



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Proyecto de Fin de Grado

Plataforma Modular para la Gestión de Ponencias de Congresos Internacionales

Autor:

Gabriel Urso Santana Reyes

Tutores

Dr. D. Javier. J. Sánchez Medina

Dr. D. Enrique Rubio Royo

DEDICATORIA

Dedico el trabajo realizado a mi familia, que me ha apoyado siempre y que sin su ayuda no hubiera podido finalizarlo.

Especialmente a mis padres que ya son informáticos de facto tras atender y escuchar todas mis ideas y quienes me ayudan indicando incluso cómo resolver los problemas.

Y a mi hermano que está siempre dispuesto a escuchar mis lamentos.

Índice de contenido

Introducción.....	7
1 Objetivos.....	7
2 Estado del Arte: Aplicaciones de Gestión de Congresos, definición y algunas actualmente en uso.....	8
2.1 Desarrollos similares de la ULPGC.....	9
2.2 Desarrollos comerciales existentes.....	12
2.2.1 Papercept / Paperplaza.....	12
2.2.2 Openconf.....	13
2.2.3 Easychair.....	16
2.2.4 Conftool.....	17
3. Metodología.....	19
4. Plan de trabajo y temporalización.....	20
5. Organización de la memoria.....	21
Competencias.....	22
Aportaciones.....	23
Análisis.....	24
1. Descripción del problema.....	24
2. Análisis de requisitos de usuario.....	24
2.1 Descripción de los requisitos de usuario.....	24
2.2 Workflow de los artículos.....	29
2.3 Descripción de la metodología y herramientas utilizadas....	36
3. Backbone: Tecnologías utilizadas y justificación.	
Ventajas y desventajas frente a otras alternativas.....	40
3.1 Tecnologías de desarrollo en el lado del cliente.....	40
3.1.1 Lenguajes.....	40

3.1.2 Frameworks.....	40
3.2 Tecnologías de desarrollo en el lado del servidor.....	43
3.2.1 Lenguajes.....	44
3.2.2 Frameworks.....	45
Requisitos hardware y software.....	47
Desarrollo.....	48
1. Descripción de la configuración de la aplicación.....	51
2. Descripción del Módulo Conference.....	53
3. Descripción del Módulo de Gestión de Usuarios.....	54
4. Descripción del Módulo de Gestión de Artículos.....	56
Manual de usuario y software.....	59
Conclusiones y líneas futuras.....	60
Anexos.....	64
Bibliografía.....	65
Glosario.....	66
El patrón modelo vista controlador.....	68
Manual de Usuario.....	71

Introducción

La organización de congresos internacionales incluye un conjunto amplio de tareas diversas, de variadas naturalezas. Algunas de ellas requieren una supervisión cuidadosa al ser factor determinante del éxito del congreso. Es posible que la tarea más delicada y compleja sea la organización del programa científico del evento, porque requiere satisfacer las necesidades, a veces contrapuestas, de muchas personas diferentes, con diferentes roles de participación en el mismo.

Las necesidades de un participante son totalmente diferentes a las de un ponente, las de un revisor, o a las del director de programa de la conferencia (*program chair*). El participante requiere recibir una información en el formato publicitado por el congreso, tal vez poder participar en discusiones o debates, y recibir una documentación en forma de actas, libro de resúmenes (abstracts), etc. Un autor requiere poder enviar para consideración un resumen o un artículo (paper), recibir el resultado de la evaluación del mismo y poder, si es aceptado, subir el artículo definitivo, conocer cuándo tendrá lugar su comunicación, etc.

Para complicar más las cosas, puede darse el caso, y se da con frecuencia, que las mismas personas actúan en el congreso con más de un rol diferente.

1 Objetivos

El propósito de este TFG no puede ser abarcar toda esta problemática, dada la fuerte restricción temporal de los Trabajos fin de Grado. Este TFG trata de esbozar un prototipo que pueda ser extendido con posterioridad este trabajo, donde se presente un prototipo con el esqueleto y estructura fundamental de una aplicación futura que englobe el máximo de elementos de la organización del congreso. También se desea que se desarrollen dos módulos fundamentales:

- un módulo para la gestión de usuarios y roles, asumiendo que podemos definir los roles, vistas y privilegios.
- y un módulo que debe cubrir el workflow de las ponencias, desde su recepción, asignación a revisores, comunicación de resultado y recepción de versión definitiva.

Los objetivos del trabajo a desarrollar en el TFG propuesto son los siguientes:

- 1) Elección de la tecnología web a utilizar, según criterios de mantenibilidad, modularidad, expandibilidad, flexibilidad y prestaciones
- 2) Desarrollo del prototipo de la aplicación global
- 3) Desarrollo de un módulo para la gestión de los usuarios y roles.
- 4) Desarrollo de un módulo para la recepción, revisión e inclusión de comunicaciones en el programa científico de un congreso internacional.

2 Estado del Arte: Aplicaciones de Gestión de Congresos, definición y algunas actualmente en uso

Una aplicación de gestión de conferencias es el software que es utilizado para la organización de conferencias, especialmente científicas. Ayuda a los diferentes tipos de usuario (director de programa (*program chair*), editor asociado (*associate editor*), revisor (*reviewer*), autor (*author*), director/codirector de sesión (*session chair/cochair*), etcétera) a realizar su labor, esas tareas que deben realizar los participantes en la conferencia junto con funcionalidades de ayuda son las siguientes:

- Recibir artículos iniciales (*initial submissions*) (con artículo (*papers*), subida de fichero, y datos del mismo)
- Recopilación de las preferencias temáticas de los revisores
- Recogida de los conflictos de intereses
- Asignación de los revisiones a los documentos
- Difusión de artículos a los revisores
- Informe de recomendaciones de los revisores
- Monitoreo de las opiniones
- Intercambio de opiniones entre el Comité del programa
- Garantizar la independencia de las recomendaciones de los revisores (quienes no pueden ver otros informes de revisión de los artículos)
- Proporcionar un foro de discusión por artículo entre los revisores
- Calificación en los informes de revisión
- Informes de comentarios de los revisores y la decisión del comité del programa para los autores
- Recopilación de versiones finales de los artículos
- Gestor de contenidos del congreso con información del mismo, de la programación, los artículos, y toda la información relevante

Diferentes aplicaciones de gestión de conferencias realizan funciones adicionales, que aunque no son las que una aplicación de gestión de conferencias básica realiza, tiene servicios que pueden ser útiles, esos servicios que pueden ayudar a una aplicación de gestión de conferencias y otros que están implementados en diferentes aplicaciones aunque no sean destinadas a tal fin, pero sirven de ayuda son los siguientes:

- Creación de la web del congreso, una especie de gestor de contenidos específico del congreso, en donde se especifica la información de la conferencia y demás información útil en la misma
- Planificador del programa del congreso, que permita planificar las sesiones del congreso dependiendo de los valores de los artículos, de necesidades específicas de cada ponente, y demás valoraciones que el director del programa decida
- Registrar a los participantes, que puede ser un registro de cada participante, que permita una carga inicial, dependiendo de las necesidades de la aplicación y del cliente
- Pago online de la matrícula al evento, existen diferentes plataformas para realizar esta tarea, pueden estar incluidas en la aplicación o ser externas

- Gestión de las reservas, facturación, servicio que permite a los participantes del congreso reservar plaza en la misma, realizando el pago, y además de reservar alojamiento y demás necesidades que podría llegar a tener un participante del congreso
- Selección de la sede, servicio que permite al organizador del congreso elegir entre diferentes localizaciones en la que podría celebrar el evento
- Organización de las salas y horarios, servicio que planifica las sesiones en las diferentes salas dependiendo de valores indicados por el cliente
- Publicitar, servicio que permite al organizador del congreso publicitar el evento, publicitándolo en redes sociales, enviando publicidad del mismo y demás formas de publicitación del evento

Existen aplicaciones que realizan las tareas de la gestión de conferencias gestionando varias conferencias: hay que tener en cuenta que la aplicación desarrollada es una aplicación que gestiona una conferencia (la información de la misma, los usuarios, los artículos (*submissions*), etcétera), no se trata de una aplicación de organización de múltiples conferencias (que gestione la información de diferentes conferencias, que muestre información de diferentes conferencias que haya, etcétera)

También hay que tener en cuenta que no es una aplicación de ayuda de creación de conferencias (en la que te indique lugares en los que organizar el evento, alojamientos para participantes, y demás servicios externos) como por ejemplo <http://www.kineticsolutions.co.uk/>.

A continuación se mostrarán algunas de las aplicaciones que actualmente están en el mercado, enseñando algunas de sus funcionalidades, hay que tener en cuenta, que algunas son de pago y no pueden ser descargadas gratuitamente para su prueba. Las aplicaciones son las siguientes:

2.1 Desarrollos similares de la ULPGC

Existen varios trabajos de fin de carrera de la Escuela de Informática de la ULPGC desarrollados centrados en la creación de una herramienta informática de apoyo a las reuniones científicas, los trabajos son:

- Reuniones científicas : organización. Herramienta informática de apoyo a la gestión de reuniones científicas, de Jesús Miguel Quintana Hernández y dirigido por Alexis Quesada Arencibia de septiembre del 2006.
- Implementación de un software de apoyo a la gestión de reuniones científicas, de Mario Martín Santana y tutorizado por Alexis Quesada Arencibia de junio del 2008.
- Ampliación, validación y despliegue de software de apoyo a la gestión de reuniones científicas de Abel Silván Vega y tutorizado por Alexis Quesada Arencibia de abril del 2010.

El primero de los trabajos realiza un estudio en profundidad de las reuniones científicas, realizando un análisis de requisitos, además de la base de datos que posteriores trabajos usarán, también da las directrices de las tecnologías que se usarán posteriormente. El siguiente trabajo normaliza el análisis de los requisitos de usuario y realiza la implementación de muchas de las funciones necesarias en el transcurso de reuniones científicas. El último de los trabajos amplía las funciones de los trabajos anteriores.

Las funciones implementadas por los trabajos se pueden resumir en las siguientes:

- Gestión de usuarios.

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

- Gestión de la asignación de revisores. a trabajos propuestos, según diferentes políticas.
- Desarrollo de herramienta para llevar a cabo copias de seguridad de los congresos.
- Gestión de pagos y confirmación de asistencia a los congresos.
- Ayuda a la confección de publicaciones escritas del congreso.
- Apoyo a la confección del programa del congreso sobre un calendario.
- Sistema de registros de usuarios en las actividades del congreso.
- Desarrollo de guía para la utilización correcta de la aplicación.
- Despliegue de la aplicación en un servidor para que sea accesible desde lar red junto con toda la documentación.

Algunas capturas del las funcionalidades del desarrollo son las siguientes:

Congresos disponibles

6 items found, displaying all items. 1

Nombre	Ediciones	Modificar	Borrar
Congress_1			
Congress_2			
Congress_3			
Congress_4			
Congress_5			
Eurocast			

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Administradores del congreso disponibles

Congreso:

Edición del congreso:

2 items found, displaying all items. 1

Login	Nombre	Apellido	Teléfono	Email	Modificar	Borrar
adminCongres	admin	congres		admin@congres.com		
bbbbbb	bbbbbb	bbbbbb		bbbbbb@bbbbbb.com		

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

2.2 Desarrollos comerciales existentes

A continuación se exponen desarrollos comerciales similares al desarrollo realizado.

2.2.1 Papercept / Paperplaza

Página principal de la aplicación:

The screenshot shows the homepage of the PaperCept Conference Manuscript Management System. At the top, there is a navigation bar with the PaperCept logo and the text "PaperCept, Inc. Conference & Journal Manuscripts Management". Below this, the main heading reads "The PaperCept Conference Manuscript Management System" with links for Home, Log In, PIN, Refresh, and Help. A welcome message is followed by instructions for submitting papers, proposals, and reviews. The "Upcoming Conferences" section lists two events: the 51st Annual Allerton Conference on Communication, Control, and Computing (October 2-4, 2013) and the 9th International Conference on MicroManufacturing (March 25-28, 2014). Each event includes submission details and links to submit contributions, contact organizers, and log in. The footer contains sections for PIN, Pdf Compliance and On-Line Pdf Test, JavaScript and Cookies, E-mail and Spam, and Supported Browsers. The URL "publish.illinois.edu/cslallertonconference/" is visible in the address bar.

2.2.2 Openconf

Página principal de la aplicación:

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

OpenConf Home Editors Demo Download Services Support Translations Pricing & Purchase

Peer-Review, Abstract and Conference Management

Known for its ease of use and clean interface, OpenConf has powered thousands of events* in over 100 countries.

Download Compare
Demo Purchase

International Conference

OpenConf Peer Review & Conference Management System

OpenConf Home Email Chair

Wednesday, 1 December

Time	Room A	Room B
8:00-9:00	Welcome & Keynote (Rooms A & B)	
9:00-9:30	Break	
9:30-12:00	CONCURRENT SESSION 1	
	Session 1	Session 2
	Chair: Venkatesh Balaji	Chair: Frank Dreyer
	On Distributed Communications	Asymmetry
	Paul Baran, MIT	R. Keshav, USC-ISI-CC
	Why Protocol Is Not My Favorite Programming Language	Up Over Arise Camera With Quality Of Service
	Brian Armstrong, AT&T Bell Laboratories	D. Watsman, bridge Networks
12:00-14:00	Lunch	
	Buffet lunch will be served in the hotel restaurant	
14:00-17:00	CONCURRENT SESSION 2	
	Session 3	Session 4
	Chair: Donald Davies	Chair: Frank Dreyer
	Communication Networks To Serve Rapid-response Computers	Are We Any There

Submission »
Assignment
Review
Bidding
Discussion
Selection
Program
Mobile Program
Proceedings
Multilingual

General Information:
Submission Title: _____
Submission Type: Paper Poster Short Paper

Author(s)
Author 1:
First/Given Name: _____
Last/Family Name: _____
Organization: _____
Country: _____
Email: _____

Current Release
5.30

OpenConf News
Version 5.30 Released - Nederlands and Italiano added
1 July 2013
Version 5.20 Released - now with 12 complete translations!
5 March 2013
Version 5.10 Released
2 November 2012
Version 5 Released with mobile program and much more
31 July 2012

Privacy Policy | License | Terms of Service | Contact

Zakon group
OpenConf is a registered trademark of Zakon Group LLC.
©2004-2013 Zakon Group LLC. All rights reserved.

Vista del autor y acceso de los miembros del congreso a openconf

www.openconf.com/demo/openconf5/openconf.php

OpenConf Conference 2014

OpenConf Peer Review & Conference Management System

OpenConf Home Email Chair

View Proceedings

Program:

- At-a-Glance Program
- Full Program
- Mobile Program

Authors:

- Make Submission
- Edit Submission
- Upload File
- View File
- Withdraw Submission
- Check Status

Review & Program Committees:

- Sign In
- Sign Up — Keycode: *****

Chair:

- Sign In

English

Powered by OpenConf®
Copyright ©2002-2012 Zakon Group LLC

Vista del program chair a openconf

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

www.openconf.com/demo/openconf5/chairst/index.php?  :nt Management System download

OpenConf Conference 2014
OpenConf Peer Review & Conference Management System v5.x
Chair Home Help Sign Out Signed in

Chair

At-a-Glance Summary
Email (log)

Settings:

- Configuration (advanced)
- Modules
- Open/Close Status (log)
- Topics
- Password
- Database: Backup | Reset

Submissions:

- List/Delete/Withdraw Submissions (log)
- View Uploaded Files [set format]
- Create Submission Stub
- List Author Names: All | Accept | Reject
- List Submissions & Topics: All | Accept | Reject
- Show Topic Count: All | Accept | Reject
- Show Country Count: All | Accept | Reject
- Export Submissions: All | Accept | Reject
- Export Submissions for ACM ICPS

Committees:

- List/Delete: All | Reviewers | Advocates
- List by Topic: All | Reviewers | Advocates
- Show Topic Count: All | Reviewers | Advocates
- Show Country Count: All | Reviewers | Advocates
- Export Committee Members: All
- Show Bid Count: Submissions | Reviewers

Assignments:

- Set Conflicts
- List/Unset Conflicts
- Assign Reviews: Manually | Automatically
- Assign Advocates: Manually | Automatically
- List/Unassign: Reviews | Advocates
- Export Review Data [fields guide]
- Clear Data: Reviews | Advocate Recommendations
- Reviewer Uploads

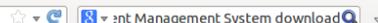
Selection:

- Review Scores & Accept/Reject
- List Submissions by Score with Topics
- Rebuttals

Program:

- Open/Close & Settings
- Create/Edit Program
- Preview Program: At-a-Glance Program | Full Program
- Download HTML Program
- Export Sessions

Vista de las presentaciones de openconf

www.openconf.com/demo/openconf5/chairst/list_papers.php  :nt Management System download

OpenConf Conference 2014
OpenConf Peer Review & Conference Management System v5.x
Chair Home Help Sign Out Signed in as:

Submissions

Number of Submissions: 28 all submissions

ID	Title	Contact Author	Stud.	Type	Review Paper	Final Paper	Presentation	Copyright
<input type="checkbox"/>	1 Man-Computer Symbiosis	J.C.R. Licklider		Paper				
<input type="checkbox"/>	2 The Computer as a Communication Device	J.C.R. Licklider		Paper				
<input type="checkbox"/>	3 As We May Think	Vannevar Bush		Paper				
<input type="checkbox"/>	4 The Next 700 Programming Languages	P. J. Landin		Paper				
<input type="checkbox"/>	5 Why Pascal Is Not My Favorite Programming Language	Brian Kernighan		Paper				
<input type="checkbox"/>	6 Information Flow in Large Communication Nets	Leonard Kleinrock	S	Paper				
<input type="checkbox"/>	7 On Distributed Communications	Paul Baran		Paper				
<input type="checkbox"/>	8 On the Design of Programming Languages	Niklaus Wirth		Paper				
<input type="checkbox"/>	9 Toward A Cooperative Network Of Time-Shared Computers	Lawrence Roberts		Paper				
<input type="checkbox"/>	10 Multiple Computer Networks and Intercomputer Communication	Lawrence Roberts		Paper				
<input type="checkbox"/>	11 Communication networks to serve rapid-response computers	Donald Davies		Paper				
<input type="checkbox"/>	12 On Computable Numbers, with an Application to the Entscheidungsproblem	Alan Turing		Paper				
<input type="checkbox"/>	13 Computer Programming as an Art	Donald Knuth		Paper				
<input type="checkbox"/>	14 Go To Statement Considered Harmful	Edsger Dijkstra		Paper				

Vista del programa de la conferencia de openconf

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

www.openconf.com/demo/openconf5/modules/request.php?module=oc_program&action=sessions.php

OpenConf Conference 2014

OpenConf Peer Review & Conference Management System v5.x

Chair Home | Help | Sign Out

Signed in as: ct

Program

[Add a Session](#) - [Preview](#)

Links:

Time - Edit concurrent session title/order.

