UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques
Universitat de Barcelona

Desenvolupament d'una plataforma web de gestió de tallers i reptes científics

---

Autor: Oriol Suils Pérez

Directors:	Dr. Sergio Escalera
	Dr. Xavier Baró
	Marc Oliu
Realitzat a:	Departament de Matemàtiques i informàtica

Barcelona, 30 de junio de 2016

Agradecimientos

Me gustaría agradecer a todas las personas que me han ayudado a todos los niveles en esta etapa de mi vida.
A mi familia por apoyarme y creer en mi siempre, sin importar el momento y la situación.
A mis grandes amigos, Rubén Nieto, Daniel Ortega, David Rodríguez y Daniel Torres, por brindarme momentos de desahogo y divertimento, que los disfruto enormemente.
A todos mis profesores, sin excepción, por darme la oportunidad de aprender y formarme en múltiples ámbitos.
A mis tutores, Sergio Escalera, Xavier Baró i Marc Oliú, por plantearme el proyecto y tener sus consejos y ayuda siempre que lo he necesitado.

Abstract

Constantly new studies and advances in computer vision-based applications appear, a good way to encourage this developments is by creating scientific challenges and workshops. This project is responsible for developing a web platform, entirely in English, which is dedicated to facilitate the creation and monitoring the progress of these competitions and events.
This platform allows visualization of the various competitions created by the organizers, where users visiting the web can see all the information about the challenges like the description, the members participating, the schedule and the results. Along with the competitions are often conducted events related like workshops or special issues, where web visitors can also learn about these kind of events.
Also, if the user is logged on the platform, they can download the data set allocated to a competition and later publish their own post-competition results, where they can compare with the final results of the competition concerned. They also have the possibility of making an event proposal that administrators will analyze, and create it if they consider it convenient, giving the possibility to the user who made the proposal to edit it.
The platform includes a part for administrator users, having the ability to organize any type of event and its components, and manage all users and sponsors. The website is implemented by Django a Python web framework, along with Bootstrap for the client side.

Resum

Constantment apareixen nous estudis i avanços en aplicacions basades en visió computacional, una bona manera d'incentivar aquests avanços és la creació de reptes i tallers científics. Aquest projecte s'encarrega de desenvolupar una plataforma web, totalment en anglès, que es dedica a facilitar la creació i el seguiment del progrés d'aquestes competicions i esdeveniments.
Aquesta plataforma permet la visualització de les diferents competicions creades per part dels organitzadors, on els usuaris que visitin la web poden observar tota la informació referida als reptes com la descripció, els membres que participen, el calendari i els resultats. Juntament amb les competicions se solen realitzar esdeveniments relacionats com tallers o problemes especials, on els visitants de la web també poden informar-se sobre aquests esdeveniments.
A més si l'usuari es registra en la plataforma, pot descarregar-se el conjunt de dades assignades a una competició i més tard, publicar els seus propis resultats post-competició, on es podrà comparar amb els resultats finals de la competició en qüestió. També tenen la possibilitat de realitzar una proposta d'esdeveniment que els administradors analitzaran, i si creuen convenient el crearan, donant la possibilitat d'editar-ho a l'usuari que ha realitzat la proposta.
La plataforma inclou una part per a usuaris de tipus administrador, que tenen la capacitat d'organitzar qualsevol tipus d'esdeveniment i els seus components, i gestionar el conjunt d'usuaris i sponsors.
La web està implementada mitjançant el *web framework* Django de Python, juntament amb Bootstrap per a la part del client.

Resumen

Constantemente aparecen nuevos estudios y avances en aplicaciones basadas en visión computacional, una buena manera de incentivar estos avances es mediante la creación de retos y talleres científicos. Este proyecto se encarga de desarrollar una plataforma web, totalmente en inglés, que se dedica a facilitar la creación y el seguimiento del progreso de estas competiciones y eventos.
Esta plataforma permite la visualización de las diferentes competiciones creadas por parte de los organizadores, donde los usuarios que visiten la web pueden observar toda la información referida a los retos como la descripción, los miembros que participan, el calendario y los resultados. Junto con las competiciones se suelen realizar eventos relacionados como talleres o problemas especiales, donde los visitantes de la web también pueden informarse sobre estos eventos.
Además si el usuario se registra en la plataforma, puede descargarse el conjunto de datos asignados a una competición y más tarde, publicar sus propios resultados post-competición, donde se podrá comparar con los resultados finales de la competición en cuestión. También tienen la posibilidad de realizar una propuesta de evento que los administradores analizarán, y si creen conveniente lo crearán, dando la posibilidad de editarlo al usuario que ha realizado la propuesta.
La plataforma incluye una parte para usuarios de tipo administrador, que tienen la capacidad de organizar cualquier tipo de evento y sus componentes, y gestionar el conjunto de usuarios y sponsors.
La web está implementada mediante el *web framework* Django de Python, junto con Bootstrap para la parte del cliente.

Índice

1. Introducción **6**

1.1. Contexto i motivación 6

1.2. Objetivos 6

1.3. Estructura de la memoria 7

2. Análisis **8**

2.1. Requerimientos 8

2.2. Casos de uso 9

2.3. Tecnologías 35

3. Planificación **39**

3.1. Valoración económica 39

4. Diseño e implementación **41**

4.1. Diseño web (*Wireframe*) 41

4.2. Modelo Relacional 44

4.3. Diagramas de secuencia 44

4.4. Detalles de implementación 57

5. Validación **59**

6. Conclusiones **60**

6.1. Futuras ampliaciones 60

7. Referencias **62**

Anexos **64**

A. Wireframe completo 64

B. Manual de usuario 68

C. Manual de administrador 73

1. Introducción

1.1. Contexto i motivación

Este proyecto surgió como una propuesta del *Departament de Matemàtiques i informàtica*. El objetivo principal de éste es facilitar la gestión y organización de talleres y retos científicos del ámbito de visión computacional mediante una aplicación web.
Estas competiciones se definen con un enunciado o problema y unas reglas que los usuarios tienen que seguir, un conjunto de datos que se van entregando siguiendo un calendario definido, y algunas indicaciones sobre cómo resolver el problema. Estos retos, también se suelen componer de diferentes tipos de eventos como puede ser un taller, con las presentaciones de los trabajos, charlas de investigadores y la entrega de premios a los ganadores de la competición.
Las competiciones siguen un calendario que define diferentes etapas. El primer paso consiste en publicar el problema, junto con los datos de entrenamiento y validación. Más tarde se entregan las predicciones de validación esperadas y los datos de test. Finalmente, mediante las predicciones de test, se clasifican los diferentes participantes.
Cuando finaliza la competición, los usuarios registrados en la aplicación web pueden seguir subiendo sus predicciones junto con una explicación sobre cómo han desarrollado el código, y posteriormente comparar sus resultados con los grupos que participaron en el reto.
La aplicación web incluye tres partes diferenciadas. La primera consiste en una zona pública visible para cualquier visitante a la plataforma, sin necesidad de estar registrado. En esta parte se puede observar la información básica de las competiciones y sus eventos. La segunda está dedicada a los usuarios registrados, éstos pueden publicar sus predicciones, y también proponer algún nuevo evento. La última, exclusiva para los administradores, consiste en entregar el control absoluto de crear, editar y eliminar competiciones y eventos.

Mi principal motivación para la realización este proyecto, ha sido poder desarrollar una aplicación web desde cero. Comenzando por la elección y aprendizaje de tecnologías, siguiendo por el desarrollo y finalizando con el *testing*. Además me brindará la oportunidad de ampliar mis conocimientos en el ámbito de la programación web, incluyendo el análisis del mercado de tecnologías y *frameworks* actuales, y la capacidad de elegir y aplicar las más convenientes para este proyecto.

1.2. Objetivos

A continuación se enumeran los objetivos propuestos para este proyecto:

- Analizar las diferentes tecnologías web actuales, con el objetivo de escoger las más convenientes para el proyecto, y estudiar e investigar en profundidad éstas para poder aplicarlas.

- Realizar el diseño de las diferentes partes de la plataforma.
- Desarrollar la web, según las necesidades requeridas por los usuarios.
- Realizar las pruebas necesarias para confirmar el buen funcionamiento de la plataforma, situada en un servidor.

1.3. Estructura de la memoria

La memoria de este proyecto se divide en 7 grandes bloques, ordenados cronológicamente, que se detallan a continuación:

- Introducción: En este apartado se describe el proyecto en ámbito general, cuáles son los objetivos y la motivación asociada.
- Análisis: En esta parte se detallan las necesidades de la plataforma y la comparativa de tecnologías a usar, junto con la justificación de su elección.
- Planificación: En este capítulo se explica como se ha organizado el trabajo y también se muestran dos gráficos para la representación de la inversión de tiempo que se ha dedicado, así como su coste económico asociado en caso de haber sido desarrollado como proyecto para una empresa.
- Diseño e implementación: En este apartado se muestran diferentes diagramas referidos al diseño de la plataforma, y también la explicación del desarrollo del código.
- Validación: En este capítulo se detallan los procedimientos realizados para asegurar un buen funcionamiento de la plataforma web, junto con una prueba de velocidad para comprobar la rapidez de la web.
- Conclusiones: Esta parte contiene los objetivos alcanzados, y algunas reflexiones sobre la posibilidad de trabajos futuros.
- Referencias: Esta apartado contiene todas las referencias que se han consultado durante el desarrollo del proyecto.

2. Análisis

La primera parte del desarrollo de la plataforma consiste en analizar todos los requerimientos necesarios e investigar las tecnologías web a implementar.

2.1. Requerimientos

En la especificación de requisitos se describen todas las características que ha de cumplir la web. Para completar la lista de requerimientos se investigó sobre las necesidades funcionales que los usuarios futuros de la plataforma necesitarían.
Se han definido dos tipos de usuarios, que se detallan a continuación.

- Usuario: Consiste en usuarios que pueden haber iniciado sesión o no en la web. Tienen la posibilidad de visualizar todo el contenido público de los diferentes eventos. En caso de iniciar sesión, tienen la posibilidad de editar su perfil y proponer nuevos retos.
- Administrador: Este tipo de usuario es el responsable de la web, ya que puede manipular todo el contenido que se muestra en la plataforma como las competiciones.

Después de detallar los diferentes actores que existen en la plataforma, se listarán los requerimientos funcionales, separados por roles:

- Todos los tipos de actores:
  - Registrar una cuenta.
  - Iniciar sesión.
  - Editar perfil de usuario.
  - Salir de la sesión.
  - Consultar toda la información referida a *challenges*.
  - Consultar toda la información referida a *datasets*.
  - Consultar toda la información referida a *workshops*.
  - Consultar toda la información referida a *special issues*.
  - Consultar el listado de *sponsors*.
  - Realizar una búsqueda.
  - Proponer un nuevo evento.
- Administrador:
  - Crear nuevos usuarios.
  - Consultar el listado de usuarios registrados.
  - Editar información de usuarios registrados, que no sean administradores.

- Crear, editar, y eliminar toda la información relacionada con *challenges*.
- Crear, editar, y eliminar toda la información relacionada con *datasets*.
- Crear, editar, y eliminar toda la información relacionada con *workshops*.
- Crear, editar, y eliminar toda la información relacionada con *special issues*.
- Crear nuevos sponsors.

2.2. Casos de uso

En este capítulo se detallan las diferentes acciones que puede realizar cada tipo de actor, mediante dos diagramas de casos de uso y la especificación de cada uno de ellos.

Figura 1: Casos de uso del usuario.

En la anterior figura 1 se pueden observar el listado de funcionalidades que pueden realizar tanto el usuario como el administrador.
A continuación se describen el conjunto de casos de uso, mediante su especificación.

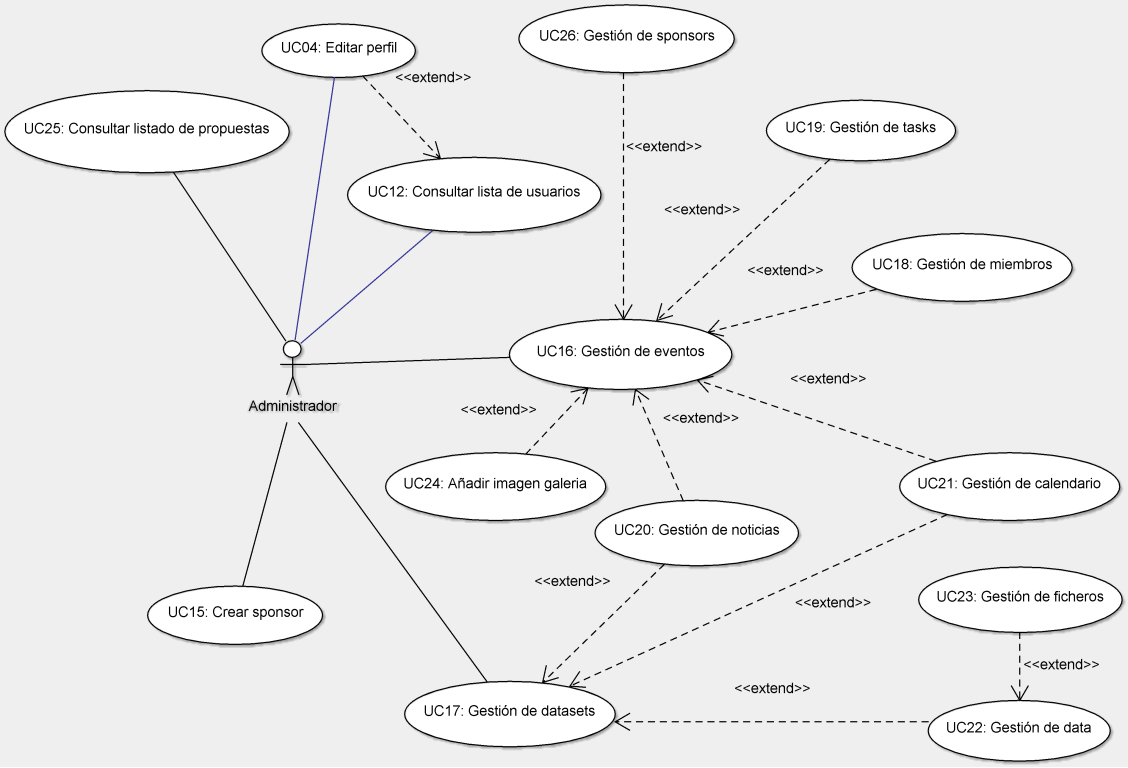

Figura 2: Casos de uso exclusivos del administrador.

En la figura 2 se puede ver el conjunto de acciones que puede realizar exclusivamente el usuario con el rol de administrador.

UC01	**Registro de cuenta**
Descripción	Permite registrarse a la plataforma.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la página principal. ▪ No tienen que haber iniciado sesión. ▪ El nombre de usuario y el correo electrónico que introduzca el usuario no ha de estar registrado.
Flujo básico	

1. El actor selecciona el botón *Sign up* situado en la parte superior de la página.
2. El sistema muestra el formulario de registro de usuarios.
3. El actor rellena todos los campos obligatorios del formulario y selecciona el botón *Register*.
4. El sistema envía un correo electrónico, a la cuenta indicada en el formulario anterior, con un enlace para activarla.
5. El actor selecciona el enlace del correo electrónico.
6. El sistema muestra un mensaje indicando que se ha activado la cuenta con éxito.

Post-condiciones	▪ Se muestra un mensaje para indicar que el actor ya puede iniciar sesión, con los datos introducidos.

Tabla 1: Caso de uso UC01

UC02	**Iniciar sesión**
Descripción	Permite que los actores inicien sesión en la plataforma.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la pagina principal. ▪ No tienen que haber iniciado sesión. ▪ Tienen que haber realizado UC01: Registro de cuenta.
Flujo básico	

1. El actor selecciona el botón *Sign in* situado en la parte superior de la página.
2. El sistema muestra un formulario con dos campos para iniciar sesión.
3. El actor introduce el nombre de usuario y la contraseña, y selecciona el botón *Sign in*.
4. El sistema muestra la página principal.

Post-condiciones	▪ Se muestra la pagina principal, pero con la diferencia que ahora, en la parte superior de la página, se puede observar un menú de usuario.

Flujo alternativo

4.a El actor ha introducido datos incorrectos.

1. El sistema muestra un mensaje de error.
2. El cliente vuelve al paso 3.

4.b El actor no recuerda su contraseña.

1. El cliente selecciona la opción *Forgot password?* UC13.

Tabla 2: Caso de uso UC02

UC03	**Salir de la sesión**
Descripción	Permite a los actores salir de la sesión iniciada.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Logout*.

Post-condiciones	▪ Se muestra la pagina principal.

Tabla 3: Caso de uso UC03

UC04	**Editar perfil**
Descripción	Permite a los actores editar su información.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Edit profile*.
4. El sistema muestra un formulario rellenado con la información introducida por el actor en el registro.
5. El actor modifica las opciones que desee y selecciona el botón *Save*.

Post-condiciones	▪ Se muestra la pagina principal, con los datos ya guardados.

Tabla 4: Caso de uso UC04

UC05	**Consultar información de *challenges***

Descripción	Permite a los actores observar toda la información pública referida a las competiciones.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor selecciona el menú llamado *Challenges* situado en la parte superior de la página.
2. El sistema muestra un desplegable con la lista de competiciones.
3. El actor selecciona una de ellas.
4. El sistema muestra una página con la descripción y las noticias del reto.
5. El actor puede acceder a diferente información relacionada con la competición, como los miembros que participan, el calendario, los *sponsors*, los *tracks*, seleccionando las opciones de la barra izquierda.

Post-condiciones	▪ Se muestra una página que detalla información de un reto concreto.

