



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Universidad de Las Palmas de Gran Canaria
Escuela de Ingeniería Informática
Ingeniería Informática

Proyecto de Fin de Carrera

**Aplicación para dispositivos
móviles de consulta para
profesorado y gestión para
proponentes de títulos propios de
la ULPGC**

Autor: Ubay González Ramírez
Tutor: Francisco J. Santana Pérez
Noviembre de 2013

Proyecto fin de carrera de la Escuela de Ingeniería Informática de la Universidad de
Las Palmas de Gran Canaria, presentado por el alumno:

UBAY GONZÁLEZ RAMÍREZ

Título del Proyecto: Aplicación para dispositivos móviles de consulta para profesorado
y gestión para proponentes de títulos propios de la ULPGC

Tutor:

- Francisco J. Santana Pérez

DEDICATORIA

A mi familia

AGRADECIMIENTOS

Agradezco en primer lugar a mi Tutor, Francisco J. Santana Pérez, por su apoyo y motivación a lo largo del desarrollo de este proyecto.

A la comunidad de software libre, que me ha proporcionado las herramientas necesarias para la realización del proyecto de manera altruista.

A la empresa Banot, responsable de la gestión de los datos, por su ayuda a la hora de estudiar el dominio del proyecto, así como la realización de cambios a medida en la aplicación web que gestiona los datos.

Por último, agradecer a mis padres y más cercanos amigos por su paciencia y apoyo constante e incondicional.

ÍNDICE

1	Introducción.....	7
2	Estado del Arte.....	10
3	Objetivos.....	35
3.1	Docentes.	35
3.2	Proponentes/Gestores.	35
3.3	Administrador.	35
4	Recursos Necesarios.....	37
4.1	Ubuntu.....	37
4.2	Java.....	37
4.3	Eclipse.....	37
4.4	ADT.....	37
4.5	Ruby on rails.....	38
4.6	Samsung.....	38
4.7	Android.....	38
5	Análisis, Diseño y Desarrollo.....	39
5.1	Análisis.....	39
5.1.1	Extracción de requisitos de usuario.....	39
5.2	Diseño.....	63
5.3	Desarrollo.....	67
5.3.1	Metodología.....	67
5.4	Patrón singleton.....	71
5.5	Interfaz de usuario.....	86
5.6	Diferencias de layout en versión tableta.....	98
6	Temporización.....	104
7	Prueba y Mantenimiento.....	110
7.1	Prueba.....	110
7.2	Mantenimiento.....	110
8	Análisis de Costos.....	111
8.1	Recursos materiales.....	111
8.2	Recursos humanos.....	111
9	Conclusiones.....	113

10	Trabajos Futuros.....	115
11	Bibliografía.....	117
12	Anexo.....	118
12.1	Código Java/Android.....	118
12.2	Código XML/Layouts/Strings.....	245
12.3	Parche Servidor Ruby on Rails.....	374
12.4	Manual de Usuario de la aplicación.....	381

1 Introducción

Desde 2001 y con modificación en 2007, mediante las Leyes Orgánicas de Universidades 6/2001 de 21 de diciembre y 4/2007 de 12 de abril, se establece el mecanismo y regulación de un nuevo tipo de enseñanzas al margen de las enseñanzas conducentes a la obtención de títulos oficiales. Estas nuevas enseñanzas se regulan en la ULPGC en el artículo 114 de sus estatutos, estableciendo que se podrán proponer e impartir si se estima necesario para el interés y las necesidades locales, permitiendo emitir titulaciones y diplomas propios, así como implantar enseñanzas de formación a lo largo de toda la vida.

Con el objetivo de la adecuación de estas enseñanzas al “Espacio Europeo de Educación Superior” se establecen nuevas exigencias dirigidas a la introducción de mecanismos de evaluación de calidad. El procedimiento para la presentación de propuestas, aprobación por parte de los distintos órganos responsables de enseñanzas y los requisitos para la aprobación e impartición de las titulaciones se encuentran recogidos en el Reglamento de Títulos propios, Formación continua y Extensión universitaria de la Universidad de Las Palmas de Gran Canaria.

En el previamente mencionado reglamento de la ULPGC se establecen las directivas para la creación del modelo para la presentación de diferentes modalidades de enseñanza de distinta carga lectiva, así como la información detallada que debe incluir la propuesta presentada, tales como las memorias académicas, económicas y memorias finales.

La iniciativa de propuesta para la implantación de todas las enseñanzas propias corresponderá al equipo de gobierno de la Universidad, a sus centros docentes, departamentos, institutos universitarios de investigación, centros de investigación, Centro de Formación continua, profesorado o cualquier entidad pública o privada que demande este tipo de formación.

En función de los objetivos que persigan, del colectivo al que vayan dirigidas y de las titulaciones o certificaciones a que den lugar, la Universidad de Las Palmas de Gran Canaria podrá organizar enseñanzas conducentes a títulos no oficiales de los siguientes tipos:

- Enseñanzas propias básicas, que corresponderán a un núcleo de formación inicial específico (estructura equivalente a un nivel de pregrado) y cuyos estudios tendrán una duración total de 120 ECTS, estructurados en dos cursos de 60 ECTS cada uno.
- Enseñanzas propias superiores (estructura equivalente a un nivel de grado), que no estén registradas en el Registro de universidades, centros y títulos (RUCT) como títulos oficiales de validez en todo el territorio nacional, que corresponderán a estudios de especialización científica o de formación profesional avanzada y cuyos estudios tendrán una duración de 240 ECTS, estructurados en cuatro cursos de 60 ECTS cada uno. Los requisitos de acceso son estar en posesión de los requisitos necesarios para el ingreso en las enseñanzas conducentes a títulos oficiales de grado. Cada titulación puede establecer requisitos específicos. Se obtendrá un título denominado “Título Superior de Estudios Universitarios”, adscrito a una rama de conocimiento específica.
- Maestría Universitaria, cuyos estudios tendrán una duración variable entre 60 y 120 ECTS,

programándose en uno o dos cursos académicos. Esta enseñanza estará dirigida a personas en posesión de títulos universitarios oficiales, títulos propios de enseñanzas universitarias superiores de la Universidad de Las Palmas de Gran Canaria u otros equivalentes de otras universidades.

- Experto Universitario, cuyos estudios tendrán una duración variable entre 18 y 42 ECTS, programados en un solo curso académico. Podrán acceder a estas enseñanzas los titulados conforme a sistemas educativos ajenos al Espacio Europeo de Educación Superior sin necesidad de homologación de sus títulos, previa comprobación por la Comisión de Títulos Oficiales y Propios de la Universidad de Las Palmas de Gran Canaria de que aquellos acrediten un nivel de formación equivalente a los correspondientes títulos universitarios oficiales españoles y de que estos faculten en el país expedidor para el acceso a enseñanzas de postgrado.

Cada propuesta de nuevo título propio ha de estar asociada a un Centro docente de la Universidad de Las Palmas de Gran Canaria. Se tendrán en consideración como Centros docentes al Centro de Formación Continua, la Plataforma de Teleformación, las Facultades y Escuelas, así como los Institutos Universitarios de Investigación, en el caso de que se trate de un título propio con perfil I+D+i.

Para facilitar la gestión académica del título o diploma, cada enseñanza deberá estar adscrita a una rama de conocimiento de entre las siguientes mencionadas:

- Arte y Humanidades
- Ciencias
- Ciencias de la Salud
- Ciencias Sociales y Jurídicas
- Ingeniería y Arquitectura

La aprobación o renovación de las propuestas de títulos propios será competencia del Consejo de Gobierno, previo informe de la Comisión de Títulos Oficiales y Propios. Deberá ser presentada según el modelo de formato que establece la Comisión de Títulos Oficiales y Propios, ante el Vicerrectorado de Títulos y Doctorado. Con la solicitud se acompañará una memoria académica y una memoria económica.

Para facilitar la presentación de propuestas y en contraposición con la entrega de la documentación en formato físico, para el curso académico 2011/2012 se pone en funcionamiento una plataforma web que permite introducir la información detallada necesaria para la aprobación de la propuesta por parte de los órganos competentes, así como la verificación de la mayoría de los requisitos detallados en el Reglamento de Títulos Propios de la ULPGC. Entre estos requisitos se encuentra la notificación y aceptación de docencia por parte de cada uno de los docentes propuestos para un título propio.

Actualmente, el uso de las plataformas móviles se encuentra ampliamente difundido y al alza. Existe un gran número de usuarios con un rango de dependencia variable con respecto a dispositivos móviles como fuente de información. El aprovechamiento de este tipo de dispositivos para la visualización y gestión de contenido ya existente a través de Web se ve limitado por diversos factores

(tamaño de pantalla, accesibilidad, etc).

Debido a las características de funcionamiento de la plataforma de gestión, su visualización en dispositivos móviles se encuentra severamente limitada, debido a que no está preparada para ser utilizada en otro dispositivo que no sea un ordenador personal de sobremesa.

El presente proyecto presenta una aplicación, dirigida a docentes, gestores o administrador de propuestas que permite visualizar propuestas y realizar diversas acciones sobre éstas, necesarias para la verificación y proceso de aprobación por parte de las diferentes comisiones ejecutivas y de control de calidad de la ULPGC.

2 Estado del Arte

Durante el periodo previo a la actual aplicación web, el proceso de verificación durante el periodo establecido para la presentación de propuestas requería la dedicación exclusiva de todo el personal administrativo asignado al VTD durante 7-10 días, así como un seguimiento continuo durante meses hasta el fin de los periodos de subsanación. Asimismo, requería la emisión continua de correos personalizados cuyo contenido informaba al proponente de los errores, omisiones u otras consideraciones con respecto a la propuesta y su ajuste al reglamento vigente. Además, para cumplir con el requisito de que todos y cada uno de los docentes propuestos, ya sean profesores de la ULPGC, de otra universidad o externos, confirmaran su docencia en el título propuesto y en el caso de los profesores externos, adjuntaran una fotocopia del título de enseñanza superior que les cualifica para impartir docencia en los títulos propios. La logística física y productividad del personal del VTD se veía seriamente mermada durante el periodo de presentación y subsanación de propuestas para responder a un proceso que puede calificarse de casi completamente mecánico.

El procedimiento completo a modelar se compone de las siguientes etapas:

1. La iniciativa de la propuesta del título propio puede proceder del equipo de gobierno de la Universidad, sus centros (Facultades o Escuelas), departamentos, institutos universitarios de investigación, Centro de Formación Continua, profesorado o de cualquier entidad pública o privada.
2. La propuesta de título propio debe adscribirse a una rama de conocimiento y a un centro docente. En el caso de que la propuesta de título proceda de entidades públicas o privadas ajenas a la universidad, el Vicerrectorado con competencias en el área lo adscribirá a la rama de conocimiento y lo asignará al centro docente que estime más acorde con sus contenidos y objetivos.
3. Una vez que el centro docente recibe la propuesta de un título propio, su órgano de gobierno nombrará una comisión para la elaboración y desarrollo del título. La comisión estará constituida, al menos, por el director del centro docente al que se adscribe el título, el director académico propuesto en el título y, al menos, un miembro más del personal docente a tiempo completo de la Universidad de Las Palmas de Gran Canaria, que se prevea que vaya a impartir docencia en dicho título. La constitución de la comisión de elaboración y desarrollo del título deberá ser informada al Vicerrector con competencias en el área.

Serán funciones de la comisión para la elaboración del título propio:

- Diseñar el título propio, que se remitirá a la comisión de asesoramiento docente (CAD) de títulos propios del centro docente correspondiente. En el caso de que esta comisión no exista, se remitirá a la Junta de Centro que desempeñará la función de la CAD.
- Remitir el borrador de la propuesta de título a la Junta de centro para su aprobación.

4. El centro docente remitirá la propuesta de título propio al Vicerrectorado de Ordenación Académica y Espacio Europeo de Educación superior.
5. La convocatoria para la implantación de títulos propios nuevos permanece abierta durante todo el año. Los centros docentes deberán remitir sus propuestas antes del 31 de Enero del curso académico anterior al que se pretenda comenzar las enseñanzas.
6. El Vicerrectorado de Ordenación Académica y Espacio Europeo de Educación Superior remitirá copia de la propuesta al Vicerrectorado de Calidad e Innovación Educativa para la obtención del informe sobre el Sistema de Garantía Interna de Calidad contenido en ella. Del mismo modo, remitirá una copia de la propuesta a una agencia de evaluación de la calidad universitaria externa a la Universidad de Las Palmas de Gran Canaria, para la evaluación de la calidad del título.
7. Tanto el Vicerrectorado de Calidad e Innovación Educativa como la agencia de evaluación de la calidad elaborarán un informe que se adjuntará a la propuesta del título.
8. Los informes remitidos por el Vicerrectorado de Calidad e Innovación Educativa y por la agencia de evaluación de la calidad universitaria externa serán presentados junto con la propuesta del título a la Comisión de Títulos Oficiales y Propios delegada del Consejo de Gobierno de la Universidad de Las Palmas de Gran Canaria, para su estudio y valoración. Dicha comisión elaborará un informe que remitirá al Consejo de Gobierno.
9. El Consejo de Gobierno de la Universidad de Las Palmas de Gran Canaria decidirá la implantación o no del título propuesto.
10. En el caso de que la propuesta de título propio sea aprobada por el Consejo de Gobierno de la Universidad de Las Palmas de Gran Canaria, el Consejo Social establecerá los precios para éste.
11. Una vez que el título propio haya sido aprobado, se procederá a su publicación e implantación.
12. La Universidad de Las Palmas de Gran Canaria solicitará la inclusión del título propio en el Registro de Títulos Centros y Universidades (RTCU).

Debido a la demanda constante por parte de docentes y alumnos de este tipo de titulación, se establece un procedimiento de apertura de periodo de presentación de propuestas sobre la plataforma web, con el objetivo de minimizar el tiempo requerido por parte del personal asignado al Vicerrectorado de Títulos y Doctorado (VTD) para la verificación de la corrección del material presentado.

Esta plataforma fue desarrollada para permitir el acceso a los proponentes de títulos propios con el objeto de crear las propuestas dentro del plazo establecido. La plataforma se ocupa de verificar gran parte de los requisitos declarados en el reglamento. Asimismo, permite el establecimiento de otros dos periodos adicionales de subsanación, durante los cuales la creación de nuevas propuestas no estará permitida. Durante los periodos de subsanación, únicamente se podrá acceder a modificar las propuestas presentadas y corregir los errores que han llevado a la propuesta a no encontrarse en estado 'Enviado', que representa la verificación automatizada de la aplicación sobre los datos introducidos.

Adicionalmente, la plataforma permite da acceso a los docentes propuestos en los títulos

propios, para cumplir otro de los requisitos presentes en la normativa vigente al respecto. Cada profesor ha de confirmar en primer lugar que quiere ser añadido a la base de datos de profesores de la aplicación y en segundo lugar, aceptar la docencia de cada una de las propuestas de título en las que haya sido añadido como profesor.

Actualmente, con la plataforma web en funcionamiento, el procedimiento a seguir para una entidad o docente de la ULPGC que quiera ser proponente de un nuevo título propio es en primer lugar ponerse en contacto con el director/decano del centro al que se pretende adscribir la propuesta. Cada director tiene un superusuario asignado, con capacidad para crear nuevos usuarios proponentes en los que delegar la carga de la información.

Una vez adquirido el usuario proponente, se procederá a seleccionar el rol de gestor para a continuación hacer login en la aplicación.



Imagen 1: Seleccionar Perfil

Después de haber introducido el nombre de usuario y la contraseña para ser autenticado, se tiene acceso al listado de propuestas previamente presentadas, con enlaces para acceder a la información detallada de cada una de ellas.

Lista de títulos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

BÚSQUEDA DE TÍTULOS

Bienvenido Isabel Luján

+ Crear un Título propio nuevo Exportar a Excel

Título propio	Curso	Código	Estado	Errores
Experto universitario en Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	2012/2013	TP3-1-059-02	Subsanacion	Ninguno
Experto universitario en Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	2013/2014	TP3-1-059-03	Enviado	Ninguno
Experto universitario en Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	2011/2012	TP3-1-059-01	Enviado	Ninguno

Imagen 2: Búsqueda de Títulos

Lista de profesores

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

BÚSQUEDA DE PROFESORES

Añadir profesor: + ULPGC + otra universidad + externo Exportar a Excel

Nombre	DNI	Tipo	Estado	Horas	
amadeo Jensana Tenahashi	[Redacted]	Externo	Pendiente de confirmación	0.0	Ver
guillermo Martínez Taberner	[Redacted]	Externo	Aprobado	0.0	Ver
Irene Bello Quintana	[Redacted]	Externo	Pendiente de confirmación	0.0	Ver
José Déniz Espinós	[Redacted]	Otra universidad	Aprobado	0.0	Ver
José Ángel Sotillo Lorenzo	[Redacted]	Otra universidad	Aprobado	0.0	Ver
luís Norberto González Alonso	[Redacted]	Otra universidad	Pendiente de confirmación	0.0	Ver

Imagen 3: Lista de Profesores

Asimismo, se tiene acceso en la pestaña superior a la sección 'Profesores', donde se listan los docentes ya propuestos para algún título propio previo. En esta pantalla, el proponente también tiene la posibilidad de proponer nuevos docentes para posteriormente ser añadidos a una propuesta de título

propio concreta.

El primer paso para la creación de una nueva propuesta de título propio comienza en la pantalla principal, pulsando el enlace con la etiqueta '+ *Crear un Título propio nuevo*'. A continuación se solicitará la información mínima necesaria para la formación de un borrador de propuesta que quedará pendiente de rellenar.

La información requerida para crear un borrador de propuesta se compone de los siguientes campos obligatorios.

Datos generales del título

- Tipo de Titulación
 - Experto Universitario
 - Maestría Universitaria
 - Diploma Profesional
 - Enseñanza Propia Básica
 - Enseñanza Propia Superior
- Denominación del Título
- Modalidad
 - Presencial
 - Semi-presencial
 - No-presencial
- Dirigido a
 - Docentes
 - Público en general
 - Personal de administración y servicios
- Competencias
- Proponente
- Director Académico
- Unidades participantes
- Rama de conocimiento
 - Arte y Humanidades

- Ciencias
- Ciencias de la Salud
- Ciencias Sociales y Jurídicas
- Ingeniería y Arquitectura
- Número de plazas mínimas
- Número de plazas máximas
- Periodo lectivo
- Número mínimo de créditos de matrícula con periodo lectivo
- Impartición fuera de las Islas Canarias
- Fecha de evaluación por ACECAU

Datos del centro

- Centro al que se adscribe el título
 - Lugar de impartición
 - Sede de la organización y dirección postal

Datos del responsable económico

- Datos del gestor económico

Una vez rellenada esta información, se pulsará el botón enviar y la petición quedará registrada en la aplicación. El nuevo título se mostrará en el listado principal de títulos y será accesible para su edición.

El siguiente paso será completar la información básica de la propuesta, que consistirá en determinar la composición de la comisión académica de la propuesta. La comisión estará integrada por el director académico, el proponente del título, un profesor de la ULPGC propuesto en el título y un representante del centro adscrito.

The screenshot shows a web browser window with the URL https://gestion.titulospropios.ulpgc.es/university_specific_degrees/209/general_information/edit. The page title is "Información general". The header includes the logo of the "UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA Vicerrectorado de Títulos y Doctorado" and a navigation menu with "Titulos propios", "Profesores", and "Ayuda". A user email "u.gonzalez@gmail.com" and a "Salir" button are also visible.

The main content area is titled "COMISIÓN ACADÉMICA" and contains the text "La comisión académica está formada por:". Below this, there are four input fields:

- El director académico:** A dropdown menu with a blurred selection.
- El proponente del título:** A text field containing "Aula Manuel Alemán. Isabel Luján Henríquez".
- Un profesor del título (de la ULPGC):** A dropdown menu with "Jesus Ariel Aleman Falcon" selected.
- Un representante del centro:** A text field containing "Daniel Barreto González".

An "Actualizar" button is located at the bottom right of the form area. A circular seal of the University of Las Palmas de Gran Canaria is visible in the bottom right corner of the page.

Imagen 4: Comisión Académica

Información General

A continuación, el proponente procederá a insertar la información correspondiente a la memoria académica, que se compondrá de:

- Información general
- Asignaturas
- Profesores

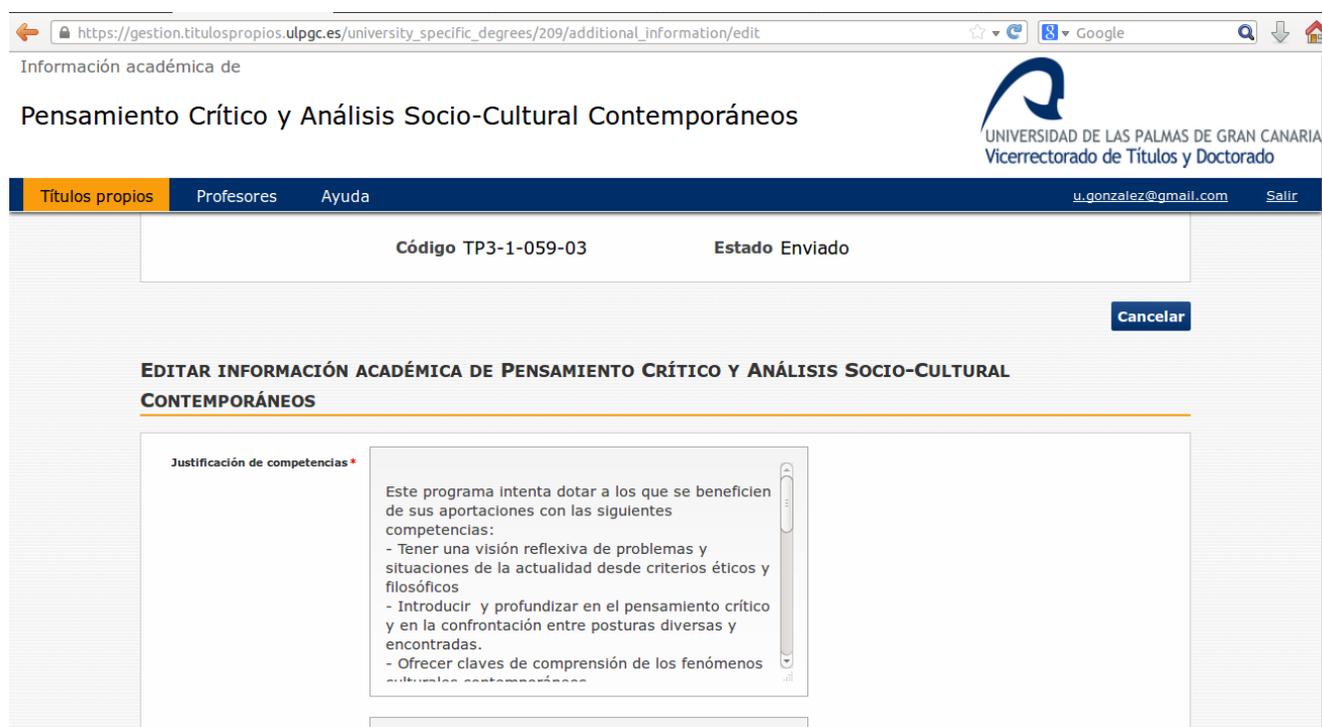


Imagen 5: Información General

La sección de información general se compone de los siguientes campos:

- Justificación de competencias
- Acceso y admisión de estudiantes
- Requisitos, materiales y servicios
- Resultados previstos
- Plan de extinción de títulos propios
- Fecha de comienzo de las enseñanzas
- Fecha de finalización de las enseñanzas

Asignaturas

En esta sección se muestra la lista de asignaturas ya agregadas a la propuesta, así como el recuento de horas de docencia que quedan pendientes por ser asignadas a un profesor de la titulación. En la parte superior de la pantalla de listado de asignaturas se encuentra el enlace '+ Añadir Asignatura', destinado a agregar una nueva unidad.

Asignaturas para

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda [u.gonzalez@gmail.com](#) [Salir](#)

Código TP3-1-059-03 Estado Enviado [Volver](#)

ASIGNATURAS + Añadir asignatura

Nombre	Tipo	Módulo	Curso/Sem.	Horas/Créditos	Horas sin asignar	Profesores	
De la filosofía del sujeto a la mística. Una aproximación a Tugendhat	Obligatoria		1 / 1	50.0 / 2.0	Completo	Isabel Pascua Febles, Pedro Sánchez Limiñana	Ver
Deseo, hiperconsumo y publicidad	Obligatoria		1 / 1	50.0 / 2.0	Completo	Tomás Acosta Santana, Isabel Pascua Febles	Ver
Después de Dios: lecturas del cristianismo en la posmodernidad	Obligatoria		1 / 1	50.0 / 2.0	Completo	Isabel Pascua Febles, Juan Francisco Comendador	Ver
El debate sobre las clases sociales	Obligatoria		1 / 1	25.0 / 1.0	Completo	Germán López Florido	Ver

Imagen 6: Listado de Asignaturas

Asignaturas para

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

u.gonzalez@gmail.com Salir

Neurociencia y cultura	Obligatoria	1 / 1	25.0 / 1.0	Completo	Manuel Rodríguez Díaz	Ver
Repensar la justicia, cambiar la política. Leyendo a Derrida	Obligatoria	1 / 1	50.0 / 2.0	Completo	Daniel Barreto González	Ver
Responsabilidad social y ética de la economía	Obligatoria	1 / 1	25.0 / 1.0	Completo	Isabel Pascua Febles, Daniel Pérez Estevez	Ver

Suma de créditos de todas las asignaturas: 20.0

Créditos pendientes de asignar a profesores: 0.0



Imagen 7: Resumen de docencia asignada a asignaturas

A la hora de agregar una nueva asignatura, la plataforma requiere una descripción completa de la misma.

Asignatura de

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

Código TP3-1-059-03 Estado Enviado

AÑADIR ASIGNATURA Cancelar

Nombre *

Módulo

Rama de conocimiento * Seleccione la rama de conocimiento

Real Decreto 1393/2007 de 29 de Octubre. BOE 260 de 30 de Octubre de 2007.

Tipo

Modalidad *

Competencias

Imagen 8: Añadir Asignatura

La información completa que se solicita en este punto es la siguiente:

- Nombre
- Rama de conocimiento
- Tipo
 - Básica
 - Obligatoria
 - Optativa
 - Prácticas Externas
 - Trabajo de Fin de Grado
 - Mixto
- Modalidad
 - Presencial
 - No-presencial
- Competencias
- Actividades Formativas

- Descripción de los Contenidos
- Sistema de Evaluación
- Curso
- Semestre
- Fecha de Comienzo
- Fecha de Finalización

Llegados a este punto, se ha de introducir el reparto de horas que presenta la asignatura, puesto que dependiendo de dicho reparto, se determina el número de ECTS de la asignatura, así como las horas de docencia que han de ser asignadas a los profesores posteriormente.

Asignatura de

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

HORAS

Clases teóricas	<input type="text" value="0.0"/>	Sesiones a través de las cuáles se imparten los conocimientos teóricos de las enseñanzas
Clases prácticas	<input type="text" value="0.0"/>	Sesiones a través de las cuáles se imparten los conocimientos prácticos de las enseñanzas
Actividades docentes complementarias	<input type="text" value="0.0"/>	Seminarios, trabajos (distintos del trabajo final), conferencias, charlas, visitas y otras actividades docentes complementarias
Prácticas en empresas/instituciones	<input type="text" value="0.0"/>	Prácticas en empresas u otras instituciones
Pruebas de evaluación	<input type="text" value="0.0"/>	

Imagen 9: Reparto de Horas

El reparto de horas se compone de las siguientes secciones:

- Clases teóricas
- Clases prácticas
- Actividades docentes complementarias
- Prácticas en empresas/instituciones

- Prueba de evaluación
- Trabajo final
- Horas dedicadas al estudio

Profesores

Suponiendo que el proponente ya haya agregado en la pestaña general de profesores anteriormente a todos los docentes que desea añadir a la propuesta, o en caso alternativo, estos profesores ya estaba agregados de ediciones anteriores, debe comenzar a proponerlos para el título propio concreto. Esto se realiza accediendo a través de la sección 'Profesores'. Cada nuevo profesor podrá ser agregado simplemente escribiendo el DNI.

Profesores para

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

Daniel Barreto González	Otra universidad	1	1.0 / 25.0	Confirmada	Eliminar
Tomás Acosta Santana	Externo	1	1.0 / 25.0	Confirmada	Eliminar
Antonio Aizpuru Rosado	Externo	1	0.5 / 12.5	Confirmada	Eliminar
Daniel Pérez Estevez	Externo	1	0.5 / 12.5	Confirmada	Eliminar
Juan Francisco Comendador	Externo	1	1.0 / 25.0	Confirmada	Eliminar
DNI *		<input type="text"/>			
Asignar profesor					

Imagen 10: Añadir profesores a propuesta

En esta última imagen se observa cómo una de las columnas declara en qué estado se encuentra el profesor propuesto para el título. '*Confirmado*' o '*Pendiente de confirmar*'. En el caso de que se encuentre '*Pendiente de confirmar*', la aplicación proporciona un enlace para notificar via correo electrónico al docente para que acceda a la aplicación y confirme o deniegue su participación en el título.

En este momento, el proponente deberá volver a la sección 'Asignaturas', acceder a cada una de las asignaturas para asignar a los profesores encargados de impartirla. Asimismo, se introduce la cantidad de horas que se le asignan a cada docente en cada asignatura, hasta que la cuenta llegue a cero. De igual manera, cada asignatura deberá tener un profesor responsable, que deberá ser perteneciente al colectivo de profesores de la ULPGC.

The screenshot shows a web interface for assigning professors to a subject. The browser address bar indicates the URL: https://gestion.titulospropios.ulpgc.es/university_specific_degrees/209/subjects/2550. The page title is 'Información general' and the logo of the 'UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA Vicerrectorado de Títulos y Doctorado' is visible. A navigation bar includes 'Títulos propios', 'Profesores', and 'Ayuda'. The main content area is titled 'PROFESORES ASIGNADOS' and contains a summary box for 'Horas pendientes de asignar' and 'Horas Completo', a table of assigned professors, and a form to add a new professor.

Nombre	Tipo	Créditos/Horas	Estado	Acciones
Isabel Pascua Febles	ULPGC	0.0 / 0.0	Aprobado (responsable)	Eliminar
Pedro Sánchez Limiñana	Externo	1.0 / 25.0	Aprobado	Eliminar

Below the table, there is a form to add a new professor:

Profesor:

Horas:

Asignar profesor

Imagen 11: Añadir profesor a asignatura

Memoria económica

Una vez completada la información perteneciente a la memoria académica, se procederá a rellenar la memoria económica con los datos de ingresos y gastos del título. Es imprescindible que la propuesta tenga un balance positivo o cero para que sea aceptada a la hora de ser evaluada.

En primer lugar, se determinan los ingresos de la propuesta mediante los conceptos:

- Matrículas
 - Estudiantes con beca
 - Estudiantes sin beca
- Otros
 - Ingresos directos
 - Subvenciones

Memoria económica de

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

Código TP3-1-059-03 Estado Enviado

INGRESOS

DERIVADOS DE LAS MATRÍCULAS [Editar](#) [Volver](#)

	Estudiantes con beca	Importe matrícula	Total
3	300,00 €		900,00 €
	Estudiantes sin beca	Importe matrícula	Total
12	600,00 €		7.200,00 €
	Total		8.100,00 €

DIRECTOS Y DE SUBVENCIONES [Editar](#)

	Subvenciones	Total
Áula Manuel Alemán. Vicerrectorado de Cultura		900,00 €
	Total	900,00 €

Imagen 12: Ingresos

Seguidamente, se especifican los gastos relacionados con la propuesta. La partida de gastos se compone de múltiples conceptos que se detallan a continuación. En la sección profesorado, se verificará que las horas que figuran como gastos correspondan con las horas de docente especificadas en la memoria académica. Por lo tanto, los conceptos se verán especificados como cantidad de horas y coste por hora.

- Profesorado
 - ULPGC
 - Clases teóricas
 - Clases prácticas
 - Seminarios
 - Pruebas de evaluación
 - Tutorías presenciales y de trabajo final
 - Tribunales para la valoración de trabajos finales
 - Directores

- Externos
 - Clases teóricas
 - Clases prácticas
 - Seminarios
 - Pruebas de evaluación
 - Tutorías presenciales y de trabajo final
 - Tribunales para la valoración de trabajos finales
 - Otros
- Apoyo administrativo
 - Personal administrativo de apoyo al título
 - Becarios de colaboración
 - Becarios específicos
 - Otro tipo de personal
 - Elaboración de memoria de presentación
 - Memoria final
 - Punto de información
 - Servicios de empresas de apoyo
 - Otros gastos de administración
- Desplazamiento y estancia
 - Locomoción de profesorado desplazado
 - Alojamiento de profesorado desplazado
 - Manutención de profesorado desplazado
 - Ayudas para estudiantes en prácticas
 - Otros
- Material fungible
 - Material de oficina
 - Material de laboratorio
 - Material informático no inventariable
 - Reprografía

- Otros
- Bibliografía y material inventariable
 - Bibliografía
 - Otros
- Publicidad
 - Material impreso
 - Publicidad en prensa u otros medios
- Otros
 - Inauguración y/o clausura
 - Traducción simultánea o sucesiva
 - Certificados
 - Campus virtual
 - Evaluación del título por agencia externa
 - Seguro de accidentes de estudiantes
 - Otros

Por último, el sistema agregará a la partida de gastos una serie de costes fijos relacionados con el uso de instalaciones universitarias y pago por gestión económica

- Compensación a la ULPGC (5%)
- Compensación al centro adscrito (5%)
- Compensación al gestor económico (5%)

Memoria económica de

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

GASTOS

PROFESORADO ULPGC Editar

	Horas	Coste horas	Total
Horas de clases teóricas	80.0	10,00 €	800,00 €
Horas de clases prácticas	40.0	10,00 €	400,00 €
Horas de seminarios	90.0	10,00 €	900,00 €
Pruebas de evaluación	40.0	10,00 €	400,00 €
	Proyectos	Coste por proyecto	Total
Tutorías presenciales y de trabajos finales			0,00 €
	Tribunales	Coste por tribunal	Total
Tribunales para valoración de trabajos finales			0,00 €
	Número		Total
Directores			
	Coste total del profesorado de la ULPGC (a)		2.500,00 €
	Beneficios para la unidad organizadora (b)		
	Total retribución bruta a profesorado de la ULPGC (a)-(b)		2.500,00 €

PROFESORADO EXTERNO A LA ULPGC Editar

Imagen 13: Gastos

Información extra

https://gestion.titulospropios.ulpgc.es/university_specific_degrees/209

Experto universitario en

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Profesores Ayuda u.gonzalez@gmail.com Salir

INFORMACIÓN EXTRA

Aún no se ha subido ningún fichero

Fichero No file selected.

Si se trata de varios ficheros, puede subirlos todos juntos mediante un archivo comprimido (.zip, .rar, etc.).

Imagen 14: Información extra

En algunos casos, el proponente necesita adjuntar una justificación de algún requisito que es incapaz de cumplir o comunicar acuerdos de prácticas externas con alguna institución o empresa. La aplicación proporciona una sección que permite remitir junto con el resto de la memoria un documento para que sea considerado de manera conjunta con el resto de la petición.

Una vez completados todos estos pasos, la aplicación puede ser enviada a evaluar automáticamente por la aplicación. La aplicación responderá con los errores que encuentre al verificar la información introducida, manteniendo la propuesta en estado '*Borrador*'. En el caso de no producirse errores, la propuesta pasaría a estado '*Enviado*', que significará que está lista para ser procesada y enviada al siguiente paso de aprobación.

The screenshot shows a web browser window with the URL https://gestion.titulospropios.ulpgc.es/university_specific_degrees/212/issues. The page header includes navigation links: 'Títulos propios', 'Profesores', 'Ayuda', and 'Salir'. The main content area displays the logo of the 'UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA Vicerrectorado de Títulos y Doctorado'. Below the logo, a box titled 'Errores de Apicultura' shows the code 'Código TP1-3-212-01' and the status 'Estado Borrador'. A button labeled 'Volver al título propio' is located below this box. Underneath, a section titled 'LISTA DE ERRORES' contains a bulleted list of errors:

- El título debe cumplir con el número de créditos ECTS establecidos por su tipo
- Todos los créditos presenciales deben estar asignados a profesores
- El 60% de los créditos ECTS tienen que ser impartidos por personal docente universitario
- El 40% de los créditos ECTS impartidos por personal docente universitario tienen que ser de personal docente de la ULPGC
- Según la memoria económica, el título no es viable
- La información académica del título no está completa

Imagen 15: Informe de errores

A continuación, se detallarán funcionalidades exclusivas para el administrador de la plataforma web.

Plazos

En esta pestaña, el administrador puede consultar cuales son los plazos vigentes de presentación de propuestas y hasta donde se extienden los dos plazos de subsanación. Asimismo, se puede editar los plazos existentes, así como crear uno nuevo. Adicionalmente, la plataforma muestra la cantidad bruta de propuestas asignadas a cada periodo lectivo.

Año	Solicitud	Primera subsanación	Segunda subsanación	Número de títulos	
2011/2012	Del 01/11/2011 al 08/11/2011	Del 28/11/2011 al 29/11/2011	Del 30/11/2011 al 01/12/2011	44	Ver Editar
2012/2013	Del 07/11/2011 al 23/03/2012	Del 24/03/2012 al 25/03/2012	Del 26/03/2012 al 25/07/2012	68	Ver Editar
2013/2014	Del 12/06/2013 al 11/07/2013	Del 12/07/2013 al 19/07/2013	Del 26/07/2013 al 18/10/2013	41	Ver Editar
2014/2015	Del 18/09/2013 al 30/09/2013	Del 01/10/2013 al 02/10/2013	Del 03/10/2013 al 04/10/2013	1	Ver Editar

Imagen 16: Ver plazos

Añadir plazo

Volver al listado

Año * Seleccione el año académico

Solicitud desde *

Solicitud hasta *

Primera subsanación desde *

Primera subsanación hasta *

Segunda subsanación desde *

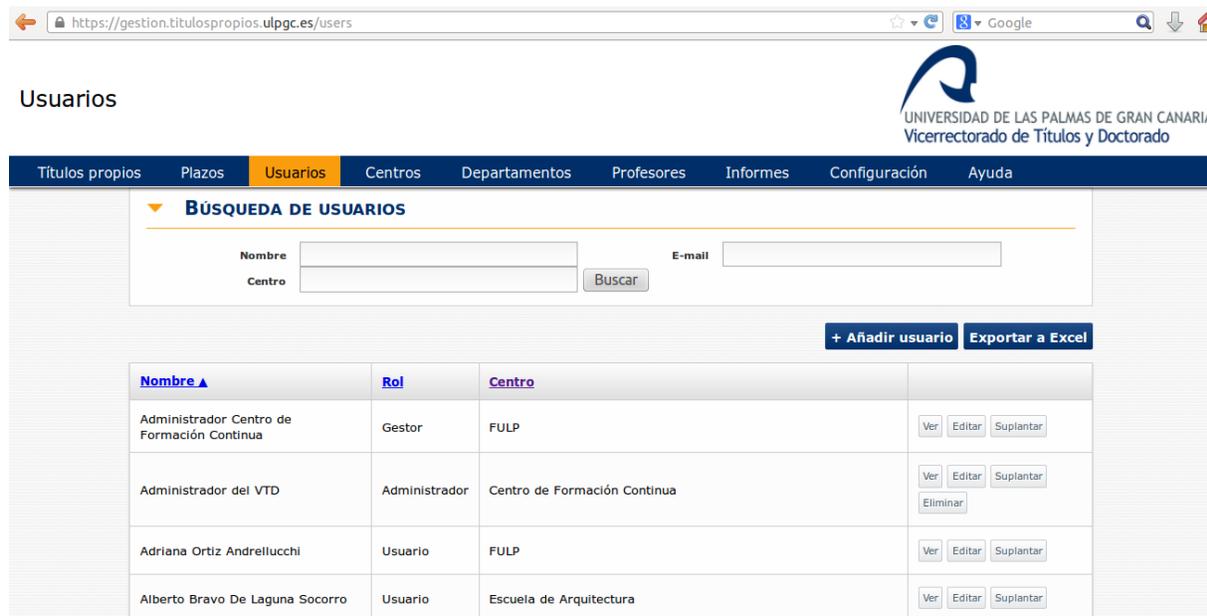
Segunda subsanación hasta *

Guardar

Imagen 17: Añadir plazo

Usuarios

En esta sección, común a los gestores(directores de centro) y administrador, se podrán crear nuevos usuarios proponentes para acceder a la aplicación. El administrador tendrá permiso en este caso para crear nuevos usuarios gestores asignados a un centro, que puedan a su vez crear nuevos usuarios proponentes.



Usuarios

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

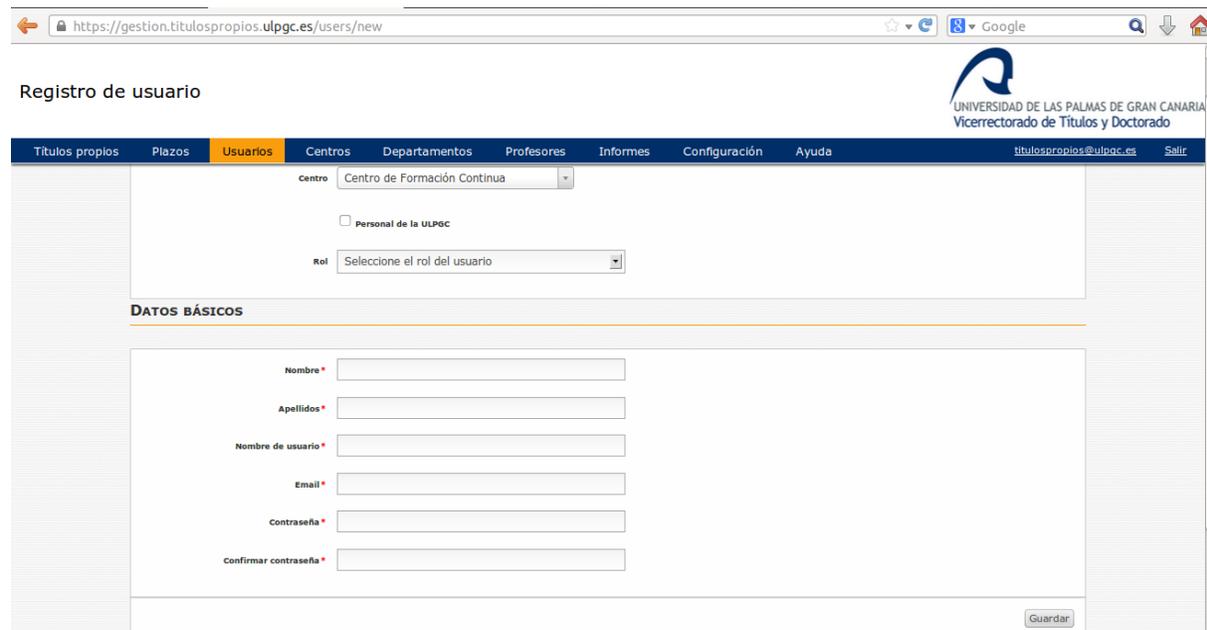
Títulos propios Plazos **Usuarios** Centros Departamentos Profesores Informes Configuración Ayuda

BÚSQUEDA DE USUARIOS

Nombre E-mail
Centro

Nombre ▲	Rol	Centro	
Administrador Centro de Formación Continua	Gestor	FULP	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Suplantar"/>
Administrador del VTD	Administrador	Centro de Formación Continua	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Suplantar"/> <input type="button" value="Eliminar"/>
Adriana Ortiz Andrellucchi	Usuario	FULP	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Suplantar"/>
Alberto Bravo De Laguna Socorro	Usuario	Escuela de Arquitectura	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Suplantar"/>

Imagen 18: Usuarios



Registro de usuario

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Plazos **Usuarios** Centros Departamentos Profesores Informes Configuración Ayuda títulospropios@ulpgc.es

Centro

Personal de la ULPGC

Rol

DATOS BÁSICOS

Nombre *

Apellidos *

Nombre de usuario *

Email *

Contraseña *

confirmar contraseña *

Imagen 19: Añadir usuario

Departamentos

La sección dedicada a departamentos se encuentra en desarrollo. Una vez implementada, su cometido será notificar a los jefes de departamentos de la carga horaria asignada para títulos propios de cada profesor propuesto. Una vez recibida la notificación, es decisión del jefe de departamento si autorizar o no la docencia de dicho profesor, basado en la posible sobrecarga de docencia.

Para gestionar la posible reorganización departamental, la plataforma permite la edición de información y la creación de nuevos departamentos.

Departamentos

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado

Títulos propios Plazos Usuarios Centros Departamentos Profesores Informes Configuración Ayuda

BÚSQUEDA DE DEPARTAMENTOS

Nombre Teléfono

Buscar

Crear nuevo departamento Exportar a Excel

Acrónimo	Nombre ▲	Director	Profesores	
daea	Análisis Económico Aplicado	Roberto	16	Ver Editar Eliminar
dact	Arte, Ciudad y Territorio	Manuel Jesus	10	Ver Editar Eliminar
dbio	Biología	Juan Luis	1	Ver Editar Eliminar
dbbf	Bioquímica y Biología Molecular, Fisiología, Genética e Inmunología	Ricardo	1	Ver Editar Eliminar

Informes

Bajo demanda del Vicerrectorado de Títulos y Doctorado, se solicita a los desarrolladores de la aplicación web que como funcionalidad sea posible extraer cierta información relevante para la presentación de propuestas en Consejo Social o Consejo de Gobierno. Extrayendo información concreta para cada etapa de aprobación, la aplicación devuelve un documento CVS. Desafortunadamente, esta funcionalidad funciona correctamente para los casos en los que un periodo contenga propuestas abandonadas en etapas previas o de prueba. Esta circunstancia hace necesario aún la extracción de determinados listados directamente a través del sistema de gestión de bases de datos.



Imagen 20: Informes

Configuración

Por último, la plataforma permite la edición de determinados parámetros de configuración que han de ser constantes durante un periodo, pero pueden ser modificados dependiendo de las consideraciones que emanen anualmente de las resoluciones del Consejo Social o la Comisión de Títulos Oficiales y Propios.

Los parámetros configurables son:

- Caducidad del código de activación (notificación a profesores)
- Ratio de compensación para el centro
- Ratio de compensación para la ULPGC
- Ratio de compensación para el administrador
- Relación de horas por crédito
- Número máximo de horas por profesor y título
- Precio de la póliza del seguro obligatorio de estudiantes
- Coste máximo asignado por título, director y periodo

- Coste de la evaluación por parte de una entidad externa (ACECAU)
- URL del manual de ayuda de la aplicación

The screenshot shows a web browser window with the URL `https://gestion.titulospropios.ulpgc.es/settings/edit`. The page title is "Editar parámetros de configuración". The header includes the logo of the "UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA Vicerrectorado de Títulos y Doctorado" and a navigation menu with items: "Títulos propios", "Plazos", "Usuarios", "Centros", "Departamentos", "Profesores", "Informes", "Configuración" (highlighted), and "Ayuda". The main content area contains several configuration fields:

Parámetro	Valor
Caducidad del código de activación *	604800
Ratio de compensación para el centro *	0.05
Ratio de compensación para la ULPGC *	0.05
Ratio de compensación para el administrador *	0.05
Relación de horas por crédito *	25
Número máximo de horas por profesor y título *	80
Precio de la póliza del seguro *	15
Coste máximo asignado por título, director y periodo *	2
Coste por defecto asignado a externos *	550
Uri de ayuda *	<code>http://titulospropios.ulpgc.es/index.php?option=com_docman&Itemid=31</code>

A "Guardar" button is located at the bottom right of the form.

Imagen 21: Configuración

Otras Universidades

Se ha realizado una tarea de investigación para averiguar si alguna otra universidad española contaba con una plataforma pública para la presentación y consulta de nuevas propuestas de títulos propios. Los resultados obtenidos se dividirán en las diferentes categorías descubiertas.

- Presentación de propuesta mediante formulario (Word, PDF) a través de **correo electrónico**:
 - Universidad Pablo de Olavide
 - Universidad de Salamanca
 - Universidad Complutense de Madrid
 - Universidad de Valencia
 - Universidad Rey Juan Carlos
 - Universidad de Extremadura
 - Universidad de Málaga
 - Universidad de Oviedo

- Presentación de propuesta mediante la plataforma **Moodle**:
 - Universidad Autónoma de Madrid

3 Objetivos

El proyecto tiene como objetivo la puesta a disposición de la información contenida en la plataforma de gestión de títulos propios de la ULPGC en un formato accesible por la mayoría de los actuales dispositivos móviles del mercado. Su desarrollo permitirá la interacción de los usuarios envueltos en el proceso con sus propuestas, para la consulta de los datos así como otras acciones destinadas a facilitar la aprobación de los nuevos títulos.

Los roles a los que deberá servir son:

3.1 *Docentes.*

El profesorado podrá ser capaz de verificar la información relacionada con la docencia en los títulos a los que ha sido propuesto, como la carga horaria, competencias, resultados previstos, etc.

3.2 *Proponentes/Gestores.*

Los responsables de la presentación de la propuesta podrán ser capaces de verificar toda la información introducida en la aplicación, como la información general del título, los datos relacionados con la memoria académica (Asignaturas, Profesorado/Carga lectiva) y la memoria económica. Asimismo, las acciones disponibles para este rol incluirán la capacidad de remitir notificaciones al profesorado pendiente de confirmar su docencia en determinado título, el envío de la propuesta a validar o descargar el borrador de la memoria del título en formato imprimible (PDF)

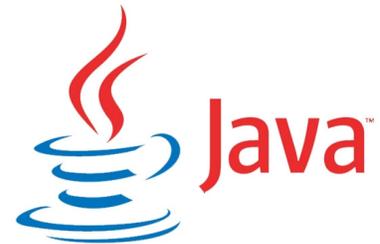
3.3 *Administrador.*

Acceso completo a la información disponible en la plataforma para el seguimiento de la totalidad de las propuestas, con el objetivo de poder consultar cualquier detalle relevante para las comisiones encargadas de la aprobación, tener la capacidad de saber en qué estado se encuentran y estimar qué volumen de éstas van a progresar en el proceso de aprobación.

Como objetivos adicionales y para responder a la necesidad de movilidad de los proponentes y docentes, se tratará de que el tráfico de datos no sea excesivo, dado el coste que se deriva de la transferencia de datos en tránsito o 'roaming'.

Para llevar a cabo los anteriores objetivos, se utilizarán numerosas tecnologías. Su aprendizaje también serán considerados como objetivos a realizar. Entre estas tecnologías que durante alguna fase del proyecto serán utilizadas se encuentran:

- **Java**



- **Plataforma Android**



- **Arquitectura: Representational State Transfer (REST)**

Despliegue de servidor con las siguientes tecnologías:

- **Ruby on Rails**



- **PostgreSQL**



4 Recursos Necesarios

Para la realización del proyecto se requiere disponer de un equipamiento informático básico: PC, impresora, etc... así como tener acceso a otras herramientas básicas e imprescindibles: editor de texto, acceso a internet, etc.

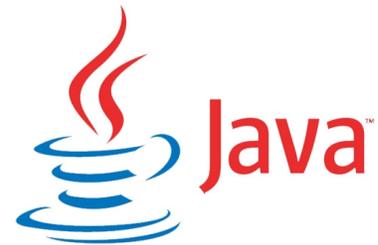
4.1 *Ubuntu*

- El sistema operativo utilizado será Ubuntu 12.04 LTS/Linux



4.2 *Java*

- El lenguaje de programación de Android necesariamente será Java.



4.3 *Eclipse*

- El entorno de desarrollo elegido (compilador, interprete, depurador) para desarrollar sobre la plataforma Android es Eclipse.



4.4 *ADT*

- Para el desarrollo específico de android se utilizará la extensión de ADT para Eclipse(Android Development Tools)



4.5 *Ruby on rails*

- Asimismo, será necesaria la puesta en funcionamiento de un servidor Rails, puesto que está desarrollada en lenguaje Ruby. El servidor local replicará la funcionalidad de la aplicación que se encuentra en producción con el objeto de ser utilizada como base de pruebas.



4.6 *Samsung*

- Con el objeto de testear en un entorno real y no sólo en el emulador incluido en el ADT proporcionado por Google, se utilizará un dispositivo Samsung Galaxy Ace 2.



4.7 *Android*

- La versión de Android del dispositivo mencionado será la 2.3.6 (Gingerbread)



El proyecto será desarrollado utilizando prácticamente en exclusiva software de carácter libre.

5 Análisis, Diseño y Desarrollo

5.1 Análisis

5.1.1 Extracción de requisitos de usuario

La etapa de análisis se extiende durante los tres años que el autor de este proyecto de fin de carrera ha estado encargado de la gestión y tramitación de la documentación de las propuestas de títulos propios. Esto ha dado al autor una clara idea de los problemas y las circunstancias especiales en las que un administrador o un proponente se encuentra habitualmente cuando trata de hacer progresar una propuesta a través del proceso completo de aprobación.

Con el objetivo de modelar e identificar las entidades de información relevantes en el sistema, se ilustran en el siguiente diagrama simplificado Entidad-Relación. Este diagrama ayuda a determinar las entidades clave en el proceso de extracción de requisitos.

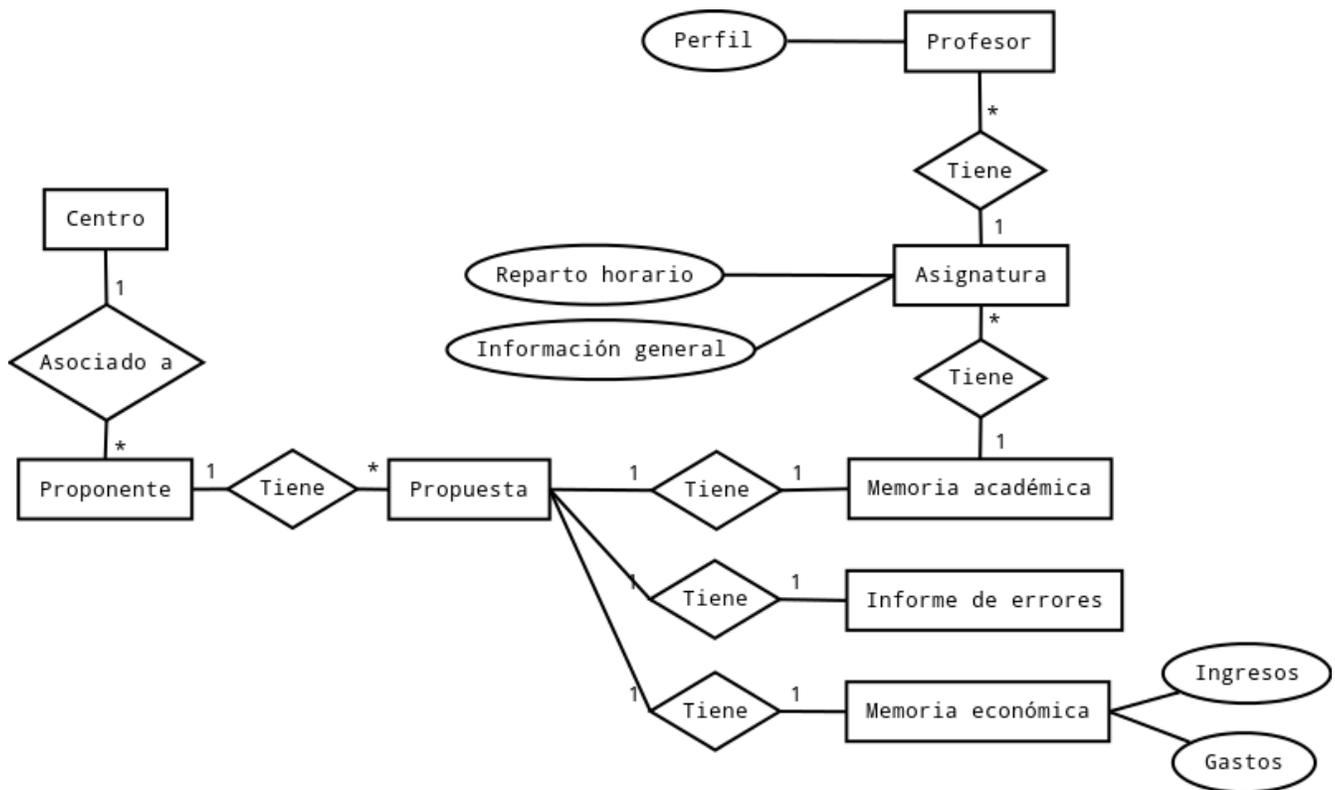


Imagen 22: Modelo Entidad-Relación

Al estudiar el modelado Entidad-Relación, se identifican claramente varias entidades principales que han de ser tratadas como tales. Estas son:

- Propuesta
- Memoria Académica
- Memoria Económica
- Asignatura
- Profesor
- Memoria Económica
- Informe de errores

El proponente se indentifica como el usuario para el modelado de los casos de uso, mientras que el centro quedaría al márgen del objetivo del presente proyecto.

Casos de uso

Con el objetivo de determinar de manera explícita lo que se espera del comportamiento de la aplicación, se especifican a continuación los casos de uso de los actores del sistema.

Nombre	Seleccionar rol
Descripción	El actor elige con qué rol va a autenticarse en la aplicación
Actores	Docentes, Proponentes
Precondiciones	El actor debe haber instalado la aplicación y ejecutarla
Flujo normal	El actor ha de pulsar sobre una de las dos opciones presentadas en pantalla para elegir el rol
Flujo alternativo	-
Postcondiciones	El rol ha sido almacenado en el sistema y se ha ejecutado la siguiente actividad

Nombre	Autenticar
Descripción	El actor, con sus credenciales, verifica su identidad y contraseña para entrar en la aplicación
Actores	Docentes, Proponentes
Precondiciones	El actor debe haber seleccionado previamente el rol con el que quiere autenticarse
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa sobre la caja de texto correspondiente al nombre de usuario y lo introduce a través del teclado en pantalla 2. El actor pulsa sobre la caja de texto correspondiente a la contraseña y la introduce a través del teclado en pantalla 3. El actor decide si pulsar la caja de opción de 'Recordar Usuario' 4. El actor pulsa sobre el botón 'Acceder'
Flujo alternativo	<ol style="list-style-type: none"> 1. El actor pulsa sobre el botón 'Acceder' puesto que la opción 'Recordar Usuario' ha rellenado el resto de los datos automáticamente
Postcondiciones	La sesión ha sido almacenada por el sistema y el usuario está autenticado

Nombre	Seleccionar tarea
Descripción	El actor elige cuál ha de ser su siguiente acción entre buscar propuestas o crear propuestas
Actores	Proponentes
Precondiciones	El actor debe haberse autenticado con el rol de proponente
Flujo normal	El actor ha de pulsar sobre una de las dos opciones presentadas en pantalla para elegir la tarea
Flujo alternativo	-
Postcondiciones	Una nueva actividad se ejecutará en pantalla

Nombre	Crear nueva propuesta
Descripción	El actor-proponente rellena los datos necesarios para la creación de una versión borrador de título propio
Actores	Proponente
Precondiciones	El periodo de presentación de solicitudes debe estar vigente
Flujo normal	<ol style="list-style-type: none"> 1. El actor-proponente pulsa sobre la caja de texto Denominación e introduce la información correspondiente 2. El actor-proponente pulsa sobre la caja de texto DNI e introduce la información correspondiente 3. El actor-proponente pulsa sobre la caja de texto Proponente e introduce la información correspondiente 4. El actor-proponente pulsa sobre la caja de texto Unidades participantes e introduce la información correspondiente 5. El actor-proponente pulsa sobre la caja de texto Plazas mínimas e introduce la información correspondiente 6. El actor-proponente pulsa sobre la caja de texto Plazas máximas e introduce la información correspondiente 7. El actor-proponente pulsa sobre la caja de texto Mínimo de créditos anuales e introduce la información correspondiente 8. El actor-proponente pulsa sobre la caja de texto Gestor económico e introduce la información correspondiente 9. El actor-proponente pulsa sobre la caja de texto Competencias e introduce la información correspondiente 10. El actor-proponente pulsa sobre el desplegable Tipo de título y selecciona la opción correspondiente 11. El actor-proponente pulsa sobre el desplegable Modalidad de enseñanza y selecciona la opción correspondiente 12. El actor-proponente pulsa sobre el desplegable Dirigido a y selecciona la opción correspondiente 13. El actor-proponente pulsa sobre el desplegable Rama de conocimiento y selecciona la opción correspondiente 14. El actor-proponente pulsa sobre el desplegable Centro adscrito y selecciona la opción correspondiente 15. El actor-proponente pulsa o no pulsa sobre la opción 'Título impartido fuera de las Islas Canarias'
Flujo alternativo	-
Postcondiciones	Si toda la información es correcta, el título quedará creado en el sistema y el actor-proponente será redirigido a la pantalla de Seleccionar tarea

Nombre	Buscar propuestas
Descripción	El actor elige una serie de parámetros de búsqueda para obtener un listado de propuestas que coincidan con su selección
Actores	Docentes-Proponentes
Precondiciones	El actor debe haberse autenticado
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa sobre el desplegable 'Tipo de título' para elegir el parámetro correspondiente 2. El actor pulsa sobre el desplegable 'Año de inicio' para elegir el parámetro correspondiente 3. El actor pulsa sobre el desplegable 'Estado de título' para elegir el parámetro correspondiente 4. El actor pulsa sobre el cuadro de texto 'Código de título' para introducir un código concreto 5. El actor pulsa sobre el cuadro de texto 'Denominación' para introducir un nombre de título 6. El actor pulsa sobre el botón 'Buscar' para obtener los resultados
Flujo alternativo	<ol style="list-style-type: none"> 1. El actor pulsa directamente sobre el botón 'Buscar' para obtener todas las propuestas de título asociadas a su cuenta de usuario
Postcondiciones	El sistema mostrará un diálogo con los resultados de búsqueda, indicándole al usuario dos opciones, volver a repetir búsqueda o iniciar una nueva actividad con el listado de propuestas que se ajustan a los criterios de búsqueda

Nombre	Seleccionar propuesta
Descripción	El actor recibe un listado de propuestas que cumplen con los parámetros de búsqueda de la actividad anterior y elige una de entre éstas para acceder a su información específica
Actores	Docentes-Proponentes
Precondiciones	El actor ha seleccionado un grupo de parámetros de búsqueda que le proporcionan un listado de propuestas
Flujo normal	<ol style="list-style-type: none"> 1. El actor elige de entre las propuestas mostradas y pulsa sobre alguna de ellas para acceder a su información
Flujo alternativo	-
Postcondiciones	El sistema tiene conocimiento de la propuesta que el actor está consultando para posterior uso

Nombre	Seleccionar sección
Descripción	Al actor se le muestran las secciones de información disponibles en cada propuesta
Actores	Docentes-Proponentes
Precondiciones	El actor ha seleccionado una propuesta concreta después de haber buscado sobre el total
Flujo normal	1. El actor elige de entre las secciones de una propuesta para determinar qué información ha de ser mostrada a continuación
Flujo alternativo	-
Postcondiciones	El sistema tiene conocimiento de la sección que el actor está solicitando

Nombre	Información general
Descripción	El actor selecciona la opción de mostrar la información general de la propuesta
Actores	Docente-Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona la opción mostrar información general de la propuesta
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla la información general de la propuesta seleccionada

Nombre	Asignaturas
Descripción	El actor selecciona la opción de mostrar las asignaturas de la propuesta
Actores	Docente-Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona la opción mostrar las asignaturas de la propuesta
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla el listado de asignaturas asignado a la propuesta concreta

Nombre	Seleccionar asignatura
Descripción	El actor selecciona una de las asignaturas en la lista de asignaturas de una propuesta solicitada previamente
Actores	Docente-Proponente
Precondiciones	El actor ha seleccionado mostrar el listado de asignaturas asignado a una propuesta concreta
Flujo normal	1. El actor selecciona una de las asignaturas del listado
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla la información detallada de la asignatura seleccionada

Nombre	Seleccionar profesorado
Descripción	El actor selecciona la opción de mostrar el profesorado de la propuesta
Actores	Docente-Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona la opción mostrar el profesorado de la propuesta
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla el listado de profesorado asignado a la propuesta concreta

Nombre	Seleccionar profesor
Descripción	El actor selecciona uno de los profesores en la lista de profesorado de una propuesta solicitada previamente
Actores	Proponente
Precondiciones	El actor ha seleccionado mostrar el listado de profesorado asignado a una propuesta concreta
Flujo normal	1. El actor selecciona uno de los profesores del listado
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla la información detallada del profesor seleccionado

Nombre	Notificar profesor
Descripción	El profesor al que se intenta notificar se encuentra en estado 'Pendiente de confirmar'
Actores	Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona el botón 'Notificar'
Flujo alternativo	-
Postcondiciones	El sistema remite un correo electrónico al profesor seleccionado notificándole de que su estado como docente se encuentra aún 'Pendiente de confirmar'

Nombre	Mostrar memoria académica
Descripción	El actor quiere visualizar la información económica de la propuesta que se encuentra explorando
Actores	Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona el botón 'Memoria académica'
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla la información económica detallada de la propuesta seleccionada

Nombre	Mostrar informe de errores
Descripción	El actor quiere visualizar el informe de errores generado por la aplicación, de la propuesta que se encuentra explorando
Actores	Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones
Flujo normal	1. El actor selecciona el botón 'Informe de errores'
Flujo alternativo	-
Postcondiciones	El sistema trae a pantalla el informe de errores de la propuesta seleccionada

Nombre	Enviar propuesta para evaluación
Descripción	Remitir la propuesta actual a la plataforma web para que sea verificada mecánicamente y comprobar si puede cambiar de estado
Actores	Proponente
Precondiciones	<ol style="list-style-type: none"> 1. El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones. 2. El periodo de subsanación se encuentra abierto
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción 'Enviar propuesta para evaluación' en el menú principal
Flujo alternativo	-
Postcondiciones	El sistema cambiará el estado de la propuesta a 'Enviado' si la propuesta pasa la verificación mecánica. En caso contrario, el sistema mostrará el informe de errores generado por la aplicación

Nombre	Descargar memoria
Descripción	El usuario se descarga en su dispositivo la memoria en formato PDF que genera la aplicación web bajo demanda
Actores	Proponente
Precondiciones	El actor ha seleccionado una propuesta concreta para explorar alguna de sus secciones.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción 'Descarga de la memoria en PDF' en el menú principal
Flujo alternativo	-
Postcondiciones	El actor recibe un documento PDF para ser almacenado y visualizado en su dispositivo móvil

RESTful web API

A continuación se describen las interfaces disponibles en la aplicación web para la obtención de documentos XML con la información solicitada a la API.

En el servidor actual de producción la BASE_URL es <http://gestion.titulospropios.ulpgc.es>

BASE_URL/university_specific_degrees.xml

En primer lugar, se describe la dirección base de la URL Esta petición a la API mostrará un array con todas las propuestas solicitadas, ya sea con criterios de búsqueda o una petición completa del listado

```
<university-specific-degrees type="array">
```

```
<university-specific-degree>
```

Las etiquetas incluidas en este documento son las siguientes:

Etiqueta	Significado	Tipo
Academic-management-units	Unidades participantes	Ristra
Administrator	Gestor económico	Ristra
Audience-id	Dirigido a: 1. Docentes 2. Público en general 3. Personal de administración y servicios	Entero
Branch-of-knowledge-id	Rama de conocimiento: 1. Arte y Humanidades 2. Ciencias 3. Ciencias de la Salud 4. Ciencias Sociales y Jurídicas 5. Ingeniería y Arquitectura	Entero
Code	Código único de la propuesta: TPX-YYY-ZZ X: Tipo de título YYY: Contador de propuestas	Ristra

	ZZ: Número de edición	
Commissioner-teacher-id	Identificador del profesor que forma parte de la comisión académica de la propuesta	Entero
Created-at	Fecha de creación de la propuesta	Fecha
Degree-type-id	Tipo de enseñanza: 1. Enseñanza propia básica 2. Enseñanza propia superior 3. Experto universitario 4. Maestría universitaria 5. Diploma profesional	Entero
Edition-number	Número de edición	Entero
Evaluated-by-acecau-at	Fecha de evaluación por parte de la ACECAU	Fecha
Expected-results	Resultados esperados	Ristra
Extinction-plan	Plan de extinción de la titulación	Ristra
Extra-information-content-type	Tipo de contenido del documento adjunto en información adicional	Ristra
Extra-information-file-name	Nombre del fichero adjunto en información adicional	Ristra
Extra-information-file-size	Tamaño del fichero adjunto en información adicional	Entero
Extra-information-updated-at	Fecha de actualización del fichero adjunto en información adicional	Fecha
Head-office	Sede de la organización	Ristra
Id	Identificador único de la propuesta	Entero
Institution-id	Identificador único del centro adscrito	Entero
Institution-representative	Representante del centro	Ristra
Justify-competences	Justificación de competencias	Ristra
Lectures-end	Finalización de las enseñanzas	Fecha
Lectures-start	Inicio de las enseñanzas	Fecha
Manager-id	Identificador único del usuario proponente	Entero
Material-resources	Recursos materiales	Ristra
Max-attendees	Número máximo de alumnos	Entero
Min-attendees	Número mínimo de alumnos	Entero
Min-credits-per-academic-period	Número mínimo de créditos a los que se debe	Entero

	matricular un alumno anualmente	
Mode-id	Modalidad de enseñanza: 1. Presencial 2. No-presencial 3. Semi-presencial	Entero
Name	Denominación de la propuesta	Ristra
Non-presential-ects	Cantidad de créditos en modalidad no presencial	Entero
Original-id	Vestigio de previa versión de desarrollo de la aplicación	No aplicable / null
Outside-canary-islands	Impartición de las enseñanzas fuera de las Islas Canarias	Booleano
Period-id	Identificador único del periodo lectivo	Entero
Proposed-by	Proponente	Ristra
State	Estado de la propuesta: borrador enviado subsanción cancelado rechazado	Ristra
Students-admission	Admisión de estudiantes	Ristra
Teaching-place	Lugar de impartición de enseñanzas	Ristra
Updated-at	Fecha de última actualización de la propuesta	Fecha
User-id	Identificador único del usuario encargado de cargar la propuesta	Entero

BASE_URL/periods.xml

En este documento se recogen los diferentes periodos de presentación de propuesta y subsanación.

<periods type="array">

<period>

Etiqueta	Significado	Tipo
Created-at	Fecha de creación del periodo	Fecha
First-rectification-from-date	Fecha de inicio de primer periodo de subsanación	Fecha
First-rectification-to-date	Fecha de finalización del primer periodo de subsanación	Fecha
Id	Identificador único del periodo	Entero
Request-from-date	Fecha de inicio de periodo de creación de nuevas propuestas	Fecha
Request-to-date	Fecha de finalización del periodo de creación de nuevas propuestas	Fecha
Second-rectification-from-date	Fecha de inicio del segundo periodo de subsanación	Fecha
Second-rectification-to-date	Fecha de finalización del segundo periodo de subsanación	Fecha
Updated-at	Fecha de la última actualización realizada sobre el periodo	Fecha
Year	Año académico correspondiente con periodo actual	Ristra

BASE_URL/ID_PROPUESTA/subjects.xml

Este documento contiene un array con las asignaturas introducidas en la propuesta que se está tratando.

<subjects type="array">

<subject>

Etiqueta	Significado	Tipo
Activities	Actividades formativas	Ristra
Assessment-system	Pruebas de evaluación	Ristra
Branch-of-knowledge-id	Rama de conocimiento: 1. Arte y Humanidades 2. Ciencias 3. Ciencias de la Salud 4. Ciencias Sociales y Jurídicas 5. Ingeniería y Arquitectura	Entero
Class-hours	Horas de teoría	Real
Competences	Competencias	Ristra
Course	Año en el que se imparte la asignatura. Puede ser mayor que uno si la propuesta contempla varios cursos académicos	Entero
Created-at	Fecha de creación de la asignatura	Fecha
Description	Descripción de los contenidos de la asignatura	Ristra
Ects-credits	ECTS de la asignatura	Real
End-date	Fecha de finalización de la asignatura	Fecha
Exams-hours	Horas de examen	Real
External-practice-hours	Horas de prácticas externas	Real
Final-work-hours	Horas de trabajo final	Real
Hours-sum	Suma completa de horas	Real
Id	Identificador único de la asignatura	Entero
Kind-id	Tipo de asignatura:	Entero

	<ul style="list-style-type: none"> 1.Normal 2.Prácticas externas 3.Trabajo de fin de título 	
Misc-activities-hours	Horas de actividades docentes complementarias	Real
Mode-id	Modalidad de enseñanza de la asignatura <ul style="list-style-type: none"> 1. Presencial 2. No-presencial 3. Semi-presencial 	Entero
Name	Denominación de la asignatura	Ristra
Practice-hours	Horas de prácticas	Real
Semester	Semestre de la asignatura	Entero
Start-date	Fecha de inicio de la asignatura	Fecha
Studying-hours	Horas dedicadas al estudio	Real
Subject-id	Vestigio de previa versión de desarrollo de la aplicación	No aplicable / null
Submit-department-report	Parámetro de aplicación en futuras iteraciones de la aplicación	No aplicable / null
Teacher-hours	Horas de profesor	Real
Unit-id	Identificador del módulo al que pertenece la asignatura	Entero
University-specific-degree-id	Identificador de la propuesta a la que pertenece la asignatura actual	Entero
Updated-at	Fecha de la última actualización de la asignatura	Fecha

BASE_URL/university_specific_degrees/ID_PROPUESTA/degree_teachers.xml

Este documento, asociado a cada propuesta, muestra la información de las participaciones de profesorado agregado a la titulación. A través de este documento se puede extraer la identificación del profesor y poder adquirir su perfil completo.

<degree-teachers type="array">

<degree-teacher>

Etiqueta	Significado	Tipo
Aasm-state	Estado de la participación del profesor 1. Confirmada 2. Pendiente de confirmar	Ristra
Confirmation-token	Token único que se adjunta en los correos de confirmación para los profesores	Ristra
Id	Identificador único de la participación del profesor en una propuesta	Entero
Last-notification-at	Fecha de la última notificación mandada al profesor	Fecha
Notifications-count	Contador de notificaciones remitidas al profesor hasta la fecha	Entero
Teacher-id	Identificador del profesor	Entero
University-specific-degree-id	Identificador único de la propuesta a la que la participación actual está agregada	Entero

BASE_URL/university_specific_degrees/TIPO_PROFESOR/ID_PROFESOR.xml

En este documento que proporciona la plataforma web, hay que conocer de antemano la categoría del profesor, si es de la ULPGC, de otra universidad o externo. Asimismo, proporcionar su identificador.

`<external-teacher>` `<other-teacher>` `<ulpgc-teacher>`
`</external-teacher>` `</other-teacher>` `</ulpgc-teacher>`

Etiqueta	Significado	Tipo
Aasm-state	Estado de confirmación del profesor a nivel global de la aplicación. 1. Aprobado 2. Pendiente de confirmar 3. Rechazado	Ristra
Active	Variable que indica si el profesor ha accedido alguna vez en la aplicación a confirmar sus datos	Booleano
Branch-of-knowledge	Rama de conocimiento. Se representa como ristra puesto que se determinan las ramas de conocimiento de manera más general que con respecto a las propuestas presentadas	Ristra
Created-at	Fecha de creación del perfil de profesor	Fecha
Crypted-password	Contraseña encriptada del profesor	Ristra
Current-login-at	Fecha de último acceso a la plataforma	Fecha
Cv-abstract	Breve descripción del Curriculum Vitae del profesor	Ristra
Degree	Titulación del profesor	Ristra
Department	Departamento de la ULPGC, en el caso de que sea de ese colectivo	Ristra
DNI	DNI	Ristra
Email	Dirección de correo electrónico del profesor, al que se le remitirán las notificaciones correspondientes	Ristra
Failed-login-count	Contador de intentos de autenticación fallidos	Entero
Grade-type-id	Grado de formación 1. Diplomado 2. Licenciado	Entero

	3. Doctor 4. Formación profesional 5. Otros	
Graduate-in-ulpgc	Graduado en la ULPGC	Booleano
Id	Identificador de profesor	Entero
Institute	Entidad a la que pertenece el profesor	Ristra
Last-login-at	Fecha de último acceso a la plataforma	Fecha
Last-login-ip	IP del dispositivo con el que se hizo el último acceso a la plataforma	Ristra
Last-name	Apellidos del profesor	Ristra
Last-request-at	Fecha de la última notificación	Fecha
Login-count	Contador de veces que se ha autenticado en la plataforma	Entero
Mobile	Número de teléfono móvil	Entero
Name	Nombre del profesor	Ristra
Password-salt	Sal(criptografía) de la contraseña	Ristra
Perishable-token	Token temporal de confirmación	Ristra
Persistence-token	Token de persistencia para las sesiones de usuario	Ristra
Professional-category	Categoría profesional del profesor	Ristra
Proposer-id	Identificador único del usuario proponente que agregó al profesor en la plataforma	Entero
Public-cv	Opción para hacer público el Currículum Vitae del profesor, para uso de otros proponentes	Booleano
Ulpgc-department-id	En el caso en el que el profesor sea perteneciente al colectivo de la ULPGC, identificador único del departamento al que pertenece	Entero
Ulpgc-login	En el caso en el que el profesor sea perteneciente al colectivo de la ULPGC, nombre de usuario necesario para hacer login	Ristra
Updated-at	Fecha de la última edición del perfil del profesor	Fecha

BASE_URL/university_specific_degrees/ID_PROPUESTA/issues.xml

En este documento se enumeran en un array cada uno de los errores generados mediante la verificación mecánica de la propuesta por parte de la plataforma web.

`<issues type="array">`

`<issue>`

Etiqueta	Significado	Tipo
Created-at	Fecha de creación del reporte de errores	Fecha
Id	Identificador único de error	Entero
Message	Mensaje de error	Ristra
University-specific-degree	Identificador único de la propuesta a la que el listado de errores se refiere	Entero
Updated-at	Fecha de actualización del reporte de errores	Fecha

BASE_URL/university_specific_degrees/ID_PROPUESTA/economic_report.xml

En este documento se detallan todos los conceptos económicos que componen la memoria económica de la propuesta.