Title - Edit session information and add/order accepted submissions.

Only show sessions for the following day:

Time	Topic	Title	Session Chair	Location	Del
Monday, 1 December 2014					
08:00 AM - 09:00 AM		Welcome & Keynote		Rooms A & B	<input type="checkbox"/>
09:00 AM - 09:30 AM		Break			<input type="checkbox"/>
09:30 AM - 12:00 PM	Computer Science	Session 1	Vannevar Bush Office of Scientific Research and Development	Room A	<input type="checkbox"/>
	Humo(u)r	Session 2	Steve Crocker UCLA	Room B	<input type="checkbox"/>
12:00 PM - 02:00 PM		Lunch			<input type="checkbox"/>
02:00 PM - 05:00 PM	Networking	Session 3	Donald Davies National Physical Laboratory	Room A	<input type="checkbox"/>
	Classic Papers	Session 4	S. Christey MonkeySeeDoo, Inc.	Room B	<input type="checkbox"/>
05:30 PM - 08:00 PM		Dinner			<input type="checkbox"/>

2.2.3 Easychair

Homepage de la aplicación



conference system

[overview](#)

[news](#)

[users](#)

[FAQ](#)

[installation <](#)

[citations](#)

[Ack](#)
[Thanks](#)
[Contact](#)

installation

Installing EasyChair for your conference is easy. You do not have to download or configure software or arrange backup procedures yourself. All you have to do is to fill out a simple form. The reason for this is that we host all conferences on our server and use our own backup, recovery, mail sending and other procedure. As far as the users are concerned, EasyChair is simply a Web service.

This page describes the installation process of EasyChair in detail.

Request

If you want to use EasyChair for a conference, you should request an installation first. To do so, [click this link](#).

Confirmation

When you submitted your request, you will immediately receive a confirmation by email. The email will be sent to the address you specified. If you do not receive it within a few minutes, it is likely that your mail server either rejected it or put it in your spam folder. If this happens, please check the spam folder or contact your system administrator.

Verification

We check the information you specified. We have to do this for security reasons. The verification process can take anything between a minute and 12 hours. On the average it is less than 2 hours. Normally, requests made during the night time in the UK are processed slower.

When we verified your request, you will receive a notification by email. The majority of our requests are granted. However, if we have insufficient information we may ask you some questions about your conference, again by email. You should check your mailbox for a notification from us.

Initialization

If your request is granted, the notification email from us will contain instructions on how to **initialize** your conference. This amounts to filling out another simple form. You will also need an EasyChair account to initialize a conference.

When you initialized the conference, it is ready to accept submissions. You will be the first chair (administrator) of your conference. You will receive instructions from EasyChair about configuring your conference.

Support

If you or your conference authors and reviewers need support, you should apply for a license that includes our helpdesk feature by using the menu item "License management" in the EasyChair menu.

Ready?

[Click here to make an installation request.](#)

2.2.4 Conftool

Homepage de la aplicación



The screenshot shows the homepage of the Conftool application. The browser address bar displays 'www.conftool.net/en.html'. The page features a navigation menu on the left with options like 'Homepage', 'Features', 'Demo Installations', and 'About Conftool'. The main content area is titled 'ConfTool: Conference Management Software' and includes a list of features, a description of the software, and information about two versions: 'VSIS ConfTool' and 'ConfTool Pro'. The 'ConfTool Pro' section lists several benefits such as hosted software, customizable forms, and enhanced payment options.

Vista del autor de conftool



The screenshot displays the user interface of the Conftool application. At the top, it shows the event details: 'Demonstration Conference 2014', '29 - 31 February 2014', and 'Hamburg, Germany'. The user is logged in as 'prueba'. A message states: 'Bienvenido(a), Prueba Prueba! Está conectado con el nombre de usuario prueba. Usted no está inscrito como participante del congreso. Puede inscribirse mediante el siguiente formulario.' Below this, there are several menu options: 'Contribuciones', 'Inscribirse como participante', 'Mostrar las sesiones del congreso', 'Mostrar datos del usuario', 'Editar datos del usuario', 'Terminar sesión', and 'Salir y volver a la página principal'. The footer contains legal information and contact details for the software.

Vista de las funcionalidades de conftool

The screenshot displays the ConfTool website interface. At the top left is the ConfTool logo, a cluster of colored dots. To its right is the text 'conftool® Conference Management Tool' and the tagline 'simple - powerful - secure'. Language selection options for 'English' and 'Deutsch' are visible in the top right. A left-hand navigation menu lists various sections, with 'Admin Documentation' highlighted in yellow. The main content area is titled 'Documentation for Administrative ConfTool Users' and contains a list of administrative tasks and a feedback statement.

conftool®
Conference Management Tool

English Deutsch

simple - powerful - secure

Homepage
Features
Demo Installations
User Documentation
Admin Documentation
User Roles
User Import
Sending Bulk E-Mails
Sending Invitations
Downloading Multiple Files
Assigning Persons to Tracks
Assigning Reviews
Review Results & Acceptance Status
Create the Conference Agenda
Data Export for the Proceedings
Create a Browsible Copy of the Session Overview
Creating Name Tags
FAQ
Configuration Documentation
Organizer Documentation
Technical Documentation
ConfTool Support Forum
ConfTool Glossary
References
About ConfTool

Documentation for Administrative ConfTool Users

This information is for users who have Administrator (Admin), Chair or Assistant user rights in the ConfTool system.

Currently you find here:

- A list of the different user roles in ConfTool
- Importing users and reviewers (ConfTool Pro)
- Sending Bulk E-Mails to different user groups (authors, reviewers, participants asf.)
- Creating and sending invitations for authors, reviewers, PC members or participants.
- An introduction on how to download multiple files.
- Assigning Chairs and Reviewers / PC Members to different tracks, sub-conferences, workshops asf. (ConfTool Pro with Track-Chair-Module).
- How to assign submissions to reviewers / pc members for evaluation.
- Inspect the review results and set the acceptance status.
- A short guide on how to export data for the conference proceedings.
- An introduction on how to create a browsible copy of the conference agenda of ConfTool Pro.
- Information about creating name badges and the list of participants (ConfTool Pro).
- Answers to administrators' most frequently asked questions.

Feedback is always appreciated.

Page last updated: May 23, 2013 · Site and Graphics © 2013 by Harald Weinreich & ConfTool GmbH, Germany · All Rights Reserved

3. Metodología

Inicialmente se ha realizado un análisis de las necesidades hardware de la aplicación, buscando necesidades del servidor en el que se implantaría la aplicación como en los cliente que van a acceder a ella.

También se ha estudiado necesidades de probar la aplicación en diferentes plataformas de cliente, para tal necesidad, se ha comprobado la necesidad de utilizar máquinas virtuales si queremos comprobar su uso en clientes diferentes Internet Explorer, ya que no es suficiente únicamente disponer de un software de testado de html de IE porque el motor de IE6 no puede ser instalado conjuntamente con el de versiones superiores de Internet Explorer, eligiendo como software de virtualización de máquinas VirtualBox.

Se ha estudiado las herramientas que se ajustan a la metodología de desarrollo de la aplicación, las herramientas de modelado, de versionado, de comunicación con el tutor y demás herramientas necesarias para el desarrollo de la aplicación.

Tras realizar las necesidades de la aplicación se estudió los lenguaje y frameworks a utilizar en el desarrollo, lenguajes y frameworks que se ajustaran a las necesidades actuales y a la evolución de las mismas. Teniendo en cuenta que va a realizarse un desarrollo en cliente y en servidor, y eligiendo lenguajes y frameworks para cada uno de ellos.

Se ha tenido en cuenta la metodología de desarrollo, intentando utilizar una metodología de desarrollo ágil, realizando entregas en periodos cortos de prototipos de la aplicación, que permitieran dar una visión del estado de la misma, que permitiera realizar modificaciones de las necesidades según se fueran observando, para lo cual ha sido necesario estar en contacto con el tutor lo más posible.

Antes de comenzar con la implementación se han realizado utilizado diferentes herramientas de modelado y diseño, como diagramas de casos de uso y storyboards que han dado una visión muy importante de las necesidades de la aplicación.

También se realizó un estudios del diseño de la base de datos de la aplicación utilizando el modelo entidad relación, que ayudó a tener una visión de las necesidades de la estructura de datos que iban a ser necesarias, aunque posteriormente se utilizó un modelo de datos orientado a objetos que fue exportado automáticamente por el framework de desarrollo a la base de datos.

La implementación de la aplicación en el servidor se ha realizado utilizando python y django, que nos ha permitido realizar el prototipo de forma muy rápida, permitiendo realizar las pruebas con el motor de aplicaciones propio del framework, y utilizando herramientas como la aplicación de migración del modelo de datos a la base de datos a utilizar antes mencionado, la aplicación de administración propia, y demás herramientas muy útiles que dispone, además de proporcionarnos una estructura muy simple y útil de estructurar el código en aplicaciones.

La implementación de la aplicación en el cliente se ha realizado utilizando HTML5, CSS3 y JavaScript, con la ayuda de jquery y API's que nos han permitido implementar algunas funcionalidades de forma mucho más sencilla.

La memoria del proyecto ha sido realizada con LibreOffice, y la documentación y código fuente, entregado en un disco.

4. Plan de trabajo y temporalización

En relación al Plan de Trabajo a llevado a cabo, las etapas a cubrir durante el desarrollo del TFG propuesto fueron las siguientes:

1. Elaboración de las **Especificaciones de Usuario**. El peticionario del proyecto especifica un conjunto de requisitos, agrupados por roles, como pueden ser autor, revisor, director de programa, etc.
2. Partiendo de la Especificación de Usuario el proyectando **diseña una base de datos relacional** en la cual se implementan la información, las entidades y las relaciones presentes en el ciclo de envío, revisión y aceptación de artículos a congresos internacionales.
3. Diseño de **Workflows, Storyboards**, y demás diagramas previos a la programación web, para detectar posibles lagunas en la especificación de usuario, errores en la comprensión de la misma, etc.
4. **Desarrollo del esqueleto de la aplicación web**, haciendo hincapié en la modularidad y la extensibilidad de la aplicación para la futura incorporación de nuevos módulos, como de comprobación de formato de los artículos, de planificación de las comunicaciones en el congreso, etc.
5. **Desarrollo y prueba del Módulo de Gestión de Usuarios y Roles**
6. **Desarrollo y prueba del Módulo** requerido por el peticionario para la implementación del ciclo de envío, revisión y aceptación/rechazo **de los artículos**.
7. **Testeo de la aplicación**, tratando de buscar posibles debilidades, líneas futuras de desarrollo, etc.

La estimación de la temporalización fue la la siguiente:

Etapas	Dedicación Estimada
Etapa 1: Elaboración de las Especificaciones de Usuario	25 horas
Etapa 2: Diseño de la Base de Datos a utilizar	40 horas
Etapa 3: Diseño de Workflows, Storyboards, y demás diagramas previos a la programación web	75 horas
Etapa 4: Desarrollo del esqueleto de la aplicación web	50 horas
Etapa 5: Desarrollo del Módulo de Gestión de Usuarios y Roles	50 horas
Etapa 6: Desarrollo del módulo de Gestión de las Comunicaciones del congreso	50 horas
Etapa 7: Testeo de la aplicación/plataforma	10 horas
Dedicación Estimada Total	300 horas

5. Organización de la memoria

Esta memoria está organizada en once capítulos:

1. Introducción: presenta el proyecto, objetivos, metodología, estado del arte, plan de trabajo, recursos necesarios.
2. Competencias:
3. Aportaciones:
4. Normativa y legislación:
5. Análisis:
6. Requisitos de Hardware y de software:
7. Desarrollo: explicación de las tecnologías usadas, justificación de las mismas, explicación de los requisitos de usuario, diagrama de flujo de los artículos (submissions), explicación de la arquitectura de la aplicación, explicación de los módulos
8. Manual de usuario y software:
9. Conclusiones y líneas futuras: se exponen las conclusiones que se han obtenido tras el desarrollo de la aplicación, y se explican líneas futuras posibles en la ampliación de la aplicación
10. Bibliografía: referencias
11. Glosario: términos usados

Competencias

El trabajo de fin de grado debe cubrir y cubre las competencias siguientes:

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 vigentes

El trabajo es precisamente el desarrollo de una prototipo de aplicación en el que se ha tenido desarrollar, diseñar además de seleccionar las herramientas de desarrollo necesarias, tanto IDE, como frameworks, lenguajes, plataformas, hardware, conectividad. Teniendo en cuenta su fiabilidad y seguridad, realizando un estudio de comparación entre las diferentes opciones existentes. Además se ha tenido en cuenta la normativa vigente, utilizando herramientas libres y desarrollando sobre un sistema completamente libres.

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

El proyecto ha sido planificado siguiendo metodologías de desarrollado, y comprobando su utilidad y adaptándola a las necesidades que se iban presentando, la finalización demuestra se ha tenido capacidad para concebir y concluir un proyecto. La creación del prototipo de la herramienta implementada tiene produce una ayuda social teniendo en cuenta la carencia de herramientas de este tipo en el mercado.

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

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

Aportaciones

Este trabajo de fin de grado, además de aportar el desarrollo del prototipo de la aplicación en sí, que permite explicar las tecnologías utilizadas, comparándolas y explicando las metodologías de desarrollo utilizada, permite dar su punto de vista acerca de la organización y metodología de trabajo de la organización de congresos, ya que explica el flujo de los artículos desde que se crea el artículo inicial, hasta que se envía el final tras ser revisado el artículo inicial por el comité organizador, y valorado, para esa tarea da un ejemplo de interfaz de usuario para los diferentes participantes de este tipo de aplicaciones.

Sobre el la comparativa de herramienta existentes, la explicación de la utilizada, la información se encuentra en el capítulo correspondiente, junto con la metodología de trabajo utilizada, además de explicar la arquitectura de la aplicación en sí.

Análisis

1. Descripción del problema

Se solicita el prototipo de una aplicación de gestión de conferencias internacionales, una aplicación que sea accesible desde internet, en la que se gestione usuarios y participantes, que puedan acceder y realizar las diferentes funcionalidades que se describen en el análisis de requisitos de usuario.

Esta aplicación va a cubrir necesidades de la gestión de un congreso, tal como la gestión de artículos, la tarea más importante de la aplicación, que se describe con detalle en el análisis de flujo, y que comprende el proceso desde la creación del artículo hasta la entrega final del mismo, en la gestión de artículos intervienen diferentes tipos de usuarios del congreso, por lo que vamos a necesitar la gestión de los mismos, describiendo los roles y las tareas de cada uno de ellos, así como las particularidades que tienen. La aplicación va a ser accesible desde internet, por lo que vamos a aprovechar esta característica y desarrollar una interfaz que permita ver información del congreso, y además vamos a dotar a la aplicación de funcionalidades que permitan realizar un portal para el congreso, creando la estructura para la posible ampliación de la aplicación a un gestor de contenidos de congresos.

2. Análisis de requisitos de usuario

Para el análisis de requisitos de usuario se ha utilizado la técnica de entrevistas con el cliente, en este caso el tutor, que ha indicado las necesidades de la aplicación, para comprobar la validez de los requisitos obtenidos se ha utilizado herramientas como la realización de storyboards, diagramas de casos de uso y diagrama de flujo

A medida que se tenía una idea más clara de los requerimientos se ha ido implementado el prototipo, adaptándolo a las necesidades que se iban encontrando, en esta fase, se realizaban encuentros cortos en los que se mostraban los avances en el desarrollo de la aplicación.

2.1 Descripción de los requisitos de usuario

Inicialmente fue dada información sobre el funcionamiento de la organización de las conferencias, indicando acciones que se tienen en cuenta, términos necesarios, tipo de usuarios, datos a incluir de cada tipo de usuario, información sobre los artículos, funcionalidades que podría tener la aplicación, incluyendo posibles ampliaciones, la información es la siguiente:

Autor (*author*): El autor es un usuario del congreso que produce una publicación de algún tipo, la cual desea sea incluida en el Programa Científico de la Conferencia, hay que tener en cuenta que un artículo puede ser realizado por varios autores, de los cuales uno será el autor para correspondencia, que será descrito más adelante.

Artículo (*submission*): Un artículo es enviado y será aceptado o rechazado, siendo o no incluido en la conferencia, debe incluir el artículo en un formato compatible, produciendo la presentación (*communication*) o exposición oral, durante el desarrollo de la conferencia. Un artículo tiene un conjunto de datos asociados.

- Título
- Título corto: versión reducida del título para los listados
- Palabras clave
- Autor para correspondencia (*Corresponding Author*): un autor del artículo al que le llega la información de las revisiones y quien debe enviar la información del artículo que va a ser presentado en la conferencia
- Autores: conjunto de autores ordenados. Un listado corto podría ser resumido indicando el apellido del primer autor y “et al.”

Artículo inicial (*Initial Submission*): El artículo que inicialmente se envía al sistema implementado para su evaluación por los colaboradores, de este artículo se producirá el informe que el autor leerá indicando las razones de la aceptación o rechazo y las modificaciones necesarias en el artículo final a entregar si fueran necesarias.

Informe de revisiones (*Reviewers Report*): Producido por los Editores Asociados, que compila las calificaciones y comentarios de las opiniones (*review templates*) de los revisores, además de los comentarios hacia el autor y el director de programa (*program chair*) y la evaluación del editor asociado.

Plantilla de la opinión (*review template*): Un formulario que permite a los revisores para anotar observaciones del artículo específico para ayudar a los editores asociados para tomar una decisión sobre su aceptación/rechazo de su inclusión dentro de la conferencia.

Autor para correspondencia (*corresponding author*): Cada artículo tiene asociado un autor para correspondencia, el cual es uno de los autores, y que será la misma persona a la que se dirigen las comunicaciones relativas a ese artículo específico. En otras palabras, el autor para correspondencia es la persona de contacto para un artículo (*submission*) específico, la que enviará el artículo tanto inicial como final y recibirá las notificaciones e informes.