Tabla 5: Caso de uso UC05

UC06	**Consultar información de *datasets***
Descripción	Permite a los actores observar toda la información pública referida a los conjuntos de datos.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor selecciona el menú llamado *Datasets* situado en la parte superior de la página.
2. El sistema muestra un desplegable con la lista de los conjuntos de datos.
3. El actor selecciona uno de ellos.
4. El sistema muestra una página con la descripción y las noticias del *dataset*.
5. El actor puede acceder a diferente información relacionada con el *dataset*, como las diferentes *data*, el calendario, los eventos relacionados, seleccionando las opciones de la barra izquierda.

Post-condiciones	▪ Se muestra una página que detalla información de un conjunto de datos concreto.

Flujo alternativo

5.a El actor quiere descargar los ficheros de datos UC14.

Tabla 6: Caso de uso UC06

UC07	**Consultar información de *workshops***
Descripción	Permite a los actores observar toda la información pública referida a los talleres.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor selecciona el menú llamado *Workshops* situado en la parte superior de la página.
2. El sistema muestra un desplegable con la lista de talleres.
3. El actor selecciona uno de ellos.
4. El sistema muestra una página con la descripción y las noticias del taller.
5. El actor puede acceder a diferente información relacionada con el taller, como el calendario, los ponentes invitados, la galería, seleccionando las opciones de la barra izquierda.

Post-condiciones	▪ Se muestra una página que detalla información de un taller concreto.

Tabla 7: Caso de uso UC07

UC08	**Consultar información de *special issues***
Descripción	Permite a los actores observar toda la información pública referida a los *special issues*.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor selecciona el menú llamado *special issues* situado en la parte superior de la página.
2. El sistema muestra un desplegable con la lista de éstos.
3. El actor selecciona uno de ellos.
4. El sistema muestra una página con la descripción y las noticias del *special issue*.
5. El actor puede acceder a diferente información relacionada con éste, como los miembros que participan, el calendario, los eventos relacionados, seleccionando las opciones de la barra izquierda.

Post-condiciones	▪ Se muestra una página que detalla información de un textitspecial issue concreto.

Tabla 8: Caso de uso UC08

UC09	**Consultar información de *sponsors***
Descripción	Permite a los actores observar la lista de *sponsors* globales de la plataforma.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor selecciona la opción *Sponsors* situada en la parte superior de la página.
2. El sistema muestra una lista donde se detallan los *sponsors* registrados.

Post-condiciones	▪ Se muestra una página que detalla información de los *sponsors*.

Tabla 9: Caso de uso UC09

UC10	**Realizar búsqueda**
Descripción	Permite a los actores realizar una búsqueda mediante una caja de entrada de texto.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor introduce algún carácter a buscar en la caja de entrada de texto situada en la parte superior de la página, y selecciona la tecla *Enter*.
2. El sistema muestra una lista con los resultados de la búsqueda.
3. El actor selecciona uno de los resultados.
4. El sistema muestra la página de descripción de este evento concreto.

Tabla 10: Caso de uso UC10

UC11	**Proponer evento**
Descripción	Permite a los actores proponer un evento futuro.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Propose event*.
4. El sistema muestra un formulario.
5. El actor introduce el tipo de evento, título y descripción, y selecciona el botón *Send*.
6. El sistema muestra la página principal.

Post-condiciones	▪ Se muestra la página principal.

Tabla 11: Caso de uso UC11

UC12	**Consultar lista de usuarios**
Descripción	Permite a los actores consultar la lista de usuarios registrados en la plataforma.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.
Flujo básico	1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página. 2. El sistema muestra un desplegable con diversas opciones. 3. El actor selecciona la opción *Users zone* dentro de *Admin site*. 4. El sistema muestra un listado de los usuarios.
Post-condiciones	▪ Se muestra una página con la lista de usuarios.
Flujo alternativo	5.a El actor quiere editar el perfil de un usuario no administrador UC04.

Tabla 12: Caso de uso UC12

UC13	**Recuperar contraseña**
Descripción	Permite a los actores reiniciar la contraseña de su cuenta.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC01: Registro de cuenta.

Flujo básico

1. El actor selecciona la opción *Forgot password?*
2. El sistema muestra un formulario con un único campo.
3. El actor introduce su correo electrónico.
4. El sistema envía un correo al cliente con un enlace para reiniciar su contraseña.
5. El actor selecciona el enlace.
6. El sistema muestra un formulario con dos campos.
7. El actor introduce la nueva contraseña dos veces.
8. El sistema muestra un mensaje de éxito.

Post-condiciones	▪ Se ha reiniciado la contraseña del actor.

Flujo alternativo

4.a El actor introduce un correo electrónico que no esta registrado en la plataforma.

1. El sistema muestra un mensaje de error.
2. El cliente vuelve al paso 3.

Tabla 13: Caso de uso UC13

UC14	**Descargar ficheros de datos**
Descripción	Permite a los actores descargar ficheros de datos dentro de un *dataset* concreto.
Actores	Usuario y administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma.

Flujo básico

1. El actor realiza UC06: Consulta de *datasets*, y selecciona la opción *Data*.
2. El sistema muestra un desplegable con un listado de elementos *data*.
3. El actor selecciona uno de ellos.
4. El sistema muestra una página con la descripción del elemento seleccionado.
5. El actor selecciona la opción *Files*.
6. El sistema muestra un listado de los enlaces de ficheros a descargar.
7. El actor selecciona el enlace que quiere descargar.

Post-condiciones	▪ Se ha descargado el fichero de datos.

Tabla 14: Caso de uso UC14

UC15	**Crear sponsor**
Descripción	Permite a los actores añadir un sponsor global para la plataforma.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.
Flujo básico	

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Create sponsor* dentro de *Admin site*.
4. El sistema muestra un formulario.
5. El actor introduce los datos necesarios para la creación de un sponsor y su contacto.
6. El sistema muestra la página principal.

Post-condiciones	▪ El sistema añade un sponsor a la plataforma.

Tabla 15: Caso de uso UC15

UC16	**Gestión de eventos**
Descripción	Permite a los actores crear, editar y eliminar competiciones, talleres y *special issues*, y sus componentes.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Event zone* dentro de *Admin site*.
4. El sistema muestra un listado de eventos creadas.

Post-condiciones	▪ El sistema realiza alguna gestión sobre un evento.

Flujo alternativo

3.a El actor quiere crear un nuevo evento.

1. El actor selecciona el botón +, para crear un nuevo evento.
2. El sistema muestra un formulario.
3. El actor introduce el título, descripción y tipo de evento.
4. El sistema vuelve al paso 3.

3.b El actor quiere eliminar un evento.

1. El actor selecciona el botón -, para eliminar un evento concreto.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 3.

3.c El actor quiere editar un evento.

1. El actor selecciona el nombre de un evento.
2. El sistema muestra una ventana, donde se puede editar todo el contenido del evento seleccionado.
3. El actor puede modificar el título y la descripción del evento, y además:
   - 3.1 Gestionar los miembros relacionados con un evento UC18.
   - 3.2 Gestionar los tasks de un evento UC19.
   - 3.3 Gestionar las noticias de un evento UC20.
   - 3.4 Gestionar el calendario un evento UC21.
   - 3.5 Gestionar las imágenes de la galería de un evento UC24.
   - 3.5 Gestionar los sponsors de un evento UC26.
4. El actor selecciona el botón *Save*.
5. El sistema vuelve al paso 3.

Tabla 16: Caso de uso UC16

UC17	**Gestión de *datasets***
Descripción	Permite a los actores crear, editar y eliminar *datasets* y sus componentes.

Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Dataset zone* dentro de *Admin site*.
4. El sistema muestra un listado de *datasets* creados.

Post-condiciones	▪ El sistema realiza alguna gestión sobre un *dataset*.

Flujo alternativo

3.a El actor quiere crear un nuevo *dataset*.

1. El cliente selecciona el botón +, para crear un nuevo *dataset*.
2. El sistema muestra el formulario del *dataset*.
3. El cliente introduce el título y descripción, y selecciona *Save*.
4. El sistema muestra el formulario de *data*.
5. El cliente introduce todos los campos, y selecciona *Save*.
6. El sistema muestra la página de edición del *dataset*.

3.b El actor quiere eliminar un *dataset*.

1. El actor selecciona el botón -, para eliminar una *dataset* concreto.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 3.

3.c El actor quiere editar un *dataset*.

1. El actor selecciona el nombre de un *dataset*.
2. El sistema muestra una ventana, donde se puede editar todo el contenido del *dataset* seleccionado.
3. El actor puede modificar el título y la descripción del *dataset*, y además:
   3.1 Gestionar los componentes *data* de un *dataset* UC22.
   3.2 Gestionar el calendario un *dataset* UC21.
4. El actor selecciona el botón *Save*.
5. El sistema vuelve al paso 3.

Tabla 17: Caso de uso UC17

UC18	**Gestión de miembros**
Descripción	Permite a los actores crear y eliminar los miembros relacionados con eventos.
Actores	Administrador.

Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre eventos y miembros.

Flujo alternativo

3.a El actor quiere crear un nuevo miembro.

1. El actor selecciona el botón + situado junto al título *People*, para crear un nuevo miembro.
2. El sistema muestra un formulario.
3. El actor selecciona el rol y introduce uno o más nombres en la caja de texto y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar un miembro.

1. El actor selecciona el botón -, para eliminar un miembro concreto.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 18: Caso de uso UC18

UC19	**Gestión de tasks**
Descripción	Permite a los actores crear y eliminar *tasks* relacionados con eventos.
Actores	Administrador.

Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre eventos y *tasks*.

Flujo alternativo

3.a El actor quiere crear un nuevo *task*.

1. El actor selecciona el botón + situado junto al título *Tasks*, para crear un nuevo *task*.
2. El sistema muestra un formulario.
3. El actor introduce título, descripción, *evaluation metrics* y *baseline* y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar un *task*.

1. El actor selecciona el botón -, para eliminar un *task* concreto.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 19: Caso de uso UC19

UC20	**Gestión de noticias**
Descripción	Permite a los actores crear y eliminar noticias relacionadas con eventos.
Actores	Administrador.

Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre eventos y noticias.

Flujo alternativo

3.a El actor quiere crear una nueva noticia.

1. El actor selecciona el botón + situado junto al título *News*, para crear una nueva noticia.
2. El sistema muestra un formulario.
3. El actor introduce título y descripción, y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar una noticia.

1. El actor selecciona el botón -, para eliminar una noticia concreta.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 20: Caso de uso UC20

UC21	**Gestión del calendario**
Descripción	Permite a los actores crear y eliminar nuevos acontecimientos del calendario del evento.
Actores	Administrador.

Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre eventos y su calendario.

Flujo alternativo

3.a El actor quiere crear un nuevo acontecimiento del calendario.

1. El actor selecciona el botón + situado junto al título *Schedule*, para crear un nuevo acontecimiento.
2. El sistema muestra un formulario.
3. El actor introduce título, fecha y descripción, y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar una noticia.

1. El actor selecciona el botón -, para eliminar un acontecimiento.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 21: Caso de uso UC21

UC22	**Gestión de *data***
Descripción	Permite a los actores crear, editar y eliminar nuevos componentes *data* dentro de un *dataset*.
Actores	Administrador.

Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC17: Gestión de *datasets*.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre un *dataset* y su *data*.

Flujo alternativo

3.a El actor quiere crear un nuevo *data*.

1. El actor selecciona el botón + situado junto al título *Data*, para crear uno nuevo.
2. El sistema muestra un formulario.
3. El actor introduce título, descripción, software y *evaluation metrics*, y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar un *data*.

1. El actor selecciona el botón -, para eliminar un *Data*.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

3.c El actor quiere editar un *Data*.

1. El actor selecciona el nombre de un *data*.
2. El sistema muestra una ventana, donde se puede editar todo el contenido del *data* seleccionado.
3. El actor puede modificar el título, la descripción del *data*, y además:
   3.1 Gestionar los ficheros que el usuario se puede descargar UC23.
4. El actor selecciona el botón *Save*.
5. El sistema vuelve al paso 3.

Tabla 22: Caso de uso UC22

UC23	**Gestión de ficheros**
Descripción	Permite a los actores crear y eliminar nuevos ficheros dentro de un *data*.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC22: Gestión de *data*.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre *data* y sus ficheros.

Flujo alternativo

3.a El actor quiere crear un nuevo fichero.

1. El actor selecciona el botón + situado junto al título *File's assigned*, para crear uno.
2. El sistema muestra un formulario.
3. El actor selecciona el fichero a subir o una url junto con un nombre, y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar un fichero.

1. El actor selecciona el botón -, para eliminar un fichero.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 23: Caso de uso UC23

UC24	**Añadir imagen galería**
Descripción	Permite a los actores crear nuevas imágenes dentro de la galería de un evento.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre un evento y su galería.

Flujo alternativo

3.a El actor quiere añadir una imagen a la galería.

1. El actor selecciona el botón + situado junto al título *Gallery pictures*, para añadir una.
2. El sistema muestra un formulario.
3. El actor selecciona la imagen junto con una breve descripción, y selecciona *Save*.
4. El sistema vuelve al paso 2.

Tabla 24: Caso de uso UC24

UC25	**Consultar listado de propuestas**
Descripción	Permite a los actores observar el listado de propuestas que han realizado los usuarios.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Event proposals list* dentro de *Admin site*.
4. El sistema muestra un listado de los eventos propuestos.

Post-condiciones	▪ El sistema muestra todos los eventos propuestos por los usuarios.

Tabla 25: Caso de uso UC25

UC26	**Gestión de sponsors**
Descripción	Permite a los actores crear y eliminar nuevos sponsors para un evento en concreto.
Actores	Administrador.
Pre-condiciones	▪ Los actores han de acceder a la plataforma. ▪ Los actores han de haber realizado UC02: Iniciar sesión.

Flujo básico

1. El actor selecciona el menú de usuario con su nombre situado en la parte superior de la página.
2. El sistema muestra un desplegable con diversas opciones.
3. El actor selecciona la opción *Event proposals list* dentro de *Admin site*.
4. El sistema muestra un listado de los eventos propuestos.

Flujo básico

1. El actor realiza UC16: Gestión de eventos.
2. El sistema muestra todos los componentes del evento.

Post-condiciones	▪ El sistema realiza alguna gestión relacionada entre un sponsor y un evento.

Flujo alternativo

3.a El actor quiere crear un nuevo sponsor.

1. El actor selecciona el botón + situado junto al título *Sponsors*, para crear uno nuevo.
2. El sistema muestra un formulario.
3. El actor selecciona el rol e introduce uno o más nombres de sponsor en la caja de texto y selecciona *Save*.
4. El sistema vuelve al paso 2.

3.b El actor quiere eliminar un sponsor.

1. El actor selecciona el botón -, para eliminar un sponsor.
2. El sistema muestra una ventana, pidiendo confirmación.
3. El actor selecciona *Delete*.
4. El sistema vuelve al paso 2.

Tabla 26: Caso de uso UC26

2.3. Tecnologías

Después de analizar los requerimientos y los conocimientos ya adquiridos, se decidió a utilizar un *framework* de Python, ya que la alternativa era usar lenguaje PHP o Ruby. En la figura 3 se puede observar las diferencias entre los tres lenguajes. Las principales características que hicieron decantarse por Python fueron por la facilidad de lectura y la usabilidad del código, junto con las múltiples y diversas librerías que están disponibles.

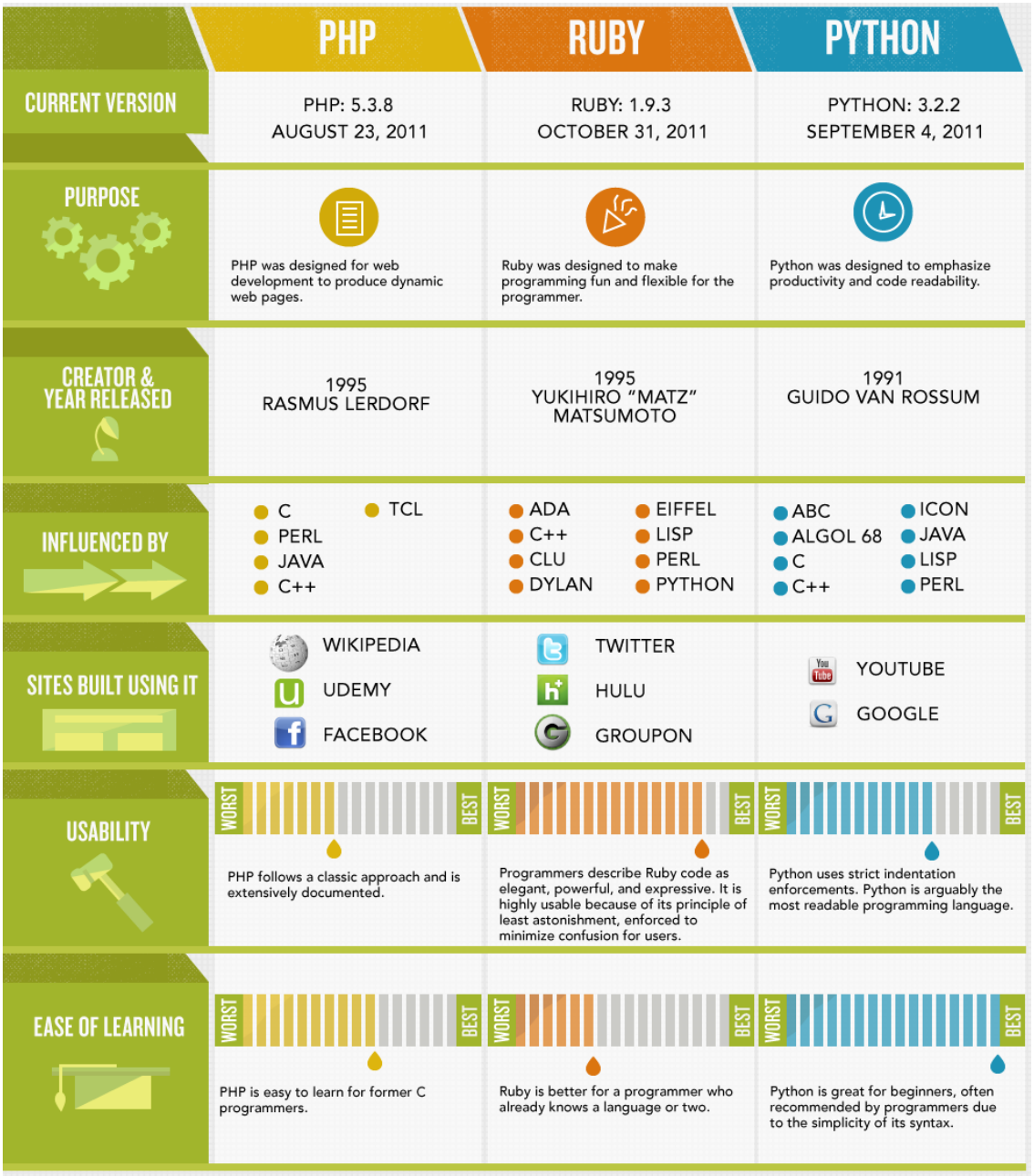

Figura 3: Diferencias entre PHP, Ruby, Python.