`<economic-report>`

`</economic-report>`

Etiqueta	Significado	Tipo
Accomodation-cost	Coste de hospedaje de profesorado externo	Real
Admin-staff-cost	Coste de personal de apoyo	Real
Admin-staff-qty	Cantidad de personal de apoyo	Entero
Admin-staff-sum	Coste total del personal de apoyo al título	Real
Admin-support-misc-cost	Otros costes de personal de apoyo	Real
Admin-support-misc-description	Concepto de los otros costes de personal de apoyo	Ristra
Admin-support-misc-sum	Coste total de los otros coste de personal de apoyo	Real
Administrator-compensation	Compensación al proponente	Real

Advertising-sum	Coste total de publicidad en medios del título	Real
Bibliography-cost	Coste de la bibliografía necesaria para el título	Real
Bibliography-sum	Coste total de bibliografía	Real
Boards-cost	Coste del tribunal de evaluación de trabajos de fin de título	Real
Boards-qty	Cantidad de tribunales de evaluación de trabajos de fin de título	Entero
Boards-sum	Coste total de tribunales de evaluación de trabajos de fin de título	Real
Certificates-cost	Coste de emisión de certificados	Real
Computer-goods-cost	Coste de material informático	Real
Content-credits-cost	Coste de creación de material por crédito	Real
Content-credits-qty	Cantidad de créditos de material	Entero
Content-credits-sum	Coste total de creación de material	Real
Created-at	Fecha de creación de la memoria económica	Fecha
Evaluation-hours-cost	Coste de las horas de pruebas de evaluación	Real
Evaluation-hours-qty	Cantidad de horas de pruebas de evaluación	Entero
Evaluation-hours-sum	Coste total de las horas de pruebas de evaluación	Real
Exams-coordination-cost	Coste de coordinación de examen	Real
Exams-coordination-qty	Cantidad de horas de coordinación de examen	Entero
Exams-coordination-sum	Coste total de horas de coordinación de examen	Real
Expenses-sum	Suma global de gastos	Real
External-assessment-cost	Coste de evaluación externa del título	Real
Final-report-cost	Coste de evaluación de la memoria final	Real
Finantial-goods-misc-cost	Coste de reprografía	Real
Finantial goods-sum	Coste total del material de oficina	Real
Granted-students-qty	Cantidad de estudiantes becados	Entero
Granted-students-quota	Coste de matrícula para estudiantes becados	Real
Granted-students-sum	Ingresos totales a través de matrícula de estudiantes becados	Real
Id	Identificador único de la propuesta a la que pertenece la memoria académica	Entero

Incomes-sum	Suma global de ingresos	Real
Information-point-cost	Coste del punto de información	Real
Institution-compensation	Coste de la compensación al centro adscrito	Real
Lab-goods-cost	Coste del material de laboratorio	Real
Locomotion-cost	Costes de desplazamiento del profesorado	Real
Locomotion-sum	Coste total de locomoción del profesorado	Real
Maintenance-cost	Coste de mantenimiento de equipamiento	Real
Media-advertising-cost	Coste de publicidad en medios	Real
Misc-misc-cost	Coste de otros conceptos	Real
Office-goods-cost	Coste de material de oficina	Real
Opening-closing-cost	Coste de ceremonias de apertura/clausura	Real
Other-bibliography-cost	Otros costes de bibliografía	Real
Other-costs-sum	Coste total de otros conceptos	Real
Other-locomotion-costs	Otros costes de locomoción	Real
Other-staff-cost	Coste de personal de apoyo adicional	Real
Other-staff-qty	Cantidad de personal de apoyo adicional	Entero
Other-staff-sum	Coste total del personal de apoyo adicional	Real
Partial-expenses-sum	Suma parcial de los gastos, previo a compensaciones	Real
Practice-hours-cost	Coste de la hora de prácticas	Real
Practice-hours-qty	Cantidad de horas de prácticas	Entero
Practice-hours-sum	Coste total de las horas de prácticas	Real
Presentation-report-cost	Coste del informe de presentación	Real
Printed-advertising-cost	Coste de la publicidad en medios impresos	Real
Regular-students-qty	Cantidad de alumnos sin beca	Entero
Regular-students-quota	Coste de la matrícula para alumnos sin beca	Real
Regular-students-sum	Ingresos totales en concepto de matrículas de estudiantes sin beca	Real
Reprographics-cost	Costes de reprografía	Real
Scholar-cost	Coste de profesorado externo	Real
Scholar-qty	Cantidad de profesores externo	Entero
Scholar-sum	Coste total de profesorado externo	Real

Seminar-hours-cost	Coste de las horas de seminario	Real
Seminar-hours-qty	Cantidad de horas de seminario	Entero
Seminar-hours-sum	Coste total de las horas de seminario	Real
Specific-scholar-cost	Coste de profesorado específico	Real
Specific-scholar-qty	Cantidad de profesorado específico	Entero
Specific-scholar-sum	Coste total de profesorado específico	Real
Students-in-practice-cost	Coste de alumnado en prácticas	Real
Students-incoming	Ingresos totales por concepto de matrículas de estudiantes	Real
Students-insurance-cost	Coste del seguro de estudiantes	Real
Sum	Balance	Real
Support-service-cost	Coste de servicio de apoyo a la gestión del título	Real
Support-service-description	Descripción de los servicios adicionales de apoyo al título	Ristra
Teachers-cost-sum	Coste de profesorado específico	Real
Teachers-misc-cost	Coste adicional específico del profesorado. Otras categorías	Real
Teachers-misc-cost-description	Descripción del profesorado de apoyo adicional específico	Ristra
Teachers-total-sum	Coste total del profesorado del título	Real
Translation-cost	Costes de traducción simultánea o sucesiva	Real
Tutorship-jobs-cost	Coste de tutelaje de trabajos finales	Real
Tutorship-jobs-qty	Cantidad de trabajos finales a tutorizar	Entero
Tutorship-jobs-sum	Coste total de tutorías de trabajos finales	Real
Ulpgc-boards-cost	Coste de tribunal para evaluación de trabajo final	Real
Ulpgc-boards-qty	Cantidad de tribunales para evaluaciones de trabajos finales	Entero
Ulpgc-boards-sum	Coste total de tribunales para la evaluación de trabajos finales	Real
Ulpgc-compensation	Compensación económica a la ULPGC	Real
Ulpgc-content-credit-cost	Coste de creación de contenido procedente de profesorado de la ULPGC	Real
Ulpgc-content-credit-qty	Cantida de créditos de contenido generado	Entero

	procedente de profesorado de la ULPGC	
Ulpgc-content-credit-sum	Coste total de creación de contenido procedente de profesorado de la ULPGC	Real
Ulpgc-evaluation-hours-cost	Coste de pruebas de evaluación de profesorado de la ULPGC	Real
Ulpgc-evaluation-hours-qty	Cantidad de horas de pruebas de evaluación de profesorado de la ULPGC	Entero
Ulpgc-evaluation-hours-sum	Coste total de pruebas de evaluación de profesorado de la ULPGC	Real
Ulpgc-exams-coordination-cost	Coste de coordinación de exámenes de profesorado de la ULPGC	Real
Ulpgc-exams-coordination-qty	Cantidad de horas de coordinación de exámenes de profesorado de la ULPGC	Entero
Ulpgc-exams-coordination-sum	Coste total de coordinación de exámenes de profesorado de la ULPGC	Real
Ulpgc-managers-cost	Coste de director académico	Real
Ulpgc-managers-qty	Número de directores académicos	Entero
Ulpgc-practice-hours-cost	Coste de hora de prácticas de profesorado de la ULPGC	Real
Ulpgc-practice-hours-qty	Cantidad de horas de prácticas de profesorado de la ULPGC	Entero
Ulpgc-practice-hours-sum	Coste total de prácticas de profesorado de la ULPGC	Real
Ulpgc-seminars-hours-cost	Coste de hora de seminarios por parte del profesorado de la ULPGC	Real
Ulpgc-seminars-hours-qty	Cantidad de horas de seminarios por parte del profesorado de la ULPGC	Entero
Ulpgc-seminars-hours-sum	Coste total de seminarios por parte del profesorado de la ULPGC	Real
Ulpgc-teachers-cost-sum	Coste total del profesorado de la ULPGC	Real
Ulpgc-teachers-gross	Coste bruto del profesorado de la ULPGC	Real
Ulpgc-teachers-organisation-profit	Beneficios para el centro adscrito	Real
Ulpgc-teachers-total-sum	Coste total del profesorado de la ULPGC, incluyendo todos los conceptos	Real
Ulpgc-teachers-unit-profit	Beneficios para la unidad organizadora	Real

Ulpgc-teaching-hours-cost	Coste de la hora teórica por parte del profesorado de la ULPGC	Real
Ulpgc-teaching-hours-qty	Cantidad de horas teóricas impartidas por el profesorado de la ULPGC	Entero
Ulpgc-teaching-hours-sum	Coste total de horas teóricas impartidas por el profesorado de la ULPGC	Real
Ulpgc-tutorship-jobs-cost	Coste de la hora de tutoría por parte del profesorado de la ULPGC	Real
Ulpgc-tutorship-jobs-qty	Cantidad de horas de tutoría por parte del profesorado de la ULPGC	Entero
Ulpgc-tutorship-jobs-sum	Coste total de tutorías por parte del profesorado de la ULPGC	Real
University-specific-degree	Identificador único de la propuesta a la que la memoria económica pertenece	Entero
Updated-at	Fecha de la última modificación de la memoria económica	Fecha
Virtual-campus-cost	Coste del campus virtual para la impartición de las enseñanzas	Real

5.2 Diseño

En la etapa de diseño se establece la arquitectura con la que se va a modelar el software para cumplir con los requisitos. Debido a las características de la plataforma web a través de la cuál la aplicación móvil que se desarrolla va a alimentarse de datos, la arquitectura que se elige es la más intuitiva en este caso, Cliente-Servidor.

El modelo cliente-servidor en computación es una estructura de aplicación distribuída que particiona tareas o carga de trabajo entre los proveedores de un recurso o servicio, llamados servidores, y los peticionarios de dicho servicio, llamados clientes. El cliente no comparte ninguno de sus recursos, al contrario, lanza peticiones al servidor solicitando contenido o servicios. Es el cliente el que inicia las sesiones de comunicación con el servidor, que aguarda las peticiones entrantes. Para describir el comportamiento dinámico que la aplicación habrá de mostrar en la última iteración del proceso, se ilustra en el siguiente diagrama de actividades.

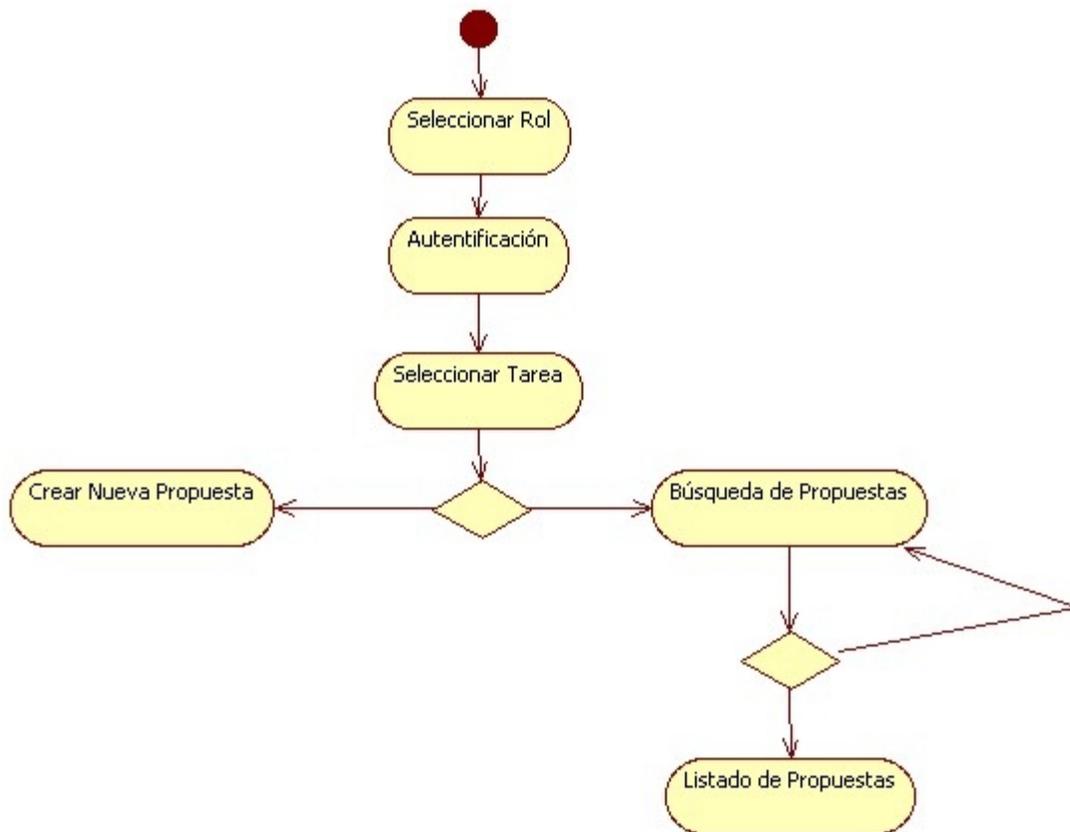


Imagen 23: Diagrama de actividades. 1ª parte

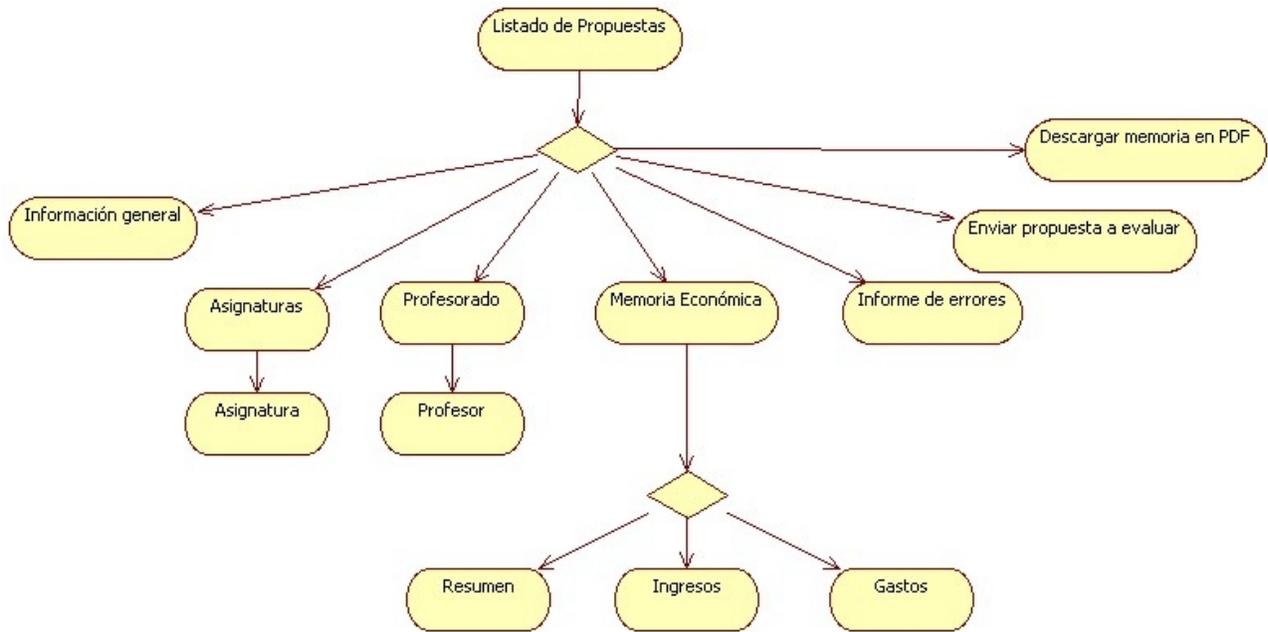


Imagen 24: Diagrama de actividades, 2ª parte

Se definen las siguientes 18 actividades, usando la actividad como unidad mínima de desarrollo en Android, puesto que cada una de ellas definirá la composición de la información de la pantalla a mostrar. Se divide en tres secciones la información de la memoria económica debido al gran número de variables que han de ser mostradas.

A continuación, se define el comportamiento de la plataforma web en base a las diferentes peticiones que el usuario puede hacer para comunicarse, obtener información o provocar una acción posterior del sistema.

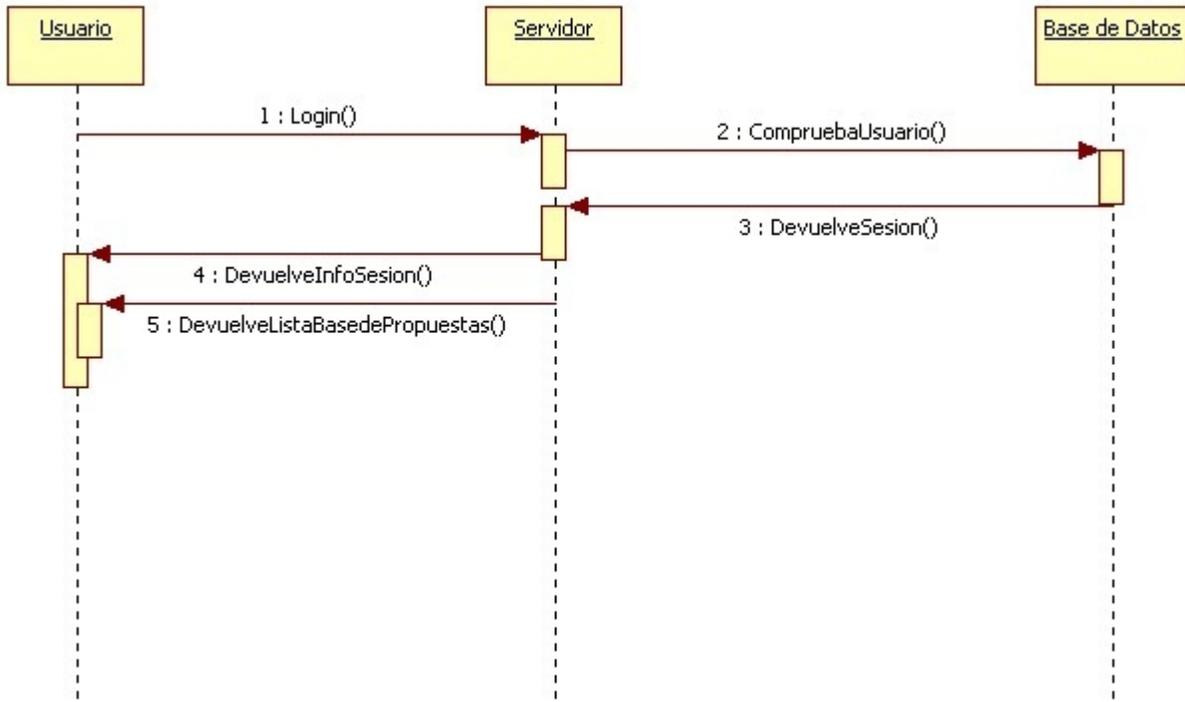


Imagen 25: Diagrama de Secuencia. Autenticación

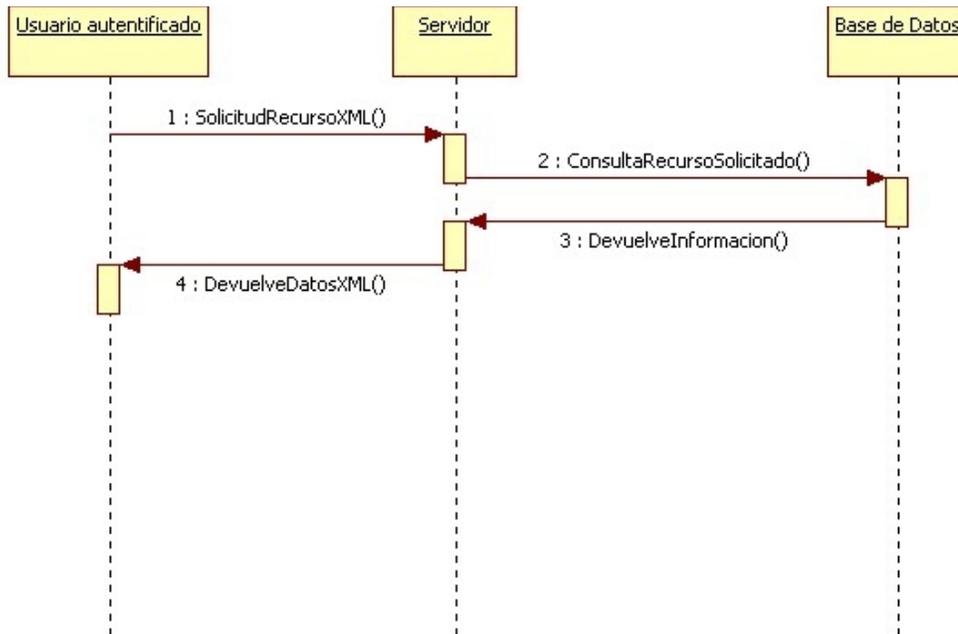


Imagen 26: Diagrama de Secuencia. Solicitud de Recurso XML

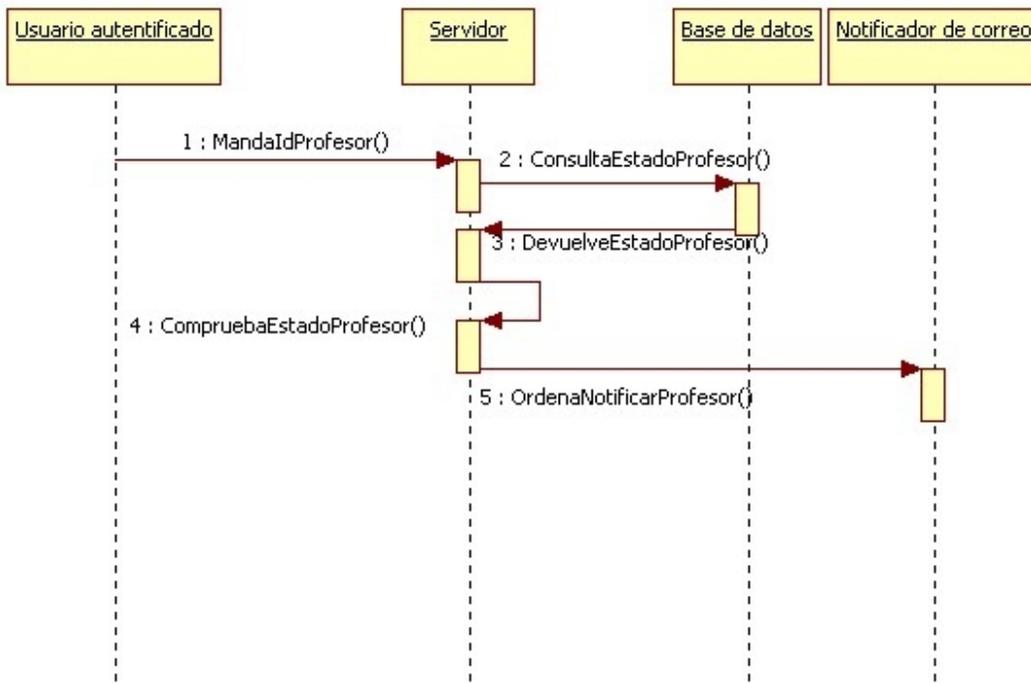


Imagen 27: Diagrama de Secuencia. Notificación de profesor

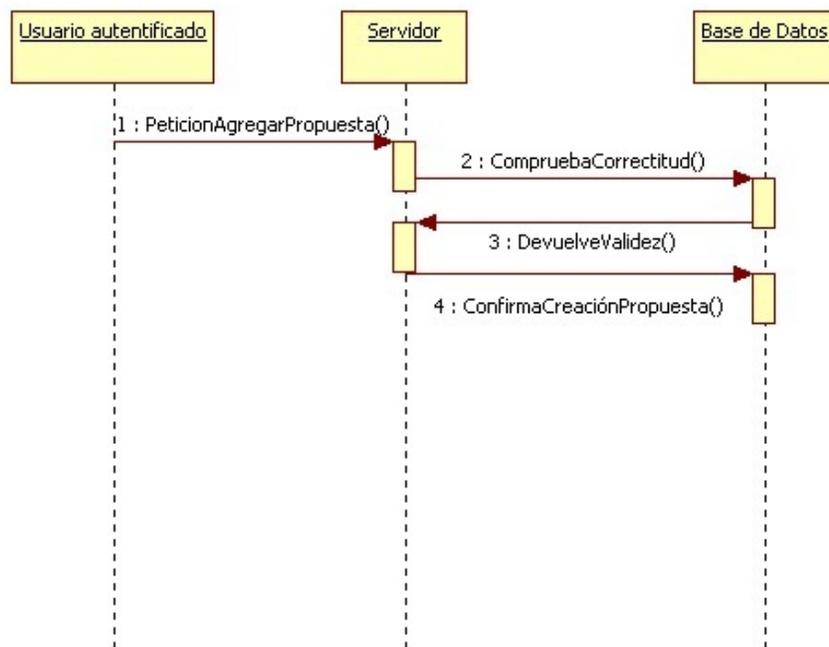


Imagen 28: Diagrama de Secuencia. Creación de Nueva Propuesta

5.3 Desarrollo

5.3.1 Metodología

Desarrollo iterativo incremental

Debido a la naturaleza del proyecto, el desconocimiento de las herramientas en el momento de inicio del proyecto y la necesidad de mostrar un producto considerado 'final' para recibir *feedback* de los usuarios con el objetivo de mejorar la calidad del código, la interfaz y el comportamiento, se escoge la metodología de desarrollo iterativo e incremental.

El desarrollo de software iterativo e incremental es un método de desarrollo software que está modelado alrededor de un incremento gradual de adiciones de funcionalidades y una liberación cíclica del producto.

Empieza con la planificación y continúa a través de ciclos de desarrollo iterativo utilizando '*feedback*' continuo por parte del usuario y concluyendo en el despliegue del software completo al final de cada ciclo.

Es una disciplina para desarrollar sistemas basado en producir resultados. En el desarrollo incremental, diferentes partes de un sistema son desarrollados en diferentes momentos o ritmos de desarrollo y son integrados basados en su finalización. En desarrollo iterativo, el desarrollador planea revisar partes del sistema con el objetivo de mejorarlos. El '*feedback*' del usuario es consultado para modificar los objetivos en los siguientes productos finalizados.

El desarrollo de software iterativo e incremental emana en respuesta de las debilidades en el modelo de desarrollo en cascada, un modelo de diseño secuencial en el cual el progreso fluye hacia abajo continuamente. Se diferencia del modelo en cascada en que es cíclico en vez de unidireccional, ofreciendo una mejor habilidad para incorporar cambios dentro de la aplicación durante el ciclo de desarrollo.

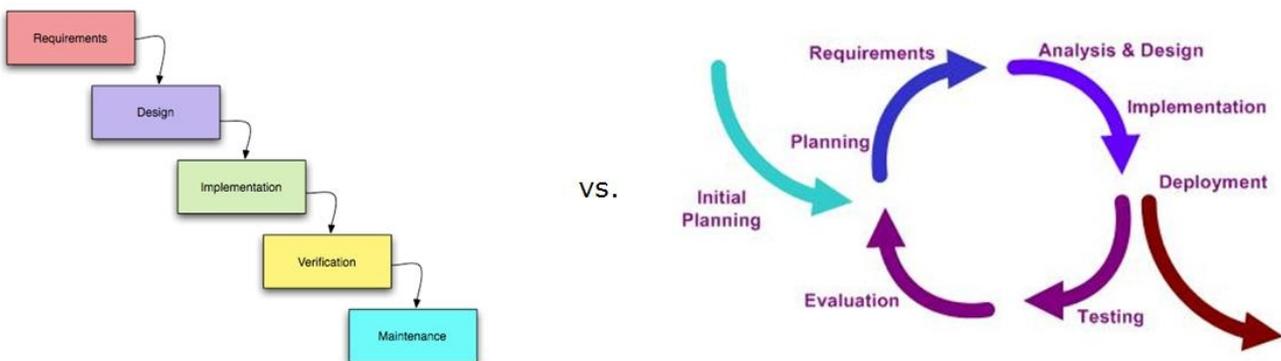


Imagen 29: Modelo en cascada vs Modelo iterativo-incremental

Primera iteración

Entre los objetivos de la primera iteración de la aplicación móvil es la de obtener un prototipo funcional de la arquitectura modelada en la etapa de diseño. Esta iteración se desarrolló durante un periodo más prolongado que las subsiguientes, debido a la falta de familiaridad con las herramientas, lenguaje, arquitectura del sistema operativo Android y las especificidades de la programación en este soporte.

Como elementos esenciales que deben estar presentes en el primer prototipo de la aplicación se encontrarán:

- Autenticación sobre el servidor Rails de desarrollo local
- Solicitud de recursos XML

Para la primera tarea, se ha obtenido una instantánea de la base de datos PostgreSQL por parte de los desarrolladores de la plataforma. Asimismo, el código fuente y sus herramientas necesarias para montar un servidor local completo en el que hacer las modificaciones adecuadas.

Se ha instalado un gestor de bases de datos **PostgreSQL 9.1.9**. En ésta, se crean tres usuarios correspondientes a las instancias de desarrollo(dev), pruebas(test) y producción(prod). A continuación, se importa la instantánea de la base de datos obtenida de los desarrolladores de la aplicación original mediante el comando

```
pg_restore -U usuario -d bbdd -Fc instantanea.dump
```

A continuación, se instala un servidor **Rails 3.2.9**. Se asocia la nueva base de datos con el código Ruby on Rails de la plataforma, agregando los usuarios y los accesos creados a su fichero de configuración en *config/database.yml*. Los comandos listados a continuación crean una base de datos importando la información deseada y asociandola con la aplicación web.

```
rake db:drop
```

```
rake db:create
```

```
rake db:migrate
```

config/database.yml

```
development:  
  adapter: postgresql  
  database: railsdev  
  username: dev  
  password: devpwd  
  host: localhost
```

```
test:  
  adapter: postgresql  
  database: railstest  
  username: test  
  password: testpwd  
  host: localhost
```

```
production:  
  adapter: postgresql  
  database: railsprod  
  username: prod  
  password: prodpwd  
  host: localhost
```

Después de completar todas estas etapas de preparación, se inicializa el servidor mediante el comando:

```
rails server
```

A partir de este punto se comienza la programación de la aplicación. El objetivo de esta primera iteración será iniciar sesión contra el servidor para seguidamente obtener un recurso XML mediante una petición *HTTP Get*.

Esta operación se decidió hacerla mediante la creación de una instancia de `AsyncTask`.

```
java.lang.Object  
  ↳ android.os.AsyncTask<Params, Progress, Result>
```

Dentro de esta operación asíncrona, que se ejecutará fuera del hilo de la interfaz de usuario, se crean las peticiones necesarias y se ejecutan para obtener un resultado.

```

CookieStore cookieStore = new BasicCookieStore();
HttpContext localContext = new BasicHttpContext();
localContext.setAttribute(ClientContext.COOKIE_STORE, cookieStore);

HttpClient client = new DefaultHttpClient();
HttpPost post = new HttpPost(urls[0].toString() + "user_session");

List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("utf8", "✓"));

params.add(new BasicNameValuePair("user_session[login]", login));
params.add(new BasicNameValuePair("user_session[password]", password));
UrlEncodedFormEntity ent = new UrlEncodedFormEntity(params, HTTP.UTF_8);
post.setEntity(ent);
HttpResponse responsePOST = client.execute(post, localContext);
HttpEntity resEntity = responsePOST.getEntity();

if (resEntity != null) {
    responsePOST.getEntity().consumeContent();
}

HttpGet httpGet = new HttpGet(addr + RESOURCE_URL);
HttpResponse responseGET = null;
String xml_response = null;
try {
    responseGET = client.execute(httpGet, localContext);
    HttpEntity getEntity = responseGET.getEntity();

    if (getEntity != null) {
        xml_response = EntityUtils.toString(getEntity);
    }

} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    error = e;
    e.printStackTrace();

    return null;
}
return xml_response;

```

De esta manera, examinando la variable *xml_response* correspondiente a la respuesta que el servidor da a nuestra petición de autenticación se determina si el inicio ha sido correcto o no. En el caso de que los datos hayan quedado validados, la respuesta contendrá el primer bloque de 20 propuestas, si las hubiera, que el usuario tiene asociadas a su cuenta.

Una vez conseguida la autenticación correctamente, se debe proporcionar un mecanismo para obtener las mismas instancias de `HttpClient`, `HttpContext` y `BasicCookieStore`, con el objetivo de seguir realizando peticiones de recursos al servidor bajo la misma sesión abierta.

5.4 Patrón singleton

Se determina que la mejor manera es establecer un patrón de diseño Singleton para mantener la información a lo largo de toda la sesión en la aplicación móvil. Asimismo, se decide mantener dentro de esta clase la dirección base del servidor al que se está trabajando, para poder cambiar desde el servidor de pruebas al de producción con facilidad.

```
[...]
public class GlobalState extends Application {

    private CookieStore cookieStore;
    private HttpContext localContext;
    private DefaultHttpClient client;
    private String base_url;
    private boolean role;

    public String getTargetAddress() {
        return base_url;
    }

    public CookieStore getCookieStore() {
        return cookieStore;
    }

    public HttpContext getLocalContext() {
        return localContext;
    }

    public DefaultHttpClient getHttpClient() {
        return client;
    }

    public boolean getRole() {
        return this.role;
    }

    public void setCookieStore(CookieStore cs) {
        this.cookieStore = cs;
    }

    public void setLocalContext(HttpContext lc) {
        this.localContext = lc;
    }

    public void setHttpClient(DefaultHttpClient hc) {
        this.client = hc;
    }
}
```

```

public void setRole(boolean role) {
    this.role = role;
}

public void setBaseAddress(String base_url) {
    this.base_url = base_url;
}
}

```

Asimismo, para que la instancia creada sea accesible en todo el ciclo de vida de la aplicación habrá que especificarlo en el documento de permisos del proyecto *AndroidManifest.xml*

```

[...]
<application
    android:name="es.ulpgc.titulospropios.gestion.ubay.GlobalState"
    android:allowBackup="true"
[...]

```

Utilizando el código hasta ahora mencionado permitirá la realización de peticiones de recursos al servidor a lo largo de la ejecución del programa.

La estructura subsiguiente de las actividades consistirán en la generación programática de los elementos del layout en base a las necesidades de presentación en la interfaz de usuario. La estructura de todas las actividades que realizan peticiones de recursos creará el *layout* en el hilo de usuario, se ejecutará la instancia de la clase *AsyncTask* para obtener y procesar la información requerida. Por último, en el método *onPostExecute* de *AsyncTask*, las vistas del *layout* se pueblan con la información procesada proveniente de *xml_response*.

Esta iteración inicial se puede calificar de especulativa y exploratoria, puesto que surgen numerosos problemas en la fase de pruebas.

- El rendimiento se ve significativamente afectado a causa de la creación de las vistas y componentes del *layout* de manera programática.
- La aparición de excepciones de entrada/salida provocan detenciones no deseadas del programa, asimismo provocando excepciones de punteros nulos durante la ejecución.
- El código generado para ejecutar una petición y a continuación procesar el documento XML para extraer las etiquetas y su información asociada es excruciantemente largo.

Segunda iteración

Los objetivos perseguidos en la segunda iteración han sido tratar de solucionar los problemas planteados en la fase de prueba y evaluación de la primera iteración, así como desligar las ristas de texto que anteriormente estaban *Hard-coded* en el código hacia su correspondiente lugar en el directorio de recursos */values/strings.xml*

Se adjunta un pequeño ejemplo del transvase de las rstras desde el código al fichero de recursos y su forma de direccionar éstas.

LoginUser.java

```
[...]  
if (manager)  
    textRole.setText(R.string.accessMessageManager);  
else  
    textRole.setText(R.string.accessMessageTeacher);  
[...]
```

Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="appName">Gestor Titulos Propios ULPGC</string>
    <string name="accessMessageManager">Acceder como Gestor</string>
    <string name="accessMessageTeacher">Acceder como Docente</string>
[...]
```

Asimismo, se establece como objetivo realizar las peticiones mediante una conexión segura (HTTPS). Se descubre que esta circunstancia se realiza de manera transparente mediante las clases utilizadas para la conexión, como HttpClient.

Se realiza el traspaso de los elementos de la interfaz hacia los *layouts*, requiriendo una modificación significativa del código elaborado en la anterior iteración. A partir de este momento se harán las referencias a las vistas de la interfaz mediante la siguiente operación, como ejemplo.

```
[...]
TextView textRole = (TextView) findViewById(R.id.login_manager);
[...]
```

Para abordar el problema de la extensión excesiva del código utilizado para realizar petición y extracción de etiquetas, se crea una nueva clase que se ocupará de tratar el documento de texto plano retornado por las peticiones HTTP, permitiendo esta clase la extracción de determinadas etiquetas significativas y la división de instancias de una misma entidad en listas de entidades independientes.

XmlExtractor.java

```
public class XmlExtractor {

    private XmlPullParserFactory factory;
    private XmlPullParser xpp;
    private String input;

    public XmlExtractor(String input) {

        try {
            factory = XmlPullParserFactory.newInstance();
            factory.setNamespaceAware(true);
            xpp = factory.newPullParser();
            xpp.setInput(new StringReader(input));

        } catch (XmlPullParserException e) {
            e.printStackTrace();
        }
    }
}
```

```

        this.input = input;
    }

    public void setInput(String newInput) throws XmlPullParserException {
        input = newInput;
        xpp.setInput(new StringReader(input));
    }

    public String extractSingleTag(String tag) {
        String output = null;

        int eventType = 0;
        try {
            eventType = xpp.getEventType();
            while (eventType != XmlPullParser.END_DOCUMENT) {
                if (eventType == XmlPullParser.START_TAG) {
                    if (tag.equals(xpp.getName())) {
                        eventType = xpp.next();
                        output = xpp.getText();
                        break;
                    }
                }
                eventType = xpp.next();
            }
        } catch (XmlPullParserException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return output;
    }

    public HashMap<String, String> extractTagsFromDoc(String[] tags) {
        HashMap<String, String> output = new HashMap<String, String>();

        int eventType = 0;
        try {
            eventType = xpp.getEventType();
            while (eventType != XmlPullParser.END_DOCUMENT) {
                if (eventType == XmlPullParser.START_TAG) {
                    if (Arrays.asList(tags).contains(xpp.getName())) {
                        String temp_tag = new String(xpp.getName());
                        eventType = xpp.next();
                        output.put(temp_tag, xpp.getText());
                    }
                }
            }
        }
    }

```

```

        eventType = xpp.next();
    }
} catch (XmlPullParserException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
return output;
}

public ArrayList<String> unitSeparator(String endTag) {
    String temp_trimdoc = input;
    ArrayList<String> trimmedDoc = new ArrayList<String>();
    boolean contains = input.contains(endTag);

    while (contains) {
        trimmedDoc.add(temp_trimdoc.substring(0,
            temp_trimdoc.indexOf(endTag) + endTag.length()));
        temp_trimdoc =
temp_trimdoc.substring(temp_trimdoc.indexOf(endTag)
            + endTag.length());
        contains = temp_trimdoc.contains(endTag);
    }
    return trimmedDoc;
}
}

```

De igual forma, para el caso de búsqueda cuando el número de propuestas es mayor a 20, que tiene la especificidad de no saber si hay que seguir haciendo peticiones para completar la descarga de todas los bloques de propuestas, se crea una nueva clase que alberga una página completa de 20 propuestas, para realizar las comprobaciones necesarias para seguir o no haciendo peticiones al servidor para completar la descarga.

XmlPage.java

```

[...]  
public class XmlPage {  
    private Context mContext;  
  
    public XmlPage(Context mContext) {  
        this.mContext = mContext;  
    }  
  
    public String getPage(String target_addr) throws IOException {  
        GlobalState gs = (GlobalState) mContext.getApplicationContext();  
        HttpContext localContext = gs.getLocalContext();  
        DefaultHttpClient client = gs.getHttpClient();

```

```

HttpGet httpGet = new HttpGet(target_addr);
HttpResponse responseGET = null;
HttpEntity getEntity = null;
String response = null;
try {
    responseGET = client.execute(httpGet, localContext);
    getEntity = responseGET.getEntity();
    response = EntityUtils.toString(getEntity);
    getEntity.consumeContent();
} catch (ClientProtocolException e) {
    e.printStackTrace();
}

return response;
}
}

```

Habiendo solucionado los problemas derivados de la primera iteración, excepto la gestión de las interrupciones de entrada/salida, surgen nuevos problemas.

Se descubre que el servidor no está preparado para responder en formato XML a las peticiones que le solicitan las entidades relacionadas con la memoria económica, profesorado y reporte de errores. Sin acceso a estas entidades, la funcionalidad de la aplicación móvil se ve severamente recortada.

Se adjunta como anexo las modificaciones que fueron enviadas a los responsables del desarrollo de la aplicación web servidor, en formato *patch*, para proporcionar al servidor la capacidad de responder a las peticiones de las mencionadas entidades.

Tercera iteración

El inicio de esta tercera iteración lo marca la aplicación del parche propuesto al servidor de producción de la aplicación web por parte de sus responsables. Esto permite completar el árbol de actividades que proporciona la información determinada en la fase de captura de requisitos.

Llegado a este punto, se descubre que la información relacionada con el profesorado no es accesible, siendo accesible únicamente la información sobre su participación en cada propuesta. Este documento disponible denominada *teacher_degree* en la base de datos del servidor, proporciona una Id única para relacionarlo con la entidad *teachers*. No obstante, es imposible el acceso a ellos, puesto que hay que conocer de antemano su tipo de profesor (ULPGC, Otra Universidad, Externo) para poder acceder a su ficha de información.

Esta modificación fue motivo de retraso en el desarrollo de la aplicación móvil por un periodo de más de dos meses. A causa de esta circunstancia se deriva un alargamiento de las fases de captura de requisitos, así como la de análisis y diseño.

Los frutos de este análisis extendido y el tiempo disponible antes de poder realizar

implementación de los cambios solicitados, en primer lugar fue la traducción de todas las ristras de la aplicación al inglés. Esto se realiza duplicando el fichero de recursos de ristras y agregar uno nuevo en el fichero

/values/strings.xml

```
<string name="searchMessage_before">Por el momento, se han encontrado </string>
<string name="searchMessage_after"> propuestas que cumplen los criterios de
                                     búsqueda</string>
<string name="searchResults_before">Resultados de la búsqueda:\nSe han encontrado
</string>
<string name="searchResults_after">títulos que coinciden con los \nparámetros de
                                     búsqueda introducidos</string>
```

/values-en/strings.xml

```
<string name="searchMessage_before">The ongoing search has found </string>  
<string name="searchMessage_after"> degree proposals that match the parameters at  
the moment</string>  
<string name="searchResults_before">Search results:\nThere has been </string>  
<string name="searchResults_after"> results</string>
```

Una vez los responsables de la plataforma web aplican los cambios propuestos, así como la modificación que permitirá acceder a las fichas de profesorado, se continúa el desarrollo de la implementación planeada. De manera análoga a como se listan las asignaturas y se tiene acceso a la ficha de cada una de ellas pulsando en su denominación, se implementa el profesorado. No obstante, su carga será asíncrona, utilizando el método *onProgressUpdate* de *AsyncTask*. Una vez descargada la lista de participaciones de profesorado de *degree_teachers*, se deben descargar una por una las fichas de profesorado para obtener su información básica con la que poblar la lista. Cada vez que una ficha de profesor sea recibida y procesada, se enviará un nuevo elemento a la lista. Se agrega un contador de profesorado en la parte inferior de la pantalla para que el usuario sepa el número total de profesores que se están cargando en ese momento.

Llegados a este punto, se agrega el botón 'Notificar' junto a la denominación del profesor, si este profesor se encuentra en estado 'Pendiente de confirmar participación'. Este código se encontrará embebido en la clase que crea y puebla de profesores la lista correspondiente.

Una vez concluida la fase de implementación de esta iteración, siguen persistiendo los antiguos errores relacionados con las excepciones de entrada/salida, que imposibilitan la recuperación de la ejecución de la aplicación. Asimismo, las aproximaciones realizadas para intentar paliar este efecto se encuentran con otro problema derivado de la creación de cuadros de diálogo en múltiples actividades concurrentemente. Esto se denomina filtración de contexto entre actividades.

Se establece como prioritario para la última iteración la corrección de los mencionados errores funcionales de la aplicación

Cuarta iteración

Como última etapa de desarrollo de la implementación, los objetivos establecidos serán la corrección de los problemas derivados y heredados de las anteriores iteraciones, interrupciones entrada/salida, filtraciones de contexto, así como una adaptación del código para que tenga una visualización adaptable para dispositivos de pantalla de gran tamaño como tabletas.

La adaptación a las tabletas requiere una nueva creación de *layouts* que se ajusten al nuevo comportamiento, así como una modificación de la estructura de control que gestiona los cambios entre actividades. En este caso se tendrán que usar fragmentos para poder componer la pantalla con la información solicitada.

Para lograr este efecto, se establece el nuevo control en dos puntos de la aplicación. La actividad de búsqueda de propuestas es la primera, para permitir que al lado del fragmento que contiene

el mecanismo para especificar parámetros de búsqueda conviva lado a lado con el fragmento que representará el listado de propuestas.

En segundo lugar, en la actividad de menú principal se agregará el control adecuado para realizar la sustitución de fragmentos necesaria para utilizar los tres sectores de pantalla en los que se divide la interfaz de usuario en ese punto de la ejecución. Como ejemplo, se adjunta el código correspondiente con la pulsación de una de sus opciones.

DegreeMainMenu.java

```
[...]
@Override
public void OnSubjectsClicked() {
    if (findViewById(R.id.central_fragment) == null) {

        Intent intent = prepareIntent();
        intent.setClass(this, SubjectsList.class);
        startActivity(intent);
    } else {
        hideFullscreenLayout();
        unfoldSmallerFrameLayouts();

        Bundle args = getIntent().getExtras();
        SubjectsListFragment subjectsListFragment = new
SubjectsListFragment();
        subjectsListFragment.setArguments(args);
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.central_fragment, subjectsListFragment)
            .commit();
    }
}
[...]
```

Asimismo, para los casos en los que se ha de gestionar el tercer fragmento, es decir, cuando se selecciona la información de un profesor o asignatura completa, así como para elegir una de las tres secciones correspondientes con la memoria económica, se ha de agregar una sección de control de fragmentos.

El funcionamiento, en el caso de asignatura y profesor, será controlado mediante una interfaz que notifica a la actividad llamadora que se ha producido un click, para que modifique el fragmento lateral derecho y lo sustituya por el fragmento requerido.

SubjectListFragment.java

```
[...]
@Override
public void OnSubjectClicked(String subject_info) {
    if (getActivity().findViewById(R.id.lateral_fragment) == null) {

        Intent intent = new Intent(getActivity(), SingleSubject.class);
```

```

        intent.putExtra("subject-info", subject_info);
        startActivity(intent);
    } else {
        Bundle args = new Bundle();
        args.putString("subject-info", subject_info);
        SingleSubjectFragment singleSubjectFragment = new
SingleSubjectFragment();
        singleSubjectFragment.setArguments(args);
        getActivity().getSupportFragmentManager().beginTransaction()
            .replace(R.id.lateral_fragment, singleSubjectFragment)
            .commit();
    }
}
[...]
```

En el caso de la memoria económica, se agrega un fragmento simple que infla un *layout* con tres alternativas correspondientes a Balance, Ingresos y Gastos. Una vez escogida la sección, en el fragmento lateral derecho aparecerá el fragmento que infla la información requerida por separado.

A continuación, se muestra la solución propuesta para solucionar la problemática de las excepciones de entrada/salida y las filtraciones de contexto. Se ha optado por extender la clase *FragmentManager*, que pertenece a la librería *support-v4*, dedicada a la compatibilidad con dispositivos con versiones antiguas de Android.

Con esta clase extendida, se establecen métodos para la creación de ventanas de diálogo. Utilizando únicamente estos métodos nos aseguramos de que no se producen filtraciones de contexto, puesto que sólo la actividad en ejecución se encarga de la creación, modificación y eliminación de dichas ventanas. Asimismo, se establece otro método al que se llama cuando se produce una excepción de entrada/salida, proporcionando una manera de mostrar un mensaje de error y a continuación recuperar la ejecución de la aplicación, ya sea repitiendo la operación, volviendo al inicio o a la actividad anterior.

MyFragmentManager.java

```

public class MyFragmentManager extends FragmentManager {
    protected Context mActivity;
    protected ProgressDialog progressDialog;

    protected void AlertDialogGoingBack(int id) {

        if (progressDialog != null) {
            if (progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
        }

        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        mActivity = this;
    }
}
```

```

        alertDialogBuilder.setTitle(R.string.error_period);

        alertDialogBuilder
            .setMessage(id)
            .setCancelable(false)
            .setPositiveButton(R.string.ok,
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog, int id) {
                        finish();
                    }
                });
        alertDialogBuilder.create().show();
    }

    protected void setUpProgressDialog(int id) {
        if (progressDialog != null) {
            if (progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
        }

        progressDialog = new ProgressDialog(this, R.style.MyTheme);
        progressDialog.setCancelable(false);
        progressDialog.setMessage(getResources().getString(id));
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setProgress(0);
        progressDialog.show();
    }

    protected void dismissWindow() {
        if (progressDialog != null) {
            if (progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
        }
    }

    protected void setUpLoginAlertDialogError() {
        if (progressDialog != null) {
            if (progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
        }
        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        alertDialogBuilder.setTitle(R.string.errorLogin);
        alertDialogBuilder
            .setMessage(R.string.message_back_start)
            .setCancelable(false)
            .setPositiveButton(R.string.ok,

```

```

        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface
dialog, int id) {
                // do nothing
            }
        });
        alertDialogBuilder.create().show();
    }

    protected void launchErrorWindow(int error_id) {
        if (progressDialog != null) {
            if (progressDialog.isShowing()) {
                progressDialog.dismiss();
            }
        }
        AlertDialog.Builder alertDialogBuilder = null;

        if (error_id != 0)
            alertDialogBuilder = IOError();
        else
            alertDialogBuilder = IOErrorWithRetry();

        alertDialogBuilder.create().show();
    }

    private Builder IOErrorWithRetry() {
        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        mActivity = this;
        alertDialogBuilder.setTitle(R.string.error_io);

        alertDialogBuilder
            .setMessage(R.string.error_io_message)
            .setCancelable(false)
            .setPositiveButton(R.string.retry,
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog, int id) {
                        Intent intent = getIntent();
                        finish();
                        startActivity(intent);
                    }
                });
        return alertDialogBuilder;
    }

    private Builder IOError() {
        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        mActivity = this;

```

```
        alertDialogBuilder.setTitle(R.string.error_io);

        alertDialogBuilder
            .setMessage(R.string.error_io_message)
            .setCancelable(false)
            .setPositiveButton(R.string.ok,
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog, int id) {
                        // Do nothing
                    }
                });
        return alertDialogBuilder;
    }

    protected void updateProgressDialog(String message) {
        progressDialog.setMessage(Html.fromHtml(message));
    }
}
```

Como ejemplo de la captura de la excepción, se adjunta parte de una implementación de la clase *AsyncTask*, en su método *onPostExecute*, que descarta la actual ventana de diálogo y comprueba si se ha producido un error. Si se ha producido tal error, se llama al método de la clase extendida *MyFragmentActivity* con el parámetro deseado relacionado con la acción para la recuperación. Si no se produce error, se llama al método encargado de poblar las vistas del *layout* con la información recibida en la implementación de *AsyncTask*.

AcademicInfoFragment.java

```
@Override
protected void onPostExecute(Void param) {

    ((MyFragmentActivity) getActivity()).dismissWindow();
    if (error != null) {
        ((MyFragmentActivity) getActivity()).launchErrorWindow(0);
    } else {
        populateViews(degree.getData());
    }
}
```

Por último, se retocan los ficheros de *layout*, para que sus componentes visuales tengan el aspecto recomendado por el manual de identidad gráfica corporativa de la ULPGC. Se adjunta un elemento básico de la creación de componentes más complejos de los elementos *drawable* de la aplicación.

invert_gradient_background_noborder.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android" >

    <gradient
        android:angle="270"
        android:endColor="#B0C4DE"
        android:startColor="#FFFFFF" />

    <padding
        android:bottom="7dp"
        android:left="7dp"
        android:right="7dp"
        android:top="0dp" />

    <corners android:radius="4dp" />

</shape>
```

5.5 Interfaz de usuario

Seleccionar Rol

En primer lugar, se establece la pantalla de inicio, donde se elegirá el rol con el que acceder a la aplicación. Este dato es imprescindible, pues la plataforma web subyacente trata a ambos roles de manera diferente y la autenticación se realiza a través de dos URL diferentes.

La selección de uno de los dos roles llevará a continuación hacia la pantalla de autenticación del usuario.



Imagen 30: Seleccionar rol

Autenticación

Esta pantalla guiará al usuario a la actividad encargada de recibir la información del usuario. Asimismo, permitirá guardar y recordar los datos introducidos previamente como funcionalidad.

Una vez introducidos los datos, la aplicación lanzará la petición HTTP Post, con los parámetros introducidos por el usuario, para a continuación realizar una petición HTTP Get. La aplicación comprobará el contenido de la entidad devuelta a través de la HTTP Get para verificar si el usuario ha hecho login o no.



Imagen 31: Autenticación

Seleccionar tarea

En el caso de que el usuario sea un proponente de títulos propios, se le mostrará una nueva pantalla de selección de tarea. En este caso, el usuario seleccionará su siguiente acción entre dos opciones. Estas opciones son 'Crear nueva propuesta' y 'Explorar propuestas existentes'.



Imagen 32: Seleccionar tarea

Crear nueva propuesta

Al iniciar esta pantalla, el sistema descarga el recurso `periods.xml` y lo procesa. Esta acción por parte de la aplicación trata de asegurar que en la aplicación web, el periodo de creación de nuevas propuestas se encuentra abierto. En caso negativo, la aplicación muestra un mensaje indicativo de este error y regresa atrás en la jerarquía de actividades.

En caso positivo, se le presenta al usuario una serie de campos obligatorios que rellenar para completar el borrador de la nueva propuesta de título. Al seleccionar alguno de los items del menú desplegable correspondiente al centro adscrito, la información sobre la sede de la organización se rellenará automáticamente con la información contenida de ese centro en la base de datos de la aplicación web. No obstante, se podrá editar si la dirección es diferente a la dirección de la sede de la organización.

En el momento en el que el usuario pulsa el botón enviar, la aplicación móvil verificará que todos los campos son distintos de la ristra vacía. A continuación, verificará que el DNI del proponente introducido es de un profesor dado de alta en la aplicación web y además es profesor de la ULPGC. Ésto se logra haciendo una búsqueda de profesores con el único parámetro del DNI introducido. El documento XML retornado determinará la existencia o no y el tipo de profesor propuesto.

Gestor TTPP

Denominación

Item 1
Sub Item 1

DNI de Director Académico

Proponente

Unidades participantes

Plazas mínimas

Plazas máximas

Mínimo de créditos anuales

Gestor económico

Competencias

Centro adscrito

Item 1
Sub Item 1

Sede de la organización

Sede de la organización

Título impartido fuera de las Islas Canarias

Enviar

Imagen 33: Crear nueva propuesta

Búsqueda de propuestas

En esta actividad, el usuario podrá introducir diferentes criterios de búsqueda para explorar las propuestas a las que tiene acceso según sus permisos. Estos criterios incluyen los tipos de título, la rama de conocimiento y el estado. Asimismo, admitirá subestructuras correspondientes a la denominación, así como su código generado.

En el caso de que el usuario no seleccione ningún criterio, se mostrarán todas las propuestas disponibles.

La búsqueda se produce mediante una petición HTTP Get, incluyendo en la URL solicitada los parámetros que el usuario ha introducido. La aplicación web devolverá un único documento XML con los títulos que cumplen con los criterios de búsqueda. No obstante, la aplicación móvil ha de comprobar si el documento contiene 20 instancias de títulos. Esta circunstancia hace que sea necesario pedir un segundo documentoXML con la siguiente página de propuestas.

Cuando la aplicación móvil ha terminado de descargar y contabilizar las propuestas disponibles, muestra un diálogo con el número de propuestas y dos opciones. Una de las opciones será repetir la búsqueda con otros criterios, siendo la otra la opción de mostrar el listado de propuestas que ha retornado la búsqueda.

Resultado de la búsqueda

Una vez aceptado el set de resultados de la búsqueda realizada en la actividad anterior, se presentará la lista de propuestas que cumplen con los parámetros de búsqueda, mostrando su información básica y permitiendo el acceso a la información de cada una de ellas a través de la actividad que se inicia al pulsar sobre la denominación del ítem correspondiente propuesta de la lista generada.

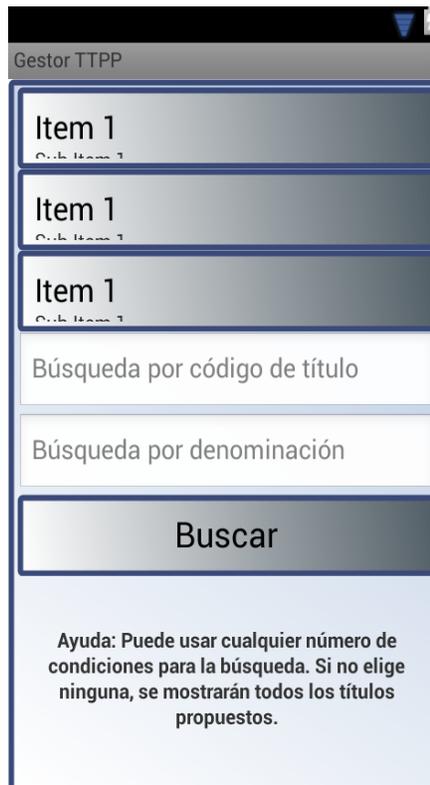


Imagen 34: Búsqueda de propuestas



Imagen 35: Resultado de la búsqueda

Menú principal

En esta actividad se representan 7 acciones que realizar sobre la propuesta de título que se escogió explorar en la actividad anterior. Esta actividad, en el caso de los dispositivos móviles *handheld*, actuará de control para iniciar el resto de actividades. En el caso de los dispositivos tableta, esta actividad actuará de mecanismo de control para la sustitución de los fragmentos correspondientes a cada una de las funcionalidades presentadas, excepto para la opción de descarga de la memoria en PDF, que se ejecutará en esta misma actividad.

Esta acción de descarga de la memoria mostrará un diálogo que hace el seguimiento de la cantidad de bytes descargados hasta el momento. Cuando finaliza la descarga, la aplicación web buscará la aplicación instalada en el dispositivo encargada de leer los documentos PDF y la abrirá automáticamente. Si no la encuentra, depositará la memoria en la carpeta de descargas por defecto y mostrará un mensaje de error relacionado con la inexistencia de la aplicación encargada de visualizar documentos PDF.

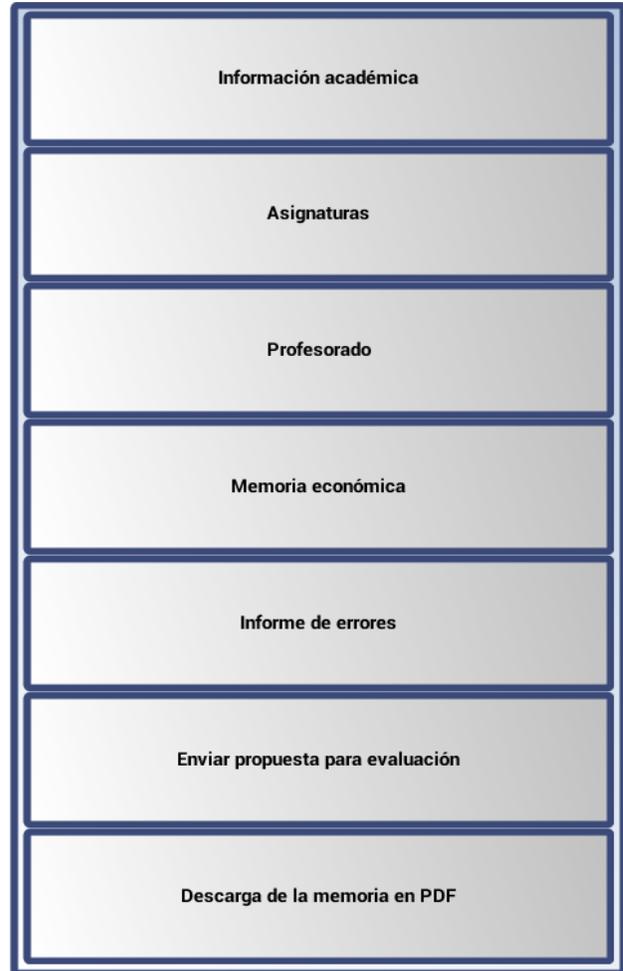


Imagen 36: Menú principal

Información general

El primer enlace del menú principal procesa la información general de la propuesta y la muestra de forma ordenada en columnas de a dos. Esta información describe las características del título y su información básica. Debido a que varios de sus campos de datos contienen un identificador numérico en vez de una versión válida para la lectura, se realiza un preprocesado de los campos 'Rama de conocimiento' y 'Tipo de enseñanza'.

Información General	
Denominación: Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	
Rama de Conocimiento: Arte y humanidades	Tipo de enseñanza: Experto
Estado de la propuesta: subsanacion	Número de edición: 2
Código: TP3-1-059-02	Proponente: Aula Manuel Alemán. José Alonso Morales e Isabel Luján Henríquez
Inicio de las enseñanzas: 2012-10-03	Fin de las enseñanzas: 2013-06-18
Unidades participantes: Vicerrectorado de Cultura y Deporte	Lugar de impartición: Escuela de Arquitectura
Gestor: Centro de Formación Continua	Representante del centro: Daniel Barreto González
Sede de la organización: Edificio Arquitectura CAMPUS UNIVERSITARIO 35017 - Las Palmas de Gran Canaria	Evaluado por acecau en: Sin especificar
Competencias: Este programa intenta dotar a los que se beneficien de sus aportaciones con las siguientes competencias: - Tener una visión reflexiva de problemas y situaciones de la actualidad desde criterios éticos y filosóficos - Introducir y profundizar en el pensamiento crítico y en la confrontación entre posturas diversas y encontradas. - Ofrecer claves de comprensión de los fenómenos culturales contemporáneos. - Familiarizarse y adiestrarse en elementos de comprensión de los textos de pensadores y filósofos especialmente del mundo actual. - Preparar, en un mundo plural y globalizado, para un diálogo intercultural en los diferentes ámbitos de la sociedad. - Dotar de elementos para analizar los comportamientos y las culturas con las que se entra en contacto en una realidad en que la inmigración aumenta progresivamente. - Tomar postura crítica frente a un pensamiento estrictamente técnico y manipulado desde poderes que dominan nuestra sociedad. - Crear conciencia social y política frente a situaciones y problemas que acaecen en nuestra realidad cercana y mundial. - Dotar de formación específica para la gestión y la política cultural	
Justificación de competencias: Este programa intenta dotar a los que se beneficien de sus aportaciones con las siguientes competencias: - Tener una visión reflexiva de problemas y situaciones de la actualidad desde criterios éticos y filosóficos - Introducir y profundizar en el pensamiento crítico y en la confrontación entre posturas diversas y encontradas. - Ofrecer claves de comprensión de los fenómenos culturales contemporáneos. - Familiarizarse y adiestrarse en elementos de comprensión de los textos de pensadores y filósofos especialmente del mundo actual. - Preparar, en un mundo plural y globalizado, para un diálogo intercultural en los diferentes ámbitos de la sociedad. - Dotar de	

Imagen 37: Información general

Asignaturas

El segundo enlace del menú principal generará una nueva lista, si hubiera asignaturas cargadas en la propuesta, con la información básica de cada asignatura y un enlace sobre su nombre para acceder a la información académica completa en una nueva actividad.

Asignaturas	
Bioética. Un puente entre ciencia y valores	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio: 2012-10-10	Fin: 2012-11-14
Biología y conciencia	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio: Sin especificar	Fin: Sin especificar
Ciencia, Tecnología y Humanismo.	
Curso: 1	Semestre: 1
13/13 asignaturas cargadas	

Imagen 38: Asignaturas

Asignatura

Al pulsar en una de las asignaturas del listado de asignaturas, se mostrará la información completa relacionada con dicha asignatura, incluyendo la información general completa, el reparto de horas, competencias, actividades formativas, descripción de los contenidos y el sistema de evaluación.

Bioética. Un puente entre ciencia y valores	
Nombre: Bioética. Un puente entre ciencia y valores	
Información general	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio de docencia: 2012-10-10	Fin de docencia: 2012-11-14
Rama de conocimiento: Arte y humanidades	Modalidad: Presencial
Creado: 2011-06-20 09:30:27	Actualizado: 2012-02-13 09:42:40
Reparto de horas	
Clases teóricas: 4.0 horas	Clases prácticas: 2.0 horas
Actividades docentes complementarias: 4.5 horas	Prácticas en empresas/instituciones: 0.0 horas
Pruebas de evaluación: 2.0 horas	Trabajo final: 0.0 horas
Dedicadas al estudio: 12.5 horas	Número de horas en total: 25.0 horas
<p>Competencias: El primero en hablar de Bioética, Van Rensselaer Potter (Madison, 1971) usó el término en un sentido amplio, queriendo significar el puente entre dos culturas: de un lado el conocimiento científico biológico, del otro los valores humanos. Para él la bioética es "Bridge to the Future", puente de encuentro entre los hechos que presentan las ciencias de la vida y el mundo del valor. Se intenta conseguir: - Aproximación a la comprensión de la Bioética y sus implicaciones - Acercamiento a los grandes debates de actualidad bioética, desde una visión interdisciplinar. - Introducción de la Bioética como posible "estilo de vida" desde la humanización y el cuidado</p>	
<p>Actividades formativas: Exposición de temas por parte de la profesora. Análisis de casos prácticos desde los que se intentará cultivar la búsqueda común y el diálogo entre las diferentes visiones que puedan surgir en el grupo</p>	
Descripción de los contenidos:	
<p>Sistema de evaluación: Evaluación continua que tendrá en cuenta la asistencia, participación y análisis de casos - Entrega de una Memoria de lo trabajado.</p>	

Imagen 39: Asignatura

Profesorado

Cuando se pulsa la tercera opción del menú principal, la aplicación descarga el recurso `degree_teachers.xml`. Este recurso no es suficiente para mostrar la información del profesor, puesto que sólo se refiere a la información sobre la participación. Para obtener la información completa del profesor hay que procesar el documento descargado en busca de la Id única del profesor y descargar su recurso xml concreto, una petición HTTP Get por cada uno de los profesores. Con esta información descargada se genera un listado y se almacena el resto de la información del profesor a la espera de si se va a acceder a la información completa del profesor al pulsar en su nombre.

Debido a la necesidad de descargar un recurso previo a descargar la información completa del profesor, la carga de la lista se hará asíncronamente

Asimismo, junto al botón situado en el nombre del profesor, se situará la opción de remitir un correo de confirmación al profesor en el caso de que su estado se encuentre pendiente de confirmar participación. Esta acción no requerirá el inicio de una nueva actividad o fragmento. Se ejecutará en esta misma actividad, creando un diálogo que se descartará cuando la notificación haya sido solicitada a la plataforma web. Esta petición se realiza haciendo un HTTP Get sobre la URL del profesor, agregando al final `/request confirmation`

Profesor

Cuando se pulsa sobre el nombre del profesor, se iniciará otra actividad o fragmento que mostrará la información completa existente en el perfil del profesor propuesto.

Memoria económica

La cuarta opción del menú principal se refiere al tratamiento y exposición del recurso `economic_report.xml`. Este documento es significativamente más largo que cualquier otro que se trata en este proyecto. Por esta razón, esta vista está dividida en tres secciones. Balance, Ingresos y Gastos.



Imagen 40: Profesorado

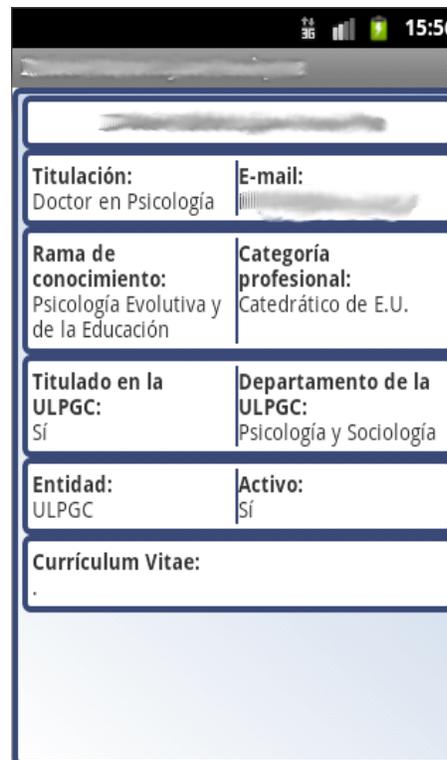


Imagen 41: Profesor

Balance

Resumen Ingresos/Gastos	
Ingresos totales	9000.0€
Suma de gastos parcial	7645.0€
Compensación a la ULPGC (5%)	450.0€
Compensación al centro (5%)	450.0€
Compensación al gestor económico (5%)	450.0€
Total de Gastos	8995.0€
Balance	5.0€

Imagen 42: Memoria económica. Balance

Ingresos

Ingresos		
Derivados de las matrículas		
Número de estudiantes becados	Precio de matrícula	Total
3	300.0	900.0 €
Número de estudiantes sin beca	Precio de matrícula	Total
12	600.0	7200.0 €
Ingresos por matrículas		8100.0 €
Directos y subvenciones		
Total		900.0 €

Imagen 43: Memoria económica. Ingresos

Gastos

La vista de gastos es la más extensa de toda la aplicación, requiriendo más de cuatro veces más espacio que el disponible en una tableta en modo handheld. Se adjunta una única captura de pantalla de su sección inicial.

Gastos		
Profesorado ULPGC		
Número de horas de docencia 84.0	Coste por hora 10.0 €	Total 840.0 €
Número de horas de seminario 89.5	Coste por hora 10.0 €	Total 895.0 €
Número de horas de prácticas 47.0	Coste por hora 10.0 €	Total 470.0 €
Número de horas de evaluación 42.0	Coste por hora 10.0 €	Total 420.0 €
Elaboración de manuales docentes 0.0	Coste por crédito 0.0 €	Total 0.0 €
Tutorías presenciales y de trabajos finales	Coste por proyecto	Total 0.0 €
Tribunales para valoración de trabajos finales	Coste por tribunal	Total 0.0 €
Coordinación de exámenes 0	Coste por examen 0.0 €	Total 0.0 €
Directores		Coste
Coste de profesorado de la ULPGC		Coste 2625.0 €
Beneficios para la unidad organizadora		Coste
Total retribución bruta a profesorado de la ULPGC		Total

Imagen 44: Memoria económica. Gastos

Reporte de errores

En la quinta opción del menú principal se muestra, si lo hubiera, el último reporte de errores que la plataforma ha generado cuando se ha tratado de enviar la propuesta. Se listan las razones por las que la propuesta no ha pasado directamente al estado 'Enviado', estado en el que se establecen las propuestas ya preparadas para ser evaluadas por las diferentes comisiones.

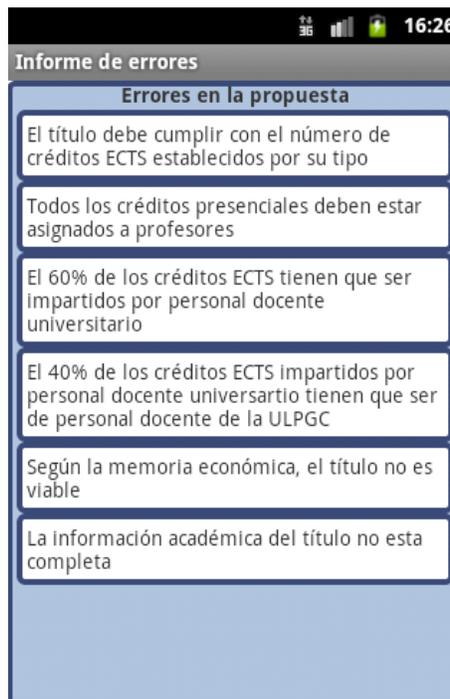


Imagen 45: Informe de errores

Enviar propuesta para evaluación

En la sexta opción del menú principal se provee la opción de remitir la propuesta actual siendo revisada a ser evaluada por la aplicación web. Como control previo, la aplicación móvil descarga el recurso periods.xml para comprobar qué plazos están abiertos, sea periodo de petición de propuestas, primera subsanación o segunda subsanación. Si ninguno de estos plazos están abiertos o el curso académico no es el correcto, la aplicación móvil avisará al usuario de esta circunstancia y no remitirá la propuesta a evaluar.

En el caso en el que los plazos sean correctos, se remite la propuesta a evaluar y se redirige al usuario a la actividad anterior, reporte de errores, para comprobar cuáles son, si hubieran, los errores que han impedido que la propuesta pase a estado 'Enviado'.

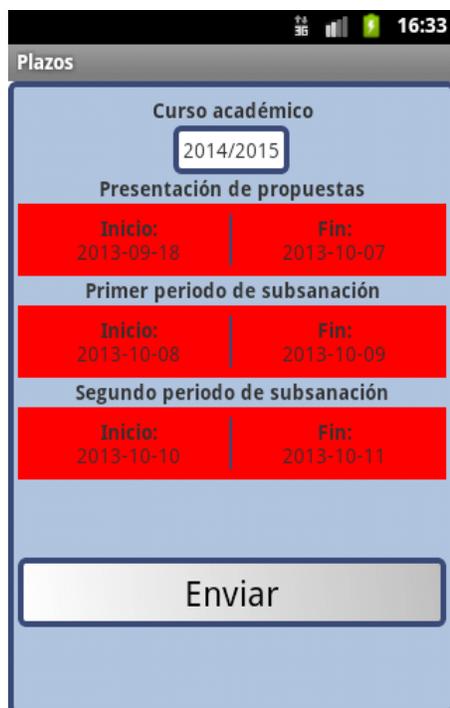


Imagen 46: Enviar propuesta para evaluación

Descarga de la memoria en PDF

Esta operación no implica un cambio de actividad, por lo que se ejecuta la operación de descarga en la actividad del menú principal, así como la apertura de la memoria descargada por parte de la aplicación establecida por defecto para la lectura de documentos PDF.

5.6 Diferencias de layout en versión *tableta*

A continuación se adjuntarán las vistas que se comportan de manera diferente en el caso de que la aplicación móvil se esté ejecutando en versión *landscape* en una tableta.

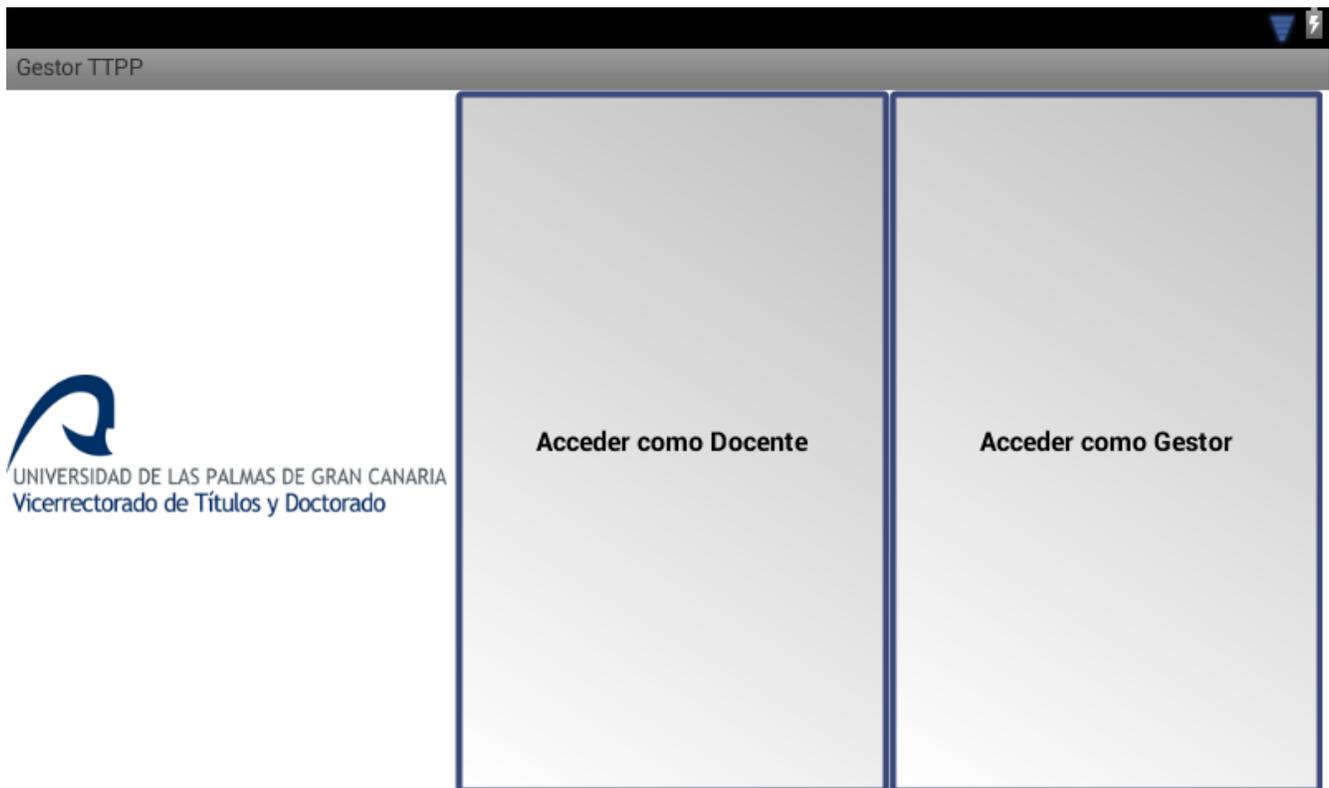
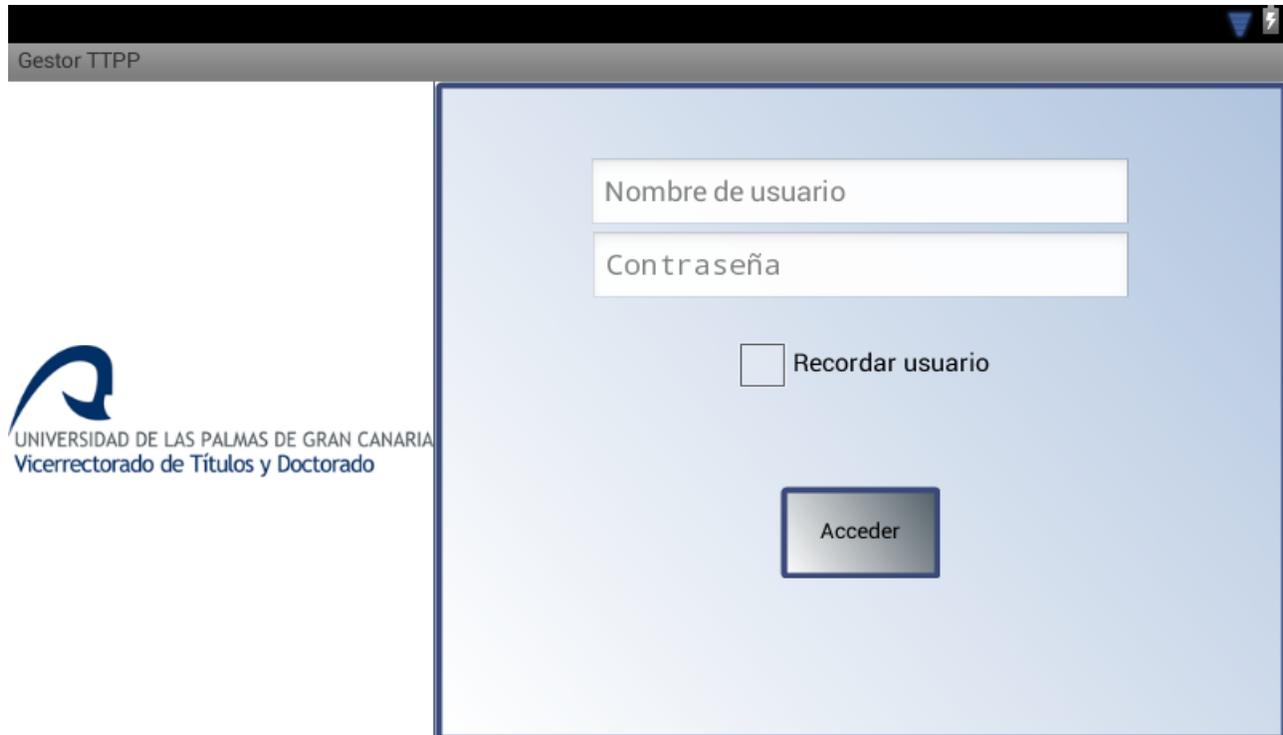


Imagen 47: Seleccionar Rol. Tableta

Seleccionar Rol. Tableta

Autenticar. Tableta



The image shows a tablet interface for a login system. At the top, a dark grey header bar contains the text "Gestor TTPP" on the left and a small blue triangle icon on the right. Below the header, the main content area has a light blue background. On the left side of this area, there is a logo consisting of a blue stylized 'U' shape, followed by the text "UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA" and "Vicerrectorado de Títulos y Doctorado" in a smaller font. In the center-right of the main area, there is a login form with two white input fields: the top one is labeled "Nombre de usuario" and the bottom one is labeled "Contraseña". Below these fields is a checkbox labeled "Recordar usuario". At the bottom center of the form area is a grey button with the text "Acceder".