Palabra clave (*keyword*): Una palabra o conjunto de palabras que expresan un tema específico dentro del alcance de la conferencia organizada. Pueden ser utilizados como etiquetas de indexación/categorización en los artículos, revisores, sesiones, talleres, tutoriales, y demás entidades participantes de las que deseemos indexar.

Sesión: Cada parte de tiempo ininterrumpida durante el desarrollo de una conferencia, donde está previsto que haya una serie de comunicaciones que se expondrán en el evento. Se espera que los artículos programados en una sesión que se agrupen atendiendo a un conjunto de palabras clave. Los artículos programados en sesión son el resultado filtrado de un proceso de revisión desarrollado antes de la conferencia.

Director de Programa (*program chair*): uno o más voluntarios en el comité organizador de la conferencia a cargo del desarrollo de todo el programa científico de la reunión. Su función principal es la gestión de los artículos. En la tarea de gestión de artículos, tendrá que asignar cada artículo a un editor asociado, para que evalúe el artículo tras comprobar los informes de los revisores que el editor haya seleccionado para cada artículo. La asignación del artículo al editor asociado y posterior evaluación del mismo se realiza para facilitar su tarea de aceptación o rechazo de los mismos. También tiene la función de gestionar los usuarios, la cual realiza indicando el rol que tiene cada

usuario en el sistema.

Editor Asociado: Un voluntario que está de acuerdo en coordinar el proceso de revisión de una serie de artículos, su tarea principal es asignar la revisión de cada artículo que tenga que evaluar a un revisor, y tras comprobar los informes enviar al director del programa su evaluación.

Director/Codirector de Sesión (*session chair/cochair*): Un voluntario que se compromete a llevar a cabo una sesión específica, presentando a los participantes, la promoción de preguntas y debate útil, controlar el tiempo y las condiciones adecuadas para la sesión, etc.

Director de inscripciones (*registration chair*): Uno o más voluntarios en el comité organizador de la conferencia a cargo del registro de participantes, la producción de registro asociada a la documentación, registro, certificados / insignias

Categoría de inscripción: El sistema debe manejar un conjunto de diferentes categorías de registro, dependiendo de registro temprano o tarde, la condición a la de registro, los tipos de registro (registro completo, inscripción de artículo adicional, inscripción reducida, pase de comida, pase de comida extra, reducción de cuota de inscripción, tasas de matrícula estudiantil, etc.)

Requisitos de cada rol:

Usuario (*user*): Una persona con derecho a actuar con uno o más de los roles definidos para el sistema, y con la siguiente información introducida en el sistema:

- nombre de usuario (*username*)
- contraseña (*password*)
- nombre (*name*)
- apellidos (*surname*)
- dirección de correo electrónico (*email addresses*)
- dirección de trabajo (*office address*)
- afiliación académica (*Affiliation/Institution*) (es necesario que sea elegible entre las existentes, y si no existiera que se pudiera añadir)
- teléfono (*telephone*)
- fax
- palabras clave (*keywords*): se utilizan para identificar el campo de la experiencia y/o interés de un usuario, que es útil para algunas funciones diferentes (autor, revisor, AE)

Autor (*author*): Un subtipo de usuario que produce una publicación de algún tipo, con derecho a ser incluidos en el Programa Científico de la Conferencia. Un autor tiene asociados algunos datos personales:

- toda la información asociada a su usuario
- relevancia (*VIP condition*)
- número de miembro IEEE (*IEEE membership number*)
- condición de miembro IEEE (*IEEE member condition*)
- condición de miembro IEEE ITSS (*IEEE ITSS member condition*)

Acciones:

- Tiene que ser capaz de cargar uno o más artículos. Sólo los autores para correspondencia de cada artículo al que corresponde puede modificarlo, así también los artículos finales y transferencia de derechos de autor
- Deberá cargar la información referente al artículo:
 - título (*title*)

- título corto (*short title*)
- resumen (*abstract*)
- palabras clave (keywords) (Permitir elegir entre las existentes e introducir nuevas)
- Al enviar el artículo debe poder descargarlo para comprobar que es su artículo
- Si el artículo es aceptado, entonces tendrá que ser capaz de recibir el informe de revisión
- A continuación, tiene que enviar el artículo final en el formato adecuado, y también una transferencia de derechos de autor
- El artículo debe pasar un filtro de plagio
- Una vez cargado correctamente el artículo, los datos asociados y la forma de transferencia de derecho de autor, el autor tiene que consultar dónde/cuándo se programará su artículo para su presentación.
- Un autor puede tener que expresar sus preferencias o necesidades en cuanto al momento en que se programará su artículo. Una de las prioridades de planificación que se utilizará en función de su relevancia (condición VIP)

Editor Asociado (AE - *associate editor*): Un usuario que está de acuerdo en coordinar el proceso de revisión de una serie de artículos.

- Inicialmente, tiene que ser asignado por el director de programa de la conferencia (*program chair*), que le enviará los artículos iniciales
- A continuación, tendrá que asignar cada artículo inicial a una serie de revisores, dependiendo de las palabras clave del artículo y la experiencia de los revisores, expresada también como un conjunto de palabras clave.
- Puede tener que enviar recordatorios a los colaboradores inicialmente asignados para confirmar que aceptan la tarea de revisión de cada artículo
- Puede tener que volver a asignar un artículo a un grupo diferente de revisores si alguno por algún motivo rechazara la revisión del mismo

Revisor (*reviewer*): un usuario que se compromete a revisar una serie de artículos, valorándolos de acuerdo a una plantilla, incluyendo algunas ideas de mejora y una recomendación de decisión ([Aceptado (*accepted*) | Aceptado con cambios menores (*accepted with minor changes*) | Aceptado con modificaciones principales (*accepted with major changes*) | Rechazado (*rejected*)]). También tendrá que comprobar el artículo asignado para detectar casos de plagio.

Director de programa (PC *program chair*): uno o más usuarios de la conferencia del comité organizador encargado de desarrollar todo el programa científico de la reunión.

- Dependiendo del alcance de conferencias, necesitará para definir los perfiles de los artículos para que sean aceptados
- Puede asignar los roles editor asociado y revisor a los usuarios.
- El proceso de revisión se inicia asignando los editores asociados a los artículos, dependiendo de la relación de las palabras clave de las mismas y el perfil de los editores
- Necesitará reasignar artículos dependiendo de diversos motivos
- Podría tener que enviar mensajes de aviso instando a los editores que confirmen su aceptación o rechazo de un envío de artículo
- Tendrá que recibir los informes de revisión de cada artículo, con el fin de tomar la decisión de aceptación.
- Puede tener que enviar recordatorios para instar a los editores que dispongan sus informes de los artículos
- Después se toman las decisiones, tendrá que enviar la notificación a todos los autores

correspondientes con respecto a sus artículos. Para ello, un conjunto de plantillas debe existir: plantilla de aceptación, plantilla de aceptación con modificaciones requeridas, plantilla de rechazo

- También puede tener que tomar decisiones definitivas acerca de plagio, si hubiera casos detectados
- Después de recibir las conclusiones finales que necesita asignará los artículos a las diferentes sesiones existentes
- Puede crear sesiones
- Tiene que asignar moderadores para cada sesión (director de la sesión y/o codirector)
- A continuación, debe comprobar si hay conflictos en la asignación: básicamente, una persona no puede estar en dos sesiones diferentes pero simultáneas como autor, presidente de la sesión, o cualquier otra función
- Además, puede tener que hacer frente a requisitos especiales de los autores, como el tiempo limitado en la conferencia, que asistir a una sesión específica, las preferencias/necesidades de tiempo. El planificador debe proponer un conjunto de opciones, dependiendo de la condición VIPness de la persona, con el objetivo de hacer frente a su / sus peticiones.

Director de inscripciones (RC - *registration chair*): uno o más participantes del comité organizador de la conferencia a cargo de la inscripción de los participantes, la producción de documentos de registro/certificados/insignias

- necesita un sistema automático/manual (que produzca fácilmente nuevos documentos en el sitio, ideal si fuera el mismo sistema), lo que debe producir es:
 - Cartas de asistencia (*Attendance Letters*) que se entregarán en el lugar de la conferencia para los que se registraron y participan
 - Cartas de participación (*Participation Letters*) que se entregarán en el momento de la conferencia o después de la conferencia a los autores de las propuestas finalmente incluidas en el programa de la conferencia
 - Recibos de inscripción (*Registration Receipts*) que se entregarán una vez el pago de inscripción on-line y/o en el sitio, incluyendo la información de la tarjeta de crédito/cuenta paypal usada para pagar
 - Carta de invitación (*Invitation letter*) para ser descargado de Internet por cada participante), incluyendo la firma del director en ella
 - Insignias (*Badges*) nombre del autor y la afiliación en ella, fácilmente imprimibles sólo introducir las características del papel microperforado para su uso. Fácil de realizar, incluso en el lugar del evento
- Un módulo de registro manual en el sitio podría ser desarrollado para ser utilizado con flexibilidad en el lugar, lo que permite pagos con tarjeta de crédito (a través de Paypal) y también en efectivo, y la producción sobre la marcha toda la documentación referida a solicitantes de registro.
- Además, un servicio en línea para producir placas y cartas de asistencia introducir manualmente los datos en ellos debe estar también disponible.
- Un servicio de listado/query flexible debería estar disponible, para configurar fácilmente los datos a ser incluidos. Podría incluirse una serie de preguntas estandarizadas, una dificultad media servicio de listado más configurable y una consola de consulta SQL para consultas específicas (con un servicio de ayuda / tutorial adjunto).

Director/codirector de sesión (SC/ScC - *session chair/cochair*): un usuario que se compromete a llevar a cabo una sesión específica, introduciendo a los participantes, la promoción de preguntas y

debate útil, controlar el tiempo y las condiciones adecuadas para la sesión, etc

- Podría tener un servicio disponible para comprobar quien asiste y quién no en la sesión de que preside
- Podría tener que comunicarse con los autores de la sesión presidida

Administrador (*admin*): Uno o varios usuarios del comité organizador de la conferencia que tienen una vista completa de todo el Sistema de Información de la Conferencia. Podrían cambiar todos los parámetros de la configuración de la conferencia. Tiene una vista de sólo lectura del espacio de trabajo del área de cada función de participante de la conferencia. El administrador puede actuar con cualquier otro rol, con una lectura y escritura a su espacio de trabajo, seleccionando la función deseada y convertirlo en el comportamiento que deseado.

Aparte de todas las capacidades de cada función en el sistema, hay un conjunto de funciones exclusivas para este papel:

- Un administrador es el único tipo de usuario con capacidades para asignar todos los roles posibles para los usuarios.
- Un administrador es el único tipo de usuario que puede eliminar a un usuario del sistema

Restricciones adicionales:

Diferentes vistas dependiendo del rol de acceso por el que se haya identificado el usuario, cada rol tiene funciones propias, por lo que debe mostrarse una vista dependiendo de las necesidades que cada usuario tenga.

Estructurar la aplicación para que pueda almacenar información del congreso, no manteniendo la información del congreso de forma estática, realizando la base de un gestor de contenidos del congreso, y permitiendo que sea un módulo ampliable en un futuro.

2.2 Workflow de los artículos

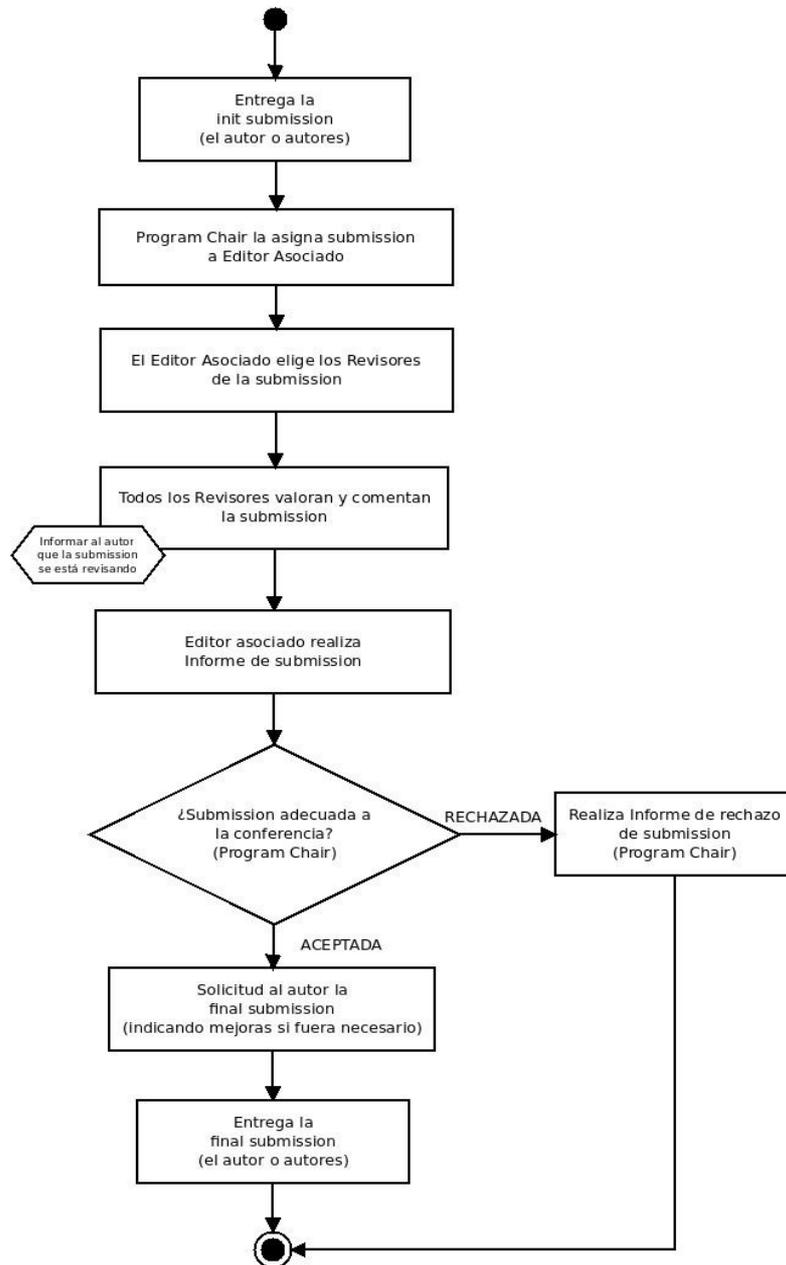
Una de las tareas más importantes de la aplicación es la gestión de artículos, para ello existe un proceso que se debe cumplir, este proceso incluye desde el momento en el que el autor crea el artículo inicial hasta que envía la final, si fuera aceptada, pasando por ser revisada y evaluada por los revisores, que son seleccionados por los editores asociados, que indican su opinión sobre la aceptación o rechazo de la misma, que la valora el director de la conferencia e incluye o no el artículo en la misma.

El diagrama de flujo nos aporta una información muy importante en el proceso de tramitación de los artículos (*submissions*), y es importante conocer este proceso incluso a los participantes del comité organizador, que deben realizar las tareas de revisión, valoración y asignación de los artículos.

En el diagrama participan los diferentes tipos de usuario, realizando cada uno de ellos las tareas correspondientes a su rol.

Puede existir un usuario que tenga diferentes roles, pero debe realizar las tareas propias dependiendo del rol que tenga en cada momento

El diagrama de flujo es el siguiente:



El proceso de aceptación de los artículos comienza en el momento en el que un autor crea un artículo, convirtiéndose en ese momento del autor para la correspondencia del mismo, puede modificarlo hasta la entrega del mismo, en ese momento, ya no puede modificarlo, y esperará al

informe del mismo, que indicará si es aceptado o no, entregando el artículo final con las modificaciones indicadas en el informe.

Una vez que el autor para correspondencia entrega el artículo, el director de programa elige el editor asociado que realizará la evaluación y el seguimiento del artículo.

El editor asociado comprobará los artículos que se le ha asignado y para su evaluación necesitará informes de revisión del artículo, que obtendrá eligiendo los revisores del artículo que se le ha asignado.

El revisor observa los artículos que le ha asignado y realizará la revisión de los mismos, valorándolos.

De las valoraciones de los revisores el editor asociado dará un informe al director de programa dando su punto de vista sobre la aceptación del artículo, el informe también servirá como referencia al autor para realizar las modificaciones oportunas. El informe del editor asociado contiene información a la que sólo el director de programa pueda acceder, en la que puede indicar opiniones sobre posibles plagios, y demás información interna que no debe ser conocida por miembros ajenos a la dirección de programa.

El director de programa atendiendo al informe del editor asociado indicará si está aceptado o no el artículo, enviando un informe de revisión, en el cual se indican las modificaciones que fueran necesarias.

El autor podrá comprobar si su artículo ha sido revisado y la valoración del mismo, viendo el informe de revisión y observando las modificaciones que harían falta en el artículo final para la entrega, la cual podrá realizar una vez que sea aceptado.

