



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Trabajo de Fin de Grado en Ingeniería Informática

DESARROLLO DE UNA APLICACIÓN MÓVIL HÍBRIDA DE
SOPORTE PARA CONGRESOS EN IOS

JULIO 2015

ALUMNO: RUYMÁN MOLINA PÉREZ

TUTOR: DR. ALEXIS QUESADA ARENCIBIA (ESCUELA DE INGENIERÍA INFORMÁTICA)

A mi familia y amigos y a mi tutor Alexis Quesada

Resumen

Esta aplicación surge de la necesidad de generar una herramienta de bajo coste capaz de dar un soporte estandarizado a diferentes tipos de congresos a través de un mismo software. Este tipo de congresos debe manejar grandes volúmenes de datos de forma eficiente.

Así, el objetivo fundamental de este proyecto es generar una aplicación para móviles que de soporte a congresos, usando para ello un desarrollo móvil que incorpore elementos del desarrollo web; es decir, un desarrollo híbrido. Como objetivo secundario, se estudiará la viabilidad de este sistema de desarrollo, evaluando las diferencias obtenidas frente a un desarrollo web completo tanto en eficiencia como en recursos utilizados.

Este proyecto demostrará, por medio de la aplicación desarrollada, su utilidad en el campo de la gestión de congresos. Por otro lado, también resultará útil como estudio de viabilidad del desarrollo híbrido, estudiando sus ventajas e inconvenientes.

Abstract

This application covers the needs of generating a standardized low cost application that provides support to different kinds of conferences through the same software. This kind of conferences always has to deal with large volume data in an efficient way.

As such, the main objective for this project will be to obtain a mobile application that provides support to conferences, using a kind of mobile development featuring elements from web development; that is, hybrid development. As a secondary objective, the viability of this development system will be subject of study, evaluating the differences obtained against complete web development in terms of efficiency and resources used.

This project will prove its utility in the congress management field. In the other hand it will also be useful as a study of the hybrid development, studying its benefits and inconveniences.

Índice de contenido

| | |
|--|----|
| Resumen | 2 |
| Abstract..... | 2 |
| Índice de contenido | 3 |
| Índice de tablas..... | 8 |
| Índice de ilustraciones | 9 |
| Estructura del documento | 12 |
| Estructura del CD | 13 |
| 1 Punto de partida..... | 14 |
| 1.1 Introducción | 14 |
| 1.2 Contextualización..... | 16 |
| 1.2.1 Congresos e introducción a la gestión de congresos | 16 |
| 1.2.2 Aplicaciones móviles nativas y aplicaciones web móviles | 16 |
| 1.2.3 Desarrollo híbrido | 17 |
| 1.3 Estado del arte | 18 |
| 1.3.1 Conclusiones obtenidas | 21 |
| 1.4 Objetivos | 22 |
| 1.4.1 Objetivos secundarios de la aplicación a desarrollar | 22 |
| 1.4.2 Objetivo secundario: Estudio de la viabilidad del desarrollo híbrido | 23 |
| 1.5 Normativa y legislación | 24 |
| 1.5.1 Ley de Protección de Datos de Carácter Personal | 24 |
| 1.5.2 Clasificación de licencias de software | 25 |
| 1.5.3 Reglamento que afecta al Trabajo de Fin de Grado..... | 25 |
| 1.6 Aportaciones | 26 |
| 1.6.1 Aportaciones a nivel social y económico | 26 |
| 1.6.2 Aportaciones a nivel científico y técnico..... | 26 |
| 1.7 Competencias desarrolladas | 27 |

| | | |
|-------|--|----|
| 2 | Elaboración del proyecto..... | 31 |
| 2.1 | Metodología..... | 31 |
| 2.1.1 | Manifiesto ágil..... | 31 |
| 2.2 | Herramientas utilizadas..... | 33 |
| 2.2.1 | Herramientas generales..... | 33 |
| 2.2.2 | Herramientas específicas a la plataforma..... | 36 |
| 2.2.3 | Herramientas para el desarrollo web..... | 40 |
| 2.3 | Análisis..... | 43 |
| 2.3.1 | Introducción a la ingeniería de requisitos..... | 43 |
| 2.3.2 | Estudio inicial..... | 43 |
| 2.3.3 | Captura e identificación de requisitos..... | 44 |
| 2.3.4 | Actores del sistema..... | 44 |
| 2.3.5 | Diccionario de conceptos..... | 45 |
| 2.3.6 | Análisis de requisitos..... | 45 |
| 2.4 | Diseño..... | 47 |
| 2.4.1 | Arquitectura del sistema..... | 47 |
| 2.4.2 | Diseño arquitectónico..... | 48 |
| 2.4.3 | Entidades del modelo..... | 49 |
| 2.5 | Desarrollo..... | 51 |
| 2.5.1 | Modelado..... | 52 |
| 2.5.2 | Conexión con el servidor <i>Back-End</i> | 55 |
| 2.5.3 | Maquetado de las vistas..... | 58 |
| 2.5.4 | Integración híbrida..... | 65 |
| 2.5.5 | Comparativa entre los desarrollos nativos Android e iOS..... | 71 |

| | | |
|-------|--|----|
| 3 | Conclusiones y trabajos futuros | 74 |
| 3.1 | Conclusiones | 74 |
| 3.1.1 | Conclusiones sobre el desarrollo de la aplicación..... | 74 |
| 3.1.2 | Conclusiones sobre el desarrollo híbrido..... | 76 |
| 3.2 | Trabajos futuros | 77 |
| 3.2.1 | Funcionalidades no desarrolladas..... | 77 |
| 3.2.2 | Propuesta de nuevas funcionalidades | 78 |
| 4 | Fuentes de información..... | 79 |
| 4.1 | Información sobre los congresos | 79 |
| 4.2 | Normativa y legislación | 79 |
| 4.3 | Metodología | 80 |
| 4.4 | Herramientas..... | 80 |
| 4.5 | Análisis | 81 |
| 4.6 | Diseño..... | 81 |

| | | |
|--------|--|-----|
| 5 | Anexos | 82 |
| 5.1 | Anexo I – Detalle de los congresos..... | 82 |
| 5.1.1 | CES – Consumer Electronics Show | 82 |
| 5.1.2 | MWC – Mobile World Congress..... | 82 |
| 5.1.3 | Gamescom | 83 |
| 5.1.4 | E3 – Electronic Entertainment Expo..... | 83 |
| 5.1.5 | DOG – Deutsche Ophthalmologische Gesellschaft | 84 |
| 5.2 | Anexo II – Capturas de pantalla de la maqueta 85 | |
| 5.2.1 | Home..... | 85 |
| 5.2.2 | My Account | 86 |
| 5.2.3 | Program..... | 87 |
| 5.2.4 | Update..... | 88 |
| 5.2.5 | Authors..... | 89 |
| 5.2.6 | Workshops | 90 |
| 5.2.7 | Keywords..... | 91 |
| 5.2.8 | Committees..... | 92 |
| 5.2.9 | News..... | 93 |
| 5.2.10 | Social Activities | 94 |
| 5.2.11 | Sponsors..... | 95 |
| 5.2.12 | Session detail | 96 |
| 5.2.13 | Author detail..... | 97 |
| 5.2.14 | Workshop detail..... | 98 |
| 5.2.15 | Keyword detail | 99 |
| 5.2.16 | New detail..... | 100 |
| 5.2.17 | Social Activity detail..... | 101 |
| 5.3 | Anexo III – Diagrama de flujo de la aplicación | 102 |

| | | |
|--------|------------------------------------|-----|
| 5.4 | Anexo IV – Manual de usuario | 103 |
| 5.4.1 | Navegación hacia atrás | 103 |
| 5.4.2 | Menú principal | 104 |
| 5.4.3 | Programa..... | 105 |
| 5.4.4 | Mi programa | 108 |
| 5.4.5 | Detalle de una charla | 109 |
| 5.4.6 | Actualización | 112 |
| 5.4.7 | Listado de autores..... | 113 |
| 5.4.8 | Detalle de un autor | 115 |
| 5.4.9 | Listado de temáticas | 118 |
| 5.4.10 | Detalle de una temática..... | 120 |
| 5.4.11 | Listado de palabras clave | 123 |
| 5.4.12 | Detalle de una palabra clave..... | 125 |
| 5.4.13 | Listado de comités | 126 |
| 5.4.14 | Listado de noticias | 129 |
| 5.4.15 | Detalle de una noticia | 130 |
| 5.4.16 | Listado de eventos | 131 |
| 5.4.17 | Detalle de un evento..... | 132 |
| 5.4.18 | Listado de patrocinadores | 133 |
| 5.4.19 | Mi cuenta | 134 |

Índice de tablas

| | |
|---|----|
| Tabla 1-1 - Comparativa del desarrollo nativo y el desarrollo web en dispositivos móviles | 17 |
| Tabla 1-2 - Congresos de la ITS | 18 |
| Tabla 1-3 - Resumen de los congresos seleccionados | 19 |
| Tabla 1-4 - Aplicaciones de congresos en Android y iOS | 20 |
| Tabla 2-1 - Actores del sistema | 44 |
| Tabla 2-2 - Diccionario de conceptos | 45 |
| Tabla 2-3 - Entidades del modelo | 50 |
| Tabla 5-1 - Detalle de <i>CES</i> | 82 |
| Tabla 5-2 - Detalle de <i>MWC</i> | 82 |
| Tabla 5-3 - Detalle de <i>Gamescom</i> | 83 |
| Tabla 5-4 - Detalle de <i>E3</i> | 83 |
| Tabla 5-5 - Detalle de <i>DOG</i> | 84 |

Índice de ilustraciones

| | |
|--|----|
| Ilustración 1-1 - Logotipo de la ITS..... | 18 |
| Ilustración 2-1 - Logotipo de Microsoft Office 2013 | 33 |
| Ilustración 2-2 - Logotipo de Google Drive | 34 |
| Ilustración 2-3 - Logotipo de Dropbox | 34 |
| Ilustración 2-4 - Logotipo de Git | 34 |
| Ilustración 2-5 - Logotipo de Bitbucket..... | 34 |
| Ilustración 2-6 - Diagrama Modelo-Vista-Controlador | 35 |
| Ilustración 2-7 - Logotipo de Swift | 36 |
| Ilustración 2-8 - Logotipo de Xcode | 37 |
| Ilustración 2-9 - Logotipo de Core Data | 37 |
| Ilustración 2-10 Logotipo de JSON | 38 |
| Ilustración 2-11 - Foto de iPhone 6..... | 39 |
| Ilustración 2-12 - Foto de iPad 2 | 39 |
| Ilustración 2-13 - Logotipo de HTML5..... | 40 |
| Ilustración 2-14 - Logotipo de CSS3 | 40 |
| Ilustración 2-15 - Logotipo de JavaScript..... | 41 |
| Ilustración 2-16 - Logotipo de Notepad++ | 41 |
| Ilustración 2-17 - Logotipo de normalize.css | 41 |
| Ilustración 2-18 - Logotipo de JQuery | 42 |
| Ilustración 2-19 - Logotipo de JQuery Mobile..... | 42 |
| Ilustración 2-20 - Logotipo de Mozilla Firefox | 42 |
| Ilustración 2-21 - Arquitectura Cliente-Servidor a dos niveles | 47 |
| Ilustración 2-22 - Diseño arquitectónico de la aplicación..... | 48 |
| Ilustración 2-23 - Captura de pantalla del entorno Xcode, entidades del modelo..... | 49 |
| Ilustración 2-24 - Clases del Modelo..... | 52 |
| Ilustración 2-25 - Implementación de objeto del contexto | 53 |
| Ilustración 2-26 - Modelado por medio de xcdatamodeld | 53 |
| Ilustración 2-27 - Controlador delegado..... | 54 |
| Ilustración 2-28 - Interfaz de consulta al motor de persistencia | 55 |
| Ilustración 2-29 - Interfaz de inserción en el motor de persistencia | 56 |
| Ilustración 2-30 - Controlador de sincronía | 57 |
| Ilustración 2-31 - Captura de pantalla de ThemeRoller | 58 |
| Ilustración 2-32 - Captura de pantalla del encabezado de la aplicación | 59 |
| Ilustración 2-33 - Captura de pantalla de la barra de herramientas de la aplicación..... | 59 |

| | |
|--|-----|
| Ilustración 2-34 - Captura de pantalla de la página principal | 60 |
| Ilustración 2-35 - Captura de pantalla del programa del congreso | 61 |
| Ilustración 2-36 - Captura de pantalla del listado de autores..... | 62 |
| Ilustración 2-37 - Captura de vista de detalle de un autor | 63 |
| Ilustración 2-38 - Grupo Views..... | 65 |
| Ilustración 2-39 – <i>JSController</i> | 66 |
| Ilustración 2-40 - Programa para el día 9, desplegado a las 9:00 | 69 |
| Ilustración 2-41 - Distribución de versiones de Android a junio de 2015..... | 73 |
| Ilustración 5-1 - Captura de pantalla de la página principal | 85 |
| Ilustración 5-2 - Captura de pantalla de la página de inicio de sesión..... | 86 |
| Ilustración 5-3 - Captura de pantalla del programa completo..... | 87 |
| Ilustración 5-4 - Captura de pantalla de la página de actualizaciones..... | 88 |
| Ilustración 5-5 - Captura de pantalla del listado de autores..... | 89 |
| Ilustración 5-6 - Captura de pantalla del listado de temáticas | 90 |
| Ilustración 5-7 - Captura de pantalla del listado de palabras clave | 91 |
| Ilustración 5-8 - Captura de pantalla del listado de comités | 92 |
| Ilustración 5-9 - Captura de pantalla del listado de noticias | 93 |
| Ilustración 5-10 - Captura de pantalla del listado de actividades sociales | 94 |
| Ilustración 5-11 - Captura de pantalla del listado de patrocinadores | 95 |
| Ilustración 5-12 - Captura de pantalla de la vista de detalle de una sesión | 96 |
| Ilustración 5-13 - Captura de pantalla de la vista de detalle de un autor..... | 97 |
| Ilustración 5-14 - Captura de pantalla de la vista de detalle de una temática | 98 |
| Ilustración 5-15 - Captura de pantalla de la vista de detalle de una palabra clave | 99 |
| Ilustración 5-16 - Captura de pantalla de la vista de detalle de una noticia..... | 100 |
| Ilustración 5-17 - Captura de pantalla de la vista de detalle de una actividad social..... | 101 |
| Ilustración 5-18 - Diagrama de flujo de la aplicación..... | 102 |
| Ilustración 5-19 - Botón Atrás | 103 |
| Ilustración 5-20 - Menú principal..... | 104 |
| Ilustración 5-21 - Programa | 105 |
| Ilustración 5-22 - Días del programa..... | 106 |
| Ilustración 5-23 - Despliegue de horas de un día del programa | 107 |
| Ilustración 5-24 - Mi programa | 108 |
| Ilustración 5-25 - Detalle de una charla..... | 109 |
| Ilustración 5-26 - Descripción de la charla..... | 110 |
| Ilustración 5-27 - Autores de una charla..... | 111 |
| Ilustración 5-28 - Actualización..... | 112 |

| | |
|---|-----|
| Ilustración 5-29 - Listado de autores | 113 |
| Ilustración 5-30 - Filtrado de autores | 114 |
| Ilustración 5-31 - Detalle de un autor | 115 |
| Ilustración 5-32 - Perfil de un autor | 116 |
| Ilustración 5-33 - Sesiones de un autor | 117 |
| Ilustración 5-34 - Listado de temáticas..... | 118 |
| Ilustración 5-35 - Filtrado de temáticas | 119 |
| Ilustración 5-36 - Detalle de una temática..... | 120 |
| Ilustración 5-37 - Descripción de una temática | 121 |
| Ilustración 5-38 - Sesiones de una temática | 122 |
| Ilustración 5-39 - Listado de palabras clave | 123 |
| Ilustración 5-40 - Filtrado de palabras clave | 124 |
| Ilustración 5-41 - Detalle de una palabra clave | 125 |
| Ilustración 5-42 - Listado de comités | 126 |
| Ilustración 5-43 - Comité organizador | 127 |
| Ilustración 5-44 - Comité del programa | 128 |
| Ilustración 5-45 - Listado de noticias | 129 |
| Ilustración 5-46 - Detalle de una noticia | 130 |
| Ilustración 5-47 - Listado de eventos..... | 131 |
| Ilustración 5-48 - Detalle de un evento | 132 |
| Ilustración 5-49 - Listado de patrocinadores | 133 |
| Ilustración 5-50 - Mi cuenta | 134 |

Estructura del documento

Para el desarrollo de esta memoria, hemos decidido dividirla en tres capítulos principales claramente acotados. Siguiendo a estos tres capítulos, hemos situado las fuentes de información y una serie de anexos disponibles para su consulta. La estructura de estos tres capítulos es la siguiente:

- **Capítulo 1: Punto de partida:** En este capítulo introducimos el proyecto y su contexto. Presentamos al mismo tiempo un análisis de mercado de las aplicaciones relevantes en el entorno, definimos los objetivos a cubrir y planteamos una serie de márgenes normativo-legales necesarios para el desarrollo del proyecto. Finalmente, señalamos las aportaciones de este proyecto a niveles socio-económico y científico-técnico, y analizamos las competencias que hemos desarrollado a través de este Trabajo de Fin de Grado.
- **Capítulo 2: Elaboración del proyecto:** Este capítulo se centra en el propio proceso de trabajo que hemos llevado a cabo. Aquí, definimos la metodología utilizada y las herramientas que hemos seleccionado para cada etapa. Presentamos también las etapas de análisis y diseño de nuestro proceso de desarrollo, y describimos las diferentes etapas de la implementación del sistema a nivel técnico.
- **Capítulo 3: Conclusiones y trabajos futuros:** Durante este capítulo cerraremos la memoria a través de dos secciones. Por un lado, expondremos nuestras conclusiones y reflexiones obtenidas tras terminar el proyecto. Para terminar, procederemos a detallar los posibles trabajos futuros.

Este proyecto, tal y como se describe en la introducción de la memoria, se ha dividido en dos proyectos de percepción similar, persiguiendo con cada uno el desarrollo híbrido en Android y en iOS, respectivamente. Esto plantea que, en general, las memorias para los proyectos "**Desarrollo de una Aplicación Móvil Híbrida de Soporte para Congresos en Android**" y "**Desarrollo de una Aplicación Móvil Híbrida de Soporte para Congresos en iOS**" serán fundamentalmente idénticas, al compartir una gran cantidad de recursos ajenos a la implementación nativa. Aunque ambas memorias son documentos completos, las secciones únicas a cada plataforma, y que por tanto difieren entre sí, son las siguientes:

- **Sección 2.2.2:** Herramientas específicas a la plataforma
- **Sección 2.5.1:** Modelado
- **Sección 2.5.2:** Conexión con el servidor *Back-End*
- **Sección 2.5.4:** Integración híbrida (excepto sección 2.5.4.4, Casos especiales)
- **Comienzo del anexo IV – Manual de usuario**

Estructura del CD

El CD que acompaña a este documento presenta la siguiente estructura:

- **EII-GII-2014-07-MolinaPerez_R-QuesadaArencibia_A-DAMHSCI.pdf** – Este documento, en formato PDF.
- **Manual.pdf** – Manual de usuario, en formato PDF. Este manual también se encuentra en el [anexo IV](#).
- **Resumen.txt** – Resumen en español de este documento, en formato TXT.
- **Abstract.txt** – Resumen en inglés de este documento, en formato TXT.
- **Project.zip** – Carpeta comprimida con el proyecto de Xcode, que incluye el código fuente de la aplicación. Esta carpeta está pensada para agregar el proyecto a Xcode, y ejecutarlo o modificarlo.
- **Source/** – Carpeta que incluye el código fuente con la jerarquía de subcarpetas pertinente para facilitar su lectura.

1 Punto de partida

1.1 Introducción

Este proyecto trata sobre el desarrollo de una aplicación híbrida destinada a dar soporte a la gestión de congresos. Este tipo de sistemas necesita un uso eficiente de grandes volúmenes de datos, por lo que hemos decidido desarrollar este manejo de datos de manera nativa, al ser esta una forma directa y eficiente de tratar los datos.

Esta aplicación parte de la idea de una aplicación ya existente, desarrollada para los congresos *Intelligent Vehicles Symposium (IV'14)* e *Intelligent Transportation Systems Conference (ITSC 2014)*, la cual fue desarrollada por completo mediante tecnologías web. Se trata de dos importantes congresos de carácter internacional centrados en el transporte inteligente, impulsados por una de las sociedades profesionales de IEEE, la *Intelligent Transport Systems Society*. Aunque nuestra aplicación se haya basado en un proyecto ya existente, todo su desarrollo ha partido de cero, omitiendo algunas etapas del análisis inicial que serán consideradas posteriormente en este documento.

Debido a la alta complejidad del desarrollo de vistas en nativo y al límite de tiempo establecido por el proyecto, tomamos la decisión de implementar las vistas usando elementos del desarrollo web, si bien inicialmente no era la intención. El uso de componentes web en una aplicación móvil ofrece un rendimiento subóptimo frente a los propios elementos de la plataforma, pero reduce enormemente el tiempo de desarrollo, consiguiendo obtener un proyecto abarcable en el tiempo disponible.

Esta decisión nos ha abierto un punto interesante, que es el estudio de la viabilidad del desarrollo híbrido, determinar la diferencia de complejidad y de eficiencia con respecto al desarrollo puramente web en las aplicaciones móviles. Por tanto, también contemplamos este camino durante el desarrollo de nuestro proyecto.

Es importante indicar que se ha dividido este proyecto en dos vertientes hermanas: Una con desarrollo nativo en Android, desarrollada por Adán Alonso Salvador, y la otra en iOS, desarrollada por Ruymán Molina Pérez. Esto ha permitido plantear el proyecto desde una visión más generalista que si se hubiera limitado a una única plataforma, y ampliar su alcance al compartir recursos en las etapas de desarrollo no nativo, permitiendo analizar las diferencias de cada plataforma. En este documento se especificarán los detalles correspondientes únicamente a una de las dos implementaciones, diferenciándose del documento hermano esencialmente en las secciones dedicadas al desarrollo propio de la plataforma.

Este proyecto se divide en diferentes partes. Para empezar, y debido al uso de tecnologías nativas en las cuales no teníamos ningún tipo de instrucción, hemos tenido que pasar un periodo de formación en el cual se ha invertido una cantidad considerable del tiempo que disponíamos para el desarrollo de nuestro proyecto.

Tras este primer contacto con dichas tecnologías y sus peculiaridades, pudimos comenzar el análisis del proyecto, determinando los actores y diversos casos de uso que formarían parte del mismo. Posteriormente, identificamos el tipo de herramientas que necesitaríamos utilizar durante el desarrollo, teniendo en algunos casos que comparar entre diferentes opciones eligiendo las que mejor se ajustasen a nuestros objetivos de eficiencia y abarcabilidad.

Finalmente, procedimos al desarrollo del proyecto, siguiendo un modelo incremental inspirado en diferentes principios ágiles y siguiendo el patrón de arquitectura de software Modelo-Vista-Controlador, centrándonos primero en el desarrollo de un *Middle-End* nativo capaz de obtener los datos desde el servidor *Back-End* ya existente y proveer a nuestras vistas web *Front-End* con los datos de manera local.

Tras obtener este *Middle-End* funcional, procedimos a desarrollar una maqueta completa que representara visualmente el estado final de la aplicación, para posteriormente conectarla con todos los componentes anteriores consiguiendo así la aplicación final.

1.2 Contextualización

1.2.1 Congresos e introducción a la gestión de congresos

Durante los congresos se llevan a cabo una serie de actividades como charlas, conferencias, visitas a lugares importantes, comidas, etc. Uno de los objetivos de nuestro proyecto es facilitar el trabajo de la organización, suministrando toda la información relativa a los eventos del congreso de una manera fácil, accesible y rápida a todos los usuarios del congreso que dispongan de un *smartphone* o tableta. Con nuestra aplicación, los asistentes al congreso podrán encontrar diversa información acerca del congreso, reduciendo así la carga de trabajo de la organización y mejorando el tiempo de respuesta, y por tanto la satisfacción de los asistentes.

1.2.2 Aplicaciones móviles nativas y aplicaciones web móviles

El desarrollo de aplicaciones móviles posee multitud de vertientes con características diferentes. De cara a este proyecto, es fundamental diferenciar entre lo que denominaremos una aplicación nativa y una aplicación web, dentro del entorno del desarrollo móvil.

El desarrollo tradicional de una aplicación móvil es lo que, a los efectos de este proyecto, definimos como aplicación nativa. En el desarrollo nativo, toda la aplicación se implementa empleando componentes propios al lenguaje de desarrollo seleccionado para la plataforma, a todos los niveles del sistema. Los lenguajes de desarrollo móvil de hoy en día proporcionan multitud de herramientas para llevar a cabo estos procesos, facilitando librerías que nos permiten acceder a los diferentes recursos disponibles en la plataforma.

En el extremo opuesto de nuestro gradiente particular, podemos encontrar lo que denominaremos aplicaciones web móviles. Estas aplicaciones nacen del desarrollo web prácticamente al completo, y se diseñan con estas plataformas en mente. Existe un gran abanico de herramientas y marcos de trabajo que permiten llevar a cabo proyectos de estas características de forma satisfactoria. El producto final habitualmente se presenta como una página web tradicional, a la que es posible acceder desde el móvil, o como una aplicación nativa en la que se incorpora un componente equivalente a un navegador web.

Es importante destacar que, aunque se trate de soluciones basadas en el desarrollo web, esto no hace que la aplicación final dependa completamente de la red para funcionar, sino que puede ofrecer diversos servicios fuera de línea. Aunque el rendimiento que estas aplicaciones presentan frente a soluciones puramente nativas tiende a ser severamente inferior, el proceso de desarrollo resulta habitualmente más ágil y rápido. También otorgan una gran ventaja en portabilidad, al no requerir desarrollar más que una única aplicación web para todas las plataformas.

| Desarrollo nativo | Desarrollo web |
|---|---------------------------------------|
| Alto rendimiento | Bajo rendimiento |
| Acceso directo a todos los recursos del dispositivo | Limitado a los recursos del navegador |
| Requiere desarrollo por plataforma | Portabilidad |
| Desarrollo complejo | Desarrollo rápido |

Tabla 1-1 - Comparativa del desarrollo nativo y el desarrollo web en dispositivos móviles

1.2.3 Desarrollo híbrido

Tras explicar los dos tipos de aplicaciones móviles que comúnmente encontramos, planteamos un punto intermedio: El desarrollo híbrido. Este tipo de desarrollo intenta combinar lo mejor de los sistemas anteriores, consiguiendo un rendimiento aceptable en un tiempo de implementación relativamente corto y con un consumo de recursos moderado. Este método híbrido permite al desarrollador emplear todas las herramientas que le puede proporcionar un entorno nativo, otorgando gran flexibilidad al sistema.

Por ello, en nuestra aproximación particular, hemos decidido limitar los componentes web a la interfaz gráfica de la aplicación, permitiendo a los componentes nativos llevar a cabo una gestión óptima de la información, penalizando el rendimiento de la visualización. La solución híbrida permite, en cierta medida, la flexibilidad de las posibilidades nativas, combinada con la rapidez y portabilidad (en sus ámbitos) del desarrollo web.

1.3 Estado del arte

Inicialmente, hemos realizado un breve análisis de los congresos en los que se centra la aplicación de referencia para nuestro proyecto. La herramienta de la que partimos es una aplicación guía para los congresos *Intelligent Transportation Systems Conference* e *Intelligent Vehicles Symposium*, dos de los congresos de sistemas de transporte inteligente impulsados por la sociedad de transporte inteligente de IEEE (ITS, *Intelligent Transport Society*).

La ITS surge con el comienzo del año 1999, uniéndose a las filas de las otras 37 sociedades y consejos de IEEE en aquella época. La ITS nace como una sociedad interesada en aspectos teóricos, experimentales y operacionales de las ingenierías eléctrica y electrónica y las tecnologías de la información, con el objeto de desarrollar y mejorar sistemas de transporte de todo tipo. En sus comienzos, la ITS partió con miembros fundadores de hasta 18 sociedades diferentes interesadas en el ámbito del transporte inteligente. Los representantes de estas sociedades proporcionan un amplio espectro de trasfondos de grandes compañías, consultorías, academias y asociaciones de intercambio.



Ilustración 1-1 - Logotipo de la ITS

Esta sociedad patrocina, entre otros muchos más, los dos congresos para nuestra aplicación de referencia, IV (*Intelligent Vehicles Symposium*) e ITSC (*Intelligent Transportation Systems Conference*). Nuestro proyecto se ha basado en las instancias de estos congresos del año 2014.

| | IV'14 | ITSC 2014 |
|----------------------|---|--|
| Título | <i>Intelligent Vehicles Symposium</i> | <i>Intelligent Transportation Systems Conference</i> |
| Logotipo |  |  |
| Género | Transporte inteligente | Transporte inteligente |
| Localización | Variable. En 2014, Dearborn, Michigan, Estados Unidos | Variable. En 2014, Qingdao, China |
| Fecha | Junio de 2014 | Octubre de 2014 |
| Frecuencia | Anual | Anual |
| Participantes | 500 | 800 - 1.500 |
| Página web | Enlace | Enlace |

Tabla 1-2 - Congresos de la ITS

Además de la aplicación de referencia que nos fue proporcionada, para determinar el estado del arte actual para el desarrollo de nuestra aplicación, hemos decidido estudiar una serie de aplicaciones que hemos encontrado en las diferentes plataformas móviles. El criterio de selección de las aplicaciones ha sido por un lado buscar las aplicaciones oficiales de algunos de los congresos de electrónica de consumo y videojuegos más importantes a nivel internacional (*CES*, *MWC*, *gamescom* y *E3*), y por otro lado aplicaciones populares en las correspondientes plataformas que aparecen cuando se efectúa una búsqueda con los términos correspondientes (*DOG*). Esto nos permite obtener referencias de algunos de los congresos de mayor escala en la actualidad.

| | <i>CES</i> | <i>MWC</i> | <i>gamescom</i> | <i>E3</i> | <i>DOG</i> |
|---------------------|---|---|---|---|---|
| Título | <i>Computer Electronics Show</i> | <i>Mobile World Congress</i> | <i>gamescom</i> | <i>Electronic Entertainment Expo</i> | <i>Deutsche Ophthalmologische Gesellschaft</i> |
| Logotipo |  |  |  |  |  |
| Género | Electrónica de consumo | Telefonía móvil | Videojuegos | Videojuegos | Oftalmología |
| Localización | Las Vegas, Nevada, Estados Unidos | Barcelona, España | Colonia, Alemania | Los Ángeles, California, Estados Unidos | Berlín, Alemania |
| Inauguración | 1967 | 1987 | 2009 | 1995 | 1857 |
| Frecuencia | Anual | Anual | Anual | Anual | Anual |
| Asistentes | 140.000 (2015) | 85.000+ (2014) | 335.000 (2014) | 48.900 (2014) | 6.600+ (2014) |
| Página web | Enlace | Enlace | Enlace | Enlace | Enlace |

Tabla 1-3 - Resumen de los congresos seleccionados

Una descripción en mayor profundidad de las características de estos congresos puede encontrarse en el [Anexo I](#).

Las aplicaciones analizadas para cada uno de estos congresos se describen en la tabla que se muestra a continuación. No se incluyen estadísticas de descargas y valoraciones en iOS puesto que en el *App Store* de Apple esta información no se encuentra disponible públicamente. También incluimos las estadísticas de la aplicación de referencia del ITSC 2014 mencionada anteriormente.

| | <i>IEEE ITSC Conference 2014</i> | <i>2015 International CES</i> | <i>My MWC - Official GSMA MWC App</i> |
|--|---|--|---|
| Congreso | <i>ITSC 2014</i> | <i>CES</i> | <i>MWC</i> |
| Icono |  |  |  |
| Aplicación Android | Play Store | Play Store | Play Store |
| Aplicación iOS | App Store | App Store | App Store |
| Tamaño Android | 3,2 MB | 26 MB | 30 MB |
| Tamaño iOS | 3,9 MB | 33,9 MB | 50,9 MB |
| Requiere Android | 2.3.3+ | 4.1 - 5.0 | 4.0+ |
| Requiere iOS | 6.0+ | 7.0+ | 7.0+ |
| Descargas Android (27/02/2015) | 100 - 500 | 10.000 - 50.000 | 10.000 - 50.000 |
| Valoración Android (27/02/2015) | 5/5 | 3,2/5 | 3,4/5 |
| Votos Android (27/02/2015) | 4 | 325 | 447 |
| Desarrollador | IUCTC | Konvuri | Genie-Connect |
| | <i>gamescom - The Official Guide</i> | <i>E3 2014</i> | <i>DOG Congress</i> |
| Congreso | <i>gamescom</i> | <i>E3</i> | <i>DOG</i> |
| Icono |  |  |  |
| Aplicación Android | Play Store | Play Store | Play Store |
| Aplicación iOS | App Store | App Store | App Store |
| Tamaño Android | 20 MB | 3,6 MB | 13 MB |
| Tamaño iOS | 20,4 MB | 13.3 MB | 12,5 MB |
| Requiere Android | 4.0+ | 2.2+ | 2.2+ |
| Requiere iOS | 6.1+ | 5.1.1+ | 6.0+ |
| Descargas Android (27/02/2015) | 50.000 - 100.000 | 10.000 - 50.000 | 100 - 500 |
| Valoración Android (27/02/2015) | 3,5/5 | 4,0/5 | 4,0/5 |
| Votos Android (27/02/2015) | 1.063 | 583 | 3 |
| Desarrollador | Inside | Core-Apps | Nfonix |

Tabla 1-4 - Aplicaciones de congresos en Android y iOS

1.3.1 Conclusiones obtenidas

Actualmente existen unas pocas aplicaciones orientadas al soporte de la gestión de congresos. No obstante, son en su mayoría aplicaciones de gran calado generadas para congresos de eventos de importancia mundial, realizadas a medida para un único cliente. Estas aplicaciones no son baratas y no tienen una fácil reutilización.

Debido a esto, nuestra aplicación debe ser estandarizable, permitiendo un despliegue sencillo y adaptable para diferentes congresos y consiguiendo minimizar los costes del software hecho a medida, con lo que poder lograr ser un software asequible y que esté en la mano de congresos de menor calado. Adaptar la aplicación a un nuevo congreso no debería ser más complicado que añadir los nuevos datos en una sección diferente de nuestro servidor *Back-End* y configurar la aplicación para que acceda a dicho congreso. Para ello debemos solicitar los datos en un formato preestablecido a los clientes los cuales introduciremos en nuestro *Back-End*, y con esto ya podríamos desplegar la aplicación adaptada al nuevo congreso, lo cual reduce los costes de portabilidad a prácticamente lo que nos cueste migrar la información y el espacio que esto suponga para nuestros servidores.

Tras el análisis de las diversas aplicaciones mencionadas, obtuvimos una serie de características deseables para nuestro software. Cabe mencionar que, debido a las limitaciones en el tiempo de desarrollo, no ha sido posible implementarlas todas. Aquellas características que no fueron implementadas serán mencionadas en la sección final de conclusiones y trabajos futuros si se consideran oportunas. Pasamos a analizar dichas conclusiones:

- **Organización de la información:** Hemos comprobado que la información debe organizarse de manera sencilla y accesible, permitiendo al usuario saber en todo momento dónde se encuentra en la aplicación de manera simple.
- **Interfaz de uso:** Debe ser sencilla y directa, ya que hemos visto que algunas de las aplicaciones estudiadas son tan complejas que incluso incorporan tutoriales de uso. Deberíamos conseguir que nuestro software tenga un uso intuitivo y evitar la necesidad de tutoriales.
- **Usuarios registrados de pago:** La aplicación podría contar con un sistema de usuarios registrados capaz de dar acceso a contenidos exclusivos, como por ejemplo la descarga de los documentos de cada charla. Esto podría generar un valor añadido tanto para los asistentes del congreso como para sus organizadores.
- **Contenido útil adicional ajeno al congreso:** Sería interesante añadir una serie de elementos de utilidad para el asistente al congreso, como por ejemplo opciones de hospedaje, información sobre rutas y transportes al centro o lugares de interés en la zona. Este sistema conseguiría lograr dos objetivos: Por un lado mejorar la visita del asistente al congreso, sugiriéndole opciones de calidad, y por el otro dar la opción a empresas externas del sector servicios a promocionarse dentro de nuestra aplicación.
- **Incorporación de un mapa:** La aplicación deberá incluir un mapa del recinto en el que tenga lugar el congreso, facilitando a los visitantes la localización de los eventos del congreso.
- **Cronología de eventos:** Se espera poder encontrar en la aplicación un listado ordenado de los eventos que tienen lugar en el congreso, incluyendo información de hora y localización para cada uno.
- **Sistema de favoritos:** Los usuarios de la aplicación podrán marcar como favoritos algunos elementos, de manera que puedan generar su plan personal de visita por el congreso. Este sistema de favoritos puede implementarse de forma local a la aplicación, o integrarse en el sistema de cuentas de usuario.

- **Rendimiento y estabilidad:** Sería ideal minimizar los tiempos de carga de la aplicación en su arranque (o notificar al usuario al respecto, si no puede ser evitado. Se espera al mismo tiempo que el tamaño de la aplicación se mantenga por debajo de unos máximos aceptables, y que sea compatible con la mayor cantidad de dispositivos del mercado posible.
- **Elemento social:** Los usuarios de la aplicación podrán valorar las sesiones, sirviendo como retroalimentación tanto para otros usuarios como para el propio congreso. La integración de la aplicación con las redes sociales también es un elemento interesante a explotar, enlazando por ejemplo los perfiles de los ponentes en *LinkedIn* o *Facebook* con el perfil de nuestra aplicación.

1.4 Objetivos

El objetivo principal de nuestro proyecto es **construir una aplicación para móviles capaz de dar soporte a la gestión de congresos valiéndose del desarrollo híbrido para dichas plataformas**. Nuestro proyecto posee a su vez una serie de objetivos secundarios, algunos de ellos directamente relacionados con la aplicación en sí, y otros con la forma en la que la misma ha sido implementada.

1.4.1 Objetivos secundarios de la aplicación a desarrollar

Relativos a la aplicación, tenemos los siguientes objetivos secundarios:

1. **La aplicación debe ser capaz de tratar grandes volúmenes de datos de manera eficiente y rápida.** Para ello, hemos decidido utilizar el desarrollo nativo ya que, como anteriormente hemos mencionado, proporciona acceso directo a los recursos del dispositivo con la correspondiente mejora en eficiencia.
2. **La generación de las vistas debe ser implementada de manera rápida y sencilla,** debido a la cantidad de tiempo disponible y la necesidad de formación en el ámbito de las tecnologías nativas. Para poder conseguir realizar estas vistas en tiempo y forma, hemos optado por el desarrollo web, cuyas características y nuestro conocimiento del mismo nos han permitido cumplir los plazos acordados.
3. **Proveer a los participantes con un sistema que les facilite el acceso a la información relativa al congreso de manera ágil, concisa y cómoda.** Para ello la aplicación contará con una serie de **secciones** que pretenden satisfacer las necesidades de los asistentes. Entre ellas, se destacan las siguientes:
 - **Programa del congreso:** Será el portal de acceso a la aplicación desde el que podremos acceder al resto de secciones y será accesible por medio de un botón desde cualquier otra sección, ya que esta es la más importante de ellas y debe ser accesible en cualquier momento.
 - **Sección de autores:** La vista inicial de esta sección constará de un listado de los autores que participan en el congreso y una herramienta de búsqueda que permita realizar filtrados sobre dicho listado. Desde este listado podremos acceder a la vista de detalle de los autores en la cual se podrá acceder a datos más precisos del autor, así como a un listado de las sesiones en las que participará.