Imagen 48: Autenticar. Tableta

Búsqueda de propuestas. Resultado de la búsqueda. Tableta

The screenshot shows a tablet application interface for searching proposals. The title bar at the top reads "Búsqueda de propuestas" and includes a status bar with the time "16:56".

Search Filter Section (Left):

- Tipo de título
- Año de inicio
- Estado del título
- Búsqueda por código de título (highlighted with an orange border)
- Búsqueda por denominación
- Buscar (button)
- Ayuda: Puede usar cualquier número de condiciones para la búsqueda. Si no elige ninguna, se mostrarán todos los títulos propuestos.

Search Results Section (Right):

Three results are displayed, each with a title and two data fields:

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	
Experto Universitario	Estado: En subsanación
Inicio: 2012-10-03	Código: TP3-1-059-02

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	
Experto Universitario	Estado: Enviado
Inicio: 2013-10-21	Código: TP3-1-059-03

Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	
Experto Universitario	Estado: Enviado
Inicio: 2011-10-03	Código: TP3-1-059-01

Imagen 49: Búsqueda de propuestas. Resultado de la búsqueda. Tableta

Información general. Tableta

Menú principal	
Información académica	Denominación: Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos
Asignaturas	Rama de Conocimiento: Arte y humanidades Tipo de enseñanza: Experto
Profesorado	Estado de la propuesta: subsanacion Número de edición: 2
Memoria económica	Código: TP3-1-059-02 Proponente: Aula Manuel Alemán, José Alonso Morales e Isabel Luján Henríquez
Informe de errores	Inicio de las enseñanzas: 2012-10-03 Fin de las enseñanzas: 2013-06-18
Enviar propuesta para evaluación	Unidades participantes: Vicerrectorado de Cultura y Deporte Lugar de impartición: Escuela de Arquitectura
Descarga de la memoria en PDF	Gestor: Centro de Formación Continua Representante del centro: Daniel Barreto González
	Sede de la organización: Edificio Arquitectura CAMPUS UNIVERSITARIO 35017 - Las Palmas de Gran Canaria Evaluado por acecau en: Sin especificar
	Competencias: Este programa intenta dotar a los que se beneficien de sus aportaciones con las siguientes competencias: - Tener una visión reflexiva de problemas y situaciones de la actualidad desde criterios éticos y filosóficos - Introducir y profundizar en el pensamiento crítico y en la confrontación entre posturas diversas y encontradas. - Ofrecer claves de comprensión de los

Imagen 50: Información general. Tableta

Listado de asignaturas. Asignatura. Tableta

Bioética. Un puente entre ciencia y valores		
Información académica	Bioética. Un puente entre ciencia y valores	
	Curso: 1	Semestre: 1
Asignaturas	Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
	Inicio: 2012-10-10	Fin: 2012-11-14
Profesorado	Biología y conciencia	
Memoria económica	Curso: 1	Semestre: 1
	Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Informe de errores	Inicio: Sin especificar	Fin: Sin especificar
	Ciencia, Tecnología y Humanismo.	
Enviar propuesta para evaluación	Curso: 1	Semestre: 1
	Tipo de asignatura: Obligatoria	Créditos ECTS: 2.0
Descarga de la memoria en PDF	13/13 asignaturas cargadas	
	Nombre: Bioética. Un puente entre ciencia y valores	
Información general		
Curso: 1		Semestre: 1
Tipo de asignatura: Obligatoria		Créditos ECTS: 1.0
Inicio de docencia: 2012-10-10	Fin de docencia: 2012-11-14	
Rama de conocimiento: Arte y humanidades	Modalidad: Presencial	
Creado: 2011-06-20 09:30:27	Actualizado: 2012-02-13 09:42:40	
Reparto de horas		
Clases teóricas: 4.0 horas	Clases prácticas: 2.0 horas	
Actividades docentes complementarias: 4.5 horas	Prácticas en empresas/instituciones: 0.0 horas	

Imagen 51: Listado de asignaturas. Asignatura. Tableta

Memoria económica. Balance. Tableta

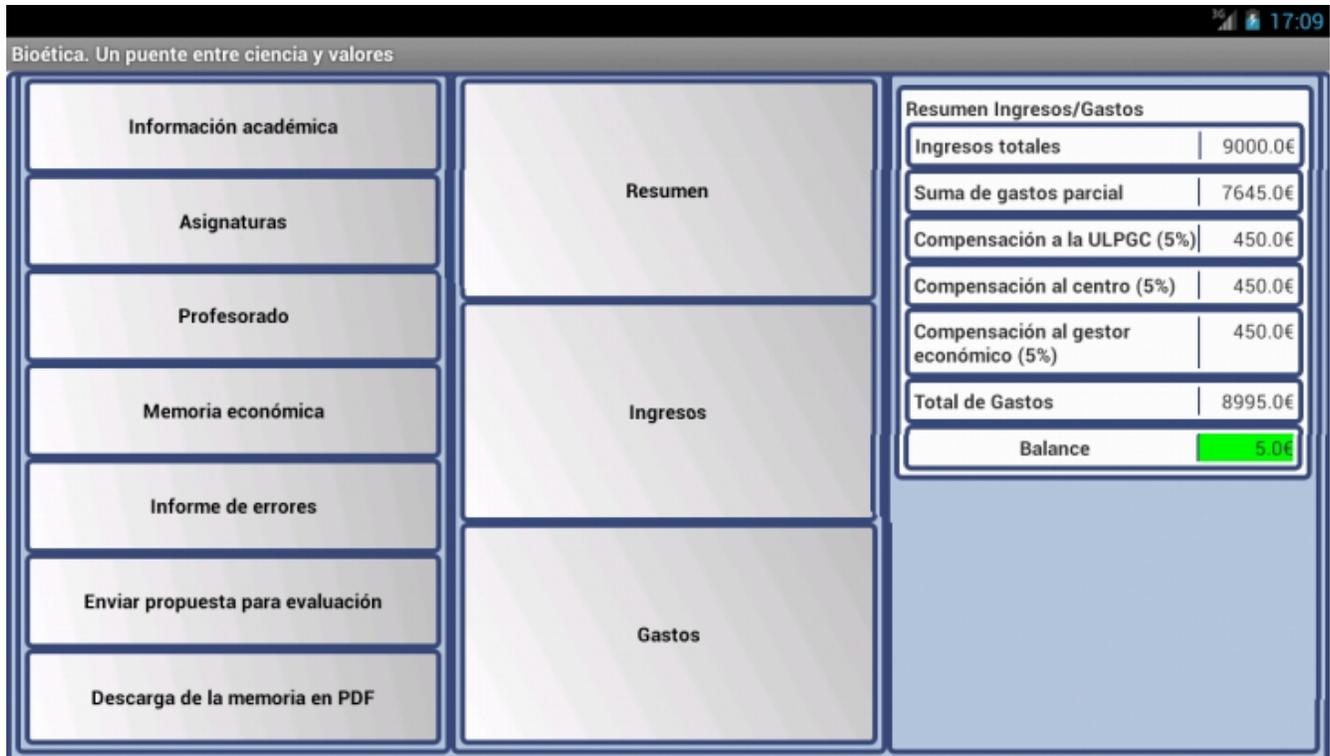


Imagen 52: Memoria económica. Balance. Tableta

6 Temporización

Como se ha comentado con anterioridad, el proyecto se ha desarrollado en cuatro iteraciones incrementales. No obstante, es de observar que una gran parte de las etapas sucesivas de planificación y requisitos, así como de análisis y diseño, se produjeron fuera del ámbito temporal del presente proyecto. Esto es debido a que el autor de este proyecto fue parte del equipo encargado de la extracción de requisitos del reglamento del que emanan las restricciones y necesidades afrontadas en el proceso automatizado de creación y modificación de propuestas de títulos propios en colaboración con el equipo del vicerrectorado y la empresa encargada de la implementación, *Banot*.

Debido a esta razón, las etapas mencionadas tienen un tamaño inusualmente inferior al que se le debería dedicar en condiciones de desarrollo normal.

La temporización se dividirá en iteraciones, así como una etapa adicional de generación de documentación.

Primera iteración

En esta primera iteración, se hace recopilación de los diversos requisitos iniciales de la aplicación. A continuación, se ejecuta la etapa de análisis y diseño. Debido al escaso conocimiento de la arquitectura Android, una pobre ejecución de esta última etapa afectará negativamente al tiempo dedicado a la implementación de esta etapa.

La prueba y evaluación se ve reducida, puesto que los errores eran numerosos y fácilmente identificables

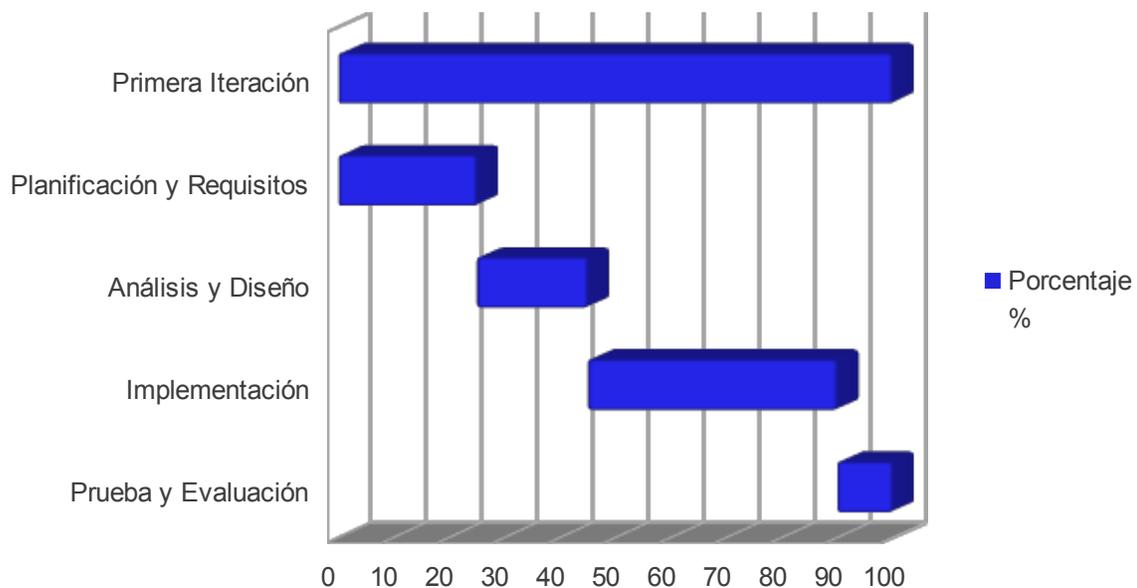


Imagen 53: Temporización. Iteración 1

Segunda iteración

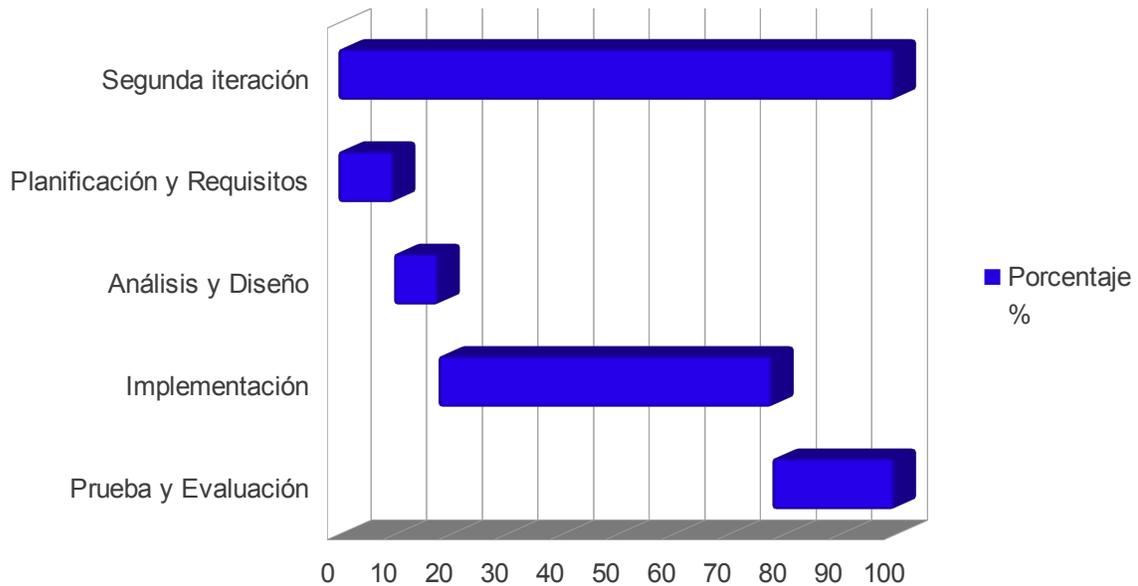


Imagen 54: Temporización. Segunda iteración

En la segunda iteración, las dos primeras etapas siguen el comportamiento anteriormente mencionado, no teniendo demasiado peso en el porcentaje de tiempo dedicado a su definición. No obstante, la etapa de implementación sigue teniendo un peso determinante en la iteración. Esto es debido a la reorganización completa de los *layouts*, trasladándolos siempre que fuera posible a las definiciones *XML* en el directorio */layouts* del proyecto.

La etapa de prueba y evaluación aumenta de peso con el objetivo de determinar la razón de la inconsistencia y poca fiabilidad en las peticiones HTTP hechas contra el servidor.

Tercera iteración

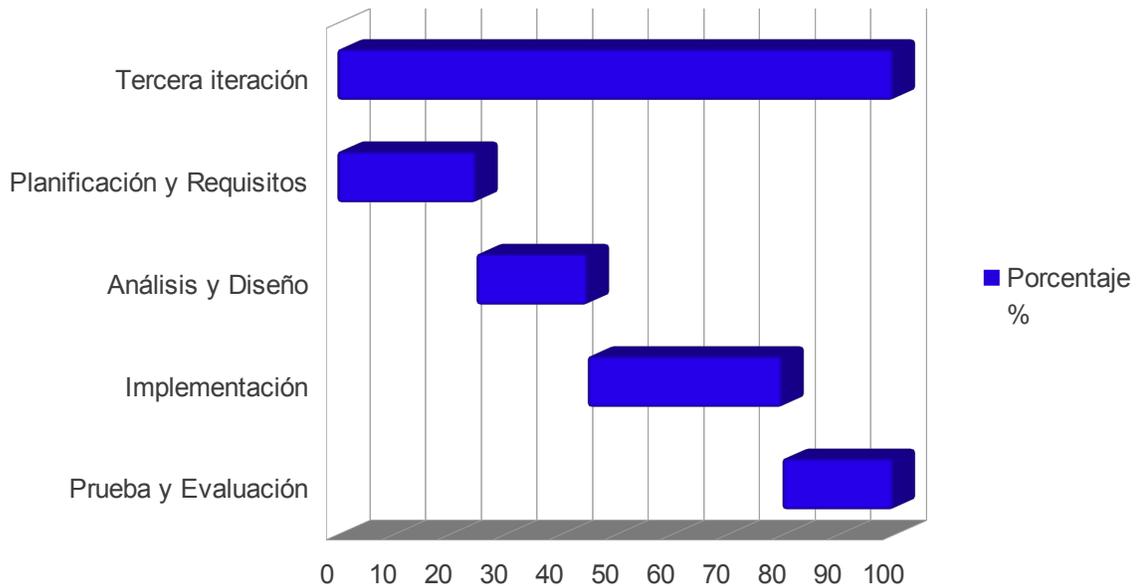


Imagen 55: Temporización. Tercera iteración

En la tercera iteración, se descubre que la plataforma web no proporciona todos los datos necesarios para completar la aplicación, necesitando pequeñas modificaciones en la forma en la que response a las peticiones de la aplicación móvil. Esto provoca que la etapa de planificación y requisitos sea más importante en esta iteración, así como la etapa de análisis y diseño, ya que fue necesario localizar y modificar el código objetivo, así como aguardar a que los desarrolladores de la plataforma web desplegaran el parche propuesto. Esta espera fue la más importante demora que este proyecto sufrió a lo largo de todo su desarrollo.

La etapa de implantación sigue teniendo el mayor peso entre todas las etapas, puesto que fue en esta iteración donde prácticamente se completó el número de actividades que la aplicación móvil iba a ser capaz de mostrar.

En la etapa de prueba y evaluación se tuvo que dedicar tiempo, puesto que fue en este momento el paso de las pruebas desde el servidor Rails local al servidor de pruebas en línea. El código local se había quedado obsoleto debido a una actualización del desarrollo de la aplicación principal.

Cuarta iteración

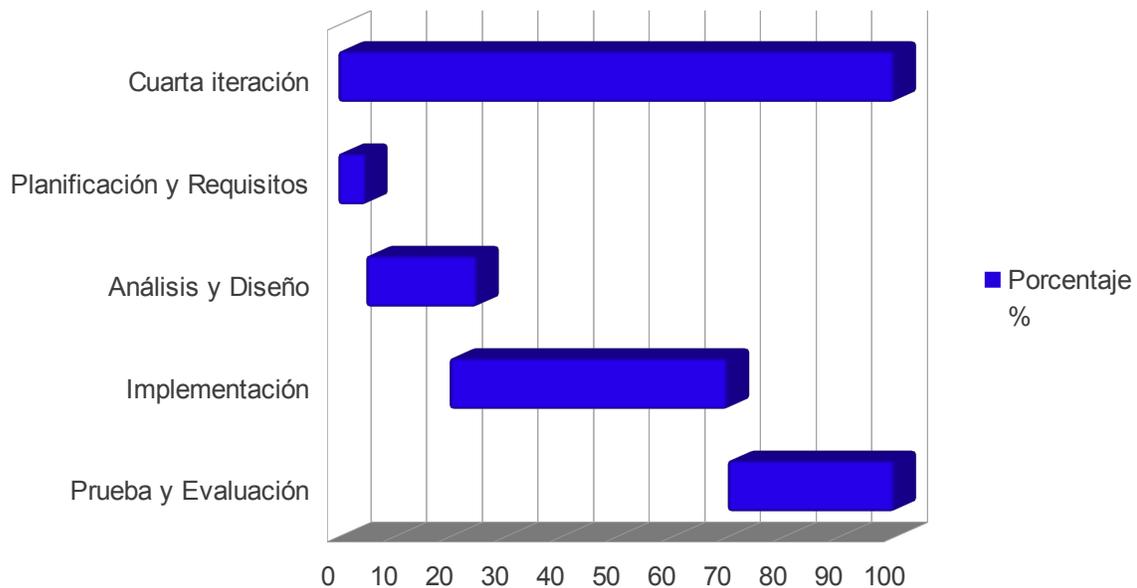


Imagen 56: Temporización. Cuarta iteración

En la última iteración, la primera etapa pierde importancia puesto que se está cerca del final del desarrollo y se tienen los objetivos de lo que falla, falta o necesita la aplicación móvil bastante mejor definidos. No obstante, la etapa de análisis y diseño necesita bastante trabajo para modificar la estructura de control de las actividades para adaptarlas al nuevo paradigma de fragmentos reutilizables, con el objetivo de adaptar el código a las tabletas.

La etapa de implantación, como ha sido habitual en todas las iteraciones, se lleva la mayor parte del trabajo. En esta etapa se decide extender las clases básicas de *FragmentActivity* para un control más fiable de las interrupciones de Entrada/Salida, así como evitar la generación de filtraciones de contexto en la creación de ventanas procedentes de diferentes actividades.

La etapa de prueba y evaluación cobra mayor peso en esta última iteración, puesto que se trata de la versión definitiva que quedará para uso público de los proponentes de títulos propios.

Temporización global de iteraciones

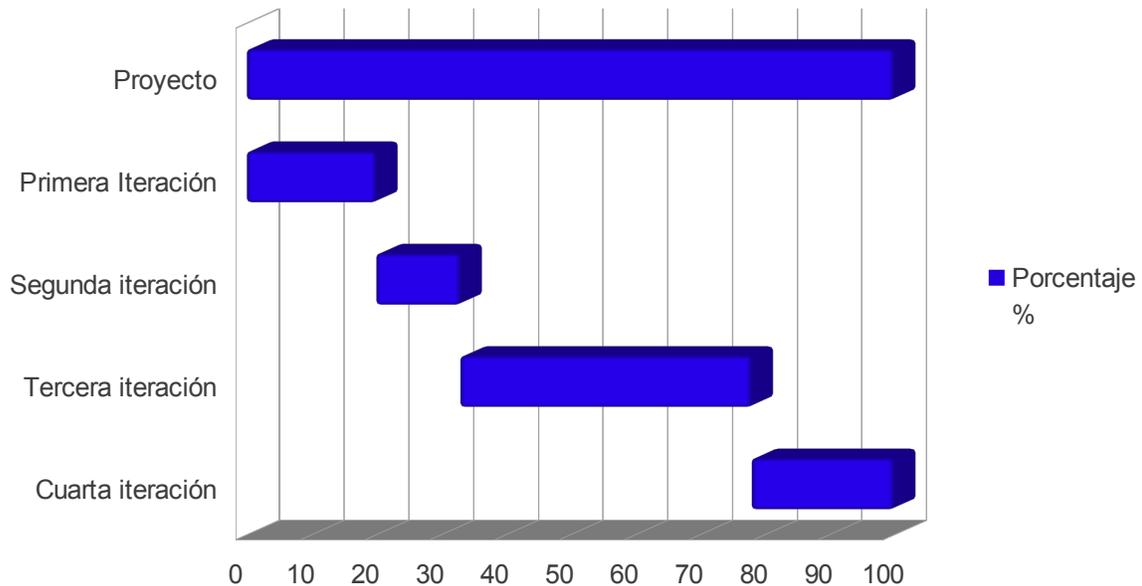


Imagen 57: Temporización. Cuarta Iteración

En este último diagrama, se observa claramente un peso desproporcionado en el tiempo dedicado a la tercera iteración, debido a la dependencia de terceras personas en la capacidad de desarrollo. Las modificaciones solicitadas a los desarrolladores del servidor de datos tomaron meses, puesto que no había obligación por su parte para realizarlas.

La cuarta iteración, a pesar de tener una gran cantidad de modificaciones, como por ejemplo la reorganización del control de la aplicación, así como la transición de *Activity* a *FragmentActivity/Fragment* y el control de interrupciones, no tuvo tanto peso comparativo como las anteriores etapas debido a adquisición de conocimientos y soltura a la hora de desarrollar en Java/Android, que permitió un desarrollo más fluido y ágil.

Temporización por fases de desarrollo

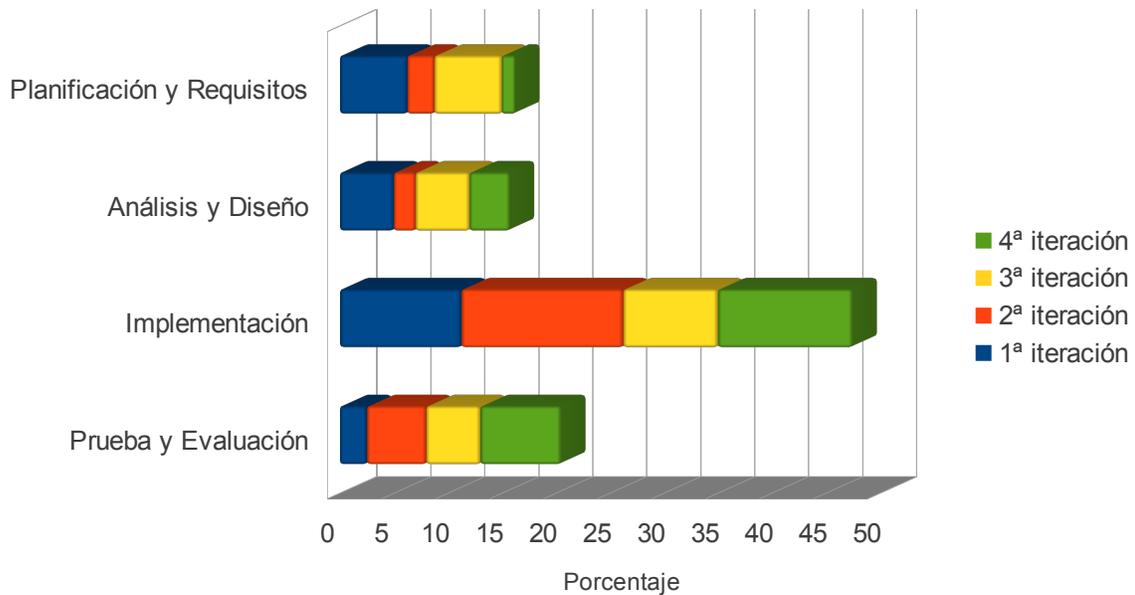


Imagen 58: Temporización. Fases de desarrollo

Temporización en horas	1ª iteración	2ª iteración	3ª iteración	4ª iteración	Total
Planificación y requisitos	45	11.7	101.25	9.9	167.85 horas
Análisis y Diseño	36	9.36	81	29.7	156.06 horas
Implementación	81	70.2	141.75	99	391.95 horas
Prueba y evaluación	18	25.74	81	59.4	184.14 horas
Total					900 horas

7 Prueba y Mantenimiento

7.1 Prueba

El mecanismo para las pruebas de la aplicación han sido los siguientes:

En primer lugar, con la instalación de la plataforma web en el equipo local, se realizan las pruebas que corresponden a las dos primeras iteraciones de desarrollo de la aplicación móvil. Al ser el código cedido del servidor una mera instantánea entre versiones completas de producción de la plataforma web, el uso de este servidor local se ve limitado cuando se ha de implementar los cambios propuestos al servidor de producción.

En la tercera iteración, se pasa utilizar como plataforma de pruebas un servidor de desarrollo que los responsables de la aplicación web ponen a disposición con el objetivo de hacer los cambios propuestos al código Ruby on Rails necesarios para continuar el progreso del proyecto.

En la cuarta iteración, ya se realizan todas las pruebas en el servidor web de producción, ya que se deben realizar pruebas de rendimiento utilizando tipos de conectividad dentro y fuera de la red de la ULPGC que se correspondan más al uso que se le va a dar a la aplicación web una vez se encuentre desplegada.

Las pruebas relacionadas con la interfaz de usuario y el comportamiento de la aplicación se realizan sobre un dispositivo android Samsung Galaxy Ace 2 con versión de Android 2.3.6.

Se encarga a una tercera persona realizar pruebas de stress de la aplicación, forzando los casos de uso e intentando la detección de casos extremos.

7.2 Mantenimiento

Se establece que el periodo de mantenimiento una vez desplegada la aplicación y puesta a disposición del público deberá cubrir el primer periodo de solicitud de propuestas para el curso académico 2014/2015. Este periodo estará abierto, previsiblemente, entre Diciembre de 2013 y Mayo de 2014, contando con sus correspondientes periodos de subsanación de propuestas. Durante este tiempo, se recibirán las incidencias a través de correo electrónico o vía telefónica y se tratará de solucionar los errores en un razonablemente corto intervalo de tiempo.

Es relevante mencionar que la dirección del Vicerrectorado de Títulos y Doctorado se encuentra en las primeras etapas del proceso de elaboración de un nuevo reglamento de títulos propios. Esta circunstancia haría que la plataforma web en la que la aplicación móvil de este proyecto se apoya para su funcionamiento, necesitaría una modificación importante. La fecha de aprobación de este nuevo reglamento y el tiempo necesario para que la aplicación vuelva a encontrarse al día con respecto a la correlación de su funcionamiento con los requisitos necesarios es indeterminada.

Es posible que la aplicación móvil deje de funcionar como fue diseñada si la modificación a la plataforma web subyacente es muy significativa a nivel estructural.

8 Análisis de Costos

8.1 Recursos materiales

Para la realización del presente proyecto ha sido imprescindible un ordenador razonablemente moderno, con capacidad para ejecutar un entorno de desarrollo como Eclipse, así como un emulador y un navegador de internet.

Como elementos opcionales pero altamente recomendables, un dispositivo móvil con Android, así como una tableta con el mismo sistema operativo.

Concepto	Coste estimado
Ordenador personal	600 €
Móvil Android	200 €
Tableta Android	200 €
Total	1000 €

8.2 Recursos humanos

En el caso de los recursos humanos, se valorará el tiempo de desarrollo por parte de un programador (el autor del proyecto), así como una pequeña aportación en la función de prueba de la aplicación en la última parte del proyecto.

Se hace la estimación de horas del proyecto en función de la suma de la proporción de tiempo de las diferentes iteraciones, contando con un desarrollo total de unos 5 meses (aproximado) de programación con una carga de trabajo de 5 horas dedicadas al día. Las horas de desarrollo de un programador novel se estimarán en 20€ la hora. No obstante, el incremento final se puede considerar como el trabajo de un programador con cierta experiencia, aumentando el coste de la hora de desarrollo.

Concepto	Horas	Coste por hora	Coste estimado
Programador			
Primera iteración	180	20	3600 €
Segunda iteración	117	20	2340 €
Tercera iteración	405	20	8100 €
Cuarta iteración	198	30	5940 €
Total			19980 €

Concepto	Horas	Coste por hora (Coste 1 ^a /2 ^a /3 ^a iteración + Coste 4 ^a iteración)	Coste estimado
Programador			
Fase 1. Planificación y Requisitos	167.85 horas	157.95h*20€ + 9.9h*30€	3456 €
Fase 2. Análisis y Diseño	156.06 horas	126.36h*20€ + 29.7h*30€	3418.2 €
Fase 3. Implementación	391.95 horas	292.95h*20€ + 99h*30€	8829 €
Fase 4. Prueba y Evaluación	184.14 horas	124.74h*20€ + 59.4h*30€	4276.8 €
Total			19980 €

Concepto	Horas	Coste por hora	Coste estimado
Tester	20	20 €	400 €
			400 €

Finalmente, agregando el coste material con el coste en recursos humanos, se obtiene el coste estimado del desarrollo de la aplicación.

Concepto	Coste estimado
Recursos materiales	1000 €
Recursos humanos. Desarrollador	19980 €
Recursos humanos. Tester	400 €
Total	21380 €

9 Conclusiones

Las conclusiones principales que se extraen de la realización del proyecto son positivas, puesto que se han cumplido la práctica totalidad de los objetivos establecidos al inicio del mismo.

En primer lugar, se ha logrado la finalización de un proyecto planteado desde cero hasta la fase de despliegue, realizando el análisis y diseño del software de antemano en cada iteración, así como la familiarización con las técnicas de testing.

Asimismo, se ha logrado un cierto dominio de la programación orientada a objetos, más concretamente en el lenguaje Java y las técnicas asociadas a la programación de dispositivos móviles en arquitectura Android. En esta última tecnología, se ha trabajado con una amplia gama de APIs, desde las versiones 2.2 liberadas en Mayo de 2010 hasta las tecnologías más recientes.

La necesidad de desplegar un servidor de pruebas de la plataforma subyacente a la aplicación ha permitido el aprendizaje del despliegue de un servidor Ruby on Rails, con sus convenientes dependencias. En este proceso también fue necesario acoplar un gestor de bases de datos PostgreSQL.

La experiencia obtenida mediante la realización de la estimación de costos y planificación temporal durante el proyecto también ha sido valiosa como aprendizaje para futuros desarrollos.

Otro punto positivo que se puede extraer de la realización del proyecto es el diseño de la fase de prueba que ha permitido la familiarización con las técnicas habituales. Asimismo, el diseño de la fase de mantenimiento también provee de una experiencia relacionada con las fases finales del desarrollo, mantenimiento correctivo y solución de *bugs*.

La primera conclusión negativa evidente que se extrae es el pobre rendimiento que presenta el emulador para linux de android. Ha sido una tarea ardua de paciencia tener que depurar código para el emulador, principalmente para tabletas, puesto que la carga a nivel gráfico que el emulador pone sobre el PC que lo ejecuta es intensa.

No obstante, el desarrollo en Windows es más fluido en este sentido, puesto que el *plugin AVT* para esta plataforma incluye los paquetes *Intel Atom x86 System Image by Intel Corporation*, *Intel Hardware Accelerated Execution Manager* y *Hardware Accelerated Execution Manager*. Estas herramientas desarrolladas por Intel mejoran el rendimiento del emulador para esta plataforma de desarrollo.

Por otra parte, el hecho de haber empezado a desarrollar sin conocer las mejores prácticas de programación en Android reduce la velocidad de desarrollo y aumenta el número de errores de difícil solución si no se tiene en cuenta desde el inicio. Esto es debido a las especificidades de la arquitectura para el desarrollo en Android. En este momento, habiendo concluido el desarrollo de la aplicación, probablemente empezaría de una manera completamente diferente, centrando los esfuerzos en un esqueleto de clases nativas extendidas para un mayor dominio de los errores comunes mencionados anteriormente.

Al contrario de lo que es naturalmente inferible, haber empezado desde el desconocimiento total de la plataforma y el lenguaje, ha permitido conocer errores comunes de programación, problemas derivados de la arquitectura del sistema Android, como las filtraciones de contexto de ventana entre actividades, las características variables de la clase *AsyncTask* entre versiones de Android o el manejo

de excepciones integral. Este camino ha sido más tortuoso, pero ha proporcionado un conocimiento más profundo de las características de la programación en sistemas Android.

Otra conclusión obvia es que la dificultad de programar contra un sistema de caja negra es considerable. Sin posibilidad de modificar un sistema del que se alimenta de datos la aplicación web y sin haber sido diseñada desde un principio para recibir estas peticiones, complica en gran magnitud la capacidad del programador de realizar un producto funcional.

Esta circunstancia ha hecho imposible desarrollar una sección para el usuario profesor que fuera ajustada a los objetivos establecidos al principio del proyecto. Esto es debido exclusivamente a la falta de permiso del profesor para acceder a su propia docencia. No obstante, sin modificar la plataforma web subyacente, esto sólo era posible superarlo utilizando una cuenta de administrador oculta dentro del código de la aplicación. Esta solución no es recomendable por varias razones, la más importante de ellas es la seguridad.

10 Trabajos Futuros

Durante el desarrollo del proyecto han surgido varias ideas que proporcionarían mayor calidad y fiabilidad al software desarrollado, pero que excedían el ámbito del presente proyecto. De esta manera, quedarán establecidos como posibles trabajos futuros.

- **Sistema de caché.** Mecanismo en segundo plano que se ocupa de almacenar los documentos ya adquiridos, con el objeto de no repetir peticiones. Este mecanismo deberá tener en cuenta las especificidades y capacidades de almacenamiento de infinidad de dispositivos móviles, puesto que el código actual es compatible con las APIs posteriores al nivel 8 (2.2 Froyo). Asimismo, esta característica tendrá que tener su propio control de caducidad de las peticiones ya almacenadas, puesto que el sistema contra el que se realizan las peticiones sólo proporciona los cualificadores de 'última modificación' de manera conjunta al resto de información.
- **Layout.** El software presentado no está suficientemente maduro en el aspecto visual. Mediante un estudio de interfaz de usuario podría ser mejorado sensiblemente, así como incorporar de manera más integral los colores de la imagen corporativa de la UPGC a nivel de tema de Android.
- **Desarrollo orientado a APIs modernas.** Existen varios elementos de la interfaz de usuario que han sido liberados en APIs posteriores, que podrían hacer más agradable la experiencia de usuario, tales como *OptionsMenu*, manejo de pestañas, entre otros.
- **Modificación de la información.** Actualmente, a través de la aplicación móvil de este proyecto, no se puede agregar una asignatura o editarla. Esto es debido a que la aplicación fue concebida para dispositivos hand-held y no para tabletas, imposibilitando la redacción de largos textos requeridos para la creación de estas entidades nuevas. No obstante, la adaptación a tabletas realizada en las últimas fases de este proyecto permitiría la modificación de estos datos.

Otras consideraciones se pueden hacer con respecto a la velocidad de funcionamiento de la aplicación móvil que necesitan apoyo de la aplicación web desarrollada en Ruby on Rails de la que se alimenta.

- Proporcionar servicios sencillos que no impliquen un documento XML con una entidad completa. Esto mejoraría la agilidad de la aplicación y permitiría nuevas funcionalidades. Por ejemplo, proporcionar los datos de 'última-modificación', 'fecha-creación'. Con esta información previa se puede determinar si es necesario volver a descargar los bloques de información de cada entidad consultada.
- Agregar un profesor o asignarle docencia a este profesor. Actualmente este proceso inmiscuye a varias otras entidades y requiere varias peticiones y comprobaciones en tiempo real, que se realizan instantáneamente en la aplicación web, pero para la aplicación móvil resultaría en un proceso insufrible de inicio-parada. Si la aplicación web estuviera preparada para estas

circunstancias, podría simplificarse el proceso de manera que no fueran necesarias tantas transacciones entre la aplicación móvil y la plataforma web.

- Emisión de errores de verificación. Actualmente, fuera de la entidad *issues.xml* que muestra los errores generados por la propuesta una vez remitida para su evaluación, no hay una manera directa de obtener la razón de muchas de las comprobaciones que se realizan a la hora de crear una propuesta, crear un profesor, crear una asignatura, etc. Este hecho hace que sea necesario examinar la respuesta HTTP que se produce al intentar validar una petición, para a continuación *parsear* el documento HTML en busca de la razón del error.
- Proporcionar acceso a los usuarios profesores a su propia docencia.
- Proporcionar acceso a los usuarios profesores a su propio perfil, con el objetivo de modificarlo si así lo desean.
- Proporcionar acceso a otras funcionalidades presentes en la aplicación web, pero inaccesibles a través de los puntos de servicios REST. Entre éstas se encuentra la generación de informes para el Consejo Social, listado de precios, resumen.

Asimismo, la portabilidad del proyecto hacia otras plataformas minoritarias en el mercado de los dispositivos portátiles es también parte de los trabajos futuros posibles. Las posibilidades de portabilidad incluyen:

- iOS
- FirefoxOs
- Windows Phone

Finalmente, varias funcionalidades de la aplicación web se encuentran actualmente todavía en periodo de pruebas, por lo que serán puestas en funcionamiento en posteriores fechas. Entre estas funcionalidades que podrían ser tratadas en el futuro se encuentra:

- Departamentos
- Gestión de usuarios
- Suplantación de usuarios por parte del administrador

11 Bibliografía

Consejo de gobierno de la ULPGC. Reglamento de Títulos Propios, Formación Continua y Extensión Universitaria de la ULPGC.

Nudelman, Greg(2013): *Android design patterns. Interaction design solutions for developers*. John Wiley & Sons, Indianapolis.

Murphy, Mark L.(2009): *Android programming tutorials*. Pearson Education, Boston.

Meier, Reto(2010): *Professional Android 2 Application Development*. Wiley Publishing, Indianapolis

Bloch, Joshua (2008): *Effective Java. Second edition*. Pearson Education, Boston.

Stack Exchange. Stack Overflow public forum
<http://www.stackoverflow.com>

12 Anexo

12.1 *Código Java/Android*

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.HashMap;
4:
5: import android.os.Bundle;
6: import android.support.v4.app.Fragment;
7: import android.text.Html;
8: import android.view.LayoutInflater;
9: import android.view.View;
10: import android.view.ViewGroup;
11: import android.widget.LinearLayout;
12: import android.widget.TextView;
13: /**
14:  *
15:  * @author ubay
16:  *
17:  *     Esta clase es lanzada cuando se pulsa en el nombre de uno de los
18:  *     profesores en la actividad TeachersList. Se aprovecha que la
19:  *     informacion de la asignatura ya ha sido descargada en la anterior
20:  *     actividad y la recibe de la misma. Con esta informacion, se puebla el
21:  *     layout correspondiente que muestra la informacion tabulada
22:  */
23: public class SingleTeacherFragment extends Fragment {
24:     private View myView;
25:
26:     @Override
27:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
28:         Bundle savedInstanceState) {
29:         myView = inflater.inflate(R.layout.single_teacher_fragment, container,
30:             false);
31:         Bundle bundle = getArguments();
32:         if (bundle == null) {
33:             bundle = getActivity().getIntent().getExtras();
34:         }
35:
36:         String teacher_data = bundle.getString("teacher-info");
37:         XmlExtractor xml_extractor = new XmlExtractor(teacher_data);
38:         String[] tags = getResources().getStringArray(
39:             R.array.generic_teachers_info);
40:         HashMap<String, String> teacher_info = xml_extractor
41:             .extractTagsFromDoc(tags);
42:
43:         getActivity().setTitle(
44:             teacher_info.get("name") + " " + teacher_info.get("last
45: -name"));
46:         populateViews(teacher_info);
47:         return myView;
48:     }
49:
50:     private void populateViews(HashMap<String, String> teacher_info) {
51:         setValuesToViews(sanitizeInput(teacher_info));
52:     }
53:
54:     private void setValuesToViews(HashMap<String, String> clean_teacher) {
55:         LinearLayout first_row = (LinearLayout) myView
56:             .findViewById(R.id.first_row);
57:         LinearLayout second_row = (LinearLayout) myView
58:             .findViewById(R.id.second_row);
59:         LinearLayout third_row = (LinearLayout) myView
60:             .findViewById(R.id.third_row);
61:         LinearLayout fourth_row = (LinearLayout) myView
62:             .findViewById(R.id.fourth_row);
```

```

63:
64:         TextView[] text_views = new TextView[9];
65:
66:         text_views[0] = (TextView) first_row.findViewById(R.id.left_item); // d
egree
67:         text_views[1] = (TextView) first_row.findViewById(R.id.right_item); //
email
68:
69:         text_views[2] = (TextView) second_row.findViewById(R.id.left_item); //
branch
70:         text_views[3] = (TextView) second_row.findViewById(R.id.right_item); //
category
71:
72:         text_views[4] = (TextView) third_row.findViewById(R.id.left_item); // u
lpgc_graduate
73:         text_views[5] = (TextView) third_row.findViewById(R.id.right_item); //
department
74:
75:         text_views[6] = (TextView) fourth_row.findViewById(R.id.left_item); //
entity
76:         text_views[7] = (TextView) fourth_row.findViewById(R.id.right_item); //
active
77:
78:         text_views[8] = (TextView) myView.findViewById(R.id.cv_textview); // cv
_abstract
79:
80:         TextView name = (TextView) myView.findViewById(R.id.teacher_name);
81:         String[] tags_asIndex = getResources().getStringArray(
82:             R.array.info_generica_profesores);
83:         String[] locale_tags = getResources().getStringArray(
84:             R.array.generic_teachers_info);
85:         name.setText(clean_teacher.get("name") + " "
86:             + clean_teacher.get("last-name"));
87:
88:         for (int i = 2; i < clean_teacher.size() - 2; i++) {
89:             text_views[i - 2].setText(Html.fromHtml(tags_asIndex[i - 2]
90:                 + clean_teacher.get(locale_tags[i])));
91:         }
92:
93:     }
94:
95:     private HashMap<String, String> sanitizeInput(
96:         HashMap<String, String> teacher_info) {
97:         if (teacher_info.get("graduate-in-ulpgc") == "false") {
98:             teacher_info.put("graduate-in-ulpgc", "No");
99:         } else {
100:             teacher_info.put("graduate-in-ulpgc",
101:                 getResources().getString(R.string.yes));
102:         }
103:         if (teacher_info.get("department") == null) {
104:             teacher_info.put("department",
105:                 getResources().getString(R.string.notUlpgcTeach
er));
106:         }
107:         if (teacher_info.get("cv-abstract") == null) {
108:             teacher_info.put("cv-abstract",
109:                 getResources().getString(R.string.notPresent));
110:         }
111:         if (teacher_info.get("active") == null) {
112:             teacher_info.put("active", "No");
113:         } else {
114:             teacher_info.put("active", getResources().getString(R.string.ye
s));

```

```
115:         }
116:         if (teacher_info.get("institute") == null) {
117:             teacher_info.put("institute",
118:                             getResources().getString(R.string.notPresent));
119:         }
120:         return teacher_info;
121:     }
122:
123: }
```

```
1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.Menu;
8: import android.view.MenuItem;
9:
10: public class ReportsInfo extends MyFragmentActivity {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.reports_info_activity);
16:     }
17:
18:     /**
19:      * Set up the {@link android.app.ActionBar}, if the API is available.
20:      */
21:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
22:     private void setupActionBar() {
23:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
24:             getActionBar().setDisplayHomeAsUpEnabled(true);
25:         }
26:     }
27:
28:     @Override
29:     public boolean onCreateOptionsMenu(Menu menu) {
30:         // Inflate the menu; this adds items to the action bar if it is present
31:
32:         getMenuInflater().inflate(R.menu.reports_info, menu);
33:         return true;
34:     }
35:
36:     @Override
37:     public boolean onOptionsItemSelected(MenuItem item) {
38:         switch (item.getItemId()) {
39:             // This ID represents the Home or Up button. In the case of thi
40:             // activity, the Up button is shown. Use NavUtils to allow user
41:             // to navigate up one level in the application structure. For
42:             // more details, see the Navigation pattern on Android Design:
43:             //
44:             // http://developer.android.com/design/patterns/navigation.html
45:             //
46:             NavUtils.navigateUpFromSameTask(this);
47:             return true;
48:         }
49:         return super.onOptionsItemSelected(item);
50:     }
51:
52: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4: import java.util.HashMap;
5: /**
6:  *
7:  * @author ubay
8:  *
9:  * Clase contenedora de la informacion de cada propuesta para la lista
10: * de propuestas en SubjectsList
11: */
12: public class SubjectInfoContainer {
13:     private String name;
14:     private String course;
15:     private String semester;
16:     private String ECTSCredits;
17:     private String starts;
18:     private String ends;
19:     private String kind;
20:     private String id;
21:     private String whole_subject;
22:
23:     public SubjectInfoContainer(HashMap<String, String> input) {
24:         name = input.get("name");
25:         course = input.get("course");
26:         if (input.containsKey("start-date")) {
27:             if (input.get("start-date") != null
28:                 && (!input.get("start-date").equals("")))) {
29:                 starts = input.get("start-date");
30:             } else
31:                 starts = "";
32:         } else {
33:             starts = "";
34:         }
35:         if (input.containsKey("end-date")) {
36:             if (input.get("end-date") != null
37:                 && (!input.get("end-date").equals("")))) {
38:                 ends = input.get("end-date");
39:             } else
40:                 ends = "";
41:         } else {
42:             ends = "";
43:         }
44:         semester = input.get("semester");
45:         ECTSCredits = input.get("ects-credits");
46:         kind = input.get("kind-id");
47:         id = input.get("id");
48:         whole_subject = input.get("whole");
49:     }
50:
51:     public String getId() {
52:         return this.id;
53:     }
54:
55:     public String getName() {
56:         return this.name;
57:     }
58:
59:     public String getCourse() {
60:         return this.course;
61:     }
62:
63:     public String getSemester() {
```

```
64:         return this.semester;
65:     }
66:
67:     public String getKind() {
68:         return this.kind;
69:     }
70:
71:     public String getStarts() {
72:         return this.starts;
73:     }
74:
75:     public String getEnds() {
76:         return this.ends;
77:     }
78:
79:     public String getEcts() {
80:         return this.ECTSCredits;
81:     }
82:
83:     public String getWhole() {
84:         return this.whole_subject;
85:     }
86:
87:     public ArrayList<String> getArrayList() {
88:         ArrayList<String> output = new ArrayList<String>();
89:
90:         output.add(name);
91:         output.add(course);
92:         output.add(semester);
93:         output.add(starts);
94:         output.add(ends);
95:         output.add(kind);
96:         output.add(ECTSCredits);
97:         output.add(id);
98:         output.add(whole_subject);
99:
100:        return output;
101:    }
102: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.app.AlertDialog;
4: import android.app.AlertDialog.Builder;
5: import android.app.ProgressDialog;
6: import android.content.Context;
7: import android.content.DialogInterface;
8: import android.content.Intent;
9: import android.support.v4.app.FragmentActivity;
10: import android.text.Html;
11:
12: /**
13:  *
14:  * @author ubay
15:  *
16:  *     Clase que extiende FragmentActivity para la gestion de interrupciones
17:  *     de entrada/salida y su envio de mensajes asociados. Asimismo, provee
18:  *     un mecanismo para la prevencion de filtracion de contexto en el
19:  *     proceso de creacion de ventanas de aviso
20:  */
21: public class MyFragmentActivity extends FragmentActivity {
22:     protected Context mActivity;
23:     protected ProgressDialog progressDialog;
24:
25:     protected void AlertDialogGoingBack(int id) {
26:
27:         if (progressDialog != null) {
28:             if (progressDialog.isShowing()) {
29:                 progressDialog.dismiss();
30:             }
31:         }
32:
33:         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
34:         mActivity = this;
35:         alertDialogBuilder.setTitle(R.string.error_period);
36:
37:         alertDialogBuilder
38:             .setMessage(id)
39:             .setCancelable(false)
40:             .setPositiveButton(R.string.ok,
41:                 new DialogInterface.OnClickListener() {
42:                     public void onClick(DialogInterface
face dialog, int id) {
43:                                     finish();
44:                                 }
45:                             });
46:         alertDialogBuilder.create().show();
47:     }
48:
49:     protected void setUpProgressDialog(int id) {
50:         if (progressDialog != null) {
51:             if (progressDialog.isShowing()) {
52:                 progressDialog.dismiss();
53:             }
54:         }
55:
56:         progressDialog = new ProgressDialog(this, R.style.MyTheme);
57:         progressDialog.setCancelable(false);
58:         progressDialog.setMessage(getResources().getString(id));
59:         progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
60:         progressDialog.setProgress(0);
61:         progressDialog.show();
62:     }

```

```

63:
64:     protected void dismissWindow() {
65:         if (progressDialog != null) {
66:             if (progressDialog.isShowing()) {
67:                 progressDialog.dismiss();
68:             }
69:         }
70:     }
71:
72:     protected void setUpLoginAlertDialogError() {
73:         if (progressDialog != null) {
74:             if (progressDialog.isShowing()) {
75:                 progressDialog.dismiss();
76:             }
77:         }
78:         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
79:         alertDialogBuilder.setTitle(R.string.errorLogin);
80:         alertDialogBuilder
81:             .setMessage(R.string.message_back_start)
82:             .setCancelable(false)
83:             .setPositiveButton(R.string.ok,
84:                 new DialogInterface.OnClickListener() {
85:                     public void onClick(DialogInterface
face dialog, int id) {
86:                                     // do nothing
87:                                     }
88:                                     });
89:         alertDialogBuilder.create().show();
90:     }
91:
92:     protected void launchErrorWindow(int error_id) {
93:         if (progressDialog != null) {
94:             if (progressDialog.isShowing()) {
95:                 progressDialog.dismiss();
96:             }
97:         }
98:         AlertDialog.Builder alertDialogBuilder = null;
99:
100:         if (error_id != 0)
101:             alertDialogBuilder = IOError();
102:         else
103:             alertDialogBuilder = IOErrorWithRetry();
104:
105:         alertDialogBuilder.create().show();
106:
107:     }
108:
109:     private Builder IOErrorWithRetry() {
110:         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
111:         mActivity = this;
112:         alertDialogBuilder.setTitle(R.string.error_io);
113:
114:         alertDialogBuilder
115:             .setMessage(R.string.error_io_message)
116:             .setCancelable(false)
117:             .setPositiveButton(R.string.retry,
118:                 new DialogInterface.OnClickListener() {
119:                     public void onClick(DialogInterface
face dialog, int id) {
120:                                     Intent intent = getInte
nt();
121:                                     finish();
122:                                     startActivity(intent);

```

```
123:                                     }
124:                                     });
125:         return alertDialogBuilder;
126:     }
127:
128:     private Builder IOError() {
129:         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
130:         mActivity = this;
131:         alertDialogBuilder.setTitle(R.string.error_io);
132:
133:         alertDialogBuilder
134:             .setMessage(R.string.error_io_message)
135:             .setCancelable(false)
136:             .setPositiveButton(R.string.ok,
137:                 new DialogInterface.OnClickListener() {
138:                     public void onClick(DialogInterface
face dialog, int id) {
139:                                     // Do nothing
140:                                     }
141:                                     });
142:         return alertDialogBuilder;
143:     }
144:
145:     protected void updateProgressDialog(String message) {
146:         progressDialog.setMessage(Html.fromHtml(message));
147:     }
148: }
```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.Menu;
8: import android.view.MenuItem;
9:
10: public class SubjectsList extends MyFragmentActivity {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.subjects_list_activity);
16:     }
17:
18:     /**
19:      * Set up the {@link android.app.ActionBar}, if the API is available.
20:      */
21:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
22:     private void setupActionBar() {
23:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
24:             getActionBar().setDisplayHomeAsUpEnabled(true);
25:         }
26:     }
27:
28:     @Override
29:     public boolean onCreateOptionsMenu(Menu menu) {
30:         // Inflate the menu; this adds items to the action bar if it is present
31:
32:         getMenuInflater().inflate(R.menu.subjects_info, menu);
33:         return true;
34:     }
35:
36:     @Override
37:     public boolean onOptionsItemSelected(MenuItem item) {
38:         switch (item.getItemId()) {
39:             // This ID represents the Home or Up button. In the case of thi
40:             // activity, the Up button is shown. Use NavUtils to allow user
41:             // to navigate up one level in the application structure. For
42:             // more details, see the Navigation pattern on Android Design:
43:             //
44:             // http://developer.android.com/design/patterns/navigation.html
45:             //
46:             NavUtils.navigateUpFromSameTask(this);
47:             return true;
48:         }
49:         return super.onOptionsItemSelected(item);
50:     }
51:
52: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4:
5: import android.content.Context;
6: import android.support.v4.app.Fragment;
7: import android.text.Html;
8: import android.util.Log;
9: import android.view.LayoutInflater;
10: import android.view.View;
11: import android.view.View.OnClickListener;
12: import android.view.ViewGroup;
13: import android.widget.AdapterView;
14: import android.widget.Button;
15: import android.widget.TextView;
16:
17: /**
18:  *
19:  * @author ubay
20:  *
21:  *     Clase encargada de almacenar las entradas de la lista de asignaturas
22:  *     que se muestra en la actividad SubjectsList para su uso posterior
23:  */
24: public class CustomSubjectsAdapter extends ArrayAdapter<ArrayList<String>> {
25:     private ArrayList<ArrayList<String>> entries;
26:     private Fragment fragment;
27:     OnSubjectClickedListener mCallback;
28:
29:     public CustomSubjectsAdapter(Fragment a, int textViewResourceId,
30:         ArrayList<ArrayList<String>> entries) {
31:         super(a.getActivity(), textViewResourceId, entries);
32:         this.entries = entries;
33:         this.fragment = a;
34:     }
35:
36:     public static class ViewHolder {
37:         public Button boton;
38:         public TextView name;
39:         public TextView course;
40:         public TextView semester;
41:         public TextView ECTSCredits;
42:         public TextView starts;
43:         public TextView ends;
44:         public TextView kind;
45:         public String id;
46:     }
47:
48:     public interface OnSubjectClickedListener {
49:         public void OnSubjectClicked(String subject_info);
50:     }
51:
52:     @Override
53:     public View getView(int position, View convertView, ViewGroup parent) {
54:         View v = convertView;
55:         ViewHolder holder;
56:         if (v == null) {
57:             LayoutInflater vi = (LayoutInflater) fragment.getActivity()
58:                 .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
59:             v = vi.inflate(R.layout.single_subject_row, null);
60:             holder = new ViewHolder();
61:             holder.boton = (Button) v.findViewById(R.id.button_subject);
62:
```

```

63:         holder.course = (TextView) v
64:             .findViewById(R.id.textView_subject_course);
65:         holder.semester = (TextView) v
66:             .findViewById(R.id.textView_subject_semester);
67:         holder.ECTSCredits = (TextView) v
68:             .findViewById(R.id.textView_subject_ects);
69:         holder.starts = (TextView) v
70:             .findViewById(R.id.textView_subject_starts);
71:         holder.ends = (TextView) v.findViewById(R.id.textView_subject_e
nds);
72:         holder.kind = (TextView) v.findViewById(R.id.textView_subject_k
ind);
73:
74:         v.setTag(holder);
75:     } else
76:         holder = (ViewHolder) v.getTag();
77:
78:     final ArrayList<String> custom = entries.get(position);
79:     final String id = custom.get(7);
80:     final String whole = custom.get(8);
81:     String[] subject_tags = fragment.getActivity().getResources()
82:         .getStringArray(R.array.subject_forAdapter);
83:     if (custom != null) {
84:         holder.id = id;
85:
86:         holder.boton.setText(Html.fromHtml("<b>" + custom.get(0) + "</b
>"));
87:
88:         holder.course
89:             .setText(Html.fromHtml(subject_tags[0] + custom
.get(1)));
90:         holder.semester.setText(Html.fromHtml(subject_tags[1]
91:             + custom.get(2)));
92:         Log.d("Starts: ", custom.get(3));
93:         Log.d("Ends: ", custom.get(4));
94:         if (custom.get(3).equals("") || custom.get(3) == null) {
95:             holder.starts.setText(Html.fromHtml(subject_tags[2]
96:                 + fragment.getActivity().getResources()
97:                 .getString(R.string.uns
pecified));
98:         } else {
99:             holder.starts.setText(Html.fromHtml(subject_tags[2]
100:                 + custom.get(3)));
101:         }
102:         if (custom.get(4).equals("") || custom.get(4) == null) {
103:             holder.ends.setText(Html.fromHtml(subject_tags[3]
104:                 + fragment.getActivity().getResources()
105:                 .getString(R.string.uns
pecified));
106:         } else {
107:             holder.ends.setText(Html.fromHtml(subject_tags[3]
108:                 + custom.get(4)));
109:         }
110:
111:         String prefix = new String(subject_tags[5]);
112:         String[] subject_types = fragment.getActivity().getResources()
113:             .getStringArray(R.array.subject_types);
114:
115:         if (custom.get(5).equals("1"))
116:             holder.kind.setText(Html.fromHtml(prefix + subject_type
s[0]));
117:         else if (custom.get(5).equals("2"))
118:             holder.kind.setText(Html.fromHtml(prefix + subject_type

```

```
s[1]));
119:         else if (custom.get(5).equals("3"))
120:             holder.kind.setText(Html.fromHtml(prefix + subject_type
s[2]));
121:         else if (custom.get(5).equals("4"))
122:             holder.kind.setText(Html.fromHtml(prefix + subject_type
s[3]));
123:         else if (custom.get(5).equals("5"))
124:             holder.kind.setText(Html.fromHtml(prefix + subject_type
s[4]));
125:
126:         holder.ECTSCredits.setText(Html.fromHtml(subject_tags[4]
127:             + custom.get(6).toString()));
128:
129:         holder.boton.setOnClickListener(new OnClickListener() {
130:
131:             @Override
132:             public void onClick(View v) {
133:                 ((OnSubjectClickedListener) fragment)
134:                     .OnSubjectClicked(whole);
135:
136:             }
137:
138:         });
139:     }
140:     return v;
141: }
142: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4: import java.util.HashMap;
5: import java.util.Locale;
6: /**
7:  *
8:  * @author ubay
9:  *
10:  *      Clase contenedora de la informacion de cada propuesta para la lista
11:  *      de propuestas en SubjectsList
12:  */
13: public class TeacherInfoContainer {
14:     @SuppressWarnings("unused")
15:     private HashMap<String, String> whole;
16:     private String name;
17:     private String last_name;
18:     private String type;
19:     private String state;
20:     private String last_notified;
21:     private String num_notice;
22:     private String degree;
23:     private String email;
24:     private String id;
25:     private String degree_id;
26:     private String participation_id;
27:     private String whole_teacher;
28:
29:     public static final String ULPGC = "ULPGC";
30:     public static final String ZERO = "0";
31:     public static final String[] EN_TAGS = { "Other university",
32:         "External teacher", "Participation confirmed",
33:         "Participation confirmation pending", "Never", "Unspecified" };
34:     public static final String[] ES_TAGS = { "Otra universidad", "Externo",
35:         "ParticipaciÃ³n confirmada", "Pendiente de confirmar participac
iÃ³n",
36:         "Nunca", "Sin especificar" };
37:
38:     public TeacherInfoContainer(HashMap<String, String> input) {
39:         String locale = Locale.getDefault().getLanguage();
40:         String[][] tags = { EN_TAGS, ES_TAGS };
41:         whole = input;
42:         int locale_index = 0;
43:         if (locale == "en") {
44:             locale_index = 0;
45:         } else {
46:             locale_index = 1;
47:         }
48:
49:         name = input.get("name");
50:         last_name = input.get("last-name");
51:
52:         if (input.get("type").equals("ulpgc")) {
53:             type = ULPGC;
54:         } else if (input.get("type").equals("other")) {
55:             type = tags[locale_index][0];
56:         } else if (input.get("type").equals("exter")) {
57:             type = tags[locale_index][1];
58:         }
59:         if (input.get("aasm-state").equals("confirmada")) {
60:             state = tags[locale_index][2];
61:         } else {
62:             state = tags[locale_index][3];

```

```
63:         }
64:         if (!(input.get("last-notification-at") == null)) {
65:             last_notified = input.get("last-notification-at").replace("Z",
""))
66:                 .replace("T", " ");
67:         };
68:     } else {
69:         last_notified = tags[locale_index][4];
70:     }
71:     if (!(input.get("notifications-count") == null)) {
72:         num_notice = input.get("notifications-count");
73:     } else {
74:         num_notice = ZERO;
75:     }
76:     if (!(input.get("degree") == null)) {
77:         degree = input.get("degree");
78:     } else {
79:         degree = tags[locale_index][5];
80:     }
81:     email = input.get("email");
82:     id = input.get("teacher-id");
83:     degree_id = input.get("university-specific-degree-id");
84:     participation_id = input.get("id");
85:     whole_teacher = input.get("whole");
86:
87:     }
88:
89:     public String getId() {
90:         return this.id;
91:     }
92:
93:     public String getName() {
94:         return this.name;
95:     }
96:
97:     public String getLastName() {
98:         return this.last_name;
99:     }
100:
101:     public String getType() {
102:         return this.type;
103:     }
104:
105:     public String getState() {
106:         return this.state;
107:     }
108:
109:     public String getLast_notified() {
110:         return this.last_notified;
111:     }
112:
113:     public String getNum_notice() {
114:         return this.num_notice;
115:     }
116:
117:     public String getDegree() {
118:         return this.degree;
119:     }
120:
121:     public String getEmail() {
122:         return this.email;
123:     }
124:
```

```
125:     public String getDegreeId() {
126:         return this.degree_id;
127:     }
128:
129:     public String getPartId() {
130:         return this.participation_id;
131:     }
132:
133:     public ArrayList<String> getArrayList() {
134:         ArrayList<String> output = new ArrayList<String>();
135:
136:         output.add(name);
137:         output.add(last_name);
138:         output.add(type);
139:         output.add(state);
140:         output.add(last_notified);
141:         output.add(num_notice);
142:         output.add(degree);
143:         output.add(email);
144:         output.add(id);
145:         output.add(degree_id);
146:         output.add(participation_id);
147:         output.add(whole_teacher);
148:
149:         return output;
150:     }
151: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4:
5: import android.app.Activity;
6: import android.content.Context;
7: import android.content.Intent;
8: import android.text.Html;
9: import android.view.LayoutInflater;
10: import android.view.View;
11: import android.view.View.OnClickListener;
12: import android.view.ViewGroup;
13: import android.widget.AdapterView;
14: import android.widget.Button;
15: import android.widget.TextView;
16:
17: /**
18:  *
19:  * @author ubay
20:  *
21:  *     Clase encargada de almacenar las entradas de la lista de titulos que
22:  *     se muestra en la actividad SearchResults para su uso posterior
23:  */
24: public class CustomDegreeAdapter extends ArrayAdapter<ArrayList<String>> {
25:
26:     private ArrayList<ArrayList<String>> entries;
27:     private Activity activity;
28:
29:     public CustomDegreeAdapter(Activity a, int textViewResourceId,
30:         ArrayList<ArrayList<String>> entries) {
31:         super(a, textViewResourceId, entries);
32:         this.entries = entries;
33:         this.activity = a;
34:     }
35:
36:     public static class ViewHolder {
37:         public Button boton;
38:         public TextView type;
39:         public TextView state;
40:         public TextView edition;
41:         public TextView code;
42:         public String id;
43:     }
44:
45:     @Override
46:     public View getView(int position, View convertView, ViewGroup parent) {
47:         View v = convertView;
48:         ViewHolder holder;
49:         if (v == null) {
50:             LayoutInflater vi = (LayoutInflater) activity
51:                 .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
52:             v = vi.inflate(R.layout.single_degree_row, null);
53:             holder = new ViewHolder();
54:             holder.boton = (Button) v.findViewById(R.id.button_degree);
55:             holder.type = (TextView) v.findViewById(R.id.textView_type);
56:             holder.state = (TextView) v.findViewById(R.id.textView_state);
57:             holder.edition = (TextView) v.findViewById(R.id.textView_edition);
58:             holder.code = (TextView) v.findViewById(R.id.textView_code);
59:             v.setTag(holder);
60:         } else
61:

```

```
62:         holder = (ViewHolder) v.getTag();
63:
64:         final ArrayList<String> custom = entries.get(position);
65:         final String id = custom.get(5);
66:         final String edition = custom.get(3);
67:         final String state = custom.get(2);
68:
69:         if (custom != null) {
70:             holder.id = id;
71:
72:             holder.boton.setText(Html.fromHtml("<b>" + custom.get(0) + "</b>
>"));
73:
74:             if (custom.get(1).equals("1"))
75:                 holder.type.setText(Html.fromHtml(activity.getResources
()
76:                     .getString(R.string.ownDegreeBasic)));
77:             else if (custom.get(1).equals("2"))
78:                 holder.type.setText(Html.fromHtml(activity.getResources
()
79:                     .getString(R.string.ownDegreeHigher)));
80:             else if (custom.get(1).equals("3"))
81:                 holder.type.setText(Html.fromHtml(activity.getResources
()
82:                     .getString(R.string.expert)));
83:             else if (custom.get(1).equals("4"))
84:                 holder.type.setText(Html.fromHtml(activity.getResources
()
85:                     .getString(R.string.mastery)));
86:             else if (custom.get(1).equals("5"))
87:                 holder.type.setText(Html.fromHtml(activity.getResources
()
88:                     .getString(R.string.professionalDiploma
)));
89:
90:             String[] degree_states = activity.getResources().getStringArray
(
91:                 R.array.degree_states);
92:
93:             String prefix = activity.getResources().getString(R.string.stat
e);
94:             if (custom.get(2).equals("borrador"))
95:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[1]));
96:             else if (custom.get(2).equals("enviado"))
97:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[2]));
98:             else if (custom.get(2).equals("rechazado"))
99:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[3]));
100:             else if (custom.get(2).equals("aprobado"))
101:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[4]));
102:             else if (custom.get(2).equals("subsanacion"))
103:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[5]));
104:             else if (custom.get(2).equals("cancelado"))
105:                 holder.state.setText(Html.fromHtml(prefix + degree_stat
es[6]));
106:             else
107:                 holder.state.setText("");
108:
109:             holder.edition.setText(Html.fromHtml(activity.getResources()
```

```
110:         .getString(R.string.starts) + custom.get(3));
111: holder.code.setText(Html.fromHtml(activity.getResources()
112:         .getString(R.string.code) + custom.get(4)));
113:
114: holder.boton.setOnClickListener(new OnClickListener() {
115:
116:         @Override
117:         public void onClick(View v) {
118:
119:             Intent intent = new Intent(activity, DegreeMain
Menu.class);
120:             intent.putExtra("id", id);
121:             intent.putExtra("edition", edition);
122:             intent.putExtra("state", state);
123:             activity.startActivity(intent);
124:
125:         }
126:     });
127: }
128: return v;
129: }
130:
131: }
```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.Menu;
8: import android.view.MenuItem;
9:
10: public class TeachersList extends MyFragementActivity {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.teachers_list_activity);
16:     }
17:
18:     /**
19:      * Set up the {@link android.app.ActionBar}, if the API is available.
20:      */
21:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
22:     private void setupActionBar() {
23:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
24:             getActionBar().setDisplayHomeAsUpEnabled(true);
25:         }
26:     }
27:
28:     @Override
29:     public boolean onCreateOptionsMenu(Menu menu) {
30:         // Inflate the menu; this adds items to the action bar if it is present
31:
32:         getMenuInflater().inflate(R.menu.teachers_info, menu);
33:         return true;
34:     }
35:
36:     @Override
37:     public boolean onOptionsItemSelected(MenuItem item) {
38:         switch (item.getItemId()) {
39:             // This ID represents the Home or Up button. In the case of thi
40:             // activity, the Up button is shown. Use NavUtils to allow user
41:             // to navigate up one level in the application structure. For
42:             // more details, see the Navigation pattern on Android Design:
43:             //
44:             // http://developer.android.com/design/patterns/navigation.html
45:             //
46:             NavUtils.navigateUpFromSameTask(this);
47:             return true;
48:         }
49:         return super.onOptionsItemSelected(item);
50:     }
51:
52: }

```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.content.pm.ActivityInfo;
5: import android.os.Build;
6: import android.os.Bundle;
7: import android.support.v4.app.FragmentActivity;
8: import android.support.v4.app.NavUtils;
9: import android.view.Menu;
10: import android.view.MenuItem;
11:
12: public class SearchResults extends FragmentActivity {
13:
14:     @Override
15:     protected void onCreate(Bundle savedInstanceState) {
16:         super.onCreate(savedInstanceState);
17:         setContentView(R.layout.search_results_activity);
18:
19:         if (getResources().getBoolean(R.bool.portrait_only)) {
20:             setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
21:         }
22:     }
23:
24:     /**
25:      * Set up the {@link android.app.ActionBar}, if the API is available.
26:      */
27:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
28:     private void setupActionBar() {
29:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
30:             getActionBar().setDisplayHomeAsUpEnabled(true);
31:         }
32:     }
33:
34:     @Override
35:     public boolean onCreateOptionsMenu(Menu menu) {
36:         // Inflate the menu; this adds items to the action bar if it is present
37:
38:         getMenuInflater().inflate(R.menu.search_results, menu);
39:         return true;
40:     }
41:
42:     @Override
43:     public boolean onOptionsItemSelected(MenuItem item) {
44:         switch (item.getItemId()) {
45:             // This ID represents the Home or Up button. In the case of thi
46:             // activity, the Up button is shown. Use NavUtils to allow user
47:             // to navigate up one level in the application structure. For
48:             // more details, see the Navigation pattern on Android Design:
49:             //
50:             // http://developer.android.com/design/patterns/navigation.html
51:             //
52:             NavUtils.navigateUpFromSameTask(this);
53:             return true;
54:         }
55:         return super.onOptionsItemSelected(item);
56:     }
57:
58: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.HashMap;
5:
6: import android.os.AsyncTask;
7: import android.os.Bundle;
8: import android.support.v4.app.Fragment;
9: import android.text.Html;
10: import android.view.LayoutInflater;
11: import android.view.View;
12: import android.view.ViewGroup;
13: import android.widget.TextView;
14:
15: /**
16:  *
17:  * @author ubay
18:  *
19:  *     La clase AcademicInfoFragment se encarga de descargar el documento
20:  *     XML con la informaci3n del titulo propio con la ID recibida de la
21:  *     actividad anterior. Una vez recibido el documento, se puebla el
22:  *     layout con la informacion extraida
23:  *
24:  */
25:
26: public class AcademicInfoFragment extends Fragment {
27:     private static final int[] textview_list = { R.id.name, R.id.branch,
28:         R.id.teaching_mode, R.id.state, R.id.edition, R.id.code,
29:         R.id.proposer, R.id.startDate, R.id.endDate, R.id.units,
30:         R.id.teachingPlace, R.id.manager, R.id.institutionRepresentativ
e,
31:         R.id.headquarters, R.id.evalAcecau, R.id.competencias,
32:         R.id.justificaCompetencias, R.id.resultadosEsperados,
33:         R.id.recursosDisponibles };
34:
35:     private View myView;
36:     private AsyncTask<String, Void, Void> task;
37:
38:     @Override
39:     public void onStop() {
40:         super.onStop();
41:         if (task != null) {
42:             task.cancel(true);
43:         }
44:     }
45:
46:     @Override
47:     public void onDestroy() {
48:         super.onStop();
49:         if (task != null) {
50:             task.cancel(true);
51:         }
52:     }
53:
54:     @Override
55:     public void onDetach() {
56:         super.onStop();
57:         if (task != null) {
58:             task.cancel(true);
59:         }
60:     }
61:
62:     @Override
```

```

63:     public void onDestroyView() {
64:         super.onStop();
65:         if (task != null) {
66:             task.cancel(true);
67:         }
68:     }
69:
70:     @Override
71:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
72:         Bundle savedInstanceState) {
73:
74:         task = new retrieveInfo();
75:         task.execute();
76:         myView = inflater.inflate(R.layout.academic_info_fragment, container,
77:             false);
78:         return myView;
79:     }
80:
81:     private class retrieveInfo extends AsyncTask<String, Void, Void> {
82:         Degree degree;
83:         IOException error;
84:
85:         @Override
86:         protected void onPreExecute() {
87:             ((MyFragmentActivity) getActivity())
88:                 .setUpProgressDialog(R.string.loadingMessage);
89:         }
90:
91:         @Override
92:         protected Void doInBackground(String... params) {
93:             String target_url = setTargetURL();
94:             String degree_data_xml = getXmlPage(target_url);
95:             if (degree_data_xml != null) {
96:                 degree = new Degree(degree_data_xml, getActivity());
97:             }
98:
99:             return null;
100:         }
101:     }
102:
103:     private String setTargetURL() {
104:         final String UNIV_SPEC_DEG = "university_specific_degrees";
105:         final String XML_EXTENSION = ".xml";
106:         final String SLASH = "/";
107:
108:         GlobalState gs = (GlobalState) getActivity().getApplication();
109:         String base_url = gs.getTargetAddress();
110:
111:         String degree_id = "";
112:         Bundle bundle = getArguments();
113:
114:         if (bundle != null) {
115:             degree_id = bundle.getString("id");
116:         } else {
117:             degree_id = getActivity().getIntent().getStringExtra("i
118: d");
119:         }
120:         String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_i
121:             d
122:                 + XML_EXTENSION;
123:         return target_url;

```

```
124:         }
125:
126:     private String getXmlPage(String target_url) {
127:         XmlPage xml_page = new XmlPage(getActivity());
128:         String xml_response = null;
129:         try {
130:             xml_response = xml_page.getPage(target_url);
131:         } catch (IOException e) {
132:             error = e;
133:             return null;
134:         }
135:         return xml_response;//
136:     }
137:
138:     @Override
139:     protected void onPostExecute(Void param) {
140:
141:         ((MyFragmentActivity) getActivity()).dismissWindow();
142:         if (error != null) {
143:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
144:         } else {
145:             populateViews(degree.getData());
146:         }
147:     }
148: }
149:
150: }
151:
152: public void populateViews(HashMap<String, String> hashMap) {
153:     TextView textview;
154:     String[] tags_eng = getResources()
155:         .getStringArray(R.array.academic_info);
156:
157:     String[] tags_es = getResources().getStringArray(
158:         R.array.informacion_academica);
159:
160:     for (int i = 0; i < tags_es.length - 1; i++) {
161:
162:         textview = (TextView) myView.findViewById(textview_list[i]);
163:
164:         switch (i) {
165:
166:         case 0: {
167:             if (hashMap.containsKey(tags_eng[i])) {
168:                 textview.setText(tags_es[i] + " "
169:                     + hashMap.get(tags_eng[i]));
170:             } else
171:                 textview.setText(tags_es[i]);
172:             break;
173:         }
174:         case 1: {
175:             String[] tags = getResources().getStringArray(
176:                 R.array.branchKnowledge);
177:
178:             if (hashMap.containsKey(tags_eng[i])) {
179:                 String rama = null;
180:                 if (hashMap.get(tags_eng[i]).equals("1")) {
181:                     rama = tags[0];
182:                 } else if (hashMap.get(tags_eng[i]).equals("2"))
183:                     rama = tags[1];
184:                 } else if (hashMap.get(tags_eng[i]).equals("3"))
```

```

) {
185:             rama = tags[2];
186:         } else if (hashMap.get(tags_eng[i]).equals("4"))
) {
187:             rama = tags[3];
188:         } else if (hashMap.get(tags_eng[i]).equals("5"))
) {
189:             rama = tags[4];
190:         }
191:         textView.setText(Html.fromHtml("<b>" + tags_es[
i]
192:             + "</b><br>" + rama));
193:     } else
194:         textView.setText(Html.fromHtml("<b>" + tags_es[
i] + "</b>"));
195:         break;
196:     }
197: case 2: {
198:     String[] tags = getResources().getStringArray(
199:         R.array.degree_types);
200:     if (hashMap.containsKey(tags_eng[i])) {
201:         String tipo = null;
202:         if (hashMap.get(tags_eng[i]).equals("1")) {
203:             tipo = tags[1];
204:         } else if (hashMap.get(tags_eng[i]).equals("2"))
) {
205:             tipo = tags[2];
206:         } else if (hashMap.get(tags_eng[i]).equals("3"))
) {
207:             tipo = tags[3];
208:         } else if (hashMap.get(tags_eng[i]).equals("4"))
) {
209:             tipo = tags[4];
210:         } else if (hashMap.get(tags_eng[i]).equals("5"))
) {
211:             tipo = tags[5];
212:         }
213:         textView.setText(Html.fromHtml("<b>" + tags_es[
i] + "</b> "
214:             + tipo));
215:
216:     } else
217:         textView.setText(tags_es[i]);
218:     break;
219: }
220: case 3:
221: case 4:
222: case 6:
223: case 7: {
224:     if (hashMap.containsKey(tags_eng[i])) {
225:         textView.setText(Html.fromHtml("<b>" + tags_es[
i] + "</b> "
226:             + hashMap.get(tags_eng[i])));
227:     } else
228:         textView.setText(Html.fromHtml("<b>" + tags_es[
i] + "</b>"));
229:     break;
230: }
231: case 8:
232: case 9: {
233:
234:     if (hashMap.containsKey(tags_eng[i])) {
235:         String temp = hashMap.get(tags_eng[i]);