2.3 Descripción de la metodología y herramientas utilizadas

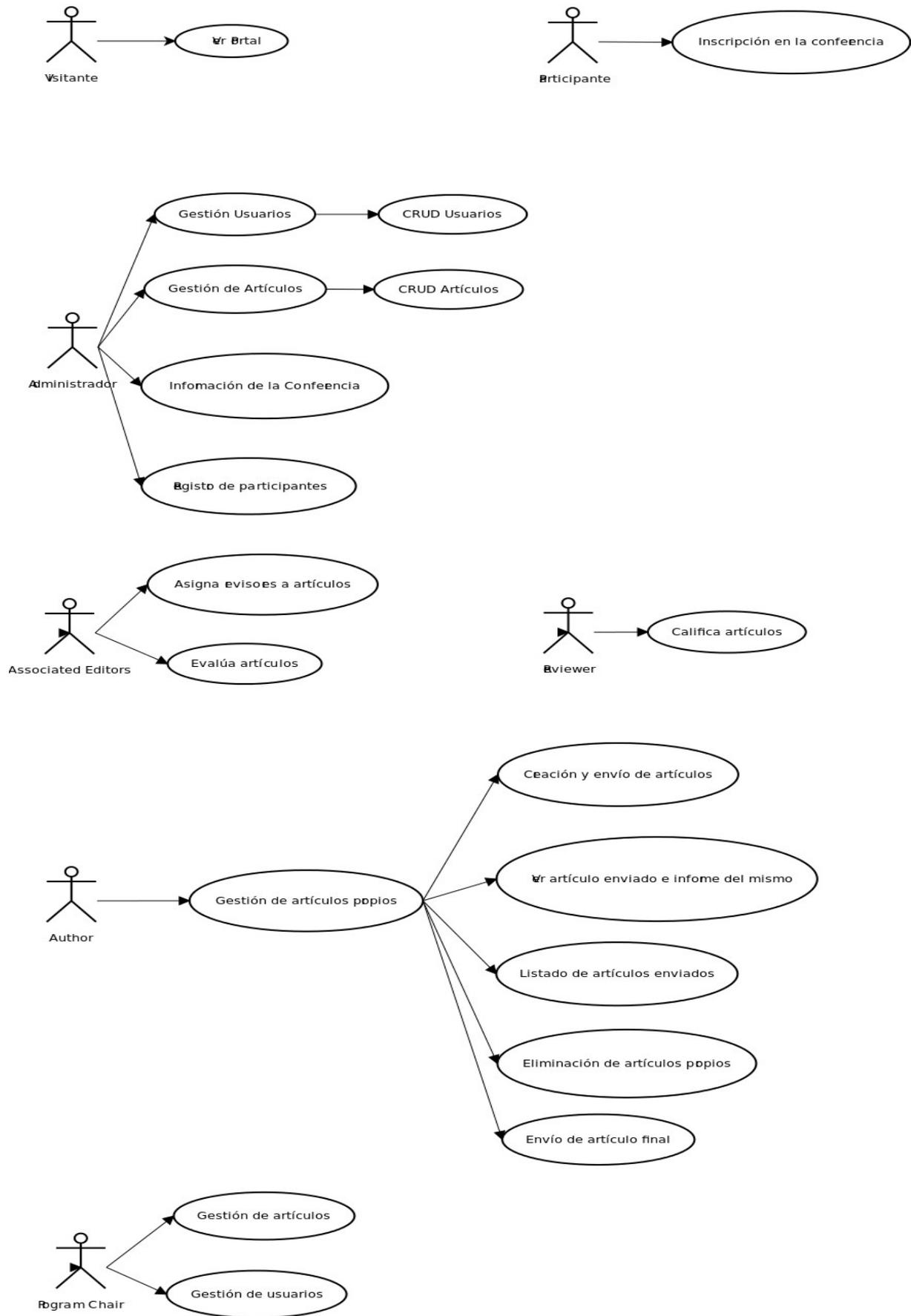
Una de las herramientas más útiles que hemos utilizado es el diagrama de casos de uso, con la que hemos definido los roles y las acciones de cada uno de los actores.

El diagrama nos ha ayudado a diseñar la interfaz de usuario, en la cual se han incluido las acciones de cada actor en la vista de cada tipo de usuario.

Ha sido útil incluso en la modularización de la aplicación agrupando las diferentes acciones de los usuarios en módulos.

El diagrama obtenido inicialmente es el siguiente:

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes



Otra de las herramientas más importante han sido los storyboards, que no sólo nos ha ayudado en la definición del diseño de la interfaz de usuario, sino que nos ha aportado información de qué es lo que el usuario necesita, y las funcionalidades que deseaba que tuviera la aplicación.

Se realizaron storyboards como el siguiente:



3. Backbone: Tecnologías utilizadas y justificación. Ventajas y desventajas frente a otras alternativas.

Las especificaciones iniciales indican que el desarrollo es el de una aplicación web, por lo que la elección de las tecnologías debe dividirse en el lado del cliente y en el lado del servidor.

Al hablar de las tecnologías utilizadas en el lado del cliente nos referimos que el cliente es un navegador web, los requisitos del mismo son que soporte HTML5, CSS3 y JavaScript, aunque muchas de las características de HTML5 y CSS3 de las que carezcan navegadores antiguos podrán ser utilizadas gracias al uso del framework 960gs que permite compatibilidad con navegadores que carezcan de muchas de las características utilizadas.

Al hablar de las tecnologías utilizadas en el lado del servidor nos referimos a un servidor que pueda soportar el servidor de aplicaciones de django y el sistema de gestor de bases de datos sqlite3.

3.1 Tecnologías de desarrollo en el lado del cliente

Según las necesidades de la aplicación en el lado del cliente podíamos elegir entre desarrollar sobre flash o HTML con CSS y JavaScript, las especificaciones de la aplicación nos permiten elegir entre cualquiera de ellas, ya que HTML-CSS-JavaScript no requieren instalación de ningún tipo de software adicional al navegador. Se ha optado por esa tecnología. Además teniendo en cuenta que los navegadores actuales en su mayoría soportan HTML5, y CSS3, se ha utilizado muchas de sus características, también teniendo en cuenta algunos navegadores más antiguos, el framework 960gs utilizado permite que las característica que no soportan sean suplidas por las que soportan.

3.1.1 Lenguajes

Los lenguajes elegidos por tanto han sido HTML5, CSS3 y JavaScript, tal como se ha indicado anteriormente tiene la ventaja de venir por defecto en los clientes sobre los que vamos a desarrollar, a diferencia de otras tecnologías como Flash.

HTML5 es un lenguaje de marcas utilizado para el diseño de la interfaz de usuario.

CSS3 es un lenguaje para describir la presentación semántica (descrita por HTML) por lo que describe el aspecto y formato.

JavaScript es un lenguaje de programación interpretado del que la mayoría de navegadores disponen de intérprete y su utilidad utilizada ha sido la validación de formularios y proporcionar características adicionales como selección múltiple en algunos campos de formularios.

3.1.2 Frameworks

Se ha utilizado el framework 960gs, para la maquetación de la aplicación, es un framework css que divide la pantalla en 12 o 16 columnas que vamos combinando según las necesidades pudiendo montar de una forma sencilla webs con múltiples secciones diferentes en tamaño, zonas para cajas, etc. Estas divisiones se configuran de forma natural, donde simplemente tendremos que ir sumando los tamaños que vamos concatenando para llegar a esas columnas que estamos usando como guía.

El uso de un framework para la maquetación como 960gs nos proporciona unas ventajas muy importantes como la compatibilidad, siendo compatible con la mayoría de los navegadores más utilizados como el Mozilla Firefox, Google Chrome, Safari, Internet Explorer y Opera, la estructura para la diagramación de los sitios web que dispone, pues evita el proceso de creación de una nueva estructura cada vez que se inicia un proyecto, un sistema como este ofrece una retícula que mejora el balance, la alineación y el espacio para lograr una mejor experiencia visual,

Por lo que el uso implica una mayor velocidad de desarrollo, ya que la compatibilidad con diferentes navegadores y la maquetación en columnas definidas está ya implementada.

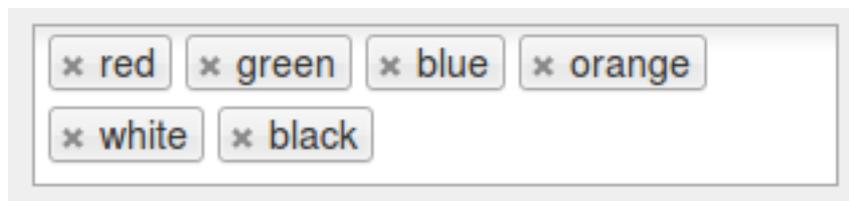


24 Column Grid	
	950
30	910
70	870
110	830
150	790
190	750
230	710
270	670
310	630

Para la creación de campos de selección múltiple se ha utilizado la API select2 que permite la selección de múltiples datos, esta API es utilizada por JQuery, por lo que también ha sido importada en la

aplicación.

La API select2 permite muchísimas personalizaciones y en su portal tiene un manual muy explicativo [1].



3.2 Tecnologías de desarrollo en el lado del servidor

En el desarrollo de una aplicación web, un factor muy importante es el framework elegido, es la herramienta con la que vamos a desarrollar la aplicación, por lo que su elección es un factor determinante.

Hay que tener en cuenta el lenguaje en el que vamos a desarrollar con ese framework, por lo que hay que conocer las características del mismo, y si es un lenguaje script, además conocer las características de su intérprete.

Los lenguajes que se ha estudiado su uso han sido: Java, ASP.NET, python, php y ruby.

Y de cada lenguaje se ha elegido un framework para comparar, de Java JSF, de ASP.NET ASP.NET MVC, de python Django, de PHP Symphony y de ruby Ruby on Rails, se han elegido estos frameworks de cada lenguaje opinando que son los que más se ajustan a las necesidades propias.

Hay que tener en cuenta también el sistema de gestión de base de datos a usar, en este caso, ya que se trata de un prototipo, se ha optado por un sistema de gestión de base de datos ligero, de fácil uso y configuración como es sqlite3, además el tiene soporte en muchas plataformas y por los diferentes frameworks a elegir.

3.2.1 Lenguajes

El lenguaje de desarrollo en el servidor elegido es python, lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

[1] <http://ivaynberg.github.io/select2/>

Además de la legibilidad, orientación a objetos y demás características que son muy importantes en la implementación de la aplicación, debemos tener en cuenta ya que se trata de un lenguaje interpretado del intérprete del que disponemos, y es una gran ventaja, ya que existen intérprete para muchas plataformas, como Windows, OS X, UNIX y GNU/Linux, y se caracteriza por su velocidad y robustez.

Una característica tenida en cuenta ha sido la comunidad de desarrolladores de python, una amplia comunidad, muy activa.

python™

search

Advanced Search

ABOUT »

NEWS »

DOCUMENTATION »

DOWNLOAD »

下載 »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

Help

Package Index

Quick Links (2.7.5)

- » Documentation
- » Windows Installer
- » Source Distribution

Quick Links (3.3.2)

- » Documentation
- » Windows Installer
- » Source Distribution

Python Jobs

Python Merchandise

Python Wiki

Python Insider Blog

Python 2 or 3?

Help Fund Python

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved [open source license](#).

New to Python or choosing between Python 2 and Python 3? Read [Python 2 or Python 3](#).

The [Python Software Foundation](#) holds the intellectual property rights behind Python, underwrites the [PyCon conference](#), and funds many other projects in the Python community.

[Read more, -or- download Python now](#)

- » **Python 3.4.0 alpha 1 has been released**
The first alpha for Python 3.4, [Python 3.4.0a1](#), has been released.
Published: Sat, 3 August 2013, 23:00 -0800
- » **EuroPython 2014/2015 Conference Team Call for Site Proposals**
The EuroPython Society announced the [Call for Proposals for EuroPython](#), to collect proposals from teams volunteering to organize the EuroPython conference in 2014-2015.
Published: Thu, 13 June 2013, 11:43 -0400
- » **Python 3.2.5 and 3.3.2 have been released**
[Python 3.2.5](#) and [Python 3.3.2](#) regression fix releases have been released.
Published: Wed, 15 May 2013, 22:30 +0100

Support the Python Community

Help the Python community by becoming an associate member or making a one-time donation.

Python 3 Poll

I wish there was Python 3 support in

(enter PyPI package name)

Vote Results

AFNIC.fr uses Python...

... joining users such as Rackspace, Industrial Light and Magic, AstraZeneca, Honeywell, and many others.

3.2.2 Frameworks

El framework de desarrollo en el lado del servidor elegido ha sido Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el paradigma conocido como Model Template View. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

En la elección del framework de desarrollo, se han comparado ASP.NET MVC, JavaServer Faces, Symphony, Ruby on Rails y Django.

La comparativa entre los frameworks elegibles para el trabajo, donde se indican algunas

características relevantes, están descritas en la siguiente tabla.

Project	Lenguaje	Versión estable actual	Versión de lanzamiento	Licencia
Django	python	1.5.1	2013-03-28	BSD
Symfony	php	2.3.1	2013-06-19	MIT
JavaServer Faces	java	2.1	2010-10-22	CDDL, GPL 2, Apache 2.0,
Ruby on Rails	ruby	4.0.0	2013-06-25	MIT, Ruby
ASP.NET MVC	ASP.NET	4.0	2012-08-15	Apache v2

Project	Ajax	MVC framework	MVC push-pull	i18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)
Django	Full stack	Push	Yes	Yes	Django ORM	Yes	Provided by South
Symfony	Any	Yes	Push	Yes	Propel, Doctrine (YAML)	Yes	Plugin exists
JavaServer Faces	Yes	Yes	Pull	using Java i18n		external, built-in	
Ruby on Rails	Prototype, script.aculo.us, jQuery	ActiveRecord, Action Pack	Push	Yes	ActiveRecord	Unit Tests, Functional Tests and Integration Tests	Yes
ASP.NET MVC	Yes	Yes	Push	Yes	ORM-independent	Unit tests, Functional Tests, Integration Tests	Entity Framework

Project	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
Django	ACL-based	Django Template Language	Cache Framework	Django Forms API
Symfony	Yes	PHP, Twig	Yes	Yes
JavaServer Faces	pluggable	pure HTML-SVG	page caching	normal Java

Project	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
Ruby on Rails	Plug-in	Yes	Yes	Yes
ASP.NET MVC	ASP.NET Forms Authentication (Default), Pluggable	Razor (Default), ASPX, Pluggable	Yes	Yes (client-side via plugins)

Las características más importantes que influido en la elección del framework están descritas a continuación:

- Intérprete del lenguaje integrado en la mayoría de los servidores
- Robustez y velocidad del intérprete
- Framework maduro
- Patrón de desarrollo del framework que se ajusta a nuestras necesidades (una descripción del mismo se puede observar en el anexo de esta memoria)
- Comunidad de desarrolladores muy amplia y activa
- Multitud de aplicaciones útiles que se pueden añadir

Requisitos hardware y software

Herramientas software utilizadas en el desarrollo:

- VIM (IDE de desarrollo para python y django)
- bash (para lanzar scripts)
- gedit (editor para HTML, CSS y JavaScript)
- firebug (herramienta de depuración de JavaScript y CSS)
- Navegadores Chrome, FireFox e Internet Explorer

Recursos software de soporte:

- Sistema operativo Ubuntu 12.04
- LibreOffice

Recursos Hardware:

- Portátil con los recursos: 4Gb RAM, procesador i5, 500Gb de HD

Frameworks e intérpretes necesarios:

- python 2.7.3
- django 1.5
- 960gs (framework css)
- jquery 1.9
- select2 (api que permite selección múltiple en formularios)

Desarrollo

En este capítulo se presenta tanto el análisis de requisito de usuario, presentando el problema a resolver, y las medidas adoptadas para resolverlos, herramientas utilizadas, metodología de desarrollo y una explicación de los módulos desarrollados.

El desarrollo de la aplicación se ha realizado teniendo en cuenta metodologías de desarrollo ágil, aunque inicialmente se realizó un análisis exhaustivo de los requisitos de usuario, utilizando para tal tarea herramientas como diagramas de casos de uso, diagramas entidad relación y storyboards, posteriormente se optó por ir desarrollando el prototipo e ir mostrándolo y mejorándolo a medida que el cliente vea su funcionamiento comprobando que sus necesidades se cubrían.

La aplicación está modulada en lo que Django denomina aplicaciones, que son ... , las aplicaciones que en las que se ha modulado la aplicación son:

- conference
- submission
- user
- admin
- auth

Django estructura el código de la aplicación almacenando cada módulo o aplicación, como lo denomina Django en un directorio, que contiene los scripts 'views.py' (script que contiene los controladores), 'test.py' (script que contiene las pruebas de la aplicación), 'models.py' (script que contiene los modelos). Además mantiene un directorio con el mismo nombre de la aplicación en el que almacena la configuración de la misma, las url ('url.py'), el script de configuración ('settings.py'). La estructura de los archivos de la aplicación es la siguiente:

```
├── djcon.db
├── initial_data.json
├── manage.py
├── installation
├── ...
├── accounts
├── ...
├── conference
├── ...
├── djcon
├──   ├── __init__.py
├──   ├── settings.py
├──   ├── settings.py.old
├──   ├── urls.py
├──   └── wsgi.py
├── news
├── ...
├── submissions
├── ...
├── users
├── ...
├── static
├── ...
├── templates
├──   ├── base.html
├──   ├── conference
├──   ├── ...
├──   ├── js
├──   ├── news
├──   ├── ...
├──   ├── submissions
├──   ├── ...
├──   ├── users
├──   └── ...
├── uploads
```

La forma con la que Django trabaja con las url es muy sencilla, elegante y potente, en el directorio de configuración de la aplicación tiene un script que tiene un hash que relaciona un patrón, que se utiliza para comparar con la entrada en el servidor de aplicaciones con un método que se ejecuta al recibir esa petición.

El script que contienen las rutas url con las que se accede a la aplicación están definidas en el fichero 'url.py' del directorio djcon de la aplicación. A continuación se encuentra una parte del script de las url de nuestra aplicación:

```
from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    url(r'^$', 'conference.views.index'),
    url(r'^conference/$', 'conference.views.edit'),
    url(r'^contact/$', 'conference.views.contact'),
    url(r'^conference/login/$', 'conference.views.login_congress'),
    url(r'^conference/login/(?P<group_id>\d+)/$', 'conference.views.login_congress'),
    url(r'^conference/logout/$', 'conference.views.logout_congress'),
    url(r'^conference/edit/$', 'conference.views.edit'),
    url(r'^conference/modify/$', 'conference.views.modify'),
    # users url
    url(r'^users/$', 'users.views.index'),
    url(r'^users/new/$', 'users.views.new'),
    url(r'^new_account/$', 'users.views.new_account'),
    url(r'^create_account/$', 'users.views.create_new_account'),
    url(r'^profile/$', 'users.views.profile'),
    url(r'^update_profile/$', 'users.views.update_profile'),
    url(r'^users/edit/(?P<user_id>\d+)/$', 'users.views.edit'),
    url(r'^users/create/$', 'users.views.create'),
    url(r'^users/update/$', 'users.views.update'),
    url(r'^users/remove/(?P<user_id>\d+)/$', 'users.views.remove'),
    # submissions url
    url(r'^submissions/$', 'submissions.views.index'),
    url(r'^submissions/download/(?P<submission_id>\d+)/$', 'submissions.views.download'),
    url(r'^submissions/edit/(?P<submission_id>\d+)/$', 'submissions.views.edit'),
    url(r'^submissions/update/$', 'submissions.views.update'),
    url(r'^submissions/new/$', 'submissions.views.new'),
    url(r'^submissions/show/(?P<submission_id>\d+)/$', 'submissions.views.show'),
    url(r'^submissions/show_final/(?P<finalsubmission_id>\d+)/$', 'submissions.views.show_final'),
    url(r'^submissions/send_final/(?P<submission_id>\d+)/$', 'submissions.views.send_final'),
    url(r'^submissions/new_author_submission/$', 'submissions.views.new_author_submission'),
    url(r'^submissions/final_submission/$', 'submissions.views.final_submission'),
    url(r'^submissions/author_update_submission/$', 'submissions.views.author_update_submission'),
    url(r'^submissions/reviewer_submission/$', 'submissions.views.reviewer_submission'),
    url(r'^submissions/editor_submission/$', 'submissions.views.editor_submission'),
    url(r'^submissions/create/$', 'submissions.views.create'),
    url(r'^submissions/program_chair_submission/$', 'submissions.views.program_chair_submission'),
    url(r'^submissions/remove/(?P<submission_id>\d+)/$', 'submissions.views.remove'),
    url(r'^submissions/remove_final/(?P<finalsubmission_id>\d+)/$', 'submissions.views.remove_final'),
)

"djcon/urls.py" 50L, 2775C
```

1,1 Comienzo

1. Descripción de la configuración de la aplicación

La aplicación se ha implementado sobre el intérprete de python 2.7.3, y utilizado django 1.5, además el SGBD usado es SQLite3 se han instalado las aplicaciones de admin, y módulos de seguridad, además de la configuración completa de la aplicación, idioma, zona horaria, ruta de instalación de la aplicación, aplicaciones instaladas, y demás datos necesarios de la configuración que se encuentra en el fichero setting.py del directorio djcon de la aplicación.