- **Sección de temáticas:** En ella se mostrará un listado de todos los géneros a tratar durante el congreso, contando a su vez con una herramienta de búsqueda capaz de filtrar dicho listado. A través de esta sección podemos acceder a cualquier tema y obtener información detallada del mismo así como acceso a las sesiones en las que se tratará dicho tema.
- **Sección de palabras clave:** En esta sección encontraremos un conjunto de etiquetas de relevancia en el ámbito del congreso que nos permitirán encontrar las sesiones en las que se establezca una relación con las mismas.
- **Sección de comités:** En esta vista se mostrarán los distintos tipos de comités que participarán en el congreso, así como los miembros que forman parte de cada uno de ellos. Existen dos tipos de comités: El organizativo, que consiste en el grupo de personas que se encargan del funcionamiento correcto y fluido del congreso, y el del programa, que se compone de las personas encargadas del control, introducción y evaluación de las charlas que se ofrecen durante el congreso.
- **Sección de noticias:** Bajo esta categoría encontraremos las noticias que consideren oportuno publicar los organizadores durante el congreso. Tendrá una vista inicial de listado desde la que poder acceder a una sección en detalle de cada noticia en detalle.
- **Sección de actividades sociales:** En esta sección se mostrarán los eventos ajenos a las charlas que se organizarán durante el evento. Siguiendo el esquema del resto de secciones contará con un listado y una vista en detalle de cada actividad.
- **Sección de patrocinadores:** Esta vista cumple el cometido de publicitar y agradecer a los patrocinadores del congreso, por medio de imágenes y enlaces a sus portales web.

1.4.2 Objetivo secundario: Estudio de la viabilidad del desarrollo híbrido

Adicionalmente, el desarrollo de la aplicación persigue un objetivo de carácter secundario en el que se plantea **estudiar la viabilidad del desarrollo híbrido** frente a otros sistemas. Este estudio se centrará en elementos de rendimiento percibido de la aplicación final, así como en los recursos que han sido necesarios para implementarla.

1.5 Normativa y legislación

1.5.1 Ley de Protección de Datos de Carácter Personal

La Ley Orgánica 15/1999 de 13 de diciembre es una ley orgánica española que tiene por objetivo garantizar y proteger las libertades públicas y los derechos fundamentales de las personas físicas en lo concerniente al tratamiento de los datos personales, con especial énfasis en el honor, la intimidad y la privacidad personal y familiar.

Esta ley fue aprobada en las Cortes Españolas el 13 de diciembre de 1999, fundamentándose en el artículo 18 de la constitución española de 1978 el cual, aparte de lo descrito anteriormente, también contempla que se garantice el secreto de las comunicaciones y la limitación del uso de la informática en tanto esto suponga un perjuicio para el honor y la intimidad de las personas.

Esta ley cataloga los ficheros según la naturaleza de los datos que contienen. Acorde con ello, se establecen diferentes niveles de seguridad que garanticen en mayor o menor medida la confidencialidad e integridad de dicha información. Los niveles que se ven descritos son los siguientes:

- **Nivel Alto:** Se considerarán de este nivel aquellos ficheros que contengan datos relativos a la ideología, afiliación social, religión, creencias, origen racial, salud o vida sexual. También forman parte de este nivel los ficheros recabados con fines policiales sin consentimiento de las personas afectadas o aquellos derivados de actos de violencia de género.
- **Nivel Medio:** Dentro de este nivel se encuentran aquellos ficheros que contengan datos relativos a la comisión de infracciones administrativas o penales, prestación de servicios de administraciones tributarias, relaciones con entidades financieras, gestorías o servicios comunes de seguridad social. También formarán parte de este nivel aquellos ficheros que contengan datos relativos a enfermedades profesionales o accidentes de trabajo, o aquellos que definan la personalidad o permitan evaluar determinados aspectos de la misma, o analizar el comportamiento de las personas. Finalmente este nivel incluye aquellos datos que permitan la localización de una persona (por ejemplo, datos de tráfico).
- **Nivel Básico:** Cualquier otro tipo de ficheros que contenga datos de carácter personal. Las medidas de seguridad de este nivel son exigibles en cualquiera de los casos. Cabe mencionar que bajo algunas excepciones ficheros de niveles superiores pueden ser considerados básicos.

Nuestra aplicación hace uso de varios datos de carácter personal de los autores, ya que en ella se muestran ciertos datos como su nombre completo, la entidad a la que representan, una foto y su nacionalidad. Siendo este el único fichero con datos de carácter personal y ajustándose a la lista de niveles anteriores, tenemos un fichero de nivel básico.

Para cumplir la LOPD se establecerán una serie de condiciones que deberán aceptar los autores que quieran verse representados en la aplicación, indicándoles claramente la naturaleza de nuestro software y solicitándoles expresamente su conformidad para que sus datos puedan ser accedidos públicamente, a la vez que les garantizamos sus derechos ARCO (Acceso, Rectificación, Cancelación y Oposición) acorde con lo que se describe en esta ley.

1.5.2 Clasificación de licencias de software

Este apartado se incluye como una contextualización en lo que a los términos de uso de un software se refiere. En la sección de herramientas utilizadas para el desarrollo de este proyecto se verán indicadas las licencias asociadas a cada herramienta.

Desde el punto de vista de los derechos reservados sobre el autor y de menor permisividad a mayor libertad, se encuentran las siguientes categorías de licencias de software:

- **Licencia de código cerrado o software propietario:** Por lo general, esta licencia no permite que el software sea modificado, desensamblado, copiado o distribuido por medio de otras licencias. Regula el número de copias que pueden ser instaladas o los fines con los que puede ser utilizado. Habitualmente implica la existencia de servicios de soporte técnico o actualizaciones durante el tiempo de vida del producto.
- **Licencia de código abierto robusta fuerte:** Se permiten obras derivadas de la original, pero deben ser licenciadas en los mismos términos y condiciones de la licencia original.
- **Licencia de código abierto robusta débil:** Se permiten modificaciones al software original licenciadas bajo los mismos términos y condiciones, pero las obras derivadas pueden ser licenciadas bajo términos y condiciones diferentes.
- **Licencia de código abierto permisiva:** Se permite la creación de obras derivadas sin obligación de protección.
- **Software de dominio público o sin licencia:** Se permite uso, copia, modificación o redistribución con o sin fines de lucro.

1.5.3 Reglamento que afecta al Trabajo de Fin de Grado

Tal y como se describe en los Estatutos de la Universidad de Las Palmas de Gran Canaria, publicados en el Decreto 30/2003, de 10 de marzo, Artículo 133:

1. *La realización de un proyecto fin de carrera o tesina llevará consigo, necesariamente, la elaboración de un trabajo en el ámbito disciplinario elegido, en régimen de tutoría. El Consejo de Gobierno establecerá un reglamento que defina e incentive la tutoría de proyectos y tesinas. El plan de estudios correspondiente establecerá la forma de evaluación de ese proyecto fin de carrera o tesina.*
2. *En virtud del artículo 7 del Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, el proyecto fin de carrera se considera una obra en colaboración entre el estudiante y el tutor o tutores, en su caso.*
3. *La explotación industrial de un proyecto fin de carrera debe ser objeto de convenio entre la Universidad y el organismo o empresa que la realizará. Para fomentar la realización de proyectos fin de carrera con empresas, la Universidad podrá, en el convenio citado, ceder la propiedad industrial de los trabajos contenidos en el proyecto fin de carrera.*

1.6 Aportaciones

1.6.1 Aportaciones a nivel social y económico

El desarrollo de esta aplicación permitirá a sus usuarios obtener información útil en su visita a los congresos, permitiendo a los asistentes optimizar su tiempo y conocer, fácilmente, qué eventos tienen lugar durante el congreso, así como su localización espacio-temporal.

Por otro lado, el uso de esta aplicación desde el punto de vista de la gestión del congreso permite agilizar los procesos de asistencia informativa a los visitantes, obteniendo una herramienta potente de una forma económica comparado a lo que podría implicar el desarrollo de una aplicación a medida desde cero. En nuestra aplicación, como se ha indicado anteriormente, resulta trivial añadir una sección para un nuevo congreso en los servidores, agregando un importante factor de reutilizabilidad a nuestro proyecto.

Adicionalmente, como ya comentamos con anterioridad, cabría la posibilidad de añadir lugares de interés, con la finalidad de acercar a los asistentes del congreso al lugar en el que este se celebra, consiguiendo así fomentar el sector servicios de la zona local.

1.6.2 Aportaciones a nivel científico y técnico

Este proyecto provee un sistema de fácil implantación que lo convierte en altamente reusable a muy bajo coste. Esto lo diferencia claramente de la mayoría de los sistemas analizados al tratarse de aplicaciones hechas a medida que cada año generan una nueva versión muy diferente de la anterior. Hemos intentado plantear una solución capaz de perdurar por varias iteraciones con bajo costo de mantenimiento.

Cabe mencionar además que a través del desarrollo híbrido de nuestra solución, se abren posibles vías futuras de investigación que podrán, partiendo del trabajo planteado en este proyecto, explotar al máximo lo que realmente supone el desarrollo híbrido en todo su potencial. Consideramos que, aunque este proyecto resulte un paso importante en la buena dirección, aún queda mucho camino por andar.

1.7 Competencias desarrolladas

Las tareas que abarcan este proyecto han permitido desarrollar y cubrir las competencias propuestas para superar la asignatura de Trabajo de Fin de Grado en el curso 2014-2015, tal y como se describen en el proyecto docente correspondiente:

G1. Poseer y comprender conocimientos en un área de estudio (Ingeniería Informática) que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

G2. Aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

- **Justificación G1 y G2:** Durante el desarrollo de este proyecto hemos tenido que buscar información de primera mano, tanto en manuales y guías como solución a problemas referentes a las versiones actuales de las herramientas. Hemos tenido que lidiar con errores recientes que no están contemplados en los manuales, con lo que hemos debido resolver problemas actuales valiéndonos de nuestras propias capacidades y aplicar soluciones a problemas similares que hemos conseguido encontrar en las comunidades en línea. Por otro lado, el mero hecho de completar este proyecto indica, sin lugar a dudas, que hemos sido capaces de aplicar nuestros conocimientos informáticos de una manera profesional.

G3. Reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

- **Justificación G3:** En la sección del estado del arte de esta memoria hemos obtenido datos a partir de otros desarrollos similares y extraído una serie de conclusiones que hemos querido plasmar en el desarrollo de nuestra aplicación. Teniendo en cuenta los rasgos positivos y negativos de dichas aplicaciones de cara al usuario, hemos intentado conseguir una solución de bajo coste que pueda llegar al máximo de clientes posibles, considerando qué opciones deben quedar descartadas para garantizar la abarcabilidad del proyecto actual. Por otro lado, en la sección de selección de herramientas hemos tenido que comparar y elegir entre diversas herramientas siguiendo criterios de eficiencia, sencillez y recursos disponibles.

G4. Transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

- **Justificación G4:** En la medida de lo posible todo el desarrollo de esta memoria está expresado de manera sencilla e intenta no ahondar en palabras excesivamente técnicas, procurando ser útil para un público menos especializado, aunque a su vez en las secciones más específicas debemos ahondar en temas bastante más especializados.

G5. Desarrollar aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

- **Justificación G5:** Para la realización de este proyecto fue necesario el aprendizaje autónomo del lenguaje nativo partiendo de cero, lo cual consumió una enorme cantidad del tiempo disponible, pero sin lugar a dudas esto es un ejemplo claro del aprendizaje de autónomo de un contenido que va más allá de lo impartido en el Grado.

N1. Comunicarse de forma adecuada y respetuosa con diferentes audiencias (clientes, colaboradores, promotores, agentes sociales, etc.), utilizando los soportes y vías de comunicación más apropiados (especialmente las nuevas tecnologías de la información y la comunicación) de modo que pueda llegar a comprender los intereses, necesidades y preocupaciones de las personas y organizaciones, así como expresar claramente el sentido de la misión que tiene encomendada y la forma en que puede contribuir, con sus competencias y conocimientos profesionales, a la satisfacción de esos intereses, necesidades y preocupaciones.

- **Justificación N1:** Es requisito indispensable de este proyecto la generación de este documento, así como la propia preparación de la presentación oral que acompaña a este trabajo. Estas tareas implican, para su correcta ejecución, un uso consciente y apropiado de las diferentes herramientas de comunicación disponibles, así como una expresión oral y escrita apropiada al registro en el que se sitúa el proyecto, de modo que este documento resulte fácilmente accesible para el público objetivo. Este proyecto también ha permitido fomentar la comunicación horizontal entre compañeros al tratarse, en gran medida, de un proyecto conjunto.

N2. Cooperar con otras personas y organizaciones en la realización eficaz de funciones y tareas propias de su perfil profesional, desarrollando una actitud reflexiva sobre sus propias competencias y conocimientos profesionales y una actitud comprensiva y empática hacia las competencias y conocimientos de otros profesionales.

- **Justificación N2:** Este proyecto ha exigido una coordinación con los desarrolladores del *Back-End* existente que nutre a nuestro proyecto. Al mismo tiempo, se ha procurado buscar opiniones expertas en los ámbitos considerados oportunos a lo largo del trabajo, facilitando nuestra toma de decisiones. Es importante mencionar de nuevo el factor del proyecto conjunto y la comunicación y coordinación que ello implica, tal y como se menciona en la justificación de la competencia N1.

N3. Contribuir a la mejora continua de su profesión así como de las organizaciones en las que desarrolla sus prácticas a través de la participación activa en procesos de investigación, desarrollo e innovación.

- **Justificación N3:** El desarrollo de este proyecto nace del corazón del Instituto Universitario de Ciencias y Tecnologías Cibernéticas como una vía de investigación hacia posibles enfoques futuros en el desarrollo de software, abriendo posibles caminos hacia el desarrollo de aplicaciones en el futuro.

N4. Comprometerse activamente en el desarrollo de prácticas profesionales respetuosas con los derechos humanos así como con las normas éticas propias de su ámbito profesional para generar confianza en los beneficiarios de su profesión y obtener la legitimidad y la autoridad que la sociedad le reconoce.

- **Justificación N4:** Hemos intentado desarrollar una aplicación que fuese de bajo coste, intentando ampliar el mercado a pequeños congresos que no tengan una economía tan opulenta. Con este proyecto se intenta demostrar que la implantación de un software de estas características está a la mano de todo el que lo necesite.

N5. Participar activamente en la integración multicultural que favorezca el pleno desarrollo humano, la convivencia y la justicia social.

- **Justificación N5:** Comprendemos con el desarrollo del proyecto la función vital de comunicación multicultural que un congreso supone, permitiendo nuevas vías de investigación a nivel internacional en cualquiera de los ámbitos en los que se centre este evento.

T1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. (G1, G2)

- **Justificación T1:** Ver la justificación de las competencias G1 y G2.

T2. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada. (G1, G2)

- **Justificación T2:** Ver la justificación de las competencias G1 y G2.

T3. Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan. (G1, G2)

- **Justificación T3:** Ver la justificación de las competencias G1 y G2.

T5. Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada. (G1, G2)

- **Justificación T5:** Ver la justificación de las competencias G1 y G2.

T6. Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada. (G1, G2)

- **Justificación T6:** Ver la justificación de las competencias G1 y G2.

T7. Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento. (N4)

- **Justificación T7:** Esta competencia se puede ver desarrollada en el apartado dedicado a la normativa y legislación que afecta a este proyecto en este mismo documento.

T8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

- **Justificación T8:** Ver justificación de la competencia G5.

T11. Capacidad para analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico en Informática.

- **Justificación T11:** Ver justificación de la competencia N5.

CII01. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

- **Justificación CII01:** Ver justificación de las competencias G1, G2, G3 y T7.

CII02. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

- **Justificación CII02:** Ver justificación de las competencias G1, G2, G4, N1 y N2.

CII04. Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

- **Justificación CII04:** Ver justificación de la competencia T7.

CII18. Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.

- **Justificación CII18:** Ver justificación de la competencia T7.

TFG01. Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sintetizen e integren las competencias adquiridas en las enseñanzas

- **Justificación TFG1:** El desarrollo de esta competencia se ve cubierto en la propia confección de este documento y proyecto, así como la presentación final a tener lugar ante el tribunal pertinente.

2 Elaboración del proyecto

2.1 Metodología

En lo que respecta al desarrollo del proyecto, hemos optado por seguir una línea incremental de desarrollo dividiendo el proyecto en bloques independientes. Para ello, concluimos que la manera óptima de trabajar pasaría por obtener una aplicación básica que simplemente administrara los datos del modelo correctamente sin apenas interfaz gráfica, y posteriormente añadir una visualización amigable como un módulo independiente.

Esto a su vez nos permitió dividir fácilmente el trabajo exclusivamente nativo a cada plataforma de los componentes comunes del desarrollo web. Siguiendo este esquema, seleccionamos el desarrollo nativo para implementar los modelos y los controladores del software nativamente, y el desarrollo web para formar las vistas finales de la aplicación.

En cuanto al desarrollo de las vistas, al tratarse de un componente relativamente independiente al resto del proyecto, pudimos plantear una breve etapa de prototipado para las mismas. De este modo, decidimos generar unas maquetas que representaran la visualización de las diferentes pantallas de la aplicación, presentando elementos de navegación básicos entre ellas. Esto nos permitió, por un lado, estudiar los aspectos gráficos de la aplicación y la distribución de los elementos de la interfaz, y por otro analizar la navegabilidad del sistema entre las diferentes zonas de la aplicación.

Una vez desarrollados los componentes de la maqueta, se incorporó esta capa al software real, integrando los datos existentes en el modelo de la aplicación con las vistas desarrolladas en las maquetas, conformando así el resultado de este proyecto.

2.1.1 Manifiesto ágil

No hemos seguido ningún tipo de metodología preestablecida al realizar este proyecto. Dado el pequeño número de desarrolladores (2 personas) que han intervenido, no consideramos asumible implantar SCRUM o XP debido al número recomendado de personas para ello o la falta de roles. Finalmente optamos por tener presente los principios del manifiesto ágil e intentar amoldar nuestra forma de trabajar a los mismos. A continuación exponemos los principios del manifiesto que consideramos más relevantes para nuestro proyecto y la manera en la que los hemos aplicado durante nuestro desarrollo.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

En un principio se orientó el proyecto a un desarrollo totalmente nativo, pero tras realizar el bloque de manejo de datos y comenzar el desarrollo de las vistas en nativo, nos dimos cuenta de la enorme complejidad del sistema y tuvimos que cambiar a un desarrollo híbrido. Por otro lado, algunas vistas se han visto alteradas debido a su enorme carga de datos, obligándonos a cambiar su desarrollo a otros sistemas (por ejemplo, paginando listados) con la finalidad de obtener una eficiencia aceptable.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Tras el desarrollo de la sección nativa, realizamos unas primeras vistas sencillas y una aplicación que las mostraba sin formato, con la finalidad de mostrar el sistema funcionando. A continuación generamos unas maquetas de la sección web de la aplicación, las cuales eran funcionales pero con datos estáticos. Finalmente fuimos integrando ambas partes vista a vista, teniendo cada vez una versión más completa de la aplicación final.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Durante todo el proyecto, ambos miembros hemos estado desarrollando conjuntamente en el mismo laboratorio. Del mismo modo, disponíamos de nuestro tutor Alexis Quesada Arencibia en dicho laboratorio, quien fue el que propuso la idea y hacía las veces de responsable de negocio. Por tanto, el contacto fue continuo durante todo el desarrollo del proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

Debemos agradecer al Instituto Universitario de Ciencias y Tecnologías Cibernéticas (IUCTC) por proveernos de un entorno y ambiente ideal para el desarrollo de nuestro proyecto. Además, el interés de la temática elegida nos ha ayudado a la hora de mantenernos motivados.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

Tal y como apuntábamos anteriormente, durante la totalidad del desarrollo, ambos miembros hemos trabajado en el mismo laboratorio, con lo que la comunicación se llevó a cabo cara a cara en todo momento.

El software funcionando es la medida principal de progreso. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

Tras cada iteración, nuestro software pasó de ser meramente funcional a ir gradualmente añadiendo más y más características conforme pasaban los días de desarrollo. Una vez metidos de lleno en el desarrollo, estos avances se llevaron a cabo de manera constante.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Para facilitar y agilizar nuestro desarrollo nativo, uno de los primeros pasos fue generar unas librerías específicas para nuestro desarrollo que nos permitieran minimizar el código repetido, de forma que los componentes de la aplicación pudieran reutilizarse mediante la parametrización de los mismos. Este paso hizo que la velocidad de implementación real se redujera considerablemente en las primeras etapas, pero sus resultados fueron recibidos gratamente a lo largo del resto del proyecto.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

Más allá de la descripción de la aplicación y la solución de dudas por parte de nuestro tutor y uno de los desarrolladores de la aplicación anterior, el resto del trabajo ha sido realizado íntegramente por nosotros, teniendo que organizarnos y solucionar los problemas que han ido surgiendo por nuestros propios medios.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Tras las sucesivas sesiones de trabajos, hemos ido perfeccionando nuestra forma de trabajar, marcándonos hitos del desarrollo, planificando nuestros siguientes objetivos a cumplir y poco a poco cumpliendo los requisitos que se nos plantearon al diseñar la aplicación. A lo largo del desarrollo del proyecto se ha ido afinando nuestro sistema de trabajo, consiguiendo finalmente un ritmo altamente productivo.

2.2 Herramientas utilizadas

A la hora de describir las herramientas que hemos utilizado para el desarrollo del proyecto, podemos dividir las en las 3 categorías que describimos a continuación.

2.2.1 Herramientas generales

Entendemos como herramientas generales aquellas que hemos empleado de forma transversal durante todo el desarrollo del proyecto.

2.2.1.1 Suite de ofimática: Microsoft Office 2013

Para el desarrollo de esta memoria y documentación asociada, hemos seleccionado la suite de ofimática de Microsoft Office 2013. Las razones detrás de esta decisión son fundamentalmente las características de la licencia que hemos utilizado, el aspecto profesional de los documentos generados con esta suite y la facilidad de uso puesto que nos resulta una suite conocida.



Ilustración 2-1 - Logotipo de Microsoft Office 2013

Esta suite emplea un tipo de licencia de software propietario conocida como *trialware*, el cual es una versión concreta de la licencia *shareware*. Esta licencia proporciona un tiempo de prueba gratuito del producto completo, tras el cual se presenta al usuario con un formulario a través del cual comprar la aplicación. Para esta licencia hemos optado por participar en la iniciativa de la ULPGC que proporciona uso gratuito de esta suite a los estudiantes de esta universidad.

2.2.1.2 Almacenamiento en la nube: Google Drive y Dropbox



Ilustración 2-2 -
Logotipo de Google
Drive

Para compartir la documentación de nuestros proyectos, hemos empleado Google Drive y Dropbox de forma simultánea. Decidimos utilizar Google Drive para gestionar los bocetos de la documentación generada para el proyecto, aprovechando las capacidades de ofimática colaborativa que la plataforma proporciona.

Una vez confeccionados los bocetos de cada sección, los incluimos poco a poco en un documento final desarrollado mediante Microsoft Word en una carpeta compartida con nuestro tutor en Dropbox, facilitando así la comunicación de documentos.

Tanto en Google Drive como en Dropbox hemos utilizado las cuentas gratuitas que ambas plataformas proporcionan.



Ilustración 2-3 -
Logotipo de
Dropbox

2.2.1.3 Sistema de control de versiones y repositorios: Git y Bitbucket

Para administrar el código de nuestra aplicación consideramos crucial emplear un sistema de control de versiones que nos permitiera mantener nuestro código actualizado de una forma cómoda y sin entorpecer la colaboración. Por ello, hemos decidido incorporar Git a nuestras herramientas para el desarrollo. De esta forma, cada vez que completábamos una etapa significativa del proyecto, generábamos una nueva versión del código, a la que pudiéramos volver más adelante si cometíamos algún error. Decidimos escoger Git frente otras herramientas de control de versiones al ya estar familiarizados con el uso de la herramienta durante nuestros estudios.



Ilustración 2-4 - Logotipo de
Git

Combinado con Git, también decidimos utilizar el servicio de alojamiento de Bitbucket, el cual permite mantener los proyectos accesibles fácilmente entre diferentes máquinas con su control de versiones correspondiente. Bitbucket además proporciona repositorios privados en los que alojar los proyectos de forma gratuita, enfrentado a alternativas como GitHub que se basan en sistemas de repositorios públicos. Por las características del proyecto, consideramos que los repositorios privados de Bitbucket se ajustaban más a nuestras necesidades.



Ilustración 2-5 - Logotipo de Bitbucket

En total hemos empleado 3 repositorios de Bitbucket diferentes: Dos de ellos dedicados cada uno al desarrollo específico de cada plataforma, y un tercer repositorio para el desarrollo de la maqueta.

Git emplea una licencia *GNU General Public License*, una clase de licencia de software gratuito que permite a los usuarios ejecutar, estudiar, compartir y modificar el software con total libertad. Para el uso de Bitbucket hemos empleado las cuentas gratuitas que esta plataforma proporciona.

2.2.1.4 Patrón de diseño: Modelo-Vista-Controlador (MVC)

Modelo-Vista-Controlador es un patrón de diseño de la arquitectura del software orientado al desarrollo de interfaces de usuario. Consiste en separar el software en tres partes interconectadas entre sí, claramente diferenciadas según su función dentro del software. Los tres componentes que forman el Modelo-Vista-Controlador son:

- **Modelo:** Se encarga de almacenar los datos que son solicitados por los controladores y mostrados por las vistas.
- **Vista:** Solicita datos al modelo por medio del controlador con la finalidad de generar una representación de los mismos para ser mostrados al usuario.
- **Controlador:** Realiza las tareas intermedias necesarias para la intercomunicación entre el modelo y la vista.

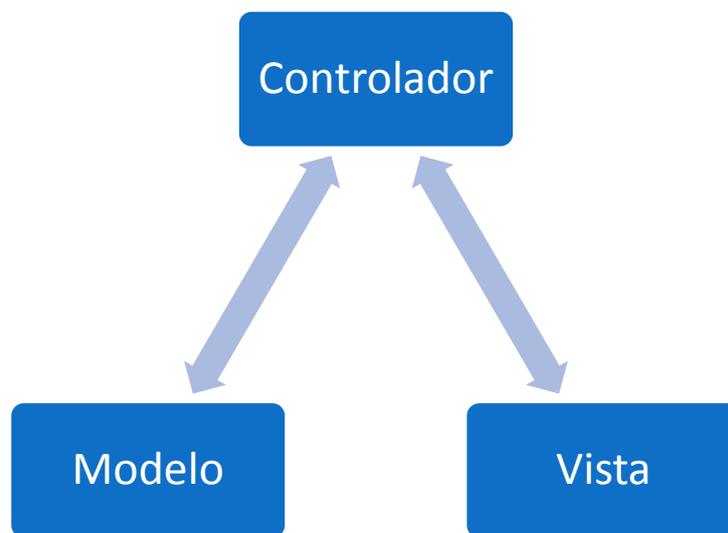


Ilustración 2-6 - Diagrama Modelo-Vista-Controlador

Hemos decidido partir de este patrón de diseño dado que lo hemos estudiado en profundidad durante nuestra formación, y comprendemos que se ajusta de manera natural a lo que queríamos llevar a cabo. Además, es un patrón ampliamente utilizado en este tipo de soluciones software.

2.2.2 Herramientas específicas a la plataforma

A continuación se describen las herramientas específicas al desarrollo en la plataforma Swift, empleadas para la implementación del código nativo en esta plataforma.

2.2.2.1 Lenguaje de programación: Swift

Swift es un lenguaje multiparadigma compilado, fue creado por Apple Inc para el desarrollo de aplicaciones para OSX y iOS. Swift está desarrollado para trabajar en conjunto con el amplio abanico de código escrito en Objective-C y los framework de Apple Cocoa y Cocoa Touch.

Swift se desarrolló con la intención de proveer un lenguaje capaz de generar un código más robusto ante errores y más conciso. Se empezó a incluir en el IDE Xcode 6 y usa el mismo entorno de ejecución que Objective-C pudiendo hacer que ambos lenguajes puedan usarse en un mismo código.



Ilustración 2-7 - Logotipo de Swift

Swift mantiene los conceptos que hicieron a Objective-C flexible y dinámico, Apple ha invertido un esfuerzo considerable en la optimización de las llamadas a métodos y la herencia, intentando solucionar los problemas de eficiencia que estos generaban en Objective-C.

Para este proyecto se optó por Swift en vez de Objective-C debido a que fue considerado más legible y mostraba una sintaxis similar a otros lenguajes ya conocidos como por ejemplo java. Por otro lado, Swift parece ser la apuesta de futuro para el desarrollo en iOS así que consideramos correcto optar por él. Finalmente el hecho de poder utilizar código de Objective-C en la misma aplicación, paliaba el problema de encontrar algo que aún no pudieras conseguir desarrollar con Swift.

2.2.2.2 Entorno de desarrollo: Xcode

Xcode es un entorno de desarrollo integrado (IDE) que contiene un conjunto de herramientas de desarrollo software desarrolladas por Apple para poder desarrollar aplicaciones para iOS y OS X.



Ilustración 2-8 - Logotipo de Xcode

Xcode fue lanzado por primera vez en 2003 y su versión estable más actualizada es la 6.3. Este IDE puede ser descargado de manera gratuita desde el Mac App Store por los usuarios de OS X Yosemite.

Este IDE resulta sencillo de usar, tiene varias herramientas indispensables para el desarrollo en Swift, como métodos de autocompletado y sugerencias, integra contenidos de los manuales de Apple, permite el uso de simuladores de la mayoría de tipos de tabletas y smartphones de Apple. Además provee un sistema de control de versiones propio, y simplifica el desarrollo de código con el uso del storyboard donde podemos construir una interfaz integrando elementos de manera sencilla arrastrándolos con el ratón y luego vincularlos a nuestro código.

2.2.2.3 Gestor de base de datos: Core Data

Core Data es un framework para el manejo de datos persistentes incluido por Apple en los sistemas operativos iOS y OS X. Cabe mencionar que las librerías que se pueden usar en xCode controlan una base de datos no relacional cuyo *Back-End* está en mi caso sustentado por un SQLite, aunque esto podría cambiarse ya que Core Data permite serializar sus datos también en XML o en ficheros binarios.

Para este proyecto usamos Core Data debido a que ya estaba incluido por defecto y dada la naturaleza sencilla de las consultas a base de datos que realizamos en nuestra aplicación no requería de una base de datos relacional.



Ilustración 2-9 - Logotipo de Core Data

2.2.2.4 Frameworks para Swift: Core Data, Webkit, Foundation, UI Kit

El *framework* Core Data, provee las herramientas necesarias para comunicarte con el gestor de bases de datos explicado anteriormente. Entre ellas uso métodos para definir los tipos de elementos que voy a almacenar en mi base de datos local, métodos para extraer dichos datos y realizar filtrados sobre los mismos.

El *framework* Webkit nos permite tener acceso a las herramientas que utilizamos para integrar un navegador web en nuestra aplicación, con el cual podemos renderizar las vistas creadas mediante el desarrollo web.

El *framework* Foundation provee una serie de herramientas de uso general necesarias en la mayoría de aplicaciones para ios. En particular nos suministra una serie de clases que definen la mayoría de objetos utilizados durante nuestro desarrollo, por ejemplo toda la gama de objetos de tipo NS (NSArray, NSString, NSData, NSDate,...).

Cabe mencionar que Foundation nos proporciona las herramientas necesarias para pasar transformar datos entre JSON y Swift y viceversa. JSON (JavaScript Object Notation, Notación de Objetos de JavaScript) es un estándar abierto que permite la comunicación entre aplicaciones a través de la web de una forma ligera. Puesto que el servidor con el que nos comunicamos para alimentar los datos de nuestra aplicación nos proporciona la información en este formato, es necesario efectuar una traducción al recibirla.



Ilustración 2-10
Logotipo de
JSON

El *framework* UI Kit nos suministra una infraestructura necesaria para construir aplicaciones para iOS. En particular nos provee las herramientas necesarias para manejar las vistas y controladores que se usan en la interfaz de usuario, además del manejo de los eventos que sucedan en ellas. Aunque nuestras vistas sean mostradas por el Webkit, este navegador está realmente contenido en una vista nativa y para esta vista y su manejo es necesario el uso de este *framework*.

2.2.2.5 Entorno de pruebas: Emulador y dispositivos reales

Para realizar las pruebas de nuestra aplicación, hemos tenido acceso al sistema de simulaciones suministrado por nuestro IDE Xcode 6.3 de las cuales hemos hablado anteriormente.

Por otro lado se ha dispuesto de una tableta iPad 2 y un *smartphone* iPhone 6, el Instituto Universitario de Ciencias y Tecnologías Cibernéticas (IUCTC) nos ha suministrado tanto el iPad en cuestión como el Mac desde el que se desarrolló la aplicación usando xCode.



Ilustración 2-12 - Foto de iPad 2



Ilustración 2-11 - Foto de iPhone 6

2.2.3 Herramientas para el desarrollo web

2.2.3.1 Lenguaje de marcado: HTML5

HyperText Markup Language (Lenguaje de Marcado de HiperTexto, HTML) es el lenguaje de marcado estándar utilizado en el desarrollo web. Los navegadores web son capaces de leer ficheros HTML y renderizarlos en forma de webs visibles y audibles. HTML es un lenguaje de marcado en vez de un lenguaje de programación debido a que su función es describir una estructura semántica de la web, y un conjunto de reglas para su representación.



Ilustración 2-13 -
Logotipo de
HTML5

En HTML, se definen una serie de etiquetas para marcar los elementos de un documento, facilitando la organización del mismo. Estas etiquetas por ejemplo indican la existencia de un párrafo, un título de una sección, una imagen o incluso otros enlaces a otros documentos, de lo que nace el nombre de "hipertexto".

Las etiquetas HTML poseen diversos atributos como la dirección hacia la que enlaza un vínculo o la localización de una imagen. Hay dos atributos especiales en HTML que permiten diferenciar a elementos particulares del documento del resto. Estos atributos son las clases (*class*) y los identificadores (*id*). Las clases permiten agrupar diferentes elementos que resulten similares a nivel semántico o de presentación, formando subgrupos de etiquetas, mientras que los identificadores proporcionan una denominación inequívoca para un elemento del documento.

HTML genera con estas etiquetas una jerarquía arbórea conocida como *Document Object Model* (Modelo de Objetos del Documento, DOM), y es el modelo estándar que emplean los navegadores para renderizar el contenido de un documento HTML.

Hemos optado por HTML5 debido a que unifica las numerosas versiones anteriores del lenguaje, reduce el número de etiquetas y provee soporte para el uso de archivos multimedia. Se trata del lenguaje de marcado estándar en el desarrollo web actual.

2.2.3.2 Lenguaje de estilo: CSS3

Cascading Style Sheets (Hojas de Estilo en Cascada, CSS) es un lenguaje empleado para describir el aspecto de un documento escrito en un lenguaje de marcado, como puede ser HTML. Este lenguaje se basa en un sistema de selectores y reglas, y es una de las vías principales de explotación de los conceptos de clases e identificadores de HTML.



Ilustración
2-14 -
Logotipo
de CSS3

CSS permite aplicar estilo como color o posicionamiento a cualquier elemento del DOM mediante el uso del nombre de la etiqueta correspondiente. Al mismo tiempo, CSS es capaz de aplicar estilo a componentes concretos del DOM a través de las clases o identificadores que estos posean. CSS define un sistema de prioridades a la hora de aplicar estas propiedades sobre el documento final.

CSS3 es la tercera versión de este lenguaje, aún en parcial desarrollo. CSS3 agrega nuevos selectores y potencia al lenguaje, y a día de hoy su uso se encuentra ampliamente extendido.

2.2.3.3 Lenguaje de programación web: JavaScript



Ilustración 2-15 -
Logotipo de
JavaScript

JavaScript es un lenguaje de programación dinámica comúnmente usado como parte de la programación web. Suele ser usado para permitir relaciones entre el usuario y scripts cliente-servidor. Permite peticiones asíncronas y alterar el contenido del documento que se está mostrando, y agrega un factor importante de interactividad del usuario con una página web. JavaScript se relaciona con HTML a través de la estructura definida en el DOM, permitiendo navegar a través de su estructura e incluso agregar elementos nuevos después de que el documento haya sido cargado.

Su uso a día de hoy va más allá de la programación web, y se ha usado en el desarrollo de juegos y aplicaciones de escritorio o incluso en aplicaciones del lado del servidor usando entornos de ejecución como Node.js.

En nuestro proyecto, hemos utilizado JavaScript para transferir datos entre el entorno nativo y la aplicación web, así como para la modificación de los documentos HTML en tiempo real que caracteriza a este lenguaje.

2.2.3.4 Entorno de desarrollo: Notepad++

Notepad++ es un editor de texto y de código fuente desarrollado para Microsoft Windows. Se distribuye como software libre y en nuestro caso nos permite mostrar los ficheros HTML, JavaScript y CSS respetando su formato y características cumpliendo las veces de IDE para dichos lenguajes. Uno de los factores que agrega mayor potencial al uso de Notepad++ es la fuerte comunidad desarrolladora de complementos que lo rodea, permitiendo ampliar las capacidades del entorno de desarrollo.



Ilustración 2-16 -
Logotipo de
Notepad++

2.2.3.5 Librerías: normalize.css, JQuery y JQuery Mobile

normalize.css

Ilustración 2-17 - Logotipo de
normalize.css

Normalize.css se trata de una librería CSS que permite que los navegadores rendericen todos los elementos de manera más consistente y sujeta a los estándares actuales. Afecta exclusivamente a los estilos de aquellos elementos que necesitan normalización, resultando una alternativa eficiente a las hojas de reset de CSS. El uso de estas herramientas permite un desarrollo uniforme a lo largo de todos los navegadores en los que se mostrará el producto final, facilitando la labor del desarrollador.

JQuery es una librería multiplataforma de JavaScript cuya finalidad es simplificar el desarrollo de los scripts del lado de un cliente HTML. JQuery es software de código abierto licenciado bajo la licencia del MIT.

La sintaxis de JQuery fue desarrollada para facilitar la navegación dentro de un documento, elegir elementos del DOM, crear animaciones, capturar eventos y desarrollar aplicaciones Ajax (*Asynchronous JavaScript And XML*), técnica de desarrollo web que permite agregar comunicación asíncrona con el servidor web en segundo plano después de que el documento haya sido cargado.

JQuery provee herramientas para desarrollar *plug-ins* sobre las librerías de JavaScript, lo que permite crear aplicaciones y páginas web muy potentes.



Ilustración 2-18 -
Logotipo de
jQuery

JQuery Mobile es un *framework* web optimizado para aplicaciones orientadas a dispositivos con pantallas táctiles, creado sobre las librerías de JavaScript. Está siendo desarrollado por el *JQuery Project Team*. Esta herramienta pretende ser compatible con un amplio espectro de tabletas y *smartphones*, y también tiene compatibilidad con *frameworks* y plataformas de aplicaciones como *PhoneGap*, el cual permite el desarrollo de aplicaciones web móviles que pueden ser desplegadas como aplicaciones independientes en los principales sistemas operativos móviles, e incluso proporcionando acceso a los diferentes sensores de estos dispositivos.



Ilustración 2-19 - Logotipo de
jQuery Mobile

2.2.3.6 Entorno de pruebas: Mozilla Firefox

Para llevar a cabo las pruebas durante el desarrollo web, hemos seleccionado como navegador Mozilla Firefox, un navegador gratuito y de código abierto multiplataforma. El criterio para seleccionar este navegador como entorno de pruebas reside en la potencia de las herramientas de desarrollador que incluye.



Ilustración 2-20 -
Logotipo de
Mozilla Firefox

2.3 Análisis

2.3.1 Introducción a la ingeniería de requisitos

Dentro de lo que supone la ingeniería de software, existe una vertiente conocida como la ingeniería de requisitos que comprende todas las tareas relacionadas con la determinación de las necesidades a cubrir por un producto o sistema. Dentro de esta ingeniería se definen una serie de etapas:

Captura e identificación: Obtención de la información necesaria para definir los requisitos del sistema. Puede llevarse a cabo mediante diferentes herramientas, como la observación del entorno de los futuros usuarios, cuestionarios, entrevistas o presentaciones de prototipos, entre otras muchas técnicas.

Análisis y negociación: Detección, corrección y selección de los requisitos identificados. Permite conocer los límites del sistema a desarrollar, priorizar sus requisitos y alcanzar un compromiso final sobre los requisitos a implementar.