Dentro de los *web framework* de Python hay mucha variedad, pero se analizaron tres:

- **Flask**: Está orientado para proyectos pequeños, se desarrolla rápido y requiere poca configuración.
- **Pyramid**: Es el más flexible de los tres, en cuanto a que puede usarse para pequeñas aplicaciones y para sitios grandes. Un problema que plantea es que tiene muchas opciones y la curva de aprendizaje es más larga que los otros dos.

- **Django**: Es el mas popular y tiene una gran cantidad de *packages* para reutilizar.

Finalmente para el desarrollo del proyecto se decidió optar por Django [1]. Django es un *web framework* libre y de código abierto que sigue el patrón *Model-View-Controller* o más específicamente un derivado suyo llamado *Model-View-Template*. La parte de modelo consiste en clases Python que mediante un ORM interviene entre éstas y las múltiples tablas de la base de datos. La parte de vista consiste en funciones Python que procesan peticiones HTTP y devuelven respuestas HTTP. La parte de controlador es el *URL dispatcher*, es decir un fichero que contiene todas las URL de la plataforma y relaciona cada una con la función de la vista correspondiente. La parte de *template* consiste en definir cómo se muestra la información al usuario.

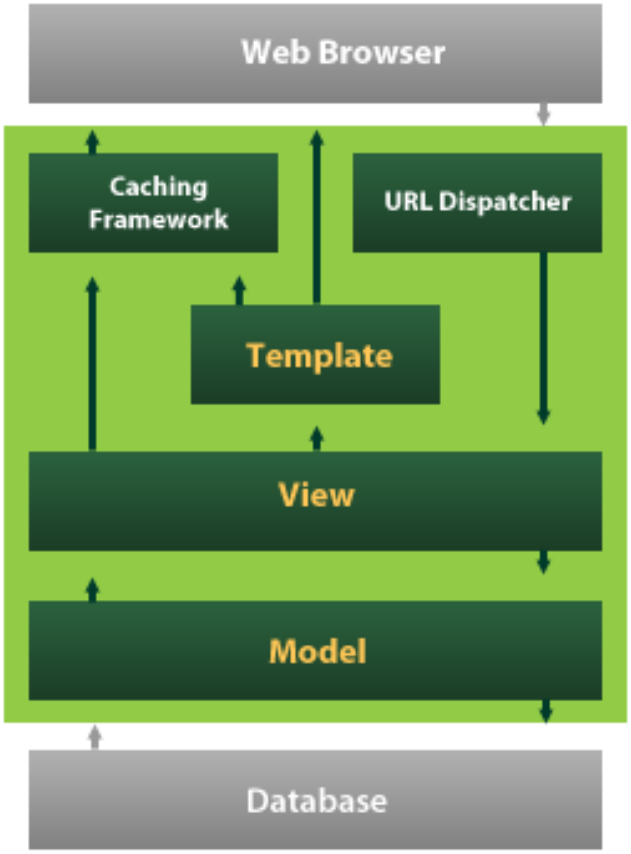

Figura 4: Arquitectura de Django.

El listado de lenguajes y tecnologías utilizados durante el desarrollo de la plataforma han sido:

- Back-end
  - Python. Para realizar toda la parte del servidor.
  - MySQL. Para la base de datos, aunque no se ha programado directamente, ya que se realiza mediante un ORM.

- Front-end
  - HTML. Para la visualización de la información en los *templates*.
  - Javascript. Para manejar eventos en la parte del cliente.
  - CSS. Para marcar el estilo de la plataforma.
  - DTL (**D**jango **T**emplate **L**anguage) [2]. Para realizar sentencias lógicas y consultar objetos en un *template*.

Para el desarrollo de la plataforma se han utilizado las siguientes herramientas:

- Balsamiq. Programa que se ha utilizado para la realización del *wireframe* completo de la web.
- ArgoUML. Software que ha servido para realizar diferentes diagramas para el diseño de la página.
- Vagrant [15]. Software que se ha utilizado para realizar un entorno de desarrollo virtual.
- Sublime Text 3. Editor que se ha usado para desarrollar todo el código.
- Pinegrow. Software que se ha utilizado para conseguir el estilo de la web, mediante CSS.
- Google Chrome. Navegador web usado para realizar las múltiples pruebas.
- GitHub. Utilizado para alojar el código de la plataforma.

Finalmente la plataforma web esta localizada en un servidor con el dominio http://chalearnlap.cvc.uab.es/.

3. Planificación

La planificación del proyecto se ha divido en diversas fases. El primer paso es el análisis del problema y las tecnologías a utilizar. La segunda fase es el aprendizaje de estas tecnologías junto con el diseño de la plataforma, tanto de la base de datos como de la parte estética. La siguiente fase es la implementación, donde se desarrolla el código. A continuación se muestra un diagrama de Gantt que contiene la planificación definida.

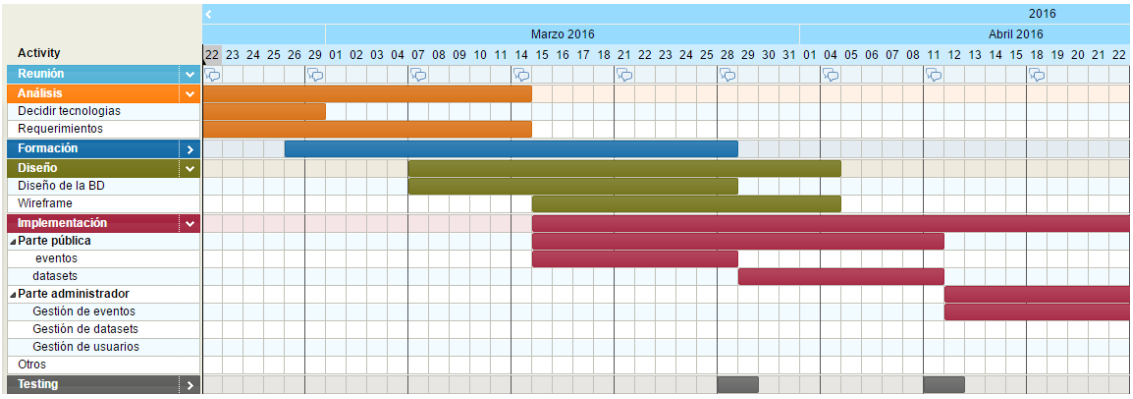

Figura 5: Diagrama de Gantt de Febrero hasta Abril.

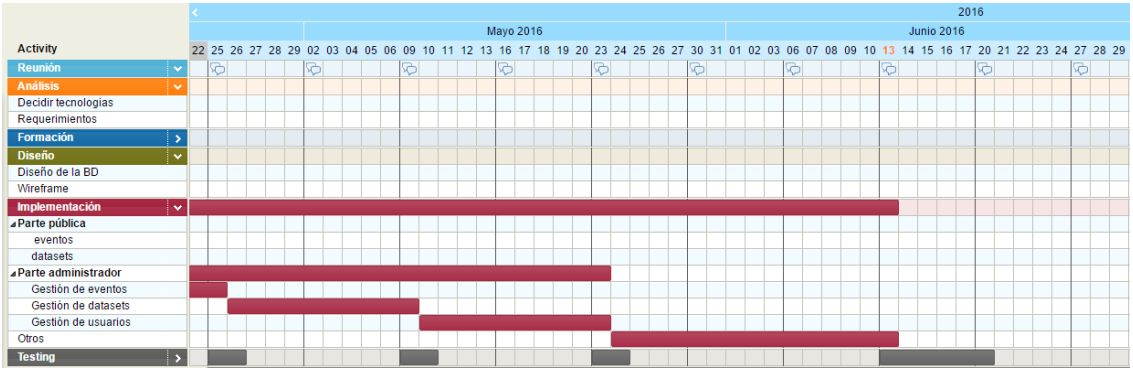

Figura 6: Diagrama de Gantt de Abril hasta Junio.

En las anteriores figuras 5 y 6, se han estimado entre tres y cuatro horas diarias, cinco días a la semana y no se ha incluido el tiempo previsto para la documentación y realización de la memoria.

3.1. Valoración económica

En este apartado se realiza una valoración económica del proyecto, tomando como referencia las horas planificadas. Este análisis está enfocado a nivel empresarial, pero solo valora superficialmente el coste que podría tener la plataforma.

Proceso	**Horas**
Análisis	35
Diseño componentes	50
Diseño gráfico	80
Implementación	110
Testing	25
Total	300

Tabla 27: Desglose de horas

En la tabla 27 se puede observar el desglose de horas estimadas, especificando el proceso del proyecto. Para las horas indicadas se ha pensado en dos puestos especializados, el primero sería un diseñador gráfico que se encargaría de definir el estilo de la plataforma, y el segundo puesto sería un programador *Full-Stack*, que se encargaría de toda la parte de programación, incluyendo el análisis de requerimientos y el *testing*.

Puesto de trabajo	**Tiempo estimado**	**Sueldo estimado**	**Coste total**
Diseñador	80 horas	21 €/h	1680 €
Programador Full-Stack	110+30+50+25 horas	26 €/h	5720 €

Tabla 28: Coste final

En la tabla 28 se puede observar las horas estimadas repartidas para cada puesto, junto con el coste parcial y total que habría en materia de sueldos.
En cuanto a costes adicionales, no habría que incluir ninguna licencia de software, ya que todos los programas utilizados son gratuitos, pero sí que se necesitaría un coste en el servidor, incluyendo el mantenimiento de éste y el DNS.

4. Diseño e implementación

En este capítulo se detallan las diversas fases de diseño y desarrollo de la plataforma, siguiendo un orden cronológico.

4.1. Diseño web (*Wireframe*)

La primera parte es diseñar el contenido que se muestra al usuario, mediante un *wireframe* o maqueta. Un *wireframe* esquematiza el diseño de la página, pero carece de estilo ya que el principal objetivo de éste es mostrar la organización de los elementos de la web. A continuación se muestran unas figuras con la maqueta de algunas páginas de la plataforma.

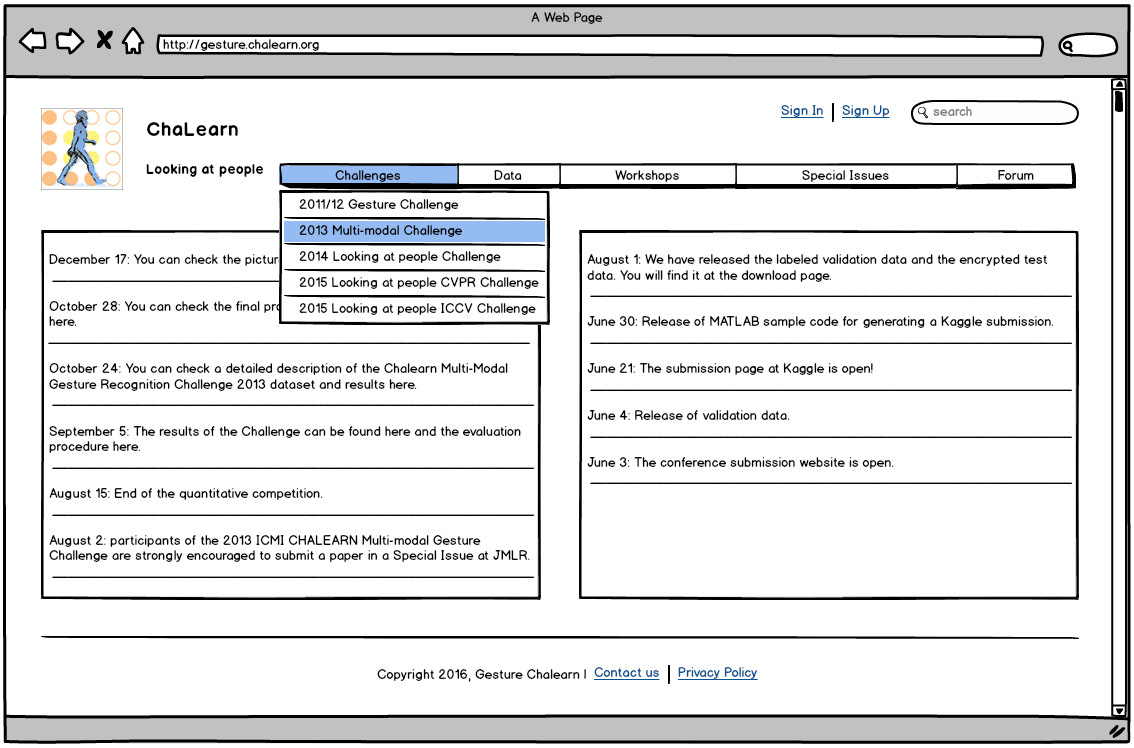

Figura 7: Página principal.

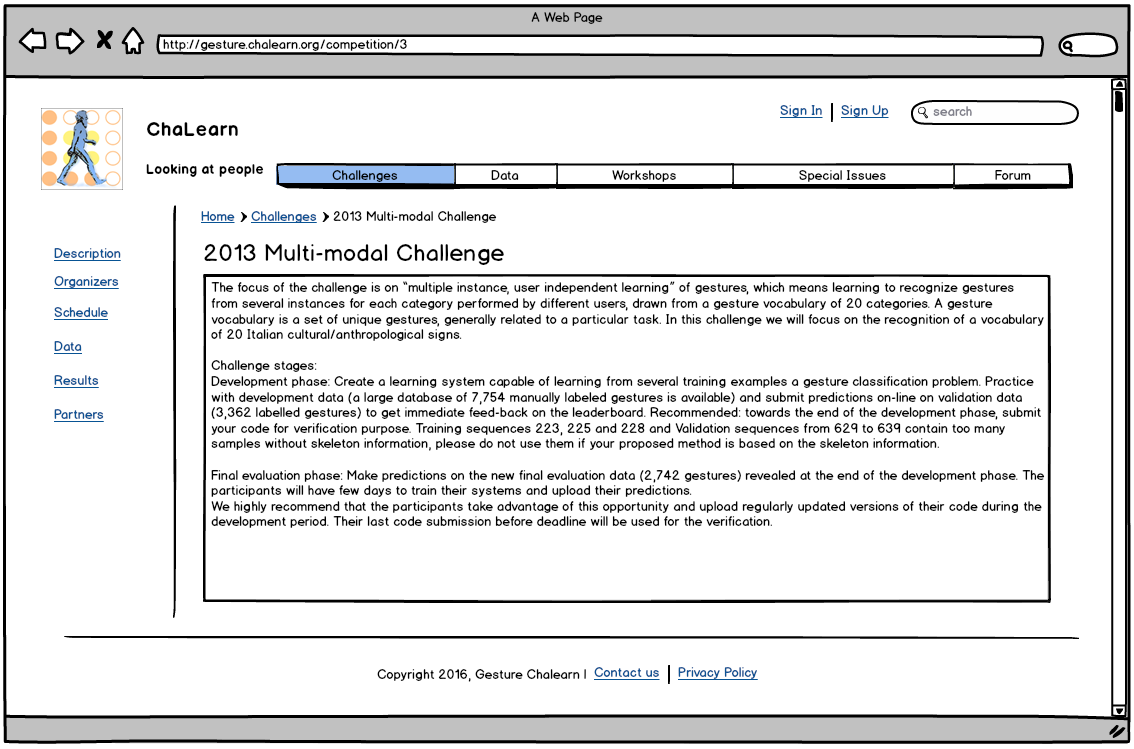

Figura 8: Página de descripción de una competición.