La aplicación utiliza en su inicialización una carga de datos necesaria, que introduce información sobre los grupos de la aplicación (roles que se tienen en el el congreso) e información inicial para el congreso. La carga cargando el fichero 'initial_data.json' realizado mediante mismo comando que también crea la estructura de la base de datos a partir de los modelos de las diferentes aplicaciones, que es 'python manage.py syncdb'.

La creación de la estructura de la base de datos, tal como indicamos anteriormente, se realiza con el comando 'python manage.py syncdb', que lee los modelos de las diferentes aplicaciones instaladas,

y la crea a partir de estos.

A continuación una muestra de setting.py

 Django settings for djcon project.

```
import os.path

DEBUG = True
TEMPLATE_DEBUG = DEBUG

ADMINS = (
    ('Gabriel Urso', 'brielsantana@gmail.com'),
)

MANAGERS = ADMINS

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3', # Add 'postgresql_psycopg2', 'mysql', 'sqlite3' or 'oracle'.
        'NAME': '/home/briel/djcon/djcon.db', # Or path to database file if using sqlite3.
        'USER': '', # Not used with sqlite3.
        'PASSWORD': '', # Not used with sqlite3.
        'HOST': '', # Set to empty string for localhost. Not used with sqlite3.
        'PORT': '', # Set to empty string for default. Not used with sqlite3.
    }
}

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = 'Atlantic/Canary'

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = 'en-us'

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
```

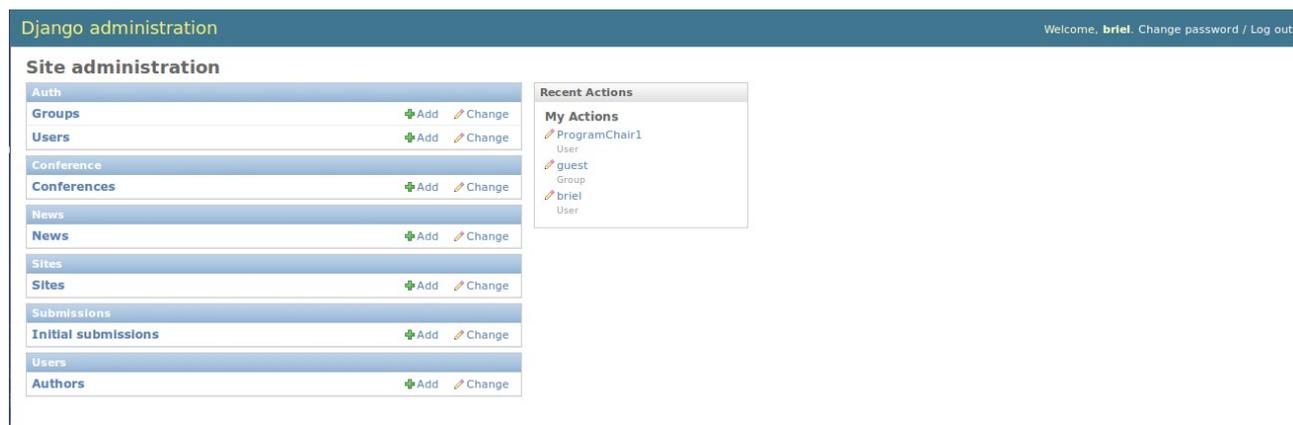
1,1 Comienzo

Se han instalado diferentes aplicaciones que nos permite utilizar muchas funcionalidades útiles para la aplicación, son:

- 'django.contrib.auth',
- 'django.contrib.contenttypes',
- 'django.contrib.sessions',
- 'django.contrib.sites',
- 'django.contrib.messages',
- 'django.contrib.staticfiles',

No sólo se ha instalado la aplicación de admin, sino que se ha configurado las diferentes aplicaciones para que el admin pueda acceder a los modelos y modificar la información sobre ellos, es muy útil ya que la aplicación admin es muy configurable y de una forma muy sencilla. Para que admin acceda a los modelos de las diferentes aplicaciones de la aplicación, se crea el archivo 'admin.py' en cada aplicación, en el que se debe indicar los modelos que van a ser accedidos por el admin y la forma de hacerlo.

A continuación se puede observar la página principal de la administración de la aplicación



2. Descripción del Módulo Conference

Conference es la aplicación que permite acceder y modificar la información del congreso, interactúa con la aplicación Users ya que hace uso de los roles de los usuarios para restringir el acceso a la modificación del congreso, además es la que gestiona el acceso a los usuarios, por tanto las funcionalidades principales son:

- Modificar información del congreso: permite únicamente al admin acceder a la modificación de los datos del congreso.
- Autenticación en el congreso (login-logout): es la que gestiona el acceso de los usuarios y modifica el rol del usuario.

Tiene el modelo de Conference, que tiene información del congreso, como nombre del mismo y detalles.

En la aplicación se puede acceder a diferentes funcionalidades dependiendo del rol que el usuario tenga en cada momento, por lo que el rol puede modificarse, esa modificación se realiza desde login, que comprueba si el usuario tiene el rol que el usuario desee adquirir y si lo tiene se lo asigna a la sesión que tiene.

Django facilita la autenticación de usuario dotando al framework de funcionalidades de autenticación, la aplicación que nos ofrece estas funcionalidades viene por defecto al crear la aplicación ('django.contrib.auth').

A continuación se puede observar el código fuente de las funciones de login y logout

```
def login_congress(request, group_id=0):
    conference = Conference.objects.all()[0]
    if request.user.is_authenticated():
        if len(request.user.groups.filter(id=group_id))>0:
            request.session['activegroup'] = request.user.groups.filter(id=group_id)[0]
            msg = "Profile changed to " + request.session['activegroup'].name
            return render_to_response('conference/myindex.html', {'conference': conference, 'msg' : msg }, context_instance=RequestContext(request))
        else:
            error_message = "User cannot access with selected profile"
            return render_to_response('conference/index.html', {'conference': conference, 'error_message' : error_message }, context_instance=RequestContext(request))
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(username=username, password=password)
    if user is not None:
        if user.is_active:
            login(request, user)
            msg = "Welcome"
            request.session['activegroup'] = user.groups.all()[0]
            return render_to_response('conference/myindex.html', {'conference': conference, 'msg' : msg }, context_instance=RequestContext(request))
        else:
            msg = "User cannot access"
            return render_to_response('conference/index.html', {'conference': conference, 'msg' : msg}, context_instance=RequestContext(request)) # Usuario no activo
    else:
        msg = "User cannot access"
        return render_to_response('conference/index.html', {'conference': conference, 'msg' : msg}, context_instance=RequestContext(request)) # Usuario no activo

def logout_congress(request):
    logout(request)
    return HttpResponseRedirect("/")
```

3. Descripción del Módulo de Gestión de Usuarios

Users es una aplicación para la gestión de usuarios y roles en la aplicación djcon, tiene los roles de djcon, los roles de la aplicación son:

- Author
- Reviewer
- Associate Editor
- Programm Chair
- Session Chair
- Registration Chair

Todos los usuarios de la aplicación tienen los atributos:

- username
- password
- name
- last name
- e-mail
- office address
- affiliation institution
- telephone
- fax
- keywords

Los autor tienen una serie de atributos propios:

- IEEE membership condition
- IEEE membership order
- IEEE member
- vip condition

La aplicación utiliza el modelo User, y Group de Django, para la creación de los usuarios de la aplicación, ya que el único atributo que diferencia a los usuarios es el tipo, excepto para el tipo de usuario author, los usuarios de la aplicación son instancias del modelo User de Django, relacionados con su grupo, el tipo de usuario author se crea mediante el modelo Author, que mantiene la información de los autores.

A continuación se muestra el código fuente de los modelos de la aplicación de gestión de usuarios, se puede observar que el único modelo es 'Author', eso es porque los diferentes usuarios de la aplicación se obtienen al importar el modelo 'User' de la aplicación 'django.contrib.auth':

```
from django.db import models
from django.contrib.auth.models import User, Group

class Author(models.Model):
    # Permisos de la clase author
    class Meta:
        permissions = (
            ("view_task", "Can see available tasks"),
            ("change_task_status", "Can change the status of tasks"),
            ("close_task", "Can remove a task by setting its status as closed"),
        )
    # Es un atributo obligatorio, no pueden existir autores sin usuario
    # Un autor esta relacionado unicamente con un usuario
    user = models.OneToOneField(User, null=False)

    ieee_membership_number = models.CharField(max_length=150)
    ieee_member_condition = models.CharField(max_length=150)
    ieee_itss_member_condition = models.CharField(max_length=150)
    vip_condition = models.CharField(max_length=150)

    def inicializar(self):
        self.user.groups.add(Group.objects.get(name='authors'))

    def __unicode__(self):
        return self.user.username
~
-
```

El tipo los usuarios tal como hemos indicado tienen datos comunes, algunos de estos atributos, no vienen por defecto en el modelo user de 'django.contrib.auth', por lo que se al crear la aplicación account amplía los datos del modelo 'User' y lo adapta a las necesidades propias.

Hay que tener en cuenta que los requisitos indican que no se permite que un usuario pueda tener varias instancias del mismo rol (ej. no puede ser dos author), aunque sí se permite que el mismo usuario tenga varios roles (puede ser author y reviewer).

Es un requisito de la aplicación, que se realicen una serie de comprobaciones en la creación del usuario para comprobar que no está en el sistema, como la verificación que el e-mail no está duplicado, y si tiene los mismos apellidos, si es así se advertirá al usuario que ya existe otro usuario con esos datos y se le preguntará si es él mismo, si responde afirmativamente, se le ofrecerá la posibilidad de proporcionarle otra pwd a su e-mail.

Va a utilizar la aplicación de Django auth, y la contenttypes, que ofrecen funcionalidades para la gestión de usuarios y permisos en la aplicación.

El uso de auth y contenttypes mejora la modularidad y la robustez de la aplicación, ya que son aplicaciones ampliamente probadas y usadas en muchos desarrollos.

La implementación de los roles que no tienen atributos propios se ha realizado usando los grupos de auth, por lo que si un usuario tiene el rol de reviewer, pertenece al grupo reviewers, los grupos se cargan inicialmente mediante una carga de datos en la instalación, almacenándose estos datos en initial_data.json.

ej. de initial_data.json

```
[
  {
    "model": "auth.Group",
    "pk": 1,
    "fields": {
      "name": "authors"
    }
  },
  {
    "model": "auth.Group",
    "pk": 2,
    "fields": {
      "name": "reviewers"
    }
  },
  {
    "model": "auth.Group",
    "pk": 3,
    "fields": {
      "name": "assosiate_editor"
    }
  },
  ...
]
```

Consecuencias en la modificación del rol de un usuario:

La modificación del rol de un usuario implica el acceso a las funcionalidades que sus roles actuales le permiten, las relaciones anteriores que permitía su anterior rol se mantienen (ej. un revisor que tenía review_templates al dejar de serlo sus reviews mantienen el usuario que las realizó).

4. Descripción del Módulo de Gestión de Artículos

Submissions es una aplicación que gestiona los artículos, por lo que interactúa con la aplicación Users, ya que tiene en cuenta los roles en la aplicación djcon, para restringir el acceso a ciertas funcionalidades, las funcionalidades son:

- Crear initial_submissions: las cuales sólo pueden ser creadas por algún author, el autor que la crea será el autor para correspondencia
- Ver submissions: sólo el author de la submission, el program chair, el editor al que se le ha asignado la initial_submission y el revisor al que se le ha asignado la initial_submission pueden ver la submission, los datos y el descargar el paper de la misma.
- Eliminar initial_submissions: sólo el author de la initial_submission y el program chair pueden eliminar initial_submission, el author, únicamente las suyas.
- Modificar initial_submissions: sólo el author de la corresponding author de la initial_submission puede modificarla
- Asignar una revisión de una submission: sólo el associate editor puede crear revisiones y sólo para las initial_submission que se las ha asignado el program chair
- Modificar o realizar una revisión: sólo el revisor al que se le ha asignado la

initial_submission puede realizar revisión sobre esta

- Revisar submissions: sólo el revisor de la submission puede realizar una revisión

Los modelos que necesita la aplicación son:

- Initial_Submission: que almacena la información de la submission inicial que el author envía, e información sobre su aceptación que el program chair añadirá.
- Final_Submission: que almacena la información de la initial_submission que sea aceptada
- Review: revisión que debe tener información acerca de la revisión, y está relacionada con la initial_submission a revisar y revisor
- Report: tiene los datos de la impresión que tiene el associate editor sobre la initial_submission, además de posibles comentarios hacia el author y program chair

Una tarea muy ardua en de la aplicación de gestión de artículos es la validación de formularios, ya que existentes muchos y son muchos campos que hay que validar, gracias a la validación de formularios de la que django dispone, se ha realizado de una forma mucho más limpia y sencilla, para tal tarea, django necesita una clase que implemente 'forms.Form' y en el controlador (view en django) obtendrá los campos del post y validará los validará tal como la clase implementada indique, los formularios también se envían a las plantillas permitiendo ser inicializados renderizándolos.

El siguiente código pertenece a varias clases que ayudan en la validación de formularios:

```
from django.http import HttpResponseRedirect
from django.core.urlresolvers import reverse
from django.shortcuts import render_to_response, get_object_or_404
from django.template import RequestContext
from django.contrib.auth.decorators import login_required, permission_required
from submissions.models import *
from conference.models import *
from users.models import *
from django import forms

REVIEW_STATUS_CHOICES = ( ('',''), ('acceptance', 'acceptance'), ('minor changes, acceptance', 'minor changes, acceptance'), ('mayor changes, acceptance', 'mayor changes, a
ceptance'), ('rejected', 'rejected') )

class ReviewerReportForm(forms.Form):
    #reviewers = forms.ModelMultipleChoiceField(queryset=User.objects.filter(groups=Group.objects.get(name='reviewer')), required=False, label='Reviewers (press control t
o multiple choice)')
    comments = forms.CharField(widget=forms.Textarea, max_length=10000, required=False)
    status_comments = forms.CharField(widget=forms.Textarea, max_length=10000, required=False, label='Confidential Comments (not visible to author)') # comments to progra
m chair
    status = forms.ChoiceField(required=False, choices=REVIEW_STATUS_CHOICES)

class ProgramChairForm(forms.Form):
    editor = forms.ModelChoiceField(queryset=User.objects.filter(groups=Group.objects.get(name='associate editor')), required=True)#, empty_label=None)
    final_status = forms.ChoiceField(required=False, choices=REVIEW_STATUS_CHOICES)
    comments_to_author = forms.CharField(widget=forms.Textarea, max_length=10000, required=False, label='Comments to author')

class ReviewerForm(forms.Form):
    score = forms.ChoiceField(required=False, choices=SCORE)
    status = forms.ChoiceField(required=False, choices=REVIEW_STATUS_CHOICES)
    comments = forms.CharField(widget=forms.Textarea, max_length=10000, required=False)

class AuthorSubmissionForm(forms.Form):
    title = forms.CharField(max_length=100, required=True)
    short_title = forms.CharField(max_length=100, required=True)
    abstract = forms.CharField(widget=forms.Textarea, max_length=10000, required=False)
    #status = forms.ChoiceField(required=False, choices=AUTHOR_STATUS_CHOICES)
    authors = forms.ModelMultipleChoiceField(queryset=Author.objects.all(), required=False, label='Authors (press control to multiple choice)')
    paper = forms.FileField(required=False)

class FinalSubmissionForm(forms.Form):
    title = forms.CharField(max_length=100, required=True)
    short_title = forms.CharField(max_length=100, required=True)
```

1,1 Comienzo

Un ejemplo en el que se ve la forma en la que se obtienen los datos del formularios y se renderiza se puede observar en el código del controlador de 'author_update_submission', el cual podemos observar a continuación:

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

```
@login_required(login_url='/')
def author_update_submission(request):
    conference = Conference.objects.all()[0]
    author_submission_form = AuthorSubmissionForm(request.POST, request.FILES)
    submission = InitialSubmission.objects.get(pk=request.POST['submission_id'])
    if author_submission_form.is_valid():
        submission.title = author_submission_form.cleaned_data['title']
        submission.short_title = author_submission_form.cleaned_data['short_title']
        submission.abstract = author_submission_form.cleaned_data['abstract']
        if request.POST['status'].lower() == 'draft':
            submission.status = 'draft'
        else:
            submission.status = 'sent'
        submission.corresponding_author = Author.objects.get(user=request.user)
        submission.authors = author_submission_form.cleaned_data['authors']
        if author_submission_form.cleaned_data['paper'] != None:
            submission.paper = author_submission_form.cleaned_data['paper']
        submission.save()
    else:
        return render_to_response('submissions/author_submission.html', {'submission': submission, 'conference': conference, 'submission_form': author_submission_form },
        context_instance=RequestContext(request))
    return HttpResponseRedirect("/submissions")
```

Manual de usuario y software

El código fuente de la aplicación se encuentra en el disco adjunto, el cual está documentado.

El manual de usuario de la aplicación se encuentra en el anexo de la memoria.