Documentación: Etapa que permite asegurar contractualmente el desarrollo del sistema al formar un documento escrito en el que se presentan las características acordadas para el proyecto. A su vez, ayuda a evitar malentendidos y asegura que todas las partes están de acuerdo con los requisitos declarados. Este documento debe describir cada requisito de una forma clara, concisa y tecnológicamente neutra. El resultado final de esta etapa se conoce como un **documento de requisitos**.

Validación: Esta fase tiene por objetivo descubrir problemas en la documentación antes de comenzar con la implementación. Consiste en la revisión completa del documento para detectar omisiones, conflictos, ambigüedades y comprobar la calidad del documento. En este momento se genera lo que se conoce como **línea base**, un conjunto de requisitos formalmente aceptados por todas las personas implicadas en el proyecto. Futuros cambios a este conjunto de requisitos sólo podrán realizarse a través de un proceso formal de gestión y aprobación de cambios.

Gestión: Proceso de administración continua de los requisitos durante el desarrollo del sistema, gestionando los cambios en los mismos y las repercusiones en el sistema que ello supone.

2.3.2 Estudio inicial

Este proyecto surge con la idea de mejorar una aplicación anterior cuyo desarrollo fue llevado a cabo mediante programación web al completo. Por tanto, no tuvimos que realizar un análisis de requisitos completo, sino heredar el del anterior proyecto.

No obstante, se descubrieron algunos requisitos particulares a nuestra implementación, ya que debíamos explorar el desarrollo híbrido con la finalidad de optimizar el manejo de recursos de la aplicación previa.

Cabe mencionar que si bien esta aplicación está basada en una anterior, el desarrollo de nuestro software ha sido completo, sólo se han reutilizado las ideas y los requisitos, pero todas las facetas de la aplicación han sido desarrolladas desde cero.

Durante esta fase, nos reunimos frecuentemente con nuestro tutor y con uno de los autores de la aplicación previa. Durante estas reuniones y mediante el uso de dicha aplicación, detectamos los puntos que debíamos cumplir y los problemas que se querían intentar mejorar de cara a nuestra versión híbrida.

2.3.3 Captura e identificación de requisitos

Para comenzar el desarrollo de una aplicación debemos previamente conocer algunos puntos importantes:

- ¿Cuál es problema?
- ¿Tiene solución?
- ¿Cuál es el contexto?
- ¿Qué necesitamos para solucionarlo?

Para hallar la respuesta a estas preguntas existen diversas técnicas. Durante nuestro análisis, utilizamos habitualmente dos de ellas:

- **La observación:** Para comenzar el desarrollo estudiamos a fondo la aplicación previa, estudiando sus secciones y obteniendo información muy valiosa sobre todos los elementos que nos serían necesarios para poder llevar a cabo nuestro software. Por otro lado, hemos asistido al congreso **Eurocast 2015** como miembros del staff con la intención de conseguir información de primera mano sobre lo que implica la gestión de un congreso.
- **Las reuniones:** Esta técnica, ampliamente utilizada, consistió en nuestro caso en reunirnos con nuestro tutor y/o con un desarrollador de la anterior aplicación. Durante dichas reuniones, se trató de obtener tanto la información relevante para poder desarrollar nuestra versión desde cero como una serie de puntos que serían deseables para nuestra aplicación. Las necesidades que surgieron durante estos eventos fueron discutidas entre todas las partes y documentadas para su posterior contemplación a la hora de desarrollar el proyecto.

2.3.4 Actores del sistema

De cara al uso del sistema, hemos detectado los siguientes actores o entidades que podrán interactuar con nuestra aplicación:

| Actor | Descripción |
|------------------------------|--|
| Usuario no registrado | Usuario común que puede acceder a toda la información del congreso, a excepción de los documentos en PDF de cada conferencia. |
| Usuario registrado | Heredando las características de la aplicación anterior, se introduce un sistema de usuarios registrados de pago que serán los que puedan acceder a los <i>papers</i> de cada conferencia, además de todas las funciones que posee un usuario no registrado. |

Tabla 2-1 - Actores del sistema

2.3.5 Diccionario de conceptos

En el contexto de la aplicación, definimos los siguientes términos:

| Concepto | Descripción |
|----------------------|--|
| Congreso | Evento compuesto de multitud de actividades como charlas, conferencias, visitas a lugares importantes, comidas... Poseen como característica principal un alto flujo de información. |
| Autor | Persona participante en las charlas del congreso, como ponente o como colaborador en la preparación del documento de la conferencia. |
| Temática | Género tratado en una charla del congreso. Una conferencia se centra en una única temática, pero una temática puede verse reflejada en diferentes conferencias. |
| Palabra clave | Etiqueta de relevancia en el ámbito del congreso. Permite concretar los temas a tratar en una conferencia. Una conferencia puede estar relacionada con varias palabras clave, y viceversa. |
| Comité | Conjunto de participantes que se encargan de administrar el correcto funcionamiento del congreso. |

Tabla 2-2 - Diccionario de conceptos

2.3.6 Análisis de requisitos

Este proyecto pretende, partiendo de cero, mejorar el resultado de un desarrollo existente. El único requisito adicional que se agregó a los propios al proyecto anterior fue intentar conseguir una **mejora del rendimiento**, puesto que el desarrollo existente ofrecía tiempos de respuesta demasiado largos en la carga de datos y en las transiciones entre las secciones de la aplicación. Para intentar paliar estos efectos se optó por tomar una serie de medidas que debían cumplirse en nuestro desarrollo:

- **Eliminación del uso de servidores web**

Para optimizar el manejo de recursos y con ello conseguir una aplicación más eficiente hemos decidido eliminar el uso de servidores web, ya que esto conlleva la comunicación entre distintos sistemas informáticos. Esta comunicación añade una capa más de manejo de datos, haciendo que éste sea menos eficiente.

- **Uso de los recursos del sistema nativo**

Para suplir el uso de servidores web, hemos decidido manejar los recursos de manera directa a través de las herramientas nativas que nos ofrece el sistema, accediendo directamente a estos recursos. Gracias a esto, se esperaba conseguir una mejora notable de la optimización de su manejo.

- **Abarcable en tiempo y forma**

En un principio, se planteó el desarrollo completamente nativo de la aplicación. Tras la generación del sistema de carga de datos nativo y conexión con el *Back-End*, se implementaron algunas vistas de la aplicación de forma nativa. Estas vistas ofrecían un rendimiento muy alto, pero a su vez implicaban un fuerte volumen de trabajo para obtener resultados visualmente poco atractivos. Si queríamos introducir elementos más complejos, como los que se incluyen en las vistas de detalle de cada elemento de la aplicación, la dificultad para su implementación se elevaba de forma considerable.

Finalmente y dada la naturaleza de este proyecto, entre otras razones por su límite de tiempo establecido, tuvimos que tomar la decisión de no generar las vistas en nativo y explorar nuevas posibilidades. Fue una decisión difícil, ya que se había dedicado bastante tiempo a generar algunas vistas y se había comprobado que la mejora en rendimiento era muy alta. No obstante, no tuvimos elección ya que de otra manera no habiéramos logrado conseguir completar el proyecto correctamente en el tiempo establecido.

2.3.6.1 Conclusión: Desarrollo híbrido

Para conseguir un rendimiento superior a la aplicación anterior pero terminarlo en el tiempo establecido, optamos por una solución basada en el **desarrollo híbrido**, considerando la facilidad y rapidez del desarrollo de las vistas en la programación web manteniendo un tiempo de respuesta razonable. Esto nos permite, además, acceder a toda la potencia que el desarrollo nativo nos puede proporcionar, resultando comparable incluso a lo que podríamos obtener con un servidor web completo.

Pese a que quizá fue un error no contemplar previamente el coste de desarrollar la aplicación íntegramente en nativo, consideramos digno de mención que, debido a nuestro enfoque ágil de desarrollo y al seguimiento del patrón Modelo-Vista-Controlador (MVC), nuestro sistema de gestión de datos no necesitó modificación y el único tiempo y trabajo desperdiciado fue el del desarrollo de las vistas descartadas.

Debido a este tipo de medidas hemos demostrado ser capaces de reaccionar ante un cambio de requisitos de una magnitud alta como ha sido este caso, pudiendo cambiar el enfoque general de la aplicación sin poner en riesgo el desarrollo exitoso de la misma.

2.4 Diseño

En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, las representaciones de la interfaz y los algoritmos. Por tanto, el diseño debe contemplar todos los requisitos explícitos obtenidos en la fase de análisis, ser una guía que puedan leer y entender los que construyen, prueban y mantienen el software, y proporcionar una idea completa de, en este caso, la aplicación.

2.4.1 Arquitectura del sistema

La arquitectura Cliente-Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes.

Para nuestro sistema, hemos decidido implementar esta arquitectura en dos niveles diferentes: De cara al servidor *Back-End*, el componente nativo actúa como cliente, y de cara al navegador nuestra aplicación haría las veces de servidor.

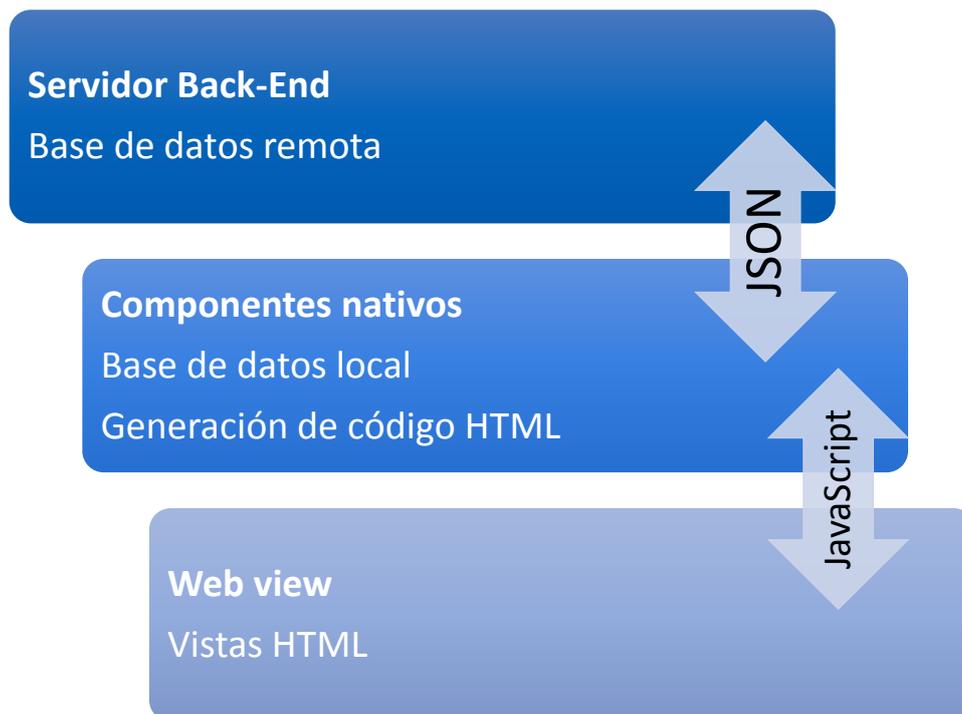


Ilustración 2-21 - Arquitectura Cliente-Servidor a dos niveles

2.4.2 Diseño arquitectónico

Entendemos por diseño arquitectónico el acto de definir una serie de patrones y abstracciones que servirán de guía a la hora de desarrollar nuestro software. Para la realización de nuestro diseño hemos optado por seguir el patrón Modelo-Vista-Controlador (MVC) unido a la arquitectura Cliente-Servidor, ambas explicadas anteriormente.

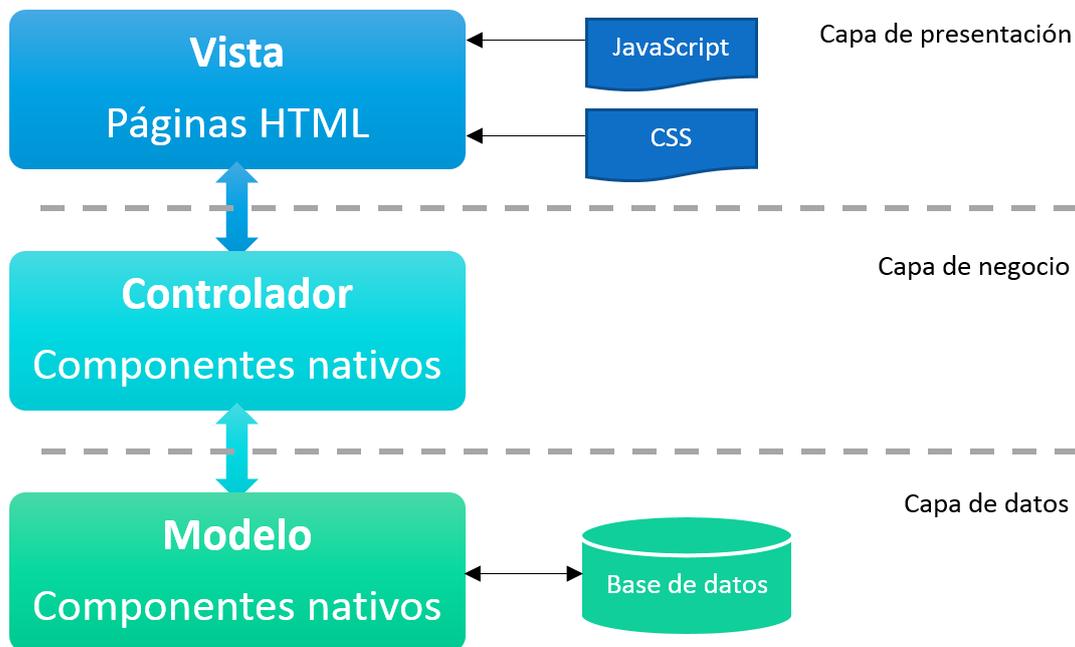


Ilustración 2-22 - Diseño arquitectónico de la aplicación

La **capa de presentación** representa a la interfaz de usuario, por medio de la cual se interactúa con la aplicación. Está desarrollada por medio de programación web empleando código HTML, librerías JavaScript y hojas de estilo (CSS), tal y como se ha explicado anteriormente. Esta capa se encarga de transmitir al usuario la información que necesita, y a su vez capturar la información que se requiere del mismo para poder interactuar con él, transfiriéndola luego a la capa de negocio.

La **capa de negocio** contiene los programas ejecutados. Recibe las peticiones del usuario, y tras ser procesadas, se envían las respuestas al mismo. Esta capa intermedia hace las veces de enlace entre la capa de usuario y la capa de datos, solicitando a esta última que gestione los procesos necesarios para transformar las peticiones del usuario en los datos que éste requiere que se muestren finalmente.

La **capa de datos** contiene toda la información organizada por entidades que representan los datos necesarios para el correcto funcionamiento de la aplicación. Esta capa está formada por el módulo del modelo, el cual contiene todas las clases y métodos externos al sistema que son llamados por la capa de negocio. También incluye un gestor de base de datos encargado de procesos como el almacenamiento persistente y recuperación de la información.

2.4.3 Entidades del modelo

A continuación se mostrará una visión general del modelo utilizado para el desarrollo de nuestra aplicación, procediendo luego a una definición más detallada de cada una de las entidades utilizadas. Cabe mencionar que estas entidades son las usadas en nuestra aplicación y no tienen que ser exactamente iguales a las del servidor *Back-End*.

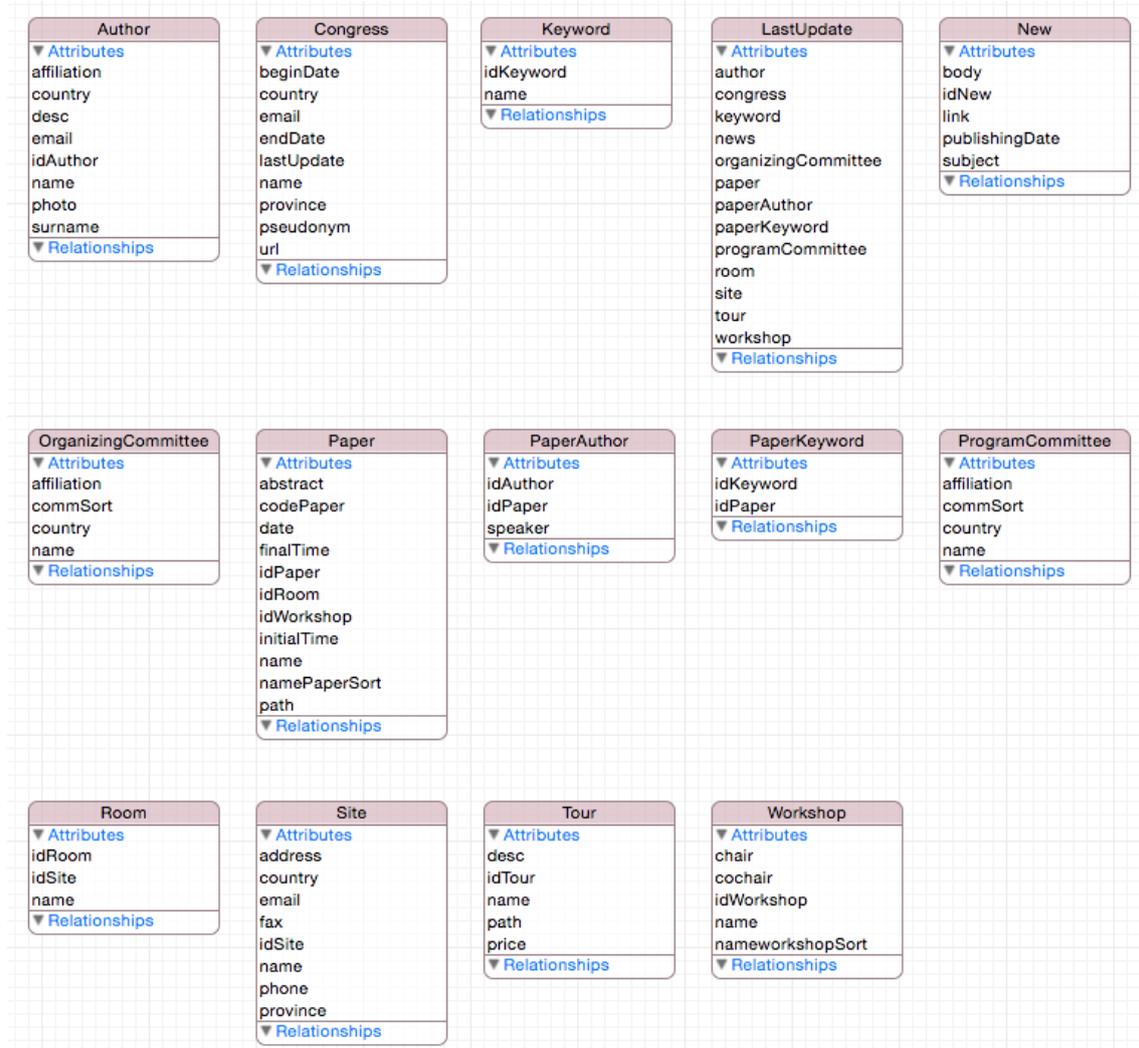


Ilustración 2-23 - Captura de pantalla del entorno Xcode, entidades del modelo

| Entidad | Descripción |
|----------------------------|--|
| Author | Representa y almacena toda la información relevante acerca de los autores que participan en el congreso. |
| Congress | Contiene los datos relativos al congreso. Sólo hay un elemento de este tipo por aplicación. |
| Keyword | Define un conjunto de palabras clave relacionadas a los <i>papers</i> . |
| LastUpdate | Contiene los datos necesarios para el control de versiones de los datos de la aplicación. Nos permitirá saber si existen actualizaciones en las entidades de la aplicación, permitiendo actualizar sólo las que sean necesarias. |
| New | Contiene las noticias que se quieran mostrar durante el congreso. Para ello, cuenta con una sección en la propia aplicación. |
| OrganizingCommittee | Contiene los datos relativos al comité de organización del congreso. |
| Paper | Contiene los datos que se requieren mostrar acerca de los <i>papers</i> de las diferentes charlas que tendrán lugar en el congreso. |
| PaperAuthor | Contiene la relación entre los <i>papers</i> y los autores de los mismos, así como el ponente de la charla. |
| PaperKeyword | Contiene la relación entre las palabras clave y los <i>papers</i> con los que están relacionadas. |
| ProgramCommittee | Contiene los datos relativos al comité encargado del programa del congreso. |
| Room | Contiene los datos de las diversas salas en las que se desarrollarán los eventos del congreso. |
| Site | Recoge los emplazamientos en los que está distribuido el congreso. |
| Tour | Sirve para representar los eventos ajenos a las charlas que pueden ser ofrecidos durante el congreso. |
| Workshop | Recoge los datos relativos a las temáticas a tratar durante el congreso. |

Tabla 2-3 - Entidades del modelo

2.5 Desarrollo

El desarrollo de nuestra aplicación fue dividido en una serie de etapas con finalidades diferentes que finalmente irán integrándose las unas con las otras para conseguir la aplicación final.

La primera etapa fue la de modelado, en la que se crearon las diversas entidades necesarias para representar los datos que sustentan la aplicación y se gestiona su persistencia mediante la creación de una base de datos. Esta etapa forma parte del componente nativo de la aplicación.

En la segunda etapa se realiza el desarrollo necesario para comunicarnos con el servidor *Back-End* y cargar nuestro gestor de base de datos con la información obtenida de dicho servidor. Esta etapa forma también parte del componente nativo de la aplicación, finalizando con el mismo.

En la tercera etapa se construyen una serie de maquetas de las vistas realizadas mediante desarrollo web, en las cuales se utilizan datos de relleno estático con la finalidad de maquetarlas y asegurarnos de que tienen la apariencia que queremos. Durante esta etapa se completa el componente web de la aplicación.

Finalmente en la cuarta y última etapa se enlazan ambos componentes, dotando a las vistas web del contenido dinámico extraído de los componentes nativos, finalizando el desarrollo de la aplicación. En esta etapa interactúan ambos componentes, siendo la sección del desarrollo híbrido.

2.5.1 Modelado

Esta etapa se dividió en tres partes. La primera de ellas consistió en determinar las entidades necesarias para la representación de los datos a utilizar por nuestra aplicación y modelarlas. En la siguiente parte se creó el motor de persistencia y se enlazaron los elementos. Finalmente, en la tercera y última parte se creó una interfaz de uso para solicitar datos al motor de persistencia y crear vectores con dichos datos, ya que era más rápido su uso en tiempo de ejecución por parte de la aplicación.

2.5.1.1 Creación de clases

Estas entidades representan la información que necesitamos durante nuestra aplicación, cada una de ellas representa a un elemento diferente con sus propias características y contenidos.

Para determinar la selección de estas entidades y sus características se realizaron varias reuniones con nuestro tutor y con un desarrollador de la anterior aplicación. También se realizó con el mismo fin un estudio de la anterior aplicación, así como de los datos almacenados en el servidor *Back-End*.

Para implementar estos elementos hemos recurrido a la estructura de clases de Swift ya que es una herramienta propia de la orientación a objetos que provee nuestro dicho lenguaje.

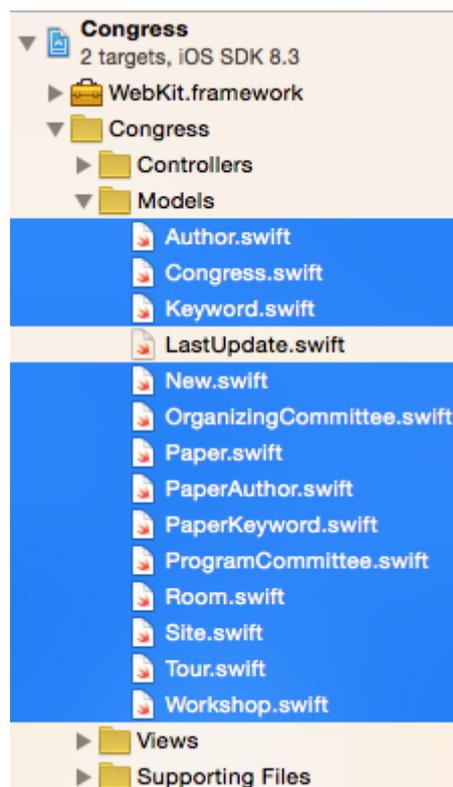


Ilustración 2-24 - Clases del Modelo

Todas estas clases cumplen una función similar, ser utilizadas por nuestra aplicación para almacenar la información y luego ser mostradas donde corresponda por medio de las vistas.

Existe una excepción dentro de estas entidades, este es el caso de *LastUpdate*, su cometido será diferente ya que no será una entidad que será mostrada pero sí necesitamos almacenar su información. Entraremos más en detalle en la sección que la utiliza.

Finalmente el conjunto de estas clases conforman el Grupo de `Models`, agrupando a todas las clases que forman parte del modelo para mayor claridad del código y del proyecto, a la vez que se respeta la configuración del patrón Modelo-Vista-Controlador.

Cabe mencionar que a diferencia de otros gestores de base de datos, al usar la librería CoreData de Swift podemos definir que estas clases sean directamente elementos manejados por un contexto persistente. De esta manera podríamos manejar dichas clases como objetos a nivel del lenguaje, pero en realidad la librería se encarga de almacenarlas y tratarlas como elementos de la base de datos bajo petición, abstrayendo al desarrollador de este elemento.

Para poder conseguir este tipo de objetos debemos implementar la interfaz `NSManagedObject` durante la creación de la clase, lo cual nos fuerza a crear un método `createInManagedObjectContext` el cual debe tener acceso a un Contexto que será el motor de persistencia. Finalmente debemos incluir el tipo `@NSManaged` en los atributos de la clase que serán controlados por el motor de persistencia.

```
class Author: NSManagedObject {
    @NSManaged var idAuthor: NSNumber
    @NSManaged var name: String
    @NSManaged var surname: String
    @NSManaged var country: String
    @NSManaged var email: String
    @NSManaged var affiliation: String
    @NSManaged var photo: String
    @NSManaged var desc: String
}
```

Ilustración 2-25 - Implementación de objeto del contexto

Estas clases deben estar representadas en el esquema `xcdatamodeld` y luego enlazadas con el código con lo que se establece la relación entre los objetos que referencian a la clase y los modelos de manejo del motor de base de datos.

The screenshot shows the XCDatamodeld editor with the 'Author' entity selected. The left pane lists various entities, and the right pane shows the attributes for the selected entity.

| Attribute ^ | Type |
|---------------|------------|
| S affiliation | String |
| S country | String |
| S desc | String |
| S email | String |
| N idAuthor | Integer 64 |
| S name | String |
| S photo | String |
| S surname | String |

Below the attributes table, there are sections for 'Relationships' and 'Fetched Properties', both currently collapsed.

Ilustración 2-26 - Modelado por medio de `xcdatamodeld`

A continuación detallaremos el funcionamiento de nuestro motor de persistencia.

2.5.1.2 Motor de persistencia

Debido a la naturaleza de nuestra aplicación, queremos que, tras la descarga de los datos desde nuestro servidor *Back-End*, podamos utilizar nuestra aplicación de manera local, sin requerir conexión a internet. Para ello nos vemos forzados a utilizar alguna herramienta que nos permita mantener los datos de manera persistente aunque se cierre la aplicación.

En el caso de iOS hemos decidido usar la librería CoreData, la cual nos suministra una capa de abstracción. Permitiendo usar un gestor de base de datos, en este caso SQLite, por medio de una interfaz amigable manejando datos con la estructura de clases de Swift y dejando el almacenamiento persistente de los mismos a cargo de esta librería.

Para acceder a esta funcionalidad debemos definir un contexto de almacenamiento de objetos, para lo cual debemos utilizar la clase `NSManagedObjectContext` que provee la librería CoreData, en general este elemento hace las veces de enlace entre nuestro código Swift y el gestor de base de datos SQLite por medio de las librerías.

Hemos trasladado la creación de este elemento a la clase `AppDelegate` cuya finalidad es suministrar métodos de apoyo a la aplicación. Entre otras funcionalidades, en esta clase se administran eventos y se proporcionan herramientas a nuestra App. En este caso hemos provisto a esta clase de la generación del contexto y sus características, además de definir un método para que transfiera los elementos de Swift a SQLite denominado `saveContext`.

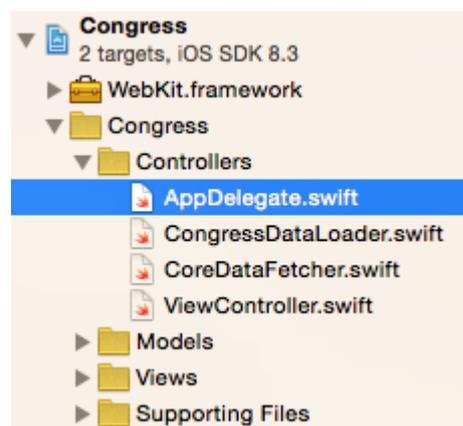


Ilustración 2-27 - Controlador delegado

Tal y como mencionamos previamente, este objeto que representa el contexto es utilizado por cada una de las clases del modelo con la finalidad de que sean todos almacenados en la misma base de datos, ya que este elemento es el enlace a la misma. Para ello se utiliza el método `createInManagedObjectContext` mencionado anteriormente.

Para terminar con esta etapa hablaremos ahora de como accede la aplicación a estos datos.

2.5.1.3 Interfaz de consulta al motor de persistencia

Tras disponer de la base de datos cargada con los elementos obtenidos del servidor *Back-End*, al ejecutar la aplicación se cargan una serie de vectores a partir de los datos almacenados por nuestro motor de persistencia local. Para ello generamos una interfaz denominada *CoreDataFetcher* a la cual puedes acceder con los métodos *fetch* y *basicfetch* a los cuales puedes pasarle una serie de predicados y/o sistemas de ordenación y como resultado obtienes un vector de elementos de la entidad escogida ordenados y filtrados como se desee.

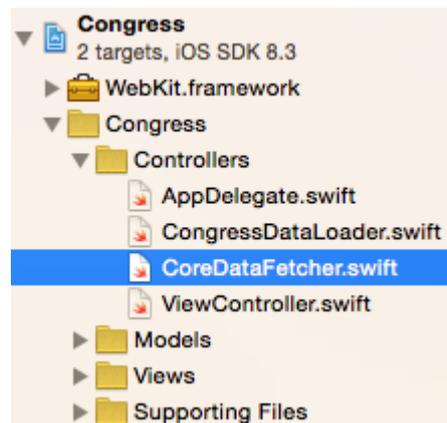


Ilustración 2-28 - Interfaz de consulta al motor de persistencia

Hemos optado por estos vectores ya que reducen los tiempos de consulta al estar precalculados y sólo se busca en ellos durante la ejecución.

2.5.2 Conexión con el servidor *Back-End*

En esta etapa hemos dividido el desarrollo en tres partes. La primera es el método usado en la transferencia de datos entre el servidor *Back-End* y nuestra aplicación. La segunda parte se encarga de convertir los datos JSON obtenidos en objetos Swift e insertar estos elementos en nuestro motor de persistencia. Finalmente la tercera parte trata acerca de la política seguida a la hora de sincronizar los datos desde el servidor *Back-End* cuando existen actualizaciones.

2.5.2.1 Obtención de datos del servidor *Back-End*

En el caso de nuestra aplicación, el acceso al servidor remoto se lleva a cabo mediante peticiones HTTP GET contra una URL, en la que se especifica el nombre del congreso y la entidad a solicitar. El servidor responde a estas peticiones con un documento en formato JSON con los datos correspondientes. Partiendo de esto, hemos dividido en fragmentos la URL de acceso para facilitar la parametrización, y hemos extraído los fragmentos correspondientes a cada entidad.

Para llevar esto a cabo hemos usado el método *NSData*, el cual realiza la petición a una URL pasada por parámetro, tras esto se obtienen los datos en formato JSON desde el servidor *Back-End*.

2.5.2.2 Deserialización JSON e inserción en el motor de persistencia

Una vez tenemos los datos del servidor, estos están en formato JSON. Para poder tratarlos en Swift debemos transferirlos a objetos de nuestro lenguaje, para ello nos valemos del método `NSJSONSerialization.JSONObjectWithData` el cual genera un objeto con el formato e información que contienen los datos en formato JSON devueltos por nuestro servidor *Back-End*.

Finalmente cada una de nuestras clases de Modelo tiene un método denominado `fromJSON`, el cual se encarga de recorrer todos los elementos del objeto que acabamos de generar, asignándolo a los atributos de un objeto de nuestra clase de modelo y finalmente insertándolos uno a uno en nuestro motor de persistencia.

2.5.2.3 Sincronización de los datos

Nuestra aplicación dispone de una interfaz para el manejo de las inserciones en nuestro motor de persistencia, esta ha sido denominada `CoreDataLoader`.

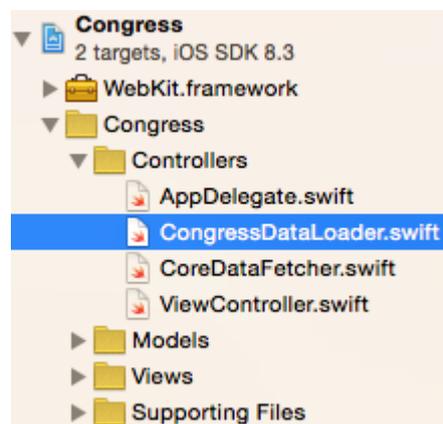


Ilustración 2-29 - Interfaz de inserción en el motor de persistencia

Esta interfaz suministra dos métodos, el método `load` y el método `update`. El método `load` es ejecutado únicamente la primera vez que se lanza la aplicación, cargando todos los datos desde el servidor *Back-End* y dejando todo preparado para el correcto funcionamiento de la aplicación en su primer lanzamiento.

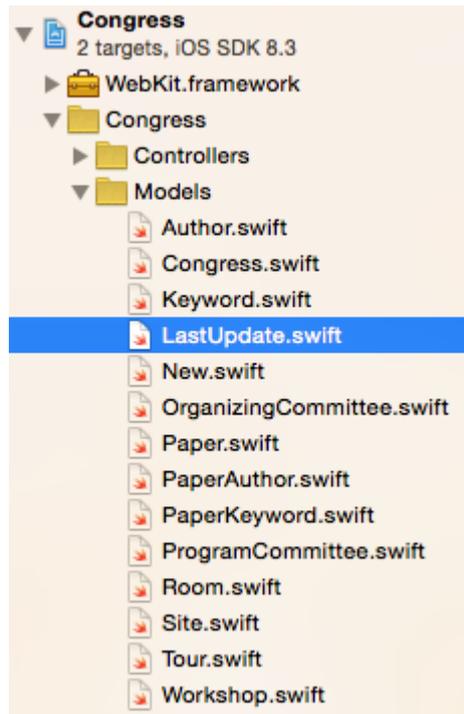


Ilustración 2-30 - Controlador de sincronía

Este método carga todos los datos y utiliza el modelo de `LastUpdate` para cargar en él las fechas de la última modificación de cada uno de las entidades del modelo. Gracias a esto podemos detectar en cualquier momento si nuestra aplicación y el servidor están o no sincronizados.

Finalmente el método `update` podrá ser lanzado desde una de las vistas web de la aplicación. Esta vista, al ser accedida por el usuario, mostrará si la aplicación está o no en sincronía con el servidor. Para ello se vale de un método que se encarga de comparar las fechas de la última modificación de cada entidad en el servidor *Back-End* con la fecha de la versión de dicha entidad en nuestra aplicación. Si existe alguna diferencia se permitirá pulsar un botón que se encargará de transferir de nuevo las entidades que no estén actualizadas.

2.5.3 Maquetado de las vistas

Durante esta etapa, se desarrollaron una serie de vistas mediante desarrollo web. Tal y como se menciona en la sección de herramientas, nos hemos valido de una serie de lenguajes y *frameworks* para poder implementar dichas vistas.

2.5.3.1 Preparación del entorno

Además del conjunto básico HTML5+CSS3+JS, para el proceso de desarrollo de las vistas web hemos empleado los *frameworks* JQuery y JQuery Mobile para agilizar esta etapa y obtener resultados vistosos sin demasiada dificultad. Hemos compuesto, mediante estas herramientas, una versión básica de lo que supone la interfaz de la aplicación completa.

Como resultado final de esta etapa, hemos obtenido un conjunto de documentos HTML totalmente navegables entre sí que dan una idea clara del funcionamiento esperado de la aplicación, y permiten posteriormente la reutilización de los elementos de esta maqueta para formar la interfaz real. Esta versión no implementa funcionalidades complejas como puede ser el inicio de sesión de usuarios o el sistema de favoritos, al tratarse de una maqueta.

El primer paso que hemos tomado para desarrollar en JQuery Mobile ha sido recurrir a la herramienta online ThemeRoller que proporcionan en la página oficial, la cual permite generar un tema personalizado para este *framework* fácilmente.

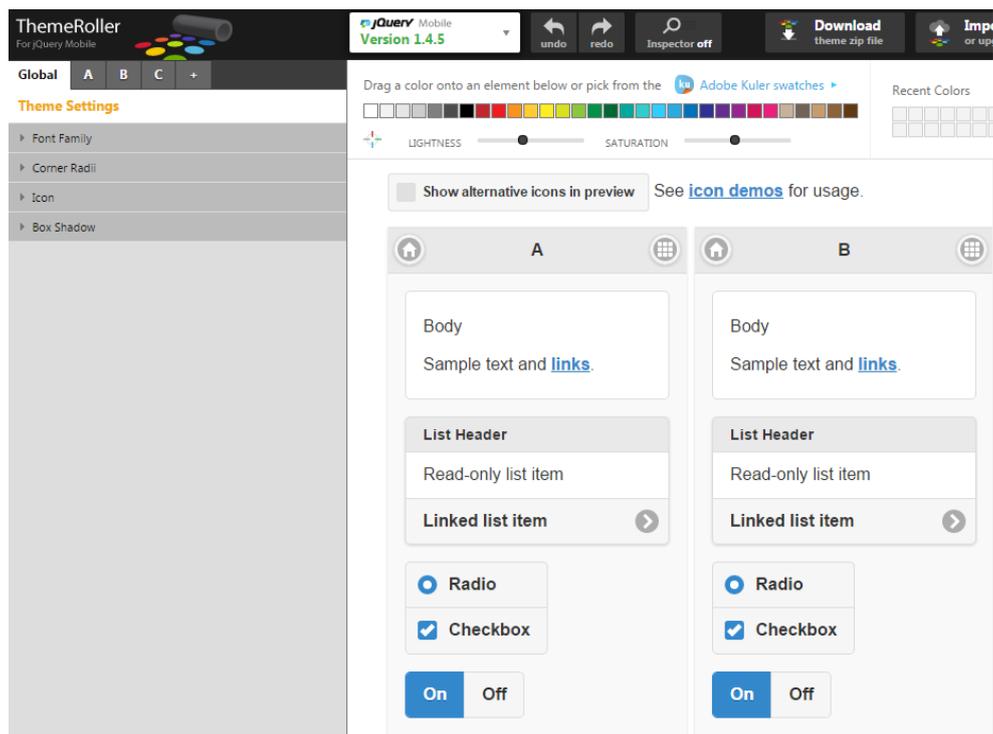


Ilustración 2-31 - Captura de pantalla de ThemeRoller

Una vez definido el tema, hemos generado un esqueleto básico para la aplicación que será reutilizado en el desarrollo de cada una de las páginas. A cada una de las vistas le corresponde una página HTML, y todas comparten el mismo estilo de encabezado y pie de página.

En el encabezado vemos el nombre del congreso para el cual se implemente la aplicación, si estamos en la página principal, o bien el nombre de la sección en la que estamos, para el resto de páginas.



Ilustración 2-32 - Captura de pantalla del encabezado de la aplicación

En el pie de página hemos incluido una barra de herramientas con cuatro botones que nos permiten un acceso rápido a las secciones más importantes de la aplicación.