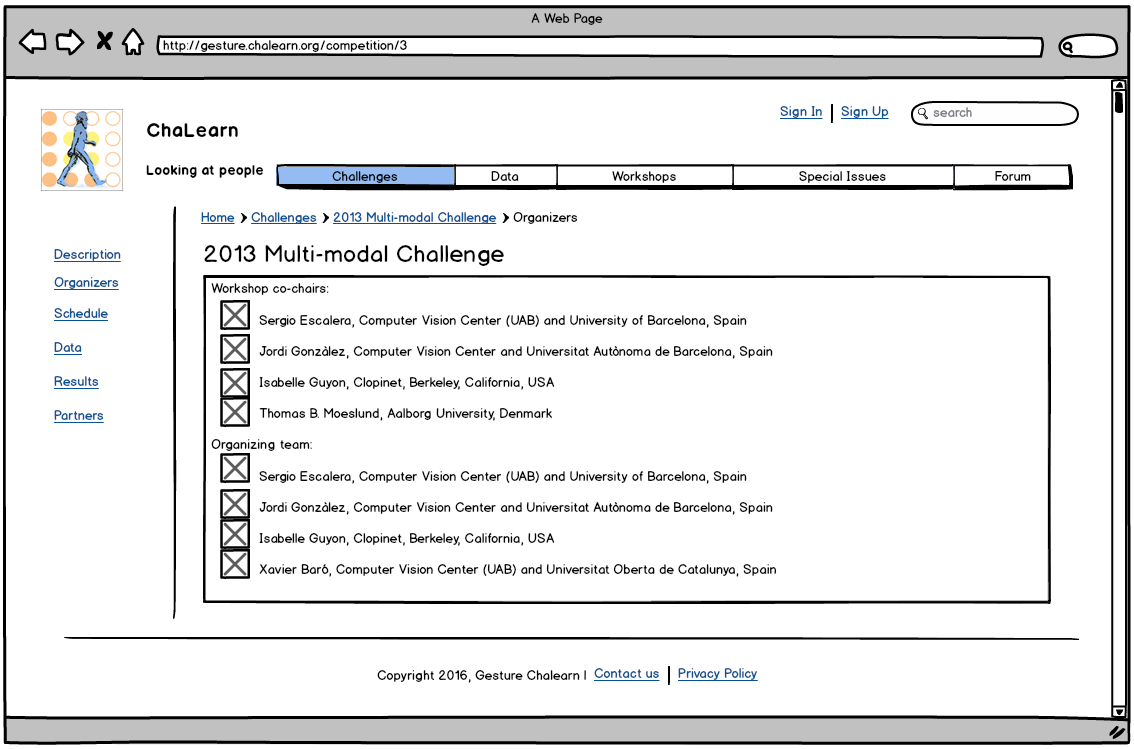

Figura 9: Página de los organizadores de una competición.

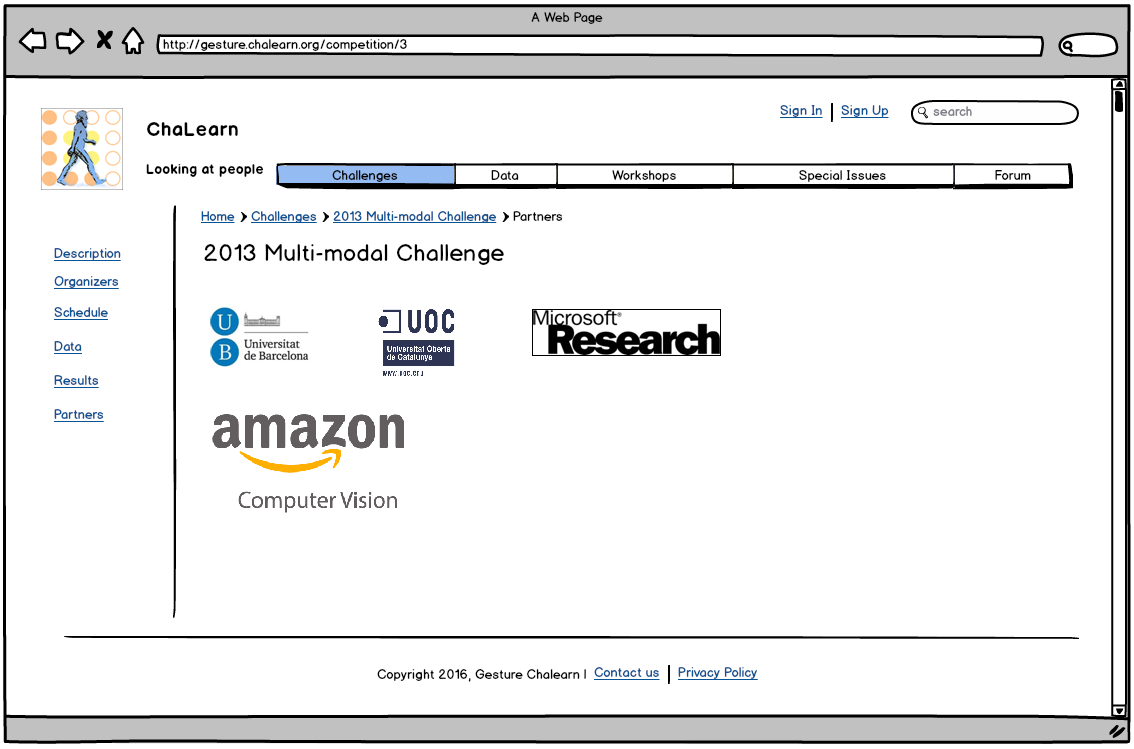

Figura 10: Página de los sponsor de una competición.

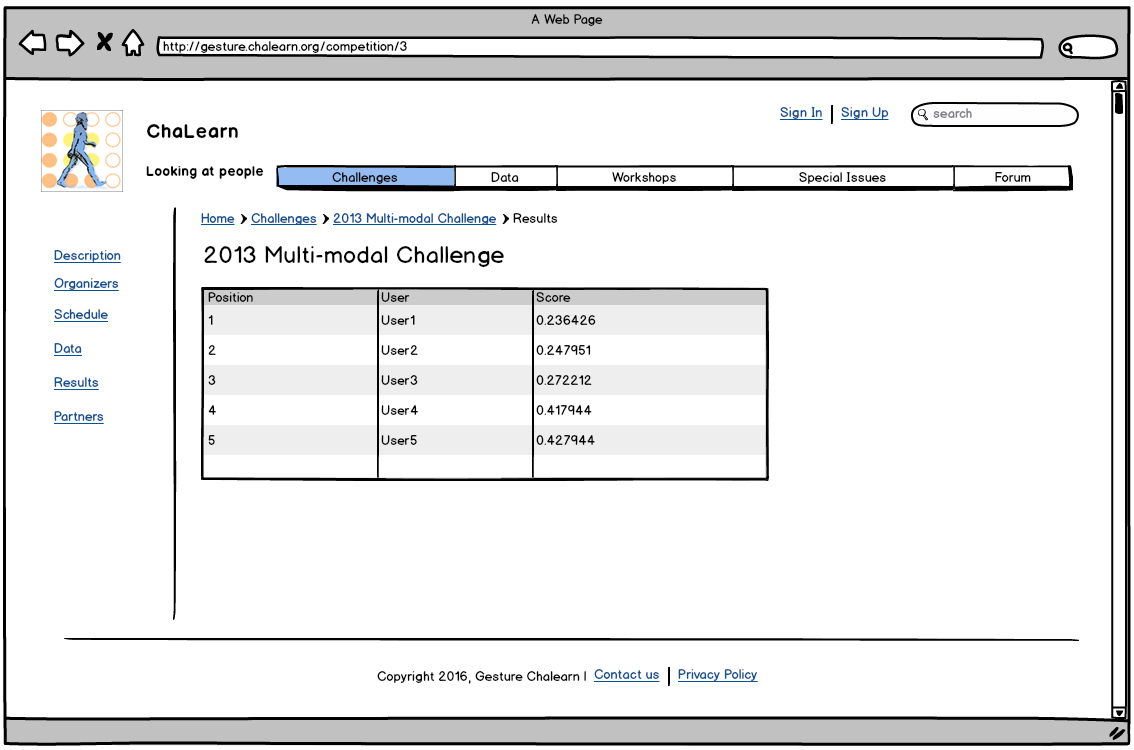

Figura 11: Página de los resultados de una competición.

Las figuras 7, 8, 9, 10 y 11 muestran la organización de los elementos de una parte representativa de la plataforma. En el anexo A se muestra el *wireframe* completo de la web.

4.2. Modelo Relacional

El siguiente paso es el diseño de la base de datos, como en este caso se maneja mediante un ORM, no se programan las tablas de la base de datos, sino se declaran clases Python que representan el apartado de modelo de la plataforma. A continuación se detalla el modelo relacional, donde se puede observar las diferentes relaciones entre modelos y los atributos de cada clase.

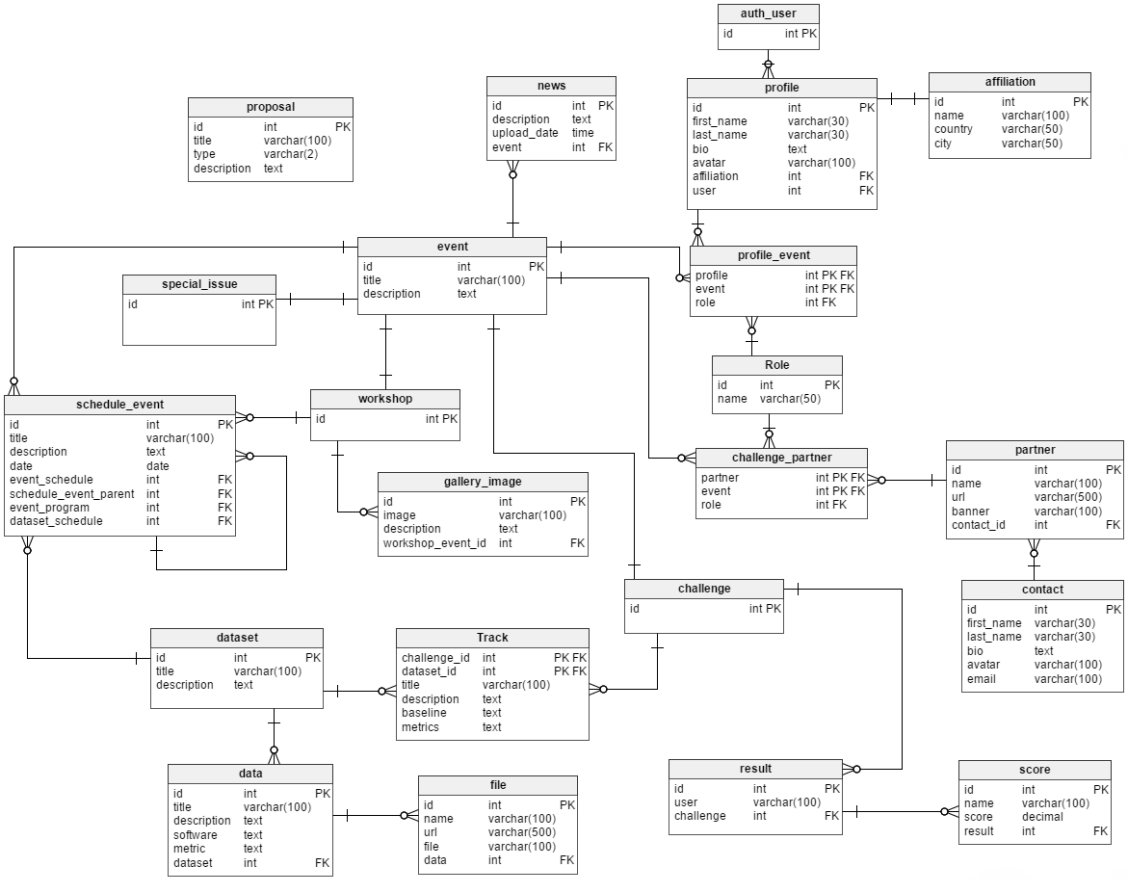

Figura 12: Modelo relacional.

4.3. Diagramas de secuencia

Finalmente, la última fase de diseño realizada son los diferentes diagramas de secuencia para cada caso de uso listados anteriormente. Los diagramas de secuencia muestran la interacción entre los usuarios y los diferentes componentes de la plataforma. Estos modelos sirven para observar los diversos escenarios de una manera simple y gráfica.

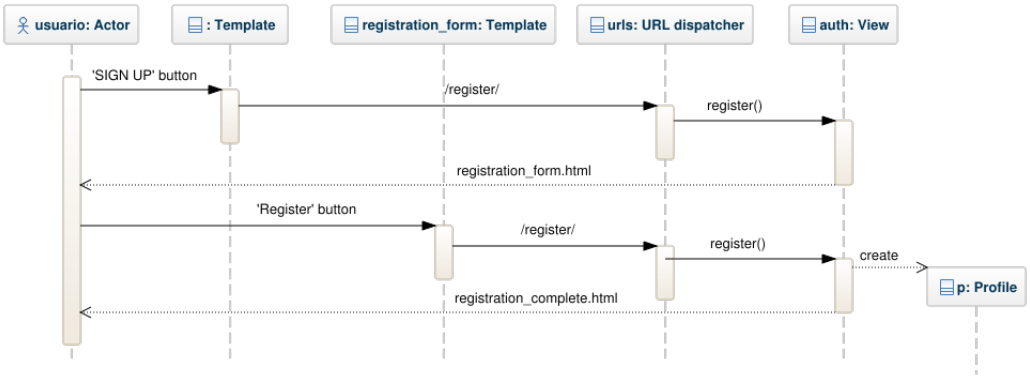

Figura 13: Diagrama de secuencia de UC01: Registrar cuenta.

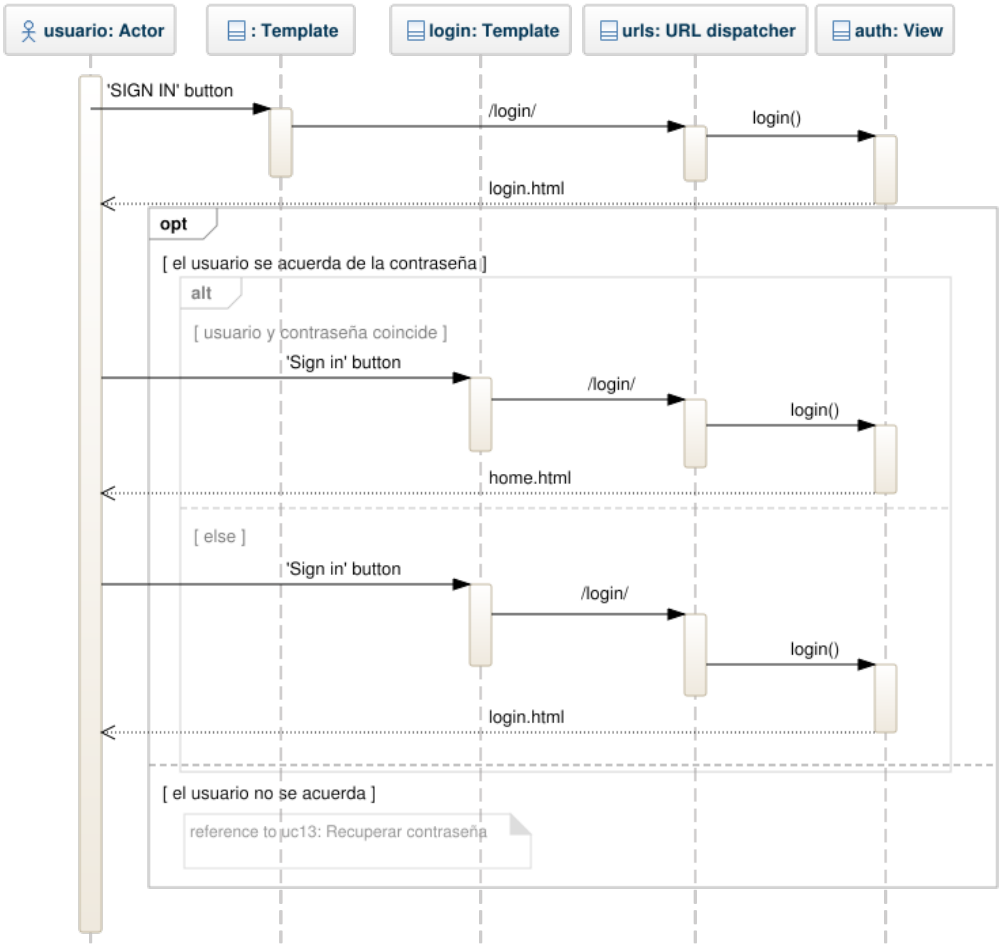

Figura 14: Diagrama de secuencia de UC02: Iniciar sesión.

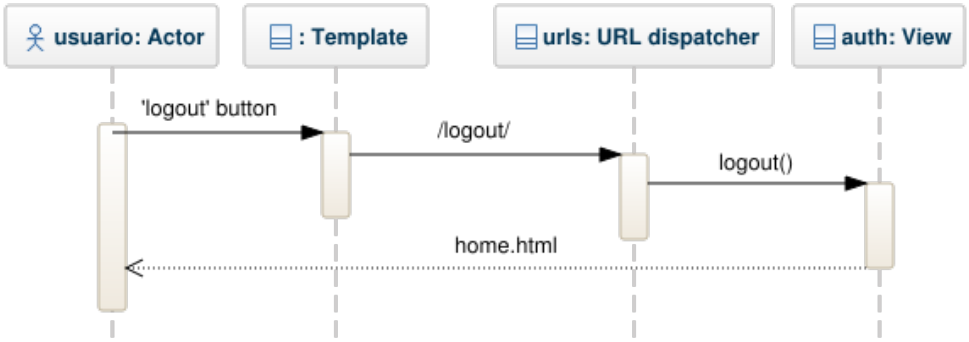

Figura 15: Diagrama de secuencia de UC03: Salir de la sesión.