Conclusiones y líneas futuras

La elaboración del trabajo de fin de grado ha ayudado a tener una visión práctica del desarrollo de una aplicación web, pues nos da un punto de vista práctico al realizar la aplicación desde cero, tarea que ha permitido observar todo el ciclo de vida del desarrollo de la aplicación, adquiriendo la experiencia que implica, realizar todas las tareas necesarias en la creación de la aplicación, desde la elección del hardware y software necesario, junto a las tecnologías con las que se va a desarrollar el proyecto, el análisis de requisitos de usuario, la elección de metodología de trabajo y demás pasos necesarios hasta la implementación de la aplicación junto a las pruebas de la misma.

El desarrollo de una aplicación que gestione las ponencias de congresos internacionales ha sido una gran idea, ya que en el mercado no existen suficientes aplicaciones que implementen las exigencias de un congreso de forma completa, por lo que al necesitar gestionar multitud de actividades, las gestionan dependiendo de la funcionalidad concreta que requieren, por lo que muchos de los requisitos se encuentran desarrollados de forma independiente, no existiendo aplicaciones que integren la totalidad de las necesidades que un congreso debe gestionar, por lo que esta aplicación de gestión de congresos internacionales puede servir de base de una aplicación mayor que implemente las ampliaciones descritas en las líneas futuras.

El desarrollo de la aplicación se ha apoyado en el uso de metodologías ágiles de desarrollo, aunque inicialmente se ha realizado un minucioso análisis de los requisitos de usuario, donde una vez tenido claro cuales eran las necesidades del cliente, y de la aplicación, se han utilizado estas metodologías, realizando entregas a medida que se iban realizando diferentes funcionalidades, y observando si las cuales, implementadas en cada entrega eran las adecuadas y si la dirección que iba tomando el desarrollo del proyecto era el correcto, permitiendo planificar el desarrollo, ajustándolo a lo que se iba comprobando, y realizando adaptaciones siempre que fuera necesario.

A medida que se iba desarrollando la aplicación, ampliando el número de entregas, recortando el tiempo entre ellas, se ha visto más útil, ya que al aumentar la comunicación con el cliente, se ha conocido mejor sus necesidades.

Las herramientas tanto de software como de hardware elegidas se ajustaron a las necesidades de desarrollo de la aplicación, y están descritas en requisitos de hardware y software, habiendo escogido en todo momento herramientas software que fueran libres, desde el sistema operativo utilizado en el desarrollo de la aplicación, al entorno de desarrollo, hasta el editor de textos utilizado en la elaboración de la memoria.

Conocer inicialmente el tipo de aplicación a desarrollar: una aplicación web accedida por navegadores web con la capacidad de html5, css3 y JavaScript. Elegir que la lógica de la aplicación estuviera principalmente en el servidor y el patrón de desarrollo modelo vista controlador permitió dirigir el desarrollo de la aplicación en la dirección tomada, eligiendo django por las razones descritas en el apartado de tecnologías en el lado del servidor descrito en la sección de Análisis.

Una vez finalizada la implementación de la aplicación se ha comprobado que la elección del framework django fue muy acertada. Ha permitido conocer una tecnología que actualmente está en auge, su uso además ha implicado conocer de la comunidad de desarrolladores, una filosofía de

desarrollo muy interesante, aunque inicialmente ha supuesto un coste de tiempo importante encontrar los sitios y foros en los que buscar la información que se iba necesitando, y la forma de solucionar los problemas que iban surgiendo a medida que se iba progresando en la elaboración de la aplicación.

El uso del framework de desarrollo django ha ayudado mucho en la modularización de la aplicación, ya que django, ofrece por defecto una normalización en la organización del código, proporcionando una estructura muy sencilla y práctica, y que siguiendo esa estructura estandarizada, que django posee, se ha utilizado facilitando la implementación de la aplicación, al estructurarla según los cánones del framework.

Se ha utilizado características de django que permiten mejorar la seguridad de la aplicación, desde el uso de los módulos de autenticación, hasta el módulo que nos permite eliminar la vulnerabilidad de secuencias de comandos en sitios cruzados, el uso de estas características en django se realiza de una forma muy sencilla.

El tratamiento de base de datos en el desarrollo de la aplicación ha sido realizado atendiendo principalmente a la manera en la que django trabaja por defecto, utilizando las características de las que dispone, tal como la configuración de la conexión de la aplicación a la base de datos, que se realiza de una forma muy sencilla, la creación de las estructuras relacionales a partir de los modelos creados en cada aplicación, evitándonos la tarea de traducción de los modelos a tablas, la validación de los campos y el uso de los métodos que nos ofrecen los modelos para el acceso a los datos, haciendo innecesario realizar consultas manualmente. Todo esto ha facilitado mucho la labor. Además nos permite en un futuro migrar a cualquier sistema de gestión de base de datos en el que queramos alojar la aplicación en producción de forma muy cómoda.

La implementación de la interfaz de la aplicación apoyándose en la tecnología de html5, junto con css3 y JavaScript permite que la aplicación sea fácilmente personalizable para diferentes plataformas, como ordenadores de sobremesa, portátiles, o móviles, que dispongan sistemas operativos varios, además sin realizar ningún tipo de instalación de software adicional del que por defecto dispone.

El uso de otras tecnologías como flash nos hubiera requerido la instalación de software adicional, y no nos da ningún tipo de funcionalidad añadida que requiramos en la aplicación desarrollada.

El uso de JavaScript ha sido muy útil ampliando características de controles de html, para lo que se ha utilizado una librería que amplía las funcionalidades del control de selección múltiple del que dispone html.

El manual de usuario realizado contempla las características de la aplicación e indica la forma en la que cada tipo de usuario trabaja, desde el director de programa, hasta un autor, describiendo las acciones que su rol tiene asignado.

En la realización de un proyecto de este estilo para el trabajo de fin de grado del curso de adaptación a grado de ingeniería informática, habría que tener los conocimientos de los lenguajes en los que se va a desarrollar y en el framework que se utilizará, ya que el desarrollo de este proyecto implica una inversión de horas para dominar los lenguajes y frameworks, puesto que sin tenerlos, sería imposible la realización del mismo en el número de horas que el trabajo de fin de grado exige.

El prototipo de la aplicación está completo, al estar modulada según los estándares de django, y estructurado el código siguiendo los cánones del mismo, resulta muy sencillo añadir módulos y funcionalidades adicionales. Entre las funcionalidades que se podrían implementar ampliando la actual aplicación estarían las siguientes:

- **Módulo de Mensajería**
En el que los diferentes usuarios pudieran comunicarse mediante mensajería interna, también podrían enviarse correos electrónicos, e incluso comunicarse de forma síncrona.
- **Módulo de Cumplimiento de Formato**
Que comprobara el formato en el que se encuentran los artículos y que indicara si cumple o no en el que se deben enviar los artículos, indicando los posibles errores que existieran si no se cumpliera el formato que el congreso exigiera, pudiendo incluso personalizar las exigencias de formato de cumplimiento que tuviera el congreso concreto para el que se utilizara la aplicación.
- **Módulo de Planificación**
Que permitiera organizar el congreso, indicando los posibles conflictos en las exposiciones de los artículos. Habría que indicar el tiempo de cada exposición ya que el tiempo del congreso es limitado, habría que tener en cuenta las localizaciones de las que dispone el congreso, indicando las salas, las necesidades de cada exposición, al aforo al que va destinado, las capacidades de las presentaciones individuales, preferencias de cada autor sobre la sesión en la que se va a celebrar, las prioridades del congreso respecto a las exposiciones dependiendo del artículo, y que el módulo indique posibles planificaciones.
- **Módulo de Comprobación de Plagio**
La necesidad de la comprobación del plagio de los artículos presentados a un congreso es habitual, por lo que el desarrollo de un módulo que facilite la gestión de la comprobación de los artículos facilitaría la tarea, siendo de ayuda en muchos congresos.
- **Módulo de Registro y Facturación**
En los congresos se realizan tareas de registro y facturación, por lo que la creación de un módulo que amplíe las características de la aplicación sería de gran ayuda.
El sistema podría manejar un conjunto de diferentes categorías de registro: condiciones de registro, elementos de registro, registro completo, registro de página adicional, entrada con comida, entrada con servicios extra, cuota reducida de inscripción, matrícula estudiantil, etc., además la facturación del mismo también se podría realizar de forma automatizada, realizando el pago online, mediante tarjeta de crédito o paypal.
Ejemplo de tareas que podría realizar este módulo serían:
 - El sistema de registro podría capturar la información de tarjeta de crédito y la información de facturación.
 - El sistema podría generar automáticamente un recibo pdf y enviarlo cuando una persona se registra.
- **Módulo de Ayuda o tutorial**
Este módulo podría estar basado en los roles, en donde a cada usuario, dependiendo de su tipo, se le indicara la forma en la que realizar su tarea según su rol, para lo cual se le indicara las funciones propias en el congreso dentro del proceso del mismo, además podría señalarse el flujo de trabajo que debe realizar.
- **Módulo de Gestor de Contenidos ampliado**
El congreso necesita un portal donde publicitarse, mostrar la información del mismo tanto a autores, como a interesados en el mismo. Ya que la mayoría de los congresos tienen estas mismas necesidades, sería conveniente que se realizara un gestor de contenidos enfocado a

congresos, en el que se añadiera la información del mismo y permitiera elegir entre diferentes visualizaciones de un congreso. Sería interesante enlazar la información del congreso con las distintas redes sociales, por lo que también podría tener la posibilidad de relacionar el portal con las redes sociales más usadas en estos momentos, facebook, twitter, google+.

- Módulo de Sistema de Log de Acciones de los usuarios
Puede ser interesante almacenar las acciones que realizan los diferentes usuarios, de esta manera, podríamos acceder a esa información, o al existir cualquier problema, seríamos capaces de dejar el sistema en el estado que anteriormente se encontraba, permitiendo restaurar dicho sistema.

Anexos

Bibliografía

<http://www.python.org/doc/>
<http://www.python.org/>
<https://www.djangoproject.com/>
<https://docs.djangoproject.com/en/1.5/>
<http://960.gs/>
<http://jquery.com/>
<http://ivaynberg.github.io/select2/>
<http://librosweb.es/xhtml/>
<http://librosweb.es/css/>
<http://librosweb.es/javascript/>

Glosario

Autor (*author*): Produce una publicación de algún tipo, la cual desea sea incluida en el Programa Científico de la Conferencia.

Artículo (*submission*): Normalmente irá acompañada de un artículo (*paper*) y es enviado y gestionado por el autor para correspondencia (*corresponding author*), inicialmente se envía un artículo inicial y una vez aceptado, se enviará el final. produciendo una presentación (*communication*) o la exposición oral, durante el desarrollo de la conferencia.

Artículo inicial (*Initial Submission*): El artículo que inicialmente se envía al sistema implementado para su evaluación por los colaboradores.

Informe de revisiones (*Reviewers Report*): Producido por los Editores Asociados, que compila las calificaciones y comentarios de las opiniones (*review templates*), además de los comentarios hacia el director de programa (*program chair*) y la evaluación del editor asociado.

Plantilla de la opinión (*review template*): Un formulario que permite a los revisores anotar observaciones del artículo específico ayudando a los editores asociados a en la valoración del artículo y sobre su aceptación en la conferencia.

Autor para correspondencia (*corresponding author*): Cada artículo tiene asociado un autor para correspondencia, el cual es uno de los autores, y que será la misma persona a la que se dirigen las comunicaciones relativas a ese artículo específico. En otras palabras, es la persona de contacto del artículo (*submission*).

Palabra clave (*keyword*): Una palabra o conjunto de palabras que expresan un tema específico dentro del alcance de la conferencia organizada. Pueden ser utilizadas de etiquetas de indexación/categorización del artículo, los revisores, las sesiones, los talleres, los tutoriales, ...

Sesión: Cada parte de tiempo ininterrumpida durante el desarrollo de una conferencia, donde está previsto que haya una serie de comunicaciones que se expondrán, comentarán, ... Se espera que todos los envíos programados dentro de una sesión que se agrupen en un conjunto de palabras clave. Todos los artículos programados en sesión son el resultado filtrado de un proceso de revisión desarrollado antes de la conferencia.

Director de Programa (*program chair*): uno o más usuarios incluidos en el comité organizador de la conferencia a cargo del desarrollo de todo el programa científico de la reunión.

Editor Asociado: Un usuario que está de acuerdo en coordinar el proceso de revisión de una serie de artículos.

Director/Codirector de Sesión (*session chair/cochair*): Un usuario que se compromete a llevar a cabo una sesión específica, presentando a los participantes, la promoción de preguntas y debate útil, controlar el tiempo y las condiciones adecuadas para la sesión, etc

Director de inscripciones (*registration chair*): Uno o más usuarios incluidos en el comité organizador de la conferencia, a cargo del registro de participantes, la generación de la documentación, certificados, insignias

Categoría de inscripción: El sistema debe manejar un conjunto de diferentes categorías de registro,

dependiendo de registro temprano o tarde, la condición de registro, los tipos de registro (registro completo, registro de página adicional, el registro en papel, pase de comida, pase de comida extra, reducción de cuota de inscripción, tasas de matrícula estudiantil, etc)

Usuario (*user*): Una persona con derecho a actuar con uno o más de los roles definidos para el sistema.

Revisor (*reviewer*): un usuario que se compromete a revisar una serie de artículos, valorándolos de acuerdo a una plantilla, incluyendo ideas de mejora y una recomendación de decisión ([Aceptada (*accepted*) | Aceptada con cambios menores (*accepted with minor changes*)| Aceptada con modificaciones principales (*accepted with mayor changes*) | Rechazada (*rejected*)]).

Director de programa (*PC program chair*): uno o más usuarios de la conferencia del comité organizador encargado de desarrollar todo el programa científico de la reunión.

Director de inscripciones (*RC - registration chair*): uno o más participantes del comité organizador de la conferencia a cargo de la inscripción de los participantes, la producción de documentos de registro/certificados/insignias

Director/Codirector de sesión (*SC/ScC - session chair/co-chair*): un usuario que se compromete a llevar a cabo una sesión específica, introduciendo a los participantes, la promoción de preguntas y debate útil, controlar el tiempo y las condiciones adecuadas para la sesión, etc

Administrador (*admin*): Uno o varios usuarios del comité organizador de la conferencia que tienen una vista completa del Sistema de Información de la Conferencia. Pueden cambiar los parámetros de la configuración de la conferencia.

El patrón modelo vista controlador

El patrón modelo vista controlador ha sido desde el primer momento del desarrollo tenido en cuenta en el desarrollo del proyecto, desde el diseño hasta la implementación de la aplicación, ha sido el patrón utilizado en la arquitectura de la aplicación, la explicación del mismo está descrita a continuación:

El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Historia

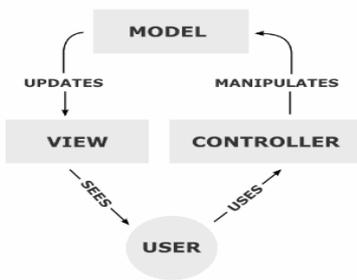
El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones.

MVC fue introducido por Trygve Reenskaug en Smalltalk-76 durante su visita a Xerox Parc en los años 70 y, seguidamente, en los años 80, Jim Althoff y otros implementaron una versión de MVC para la biblioteca de clases de Smalltalk-80. Sólo más tarde, en 1988, MVC se expresó como un concepto general en un artículo.

En esta primera definición de MVC el **controlador** se definía como "*el módulo que se ocupa de la entrada*" (de forma similar a como la **vista** "*se ocupa de la salida*"). Esta definición no tiene cabida en las aplicaciones modernas en las que esta funcionalidad es asumida por una combinación de la 'vista' y algún framework moderno para desarrollo. El 'controlador', en las aplicaciones modernas de la década de 2000, es un módulo o una sección intermedia de código, que hace de intermediario de la comunicación entre el 'modelo' y la 'vista', y unifica la validación (utilizando llamadas directas o el "*observer*" para desacoplar el 'modelo' de la 'vista' en el 'modelo' activo).

Descripción del patrón

Una típica colaboración entre los componentes de un MVC es la siguiente:



De manera genérica, los componentes de MVC se podrían definir como sigue:

- El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- La **Vista**: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

Interacción de los componentes

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el

modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. *Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro*

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

MVC y bases de datos

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre *capa de presentación* y *capa de negocio* conocido en el paradigma de la Programación por capas representaría la integración entre la **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la *capa visual gráfica* de su correspondiente *programación y acceso a datos*, algo que mejora el desarrollo y mantenimiento de la *Vista* y el *Controlador* en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Manual de Usuario

Este manual explica gráficamente las funciones de cada uno de los tipos de usuarios de la aplicación desarrollada.

Inicialmente se muestra la página inicial de la aplicación y posteriormente se van explicando cada una de las acciones que realiza cada tipo de usuario, desde la creación de los mismos.

Página principal de la aplicación



Creación de nueva cuenta

En la página de la conferencia, clicamos sobre “New Account”



Rellenamos el siguiente formulario y rellenamos los campos que nos indica, hay que tener en cuenta que el nombre de usuario no puede repetirse en el sistema, por lo que si por casualidad introducimos uno existente, nos lo indica, también comprueba si las contraseñas coinciden, y si el formato del e-mail es correcto. Tenemos que tener en cuenta que si nos interesa introducir un keyword o affiliation institution que no se encuentra en la lista, lo podemos añadir escribiendo el valor e introduciendo enter.