- El primero de ellos es el botón *"Home"*, el cual nos traslada a la página principal.
- El segundo botón denominado *"Program"* nos lleva a la vista del programa del congreso.
- El tercer botón es el de *"My Program"*, donde encontraremos los elementos del programa que marquemos como favoritos. Esto nos permitirá configurar nuestro propio subprograma a fin de crearnos una guía de las charlas a las que queremos asistir.
- Para terminar, el cuarto botón es el botón *"Updates"* que nos llevará a la vista que se encarga de mostrarnos si hay actualizaciones de los datos disponibles y nos permite realizar dicha actualización.

Esta serie de elementos son comunes a todas las vistas de la aplicación y permiten poder retornar al índice principal y a las secciones que consideramos más importantes de una manera rápida y accesible desde cualquier punto de la aplicación.



Ilustración 2-33 - Captura de pantalla de la barra de herramientas de la aplicación

2.5.3.2 Vistas de la aplicación

A continuación pasamos a describir con detalle cada una de las vistas de la aplicación y sus características principales. Las vistas de nuestra aplicación se distribuyen generalmente en dos tipos: Vistas de índices y vistas de detalle.

Incluiremos en esta sección capturas de las secciones más relevantes. Pueden encontrarse más capturas de la maqueta en el [Anexo II](#).

- *Index* (Página principal): Es la primera pantalla que vemos de la aplicación nada más abrirla. Desde ella podemos acceder a las demás vistas de índices, haciendo las veces de nexo central de nuestra aplicación. Podemos acceder a ella en cualquier momento desde el botón “Home” de la barra de herramientas situada en el pie de página de todas las vistas de nuestra aplicación.

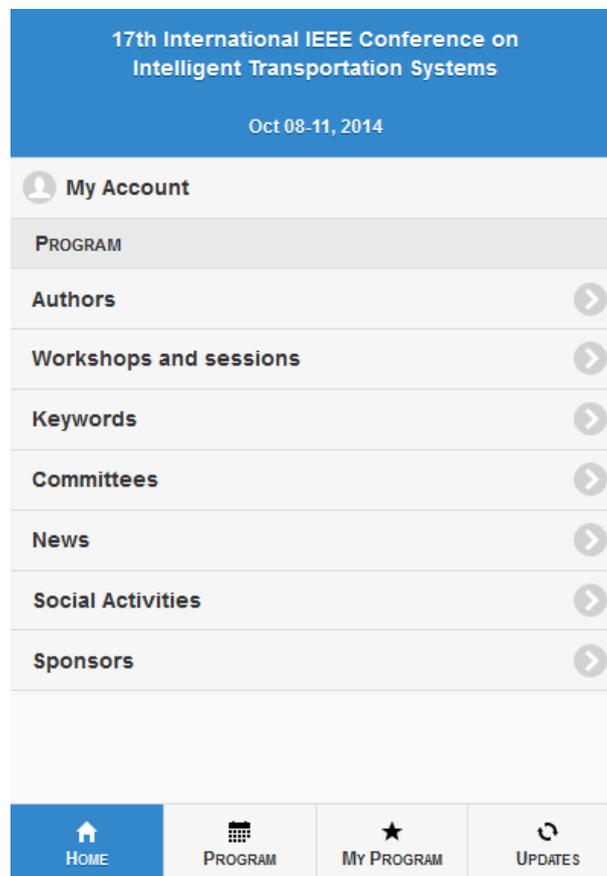


Ilustración 2-34 - Captura de pantalla de la página principal

- *My Account* (Mi cuenta): Mediante esta vista se le permite al usuario autenticarse contra nuestro servidor de *Back-End*. Cuando consigue identificarse de manera exitosa se le permitirá al usuario descargarse los *papers* desde la sección pertinente, ya que esto es un derecho al que sólo tienen acceso los usuarios registrados.

- *Full program* (Programa): Desde esta vista se podrá obtener la información relativa al programa del congreso. En la zona superior existe un selector de los días en los que se desarrolla el congreso, y en la zona inferior se muestran las conferencias del día seleccionado. Para cada conferencia se presenta una breve descripción y datos de interés de las mismas, como la sala en la que tendrán lugar y su horario. Desde esta vista podemos acceder a las vistas de detalle de cada sesión.

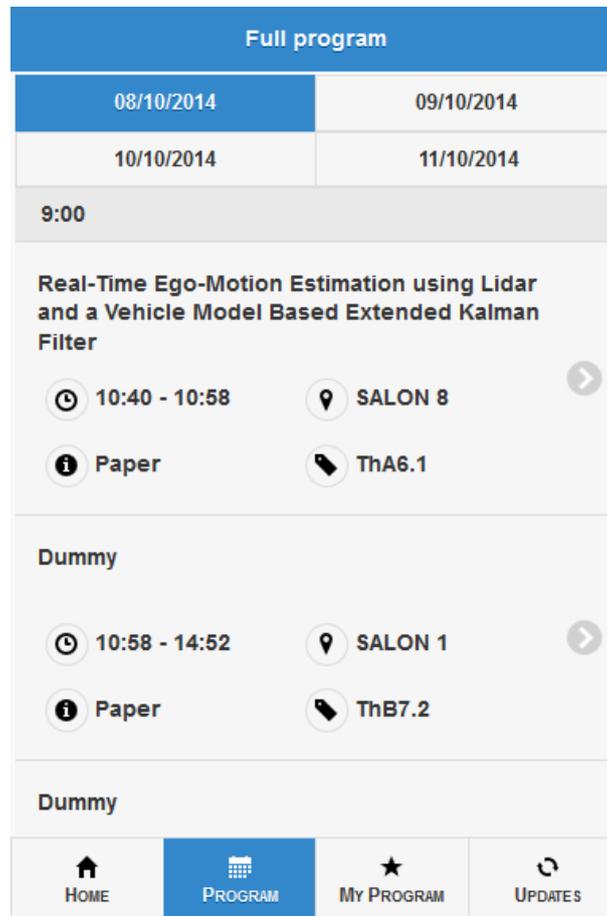


Ilustración 2-35 - Captura de pantalla del programa del congreso

- *My program* (Mi programa): Se trata de una vista idéntica a la del programa completo, con la diferencia de que en esta sólo se mostrarán las sesiones que seleccione el usuario de la aplicación.
- *Updates* (Actualizaciones): En esta vista se mostrará al usuario el estado de sincronía de la aplicación con el servidor *Back-End*, indicándole si su aplicación está actualizada o no. Si fuera necesario, se le facilitará al usuario un botón con el que podrá comenzar la actualización de la aplicación.

- *Authors* (Listado de autores): En esta vista podemos encontrar un listado de los autores que participan en el congreso y algunos datos de los mismos, como por ejemplo su nombre y su nacionalidad. También se dispone de un buscador para filtrar el índice. Desde esta vista podemos acceder a la vista de detalle de los autores.
- *Workshops* (Listado de temáticas): En esta vista podemos ver un listado de las temáticas a tratar durante el congreso, incluyendo su título y el tipo de sesiones que la compone. También se dispone de un buscador para filtrar el índice. Desde esta vista podemos acceder a la vista de detalle de las temáticas.

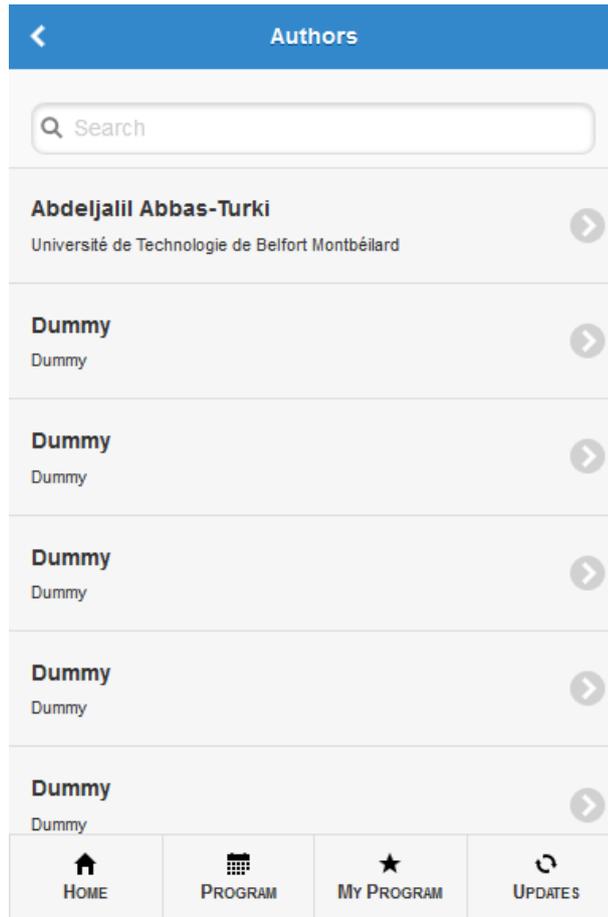


Ilustración 2-36 - Captura de pantalla del listado de autores

- *Keywords* (Listado de palabras clave): En esta vista podemos ver un listado de las palabras relacionadas a las sesiones que se llevarán a cabo durante el congreso. También se dispone de un buscador para filtrar el índice. Desde esta vista se puede acceder a la vista de detalle de las palabras clave.
- *Committees* (Listado de comités): Desde esta vista podemos acceder a un listado de los miembros de ambos comités, el organizador y el del programa. En la parte superior se puede elegir qué tipo de comité queremos ver y en la parte inferior se mostrarán los miembros de dicho comité, pudiendo verse alguna información como su nombre y su afiliación.
- *News* (Listado de noticias): Desde esta vista podemos acceder a un listado de las noticias que se decidan publicar durante el congreso. Se presenta para cada noticia su título y la fecha y hora de su publicación. Desde esta vista se podrá acceder a la vista de detalle de las noticias.

- *Social Activities* (Listado de eventos del congreso): En esta vista podemos ver un listado de los eventos ajenos a las charlas que se celebren durante el congreso. Para cada evento, se presenta su título en el listado. Desde esta vista se podrá acceder a la vista de detalle de los eventos.
- *Sponsors* (Listado de los patrocinadores): Desde esta vista podremos acceder a un listado de los patrocinadores del congreso, incluyendo su nombre y su logotipo.
- *Session* (Detalle de las sesiones): En esta vista podemos ver gran parte de la información de las sesiones, incluyendo su horario, la sala en la que se llevará a cabo, un enlace al *paper* de la sesión si el usuario está registrado. Finalmente se mostrará un listado de los autores de la sesión y una descripción de la misma.
- *Author* (Detalle de un autor): En esta vista veremos información detallada de un autor, incluyendo su nombre completo, nacionalidad, afiliación y una foto. En la sección inferior podemos ver un listado de las sesiones en las que participa y una descripción del autor.

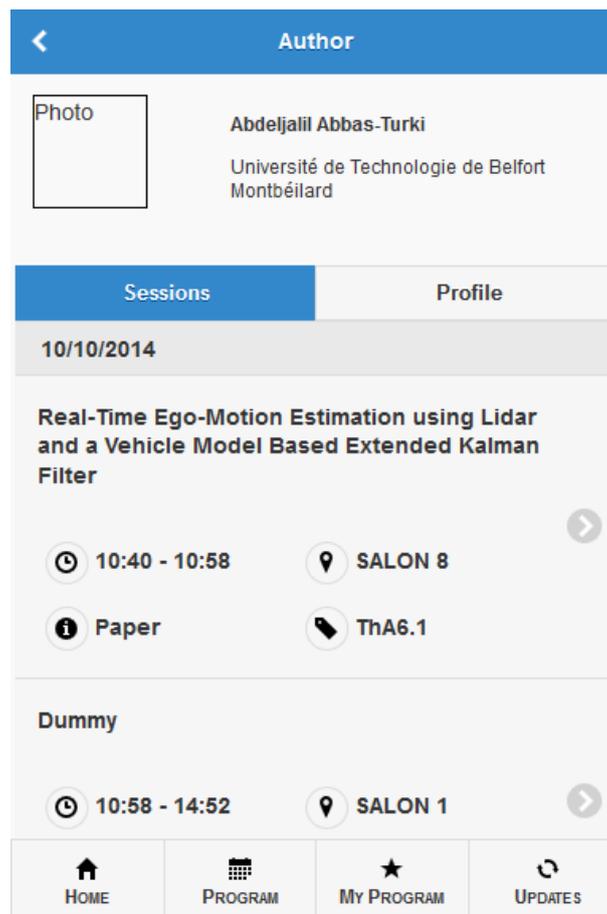


Ilustración 2-37 - Captura de vista de detalle de un autor

- *Workshop* (Detalle de una temática): En esta vista podemos ver un listado de todas las sesiones en las que se trata una misma temática, además de una descripción de dicha temática. Esta descripción la cual incluye el tipo de sesiones que la compone y los miembros del comité del programa que las moderan.
- *Keyword* (Detalle de una palabra clave): En esta vista podremos obtener un listado de las sesiones en las que se hace referencia a una palabra clave en particular.
- *New* (Detalle de una noticia): Por medio de esta vista se muestra el contenido de las noticias publicadas por los organizadores del evento, incluyendo su título, contenido y fecha de publicación así como enlaces externos si fueran necesarios.
- *Social Activity*: (Detalle de un evento): Desde esta vista podemos acceder a los detalles de los eventos ajenos a las charlas que se celebren durante el congreso, incluyendo el plan del evento, su precio, descripción y enlaces a elementos externos si fueran necesarios.

2.5.4 Integración híbrida

En esta etapa final, en la que ya disponemos de los datos cargados en código nativo y las vistas web realizadas, sólo nos falta unir ambos componentes. Para ello hemos dividido esta etapa en cuatro partes. La primera parte consistirá en cargar todos los elementos web estáticos en el componente nativo de navegador *webView* de nuestra aplicación. Tras esto, en la segunda parte insertaremos el contenido dinámico en las vistas de índices a partir de los datos almacenados en nuestra aplicación. La tercera parte consistirá en la integración del contenido dinámico en las vistas de detalle, para lo cual necesitaremos establecer unas relaciones que indiquen el elemento del que queremos mostrar el detalle (este caso no se da en las vistas de índices). Finalmente la última parte tratará acerca de los casos especiales que nos hemos encontrado en algunas vistas y como los hemos solucionado.

2.5.4.1 Inicialización del componente *WebView*

Para poder cargar nuestra maqueta en nuestro componente *WebView*, debemos agregar los recursos web a nuestro proyecto. Para ello generamos el grupo *Views*, en el cual se encuentran todos los ficheros necesarios para las vistas de nuestra aplicación.

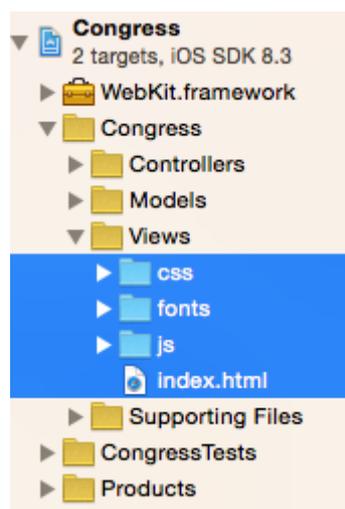


Ilustración 2-38 - Grupo *Views*

Una vez disponemos de todos estos elementos podremos configurar el *WebView*. Entre otros parámetros, le hemos configurado el espacio de pantalla que ocupará. De esta manera se respeta la sección superior donde se muestra la barra de estado de iOS, ya que por defecto este componente ocupa toda la pantalla.

Para finalizar la configuración del componente *WebView*, nos queda por realizar un último ajuste. Debemos establecer el método de comunicación entre el código nativo y la maqueta Web, para ello vamos a utilizar JavaScript.

En Swift debemos configurar una serie de manejadores de eventos, para ello debemos añadirlos a la configuración del componente *WebView*. Estos manejadores esperan que ocurra un evento en JavaScript y lo capturarán, pudiendo durante esa captura realizar todo el cálculo nativo necesario. Finalmente, tras estos cálculos, los manejadores podrán inyectar código JavaScript a las maquetas Web cargadas en el componente *WebView*. Rellenando éstas con el contenido dinámico disponible en nuestro componente nativo.

El tipo de evento que capturamos normalmente desde estos manejadores fueron los cambios de página, es decir, cuando navegamos entre las distintas vistas Web de la aplicación. Para ello hicimos uso del evento *pagechange*. Al ocurrir este evento se detectaba la vista Web que se había cargado mediante una consulta a los elementos HTML visibles. De esta forma, si el listado de autores está visible al ocurrir el evento *pagechange*, se realizará una llamada que atenderá el manejador que rellena dicho listado.

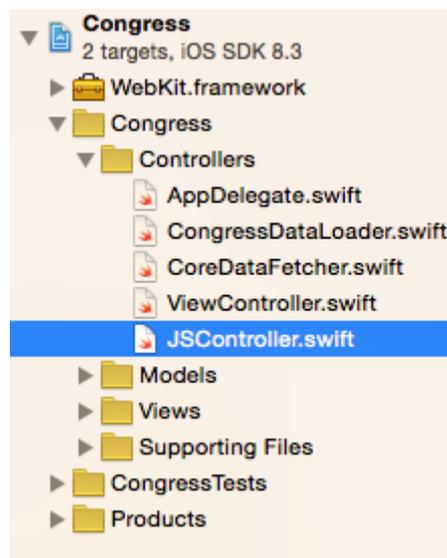


Ilustración 2-39 – *JSController*

Estos manejadores se encuentran en el controlador denominado *JSController*.

2.5.4.2 Generación de índices dinámicos

Decidimos desarrollar primero este tipo de contenido debido a su menor complejidad, ya que no usaban relaciones entre otros modelos. Mostrando los listados en su totalidad, sin necesidad de elementos externos que los filtraran.

Dado que no queríamos realizar consultas al motor de base de datos en tiempo de ejecución, decidimos cargar el contenido de las tablas en vectores en nuestro componente nativo. Estos vectores están ya ordenados y listos para ser mostrados en las vistas Web.

Una vez se activa un manejador de eventos en el componente nativo, este recorrerá el vector asociado al tipo de elementos a mostrar. Almacenando en una ristra el código JavaScript necesario para insertar todo el contenido dinámico en la vista Web. Finalmente esta ristra con código JavaScript será inyectada en el documento cargado por nuestro componente *WebView*.

Cabe mencionar que algunos de los listados requieren divisores entre los elementos, agrupándolos por algún rasgo particular como puede ser la primera letra del nombre o la hora de comienzo de la sesión. JQuery Mobile incluye herramientas que facilitan la generación de estos divisores, tomando por defecto el texto que se muestra en cada elemento como criterio para definir los divisores. No obstante, para divisores basados en datos que no se muestran en la interfaz, debemos configurar la generación de estas listas. Para cumplir este objetivo, inyectamos en cada elemento de la lista un elemento oculto con el dato necesario para aplicar la agrupación. En esos casos, se le indica a JQuery Mobile que utilice ese dato para generar los divisores correspondientes, en lugar de los valores por defecto.

Finalmente, en aquellos listados cuyos elementos tengan una vista de detalle, necesitan una forma de enlazar ambos elementos. Gracias a esto podremos navegar desde un elemento del listado hasta su vista de detalle correspondiente. Para esto hemos decidido utilizar una pila de identificadores que irá añadiendo y extrayendo elementos según naveguemos por la aplicación. Cada uno de los elementos del listado, dispone de un evento *onClick()*, que llamará a la función JavaScript *SetId(id)*, este id se generará dinámicamente junto con el listado y cada elemento establecerá su posición del vector como *id*. Esta función cambiará el valor de la variable *id* de JavaScript por el *id* proveído por el elemento pulsado para su posterior recuperación en la vista de detalle. Finalmente cada vez que se realice una navegación “hacia adelante” se insertará en la pila el valor de la variable *id* y cuando la navegación sea “hacia atrás” se extraerá el último elemento de la pila.

2.5.4.3 Generación de vistas de detalle dinámicas

Siguiendo lo explicado anteriormente, las vistas de detalle necesitan saber que elemento deben mostrar, pudiendo así acceder a dicho elemento en el vector correspondiente. Para ello desde JavaScript se le pasa como mensaje al manejador de eventos pertinente el último valor almacenado en la pila de identificadores.

Una vez disponemos de este identificador ya sólo nos queda acceder a esa posición del vector pertinente y extraer todos los datos necesarios para generar el contenido que se quiere mostrar.

Finalmente debemos mencionar que algunas vistas de detalle requieren el uso de datos de otros elementos, por ejemplo la vista de detalle de autor muestra un listado de las sesiones en las que dicho autor participa. Para esto disponemos de un vector con las relaciones entre los elementos en nuestro componente nativo. Debido a que queremos minimizar al máximo las consultas al motor de base de datos durante la ejecución del programa, nos hemos limitado a recorrer secuencialmente los vectores, comparando sus identificadores y quedándonos sólo con aquellos válidos. Este método puede resultar subóptimo, pero debido al pequeño tamaño de los vectores a recorrer no supone ninguna desventaja apreciable en el rendimiento.

2.5.4.4 Casos especiales

Durante el desarrollo, hemos encontrado algunas vistas de la aplicación que han requerido un tratamiento diferente en su generación debido a las características de la información que muestran. A continuación, exponemos estas situaciones y las medidas que hemos tomado al respecto.

2.5.4.4.1 *Índice de autores*

En un congreso pueden participar una gran cantidad de autores. Debido a esto, el número de elementos del listado de autores puede ser demasiado grande para mostrarlo de una vez. Con el fin de reducir el tiempo de carga de este listado se ha decidido mostrar los autores en grupos de veinte, mostrando nuevos grupos según nos vamos desplazando hacia el final del listado.

Para llevar esto a cabo, se cargan todos los autores en la página, estableciendo para todos los elementos excepto los veinte primeros el atributo `display: none` de CSS el cual evita que sean renderizados por nuestro navegador. Cuando nos desplazamos por el listado, iremos eliminando dicho atributo del siguiente grupo de veinte, con lo que conseguimos que se renderice por grupos, evitando molestos tiempos de espera.

2.5.4.4.2 Programa

Durante el congreso se llevan a cabo múltiples conferencias. La sección del programa contiene toda la información necesaria acerca de las mismas. Debido al gran número de conferencias que pueden existir en el congreso, hemos decidido mostrarlas separadas por días, contando cada día con su propia pestaña.

En cada una de estas pestañas se pueden ver las charlas que sucederán durante el día seleccionado, ordenadas según su hora de comienzo y existiendo una línea de división entre las horas que divide las charlas según la hora en la que tendrán lugar. Con el fin de proveer al usuario de una manera rápida de encontrar las charlas que suceden en un momento dado, hemos decidido mostrar en el programa sólo estas líneas con las horas, pudiendo interactuar con ellas para que se muestre el contenido del programa para dichas horas. Con esto además logramos reducir el tiempo de carga de esta sección, la cual es una de las vistas con mayor contenido de la aplicación.

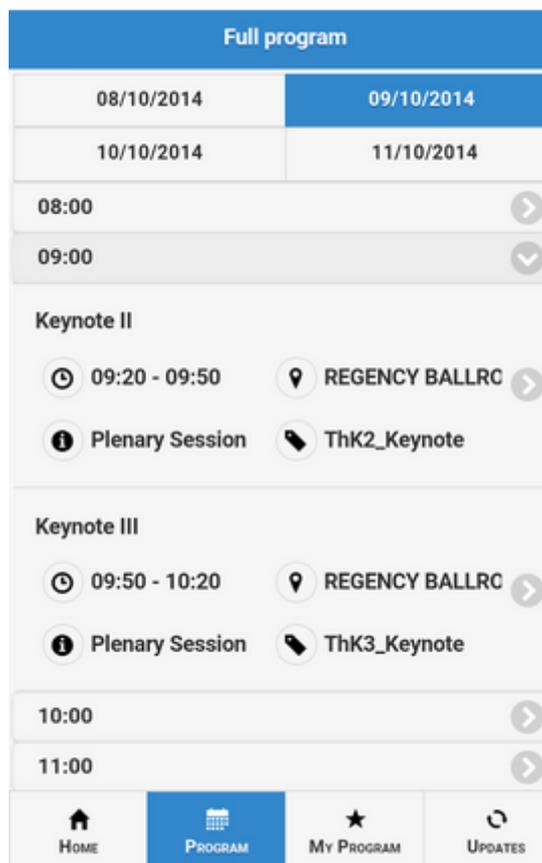


Ilustración 2-40 - Programa para el día 9, desplegado a las 9:00

Cabe mencionar que para esta versión de la aplicación hemos establecido un número de cuatro días para la duración del congreso de manera estática, aunque su contenido se genere de manera dinámica. En un futuro cercano planeamos realizar una carga dinámica del número de días del congreso a partir de los datos del servidor Back-End, ya que los congresos no deberían estar obligados a durar cuatro días.

2.5.4.4.3 Vistas con pestañas

En JQuery Mobile, existen dos formas principales para construir estos componentes: Describiéndolos en el código HTML, mediante atributos data-role, o bien a través de JavaScript empleando selectores desde JQuery. Este segundo método permite, por ejemplo, generar código HTML dinámicamente desde JavaScript, y posteriormente estilizarlo mediante JQuery Mobile para agregarle comportamientos especiales.

Nuestra aproximación inicial para construir las vistas que incluyeran elementos con pestañas partía de este sistema. El plan consistía en crear todo el código HTML de la página a mostrar, y posteriormente estilizarlo con JQuery Mobile para formar las pestañas que dividieran el contenido. Este es el proceso que empleamos para construir elementos como listados o botones, pero no pudo ser posible completarlo correctamente para el caso de las pestañas. Esto se debe a que existe un error en las librerías de JQuery Mobile que impide que la estilización programática de elementos se complete satisfactoriamente.

Como resultado de este error, nuestra solución para estos casos sufrió algunas modificaciones. En lugar del plan inicial, las páginas que incluyan pestañas son construidas, primero, con las pestañas vacías, empleando atributos data-role para ello y, a continuación, inyectamos el código HTML por separado para cada región de la página resultante.

El mayor inconveniente de la aproximación que hemos seleccionado es que se ha multiplicado el número de veces que una función debe llamarse para el renderizado de ciertas páginas por el número de regiones diferentes que posea. Frente a una página que no presente este problema, en ocasiones debemos invocar a la misma función de renderizado hasta cuatro veces, con parámetros diferentes, para que genere el código HTML correspondiente a cada área. A pesar de todo, cabe mencionar que este problema no presenta efectos negativos severos en el rendimiento de la aplicación.

Las vistas afectadas por este error son:

- Índice de sesiones (programa completo)
- Índice de comités
- Vista de detalle de un autor
- Vista de detalle de una sesión
- Vista de detalle de una temática

2.5.5 Comparativa entre los desarrollos nativos Android e iOS

Más allá de las diferencias obvias entre el desarrollo de ambas plataformas, en esta sección pretendemos explicar ciertos problemas a los que nos hemos enfrentado durante el desarrollo. Es el caso de soluciones que se plantearon en un principio pero luego no pudieron ser implementadas en ambas plataformas, o sencillamente limitaciones y errores ajenos al proyecto que nos encontramos durante el desarrollo del mismo.

2.5.5.1 Esquema de documentos HTML

Durante el desarrollo de las maquetas para las vistas web, planteamos cada vista como un documento HTML aparte, teniendo finalmente más de una decena de documentos HTML. Esta configuración nos pareció en principio la más lógica y legible para el código.

Tras completar la carga del maquetado web estático en la aplicación, procedimos a integrar el contenido dinámico en las vistas web. En el desarrollo para Android no encontramos ningún problema, puesto que la comunicación entre Android y JavaScript es directa, y es el propio código JavaScript el que recibe el resultado de la solicitud al código nativo de Android. Sin embargo durante el desarrollo para iOS, detectamos que al ser el propio código nativo de Swift el que debía inyectar los datos a JavaScript por medio del documento cargado en el *WebView* (la vista `index.html` que contiene el menú principal), por tanto sólo podía alterar los elementos contenidos en esa vista, independientemente de la vista que se estuviera mostrando en el navegador en ese momento.

Para solucionar esto decidimos realizar un único documento HTML con todas las vistas en él, separándolas mediante las herramientas que proporciona JQuery Mobile. Tras esto pudimos realizar la integración sin problemas en iOS. No obstante encontramos utilizar este documento único en Android suponía un perjuicio notable al rendimiento (esto no fue así en iOS). Esto nos obligó a utilizar el sistema de páginas separadas en Android.

Hay que mencionar que esta decisión también nos obligó a manejar de forma diferente la transferencia de identificadores entre los listados y las vistas de detalle, ya que en un principio planteamos realizarlo empleando parámetros GET de HTTP. Esto no fue posible en el documento único utilizado por iOS, por lo que hemos planteado una solución basada una pila de identificadores que crece y decrece a la par que el historial de navegación de la aplicación para cumplir el mismo fin.

2.5.5.2 Tiempo de carga de datos

Hemos apreciado que la inserción de los datos en la base de datos presenta importantes variaciones de tiempo entre ambas plataformas. El tiempo de carga de datos en iOS es del orden de segundos y ocurre de forma totalmente transparente al usuario, mientras que en Android este proceso se toma varios minutos, por lo que ha sido necesario notificar al usuario del progreso de la carga durante su desarrollo.

No podemos concluir con certeza al respecto de una causa concreta para esta diferencia en los tiempos de ejecución. Planteamos así un estudio de las diferencias de rendimiento que pueden implicar el uso de bases de datos no relacionales, como es el caso de Swift, o una revisión del código empleado para el procesado de los datos JSON.

2.5.5.3 Lenguaje de programación

Es una diferencia importante en el desarrollo apreciar que Android no son más que librerías de Java, frente al hecho de que Swift es un lenguaje completo. Partiendo de la base de que Java nos resulta un lenguaje conocido que hemos estudiado ampliamente durante el desarrollo de nuestra vida académica universitaria, la complejidad para aprender a manejar un lenguaje frente a entender el uso de unas librerías no resulta comparable al no poseer cualidades similares.

Cada plataforma posee sus propios mecanismos y herramientas para el desarrollo, y este trabajo nos ha servido para conocer no sólo la plataforma en la que cada uno ha desarrollado, sino también la del otro proyecto hermano.

2.5.5.4 Comunicación entre nativo y JavaScript

Hemos apreciado que las metodologías a emplear para desarrollar elementos de comunicación entre nativo y JavaScript en ambas plataformas presentan aproximaciones radicalmente opuestas. Un proceso como la invocación directa de métodos de Java desde JavaScript, como es el caso de Android, debe ser replicado en iOS mediante la inyección del propio código JavaScript desde Swift como respuesta a un evento concreto. Para esta comunicación utiliza un sistema de mensajería desde JavaScript el cual es rescatado por unos manejadores de eventos en Swift, procediendo a inyectar el código JavaScript en el documento cargado en el *WebView*. Esta inconveniencia ha supuesto que, para la etapa de integración, los desarrollos pasaran a ser completamente independientes y apenas pudieran compartir planteamientos algorítmicos.

2.5.5.5 Errores de iOS

Durante el desarrollo en iOS nos encontramos con dos complicaciones reconocidas por Apple. Estos problemas pretenden ser resueltos en las futuras versiones del sistema de Apple.

El primer problema que encontramos es un error que produce que el código CSS no se aplique correctamente al cambiar de orientación el dispositivo móvil. La solución no era precisamente trivial y requería una cantidad de tiempo de la que no disponíamos, por tanto optamos por una solución temporal hasta que Apple corrija este problema.

La solución que implementamos fue bloquear la rotación del dispositivo, permitiendo que la aplicación se pueda ejecutar con el dispositivo en vertical o en horizontal. No obstante, la visualización permanecerá con esa orientación durante toda su ejecución.

El segundo problema es de una envergadura bastante superior. En la última versión de Apple existe un error que impide cargar ficheros externos en el proyecto desde un dispositivo real (en el emulador no existe tal problema). Debido a la naturaleza de nuestras vistas que se constituyen de ficheros externos, es necesario acceder a ellos. Debido a este error, no basta únicamente con situarlos dentro del proyecto.

Existen al menos dos soluciones para este problema. La primera consistía en integrar un servidor web del que extraer los ficheros. Dado que uno de los objetivos de nuestro desarrollo era no utilizar servidor web, decidimos descartar esta solución directamente. La solución implementada consiste en copiar los documentos externos al sistema de ficheros temporales, ya que este sistema de directorios sí es accesible desde el dispositivo. Esto se realiza al ejecutar la aplicación, y no consideramos razonable tener que utilizar este tipo de medidas. No obstante, el código que realiza esta transferencia está encapsulado, y procederemos a eliminarlo tan pronto Apple solucione el problema en cuestión.

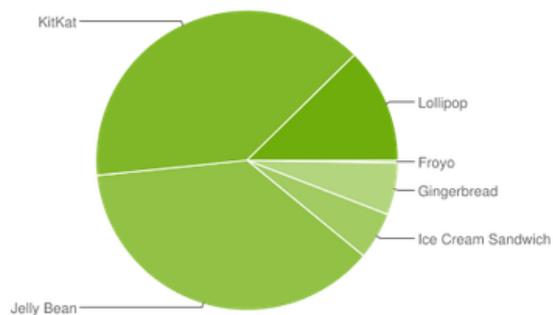
2.5.5.6 Fragmentación de versiones en Android

En Android, el desarrollo de una aplicación implica ajustarse a las librerías que proporciona una versión concreta del kit de desarrollo. Usualmente, cuanto más potente es una librería, mayor versión requiere el dispositivo para poder ejecutar la aplicación, con el consiguiente descenso en el porcentaje global de dispositivos que podrían instalarla en primer lugar.

Hemos sacrificado un pequeño porcentaje de dispositivos Android que no podrán hacer uso de nuestra herramienta, con la justificación de obtener un producto final funcional ajustándose a los recursos de tiempo de los que disponíamos para llevar a cabo este proyecto. Se plantea para el futuro el reajuste del código para mejorar la compatibilidad de la aplicación.

Esta versión de la aplicación requiere Android 4.1 Jelly Bean (API nivel 16) o superior para ser instalada, versión mínima para cerca del 90% de los dispositivos Android activos en el mercado. A continuación se incluye un gráfico descriptivo de la distribución actual de versiones de Android obtenido de [la página oficial de la plataforma](#).

| Version | Codename | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.2 | Froyo | 8 | 0.3% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 5.6% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 5.1% |
| 4.1.x | Jelly Bean | 16 | 14.7% |
| 4.2.x | | 17 | 17.5% |
| 4.3 | | 18 | 5.2% |
| 4.4 | KitKat | 19 | 39.2% |
| 5.0 | Lollipop | 21 | 11.6% |
| 5.1 | | 22 | 0.8% |



Data collected during a 7-day period ending on June 1, 2015.

Any versions with less than 0.1% distribution are not shown.

Ilustración 2-41 - Distribución de versiones de Android a junio de 2015

3 Conclusiones y trabajos futuros

En este capítulo final expondremos las conclusiones a las que hemos llegado tras el desarrollo de este Trabajo de Fin de Grado. Explicaremos algunos problemas con los que hemos lidiado durante el desarrollo, y analizaremos si se han logrado los objetivos establecidos para este desarrollo. Seguidamente, expondremos nuestros pensamientos sobre el desarrollo híbrido tanto para este proyecto como para otro tipo de trabajos.

Finalmente, hablaremos de los posibles planes de futuro para la aplicación, donde se incluirán posibles mejoras a la misma y desarrollos que no se han podido terminar durante el tiempo establecido para este proyecto.

3.1 Conclusiones

Debido a que el desarrollo de este TFG tiene como objetivo principal el desarrollo de una aplicación, pero además incluye como objetivo secundario explorar y compartir nuestra experiencia con el desarrollo híbrido, hemos decidido dividir esta sección en dos partes donde tratar ambos objetivos.

3.1.1 Conclusiones sobre el desarrollo de la aplicación

Aunque los TFG no suelen verse sometidos a la misma presión que un proyecto real, como por ejemplo presiones por parte de inversores o fechas inamovibles, sí que existen unas fechas de entrega y un límite de horas que no deberían sobrepasarse. Dada la naturaleza de nuestro proyecto hemos tenido que estudiar en detalle las últimas tendencias y tecnologías utilizadas tanto en el desarrollo web como en el desarrollo de aplicaciones para dispositivos móviles.

Nos hemos visto especialmente obligados a destinar un tiempo considerable a aprender el uso de los lenguajes nativos para iOS y Android, ya que esto era una necesidad indispensable para poder llevar a cabo nuestro proyecto. Esto nos ha limitado severamente en las horas destinadas al desarrollo de la aplicación. No obstante, creemos más que justificada la inversión dado nuestro enorme interés en aprender el uso de estas tecnologías, ya que un TFG no sólo sirve para demostrar nuestros conocimientos, sino para aprender y crecer como profesionales, así como para demostrar que podemos adquirir nuevas destrezas de manera autónoma tal y como se solicita en las competencias a cubrir durante el desarrollo del TFG.

Aunque nos encontrásemos con este tipo de problemas, hemos sido capaces de desarrollar un primer incremento de la aplicación funcional en el cual están ya desarrolladas la enorme mayoría de las funcionalidades de la misma, quedando el resto como trabajos futuros. Por tanto creemos férreamente haber alcanzado exitosamente los objetivos académicos planteados por el TFG.

Para lograr todo lo que ha supuesto el desarrollo de este proyecto hemos puesto en práctica varios de los conocimientos adquiridos durante nuestra formación académica, especialmente en el ámbito de las asignaturas que tratan los siguientes ámbitos:

- Bases de datos
- Desarrollo de aplicaciones web
- Ingeniería del software
- Seguridad en los sistemas de información
- Tecnologías de los sistemas de información

Consideramos que nuestro proyecto genera valor al aportar una aplicación útil que cubre una necesidad real, ya que hemos realizado un sistema de bajo coste para el soporte de la gestión de congresos que puede ser rentable y accesible a congresos que normalmente no pueden permitirse una aplicación de este tipo con su presupuesto. Aunque para ello tendrían que finalizarse los desarrollos que planteamos en la sección de trabajos futuros.

A su vez, creemos haber creado algo útil durante nuestro proyecto al crear un acercamiento hacia el desarrollo híbrido, hablando del mismo y explicando a cada paso donde lo hemos integrado. También estableceremos nuestras conclusiones al respecto en la siguiente sección.

Finalmente nos gustaría mencionar que nos sentimos muy contentos con el desarrollo del proyecto. Hemos aprendido mucho por el camino y hemos mejorado como profesionales, ya que nos hemos enfrentado a un desarrollo muy semejante a lo que nos encontraremos en nuestra futura vida laboral. Además, consideramos que hemos aprendido mucho sobre trabajar en equipo, superando constantemente nuestras diferencias y creando algo que ha merecido la pena.

3.1.2 Conclusiones sobre el desarrollo híbrido

Pasada la experiencia del desarrollo de este proyecto, hemos podido percibir las principales características de una aplicación híbrida y lo que su implementación supone a grandes rasgos. El desarrollo híbrido es una herramienta muy potente y con un enorme potencial que permite obtener aplicaciones de complejidad medio-alta en tiempos asumibles. Esto se debe a que, como hemos podido apreciar, obtenemos por un lado todas las ventajas de un lenguaje de programación nativo, combinado con la agilidad de implementación de una página web como interfaz gráfica.

No obstante, también comprendemos que el desarrollo híbrido tal y como lo hemos llevado a cabo no es la solución ideal para todas las aplicaciones móviles. Existiendo herramientas que permiten obtener soluciones a través de un desarrollo web completo, la pregunta sobre la utilidad del desarrollo híbrido en proyectos de baja complejidad computacional no posee una respuesta inmediata. Entendemos de esta forma que quizás no seleccionamos el mejor caso de uso para experimentar hacia el desarrollo híbrido, puesto que nuestro proyecto habría sido implementado con un uso de recursos, especialmente en tiempo, mucho más óptimo de haber suprimido la parte nativa del desarrollo. Bajo nuestro punto de vista y por las características de la aplicación, no hemos sido capaces de exprimir al máximo tanto la parte nativa del desarrollo como la interfaz web.