```

```

236:         if (!(temp == null)) {
237:             textview.setText(Html.fromHtml("<b>" +
tags_es[i]
238:                 + "</b><br> " + hashMap
.get(tags_eng[i]));
239:         } else
240:             textview.setText(Html.fromHtml("<b>"
241:                 + tags_es[i]
242:                 + "</b><br> "
243:                 + getResources()
244:                     .getStr
ing(R.string.unspecified));
245:         }
246:         break;
247:     }
248:     }
249:     case 5:
250:     case 10:
251:     case 11:
252:     case 12:
253:
254:     case 15:
255:     case 16:
256:     case 17:
257:     case 18: {
258:         textview = (TextView) myView.findViewById(textview_list
[i]);
259:         if (hashMap.containsKey(tags_eng[i])) {
260:             textview.setText(Html.fromHtml("<b>" + tags_es[
i]
261:                 + "</b><br> " + hashMap.get(tag
s_eng[i]));
262:         } else
263:             textview.setText(Html.fromHtml("<b>" + tags_es[
i]
264:                 + "</b><br>"));
265:         break;
266:     }
267:     case 13: {
268:         textview = (TextView) myView.findViewById(textview_list
[i]);
269:         if (hashMap.containsKey(tags_eng[i])) {
270:             String temp = hashMap.get(tags_eng[i]);
271:             if (!(temp == null)) {
272:                 textview.setText(Html.fromHtml("<b>" +
tags_es[i]
273:                 + "</b><br> " + hashMap
.get(tags_eng[i]));
274:             } else
275:                 textview.setText(Html.fromHtml("<b>"
276:                 + tags_es[i]
277:                 + "</b><br> "
278:                 + getResources()
279:                     .getStr
ing(R.string.unspecified));
280:             }
281:             break;
282:         }
283:     case 14: {
284:         if (hashMap.containsKey(tags_eng[i])) {
285:             String temp = hashMap.get(tags_eng[i]);
286:             if (!(temp == null)) {
287:                 textview.setText(Html.fromHtml("<b>" +

```

```
tags_es[i]
288:                                     + "</b><br> " + hashMap
.get(tags_eng[i]));
289:                                     } else
290:                                     textView.setText(Html.fromHtml("<b>"
291:                                     + tags_es[i]
292:                                     + "</b><br> "
293:                                     + getResources()
294:                                     .getStr
ing(R.string.unspecified));
295:                                     }
296:                                     break;
297:                                     }
298:                                     }
300:                                     }
301:                                     }
302: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.HashMap;
4:
5: import android.graphics.Color;
6: import android.graphics.Typeface;
7: import android.os.Bundle;
8: import android.support.v4.app.Fragment;
9: import android.text.Html;
10: import android.view.Gravity;
11: import android.view.LayoutInflater;
12: import android.view.View;
13: import android.view.ViewGroup;
14: import android.widget.LinearLayout;
15: import android.widget.ScrollView;
16: import android.widget.TextView;
17:
18: /**
19:  *
20:  * @author ubay
21:  *
22:  *     TabOverview representa una de las pestañas de las tres que componen
23:  *     la informacion economica. Utiliza la informacion obtenida de la
24:  *     actividad padre para poblar el layout correspondiente
25:  */
26: public class TabOverview extends Fragment {
27:
28:     public static final String EURO = "â\202-";
29:
30:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
31:         Bundle savedInstanceState) {
32:         if (container == null) {
33:             return null;
34:         }
35:
36:         HashMap<String, String> info;
37:
38:         if (getActivity()
39:             .getClass()
40:             .toString()
41:             .equals("class es.ulpgc.titulospropios.gestion.ubay.Deg
reeMainMenu")) {
42:             info = EconomicInfoFragment.hashMap;
43:         } else {
44:             info = EconomicInfo.hashMap;
45:         }
46:
47:         ScrollView outer_layout = (ScrollView) inflater.inflate(
48:             R.layout.degree_overview_tab, container, false);
49:
50:         return populateViews(info, outer_layout);
51:     }
52:
53:
54:     private View populateViews(HashMap<String, String> info,
55:         ScrollView outer_layout) {
56:         int[] views = { R.id.incomes_total, R.id.parcial_cost,
57:             R.id.ulpgc_compensation, R.id.institute_compensation,
58:             R.id.manager_compensation, R.id.total, R.id.balance };
59:         String[] tags = getResources().getStringArray(R.array.overview);
60:         String[] etiquetas = getResources().getStringArray(
61:             R.array.overviewExpenses);
62:

```

```
63:     LinearLayout inner_layout = null;
64:     for (int i = 0; i < tags.length - 1; i++) {
65:         inner_layout = (LinearLayout) outer_layout.findViewById(views[
]);
66:         TextView left_item = (TextView) inner_layout
67:             .findViewById(R.id.left_item);
68:         TextView right_item = (TextView) inner_layout
69:             .findViewById(R.id.right_item);
70:         left_item.setText(Html.fromHtml(etiquetas[i]));
71:         if (info.get(tags[i]) == null) {
72:             right_item.setText("0");
73:         } else {
74:             right_item.setText(info.get(tags[i]) + "â\202r");
75:             right_item.setGravity(Gravity.RIGHT);
76:         }
77:     }
78:     LinearLayout layout_balance = (LinearLayout) outer_layout
79:         .findViewById(R.id.balance);
80:     TextView balance_tag = (TextView) layout_balance
81:         .findViewById(R.id.left_item);
82:     balance_tag.setText(R.string.balance);
83:     balance_tag.setGravity(Gravity.CENTER);
84:     balance_tag.setTypeface(null, Typeface.BOLD);
85:     TextView balance_sum = (TextView) layout_balance
86:         .findViewById(R.id.right_item);
87:     balance_sum.setText(info.get("sum") + EURO);
88:     balance_sum.setGravity(Gravity.RIGHT);
89:
90:
91:     if (Float.parseFloat(info.get("sum")) >= 0) {
92:         balance_sum.setBackgroundColor(Color.GREEN);
93:     } else {
94:         balance_sum.setBackgroundColor(Color.RED);
95:     }
96:     return outer_layout;
97: }
98: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.os.Bundle;
4: import android.support.v4.app.Fragment;
5: import android.view.LayoutInflater;
6: import android.view.View;
7: import android.view.ViewGroup;
8:
9: /**
10:  *
11:  * @author ubay
12:  *
13:  *      Clase vacia que infla un fragmento vacio como placeholder
14:  */
15: public class BlankFragment extends Fragment {
16:     @Override
17:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
18:         Bundle savedInstanceState) {
19:         // Inflate the layout for this fragment
20:         return inflater.inflate(R.layout.blank_fragment, container, false);
21:     }
22: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.io.StringReader;
5: import java.util.Arrays;
6: import java.util.HashMap;
7:
8: import org.xmlpull.v1.XmlPullParser;
9: import org.xmlpull.v1.XmlPullParserException;
10: import org.xmlpull.v1.XmlPullParserFactory;
11:
12: import android.content.Context;
13:
14: /**
15:  *
16:  * @author ubay
17:  *
18:  *     Clase que representa una propuesta, con metodos para extraer y
19:  *     acceder a sus etiquetas
20:  */
21: public class Degree {
22:     String[] tags;
23:     HashMap<String, String> information;
24:
25:     public Degree(String xml_document, Context mContext) {
26:         tags = mContext.getResources().getStringArray(R.array.academic_info);
27:         information = new HashMap<String, String>(extractTags(xml_document));
28:     }
29:
30:     private HashMap<String, String> extractTags(String xml_document) {
31:         XmlPullParserFactory factory;
32:         XmlPullParser xpp = null;
33:         try {
34:             factory = XmlPullParserFactory.newInstance();
35:             factory.setNamespaceAware(true);
36:             xpp = factory.newPullParser();
37:             xpp.setInput(new StringReader(xml_document));
38:         } catch (XmlPullParserException e) {
39:             e.printStackTrace();
40:         }
41:         HashMap<String, String> output = new HashMap<String, String>();
42:
43:         int eventType = 0;
44:         try {
45:             eventType = xpp.getEventType();
46:         } catch (XmlPullParserException e) {
47:             e.printStackTrace();
48:         }
49:         while (eventType != XmlPullParser.END_DOCUMENT) {
50:             if (eventType == XmlPullParser.START_TAG) {
51:                 if (Arrays.asList(tags).contains(xpp.getName())) {
52:                     String current_tag = new String(xpp.getName());
53:                     try {
54:                         eventType = xpp.next();
55:                     } catch (XmlPullParserException e) {
56:                         e.printStackTrace();
57:                     } catch (IOException e) {
58:                         e.printStackTrace();
59:                     }
60:                     output.put(current_tag, xpp.getText());
61:                 }
62:             }
63:         }
64:     }
65: }
```

```
64:         eventType = xpp.next();
65:     } catch (XmlPullParserException e) {
66:         e.printStackTrace();
67:     } catch (IOException e) {
68:         e.printStackTrace();
69:     }
70:     }
71:     return output;
72:
73: }
74:
75: public HashMap<String, String> getData() {
76:     return information;
77: }
78:
79: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.io.StringReader;
5: import java.util.ArrayList;
6: import java.util.Arrays;
7: import java.util.HashMap;
8:
9: import org.xmlpull.v1.XmlPullParser;
10: import org.xmlpull.v1.XmlPullParserException;
11: import org.xmlpull.v1.XmlPullParserFactory;
12:
13: /**
14:  *
15:  * @author ubay
16:  *
17:  *     XmlExtractor se ocupa de recibir un documento XML y tratarlo, ya sea
18:  *     recorriendolo para extraer el listado de etiquetas que le provea o
19:  *     para dividir el documento XML en funcion de un texto separador
20:  */
21: public class XmlExtractor {
22:
23:     private XmlPullParserFactory factory;
24:     private XmlPullParser xpp;
25:     private String input;
26:
27:     public XmlExtractor(String input) {
28:
29:         try {
30:             factory = XmlPullParserFactory.newInstance();
31:             factory.setNamespaceAware(true);
32:             xpp = factory.newPullParser();
33:             xpp.setInput(new StringReader(input));
34:
35:         } catch (XmlPullParserException e) {
36:             e.printStackTrace();
37:         }
38:
39:         this.input = input;
40:
41:     }
42:
43:     public void setInput(String newInput) throws XmlPullParserException {
44:         input = newInput;
45:         xpp.setInput(new StringReader(input));
46:     }
47:
48:     public String extractSingleTag(String tag) {
49:         String output = null;
50:
51:         int eventType = 0;
52:         try {
53:             eventType = xpp.getEventType();
54:             while (eventType != XmlPullParser.END_DOCUMENT) {
55:                 if (eventType == XmlPullParser.START_TAG) {
56:                     if (tag.equals(xpp.getName())) {
57:                         eventType = xpp.next();
58:                         output = xpp.getText();
59:                         break;
60:                     }
61:                 }
62:                 eventType = xpp.next();
63:
```

```

64:         }
65:     } catch (XmlPullParserException e) {
66:         // TODO Auto-generated catch block
67:         e.printStackTrace();
68:     } catch (IOException e) {
69:         // TODO Auto-generated catch block
70:         e.printStackTrace();
71:     }
72:     return output;
73:
74: }
75:
76: public HashMap<String, String> extractTagsFromDoc(String[] tags) {
77:
78:     HashMap<String, String> output = new HashMap<String, String>();
79:
80:     int eventType = 0;
81:     try {
82:         eventType = xpp.getEventType();
83:         while (eventType != XmlPullParser.END_DOCUMENT) {
84:             if (eventType == XmlPullParser.START_TAG) {
85:                 if (Arrays.asList(tags).contains(xpp.getName())
) {
86:                     String temp_tag = new String(xpp.getNam
e());
87:                     eventType = xpp.next();
88:                     output.put(temp_tag, xpp.getText());
89:                 }
90:             }
91:             eventType = xpp.next();
92:
93:         }
94:     } catch (XmlPullParserException e) {
95:         // TODO Auto-generated catch block
96:         e.printStackTrace();
97:     } catch (IOException e) {
98:         // TODO Auto-generated catch block
99:         e.printStackTrace();
100:    }
101:    return output;
102: }
103:
104: public ArrayList<String> unitSeparator(String endTag) {
105:     String temp_trimdoc = input;
106:     ArrayList<String> trimmedDoc = new ArrayList<String>();
107:     boolean contains = input.contains(endTag);
108:
109:     while (contains) {
110:         trimmedDoc.add(temp_trimdoc.substring(0,
111:             temp_trimdoc.indexOf(endTag) + endTag.length())
);
112:         temp_trimdoc = temp_trimdoc.substring(temp_trimdoc.indexOf(endT
ag)
113:             + endTag.length());
114:         contains = temp_trimdoc.contains(endTag);
115:     }
116:     return trimmedDoc;
117: }
118: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.HashMap;
4:
5: import android.os.Bundle;
6: import android.support.v4.app.Fragment;
7: import android.text.Html;
8: import android.view.LayoutInflater;
9: import android.view.View;
10: import android.view.ViewGroup;
11: import android.widget.TextView;
12:
13: /**
14:  *
15:  * @author ubay
16:  *
17:  *     Esta clase es lanzada cuando se pulsa en el nombre de uno de los
18:  *     profesores en la actividad SubjectsList. Se aprovecha que la
19:  *     informacion de la asignatura ya ha sido descargada en la anterior
20:  *     actividad y la recibe de la misma. Con esta informacion, se puebla el
21:  *     layout correspondiente que muestra la informacion tabulada
22:  */
23: public class SingleSubjectFragment extends Fragment {
24:     private static final int[] textview_list = { R.id.name, R.id.branch,
25:         R.id.teaching_mode, R.id.state, R.id.edition, R.id.start_date,
26:         R.id.end_date, R.id.startDate, R.id.endDate, R.id.units,
27:         R.id.teachingPlace, R.id.class_hours,
28:         R.id.institutionRepresentative, R.id.misc_activities_hours,
29:         R.id.headquarters, R.id.exams_hours, R.id.final_work_hours,
30:         R.id.studying_hours, R.id.hours_sum, R.id.competences,
31:         R.id.activities, R.id.description, R.id.assesment_system };
32:     View myView;
33:
34:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
35:         Bundle savedInstanceState) {
36:         myView = inflater.inflate(R.layout.single_subject_fragment, container,
37:             false);
38:
39:         Bundle bundle = getArguments();
40:         String subject_info;
41:
42:         if (bundle != null) {
43:             subject_info = bundle.getString("subject-info");
44:         } else {
45:             subject_info = getActivity().getIntent().getStringExtra(
46:                 "subject-info");
47:         }
48:         String[] tags = getActivity().getResources().getStringArray(
49:             R.array.complete_subject_info);
50:         XmlExtractor xml_extractor = new XmlExtractor(subject_info);
51:         HashMap<String, String> subject_data = xml_extractor
52:             .extractTagsFromDoc(tags);
53:
54:         getActivity().setTitle(subject_data.get("name"));
55:         populateViews(subject_data);
56:
57:         return myView;
58:     }
59:
60:     private void populateViews(HashMap<String, String> subject_data) {
61:         String[] tags = getResources().getStringArray(
62:             R.array.completeSubjectTags);
63:         String[] tags_inOrder = getResources().getStringArray(
```

```

64:         R.array.complete_subject_info);
65:
66:     getActivity().setTitle(subject_data.get("name"));
67:
68:     for (int i = 0; i < tags_inOrder.length; i++) {
69:         if (subject_data.get(tags_inOrder[i]) == null) {
70:             subject_data.put(tags_inOrder[i], "");
71:         }
72:     }
73:
74:     String kind = determineKind(Integer.parseInt(subject_data
75:         .get(tags_inOrder[3])));
76:     subject_data.put(tags_inOrder[3], kind);
77:
78:     String branch = determineBranch(Integer.parseInt(subject_data
79:         .get(tags_inOrder[7])));
80:     subject_data.put(tags_inOrder[7], branch);
81:
82:     String mode = determineMode(Integer.parseInt(subject_data
83:         .get(tags_inOrder[8])));
84:     subject_data.put(tags_inOrder[8], mode);
85:
86:     for (int i = 0; i < 9; i++) {
87:         TextView text_holder = (TextView) myView
88:             .findViewById(textview_list[i]);
89:         text_holder.setText(Html.fromHtml("<b>" + tags[i] + ": " + "</b>
>"
90:             + subject_data.get(tags_inOrder[i])));
91:     }
92:     for (int i = 9; i < 11; i++) {
93:         TextView text_holder = (TextView) myView
94:             .findViewById(textview_list[i]);
95:         text_holder.setText(Html.fromHtml("<b>"
96:             + tags[i]
97:             + ": "
98:             + "</b><br>"
99:             + subject_data.get(tags_inOrder[i]).replace("Z"
, ""))
100:             .replace("T", " "));
101:     }
102:     for (int i = 11; i < 19; i++) {
103:         TextView text_holder = (TextView) myView
104:             .findViewById(textview_list[i]);
105:         text_holder.setText(Html.fromHtml("<b>" + tags[i] + ": " + "</b>
>"
106:             + subject_data.get(tags_inOrder[i]) + " "
107:             + getResources().getString(R.string.hours)));
108:     }
109:     for (int i = 19; i < tags.length - 1; i++) {
110:         TextView text_holder = (TextView) myView
111:             .findViewById(textview_list[i]);
112:         text_holder.setText(Html.fromHtml("<b>" + tags[i] + ": " + "</b>
>"
113:             + subject_data.get(tags_inOrder[i])));
114:     }
115:
116:     return;
117: }
118:
119: private String determineMode(int value) {
120:     String[] mode_tags = getResources()
121:         .getStringArray(R.array.teachingMode);
122:     return mode_tags[value - 1];

```

```
123:     }
124:
125:     private String determineBranch(int value) {
126:         String[] branch_tags = getResources().getStringArray(
127:             R.array.branchKnowledge);
128:         return branch_tags[value - 1];
129:     }
130:
131:     private String determineKind(int value) {
132:         String[] subject_tags = getResources().getStringArray(
133:             R.array.subject_types);
134:         return subject_tags[value - 1];
135:     }
136: }
137:
138: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4:
5: /**
6:  *
7:  * @author ubay
8:  *
9:  * Clase que representa una pagina de resultados a la hora de descargar
10:  * el listado de la aplicacion. Necesario porque el maximo de resultados
11:  * en una peticion de propuestas es 20. Se decide paginar los resultados
12:  * en DegreePage de 20 titulos cada uno
13:  */
14: public class DegreePage {
15:     private String xml_document;
16:     private static final String endTag = "</university-specific-degree>";
17:     private static final String emptyTag = "<nil-classes type=\"array\"/>";
18:
19:     public DegreePage(String xml_document) {
20:         this.xml_document = xml_document;
21:     }
22:
23:     public ArrayList<String> trimXML() {
24:
25:         String temp_trimdoc = xml_document;
26:         ArrayList<String> trimmedDoc = new ArrayList<String>();
27:         boolean contains = xml_document.contains(endTag);
28:
29:         while (contains) {
30:             trimmedDoc.add(temp_trimdoc.substring(0,
31:                 temp_trimdoc.indexOf(endTag) + endTag.length())
32:
33:                 + endTag.length());
34:             contains = temp_trimdoc.contains(endTag);
35:         }
36:         return trimmedDoc;
37:     }
38:
39:     public boolean isEmpty() {
40:         return xml_document.contains(emptyTag);
41:     }
42:
43: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.io.UnsupportedEncodingException;
5: import java.util.ArrayList;
6: import java.util.List;
7:
8: import org.apache.http.HttpEntity;
9: import org.apache.http.HttpResponse;
10: import org.apache.http.NameValuePair;
11: import org.apache.http.client.ClientProtocolException;
12: import org.apache.http.client.CookieStore;
13: import org.apache.http.client.entity.UrlEncodedFormEntity;
14: import org.apache.http.client.methods.HttpGet;
15: import org.apache.http.client.methods.HttpPost;
16: import org.apache.http.client.protocol.ClientContext;
17: import org.apache.http.impl.client.BasicCookieStore;
18: import org.apache.http.impl.client.DefaultHttpClient;
19: import org.apache.http.message.BasicNameValuePair;
20: import org.apache.http.protocol.BasicHttpContext;
21: import org.apache.http.protocol.HTTP;
22: import org.apache.http.protocol.HttpContext;
23: import org.apache.http.util.EntityUtils;
24:
25: import android.annotation.SuppressLint;
26: import android.annotation.TargetApi;
27: import android.content.Context;
28: import android.content.Intent;
29: import android.content.SharedPreferences;
30: import android.content.SharedPreferences.Editor;
31: import android.content.pm.ActivityInfo;
32: import android.os.AsyncTask;
33: import android.os.Build;
34: import android.os.Bundle;
35: import android.support.v4.app.NavUtils;
36: import android.view.Menu;
37: import android.view.MenuItem;
38: import android.view.View;
39: import android.view.WindowManager;
40: import android.widget.Button;
41: import android.widget.CheckBox;
42: import android.widget.EditText;
43: import android.widget.TextView;
44:
45: /**
46:  *
47:  * @author ubay
48:  *
49:  * Clase que muestra la pantalla de autentificacion de usuario. Recibe la
50:  * informacion de login y contraseña, controla la opcion de recordar
51:  * usuario, realiza el intento de autentificacion y procede a la siguiente
52:  * pantalla en caso correcto. Lanza error en caso de fallo de
53:  * autentificacion
54:  */
55: public class LoginUser extends MyFragmentActivity {
56:     private String target_addr;
57:     private String ROLE = "manager";
58:     private static final String REMEMBER_USER = "remember_user";
59:     private static final String USER_INFO = "Preferences";
60:     private static final String LOGIN = "login";
61:     private static final String PASSWORD = "passwd";
62:     private static final String PARAM_LOGIN = "user_session[login]";
63:     private static final String PARAM_PASSWD = "user_session[password]";
```

```

64:     private static final String PARAM_DNI = "user_session[dni]";
65:     private static final String MANAGER_SESSION = "user_session";
66:     private static final String TEACHER_SESSION = "teacher_session";
67:     private static final String RESOURCE_URL = "university_specific_degrees.xml";
68:     private static final String ROLE_PREFERENCES = "role";
69:
70:     private Context baseContext = this;
71:     private boolean manager;
72:
73:     @Override
74:     protected void onCreate(Bundle savedInstanceState) {
75:         super.onCreate(savedInstanceState);
76:
77:         if (getResources().getBoolean(R.bool.portrait_only)) {
78:             setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
79:         }
80:         setContentView(R.layout.login_user_activity);
81:
82:         setBaseAddress();
83:
84:         setRoleMessageOnScreen();
85:
86:         if (rememberUserFeatureisOn()) {
87:             setStoredUserDataOnEditBoxes();
88:         }
89:
90:         setUpLoginButton();
91:
92:         setUpRememberUserButton();
93:     }
94:
95:     private void setBaseAddress() {
96:         target_addr = getResources().getString(R.string.production_addr);
97:     }
98:
99:     private void setUpRememberUserButton() {
100:         CheckBox check_box = (CheckBox) findViewById(R.id.rememberButton);
101:         check_box.setOnClickListener(new View.OnClickListener() {
102:             public void onClick(View v) {
103:                 SharedPreferences settings = getSharedPreferences(USER_
104:                 INFO, 0);
105:                 SharedPreferences.Editor editor = settings.edit();
106:                 editor.putBoolean(REMEMBER_USER, ((CheckBox) v).isChecked());
107:                 editor.commit();
108:             }
109:         });
110:     }
111:
112:     private void setUpLoginButton() {
113:         Button login_button = (Button) findViewById(R.id.loginButton);
114:         login_button.setOnClickListener(new View.OnClickListener() {
115:             public void onClick(View v) {
116:                 new performLogin().execute(target_addr);
117:             }
118:         });
119:     }
120:
121:     private class performLogin extends AsyncTask<String, Void, Integer> {
122:         HttpContext localContext;
123:         CookieStore cookieStore;

```

```

124:      DefaultHttpClient client;
125:      HttpPost post;
126:      IOException error;
127:      int response_length;
128:
129:      @Override
130:      protected void onPreExecute() {
131:
132:          LoginUser.this.setUpProgressDialog(R.string.loggingIn);
133:
134:      }
135:
136:      @Override
137:      protected Integer doInBackground(String... url) {
138:          List<NameValuePair> params;
139:
140:          initializeConnectionElements();
141:          if (manager) {
142:              params = setUpHTTPParametersForManager(url[0]);
143:          } else
144:              params = setUpHTTPParametersForTeacher(url[0]);
145:
146:          sendPOSTPetition(params);
147:
148:          String login_response = retrieveResponse(url[0]);
149:          if (error != null) {
150:              return null;
151:          }
152:          response_length = login_response.length();
153:
154:          if (login_response != null) {
155:              saveSessionContext();
156:              saveLoginPreferences();
157:          }
158:
159:          return null;
160:      }
161:
162:      protected void onPostExecute(Integer i) {
163:
164:          if (error != null) {
165:              launchErrorWindow(1);
166:          } else {
167:              if (response_length == 2172 || response_length == 2173)
168:                  setUpLoginAlertDialogError();
169:              } else {
170:                  dismissWindow();
171:
172:                  manager = getIntent().getExtras().getBoolean(RO
LE);
173:
174:                  if (manager)
175:                      startActivity(new Intent(baseContext, S
electTask.class));
176:                  else
177:                      startActivity(new Intent(baseContext, S
earchDegrees.class));
178:
179:
180:              }
181:          }
182:          return;
183:      }

```

```

184:
185:     private void saveSessionContext() {
186:         GlobalState gs = (GlobalState) getApplication();
187:         gs.setCookieStore(cookieStore);
188:         gs.setHttpClient(client);
189:         gs.setLocalContext(localContext);
190:         gs.setRole(manager);
191:         gs.setBaseAddress(target_addr);
192:
193:     }
194:
195:     @SuppressWarnings("CommitPrefEdits")
196:     private void saveLoginPreferences() {
197:
198:         SharedPreferences settings = getSharedPreferences(USER_INFO, 0)
;
199:         SharedPreferences.Editor editor = settings.edit();
200:         boolean remember_next_session = settings.getBoolean(REMEMBER_US
ER,
201:             false);
202:
203:         if (remember_next_session) {
204:             saveLoginSession(editor);
205:         } else
206:             deleteLoginSession(editor);
207:
208:     }
209:
210:     private void deleteLoginSession(Editor editor) {
211:         editor.remove(LOGIN).remove(PASSWORD).remove(ROLE_PREFERENCES)
212:             .remove(REMEMBER_USER).commit();
213:     }
214:
215:     private void saveLoginSession(Editor editor) {
216:         EditText login = (EditText) findViewById(R.id.login_message);
217:         EditText passwd = (EditText) findViewById(R.id.pass_message);
218:         editor.putString(LOGIN, login.getText().toString())
219:             .putString(PASSWORD, passwd.getText().toString(
))
220:             .putBoolean(ROLE_PREFERENCES, manager).commit()
;
221:     }
222:
223:     private String retrieveResponse(String addr) {
224:         HttpGet httpGet = new HttpGet(addr + RESOURCE_URL);
225:         HttpResponse responseGET = null;
226:         String xml_response = null;
227:         try {
228:             responseGET = client.execute(httpGet, localContext);
229:             HttpEntity getEntity = responseGET.getEntity();
230:
231:             if (getEntity != null) {
232:                 xml_response = EntityUtils.toString(getEntity);
233:             }
234:
235:         } catch (ClientProtocolException e) {
236:             e.printStackTrace();
237:         } catch (IOException e) {
238:             error = e;
239:             e.printStackTrace();
240:
241:             return null;
242:         }

```

```

243:         return xml_response;
244:     }
245: }
246:
247: private void sendPOSTpetition(List<NameValuePair> post_params) {
248:     UrlEncodedFormEntity ent = null;
249:     try {
250:         ent = new UrlEncodedFormEntity(post_params, HTTP.UTF_8)
;
251:         post.setEntity(ent);
252:         HttpResponse responsePOST = null;
253:         responsePOST = client.execute(post, localContext);
254:         responsePOST.getEntity().consumeContent();
255:
256:     } catch (UnsupportedEncodingException e) {
257:         e.printStackTrace();
258:     } catch (ClientProtocolException e) {
259:         e.printStackTrace();
260:     } catch (IOException e) {
261:         e.printStackTrace();
262:     }
263: }
264:
265: private List<NameValuePair> setUpHTTPParametersForTeacher(String addr)
{
266:     EditText login = (EditText) findViewById(R.id.login_message);
267:     EditText passwd = (EditText) findViewById(R.id.pass_message);
268:     List<NameValuePair> params = new ArrayList<NameValuePair>();
269:     params.add(new BasicNameValuePair("utf8", "â\234\223"));
270:     post = new HttpPost(addr + TEACHER_SESSION);
271:     params.add(new BasicNameValuePair(PARAM_DNI, login.getText()
272:         .toString()));
273:     params.add(new BasicNameValuePair(PARAM_PASSWD, passwd.getText(
)
274:         .toString()));
275:     return params;
276: }
277:
278: private List<NameValuePair> setUpHTTPParametersForManager(String addr)
{
279:     EditText login = (EditText) findViewById(R.id.login_message);
280:     EditText passwd = (EditText) findViewById(R.id.pass_message);
281:     List<NameValuePair> params = new ArrayList<NameValuePair>();
282:     params.add(new BasicNameValuePair("utf8", "â\234\223"));
283:     post = new HttpPost(addr + MANAGER_SESSION);
284:     params.add(new BasicNameValuePair(PARAM_LOGIN, login.getText()
285:         .toString()));
286:     params.add(new BasicNameValuePair(PARAM_PASSWD, passwd.getText(
)
287:         .toString()));
288:     return params;
289: }
290:
291: private void initializeConnectionElements() {
292:     localContext = new BasicHttpContext();
293:     cookieStore = new BasicCookieStore();
294:     localContext.setAttribute(ClientContext.COOKIE_STORE, cookieSto
re);
295:     client = new DefaultHttpClient();
296: }
297:
298: }
299:

```

```

300:     private void setStoredUserDataOnEditBoxes() {
301:         SharedPreferences settings = getSharedPreferences(USER_INFO, 0);
302:
303:         CheckBox check_box = (CheckBox) findViewById(R.id.rememberButton);
304:         EditText user = (EditText) findViewById(R.id.login_message);
305:         EditText passwd = (EditText) findViewById(R.id.pass_message);
306:
307:         check_box.setChecked(true);
308:         user.setText(settings.getString(LOGIN, ""));
309:         passwd.setText(settings.getString(PASSWORD, ""));
310:     }
311:
312:     private boolean rememberUserFeatureisOn() {
313:         SharedPreferences settings = getSharedPreferences(USER_INFO, 0);
314:         return settings.getBoolean(REMEMBER_USER, false);
315:     }
316:
317:
318:     private void setRoleMessageOnScreen() {
319:
320:         getWindow().setSoftInputMode(
321:             WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
322:
323:         TextView textRole = (TextView) findViewById(R.id.login_manager);
324:
325:         manager = getIntent().getExtras().getBoolean(ROLE);
326:
327:         if (manager)
328:             textRole.setText(R.string.accessMessageManager);
329:         else
330:             textRole.setText(R.string.accessMessageTeacher);
331:
332:     }
333:
334:     /**
335:      * Set up the {@link android.app.ActionBar}, if the API is available.
336:      */
337:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
338:     private void setupActionBar() {
339:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
340:             getActionBar().setDisplayHomeAsUpEnabled(true);
341:         }
342:     }
343:
344:     @Override
345:     public boolean onCreateOptionsMenu(Menu menu) {
346:         // Inflate the menu; this adds items to the action bar if it is present
347:
348:         getMenuInflater().inflate(R.menu.login_user, menu);
349:         return true;
350:     }
351:
352:     @Override
353:     public boolean onOptionsItemSelected(MenuItem item) {
354:         switch (item.getItemId()) {
355:             // This ID represents the Home or Up button. In the case of thi
356:             // activity, the Up button is shown. Use NavUtils to allow user
357:             // to navigate up one level in the application structure. For
358:             // more details, see the Navigation pattern on Android Design:

```

./LoginUser.java

Wed Nov 06 12:11:56 2013

7

```
359:                                     //
360:                                     // http://developer.android.com/design/patterns/navigation.html
#up-vs-back
361:                                     //
362:                                     NavUtils.navigateUpFromSameTask(this);
363:                                     return true;
364:                                     }
365: return super.onOptionsItemSelected(item);
366:     }
367:
368: }
```

```
1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.Menu;
8: import android.view.MenuItem;
9:
10: public class AcademicInfo extends MyFragmentActivity {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.academic_info_activity);
16:     }
17:
18:     /**
19:      * Set up the {@link android.app.ActionBar}, if the API is available.
20:      */
21:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
22:     private void setupActionBar() {
23:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
24:             getActionBar().setDisplayHomeAsUpEnabled(true);
25:         }
26:     }
27:
28:     @Override
29:     public boolean onCreateOptionsMenu(Menu menu) {
30:         // Inflate the menu; this adds items to the action bar if it is present
31:
32:         getMenuInflater().inflate(R.menu.academic_info, menu);
33:         return true;
34:     }
35:
36:     @Override
37:     public boolean onOptionsItemSelected(MenuItem item) {
38:         switch (item.getItemId()) {
39:             // This ID represents the Home or Up button. In the case of thi
40:             // activity, the Up button is shown. Use NavUtils to allow user
41:             // to navigate up one level in the application structure. For
42:             // more details, see the Navigation pattern on Android Design:
43:             //
44:             // http://developer.android.com/design/patterns/navigation.html
45:             //
46:             NavUtils.navigateUpFromSameTask(this);
47:             return true;
48:         }
49:         return super.onOptionsItemSelected(item);
50:     }
51:
52: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.app.Activity;
4: import android.os.Bundle;
5: import android.support.v4.app.Fragment;
6: import android.view.LayoutInflater;
7: import android.view.View;
8: import android.view.ViewGroup;
9: import android.view.WindowManager;
10:
11: /**
12:  *
13:  * @author ubay
14:  *
15:  * Clase que muestra la lista de opciones principal de cada propuesta,
16:  * implementando una interfaz por cada boton pulsable para acceder a la
17:  * siguiente pantalla
18:  */
19: public class DegreeMainMenuFragment extends Fragment {
20:     OnAcademicInfoClickedListener mAcademicCallback;
21:     OnEconomicInfoClickedListener mEconomicCallback;
22:     OnTeachersClickedListener mTeachersCallback;
23:     OnSubjectsClickedListener mSubjectsCallback;
24:     OnReportClickedListener mReportCallback;
25:
26:     public interface OnAcademicInfoClickedListener {
27:         public void OnAcademicInfoClicked();
28:     }
29:
30:     public interface OnEconomicInfoClickedListener {
31:         public void OnEconomicInfoClicked();
32:     }
33:
34:     public interface OnTeachersClickedListener {
35:         public void OnTeachersClicked();
36:     }
37:
38:     public interface OnSubjectsClickedListener {
39:         public void OnSubjectsClicked();
40:     }
41:
42:     public interface OnReportClickedListener {
43:         public void OnReportClicked();
44:     }
45:
46:     @Override
47:     public void onAttach(Activity activity) {
48:         super.onAttach(activity);
49:
50:         /*
51:          * Esta secci3n se asegura de que el listener est3; implementado antes
de
52:          * asociar el fragmento a la actividad
53:          */
54:         try {
55:             mAcademicCallback = (OnAcademicInfoClickedListener) activity;
56:         } catch (ClassCastException e) {
57:             throw new ClassCastException(activity.toString()
58:                 + " must implement OnAcademicInfoClickedListene
r");
59:         }
60:
61:         try {

```

```
62:         mEconomicCallback = (OnEconomicInfoClickedListener) activity;
63:     } catch (ClassCastException e) {
64:         throw new ClassCastException(activity.toString()
65:             + " must implement OnEconomicInfoClickedListene
r");
66:     }
67:
68:     try {
69:         mTeachersCallback = (OnTeachersClickedListener) activity;
70:     } catch (ClassCastException e) {
71:         throw new ClassCastException(activity.toString()
72:             + " must implement OnTeachersClickedListener");
73:     }
74:     try {
75:         mSubjectsCallback = (OnSubjectsClickedListener) activity;
76:     } catch (ClassCastException e) {
77:         throw new ClassCastException(activity.toString()
78:             + " must implement OnSubjectsClickedListener");
79:     }
80:     try {
81:         mReportCallback = (OnReportClickedListener) activity;
82:     } catch (ClassCastException e) {
83:         throw new ClassCastException(activity.toString()
84:             + " must implement OnReportClickedListener");
85:     }
86: }
87:
88: public View onCreateView(LayoutInflater inflater, ViewGroup container,
89:     Bundle savedInstanceState) {
90:
91:     View myFragmentView = inflater.inflate(
92:         R.layout.degree_main_menu_fragment, container, false);
93:
94:     getActivity().getWindow().setSoftInputMode(
95:         WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDD
EN);
96:     return myFragmentView;
97: }
98: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4: import java.util.HashMap;
5:
6: /**
7:  *
8:  * @author ubay
9:  *
10:  *     Clase contenedora de la informacion de cada propuesta para la lista
11:  *     de propuestas en SearchDegrees
12:  */
13: public class DegreeInfoContainer {
14:     private String name;
15:     private String edition;
16:     private String state;
17:     private String code;
18:     private String type;
19:     private String id;
20:
21:     public DegreeInfoContainer(HashMap<String, String> input) {
22:         name = input.get("name");
23:         code = input.get("code");
24:         if (input.containsKey("lectures-start")) {
25:             if (input.get("lectures-start") != null) {
26:                 edition = input.get("lectures-start");
27:             } else
28:                 edition = "";
29:         } else
30:             edition = "";
31:         type = input.get("degree-type-id");
32:         state = input.get("state");
33:         id = input.get("id");
34:     }
35:
36:     public String getId() {
37:         return this.id;
38:     }
39:
40:     public String getName() {
41:         return this.name;
42:     }
43:
44:     public String getCode() {
45:         return this.code;
46:     }
47:
48:     public String getEdition() {
49:         return this.edition;
50:     }
51:
52:     public String getType() {
53:         return this.type;
54:     }
55:
56:     public String getState() {
57:         return this.state;
58:     }
59:
60:     public ArrayList<String> getArrayList() {
61:         ArrayList<String> output = new ArrayList<String>();
62:
63:         output.add(name);
```

```
64:         output.add(type);
65:         output.add(state);
66:         output.add(edition);
67:         output.add(code);
68:         output.add(id);
69:
70:         return output;
71:     }
72: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.ArrayList;
5: import java.util.HashMap;
6:
7: import android.content.Intent;
8: import android.os.AsyncTask;
9: import android.os.Bundle;
10: import android.os.Handler;
11: import android.os.Message;
12: import android.support.v4.app.Fragment;
13: import android.text.Html;
14: import android.view.LayoutInflater;
15: import android.view.View;
16: import android.view.View.OnClickListener;
17: import android.view.ViewGroup;
18: import android.widget.Button;
19: import android.widget.ListView;
20:
21: /**
22:  *
23:  * @author ubay
24:  *
25:  * TeachersListFragment recibe una id de la actividad padre, MainMenu, y
26:  * descarga la lista de participaciones asociada a la propuesta que
27:  * coincide con el id. A continuacion, extrae la id de cada profesor
28:  * contenida en cada participacion asociada a la propuesta. Por ultimo,
29:  * ira poblando de manera asincrona a medida que lleguen los documentos
30:  * de profesor uno a uno en el listado del layout de esta actividad
31:  */
32: public class TeachersListFragment extends Fragment implements
33:     CustomTeacherAdapter.OnTeacherClickedListener, OnClickListener {
34:     private ArrayList<ArrayList<String>> fetch = new ArrayList<ArrayList<String>>()
;
35:     private ArrayList<String> arrayList_degreeTeachers;
36:     private CustomTeacherAdapter custom_adapter;
37:     private Handler mDegreeTeachersHandler;
38:     private Handler mFullTeachersHandler;
39:     private ListView listView;
40:     private View myView;
41:     private Button footer;
42:     private int total_teachers;
43:     private int current_teachers;
44:     private Runnable run;
45:     private ArrayList<HashMap<String, String>> completeDegreeTeachers = new ArrayLi
st<HashMap<String, String>>();
46:     private AsyncTask<String, Void, Void> first_task;
47:     private AsyncTask<String, Integer, Void> second_task;
48:
49:     @Override
50:     public void onStop() {
51:         super.onStop();
52:         if (first_task != null) {
53:             first_task.cancel(true);
54:         }
55:         if (second_task != null) {
56:             second_task.cancel(true);
57:         }
58:     }
59:
60:     @Override
61:     public void onDestroyView() {
```

```

62:         super.onDestroyView();
63:         if (first_task != null) {
64:             first_task.cancel(true);
65:         }
66:         if (second_task != null) {
67:             second_task.cancel(true);
68:         }
69:     }
70:
71:     @Override
72:     public void onDestroy() {
73:         super.onDestroy();
74:         if (first_task != null) {
75:             first_task.cancel(true);
76:         }
77:         if (second_task != null) {
78:             second_task.cancel(true);
79:         }
80:     }
81:
82:     @Override
83:     public void onDetach() {
84:         super.onDetach();
85:         if (first_task != null) {
86:             first_task.cancel(true);
87:         }
88:         if (second_task != null) {
89:             second_task.cancel(true);
90:         }
91:     }
92:
93:     @Override
94:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
95:         Bundle savedInstanceState) {
96:
97:         first_task = new retrieveDegreeParticipations();
98:         first_task.execute();
99:
100:         myView = inflater.inflate(R.layout.teachers_list_fragment, container,
101:             false);
102:         listview = (ListView) myView.findViewById(R.id.teachers_list);
103:         custom_adapter = new CustomTeacherAdapter(this, R.id.teachers_list,
104:             fetch, getActivity().getApplicationContext());
105:         listview.setAdapter(custom_adapter);
106:
107:         footer = (Button) myView.findViewById(R.id.amount_footer);
108:         footer.setOnClickListener(this);
109:
110:         mDegreeTeachersHandler = setUpDegreeTeachersHandler();
111:
112:         mFullTeachersHandler = setUpFullTeachersHandler();
113:
114:         run = new Runnable() {
115:             public void run() {
116:                 custom_adapter.notifyDataSetChanged();
117:                 footer.setText(Html.fromHtml("<b>" + current_teachers +
118:                 + total_teachers + " "
119:                 + getResources().getString(R.string.loadedTeachers)
120:                 + "</b>"));
121:
122:     }

```

```

123:         };
124:
125:         return myView;
126:     }
127:
128:     @Override
129:     public void onClick(View v) {
130:         switch (v.getId()) {
131:             case R.id.amount_footer: {
132:                 v.setVisibility(View.INVISIBLE);
133:                 break;
134:             }
135:         }
136:     }
137:
138:     private Handler setUpFullTeachersHandler() {
139:         return new Handler() {
140:             @SuppressWarnings("unchecked")
141:             @Override
142:             public void handleMessage(Message msg) {
143:                 /*
144:                  * ArrayList<HashMap<String, String>> temp =
145:                  * (ArrayList<HashMap<String, String>>) msg.obj; for (i
nt i = 0;
146:                  * i < temp.size(); i++) { TeacherInfoContainer teacher
_info =
147:                  * new TeacherInfoContainer( temp.get(i));
148:                  * custom_adapter.add(teacher_info.getArrayList()); }
149:                  */
150:                 TeacherInfoContainer teacher_info = new TeacherInfoCont
ainer(
151:                     (HashMap<String, String>) msg.obj);
152:                 custom_adapter.add(teacher_info.getArrayList());
153:                 getActivity().runOnUiThread(run);
154:             }
155:         };
156:     }
157:
158:     private Handler setUpDegreeTeachersHandler() {
159:         return new Handler() {
160:             @SuppressWarnings("unchecked")
161:             @Override
162:             public void handleMessage(Message msg) {
163:                 ArrayList<String> temp = (ArrayList<String>) msg.obj;
164:                 XmlExtractor xml_extractor;
165:                 String[] tags = getResources().getStringArray(
166:                     R.array.degree_teachers_info);
167:                 total_teachers = temp.size();
168:                 for (int i = 0; i < temp.size(); i++) {
169:
170:                     xml_extractor = new XmlExtractor(temp.get(i));
171:                     HashMap<String, String> single_teacher_part = n
ew HashMap<String, String>(
172:                         xml_extractor.extractTagsFromDo
c(tags));
173:                     completeDegreeTeachers.add(single_teacher_part)
;
174:                 }
175:                 second_task = new retrieveFullTeachers();
176:                 second_task.execute();
177:             }
178:         };
179:     }

```

```
180:
181:     private class retrieveDegreeParticipations extends
182:         AsyncTask<String, Void, Void> {
183:         IOException error;
184:
185:         protected Void doInBackground(String... params) {
186:             String target_url = setTargetUrl();
187:             XmlPage xml_retriever = new XmlPage(getActivity());
188:             String xml_response = null;
189:             try {
190:                 xml_response = xml_retriever.getPage(target_url);
191:             } catch (IOException e) {
192:                 error = e;
193:                 return null;
194:             }
195:
196:             XmlExtractor xml_extractor = new XmlExtractor(xml_response);
197:
198:             arrayList_degreeTeachers = new ArrayList<String>(
199:                 xml_extractor.unitSeparator("</degree-teacher>"
200: ));
201:             return null;
202:         }
203:
204:         private String setTargetUrl() {
205:             final String UNIV_SPEC_DEG = "university_specific_degrees";
206:             final String XML_EXTENSION = ".xml";
207:             final String SLASH = "/";
208:             final String DEGREE_TEACHERS = "degree_teachers";
209:
210:             GlobalState gs = (GlobalState) getActivity().getApplication();
211:             String base_url = gs.getTargetAddress();
212:
213:             String degree_id = "";
214:             Bundle bundle = getArguments();
215:
216:             if (bundle != null) {
217:                 degree_id = bundle.getString("id");
218:             } else {
219:                 degree_id = getActivity().getIntent().getStringExtra("i
220: d");
221:             }
222:
223:             String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_i
224: d
225:                 + SLASH + DEGREE_TEACHERS + XML_EXTENSION;
226:
227:             return target_url;
228:         }
229:
230:         protected void onPostExecute(Void param) {
231:             if (error != null) {
232:                 ((MyFragmentActivity) getActivity()).dismissWindow();
233:                 ((MyFragmentActivity) getActivity()).launchErrorWindow(
234: 0);
235:             } else {
236:                 Message msg = new Message();
237:                 msg.obj = arrayList_degreeTeachers;
238:                 mDegreeTeachersHandler.sendMessage(msg);
```

```

239:         }
240:     }
241: }
242: }
243:
244: private class retrieveFullTeachers extends AsyncTask<String, Integer, Void> {
245:     IOException error;
246:
247:     protected Void doInBackground(String... params) {
248:         XmlPage xml_retriever = new XmlPage(getActivity());
249:
250:         String[] tags = getResources().getStringArray(
251:             R.array.generic_teachers_info);
252:
253:         for (int i = 0; i < completeDegreeTeachers.size(); i++) {
254:
255:             String target_url = setTargetUrl(i);
256:
257:             String xml_response;
258:             try {
259:                 xml_response = xml_retriever.getPage(target_url
);
260:             } catch (IOException e) {
261:                 error = e;
262:                 return null;
263:             }
264:
265:             XmlExtractor xml_extractor = new XmlExtractor(xml_respo
nse);
266:             HashMap<String, String> extracted_teacher = xml_extract
or
267:                 .extractTagsFromDoc(tags);
268:             String teacher_type = determineTeacherTypeFromBaseXMLTa
g(xml_response);
269:             completeDegreeTeachers.get(i).putAll(extracted_teacher)
;
270:             completeDegreeTeachers.get(i).put("type", teacher_type)
;
271:             completeDegreeTeachers.get(i).put("whole", xml_response
);
272:             publishProgress(i);
273:         }
274:     }
275:
276:     return null;
277: }
278:
279:     protected void onProgressUpdate(Integer... values) {
280:
281:         Message msg = new Message();
282:         msg.obj = completeDegreeTeachers.get(values[0]);
283:         mFullTeachersHandler.sendMessage(msg);
284:         current_teachers = values[0] + 1;
285:     }
286: }
287:
288:     protected void onPostExecute(Void param) {
289:
290:         if (error != null) {
291:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
292:         }
293:     }

```

```

294:         }
295:
296:     private String determineTeacherTypeFromBaseXMLTag(String xml_response)
{
297:
298:         return xml_response.substring(40, 45);
299:     }
300:
301:     private String setTargetUrl(int i) {
302:         final String PARTICIPATIONS_URL = "teachers";
303:         final String XML_EXTENSION = ".xml";
304:         final String SLASH = "/";
305:
306:         GlobalState gs = (GlobalState) getActivity().getApplication();
307:         String base_url = gs.getTargetAddress();
308:
309:         String target_url = base_url + PARTICIPATIONS_URL + SLASH
310:             + completeDegreeTeachers.get(i).get("teacher-id
")
311:             + XML_EXTENSION;
312:         return target_url;
313:     }
314:
315: }
316:
317: @Override
318: public void OnTeacherClicked(String teacher_info) {
319:     if (getActivity().findViewById(R.id.lateral_fragment) == null) {
320:
321:         Intent intent = new Intent(getActivity(), SingleTeacher.class);
322:         intent.putExtra("teacher-info", teacher_info);
323:         startActivity(intent);
324:     } else {
325:         Bundle args = new Bundle();
326:         args.putString("teacher-info", teacher_info);
327:         SingleTeacherFragment singleTeacherFragment = new SingleTeacher
Fragment();
328:         singleTeacherFragment.setArguments(args);
329:         getActivity().getSupportFragmentManager().beginTransaction()
330:             .replace(R.id.lateral_fragment, singleTeacherFr
agment)
331:             .commit();
332:     }
333: }
334: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.ArrayList;
5:
6: import android.app.Activity;
7: import android.app.AlertDialog;
8: import android.content.DialogInterface;
9: import android.os.AsyncTask;
10: import android.os.Bundle;
11: import android.support.v4.app.Fragment;
12: import android.view.LayoutInflater;
13: import android.view.View;
14: import android.view.ViewGroup;
15: import android.view.WindowManager;
16: import android.widget.AdapterView;
17: import android.widget.AdapterView.OnItemClickListener;
18: import android.widget.ArrayAdapter;
19: import android.widget.Button;
20: import android.widget.EditText;
21: import android.widget.LinearLayout;
22: import android.widget.Spinner;
23:
24: /**
25:  *
26:  * @author ubay
27:  *
28:  *     La clase SearchDegreesFragment puebla un layout con opciones de
29:  *     busqueda, recibe las opciones del usuario y realiza la busqueda
30:  *     contra la aplicacion web. Muestra un mensaje con el numero de
31:  *     resultados obtenidos, da opcion de volver a buscar o mostrar los
32:  *     resultados obtenidos y se manda el listado de propuestas a la
33:  *     siguiente actividad SearchResults
34:  */
35: public class SearchDegreesFragment extends Fragment {
36:
37:     private ArrayAdapter<CharSequence> types_adapter, states_adapter,
38:         lectures_adapter;
39:     private String edicion, estado, tipo;
40:     private Spinner option1, option2, option3;
41:     private Button search_button;
42:     private Degree[] degree_list;
43:     private View myFragmentView;
44:     private OnShowResultsListener mCallback;
45:
46:     public interface OnShowResultsListener {
47:         public void onShowResults(Degree[] degrees);
48:     }
49:
50:     @Override
51:     public void onAttach(Activity activity) {
52:         super.onAttach(activity);
53:
54:         // This makes sure that the container activity has implemented
55:         // the callback interface. If not, it throws an exception
56:         try {
57:             mCallback = (OnShowResultsListener) activity;
58:         } catch (ClassCastException e) {
59:             throw new ClassCastException(activity.toString()
60:                 + " must implement OnShowResultsListener");
61:         }
62:     }
63:
```

```
64:         public View onCreateView(LayoutInflater inflater, ViewGroup container,
65:             Bundle savedInstanceState) {
66:
67:             myFragmentView = inflater.inflate(R.layout.search_degrees_fragment,
68:                 container, false);
69:
70:             getActivity().getWindow().setSoftInputMode(
71:                 WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDD
EN);
72:
73:             setSpinnerValues();
74:
75:             setSearchButton();
76:             return myFragmentView;
77:         }
78:
79:         private class search extends AsyncTask<String, Integer, Integer> {
80:             private static final String PAGE_URL = "page=";
81:             private IOException error;
82:
83:             @Override
84:             protected void onPreExecute() {
85:                 super.onPreExecute();
86:
87:                 ((MyFragmentActivity) getActivity())
88:                     .setUpProgressDialog(R.string.searching);
89:
90:             }
91:
92:             @Override
93:             protected Integer doInBackground(String... params) {
94:
95:                 int page_number = 1;
96:                 String base_url = params[0];
97:                 String resource_url = params[1];
98:
99:                 String target_url = base_url + PAGE_URL + page_number
100:                     + resource_url;
101:
102:                 String response = getXmlPage(target_url);
103:                 if (response == null) {
104:
105:                     return 0;
106:                 }
107:
108:                 DegreePage degrees_page = new DegreePage(response);
109:                 ArrayList<String> degrees = new ArrayList<String>();
110:
111:                 while (!degrees_page.isEmpty()) {
112:                     degrees.addAll(degrees_page.trimXML());
113:                     target_url = setNewPageAddress(base_url, resource_url,
114:                         ++page_number);
115:                     response = getXmlPage(target_url);
116:                     if (response == null) {
117:                         return null;
118:                     }
119:                     degrees_page = new DegreePage(response);
120:                     if (page_number > 2)
121:                         publishProgress(page_number - 2);
122:
123:                 }
124:                 degree_list = new Degree[degrees.size()];
125:
```

```

126:         for (int i = 0; i < degrees.size(); i++) {
127:             degree_list[i] = new Degree(degrees.get(i), getActivity
128: ()
129:             .getApplication());
130:         }
131:
132:         return degrees.size();
133:     }
134:
135:     protected void onProgressUpdate(Integer... values) {
136:         super.onProgressUpdate(values);
137:
138:         ((MyFragmentActivity) getActivity())
139:             .updateProgressDialog(getResources().getString(
140:                 R.string.searchMessage_before)
141:                 + " "
142:                 + Integer.toString(values[0] *
20)
143:                 + " "
144:                 + getResources().getString(
145:                     R.string.search
Message_after));
146:     }
147:
148:
149:     @Override
150:     protected void onPostExecute(Integer i) {
151:         ((MyFragmentActivity) getActivity()).dismissWindow();
152:
153:         if (error != null) {
154:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
2)
155:         }
156:         } else {
157:             showAlertDialog(i);
158:         }
159:     }
160:
161:
162:     private void showAlertDialog(Integer final_result_number) {
163:         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder
r(
164:             getActivity());
165:
166:         alertDialogBuilder.setTitle(getResources().getString(
167:             R.string.searchResults_before)
168:             + " "
169:             + final_result_number
170:             + " "
171:             + getResources().getString(R.string.searchResul
ts_after));
172:
173:         alertDialogBuilder
174:             .setMessage(R.string.finishSearch)
175:             .setCancelable(false)
176:             .setPositiveButton(R.string.showResults,
177:                 new DialogInterface.OnClickListener() {
178:                     public void onClick(Dia
logInterface dialog,
179:                                     int id)
{

```

```

180:                                                                 mCallback.onSho
wResults(degree_list);
181:                                                                 }
182:                                                                 })
183:                                                                 .setNegativeButton(R.string.repeatSearch,
184:                                                                 new DialogInterface.OnClickListener
ener() {
185:                                                                 public void onClick(Dia
logInterface dialog,
186:                                                                 int id)
{
187:                                                                 getActivity().f
inish();
188:                                                                 startActivity(g
etActivity().getIntent());
189:                                                                 }
190:                                                                 });
191:                                                                 AlertDialog alertDialog = alertDialogBuilder.create();
192:                                                                 alertDialog.show();
193:                                                                 }
194:
195:                                                                 private String setNewPageAddress(String base_url, String resource_url,
196:                                                                 int page_number) {
197:                                                                 return base_url + PAGE_URL + page_number + resource_url;
198:                                                                 }
199:
200:                                                                 private String getXmlPage(String target_url) {
201:                                                                 XmlPage xml_page = new XmlPage(getActivity());
202:                                                                 String xml_response = null;
203:                                                                 try {
204:                                                                 xml_response = xml_page.getPage(target_url);
205:                                                                 } catch (IOException e) {
206:                                                                 error = e;
207:                                                                 return null;
208:                                                                 }
209:
210:                                                                 return xml_response;
211:                                                                 }
212:
213:                                                                 }
214:
215:                                                                 private void setSearchButton() {
216:                                                                 search_button = (Button) myFragmentView
217:                                                                 .findViewById(R.id.search_button);
218:                                                                 search_button.setOnClickListener(new View.OnClickListener() {
219:                                                                 public void onClick(View v) {
220:
221:                                                                 String[] search_url = createSearchURLFromInput();
222:
223:                                                                 new search().execute(search_url);
224:                                                                 }
225:
226:                                                                 private String[] createSearchURLFromInput() {
227:                                                                 final String BASE_URL = "university_specific_degrees.xml
l?";
228:                                                                 final String SEARCH_BY_CODE = "&search[code_like]=";
229:                                                                 final String SEARCH_BY_NAME = "&search[name_like]=";
230:                                                                 final String SEARCH_BY_DATE = "&search[period_id_eq]=";
231:                                                                 final String SEARCH_BY_TYPE = "&search[degree_type_id_e
q]=";
232:                                                                 final String SEARCH_BY_STAT = "&search[state_in][]=";
233:                                                                 final String COMMIT = "&commit=Buscar";
234:

```

```
235:         GlobalState gs = (GlobalState) getActivity().getApplica
tion();
236:         String base_url = new String(gs.getTargetAddress());
237:         String resource_url;
238:
239:         int edition, type;
240:         String state, code, name;
241:
242:         name = getDegreeName();
243:         code = getDegreeCode();
244:         edition = getDegreeEditionNumber();
245:         type = getDegreeType();
246:         state = getDegreeState();
247:
248:         base_url += BASE_URL;
249:         resource_url = SEARCH_BY_CODE + code;
250:         resource_url += SEARCH_BY_NAME + name;
251:
252:         resource_url += SEARCH_BY_DATE;
253:         if (edition != 0) {
254:             resource_url += edition;
255:         }
256:
257:         resource_url += SEARCH_BY_TYPE;
258:         if (type != 0) {
259:             resource_url += type;
260:         }
261:         if (!state.equals("")) {
262:             resource_url += SEARCH_BY_STAT + state;
263:         }
264:         resource_url += COMMIT;
265:
266:         String[] output_urls = new String[2];
267:         output_urls[0] = base_url;
268:         output_urls[1] = resource_url;
269:         return output_urls;
270:     }
271:
272:     private String getDegreeCode() {
273:         EditText search_bycode;
274:         search_bycode = (EditText) myFragmentManager
                .findViewById(R.id.busqueda_codigo);
275:
276:
277:         return search_bycode.getText().toString();
278:     }
279:
280:     private String getDegreeName() {
281:         EditText search_byname;
282:
283:         search_byname = (EditText) myFragmentManager
                .findViewById(R.id.busqueda_nombre);
284:
285:         return search_byname.getText().toString();
286:     }
287:
288:
289:     private String getDegreeState() {
290:         if (estado.equals("Borrador"))
291:             return "borrador";
292:         else if (estado.equals("Enviado"))
293:             return "enviado";
294:         else if (estado.equals("Rechazado"))
295:             return "rechazado";
296:         else if (estado.equals("Aprobado"))
```

```
297:         return "aprobado";
298:     else if (estado.equals("En subsanaciÃ³n"))
299:         return "subsanacion";
300:     else if (estado.equals("Cancelado"))
301:         return "cancelado";
302:     else
303:         return "";
304: }
305:
306: private int getDegreeType() {
307:     if (tipo.equals("Enseñanza Propia Básica"))
308:         return 1;
309:     else if (tipo.equals("Enseñanza Propia Superior"))
310:         return 2;
311:     else if (tipo.equals("Experto"))
312:         return 3;
313:     else if (tipo.equals("Maestría"))
314:         return 4;
315:     else if (tipo.equals("Diploma Profesional"))
316:         return 5;
317:     else
318:         return 0;
319: }
320:
321: private int getDegreeEditionNumber() {
322:     if (edicion.equals("2011/2012"))
323:         return 2;
324:     else if (edicion.equals("2012/2013"))
325:         return 3;
326:     else if (edicion.equals("2013/2014"))
327:         return 4;
328:     else if (edicion.equals("2014/2015"))
329:         return 5;
330:     else
331:         return 0;
332: }
333:     });
334: }
335:
336: private void setSpinnerValues() {
337:
338:     createSpinnersFromResources();
339:     connectSpinnersWithAdapters();
340:
341: }
342:
343: private void connectSpinnersWithAdapters() {
344:     final String NONE_SELECTED = "nothing";
345:
346:     LinearLayout linear_layout = (LinearLayout) myFragmentManager
347:         .findViewById(R.id.base_linear_search);
348:
349:     option1 = (Spinner) linear_layout.findViewById(R.id.edicion_spinner);
350:     option2 = (Spinner) linear_layout.findViewById(R.id.tipo_spinner);
351:     option3 = (Spinner) linear_layout.findViewById(R.id.estado_spinner);
352:
353:     option1.setAdapter(types_adapter);
354:     option2.setAdapter(lectures_adapter);
355:     option3.setAdapter(states_adapter);
356:
357:     option1.setOnItemClickListener(new OnItemSelectedListener() {
358:         public void onItemClick(AdapterView<?> parentView,
359:             View selectedItemView, int position, long id) {
```

```
360:         tipo = (String) types_adapter.getItem(position);
361:     }
362:
363:     public void onNothingSelected(AdapterView<?> parentView) {
364:         tipo = NONE_SELECTED;
365:     }
366: });
367: option2.setOnItemSelectedListener(new OnItemSelectedListener() {
368:     public void onItemSelected(AdapterView<?> parentView,
369:         View selectedItemView, int position, long id) {
370:         edicion = (String) lectures_adapter.getItem(position);
371:     }
372:
373:     public void onNothingSelected(AdapterView<?> parentView) {
374:         edicion = NONE_SELECTED;
375:     }
376: });
377: option3.setOnItemSelectedListener(new OnItemSelectedListener() {
378:     public void onItemSelected(AdapterView<?> parentView,
379:         View selectedItemView, int position, long id) {
380:         estado = (String) states_adapter.getItem(position);
381:     }
382:
383:     public void onNothingSelected(AdapterView<?> parentView) {
384:         estado = NONE_SELECTED;
385:     }
386: });
387: }
388:
389: private void createSpinnersFromResources() {
390:     types_adapter = ArrayAdapter.createFromResource(getActivity(),
391:         R.array.degree_types, android.R.layout.simple_spinner_i
tem);
392:     states_adapter = ArrayAdapter.createFromResource(getActivity(),
393:         R.array.degree_states, android.R.layout.simple_spinner_
item);
394:     lectures_adapter = ArrayAdapter.createFromResource(getActivity(),
395:         R.array.degree_editions, android.R.layout.simple_spinne
r_item);
396:
397:     types_adapter
398:         .setDropDownViewResource(android.R.layout.simple_spinne
r_dropdown_item);
399:     states_adapter
400:         .setDropDownViewResource(android.R.layout.simple_spinne
r_dropdown_item);
401:     lectures_adapter
402:         .setDropDownViewResource(android.R.layout.simple_spinne
r_dropdown_item);
403:     }
404: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.HashMap;
4: import android.annotation.SuppressLint;
5: import android.graphics.Typeface;
6: import android.os.Bundle;
7: import android.support.v4.app.Fragment;
8: import android.text.Html;
9: import android.view.Gravity;
10: import android.view.LayoutInflater;
11: import android.view.View;
12: import android.view.ViewGroup;
13: import android.widget.LinearLayout;
14: import android.widget.ScrollView;
15: import android.widget.TextView;
16:
17: /**
18:  *
19:  * @author ubay
20:  *
21:  *     TabIncome representa una de las pestañas de las tres que
22:  *     componen la informacion economica. Utiliza la informacion obtenida de
23:  *     la actividad padre para poblar el layout correspondiente
24:  */
25: public class TabIncome extends Fragment {
26:     /**
27:      * (non-Javadoc)
28:      *
29:      * @see android.support.v4.app.Fragment#onCreateView(android.view.LayoutInflater
r,
30:      *     android.view.ViewGroup, android.os.Bundle)
31:      */
32:
33:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
34:         Bundle savedInstanceState) {
35:         if (container == null) {
36:             return null;
37:         }
38:
39:         HashMap<String, String> info;
40:
41:         if (getActivity()
42:             .getClass()
43:             .toString()
44:             .equals("class es.ulpgc.titulospropios.gestion.ubay.Deg
reeMainMenu")) {
45:             info = EconomicInfoFragment.hashMap;
46:         } else {
47:             info = EconomicInfo.hashMap;
48:         }
49:         ScrollView outer_layout = (ScrollView) inflater.inflate(
50:             R.layout.degree_income_tab, container, false);
51:         return populateViews(info, outer_layout);
52:
53:     }
54:
55:     @SuppressWarnings("CutPasteId")
56:     private View populateViews(HashMap<String, String> info,
57:         ScrollView outer_layout) {
58:         int[] views = { R.id.first_row, R.id.second_row, R.id.students_sum,
59:             R.id.direct_income_row };
60:
61:         String[] tags = getResources().getStringArray(R.array.incomes);

```

```

62: String[] etiquetas = getResources().getStringArray(R.array.incomeTags);
63:
64: LinearLayout inner_layout = null;
65: float student_incomes = 0;
66: for (int i = 0; i <= 1; i++) {
67:     inner_layout = (LinearLayout) outer_layout.findViewById(views[i
]);
68:     TextView left_item = (TextView) inner_layout
69:         .findViewById(R.id.left_item);
70:     TextView center = (TextView) inner_layout.findViewById(R.id.cen
ter);
71:     TextView right_item = (TextView) inner_layout
72:         .findViewById(R.id.right_item);
73:
74:     left_item.setText(Html.fromHtml(etiquetas[i * 2]
75:         + info.get(tags[i * 2])));
76:     center.setText(Html.fromHtml(etiquetas[i * 2 + 1]
77:         + info.get(tags[i * 2 + 1])));
78:     float income = (Float.parseFloat(info.get(tags[i * 2])) * Float
79:         .parseFloat(info.get(tags[i * 2 + 1])));
80:     student_incomes += income;
81:     left_item.setGravity(Gravity.CENTER);
82:     center.setGravity(Gravity.CENTER);
83:     right_item.setText(Html.fromHtml(etiquetas[4]
84:         + Float.toString(income) + " â\202r");
85:     right_item.setGravity(Gravity.RIGHT);
86: }
87: inner_layout = (LinearLayout) outer_layout
88:     .findViewById(R.id.students_sum);
89: TextView student_sum_tag = (TextView) inner_layout
90:     .findViewById(R.id.left_item);
91: TextView student_sum = (TextView) inner_layout
92:     .findViewById(R.id.right_item);
93: student_sum_tag.setText(R.string.student_sum);
94: student_sum_tag.setGravity(Gravity.CENTER);
95: student_sum_tag.setTypeface(null, Typeface.BOLD);
96: student_sum.setText(Float.toString(student_incomes) + " â\202r");
97: student_sum.setGravity(Gravity.RIGHT);
98:
99: inner_layout = (LinearLayout) outer_layout
100:     .findViewById(R.id.direct_income_row);
101: TextView external_income = (TextView) inner_layout
102:     .findViewById(R.id.left_item);
103: TextView external_income_sum = (TextView) inner_layout
104:     .findViewById(R.id.right_item);
105:
106: external_income.setText(R.string.external_incomes);
107: external_income.setGravity(Gravity.CENTER);
108: external_income.setTypeface(null, Typeface.BOLD);
109:
110: external_income_sum.setText(info.get(tags[4]) + " â\202r");
111: external_income_sum.setGravity(Gravity.RIGHT);
112:
113: return outer_layout;
114: }
115: }

```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import java.util.HashMap;
4:
5: import android.os.Bundle;
6: import android.support.v4.app.Fragment;
7: import android.text.Html;
8: import android.util.Log;
9: import android.view.Gravity;
10: import android.view.LayoutInflater;
11: import android.view.View;
12: import android.view.ViewGroup;
13: import android.widget.LinearLayout;
14: import android.widget.ScrollView;
15: import android.widget.TextView;
16:
17: /**
18:  *
19:  * @author ubay
20:  *
21:  *     TabExpenditure representa una de las pestañas de las tres que
22:  *     componen la informacion economica. Utiliza la informacion obtenida de
23:  *     la actividad padre para poblar el layout correspondiente
24:  */
25: public class TabExpenditure extends Fragment {
26:     /**
27:      * (non-Javadoc)
28:      *
29:      * @see android.support.v4.app.Fragment#onCreateView(android.view.LayoutInflater
r,
30:      *     android.view.ViewGroup, android.os.Bundle)
31:      */
32:     public static final String ANECA_COST = "550.0";
33:     public static final float INSURANCE_PER_STUDENT = 15;
34:
35:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
36:         Bundle savedInstanceState) {
37:         if (container == null) {
38:             return null;
39:         }
40:         ScrollView outer_layout = (ScrollView) inflater.inflate(
41:             R.layout.degree_expenditures_tab, container, false);
42:
43:         HashMap<String, String> info;
44:
45:         if (getActivity()
46:             .getClass()
47:             .toString()
48:             .equals("class es.ulpgc.titulosproprios.gestion.ubay.Deg
reeMainMenu")) {
49:             info = EconomicInfoFragment.hashMap;
50:         } else {
51:             info = EconomicInfo.hashMap;
52:         }
53:
54:         return populateViews(info, outer_layout);
55:     }
56:
57:
58:     private View populateViews(HashMap<String, String> info,
59:         ScrollView outer_layout) {
60:         final int[] text_views = { R.id.teory_ulpgc, R.id.practice_ulpgc,
61:             R.id.seminars_ulpgc, R.id.exams_ulpgc, R.id.content_ulp

```

```

gc,
62:         R.id.tutorship_ulpgc, R.id.final_project_ulpgc,
63:         R.id.exam_coordinator_ulpgc, R.id.directors, R.id.cost_
sum,
64:         R.id.unit_profit, R.id.cost_total, R.id.teory, R.id.pra
ctice,
65:         R.id.seminars, R.id.exams, R.id.content, R.id.tutorship
',
66:         R.id.final_project, R.id.exam_coordinator,
67:         R.id.others_external, R.id.external_cost_total, R.id.st
aff,
68:         R.id.scholarship_students, R.id.scholarship_specific,
69:         R.id.other_staff, R.id.presentation_report, R.id.final_
report,
70:         R.id.information_point, R.id.support, R.id.other_suppor
t,
71:         R.id.travel_cost, R.id.hosting_cost, R.id.food_cost,
72:         R.id.scholarship_grant, R.id.other_travelcost,
73:         R.id.travel_total, R.id.office_material, R.id.lab_materi
ial,
74:         R.id.non_inventory, R.id.reprography, R.id.other_materi
al,
75:         R.id.fungible_total, R.id.bibliography_cost,
76:         R.id.other_bibliography, R.id.bibliography_total,
77:         R.id.print_ads, R.id.press_ads, R.id.ads_total,
78:         R.id.inauguration, R.id.translation, R.id.certs,
79:         R.id.virtual_campus, R.id.evaluation, R.id.insurance,
80:         R.id.other_other, R.id.others_total, R.id.total_expense
s };
81:         final String EURO = " â\202-";
82:
83:         String[] tags = getResources().getStringArray(R.array.economic_tags);
84:         String[] etiquetas = getResources()
85:             .getStringArray(R.array.expensesTags);
86:
87:         for (int i = 0; i < etiquetas.length; i++) {
88:             Log.d("Etiqueta: " + etiquetas[i], "Tag: " + tags[i]);
89:         }
90:         int count = 0;
91:         for (int i = 0; i < 58; i++) {
92:             LinearLayout inner_layout = (LinearLayout) outer_layout
93:                 .findViewById(text_views[i]);
94:
95:             if (inner_layout.getChildCount() == 3) {
96:                 TextView left_view = (TextView) inner_layout.getChildAt
(0);
97:                 TextView right_view = (TextView) inner_layout.getChildA
t(2);
98:
99:                 if (info.get(tags[count]) == null) {
100:                     left_view.setText(Html.fromHtml(etiquetas[count
]));
101:                 } else {
102:                     left_view.setText(Html.fromHtml(etiquetas[count
]
103:                                     + info.get(tags[count])));
104:                 }
105:                 if (info.get(tags[count + 1]) == null) {
106:                     right_view.setText(Html.fromHtml(etiquetas[coun
t + 1]));
107:                 } else {
108:                     right_view.setText(Html.fromHtml(etiquetas[coun
t + 1]

```

```

109:                                     + info.get(tags[count + 1]) + E
URO));
110:     }
111:     right_view.setGravity(Gravity.RIGHT);
112:     count += 2;
113: } else if (inner_layout.getChildCount() == 5) {
114:     TextView left_view = (TextView) inner_layout.getChildAt
(0);
115:     TextView center = (TextView) inner_layout.getChildAt(2)
;
116:     TextView right_view = (TextView) inner_layout.getChildA
t(4);
117:     if (info.get(tags[count]) == null) {
118:         left_view.setText(Html.fromHtml(etiquetas[count
]));
119:     } else {
120:         left_view.setText(Html.fromHtml(etiquetas[count
]
121:                                     + info.get(tags[count])));
122:     }
123:     if (info.get(tags[count + 1]) == null) {
124:         center.setText(Html.fromHtml(etiquetas[count +
1]));
125:     } else {
126:         center.setText(Html.fromHtml(etiquetas[count +
1]
127:                                     + info.get(tags[count + 1]) + E
URO));
128:     }
129:     if (info.get(tags[count + 2]) == null) {
130:         right_view.setText(Html.fromHtml(etiquetas[coun
t + 2]));
131:     } else {
132:         right_view.setText(Html.fromHtml(etiquetas[coun
t + 2]
133:                                     + info.get(tags[count + 2]) + E
URO));
134:     }
135:     left_view.setGravity(Gravity.CENTER);
136:     center.setGravity(Gravity.CENTER);
137:     right_view.setGravity(Gravity.RIGHT);
138:     count += 3;
139: }
140: }
141: LinearLayout aneca_cost = (LinearLayout) outer_layout
142:     .findViewById(text_views[52]);
143: TextView aneca_view = (TextView) aneca_cost.getChildAt(2);
144: aneca_view.setText(Html.fromHtml(etiquetas[25] + ANECA_COST + EURO));
145: aneca_view.setGravity(Gravity.RIGHT);
146:
147: LinearLayout insurance_cost = (LinearLayout) outer_layout
148:     .findViewById(text_views[53]);
149: TextView insurance_view = (TextView) insurance_cost.getChildAt(2);
150:
151: float insurance_amount = (Float.parseFloat(info
152:     .get("granted-students-qty")) + Float.parseFloat(info
153:     .get("regular-students-qty")))
154:     * INSURANCE_PER_STUDENT;
155:
156: insurance_view.setText(Html.fromHtml(etiquetas[25]
157:     + String.valueOf(insurance_amount) + EURO));
158: insurance_view.setGravity(Gravity.RIGHT);
159:

```

```
160:
161:     }
162: }
```

return outer_layout;

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import org.apache.http.client.CookieStore;
4: import org.apache.http.impl.client.DefaultHttpClient;
5: import org.apache.http.protocol.HttpContext;
6:
7: import android.app.Application;
8:
9: /**
10:  *
11:  * @author ubay
12:  *
13:  *     Clase singleton para mantener los datos de usuario y sesion a traves
14:  *     de la ejecucion de la aplicacion, asi como mantener la URL destino
15:  *     para las peticiones
16:  */
17: public class GlobalState extends Application {
18:
19:     private CookieStore cookieStore;
20:     private HttpContext localContext;
21:     private DefaultHttpClient client;
22:     private String base_url;
23:     private boolean role;
24:
25:     public String getTargetAddress() {
26:         return base_url;
27:     }
28:
29:     public CookieStore getCookieStore() {
30:         return cookieStore;
31:     }
32:
33:     public HttpContext getLocalContext() {
34:         return localContext;
35:     }
36:
37:     public DefaultHttpClient getHttpClient() {
38:         return client;
39:     }
40:
41:     public boolean getRole() {
42:         return this.role;
43:     }
44:
45:     public void setCookieStore(CookieStore cs) {
46:         this.cookieStore = cs;
47:     }
48:
49:     public void setLocalContext(HttpContext lc) {
50:         this.localContext = lc;
51:     }
52:
53:     public void setHttpClient(DefaultHttpClient hc) {
54:         this.client = hc;
55:     }
56:
57:     public void setRole(boolean role) {
58:         this.role = role;
59:     }
60:
61:     public void setBaseAddress(String base_url) {
62:         this.base_url = base_url;
63:     }
}
```

64: }

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.app.Activity;
5: import android.content.Intent;
6: import android.os.Build;
7: import android.os.Bundle;
8: import android.support.v4.app.NavUtils;
9: import android.view.Menu;
10: import android.view.MenuItem;
11: import android.view.View;
12:
13: /**
14:  *
15:  * @author ubay
16:  *
17:  * Simple clase que muestra un layout que permite elegir entre las dos
18:  * opciones, crear propuestas o buscar propuestas. En funcion de la
19:  * eleccion, lanzara la clase CreateDegree o SearchDegree
20:  */
21: public class SelectTask extends Activity {
22:
23:     @Override
24:     protected void onCreate(Bundle savedInstanceState) {
25:         super.onCreate(savedInstanceState);
26:         setContentView(R.layout.select_task_activity);
27:
28:     }
29:
30:     public void newProposal(View v) {
31:         startActivity(new Intent(this, CreateDegree.class));
32:     }
33:
34:     public void exploreProposals(View v) {
35:         startActivity(new Intent(this, SearchDegrees.class));
36:     }
37:
38:     /**
39:      * Set up the {@link android.app.ActionBar}, if the API is available.
40:      */
41:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
42:     private void setupActionBar() {
43:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
44:             getActionBar().setDisplayHomeAsUpEnabled(true);
45:         }
46:     }
47:
48:     @Override
49:     public boolean onCreateOptionsMenu(Menu menu) {
50:         // Inflate the menu; this adds items to the action bar if it is present
51:
52:         getMenuInflater().inflate(R.menu.select_task, menu);
53:         return true;
54:     }
55:
56:     @Override
57:     public boolean onOptionsItemSelected(MenuItem item) {
58:         switch (item.getItemId()) {
59:             // This ID represents the Home or Up button. In the case of thi
60:             // activity, the Up button is shown. Use NavUtils to allow user
```

./SelectTask.java

Wed Nov 06 12:18:22 2013

2