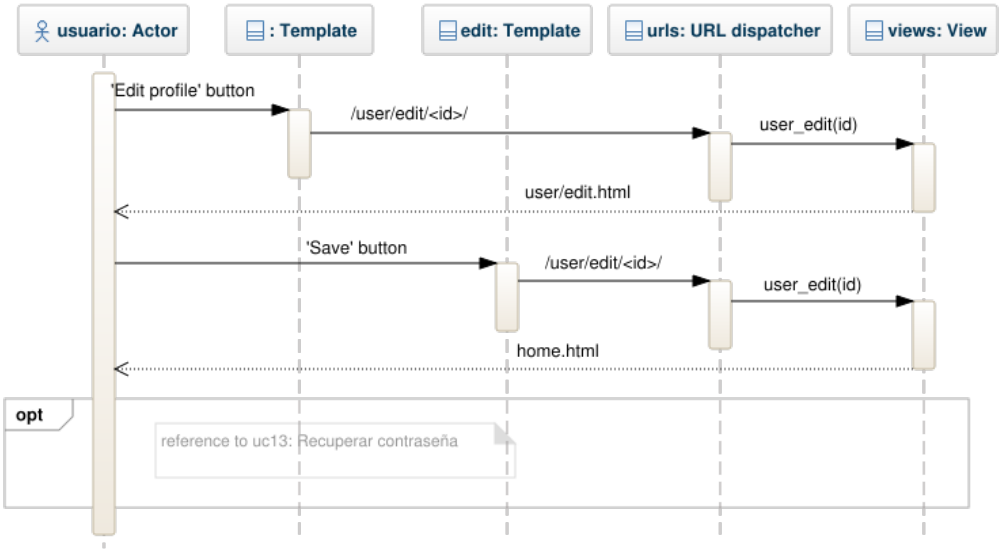

Figura 16: Diagrama de secuencia de UC04: Editar perfil.

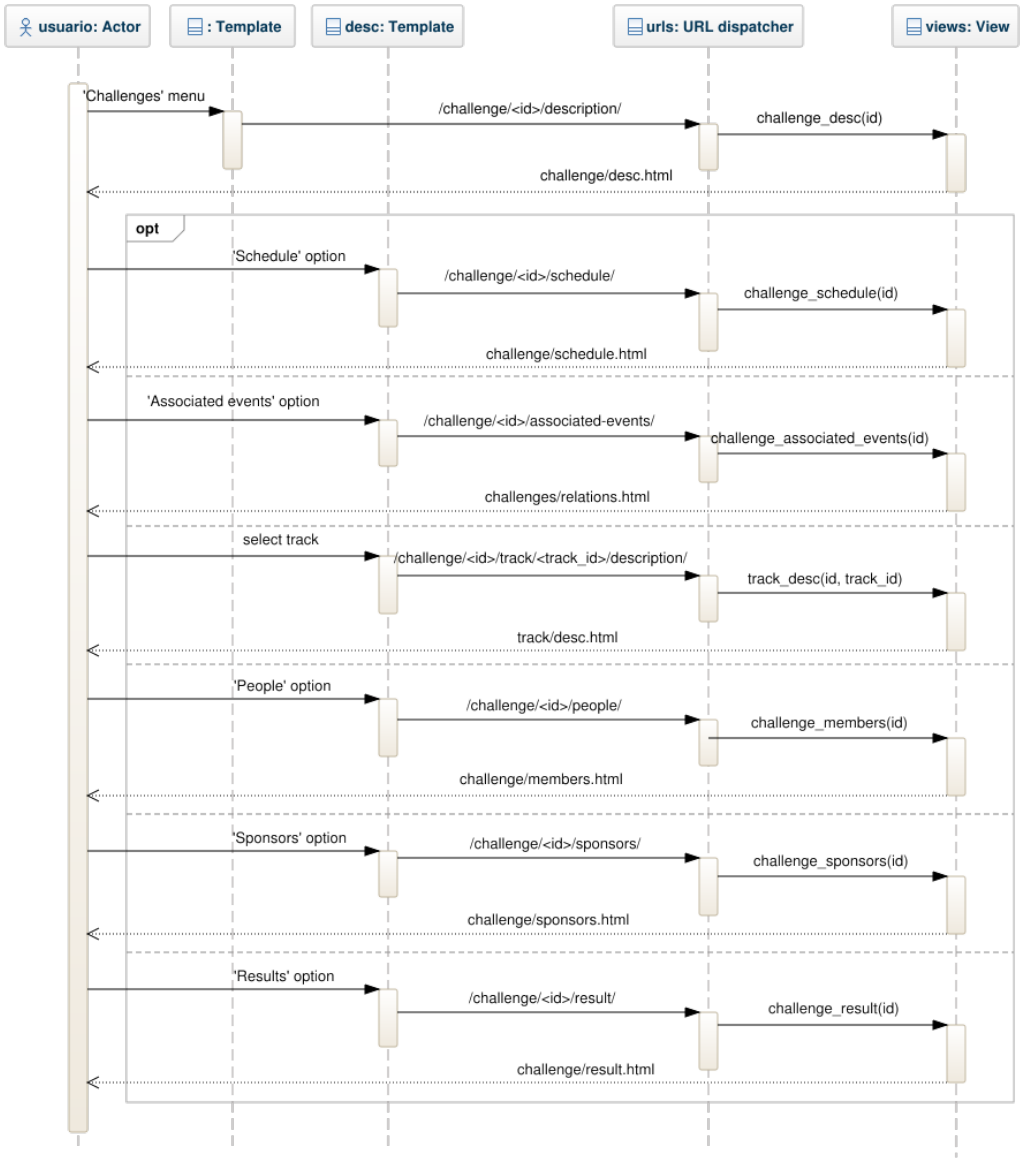

Figura 17: Diagrama de secuencia de UC05: Consultar información de challenges.

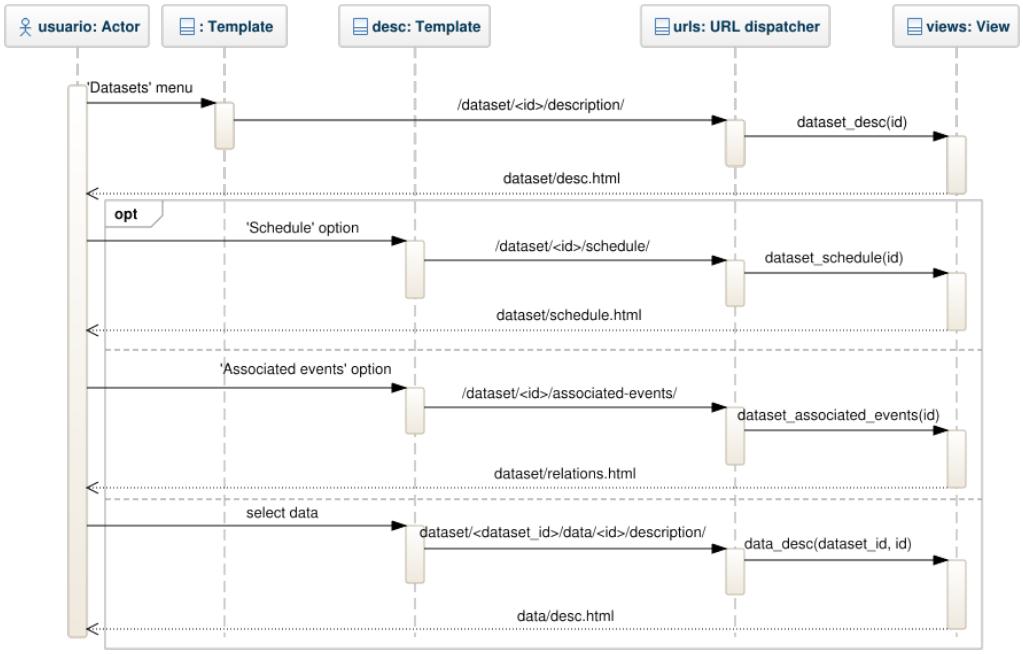

Figura 18: Diagrama de secuencia de UC06: Consultar información de datasets.

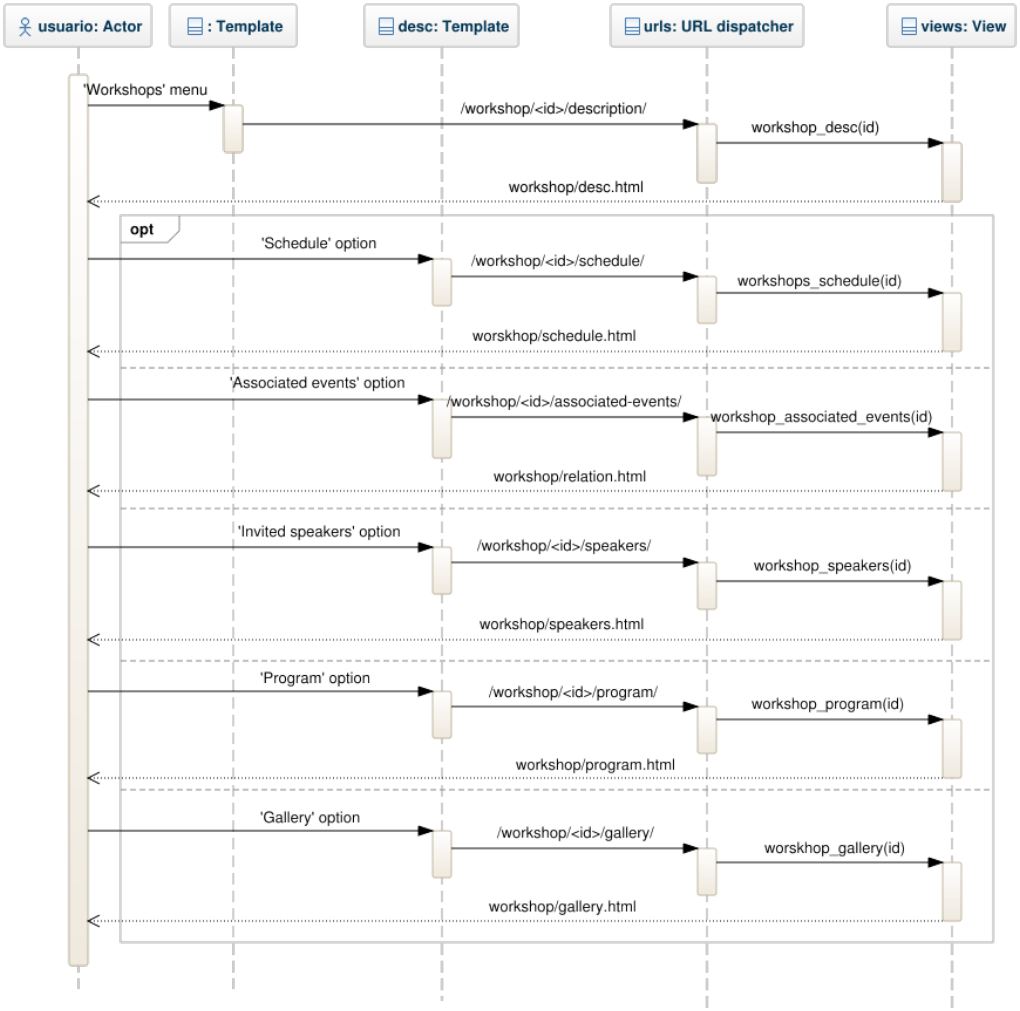

Figura 19: Diagrama de secuencia de UC07: Consultar información de workshops.

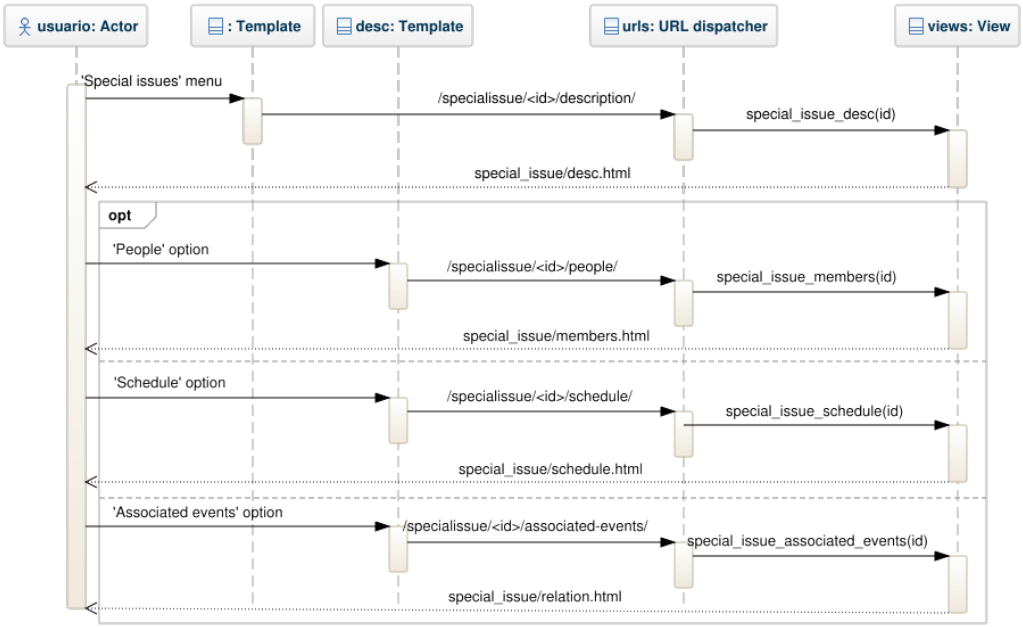

Figura 20: Diagrama de secuencia de UC08: Consultar información de special issues.

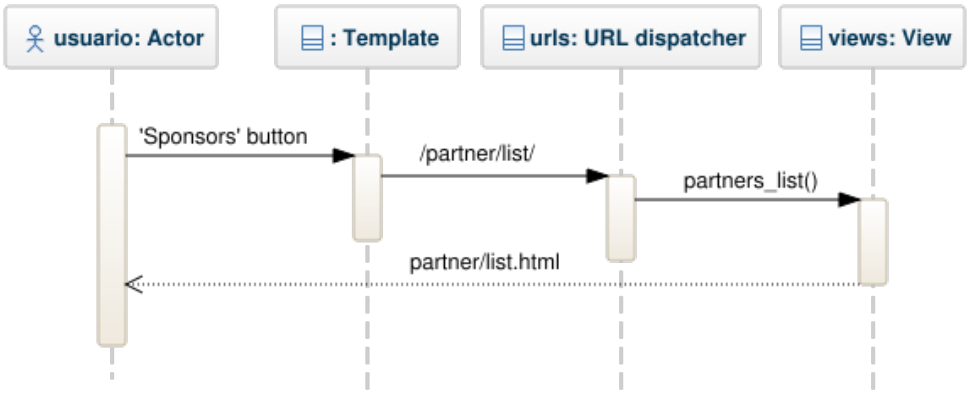

Figura 21: Diagrama de secuencia de UC09: Consultar información de sponsors.

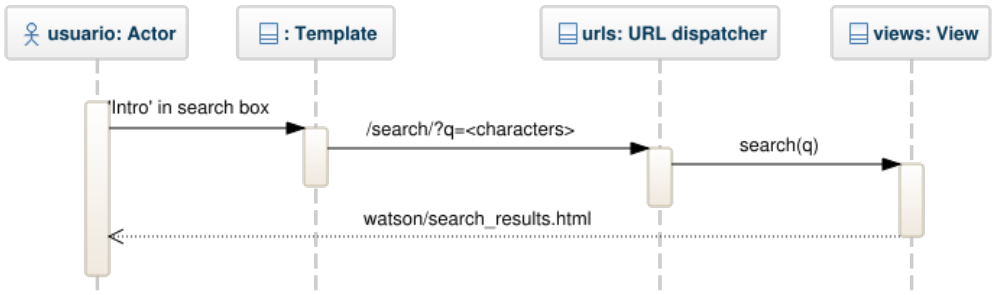

Figura 22: Diagrama de secuencia de UC10: Realizar búsqueda.

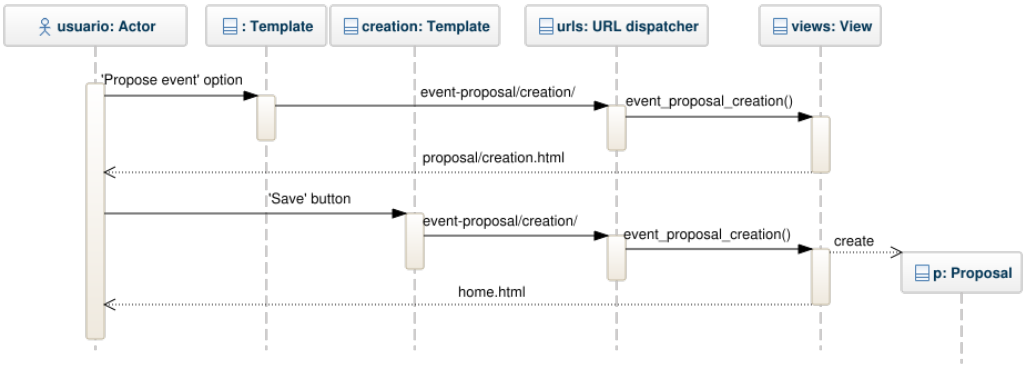

Figura 23: Diagrama de secuencia de UC11: Proponer evento.

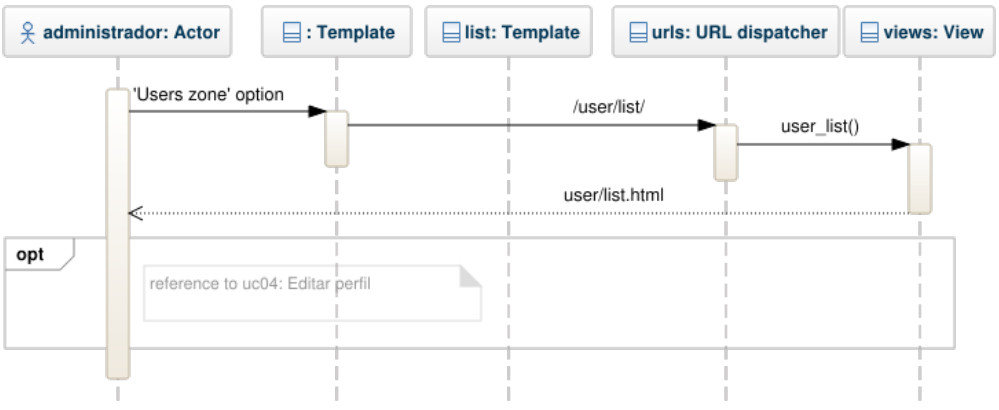

Figura 24: Diagrama de secuencia de UC12: Consultar lista de usuarios.