En respuesta a estos hechos, planteamos, para posibles proyectos futuros, la posibilidad de lanzar una nueva aplicación empleando el desarrollo híbrido como herramienta. Esta aplicación debería poseer un gran número de características diferentes frente a las de este TFG, aproximándose a la tecnología escogida con mayor ambición. Esto permitiría arrojar más luz sobre los resultados que puede ofrecer este sistema de desarrollo y sus límites, agregando la posibilidad de emplear nuevas herramientas o utilizar las actuales de formas radicalmente diferentes. Sugerimos, por ejemplo, estudios de la viabilidad del elemento *canvas* de HTML5 en el desarrollo híbrido, una de las herramientas con mayor potencial en el desarrollo web actual, o explotar las capacidades de los lenguajes nativos en tareas más complejas que la gestión y el acceso a la capa de datos de la aplicación.

Para poder entender una herramienta, es preciso conocer no lo que puede conseguir, sino lo que no puede conseguir, y estudiar estos límites. Creemos que el desarrollo híbrido no termina aquí, sino que esto no es más que el comienzo.

3.2 Trabajos futuros

Para finalizar esta memoria propondremos una serie de mejoras y/o cambios en la aplicación que podrían llevarse a cabo en un futuro cercano. Algunas de estas funcionalidades ya se contemplaban en la aplicación de partida, por lo que hemos optado por dividir esta sección final en dos partes. En la primera parte hablaremos de las funcionalidades que no se han llegado a implementar debido al carácter limitado del tiempo disponible durante el proyecto, y en la segunda parte propondremos nuevas funcionalidades que hemos considerado interesantes tras realizar nuestro pertinente estudio del estado del arte.

3.2.1 Funcionalidades no desarrolladas

3.2.1.1 Generación de la vista “My Program”

Esta vista concede al usuario la posibilidad de construir un listado de conferencias a la carta, pudiendo crear su propio programa sólo con las charlas que le interesen.

Este sistema de “favoritos” deberá ser desarrollado a nivel de la aplicación, ya que es completamente ajeno al servidor *Back-End* y propio de cada dispositivo en el que se utilice la aplicación.

3.2.1.2 Sistema de usuarios registrados

Con este sistema se pretendía añadir contenido de valor accesible sólo para los usuarios registrados. Entre otros privilegios, se les permitiría descargar los *papers* de las charlas disponibles, acceder a ficheros externos de los eventos sociales, etc.

Para llevar a cabo este sistema, se enviarían los datos del usuario desde la vista “My Account”, cifrando los datos en la transferencia. Tras esto, desde el servidor *Back-End* se procedería a descifrar dichos datos y buscar en su base de datos de usuarios. Finalmente se respondería a nuestra aplicación con el resultado de dicha búsqueda. En caso negativo en la autenticación, se procedería a mostrar un mensaje de error al usuario, y en caso contrario se procedería a desbloquear el acceso al contenido limitado a los usuarios registrados desde la aplicación.

3.2.1.3 Generación dinámica del número de días del congreso

Actualmente la vista del programa presenta cuatro pestañas de días de manera estática, y dentro de cada una de estas pestañas se inyecta de manera dinámica el contenido de dicho día. No obstante, en esta versión de la aplicación, no se realizan consultas a la base de datos para generar el número de días del congreso y luego generar una pestaña para cada día. Por tanto, una de las primeras modificaciones a realizar en la aplicación, y para mantener un sistema flexible a congresos de diferentes duraciones, se debería cambiar la generación de este contenido, procediendo a crear un número dinámico de pestañas según los días del congreso a partir de los datos almacenados en la base de datos.

3.2.2 Propuesta de nuevas funcionalidades

3.2.2.1 Interacción con las redes sociales

Las redes sociales son utilizadas de manera global a día de hoy, por lo que consideramos muy importante dar la posibilidad de comunicarse con las mismas por desde nuestra aplicación.

Posibles usos al respecto podrían incluir desde compartir la aplicación, enlazar los perfiles de los autores con los de las redes sociales, publicar eventos, o sencillamente usarlas con fines publicitarios para el congreso.

3.2.2.2 Sistema de valoración de las conferencias

Consideramos valioso el realizar un sistema que permita al usuario calificar las charlas a las que ha asistido, incluso pudiendo dejar algún comentario sobre las mismas.

Con esta funcionalidad se podría conseguir una importante fuente de retroalimentación, generando información de valor para los autores, ayudándoles a descubrir sus puntos fuertes y débiles con la ayuda de los asistentes al congreso.

3.2.2.3 Inclusión de planos de las instalaciones

Creemos que podría ser muy interesante incluir planos, aunque sea en formato de imágenes, de los recintos y las salas en las que tendrán lugar los eventos del congreso. Esta sería una información de valor para los asistentes al congreso, facilitándoles su orientación en el recinto.

Por otro lado creemos que este tipo de medidas ayudarán finalmente a los encargados de la gestión del congreso, ya que evitarán varias dudas, reduciendo la cantidad de asistentes que requieren ayuda por parte de los organizadores para saber dónde se encuentran las salas.

3.2.2.4 Aumento del contenido multimedia en la interfaz

Actualmente nuestra interfaz carece casi por completo de elementos multimedia, pudiendo crear una sensación de sobriedad. Quizá sería interesante desarrollar una nueva interfaz que se valiera, por ejemplo, de imágenes en vez de textos en el menú principal o algún tipo de elemento más atractivo visualmente para los usuarios.

4 Fuentes de información

4.1 Información sobre los congresos

- <http://its.ieee.org/>
- https://en.wikipedia.org/wiki/IEEE_Intelligent_Transportation_Systems_Society

- <http://www.cesweb.org/>
- https://en.wikipedia.org/wiki/Consumer_Electronics_Show

- <http://www.mobileworldcongress.com/>
- https://en.wikipedia.org/wiki/Mobile_World_Congress

- <http://www.gamescom-cologne.com/>
- <https://en.wikipedia.org/wiki/Gamescom>

- <http://www.e3expo.com/>
- https://en.wikipedia.org/wiki/Electronic_Entertainment_Expo

- <http://www.dog.org/>
- https://de.wikipedia.org/wiki/Deutsche_Ophthalmologische_Gesellschaft

4.2 Normativa y legislación

- <http://www.agpd.es/>
- https://es.wikipedia.org/wiki/Ley_Orgánica_de_Protección_de_Datos_de_Carácter_Personal_de_España
- <http://www.congreso.es/consti/constitucion/indice/titulos/articulos.jsp?ini=18&tipo=2>

- https://es.wikipedia.org/wiki/Licencia_de_software
- <https://en.wikipedia.org/wiki/Shareware#Trialware>
- https://en.wikipedia.org/wiki/GNU_General_Public_License

- http://www2.ulpgc.es/hege/almacen/download/1/1966/Estatutos_definitivo.pdf

4.3 Metodología

- <http://agilemanifesto.org/>

4.4 Herramientas

- https://en.wikipedia.org/wiki/Microsoft_Office_2013
- <http://www.si.ulpgc.es/Office-para-los-estudiantes>
- https://en.wikipedia.org/wiki/Revision_control
- [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))
- <https://en.wikipedia.org/wiki/Bitbucket>
- <https://en.wikipedia.org/wiki/Model-view-controller>

- [https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))
- <https://en.wikipedia.org/wiki/Xcode>
- <https://developer.apple.com/xcode/ide/>
- https://en.wikipedia.org/wiki/Core_Data
- <https://developer.apple.com/library/ios/navigation/>
- <https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html>
- https://developer.apple.com/library/prerelease/ios/documentation/Cocoa/Reference/Foundation/ObjC_classic/index.html
- https://developer.apple.com/library/prerelease/mac/documentation/Cocoa/Reference/UIKit/ObjC_classic/index.html
- https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIKit_Framework/

- <https://en.wikipedia.org/wiki/HTML>
- <https://en.wikipedia.org/wiki/HTML5>
- https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- <https://en.wikipedia.org/wiki/JavaScript>
- <https://en.wikipedia.org/wiki/Notepad++>
- <http://nicolas.github.io/normalize.css/>
- <https://en.wikipedia.org/wiki/JQuery>
- [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))
- https://en.wikipedia.org/wiki/JQuery_Mobile
- <https://en.wikipedia.org/wiki/PhoneGap>
- <https://en.wikipedia.org/wiki/Firefox>

4.5 Análisis

- https://es.wikipedia.org/wiki/Ingeniería_de_requisitos
- https://en.wikipedia.org/wiki/Requirements_engineering
- Apuntes de la asignatura "Tecnología Software para el Desarrollo de Sistemas de Información" sobre la gestión de requisitos – Sonia Marrero Cáceres (Escuela de Ingeniería Informática)

4.6 Diseño

- https://en.wikipedia.org/wiki/Software_design_pattern
- https://en.wikipedia.org/wiki/Software_architecture
- https://en.wikipedia.org/wiki/Client-server_model

5 Anexos

5.1 Anexo I – Detalle de los congresos

5.1.1 CES – Consumer Electronics Show

El *International CES*, más conocido como el *Consumer Electronics Show* (Feria Internacional de Electrónica de Consumo), es un renombrado evento internacional de electrónica y tecnología en el cual se reúnen las principales compañías y profesionales de la industria a nivel mundial. Esta feria de muestras tiene lugar en enero en el Centro de Convenciones de Las Vegas (Nevada, Estados Unidos).

| <i>International CES</i> | | |
|---|----------------------|---|
| Página web | Estado | Activo |
|  | Género | Electrónica de consumo |
| | Localización | Las Vegas, Nevada, Estados Unidos |
| | Inauguración | 1967 |
| | Asistentes | 140.000 (2015) |
| | Organizadores | <i>Consumer Electronics Association</i> |

Tabla 5-1 - Detalle de CES

5.1.2 MWC – Mobile World Congress

El *GSMA Mobile World Congress* (Congreso Mundial de Telefonía Móvil) es una combinación de la mayor exhibición de la industria móvil del mundo y una serie de conferencias de importantes ejecutivos en representación de operadores móviles, fabricantes de dispositivos, proveedores de tecnología, vendedores y dueños de contenidos de todo el mundo.

| <i>GSMA Mobile World Congress</i> | | |
|---|----------------------|-------------------|
| Página web | Estado | Activo |
|  | Género | Telefonía móvil |
| | Localización | Barcelona, España |
| | Inauguración | 1987 |
| | Asistentes | 85.000+ (2014) |
| | Organizadores | <i>GSMA</i> |

Tabla 5-2 - Detalle de MWC

5.1.3 Gamescom

La *Gamescom* es una feria de muestras centrada en videojuegos que tiene lugar anualmente en Colonia, Alemania. Es organizada por el *BIU* (*Bundesverband Interaktive Unterhaltungssoftware*, Asociación Federal del Software de Entretenimiento Interactivo), y multitud de desarrolladores de videojuegos se reúnen aquí para presentar sus próximos productos software y hardware relacionado.

| <i>Gamescom</i> | | |
|---|---------------|-------------------|
| Página web | Estado | Activo |
|  | Género | Videjuegos |
| | Localización | Colonia, Alemania |
| | Inauguración | 2009 |
| | Asistentes | 335.000 (2014) |
| | Organizadores | <i>BIU</i> |

Tabla 5-3 - Detalle de *Gamescom*

5.1.4 E3 – Electronic Entertainment Expo

La *Electronic Entertainment Expo* (Exposición de Entretenimiento Electrónico), más conocida como *E3*, es una feria de muestras internacional dedicada a los videojuegos, presentada por la *ESA* (*Entertainment Software Association*, Asociación de Software de Entretenimiento). Un amplio número de distribuidores de videojuegos y fabricantes asisten a este evento para presentar sus próximos productos.

| <i>E3</i> | | |
|---|---------------|---|
| Página web | Estado | Activo |
|  | Género | Videjuegos |
| | Localización | Los Ángeles, California, Estados Unidos |
| | Inauguración | 1995 |
| | Asistentes | 48.900 (2014) |
| | Organizadores | <i>ESA</i> |

Tabla 5-4 - Detalle de *E3*

5.1.5 DOG – Deutsche Ophthalmologische Gesellschaft

La *Deutsche Ophthalmologische Gesellschaft* (Sociedad Alemana de Oftalmología, DOG) es la sociedad científica profesional de Alemania centrada en oftalmología. Sus principales actividades son la investigación, el desarrollo y la promoción de la oftalmología.

| <i>DOG</i> | | |
|---|----------------------|------------------|
| Página web | Estado | Activo |
|  | Género | Oftalmología |
| | Localización | Berlín, Alemania |
| | Inauguración | 1857 |
| | Asistentes | 6.600+ (2014) |
| | Organizadores | <i>DOG</i> |