```
61:          // to navigate up one level in the application structure. For
62:          // more details, see the Navigation pattern on Android Design:
63:          //
64:          // http://developer.android.com/design/patterns/navigation.html
#up-vs-back
65:          //
66:          NavUtils.navigateUpFromSameTask(this);
67:          return true;
68:      }
69:      return super.onOptionsItemSelected(item);
70:  }
71:
72: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import es.ulpgc.titulospropios.gestion.ubay.R;
4: import java.io.IOException;
5: import java.util.HashMap;
6:
7: import android.content.Context;
8: import android.os.AsyncTask;
9: import android.os.Bundle;
10: import android.support.v4.app.Fragment;
11: import android.support.v4.app.FragmentTransaction;
12: import android.view.View;
13: import android.widget.TabHost;
14: import android.widget.TabHost.TabContentFactory;
15:
16: public class EconomicInfo extends MyFragmentActivity implements
17:     TabHost.OnTabChangeListener {
18:     @SuppressWarnings("rawtypes")
19:     private HashMap mapTabInfo = new HashMap();
20:     private TabInfo mLastTab = null;
21:     public static HashMap<String, String> hashMap;
22:     private TabHost mTabHost;
23:     private Bundle bundle;
24:     private Context baseContext = this;
25:
26:     private class TabInfo {
27:     private String tag;
28:     @SuppressWarnings("rawtypes")
29:     private Class clazz;
30:     private Bundle args;
31:     private Fragment fragment;
32:
33:     TabInfo(String tag, @SuppressWarnings("rawtypes") Class clazz,
34:         Bundle args) {
35:         this.tag = tag;
36:         this.clazz = clazz;
37:         this.args = args;
38:     }
39:
40: }
41:
42: class TabFactory implements TabContentFactory {
43:
44:     private final Context mContext;
45:
46:     public TabFactory(Context context) {
47:         mContext = context;
48:     }
49:
50:     public View createTabContent(String tag) {
51:         View v = new View(mContext);
52:         v.setMinimumWidth(0);
53:         v.setMinimumHeight(0);
54:         return v;
55:     }
56: }
57:
58: @Override
59: public void onCreate(Bundle savedInstanceState) {
60:     // Inflate the layout for this fragment
61:     super.onCreate(savedInstanceState);
62:     setContentView(R.layout.economic_info_activity);
63:     bundle = savedInstanceState;
```

```

64:         new retrieveInfo().execute();
65:
66:     }
67:
68:     private class retrieveInfo extends AsyncTask<String, Void, Void> {
69:         IOException error;
70:
71:         @Override
72:         protected void onPreExecute() {
73:             super.onPreExecute();
74:             setUpProgressDialog(R.string.loadingMessage);
75:         }
76:
77:         @Override
78:         protected Void doInBackground(String... params) {
79:             String target_url = setTargetURL();
80:             String economic_data = getXmlPage(target_url);
81:             String[] tags = getResources()
82:                 .getStringArray(R.array.economic_info);
83:
84:             XmlExtractor xml_extractor = new XmlExtractor(economic_data);
85:             hashMap = xml_extractor.extractTagsFromDoc(tags);
86:             return null;
87:
88:         }
89:
90:         private String setTargetURL() {
91:             final String UNIV_SPEC_DEG = "university_specific_degrees";
92:             final String ECONOMIC_REP = "economic_report";
93:             final String XML_EXTENSION = ".xml";
94:             final String SLASH = "/";
95:
96:             GlobalState gs = (GlobalState) getApplication();
97:             String base_url = gs.getTargetAddress();
98:
99:             String degree_id = getIntent().getStringExtra("id");
100:
101:             String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_i
d
102:                 + SLASH + ECONOMIC_REP + XML_EXTENSION;
103:
104:             return target_url;
105:         }
106:
107:         private String getXmlPage(String target_url) {
108:             XmlPage xml_page = new XmlPage(baseContext);
109:             String xml_response = null;
110:             try {
111:                 xml_response = xml_page.getPage(target_url);
112:             } catch (IOException e) {
113:                 error = e;
114:                 return null;
115:             }
116:             return xml_response;
117:         }
118:
119:         @Override
120:         protected void onPostExecute(Void param) {
121:
122:             dismissWindow();
123:             if (error != null) {
124:                 launchErrorWindow(0);
125:             } else {

```

```

126:             initialiseTabHost(bundle);
127:             if (bundle != null) {
128:                 mTabHost.setCurrentTabByTag(bundle.getString("tab"));
129:             }
130:         }
131:     }
132: }
133:
134: }
135:
136: @SuppressWarnings("unchecked")
137: private void initialiseTabHost(Bundle args) {
138:     mTabHost = (TabHost) findViewById(android.R.id.tabhost);
139:     mTabHost.setup();
140:     TabInfo tabInfo = null;
141:     EconomicInfo.addTab(
142:         this,
143:         this.mTabHost,
144:         this.mTabHost.newTabSpec("Tab1").setIndicator(
145:             getResources().getString(R.string.overviewTab)),
146:         (tabInfo = new TabInfo("Tab1", TabOverview.class, args)));
147:     this.mapTabInfo.put(tabInfo.tag, tabInfo);
148:     EconomicInfo.addTab(
149:         this,
150:         this.mTabHost,
151:         this.mTabHost.newTabSpec("Tab2").setIndicator(
152:             getResources().getString(R.string.incomeTab)),
153:         (tabInfo = new TabInfo("Tab2", TabIncome.class, args)));
154:     this.mapTabInfo.put(tabInfo.tag, tabInfo);
155:     EconomicInfo.addTab(
156:         this,
157:         this.mTabHost,
158:         this.mTabHost.newTabSpec("Tab3").setIndicator(
159:             getResources().getString(R.string.expensesTab)),
160:         (tabInfo = new TabInfo("Tab3", TabExpenditure.class, args)));
161:     this.mapTabInfo.put(tabInfo.tag, tabInfo);
162:     // Default to first tab
163:     this.onTabChanged("Tab1");
164:     //
165:     mTabHost.setOnTabChangeListener(this);
166: }
167:
168: private static void addTab(EconomicInfo activity, TabHost tabHost,
169:     TabHost.TabSpec tabSpec, TabInfo tabInfo) {
170:     // Attach a Tab view factory to the spec
171:     tabSpec.setContent(activity.new TabFactory(activity));
172:     String tag = tabSpec.getTag();
173:
174:     // Check to see if we already have a fragment for this tab, probably
175:     // from a previously saved state. If so, deactivate it, because our
176:     // initial state is that a tab isn't shown.
177:     tabInfo.fragment = activity.getSupportFragmentManager()
178:         .findFragmentByTag(tag);
179:     if (tabInfo.fragment != null && !tabInfo.fragment.isDetached()) {
180:         FragmentTransaction ft = activity.getSupportFragmentManager()
181:             .beginTransaction();

```

```
182:         ft.detach(tabInfo.fragment);
183:         ft.commit();
184:         activity.getSupportFragmentManager().executePendingTransactions
();
185:     }
186:     tabHost.addTab(tabSpec);
187: }
188:
189: public void onTabChanged(String tag) {
190:     TabInfo newTab = (TabInfo) this.mapTabInfo.get(tag);
191:     if (mLastTab != newTab) {
192:         FragmentTransaction ft = this.getSupportFragmentManager()
193:             .beginTransaction();
194:         if (mLastTab != null) {
195:             if (mLastTab.fragment != null) {
196:                 ft.detach(mLastTab.fragment);
197:             }
198:         }
199:         if (newTab != null) {
200:             if (newTab.fragment == null) {
201:                 newTab.fragment = Fragment.instantiate(this,
202:                     newTab.clss.getName(), newTab.a
rgs);
203:                 ft.add(R.id.realtabcontent, newTab.fragment, ne
wTab.tag);
204:             } else {
205:                 ft.attach(newTab.fragment);
206:             }
207:         }
208:         mLastTab = newTab;
209:         ft.commit();
210:         this.getSupportFragmentManager().executePendingTransactions();
211:     }
212: }
213: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.io.UnsupportedEncodingException;
5: import java.text.SimpleDateFormat;
6: import java.util.ArrayList;
7: import java.util.Date;
8: import java.util.HashMap;
9: import java.util.List;
10: import java.util.regex.Matcher;
11: import java.util.regex.Pattern;
12:
13: import org.apache.http.HttpResponse;
14: import org.apache.http.NameValuePair;
15: import org.apache.http.client.ClientProtocolException;
16: import org.apache.http.client.HttpClient;
17: import org.apache.http.client.entity.UrlEncodedFormEntity;
18: import org.apache.http.client.methods.HttpPost;
19: import org.apache.http.message.BasicNameValuePair;
20: import org.apache.http.protocol.HTTP;
21: import org.apache.http.protocol.HttpContext;
22:
23: import android.annotation.SuppressLint;
24: import android.annotation.TargetApi;
25: import android.app.Activity;
26: import android.content.Context;
27: import android.os.AsyncTask;
28: import android.os.Build;
29: import android.os.Bundle;
30: import android.os.Handler;
31: import android.os.Message;
32: import android.support.v4.app.NavUtils;
33: import android.text.TextUtils;
34: import android.view.Menu;
35: import android.view.MenuItem;
36: import android.view.View;
37: import android.widget.AdapterView;
38: import android.widget.AdapterView.OnItemClickListener;
39: import android.widget.ArrayAdapter;
40: import android.widget.CheckBox;
41: import android.widget.EditText;
42: import android.widget.Spinner;
43: import android.widget.TextView;
44: import android.widget.Toast;
45:
46: /**
47:  *
48:  * @author ubay
49:  *
50:  * Esta clase, al ser instanciada, controla las fechas de creacion de
51:  * nuevos titulos mediante la descarga de periods.xml. Si esta dentro de
52:  * los plazos adecuados muestra el layout con los campos necesarios para
53:  * la creacion de una nueva propuesta.
54:  *
55:  * El campo del centro asociado se carga automaticamente en funcion del
56:  * valor del Spinner que se situa inmediatamente por encima.
57:  *
58:  * Si todos los campos estan rellenos, la clase comprobara el dni del
59:  * proponente que se haya introducido para comprobar que esta
60:  * autorizado. Si lo esta, se remitira la propuesta a la plataforma.
61:  *
62:  * Si en algun momento se produce un error con los datos introducidos,
63:  * la aplicacion mostrara un mensaje de error
```

```

64:  *
65:  *      Muchos de los posibles errores generados por la aplicacion web
66:  *      despues de remitir la propuesta no se capturan en esta clase, al ser
67:  *      devueltos dentro del documento HTML
68:  *
69:  *
70:  */
71: public class CreateDegree extends MyFragmentActivity {
72:
73:     private Context baseContext = this;
74:     private Handler mHandler, senderHandler;
75:     private ArrayAdapter<CharSequence> types_adapter, modes_adapter,
76:         oriented_adapter, knowledge_adapter, institutions_adapter;
77:     private String type, mode, oriented, knowledge, institutions;
78:     private Spinner option1, option2, option3, option4, option5;
79:     private static final String NONE_SELECTED = "nothing";
80:     private String period_id;
81:     private String institution_id;
82:     private int institution_index;
83:     private boolean outsideCanaryIslands = false;
84:     private int type_index, mode_index, oriented_index, knowledge_index;
85:
86:     @Override
87:     protected void onCreate(Bundle savedInstanceState) {
88:         super.onCreate(savedInstanceState);
89:
90:         new downloadPeriods().execute();
91:
92:         /*
93:          * mHandler: recibe el periodo actual antes de generar el layout para
94:          * comprobar si estÃ¡; dentro del plazo permitido para que la propuesta
95:          * sea remitida. Si lo estÃ¡; establece el layout a aplicar. Si no,
96:          * muestra mensaje y manda al usuario atrÃ¡s
97:          */
98:         mHandler = new Handler() {
99:             @SuppressWarnings("SimpleDateFormat")
100:             @Override
101:             @SuppressWarnings("unchecked")
102:             public void handleMessage(Message msg) {
103:                 String[] period_tags = { "request-from-date", "request-
to-date" };
104:                 Date cDate = new Date();
105:                 String fDate = new SimpleDateFormat("yyyy-MM-dd").forma
t(cDate);
106:                 HashMap<String, String> current_period = (HashMap<Strin
g, String>) msg.obj;
107:
108:                 if (!(fDate.compareTo(current_period.get(period_tags[0]
)) >= 0 && fDate
109:                     .compareTo(current_period.get(period_ta
gs[1])) <= 0)) {
110:
111:                     AlertDialogGoingBack(R.string.outOfPeriodForPro
posal);
112:
113:                 } else {
114:                     setContentView(R.layout.create_degree_activity)
;
115:                     setSpinnerValues();
116:                     setUpCheckBox();
117:
118:                     TextView period = (TextView) findViewById(R.id.
period);

```

```
119:         period.setText(current_period.get("year"));
120:         period_id = current_period.get("id");
121:     }
122: }
123: }
124: };
125:
126: senderHandler = new Handler() {
127:     public void handleMessage(Message msg) {
128:         new sendVerifiedProposal().execute();
129:     }
130: };
131:
132: }
133:
134: private void setUpCheckBox() {
135:     CheckBox check_box = (CheckBox) findViewById(R.id.teaching_outside);
136:     check_box.setOnClickListener(new View.OnClickListener() {
137:         public void onClick(View v) {
138:             outsideCanaryIslands = ((CheckBox) v).isChecked();
139:         }
140:     });
141: }
142:
143: private void setSpinnerValues() {
144:
145:     createSpinnersFromResources();
146:     connectSpinnersWithAdapters();
147:
148: }
149:
150: private void connectSpinnersWithAdapters() {
151:
152:     option1 = (Spinner) findViewById(R.id.type_spinner);
153:     option2 = (Spinner) findViewById(R.id.mode_spinner);
154:     option3 = (Spinner) findViewById(R.id.oriented_spinner);
155:     option4 = (Spinner) findViewById(R.id.knowledge_spinner);
156:     option5 = (Spinner) findViewById(R.id.institutions_spinner);
157:
158:     option1.setAdapter(types_adapter);
159:     option2.setAdapter(modes_adapter);
160:     option3.setAdapter(oriented_adapter);
161:     option4.setAdapter(knowledge_adapter);
162:     option5.setAdapter(institutions_adapter);
163:
164:     option1.setOnItemClickListener(new OnItemSelectedListener() {
165:         public void onItemSelected(AdapterView<?> parentView,
166:             View selectedItemView, int position, long id) {
167:             type = (String) types_adapter.getItem(position);
168:             type_index = position;
169:         }
170:
171:         public void onNothingSelected(AdapterView<?> parentView) {
172:             type = NONE_SELECTED;
173:         }
174:     });
175:     option2.setOnItemClickListener(new OnItemSelectedListener() {
176:         public void onItemSelected(AdapterView<?> parentView,
177:             View selectedItemView, int position, long id) {
178:             mode = (String) modes_adapter.getItem(position);
179:             mode_index = position;
180:         }
181:     }
```

```

182:         public void onNothingSelected(AdapterView<?> parentView) {
183:             mode = NONE_SELECTED;
184:         }
185:     });
186:     option3.setOnItemClickListener(new OnItemSelectedListener() {
187:         public void onItemSelected(AdapterView<?> parentView,
188:             View selectedItemView, int position, long id) {
189:             oriented = (String) oriented_adapter.getItem(position);
190:             oriented_index = position;
191:         }
192:
193:         public void onNothingSelected(AdapterView<?> parentView) {
194:             oriented = NONE_SELECTED;
195:         }
196:     });
197:     option4.setOnItemClickListener(new OnItemSelectedListener() {
198:         public void onItemSelected(AdapterView<?> parentView,
199:             View selectedItemView, int position, long id) {
200:             knowledge = (String) knowledge_adapter.getItem(position
201:
202:             knowledge_index = position;
203:         }
204:
205:         public void onNothingSelected(AdapterView<?> parentView) {
206:             knowledge = NONE_SELECTED;
207:         }
208:     });
209:     option5.setOnItemClickListener(new OnItemSelectedListener() {
210:         public void onItemSelected(AdapterView<?> parentView,
211:             View selectedItemView, int position, long id) {
212:             institutions = (String) institutions_adapter.getItem(po
213:             institutions_address = (EditText) findViewById
214:             institutions_address.setText(getResources().getStringAr
215:             R.array.institutions_addresses)[position
216:             institution_id = getResources().getStringArray(
217:             R.array.institutions_id)[position];
218:             institution_index = position;
219:         }
220:
221:         public void onNothingSelected(AdapterView<?> parentView) {
222:             institutions = NONE_SELECTED;
223:             institution_id = "24";
224:             institution_index = 0;
225:         }
226:     });
227: }
228: private void createSpinnersFromResources() {
229:     types_adapter = ArrayAdapter.createFromResource(this,
230:         R.array.degree_types, android.R.layout.simple_spinner_i
231:     modes_adapter = ArrayAdapter.createFromResource(this,
232:         R.array.teachingModeComplete,
233:         android.R.layout.simple_spinner_item);
234:     oriented_adapter = ArrayAdapter.createFromResource(this,
235:         R.array.orientedTo, android.R.layout.simple_spinner_ite
236:     knowledge_adapter = ArrayAdapter.createFromResource(this,
237:         R.array.branchKnowledgeComplete,

```

```

238:         android.R.layout.simple_spinner_item);
239:     institutions_adapter = ArrayAdapter.createFromResource(this,
240:         R.array.institutions, android.R.layout.simple_spinner_i
tem);
241:
242:     types_adapter
243:         .setDropDownViewResource(R.layout.multiline_spinner_dro
pdown_item);
244:     modes_adapter
245:         .setDropDownViewResource(R.layout.multiline_spinner_dro
pdown_item);
246:     oriented_adapter
247:         .setDropDownViewResource(R.layout.multiline_spinner_dro
pdown_item);
248:     knowledge_adapter
249:         .setDropDownViewResource(R.layout.multiline_spinner_dro
pdown_item);
250:     institutions_adapter
251:         .setDropDownViewResource(R.layout.multiline_spinner_dro
pdown_item);
252:     }
253:
254:     private class downloadPeriods extends AsyncTask<Void, Void, Void> {
255:
256:         IOException error;
257:         HashMap<String, String> current_period;
258:
259:         @Override
260:         protected void onPreExecute() {
261:             setUpProgressDialog(R.string.loadingMessage);
262:         }
263:
264:         @Override
265:         protected void doInBackground(Void... params) {
266:             final String PERIOD_ENDTAG = "</period>";
267:             String[] tags = getResources().getStringArray(R.array.periods);
268:             XmlPage periods = new XmlPage(baseContext);
269:             String target_url = setPeriodsTargetUrl();
270:             String xml_periods;
271:             try {
272:                 xml_periods = periods.getPage(target_url);
273:             } catch (IOException e) {
274:                 error = e;
275:                 return null;
276:             }
277:             XmlExtractor xml_extractor = new XmlExtractor(xml_periods);
278:             ArrayList<String> periods_list = xml_extractor
279:                 .unitSeparator(PERIOD_ENDTAG);
280:
281:             int[] year = { 0, 0 };
282:             ArrayList<HashMap<String, String>> single_period_list = new Arr
ayList<HashMap<String, String>>();
283:
284:             for (int i = 0; i < periods_list.size(); i++) {
285:                 XmlExtractor single_period = new XmlExtractor(
286:                     periods_list.get(i));
287:
288:                 current_period = single_period.extractTagsFromDoc(tags)
;
289:                 if (year[0] < Integer.parseInt(current_period.get("year
")
290:                     .substring(0, 4))) {
291:                     year[0] = Integer.parseInt(current_period.get("

```

```

year")
292:                                     .substring(0, 4));
293:                                     year[1] = i;
294:                                     }
295:                                     single_period_list.add(current_period);
296:                                     }
297:                                     current_period = single_period_list.get(year[1]);
298:                                     return null;
299:
300:                                     }
301:
302:                                     protected void onPostExecute(Void result) {
303:                                         dismissWindow();
304:
305:                                         if (error != null) {
306:                                             launchErrorWindow(1);
307:                                             return;
308:                                         }
309:
310:                                         Message msg = new Message();
311:                                         msg.obj = current_period;
312:                                         mHandler.sendMessage(msg);
313:
314:                                     }
315:
316:                                     private String setPeriodsTargetUrl() {
317:                                         final String XML_EXTENSION = ".xml";
318:                                         final String PERIODS = "periods";
319:
320:                                         GlobalState gs = (GlobalState) getApplication();
321:                                         String base_url = gs.getTargetAddress();
322:
323:                                         String target_url = base_url + PERIODS + XML_EXTENSION;
324:
325:                                         return target_url;
326:                                     }
327:                                     }
328:
329:                                     /**
330:                                     * Set up the {@link android.app.ActionBar}, if the API is available.
331:                                     */
332:                                     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
333:                                     private void setupActionBar() {
334:                                         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
335:                                             getActionBar().setDisplayHomeAsUpEnabled(true);
336:                                         }
337:                                     }
338:
339:                                     @Override
340:                                     public boolean onCreateOptionsMenu(Menu menu) {
341:                                         // Inflate the menu; this adds items to the action bar if it is present
342:
343:                                         getMenuInflater().inflate(R.menu.create_degree, menu);
344:                                         return true;
345:                                     }
346:                                     @Override
347:                                     public boolean onOptionsItemSelected(MenuItem item) {
348:                                         switch (item.getItemId()) {
349:                                             case android.R.id.home:
350:                                                 // This ID represents the Home or Up button. In the case of thi
351:                                                 // activity, the Up button is shown. Use NavUtils to allow user

```

```

S
352:         // to navigate up one level in the application structure. For
353:         // more details, see the Navigation pattern on Android Design:
354:         //
355:         // http://developer.android.com/design/patterns/navigation.html
#up-vs-back
356:         //
357:         NavUtils.navigateUpFromSameTask(this);
358:         return true;
359:     }
360:     return super.onOptionsItemSelected(item);
361: }
362:
363: public void sendNewProposal(View v) {
364:     ArrayList<Integer> fields_with_no_content = new ArrayList<Integer>();
365:     fields_with_no_content = checkFieldsForBlanksAndNonNumbers();
366:     fields_with_no_content.addAll(checkSpinnersForNoneSelected());
367:     ArrayList<Integer> fields_with_no_number = new ArrayList<Integer>();
368:     fields_with_no_number = checkFieldsForNoNumbers();
369:
370:     if (fields_with_no_content.size() != 0) {
371:         Toast.makeText(baseContext,
372:             getResources().getString(R.string.emptyFields),
373:             Toast.LENGTH_LONG).show();
374:         highlightFieldsWithErrors(fields_with_no_content);
375:     } else {
376:         if (fields_with_no_number.size() != 0) {
377:             Toast.makeText(baseContext,
378:                 getResources().getString(R.string.noNum
berFields),
379:                 Toast.LENGTH_LONG).show();
380:             highlightFieldsWithErrors(fields_with_no_number);
381:         } else {
382:             checkId();
383:
384:         }
385:     }
386: }
387:
388:
389: private ArrayList<Integer> checkFieldsForNoNumbers() {
390:     int[] numberFields = { R.id.degree_min_alumns, R.id.degree_max_alumns,
391:         R.id.degree_min_credits };
392:
393:     ArrayList<Integer> fields_with_no_number = new ArrayList<Integer>();
394:
395:     for (int i = 0; i < numberFields.length; i++) {
396:         EditText textBox = (EditText) findViewById(numberFields[i]);
397:         String content = textBox.getText().toString();
398:         if (!(TextUtils.isDigitsOnly(content))) {
399:             fields_with_no_number.add(numberFields[i]);
400:         }
401:     }
402:
403:     return fields_with_no_number;
404: }
405:
406: private void highlightFieldsWithErrors(ArrayList<Integer> fields_with_errors) {
407:     int[] spinners = { R.id.type_spinner, R.id.mode_spinner,
408:         R.id.oriented_spinner, R.id.knowledge_spinner };
409:     for (int i = 0; i < fields_with_errors.size(); i++) {
410:         if (fields_with_errors.get(i).equals(spinners[0])
411:             | fields_with_errors.get(i).equals(spinners[1]

```

```

)
412:                || fields_with_errors.get(i).equals(spinner[2])
)
413:                || fields_with_errors.get(i).equals(spinner[3])
)) {
414:        Spinner field = (Spinner) findViewById(fields_with_errors
rs
415:                .get(i));
416:        field.setBackgroundResource(R.drawable.grey_buttons_wit
h_red_border);
417:
418:        } else {
419:
420:        EditText field = (EditText) findViewById(fields_with_er
rors
421:                .get(i));
422:        field.setBackgroundResource(R.drawable.red_border);
423:        }
424:    }
425: }
426:
427: private ArrayList<Integer> checkSpinnersForNoneSelected() {
428:     ArrayList<Integer> spinners_with_errors = new ArrayList<Integer>();
429:
430:     if (type.equals(getResources().getStringArray(R.array.degree_types)[0])
) {
431:         spinners_with_errors.add(R.id.type_spinner);
432:     } else {
433:         Spinner spinner = (Spinner) findViewById(R.id.type_spinner);
434:         spinner.setBackgroundResource(R.drawable.grey_buttons);
435:     }
436:     if (mode.equals(getResources().getStringArray(
437:         R.array.teachingModeComplete)[0])) {
438:         spinners_with_errors.add(R.id.mode_spinner);
439:     } else {
440:         Spinner spinner = (Spinner) findViewById(R.id.mode_spinner);
441:         spinner.setBackgroundResource(R.drawable.grey_buttons);
442:     }
443:     if (oriented
444:         .equals(getResources().getStringArray(R.array.orientedT
o)[0])) {
445:         spinners_with_errors.add(R.id.oriented_spinner);
446:     } else {
447:         Spinner spinner = (Spinner) findViewById(R.id.oriented_spinner)
;
448:         spinner.setBackgroundResource(R.drawable.grey_buttons);
449:     }
450:     if (knowledge.equals(getResources().getStringArray(
451:         R.array.branchKnowledgeComplete)[0])) {
452:         spinners_with_errors.add(R.id.knowledge_spinner);
453:     } else {
454:         Spinner spinner = (Spinner) findViewById(R.id.knowledge_spinner
);
455:         spinner.setBackgroundResource(R.drawable.grey_buttons);
456:     }
457:
458:     return spinners_with_errors;
459: }
460:
461: private ArrayList<Integer> checkFieldsForBlanksAndNonNumbers() {
462:
463:     int[] textFields = { R.id.degree_denomination, R.id.degree_competences,
464:

```

```

465:         R.id.degree_proposed_by, R.id.degree_units,
466:         R.id.degree_min_alumns, R.id.degree_max_alumns,
467:         R.id.degree_min_credits, R.id.degree_economic_manager,
468:         R.id.teaching_address };
469:
470:     ArrayList<Integer> textFields_with_errors = new ArrayList<Integer>();
471:
472:     for (int i = 0; i < textFields.length; i++) {
473:         EditText textBox = (EditText) findViewById(textFields[i]);
474:         String content = textBox.getText().toString();
475:         if (content.equals("")) {
476:             textFields_with_errors.add(textFields[i]);
477:         } else
478:             textBox.setBackgroundResource(R.drawable.blue_border);
479:     }
480:     return textFields_with_errors;
481: }
482:
483: private void checkId() {
484:     EditText box_teacher_dni = (EditText) findViewById(R.id.degree_academic
_director);
485:     String teacher_dni = box_teacher_dni.getText().toString();
486:
487:     Pattern p2 = Pattern.compile("^\\d{8}[A-Za-z]$");
488:     Matcher m2 = p2.matcher(teacher_dni);
489:
490:     if (m2.matches()) {
491:         String teacher_dni_verified = teacher_dni.substring(0,
492:             teacher_dni.length() - 1)
493:             + teacher_dni.substring(teacher_dni.length() -
1,
494:                 teacher_dni.length()).toUpperCase();
495:
496:         EditText dni_box = (EditText) findViewById(R.id.degree_academic
_director);
497:         dni_box.setText(teacher_dni_verified);
498:         new checkTeacherExists().execute(teacher_dni_verified);
499:     } else {
500:         box_teacher_dni.setBackgroundResource(R.drawable.red_border);
501:         Toast.makeText(baseContext,
502:             getResources().getString(R.string.teacherNotPre
sent),
503:                 Toast.LENGTH_LONG).show();
504:     }
505: }
506:
507: private class checkTeacherExists extends AsyncTask<String, Void, Void> {
508:
509:     IOException error;
510:     String xml_teachers;
511:
512:     @Override
513:     protected void onPreExecute() {
514:         setUpProgressDialog(R.string.checkingMessage);
515:     }
516:
517:     @Override
518:     protected Void doInBackground(String... params) {
519:         XmlPage teacher = new XmlPage(baseContext);
520:         String target_url = setTeacherTargetUrl(params[0]);
521:         try {
522:             xml_teachers = teacher.getPage(target_url);

```

```

523:         } catch (IOException e) {
524:             error = e;
525:             return null;
526:         }
527:         return null;
528:     }
529:
530:     protected void onPostExecute(Void result) {
531:         dismissWindow();
532:
533:         if (error != null) {
534:             launchErrorWindow(1);
535:             return;
536:         }
537:         if (xml_teachers.length() == 67) {
538:             Toast.makeText(baseContext,
539:                 getResources().getString(R.string.teach
erNotPresent),
540:                     Toast.LENGTH_LONG).show();
541:         } else {
542:             if (xml_teachers.subSequence(40, 45).equals("ulpgc")) {
543:                 Message msg = new Message();
544:                 senderHandler.sendMessage(msg);
545:             } else
546:                 Toast.makeText(
547:                     baseContext,
548:                     getResources()
549:                         .getString(R.st
ring.teacherNotPresent),
550:                         Toast.LENGTH_LONG).show();
551:         }
552:     }
553:
554: }
555:
556:
557:     private String setTeacherTargetUrl(String dni) {
558:         final String PARAMS_PRE_DNI = "teachers.xml?utf8=â\234\223&search[dni_1
ike]=";
559:         final String PARAMS_POST_DNI = "&search[full_name_like]=&search[type_eq
]=&search[aasm_state_eq]=&search[ulpgc_department_id_eq]=&commit=Buscar";
560:         GlobalState gs = (GlobalState) getApplication();
561:         String base_url = gs.getTargetAddress();
562:
563:         return base_url + PARAMS_PRE_DNI + dni + PARAMS_POST_DNI;
564:     }
565:
566:     private class sendVerifiedProposal extends AsyncTask<Void, Void, Void> {
567:
568:         IOException error;
569:         int statusCode;
570:
571:         @Override
572:         protected void onPreExecute() {
573:             setUpProgressDialog(R.string.sendingDegree);
574:         }
575:
576:         @Override
577:         protected Void doInBackground(Void... params) {
578:
579:             final String UNIV_SP_DEG = "university_specific_degrees";
580:             GlobalState gs = (GlobalState) ((Activity) baseContext)
581:                 .getApplication();

```

```

582:         HttpClient client = gs.getHttpClient();
583:         HttpContext localContext = gs.getLocalContext();
584:         String target_address = gs.getTargetAddress() + UNIV_SP_DEG;
585:
586:         HttpPost post = new HttpPost(target_address);
587:         List<NameValuePair> post_params = new ArrayList<NameValuePair>(
);
588:
589:         post_params = setPostParams();
590:
591:         UrlEncodedFormEntity ent = null;
592:         try {
593:             ent = new UrlEncodedFormEntity(post_params, HTTP.UTF_8)
;
594:
595:             post.setEntity(ent);
596:             HttpResponse responsePOST = null;
597:             responsePOST = client.execute(post, localContext);
598:             responsePOST.getEntity().consumeContent();
599:             statusCode = responsePOST.getStatusLine().getStatusCode
();
600:
601:         } catch (UnsupportedEncodingException e) {
602:             e.printStackTrace();
603:         } catch (ClientProtocolException e) {
604:             e.printStackTrace();
605:         } catch (IOException e) {
606:             error = e;
607:             e.printStackTrace();
608:         }
609:         return null;
610:
611:     }
612:
613:     protected void onPostExecute(Void result) {
614:         dismissWindow();
615:         if (error != null) {
616:             launchErrorWindow(1);
617:             return;
618:         }
619:         if (statusCode == 200) {
620:             Toast.makeText(baseContext,
621:                 getResources().getString(R.string.propo
salSent),
622:                 Toast.LENGTH_LONG).show();
623:         } else {
624:             Toast.makeText(baseContext,
625:                 getResources().getString(R.string.unide
ntifiedError),
626:                 Toast.LENGTH_LONG).show();
627:         }
628:
629:     }
630:
631:     private List<NameValuePair> setPostParams() {
632:
633:         String[] string_params = { "name", "mode_id", "audience_id",
634:             "competences", "proposed_by", "manager_dni",
635:             "academic_management_units", "branch_of_knowled
ge_id",
636:             "min_attendees", "max_attendees", "period_id",
637:             "min_credits_per_academic_period",
638:             "outside_canary_islands", "institution_id",

```

```

639:         "teaching_place", "head_office", "administrator
" };
640:
641:         String[] PARAM_UTF = { "utf8", "â\234\223" };
642:         String[] COMMIT = { "commit", "Guardar" };
643:         String[] EVAL_ACECAU = { "evaluated_by_acecau_at", "" };
644:         String UNIV_PREFIX = "university_specific_degree[";
645:         String UNIV_POST = "]";
646:         String PERIOD_ID = "period_id";
647:         String HEAD_OFFICE = "head_office";
648:         String INSTITUTION_ID = "institution_id";
649:         String OUTSIDE = "outside_canary_islands";
650:         String DEGREE_TYPE = "degree_type_id";
651:         String DEGREE_MODE = "mode_id";
652:         String DEGREE_AUDIENCE = "audience_id";
653:         String DEGREE_BRANCH = "branch_of_knowledge_id";
654:
655:         int[] string_views = { R.id.degree_denomination,
656:             R.id.degree_competences, R.id.degree_proposed_b
y,
657:             R.id.degree_units, R.id.degree_min_alumns,
658:             R.id.degree_max_alumns, R.id.degree_min_credits
,
659:             R.id.degree_economic_manager, R.id.teaching_add
ress,
660:             R.id.degree_academic_director };
661:
662:         String[] param_string_views = { "name", "competences",
663:             "proposed_by", "academic_management_units",
664:             "min_attendees", "max_attendees",
665:             "min_credits_per_academic_period", "administrat
or",
666:             "teaching_place", "manager_dni" };
667:
668:         List<NameValuePair> post_params = new ArrayList<NameValuePair>(
);
669:         post_params.add(new BasicNameValuePair(PARAM_UTF[0], PARAM_UTF[
1]));
670:
671:         for (int i = 0; i < param_string_views.length; i++) {
672:             EditText content = (EditText) findViewById(string_views
[i]);
673:             String param_name = UNIV_PREFIX + param_string_views[i]
+ UNIV_POST;
674:             String string_content = content.getText().toString();
675:             post_params.add(new BasicNameValuePair(param_name,
string_content));
676:         }
677:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + EVAL_ACECA
U[0]
678:             + UNIV_POST, EVAL_ACECAU[1]));
679:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + PERIOD_ID
+ UNIV_POST, period_id));
680:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + INSTITUTIO
N_ID
681:             + UNIV_POST, getResources().getStringArray(
R.array.institutions_id)[institution_index]));
682:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + HEAD_OFFIC
E
683:             + UNIV_POST, getResources().getStringArray(
R.array.institutions_addresses)[institution_inde
x]));
684:
685:
686:
687:
688:
689:

```

```

690:         if (outsideCanaryIslands) {
691:             post_params.add(new BasicNameValuePair(UNIV_PREFIX + OU
TSIDE
692:                 + UNIV_POST, "1"));
693:         } else
694:             post_params.add(new BasicNameValuePair(UNIV_PREFIX + OU
TSIDE
695:                 + UNIV_POST, "0"));
697:         post_params.add(new BasicNameValuePair(COMMIT[0], COMMIT[1]));
698:
699:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + DEGREE_TYP
E
700:             + UNIV_POST, Integer.toString(type_index)));
701:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + DEGREE_MOD
E
702:             + UNIV_POST, Integer.toString(mode_index)));
703:         post_params.add(new BasicNameValuePair(UNIV_PREFIX
704:             + DEGREE_AUDIENCE + UNIV_POST, Integer
705:             .toString(oriented_index)));
706:         post_params.add(new BasicNameValuePair(UNIV_PREFIX + DEGREE_BRA
NCH
707:             + UNIV_POST, Integer.toString(knowledge_index)
);
708:
709:         return post_params;
710:     }
711: }
712: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.text.SimpleDateFormat;
5: import java.util.ArrayList;
6: import java.util.Date;
7: import java.util.HashMap;
8: import android.annotation.SuppressLint;
9: import android.graphics.Color;
10: import android.os.AsyncTask;
11: import android.os.Bundle;
12: import android.support.v4.app.Fragment;
13: import android.text.Html;
14: import android.view.Gravity;
15: import android.view.LayoutInflater;
16: import android.view.View;
17: import android.view.View.OnClickListener;
18: import android.view.ViewGroup;
19: import android.view.WindowManager;
20: import android.widget.Button;
21: import android.widget.LinearLayout;
22: import android.widget.TextView;
23: import android.widget.Toast;
24:
25: /**
26:  *
27:  * @author ubay
28:  *
29:  *     Clase que se ocupa de descargar el documento periods.xml de la
30:  *     aplicacion web y poblar su layout con dicha informacion
31:  */
32: public class PeriodsFragment extends Fragment implements OnClickListener {
33:     private HashMap<String, String> current_period;
34:     private String current_period_timeframe;
35:     private Boolean time_sendable = false;
36:
37:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
38:         Bundle savedInstanceState) {
39:
40:         View myFragmentView = inflater.inflate(R.layout.periods_fragment,
41:             container, false);
42:
43:         getActivity().getWindow().setSoftInputMode(
44:             WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDD
EN);
45:
46:         new downloadPeriods().execute();
47:
48:         Button send_button = (Button) myFragmentView
49:             .findViewById(R.id.send_button);
50:         send_button.setOnClickListener(this);
51:
52:         return myFragmentView;
53:     }
54:
55:     private class downloadPeriods extends AsyncTask<Void, Void, Void> {
56:
57:         IOException error;
58:
59:         @Override
60:         protected void onPreExecute() {
61:             ((MyFragmentActivity) getActivity())
62:                 .setUpProgressDialog(R.string.loadingMessage);
```

```

63:     }
64:
65:     @Override
66:     protected Void doInBackground(Void... params) {
67:         final String PERIOD_ENDTAG = "</period>";
68:         String[] tags = getResources().getStringArray(R.array.periods);
69:         XmlPage periods = new XmlPage(getActivity());
70:         String target_url = setTargetUrl();
71:         String xml_periods;
72:         try {
73:             xml_periods = periods.getPage(target_url);
74:         } catch (IOException e) {
75:             error = e;
76:             return null;
77:         }
78:         XmlExtractor xml_extractor = new XmlExtractor(xml_periods);
79:         ArrayList<String> periods_list = xml_extractor
80:             .unitSeparator(PERIOD_ENDTAG);
81:
82:         int[] year = { 0, 0 };
83:         ArrayList<HashMap<String, String>> single_period_list = new Arr
ayList<HashMap<String, String>>();
84:
85:         for (int i = 0; i < periods_list.size(); i++) {
86:             XmlExtractor single_period = new XmlExtractor(
87:                 periods_list.get(i));
88:
89:             current_period = single_period.extractTagsFromDoc(tags)
;
90:             if (year[0] < Integer.parseInt(current_period.get("year
")
91:                 .substring(0, 4))) {
92:                 year[0] = Integer.parseInt(current_period.get("
year")
93:                     .substring(0, 4));
94:                 year[1] = i;
95:             }
96:             single_period_list.add(current_period);
97:         }
98:         current_period = single_period_list.get(year[1]);
99:         return null;
100:     }
101:
102:
103:     protected void onPostExecute(Void result) {
104:         ((MyFragmentActivity) getActivity()).dismissWindow();
105:
106:         if (error != null) {
107:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
1);
108:             return;
109:         }
110:         populateViews();
111:     }
112:
113:     @SuppressWarnings("SimpleDateFormat")
114:     private void populateViews() {
115:
116:         int[] rows = { R.id.second_row, R.id.third_row, R.id.fourth_row
};
117:         String[] tags_order = { "request-from-date", "request-to-date",
118:             "first-rectification-from-date",
119:             "first-rectification-to-date",

```

```

120:         "second-rectification-from-date",
121:         "second-rectification-to-date" };
122:     String[] locale_tags = {
123:         getResources().getString(R.string.period_start)
124:         ,
125:         getResources().getString(R.string.period_end) }
126:     ;
127:     Date cDate = new Date();
128:     String fDate = new SimpleDateFormat("yyyy-MM-dd").format(cDate)
129:     ;
130:     LinearLayout lView = (LinearLayout) getActivity().findViewById(
131:         R.id.first_row);
132:     TextView year = (TextView) lView.findViewById(R.id.item);
133:     current_period_timeframe = current_period.get("year");
134:     year.setText(current_period_timeframe);
135:     int tag_counter = 0;
136:     for (int i = 0; i < rows.length; i++) {
137:         LinearLayout lView = (LinearLayout) getActivity().findViewById(rows[
138:         i]);
139:         TextView left_item = (TextView) lView
140:             .findViewById(R.id.left_item);
141:         TextView right_item = (TextView) lView
142:             .findViewById(R.id.right_item);
143:         left_item
144:             .setText(Html.fromHtml("<b>" + locale_t
145:             ags[0] + "</b>"
146:             + "<br>"
147:             + current_period.get(ta
148:             gs_order[tag_counter]));
149:         right_item.setText(Html.fromHtml("<b>" + locale_tags[1]
150:             + "</b>" + "<br>"
151:             + current_period.get(tags_order[tag_cou
152:             nter + 1]));
153:         left_item.setGravity(Gravity.CENTER);
154:         right_item.setGravity(Gravity.CENTER);
155:         if (fDate
156:             .compareTo(current_period.get(tags_orde
157:             r[tag_counter])) >= 0
158:             && fDate.compareTo(current_period
159:             .get(tags_order[tag_cou
160:             nter + 1])) <= 0) {
161:             lView.setBackgroundColor(Color.GREEN);
162:             time_sendable = true;
163:         } else {
164:             lView.setBackgroundColor(Color.RED);
165:         }
166:         tag_counter = tag_counter + 2;
167:     }
168:     }
169:     private String setTargetUrl() {
170:         final String XML_EXTENSION = ".xml";
171:         final String PERIODS = "periods";
172:         GlobalState gs = (GlobalState) getActivity().getApplication();

```

```

174:         String base_url = gs.getTargetAddress();
175:
176:         String target_url = base_url + PERIODS + XML_EXTENSION;
177:
178:         return target_url;
179:     }
180: }
181:
182: @Override
183: public void onClick(View v) {
184:
185:     Boolean state_sendable = determineState();
186:     String academic_year = determineAcademicYear();
187:     String reason = null;
188:     if (time_sendable && state_sendable) {
189:         if ((current_period_timeframe.equals(academic_year))) {
190:             new sendDegreeToEvaluate().execute();
191:         } else {
192:             reason = getResources().getString(R.string.outOfPeriod)
;
193:             Toast.makeText(getActivity(), reason, Toast.LENGTH_LONG
).show();
194:
195:         }
196:
197:     } else {
198:
199:         if (!time_sendable) {
200:             reason = getResources().getString(R.string.outOfTime);
201:         }
202:         if (!state_sendable) {
203:             if (reason != null) {
204:                 reason = reason + " "
205:                     + getResources().getString(R.st
ring.incorrectState);
206:             } else
207:                 reason = getResources().getString(R.string.inco
rrectState);
208:
209:         }
210:         Toast.makeText(getActivity(), reason, Toast.LENGTH_LONG).show()
;
211:     }
212: }
213:
214: private Boolean determineState() {
215:     Bundle bundle = getArguments();
216:     String state;
217:     Boolean state_sendable = false;
218:     if (bundle != null) {
219:         state = bundle.getString("state");
220:     } else {
221:         state = getActivity().getIntent().getStringExtra("state");
222:     }
223:     if (state.equals("enviado") || state.equals("cancelado")
|| state.equals("rechazado")) {
224:         state_sendable = false;
225:     } else {
226:         state_sendable = true;
227:     }
228: }
229: return state_sendable;
230: }
231:

```

```
232:     private class sendDegreeToEvaluate extends AsyncTask<Void, Void, Void> {
233:         IOException error;
234:
235:         @Override
236:         protected void onPreExecute() {
237:             ((MyFragmentActivity) getActivity())
238:                 .setUpProgressDialog(R.string.sendingDegree);
239:         }
240:
241:         @Override
242:         protected Void doInBackground(Void... params) {
243:             String target_url = setTargetUrl();
244:             XmlPage xml_page = new XmlPage(getActivity());
245:             try {
246:                 xml_page.getPage(target_url);
247:             } catch (IOException e) {
248:                 error = e;
249:                 return null;
250:             }
251:             return null;
252:         }
253:
254:         @Override
255:         protected void onPostExecute(Void param) {
256:
257:             ((MyFragmentActivity) getActivity()).dismissWindow();
258:             if (error != null) {
259:                 ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
260:             } else {
261:                 Toast.makeText(getActivity(),
262:                     getResources().getString(R.string.sent)
263:
264:                         Toast.LENGTH_LONG).show();
265:             }
266:         }
267:     }
268:
269:     private String setTargetUrl() {
270:         final String UNIV_SPEC_DEG = "university_specific_degrees";
271:         final String CHECK = "check";
272:         final String SLASH = "/";
273:
274:         GlobalState gs = (GlobalState) getActivity().getApplication();
275:         String base_url = gs.getTargetAddress();
276:         Bundle bundle = getArguments();
277:         String id;
278:
279:         if (bundle != null) {
280:             id = bundle.getString("id");
281:         } else {
282:             id = getActivity().getIntent().getStringExtra("id");
283:         }
284:
285:         String target_url = base_url + UNIV_SPEC_DEG + id + SLASH + CHECK;
286:
287:         return target_url;
288:     }
289:
290:     private String determineAcademicYear() {
291:         String edition;
292:         Bundle bundle = getArguments();
```

```
293:         if (bundle != null) {
294:             edition = bundle.getString("edition");
295:         } else {
296:             edition = getActivity().getIntent().getStringExtra("edition");
297:         }
298:
299:         String year = edition.substring(0, 4);
300:         String month = edition.substring(5, 7);
301:         String academic_year;
302:         if (Integer.parseInt(month) > 8) {
303:             int academic_year_end = (Integer.parseInt(year) + 1);
304:             academic_year = year + "/" + String.valueOf(academic_year_end);
305:         } else {
306:             int academic_year_start = (Integer.parseInt(year) - 1);
307:             academic_year = academic_year_start + "/" + year;
308:         }
309:
310:         return academic_year;
311:     }
312: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.HashMap;
5:
6: import android.os.AsyncTask;
7: import android.os.Bundle;
8: import android.support.v4.app.Fragment;
9: import android.view.LayoutInflater;
10: import android.view.View;
11: import android.view.ViewGroup;
12: import android.view.WindowManager;
13:
14: /**
15:  *
16:  * @author ubay
17:  *
18:  *     Clase encargada de descargar la informacion economica de una
19:  *     propuesta con ID recibida de la actividad padre
20:  */
21: public class EconomicInfoFragment extends Fragment {
22:     public static HashMap<String, String> hashMap;
23:
24:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
25:         Bundle savedInstanceState) {
26:         View myFragmentView = inflater.inflate(R.layout.economic_info_fragment,
27:             container, false);
28:
29:         getActivity().getWindow().setSoftInputMode(
30:             WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDD
EN);
31:
32:         new retrieveInfo().execute();
33:
34:         return myFragmentView;
35:     }
36:
37:     private class retrieveInfo extends AsyncTask<String, Void, Void> {
38:         IOException error;
39:
40:         @Override
41:         protected void onPreExecute() {
42:             ((MyFragmentActivity) getActivity())
43:                 .setUpProgressDialog(R.string.loadingMessage);
44:         }
45:
46:         @Override
47:         protected Void doInBackground(String... params) {
48:             String target_url = setTargetURL();
49:             String economic_data = getXmlPage(target_url);
50:             String[] tags = getResources()
51:                 .getStringArray(R.array.economic_info);
52:
53:             XmlExtractor xml_extractor = new XmlExtractor(economic_data);
54:             hashMap = xml_extractor.extractTagsFromDoc(tags);
55:             return null;
56:
57:         }
58:
59:         private String setTargetURL() {
60:             final String UNIV_SPEC_DEG = "university_specific_degrees";
61:             final String ECONOMIC_REP = "economic_report";
62:             final String XML_EXTENSION = ".xml";

```

```
63:         final String SLASH = "/";
64:
65:         GlobalState gs = (GlobalState) getActivity().getApplication();
66:         String base_url = gs.getTargetAddress();
67:
68:         String degree_id = getActivity().getIntent().getStringExtra("id
");
69:
70:         String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_i
d
71:             + SLASH + ECONOMIC_REP + XML_EXTENSION;
72:
73:         return target_url;
74:     }
75:
76:     private String getXmlPage(String target_url) {
77:         XmlPage xml_page = new XmlPage(getActivity());
78:         String xml_response = null;
79:         try {
80:             xml_response = xml_page.getPage(target_url);
81:         } catch (IOException e) {
82:             error = e;
83:             return null;
84:         }
85:         return xml_response;
86:     }
87:
88:     @Override
89:     protected void onPostExecute(Void param) {
90:
91:         ((MyFragmentActivity) getActivity()).dismissWindow();
92:         if (error != null) {
93:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
94:         }
95:
96:     }
97:
98: }
99:
100: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.ArrayList;
5:
6: import android.os.AsyncTask;
7: import android.os.Bundle;
8: import android.support.v4.app.Fragment;
9: import android.view.LayoutInflater;
10: import android.view.View;
11: import android.view.ViewGroup;
12: import android.widget.LinearLayout;
13: import android.widget.TextView;
14:
15: /**
16:  *
17:  * @author ubay
18:  *
19:  *     Clase que se ocupa de descargar los mensajes de error asociados a la
20:  *     evaluacion de una propuesta y poblar el layout con dicha informacion
21:  */
22: public class ReportsInfoFragment extends Fragment {
23:     public static final String ENDTAG = "</issue>";
24:     public static final String MESSAGE = "message";
25:     View myView;
26:
27:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
28:         Bundle savedInstanceState) {
29:         myView = inflater.inflate(R.layout.reports_info_fragment, container,
30:             false);
31:         Bundle bundle = getArguments();
32:
33:         if (bundle == null) {
34:             bundle = getActivity().getIntent().getExtras();
35:         }
36:         String id = bundle.getString("id");
37:
38:         new retrieveIssues().execute(id);
39:
40:         return myView;
41:     }
42:
43:     private class retrieveIssues extends AsyncTask<String, Void, Integer> {
44:         String issues = null;
45:         ArrayList<String> message_list;
46:         IOException error;
47:
48:         @Override
49:         protected void onPreExecute() {
50:             ((MyFragmentActivity) getActivity())
51:                 .setUpProgressDialog(R.string.loadingMessage);
52:         }
53:
54:         protected Integer doInBackground(String... id) {
55:             XmlPage xml_issues = new XmlPage(getActivity());
56:             String target_url = setTargetUrl();
57:             try {
58:                 issues = xml_issues.getPage(target_url);
59:             } catch (IOException e) {
60:                 error = e;
61:                 return null;
62:             }
63:         }

```

```

64:         message_list = extractMessagesFromXmlPage(issues);
65:         return 1;
66:     }
67:
68:
69:     protected void onPostExecute(Integer result) {
70:         ((MyFragmentActivity) getActivity()).dismissWindow();
71:
72:         if (error != null) {
73:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
74:                 return;
75:         }
76:
77:         if (message_list.size() == 0) {
78:             setNoIssuesMessage();
79:
80:         } else {
81:             populateLayoutWithIssues(message_list);
82:
83:         }
84:     }
85: }
86:
87: private void populateLayoutWithIssues(ArrayList<String> message_list) {
88:     LinearLayout base_layout = (LinearLayout) myView
89:         .findViewById(R.id.base_issues_layout);
90:     for (int i = 0; i < message_list.size(); i++) {
91:         TextView message = new TextView(getActivity());
92:         message.setText(message_list.get(i));
93:         message.setBackgroundResource(R.drawable.blue_border);
94:         base_layout.addView(message);
95:     }
96: }
97:
98: private void setNoIssuesMessage() {
99:     LinearLayout base_layout = (LinearLayout) myView
100:         .findViewById(R.id.base_issues_layout);
101:     TextView message = new TextView(getActivity());
102:     message.setText(R.string.message_no_errors);
103:     message.setBackgroundResource(R.drawable.blue_border);
104:     base_layout.addView(message);
105: }
106:
107: private ArrayList<String> extractMessagesFromXmlPage(String issues) {
108:     XmlExtractor myXmlExtractor = new XmlExtractor(issues);
109:     ArrayList<String> message_list = new ArrayList<String>();
110:
111:     ArrayList<String> issues_list = myXmlExtractor.unitSeparator(ENDTAG);
112:
113:     for (int i = 0; i < issues_list.size(); i++) {
114:         String single_issue = issues_list.get(i);
115:         XmlExtractor single_issueXmlExtractor = new XmlExtractor(
116:             single_issue);
117:         message_list
118:             .add(single_issueXmlExtractor.extractSingleTag(
MESSAGE));
119:     }
120:     return message_list;
121: }
122:
123: private String setTargetUrl() {
124:     final String UNIV_SPEC_DEG = "university_specific_degrees";

```

```
125:         final String XML_EXTENSION = ".xml";
126:         final String SLASH = "/";
127:         final String ISSUES = "issues";
128:
129:         GlobalState gs = (GlobalState) getActivity().getApplication();
130:         String base_url = gs.getTargetAddress();
131:
132:         String degree_id = "";
133:         Bundle bundle = getArguments();
134:
135:         if (bundle != null) {
136:             degree_id = bundle.getString("id");
137:         } else {
138:             degree_id = getActivity().getIntent().getStringExtra("id");
139:         }
140:
141:         String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_id
142:             + SLASH + ISSUES + XML_EXTENSION;
143:
144:         return target_url;
145:     }
146: }
```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import java.io.File;
4: import java.io.FileOutputStream;
5: import java.io.IOException;
6: import java.io.InputStream;
7: import org.apache.http.HttpEntity;
8: import org.apache.http.HttpResponse;
9: import org.apache.http.client.ClientProtocolException;
10: import org.apache.http.client.methods.HttpGet;
11: import org.apache.http.impl.client.DefaultHttpClient;
12: import org.apache.http.protocol.HttpContext;
13: import android.annotation.SuppressLint;
14: import android.annotation.TargetApi;
15: import android.app.DownloadManager;
16: import android.content.ActivityNotFoundException;
17: import android.content.Context;
18: import android.content.Intent;
19: import android.content.pm.ActivityInfo;
20: import android.net.Uri;
21: import android.os.AsyncTask;
22: import android.os.Build;
23: import android.os.Bundle;
24: import android.os.Environment;
25: import android.support.v4.app.NavUtils;
26: import android.support.v4.view.ViewPager.LayoutParams;
27: import android.view.Menu;
28: import android.view.MenuItem;
29: import android.view.View;
30: import android.widget.FrameLayout;
31: import android.widget.LinearLayout;
32: import android.widget.Toast;
33:
34: public class DegreeMainMenu extends MyFragmentActivity implements
35:     DegreeMainMenuFragment.OnAcademicInfoClickedListener,
36:     DegreeMainMenuFragment.OnEconomicInfoClickedListener,
37:     DegreeMainMenuFragment.OnReportClickedListener,
38:     DegreeMainMenuFragment.OnSubjectsClickedListener,
39:     DegreeMainMenuFragment.OnTeachersClickedListener {
40:
41:     private Context baseContext = this;
42:
43:     @Override
44:     protected void onCreate(Bundle savedInstanceState) {
45:         super.onCreate(savedInstanceState);
46:         setContentView(R.layout.degree_main_menu_activity);
47:
48:         if (getResources().getBoolean(R.bool.portrait_only)) {
49:             setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
50:         }
51:     }
52:
53:     /**
54:      * Set up the {@link android.app.ActionBar}, if the API is available.
55:      */
56:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
57:     private void setupActionBar() {
58:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
59:             getActionBar().setDisplayHomeAsUpEnabled(true);
60:         }
61:     }
62:

```

```

63:         @Override
64:         public boolean onCreateOptionsMenu(Menu menu) {
65:             // Inflate the menu; this adds items to the action bar if it is present
.
66:             getMenuInflater().inflate(R.menu.degree_main_menu, menu);
67:             return true;
68:         }
69:
70:         @Override
71:         public boolean onOptionsItemSelected(MenuItem item) {
72:             switch (item.getItemId()) {
73:                 case android.R.id.home:
74:                     // This ID represents the Home or Up button. In the case of thi
S
75:                     // activity, the Up button is shown. Use NavUtils to allow user
S
76:                     // to navigate up one level in the application structure. For
77:                     // more details, see the Navigation pattern on Android Design:
78:                     //
79:                     // http://developer.android.com/design/patterns/navigation.html
#up-vs-back
80:                     //
81:                     NavUtils.navigateUpFromSameTask(this);
82:                     return true;
83:                 }
84:             return super.onOptionsItemSelected(item);
85:         }
86:
87:         public void academicInfo(View view) {
88:             OnAcademicInfoClicked();
89:         }
90:
91:         public void subjectsInfo(View view) {
92:             OnSubjectsClicked();
93:         }
94:
95:         public void economicInfo(View view) {
96:             OnEconomicInfoClicked();
97:         }
98:
99:         public void teachersInfo(View view) {
100:             OnTeachersClicked();
101:         }
102:
103:         public void reportInfo(View view) {
104:             OnReportClicked();
105:         }
106:
107:         public void sendDegreeToEvaluate(View view) {
108:             if (findViewById(R.id.central_fragment) == null) {
109:                 Intent intent = prepareIntent();
110:                 intent.putExtra("edition", getIntent().getStringExtra("edition"
));
111:                 intent.putExtra("state", getIntent().getStringExtra("state"));
112:                 intent.setClass(this, Periods.class);
113:                 startActivity(intent);
114:             } else {
115:                 hideUnwantedFragmentLayouts();
116:                 unfoldFullScreenFragmentLayout();
117:
118:                 Bundle args = getIntent().getExtras();
119:                 PeriodsFragment periodsFragment = new PeriodsFragment();
120:                 periodsFragment.setArguments(args);

```

```

121:         getSupportFragmentManager().beginTransaction()
122:             .replace(R.id.full_screen_fragment, periodsFrag
ment)
123:             .commit();
124:     }
125: }
126:
127: @Override
128: public void OnAcademicInfoClicked() {
129:     if (findViewById(R.id.central_fragment) == null) {
130:
131:         Intent intent = prepareIntent();
132:         intent.setClass(this, AcademicInfo.class);
133:         startActivity(intent);
134:     } else {
135:         hideUnwantedFragmentLayouts();
136:         unfoldFullScreenFragmentLayout();
137:         Bundle args = getIntent().getExtras();
138:         AcademicInfoFragment academicInfoFragment = new AcademicInfoFra
gment();
139:         academicInfoFragment.setArguments(args);
140:         getSupportFragmentManager().beginTransaction()
141:             .replace(R.id.full_screen_fragment, academicInf
oFragment)
142:             .commit();
143:     }
144: }
145: }
146:
147: @Override
148: public void OnSubjectsClicked() {
149:     if (findViewById(R.id.central_fragment) == null) {
150:
151:         Intent intent = prepareIntent();
152:         intent.setClass(this, SubjectsList.class);
153:         startActivity(intent);
154:     } else {
155:         hideFullScreenLayout();
156:         unfoldSmallerFrameLayouts();
157:
158:         Bundle args = getIntent().getExtras();
159:         SubjectsListFragment subjectsListFragment = new SubjectsListFra
gment();
160:         subjectsListFragment.setArguments(args);
161:         getSupportFragmentManager().beginTransaction()
162:             .replace(R.id.central_fragment, subjectsListFra
gment)
163:             .commit();
164:     }
165: }
166:
167: @Override
168: public void OnTeachersClicked() {
169:     if (findViewById(R.id.central_fragment) == null) {
170:
171:         Intent intent = prepareIntent();
172:         intent.setClass(this, TeachersList.class);
173:         startActivity(intent);
174:     } else {
175:         hideFullScreenLayout();
176:         unfoldSmallerFrameLayouts();
177:
178:         Bundle args = getIntent().getExtras();

```

```

179:         TeachersListFragment teachersListFragment = new TeachersListFra
gment());
180:         teachersListFragment.setArguments(args);
181:         getSupportFragmentManager().beginTransaction()
182:             .replace(R.id.central_fragment, teachersListFra
gment)
183:             .commit();
184:     }
185: }
186:
187: @Override
188: public void OnEconomicInfoClicked() {
189:     if (findViewById(R.id.central_fragment) == null) {
190:
191:         Intent intent = prepareIntent();
192:         intent.setClass(this, EconomicInfo.class);
193:         startActivity(intent);
194:     } else {
195:         hideFullscreenLayout();
196:         unfoldSmallerFrameLayouts();
197:
198:         Bundle args = getIntent().getExtras();
199:         EconomicInfoFragment economicInfoFragment = new EconomicInfoFra
gment());
200:         economicInfoFragment.setArguments(args);
201:         getSupportFragmentManager().beginTransaction()
202:             .replace(R.id.central_fragment, economicInfoFra
gment)
203:             .commit();
204:     }
205: }
206:
207: @Override
208: public void OnReportClicked() {
209:     if (findViewById(R.id.central_fragment) == null) {
210:
211:         Intent intent = prepareIntent();
212:         intent.setClass(this, ReportsInfo.class);
213:         startActivity(intent);
214:     } else {
215:         hideUnwantedFragmentLayouts();
216:         unfoldFullscreenFragmentLayout();
217:
218:         Bundle args = getIntent().getExtras();
219:         ReportsInfoFragment reportsInfoFragment = new ReportsInfoFragme
nt());
220:         reportsInfoFragment.setArguments(args);
221:         getSupportFragmentManager().beginTransaction()
222:             .replace(R.id.full_screen_fragment, reportsInfo
Fragment)
223:             .commit();
224:     }
225: }
226:
227: public void downloadDegree(View view) {
228:     String pdf_url = getPDF_URL();
229:
230:     new downloadPdf().execute(pdf_url);
231: }
232: }
233:
234: public void overview(View view) {
235:     hideFullscreenLayout();

```

```
236:         unfoldSmallerFrameLayouts();
237:
238:         Bundle args = getIntent().getExtras();
239:         TabOverview economicInfoFragment = new TabOverview();
240:         economicInfoFragment.setArguments(args);
241:         getSupportFragmentManager().beginTransaction()
242:             .replace(R.id.lateral_fragment, economicInfoFragment).c
ommit();
243:     }
244:
245:     public void income(View view) {
246:         hideFullscreenLayout();
247:         unfoldSmallerFrameLayouts();
248:
249:         Bundle args = getIntent().getExtras();
250:         TabIncome tabIncomeFragment = new TabIncome();
251:         tabIncomeFragment.setArguments(args);
252:         getSupportFragmentManager().beginTransaction()
253:             .replace(R.id.lateral_fragment, tabIncomeFragment).comm
it();
254:     }
255:
256:     public void expenditure(View view) {
257:         hideFullscreenLayout();
258:         unfoldSmallerFrameLayouts();
259:
260:         Bundle args = getIntent().getExtras();
261:         TabExpenditure tabExpenditureFragment = new TabExpenditure();
262:         tabExpenditureFragment.setArguments(args);
263:         getSupportFragmentManager().beginTransaction()
264:             .replace(R.id.lateral_fragment, tabExpenditureFragment)
265:             .commit();
266:     }
267:
268:     private void unfoldFullscreenFragmentLayout() {
269:         FrameLayout fullScreenLayout = (FrameLayout) findViewById(R.id.full_scr
een_fragment);
270:         LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(0,
271:             LayoutParams.MATCH_PARENT, 2);
272:
273:         fullScreenLayout.setLayoutParams(lp);
274:
275:     }
276:
277:     private void hideUnwantedFragmentLayouts() {
278:         FrameLayout centralLayout = (FrameLayout) findViewById(R.id.central_fra
gment);
279:         FrameLayout lateralLayout = (FrameLayout) findViewById(R.id.lateral_fra
gment);
280:
281:         LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(0,
282:             LayoutParams.MATCH_PARENT, 0);
283:         centralLayout.setLayoutParams(lp);
284:         lateralLayout.setLayoutParams(lp);
285:     }
286:
287:     private void unfoldSmallerFrameLayouts() {
288:         FrameLayout centralLayout = (FrameLayout) findViewById(R.id.central_fra
gment);
289:         FrameLayout lateralLayout = (FrameLayout) findViewById(R.id.lateral_fra
gment);
290:
291:         LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(0,
```

```

292:             LayoutParams.MATCH_PARENT, 1);
293:         centralLayout.setLayoutParams(lp);
294:         lateralLayout.setLayoutParams(lp);
295:     }
296:
297:     private void hideFullscreenLayout() {
298:         FrameLayout fullScreenLayout = (FrameLayout) findViewById(R.id.full_scr
een_fragment);
299:
300:         LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(0,
301:             LayoutParams.MATCH_PARENT, 0);
302:         fullScreenLayout.setLayoutParams(lp);
303:     }
304:
305:     private Intent prepareIntent() {
306:         Bundle bundle = getIntent().getExtras();
307:         Intent intent = new Intent();
308:         intent.putExtra("id", bundle.getString("id"));
309:         return intent;
310:     }
311:
312:     private String getPDF_URL() {
313:         final String UNIV_SPEC_DEG = "university_specific_degrees";
314:         final String PDF_EXTENSION = ".pdf";
315:         final String SLASH = "/";
316:
317:         GlobalState gs = (GlobalState) getApplication();
318:         String base_url = gs.getTargetAddress();
319:         String degree_id = getIntent().getExtras().getString("id");
320:
321:         String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_id
322:             + PDF_EXTENSION;
323:         return target_url;
324:     }
325:
326:     private class downloadPdf extends AsyncTask<String, Integer, Integer> {
327:
328:         IOException error;
329:         File file;
330:
331:         @Override
332:         protected void onPreExecute() {
333:             ((MyFragmentActivity) baseContext)
334:                 .setUpProgressDialog(R.string.downloading);
335:         }
336:
337:         @Override
338:         protected Integer doInBackground(String... params) {
339:             GlobalState gs = (GlobalState) baseContext.getApplicationContext
t();
340:             HttpContext localContext = gs.getLocalContext();
341:             DefaultHttpClient client = gs.getHttpClient();
342:
343:             HttpGet httpGet = new HttpGet(params[0]);
344:
345:             HttpResponse responseGET = null;
346:             try {
347:                 responseGET = client.execute(httpGet, localContext);
348:             } catch (ClientProtocolException e) {
349:                 e.printStackTrace();
350:             } catch (IOException e) {
351:                 error = e;
352:                 e.printStackTrace();

```

```

353:         return null;
354:     }
355:     HttpEntity getEntity = responseGET.getEntity();
356:     File path;
357:
358:     if (getEntity != null) {
359:         try {
360:             InputStream in = getEntity.getContent();
361:             if (!Environment.getExternalStorageState().equals(
ls(
362:                 Environment.MEDIA_MOUNTED)) {
363:                 // SD card is not available
364:                 return null;
365:             }
366:             path = Environment
367:                 .getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
368:             path.mkdirs();
369:             String name_id_degree = getIntent().getStringExtra(
.getString(
370:                 "id");
371:             file = new File(path, name_id_degree + ".pdf");
372:             FileOutputStream fos = new FileOutputStream(file);
e);
373:
374:             byte[] buffer = new byte[1024];
375:             int len1 = 0;
376:             int cont = 0;
377:             while ((len1 = in.read(buffer)) > 0) {
378:                 fos.write(buffer, 0, len1);
379:                 cont += len1;
380:                 publishProgress(cont);
381:             }
382:             fos.close();
383:         } catch (IOException ex) {
384:             ex.printStackTrace();
385:             error = ex;
386:             return null;
387:
388:         } catch (RuntimeException ex) {
389:             httpGet.abort();
390:             throw ex;
391:         }
392:     }
393:     return null;
394:
395: }
396:
397: protected void onProgressUpdate(Integer... values) {
398:     updateProgressDialog(baseContext.getResources().getString(
399:         R.string.downloading)
400:         + "<br>"
401:         + values[0]
402:         + " "
403:         + baseContext.getResources().getString(R.string
.downloaded));
404: }
405:
406: @SuppressWarnings("NewApi")
407: protected void onPostExecute(Integer result) {
408:     ((MyFragmentActivity) baseContext).dismissWindow();
409:
410:     if (error != null) {

```

```

411:         ((MyFragmentActivity) baseContext).launchErrorWindow(1)
;
412:         return;
413:     }
414:
415:     if (file != null) {
416:         if (file.exists()) {
417:             Uri path = Uri.fromFile(file);
418:             Intent intent = new Intent(Intent.ACTION_VIEW);
419:             intent.setDataAndType(path, "application/pdf");
420:             intent.setFlags(Intent.FLAG_ACTIVITY_BROUGHT_TO
_FRONT);
421:
422:             try {
423:                 startActivity(intent);
424:             } catch (ActivityNotFoundException e) {
425:                 Toast.makeText(
426:                     baseContext,
427:                     getResources().getStrin
g(
428:                         R.strin
g.noAppAvailable),
429:                     Toast.LENGTH_LONG).show
());
430:             }
431:         }
432:     } else
433:         Toast.makeText(baseContext,
434:             getResources().getString(R.string.noSto
rageAvailable),
435:             Toast.LENGTH_LONG).show();
436:
437:     if (((int) Build.VERSION.SDK_INT) >= 12) {
438:         DownloadManager mgr = (DownloadManager) getSystemServic
e(DOWNLOAD_SERVICE);
439:         mgr.addCompletedDownload(
440:             getIntent().getExtras().getString("id")
',
441:             "Memoria propuesta tÃ-tulo propio",
442:             false,
443:             "application/pdf",
444:             Environment
445:                 .getExternalStoragePubl
icDirectory(Environment.DIRECTORY_DOWNLOADS)
446:                 + "/"
447:                 + getIntent().getExtras
().getString("id")
448:                 + ".pdf", file.length()
, true);
449:     }
450:     return;
451: }
452: }
453:
454: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.util.ArrayList;
4:
5: import android.os.Bundle;
6: import android.support.v4.app.Fragment;
7: import android.view.LayoutInflater;
8: import android.view.View;
9: import android.view.ViewGroup;
10: import android.widget.ListView;
11:
12: /**
13:  *
14:  * @author ubay
15:  *
16:  *     La clase SearchResultsFragment recibe la informacion de las
17:  *     propuestas incluidas en la busqueda realizada en la actividad padre,
18:  *     en forma de listado
19:  */
20: public class SearchResultsFragment extends Fragment {
21:     private View myFragmentView;
22:     ArrayList<ArrayList<String>> degree_list;
23:     ArrayList<ArrayList<String>> fetch;
24:
25:     String[] degree_codes;
26:
27:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
28:         Bundle savedInstanceState) {
29:
30:         Bundle bundle = getArguments();
31:         if (bundle == null) {
32:             bundle = getActivity().getIntent().getExtras();
33:         }
34:         myFragmentView = inflater.inflate(R.layout.search_results_fragment,
35:             container, false);
36:
37:         populateDegreeListFromInput(bundle);
38:
39:         populateListAdapter();
40:
41:         initializeListView();
42:
43:         return myFragmentView;
44:     }
45:
46:
47:     private void initializeListView() {
48:
49:         ListView listview = (ListView) myFragmentView.findViewById(R.id.list);
50:         CustomDegreeAdapter custom_adapter = new CustomDegreeAdapter(
51:             getActivity(), R.id.list, fetch);
52:         listview.setAdapter(custom_adapter);
53:     }
54:
55:     private void populateListAdapter() {
56:         fetch = new ArrayList<ArrayList<String>>();
57:         for (int i = 0; i < degree_codes.length; i++) {
58:
59:             fetch.add(degree_list.get(i));
60:         }
61:     }
62:
63:     private void populateDegreeListFromInput(Bundle bundle) {
```

```
64:
65:         degree_codes = bundle.getStringArray("codes");
66:         degree_list = new ArrayList<ArrayList<String>>();
67:
68:         for (int i = 0; i < degree_codes.length; i++) {
69:             degree_list.add((ArrayList<String>) bundle
70:                 .getStringArrayList(degree_codes[i]));
71:         }
72:     }
73: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.FragmentActivity;
7: import android.support.v4.app.NavUtils;
8: import android.view.Menu;
9: import android.view.MenuItem;
10:
11: public class SingleSubject extends FragmentActivity {
12:
13:     @Override
14:     protected void onCreate(Bundle savedInstanceState) {
15:         super.onCreate(savedInstanceState);
16:         setContentView(R.layout.single_subject_activity);
17:     }
18:
19:     /**
20:      * Set up the {@link android.app.ActionBar}, if the API is available.
21:      */
22:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
23:     private void setupActionBar() {
24:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
25:             getActionBar().setDisplayHomeAsUpEnabled(true);
26:         }
27:     }
28:
29:     @Override
30:     public boolean onCreateOptionsMenu(Menu menu) {
31:         // Inflate the menu; this adds items to the action bar if it is present
32:
33:         getMenuInflater().inflate(R.menu.single_subject, menu);
34:         return true;
35:     }
36:
37:     @Override
38:     public boolean onOptionsItemSelected(MenuItem item) {
39:         switch (item.getItemId()) {
40:             case android.R.id.home:
41:                 // This ID represents the Home or Up button. In the case of thi
42:                 // activity, the Up button is shown. Use NavUtils to allow user
43:                 // to navigate up one level in the application structure. For
44:                 // more details, see the Navigation pattern on Android Design:
45:                 // http://developer.android.com/design/patterns/navigation.html
46:                 //
47:                 NavUtils.navigateUpFromSameTask(this);
48:                 return true;
49:             }
50:         return super.onOptionsItemSelected(item);
51:     }
52:
53: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.util.ArrayList;
5: import java.util.HashMap;
6:
7: import android.content.Intent;
8: import android.os.AsyncTask;
9: import android.os.Bundle;
10: import android.os.Handler;
11: import android.os.Message;
12: import android.support.v4.app.Fragment;
13: import android.text.Html;
14: import android.view.LayoutInflater;
15: import android.view.View;
16: import android.view.View.OnClickListener;
17: import android.view.ViewGroup;
18: import android.widget.Button;
19: import android.widget.ListView;
20:
21: /**
22:  *
23:  * @author ubay
24:  *
25:  * SubjectsListFragment recibe una id de la actividad padre, MainMenu, y
26:  * descarga la lista de asignaturas asociada a la propuesta que coincide
27:  * con el id. A continuacion, puebla el layout con la lista de
28:  * asignaturas y su informacion abreviada
29:  */
30: public class SubjectsListFragment extends Fragment implements
31:     CustomSubjectsAdapter.OnSubjectClickedListener, OnClickListener {
32:     private ListView listview;
33:     private CustomSubjectsAdapter custom_adapter;
34:     private ArrayList<ArrayList<String>> fetch = new ArrayList<ArrayList<String>>()
;
35:     private View myView;
36:     private Handler mHandler;
37:
38:     @Override
39:     public View onCreateView(LayoutInflater inflater, ViewGroup container,
40:         Bundle savedInstanceState) {
41:
42:         new retrieveInfo().execute();
43:
44:         myView = inflater.inflate(R.layout.subjects_list_fragment, container,
45:             false);
46:
47:         listview = (ListView) myView.findViewById(R.id.subject_list);
48:         custom_adapter = new CustomSubjectsAdapter(this, R.id.subject_list,
49:             fetch);
50:         listview.setAdapter(custom_adapter);
51:
52:         Button footer = (Button) myView.findViewById(R.id.amount_footer);
53:         footer.setOnClickListener(this);
54:
55:         // setUpFooterButton();
56:
57:         mHandler = new Handler() {
58:             @Override
59:             @SuppressWarnings("unchecked")
60:             public void handleMessage(Message msg) {
61:                 custom_adapter.add((ArrayList<String>) msg.obj);
62:                 custom_adapter.notifyDataSetChanged();

```

```
63:         }
64:     };
65:
66:     return myView;
67: }
68:
69: @Override
70: public void onClick(View v) {
71:     switch (v.getId()) {
72:     case R.id.amount_footer: {
73:         v.setVisibility(View.INVISIBLE);
74:         break;
75:     }
76:     }
77: }
78:
79: private class retrieveInfo extends AsyncTask<String, Void, Void> {
80:     ArrayList<String> subject_list;
81:     IOException error;
82:
83:     @Override
84:     protected void onPreExecute() {
85:         ((MyFragmentActivity) getActivity())
86:             .setUpProgressDialog(R.string.loadingMessage);
87:     }
88:
89:     @Override
90:     protected Void doInBackground(String... params) {
91:
92:         String target_url = setTargetUrl();
93:         XmlPage xml_subjects_list = new XmlPage(getActivity());
94:
95:         String subjects_list;
96:         try {
97:             subjects_list = xml_subjects_list.getPage(target_url);
98:         } catch (IOException e) {
99:             error = e;
100:            return null;
101:        }
102:
103:        XmlExtractor xml_extractor = new XmlExtractor(subjects_list);
104:        subject_list = new ArrayList<String>(
105:            xml_extractor.unitSeparator("</subject>"));
106:
107:        return null;
108:    }
109:
110:    private String setTargetUrl() {
111:        final String UNIV_SPEC_DEG = "university_specific_degrees";
112:        final String XML_EXTENSION = ".xml";
113:        final String SLASH = "/";
114:        final String SUBJECTS = "subjects";
115:
116:        GlobalState gs = (GlobalState) getActivity().getApplication();
117:        String base_url = gs.getTargetAddress();
118:
119:        String degree_id = "";
120:        Bundle bundle = getArguments();
121:
122:        if (bundle != null) {
123:            degree_id = bundle.getString("id");
124:        } else {
125:            degree_id = getActivity().getIntent().getStringExtra("i
```

```

d");
126:         }
127:
128:         String target_url = base_url + UNIV_SPEC_DEG + SLASH + degree_i
d
129:             + SLASH + SUBJECTS + XML_EXTENSION;
130:
131:         return target_url;
132:     }
133:
134:     protected void onPostExecute(Void param) {
135:
136:         ((MyFragmentActivity) getActivity()).dismissWindow();
137:         if (error != null) {
138:             ((MyFragmentActivity) getActivity()).launchErrorWindow(
0);
139:         } else {
140:             populateViews(subject_list);
141:         }
142:
143:     }
144:
145:     private void populateViews(ArrayList<String> subject_list) {
146:
147:         for (int i = 0; i < subject_list.size(); i++) {
148:             XmlExtractor xml_extractor = null;
149:             xml_extractor = new XmlExtractor(subject_list.get(i));
150:             HashMap<String, String> single_subject_data = null;
151:
152:             single_subject_data = new HashMap<String, String>(
153:                 xml_extractor.extractTagsFromDoc(getRes
ources()
154:                 .getStringArray(R.array
.complete_subject_info)));
155:             single_subject_data.put("whole", subject_list.get(i));
156:             SubjectInfoContainer subject_info = new SubjectInfoCont
ainer(
157:                 single_subject_data);
158:
159:             Message msg = new Message();
160:             msg.obj = subject_info.getArrayList();
161:             mHandler.sendMessage(msg);
162:         }
163:         Button footer = (Button) myView.findViewById(R.id.amount_footer
);
164:         footer.setText(Html.fromHtml("<b>" + subject_list.size() + "/"
165:             + subject_list.size() + " "
166:             + getResources().getString(R.string.loadedSubje
cts)
167:             + "</b>"));
168:
169:     }
170: }
171:
172: @Override
173: public void OnSubjectClicked(String subject_info) {
174:     if (getActivity().findViewById(R.id.lateral_fragment) == null) {
175:
176:         Intent intent = new Intent(getActivity(), SingleSubject.class);
177:         intent.putExtra("subject-info", subject_info);
178:         startActivity(intent);
179:     } else {
180:         Bundle args = new Bundle();