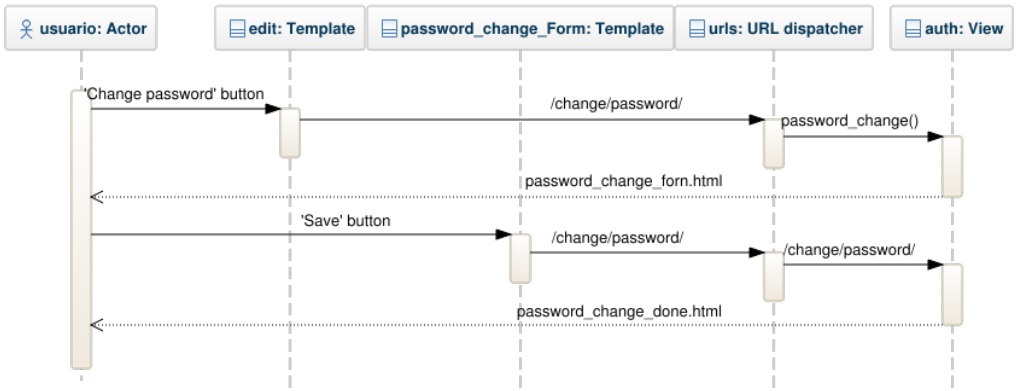

Figura 25: Diagrama de secuencia de UC13: Recuperar contraseña.

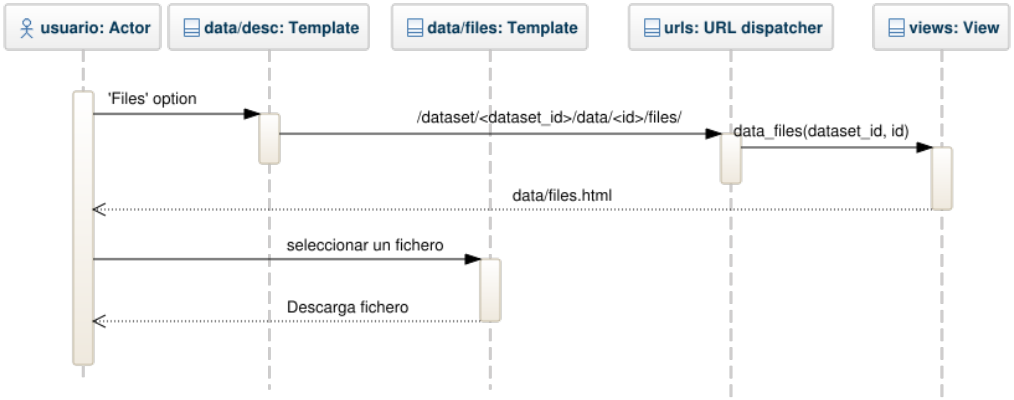

Figura 26: Diagrama de secuencia de UC14: Descargar ficheros.

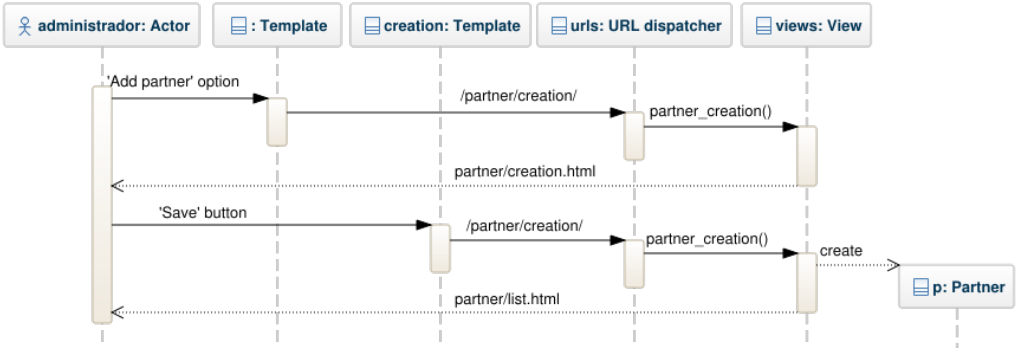

Figura 27: Diagrama de secuencia de UC15: Crear sponsor.

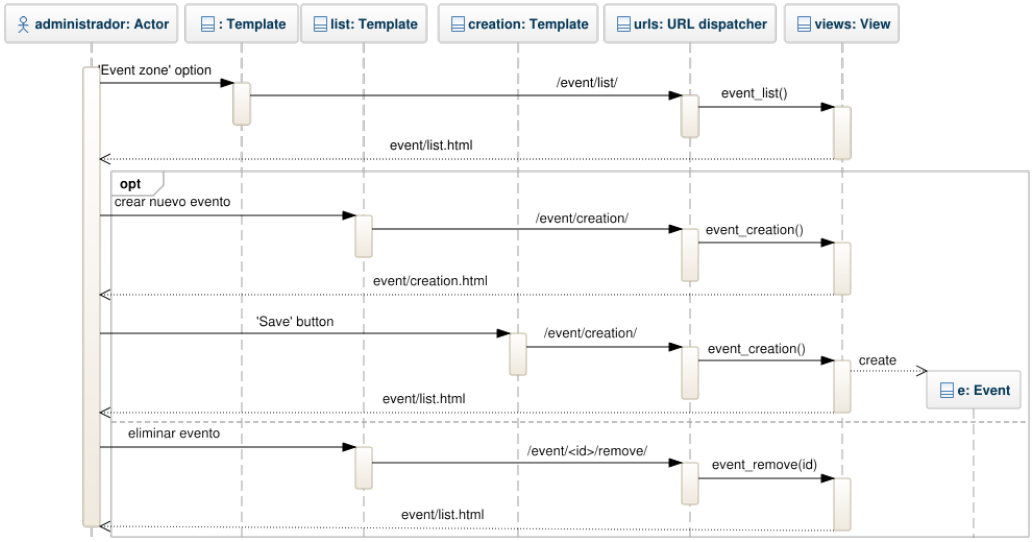

Figura 28: Diagrama de secuencia de UC16: Gestión de eventos.

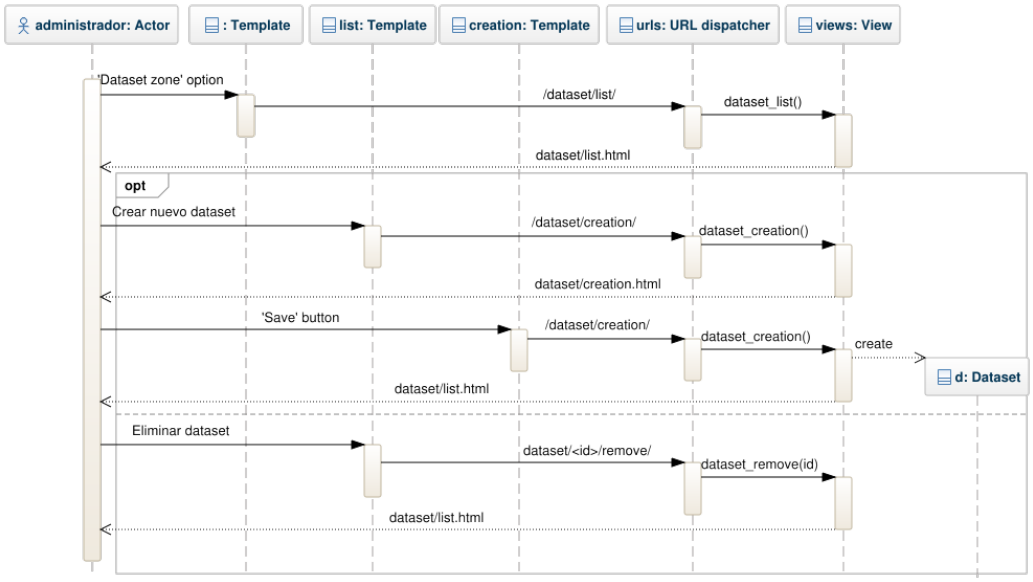

Figura 29: Diagrama de secuencia de UC17: Gestión de datasets.

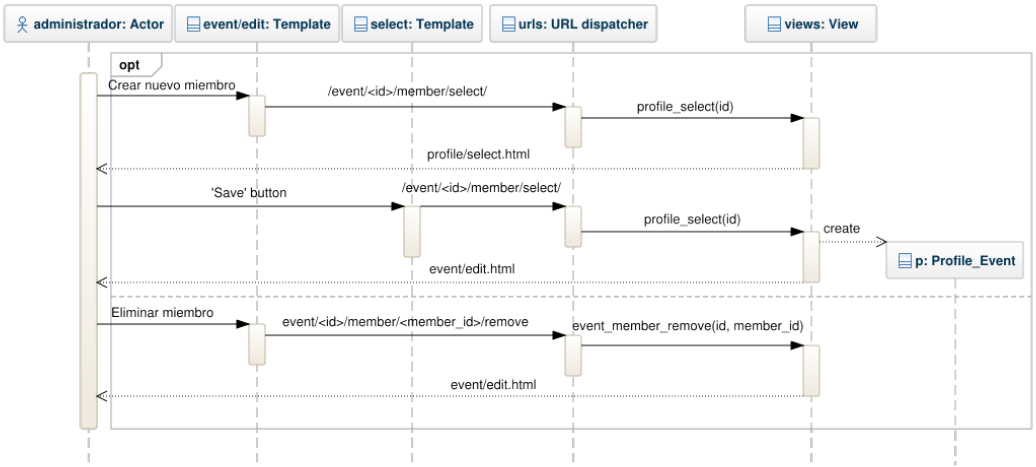

Figura 30: Diagrama de secuencia de UC18: Gestión de miembros.

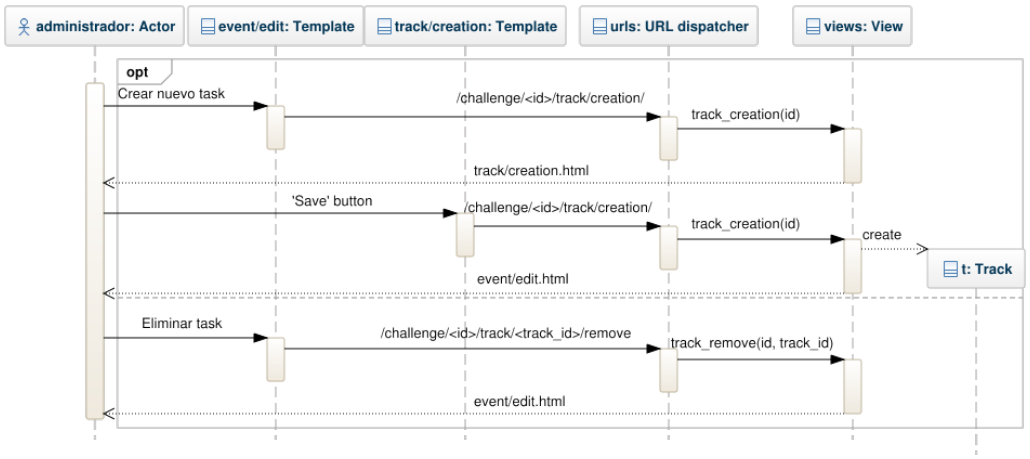

Figura 31: Diagrama de secuencia de UC19: Gestión de tasks.

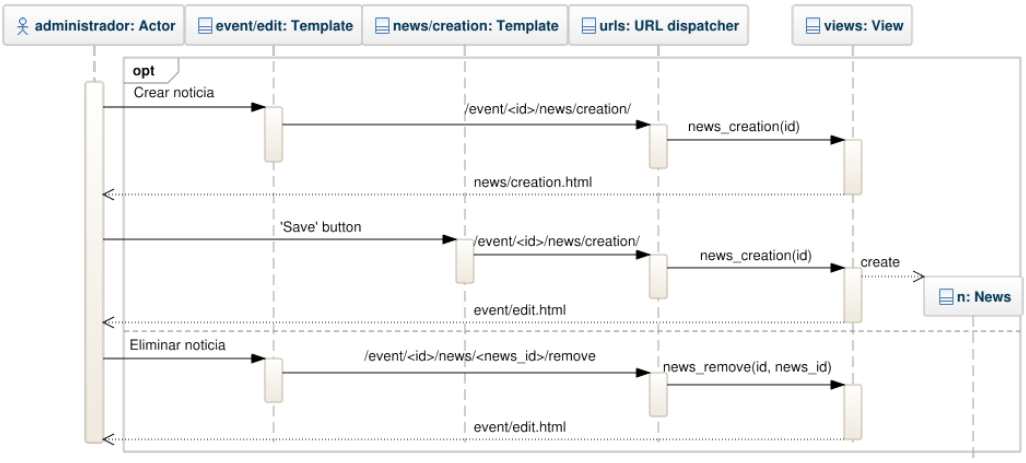

Figura 32: Diagrama de secuencia de UC20: Gestión de noticias.

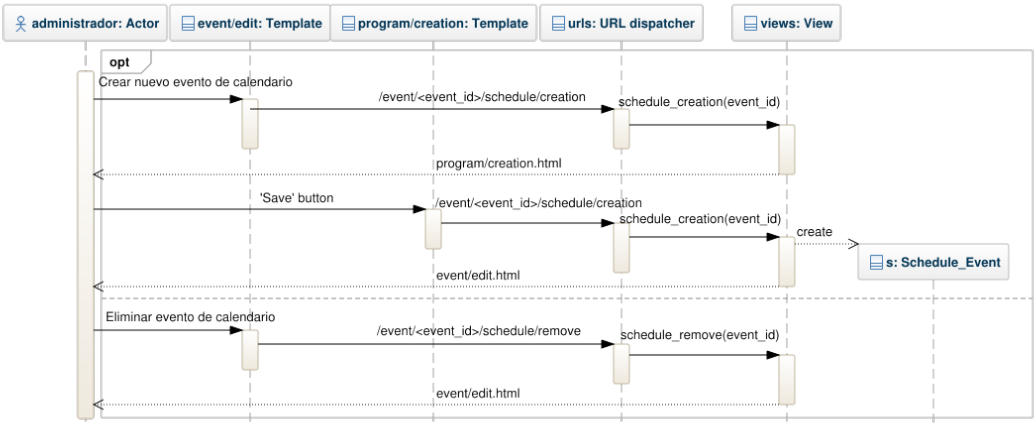

Figura 33: Diagrama de secuencia de UC21: Gestión del calendario.

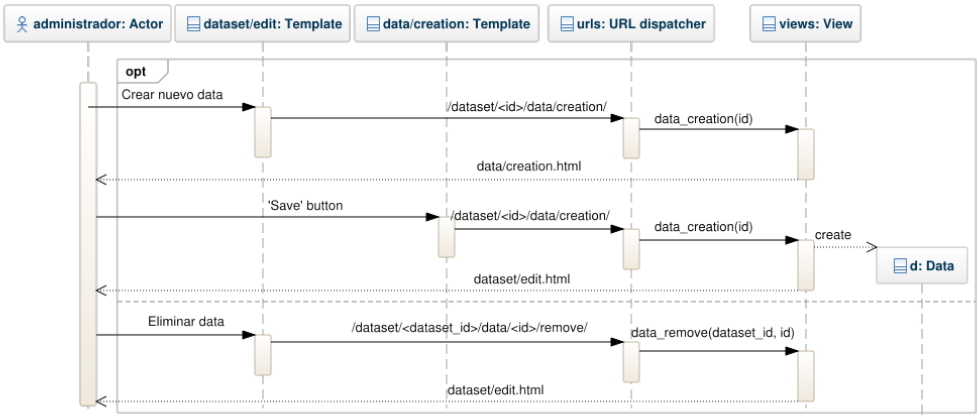

Figura 34: Diagrama de secuencia de UC22: Gestión de data.

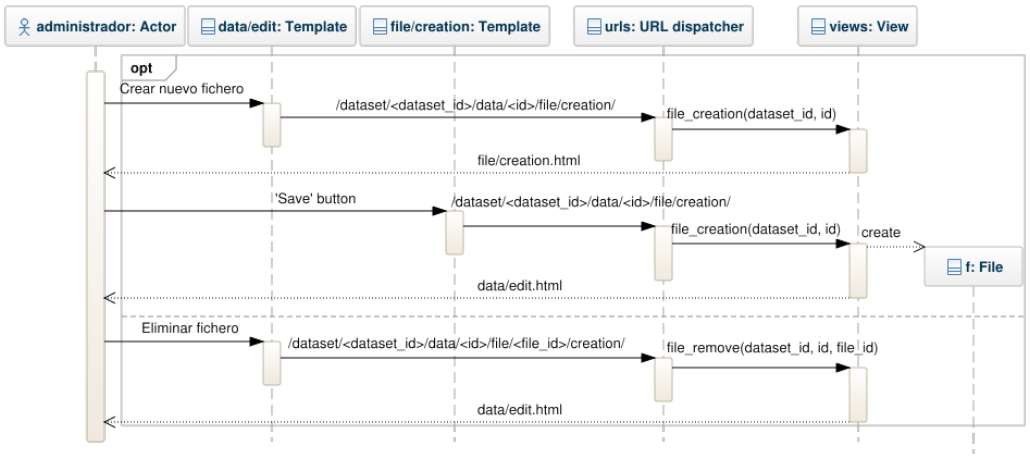

Figura 35: Diagrama de secuencia de UC23: Gestión de ficheros.