Tabla 5-5 - Detalle de *DOG*

5.2 Anexo II – Capturas de pantalla de la maqueta

5.2.1 Home

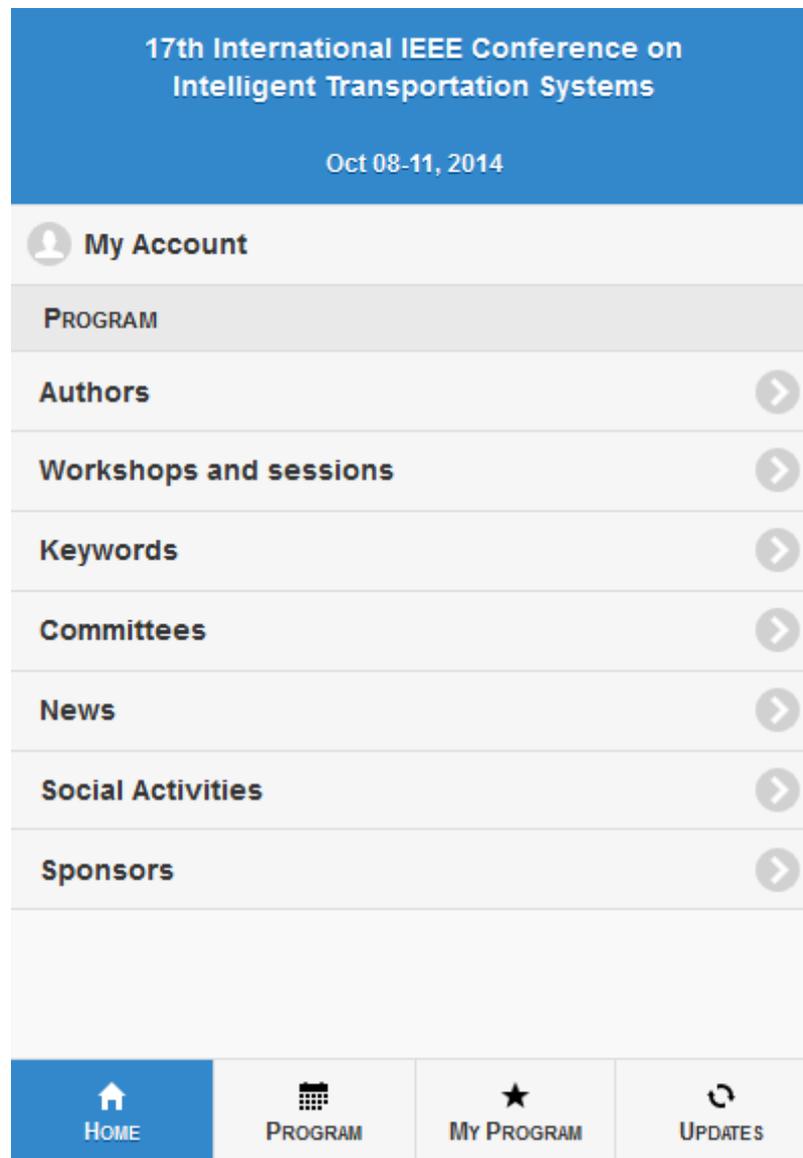


Ilustración 5-1 - Captura de pantalla de la página principal

5.2.2 My Account

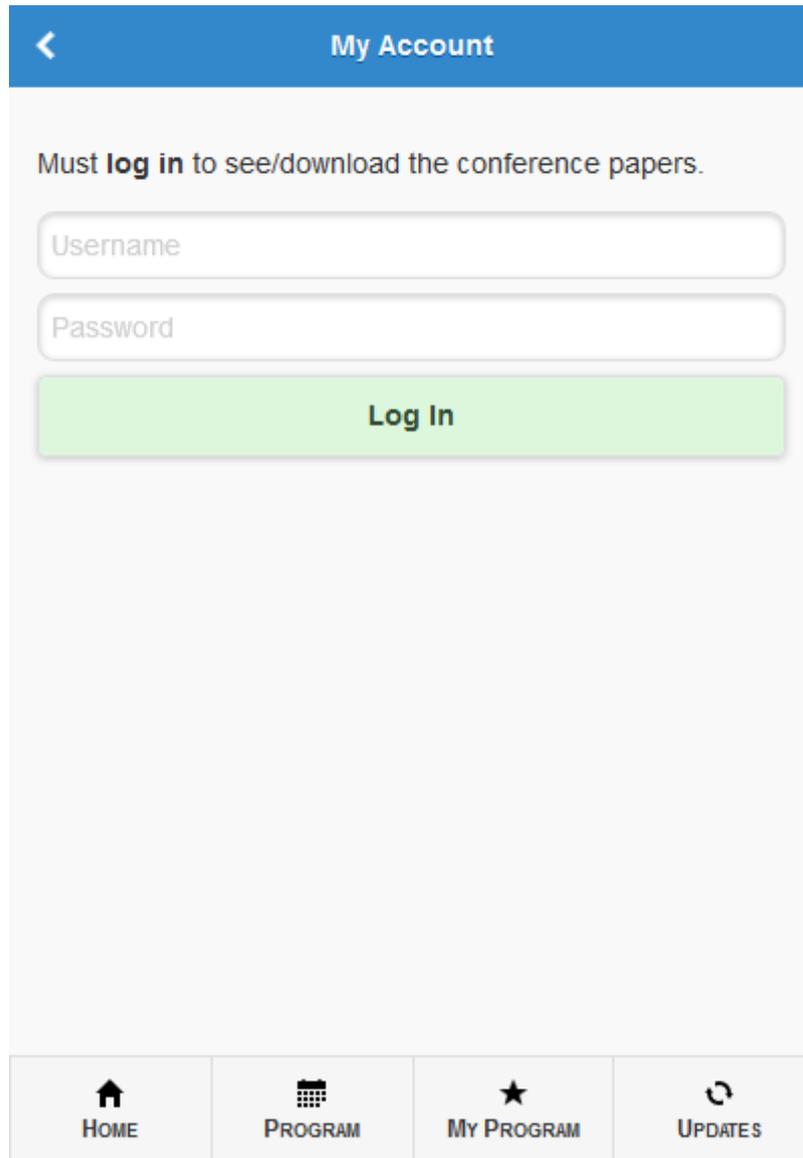


Ilustración 5-2 - Captura de pantalla de la página de inicio de sesión

5.2.3 Program

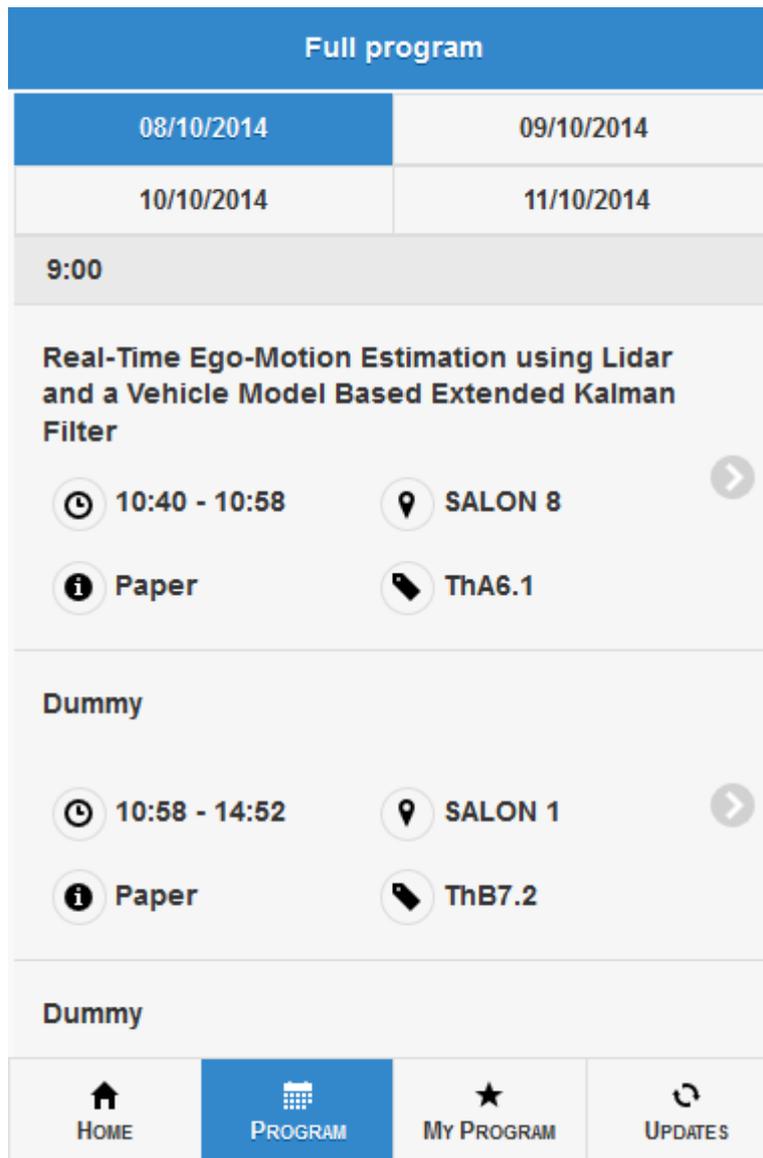


Ilustración 5-3 - Captura de pantalla del programa completo

5.2.4 Update

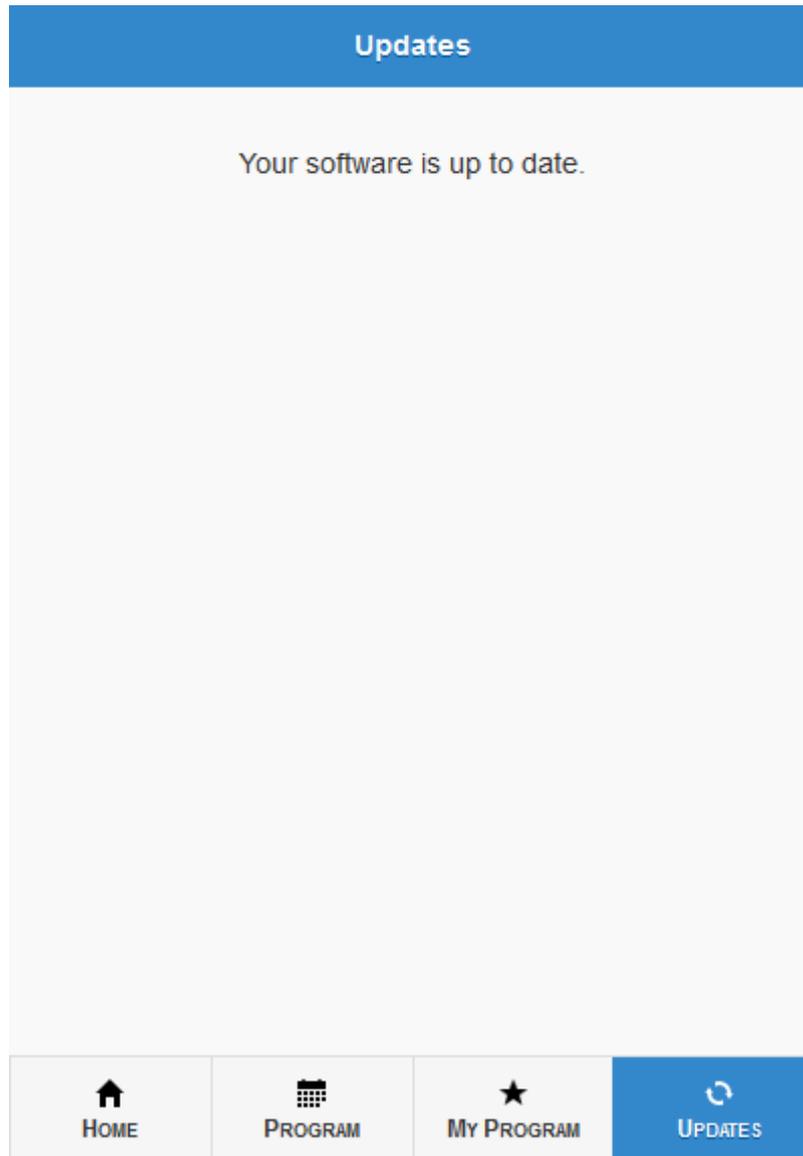


Ilustración 5-4 - Captura de pantalla de la página de actualizaciones

5.2.5 Authors

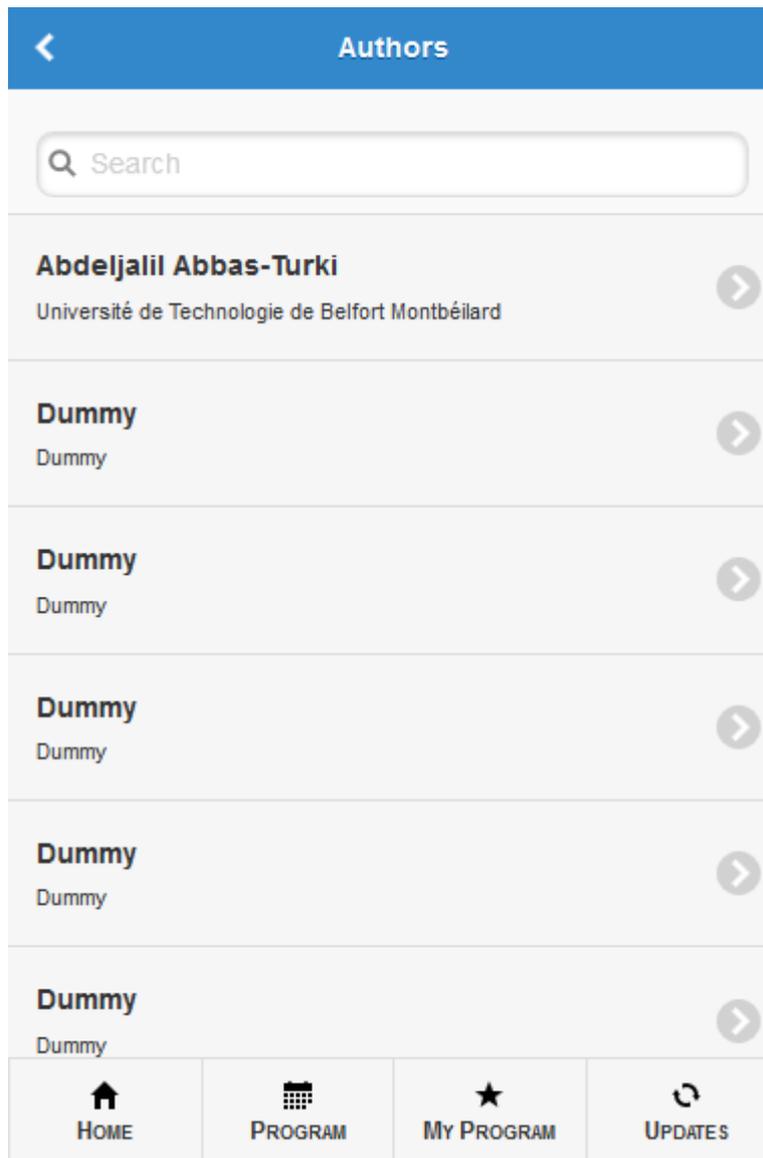


Ilustración 5-5 - Captura de pantalla del listado de autores

5.2.6 Workshops

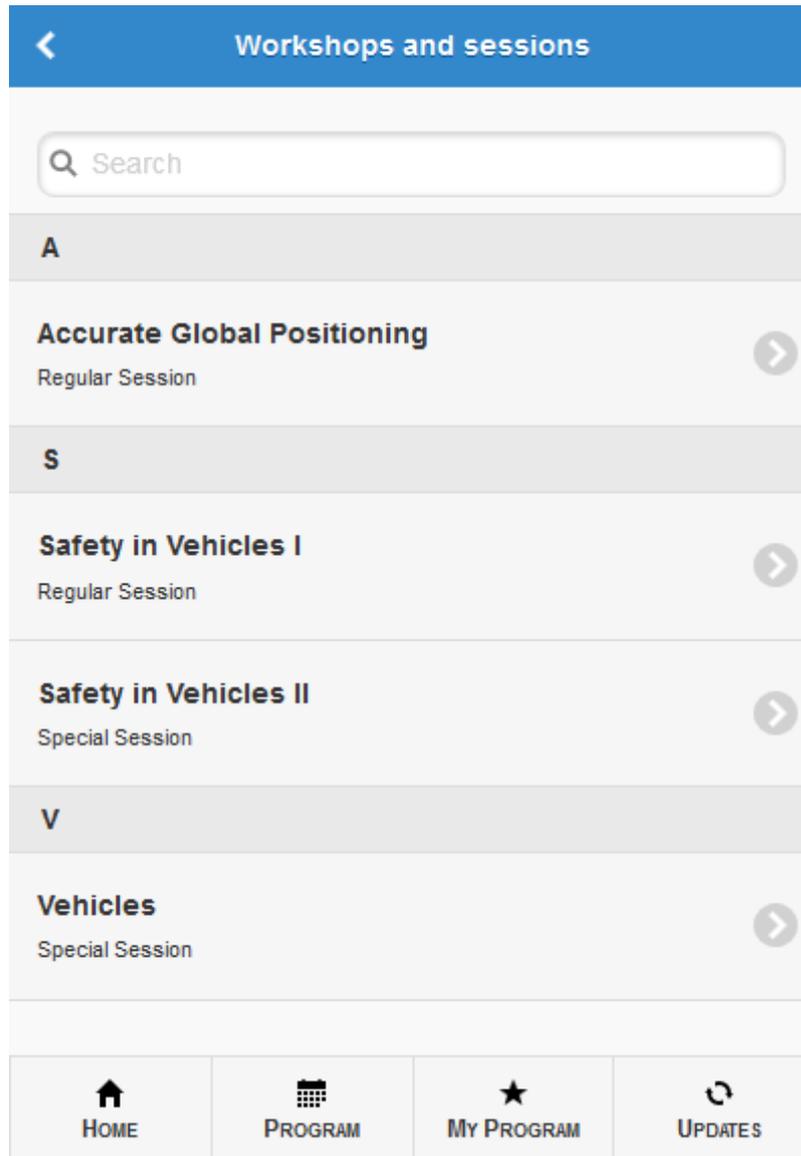


Ilustración 5-6 - Captura de pantalla del listado de temáticas

5.2.7 Keywords

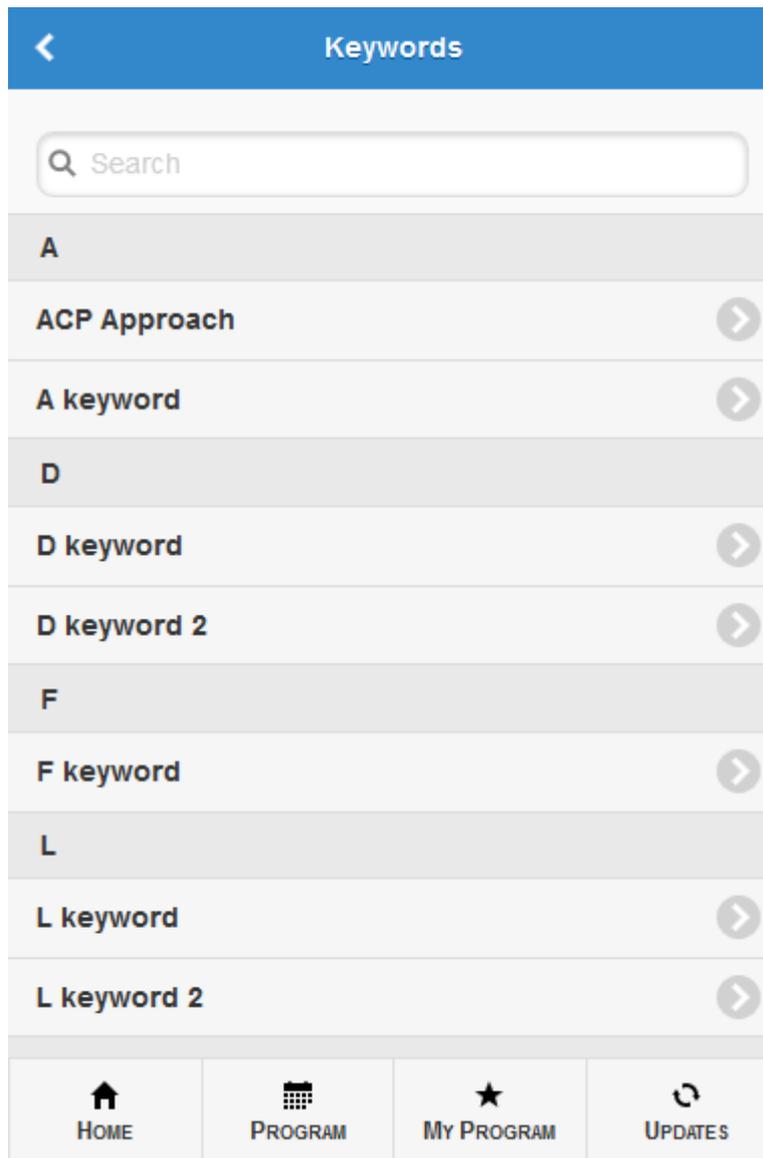


Ilustración 5-7 - Captura de pantalla del listado de palabras clave

5.2.8 Committees

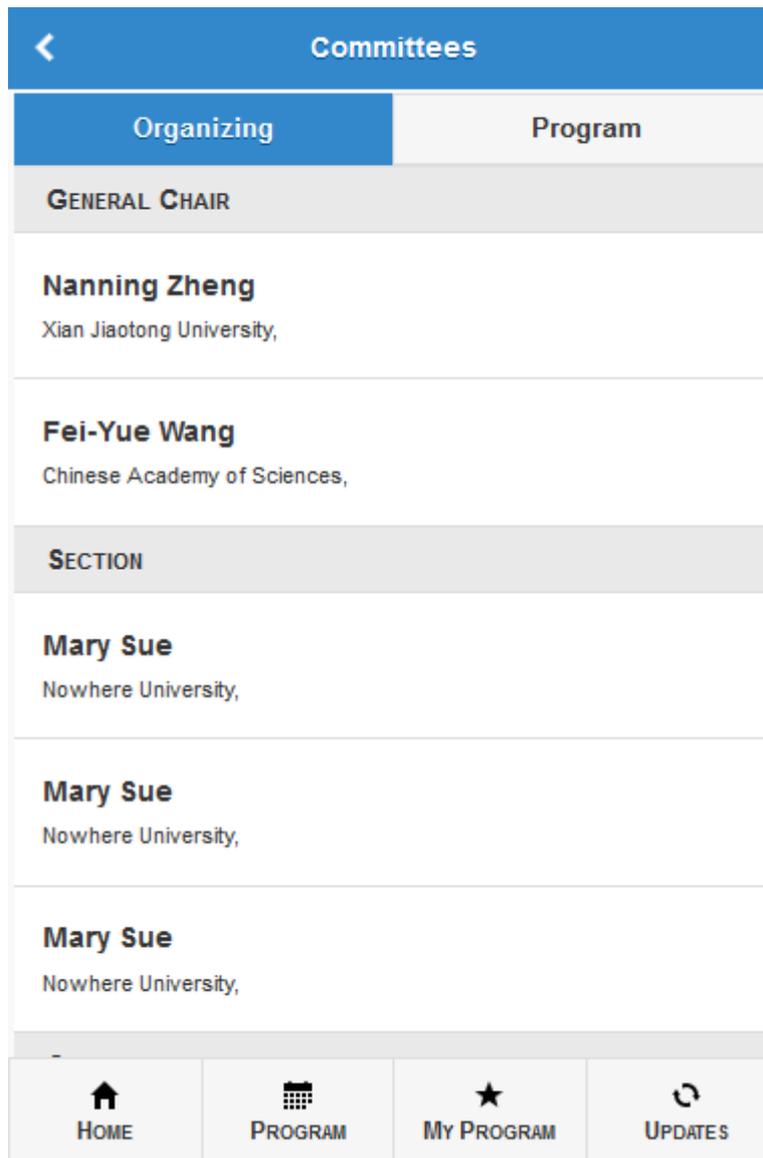


Ilustración 5-8 - Captura de pantalla del listado de comités

5.2.9 News

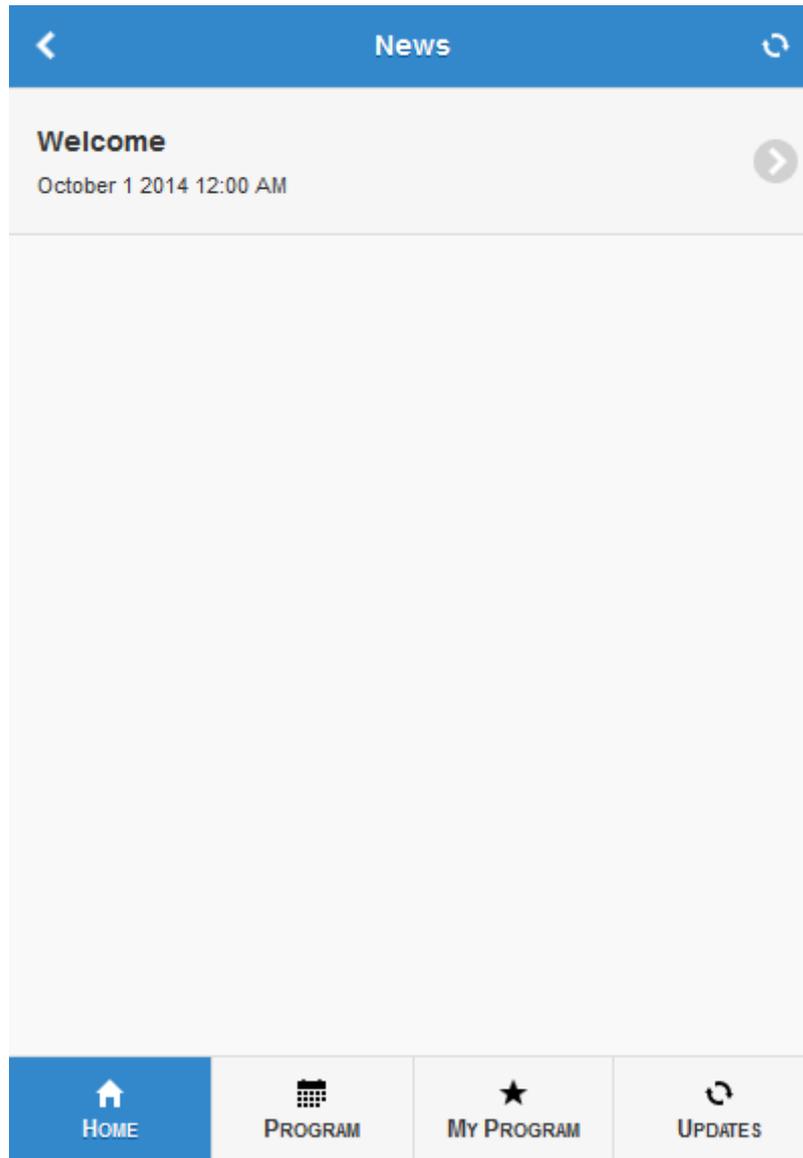


Ilustración 5-9 - Captura de pantalla del listado de noticias

5.2.10 Social Activities

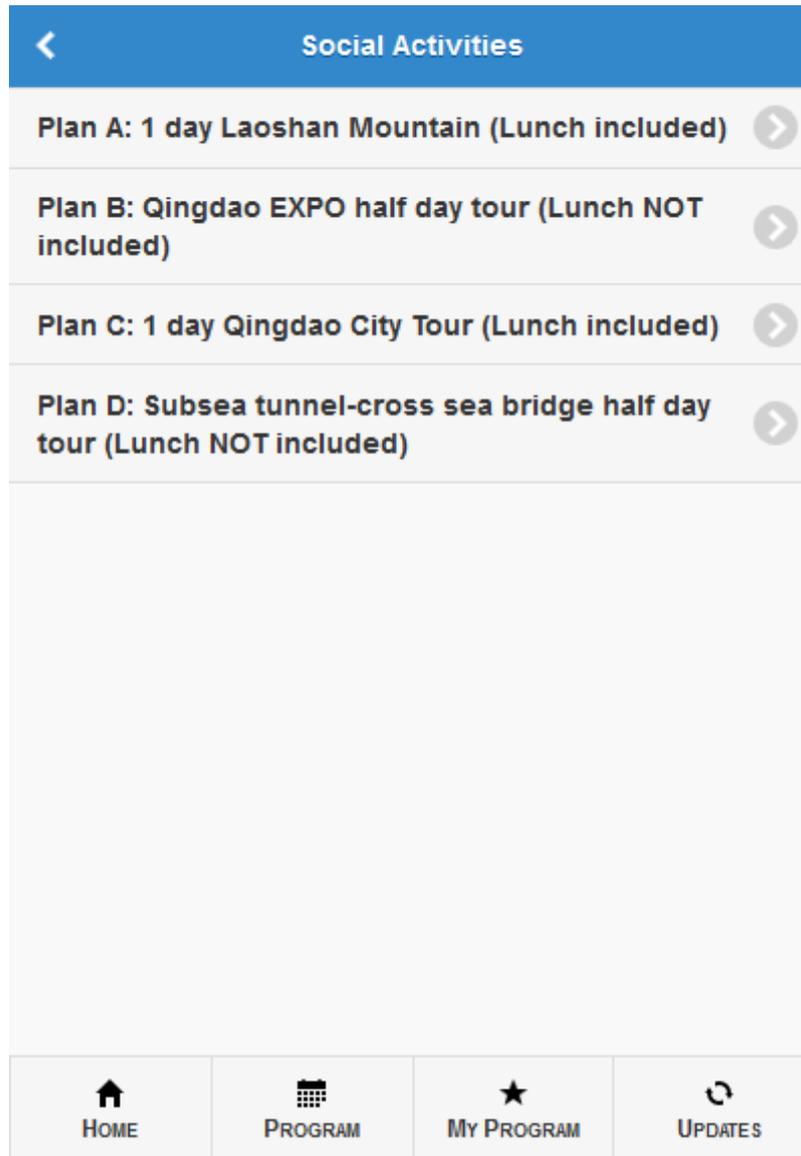


Ilustración 5-10 - Captura de pantalla del listado de actividades sociales

5.2.11 Sponsors

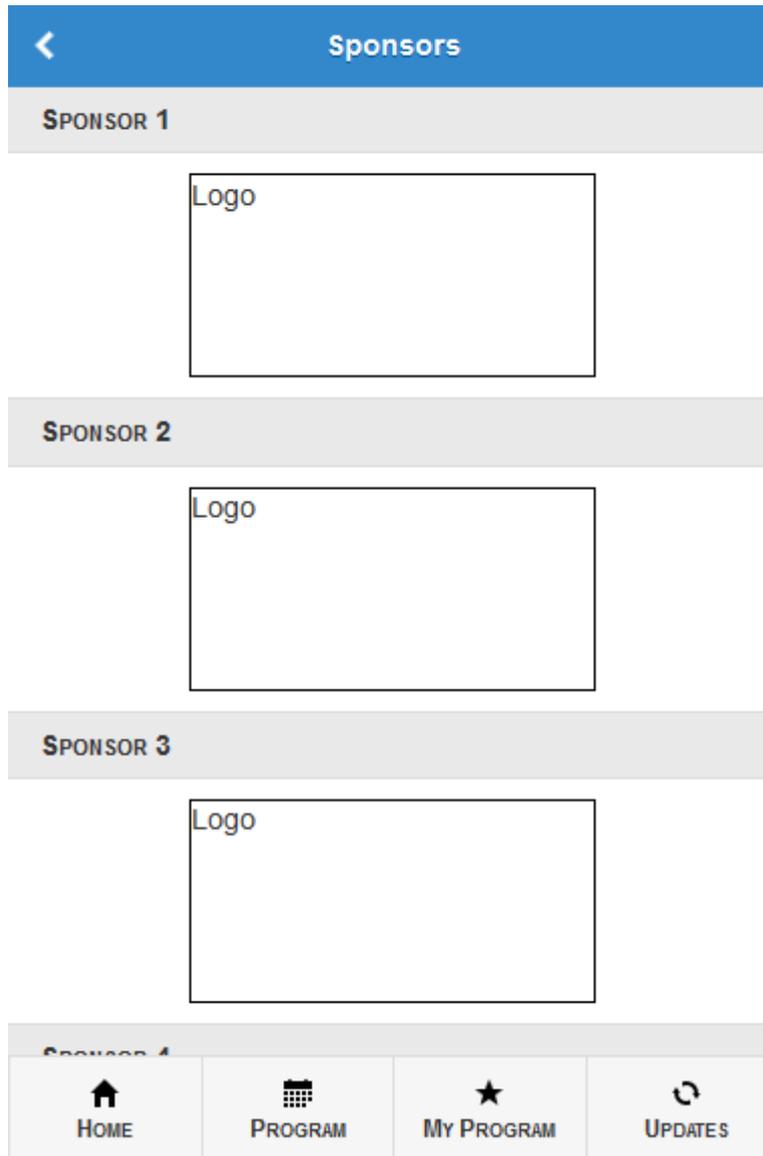
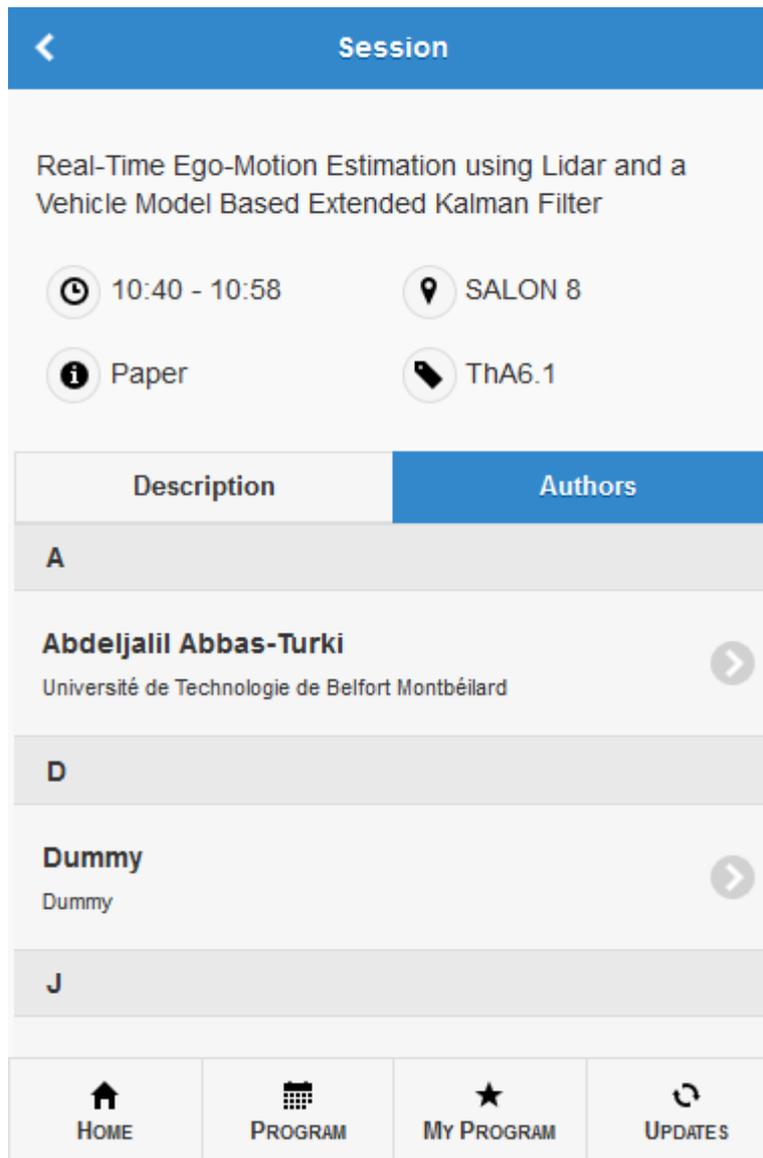


Ilustración 5-11 - Captura de pantalla del listado de patrocinadores

5.2.12 Session detail



Ilustraci3n 5-12 - Captura de pantalla de la vista de detalle de una sesi3n

5.2.13 Author detail

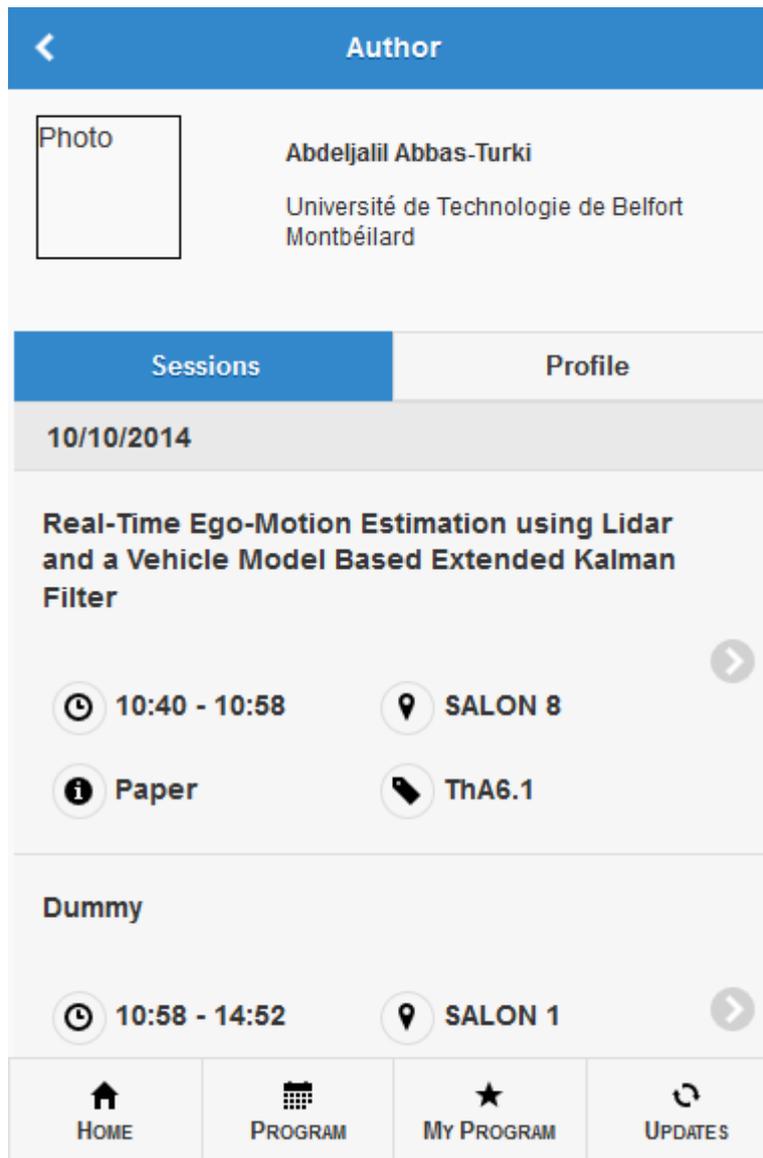


Ilustración 5-13 - Captura de pantalla de la vista de detalle de un autor

5.2.14 Workshop detail

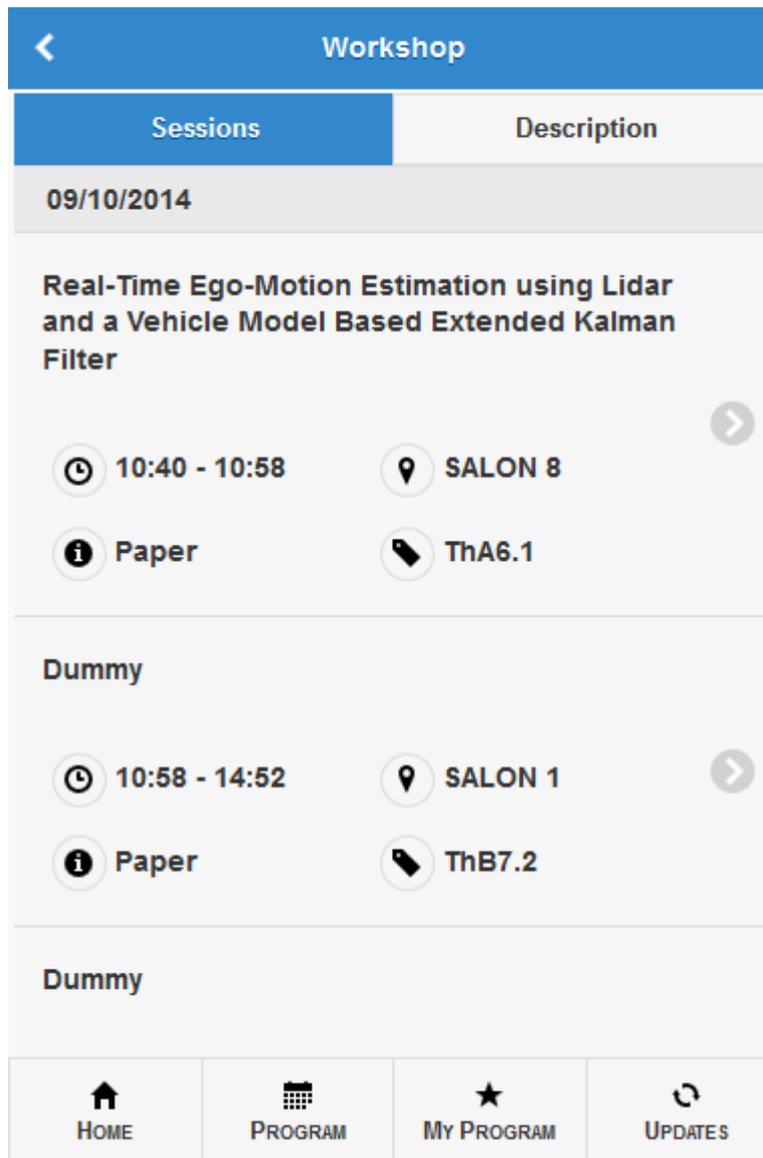


Ilustración 5-14 - Captura de pantalla de la vista de detalle de una temática

5.2.15 Keyword detail

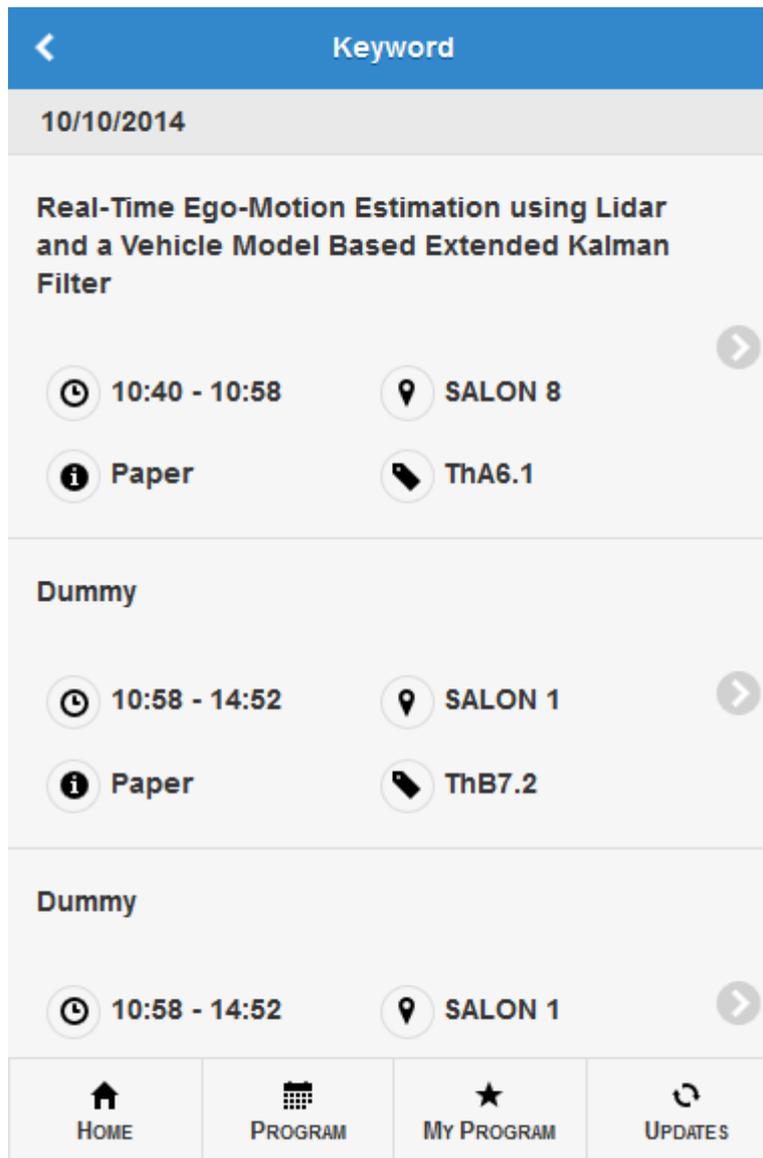


Ilustración 5-15 - Captura de pantalla de la vista de detalle de una palabra clave

5.2.16 New detail

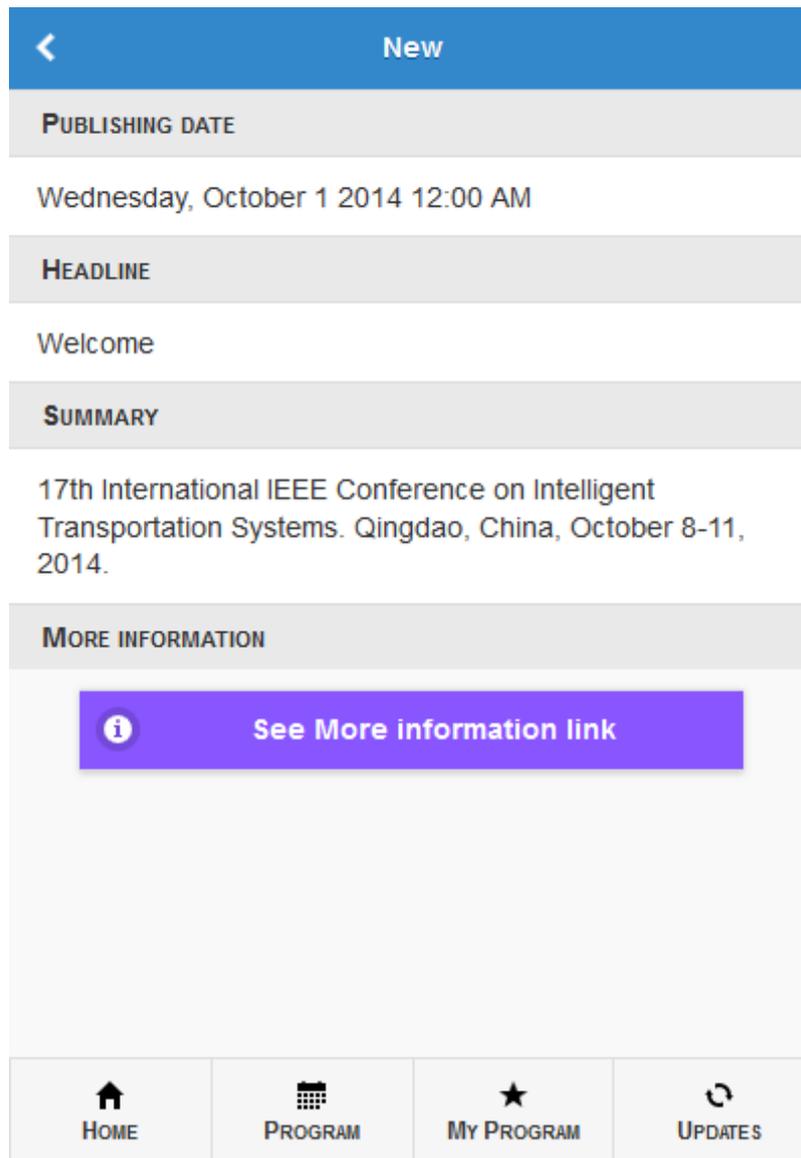


Ilustración 5-16 - Captura de pantalla de la vista de detalle de una noticia

5.2.17 Social Activity detail

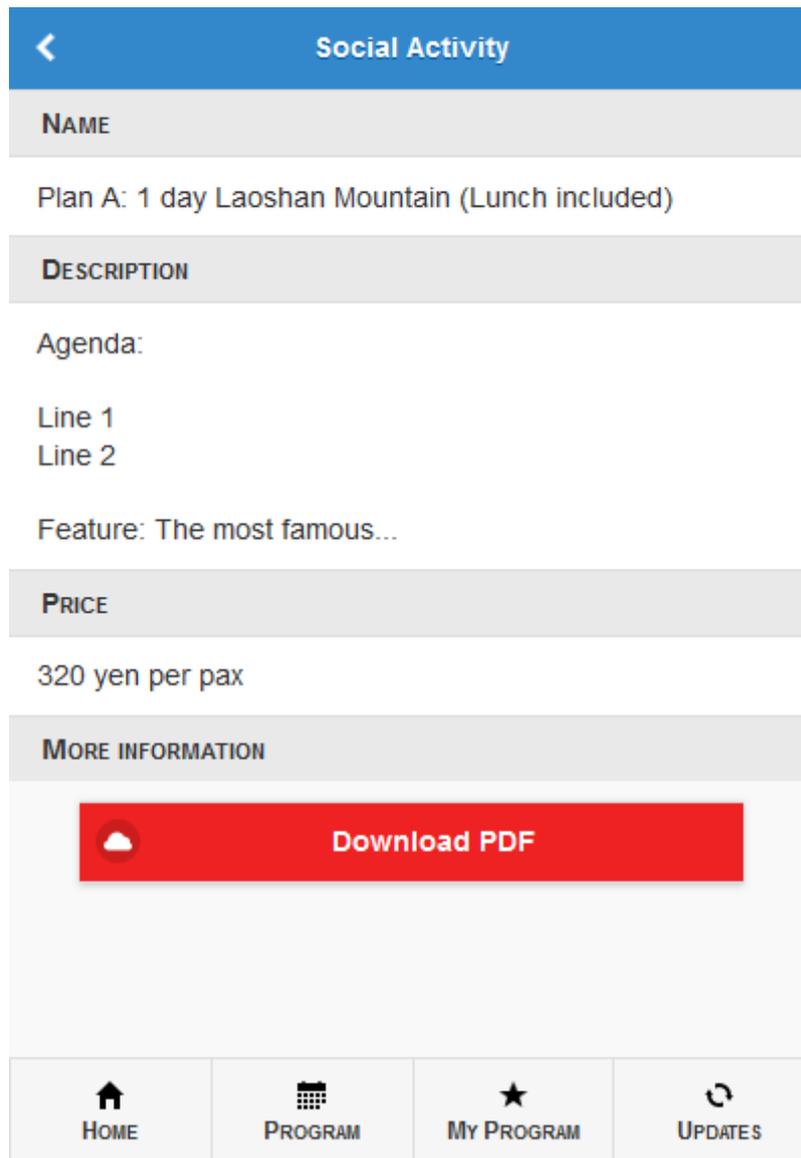


Ilustración 5-17 - Captura de pantalla de la vista de detalle de una actividad social

5.3 Anexo III – Diagrama de flujo de la aplicación

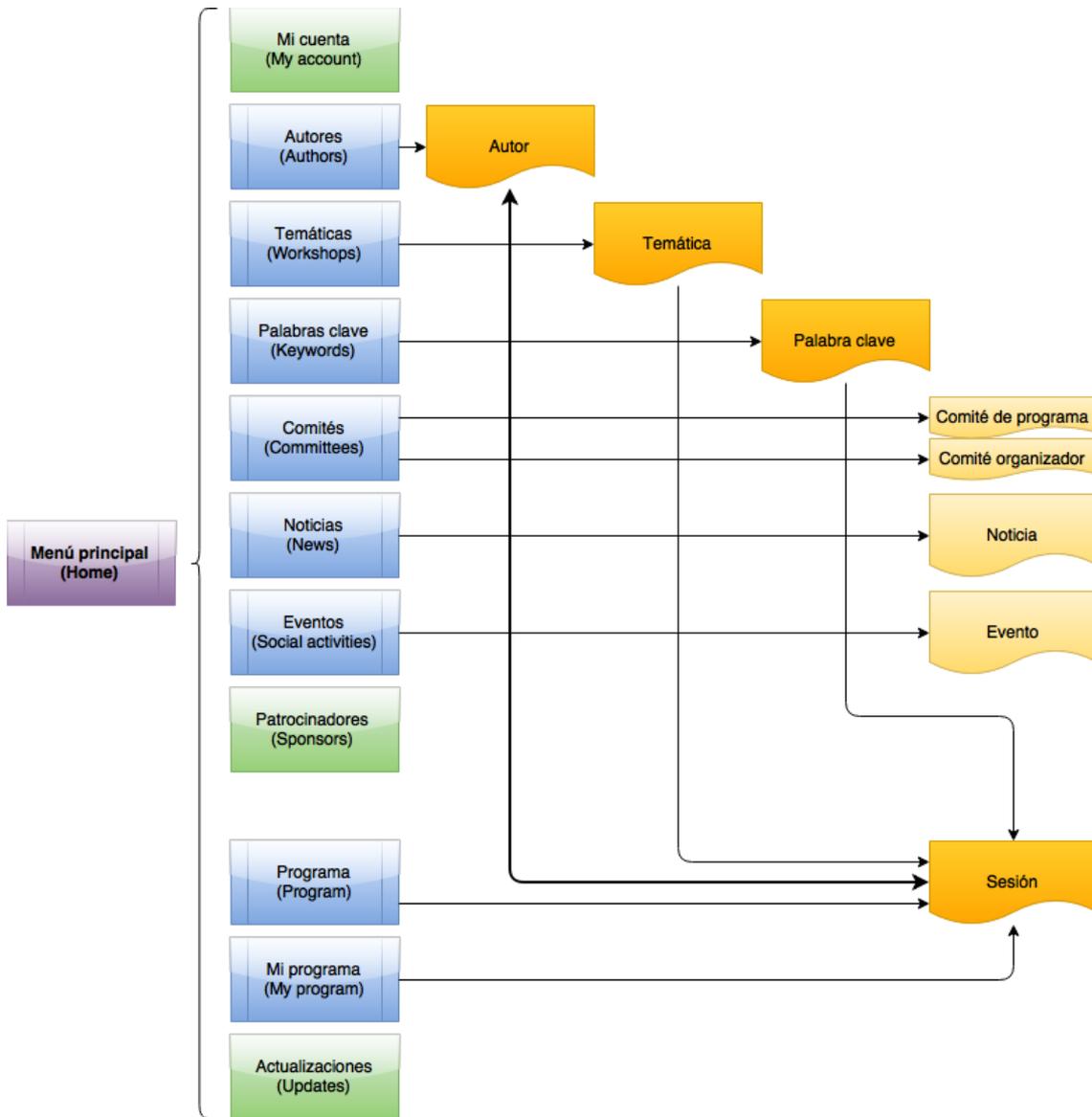


Ilustración 5-18 - Diagrama de flujo de la aplicación

5.4 Anexo IV – Manual de usuario

Debido a que no poseemos una licencia de desarrollador de Apple, la cual no es gratuita y cuesta alrededor de \$99 dólares al año, no podemos ofrecer un ejecutable *.ipa* de la aplicación por el momento.

Si se quisiera instalar la aplicación tendrían que abrir el proyecto entregado en el CD (sección de código fuente) con el IDE Xcode 8.3, pudiendo luego transferirla a su dispositivo real o emulador.

Para la realización de este manual de usuario se ha utilizado un dispositivo móvil Android Motorola Moto G (2ª generación). El uso de la aplicación es exactamente igual para Android e iOS.

5.4.1 Navegación hacia atrás

Antes de empezar, nos gustaría recordar que en toda la aplicación existe un historial de navegación. Por tanto, se puede acceder a la pantalla anterior pulsando el botón Atrás desde cualquier página que lo posea.



Ilustración 5-19 - Botón Atrás

Este botón se encuentra en la sección superior izquierda de la aplicación.

5.4.2 Menú principal

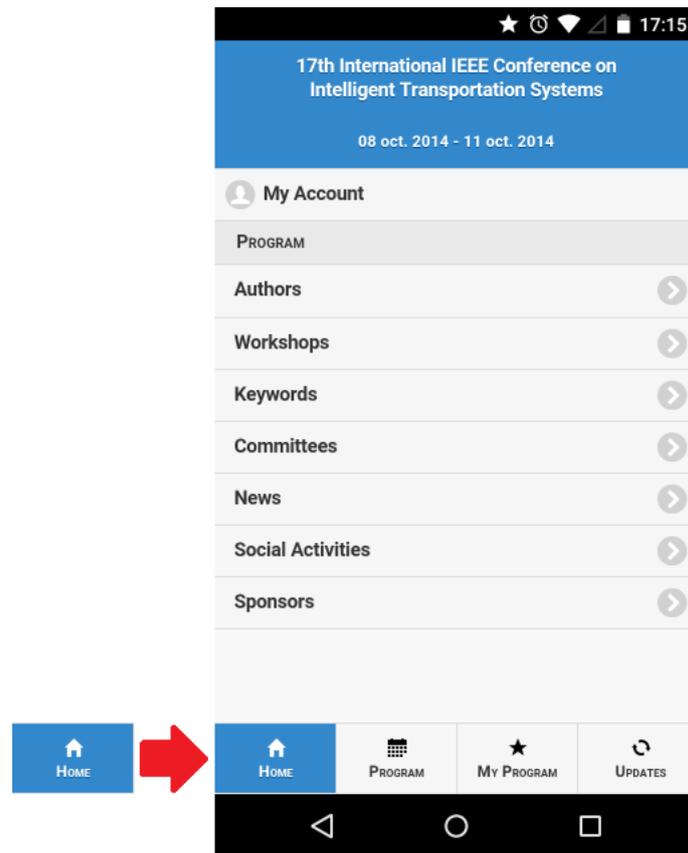


Ilustración 5-20 - Menú principal

Esta es la primera vista mostrada al arrancar la aplicación, y podrá ser accedida desde el botón *Home* en cualquier punto de la misma.

Desde aquí podremos ir al resto de vistas de la aplicación. Para ello, bastará con pulsar en la sección a la que queremos ir.

5.4.3 Programa

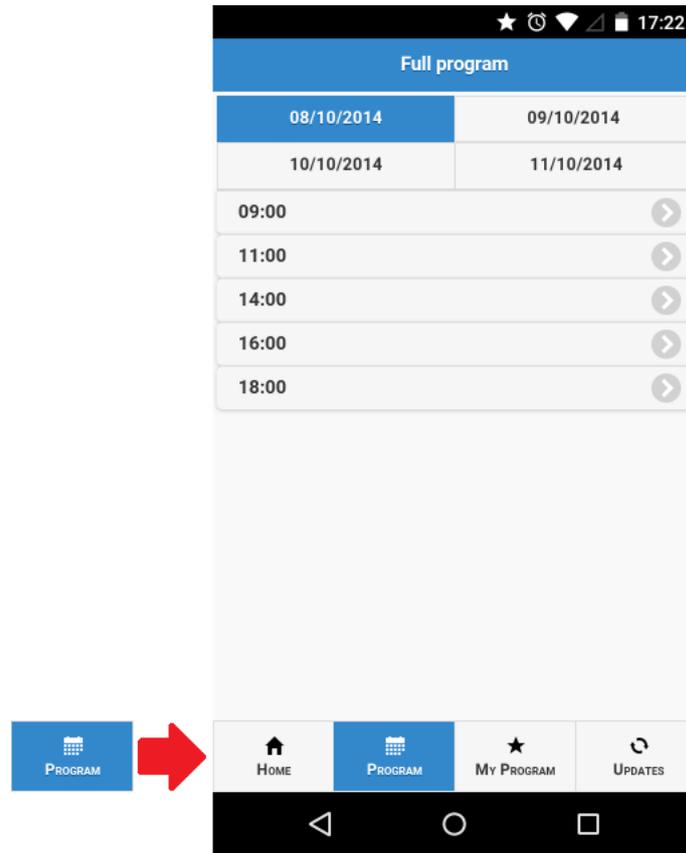


Ilustración 5-21 - Programa

En esta vista podemos ver todas las charlas del congreso.

Las sesiones del congreso están divididas en pestañas por días. Para cambiar de día basta con pulsar en la pestaña seleccionada.

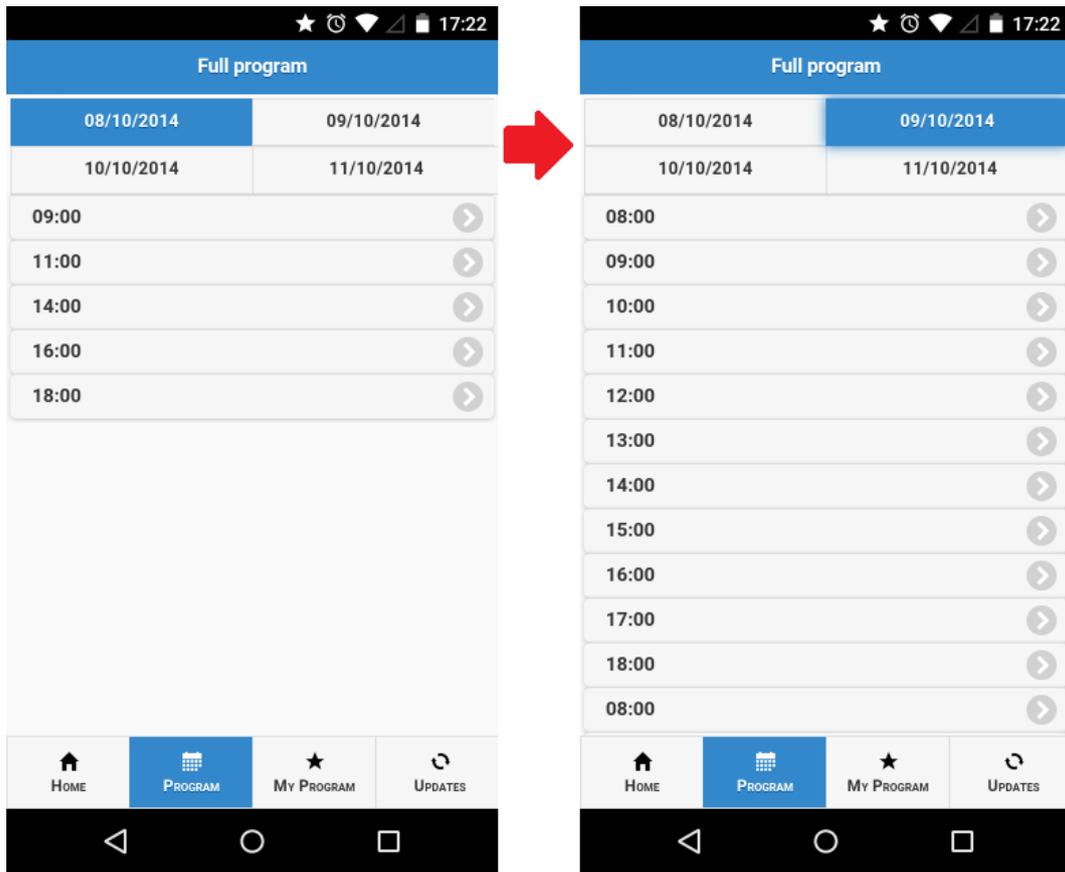


Ilustración 5-22 - Días del programa

Al principio sólo se mostrarán las horas en las que hay charlas ese día. Para ver las charlas debemos pulsar en la franja horaria deseada.

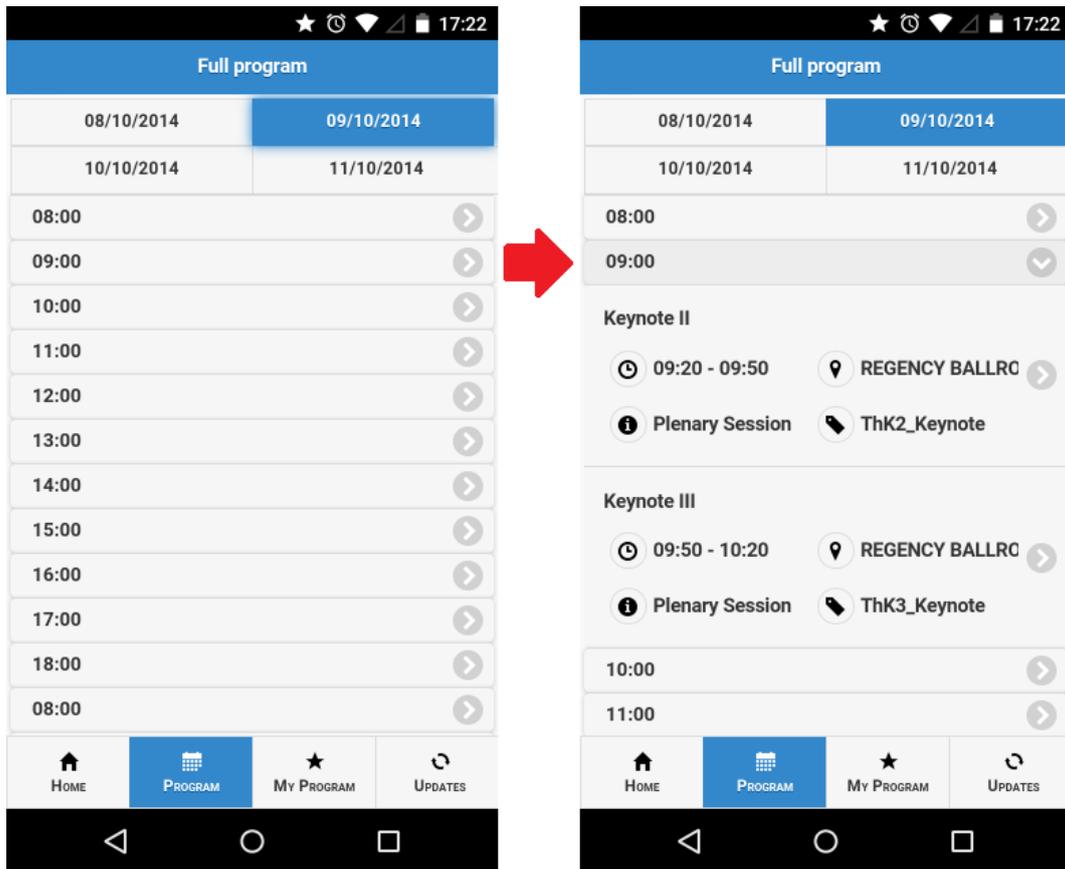


Ilustración 5-23 - Despliegue de horas de un día del programa

5.4.4 Mi programa

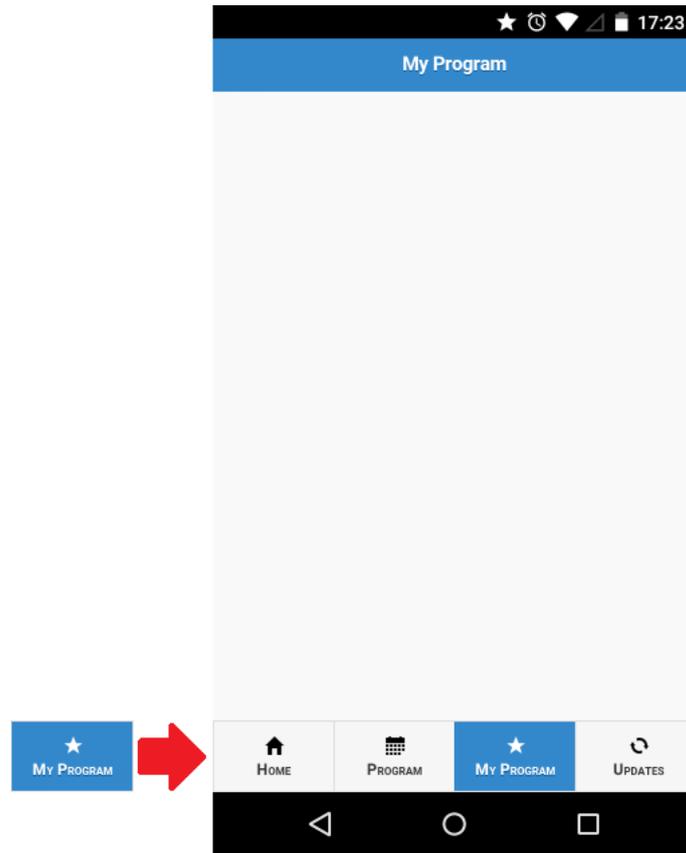


Ilustración 5-24 - Mi programa

En esta vista podremos ver un listado personalizado de las charlas, para generar este listado basta con agregar una charla desde su vista de detalle. **Esta sección no es funcional en esta versión.**

5.4.5 Detalle de una charla

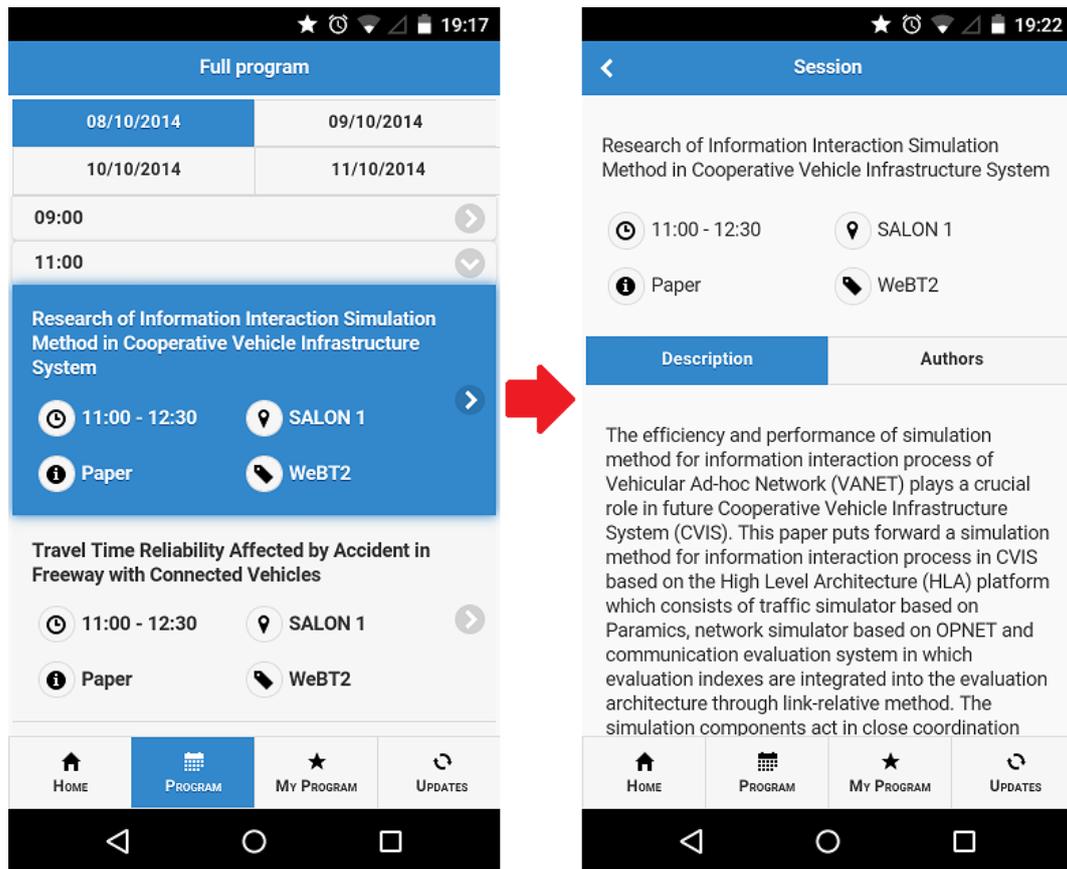


Ilustración 5-25 - Detalle de una charla

En esta vista podemos ver los datos en detalle de una charla. Desde esta vista se podrán añadir las charlas a Mi programa **(no implementado en esta versión)**.