```

./SubjectsListFragment.java

Wed Nov 06 12:22:31 2013

4

```
181:         args.putString("subject-info", subject_info);
182:         SingleSubjectFragment singleSubjectFragment = new SingleSubject
Fragment();
183:         singleSubjectFragment.setArguments(args);
184:         getActivity().getSupportFragmentManager().beginTransaction()
185:             .replace(R.id.lateral_fragment, singleSubjectFr
agment)
186:             .commit();
187:     }
188: }
189:
190: }
```

```

1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.content.Intent;
4: import android.content.pm.ActivityInfo;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.MenuItem;
8:
9: public class SearchDegrees extends MyFragmentActivity implements
10:     SearchDegreesFragment.OnShowResultsListener {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.search_degrees_activity);
16:
17:         if (getResources().getBoolean(R.bool.portrait_only)) {
18:             setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAI
T);
19:         }
20:
21:         if (findViewById(R.id.fragment_container) != null) {
22:             if (savedInstanceState != null) {
23:                 return;
24:             }
25:             BlankFragment initialFragment = new BlankFragment();
26:             getSupportFragmentManager().beginTransaction()
27:                 .add(R.id.fragment_container, initialFragment).
commit();
28:
29:         }
30:     }
31:
32:     public void onShowResults(Degree[] degrees) {
33:         if (findViewById(R.id.fragment_container) == null) {
34:             Intent intent = new Intent(this, SearchResults.class);
35:             String[] codes = new String[degrees.length];
36:             for (int i = 0; i < degrees.length; i++) {
37:                 codes[i] = degrees[i].getData().get("code");
38:                 DegreeInfoContainer degree_with_simple_data = new Degree
eInfoContainer(
39:                     degrees[i].getData());
40:                 intent.putExtra(degrees[i].getData().get("code"),
41:                     degree_with_simple_data.getArrayList());
;
42:             }
43:             intent.putExtra("codes", codes);
44:             startActivity(intent);
45:         } else {
46:             Bundle args = new Bundle();
47:             String[] codes = new String[degrees.length];
48:
49:             for (int i = 0; i < degrees.length; i++) {
50:                 codes[i] = degrees[i].getData().get("code");
51:                 DegreeInfoContainer degree_with_simple_data = new Degree
eInfoContainer(
52:                     degrees[i].getData());
53:                 args.putStringArrayList(degrees[i].getData().get("code"
),
54:                     degree_with_simple_data.getArrayList())
;
55:             }
56:             args.putStringArray("codes", codes);

```

```
57:         SearchResultsFragment searchResultsFragment = new SearchResults
Fragment();
58:         searchResultsFragment.setArguments(args);
59:         getSupportFragmentManager().beginTransaction()
60:             .replace(R.id.fragment_container, searchResults
Fragment)
61:             .commit();
62:     }
63:
64: }
65:
66: @Override
67: public boolean onOptionsItemSelected(MenuItem item) {
68:     switch (item.getItemId()) {
69:     case android.R.id.home:
70:         // This ID represents the Home or Up button. In the case of thi
s
71:         // activity, the Up button is shown. Use NavUtils to allow user
s
72:         // to navigate up one level in the application structure. For
73:         // more details, see the Navigation pattern on Android Design:
74:         //
75:         // http://developer.android.com/design/patterns/navigation.html
#up-vs-back
76:         //
77:         NavUtils.navigateUpFromSameTask(this);
78:         return true;
79:     }
80:     return super.onOptionsItemSelected(item);
81: }
82:
83: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.app.Activity;
4: import android.content.Intent;
5: import android.content.pm.ActivityInfo;
6: import android.os.Bundle;
7: import android.view.Menu;
8: import android.view.View;
9: import android.widget.Button;
10:
11: /**
12:  *
13:  * @author ubay
14:  *
15:  *     Simple clase que muestra el layout que permite pulsar en la opcion de
16:  *     rol de la aplicacion. Una vez seleccionado el rol, lanza la
17:  *     aplicacion LoginUser
18:  */
19: public class SelectRole extends Activity {
20:
21:     private String ROLE = "manager";
22:
23:     @Override
24:     protected void onCreate(Bundle savedInstanceState) {
25:         super.onCreate(savedInstanceState);
26:         setContentView(R.layout.select_role_activity);
27:
28:         if (getResources().getBoolean(R.bool.portrait_only)) {
29:             setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAI
T);
30:         }
31:
32:         Button button_teacher = (Button) findViewById(R.id.button_teacher);
33:         Button button_manager = (Button) findViewById(R.id.button_manager);
34:
35:         button_teacher.setOnClickListener(new View.OnClickListener() {
36:             public void onClick(View v) {
37:                 Intent intent = new Intent(getApplicationContext(), LoginUser.
class);
38:                 intent.putExtra(ROLE, false);
39:                 startActivity(intent);
40:             }
41:         });
42:         button_manager.setOnClickListener(new View.OnClickListener() {
43:             public void onClick(View v) {
44:                 Intent intent = new Intent(getApplicationContext(), LoginUser.
class);
45:                 intent.putExtra(ROLE, true);
46:                 startActivity(intent);
47:             }
48:         });
49:
50:     }
51:
52:     @Override
53:     public boolean onCreateOptionsMenu(Menu menu) {
54:         // Inflate the menu; this adds items to the action bar if it is present
55:
56:         getMenuInflater().inflate(R.menu.select_role, menu);
57:         return true;
58:     }
59: }

```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4:
5: import org.apache.http.HttpEntity;
6: import org.apache.http.HttpResponse;
7: import org.apache.http.client.ClientProtocolException;
8: import org.apache.http.client.methods.HttpGet;
9: import org.apache.http.impl.client.DefaultHttpClient;
10: import org.apache.http.protocol.HttpContext;
11: import org.apache.http.util.EntityUtils;
12:
13: import android.content.Context;
14:
15: /**
16:  *
17:  * @author ubay
18:  *
19:  *     XmlPage se ocupa de realizar las peticiones contra la aplicacion web,
20:  *     proporcionandole una direccion destino. Es necesario inicializar la
21:  *     clase incluyendo contexto para poder tener acceso a la informacion
22:  *     contenida en el Singleton que proporciona los datos de sesion para la
23:  *     conexion
24:  */
25: public class XmlPage {
26:     private Context mContext;
27:
28:     public XmlPage(Context mContext) {
29:         this.mContext = mContext;
30:     }
31:
32:     public String getPage(String target_addr) throws IOException {
33:         GlobalState gs = (GlobalState) mContext.getApplicationContext();
34:         HttpContext localContext = gs.getLocalContext();
35:         DefaultHttpClient client = gs.getHttpClient();
36:
37:         HttpGet httpGet = new HttpGet(target_addr);
38:         HttpResponse responseGET = null;
39:         HttpEntity getEntity = null;
40:         String response = null;
41:         try {
42:             responseGET = client.execute(httpGet, localContext);
43:             getEntity = responseGET.getEntity();
44:             response = EntityUtils.toString(getEntity);
45:             getEntity.consumeContent();
46:         } catch (ClientProtocolException e) {
47:             e.printStackTrace();
48:         }
49:
50:         return response;
51:     }
52: }
53: }
```

```

1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.NavUtils;
7: import android.view.Menu;
8: import android.view.MenuItem;
9:
10: public class Periods extends MyFragmentActivity {
11:
12:     @Override
13:     protected void onCreate(Bundle savedInstanceState) {
14:         super.onCreate(savedInstanceState);
15:         setContentView(R.layout.periods_activity);
16:     }
17:
18:     /**
19:      * Set up the {@link android.app.ActionBar}, if the API is available.
20:      */
21:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
22:     private void setupActionBar() {
23:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
24:             getActionBar().setDisplayHomeAsUpEnabled(true);
25:         }
26:     }
27:
28:     @Override
29:     public boolean onCreateOptionsMenu(Menu menu) {
30:         // Inflate the menu; this adds items to the action bar if it is present
31:
32:         getMenuInflater().inflate(R.menu.periods, menu);
33:         return true;
34:     }
35:
36:     @Override
37:     public boolean onOptionsItemSelected(MenuItem item) {
38:         switch (item.getItemId()) {
39:             // This ID represents the Home or Up button. In the case of thi
40:             // activity, the Up button is shown. Use NavUtils to allow user
41:             // to navigate up one level in the application structure. For
42:             // more details, see the Navigation pattern on Android Design:
43:             //
44:             // http://developer.android.com/design/patterns/navigation.html
45:             //
46:             NavUtils.navigateUpFromSameTask(this);
47:             return true;
48:         }
49:         return super.onOptionsItemSelected(item);
50:     }
51:
52: }

```

```
1: package es.ulpgc.titulosproprios.gestion.ubay;
2:
3: import android.annotation.TargetApi;
4: import android.os.Build;
5: import android.os.Bundle;
6: import android.support.v4.app.FragmentActivity;
7: import android.support.v4.app.NavUtils;
8: import android.view.Menu;
9: import android.view.MenuItem;
10:
11: public class SingleTeacher extends FragmentActivity {
12:
13:     @Override
14:     protected void onCreate(Bundle savedInstanceState) {
15:         super.onCreate(savedInstanceState);
16:         setContentView(R.layout.single_teacher_activity);
17:     }
18:
19:     /**
20:      * Set up the {@link android.app.ActionBar}, if the API is available.
21:      */
22:     @TargetApi(Build.VERSION_CODES.HONEYCOMB)
23:     private void setupActionBar() {
24:         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
25:             getActionBar().setDisplayHomeAsUpEnabled(true);
26:         }
27:     }
28:
29:     @Override
30:     public boolean onCreateOptionsMenu(Menu menu) {
31:         // Inflate the menu; this adds items to the action bar if it is present
32:
33:         getMenuInflater().inflate(R.menu.single_teacher, menu);
34:         return true;
35:     }
36:
37:     @Override
38:     public boolean onOptionsItemSelected(MenuItem item) {
39:         switch (item.getItemId()) {
40:             case android.R.id.home:
41:                 // This ID represents the Home or Up button. In the case of thi
42:                 // activity, the Up button is shown. Use NavUtils to allow user
43:                 // to navigate up one level in the application structure. For
44:                 // more details, see the Navigation pattern on Android Design:
45:                 // http://developer.android.com/design/patterns/navigation.html
46:                 //
47:                 NavUtils.navigateUpFromSameTask(this);
48:                 return true;
49:             }
50:         return super.onOptionsItemSelected(item);
51:     }
52:
53: }
```

```
1: package es.ulpgc.titulospropios.gestion.ubay;
2:
3: import java.io.IOException;
4: import java.io.UnsupportedEncodingException;
5: import java.util.ArrayList;
6: import java.util.List;
7:
8: import org.apache.http.HttpEntity;
9: import org.apache.http.HttpResponse;
10: import org.apache.http.NameValuePair;
11: import org.apache.http.client.HttpClient;
12: import org.apache.http.client.entity.UrlEncodedFormEntity;
13: import org.apache.http.client.methods.HttpPut;
14: import org.apache.http.message.BasicNameValuePair;
15: import org.apache.http.protocol.HTTP;
16: import org.apache.http.protocol.HttpContext;
17: import org.apache.http.util.EntityUtils;
18:
19: import android.app.Activity;
20: import android.content.Context;
21: import android.os.AsyncTask;
22: import android.support.v4.app.Fragment;
23: import android.text.Html;
24: import android.util.Log;
25: import android.view.LayoutInflater;
26: import android.view.View;
27: import android.view.View.OnClickListener;
28: import android.view.ViewGroup;
29: import android.widget.AdapterView;
30: import android.widget.Button;
31: import android.widget.TextView;
32:
33: /**
34:  *
35:  * @author ubay
36:  *
37:  *     Clase encargada de almacenar las entradas de la lista de profesores que
38:  *     se muestra en la actividad TeachersList para su uso posterior
39:  */
40:
41: public class CustomTeacherAdapter extends ArrayAdapter<ArrayList<String>> {
42:     private ArrayList<ArrayList<String>> entries;
43:     private Fragment fragment;
44:
45:     OnTeacherClickedListener mCallback;
46:
47:     public static final String CONFIRMATION_PENDING = "Pendiente de confirmar parti
cipaci3n";
48:
49:     public CustomTeacherAdapter(Fragment a, int textViewResourceId,
50:         ArrayList<ArrayList<String>> entries, Context baseContext) {
51:         super(a.getActivity(), textViewResourceId, entries);
52:         this.entries = entries;
53:         this.fragment = a;
54:     }
55:
56:     public static class ViewHolder {
57:         public Button boton_nombre;
58:         public Button boton_notify;
59:
60:         public TextView type;
61:         public TextView state;
62:         public TextView last_notice;
```

```

63:         public TextView num_notice;
64:         public TextView degree;
65:         public TextView email;
66:         public String id;
67:         public String degree_id;
68:     }
69:
70:     public interface OnTeacherClickedListener {
71:         public void OnTeacherClicked(String teacher_info);
72:     }
73:
74:     @Override
75:     public View getView(int position, View convertView, ViewGroup parent) {
76:         View v = convertView;
77:         ViewHolder holder;
78:         if (v == null) {
79:             LayoutInflater vi = (LayoutInflater) fragment.getActivity()
80:                 .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
81:             v = vi.inflate(R.layout.single_teacher_row, null);
82:             holder = new ViewHolder();
83:             holder.boton_nombre = (Button) v.findViewById(R.id.boton_teacher);
84:             holder.boton_notify = (Button) v.findViewById(R.id.boton_notify);
85:             holder.type = (TextView) v.findViewById(R.id.textView_teacher_type);
86:             holder.state = (TextView) v
87:                 .findViewById(R.id.textView_teacher_state);
88:             holder.last_notice = (TextView) v
89:                 .findViewById(R.id.textView_teacher_lastnotified);
90:             holder.num_notice = (TextView) v
91:                 .findViewById(R.id.textView_teacher_numnotice);
92:             holder.degree = (TextView) v
93:                 .findViewById(R.id.textView_teacher_degree);
94:             holder.email = (TextView) v
95:                 .findViewById(R.id.textView_teacher_email);
96:
97:             v.setTag(holder);
98:         } else
99:             holder = (ViewHolder) v.getTag();
100:
101:         String[] teacher_tags = fragment.getActivity().getResources()
102:             .getStringArray(R.array.teacher_forAdapter);
103:         final ArrayList<String> custom = entries.get(position);
104:         @SuppressWarnings("unused")
105:         final Context baseContext = fragment.getActivity()
106:             .getApplicationContext();
107:         final String id = custom.get(8);
108:         final String degree_id = custom.get(9);
109:         final String participation_id = custom.get(10);
110:         final String whole_teacher = custom.get(11);
111:         final String name = new String(custom.get(0) + " " + custom.get(1));
112:         if (custom != null) {
113:             holder.id = id;
114:             holder.degree_id = degree_id;
115:             holder.boton_nombre.setText(Html.fromHtml("<b>" + name + "</b>"));
116:
117:             if (custom.get(3).equals(CONFIRMATION_PENDING)) {
118:                 holder.boton_notify.setText(Html.fromHtml(fragment
119:                     .getActivity().getResources()

```

```

120:                                     .getString(R.string.notify));
121:         holder.boton_notify.setVisibility(Button.VISIBLE);
122:     } else {
123:         holder.boton_notify.setVisibility(Button.GONE);
124:     }
125:     holder.type.setText(Html.fromHtml(teacher_tags[0] + custom.get(
2)))");
126:     holder.state
127:         .setText(Html.fromHtml(teacher_tags[1] + custom
.get(3)));
128:     holder.last_notice.setText(Html.fromHtml(teacher_tags[2]
129:         + custom.get(4)));
130:     holder.num_notice.setText(Html.fromHtml(teacher_tags[3]
131:         + custom.get(5)));
132:     Log.d(custom.get(1), custom.get(5));
133:     holder.degree
134:         .setText(Html.fromHtml(teacher_tags[4] + custom
.get(6)));
135:     holder.email
136:         .setText(Html.fromHtml(teacher_tags[5] + custom
.get(7)));
137:     holder.boton_nombre.setOnClickListener(new OnClickListener() {
138:
139:         @Override
140:         public void onClick(View v) {
141:             ((OnTeacherClickedListener) fragment)
142:                 .OnTeacherClicked(whole_teacher
);
143:
144:         }
145:     });
146:
147:     final String degree_id_final = custom.get(9);
148:
149:     holder.boton_notify.setOnClickListener(new OnClickListener() {
150:
151:         @Override
152:         public void onClick(View v) {
153:             v.setVisibility(Button.GONE);
154:             new notifyTeacher().execute(participation_id,
155:                 degree_id_final);
156:         }
157:     });
158:     }
159:     return v;
160: }
161:
162: private class notifyTeacher extends AsyncTask<String, Void, Integer> {
163:     IOException error;
164:     private static final String BASE_URL = "university_specific_degrees/";
165:     private static final String TEACHERS_URL = "/degree_teachers/";
166:     private static final String REQUEST_CONF = "/request_confirmation";
167:     private static final String PARAM_DEGREE = "university_specific_degree_
id";
168:
169:     protected void onPreExecute() {
170:
171:         ((MyFragmentActivity) fragment.getActivity())
172:             .setUpProgressDialog(R.string.notifying);
173:
174:     }
175:
176:     @Override

```

```

177:         protected Integer doInBackground(String... params) {
178:             GlobalState gs = (GlobalState) ((Activity) fragment.getActivity
179:             ())
180:                 .getApplication();
181:             HttpClient client = gs.getHttpClient();
182:             HttpContext localContext = gs.getLocalContext();
183:             String target_address = gs.getTargetAddress();
184:             target_address = target_address + BASE_URL + params[1];
185:             target_address = target_address + TEACHERS_URL + params[0]
186:                 + REQUEST_CONF;
187:             HttpPut httpPut = new HttpPut(target_address);
188:             List<NameValuePair> put_params = new ArrayList<NameValuePair>()
;
189:             put_params.add(new BasicNameValuePair(PARAM_DEGREE, params[0]))
;
190:             put_params.add(new BasicNameValuePair("id", params[1]));
191:             UrlEncodedFormEntity ent = null;
192:             @SuppressWarnings("unused")
193:             String xml_response = null;
194:
195:             try {
196:                 ent = new UrlEncodedFormEntity(put_params, HTTP.UTF_8);
197:                 httpPut.setEntity(ent);
198:
199:                 HttpResponse responsePUT = client
200:                     .execute(httpPut, localContext);
201:                 HttpEntity getEntity = responsePUT.getEntity();
202:                 xml_response = EntityUtils.toString(getEntity);
203:                 getEntity.consumeContent();
204:
205:             } catch (UnsupportedEncodingException e) {
206:                 e.printStackTrace();
207:                 error = e;
208:                 return null;
209:             } catch (IOException e) {
210:                 e.printStackTrace();
211:                 error = e;
212:                 return null;
213:             }
214:
215:             return 1;
216:         }
217:
218:         protected void onPostExecute(Integer result) {
219:             ((MyFragmentActivity) fragment.getActivity()).dismissWindow();
220:             if (error != null) {
221:                 ((MyFragmentActivity) fragment.getActivity())
222:                     .launchErrorWindow(1);
223:
224:                 return;
225:             }
226:             return;
227:         }
228:     }
229: }

```

12.2 Código XML/Layouts/Strings

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="wrap_content"
4:     android:layout_height="wrap_content"
5:     android:background="@drawable/blue_border"
6:     android:orientation="horizontal" >
7:
8:     <include
9:         android:id="@+id/item"
10:        layout="@layout/info_cell" />
11:
12: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent">
5:     <fragment class="es.ulpgc.titulosproprios.gestion.ubay.PeriodsFragment"
6:         android:id="@+id/periods_fragment"
7:         android:layout_width="match_parent"
8:         android:layout_height="match_parent"/>
9: </FrameLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent"
5:     android:background="@drawable/background_colours"
6:     android:orientation="vertical" >
7:
8:     <LinearLayout
9:         android:layout_width="match_parent"
10:        android:layout_height="wrap_content"
11:        android:layout_marginTop="5dp"
12:        android:orientation="horizontal" >
13:
14:        <Button
15:            android:id="@+id/button_teacher"
16:            android:layout_width="0dp"
17:            android:layout_height="wrap_content"
18:            android:layout_gravity="center_vertical"
19:            android:layout_weight="7"
20:            android:background="@drawable/mybutton_layout" />
21:
22:        <Button
23:            android:id="@+id/button_notify"
24:            android:layout_width="0dp"
25:            android:layout_height="wrap_content"
26:            android:layout_gravity="center_vertical"
27:            android:layout_weight="3"
28:            android:background="@drawable/mybutton_layout"
29:            android:visibility="invisible" />
30:    </LinearLayout>
31:
32:    <LinearLayout
33:        android:layout_width="match_parent"
34:        android:layout_height="wrap_content"
35:        android:background="@drawable/blue_border"
36:        android:orientation="horizontal" >
37:
38:        <TextView
39:            android:id="@+id/textView_teacher_type"
40:            android:layout_width="0dp"
41:            android:layout_height="wrap_content"
42:            android:layout_weight="90" />
43:
44:        <View
45:            android:id="@+id/firstDivider"
46:            android:layout_width="1dp"
47:            android:layout_height="fill_parent"
48:            android:layout_weight="1"
49:            android:background="#3A4978" />
50:
51:        <TextView
52:            android:id="@+id/textView_teacher_state"
53:            android:layout_width="0dp"
54:            android:layout_height="wrap_content"
55:            android:layout_weight="90" />
56:    </LinearLayout>
57:
58:    <LinearLayout
59:        android:layout_width="match_parent"
60:        android:layout_height="wrap_content"
61:        android:background="@drawable/blue_border"
62:        android:orientation="horizontal" >
63:
```

```
64:         <TextView
65:             android:id="@+id/textView_teacher_lastnotified"
66:             android:layout_width="0dp"
67:             android:layout_height="wrap_content"
68:             android:layout_weight="90" />
69:
70:         <View
71:             android:id="@+id/secondDivider"
72:             android:layout_width="1dp"
73:             android:layout_height="fill_parent"
74:             android:layout_weight="1"
75:             android:background="#3A4978" />
76:
77:         <TextView
78:             android:id="@+id/textView_teacher_numnotice"
79:             android:layout_width="0dp"
80:             android:layout_height="wrap_content"
81:             android:layout_weight="90" />
82:     </LinearLayout>
83:
84:     <LinearLayout
85:         android:layout_width="match_parent"
86:         android:layout_height="wrap_content"
87:         android:background="@drawable/blue_border"
88:         android:orientation="horizontal" >
89:
90:         <TextView
91:             android:id="@+id/textView_teacher_degree"
92:             android:layout_width="0dp"
93:             android:layout_height="wrap_content"
94:             android:layout_weight="90" />
95:
96:         <View
97:             android:id="@+id/thirdDivider"
98:             android:layout_width="1dp"
99:             android:layout_height="fill_parent"
100:             android:layout_weight="1"
101:             android:background="#3A4978" />
102:
103:         <TextView
104:             android:id="@+id/textView_teacher_email"
105:             android:layout_width="0dp"
106:             android:layout_height="wrap_content"
107:             android:layout_weight="90" />
108:     </LinearLayout>
109:
110: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent"
5:     android:background="@drawable/background_colours"
6:     android:orientation="vertical" >
7:
8:     <Button
9:         android:id="@+id/button_subject"
10:        android:layout_marginTop="5dp"
11:        android:layout_width="match_parent"
12:        android:layout_height="wrap_content"
13:        android:background="@drawable/mybutton_layout"
14:        android:onClick="singleSubject" />
15:    <LinearLayout
16:        android:layout_width="match_parent"
17:        android:layout_height="wrap_content"
18:        android:orientation="horizontal"
19:        android:background="@drawable/blue_border" >
20:        <TextView
21:            android:id="@+id/textView_subject_course"
22:            android:layout_height="wrap_content"
23:            android:layout_width="0dp"
24:            android:layout_weight="90" />
25:        <View android:id="@+id/firstDivider"
26:            android:layout_height="fill_parent"
27:            android:layout_weight="1"
28:            android:layout_width="1dp"
29:            android:background="#3A4978" />
30:        <TextView
31:            android:id="@+id/textView_subject_semester"
32:            android:layout_height="wrap_content"
33:            android:layout_width="0dp"
34:            android:layout_weight="90" />
35:        </LinearLayout>
36:    <LinearLayout
37:        android:layout_width="match_parent"
38:        android:layout_height="wrap_content"
39:        android:orientation="horizontal"
40:        android:background="@drawable/blue_border" >
41:        <TextView
42:            android:layout_height="wrap_content"
43:            android:id="@+id/textView_subject_kind"
44:            android:layout_width="0dp"
45:            android:layout_weight="90" />
46:        <View android:id="@+id/secondDivider"
47:            android:layout_height="fill_parent"
48:            android:layout_weight="1"
49:            android:layout_width="1dp"
50:            android:background="#3A4978" />
51:        <TextView
52:            android:layout_height="match_parent"
53:            android:id="@+id/textView_subject_ects"
54:            android:layout_width="0dp"
55:            android:layout_weight="90" />
56:    </LinearLayout>
57:    <LinearLayout
58:        android:layout_width="match_parent"
59:        android:layout_height="wrap_content"
60:        android:orientation="horizontal"
61:        android:background="@drawable/blue_border" >
62:        <TextView
63:            android:id="@+id/textView_subject_starts"
```

```
64:         android:layout_height="wrap_content"
65:         android:layout_width="0dp"
66:         android:layout_weight="90" />
67:     <View    android:id="@+id/thirdDivider"
68:           android:layout_height="fill_parent"
69:           android:layout_weight="1"
70:           android:layout_width="1dp"
71:           android:background="#3A4978" />
72:     <TextView
73:       android:id="@+id/textView_subject_ends"
74:       android:layout_height="match_parent"
75:       android:layout_width="0dp"
76:       android:layout_weight="90" />
77: </LinearLayout>
78: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:baselineAligned="false"
4:     android:layout_width="match_parent"
5:     android:layout_height="wrap_content"
6:     android:orientation="horizontal"
7:     android:background="@drawable/blue_border" >
8:     <include android:id="@+id/left_item" android:layout_weight="90" layout="@layout/info_cell"/>
9:     <include android:layout_weight="1" layout="@layout/divider"/>
10:    <include android:id="@+id/center" android:layout_weight="90" layout="@layout/info_cell"/>
11:    <include android:layout_weight="1" layout="@layout/divider"/>
12:    <include android:id="@+id/right_item" android:layout_weight="90" layout="@layout/info_cell"/>
13: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:id="@+id/base_linear_search"
4:   android:layout_width="fill_parent"
5:   android:layout_height="fill_parent"
6:   android:background="@drawable/gradient_background"
7:   android:orientation="vertical" >
8:
9:   <Spinner
10:    android:id="@+id/edicion_spinner"
11:    android:layout_width="fill_parent"
12:    android:layout_height="0dp"
13:    android:layout_marginTop="5dp"
14:    android:layout_weight="10"
15:    android:background="@drawable/grey_buttons"
16:    android:prompt="@string/tipo" />
17:
18:   <Spinner
19:    android:id="@+id/tipo_spinner"
20:    android:layout_width="fill_parent"
21:    android:layout_height="0dp"
22:    android:layout_weight="10"
23:    android:background="@drawable/grey_buttons"
24:    android:prompt="@string/edicion" />
25:
26:   <Spinner
27:    android:id="@+id/estado_spinner"
28:    android:layout_width="fill_parent"
29:    android:layout_height="0dp"
30:    android:layout_weight="10"
31:    android:background="@drawable/grey_buttons"
32:    android:prompt="@string/estado" />
33:
34:   <EditText
35:    android:id="@+id/busqueda_codigo"
36:    android:layout_width="fill_parent"
37:    android:layout_height="0dp"
38:    android:layout_weight="10"
39:    android:hint="@string/busqueda_codigo" />
40:
41:   <EditText
42:    android:id="@+id/busqueda_nombre"
43:    android:layout_width="fill_parent"
44:    android:layout_height="0dp"
45:    android:layout_weight="10"
46:    android:hint="@string/busqueda_nombre" />
47:
48:   <Button
49:    android:id="@+id/search_button"
50:    android:layout_width="fill_parent"
51:    android:layout_height="0dp"
52:    android:layout_gravity="right"
53:    android:layout_weight="10"
54:    android:background="@drawable/grey_buttons"
55:    android:text="@string/buscar"
56:    android:textSize="24sp" />
57:
58:   <TextView
59:    android:layout_width="fill_parent"
60:    android:layout_height="0dp"
61:    android:layout_weight="25"
62:    android:gravity="center"
63:    android:text="@string/search_hint"
```

```
64:         android:textStyle="bold" />
65:
66: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="wrap_content"
5:     android:orientation="horizontal"
6:     android:background="@drawable/blue_border" >
7:     <TextView
8:         android:id="@+id/left_item"
9:         android:layout_height="wrap_content"
10:        android:layout_width="0dp"
11:        android:layout_weight="4" />
12:        <View
13:            android:layout_height="fill_parent"
14:            android:layout_weight="0.025"
15:            android:layout_width="1dp"
16:            android:background="#3A4978" />
17:        <TextView
18:            android:id="@+id/right_item"
19:            android:layout_height="wrap_content"
20:            android:layout_width="0dp"
21:            android:layout_weight="1.3" />
22: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent">
5:     <fragment class="es.ulpgc.titulosproprios.gestion.ubay.SingleSubjectFragment"
6:         android:id="@+id/single_subject_fragment"
7:         android:layout_width="match_parent"
8:         android:layout_height="match_parent"/>
9: </FrameLayout>
10:
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/background_colours"
6:   android:orientation="vertical"
7:   tools:context=".SelectTask" >
8:
9:   <Button
10:     android:layout_width="match_parent"
11:     android:layout_height="0dp"
12:     android:layout_marginTop="5dp"
13:     android:layout_weight="1"
14:     android:background="@drawable/mybutton_layout"
15:     android:gravity="center"
16:     android:onClick="newProposal"
17:     android:text="@string/new_proposal"
18:     android:textStyle="bold" />
19:
20:   <Button
21:     android:layout_width="match_parent"
22:     android:layout_height="0dp"
23:     android:layout_weight="1"
24:     android:background="@drawable/mybutton_layout"
25:     android:gravity="center"
26:     android:onClick="exploreProposals"
27:     android:text="@string/explore_proposals"
28:     android:textStyle="bold" />
29: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="match_parent"
3:     android:layout_height="match_parent"
4:     android:background="@drawable/lightblue_border"
5:     android:orientation="vertical" >
6:
7:     <LinearLayout
8:         android:layout_width="match_parent"
9:         android:layout_height="wrap_content"
10:        android:orientation="vertical" >
11:
12:        <TextView
13:            android:layout_width="fill_parent"
14:            android:layout_height="wrap_content"
15:            android:layout_marginTop="5dp"
16:            android:gravity="center"
17:            android:text="@string/title_expenditures"
18:            android:textStyle="bold" />
19:
20:        <LinearLayout
21:            android:id="@+id/ulpgc_layout"
22:            android:layout_width="match_parent"
23:            android:layout_height="wrap_content"
24:            android:background="@drawable/blue_border"
25:            android:orientation="vertical" >
26:
27:            <TextView
28:                android:layout_width="fill_parent"
29:                android:layout_height="wrap_content"
30:                android:layout_marginTop="5dp"
31:                android:gravity="left"
32:                android:text="@string/title_ULPGCteacherExpenditures"
33:                android:textStyle="bold" />
34:
35:            <include
36:                android:id="@+id/teory_ulpgc"
37:                layout="@layout/three_items_row" />
38:
39:            <include
40:                android:id="@+id/practice_ulpgc"
41:                layout="@layout/three_items_row" />
42:
43:            <include
44:                android:id="@+id/seminars_ulpgc"
45:                layout="@layout/three_items_row" />
46:
47:            <include
48:                android:id="@+id/exams_ulpgc"
49:                layout="@layout/three_items_row" />
50:
51:            <include
52:                android:id="@+id/content_ulpgc"
53:                layout="@layout/three_items_row" />
54:
55:            <include
56:                android:id="@+id/tutorship_ulpgc"
57:                layout="@layout/three_items_row" />
58:
59:            <include
60:                android:id="@+id/final_project_ulpgc"
61:                layout="@layout/three_items_row" />
62:
63:            <include
```

```
64:         android:id="@+id/exam_coordinator_ulpgc"
65:         layout="@layout/three_items_row" />
66:
67:     <include
68:         android:id="@+id/directors"
69:         layout="@layout/two_items_4_tol" />
70:
71:     <include
72:         android:id="@+id/cost_sum"
73:         layout="@layout/two_items_4_tol" />
74:
75:     <include
76:         android:id="@+id/unit_profit"
77:         layout="@layout/two_items_4_tol" />
78:
79:     <include
80:         android:id="@+id/cost_total"
81:         layout="@layout/two_items_4_tol" />
82: </LinearLayout>
83:
84: <LinearLayout
85:     android:id="@+id/external_layout"
86:     android:layout_width="match_parent"
87:     android:layout_height="wrap_content"
88:     android:background="@drawable/blue_border"
89:     android:orientation="vertical" >
90:
91:     <TextView
92:         android:layout_width="fill_parent"
93:         android:layout_height="wrap_content"
94:         android:layout_marginTop="5dp"
95:         android:gravity="left"
96:         android:text="@string/title_ExternalteacherExpenditures"
97:         android:textStyle="bold" />
98:
99:     <include
100:         android:id="@+id/teory"
101:         layout="@layout/three_items_row" />
102:
103:     <include
104:         android:id="@+id/practice"
105:         layout="@layout/three_items_row" />
106:
107:     <include
108:         android:id="@+id/seminars"
109:         layout="@layout/three_items_row" />
110:
111:     <include
112:         android:id="@+id/exams"
113:         layout="@layout/three_items_row" />
114:
115:     <include
116:         android:id="@+id/content"
117:         layout="@layout/three_items_row" />
118:
119:     <include
120:         android:id="@+id/tutorship"
121:         layout="@layout/three_items_row" />
122:
123:     <include
124:         android:id="@+id/final_project"
125:         layout="@layout/three_items_row" />
126:
```

```
127:         <include
128:             android:id="@+id/exam_coordinator"
129:             layout="@layout/three_items_row" />
130:
131:         <include
132:             android:id="@+id/others_external"
133:             layout="@layout/two_items_4_tol" />
134:
135:         <include
136:             android:id="@+id/external_cost_total"
137:             layout="@layout/two_items_4_tol" />
138:     </LinearLayout>
139:
140: <LinearLayout
141:     android:id="@+id/admin_costs"
142:     android:layout_width="match_parent"
143:     android:layout_height="wrap_content"
144:     android:background="@drawable/blue_border"
145:     android:orientation="vertical" >
146:
147:     <TextView
148:         android:layout_width="fill_parent"
149:         android:layout_height="wrap_content"
150:         android:layout_marginTop="5dp"
151:         android:gravity="left"
152:         android:text="@string/admin_support"
153:         android:textStyle="bold" />
154:
155:     <include
156:         android:id="@+id/staff"
157:         layout="@layout/three_items_row" />
158:
159:     <include
160:         android:id="@+id/scholarship_students"
161:         layout="@layout/three_items_row" />
162:
163:     <include
164:         android:id="@+id/scholarship_specific"
165:         layout="@layout/three_items_row" />
166:
167:     <include
168:         android:id="@+id/other_staff"
169:         layout="@layout/three_items_row" />
170:
171:     <include
172:         android:id="@+id/presentation_report"
173:         layout="@layout/two_items_4_tol" />
174:
175:     <include
176:         android:id="@+id/final_report"
177:         layout="@layout/two_items_4_tol" />
178:
179:     <include
180:         android:id="@+id/information_point"
181:         layout="@layout/two_items_4_tol" />
182:
183:     <include
184:         android:id="@+id/support"
185:         layout="@layout/two_items_4_tol" />
186:
187:     <include
188:         android:id="@+id/other_support"
189:         layout="@layout/two_items_4_tol" />
```

```
190:         </LinearLayout>
191:
192:         <LinearLayout
193:             android:id="@+id/travel_n_accomodation"
194:             android:layout_width="match_parent"
195:             android:layout_height="wrap_content"
196:             android:background="@drawable/blue_border"
197:             android:orientation="vertical" >
198:
199:             <TextView
200:                 android:layout_width="fill_parent"
201:                 android:layout_height="wrap_content"
202:                 android:layout_marginTop="5dp"
203:                 android:gravity="left"
204:                 android:text="@string/travel_accomodation"
205:                 android:textStyle="bold" />
206:
207:             <include
208:                 android:id="@+id/travel_cost"
209:                 layout="@layout/two_items_4_tol" />
210:
211:             <include
212:                 android:id="@+id/hosting_cost"
213:                 layout="@layout/two_items_4_tol" />
214:
215:             <include
216:                 android:id="@+id/food_cost"
217:                 layout="@layout/two_items_4_tol" />
218:
219:             <include
220:                 android:id="@+id/scholarship_grant"
221:                 layout="@layout/two_items_4_tol" />
222:
223:             <include
224:                 android:id="@+id/other_travelcost"
225:                 layout="@layout/two_items_4_tol" />
226:
227:             <include
228:                 android:id="@+id/travel_total"
229:                 layout="@layout/two_items_4_tol" />
230:         </LinearLayout>
231:
232:         <LinearLayout
233:             android:id="@+id/materials"
234:             android:layout_width="match_parent"
235:             android:layout_height="wrap_content"
236:             android:background="@drawable/blue_border"
237:             android:orientation="vertical" >
238:
239:             <TextView
240:                 android:layout_width="fill_parent"
241:                 android:layout_height="wrap_content"
242:                 android:layout_marginTop="5dp"
243:                 android:gravity="left"
244:                 android:text="@string/fungible"
245:                 android:textStyle="bold" />
246:
247:             <include
248:                 android:id="@+id/office_material"
249:                 layout="@layout/two_items_4_tol" />
250:
251:             <include
252:                 android:id="@+id/lab_material"
```

```
253:         layout="@layout/two_items_4_tol" />
254:
255:     <include
256:         android:id="@+id/non_inventory"
257:         layout="@layout/two_items_4_tol" />
258:
259:     <include
260:         android:id="@+id/reprography"
261:         layout="@layout/two_items_4_tol" />
262:
263:     <include
264:         android:id="@+id/other_material"
265:         layout="@layout/two_items_4_tol" />
266:
267:     <include
268:         android:id="@+id/fungible_total"
269:         layout="@layout/two_items_4_tol" />
270: </LinearLayout>
271:
272: <LinearLayout
273:     android:id="@+id/bibliog_costs"
274:     android:layout_width="match_parent"
275:     android:layout_height="wrap_content"
276:     android:background="@drawable/blue_border"
277:     android:orientation="vertical" >
278:
279:     <TextView
280:         android:layout_width="fill_parent"
281:         android:layout_height="wrap_content"
282:         android:layout_marginTop="5dp"
283:         android:gravity="left"
284:         android:text="@string/bibliography"
285:         android:textStyle="bold" />
286:
287:     <include
288:         android:id="@+id/bibliography_cost"
289:         layout="@layout/two_items_4_tol" />
290:
291:     <include
292:         android:id="@+id/other_bibliography"
293:         layout="@layout/two_items_4_tol" />
294:
295:     <include
296:         android:id="@+id/bibliography_total"
297:         layout="@layout/two_items_4_tol" />
298: </LinearLayout>
299:
300: <LinearLayout
301:     android:id="@+id/press_costs"
302:     android:layout_width="match_parent"
303:     android:layout_height="wrap_content"
304:     android:background="@drawable/blue_border"
305:     android:orientation="vertical" >
306:
307:     <TextView
308:         android:layout_width="fill_parent"
309:         android:layout_height="wrap_content"
310:         android:layout_marginTop="5dp"
311:         android:gravity="left"
312:         android:text="@string/ads"
313:         android:textStyle="bold" />
314:
315:     <include
```

```
316:         android:id="@+id/print_ads"
317:         layout="@layout/two_items_4_tol" />
318:
319:     <include
320:         android:id="@+id/press_ads"
321:         layout="@layout/two_items_4_tol" />
322:
323:     <include
324:         android:id="@+id/ads_total"
325:         layout="@layout/two_items_4_tol" />
326: </LinearLayout>
327:
328: <LinearLayout
329:     android:id="@+id/other_costs"
330:     android:layout_width="match_parent"
331:     android:layout_height="wrap_content"
332:     android:background="@drawable/blue_border"
333:     android:orientation="vertical" >
334:
335:     <TextView
336:         android:layout_width="fill_parent"
337:         android:layout_height="wrap_content"
338:         android:layout_marginTop="5dp"
339:         android:gravity="left"
340:         android:text="@string/other_expenditures"
341:         android:textStyle="bold" />
342:
343:     <include
344:         android:id="@+id/inauguration"
345:         layout="@layout/two_items_4_tol" />
346:
347:     <include
348:         android:id="@+id/translation"
349:         layout="@layout/two_items_4_tol" />
350:
351:     <include
352:         android:id="@+id/certs"
353:         layout="@layout/two_items_4_tol" />
354:
355:     <include
356:         android:id="@+id/virtual_campus"
357:         layout="@layout/two_items_4_tol" />
358:
359:     <include
360:         android:id="@+id/evaluation"
361:         layout="@layout/two_items_4_tol" />
362:
363:     <include
364:         android:id="@+id/insurance"
365:         layout="@layout/two_items_4_tol" />
366:
367:     <include
368:         android:id="@+id/other_other"
369:         layout="@layout/two_items_4_tol" />
370:
371:     <include
372:         android:id="@+id/others_total"
373:         layout="@layout/two_items_4_tol" />
374: </LinearLayout>
375:
376: <LinearLayout
377:     android:layout_width="match_parent"
378:     android:layout_height="wrap_content"
```

```
379:         android:background="@drawable/blue_border"
380:         android:orientation="vertical" >
381:
382:         <TextView
383:             android:layout_width="fill_parent"
384:             android:layout_height="wrap_content"
385:             android:layout_marginTop="5dp"
386:             android:gravity="left"
387:             android:text="@string/total_expenses"
388:             android:textStyle="bold" />
389:
390:         <include
391:             android:id="@+id/total_expenses"
392:             layout="@layout/two_items_4_tol" />
393:     </LinearLayout>
394: </LinearLayout>
395:
396: </ScrollView>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent"
5:     android:background="@drawable/lightblue_border" >
6:
7:     <fragment
8:         android:id="@+id/search_results_fragment"
9:         android:layout_width="match_parent"
10:        android:layout_height="match_parent"
11:        class="es.ulpgc.titulospropios.gestion.ubay.SearchResultsFragment" />
12:
13: </FrameLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/lightblue_border"
6:   android:orientation="vertical"
7:   tools:context=".SingleDegreeMain" >
8:
9:   <Button
10:     android:layout_width="match_parent"
11:     android:layout_height="0dp"
12:     android:layout_marginTop="5dp"
13:     android:layout_weight="1"
14:     android:background="@drawable/mybutton_layout"
15:     android:gravity="center"
16:     android:onClick="overview"
17:     android:text="@string/overviewTab"
18:     android:textStyle="bold" />
19:
20:   <Button
21:     android:layout_width="match_parent"
22:     android:layout_height="0dp"
23:     android:layout_weight="1"
24:     android:background="@drawable/mybutton_layout"
25:     android:gravity="center"
26:     android:onClick="income"
27:     android:text="@string/incomeTab"
28:     android:textStyle="bold" />
29:
30:   <Button
31:     android:layout_width="match_parent"
32:     android:layout_height="0dp"
33:     android:layout_weight="1"
34:     android:background="@drawable/mybutton_layout"
35:     android:gravity="center"
36:     android:onClick="expenditure"
37:     android:text="@string/expensesTab"
38:     android:textStyle="bold" />
39:
40: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="wrap_content"
5:     android:background="@drawable/blue_border"
6:     android:baselineAligned="false"
7:     android:orientation="horizontal" >
8:
9:     <include
10:         android:id="@+id/left_item"
11:         android:layout_weight="90"
12:         layout="@layout/info_cell"
13:         android:gravity="center" />
14:
15:     <include
16:         android:layout_weight="1"
17:         layout="@layout/divider" />
18:
19:     <include
20:         android:id="@+id/right_item"
21:         android:layout_weight="90"
22:         layout="@layout/info_cell"
23:         android:gravity="center" />
24:
25: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:id="@+id/layout_root"
3:     android:layout_width="fill_parent"
4:     android:layout_height="fill_parent"
5:     android:orientation="horizontal"
6:     android:padding="10dp" >
7:
8:     <ImageView
9:         android:id="@+id/image"
10:        android:layout_width="wrap_content"
11:        android:layout_height="fill_parent"
12:        android:layout_marginRight="10dp" />
13:
14:     <TextView
15:         android:id="@+id/text"
16:         android:layout_width="wrap_content"
17:         android:layout_height="fill_parent"
18:         android:textColor="#FFF" />
19:
20: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/background_colours" >
5:
6:     <LinearLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content"
9:         android:orientation="vertical"
10:        android:weightSum="16" >
11:
12:        <TextView
13:            android:id="@+id/period"
14:            android:layout_width="fill_parent"
15:            android:layout_height="wrap_content"
16:            android:layout_marginTop="5dp"
17:            android:background="@drawable/blue_border"
18:            android:gravity="center"
19:            android:textStyle="bold" />
20:
21:        <EditText
22:            android:id="@+id/degree_denomination"
23:            android:layout_width="fill_parent"
24:            android:layout_height="0dp"
25:            android:layout_marginTop="@dimen/activity_horizontal_margin"
26:            android:layout_weight="1"
27:            android:hint="@string/degree_denomination" />
28:
29:        <Spinner
30:            android:id="@+id/type_spinner"
31:            android:layout_width="fill_parent"
32:            android:layout_height="0dp"
33:            android:layout_weight="1"
34:            android:background="@drawable/grey_buttons"
35:            android:prompt="@string/tipo" />
36:
37:        <Spinner
38:            android:id="@+id/mode_spinner"
39:            android:layout_width="fill_parent"
40:            android:layout_height="0dp"
41:            android:layout_weight="1"
42:            android:background="@drawable/grey_buttons"
43:            android:prompt="@string/mode" />
44:
45:        <Spinner
46:            android:id="@+id/oriented_spinner"
47:            android:layout_width="fill_parent"
48:            android:layout_height="0dp"
49:            android:layout_weight="1"
50:            android:background="@drawable/grey_buttons"
51:            android:prompt="@string/oriented" />
52:
53:        <Spinner
54:            android:id="@+id/knowledge_spinner"
55:            android:layout_width="fill_parent"
56:            android:layout_height="0dp"
57:            android:layout_weight="1"
58:            android:background="@drawable/grey_buttons"
59:            android:prompt="@string/knowledge_branch" />
60:
61:        <EditText
62:            android:id="@+id/degree_academic_director"
63:            android:layout_width="fill_parent"
```

```
64:         android:layout_height="0dp"
65:         android:layout_width="1"
66:         android:hint="@string/degree_academic_director"
67:         android:imeOptions="actionNext"
68:         android:singleLine="true" >
69:     </EditText>
70:
71:     <EditText
72:         android:id="@+id/degree_proposed_by"
73:         android:layout_width="fill_parent"
74:         android:layout_height="0dp"
75:         android:layout_width="1"
76:         android:hint="@string/degree_proposed_by"
77:         android:imeOptions="actionNext"
78:         android:singleLine="true" />
79:
80:     <EditText
81:         android:id="@+id/degree_units"
82:         android:layout_width="fill_parent"
83:         android:layout_height="0dp"
84:         android:layout_width="1"
85:         android:hint="@string/degree_units"
86:         android:imeOptions="actionNext"
87:         android:singleLine="true" />
88:
89:     <EditText
90:         android:id="@+id/degree_min_alumns"
91:         android:layout_width="fill_parent"
92:         android:layout_height="0dp"
93:         android:layout_width="1"
94:         android:inputType="number"
95:         android:hint="@string/degree_min_alumns"
96:         android:imeOptions="actionNext"
97:         android:singleLine="true" />
98:
99:     <EditText
100:         android:id="@+id/degree_max_alumns"
101:         android:layout_width="fill_parent"
102:         android:layout_height="0dp"
103:         android:layout_width="1"
104:         android:inputType="number"
105:         android:hint="@string/degree_max_alumns"
106:         android:imeOptions="actionNext"
107:         android:singleLine="true" />
108:
109:     <EditText
110:         android:id="@+id/degree_min_credits"
111:         android:layout_width="fill_parent"
112:         android:layout_height="0dp"
113:         android:layout_width="1"
114:         android:inputType="number"
115:         android:hint="@string/degree_min_credits"
116:         android:imeOptions="actionNext"
117:         android:singleLine="true" />
118:
119:     <EditText
120:         android:id="@+id/degree_economic_manager"
121:         android:layout_width="fill_parent"
122:         android:layout_height="0dp"
123:         android:layout_width="1"
124:         android:hint="@string/degree_economic_manager"
125:         android:imeOptions="actionNext"
126:         android:singleLine="true" />
```

```
127:
128:     <EditText
129:         android:id="@+id/degree_competences"
130:         android:layout_width="fill_parent"
131:         android:layout_height="0dp"
132:         android:layout_weight="3"
133:         android:hint="@string/degree_competences"
134:         android:imeOptions="actionNext"
135:         android:inputType="textCapSentences|textMultiLine" />
136:
137:     <TextView
138:         android:layout_width="fill_parent"
139:         android:layout_height="wrap_content"
140:         android:layout_marginTop="5dp"
141:         android:gravity="left"
142:         android:text="@string/institutions"
143:         android:textStyle="bold" />
144:
145:     <Spinner
146:         android:id="@+id/institutions_spinner"
147:         android:layout_width="fill_parent"
148:         android:layout_height="0dp"
149:         android:layout_weight="1"
150:         android:background="@drawable/grey_buttons"
151:         android:prompt="@string/institutions" />
152:
153:     <TextView
154:         android:layout_width="fill_parent"
155:         android:layout_height="wrap_content"
156:         android:layout_marginTop="5dp"
157:         android:gravity="left"
158:         android:text="@string/teaching_address"
159:         android:textStyle="bold" />
160:
161:     <EditText
162:         android:id="@+id/teaching_address"
163:         android:layout_width="fill_parent"
164:         android:layout_height="0dp"
165:         android:layout_weight="3"
166:         android:hint="@string/teaching_address"
167:         android:imeOptions="actionDone"
168:         android:inputType="textCapSentences|textMultiLine" />
169:
170:     <TextView
171:         android:layout_width="fill_parent"
172:         android:layout_height="wrap_content"
173:         android:layout_marginTop="5dp"
174:         android:gravity="left"
175:         android:text="@string/teaching_outside"
176:         android:textStyle="bold" />
177:
178:     <CheckBox
179:         android:id="@+id/teaching_outside"
180:         android:layout_width="wrap_content"
181:         android:layout_height="wrap_content" />
182:
183:     <Button
184:         android:id="@+id/send_button"
185:         android:layout_width="fill_parent"
186:         android:layout_height="0dp"
187:         android:layout_marginTop="@dimen/activity_horizontal_margin"
188:         android:layout_weight="1"
189:         android:background="@drawable/grey_buttons"
```

```
190:         android:onClick="sendNewProposal"
191:         android:text="@string/send"
192:         android:textSize="24sp" />
193:     </LinearLayout>
194:
195: </ScrollView>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="fill_parent"
4:     android:layout_height="fill_parent"
5:     android:orientation="vertical" >
6:
7:     <TabHost
8:         android:id="@android:id/tabhost"
9:         android:layout_width="fill_parent"
10:        android:layout_height="fill_parent" >
11:
12:         <LinearLayout
13:             android:layout_width="fill_parent"
14:             android:layout_height="fill_parent"
15:             android:orientation="vertical" >
16:
17:             <TabWidget
18:                 android:id="@android:id/tabs"
19:                 android:layout_width="fill_parent"
20:                 android:layout_height="wrap_content"
21:                 android:layout_weight="0"
22:                 android:orientation="horizontal" />
23:
24:             <FrameLayout
25:                 android:id="@android:id/tabcontent"
26:                 android:layout_width="0dp"
27:                 android:layout_height="0dp"
28:                 android:layout_weight="0" />
29:
30:             <FrameLayout
31:                 android:id="@+android:id/realtabcontent"
32:                 android:layout_width="fill_parent"
33:                 android:layout_height="0dp"
34:                 android:layout_weight="1" />
35:         </LinearLayout>
36:     </TabHost>
37:
38: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/lightblue_border" >
5:
6:     <LinearLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content"
9:         android:orientation="vertical" >
10:
11:         <TextView
12:             android:layout_width="match_parent"
13:             android:layout_height="wrap_content"
14:             android:gravity="center"
15:             android:text="@string/errors_activity"
16:             android:textStyle="bold" />
17:
18:         <LinearLayout
19:             android:id="@+id/base_issues_layout"
20:             android:layout_width="match_parent"
21:             android:layout_height="wrap_content"
22:             android:orientation="vertical" >
23:             </LinearLayout>
24:         </LinearLayout>
25:
26: </ScrollView>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:orientation="vertical"
5:     android:weightSum="3" >
6:
7:     <LinearLayout
8:         android:layout_width="match_parent"
9:         android:layout_height="0dp"
10:        android:layout_weight="1"
11:        android:orientation="vertical" >
12:
13:        <ImageView
14:            android:layout_width="match_parent"
15:            android:layout_height="match_parent"
16:            android:contentDescription="@string/desc"
17:            android:src="@drawable/logo_vtd_rgb" />
18:    </LinearLayout>
19:
20:    <LinearLayout
21:        android:layout_width="match_parent"
22:        android:layout_height="0dp"
23:        android:layout_weight="2"
24:        android:background="@drawable/invert_gradient_background_noborder"
25:        android:orientation="vertical" >
26:
27:        <TextView
28:            android:id="@+id/login_manager"
29:            android:layout_width="wrap_content"
30:            android:layout_height="wrap_content"
31:            android:layout_gravity="center" />
32:
33:        <EditText
34:            android:id="@+id/login_message"
35:            android:layout_width="fill_parent"
36:            android:layout_height="wrap_content"
37:            android:hint="@string/login_string" />
38:
39:        <EditText
40:            android:id="@+id/pass_message"
41:            android:layout_width="fill_parent"
42:            android:layout_height="wrap_content"
43:            android:hint="@string/pass_string"
44:            android:inputType="textPassword" />
45:
46:        <CheckBox
47:            android:id="@+id/rememberButton"
48:            android:layout_width="wrap_content"
49:            android:layout_height="wrap_content"
50:            android:text="@string/rememberMe_message" />
51:
52:        <RelativeLayout
53:            android:layout_width="match_parent"
54:            android:layout_height="fill_parent" >
55:
56:            <Button
57:                android:id="@+id/loginButton"
58:                android:layout_width="100dp"
59:                android:layout_height="60dp"
60:                android:layout_centerHorizontal="true"
61:                android:layout_centerVertical="true"
62:                android:background="@drawable/mybutton_layout"
63:                android:text="@string/button_login" />
```

```
64:  
65:         </RelativeLayout>  
66:     </LinearLayout>  
67:  
68: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent" >
5:
6:     <fragment
7:         android:id="@+id/teachers_list_fragment"
8:         android:layout_width="match_parent"
9:         android:layout_height="match_parent"
10:        class="es.ulpgc.titulospropios.gestion.ubay.TeachersListFragment" />
11:
12: </FrameLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent"
5:     android:background="@drawable/background_colours"
6:     android:orientation="vertical" >
7:
8:     <Button
9:         android:id="@+id/button_degree"
10:        android:layout_width="match_parent"
11:        android:layout_height="wrap_content"
12:        android:layout_marginTop="5dp"
13:        android:background="@drawable/mybutton_layout" />
14:
15:     <LinearLayout
16:         android:layout_width="match_parent"
17:         android:layout_height="wrap_content"
18:         android:background="@drawable/blue_border"
19:         android:orientation="horizontal" >
20:
21:         <TextView
22:             android:id="@+id/textView_type"
23:             android:layout_width="0dp"
24:             android:layout_height="wrap_content"
25:             android:layout_weight="90" />
26:
27:         <View
28:             android:id="@+id/firstDivider"
29:             android:layout_width="1dp"
30:             android:layout_height="fill_parent"
31:             android:layout_weight="1"
32:             android:background="#3A4978" />
33:
34:         <TextView
35:             android:id="@+id/textView_state"
36:             android:layout_width="0dp"
37:             android:layout_height="wrap_content"
38:             android:layout_weight="90" />
39:     </LinearLayout>
40:
41:     <LinearLayout
42:         android:layout_width="match_parent"
43:         android:layout_height="wrap_content"
44:         android:background="@drawable/blue_border"
45:         android:orientation="horizontal" >
46:
47:         <TextView
48:             android:id="@+id/textView_edition"
49:             android:layout_width="0dp"
50:             android:layout_height="wrap_content"
51:             android:layout_weight="90" />
52:
53:         <View
54:             android:id="@+id/secondDivider"
55:             android:layout_width="1dp"
56:             android:layout_height="fill_parent"
57:             android:layout_weight="1"
58:             android:background="#3A4978" />
59:
60:         <TextView
61:             android:id="@+id/textView_code"
62:             android:layout_width="0dp"
63:             android:layout_height="wrap_content"
```

```
64:         android:layout_weight="90" />
65:     </LinearLayout>
66:
67: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     xmlns:tools="http://schemas.android.com/tools"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent"
5:     android:orientation="vertical" >
6:
7:     <TextView
8:         android:layout_width="fill_parent"
9:         android:layout_height="fill_parent"
10:        android:gravity="center"
11:        android:text="@string/empty"
12:        android:textSize="@dimen/activity_horizontal_margin"
13:        android:textStyle="bold" />
14:
15: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <TextView xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="0dp"
4:     android:layout_height="wrap_content"
5:     android:layout_weight="90" />
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent">
5:     <fragment class="es.ulpgc.titulosproprios.gestion.ubay.SearchDegreesFragment"
6:         android:id="@+id/search_degrees_fragment"
7:         android:layout_width="match_parent"
8:         android:layout_height="match_parent"/>
9: </FrameLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent" >
5:
6:     <fragment
7:         android:id="@+id/single_teacher_fragment"
8:         android:layout_width="match_parent"
9:         android:layout_height="match_parent"
10:         class="es.ulpgc.titulospropios.gestion.ubay.SingleTeacherFragment" />
11:
12: </FrameLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent" >
5:
6:     <fragment
7:         android:id="@+id/subjects_list_fragment"
8:         android:layout_width="match_parent"
9:         android:layout_height="match_parent"
10:         class="es.ulpgc.titulospropios.gestion.ubay.SubjectsListFragment" />
11:
12: </FrameLayout>
```

```
1: <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="wrap_content"
3:     android:layout_height="wrap_content"
4:     android:background="@drawable/blue_border" >
5:
6:     <ProgressBar
7:         android:id="@+id/progressBar1"
8:         style="?android:attr/progressBarStyleHorizontal"
9:         android:layout_width="wrap_content"
10:        android:layout_height="wrap_content"
11:        android:layout_centerHorizontal="true"
12:        android:layout_centerVertical="true" />
13:
14:    <TextView
15:        android:id="@+id/textView1"
16:        android:layout_width="wrap_content"
17:        android:layout_height="wrap_content"
18:        android:layout_above="@+id/progressBar1"
19:        android:layout_centerHorizontal="true"
20:        android:layout_marginBottom="30dp"
21:        android:text="@string/progressDownload"
22:        android:textAppearance="?android:attr/textAppearanceMedium" />
23:
24:    <TextView
25:        android:id="@+id/textView2"
26:        android:layout_width="wrap_content"
27:        android:layout_height="wrap_content"
28:        android:layout_below="@+id/progressBar1"
29:        android:layout_centerHorizontal="true"
30:        android:layout_marginTop="26dp" />
31:
32: </RelativeLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent" >
5:
6:     <fragment
7:         android:id="@+id/reports_info_fragment"
8:         android:layout_width="match_parent"
9:         android:layout_height="match_parent"
10:        class="es.ulpgc.titulospropios.gestion.ubay.ReportsInfoFragment" />
11:
12: </FrameLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/background_colours"
6:   android:orientation="vertical"
7:   tools:context=".SingleDegreeMain" >
8:
9:   <Button
10:    android:layout_width="match_parent"
11:    android:layout_height="0dp"
12:    android:layout_marginTop="5dp"
13:    android:layout_weight="1"
14:    android:background="@drawable/mybutton_layout"
15:    android:gravity="center"
16:    android:onClick="academicInfo"
17:    android:text="@string/academic_info"
18:    android:textStyle="bold" />
19:
20:   <Button
21:    android:layout_width="match_parent"
22:    android:layout_height="0dp"
23:    android:layout_weight="1"
24:    android:background="@drawable/mybutton_layout"
25:    android:gravity="center"
26:    android:onClick="subjectsInfo"
27:    android:text="@string/subjects"
28:    android:textStyle="bold" />
29:
30:   <Button
31:    android:layout_width="match_parent"
32:    android:layout_height="0dp"
33:    android:layout_weight="1"
34:    android:background="@drawable/mybutton_layout"
35:    android:gravity="center"
36:    android:onClick="teachersInfo"
37:    android:text="@string/teachers"
38:    android:textStyle="bold" />
39:
40:   <Button
41:    android:layout_width="match_parent"
42:    android:layout_height="0dp"
43:    android:layout_weight="1"
44:    android:background="@drawable/mybutton_layout"
45:    android:gravity="center"
46:    android:onClick="economicInfo"
47:    android:text="@string/economic_info"
48:    android:textStyle="bold" />
49:
50:   <Button
51:    android:layout_width="match_parent"
52:    android:layout_height="0dp"
53:    android:layout_weight="1"
54:    android:background="@drawable/mybutton_layout"
55:    android:gravity="center"
56:    android:onClick="reportInfo"
57:    android:text="@string/report"
58:    android:textStyle="bold" />
59:
60:   <Button
61:    android:layout_width="match_parent"
62:    android:layout_height="0dp"
63:    android:layout_weight="1"
```

```
64:         android:background="@drawable/mybutton_layout"
65:         android:gravity="center"
66:         android:onClick="sendDegreeToEvaluate"
67:         android:text="@string/send_to_evaluate"
68:         android:textStyle="bold" />
69:
70:     <Button
71:         android:layout_width="match_parent"
72:         android:layout_height="0dp"
73:         android:layout_weight="1"
74:         android:background="@drawable/mybutton_layout"
75:         android:gravity="center"
76:         android:onClick="downloadDegree"
77:         android:text="@string/download_degree"
78:         android:textStyle="bold" />
79: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/background_colours" >
5:
6:     <RelativeLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content" >
9:
10:        <TextView
11:            android:id="@+id/name"
12:            android:layout_width="wrap_content"
13:            android:layout_height="wrap_content"
14:            android:layout_alignParentLeft="true"
15:            android:layout_alignParentRight="true"
16:            android:layout_alignParentTop="true"
17:            android:background="@drawable/blue_border"
18:            android:gravity="center"
19:            android:textStyle="bold" />
20:
21:        <LinearLayout
22:            android:id="@+id/linearLayout0"
23:            android:layout_width="wrap_content"
24:            android:layout_height="wrap_content"
25:            android:layout_below="@+id/name"
26:            android:orientation="horizontal" >
27:
28:            <TextView
29:                android:layout_width="0dp"
30:                android:layout_height="match_parent"
31:                android:layout_weight="1"
32:                android:text="@string/title_generalInfo"
33:                android:textStyle="bold" />
34:        </LinearLayout>
35:
36:        <LinearLayout
37:            android:id="@+id/linearLayout1"
38:            android:layout_width="match_parent"
39:            android:layout_height="wrap_content"
40:            android:layout_below="@+id/linearLayout0"
41:            android:background="@drawable/blue_border"
42:            android:orientation="horizontal" >
43:
44:            <TextView
45:                android:id="@+id/branch"
46:                android:layout_width="0dp"
47:                android:layout_height="match_parent"
48:                android:layout_weight="90" />
49:
50:            <View
51:                android:layout_width="1dp"
52:                android:layout_height="fill_parent"
53:                android:layout_weight="1"
54:                android:background="#3A4978" />
55:
56:            <TextView
57:                android:id="@+id/teaching_mode"
58:                android:layout_width="0dp"
59:                android:layout_height="match_parent"
60:                android:layout_weight="90" />
61:        </LinearLayout>
62:
63:        <LinearLayout
```

```
64:         android:id="@+id/linearLayout2"
65:         android:layout_width="match_parent"
66:         android:layout_height="wrap_content"
67:         android:layout_below="@+id/linearLayout1"
68:         android:background="@drawable/blue_border"
69:         android:orientation="horizontal" >
70:
71:         <TextView
72:             android:id="@+id/state"
73:             android:layout_width="0dp"
74:             android:layout_height="match_parent"
75:             android:layout_weight="90" />
76:
77:         <View
78:             android:layout_width="1dp"
79:             android:layout_height="fill_parent"
80:             android:layout_weight="1"
81:             android:background="#3A4978" />
82:
83:         <TextView
84:             android:id="@+id/edition"
85:             android:layout_width="0dp"
86:             android:layout_height="match_parent"
87:             android:layout_weight="90" />
88:     </LinearLayout>
89:
90:     <LinearLayout
91:         android:id="@+id/linearLayout3"
92:         android:layout_width="match_parent"
93:         android:layout_height="wrap_content"
94:         android:layout_below="@+id/linearLayout2"
95:         android:background="@drawable/blue_border"
96:         android:orientation="horizontal" >
97:
98:         <TextView
99:             android:id="@+id/start_date"
100:            android:layout_width="0dp"
101:            android:layout_height="match_parent"
102:            android:layout_weight="90" />
103:
104:         <View
105:             android:layout_width="1dp"
106:             android:layout_height="fill_parent"
107:             android:layout_weight="1"
108:             android:background="#3A4978" />
109:
110:         <TextView
111:             android:id="@+id/end_date"
112:             android:layout_width="0dp"
113:             android:layout_height="match_parent"
114:             android:layout_weight="90" />
115:     </LinearLayout>
116:
117:     <LinearLayout
118:         android:id="@+id/linearLayout4"
119:         android:layout_width="match_parent"
120:         android:layout_height="wrap_content"
121:         android:layout_below="@+id/linearLayout3"
122:         android:background="@drawable/blue_border"
123:         android:orientation="horizontal" >
124:
125:         <TextView
126:             android:id="@+id/startDate"
```

```
127:         android:layout_width="0dp"
128:         android:layout_height="match_parent"
129:         android:layout_weight="90" />
130:
131:     <View
132:         android:layout_width="1dp"
133:         android:layout_height="fill_parent"
134:         android:layout_weight="1"
135:         android:background="#3A4978" />
136:
137:     <TextView
138:         android:id="@+id/endDate"
139:         android:layout_width="0dp"
140:         android:layout_height="match_parent"
141:         android:layout_weight="90" />
142: </LinearLayout>
143:
144: <LinearLayout
145:     android:id="@+id/linearLayout5"
146:     android:layout_width="match_parent"
147:     android:layout_height="wrap_content"
148:     android:layout_below="@+id/linearLayout4"
149:     android:background="@drawable/blue_border"
150:     android:orientation="horizontal" >
151:
152:     <TextView
153:         android:id="@+id/units"
154:         android:layout_width="0dp"
155:         android:layout_height="match_parent"
156:         android:layout_weight="90" />
157:
158:     <View
159:         android:layout_width="1dp"
160:         android:layout_height="fill_parent"
161:         android:layout_weight="1"
162:         android:background="#3A4978" />
163:
164:     <TextView
165:         android:id="@+id/teachingPlace"
166:         android:layout_width="0dp"
167:         android:layout_height="match_parent"
168:         android:layout_weight="90" />
169: </LinearLayout>
170:
171: <LinearLayout
172:     android:id="@+id/linearLayout_"
173:     android:layout_width="wrap_content"
174:     android:layout_height="wrap_content"
175:     android:layout_below="@id/linearLayout5"
176:     android:orientation="horizontal" >
177:
178:     <TextView
179:         android:layout_width="0dp"
180:         android:layout_height="match_parent"
181:         android:layout_weight="1"
182:         android:text="@string/title_hourDistribution"
183:         android:textStyle="bold" />
184: </LinearLayout>
185:
186: <LinearLayout
187:     android:id="@+id/manager"
188:     android:layout_width="match_parent"
189:     android:layout_height="wrap_content"
```

```
190:         android:layout_below="@id/linearLayout_"
191:         android:background="@drawable/blue_border"
192:         android:orientation="horizontal" >
193:
194:         <TextView
195:             android:id="@+id/class_hours"
196:             android:layout_width="0dp"
197:             android:layout_height="match_parent"
198:             android:layout_weight="90" />
199:
200:         <View
201:             android:layout_width="1dp"
202:             android:layout_height="fill_parent"
203:             android:layout_weight="1"
204:             android:background="#3A4978" />
205:
206:         <TextView
207:             android:id="@+id/institutionRepresentative"
208:             android:layout_width="0dp"
209:             android:layout_height="match_parent"
210:             android:layout_weight="90" />
211:     </LinearLayout>
212:
213:     <LinearLayout
214:         android:id="@+id/linearLayout7"
215:         android:layout_width="match_parent"
216:         android:layout_height="wrap_content"
217:         android:layout_below="@+id/manager"
218:         android:background="@drawable/blue_border"
219:         android:orientation="horizontal" >
220:
221:         <TextView
222:             android:id="@+id/misc_activities_hours"
223:             android:layout_width="0dp"
224:             android:layout_height="match_parent"
225:             android:layout_weight="90" />
226:
227:         <View
228:             android:layout_width="1dp"
229:             android:layout_height="fill_parent"
230:             android:layout_weight="1"
231:             android:background="#3A4978" />
232:
233:         <TextView
234:             android:id="@+id/headquarters"
235:             android:layout_width="0dp"
236:             android:layout_height="match_parent"
237:             android:layout_weight="90" />
238:     </LinearLayout>
239:
240:     <LinearLayout
241:         android:id="@+id/linearLayout8"
242:         android:layout_width="match_parent"
243:         android:layout_height="wrap_content"
244:         android:layout_below="@+id/linearLayout7"
245:         android:background="@drawable/blue_border"
246:         android:orientation="horizontal" >
247:
248:         <TextView
249:             android:id="@+id/exams_hours"
250:             android:layout_width="0dp"
251:             android:layout_height="match_parent"
252:             android:layout_weight="90" />
```

```
253:
254:     <View
255:         android:layout_width="1dp"
256:         android:layout_height="fill_parent"
257:         android:layout_weight="1"
258:         android:background="#3A4978" />
259:
260:     <TextView
261:         android:id="@+id/final_work_hours"
262:         android:layout_width="0dp"
263:         android:layout_height="match_parent"
264:         android:layout_weight="90" />
265: </LinearLayout>
266:
267: <LinearLayout
268:     android:id="@+id/linearLayout9"
269:     android:layout_width="match_parent"
270:     android:layout_height="wrap_content"
271:     android:layout_below="@+id/linearLayout8"
272:     android:background="@drawable/blue_border"
273:     android:orientation="horizontal" >
274:
275:     <TextView
276:         android:id="@+id/studying_hours"
277:         android:layout_width="0dp"
278:         android:layout_height="match_parent"
279:         android:layout_weight="90" />
280:
281:     <View
282:         android:layout_width="1dp"
283:         android:layout_height="fill_parent"
284:         android:layout_weight="1"
285:         android:background="#3A4978" />
286:
287:     <TextView
288:         android:id="@+id/hours_sum"
289:         android:layout_width="0dp"
290:         android:layout_height="match_parent"
291:         android:layout_weight="90" />
292: </LinearLayout>
293:
294: <LinearLayout
295:     android:layout_width="match_parent"
296:     android:layout_height="match_parent"
297:     android:layout_below="@+id/linearLayout9"
298:     android:orientation="vertical"
299:     android:paddingTop="7dp" >
300:
301:     <TextView
302:         android:id="@+id/competences"
303:         android:layout_width="match_parent"
304:         android:layout_height="wrap_content"
305:         android:background="@drawable/blue_border" />
306:
307:     <TextView
308:         android:id="@+id/activities"
309:         android:layout_width="match_parent"
310:         android:layout_height="wrap_content"
311:         android:background="@drawable/blue_border" />
312:
313:     <TextView
314:         android:id="@+id/description"
315:         android:layout_width="match_parent"
```

```
316:             android:layout_height="wrap_content"
317:             android:background="@drawable/blue_border" />
318:
319:         <TextView
320:             android:id="@+id/assessment_system"
321:             android:layout_width="match_parent"
322:             android:layout_height="wrap_content"
323:             android:background="@drawable/blue_border" />
324:     </LinearLayout>
325: </RelativeLayout>
326:
327: </ScrollView>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <View xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="1dp"
4:     android:layout_height="fill_parent"
5:     android:layout_weight="1"
6:     android:background="#3A4978" />
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="match_parent">
5:     <fragment class="es.ulpgc.titulosproprios.gestion.ubay.AcademicInfoFragment"
6:         android:id="@+id/academic_info_fragment"
7:         android:layout_width="match_parent"
8:         android:layout_height="match_parent"/>
9: </FrameLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="fill_parent"
4:     android:layout_height="fill_parent"
5:     android:background="@drawable/background_colours" >
6:
7:     <LinearLayout
8:         android:id="@+id/base_teacherlayout"
9:         android:layout_width="match_parent"
10:        android:layout_height="wrap_content"
11:        android:orientation="vertical" >
12:
13:        <TextView
14:            android:id="@+id/teacher_name"
15:            android:layout_width="fill_parent"
16:            android:layout_height="wrap_content"
17:            android:layout_marginTop="5dp"
18:            android:background="@drawable/blue_border"
19:            android:gravity="center"
20:            android:textStyle="bold" />
21:
22:        <include
23:            android:id="@+id/first_row"
24:            layout="@layout/row_layout" />
25:
26:        <include
27:            android:id="@+id/second_row"
28:            layout="@layout/row_layout" />
29:
30:        <include
31:            android:id="@+id/third_row"
32:            layout="@layout/row_layout" />
33:
34:        <include
35:            android:id="@+id/fourth_row"
36:            layout="@layout/row_layout" />
37:
38:        <LinearLayout
39:            android:layout_width="match_parent"
40:            android:layout_height="match_parent"
41:            android:orientation="vertical" >
42:
43:            <TextView
44:                android:id="@+id/cv_textview"
45:                android:layout_width="match_parent"
46:                android:layout_height="wrap_content"
47:                android:background="@drawable/blue_border" />
48:        </LinearLayout>
49:    </LinearLayout>
50:
51: </ScrollView>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:id="@+id/base_layout"
4:     android:layout_width="match_parent"
5:     android:layout_height="wrap_content"
6:     android:orientation="vertical" >
7:
8:     <ListView
9:         android:id="@+id/subject_list"
10:        android:layout_width="wrap_content"
11:        android:layout_height="wrap_content" />
12:
13:     <Button
14:         android:id="@+id/amount_footer"
15:         android:layout_width="match_parent"
16:         android:layout_height="wrap_content"
17:         android:layout_alignParentBottom="true"
18:         android:background="@drawable/mybutton_layout"
19:         android:gravity="center" />
20:
21: </RelativeLayout>
```

```
1: <CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"  
2:   android:id="@android:id/text1"  
3:   style="?android:attr/spinnerDropDownItemStyle"  
4:   android:layout_width="match_parent"  
5:   android:layout_height="?android:attr/listPreferredItemHeight"  
6:   android:ellipsize="marquee"  
7:   android:singleLine="false" />
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="wrap_content"
5:     android:orientation="vertical" >
6:
7:     <ListView
8:         android:id="@+id/teachers_list"
9:         android:layout_width="match_parent"
10:        android:layout_height="wrap_content" />
11:
12:     <Button
13:         android:id="@+id/amount_footer"
14:         android:layout_width="match_parent"
15:         android:layout_height="wrap_content"
16:         android:layout_alignParentBottom="true"
17:         android:background="@drawable/mybutton_layout"
18:         android:gravity="center" />
19:
20: </RelativeLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     xmlns:tools="http://schemas.android.com/tools"
3:     android:layout_width="match_parent"
4:     android:layout_height="wrap_content"
5:     tools:context=".SearchResults" >
6:     <ListView
7:         android:id="@+id/list"
8:         android:layout_width="match_parent"
9:         android:layout_height="wrap_content" />
10: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/background_colours" >
5:
6:     <RelativeLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content" >
9:
10:        <TextView
11:            android:id="@+id/name"
12:            android:layout_width="wrap_content"
13:            android:layout_height="wrap_content"
14:            android:layout_alignParentLeft="true"
15:            android:layout_alignParentRight="true"
16:            android:layout_alignParentTop="true"
17:            android:layout_marginTop="5dp"
18:            android:background="@drawable/blue_border"
19:            android:gravity="center"
20:            android:textStyle="bold" />
21:
22:        <LinearLayout
23:            android:id="@+id/linearLayout1"
24:            android:layout_width="match_parent"
25:            android:layout_height="wrap_content"
26:            android:layout_below="@+id/name"
27:            android:background="@drawable/blue_border"
28:            android:orientation="horizontal" >
29:
30:            <TextView
31:                android:id="@+id/branch"
32:                android:layout_width="0dp"
33:                android:layout_height="match_parent"
34:                android:layout_weight="90" />
35:
36:            <View
37:                android:id="@+id/firstDivider"
38:                android:layout_width="1dp"
39:                android:layout_height="fill_parent"
40:                android:layout_weight="1"
41:                android:background="#3A4978" />
42:
43:            <TextView
44:                android:id="@+id/teaching_mode"
45:                android:layout_width="0dp"
46:                android:layout_height="match_parent"
47:                android:layout_weight="90" />
48:        </LinearLayout>
49:
50:        <LinearLayout
51:            android:id="@+id/linearLayout2"
52:            android:layout_width="match_parent"
53:            android:layout_height="wrap_content"
54:            android:layout_below="@+id/linearLayout1"
55:            android:background="@drawable/blue_border"
56:            android:orientation="horizontal" >
57:
58:            <TextView
59:                android:id="@+id/state"
60:                android:layout_width="0dp"
61:                android:layout_height="match_parent"
62:                android:layout_weight="90" />
63:
```

```
64:         <View
65:             android:id="@+id/secondDivider"
66:             android:layout_width="1dp"
67:             android:layout_height="fill_parent"
68:             android:layout_weight="1"
69:             android:background="#3A4978" />
70:
71:         <TextView
72:             android:id="@+id/edition"
73:             android:layout_width="0dp"
74:             android:layout_height="match_parent"
75:             android:layout_weight="90" />
76:     </LinearLayout>
77:
78:     <LinearLayout
79:         android:id="@+id/linearLayout3"
80:         android:layout_width="match_parent"
81:         android:layout_height="wrap_content"
82:         android:layout_below="@+id/linearLayout2"
83:         android:background="@drawable/blue_border"
84:         android:orientation="horizontal" >
85:
86:         <TextView
87:             android:id="@+id/code"
88:             android:layout_width="0dp"
89:             android:layout_height="match_parent"
90:             android:layout_weight="90" />
91:
92:         <View
93:             android:id="@+id/thirdDivider"
94:             android:layout_width="1dp"
95:             android:layout_height="fill_parent"
96:             android:layout_weight="1"
97:             android:background="#3A4978" />
98:
99:         <TextView
100:             android:id="@+id/proposer"
101:             android:layout_width="0dp"
102:             android:layout_height="match_parent"
103:             android:layout_weight="90" />
104:     </LinearLayout>
105:
106:     <LinearLayout
107:         android:id="@+id/linearLayout4"
108:         android:layout_width="match_parent"
109:         android:layout_height="wrap_content"
110:         android:layout_below="@+id/linearLayout3"
111:         android:background="@drawable/blue_border"
112:         android:orientation="horizontal" >
113:
114:         <TextView
115:             android:id="@+id/startDate"
116:             android:layout_width="0dp"
117:             android:layout_height="match_parent"
118:             android:layout_weight="90" />
119:
120:         <View
121:             android:id="@+id/fourthDivider"
122:             android:layout_width="1dp"
123:             android:layout_height="fill_parent"
124:             android:layout_weight="1"
125:             android:background="#3A4978" />
126:
```

```
127:         <TextView
128:             android:id="@+id/endDate"
129:             android:layout_width="0dp"
130:             android:layout_height="match_parent"
131:             android:layout_weight="90" />
132:     </LinearLayout>
133:
134:     <LinearLayout
135:         android:id="@+id/linearLayout5"
136:         android:layout_width="match_parent"
137:         android:layout_height="wrap_content"
138:         android:layout_below="@+id/linearLayout4"
139:         android:background="@drawable/blue_border"
140:         android:orientation="horizontal" >
141:
142:         <TextView
143:             android:id="@+id/units"
144:             android:layout_width="0dp"
145:             android:layout_height="match_parent"
146:             android:layout_weight="90" />
147:
148:         <View
149:             android:id="@+id/fifthDivider"
150:             android:layout_width="1dp"
151:             android:layout_height="fill_parent"
152:             android:layout_weight="1"
153:             android:background="#3A4978" />
154:
155:         <TextView
156:             android:id="@+id/teachingPlace"
157:             android:layout_width="0dp"
158:             android:layout_height="match_parent"
159:             android:layout_weight="90" />
160:     </LinearLayout>
161:
162:     <LinearLayout
163:         android:id="@+id/linearLayout6"
164:         android:layout_width="match_parent"
165:         android:layout_height="wrap_content"
166:         android:layout_below="@+id/linearLayout5"
167:         android:background="@drawable/blue_border"
168:         android:orientation="horizontal" >
169:
170:         <TextView
171:             android:id="@+id/manager"
172:             android:layout_width="0dp"
173:             android:layout_height="match_parent"
174:             android:layout_weight="90" />
175:
176:         <View
177:             android:id="@+id/sixthDivider"
178:             android:layout_width="1dp"
179:             android:layout_height="fill_parent"
180:             android:layout_weight="1"
181:             android:background="#3A4978" />
182:
183:         <TextView
184:             android:id="@+id/institutionRepresentative"
185:             android:layout_width="0dp"
186:             android:layout_height="match_parent"
187:             android:layout_weight="90" />
188:     </LinearLayout>
189:
```

```
190:         <LinearLayout
191:             android:id="@+id/linearLayout7"
192:             android:layout_width="match_parent"
193:             android:layout_height="wrap_content"
194:             android:layout_below="@+id/linearLayout6"
195:             android:background="@drawable/blue_border"
196:             android:orientation="horizontal" >
197:
198:             <TextView
199:                 android:id="@+id/headquarters"
200:                 android:layout_width="0dp"
201:                 android:layout_height="match_parent"
202:                 android:layout_weight="90" />
203:
204:             <View
205:                 android:id="@+id/seventhDivider"
206:                 android:layout_width="1dp"
207:                 android:layout_height="fill_parent"
208:                 android:layout_weight="1"
209:                 android:background="#3A4978" />
210:
211:             <TextView
212:                 android:id="@+id/evalAcecau"
213:                 android:layout_width="0dp"
214:                 android:layout_height="match_parent"
215:                 android:layout_weight="90" />
216:         </LinearLayout>
217:
218:         <LinearLayout
219:             android:layout_width="match_parent"
220:             android:layout_height="match_parent"
221:             android:layout_below="@+id/linearLayout7"
222:             android:orientation="vertical" >
223:
224:             <TextView
225:                 android:id="@+id/competencias"
226:                 android:layout_width="match_parent"
227:                 android:layout_height="wrap_content"
228:                 android:background="@drawable/blue_border" />
229:
230:             <TextView
231:                 android:id="@+id/justificaCompetencias"
232:                 android:layout_width="match_parent"
233:                 android:layout_height="wrap_content"
234:                 android:background="@drawable/blue_border" />
235:
236:             <TextView
237:                 android:id="@+id/resultadosEsperados"
238:                 android:layout_width="match_parent"
239:                 android:layout_height="wrap_content"
240:                 android:background="@drawable/blue_border" />
241:
242:             <TextView
243:                 android:id="@+id/recursosDisponibles"
244:                 android:layout_width="match_parent"
245:                 android:layout_height="wrap_content"
246:                 android:background="@drawable/blue_border" />
247:         </LinearLayout>
248:     </RelativeLayout>
249:
250: </ScrollView>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/lightblue_border" >
5:
6:     <LinearLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content"
9:         android:layout_marginTop="10dp"
10:        android:background="@drawable/blue_border"
11:        android:orientation="vertical" >
12:
13:        <TextView
14:            android:layout_width="fill_parent"
15:            android:layout_height="wrap_content"
16:            android:gravity="left"
17:            android:text="@string/final_budget"
18:            android:textStyle="bold" />
19:
20:        <include
21:            android:id="@+id/incomes_total"
22:            layout="@layout/two_items_4_tol" />
23:
24:        <include
25:            android:id="@+id/parcial_cost"
26:            layout="@layout/two_items_4_tol" />
27:
28:        <include
29:            android:id="@+id/ulpgc_compensation"
30:            layout="@layout/two_items_4_tol" />
31:
32:        <include
33:            android:id="@+id/institute_compensation"
34:            layout="@layout/two_items_4_tol" />
35:
36:        <include
37:            android:id="@+id/manager_compensation"
38:            layout="@layout/two_items_4_tol" />
39:
40:        <include
41:            android:id="@+id/total"
42:            layout="@layout/two_items_4_tol" />
43:
44:        <include
45:            android:id="@+id/balance"
46:            layout="@layout/two_items_4_tol"
47:            android:background="@null" />
48:    </LinearLayout>
49:
50: </ScrollView>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/lightblue_border"
5:     android:orientation="vertical"
6:     android:paddingTop="10dp"
7:     android:weightSum="2" >
8:
9:     <LinearLayout
10:         android:layout_width="match_parent"
11:         android:layout_height="wrap_content"
12:         android:layout_weight="1"
13:         android:orientation="vertical" >
14:
15:         <TextView
16:             android:layout_width="match_parent"
17:             android:layout_height="wrap_content"
18:             android:gravity="center"
19:             android:text="@string/year"
20:             android:textStyle="bold" />
21:
22:         <RelativeLayout
23:             android:layout_width="fill_parent"
24:             android:layout_height="wrap_content"
25:             android:gravity="center_horizontal" >
26:
27:             <include
28:                 android:id="@+id/first_row"
29:                 android:layout_width="wrap_content"
30:                 layout="@layout/single_row"
31:                 android:gravity="center" />
32:         </RelativeLayout>
33:
34:         <TextView
35:             android:layout_width="match_parent"
36:             android:layout_height="wrap_content"
37:             android:gravity="center"
38:             android:text="@string/application_dates"
39:             android:textStyle="bold" />
40:
41:         <include
42:             android:id="@+id/second_row"
43:             layout="@layout/two_items_row" />
44:
45:         <TextView
46:             android:layout_width="match_parent"
47:             android:layout_height="wrap_content"
48:             android:gravity="center"
49:             android:text="@string/first_rectification"
50:             android:textStyle="bold" />
51:
52:         <include
53:             android:id="@+id/third_row"
54:             layout="@layout/two_items_row" />
55:
56:         <TextView
57:             android:layout_width="match_parent"
58:             android:layout_height="wrap_content"
59:             android:gravity="center"
60:             android:text="@string/second_rectification"
61:             android:textStyle="bold" />
62:
63:         <include
```

```
64:         android:id="@+id/fourth_row"
65:         layout="@layout/two_items_row" />
66:     </LinearLayout>
67:
68:     <LinearLayout
69:         android:layout_width="match_parent"
70:         android:layout_height="wrap_content"
71:         android:layout_weight="1"
72:         android:orientation="vertical" >
73:
74:         <Button
75:             android:id="@+id/send_button"
76:             android:layout_width="fill_parent"
77:             android:layout_height="wrap_content"
78:             android:layout_gravity="right"
79:             android:background="@drawable/mybutton_layout"
80:             android:text="@string/send"
81:             android:textSize="24sp" />
82:     </LinearLayout>
83:
84: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:layout_width="match_parent"
4:     android:layout_height="wrap_content"
5:     android:background="@drawable/blue_border"
6:     android:orientation="horizontal" >
7:
8:     <include
9:         android:id="@+id/left_item"
10:        layout="@layout/info_cell" />
11:
12:     <include layout="@layout/divider" />
13:
14:     <include
15:         android:id="@+id/right_item"
16:        layout="@layout/info_cell" />
17:
18: </LinearLayout>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/lightblue_border" >
5:
6:     <LinearLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content"
9:         android:orientation="vertical" >
10:
11:         <TextView
12:             android:layout_width="fill_parent"
13:             android:layout_height="wrap_content"
14:             android:layout_marginTop="5dp"
15:             android:gravity="center"
16:             android:text="@string/incomeTab"
17:             android:textStyle="bold" />
18:
19:         <LinearLayout
20:             android:layout_width="match_parent"
21:             android:layout_height="wrap_content"
22:             android:background="@drawable/blue_border"
23:             android:orientation="vertical" >
24:
25:             <TextView
26:                 android:layout_width="fill_parent"
27:                 android:layout_height="wrap_content"
28:                 android:layout_marginTop="5dp"
29:                 android:gravity="left"
30:                 android:text="@string/title_incomeRegistration"
31:                 android:textStyle="bold" />
32:
33:             <include
34:                 android:id="@+id/first_row"
35:                 layout="@layout/three_items_row" />
36:
37:             <include
38:                 android:id="@+id/second_row"
39:                 layout="@layout/three_items_row" />
40:
41:             <include
42:                 android:id="@+id/students_sum"
43:                 layout="@layout/two_items_4_tol"
44:                 android:background="@null" />
45:
46:             <TextView
47:                 android:layout_width="fill_parent"
48:                 android:layout_height="wrap_content"
49:                 android:layout_marginTop="5dp"
50:                 android:gravity="left"
51:                 android:text="@string/title_directIncome"
52:                 android:textStyle="bold" />
53:
54:             <include
55:                 android:id="@+id/direct_income_row"
56:                 layout="@layout/two_items_4_tol" />
57:         </LinearLayout>
58:     </LinearLayout>
59:
60: </ScrollView>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     xmlns:tools="http://schemas.android.com/tools"
4:     android:layout_width="match_parent"
5:     android:layout_height="match_parent"
6:     android:orientation="vertical"
7:     android:weightSum="3" >
8:
9:     <LinearLayout
10:         android:layout_width="match_parent"
11:         android:layout_height="0dp"
12:         android:layout_weight="1"
13:         android:orientation="vertical" >
14:
15:         <ImageView
16:             android:layout_width="match_parent"
17:             android:layout_height="match_parent"
18:             android:contentDescription="@string/desc"
19:             android:src="@drawable/logo_vtd_rgb" />
20:     </LinearLayout>
21:
22:     <LinearLayout
23:         android:layout_width="match_parent"
24:         android:layout_height="0dp"
25:         android:layout_weight="2"
26:         android:background="@drawable/invert_gradient_background_noborder"
27:         android:orientation="horizontal"
28:         android:paddingBottom="25dp"
29:         android:paddingLeft="25dp"
30:         android:paddingRight="25dp"
31:         android:paddingTop="25dp"
32:         android:weightSum="2" >
33:
34:         <Button
35:             android:id="@+id/button_teacher"
36:             android:layout_width="0dp"
37:             android:layout_height="match_parent"
38:             android:layout_weight="1"
39:             android:background="@drawable/mybutton_layout"
40:             android:text="@string/accessMessageTeacher"
41:             android:textColor="#3A4978"
42:             android:textSize="16sp"
43:             android:textStyle="bold"
44:             android:typeface="sans"
45:             tools:ignore="NestedWeights" />
46:
47:         <Button
48:             android:id="@+id/button_manager"
49:             android:layout_width="0dp"
50:             android:layout_height="match_parent"
51:             android:layout_weight="1"
52:             android:background="@drawable/mybutton_layout"
53:             android:text="@string/accessMessageManager"
54:             android:textColor="#3A4978"
55:             android:textSize="16sp"
56:             android:textStyle="bold" />
57:     </LinearLayout>
58:
59: </LinearLayout>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#3A4978" />
6:
7:     <solid android:color="#B0C4DE" />
8:
9:     <padding
10:        android:bottom="7dp"
11:        android:left="7dp"
12:        android:right="7dp"
13:        android:top="0dp" />
14:
15:     <corners android:radius="4dp" />
16:
17: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#FF0000" />
6:
7:     <solid android:color="#525E66" />
8:
9:     <gradient
10:        android:angle="225"
11:        android:endColor="#FFFFFF"
12:        android:startColor="#525E66" />
13:
14:     <padding
15:        android:bottom="7dp"
16:        android:left="7dp"
17:        android:right="7dp"
18:        android:top="7dp" />
19:
20:     <corners android:radius="4dp" />
21:
22: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <gradient
4:         android:angle="225"
5:         android:endColor="#FFFFFF"
6:         android:startColor="#B0C4DE" />
7:
8:     <padding
9:         android:bottom="7dp"
10:        android:left="7dp"
11:        android:right="7dp"
12:        android:top="0dp" />
13:
14:     <corners android:radius="4dp" />
15:
16: </shape>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <selector xmlns:android="http://schemas.android.com/apk/res/android">
3:
4:     <item android:drawable="@drawable/lightblue_border" android:state_pressed="true"/>
<!-- pressed -->
5:     <item android:drawable="@drawable/darkerwhite_border"/> <!-- Normal -->
6:
7: </selector>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android">
2:   <stroke android:width="4dp" android:color="#FF0000" />
3:   <solid android:color="#ffffff" />
4:   <padding android:left="7dp" android:top="7dp"
5:           android:right="7dp" android:bottom="7dp" />
6:   <corners android:radius="4dp" />
7: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#3A4978" />
6:
7:     <solid android:color="#B0C4DE" />
8:
9:     <gradient
10:        android:angle="225"
11:        android:endColor="#FFFFFF"
12:        android:startColor="#B0C4DE" />
13:
14:     <padding
15:        android:bottom="7dp"
16:        android:left="7dp"
17:        android:right="7dp"
18:        android:top="0dp" />
19:
20:     <corners android:radius="4dp" />
21:
22: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android">
2:   <stroke android:width="4dp" android:color="#3A4978" />
3:   <solid android:color="#ffffff" />
4:   <padding android:left="7dp" android:top="7dp"
5:           android:right="7dp" android:bottom="7dp" />
6:   <corners android:radius="4dp" />
7: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <gradient
4:         android:angle="270"
5:         android:endColor="#B0C4DE"
6:         android:startColor="#FFFFFF" />
7:
8:     <padding
9:         android:bottom="7dp"
10:        android:left="7dp"
11:        android:right="7dp"
12:        android:top="0dp" />
13:
14:     <corners android:radius="4dp" />
15:
16: </shape>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#3A4978" />
6:
7:     <solid android:color="#ffffff0" />
8:
9:     <gradient
10:        android:angle="225"
11:        android:endColor="#FFFFFF"
12:        android:startColor="#C0C0C0" />
13:
14:     <padding
15:        android:bottom="7dp"
16:        android:left="7dp"
17:        android:right="7dp"
18:        android:top="7dp" />
19:
20:     <corners android:radius="4dp" />
21:
22: </shape>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <selector xmlns:android="http://schemas.android.com/apk/res/android">
3:
4:     <item android:drawable="@drawable/lightblue_border" android:state_pressed="true"/>
<!-- pressed -->
5:     <item android:drawable="@drawable/grey_colour_gradient_with_red_border"/> <!-- Normal -->
6:
7: </selector>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#3A4978" />
6:
7:     <solid android:color="#B0C4DE" />
8:
9:     <gradient
10:        android:angle="225"
11:        android:startColor="#B0C4DE"
12:        android:endColor="#FFFFFF" />
13:
14:     <padding
15:        android:bottom="7dp"
16:        android:left="7dp"
17:        android:right="7dp"
18:        android:top="0dp" />
19:
20:     <corners android:radius="4dp" />
21:
22: </shape>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <selector xmlns:android="http://schemas.android.com/apk/res/android">
3:
4:     <item android:drawable="@drawable/lightblue_border" android:state_pressed="true"/>
<!-- pressed -->
5:     <item android:drawable="@drawable/grey_colour_gradient"/> <!-- Normal -->
6:
7: </selector>
```

```
1: <shape xmlns:android="http://schemas.android.com/apk/res/android" >
2:
3:     <stroke
4:         android:width="4dp"
5:         android:color="#3A4978" />
6:
7:     <solid android:color="#525E66" />
8:
9:     <gradient
10:        android:angle="225"
11:        android:endColor="#FFFFFF"
12:        android:startColor="#525E66" />
13:
14:     <padding
15:        android:bottom="7dp"
16:        android:left="7dp"
17:        android:right="7dp"
18:        android:top="7dp" />
19:
20:     <corners android:radius="4dp" />
21:
22: </shape>
```

```
1: <resources>
2:
3:     <string name="appName">ULPGC Postgraduate Studies Manager</string>
4:     <string name="accessMessageManager">Login as Manager</string>
5:     <string name="accessMessageTeacher">Login as Teacher</string>
6:     <string name="rememberMe_message">Remember user</string>
7:     <string name="button_login">Sign In</string>
8:     <string name="login_string">Username</string>
9:     <string name="pass_string">Password</string>
10:    <string name="loggingIn">Logging In&#8230;</string>
11:    <string name="errorLogin">Login/Password incorrect</string>
12:    <string name="message_back_start">Back to login screen</string>
13:    <string name="retry">Retry</string>
14:    <string name="cancel">Cancel</string>
15:    <string name="searching">Searching&#8230;</string>
16:    <string name="searchMessage_before">The ongoing search has found </string>
17:    <string name="searchMessage_after"> degree proposals that match the parameters at t
he moment</string>
18:    <string name="searchResults_before">Search results:\nThere has been </string>
19:    <string name="searchResults_after"> results</string>
20:    <string name="finishSearch">End search?</string>
21:    <string name="showResults">Display results</string>
22:    <string name="repeatSearch">Search again</string>
23:    <string name="checkingMessage">Checking&#8230;</string>
24:    <string name="teacherNotPresent">The ID number does not match any ULPGC teacher alr
eady in the system</string>
25:    <string name="directorValidated">Academic director ID validated</string>
26:    <string name="logoutMessageTitle">Log out</string>
27:    <string name="logoutMessage">Are you sure you want to log out of the system?</strin
g>
28:    <string name="edicion">Period</string>
29:    <string name="tipo">Kind</string>
30:    <string name="estado">State</string>
31:    <string name="buscar">Search</string>
32:    <string name="busqueda_nombre">Search by name or part of it</string>
33:    <string name="busqueda_codigo">Search by degree code</string>
34:    <string name="ownDegreeBasic">&lt;b>Elemental Own Degree</b></string>
35:    <string name="ownDegreeHigher">&lt;b>Post-Elemental Own Degree</b></string>
>
36:    <string name="expert">&lt;b>Expert</b></string>
37:    <string name="mastery">&lt;b>Mastery</b></string>
38:    <string name="professionalDiploma">&lt;b>Professional Diploma</b></string>
39:    <string name="search_hint">Tip: Any number of search condition can be used. If none
are specified, all degrees available will be shown.</string>
40:    <string name="state">&lt;b>State: </b></string>
41:    <string name="starts">&lt;b>Starts: </b></string>
42:    <string name="code">&lt;b>Code: </b></string>
43:    <string name="loadingMessage">Loading&#8230;</string>
44:    <string name="noNumberFields">The data introduced isn't a number</string>
45:    <string name="errorReport">Validation errors report</string>
46:    <string name="msgNoValidationErrors">This degree proposal is already in \'Sent\' st
ate, hence it has no validation errors</string>
47:    <string name="academic_info">Academic information</string>
48:    <string name="subjects">Subjects</string>
49:    <string name="economic_info">Economic report</string>
50:    <string name="teachers">Teachers</string>
51:    <string name="send_to_evaluate">Submit proposal for validation</string>
52:    <string name="download_degree">Download complete report in PDF format</string>
53:    <string name="report">Validation errors</string>
54:    <string name="unspecified">Unspecified</string>
55:    <string name="hours">hours</string>
56:    <string name="error_io">Network error</string>
57:    <string name="loadMoreTeachers_first">Load more teachers&lt;br>4/</string>
```

```
58: <string name="loadedTeachers"> teachers already loaded</string>
59: <string name="loadMoreTeachers">Load more teachers<br></string>
60: <string name="notify">&lt;b>Notify</b></string>
61: <string name="yes">Yes</string>
62: <string name="notPresent">Not present</string>
63: <string name="notUlpgcTeacher">Not an ULPGC teacher</string>
64: <string name="notifying">Notifying&#8230;</string>
65: <string name="final_budget">Income/Expenses Overview</string>
66: <string name="overviewTab">Overview</string>
67: <string name="incomeTab">Income</string>
68: <string name="expensesTab">Expenses</string>
69: <string name="student_sum">Tuition income</string>
70: <string name="empty">Select an option on the left to display</string>
71: <string name="error_io_message">Please check your internet connection</string>
72: <string name="error_period">Error. Period closed</string>
73: <string name="loadedSubjects">loaded subjects</string>
74: <string name="noStorageAvailable">SD card not present</string>
75: <string name="noAppAvailable">No Application Available to View PDF</string>
76: <string name="year">Academic year</string>
77: <string name="application_dates">Proposal application</string>
78: <string name="first_rectification">First rectification period</string>
79: <string name="second_rectification">Second rectification period</string>
80: <string name="send">Send</string>
81: <string name="period_start">Starts:</string>
82: <string name="period_end">Ends:</string>
83: <string name="outOfTime">The deadline has expired</string>
84: <string name="incorrectState">A proposal in sent/rejected/canceled state can't be
send to be evaluated</string>
85: <string name="sendingDegree">Sending&#8230;</string>
86: <string name="sent">Proposal sent</string>
87: <string name="outOfPeriod">The proposal intended for verification can't be send to
evaluate. The academic year is different for the current verification period</string>
88: <string name="outOfPeriodForProposal">Currently, the period for new proposal applic
ation is closed</string>
89: <string name="unidentifiedError">The proposal could not be sent. Please try again u
sing the web application</string>
90: <string name="progressDownload">Download progress</string>
91: <string name="downloading">Downloading&#8230;</string>
92: <string name="downloaded">Bytes downloaded</string>
93: <string name="new_proposal">Create new proposal</string>
94: <string name="explore_proposals">Explore existing proposals</string>
95: <string name="degree_denomination">Name</string>
96: <string name="degree_competences">Competences</string>
97: <string name="degree_proposed_by">Proponent</string>
98: <string name="degree_academic_director">Academic Director's ID Number</string>
99: <string name="emptyFields">The proposal cannot be send. There are empty text fields
left</string>
100: <string name="proposalSent">The proposal has been sent. Remember that is only a dra
ft. The rest of the proposal has to be introduced via web application</string>
101:
102: <string name="check_academic_director">Check ID Number</string>
103: <string name="degree_units">Units</string>
104: <string name="degree_min_alumns">Min students</string>
105: <string name="degree_max_alumns">Max students</string>
106: <string name="degree_economic_manager">Economic Manager</string>
107: <string name="mode">Mode</string>
108: <string name="oriented">Oriented to</string>
109: <string name="degree_min_credits">Min annual credits</string>
110: <string name="knowledge_branch">Branch of knowledge</string>
111: <string name="institutions">Institution</string>
112: <string name="teaching_address">Institution headquarters</string>
113: <string name="teaching_outside">The degree is taught outside of the Canary Islands<
/string>
```

```
114:     <string name="title_activity_login_user">Authentication</string>
115:     <string name="title_activity_search_degrees">Degree proposals search</string>
116:     <string name="title_activity_search_results">Search results</string>
117:     <string name="title_generalInfo">Overview</string>
118:     <string name="title_hourDistribution">Hour distribution</string>
119:     <string name="title_expenditures">Expenses</string>
120:     <string name="title_ULPGCteacherExpenditures">ULPGC Teachers</string>
121:     <string name="title_ExternalteacherExpenditures">External Teachers</string>
122:     <string name="title_incomeRegistration">Tuition related income</string>
123:     <string name="title_directIncome">Direct income and grants</string>
124:     <string name="title_activity_degree_main_menu">Main menu</string>
125:     <string name="title_activity_academic_info">General information</string>
126:     <string name="title_activity_subjects_info">Subjects</string>
127:     <string name="title_activity_teachers_info">Teachers</string>
128:     <string name="title_activity_economic_info">Economic report</string>
129:     <string name="title_activity_reports_info">Issues report</string>
130:     <string name="title_activity_single_subject">SingleSubject</string>
131:     <string name="title_activity_single_teacher">SingleTeacher</string>
132:     <string name="message_no_errors">The proposal have no errors</string>
133:     <string name="errors_activity">Proposal generated errors</string>
134:     <string name="title_activity_periods">Periods</string>
135:     <string name="title_activity_select_task">Select task</string>
136:     <string name="title_activity_create_degree">Create new proposal</string>
137:
138: </resources>
```

```
1: <resources>
2:     <string-array name="degree_types">
3:         <item>Kind of degree</item>
4:         <item>Elemental Own Degree</item>
5:         <item>Post-Elemental Own Degree</item>
6:         <item>Expert</item>
7:         <item>Mastery</item>
8:         <item>Professional Diploma</item>
9:     </string-array>
10:    <string-array name="degree_states">
11:        <item>Degree state</item>
12:        <item>Draft</item>
13:        <item>Sent</item>
14:        <item>Rejected</item>
15:        <item>Approved</item>
16:        <item>Correction Period</item>
17:        <item>Canceled</item>
18:    </string-array>
19:    <string-array name="degree_editions">
20:        <item>Starting academic year</item>
21:    <!--        <item>2014/2015</item> -->
22:        <item>2013/2014</item>
23:        <item>2012/2013</item>
24:        <item>2011/2012</item>
25:    </string-array>
26: </resources>
```

```
1: <resources xmlns:android="http://schemas.android.com/apk/res/android">
2:
3:     <string-array name="informacion_academica">
4:         <item>Denomination: </item>
5:         <item>Branch of knowledge: </item>
6:         <item>Kind of studies: </item>
7:         <item>Proposal state: </item>
8:         <item>Edition number: </item>
9:         <item>Code: </item>
10:        <item>Proposer: </item>
11:        <item>Start of lectures: </item>
12:        <item>End of lectures: </item>
13:        <item>Academic management units: </item>
14:        <item>Place of teaching: </item>
15:        <item>Manager: </item>
16:        <item>Institution representative: </item>
17:        <item>Head office: </item>
18:        <item>Evaluated by ACECAU: </item>
19:        <item>Competences: </item>
20:        <item>Competences justification: </item>
21:        <item>Expected results: </item>
22:        <item>Resources availables: </item>
23:        <item>Id</item>
24:    </string-array>
25:    <string-array name="subject_forAdapter">
26:        <item><b>Year: </b></item>
27:        <item><b>Semester: </b></item>
28:        <item><b>Starts: </b></item>
29:        <item><b>Ends: </b></item>
30:        <item><b>ECTS Credits: </b></item>
31:        <item><b>Type of subject: </b></item>
32:    </string-array>
33:    <string-array name="subject_types">
34:        <item>Core </item>
35:        <item>Mandatory </item>
36:        <item>Elective </item>
37:        <item>External internship </item>
38:        <item>Final thesis </item>
39:        <item>Mixed </item>
40:    </string-array>
41:    <string-array name="completeSubjectTags">
42:        <item>Name</item>
43:        <item>Year</item>
44:        <item>Semester</item>
45:        <item>Type of subject</item>
46:        <item>ECTS Credits</item>
47:        <item>Starts</item>
48:        <item>Ends</item>
49:        <item>Branch of knowledge</item>
50:        <item>Mode</item>
51:        <item>Created</item>
52:        <item>Updated</item>
53:        <item>Theory classes</item>
54:        <item>Practical classes</item>
55:        <item>Seminars</item>
56:        <item>External internships</item>
57:        <item>Evaluation tests</item>
58:        <item>Final thesis</item>
59:        <item>Student study hours</item>
60:        <item>Total amount of hours</item>
61:        <item>Competences</item>
62:        <item>Activities</item>
63:        <item>Content description</item>
```

```
64:     <item>Evaluation system</item>
65: </string-array>
66: <string-array name="branchKnowledge">
67:     <item>Arts and humanities</item>
68:     <item>Science</item>
69:     <item>Social sciences and Law</item>
70:     <item>Health sciences</item>
71:     <item>Engineering and architecture</item>
72: </string-array>
73: <string-array name="branchKnowledgeComplete">
74:     <item>Branch of knowledge</item>
75:     <item>Arts and humanities</item>
76:     <item>Science</item>
77:     <item>Social sciences and Law</item>
78:     <item>Health sciences</item>
79:     <item>Engineering and architecture</item>
80: </string-array>
81: <string-array name="teachingMode">
82:     <item>&lt;br&gt;In-person class</item>
83:     <item>&lt;br&gt;Distance learning </item>
84: </string-array>
85: <string-array name="teachingModeComplete">
86:     <item>Teaching mode</item>
87:     <item>In-person class</item>
88:     <item>Mixed teaching mode </item>
89:     <item>Distance learning </item>
90: </string-array>
91: <string-array name="orientedTo">
92:     <item>Oriented to</item>
93:     <item>General public</item>
94:     <item>Teachers</item>
95:     <item>Administration and service personnel</item>
96: </string-array>
97: <string-array name="teacher_forAdapter">
98:     <item>&lt;b&gt;Origin:&lt;/b&gt;&lt;br&gt;</item>
99:     <item>&lt;b&gt;State:&lt;/b&gt;&lt;br&gt;</item>
100:    <item>&lt;b&gt;Last notification:&lt;/b&gt;&lt;br&gt;</item>
101:    <item>&lt;b&gt;Amount of notifications:&lt;/b&gt;&lt;br&gt;</item>
102:    <item>&lt;b&gt;Degree:&lt;/b&gt;&lt;br&gt;</item>
103:    <item>&lt;b&gt;E-mail:&lt;/b&gt;&lt;br&gt;</item>
104: </string-array>
105: <string-array name="info_generica_profesores">
106:    <item>&lt;b&gt;Degree:&lt;/b&gt;&lt;br&gt;</item>
107:    <item>&lt;b&gt;E-mail:&lt;/b&gt;&lt;br&gt;</item>
108:    <item>&lt;b&gt;Branch of knowledge:&lt;/b&gt;&lt;br&gt;</item>
109:    <item>&lt;b&gt;Professional category:&lt;/b&gt;&lt;br&gt;</item>
110:    <item>&lt;b&gt;ULPGC graduate:&lt;/b&gt;&lt;br&gt;</item>
111:    <item>&lt;b&gt;ULPGC Department:&lt;/b&gt;&lt;br&gt;</item>
112:    <item>&lt;b&gt;Institution:&lt;/b&gt;&lt;br&gt;</item>
113:    <item>&lt;b&gt;Active:&lt;/b&gt;&lt;br&gt;</item>
114:    <item>&lt;b&gt;Curriculum Vitae:&lt;/b&gt;&lt;br&gt;</item>
115: </string-array>
116: <string-array name="overviewExpenses">
117:    <item>&lt;b&gt;Income total&lt;/b&gt;</item>
118:    <item>&lt;b&gt;Parcial expenses amount&lt;/b&gt;</item>
119:    <item>&lt;b&gt;ULPGC compensation (5*)&lt;/b&gt;</item>
120:    <item>&lt;b&gt;Institution compensation (5*)&lt;/b&gt;</item>
121:    <item>&lt;b&gt;Economic manager compensation (5*)&lt;/b&gt;</item>
122:    <item>&lt;b&gt;Expenses total&lt;/b&gt;</item>
123:    <item>&lt;b&gt;Balance&lt;/b&gt;</item>
124: </string-array>
125: <string-array name="incomeTags">
126:    <item>&lt;b&gt;Amount of scholarships&lt;/b&gt;&lt;br&gt;</item>
```

```
127: <item><b>Price</b><br></item>
128: <item><b>Amount of full price students </b><br></item>
129: <item><b>Price</b><br></item>
130: <item><b>Total</b><br></item>
131: <item><b>External income</b><br></item>
132: <item><b>Income total</b><br></item>
133: </string-array>
134: <string-array name="expensesTags">
135: <item><b>Teaching hours</b><br></item>
136: <item><b>Cost per hour</b><br><br></item>
137: <item><b>Total</b><br><br></item>
138: <item><b>Seminar hours</b><br></item>
139: <item><b>Cost per hour</b><br><br><br></item>
140: <item><b>Total</b><br><br><br></item>
141: <item><b>Lab hours</b><br><br><br></item>
142: <item><b>Cost per hour</b><br><br><br><br></item>
143: <item><b>Total</b><br><br><br></item>
144: <item><b>Evaluation hours</b><br><br><br><br></item>
145: <item><b>Cost per hour</b><br><br><br><br></item>
146: <item><b>Total</b><br><br><br><br></item>
147: <item><b>Content creation</b><br><br><br><br></item>
148: <item><b>Cost per credit</b><br><br><br><br></item>
149: <item><b>Total</b><br><br><br><br></item>
150: <item><b>In-person tutorships and final project related</b><br><br>
151: <item><b>Cost per project</b><br><br><br><br></item>
152: <item><b>Total</b><br><br><br><br></item>
153: <item><b>Final project evaluation tribunals</b><br><br><br></item>
154: <item><b>Cost per tribunal</b><br><br><br><br></item>
155: <item><b>Total</b><br><br><br><br></item>
156: <item><b>Exam coordination</b><br><br><br><br></item>
157: <item><b>Cost per exam</b><br><br><br><br></item>
158: <item><b>Total</b><br><br><br><br></item>
159: <item><b>Directors</b><br><br><br></item>
160: <item><b>Cost</b><br><br><br></item>
161: <item><b>ULPGC teachers cost</b><br><br><br></item>
162: <item><b>Cost</b><br><br><br></item>
163: <item><b>Organizing unit profit</b><br><br><br></item>
164: <item><b>Cost</b><br><br><br></item>
165: <item><b>ULPGC teachers gross income</b><br><br><br></item>
166: <item><b>Total</b><br><br><br><br></item>
167: <item><b>Teaching hours</b><br><br><br></item>
168: <item><b>Cost per hour</b><br><br><br><br></item>
169: <item><b>Total</b><br><br><br><br></item>
170: <item><b>Seminar hours</b><br><br><br></item>
171: <item><b>Cost per hour</b><br><br><br><br></item>
172: <item><b>Total</b><br><br><br><br></item>
173: <item><b>Lab hours</b><br><br><br><br></item>
174: <item><b>Cost per hour</b><br><br><br><br></item>
175: <item><b>Total</b><br><br><br><br></item>
176: <item><b>Evaluation hours</b><br><br><br><br><br></item>
177: <item><b>Cost per hour</b><br><br><br><br><br><br></item>
178: <item><b>Total</b><br><br><br><br><br><br></item>
179: <item><b>Content creation</b><br><br><br><br><br></item>
180: <item><b>Cost per credit</b><br><br><br><br><br></item>
181: <item><b>Total</b><br><br><br><br><br></item>
182: <item><b>In-person tutorships and final project related</b><br><br>
183: <item><b>Cost per project</b><br><br><br><br></item>
184: <item><b>Total</b><br><br><br><br></item>
185: <item><b>Final project evaluation tribunals</b><br><br><br></item>
186: <item><b>Cost per tribunal</b><br><br><br><br></item>
187: <item><b>Total</b><br><br><br><br></item>
```