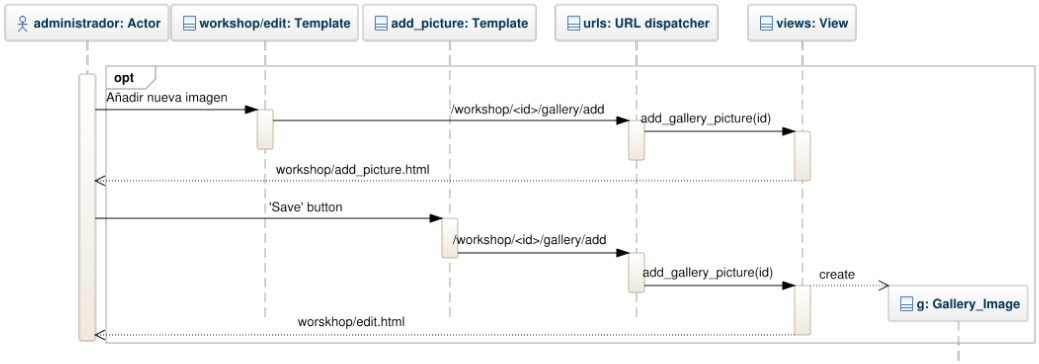

Figura 36: Diagrama de secuencia de UC24: Añadir imagen a la galería.

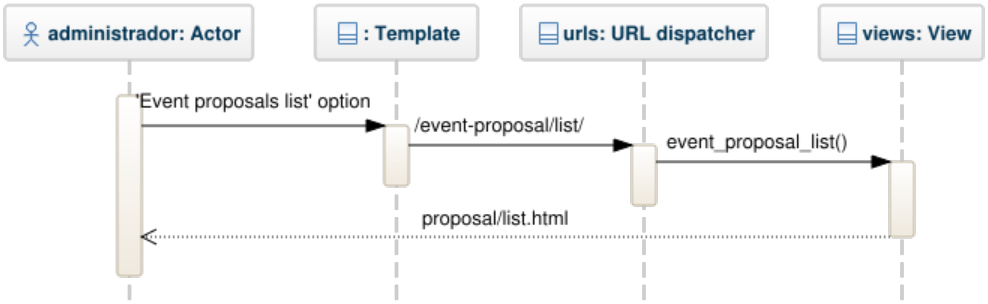

Figura 37: Diagrama de secuencia de UC25: Consultar listado de propuestas.

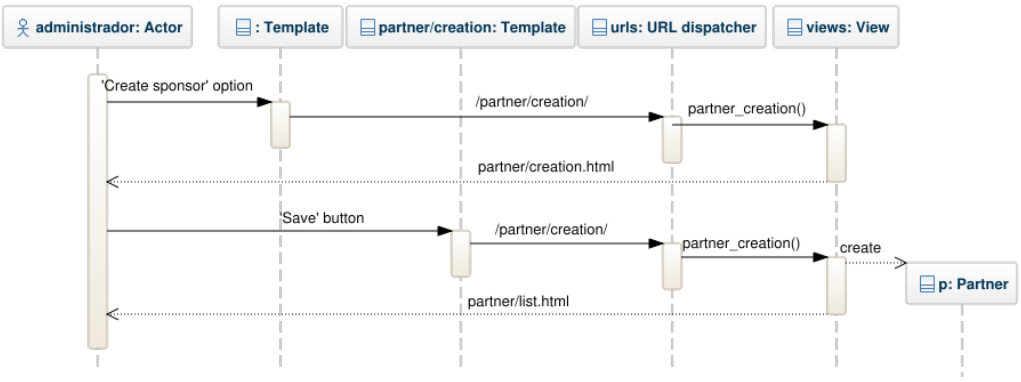

Figura 38: Diagrama de secuencia de UC26: Gestión de sponsors.

4.4. Detalles de implementación

Después de analizar los requerimientos y diseñar todos los componentes de la plataforma, es el momento del desarrollo del código siguiendo los diferentes modelos y diagramas. Para la realización de éste se ha seguido la técnica *Vertical Slice* [17], que consiste en construir pequeñas funcionalidades completas, des de la base de datos hasta la parte del cliente, e ir añadiéndolas a la plataforma.

Figura 39: Vertical slice.

Durante el desarrollo de la web, se han utilizado múltiples librerías que facilitan la implementación de una funcionalidad extra. A continuación se detallan los paquetes usados y se explica su cometido en la plataforma.

- jQuery [4]. Es una librería de JavaScript que se ha utilizado para simplificar la interacción en el código HTML.

- Bootstrap [3]. Es un *front-end web framework* que se ha usado para el diseño estético de la plataforma. Permite realizar un estilo *responsive* de una manera sencilla.

- django-registration-redux [5]. Es un paquete de Django utilizado para manejar la sesión de un usuario.

- django-ckeditor [6]. Es un paquete de Django que sirve para añadir un campo de *Rich Text* en un formulario.

- django-watson [8]. Es un paquete de Django usado para que el usuario pueda realizar búsquedas en los contenidos de la web.

- django-datetime-widget [9]. Es un paquete de Django que contiene un campo para indicar fecha y hora en un formulario.

- django-jfu [10]. Es un paquete de Django utilizado para manejar la subida de ficheros en la plataforma.

- django-sorl-thumbnail [11]. Es un paquete de Django usado para la visualización de imágenes.

- select2 [7]. Es un paquete de JQuery que contiene un campo para insertar diferentes elementos, para los formularios.

5. Validación

Al desarrollar cada pequeña funcionalidad se introducía en la plataforma y se realizaban pruebas con diversos usuarios, para comprobar los posibles fallos de este componente y posteriormente corregirlos.
Al finalizar la implementación de todas las funcionalidades e insertarlas en la plataforma, se procedió a crear datos dentro de la web, para poder realizar una prueba completa e intentar encontrar fallos en la usabilidad de la plataforma. Por separado se realizó un test de velocidad mediante la página de Google para desarrolladores [18].

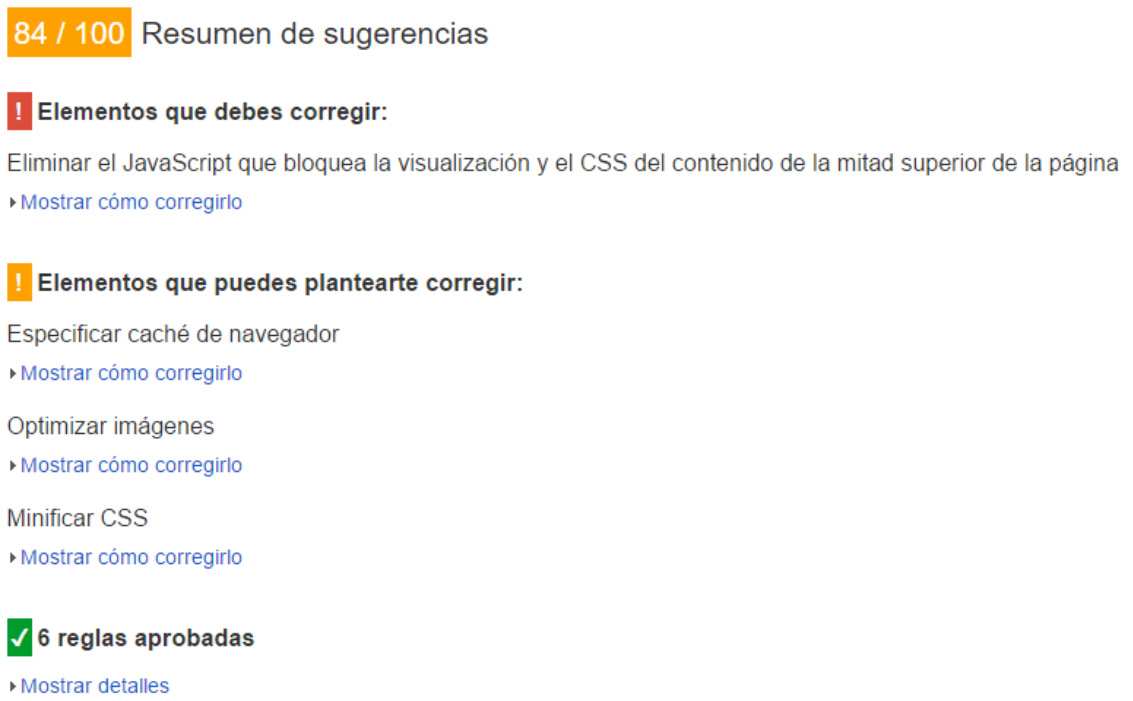

Figura 40: Test de velocidad.

En la figura 40 se puede observar que la velocidad es notable, ya que obtiene 84 puntos sobre 100. Aun así, se puede mejorar el rendimiento modificando la carga de los ficheros de estilo (CSS) y los archivos de Javascript, situándolos al final de la página. Según el test, también se puede optimizar las imágenes y hacer más pequeños los ficheros de estilo para aumentar la velocidad de carga.

6. Conclusiones

A continuación se muestran los objetivos definidos al principio del proyecto, y las conclusiones finales sobre cada uno.

- Analizar las diferentes tecnologías web actuales, con el objetivo de escoger las más convenientes para el proyecto, y estudiar e investigar en profundidad éstas para poder aplicarlas.
  Durante aproximadamente diez días se analizaron diversas opciones sobre las tecnologías y *web framework* a utilizar en la plataforma, teniendo en cuenta los requerimientos de la web. Finalmente se decidió por usar el *web framework* de Python, Django.

- Realizar el diseño de las diferentes partes de la plataforma.
  A partir del conjunto de requerimientos de la web, se realizaron los casos de uso para cada tipo de usuario, junto con el diseño de la base de datos mediante un modelo relacional.

- Desarrollar la web, según las necesidades requeridas por los usuarios.
  Se implementaron las múltiples funcionalidades de la plataforma, siguiendo los diseños anteriormente realizados, y utilizando los lenguajes HTML, CSS, Javascript y Python.

- Realizar las pruebas necesarias para confirmar el buen funcionamiento de la plataforma, situada en un servidor.
  Durante el desarrollo se realizaron pruebas al finalizar cada funcionalidad, corrigiendo los problemas que aparecían.

La implementación ha sido la fase que más tiempo se ha dedicado, más específicamente en la inclusión de las librerías citadas en el apartado 4.4, donde la principal dificultad ha estado en la personalización de algunos componentes de éstas, ya que la mayoría están enfocadas a un uso de forma estándar.
Finalmente, considero que he alcanzado un buen nivel en el proyecto, cumpliendo todos los objetivos marcados al principio.

6.1. Futuras ampliaciones

Para mejorar la plataforma necesitaría ser probada en un entorno real para poder detectar posibles nuevas mejoras funcionales, e ir añadiéndolas a la web.
Una posible mejora es la opción, por parte de los usuarios de tipo administrador, de conceder la posibilidad de exportar información de la plataforma en un fichero, como por ejemplo obtener todos los usuarios registrados en la plataforma.
Otra posible mejora podría ser incluir *scripts* de evaluación, que analizarían los resultados subidos por los usuarios en la plataforma, para poder realizar una buena criba.
Otra posible mejora podría ser cambiar del protocolo HTTP a HTTPS, para incluir

más seguridad en la transferencia de datos sensibles, como pueden ser las contraseñas de los usuarios.
Finalmente se podría mejorar el diseño responsivo de la plataforma, ya que actualmente está incompleto.

7. Referencias

Referencias

[1] Documentación Django
https://docs.djangoproject.com/en/1.9/

[2] Documentación Django Template Language
https://docs.djangoproject.com/es/1.9/ref/templates/language/

[3] Documentación Bootstrap
http://getbootstrap.com/css/

[4] Documentación jQuery
http://api.jquery.com/

[5] Documentación librería django-registration-redux
http://django-registration-redux.readthedocs.io/en/stable/

[6] Documentación librería ckeditor
https://github.com/django-ckeditor/django-ckeditor

[7] Documentación select2
https://select2.github.io/examples.html

[8] Documentación librería django-watson
https://github.com/etianen/django-watson/wiki

[9] Documentación librería django-datetime-widget
https://github.com/asaglimbeni/django-datetime-widget/

[10] Documentación librería django-jfu
https://github.com/Alem/django-jfu

[11] Documentación librería sorl-thumbnail
http://sorl-thumbnail.readthedocs.io/en/stable/

[12] Esqueleto casos de uso
https://github.com/izimakin/LaTeX/tree/master/UseCase

[13] Comparación de lenguages
https://blog.udemy.com/modern-language-wars/

[14] Arquitectura Django
http://es.slideshare.net/balakumarp/django-framework

[15] Documentación de Vagrant
https://www.vagrantup.com/docs/

[16] Diagrama de Gantt
http://www.tomsplanner.es/

[17] Vertical slice
https://en.wikipedia.org/wiki/Vertical_slice

[18] Test de velocidad
https://developers.google.com/speed/pagespeed/insights/

Anexos

A. Wireframe completo

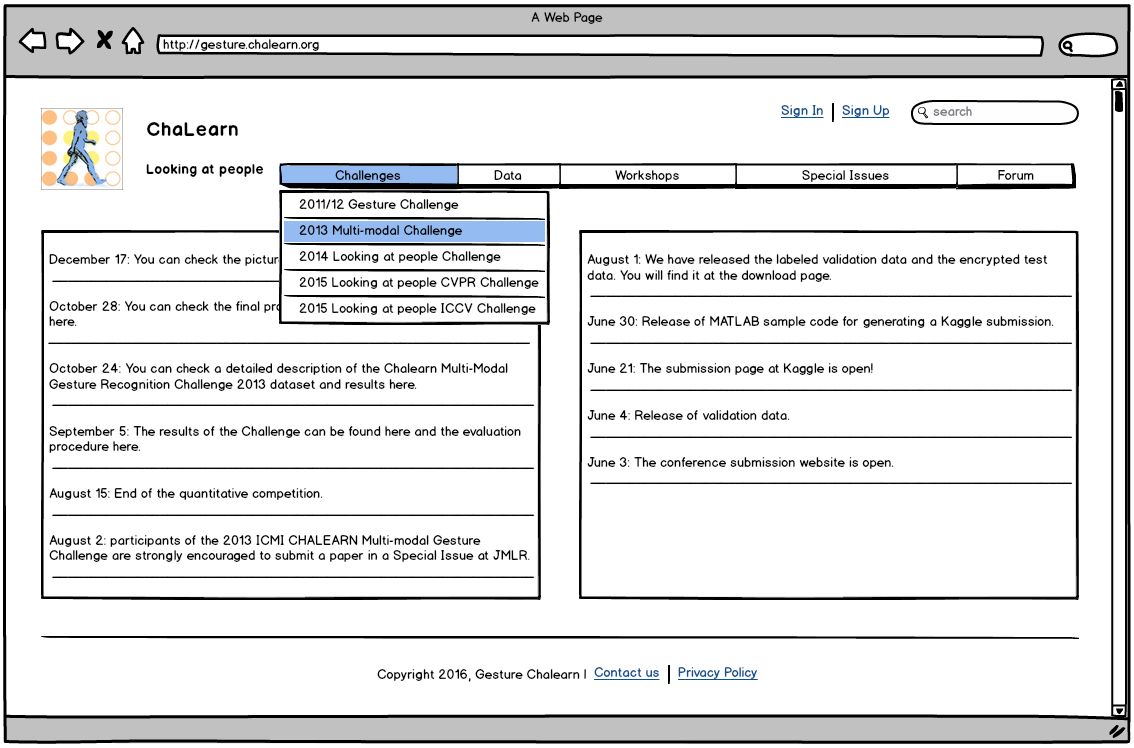

Figura 41: Página principal con challenges.

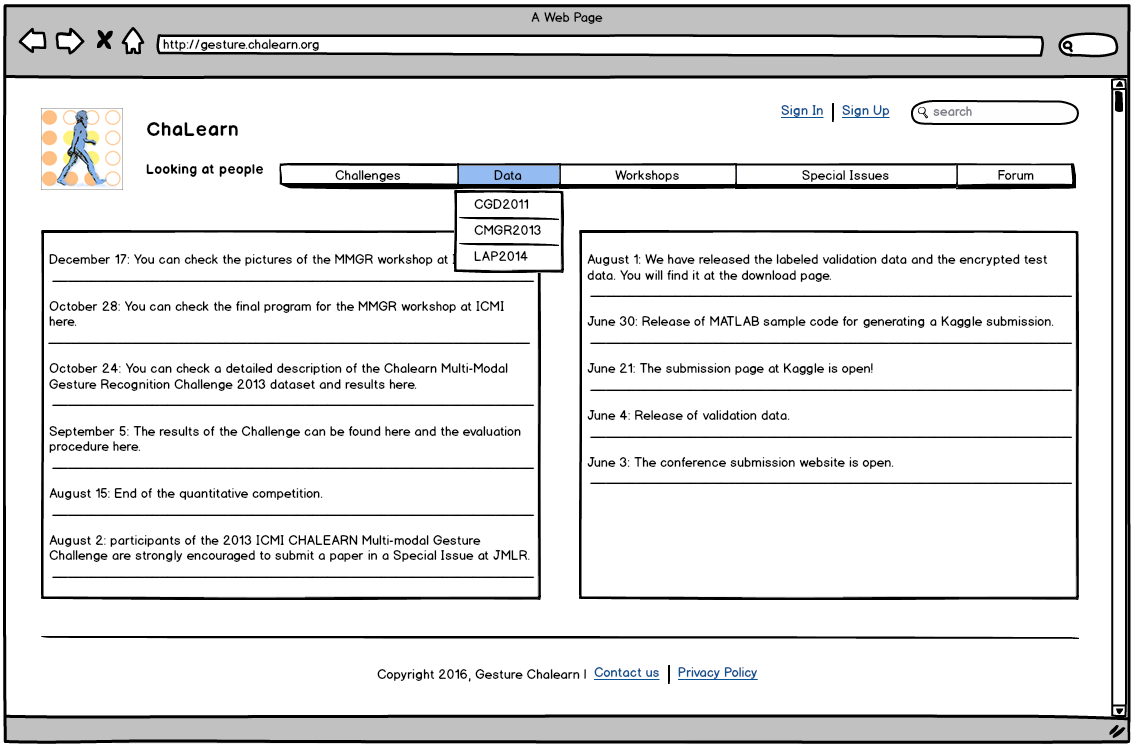

Figura 42: Página principal con datasets.