Esta vista tiene dos secciones entre las que podemos alternar pulsando en su nombre.

- **Description:** Aquí se muestra una descripción de la charla.

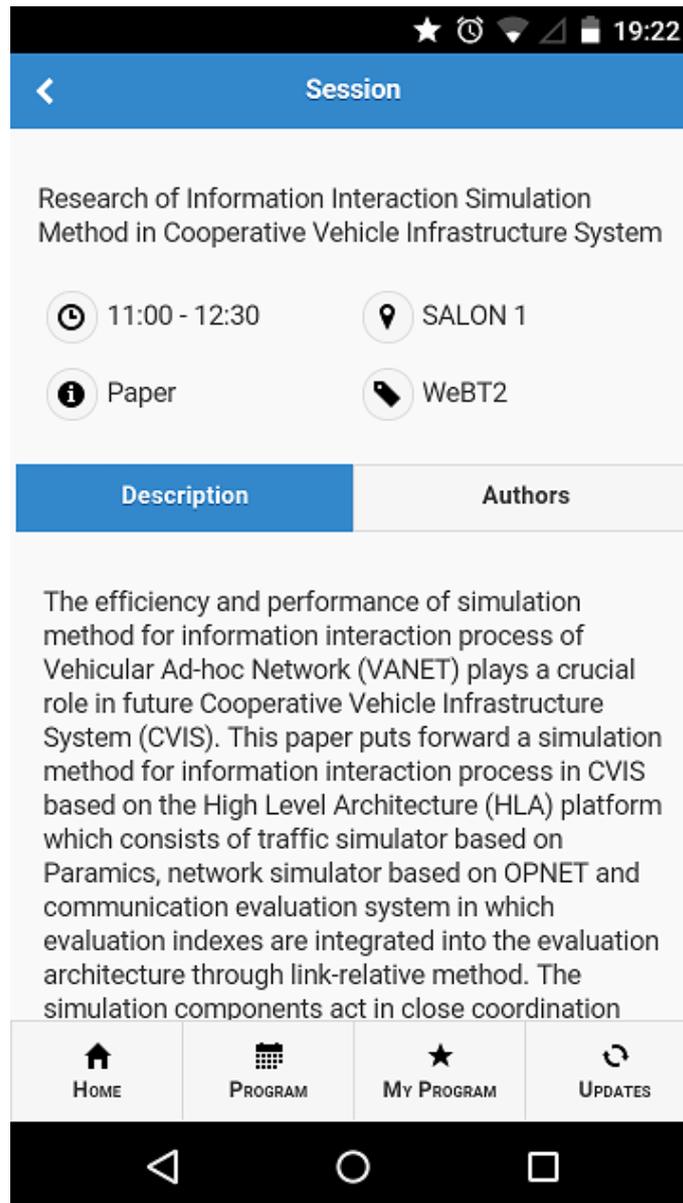


Ilustración 5-26 - Descripción de la charla

- **Authors:** Aquí se muestra un listado de los autores que participan en la charla, pudiendo consultarlos si pulsamos en ellos.

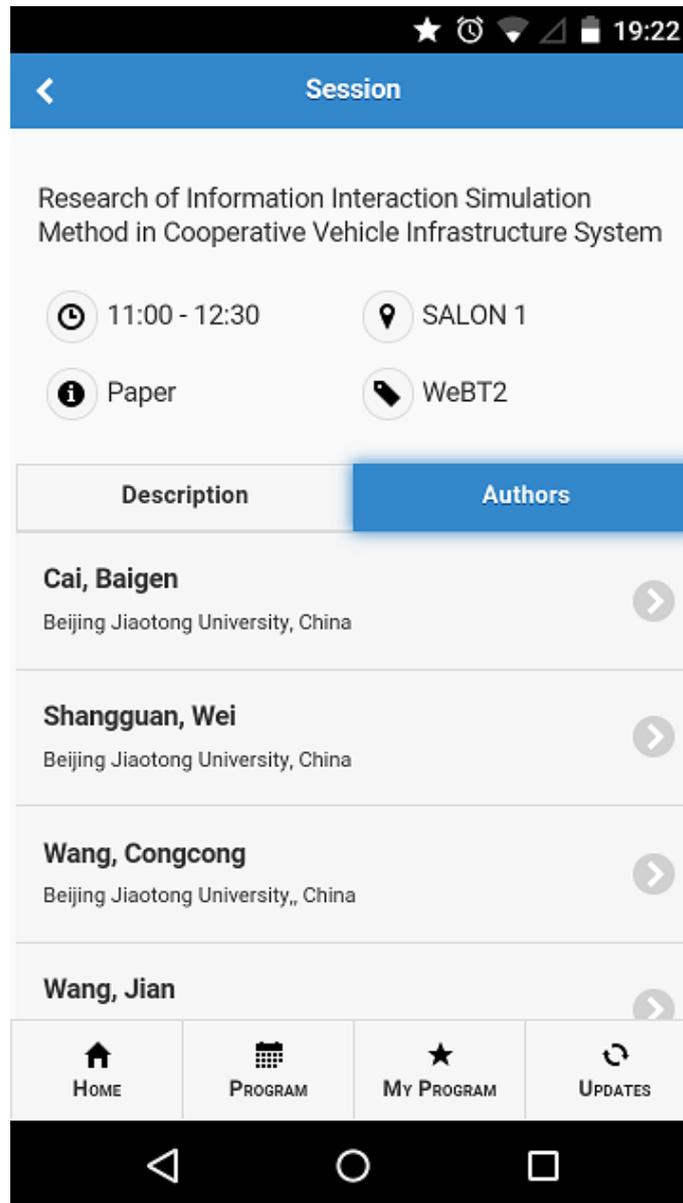


Ilustración 5-27 - Autores de una charla

5.4.6 Actualización

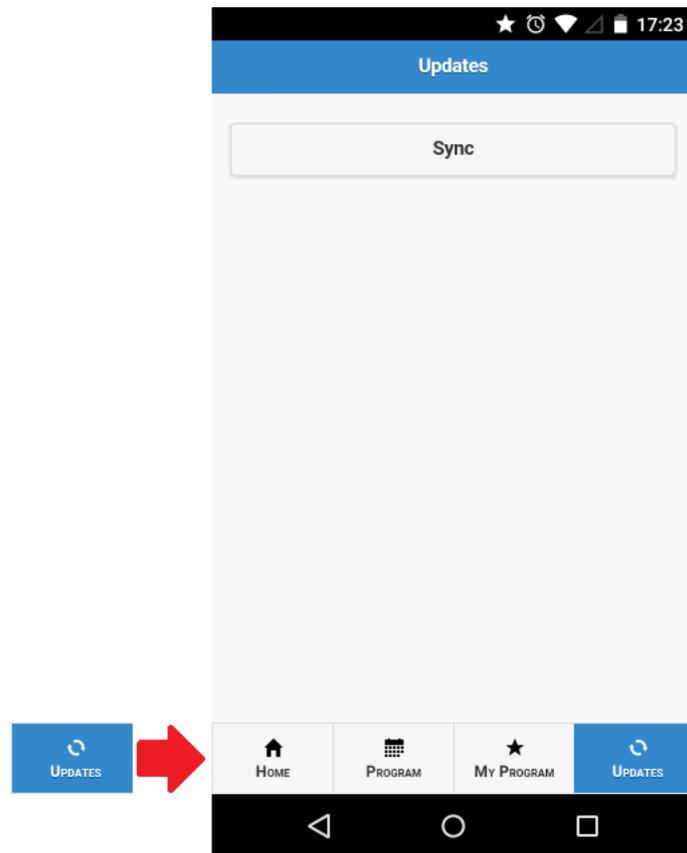


Ilustración 5-28 - Actualización

Desde esta vista se podrán actualizar los datos de la aplicación, para ello pulsaremos en el botón *Sync*.

5.4.7 Listado de autores

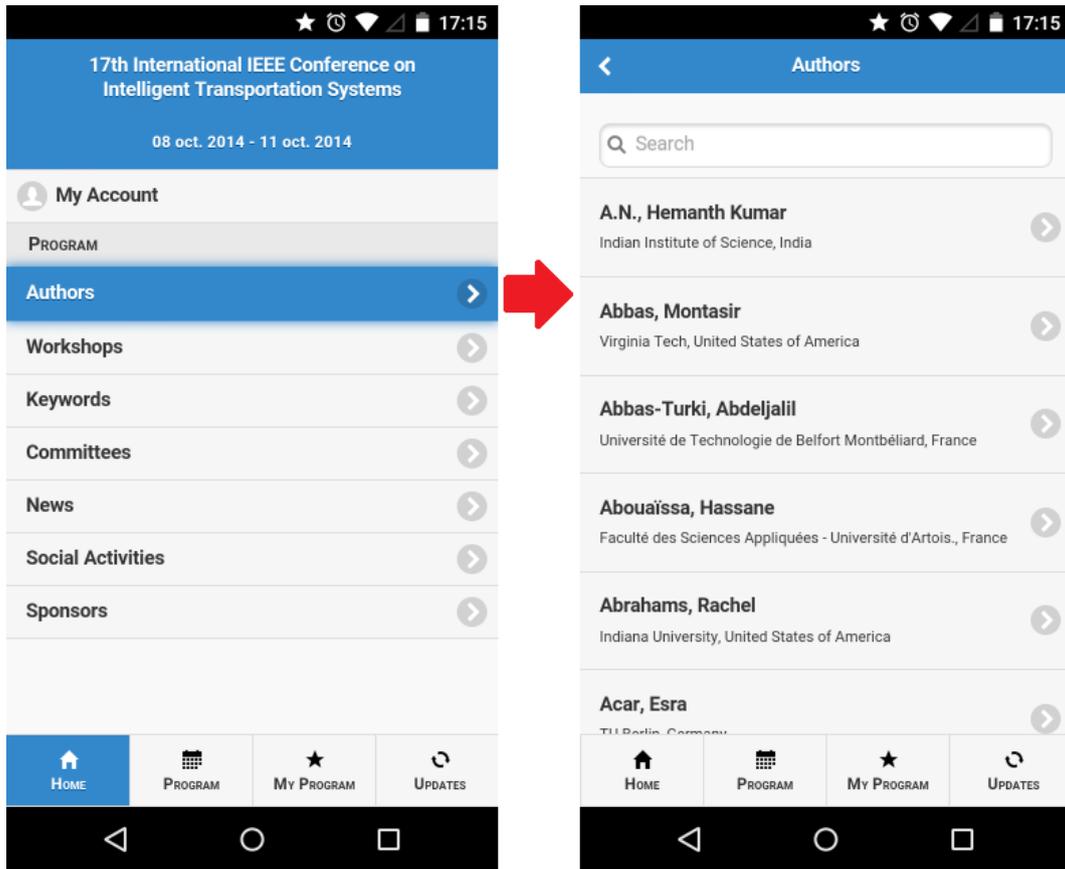


Ilustración 5-29 - Listado de autores

Esta vista nos proporciona un listado de los autores participantes en el congreso, podemos desplazarnos entre ellos pulsando y arrastrando hacia arriba o abajo. Desde esta vista podemos ir a la vista de detalle de un autor. Para ello, debemos pulsar en el autor deseado.

También podemos filtrar el resultado pulsando en la barra de búsqueda y escribiendo en ella los valores a buscar.

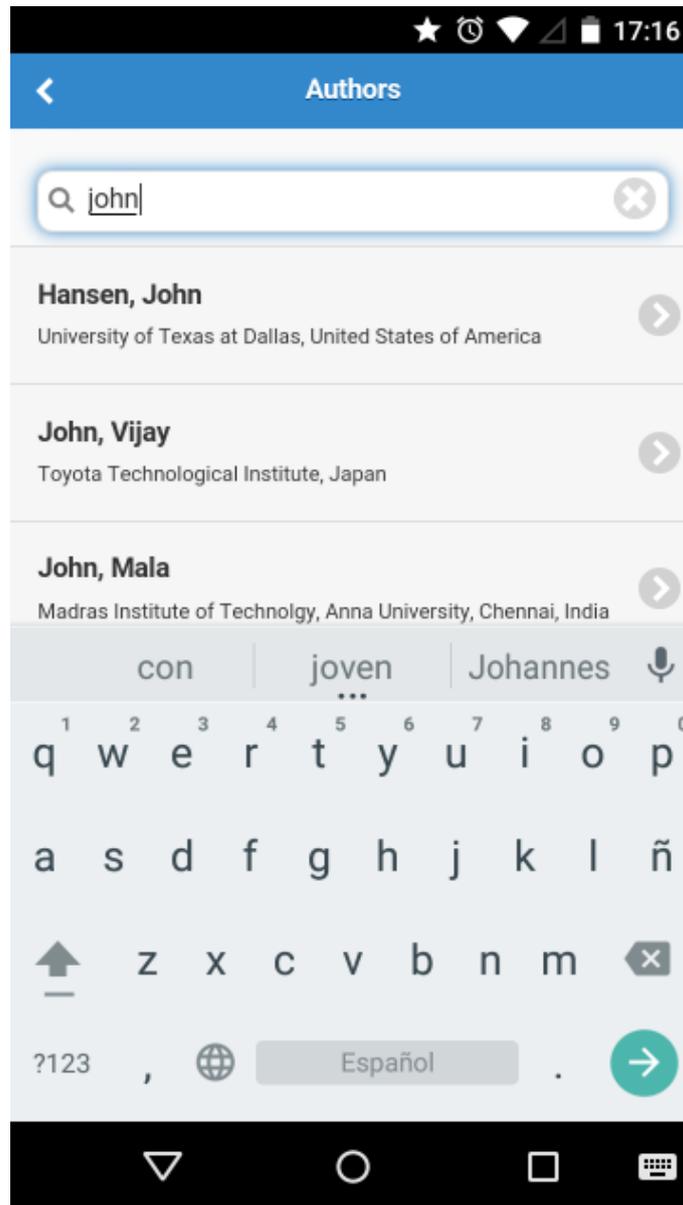


Ilustración 5-30 - Filtrado de autores

5.4.8 Detalle de un autor

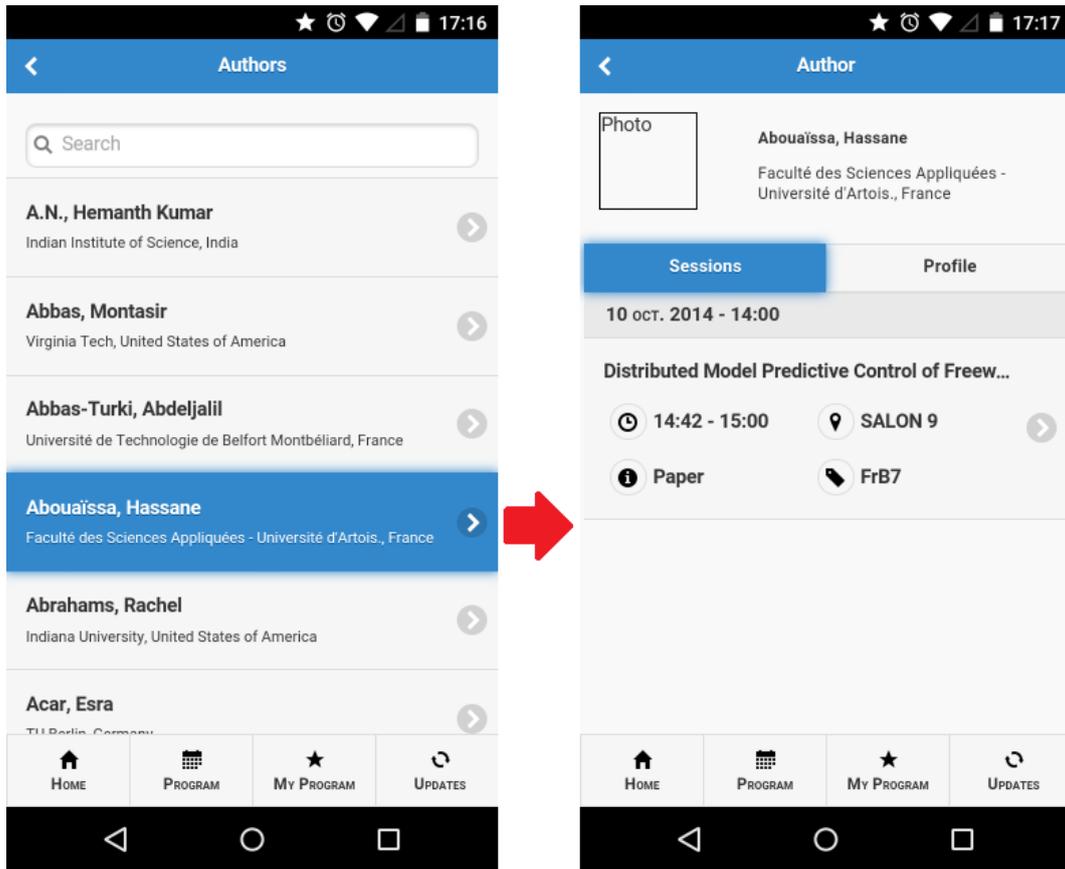


Ilustración 5-31 - Detalle de un autor

En esta vista podemos ver los datos en detalle de un autor. Esta vista tiene dos secciones entre las que podemos alternar pulsando en su nombre.

- **Profile:** Aquí se muestra una descripción proveída por el autor.

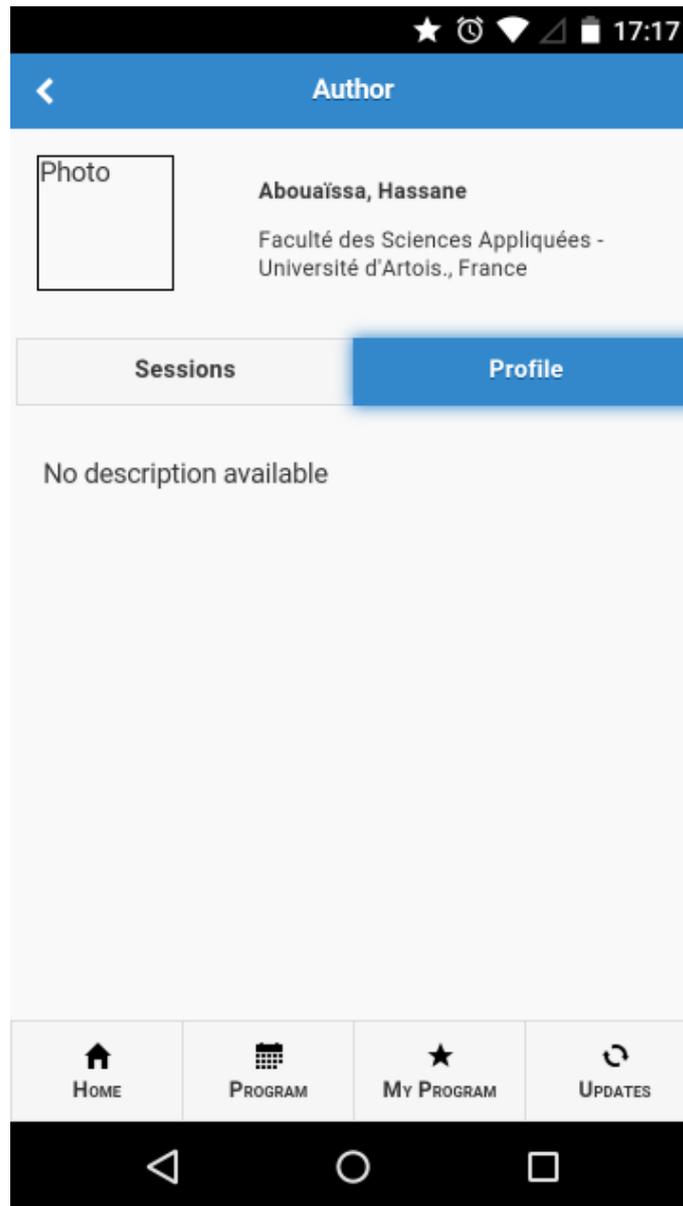


Ilustración 5-32 - Perfil de un autor

- **Sessions:** Aquí se muestra el listado de las charlas en las que participa el autor, pudiendo consultarlas si pulsamos en ellas.

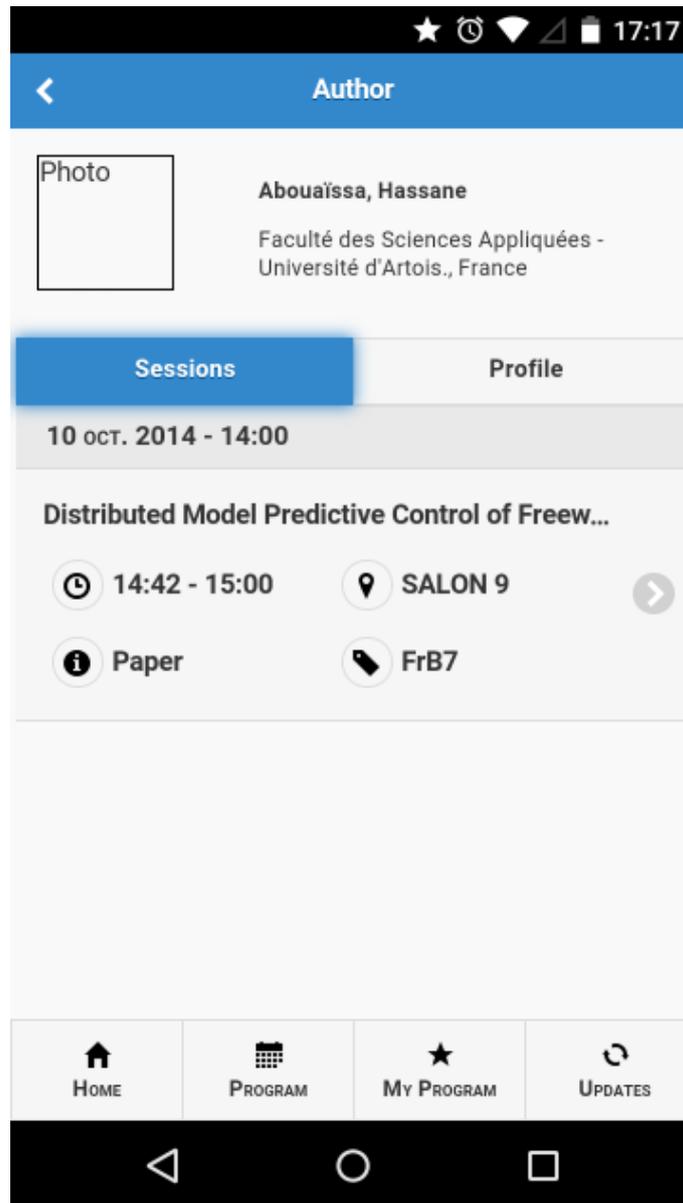


Ilustración 5-33 - Sesiones de un autor

5.4.9 Listado de temáticas

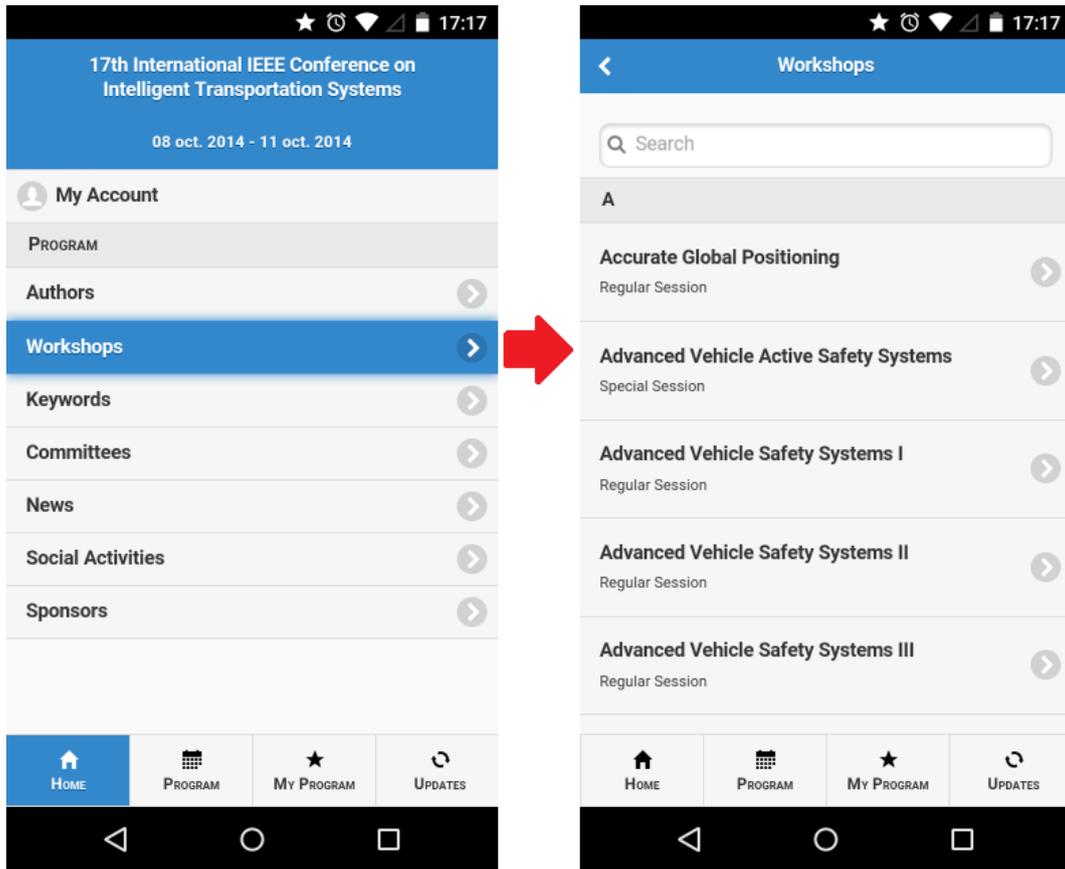


Ilustración 5-34 - Listado de temáticas

Esta vista nos proporciona un listado de las temáticas a tratar durante el congreso, podemos desplazarnos entre ellas pulsando y arrastrando hacia arriba o abajo. Desde esta vista podemos ir a la vista de detalle de una temática. Para ello debemos pulsar en la temática deseada.

También podemos filtrar el resultado pulsando en la barra de búsqueda y escribiendo en ella los valores a buscar.

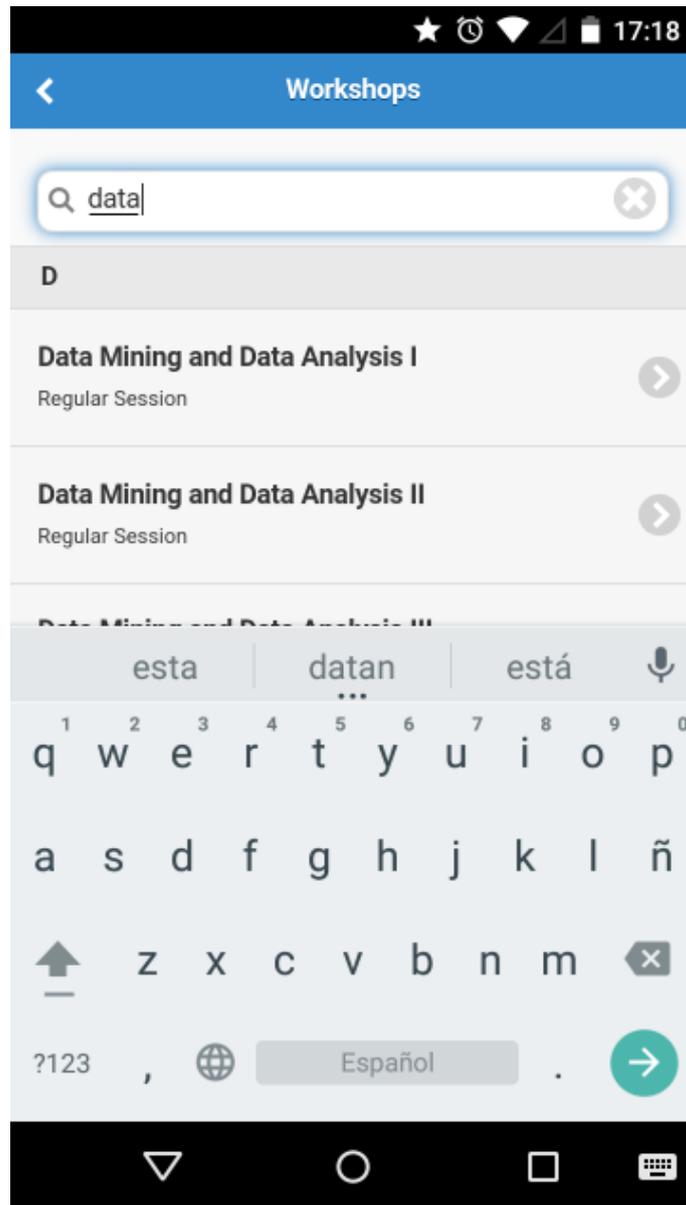


Ilustración 5-35 - Filtrado de temáticas

5.4.10 Detalle de una temática

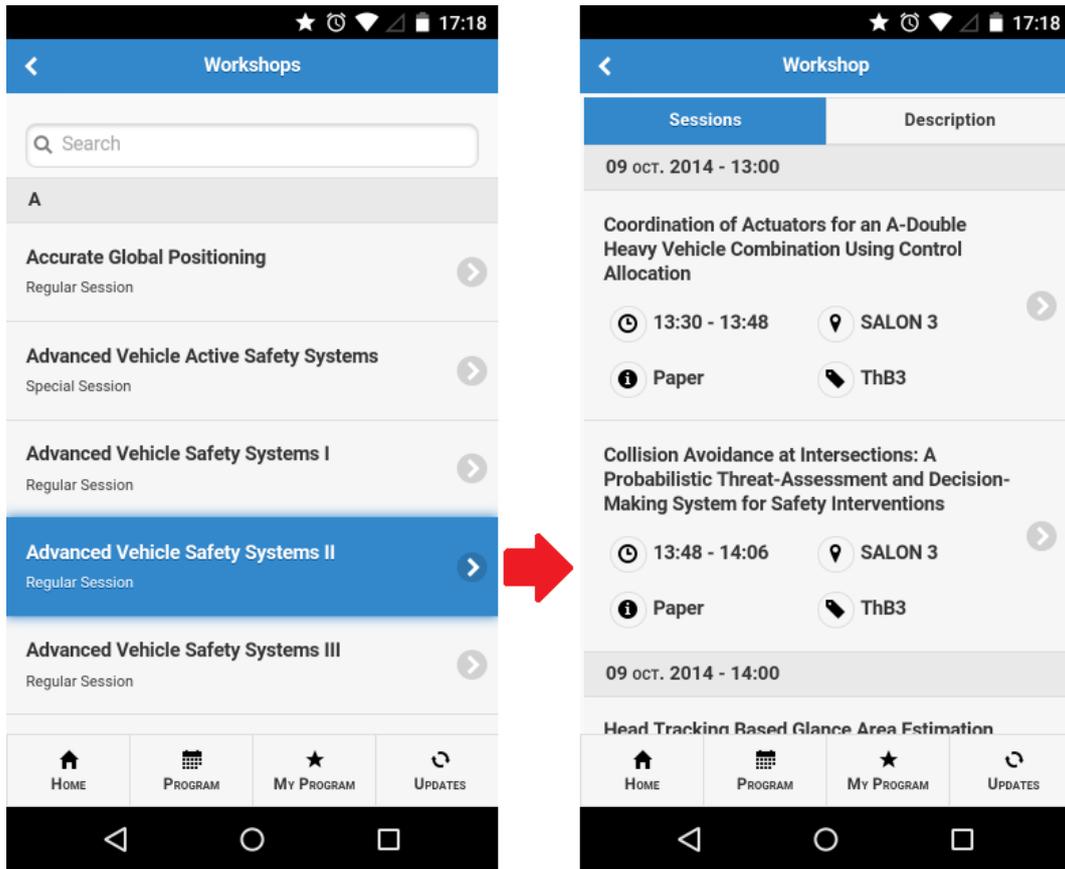


Ilustración 5-36 - Detalle de una temática

En esta vista podemos ver los datos en detalle de una temática. Esta vista tiene dos secciones entre las que podemos alternar pulsando en su nombre.

- **Description:** Aquí se muestra una descripción de la temática.

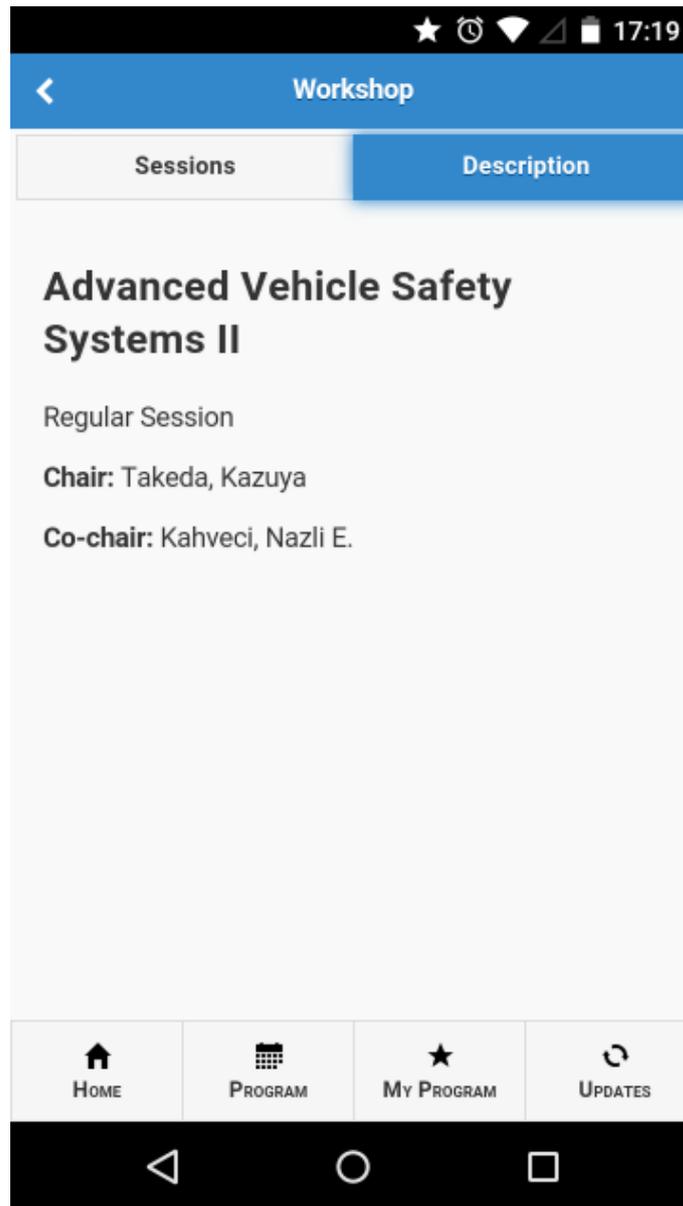


Ilustración 5-37 - Descripción de una temática

- **Sessions:** Aquí se muestra el listado de las charlas en las que se trata la temática seleccionada, pudiendo consultarlas pulsando en ellas.