188: <item>Exam coordination
</item>
189: <item>Cost per exam

</item>
190: <item>Total

</item>
191: <item>Others
</item>
192: <item>Total
</item>
193: <item>External teachers expense
</item>
194: <item>Total
</item>
195: <item>Administrative support personnel
</item>
196: <item>Cost

</item>
197: <item>Total

</item>
198: <item>Interns

</item>
199: <item>Cost

</item>
200: <item>Total

</item>
201: <item>Specific purpose interns
</item>
202: <item>Cost

</item>
203: <item>Total

</item>
204: <item>Other kind of personnel
</item>
205: <item>Cost

</item>
206: <item>Total

</item>
207: <item>Presentation report
</item>
208: <item>Cost

</item>
209: <item>Final report

</item>
210: <item>Cost

</item>
211: <item>Information point

</item>
212: <item>Cost

</item>
213: <item>External support services

</item>
214: <item>Cost

</item>
215: <item>Others

</item>
216: <item>Cost

</item>
217: <item>Teachers transportation

</item>
218: <item>Cost

</item>
219: <item>Teachers accomodation

</item>
220: <item>Cost

</item>
221: <item>Incoming teachers provision

</item>
222: <item>Cost

</item>
223: <item>Intern students support

</item>
224: <item>Cost

</item>
225: <item>Other expenses

</item>
226: <item>Cost

</item>
227: <item>Locomotions costs

</item>
228: <item>Total

</item>
229: <item>Office supplies

</item>
230: <item>Cost

</item>
231: <item>Lab supplies

</item>
232: <item>Coste

</item>
233: <item>Non-inventoried computer supplies

</item>
234: <item>Cost

</item>
235: <item>Reprography

</item>
236: <item>Cost

</item>
237: <item>Others

</item>
238: <item>Cost

</item>
239: <item>Equipment expenses

</item>
240: <item>Total

</item>
241: <item>Bibliography

</item>
242: <item>Cost

</item>
243: <item>Others

</item>
244: <item>Cost

</item>
245: <item>Bibliography expenses

</item>
246: <item>Total

</item>
247: <item>Printed materials

</item>
248: <item>Cost

</item>
249: <item>Advertising in press and other ways

</item>
250: <item>Cost

</item>

```
251:      <item>&lt;b&gt;Advertising expenses&lt;/b&gt;&lt;br&gt;</item>
252:      <item>&lt;b&gt;Total&lt;/b&gt;&lt;br&gt;</item>
253:      <item>&lt;b&gt;Opening/Closing ceremonies&lt;/b&gt;&lt;br&gt;</item>
254:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
255:      <item>&lt;b&gt;Simultaneous translation&lt;/b&gt;&lt;br&gt;</item>
256:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
257:      <item>&lt;b&gt;Certificates&lt;/b&gt;&lt;br&gt;</item>
258:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
259:      <item>&lt;b&gt;Mandatory insurance&lt;/b&gt;&lt;br&gt;</item>
260:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
261:      <item>&lt;b&gt;Quality assessment by an external entity&lt;/b&gt;&lt;br&gt;</item>
em>
262:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
263:      <item>&lt;b&gt;Virtual campus&lt;/b&gt;&lt;br&gt;</item>
264:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
265:      <item>&lt;b&gt;Others&lt;/b&gt;&lt;br&gt;</item>
266:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
267:      <item>&lt;b&gt;Other expenses section&lt;/b&gt;&lt;br&gt;</item>
268:      <item>&lt;b&gt;Cost&lt;/b&gt;&lt;br&gt;</item>
269:      <item>&lt;b&gt;Added expenses&lt;/b&gt;&lt;br&gt;</item>
270:      <item>&lt;b&gt;Total&lt;/b&gt;&lt;br&gt;</item>
271:  </string-array>
272:
273: </resources>
```

```
1: <resources>
2:
3:     <bool name="portrait_only">false</bool>
4:
5: </resources>
```

```
1: <resources>
2:
3:     <!--
4:         Customize dimensions originally defined in res/values/dimens.xml (such as
5:         screen margins) for sw720dp devices (e.g. 10" tablets) in landscape here.
6:     -->
7:     <dimen name="activity_horizontal_margin">128dp</dimen>
8:
9: </resources>
```

```
1: <resources>
2:
3:     <item name="contained_fragment" type="id"/>
4:
5: </resources>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <resources>
3:
4:     <string name="appName">Gestor Titulos Propios ULPGC</string>
5:     <string name="accessMessageManager">Acceder como Gestor</string>
6:     <string name="accessMessageTeacher">Acceder como Docente</string>
7:     <string name="rememberMe_message">Recordar usuario</string>
8:     <string name="button_login">Acceder</string>
9:     <string name="login_string">Nombre de usuario</string>
10:    <string name="pass_string">Contraseña<+a</string>
11:    <string name="loggingIn">Accediendo&#8230;</string>
12:    <string name="errorLogin">Nombre de usuario/Contraseña<+a incorrecto</string>
13:    <string name="message_back_start">Volver a la pantalla de inicio</string>
14:    <string name="retry">Reintentar</string>
15:    <string name="cancel">Cancelar</string>
16:    <string name="searching">Buscando&#8230;</string>
17:    <string name="searchMessage_before">Por el momento, se han encontrado </string>
18:    <string name="searchMessage_after"> propuestas que cumplen los criterios de bÃ³squeda</string>
19:    <string name="searchResults_before">Resultados de la bÃ³squeda:\nSe han encontrado
</string>
20:    <string name="searchResults_after">tÃ³tulos que coinciden con los \nparÃ³metros de
bÃ³squeda introducidos</string>
21:    <string name="finishSearch">Â¿Concluir bÃ³squeda?</string>
22:    <string name="showResults">Mostrar resultados</string>
23:    <string name="repeatSearch">Repetir bÃ³squeda</string>
24:    <string name="logoutMessageTitle">Salir</string>
25:    <string name="logoutMessage">Â¿EstÃ³ seguro de querer salir de la sesiÃ³n iniciada?
</string>
26:    <string name="edicion">EdiciÃ³n</string>
27:    <string name="tipo">Tipo</string>
28:    <string name="estado">Estado</string>
29:    <string name="buscar">Buscar</string>
30:    <string name="directorValidated">DNI de Director acadÃ©mico validado</string>
31:    <string name="busqueda_nombre">BÃ³squeda por denominaciÃ³n</string>
32:    <string name="busqueda_codigo">BÃ³squeda por cÃ³digo de tÃ³tulo</string>
33:    <string name="search_hint">Ayuda: Puede usar cualquier nÃºmero de condiciones para
la bÃ³squeda. Si no elige ninguna, se mostrarÃ³n todos los tÃ³tulos propuestos.</string>
34:    <string name="ownDegreeBasic">&lt;/b>&gt;EnseÃ±anza Propia BÃ¡sica&lt;/b>&lt;/string>
35:    <string name="ownDegreeHigher">&lt;/b>&gt;EnseÃ±anza Propia Superior&lt;/b>&lt;/string>
36:    <string name="expert">&lt;/b>&gt;Experto Universitario&lt;/b>&lt;/string>
37:    <string name="mastery">&lt;/b>&gt;MaestrÃ­a Universitaria&lt;/b>&lt;/string>
38:    <string name="professionalDiploma">&lt;/b>&gt;Diploma Profesional&lt;/b>&lt;/string>
39:    <string name="state">&lt;/b>&gt;Estado: &lt;/b>&lt;/string>
40:    <string name="starts">&lt;/b>&gt;Inicio: &lt;/b>&lt;/string>
41:    <string name="code">&lt;/b>&gt;CÃ³digo: &lt;/b>&lt;/string>
42:    <string name="loadingMessage">Cargando&#8230;</string>
43:    <string name="errorReport">Informe de errores</string>
44:    <string name="emptyFields">No se ha podido remitir la propuesta. Existen campos sin
rellenar</string>
45:    <string name="msgNoValidationErrors">Esta propuesta se encuentra en estado enviado.
No se han producido errores de verificaciÃ³n</string>
46:    <string name="general_info">InformaciÃ³n general de la propuesta</string>
47:    <string name="academic_info">InformaciÃ³n acadÃ©mica</string>
48:    <string name="subjects">Asignaturas</string>
49:    <string name="economic_info">Memoria econÃ³mica</string>
50:    <string name="teachers">Profesorado</string>
51:    <string name="send_to_evaluate">Enviar propuesta para evaluaciÃ³n</string>
52:    <string name="download_degree">Descarga de la memoria en PDF</string>
53:    <string name="report">Informe de errores</string>
54:    <string name="unspecified">Sin especificar</string>
55:    <string name="hours">horas</string>
```

```
56: <string name="loadMoreTeachers_first">Cargar más docentes<br>4/</string>
57: <string name="loadedTeachers"> profesores cargados</string>
58: <string name="loadMoreTeachers">Cargar más docentes<br></string>
59: <string name="notify"><br>Notificar<br></string>
60: <string name="yes">Sí</string>
61: <string name="notPresent">No introducido</string>
62: <string name="notUlpgcTeacher">No es docente en la ULPGC</string>
63: <string name="checkingMessage">Comprobando<br></string>
64: <string name="teacherNotPresent">El DNI introducido no pertenece a ningún profesor
de la ULPGC propuesto en la plataforma</string>
65: <string name="notifying">Remitiendo notificación<br></string>
66: <string name="final_budget">Resumen Ingresos/Gastos</string>
67: <string name="title_activity_degree_economic_info">Memoria Económica</string>
68: <string name="overviewTab">Resumen</string>
69: <string name="incomeTab">Ingresos</string>
70: <string name="expensesTab">Gastos</string>
71: <string name="student_sum">Ingresos por matrículas</string>
72: <string name="income">Ingresos</string>
73: <string name="balance">Balance</string>
74: <string name="external_incomes">Total</string>
75: <string name="app_name">Gestor TTPP</string>
76: <string name="empty">Seleccione un apartado en el menú de la izquierda</string>
77: <string name="admin_support">Apoyo administrativo</string>
78: <string name="travel_accomodation">Estancia y alojamiento</string>
79: <string name="fungible">Material fungible</string>
80: <string name="bibliography">Bibliografía y material inventariable</string>
81: <string name="ads">Publicidad</string>
82: <string name="other_expenditures">Otros</string>
83: <string name="error_period">Error. Fuera de plazo</string>
84: <string name="error_io">Error de red</string>
85: <string name="error_io_message">Por favor, compruebe su conexión a internet</string>
g>
86: <string name="total_expenses">Total de gastos</string>
87: <string name="errors_activity">Errores en la propuesta</string>
88: <string name="message_no_errors">La propuesta no ha generado ningún error</string>
89: <string name="noStorageAvailable">La tarjeta SD no se encuentra presente</string>
90: <string name="noAppAvailable">No existe aplicación para la visualización del fichero PDF</string>
91: <string name="loadedSubjects">asignaturas cargadas</string>
92: <string name="year">Curso académico</string>
93: <string name="application_dates">Presentación de propuestas</string>
94: <string name="first_rectification">Primer periodo de subsanación</string>
95: <string name="second_rectification">Segundo periodo de subsanación</string>
96: <string name="send">Enviar</string>
97: <string name="period_start">Inicio:</string>
98: <string name="period_end">Fin:</string>
99: <string name="outOfTime">La propuesta se encuentra fuera de plazo</string>
100: <string name="incorrectState">No se puede enviar a evaluar una propuesta en estado rechazado/enviado/cancelado</string>
101: <string name="sendingDegree">Enviando<br></string>
102: <string name="sent">Propuesta enviada</string>
103: <string name="outOfPeriod">La propuesta que se intenta evaluar pertenece a un periodo diferente al curso académico actual</string>
104: <string name="outOfPeriodForProposal">Actualmente, el plazo de solicitud de nuevas propuestas se encuentra cerrado</string>
105: <string name="progressDownload">Progreso de la descarga</string>
106: <string name="downloading">Descargando<br></string>
107: <string name="noNumberFields">El dato introducido no es un número</string>
108: <string name="downloaded">Bytes descargados</string>
109: <string name="degree_denomination">Denominación</string>
110: <string name="degree_competences">Competencias</string>
111: <string name="degree_proposed_by">Proponente</string>
112: <string name="degree_academic_director">DNI de Director Académico</string>
```

```
113: <string name="check_academic_director">Comprobar DNI</string>
114: <string name="degree_units">Unidades participantes</string>
115: <string name="degree_min_alumns">Plazas mÃ¡ximas</string>
116: <string name="degree_max_alumns">Plazas mÃ¡ximas</string>
117: <string name="degree_economic_manager">Gestor econÃ³mico</string>
118: <string name="hello_world">Hello world!</string>
119: <string name="menu_settings">Settings</string>
120: <string name="action_settings">Settings</string>
121: <string name="new_proposal">Crear nueva propuesta</string>
122: <string name="explore_proposals">Explorar propuestas existentes</string>
123: <string name="institutions">Centro adscrito</string>
124: <string name="teaching_address">Sede de la organizaciÃ³n</string>
125: <string name="teaching_outside">TÃ­tulo impartido fuera de las Islas Canarias</stri
ng>
126: <string name="proposalSent">La propuesta ha sido remitida. Recuerde que se trata de
un borrador que ha de ser completado a travÃ©s de la aplicaciÃ³n web</string>
127: <string name="unidentifiedError">La propuesta no se ha podido verificar. Por favor,
intÃ©ntelo de nuevo desde la aplicaciÃ³n web</string>
128: <string name="title_activity_login_user">AutenticaciÃ³n</string>
129: <string name="title_activity_search_degrees">BÃºsqueda de propuestas</string>
130: <string name="title_activity_search_results">Resultado de la bÃºsqueda</string>
131: <string name="title_generalInfo">InformaciÃ³n general</string>
132: <string name="title_hourDistribution">Reparto de horas</string>
133: <string name="title_expenditures">Gastos</string>
134: <string name="title_ULPGCteacherExpenditures">Profesorado ULPGC</string>
135: <string name="title_ExternalteacherExpenditures">Profesorado Externo</string>
136: <string name="title_incomeRegistration">Derivados de las matrÃ©culas</string>
137: <string name="title_directIncome">Directos y subvenciones</string>
138: <string name="title_activity_degree_main_menu">MenÃº principal</string>
139: <string name="title_activity_academic_info">InformaciÃ³n General</string>
140: <string name="title_activity_subjects_info">Asignaturas</string>
141: <string name="title_activity_teachers_info">Profesorado</string>
142: <string name="title_activity_economic_info">Memoria AcadÃ©mica</string>
143: <string name="title_activity_reports_info">Informe de errores</string>
144: <string name="title_activity_single_subject">SingleSubject</string>
145: <string name="title_activity_single_teacher">SingleTeacher</string>
146: <string name="title_activity_periods">Plazos</string>
147: <string name="title_activity_select_task">Seleccionar tarea</string>
148: <string name="title_activity_create_degree">Crear nueva propuesta</string>
149: <string name="mode">Modalidad</string>
150: <string name="oriented">Orientado a</string>
151: <string name="degree_min_credits">MÃ¡ximo de crÃ©ditos anuales</string>
152: <string name="knowledge_branch">Rama de conocimiento</string>
153:
154: </resources>
```

```
1: <resources>
2:
3:     <!--
4:         Base application theme, dependent on API level. This theme is replaced
5:         by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
6:     -->
7:     <style name="AppBaseTheme" parent="android:Theme.Light">
8:         <!--
9:             Theme customizations available in newer API levels can go in
10:            res/values-vXX/styles.xml, while customizations related to
11:            backward-compatibility can go here.
12:         -->
13:     </style>
14:
15:     <!-- Application theme. -->
16:     <style name="AppBase" parent="android:Theme.Light"></style>
17:
18:     <style name="MyTheme" parent="android:Theme.Dialog">
19:         <item name="android:alertDialogStyle">@style/CustomAlertDialogStyle</item>
20:         <item name="android:textColorPrimary">#FFFFFF</item>
21:     </style>
22:
23:     <style name="CustomAlertDialogStyle">
24:         <item name="android:bottomBright">#FFFFFF</item>
25:         <item name="android:bottomDark">#FFFFFF</item>
26:         <item name="android:bottomMedium">#FFFFFF</item>
27:         <item name="android:centerBright">#FFFFFF</item>
28:         <item name="android:centerDark">#FFFFFF</item>
29:         <item name="android:centerMedium">#FFFFFF</item>
30:         <item name="android:fullBright">#FFFFFF</item>
31:         <item name="android:fullDark">#FFFFFF</item>
32:         <item name="android:topBright">#FFFFFF</item>
33:         <item name="android:topDark">#FFFFFF</item>
34:     </style>
35:
36: </resources>
```

./values/bool.xml

Wed Sep 11 14:16:32 2013

1

```
1: <resources>
2:
3:     <bool name="portrait_only">true</bool>
4:
5: </resources>
```

```
1: <resources>
2:
3:     <string name="desc">Logo ULPGC</string>
4:     <string name="ok">OK</string>
5:     <string name="globalState">Singleton HTTP Connection</string>
6:     <string name="vertodos">Ver Todos</string>
7:     <string name="background">Imagen de fondo</string>
8:     <string name="local_emul_addr">http://10.0.2.2:3000/"</string>
9:     <string name="local_phone_addr">http://10.230.156.51:3000/"</string>
10:    <string name="test_env_addr">https://titulospropios-desas.ulpgc.es/"</string>
11:    <string name="production_addr">https://gestion.titulospropios.ulpgc.es/"</string>
12:    <string name="footer_subject_message">Cargando más profesores de la propuesta</string>
13:
14: </resources>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <resources>
3:     <string-array name="degree_types">
4:         <item>Tipo de t tulo</item>
5:         <item>Ense anza Propia B sica</item>
6:         <item>Ense anza Propia Superior</item>
7:         <item>Experto</item>
8:         <item>Maestr a</item>
9:         <item>Diploma Profesional</item>
10:    </string-array>
11:    <string-array name="degree_states">
12:        <item>Estado del t tulo</item>
13:        <item>Borrador</item>
14:        <item>Enviado</item>
15:        <item>Rechazado</item>
16:        <item>Aprobado</item>
17:        <item>En subsanaci n</item>
18:        <item>Cancelado</item>
19:    </string-array>
20:    <string-array name="original_tags_degree_states">
21:        <item>Estado del t tulo</item>
22:        <item>borrador</item>
23:        <item>enviado</item>
24:        <item>rechazado</item>
25:        <item>aprobado</item>
26:        <item>subsanacion</item>
27:        <item>cancelado</item>
28:    </string-array>
29:    <string-array name="degree_editions">
30:        <item>A o de inicio</item>
31:    <!--        <item>2014/2015</item> -->
32:        <item>2013/2014</item>
33:        <item>2012/2013</item>
34:        <item>2011/2012</item>
35:    </string-array>
36: </resources>
```

```
1: <resources>
2:
3:     <!-- Default screen margins, per the Android Design guidelines. -->
4:     <dimen name="activity_horizontal_margin">16dp</dimen>
5:     <dimen name="activity_vertical_margin">16dp</dimen>
6:
7: </resources>
```

```
1: <resources xmlns:android="http://schemas.android.com/apk/res/android">
2:
3:     <string-array name="academic_info">
4:         <item>name</item>
5:         <item>branch-of-knowledge-id</item>
6:         <item>degree-type-id</item>
7:         <item>state</item>
8:         <item>edition-number</item>
9:         <item>code</item>
10:        <item>proposed-by</item>
11:        <item>lectures-start</item>
12:        <item>lectures-end</item>
13:        <item>academic-management-units</item>
14:        <item>teaching-place</item>
15:        <item>administrator</item>
16:        <item>institution-representative</item>
17:        <item>head-office</item>
18:        <item>evaluated-by-acecau-at</item>
19:        <item>competences</item>
20:        <item>justify-competences</item>
21:        <item>expected-results</item>
22:        <item>material-resources</item>
23:        <item>id</item>
24:    </string-array>
25:    <string-array name="informacion_academica">
26:        <item>DenominaciÃ³n: </item>
27:        <item>Rama de Conocimiento: </item>
28:        <item>Tipo de enseÃ±anza: </item>
29:        <item>Estado de la propuesta: </item>
30:        <item>NÃºmero de ediciÃ³n: </item>
31:        <item>CÃ³digo: </item>
32:        <item>Proponente: </item>
33:        <item>Inicio de las enseÃ±anzas: </item>
34:        <item>Fin de las enseÃ±anzas: </item>
35:        <item>Unidades participantes: </item>
36:        <item>Lugar de imparticiÃ³n: </item>
37:        <item>Gestor: </item>
38:        <item>Representante del centro: </item>
39:        <item>Sede de la organizaciÃ³n: </item>
40:        <item>Evaluado por acecau en: </item>
41:        <item>Competencias: </item>
42:        <item>JustificaciÃ³n de competencias: </item>
43:        <item>Resultados esperados: </item>
44:        <item>Recursos disponibles: </item>
45:        <item>Id</item>
46:    </string-array>
47:    <string-array name="subject_forAdapter">
48:        <item><b>Curso: </b></item>
49:        <item><b>Semestre: </b></item>
50:        <item><b>Inicio: </b></item>
51:        <item><b>Fin: </b></item>
52:        <item><b>CrÃ©ditos ECTS: </b></item>
53:        <item><b>Tipo de asignatura: </b></item>
54:    </string-array>
55:    <string-array name="subject_types">
56:        <item>Troncal </item>
57:        <item>Obligatoria </item>
58:        <item>Optativa </item>
59:        <item>PrÃ¡cticas Externas </item>
60:        <item>Trabajo de fin de tÃ­tulo </item>
61:        <item>Mixto </item>
62:    </string-array>
63:    <string-array name="teacher_forAdapter">
```

```
64:         <item>&lt;b&gt;Procedencia:&lt;/b&gt;&lt;br&gt;</item>
65:         <item>&lt;b&gt;Estado:&lt;/b&gt;&lt;br&gt;</item>
66:         <item>&lt;b&gt;\232ltima notificaciÃ³n:&lt;/b&gt;&lt;br&gt;</item>
67:         <item>&lt;b&gt;NÃºmero de notificaciones:&lt;/b&gt;&lt;br&gt;</item>
68:         <item>&lt;b&gt;TitulaciÃ³n:&lt;/b&gt;&lt;br&gt;</item>
69:         <item>&lt;b&gt;E-mail:&lt;/b&gt;&lt;br&gt;</item>
70:     </string-array>
71:     <string-array name="branchKnowledge">
72:         <item>Arte y humanidades</item>
73:         <item>Ciencias</item>
74:         <item>Ciencias sociales y jurÃ©dicas</item>
75:         <item>Ciencias de la salud</item>
76:         <item>IngenierÃ-a y arquitectura</item>
77:     </string-array>
78:     <string-array name="branchKnowledgeComplete">
79:         <item>Rama de conocimiento</item>
80:         <item>Arte y humanidades</item>
81:         <item>Ciencias</item>
82:         <item>Ciencias sociales y jurÃ©dicas</item>
83:         <item>Ciencias de la salud</item>
84:         <item>IngenierÃ-a y arquitectura</item>
85:     </string-array>
86:     <string-array name="teachingMode">
87:         <item>Presencial</item>
88:         <item>No-Presencial</item>
89:     </string-array>
90:     <string-array name="orientedTo">
91:         <item>Dirigido a</item>
92:         <item>PÃºblico en general</item>
93:         <item>Docentes</item>
94:         <item>Personal de administraciÃ³n y servicios</item>
95:     </string-array>
96:     <string-array name="teachingModeComplete">
97:         <item>Modalidad de enseÃ±anza</item>
98:         <item>Presencial</item>
99:         <item>Semi-Presencial</item>
100:        <item>No-Presencial</item>
101:     </string-array>
102:     <string-array name="ulpgc_teachers">
103:         <item>ulpgc-teaching-hours-qty</item>
104:         <item>ulpgc-teaching-hours-cost</item>
105:         <item>ulpgc-teaching-hours-sum</item>
106:         <item>ulpgc-practice-hours-qty</item>
107:         <item>ulpgc-practice-hours-cost</item>
108:         <item>ulpgc-practice-hours-sum</item>
109:         <item>ulpgc-seminar-hours-qty</item>
110:         <item>ulpgc-seminar-hours-cost</item>
111:         <item>ulpgc-seminar-hours-sum</item>
112:         <item>ulpgc-evaluation-hours-qty</item>
113:         <item>ulpgc-evaluation-hours-cost</item>
114:         <item>ulpgc-evaluation-hours-sum</item>
115:         <item>ulpgc-content-credits-qty</item>
116:         <item>ulpgc-content-credits-cost</item>
117:         <item>ulpgc-content-credits-sum</item>
118:         <item>ulpgc-tutorship-jobs-qty</item>
119:         <item>ulpgc-tutorship-jobs-cost</item>
120:         <item>ulpgc-tutorship-jobs-sum</item>
121:         <item>ulpgc-boards-qty</item>
122:         <item>ulpgc-boards-cost</item>
123:         <item>ulpgc-boards-sum</item>
124:         <item>ulpgc-exams-coordination-qty</item>
125:         <item>ulpgc-exams-coordination-cost</item>
126:         <item>ulpgc-exams-coordination-sum</item>
```

```

127:     </string-array>
128:     <string-array name="external_teachers">
129:         <item>teaching-hours-qty</item>
130:         <item>teaching-hours-cost</item>
131:         <item>teaching-hours-sum</item>
132:         <item>seminar-hours-qty</item>
133:         <item>seminar-hours-cost</item>
134:         <item>seminar-hours-sum</item>
135:         <item>practice-hours-qty</item>
136:         <item>practice-hours-cost</item>
137:         <item>practice-hours-sum</item>
138:         <item>evaluation-hours-qty</item>
139:         <item>evaluation-hours-cost</item>
140:         <item>evaluation-hours-sum</item>
141:         <item>content-credits-qty</item>
142:         <item>content-credits-cost</item>
143:         <item>content-credits-sum</item>
144:         <item>tutorship-jobs-qty</item>
145:         <item>tutorship-jobs-cost</item>
146:         <item>tutorship-jobs-sum</item>
147:         <item>boards-qty</item>
148:         <item>boards-cost</item>
149:         <item>boards-sum</item>
150:         <item>exams-coordination-qty</item>
151:         <item>exams-coordination-cost</item>
152:         <item>exams-coordination-sum</item>
153:     </string-array>
154:     <string-array name="ulpgc_extra">
155:         <item>ulpgc-managers-cost</item>
156:         <item>ulpgc-teachers-cost-sum</item>
157:         <item>ulpgc-teachers-unit-profit</item>
158:         <item>ulpgc-teachers-gross</item>
159:     </string-array>
160:     <string-array name="teacher_hours">
161:         <item>&lt;b>Horas de docencia teórica</b>&lt;br></item>
162:         <item>&lt;b>Coste por hora</b>&lt;br></item>
163:         <item>&lt;b>Total</b>&lt;br></item>
164:         <item>&lt;b>Horas de prácticas</b>&lt;br></item>
165:         <item>&lt;b>Coste por hora</b>&lt;br></item>
166:         <item>&lt;b>Total</b>&lt;br></item>
167:         <item>&lt;b>Horas de seminario</b>&lt;br></item>
168:         <item>&lt;b>Coste por hora</b>&lt;br></item>
169:         <item>&lt;b>Total</b>&lt;br></item>
170:         <item>&lt;b>Horas de evaluación</b>&lt;br></item>
171:         <item>&lt;b>Coste por hora</b>&lt;br></item>
172:         <item>&lt;b>Total</b>&lt;br></item>
173:         <item>&lt;b>Elaboración de manuales docentes</b>&lt;br></item>
174:         <item>&lt;b>Coste por crédito</b>&lt;br></item>
175:         <item>&lt;b>Total</b>&lt;br></item>
176:         <item>&lt;b>Tutorías presenciales y de trabajos finales</b>&lt;br></item>
177:         <item>&lt;b>Coste por proyecto</b>&lt;br></item>
178:         <item>&lt;b>Total</b>&lt;br></item>
179:         <item>&lt;b>Tribunales para valoración de trabajos finales</b>&lt;br></item>
180:         <item>&lt;b>Coste por tribunal</b>&lt;br></item>
181:         <item>&lt;b>Total</b>&lt;br></item>
182:         <item>&lt;b>Coordinación de exámenes</b>&lt;br></item>
183:         <item>&lt;b>Coste por examen</b>&lt;br></item>
184:         <item>&lt;b>Total</b>&lt;br></item>
185:     </string-array>
186:     <string-array name="extra_ulpgc">
187:         <item>&lt;b>Directores</b></item>

```

```
188:     <item>&lt;b>Coste total de profesorado de la ULPGC</b></item>
189:     <item>&lt;b>Beneficios para la unidad organizadora</b></item>
190:     <item>&lt;b>Total retribuci3n bruta a profesorado de la ULPGC</b></it
em>
191: </string-array>
192: <string-array name="admin_costs">
193:     <item>admin-staff-qty</item>
194:     <item>admin-staff-cost</item>
195:     <item>admin-staff-sum</item>
196:     <item>scholar-qty</item>
197:     <item>scholar-cost</item>
198:     <item>scholar-sum</item>
199:     <item>specific-scholar-qty</item>
200:     <item>specific-scholar-cost</item>
201:     <item>specific-scholar-sum</item>
202:     <item>other-staff-qty</item>
203:     <item>other-staff-cost</item>
204:     <item>other-staff-sum</item>
205: </string-array>
206: <string-array name="coste_administracion">
207:     <item>&lt;b>Personal administrativo de apoyo</b></item>
208:     <item>&lt;b>Retribuci3n</b></item>
209:     <item>&lt;b>Total</b></item>
210:     <item>&lt;b>Becarios de colaboraci3n</b></item>
211:     <item>&lt;b>Retribuci3n</b></item>
212:     <item>&lt;b>Total</b></item>
213:     <item>&lt;b>Becarios espec3ficos</b></item>
214:     <item>&lt;b>Retribuci3n</b></item>
215:     <item>&lt;b>Total</b></item>
216:     <item>&lt;b>Otro tipo de personal</b></item>
217:     <item>&lt;b>Retribuci3n</b></item>
218:     <item>&lt;b>Total</b></item>
219: </string-array>
220: <string-array name="economic_info">
221:     <item>granted-students-qty</item>
222:     <item>granted-students-quota</item>
223:     <item>regular-students-qty</item>
224:     <item>regular-students-quota</item>
225:     <item>external-incomes-sum</item>
226:     <item>incomes-sum</item>
227:     <item>ulpgc-teaching-hours-qty</item>
228:     <item>ulpgc-teaching-hours-cost</item>
229:     <item>ulpgc-teaching-hours-sum</item>
230:     <item>ulpgc-seminar-hours-qty</item>
231:     <item>ulpgc-seminar-hours-cost</item>
232:     <item>ulpgc-seminar-hours-sum</item>
233:     <item>ulpgc-practice-hours-qty</item>
234:     <item>ulpgc-practice-hours-cost</item>
235:     <item>ulpgc-practice-hours-sum</item>
236:     <item>ulpgc-evaluation-hours-qty</item>
237:     <item>ulpgc-evaluation-hours-cost</item>
238:     <item>ulpgc-evaluation-hours-sum</item>
239:     <item>ulpgc-content-credits-qty</item>
240:     <item>ulpgc-content-credits-cost</item>
241:     <item>ulpgc-content-credits-sum</item>
242:     <item>ulpgc-tutorship-jobs-qty</item>
243:     <item>ulpgc-tutorship-jobs-cost</item>
244:     <item>ulpgc-tutorship-jobs-sum</item>
245:     <item>ulpgc-boards-qty</item>
246:     <item>ulpgc-boards-cost</item>
247:     <item>ulpgc-boards-sum</item>
248:     <item>ulpgc-exams-coordination-qty</item>
249:     <item>ulpgc-exams-coordination-cost</item>
```

250: <item>ulpgc-exams-coordination-sum</item>
251: <item>ulpgc-managers-qty</item>
252: <item>ulpgc-managers-cost</item>
253: <item>ulpgc-teachers-cost-sum</item>
254: <item>ulpgc-teachers-unit-profit</item>
255: <item>ulpgc-teachers-gross</item>
256: <item>teaching-hours-qty</item>
257: <item>teaching-hours-cost</item>
258: <item>teaching-hours-sum</item>
259: <item>seminar-hours-qty</item>
260: <item>seminar-hours-cost</item>
261: <item>seminar-hours-sum</item>
262: <item>practice-hours-qty</item>
263: <item>practice-hours-cost</item>
264: <item>practice-hours-sum</item>
265: <item>evaluation-hours-qty</item>
266: <item>evaluation-hours-cost</item>
267: <item>evaluation-hours-sum</item>
268: <item>content-credits-qty</item>
269: <item>content-credits-cost</item>
270: <item>content-credits-sum</item>
271: <item>tutorship-jobs-qty</item>
272: <item>tutorship-jobs-cost</item>
273: <item>tutorship-jobs-sum</item>
274: <item>boards-qty</item>
275: <item>boards-cost</item>
276: <item>boards-sum</item>
277: <item>exams-coordination-qty</item>
278: <item>exams-coordination-cost</item>
279: <item>exams-coordination-sum</item>
280: <item>teachers-misc-cost</item>
281: <item>teachers-misc-description</item>
282: <item>teachers-cost-sum</item>
283: <item>admin-staff-qty</item>
284: <item>admin-staff-cost</item>
285: <item>admin-staff-sum</item>
286: <item>admin-support-misc-cost</item>
287: <item>admin-support-misc-description</item>
288: <item>scholar-qty</item>
289: <item>scholar-cost</item>
290: <item>scholar-sum</item>
291: <item>specific-scholar-qty</item>
292: <item>specific-scholar-cost</item>
293: <item>specific-scholar-sum</item>
294: <item>other-staff-qty</item>
295: <item>other-staff-cost</item>
296: <item>other-staff-sum</item>
297: <item>presentation-report-cost</item>
298: <item>final-report-cost</item>
299: <item>information-point-cost</item>
300: <item>support-services-description</item>
301: <item>support-services-cost</item>
302: <item>admin-support-misc-description</item>
303: <item>admin-support-misc-cost</item>
304: <item>locomotion-cost</item>
305: <item>accommodation-cost</item>
306: <item>maintenance-cost</item>
307: <item>students-in-practice-cost</item>
308: <item>other-locomotion-cost</item>
309: <item>locomotion-sum</item>
310: <item>office-goods-cost</item>
311: <item>lab-goods-cost</item>
312: <item>computer-goods-cost</item>

```
313:      <item>reprographics-cost</item>
314:      <item>financial-goods-misc-cost</item>
315:      <item>financial-goods-sum</item>
316:      <item>bibliography-cost</item>
317:      <item>other-bibliography-cost</item>
318:      <item>bibliography-sum</item>
319:      <item>printed-advertising-cost</item>
320:      <item>media-advertising-cost</item>
321:      <item>advertising-sum</item>
322:      <item>opening-closure-cost</item>
323:      <item>translation-cost</item>
324:      <item>certificates-cost</item>
325:      <item>Seguro Obligatorio: 15â\202¬ por alumno</item>
326:      <item>EvaluaciÃ³n del tÃ­tulo por agencia externa: 550â\202¬</item>
327:      <item>virtual-campus-cost</item>
328:      <item>misc-misc-cost</item>
329:      <item>other-costs-sum</item>
330:      <item>partial-expenses-sum</item>
331:      <item>ulpgc-compensation</item>
332:      <item>institution-compensation</item>
333:      <item>administrator-compensation</item>
334:      <item>expenses-sum</item>
335:      <item>sum</item>
336:      <item>university-specific-degree-id</item>
337:    </string-array>
338:    <string-array name="overview">
339:      <item>incomes-sum</item>
340:      <item>partial-expenses-sum</item>
341:      <item>ulpgc-compensation</item>
342:      <item>institution-compensation</item>
343:      <item>administrator-compensation</item>
344:      <item>expenses-sum</item>
345:      <item>sum</item>
346:    </string-array>
347:    <string-array name="incomes">
348:      <item>granted-students-qty</item>
349:      <item>granted-students-quota</item>
350:      <item>regular-students-qty</item>
351:      <item>regular-students-quota</item>
352:      <item>external-incomes-sum</item>
353:      <item>incomes-sum</item>
354:    </string-array>
355:    <string-array name="incomeTags">
356:      <item>&lt;b>Nmero de estudiantes becados&lt;/b>&lt;br></item>
357:      <item>&lt;b>Precio de matrcula&lt;/b>&lt;br></item>
358:      <item>&lt;b>Nmero de estudiantes sin beca&lt;/b>&lt;br></item>
359:      <item>&lt;b>Precio de matrcula&lt;/b>&lt;br></item>
360:      <item>&lt;b>Total&lt;/b>&lt;br></item>
361:      <item>&lt;b>Ingresos externos&lt;/b>&lt;br></item>
362:      <item>&lt;b>Ingresos totales&lt;/b>&lt;br></item>
363:    </string-array>
364:    <string-array name="informacion_economica">
365:      <item>Nmero de estudiantes becados</item>
366:      <item>Precio de matrcula</item>
367:      <item>Nmero de estudiantes sin beca</item>
368:      <item>Precio de matrcula</item>
369:      <item>Ingresos externos</item>
370:      <item>Ingresos totales</item>
371:      <item>Nmero de horas de docencia</item>
372:      <item>Coste por hora</item>
373:      <item>Total</item>
374:      <item>Nmero de horas de seminario</item>
375:      <item>Coste por hora</item>
```