Si el usuario es autor, debe indicar que va a presentar alguna presentación, si no lo hace en el momento de creación de la cuenta, lo puede realizar posteriormente al editar su perfil

localhost:8000/create_account/#

Una vez ya hemos creado la cuenta de usuario, podemos acceder a la aplicación, introduciendo en el cuadro superior derecho de la aplicación el nombre de usuario y contraseña

usuario|

Login New Account

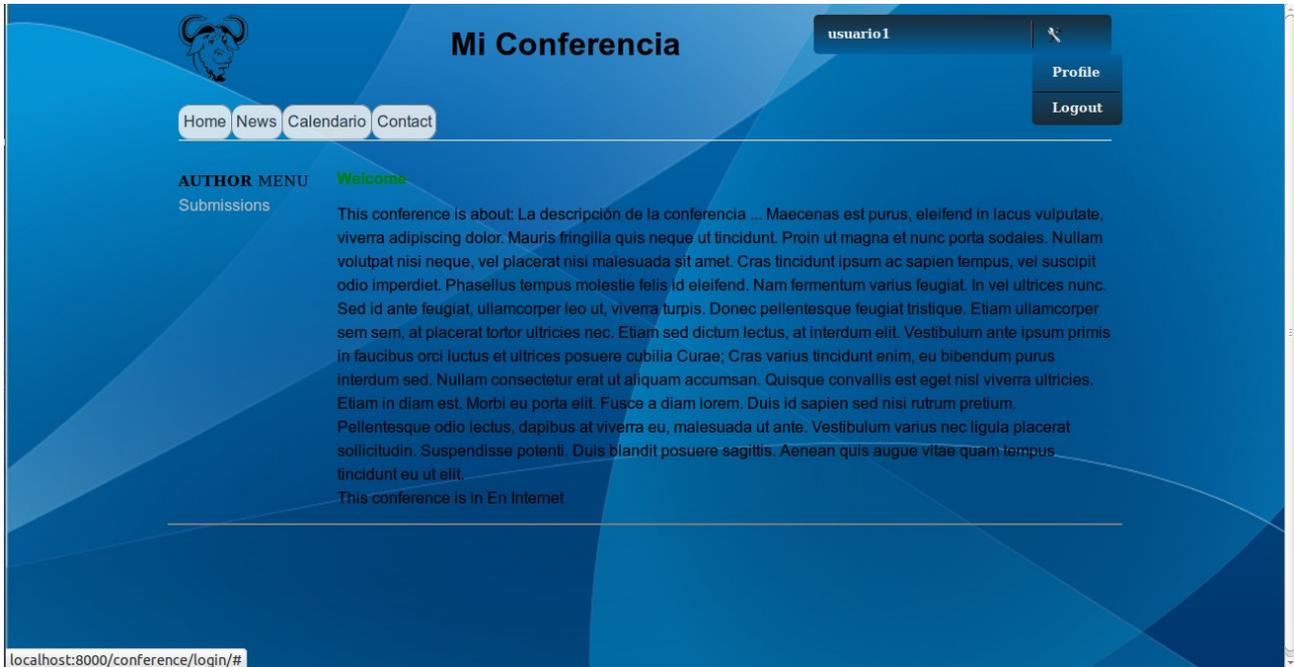
Home News Calendario Contact

La descripción de la conferencia ... Maecenas est purus, eleifend in lacus vulputate, viverra adipiscing dolor. Mauris fringilla quis neque ut tincidunt. Proin ut magna et nunc porta sodales. Nullam volutpat nisi neque, vel placerat nisi malesuada sit amet. Cras tincidunt ipsum ac sapien tempus, vel suscipit odio imperdiet. Phasellus tempus molestie felis id eleifend, Nam fermentum varius feugiat. In vel ultrices nunc. Sed id ante feugiat, ullamcorper leo ut, viverra turpis. Donec pellentesque feugiat tristique. Etiam ullamcorper sem sem, at placerat tortor ultricies nec. Etiam sed dictum lectus, at interdum elit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Cras varius tincidunt enim, eu bibendum purus interdum sed. Nullam consectetur erat ut aliquam accumsan. Quisque convallis est eget nisi viverra ultricies. Etiam in diam est. Morbi eu porta elit. Fusce a diam lorem. Duis id sapien sed nisi rutrum pretium. Pellentesque odio lectus, dapibus at viverra eu, malesuada ut ante. Vestibulum varius nec ligula placerat sollicitudin. Suspendisse potenti. Duis blandit posuere sagittis. Aenean quis augue vitae quam tempus tincidunt eu ut elit.

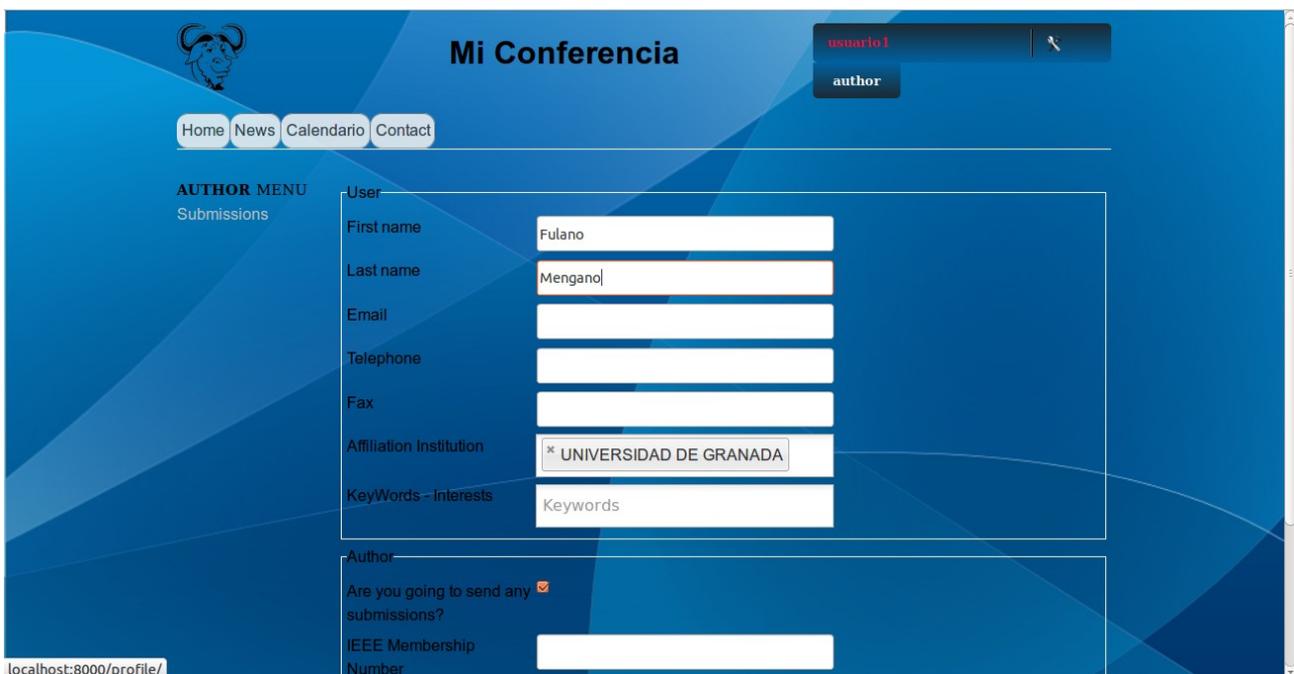
En Internet

JORNADAS TECNOLÓGICAS DE LAS PALMAS
DÍAS 3 Y 4 DE FEBRERO DE 2011

Si queremos modificación los datos de la cuenta, podemos situarnos sobre el icono de herramientas que se encuentra a la derecha del nombre de usuario que está en la esquina superior derecha y al expandir la información de profile, clicamos sobre ella, también podemos clicar sobre el nombre de usuario y accederemos también a los datos del perfil



Una vez accedemos a los datos del perfil, podemos modificarlos, aquí también podemos indicar si el usuario es autor seleccionando que el usuario va a enviar presentaciones



La cuenta de autor

Listado de presentaciones

Para ver las presentaciones que hemos creado, los borradores de las presentaciones que enviaremos, el estado de las presentaciones enviadas y las presentaciones finales, accedemos con el perfil de autor y en el menú que se encuentra en la barra izquierda, seleccionamos “Submissions”



La creación de una submission se realiza seleccionando “new submission” que está en la parte superior de los listados de submissions



Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

En la introducción de datos, hay que tener en cuenta que una presentación puede tener múltiples autores, los seleccionamos clicando sobre ellos mientras apretamos la tecla de “control”, si deseamos introducir un usuario que no se encuentra en el listado, hay que crear la cuenta de la forma antes indicada, tras introducir los datos de la submission podemos elegir si enviarla o crear un borrador, las opciones están tras los campos de los datos “Draft” y “Send”



The screenshot shows the 'Mi Conferencia' submission form. At the top right, the user is logged in as 'usuario 1'. The navigation menu includes 'Home', 'News', 'Calendario', and 'Contact'. The 'AUTHOR MENU' is on the left, with 'Submissions' selected. The form fields are:

- Title:** Mi presentación
- Short title:** Presentación
- Abstract:** Maesenas est purus, eleifend in lacus vulputate, viverra adipiscing dolor. Mauris fringilla quis neque ut tincidunt. Proin ut magna et nunc porta sodales. Nullam vulputat nisi neque, vel placerat nisi malesuada sit amet. Cras tincidunt ipsum ac sapien tempus, vel suscipit odio imperdiet. Phasellus tempus molestie felis id eleifend. Nam fermentum varius feugiat. In vel ultrices nunc. Sed id ante feugiat, ullamcorper leo ut, viverra turpis. Donec pellentesque feugiat tristique. Etiam ullamcorper sem sem, at placerat tortor ultricies nec. Etiam sed dictum lectus, at interdum elit. Vestibulum ante ipsum eribus in faucibus orci luctus et ultrices posuere cubilia Curae; Cras varius tincidunt enim, eu bibendum purus interdum sed. Nullam consectetur erat ut aliquam accumsan. Quisque conwallis est eget.
- Authors (press control to multiple choice):** user, author, briel, usuario 1
- Paper:** Examinar... No se ha seleccionado ningún

Buttons for 'Draft' and 'Send' are at the bottom.

Si queremos enviar un borrador, en el listado de submission, al que accederemos clicando sobre “Submissions” en el menu de autor que se encuentra a la izquierda, clicamos sobre editar.

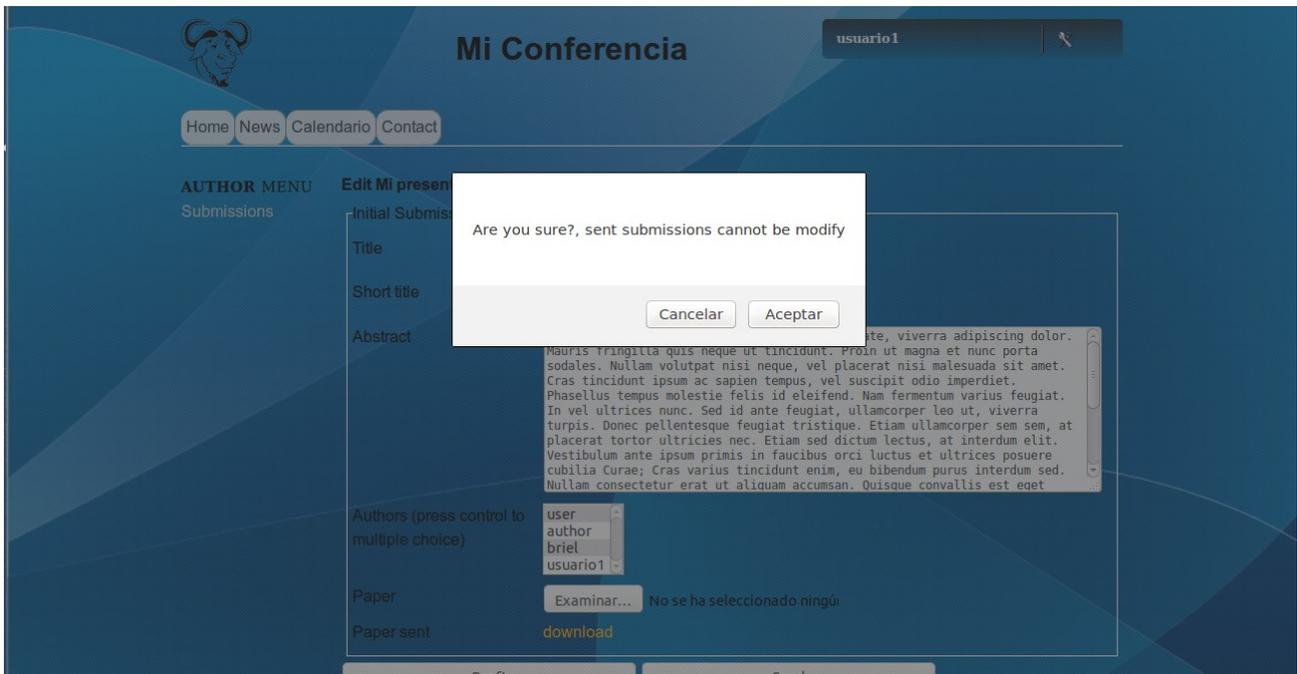


The screenshot shows the 'Mi Conferencia' submission list. At the top right, the user is logged in as 'usuario 1'. The navigation menu includes 'Home', 'News', 'Calendario', and 'Contact'. The 'AUTHOR MENU' is on the left, with 'Submissions' selected. The list shows:

- new submission**
- Initial Submissions**
- Submission ID** | **Submission Status**
- Any submission sent
- Draft Submissions**
- Submission Short Title**
- Presentación | [edit](#) [remove](#)

The browser address bar shows 'localhost:8000/submissions/edit/3'.

Tras seleccionar editar el borrador, en vez de almacenarlo clicando sobre “Draft”, podemos enviarla clicando sobre “Send”



Una vez hemos enviado una presentación, esperaremos a conocer la evaluación de la misma, para saberlo, en el listado de presentaciones enviadas, comprobaremos “status”, cuando la presentación sera corregida, pasará el estado “Reviewing” en revisión a “Accepted”, aceptada o “Rejected”, rechazada

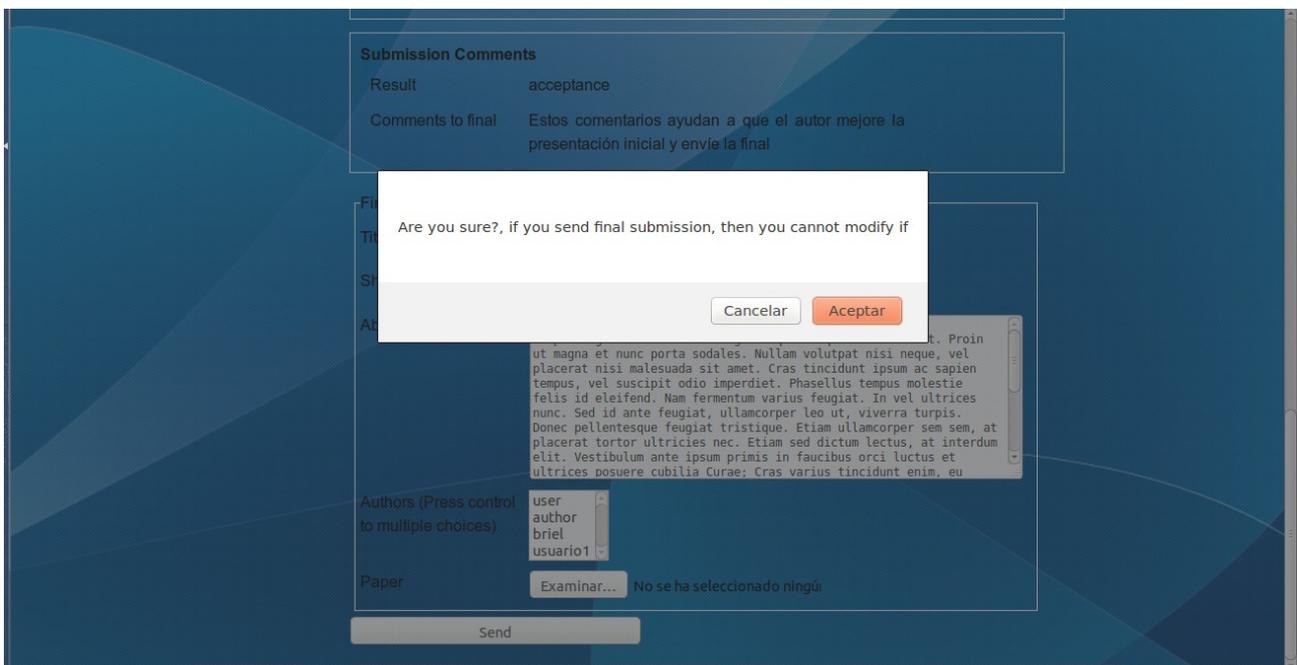


Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

Enviar una presentación final, se realizará cuando el estado de la presentación sea aceptada, comprobaremos que aparecerá la opción “send final” enviar presentación final: -----



Al seleccionar la presentación final, comprobaremos los comentarios que se han realizado sobre la presentación para mejorarla y enviar la presentación final, comprobamos que los datos previamente introducidos de la presentación final están introducidos para facilitar la introducción de datos



También podremos comprobar la presentaciones finales que hayamos enviado, y si queremos commprobar la información de las mismas, clicar sobre “show” de la misma



La cuenta del program chair

El program chair puede comprobar las presentaciones iniciales existentes, su estado, ver los datos, asignar las presentaciones a los editores asociados, aceptar las presentaciones, introducir comentarios de las presentaciones para que el autor de las mismas, listar las presentaciones finales, sus datos.

Para ver el listado de presentaciones iniciales y finales accede con el perfil program chair y selecciona “Submissions” del menú de la izquierda



Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

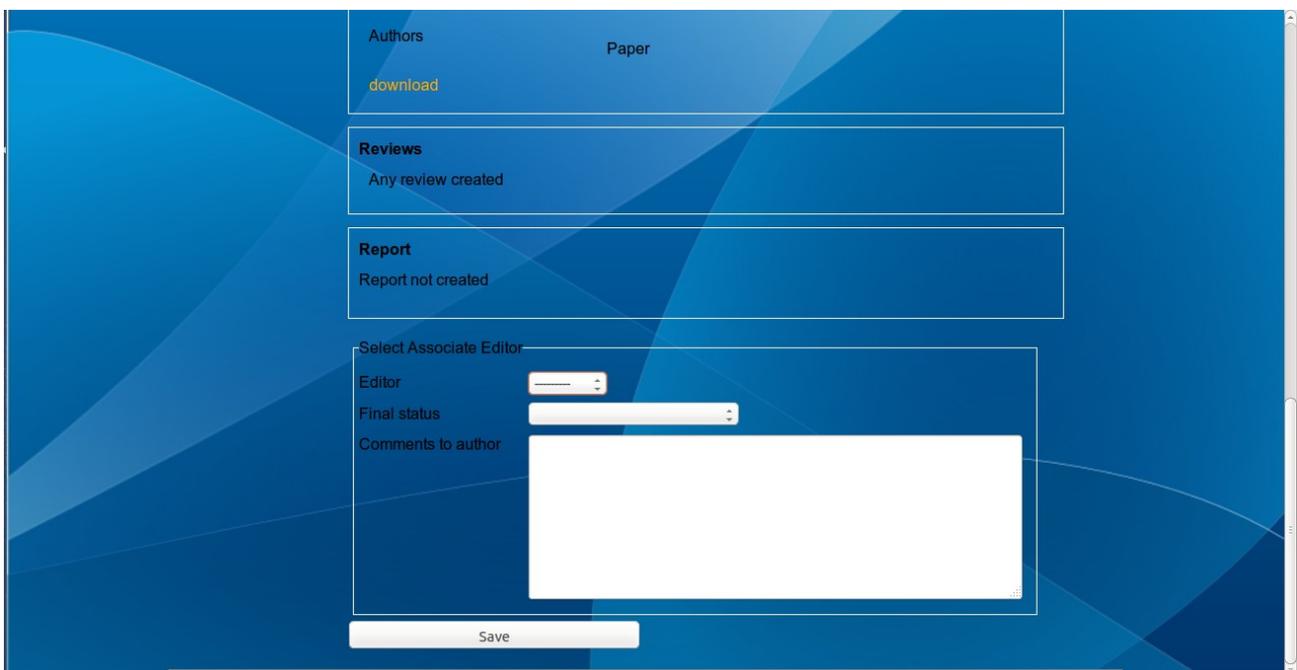
Para asignar la presentación al editor asociado, se debe seleccionar “review” del listado de presentaciones



The screenshot shows the 'Mi Conferencia' web application interface. The header includes a logo of a bull, the title 'Mi Conferencia', and a search bar with the text 'briel'. Below the header is a navigation menu with links for 'Home', 'News', 'Calendario', and 'Contact'. The main content area is divided into two sections: 'Final Submissions' and 'Initial Submissions'. The 'Final Submissions' section contains a table with one row: Submission ID 1, Submission 'final', and a 'show remove' link. The 'Initial Submissions' section contains a table with three rows: Submission ID 1 (Final Submission, Reviewing, show remove review), Submission ID 2 (final, acceptance, show remove review), and Submission ID 3 (Presentación, Reviewing, show remove review). The browser address bar shows 'localhost:8000/submissions/'.