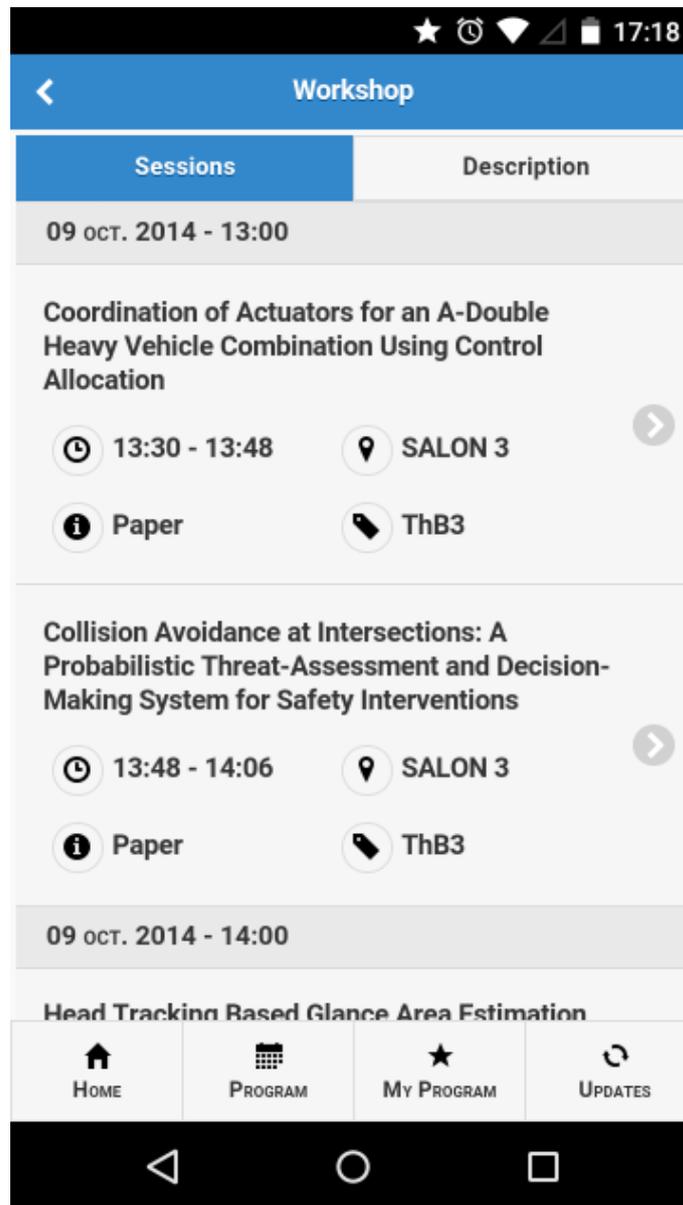


Ilustración 5-38 - Sesiones de una temática

5.4.11 Listado de palabras clave

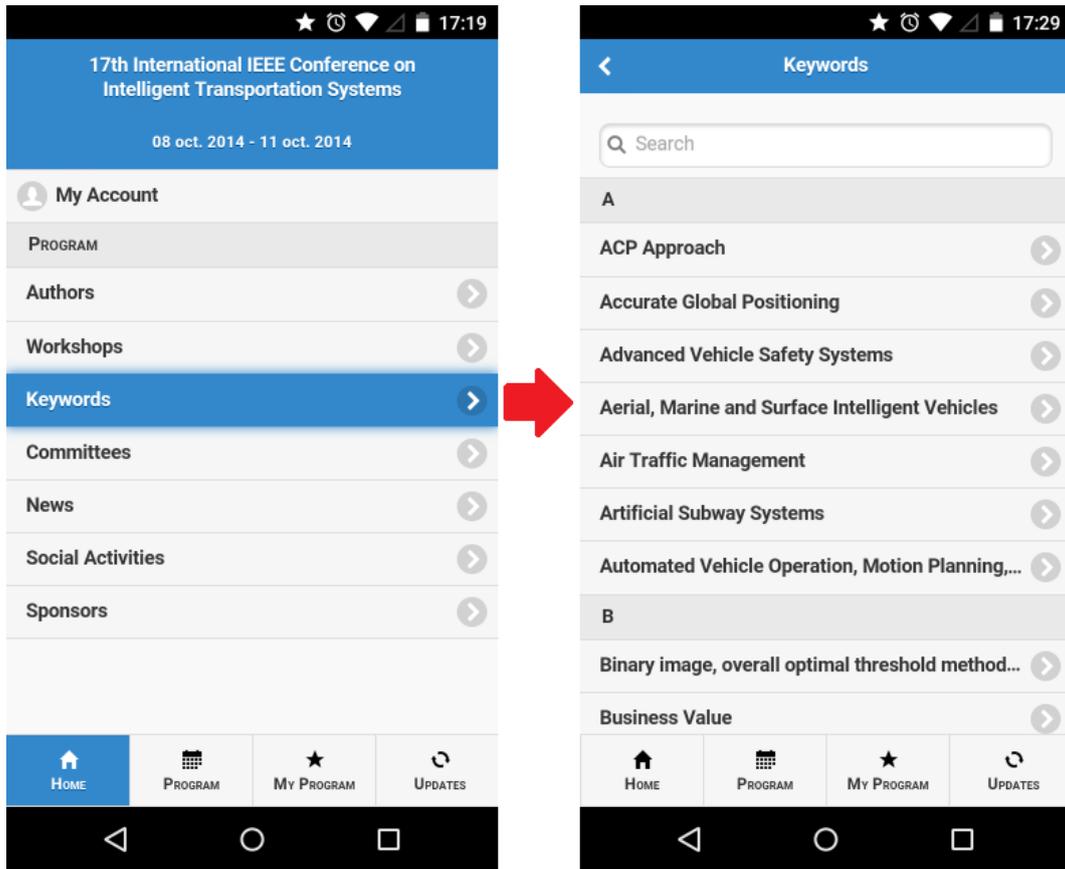


Ilustración 5-39 - Listado de palabras clave

Esta vista nos proporciona un listado de palabras clave relacionadas a conceptos a tratar durante el congreso, podemos desplazarnos entre ellas pulsando y arrastrando hacia arriba o abajo. Desde esta vista podemos ir a la vista de detalle de una palabra clave. Para ello debemos pulsar en la palabra clave deseada.

También podemos filtrar el resultado pulsando en la barra de búsqueda y escribiendo en ella los valores a buscar.

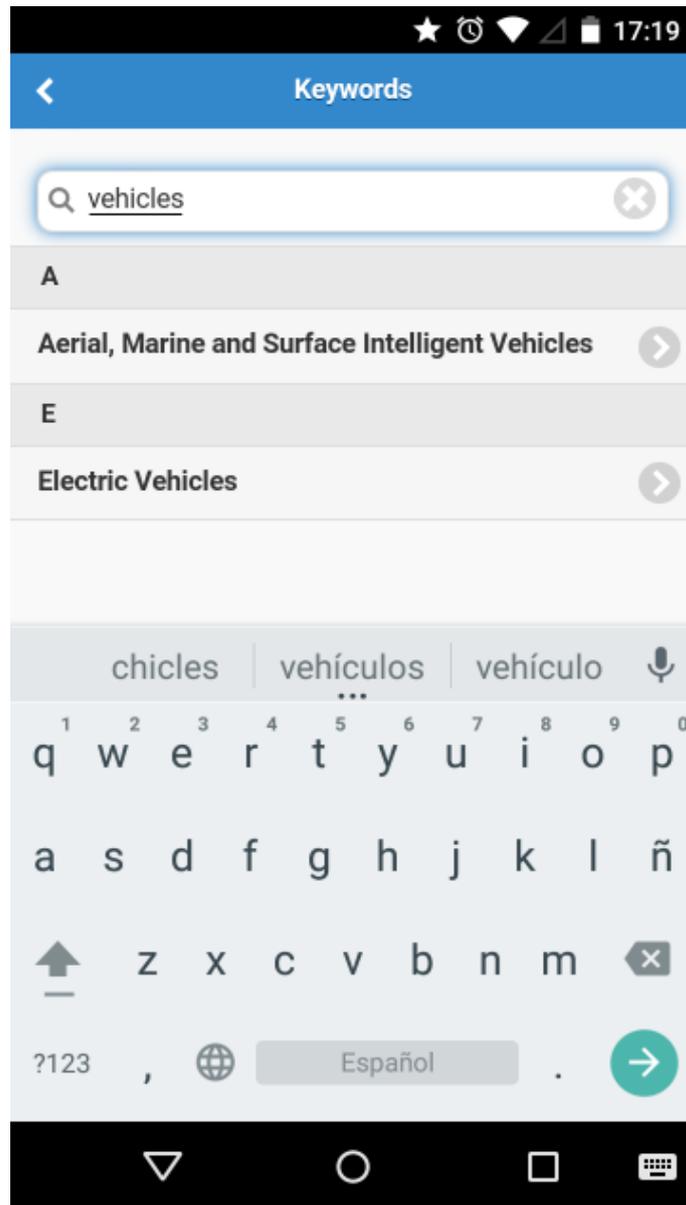


Ilustración 5-40 - Filtrado de palabras clave

5.4.12 Detalle de una palabra clave

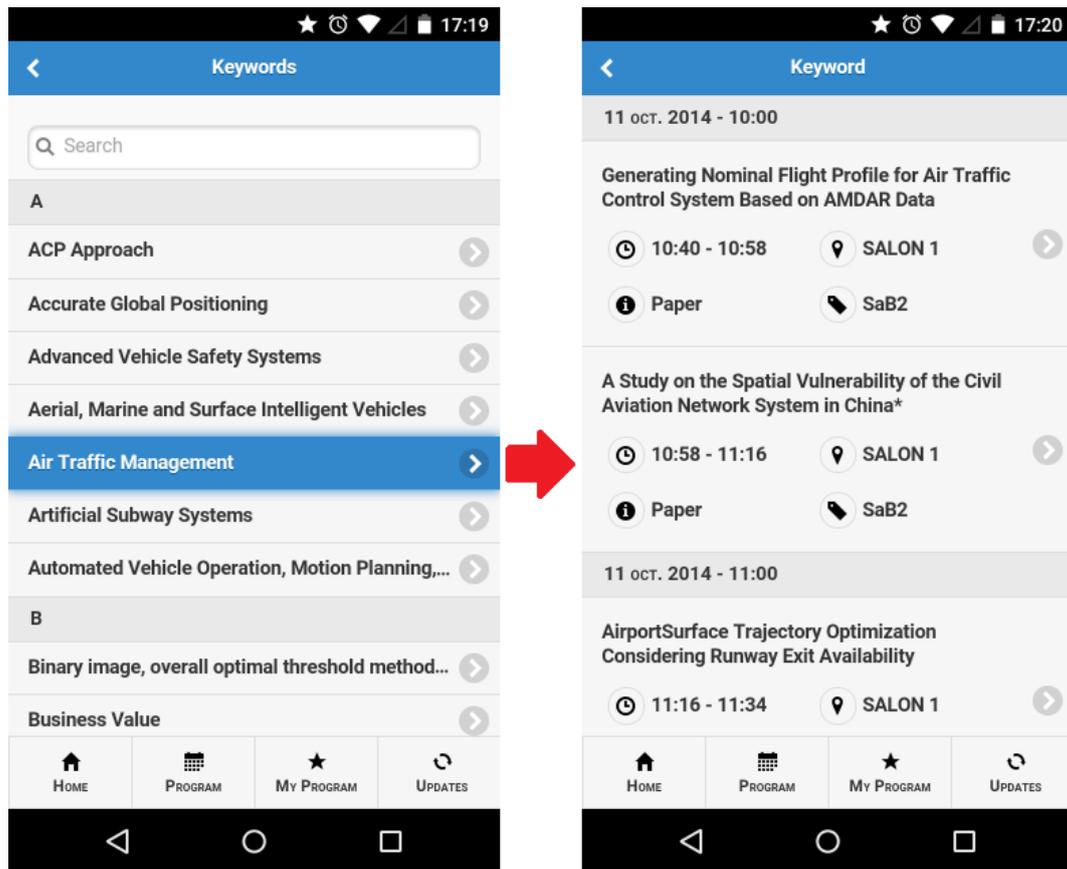
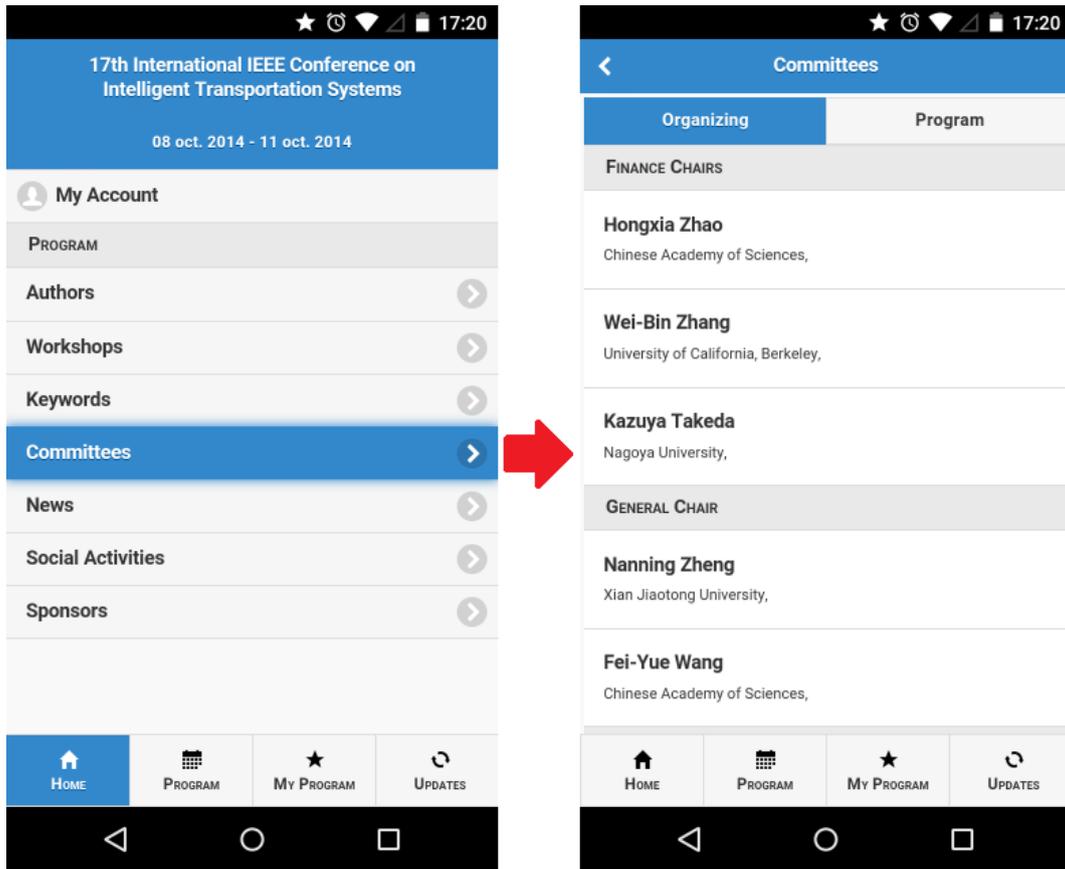


Ilustración 5-41 - Detalle de una palabra clave

En esta vista podemos ver un listado de las charlas relacionadas con la palabra clave seleccionada, pudiendo consultarlas pulsando en ellas.

5.4.13 Listado de comités



Esta vista nos proporciona un listado de los comités del congreso, podemos desplazarnos entre ellos pulsando y arrastrando hacia arriba o abajo.

Esta vista tiene dos secciones entre las que podemos alternar pulsando en su nombre.

- **Organizing:** Aquí encontramos un listado del comité organizador del evento.

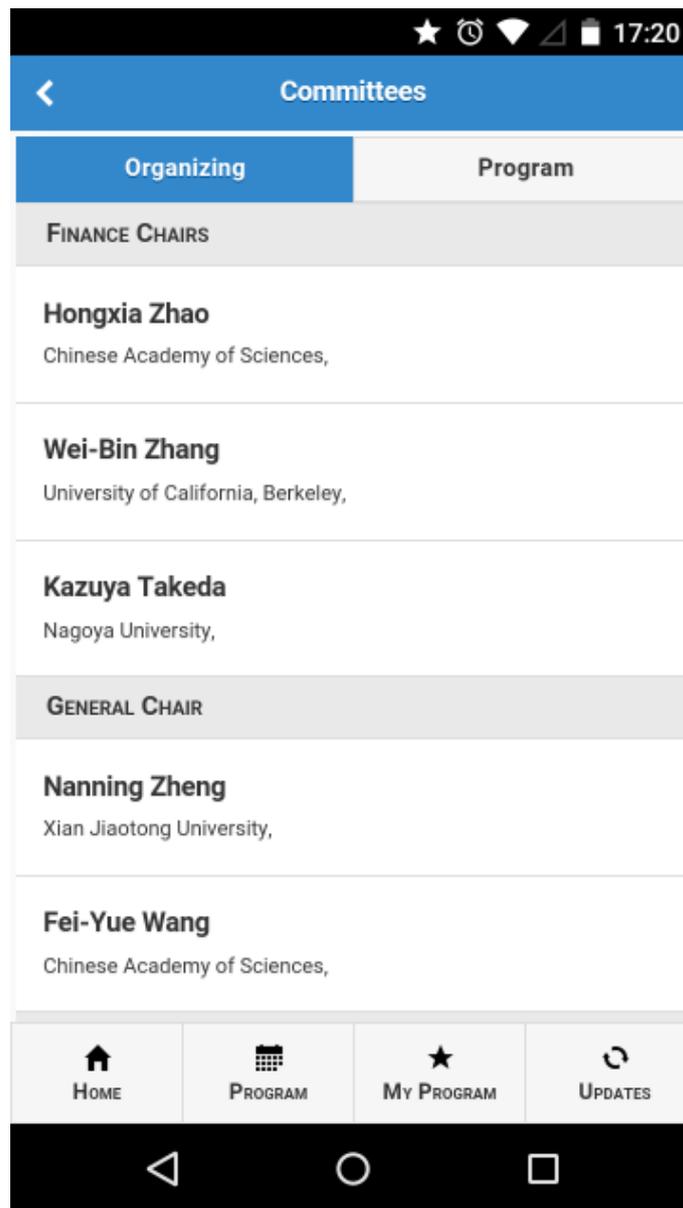


Ilustración 5-43 - Comité organizador

- **Program:** Aquí encontramos un listado del comité del programa.

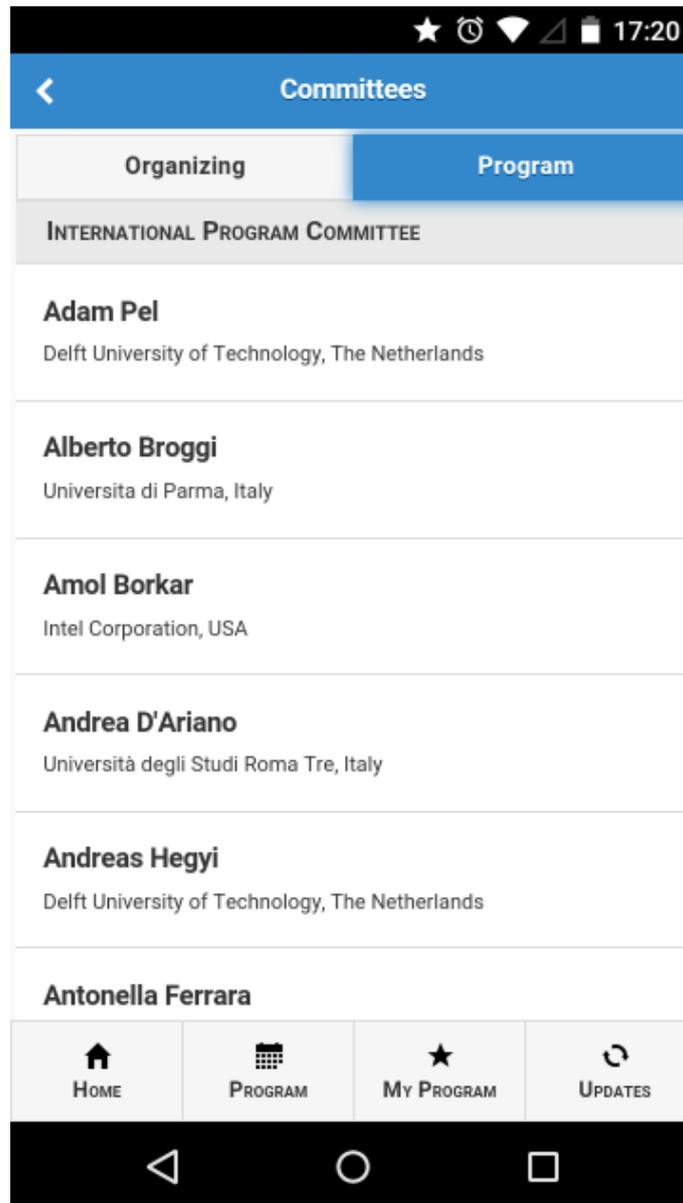


Ilustración 5-44 - Comité del programa

5.4.14 Listado de noticias

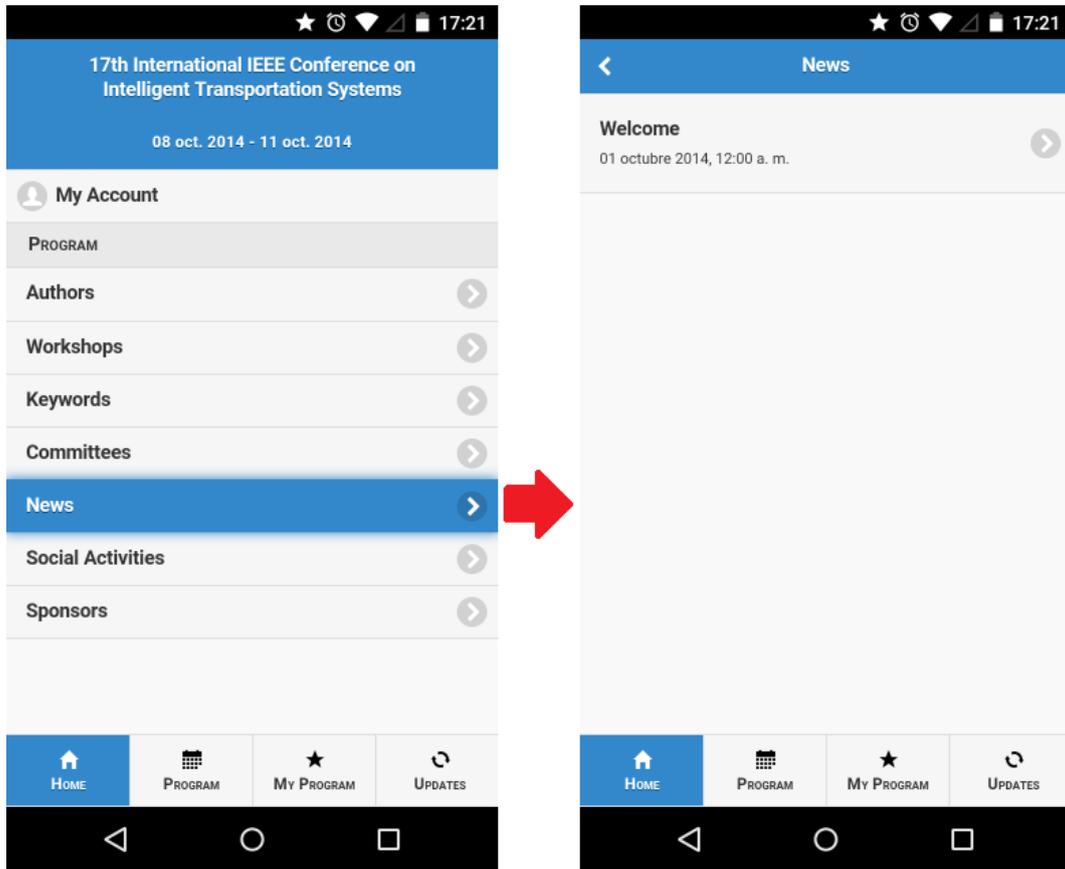


Ilustración 5-45 - Listado de noticias

Esta vista nos proporciona un listado de las noticias publicadas por los organizadores del congreso. Podemos desplazarnos entre ellas pulsando y arrastrando hacia arriba o abajo.

Desde esta vista podemos ir a la vista de detalle de una noticia. Para ello debemos pulsar en la noticia deseada.

5.4.15 Detalle de una noticia

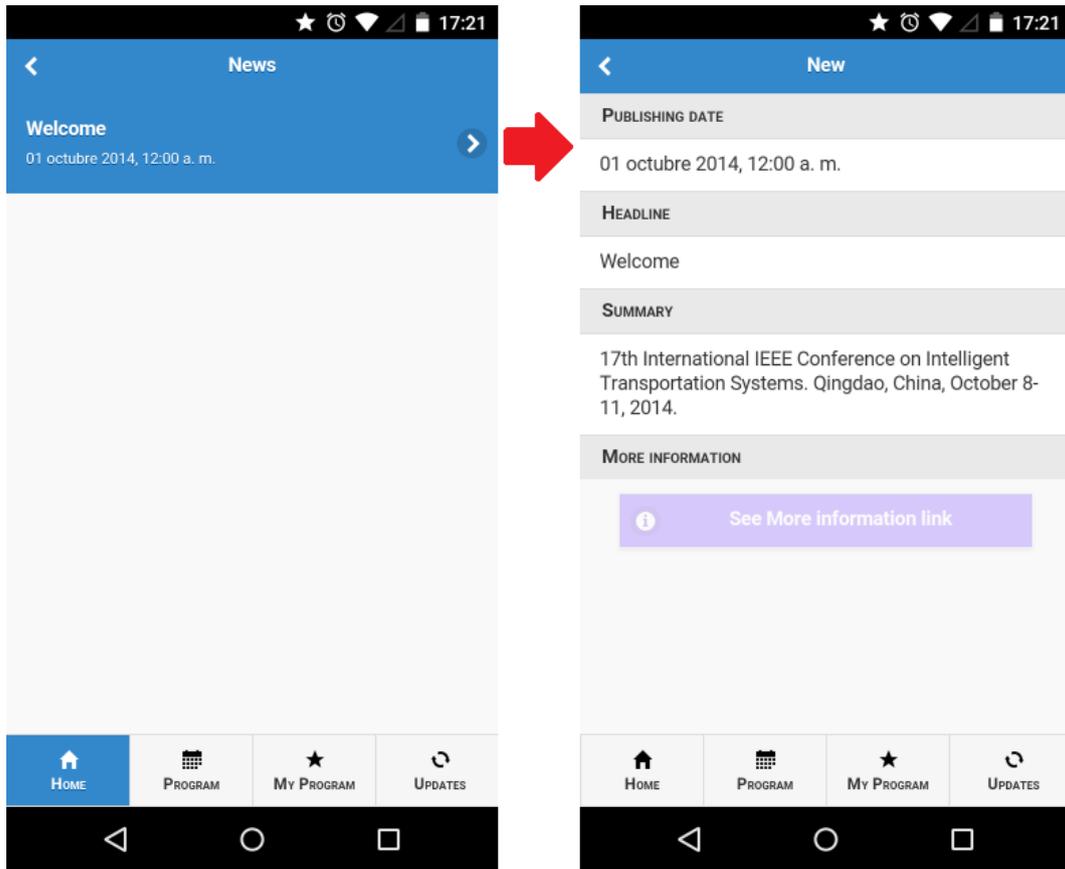


Ilustración 5-46 - Detalle de una noticia

En esta vista podemos ver los datos en detalle de una noticia, así como acceder a datos externos relacionados con la misma si fuera necesario, pudiendo consultarlos pulsando en ellos (**sólo para usuarios registrados, no funcional en esta versión**).

5.4.16 Listado de eventos

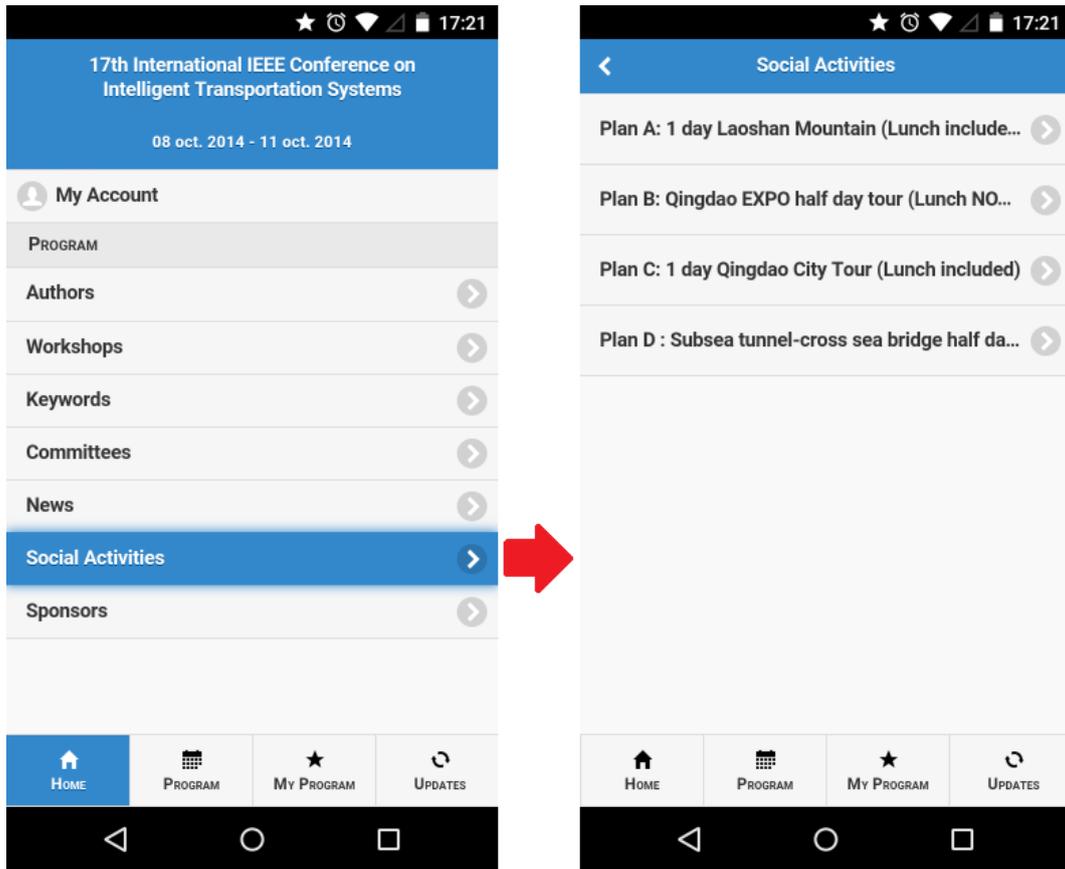


Ilustración 5-47 - Listado de eventos

Esta vista nos proporciona un listado de los eventos publicados por los organizadores del congreso, podemos desplazarnos entre ellos pulsando y arrastrando hacia arriba o abajo.

Desde esta vista podemos ir a la vista de detalle de un evento. Para ello debemos pulsar en el evento deseado.

5.4.17 Detalle de un evento

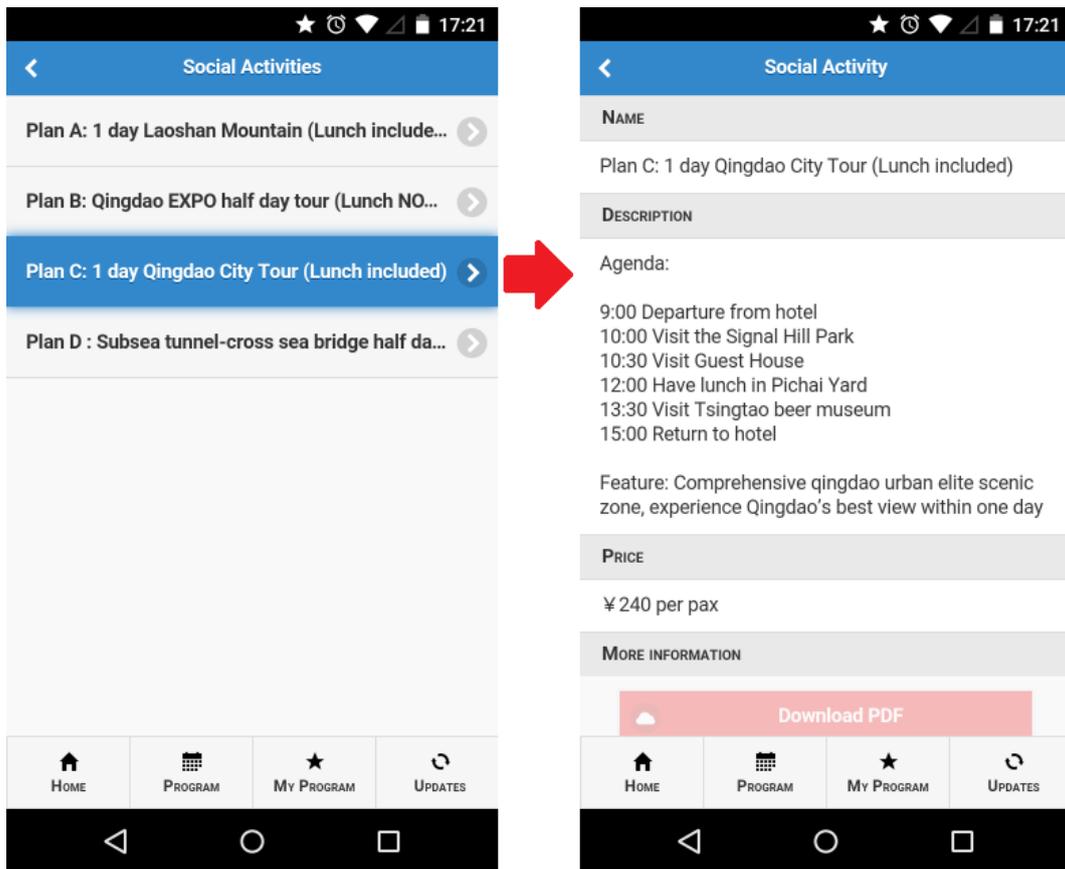
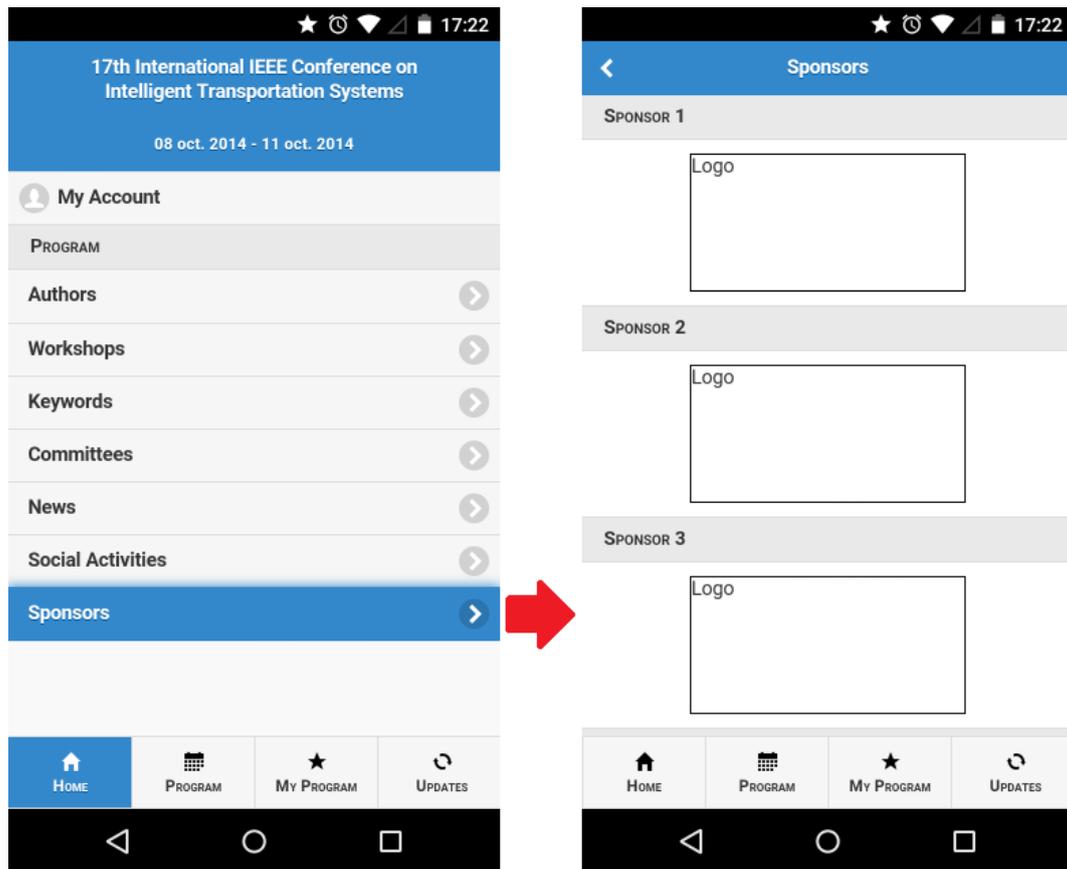


Ilustración 5-48 - Detalle de un evento

En esta vista podemos ver los datos en detalle de un evento, así como acceder a datos externos relacionados con el mismo si fuera necesario, pudiendo consultarlos pulsando en ellos (**sólo para usuarios registrados, no funcional en esta versión**).

5.4.18 Listado de patrocinadores



Esta vista nos proporciona un listado de los patrocinadores del congreso.

5.4.19 Mi cuenta

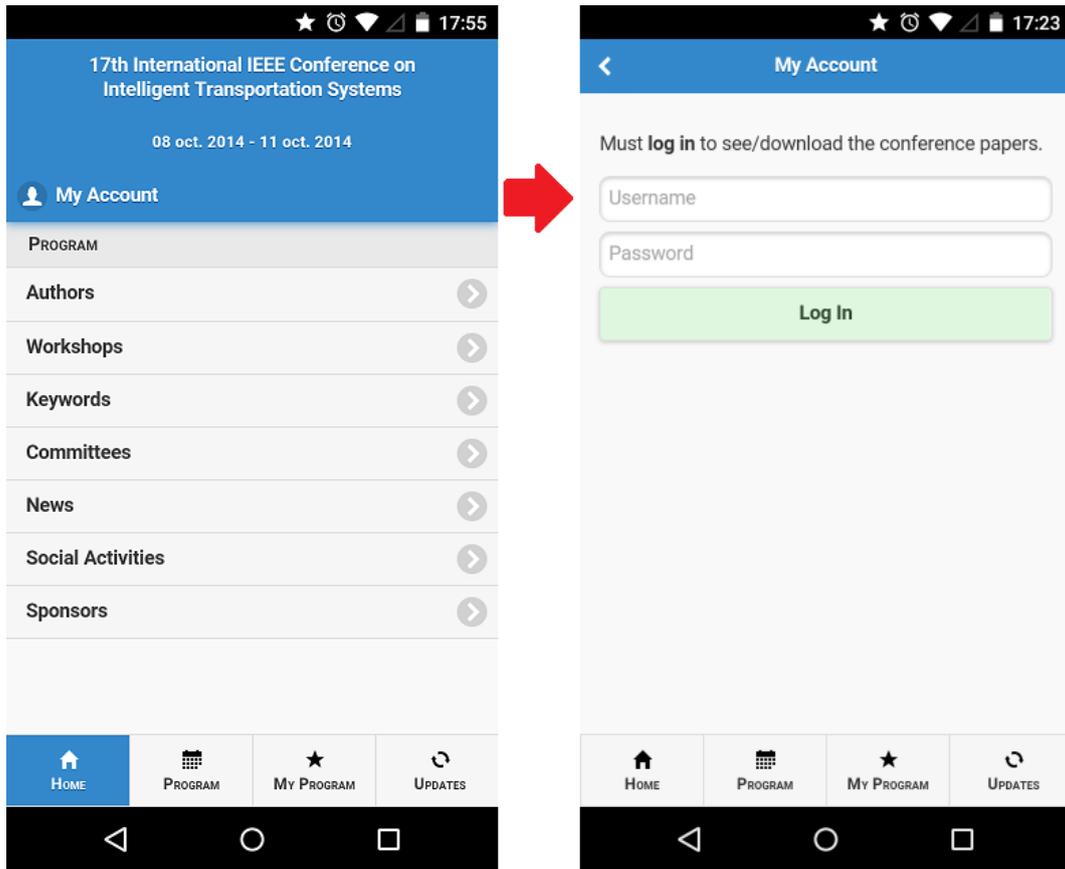


Ilustración 5-50 - Mi cuenta

Esta vista nos proporciona un sistema para autenticarnos en la aplicación. Con esto podremos acceder a contenidos adicionales. Para realizar esto, podemos pulsar en *Username* y *Password*, rellenarlos y finalmente pulsar en *Log In*. **Esta sección no es funcional en esta versión.**