376: <item>Total</item>
377: <item>N mero de horas de pr cticas</item>
378: <item>Coste por hora</item>
379: <item>Total</item>
380: <item>N mero de horas de evaluaci n</item>
381: <item>Coste por hora</item>
382: <item>Total</item>
383: <item>Elaboraci n de manuales docentes</item>
384: <item>Coste por cr dito</item>
385: <item>Total</item>
386: <item>Tutor as presenciales y de trabajos finales</item>
387: <item>Coste por proyecto</item>
388: <item>Total</item>
389: <item>Tribunales para valoraci n de trabajos finales</item>
390: <item>Coste por tribunal</item>
391: <item>Total</item>
392: <item>Coordinaci n de ex menes</item>
393: <item>Coste por examen</item>
394: <item>Total</item>
395: <item>Directores</item>
396: <item>Coste por director</item>
397: <item>Coste total de profesorado de la ULPGC</item>
398: <item>Beneficios para la unidad organizadora</item>
399: <item>Total retribuci n bruta a profesorado de la ULPGC</item>
400: <item>N mero de horas de docencia</item>
401: <item>Coste por hora</item>
402: <item>Total</item>
403: <item>N mero de horas de seminario</item>
404: <item>Coste por hora</item>
405: <item>Total</item>
406: <item>N mero de horas de pr cticas</item>
407: <item>Coste por hora</item>
408: <item>Total</item>
409: <item>N mero de horas de evaluaci n</item>
410: <item>Coste por hora</item>
411: <item>Total</item>
412: <item>Elaboraci n de manuales docentes</item>
413: <item>Coste por cr dito</item>
414: <item>Total</item>
415: <item>Tutor as presenciales y de trabajos finales</item>
416: <item>Coste por proyecto</item>
417: <item>Total</item>
418: <item>Tribunales para valoraci n de trabajos finales</item>
419: <item>Coste por tribunal</item>
420: <item>Total</item>
421: <item>Coordinaci n de ex menes</item>
422: <item>Coste por examen</item>
423: <item>Total</item>
424: <item>Otros</item>
425: <item>Concepto</item>
426: <item>Coste total de profesorado externo</item>
427: <item>Personal administrativo de apoyo al t tulo</item>
428: <item>Coste</item>
429: <item>Total</item>
430: <item>Becarios de colaboraci n</item>
431: <item>Coste</item>
432: <item>Total</item>
433: <item>Becarios espec ficos</item>
434: <item>Coste</item>
435: <item>Total</item>
436: <item>Otro tipo de personal</item>
437: <item>Coste</item>
438: <item>Total</item>

```
439:      <item>Elaboraci3n de memoria de presentaci3n</item>
440:      <item>Memoria final</item>
441:      <item>Punto de informaci3n</item>
442:      <item>Otros</item>
443:      <item>Concepto</item>
444:      <item>Coste</item>
445:      <item>Otros</item>
446:      <item>Concepto</item>
447:      <item>Coste</item>
448:      <item>Locomoci3n de profesorado desplazado</item>
449:      <item>Alojamiento de profesorado desplazado</item>
450:      <item>Manutenci3n de profesorado desplazado</item>
451:      <item>Ayudas para estudiantes en pr3cticas</item>
452:      <item>Otros gastos</item>
453:      <item>Total</item>
454:      <item>Material de oficina</item>
455:      <item>Material de laboratorio</item>
456:      <item>Material inform3tico no inventariable</item>
457:      <item>Reprograf3a</item>
458:      <item>Otros</item>
459:      <item>Total</item>
460:      <item>Bibliograf3a</item>
461:      <item>Otros</item>
462:      <item>Total</item>
463:      <item>Material impreso</item>
464:      <item>Publicidad en prensa u otros medios</item>
465:      <item>Total</item>
466:      <item>Inauguraci3n y/o clausura</item>
467:      <item>Traducci3n simult3nea o sucesiva</item>
468:      <item>Certificados</item>
469:      <item>Seguro Obligatorio</item>
470:      <item>Evaluaci3n del t3tulo por agencia externa</item>
471:      <item>Campus Virtual</item>
472:      <item>Otros</item>
473:      <item>Total</item>
474:    </string-array>
475:    <string-array name="overviewExpenses">
476:      <item><b>Ingresos totales</b></item>
477:      <item><b>Suma de gastos parcial</b></item>
478:      <item><b>Compensaci3n a la ULPGC (5%)</b></item>
479:      <item><b>Compensaci3n al centro (5%)</b></item>
480:      <item><b>Compensaci3n al gestor econ3mico (5%)</b></item>
481:      <item><b>Total de Gastos</b></item>
482:      <item><b>Balance</b></item>
483:    </string-array>
484:    <string-array name="degree_teachers_info">
485:      <item>aasm-state</item>
486:      <item>teacher-id</item>
487:      <item>last-notification-at</item>
488:      <item>notifications-count</item>
489:      <item>university-specific-degree-id</item>
490:      <item>id</item>
491:    </string-array>
492:    <string-array name="generic_teachers_info">
493:      <item>name</item>
494:      <item>last-name</item>
495:      <item>degree</item>
496:      <item>email</item>
497:      <item>branch-of-knowledge</item>
498:      <item>professional-category</item>
499:      <item>graduate-in-ulpgc</item>
500:      <item>department</item>
501:      <item>institute</item>
```

```
502:         <item>active</item>
503:         <item>cv-abstract</item>
504:         <item>dni</item>
505:         <item>proposer-id</item>
506:         <item>type</item>
507:     </string-array>
508:     <string-array name="info_generica_profesores">
509:         <item>&lt;b>Titulaci3n:&lt;/b>&lt;br>&lt;/item>
510:         <item>&lt;b>E-mail:&lt;/b>&lt;br>&lt;/item>
511:         <item>&lt;b>Rama de conocimiento:&lt;/b>&lt;br>&lt;/item>
512:         <item>&lt;b>Categor3a profesional:&lt;/b>&lt;br>&lt;/item>
513:         <item>&lt;b>Titulado en la ULPGC:&lt;/b>&lt;br>&lt;/item>
514:         <item>&lt;b>Departamento de la ULPGC:&lt;/b>&lt;br>&lt;/item>
515:         <item>&lt;b>Entidad:&lt;/b>&lt;br>&lt;/item>
516:         <item>&lt;b>Activo:&lt;/b>&lt;br>&lt;/item>
517:         <item>&lt;b>Curr3culum Vitae:&lt;/b>&lt;br>&lt;/item>
518:         <item>&lt;b>Tipo&lt;b>&lt;/item>
519:     </string-array>
520:     <string-array name="subjects_info">
521:         <item>name</item>
522:         <item>course</item>
523:         <item>semester</item>
524:         <item>ects-credits</item>
525:         <item>start-date</item>
526:         <item>end-date</item>
527:         <item>kind-id</item>
528:         <item>id</item>
529:     </string-array>
530:     <string-array name="info_asignaturas">
531:         <item>Nombre</item>
532:         <item>Curso</item>
533:         <item>Semestre</item>
534:         <item>Cr3ditos ECTS</item>
535:         <item>Inicio de docencia</item>
536:         <item>Fin de docencia</item>
537:         <item>Tipo de asignatura</item>
538:         <item>Id</item>
539:     </string-array>
540:     <string-array name="complete_subject_info">
541:         <item>name</item>
542:         <item>course</item>
543:         <item>semester</item>
544:         <item>kind-id</item>
545:         <item>ects-credits</item>
546:         <item>start-date</item>
547:         <item>end-date</item>
548:         <item>branch-of-knowledge-id</item>
549:         <item>mode-id</item>
550:         <item>created-at</item>
551:         <item>updated-at</item>
552:         <item>class-hours</item>
553:         <item>practice-hours</item>
554:         <item>misc-activities-hours</item>
555:         <item>external-practices-hours</item>
556:         <item>exams-hours</item>
557:         <item>final-work-hours</item>
558:         <item>studying-hours</item>
559:         <item>hours-sum</item>
560:         <item>competences</item>
561:         <item>activities</item>
562:         <item>description</item>
563:         <item>assessment-system</item>
564:         <item>id</item>
```

```
565:     </string-array>
566:     <string-array name="completeSubjectTags">
567:         <item>Nombre</item>
568:         <item>Curso</item>
569:         <item>Semestre</item>
570:         <item>Tipo de asignatura</item>
571:         <item>Créditos ECTS</item>
572:         <item>Inicio de docencia</item>
573:         <item>Fin de docencia</item>
574:         <item>Rama de conocimiento</item>
575:         <item>Modalidad</item>
576:         <item>Creado</item>
577:         <item>Actualizado</item>
578:         <item>Clases teóricas</item>
579:         <item>Clases prácticas</item>
580:         <item>Actividades docentes complementarias</item>
581:         <item>Prácticas en empresas/instituciones</item>
582:         <item>Pruebas de evaluación</item>
583:         <item>Trabajo final</item>
584:         <item>Dedicadas al estudio</item>
585:         <item>Número de horas en total</item>
586:         <item>Competencias</item>
587:         <item>Actividades formativas</item>
588:         <item>Descripción de los contenidos</item>
589:         <item>Sistema de evaluación</item>
590:         <item>id</item>
591:     </string-array>
592:     <string-array name="report_info">
593:         <item>message</item>
594:     </string-array>
595:     <string-array name="basic_information">
596:         <item>name</item>
597:         <item>code</item>
598:         <item>lectures-start</item>
599:         <item>degree-type-id</item>
600:         <item>state</item>
601:         <item>id</item>
602:     </string-array>
603:     <string-array name="economic_tags">
604:         <item>ulpgc-teaching-hours-qty</item>
605:         <item>ulpgc-teaching-hours-cost</item>
606:         <item>ulpgc-teaching-hours-sum</item>
607:         <item>ulpgc-seminar-hours-qty</item>
608:         <item>ulpgc-seminar-hours-cost</item>
609:         <item>ulpgc-seminar-hours-sum</item>
610:         <item>ulpgc-practice-hours-qty</item>
611:         <item>ulpgc-practice-hours-cost</item>
612:         <item>ulpgc-practice-hours-sum</item>
613:         <item>ulpgc-evaluation-hours-qty</item>
614:         <item>ulpgc-evaluation-hours-cost</item>
615:         <item>ulpgc-evaluation-hours-sum</item>
616:         <item>ulpgc-content-credits-qty</item>
617:         <item>ulpgc-content-credits-cost</item>
618:         <item>ulpgc-content-credits-sum</item>
619:         <item>ulpgc-tutorship-jobs-qty</item>
620:         <item>ulpgc-tutorship-jobs-cost</item>
621:         <item>ulpgc-tutorship-jobs-sum</item>
622:         <item>ulpgc-boards-qty</item>
623:         <item>ulpgc-boards-cost</item>
624:         <item>ulpgc-boards-sum</item>
625:         <item>ulpgc-exams-coordination-qty</item>
626:         <item>ulpgc-exams-coordination-cost</item>
627:         <item>ulpgc-exams-coordination-sum</item>
```

628: <item>ulpgc-managers-qty</item>
629: <item>ulpgc-managers-cost</item>
630: <item>placeholder</item>
631: <item>ulpgc-teachers-cost-sum</item>
632: <item>placeholder</item>
633: <item>ulpgc-teachers-unit-profit</item>
634: <item>placeholder</item>
635: <item>ulpgc-teachers-gross</item>
636: <item>teaching-hours-qty</item>
637: <item>teaching-hours-cost</item>
638: <item>teaching-hours-sum</item>
639: <item>seminar-hours-qty</item>
640: <item>seminar-hours-cost</item>
641: <item>seminar-hours-sum</item>
642: <item>practice-hours-qty</item>
643: <item>practice-hours-cost</item>
644: <item>practice-hours-sum</item>
645: <item>evaluation-hours-qty</item>
646: <item>evaluation-hours-cost</item>
647: <item>evaluation-hours-sum</item>
648: <item>content-credits-qty</item>
649: <item>content-credits-cost</item>
650: <item>content-credits-sum</item>
651: <item>tutorship-jobs-qty</item>
652: <item>tutorship-jobs-cost</item>
653: <item>tutorship-jobs-sum</item>
654: <item>boards-qty</item>
655: <item>boards-cost</item>
656: <item>boards-sum</item>
657: <item>exams-coordination-qty</item>
658: <item>exams-coordination-cost</item>
659: <item>exams-coordination-sum</item>
660: <item>teachers-misc-cost</item>
661: <item>teachers-misc-description</item>
662: <item>placeholder</item>
663: <item>teachers-cost-sum</item>
664: <item>admin-staff-qty</item>
665: <item>admin-staff-cost</item>
666: <item>admin-staff-sum</item>
667: <item>scholar-qty</item>
668: <item>scholar-cost</item>
669: <item>scholar-sum</item>
670: <item>specific-scholar-qty</item>
671: <item>specific-scholar-cost</item>
672: <item>specific-scholar-sum</item>
673: <item>other-staff-qty</item>
674: <item>other-staff-cost</item>
675: <item>other-staff-sum</item>
676: <item>placeholder</item>
677: <item>presentation-report-cost</item>
678: <item>placeholder</item>
679: <item>final-report-cost</item>
680: <item>placeholder</item>
681: <item>information-point-cost</item>
682: <item>support-services-description</item>
683: <item>support-services-cost</item>
684: <item>admin-support-misc-description</item>
685: <item>admin-support-misc-cost</item>
686: <item>placeholder</item>
687: <item>locomotion-cost</item>
688: <item>placeholder</item>
689: <item>accommodation-cost</item>
690: <item>placeholder</item>

```
691:      <item>maintenance-cost</item>
692:      <item>placeholder</item>
693:      <item>students-in-practice-cost</item>
694:      <item>placeholder</item>
695:      <item>other-locomotion-cost</item>
696:      <item>placeholder</item>
697:      <item>locomotion-sum</item>
698:      <item>placeholder</item>
699:      <item>office-goods-cost</item>
700:      <item>placeholder</item>
701:      <item>lab-goods-cost</item>
702:      <item>placeholder</item>
703:      <item>computer-goods-cost</item>
704:      <item>placeholder</item>
705:      <item>reprographics-cost</item>
706:      <item>placeholder</item>
707:      <item>financial-goods-misc-cost</item>
708:      <item>placeholder</item>
709:      <item>financial-goods-sum</item>
710:      <item>placeholder</item>
711:      <item>bibliography-cost</item>
712:      <item>placeholder</item>
713:      <item>other-bibliography-cost</item>
714:      <item>placeholder</item>
715:      <item>bibliography-sum</item>
716:      <item>placeholder</item>
717:      <item>printed-advertising-cost</item>
718:      <item>placeholder</item>
719:      <item>media-advertising-cost</item>
720:      <item>placeholder</item>
721:      <item>advertising-sum</item>
722:      <item>placeholder</item>
723:      <item>opening-closure-cost</item>
724:      <item>placeholder</item>
725:      <item>translation-cost</item>
726:      <item>placeholder</item>
727:      <item>certificates-cost</item>
728:      <item>placeholder</item>
729:      <item>Seguro Obligatorio</item>
730:      <item>placeholder</item>
731:      <item>Evaluaci3n del t3-tulo por agencia externa</item>
732:      <item>placeholder</item>
733:      <item>virtual-campus-cost</item>
734:      <item>placeholder</item>
735:      <item>misc-misc-cost</item>
736:      <item>placeholder</item>
737:      <item>other-costs-sum</item>
738:      <item>placeholder</item>
739:      <item>expenses-sum</item>
740:    </string-array>
741:    <string-array name="expensesTags">
742:      <item>&lt;b>N3mero de horas de docencia&lt;br>&lt;/b>&lt;/item>
743:      <item>&lt;b>Coste por hora&lt;br>&lt;/b>&lt;/item>
744:      <item>&lt;b>Total&lt;br>&lt;/b>&lt;/item>
745:      <item>&lt;b>N3mero de horas de seminario&lt;br>&lt;/b>&lt;/item>
746:      <item>&lt;b>Coste por hora&lt;br>&lt;/b>&lt;/item>
747:      <item>&lt;b>Total&lt;br>&lt;/b>&lt;/item>
748:      <item>&lt;b>N3mero de horas de pr3cticas&lt;br>&lt;/b>&lt;/item>
749:      <item>&lt;b>Coste por hora&lt;br>&lt;/b>&lt;/item>
750:      <item>&lt;b>Total&lt;br>&lt;/b>&lt;/item>
751:      <item>&lt;b>N3mero de horas de evaluaci3n&lt;br>&lt;/b>&lt;/item>
752:      <item>&lt;b>Coste por hora&lt;br>&lt;/b>&lt;/item>
753:      <item>&lt;b>Total&lt;br>&lt;/b>&lt;/item>
```

```
754: <item><b>Elaboraci3n de manuales docentes</b><br></item>
755: <item><b>Coste por cr3dito</b><br><br></item>
756: <item><b>Total</b><br><br></item>
757: <item><b>Tutor3as presenciales y de trabajos finales</b><br><br>
</item>
758: <item><b>Coste por proyecto</b><br><br></item>
759: <item><b>Total</b><br><br></item>
760: <item><b>Tribunales para valoraci3n de trabajos finales</b><br>
gt;</item>
761: <item><b>Coste por tribunal</b><br><br></item>
762: <item><b>Total</b><br><br></item>
763: <item><b>Coordinaci3n de ex3menes</b><br><br></item>
764: <item><b>Coste por examen</b><br><br></item>
765: <item><b>Total</b><br><br></item>
766: <item><b>Directores</b><br><br></item>
767: <item><b>Coste</b><br><br></item>
768: <item><b>Coste de profesorado de la UPGC</b><br><br></item>
769: <item><b>Coste</b><br><br></item>
770: <item><b>Beneficios para la unidad organizadora</b><br><br></item>
>
771: <item><b>Coste</b><br><br></item>
772: <item><b>Total retribuci3n bruta a profesorado de la UPGC</b><br>
br></item>
773: <item><b>Total</b><br><br></item>
774: <item><b>N3mero de horas de docencia</b><br><br></item>
775: <item><b>Coste por hora</b><br><br></item>
776: <item><b>Total</b><br><br></item>
777: <item><b>N3mero de horas de seminario</b><br><br></item>
778: <item><b>Coste por hora</b><br><br></item>
779: <item><b>Total</b><br><br></item>
780: <item><b>N3mero de horas de pr3cticas</b><br><br></item>
781: <item><b>Coste por hora</b><br><br></item>
782: <item><b>Total</b><br><br></item>
783: <item><b>N3mero de horas de evaluaci3n</b><br><br></item>
784: <item><b>Coste por hora</b><br><br></item>
785: <item><b>Total</b><br><br></item>
786: <item><b>Elaboraci3n de manuales docentes</b><br><br></item>
787: <item><b>Coste por cr3dito</b><br><br></item>
788: <item><b>Total</b><br><br></item>
789: <item><b>Tutor3as presenciales y de trabajos finales</b><br><br>
</item>
790: <item><b>Coste por proyecto</b><br><br></item>
791: <item><b>Total</b><br><br></item>
792: <item><b>Tribunales para valoraci3n de trabajos finales</b><br>
gt;</item>
793: <item><b>Coste por tribunal</b><br><br></item>
794: <item><b>Total</b><br><br></item>
795: <item><b>Coordinaci3n de ex3menes</b><br><br></item>
796: <item><b>Coste por examen</b><br><br></item>
797: <item><b>Total</b><br><br></item>
798: <item><b>Otros. Concepto</b><br><br></item>
799: <item><b>Total</b><br><br></item>
800: <item><b>Coste de profesorado externo</b><br><br></item>
801: <item><b>Total</b><br><br></item>
802: <item><b>Personal administrativo de apoyo al t3-tulo</b><br><br><
/ item>
803: <item><b>Coste</b><br><br><br></item>
804: <item><b>Total</b><br><br><br></item>
805: <item><b>Becarios de colaboraci3n</b><br><br></item>
806: <item><b>Coste</b><br><br><br></item>
807: <item><b>Total</b><br><br><br></item>
808: <item><b>Becarios espec3ficos</b><br><br></item>
809: <item><b>Coste</b><br><br><br></item>
```

```
810: <item><b>Total</b></item>
811: <item><b>Otro tipo de personal</b></item>
812: <item><b>Coste</b></item>
813: <item><b>Total</b></item>
814: <item><b>Elaboraci3n de memoria de presentaci3n</b></item>
em>
815: <item><b>Coste</b></item>
816: <item><b>Memoria final</b></item>
817: <item><b>Coste</b></item>
818: <item><b>Punto de informaci3n</b></item>
819: <item><b>Coste</b></item>
820: <item><b>Servicios de empresas de apoyo</b></item>
821: <item><b>Coste</b></item>
822: <item><b>Otros</b></item>
823: <item><b>Coste</b></item>
824: <item><b>Locomoci3n de profesorado desplazado</b></item>
825: <item><b>Coste</b></item>
826: <item><b>Alojamiento de profesorado desplazado</b></item>
827: <item><b>Coste</b></item>
828: <item><b>Manutenci3n de profesorado desplazado</b></item>
>
829: <item><b>Coste</b></item>
830: <item><b>Ayudas para estudiantes en pr3cticas</b></item>
831: <item><b>Coste</b></item>
832: <item><b>Otros gastos</b></item>
833: <item><b>Coste</b></item>
834: <item><b>Gastos en locomoci3n</b></item>
835: <item><b>Total</b></item>
836: <item><b>Material de oficina</b></item>
837: <item><b>Coste</b></item>
838: <item><b>Material de laboratorio</b></item>
839: <item><b>Coste</b></item>
840: <item><b>Material inform3tico no inventariable</b></item>
>
841: <item><b>Coste</b></item>
842: <item><b>Reprograf3a</b></item>
843: <item><b>Coste</b></item>
844: <item><b>Otros</b></item>
845: <item><b>Coste</b></item>
846: <item><b>Gastos de material</b></item>
847: <item><b>Total</b></item>
848: <item><b>Bibliograf3a</b></item>
849: <item><b>Coste</b></item>
850: <item><b>Otros</b></item>
851: <item><b>Coste</b></item>
852: <item><b>Gastos en bibliograf3a</b></item>
853: <item><b>Total</b></item>
854: <item><b>Material impreso</b></item>
855: <item><b>Coste</b></item>
856: <item><b>Publicidad en prensa u otros medios</b></item>
857: <item><b>Coste</b></item>
858: <item><b>Gastos en publicidad</b></item>
859: <item><b>Total</b></item>
860: <item><b>Inauguraci3n y/o clausura</b></item>
861: <item><b>Coste</b></item>
862: <item><b>Traducci3n simult3nea o sucesiva</b></item>
863: <item><b>Coste</b></item>
864: <item><b>Certificados</b></item>
865: <item><b>Coste</b></item>
866: <item><b>Seguro Obligatorio</b></item>
867: <item><b>Coste</b></item>
868: <item><b>Evaluaci3n del t3ulo por agencia externa</b></item>
```

```

869:      <item><b>Coste</b><br></item>
870:      <item><b>Campus Virtual</b><br></item>
871:      <item><b>Coste</b><br></item>
872:      <item><b>Otros</b><br></item>
873:      <item><b>Coste</b><br></item>
874:      <item><b>Apartado de otros gastos</b><br></item>
875:      <item><b>Coste</b><br></item>
876:      <item><b>Importe de partida de gastos</b><br></item>
877:      <item><b>Total</b><br></item>
878:    </string-array>
879:    <string-array name="periods">
880:      <item>first-rectification-from-date</item>
881:      <item>first-rectification-to-date</item>
882:      <item>request-from-date</item>
883:      <item>request-to-date</item>
884:      <item>second-rectification-from-date</item>
885:      <item>second-rectification-to-date</item>
886:      <item>year</item>
887:      <item>id</item>
888:    </string-array>
889:    <string-array name="institutions">
890:      <item>Centro de Formaci3n Continua</item>
891:      <item>Escuela de Arquitectura</item>
892:      <item>Escuela de Ingenier3a de Telecomunicaci3n y Electr3nica</item>
893:      <item>Escuela de Ingenier3a Inform3tica</item>
894:      <item>Escuela de Ingenier3as Industriales y Civiles</item>
895:      <item>Escuela Universitaria Adscrita de Turismo de Lanzarote</item>
896:      <item>Estructura de Teleformaci3n</item>
897:      <item>Facultad de Ciencias de la Actividad F3sica y el Deporte</item>
898:      <item>Facultad de Ciencias de La Salud</item>
899:      <item>Facultad de Ciencias del Mar</item>
900:      <item>Facultad de Ciencias Jur3dicas</item>
901:      <item>Facultad de Econom3a, Empresa y Turismo</item>
902:      <item>Facultad de Filolog3a</item>
903:      <item>Facultad de Formaci3n del Profesorado</item>
904:      <item>Facultad de Geograf3a e Historia</item>
905:      <item>Facultad de Traducci3n y Interpretaci3n</item>
906:      <item>Facultad de Veterinaria</item>
907:      <item>FULP</item>
908:      <item>Instituto para el desarrollo tecnol3gico y la innovaci3n en comunicacio
nes</item>
909:      <item>Instituto Universitario de Ciencias y Tecnolog3as Cibern3ticas</item>
910:      <item>Instituto Universitario de Microelectr3nica Aplicada</item>
911:      <item>Instituto Universitario de Oceanograf3a y Cambio Global</item>
912:      <item>Instituto Universitario de Sanidad Animal y Seguridad Alimentaria</item>
913:      <item>Instituto Universitario de Sistemas Inteligentes y Aplicaciones Num3rica
s en Ingenier3a</item>
914:      <item>Instituto Universitario de Turismo y Desarrollo Sostenible (TYDES)</item>
915:    </string-array>
916:    <string-array name="institutions_addresses">
917:      <item>Juan de Quesada n330
918: CP 35001</item>
919:      <item>Edificio Arquitectura
920: 35017 - Las Palmas de Gran Canaria</item>
921:      <item>Edificio Electr3nica y Telecomunicaci3n
922: 35017 - Las Palmas de Gran Canaria</item>
923:      <item>Edificio Inform3tica y Matem3ticas
924: 35017 - Las Palmas de Gran Canaria</item>
925:      <item>Edificio Ingenier3as
926: 35017 - Las Palmas de Gran Canaria</item>
927:      <item>Centro Adscrito de Turismo de Lanzarote
928: 35507 - Teguiise</item>
929:      <item>Sede Institucional

```

930: 35001 - Las Palmas de Gran Canaria</item>
931: <item>Edificio Educaci3n F3sica</item>
932: 35017 - Las Palmas de Gran Canaria</item>
933: <item>Edificio Ciencias de la Salud</item>
934: 35016 - Las Palmas de Gran Canaria</item>
935: <item>Edificio Ciencias B3sicas</item>
936: 35017 - Las Palmas de Gran Canaria</item>
937: <item>Edificio Ciencias Jur3dicas</item>
938: 35017 - Las Palmas de Gran Canaria</item>
939: <item>Edificio CC. Econ3micas y Empresariales</item>
940: 35017 - Las Palmas de Gran Canaria</item>
941: <item>Edificio Humanidades</item>
942: 35003 - Las Palmas de Gran Canaria</item>
943: <item>Edificio Formaci3n del Profesorado</item>
944: 35004 - Las Palmas de Gran Canaria</item>
945: <item>Edificio Humanidades</item>
946: 35003 - Las Palmas de Gran Canaria</item>
947: <item>Edificio Humanidades</item>
948: 35003 - Las Palmas de Gran Canaria</item>
949: <item>Edificio Veterinaria</item>
950: 35416 - Arucas</item>
951: <item>C/ Juan de Quesada, 30. Las Palmas de Gran Canaria, 35001</item>
952: <item>Parque Cient3fico Tecnol3gico de la ULPGC</item>
953: Las Palmas (Spain)</item>
954: <item>Edificio Parque Cient3fico Tecnol3gico</item>
955: 35017 - Las Palmas de Gran Canaria</item>
956: <item>Edificio Parque Cient3fico Tecnol3gico</item>
957: 35017 - Las Palmas de Gran Canaria</item>
958: <item>C/ Practicante Ignacio Rodr3guez, s/n. Edificio Polivalente II del Campu
s Universitario de Tafira</item>
959: CP 35017 Las Palmas de Gran Canaria, Espa3a</item>
960: <item>Edificio Parque Cient3fico Tecnol3gico</item>
961: 35017 - Las Palmas de Gran Canaria</item>
962: <item>Edificio Parque Cient3fico Tecnol3gico</item>
963: 35017 - Las Palmas de Gran Canaria</item>
964: <item>-</item>
965: </string-array>
966: <string-array name="institutions_id">
967: <item>24</item>
968: <item>1</item>
969: <item>2</item>
970: <item>14</item>
971: <item>3</item>
972: <item>4</item>
973: <item>5</item>
974: <item>6</item>
975: <item>7</item>
976: <item>8</item>
977: <item>10</item>
978: <item>9</item>
979: <item>11</item>
980: <item>12</item>
981: <item>13</item>
982: <item>15</item>
983: <item>16</item>
984: <item>17</item>
985: <item>22</item>
986: <item>18</item>
987: <item>19</item>
988: <item>25</item>
989: <item>21</item>
990: <item>20</item>
991: <item>23</item>

```
992:     </string-array>
993:     <string-array name="institutions_location">
994:         <item>Juan de Quesada nÂ°30</item>
995:         <item>Escuela de Arquitectura</item>
996:         <item>Escuela de IngenierÃ-a de TelecomunicaciÃ³n y ElectrÃ³nica</item>
997:         <item>Escuela de IngenierÃ-a InformÃ;tica</item>
998:         <item>Escuela de IngenierÃ-as Industriales y Civiles</item>
999:         <item>Escuela Universitaria Adscrita de Turismo de Lanzarote</item>
1000:        <item>Estructura de TeleformaciÃ³n</item>
1001:        <item>Facultad de Ciencias de la Actividad FÃ-sica y el Deporte</item>
1002:        <item>Facultad de Ciencias de La Salud</item>
1003:        <item>Facultad de Ciencias del Mar</item>
1004:        <item>Facultad de Ciencias JurÃ-dicas</item>
1005:        <item>Facultad de EconomÃ-a, Empresa y Turismo</item>
1006:        <item>Facultad de FilologÃ-a</item>
1007:        <item>Facultad de FormaciÃ³n del Profesorado</item>
1008:        <item>Facultad de GeografÃ-a e Historia</item>
1009:        <item>Facultad de TraducciÃ³n y InterpretaciÃ³n</item>
1010:        <item>Facultad de Veterinaria</item>
1011:        <item>FundaciÃ³n Universitaria</item>
1012:        <item>Instituto para el desarrollo tecnolÃ³gico y la innovaciÃ³n en comunicacio
nes</item>
1013:        <item>Instituto Universitario de Ciencias y TecnologÃ-as CibernÃ©ticas</item>
1014:        <item>Instituto Universitario de MicroelectrÃ³nica Aplicada</item>
1015:        <item>Edificio Polivalente II del Campus Universitario de Tafira</item>
1016:        <item>Instituto Universitario de Sanidad Animal y Seguridad Alimentaria</item>
1017:        <item>Instituto Universitario de Sistemas Inteligentes y Aplicaciones NumÃ©rica
s en IngenierÃ-a</item>
1018:        <item>Instituto Universitario de Turismo y Desarrollo Sostenible (TYDES)</item>
1019:     </string-array>
1020:
1021: </resources>
```

```
1: <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:background="@drawable/background_colours" >
5:
6:     <LinearLayout
7:         android:layout_width="match_parent"
8:         android:layout_height="wrap_content"
9:         android:orientation="vertical"
10:        android:weightSum="16" >
11:
12:        <TextView
13:            android:id="@+id/period"
14:            android:layout_width="fill_parent"
15:            android:layout_height="wrap_content"
16:            android:layout_marginTop="5dp"
17:            android:background="@drawable/blue_border"
18:            android:gravity="center"
19:            android:textStyle="bold" />
20:
21:        <EditText
22:            android:id="@+id/degree_denomination"
23:            android:layout_width="fill_parent"
24:            android:layout_height="0dp"
25:            android:layout_marginTop="@dimen/activity_horizontal_margin"
26:            android:layout_weight="1"
27:            android:hint="@string/degree_denomination" />
28:
29:        <Spinner
30:            android:id="@+id/type_spinner"
31:            android:layout_width="fill_parent"
32:            android:layout_height="0dp"
33:            android:layout_weight="1"
34:            android:background="@drawable/grey_buttons"
35:            android:prompt="@string/tipo" />
36:
37:        <Spinner
38:            android:id="@+id/mode_spinner"
39:            android:layout_width="fill_parent"
40:            android:layout_height="0dp"
41:            android:layout_weight="1"
42:            android:background="@drawable/grey_buttons"
43:            android:prompt="@string/mode" />
44:
45:        <Spinner
46:            android:id="@+id/oriented_spinner"
47:            android:layout_width="fill_parent"
48:            android:layout_height="0dp"
49:            android:layout_weight="1"
50:            android:background="@drawable/grey_buttons"
51:            android:prompt="@string/oriented" />
52:
53:        <Spinner
54:            android:id="@+id/knowledge_spinner"
55:            android:layout_width="fill_parent"
56:            android:layout_height="0dp"
57:            android:layout_weight="1"
58:            android:background="@drawable/grey_buttons"
59:            android:prompt="@string/knowledge_branch" />
60:
61:        <EditText
62:            android:id="@+id/degree_academic_director"
63:            android:layout_width="fill_parent"
```

```
64:         android:layout_height="0dp"
65:         android:layout_width="1"
66:         android:hint="@string/degree_academic_director"
67:         android:imeOptions="actionNext"
68:         android:singleLine="true" >
69:     </EditText>
70:
71:     <EditText
72:         android:id="@+id/degree_proposed_by"
73:         android:layout_width="fill_parent"
74:         android:layout_height="0dp"
75:         android:layout_weight="1"
76:         android:hint="@string/degree_proposed_by"
77:         android:imeOptions="actionNext"
78:         android:singleLine="true" />
79:
80:     <EditText
81:         android:id="@+id/degree_units"
82:         android:layout_width="fill_parent"
83:         android:layout_height="0dp"
84:         android:layout_weight="1"
85:         android:hint="@string/degree_units"
86:         android:imeOptions="actionNext"
87:         android:singleLine="true" />
88:
89:     <LinearLayout
90:         android:layout_width="match_parent"
91:         android:layout_height="wrap_content"
92:         android:orientation="horizontal"
93:         android:weightSum="2" >
94:
95:         <EditText
96:             android:id="@+id/degree_min_alumns"
97:             android:layout_width="0dp"
98:             android:layout_height="match_parent"
99:             android:layout_weight="1"
100:            android:hint="@string/degree_min_alumns"
101:            android:imeOptions="actionNext"
102:            android:inputType="number"
103:            android:singleLine="true" />
104:
105:         <EditText
106:             android:id="@+id/degree_max_alumns"
107:             android:layout_width="0dp"
108:             android:layout_height="match_parent"
109:             android:layout_weight="1"
110:             android:ems="10"
111:             android:hint="@string/degree_max_alumns"
112:             android:imeOptions="actionNext"
113:             android:inputType="number"
114:             android:singleLine="true" >
115:
116:             <requestFocus />
117:         </EditText>
118:     </LinearLayout>
119:
120:     <EditText
121:         android:id="@+id/degree_min_credits"
122:         android:layout_width="fill_parent"
123:         android:layout_height="0dp"
124:         android:layout_weight="1"
125:         android:hint="@string/degree_min_credits"
126:         android:imeOptions="actionNext"
```

```
127:         android:inputType="number"
128:         android:singleLine="true" />
129:
130:     <EditText
131:         android:id="@+id/degree_economic_manager"
132:         android:layout_width="fill_parent"
133:         android:layout_height="0dp"
134:         android:layout_weight="1"
135:         android:hint="@string/degree_economic_manager"
136:         android:imeOptions="actionNext"
137:         android:singleLine="true" />
138:
139:     <EditText
140:         android:id="@+id/degree_competences"
141:         android:layout_width="fill_parent"
142:         android:layout_height="0dp"
143:         android:layout_weight="3"
144:         android:hint="@string/degree_competences"
145:         android:imeOptions="actionNext"
146:         android:inputType="textCapSentences|textMultiLine" />
147:
148:     <TextView
149:         android:layout_width="fill_parent"
150:         android:layout_height="wrap_content"
151:         android:layout_marginTop="5dp"
152:         android:gravity="left"
153:         android:text="@string/institutions"
154:         android:textStyle="bold" />
155:
156:     <Spinner
157:         android:id="@+id/institutions_spinner"
158:         android:layout_width="fill_parent"
159:         android:layout_height="0dp"
160:         android:layout_weight="1"
161:         android:background="@drawable/grey_buttons"
162:         android:prompt="@string/institutions" />
163:
164:     <TextView
165:         android:layout_width="fill_parent"
166:         android:layout_height="wrap_content"
167:         android:layout_marginTop="5dp"
168:         android:gravity="left"
169:         android:text="@string/teaching_address"
170:         android:textStyle="bold" />
171:
172:     <EditText
173:         android:id="@+id/teaching_address"
174:         android:layout_width="fill_parent"
175:         android:layout_height="0dp"
176:         android:layout_weight="3"
177:         android:hint="@string/teaching_address"
178:         android:imeOptions="actionDone"
179:         android:inputType="textCapSentences|textMultiLine" />
180:
181:     <TextView
182:         android:layout_width="fill_parent"
183:         android:layout_height="wrap_content"
184:         android:layout_marginTop="5dp"
185:         android:gravity="left"
186:         android:text="@string/teaching_outside"
187:         android:textStyle="bold" />
188:
189:     <CheckBox
```

```
190:         android:id="@+id/teaching_outside"
191:         android:layout_width="wrap_content"
192:         android:layout_height="wrap_content" />
193:
194:     <Button
195:         android:id="@+id/send_button"
196:         android:layout_width="fill_parent"
197:         android:layout_height="0dp"
198:         android:layout_marginTop="@dimen/activity_horizontal_margin"
199:         android:layout_weight="1"
200:         android:background="@drawable/grey_buttons"
201:         android:onClick="sendNewProposal"
202:         android:text="@string/send"
203:         android:textSize="24sp" />
204: </LinearLayout>
205:
206: </ScrollView>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:     android:layout_width="fill_parent"
3:     android:layout_height="fill_parent"
4:     android:orientation="horizontal"
5:     android:weightSum="3" >
6:
7:     <LinearLayout
8:         android:layout_width="0dp"
9:         android:layout_height="match_parent"
10:        android:layout_weight="1"
11:        android:orientation="vertical" >
12:
13:        <ImageView
14:            android:layout_width="match_parent"
15:            android:layout_height="match_parent"
16:            android:contentDescription="@string/desc"
17:            android:src="@drawable/logo_vtd_rgb" />
18:    </LinearLayout>
19:
20:    <View
21:        android:layout_width="1dp"
22:        android:layout_height="fill_parent"
23:        android:background="#3A4978" />
24:
25:    <LinearLayout
26:        android:layout_width="0dp"
27:        android:layout_height="match_parent"
28:        android:layout_weight="2"
29:        android:orientation="vertical" >
30:
31:        <LinearLayout
32:            android:layout_width="match_parent"
33:            android:layout_height="match_parent"
34:            android:background="@drawable/gradient_background"
35:            android:orientation="vertical"
36:            android:weightSum="7" >
37:
38:            <TextView
39:                android:id="@+id/login_manager"
40:                android:layout_width="wrap_content"
41:                android:layout_height="wrap_content"
42:                android:layout_gravity="center"
43:                android:layout_weight="1"
44:                android:textSize="18sp" />
45:
46:            <EditText
47:                android:id="@+id/login_message"
48:                android:layout_width="340dp"
49:                android:layout_height="wrap_content"
50:                android:layout_gravity="center"
51:                android:hint="@string/login_string" />
52:
53:            <EditText
54:                android:id="@+id/pass_message"
55:                android:layout_width="339dp"
56:                android:layout_height="wrap_content"
57:                android:layout_gravity="center"
58:                android:hint="@string/pass_string"
59:                android:inputType="textPassword" />
60:
61:            <CheckBox
62:                android:id="@+id/rememberButton"
63:                android:layout_width="wrap_content"
```

```
64:         android:layout_height="wrap_content"
65:         android:layout_gravity="center"
66:         android:layout_weight="1"
67:         android:text="@string/rememberMe_message" />
68:
69:     <RelativeLayout
70:         android:layout_width="match_parent"
71:         android:layout_height="wrap_content"
72:         android:layout_weight="3"
73:         android:gravity="center_vertical|center_horizontal|center" >
74:
75:         <Button
76:             android:id="@+id/loginButton"
77:             android:layout_width="100dp"
78:             android:layout_height="60dp"
79:             android:layout_centerHorizontal="true"
80:             android:layout_centerVertical="true"
81:             android:background="@drawable/grey_buttons"
82:             android:text="@string/button_login"
83:             android:textColor="#000000" />
84:
85:     </RelativeLayout>
86: </LinearLayout>
87: </LinearLayout>
88:
89: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/lightblue_border"
6:   android:baselineAligned="false"
7:   android:orientation="horizontal"
8:   android:weightSum="2" >
9:
10:  <fragment
11:    android:id="@+id/search_degrees_fragment"
12:    android:layout_width="0dp"
13:    android:layout_height="match_parent"
14:    android:layout_weight="1"
15:    class="es.ulpgc.titulosproprios.gestion.ubay.SearchDegreesFragment" />
16:
17:  <FrameLayout
18:    android:id="@+id/fragment_container"
19:    android:layout_width="0dp"
20:    android:layout_height="match_parent"
21:    android:layout_weight="1" />
22:
23: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/lightblue_border"
6:   android:orientation="vertical"
7:   tools:context=".SingleDegreeMain" >
8:
9:   <Button
10:     android:layout_width="match_parent"
11:     android:layout_height="0dp"
12:     android:layout_marginTop="5dp"
13:     android:layout_weight="1"
14:     android:background="@drawable/mybutton_layout"
15:     android:gravity="center"
16:     android:onClick="academicInfo"
17:     android:text="@string/academic_info"
18:     android:textStyle="bold" />
19:
20:   <Button
21:     android:layout_width="match_parent"
22:     android:layout_height="0dp"
23:     android:layout_weight="1"
24:     android:background="@drawable/mybutton_layout"
25:     android:gravity="center"
26:     android:onClick="subjectsInfo"
27:     android:text="@string/subjects"
28:     android:textStyle="bold" />
29:
30:   <Button
31:     android:layout_width="match_parent"
32:     android:layout_height="0dp"
33:     android:layout_weight="1"
34:     android:background="@drawable/mybutton_layout"
35:     android:gravity="center"
36:     android:onClick="teachersInfo"
37:     android:text="@string/teachers"
38:     android:textStyle="bold" />
39:
40:   <Button
41:     android:layout_width="match_parent"
42:     android:layout_height="0dp"
43:     android:layout_weight="1"
44:     android:background="@drawable/mybutton_layout"
45:     android:gravity="center"
46:     android:onClick="economicInfo"
47:     android:text="@string/economic_info"
48:     android:textStyle="bold" />
49:
50:   <Button
51:     android:layout_width="match_parent"
52:     android:layout_height="0dp"
53:     android:layout_weight="1"
54:     android:background="@drawable/mybutton_layout"
55:     android:gravity="center"
56:     android:onClick="reportInfo"
57:     android:text="@string/report"
58:     android:textStyle="bold" />
59:
60:   <Button
61:     android:layout_width="match_parent"
62:     android:layout_height="0dp"
63:     android:layout_weight="1"
```

```
64:         android:background="@drawable/mybutton_layout"
65:         android:gravity="center"
66:         android:onClick="sendDegreeToEvaluate"
67:         android:text="@string/send_to_evaluate"
68:         android:textStyle="bold" />
69:
70:     <Button
71:         android:layout_width="match_parent"
72:         android:layout_height="0dp"
73:         android:layout_weight="1"
74:         android:background="@drawable/mybutton_layout"
75:         android:gravity="center"
76:         android:onClick="downloadDegree"
77:         android:text="@string/download_degree"
78:         android:textStyle="bold" />
79: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/lightblue_border"
6:   android:baselineAligned="false"
7:   android:weightSum="3"
8:   android:orientation="horizontal" >
9:
10:  <fragment
11:    android:id="@+id/single_degree_menu_fragment"
12:    android:layout_width="0px"
13:    android:layout_height="match_parent"
14:    android:layout_weight="1"
15:    class="es.ulpgc.titulospropios.gestion.ubay.DegreeMainMenuFragment" />
16:
17:  <FrameLayout
18:    android:id="@+id/central_fragment"
19:    android:layout_width="0px"
20:    android:layout_height="match_parent"
21:    android:layout_weight="1" />
22:
23:  <FrameLayout
24:    android:id="@+id/lateral_fragment"
25:    android:layout_width="0px"
26:    android:layout_height="match_parent"
27:    android:layout_weight="1" />
28:
29:  <FrameLayout
30:    android:id="@+id/full_screen_fragment"
31:    android:layout_width="0px"
32:    android:layout_height="match_parent"
33:    android:layout_weight="0" />
34:
35: </LinearLayout>
```

```
1: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2:   xmlns:tools="http://schemas.android.com/tools"
3:   android:layout_width="match_parent"
4:   android:layout_height="match_parent"
5:   android:background="@drawable/lightblue_border"
6:   android:baselineAligned="false"
7:   android:orientation="horizontal" >
8:
9:   <fragment
10:    android:id="@+id/single_degree_menu_fragment"
11:    android:layout_width="0px"
12:    android:layout_height="match_parent"
13:    android:layout_weight="1"
14:    class="es.ubay.degreemanagerulpgc.SingleDegreeMenuFragment" />
15:
16:   <FrameLayout
17:    android:id="@+id/central_fragment"
18:    android:layout_width="0px"
19:    android:layout_height="match_parent"
20:    android:layout_weight="1" />
21:
22:   <FrameLayout
23:    android:id="@+id/lateral_fragment"
24:    android:layout_width="0px"
25:    android:layout_height="match_parent"
26:    android:layout_weight="1" />
27:
28: </LinearLayout>
```

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     xmlns:tools="http://schemas.android.com/tools"
4:     android:layout_width="match_parent"
5:     android:layout_height="match_parent"
6:     android:orientation="horizontal"
7:     android:weightSum="3" >
8:
9:     <LinearLayout
10:         android:layout_width="0dp"
11:         android:layout_height="match_parent"
12:         android:layout_weight="1"
13:         android:orientation="vertical" >
14:
15:         <ImageView
16:             android:layout_width="match_parent"
17:             android:layout_height="match_parent"
18:             android:contentDescription="@string/desc"
19:             android:src="@drawable/logo_vtd_rgb" />
20:     </LinearLayout>
21:
22:     <LinearLayout
23:         android:layout_width="0dp"
24:         android:layout_height="match_parent"
25:         android:layout_weight="2"
26:         android:orientation="horizontal"
27:         android:paddingBottom="5dp"
28:         android:paddingLeft="5dp"
29:         android:paddingRight="5dp"
30:         android:weightSum="2" >
31:
32:         <Button
33:             android:id="@+id/button_teacher"
34:             android:layout_width="0dp"
35:             android:layout_height="match_parent"
36:             android:layout_weight="1"
37:             android:background="@drawable/mybutton_layout"
38:             android:text="@string/accessMessageTeacher"
39:             android:textColor="#000000"
40:             android:textSize="16sp"
41:             android:textStyle="bold"
42:             android:typeface="sans" />
43:
44:         <Button
45:             android:id="@+id/button_manager"
46:             android:layout_width="0dp"
47:             android:layout_height="match_parent"
48:             android:layout_weight="1"
49:             android:background="@drawable/mybutton_layout"
50:             android:text="@string/accessMessageManager"
51:             android:textColor="#000000"
52:             android:textSize="16sp"
53:             android:textStyle="bold" />
54:     </LinearLayout>
55:
56: </LinearLayout>
```

12.3 *Parche Servidor Ruby on Rails*

1: From 43b3ab3bc39d077fa61a99e50e3ba115bb1b553f Mon Sep 17 00:00:00 2001
2: From: electr0naut <u.gonzalez@gmail.com>
3: Date: Mon, 11 Mar 2013 13:02:59 +0000
4: Subject: [PATCH] changes needed for android app usage

5:
6: ---
7: .../additional_informations_controller.rb | 2 +-
8: app/controllers/degree_teachers_controller.rb | 1 +
9: app/controllers/economic_reports_controller.rb | 2 +
10: app/controllers/general_informations_controller.rb | 2 +-
11: app/controllers/incomes_controller.rb | 1 +
12: app/controllers/issues_controller.rb | 2 +-
13: app/controllers/subjects_controller.rb | 2 +
14: .../teacher_authorizations_controller.rb | 2 +-
15: app/controllers/teacher_sessions_controller.rb | 1 +
16: app/controllers/teachers_controller.rb | 2 +-
17: app/controllers/user_sessions_controller.rb | 2 +
18: db/schema.rb | 44 ++++++++-----

19: 12 files changed, 36 insertions(+), 27 deletions(-)

21: diff --git a/app/controllers/additional_informations_controller.rb b/app/controllers/additional_informations_controller.rb

22: index 15fd4b2..9132153 100644
23: --- a/app/controllers/additional_informations_controller.rb
24: +++ b/app/controllers/additional_informations_controller.rb
25: @@ -3,7 +3,7 @@
26: class AdditionalInformationsController < InheritedResources::Base
27: singleton_belongs_to :university_specific_degree
28: actions :show, :edit, :update
29: - respond_to :html
30: + respond_to :html, :xml
31:
32: load_and_authorize_resource :class => "UniversitySpecificDegree"

34: diff --git a/app/controllers/degree_teachers_controller.rb b/app/controllers/degree_teachers_controller.rb

35: index 59df0c3..6511233 100644
36: --- a/app/controllers/degree_teachers_controller.rb
37: +++ b/app/controllers/degree_teachers_controller.rb
38: @@ -6,6 +6,7 @@ class DegreeTeachersController < InheritedResources::Base
39:
40: load_and_authorize_resource :university_specific_degree
41: load_and_authorize_resource :degree_teacher, :through => :university_specific_degree
42: + respond_to :html, :xml
43:
44: def create
45: @university_specific_degree = UniversitySpecificDegree.find(params[:university_specific_degree_id])

46: diff --git a/app/controllers/economic_reports_controller.rb b/app/controllers/economic_reports_controller.rb

47: index 6cd3ce9..16df3e9 100644
48: --- a/app/controllers/economic_reports_controller.rb
49: +++ b/app/controllers/economic_reports_controller.rb
50: @@ -4,10 +4,12 @@ class EconomicReportsController < InheritedResources::Base
51: authorize_resource
52: singleton_belongs_to :university_specific_degree
53: actions :all, :except => [:index, :destroy]
54: + respond_to :html, :xml
55:
56: load_and_authorize_resource :university_specific_degree
57: load_and_authorize_resource :economic_report, :through => :university_specific_degree

58:

```

59: +
60:   alias_method :edit_income, :edit
61:   alias_method :edit_ulpgc_teachers, :edit
62:   alias_method :edit_non_ulpgc_teachers, :edit
63: diff --git a/app/controllers/general_informations_controller.rb b/app/controllers/gener
al_informations_controller.rb
64: index edeca3c..b90737d 100644
65: --- a/app/controllers/general_informations_controller.rb
66: +++ b/app/controllers/general_informations_controller.rb
67: @@ -3,7 +3,7 @@
68:  class GeneralInformationsController < InheritedResources::Base
69:    singleton_belongs_to :university_specific_degree
70:    actions :show, :edit, :update
71: - respond_to :html
72: + respond_to :html, :xml
73:
74:   load_and_authorize_resource :class => "UniversitySpecificDegree"
75:
76: diff --git a/app/controllers/incomes_controller.rb b/app/controllers/incomes_controller
.rb
77: index 8917be3..7e0ef95 100644
78: --- a/app/controllers/incomes_controller.rb
79: +++ b/app/controllers/incomes_controller.rb
80: @@ -8,6 +8,7 @@ class IncomesController < InheritedResources::Base
81:   load_and_authorize_resource :university_specific_degree
82:   load_and_authorize_resource :economic_report, :through => :university_specific_degre
e
83:   load_and_authorize_resource :income, :through => :economic_report
84: + respond_to :html, :xml
85:
86:   def create
87:     create!(:notice => "Se aÃ±adiÃ³ el ingreso directo.") { collection_url }
88: diff --git a/app/controllers/issues_controller.rb b/app/controllers/issues_controller.r
b
89: index 024618a..07ea499 100644
90: --- a/app/controllers/issues_controller.rb
91: +++ b/app/controllers/issues_controller.rb
92: @@ -3,7 +3,7 @@
93:  class IssuesController < InheritedResources::Base
94:    belongs_to :university_specific_degree
95:    actions :index
96: -
97: + respond_to :html, :xml
98:   load_and_authorize_resource :university_specific_degree
99:   load_and_authorize_resource :issue, :through => :university_specific_degree
100: end
101: diff --git a/app/controllers/subjects_controller.rb b/app/controllers/subjects_controll
er.rb
102: index 5d686eb..3f91031 100644
103: --- a/app/controllers/subjects_controller.rb
104: +++ b/app/controllers/subjects_controller.rb
105: @@ -5,9 +5,11 @@ class SubjectsController < InheritedResources::Base
106:   belongs_to :university_specific_degree
107:   load_and_authorize_resource :university_specific_degree
108:   load_and_authorize_resource :subject, :through => :university_specific_degree
109: + respond_to :html, :xml
110:   before_filter :load_branch_of_knowledge_subjects,
111:     :only => [ :new, :create, :edit, :update ]
112:
113: +
114:   def create
115:     create!(:notice => "Se aÃ±adiÃ³ la asignatura al tÃ­tulo propio") { [ @university_
specific_degree, @subject ] }

```

```
116: end
117: diff --git a/app/controllers/teacher_authorizations_controller.rb b/app/controllers/teacher_authorizations_controller.rb
118: index 38574a1..e018426 100644
119: --- a/app/controllers/teacher_authorizations_controller.rb
120: +++ b/app/controllers/teacher_authorizations_controller.rb
121: @@ -2,7 +2,7 @@ class TeacherAuthorizationsController < InheritedResources::Base
122:   load_and_authorize_resource :only => [ :edit, :update ]
123:
124:   actions :index, :edit, :update
125: - respond_to :js
126: + respond_to :js, :xml
127:
128:   protected
129:
130: diff --git a/app/controllers/teacher_sessions_controller.rb b/app/controllers/teacher_sessions_controller.rb
131: index flaca6e..2dab849 100644
132: --- a/app/controllers/teacher_sessions_controller.rb
133: +++ b/app/controllers/teacher_sessions_controller.rb
134: @@ -2,6 +2,7 @@
135:
136: class TeacherSessionsController < ApplicationController
137:   skip_before_filter :require_user
138: + skip_before_filter :verify_authenticity_token
139:   before_filter :require_no_teacher, :only => [:new, :create]
140:   before_filter :require_teacher, :only => :destroy
141:
142: diff --git a/app/controllers/teachers_controller.rb b/app/controllers/teachers_controller.rb
143: index b1bdd4b..ab755a1 100644
144: --- a/app/controllers/teachers_controller.rb
145: +++ b/app/controllers/teachers_controller.rb
146: @@ -7,7 +7,7 @@ class TeachersController < InheritedResources::Base
147:   before_filter :collection, :only => :index
148:
149:   actions :index, :edit, :update
150: - respond_to :html, :json
151: + respond_to :html, :json, :xml
152:
153:   load_and_authorize_resource :except => :activate
154:
155: diff --git a/app/controllers/user_sessions_controller.rb b/app/controllers/user_sessions_controller.rb
156: index e0dd75c..9c66b33 100644
157: --- a/app/controllers/user_sessions_controller.rb
158: +++ b/app/controllers/user_sessions_controller.rb
159: @@ -1,6 +1,8 @@
160:  # encoding: utf-8
161:
162: class UserSessionsController < ApplicationController
163: + skip_before_filter :require_user
164: + skip_before_filter :verify_authenticity_token
165:   before_filter :require_no_user, :only => [:new, :create]
166:   before_filter :require_user, :only => :destroy
167:
168: diff --git a/db/schema.rb b/db/schema.rb
169: index 730f145..8b4357d 100644
170: --- a/db/schema.rb
171: +++ b/db/schema.rb
172: @@ -17,8 +17,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
173:   t.text "reason"
174:   t.integer "university_specific_degree_id"
```

```

175:      t.integer   "user_id"
176: -      t.datetime "created_at"
177: -      t.datetime "updated_at"
178: +      t.datetime "created_at",           :null => false
179: +      t.datetime "updated_at",         :null => false
180:      t.boolean   "notify",               :default => true
181:      t.string    "type"
182:      t.boolean   "definitive"
183: @@ -118,8 +118,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
184:      t.decimal   "incomes_sum"
185:      t.decimal   "expenses_sum"
186:      t.decimal   "sum"
187: -      t.datetime "created_at"
188: -      t.datetime "updated_at"
189: +      t.datetime "created_at",           :nu
ll => false
190: +      t.datetime "updated_at",         :nu
ll => false
191:      t.integer   "specific_scholar_qty"
192:      t.decimal   "specific_scholar_cost"
193:      t.decimal   "specific_scholar_sum"
194: @@ -165,8 +165,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
195:      t.decimal   "amount"
196:      t.integer   "economic_report_id"
197:      t.integer   "income_type_id"
198: -      t.datetime "created_at"
199: -      t.datetime "updated_at"
200: +      t.datetime "created_at",           :null => false
201: +      t.datetime "updated_at",         :null => false
202:      end
203:
204:      create_table "institutions", :force => true do |t|
205: @@ -175,8 +175,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
206:      t.string    "other_phone"
207:      t.string    "email"
208:      t.text     "address"
209: -      t.datetime "created_at"
210: -      t.datetime "updated_at"
211: +      t.datetime "created_at",         :null => false
212: +      t.datetime "updated_at",         :null => false
213:      t.string    "location"
214:      t.boolean   "educational"
215:      t.integer   "parent_id"
216: @@ -186,8 +186,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
217:      create_table "issues", :force => true do |t|
218:      t.text     "message"
219:      t.integer   "university_specific_degree_id"
220: -      t.datetime "created_at"
221: -      t.datetime "updated_at"
222: +      t.datetime "created_at",         :null => false
223: +      t.datetime "updated_at",         :null => false
224:      end
225:
226:      create_table "periods", :force => true do |t|
227: @@ -198,8 +198,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
228:      t.date      "first_rectification_to_date"
229:      t.date      "second_rectification_from_date"
230:      t.date      "second_rectification_to_date"
231: -      t.datetime "created_at"
232: -      t.datetime "updated_at"
233: +      t.datetime "created_at",         :null => false
234: +      t.datetime "updated_at",         :null => false
235:      end

```

```

236:
237:     create_table "settings", :force => true do |t|
238:   @@ -234,8 +234,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
239:     t.decimal "exams_hours", :precision => 6, :scale => 2, :default
=> 0.0
240:     t.decimal "hours_sum", :precision => 6, :scale => 2, :default
=> 0.0
241:     t.integer "university_specific_degree_id"
242: -   t.datetime "created_at"
243: -   t.datetime "updated_at"
244: +   t.datetime "created_at",
:   :null => false
245: +   t.datetime "updated_at",
:   :null => false
246:     t.integer "subject_id"
247:     t.decimal "teacher_hours", :precision => 6, :scale => 2
248:     t.integer "branch_of_knowledge_id"
249:   @@ -262,8 +262,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
250:     t.decimal "ects_credits"
251:     t.integer "degree_teacher_id"
252:     t.integer "subject_id"
253: -   t.datetime "created_at"
254: -   t.datetime "updated_at"
255: +   t.datetime "created_at",
:   :
null => false
256: +   t.datetime "updated_at",
:   :
null => false
257:     t.boolean "responsible", :default => false
258:   end
259:
260:   @@ -277,8 +277,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
261:     t.string "institute"
262:     t.string "department"
263:     t.text "cv_abstract"
264: -   t.datetime "created_at"
265: -   t.datetime "updated_at"
266: +   t.datetime "created_at", :null => false
267: +   t.datetime "updated_at", :null => false
268:     t.string "aasm_state"
269:     t.integer "proposer_id"
270:     t.string "type"
271:   @@ -339,8 +339,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
272:     t.string "state"
273:     t.integer "user_id"
274:     t.integer "period_id"
275: -   t.datetime "created_at"
276: -   t.datetime "updated_at"
277: +   t.datetime "created_at", :null
=> false
278: +   t.datetime "updated_at", :null
=> false
279:     t.string "extra_information_file_name"
280:     t.string "extra_information_content_type"
281:     t.integer "extra_information_file_size"
282:   @@ -370,8 +370,8 @@ ActiveRecord::Schema.define(:version => 20121112091744) do
283:   create_table "users", :force => true do |t|
284:     t.string "name"
285:     t.string "last_name"
286: -   t.datetime "created_at"
287: -   t.datetime "updated_at"
288: +   t.datetime "created_at", :null => false
289: +   t.datetime "updated_at", :null => false
290:     t.string "login"

```

./android_app.patch

Thu Oct 17 14:30:44 2013

6

```
291:      t.string  "email"
292:      t.string  "crypted_password"
293: --
294: 1.7.9.5
295:
```

12.4 *Manual de Usuario de la aplicación*

Manual de usuario de la Aplicación móvil de gestión de títulos propios 2013/2014





UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Vicerrectorado de Títulos y Doctorado



Índice

Seleccionar Rol.....	4
Autenticación.....	4
Seleccionar tarea.....	5
Crear nueva propuesta.....	5
Búsqueda de propuestas.....	7
Resultado de la búsqueda.....	7
Menú principal.....	8
Información general.....	9
Asignaturas.....	10
Asignatura.....	11
Profesorado.....	12
Profesor.....	12
Memoria económica.....	13
Reporte de errores.....	15
Enviar propuesta para evaluación.....	15

Seleccionar Rol

En primer lugar, se establece la pantalla de inicio, donde se elegirá el rol con el que acceder a la aplicación. Este dato es imprescindible, pues la plataforma web subyacente trata a ambos roles de manera diferente y la autenticación se realiza a través de dos URL diferentes.

La selección de uno de los dos roles llevará a continuación hacia la pantalla de autenticación del usuario.



Imagen 1: Seleccionar rol

Autenticación

Esta pantalla guiará al usuario a la actividad encargada de recibir la información del usuario. Asimismo, permitirá guardar y recordar los datos introducidos previamente como funcionalidad.

Para el caso de los profesores o gestores pertenecientes a la plantilla docente de la ULPGC, el correo de acceso es el mismo que para acceder a la plataforma MiULPGC.

En el caso de los gestores de fuera de la ULPGC, el nombre de usuario y contraseña se establece a la hora de crear el usuario. Recordamos que los encargados de crear usuarios derivados son los directores de los centros de la ULPGC, así como los usuarios en los que dichos directores hayan delegado.

En el caso de los profesores externos a la ULPGC, el nombre de usuario será su DNI. Su contraseña la establecerán los mismos usuarios la primera vez que sean propuestos en la plataforma y reciban el correo de confirmación correspondiente.



Imagen 2: Autenticación

Seleccionar tarea

En el caso de que el usuario sea un proponente de títulos propios, se le mostrará una nueva pantalla de selección de tarea. En este caso, el usuario seleccionará su siguiente acción entre dos opciones. Estas opciones son 'Crear nueva propuesta' y 'Explorar propuestas existentes'.



Imagen 3: Seleccionar tarea

Crear nueva propuesta

Si el periodo de creación de propuestas se encuentra vigente, se le presenta al usuario una serie de campos obligatorios que rellenar para completar el borrador de la nueva propuesta de título. Al seleccionar alguno de los items del menú desplegable correspondiente al centro adscrito, la información sobre la sede de la organización se rellenará automáticamente con la información contenida de ese centro en la base de datos de la aplicación web. No obstante, se podrá editar si la dirección es diferente a la dirección de la sede de la organización.

En el momento en el que el usuario pulsa el botón enviar, la aplicación móvil verificará que todos los campos son distintos de la ristra vacía. A continuación, verificará que el DNI del proponente introducido es de un profesor dado de alta en la aplicación web y además es profesor de la ULPGC. Ésto se logra haciendo una búsqueda de profesores con el único parámetro del DNI introducido.



Gestor TTPP

Denominación

Item 1
Sub Item 1

DNI de Director Académico

Proponente

Unidades participantes

Plazas mínimas

Plazas máximas

Mínimo de créditos anuales

Gestor económico

Competencias

Centro adscrito

Item 1
Sub Item 1

Sede de la organización

Sede de la organización

Título impartido fuera de las Islas Canarias

Enviar

Imagen 4: Crear nueva propuesta

Búsqueda de propuestas

En esta pantalla, el usuario podrá introducir diferentes criterios de búsqueda para explorar las propuestas a las que tiene acceso según sus permisos. Estos criterios incluyen los tipos de título, la rama de conocimiento y el estado. Asimismo, admitirá subítems correspondientes a la denominación, así como su código generado.

En el caso de que el usuario no seleccione ningún criterio, se mostrarán todas las propuestas disponibles.

Cuando la aplicación móvil ha terminado de descargar y contabilizar las propuestas disponibles, muestra un diálogo con el número de propuestas y dos opciones. Una de las opciones será repetir la búsqueda con otros criterios, siendo la otra la opción de mostrar el listado de propuestas que ha retornado la búsqueda.

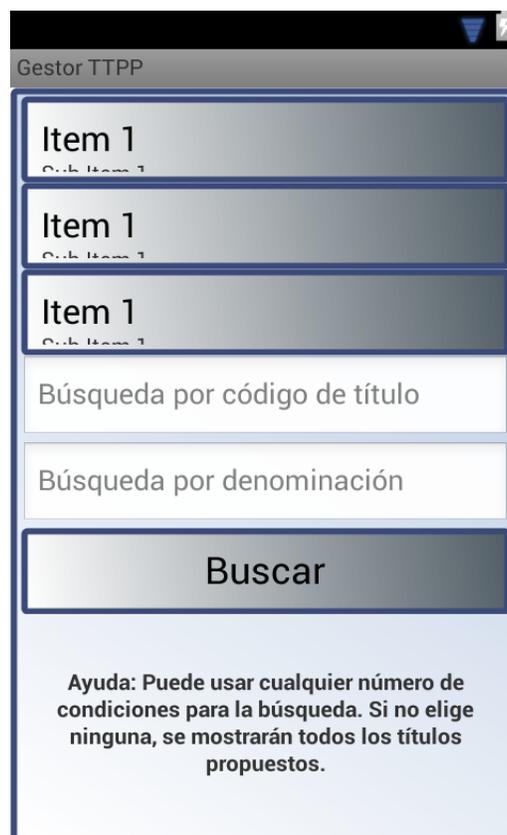


Imagen 5: Búsqueda de propuestas

Resultado de la búsqueda

Una vez aceptado el set de resultados de la búsqueda realizada en la pantalla anterior, se presentará la lista de propuestas que cumplen con los parámetros de búsqueda, mostrando su información básica y permitiendo el acceso a la información de cada una de ellas a través de la pantalla que se inicia al pulsar sobre la denominación del ítem correspondiente propuesta de la lista generada.



Imagen 6: Resultado de la búsqueda



Menú principal

En esta pantalla se representan 7 acciones que realizar sobre la propuesta de título que se escogió explorar en la pantalla anterior. Esta pantalla, en el caso de los dispositivos móviles *handheld*, actuará de control para iniciar el resto de actividades. En el caso de los dispositivos tableta, esta actividad actuará de mecanismo de control para la sustitución de los fragmentos correspondientes a cada una de las funcionalidades presentadas, excepto para la opción de descarga de la memoria en PDF, que se ejecutará desde esta misma pantalla.

Esta acción de descarga de la memoria mostrará un diálogo que hace el seguimiento de la cantidad de bytes descargados hasta el momento. Cuando finaliza la descarga, la aplicación web buscará la aplicación instalada en el dispositivo encargada de leer los documentos PDF y la abrirá automáticamente. Si no la encuentra, depositará la memoria en la carpeta de descargas por defecto y mostrará un mensaje de error relacionado con la inexistencia de la aplicación encargada de visualizar documentos PDF.

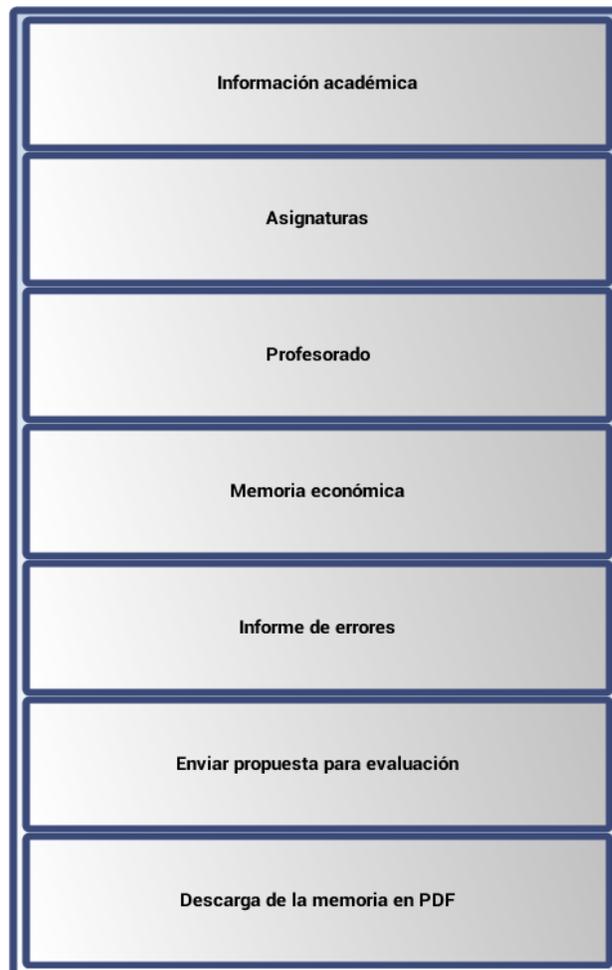


Imagen 7: Menú principal

Información general

El primer enlace del menú principal procesa la información general de la propuesta y la muestra de forma ordenada en columnas de a dos. Esta información describe las características del título y su información básica. Debido a que varios de sus campos de datos contienen un identificador numérico en vez de una versión válida para la lectura, se realiza un preprocesado de los campos 'Rama de conocimiento' y 'Tipo de enseñanza'.

Información General	
Denominación: Pensamiento Crítico y Análisis Socio-Cultural Contemporáneos	
Rama de Conocimiento: Arte y humanidades	Tipo de enseñanza: Experto
Estado de la propuesta: subsanacion	Número de edición: 2
Código: TP3-1-059-02	Proponente: Aula Manuel Alemán, José Alonso Morales e Isabel Luján Henríquez
Inicio de las enseñanzas: 2012-10-03	Fin de las enseñanzas: 2013-06-18
Unidades participantes: Vicerrectorado de Cultura y Deporte	Lugar de impartición: Escuela de Arquitectura
Gestor: Centro de Formación Continua	Representante del centro: Daniel Barreto González
Sede de la organización: Edificio Arquitectura CAMPUS UNIVERSITARIO 35017 - Las Palmas de Gran Canaria	Evaluado por acecau en: Sin especificar
Competencias: Este programa intenta dotar a los que se beneficien de sus aportaciones con las siguientes competencias: - Tener una visión reflexiva de problemas y situaciones de la actualidad desde criterios éticos y filosóficos - Introducir y profundizar en el pensamiento crítico y en la confrontación entre posturas diversas y encontradas. - Ofrecer claves de comprensión de los fenómenos culturales contemporáneos. - Familiarizarse y adiestrarse en elementos de comprensión de los textos de pensadores y filósofos especialmente del mundo actual. - Preparar, en un mundo plural y globalizado, para un diálogo intercultural en los diferentes ámbitos de la sociedad. - Dotar de elementos para analizar los comportamientos y las culturas con las que se entra en contacto en una realidad en que la inmigración aumenta progresivamente. - Tomar postura crítica frente a un pensamiento estrictamente técnico y manipulado desde poderes que dominan nuestra sociedad. - Crear conciencia social y política frente a situaciones y problemas que acaecen en nuestra realidad cercana y mundial. - Dotar de formación específica para la gestión y la política cultural	
Justificación de competencias: Este programa intenta dotar a los que se beneficien de sus aportaciones con las siguientes competencias: - Tener una visión reflexiva de problemas y situaciones de la actualidad desde criterios éticos y filosóficos - Introducir y profundizar en el pensamiento crítico y en la confrontación entre posturas diversas y encontradas. - Ofrecer claves de comprensión de los fenómenos culturales contemporáneos. - Familiarizarse y adiestrarse en elementos de comprensión de los textos de pensadores y filósofos especialmente del mundo actual. - Preparar, en un mundo plural y globalizado, para un diálogo intercultural en los diferentes ámbitos de la sociedad. - Dotar de	

Imagen 8: Información general

Asignaturas

El segundo enlace del menú principal generará una nueva lista, si hubiera asignaturas cargadas en la propuesta, con la información básica de cada asignatura y un enlace sobre su nombre para acceder a la información académica completa en una nueva actividad.

Asignaturas	
Bioética. Un puente entre ciencia y valores	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio: 2012-10-10	Fin: 2012-11-14
Biología y conciencia	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio: Sin especificar	Fin: Sin especificar
Ciencia, Tecnología y Humanismo.	
Curso: 1	Semestre: 1
13/13 asignaturas cargadas	

Imagen 9: Asignaturas



Asignatura

Al pulsar en una de las asignaturas del listado de asignaturas, se mostrará la información completa relacionada con dicha asignatura, incluyendo la información general completa, el reparto de horas, competencias, actividades formativas, descripción de los contenidos y el sistema de evaluación.

Bioética. Un puente entre ciencia y valores	
Nombre: Bioética. Un puente entre ciencia y valores	
Información general	
Curso: 1	Semestre: 1
Tipo de asignatura: Obligatoria	Créditos ECTS: 1.0
Inicio de docencia: 2012-10-10	Fin de docencia: 2012-11-14
Rama de conocimiento: Arte y humanidades	Modalidad: Presencial
Creado: 2011-06-20 09:30:27	Actualizado: 2012-02-13 09:42:40
Reparto de horas	
Clases teóricas: 4.0 horas	Clases prácticas: 2.0 horas
Actividades docentes complementarias: 4.5 horas	Prácticas en empresas/instituciones: 0.0 horas
Pruebas de evaluación: 2.0 horas	Trabajo final: 0.0 horas
Dedicadas al estudio: 12.5 horas	Número de horas en total: 25.0 horas
Competencias: El primero en hablar de Bioética, Van Rensselaer Potter (Madison, 1971) usó el término en un sentido amplio, queriendo significar el puente entre dos culturas: de un lado el conocimiento científico biológico, del otro los valores humanos. Para él la bioética es "Bridge to the Future", puente de encuentro entre los hechos que presentan las ciencias de la vida y el mundo del valor. Se intenta conseguir: - Aproximación a la comprensión de la Bioética y sus implicaciones - Acercamiento a los grandes debates de actualidad bioética, desde una visión interdisciplinar. - Introducción de la Bioética como posible "estilo de vida" desde la humanización y el cuidado	
Actividades formativas: Exposición de temas por parte de la profesora. Análisis de casos prácticos desde los que se intentará cultivar la búsqueda común y el diálogo entre las diferentes visiones que puedan surgir en el grupo	
Descripción de los contenidos:	
Sistema de evaluación: Evaluación continua que tendrá en cuenta la asistencia, participación y análisis de casos - Entrega de una Memoria de lo trabajado.	

Imagen 10: Asignatura

Profesorado

Cuando se pulsa la opción de profesorado del título, se mostrará un listado del profesorado asignado a la propuesta, así como una opción de notificación de un profesor concreto junto al enlace de su nombre, en el caso de que se encuentre en estado 'Pendiente de confirmar participación'.

Si se pulsa en el nombre del profesor, se dará paso a la pantalla de información del profesor.



Imagen 11: Profesorado

Profesor

Cuando se pulsa sobre el nombre del profesor, se iniciará otra actividad o fragmento que mostrará la información completa existente en el perfil del profesor propuesto.

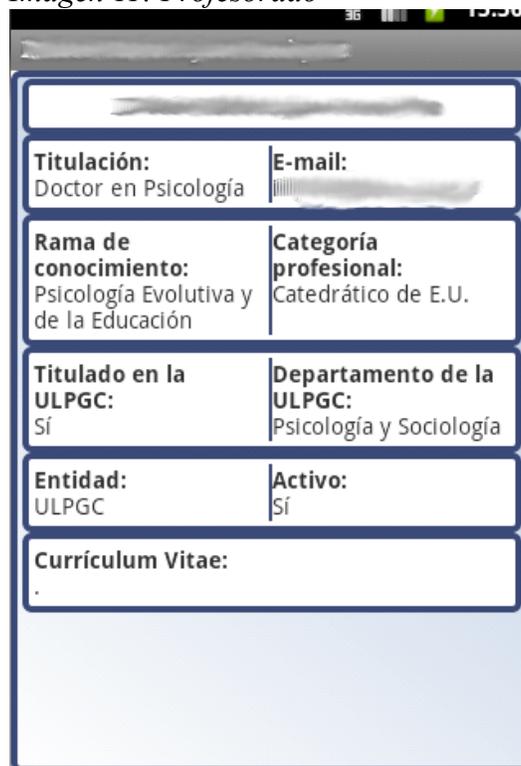


Imagen 12: Profesor

Memoria económica

La cuarta opción del menú principal mostrará la información relacionada con la memoria económica de la propuesta. Esta pantalla es significativamente más larga que cualquier otra en la aplicación. Por esta razón, esta vista está dividida en tres secciones. Balance, Ingresos y Gastos.

Balance



Resumen Ingresos/Gastos	
Ingresos totales	9000.0€
Suma de gastos parcial	7645.0€
Compensación a la ULPGC (5%)	450.0€
Compensación al centro (5%)	450.0€
Compensación al gestor económico (5%)	450.0€
Total de Gastos	8995.0€
Balance	5.0€

Imagen 13: Memoria económica. Balance

Ingresos



Ingresos		
Derivados de las matrículas		
Número de estudiantes becados	Precio de matrícula	Total
3	300.0	900.0 €
Número de estudiantes sin beca	Precio de matrícula	Total
12	600.0	7200.0 €
Ingresos por matrículas		8100.0 €
Directos y subvenciones		
Total		900.0 €

Imagen 14: Memoria económica. Ingresos



Gastos

La vista de gastos es la más extensa de toda la aplicación, requiriendo más de cuatro veces más espacio que el disponible en una tableta en modo handheld. Se adjunta una única captura de pantalla de su sección inicial.

Memoria Académica		
Resumen	Ingresos	Gastos
Gastos		
Profesorado ULPGC		
Número de horas de docencia	Coste por hora	Total
84.0	10.0 €	840.0 €
Número de horas de seminario	Coste por hora	Total
89.5	10.0 €	895.0 €
Número de horas de prácticas	Coste por hora	Total
47.0	10.0 €	470.0 €
Número de horas de evaluación	Coste por hora	Total
42.0	10.0 €	420.0 €
Elaboración de manuales docentes	Coste por crédito	Total
0.0	0.0 €	0.0 €
Tutorías presenciales y de trabajos finales	Coste por proyecto	Total
		0.0 €
Tribunales para valoración de trabajos finales	Coste por tribunal	Total
		0.0 €
Coordinación de exámenes	Coste por examen	Total
0	0.0 €	0.0 €
Directores		Coste
Coste de profesorado de la ULPGC		Coste
		2625.0 €
Beneficios para la unidad organizadora		Coste
Total retribución bruta a profesorado de la ULPGC		Total

Imagen 15: Memoria económica. Gastos

Reporte de errores

En la quinta opción del menú principal se muestra, si lo hubiera, el último reporte de errores que la plataforma ha generado cuando se ha tratado de enviar la propuesta. Se listan las razones por las que la propuesta no ha pasado directamente al estado 'Enviado', estado en el que se establecen las propuestas ya preparadas para ser evaluadas por las diferentes comisiones.

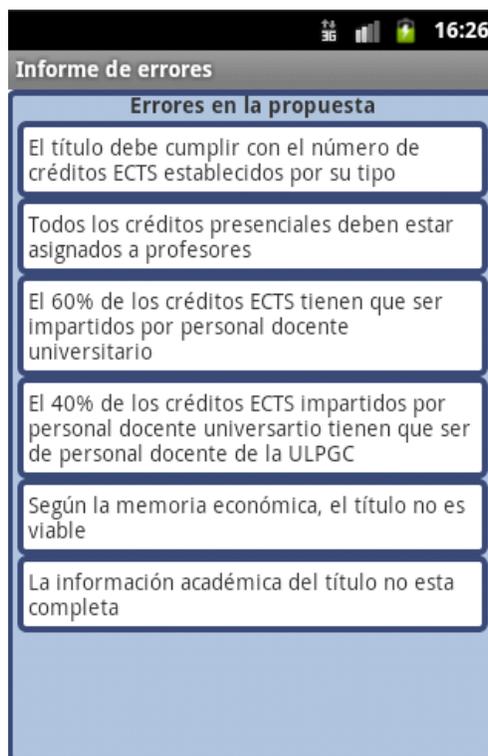


Imagen 16: Informe de errores

Enviar propuesta para evaluación

En la sexta opción del menú principal se provee la opción de remitir la propuesta actual siendo revisada a ser evaluada por la aplicación web. Como control previo, la aplicación móvil descarga el recurso periods.xml para comprobar qué plazos están abiertos, sea periodo de petición de propuestas, primera subsanación o segunda subsanación. Si ninguno de estos plazos están abiertos o el curso académico no es el correcto, la aplicación móvil avisará al usuario de esta circunstancia y no remitirá la propuesta a evaluar.

En el caso en el que los plazos sean correctos, se remite la propuesta a evaluar y se redirige al usuario a la actividad anterior, reporte de errores, para comprobar cuáles son, si hubieran, los errores que han impedido que la propuesta pase a estado 'Enviado'.

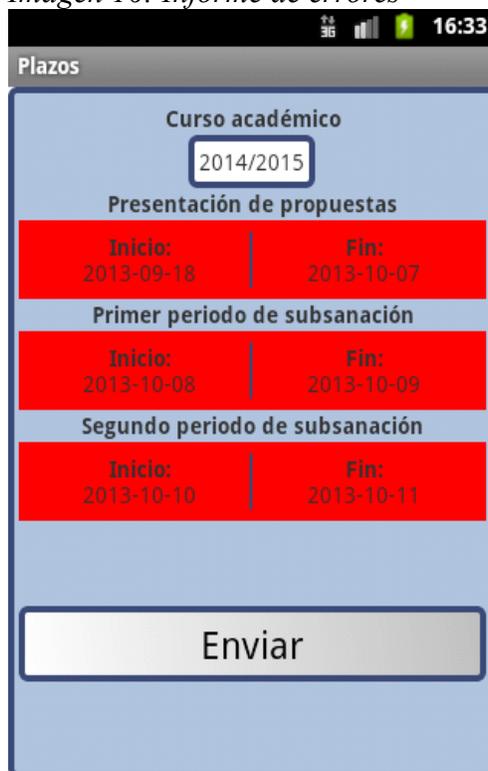


Imagen 17: Enviar propuesta para evaluación