Submission ID	Submission	Status	Actions
1	final		show remove
1	Final Submission	Reviewing	show remove review
2	final	acceptance	show remove review
3	Presentación	Reviewing	show remove review

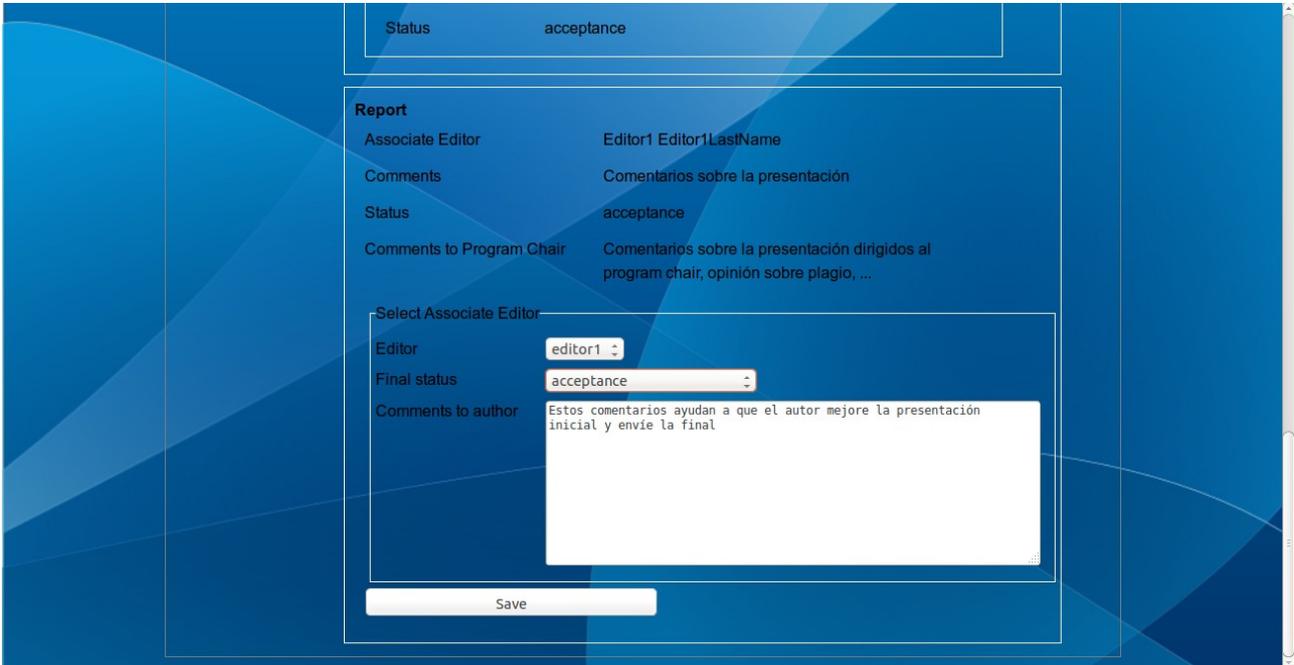
Una vez hemos seleccionado la presentación inicial, elegimos el editor que va a dar su opinión de la presentación ayudado de los revisores



The screenshot shows the 'Select Associate Editor' form in the web application. The form is divided into several sections: 'Authors' with a 'Paper' label and a 'download' link; 'Reviews' with the text 'Any review created'; 'Report' with the text 'Report not created'; and 'Select Associate Editor' which includes a dropdown menu for 'Editor', a dropdown menu for 'Final status', and a text area for 'Comments to author'. A 'Save' button is located at the bottom of the form.

Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

Cuando se haya realizado una valoración de la presentación se aceptará o rechazará, apoyándose de los informes del editor asociado y los revisores, que aparecen tras la información de la presentación, y puede introducir información al autor sobre la presentación inicial, dando información sobre lo que debe mejorar en la presentación final

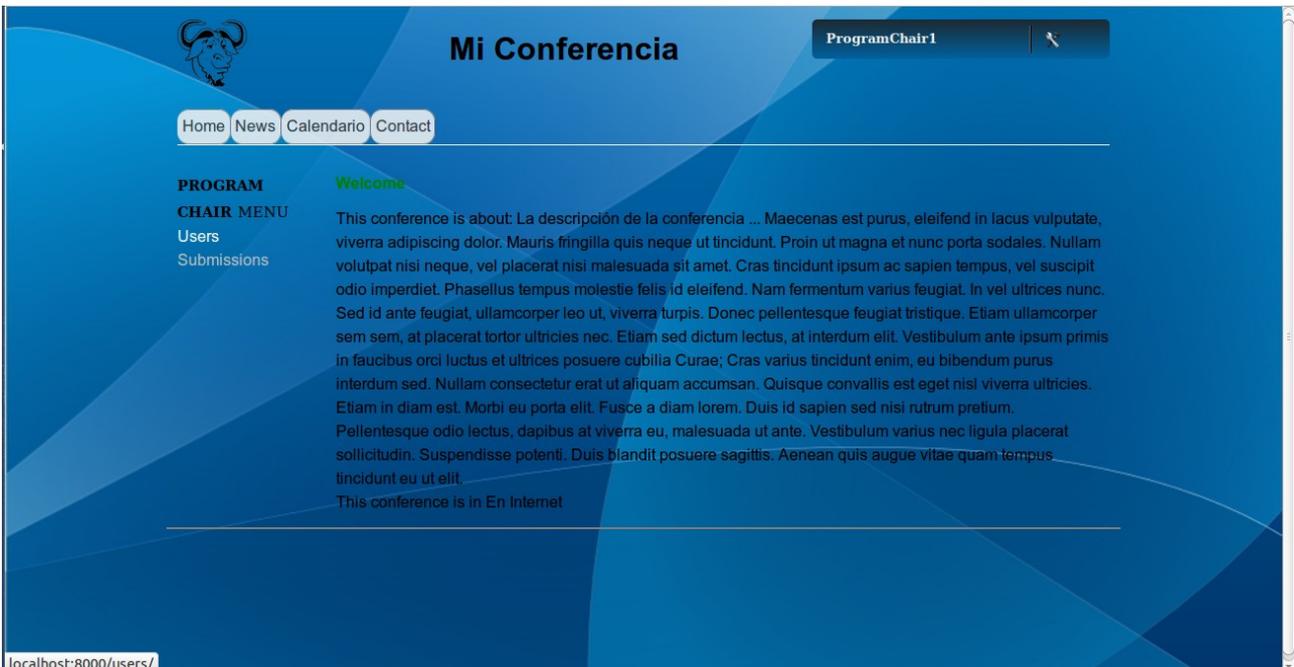


The screenshot shows a web form with a blue background. At the top, there is a 'Status' field with the value 'acceptance'. Below it is a 'Report' section with the following fields:

- Associate Editor: Editor1 Editor1LastName
- Comments: Comentarios sobre la presentación
- Status: acceptance
- Comments to Program Chair: Comentarios sobre la presentación dirigidos al program chair, opinión sobre plagio, ...

Below the report section is a 'Select Associate Editor' section with a dropdown menu showing 'editor1'. Below that is a 'Final status' dropdown menu showing 'acceptance'. At the bottom of the form is a 'Comments to author' text area containing the text: 'Estos comentarios ayudan a que el autor mejore la presentación inicial y envíe la final'. A 'Save' button is located at the bottom of the form.

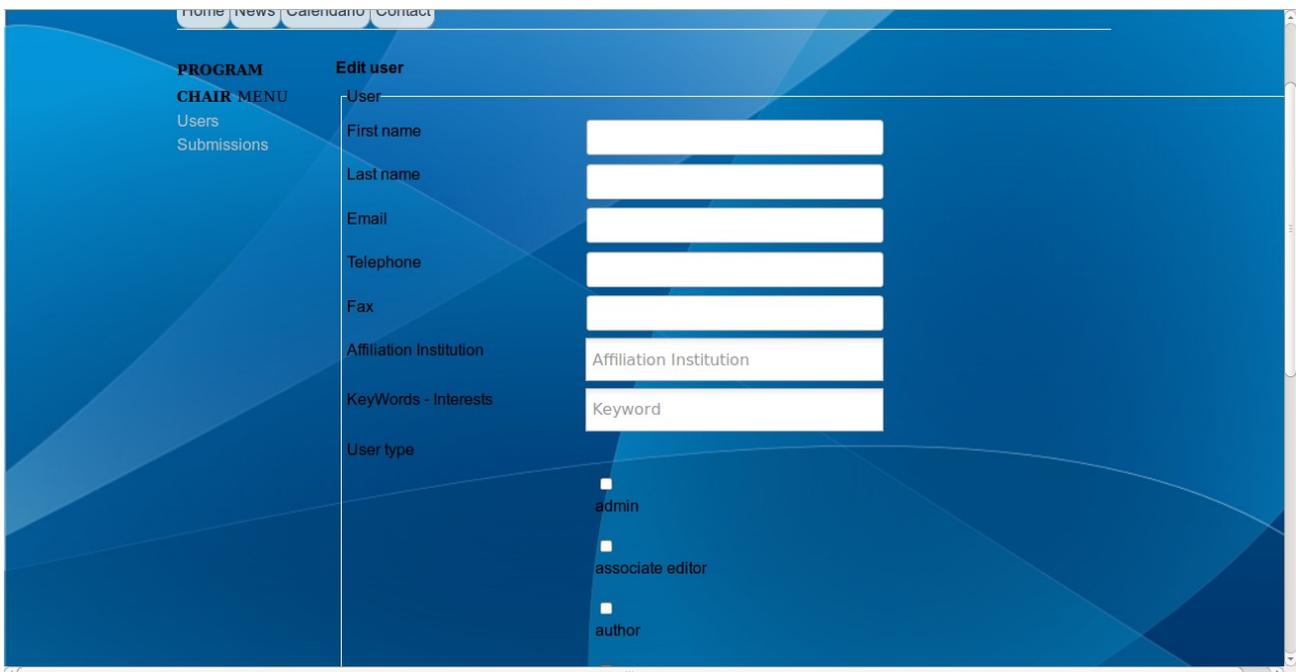
El program chair puede gestionar los roles de los usuarios, para ello, debe seleccionar sobre su menú la opción “Users”



Al seleccionar la opción indicada, aparecerá un listado de los usuarios del listado



Seleccionamos editar sobre del usuario que deseemos modificar los datos personales y/o rol y nos aparecerá la siguiente ventana en la cual modificaremos los datos que deseemos



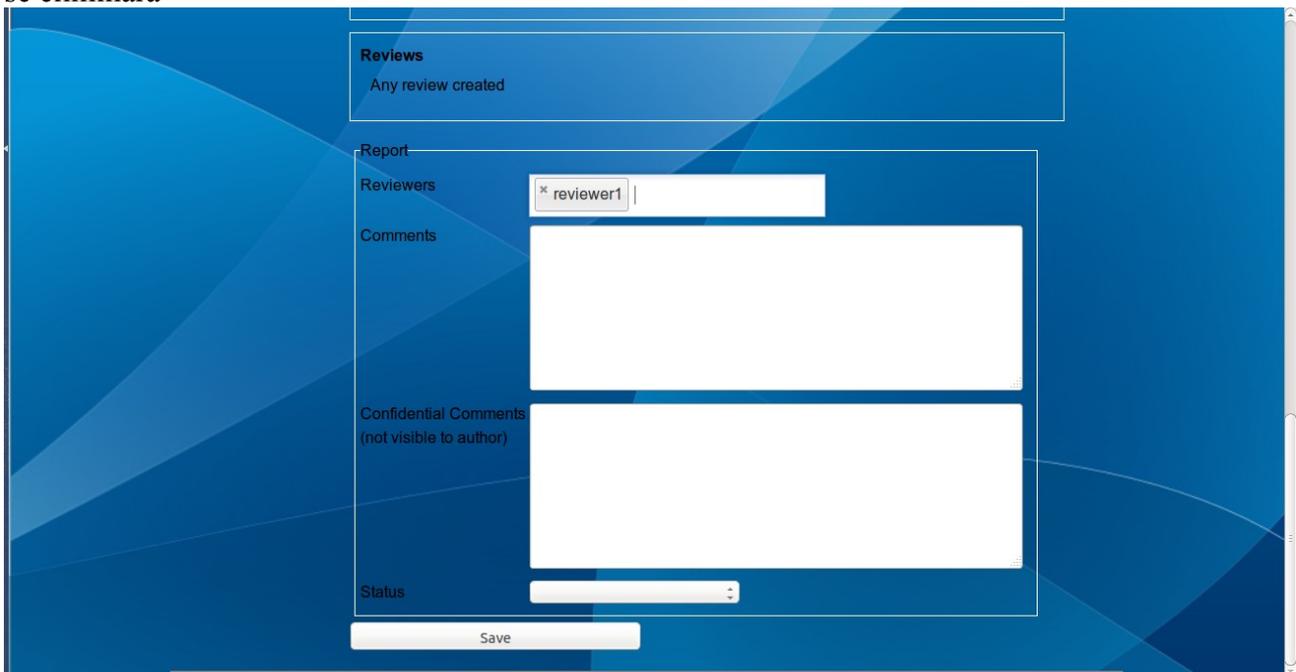
La cuenta del associate editor

El associate editor selecciona los revisores de las presentaciones iniciales a las que ha sido asignado, y posteriormente da una opinión sobre la aceptación o rechazo.

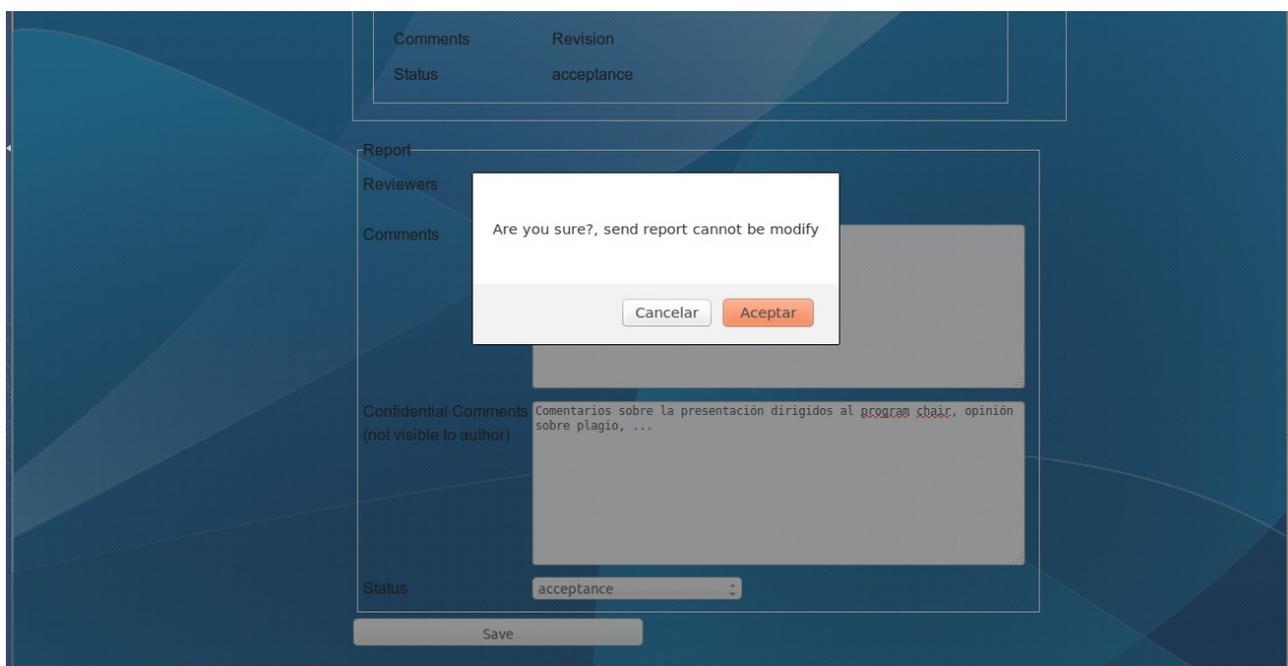
Listar las presentaciones que le han asignado:



Asignar la presentación a los revisores, e introducir los comentarios lo hará desde la siguiente ventana, hay que tener en cuenta que si el editor asociado, elimina un revisor, la revisión asociada se eliminará



Dar opinión de aceptación o rechazo de la presentación, una vez introduzca la aceptación de la presentación, no podrá volver a modificarla



La cuenta del revisor

El reviewer revisa las presentaciones que le han asignado

Listar las presentaciones que le han asignado



Trabajo de Fin de Grado – Gabriel Urso Santana Reyes

Revisar presentación se realiza seleccionando sobre el enlace review de la presentación que queremos presentar, podemos ir almacenando la revisión hasta que seleccionemos el “status” de la revisión una vez indiquemos el “status”, ya no podremos modificarlo