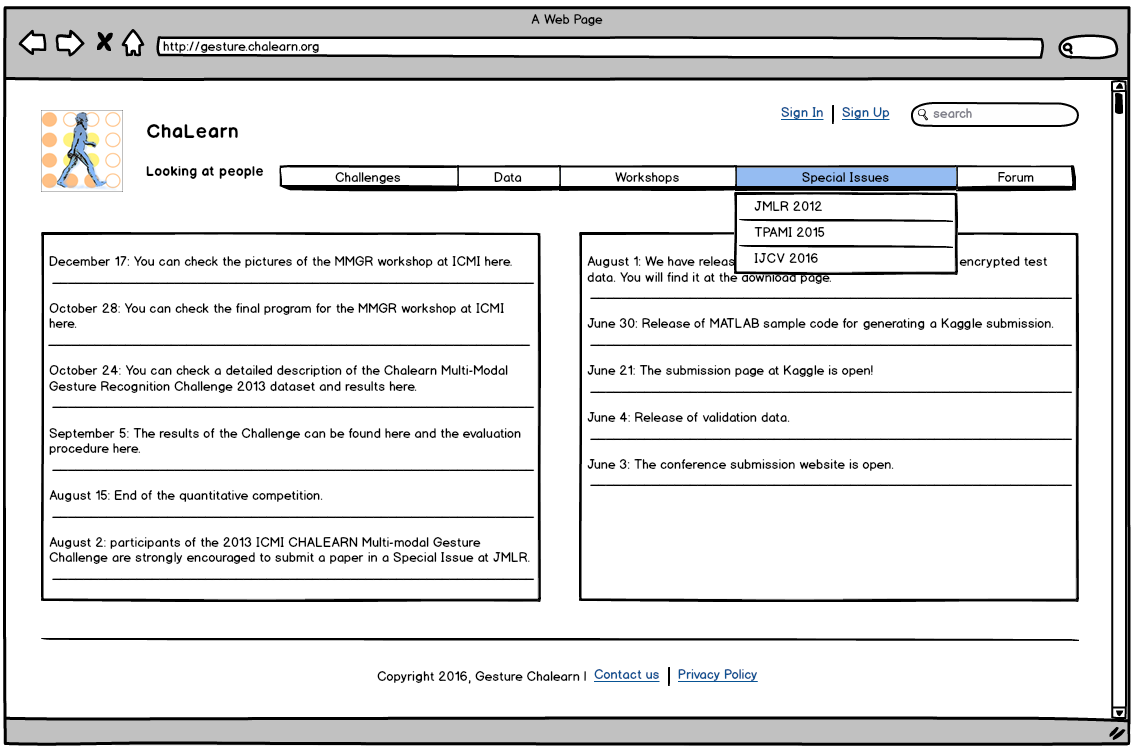

Figura 43: Página principal con special issues.

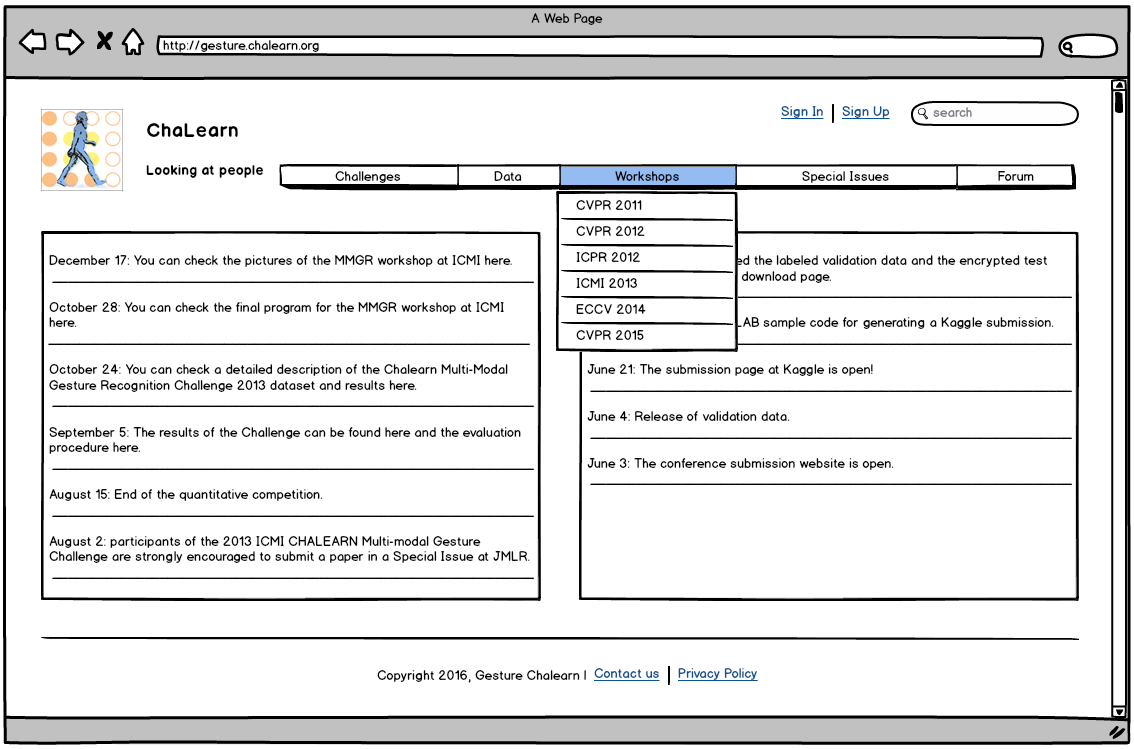

Figura 44: Página principal con workshops.

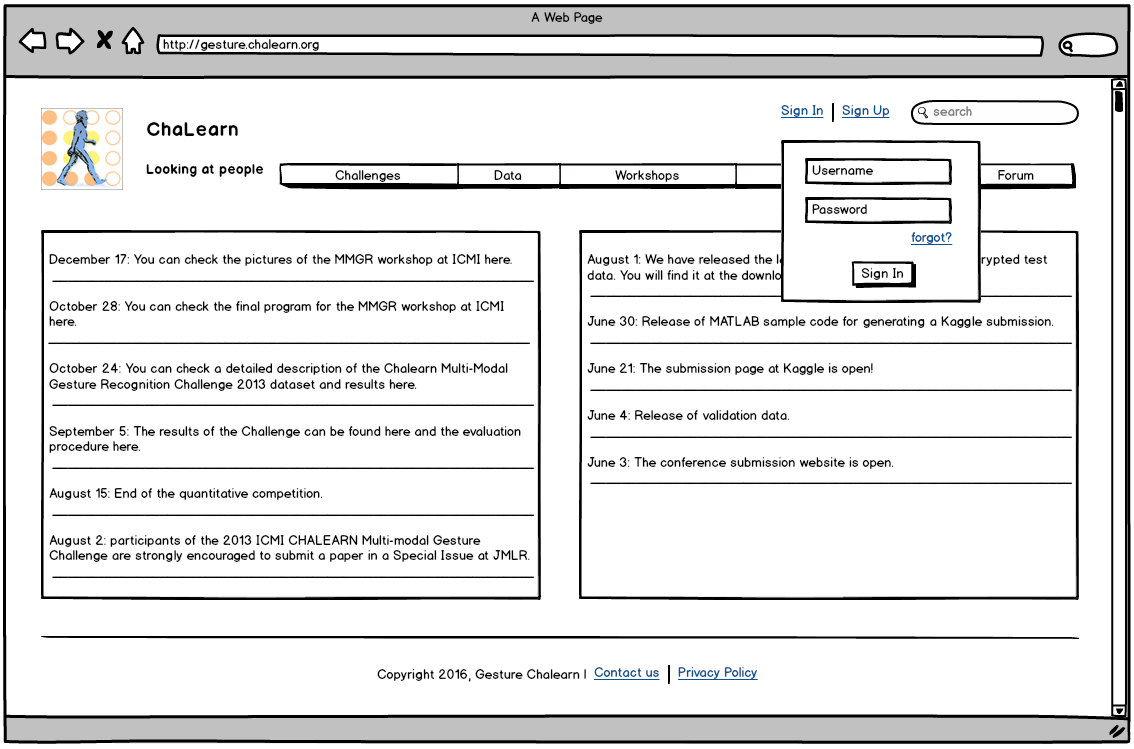

Figura 45: Página de inicio de sesión, versión 1.

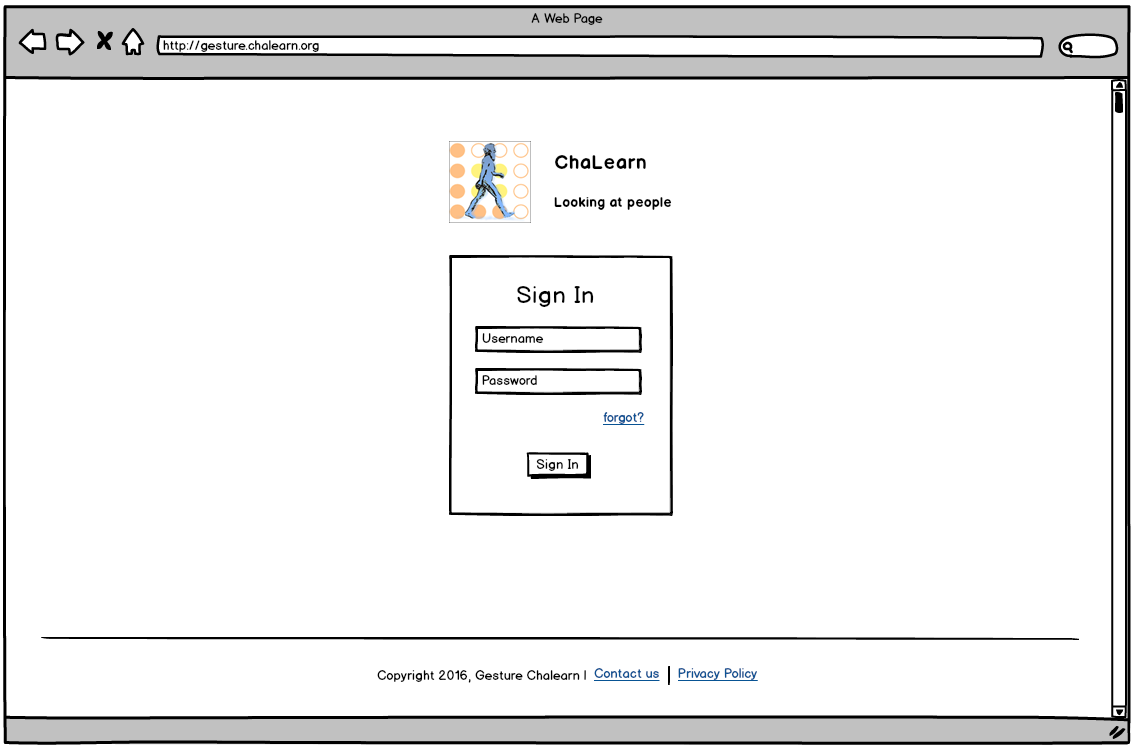

Figura 46: Página de inicio de sesión, versión 2.

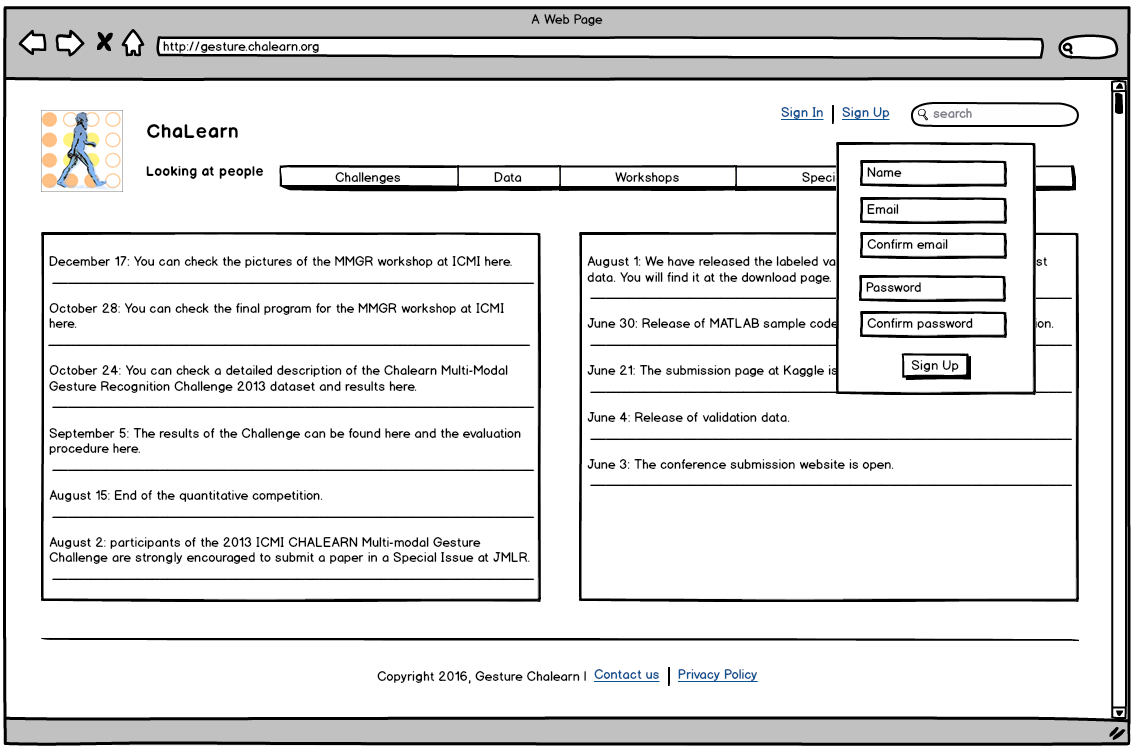

Figura 47: Página de registro, versión 1.

Figura 48: Página de registro, versión 2.

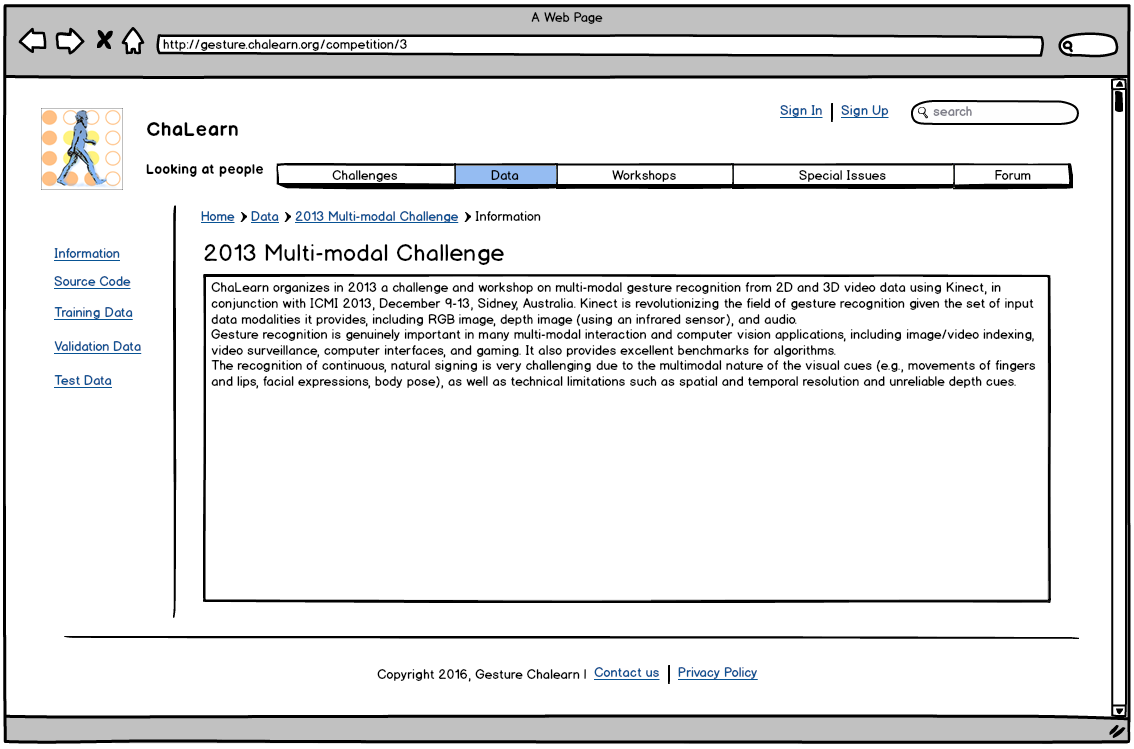

Figura 49: Página de descripción de un dataset.

B. Manual de usuario

Este manual está dirigido tanto a los usuarios como a los administradores de la plataforma. Se describen las funcionalidades más importantes, guiando paso a paso la forma de realizarlas.

Registro de cuenta

El primer paso consiste en seleccionar el botón *SIGN UP*, como refleja la figura 50.

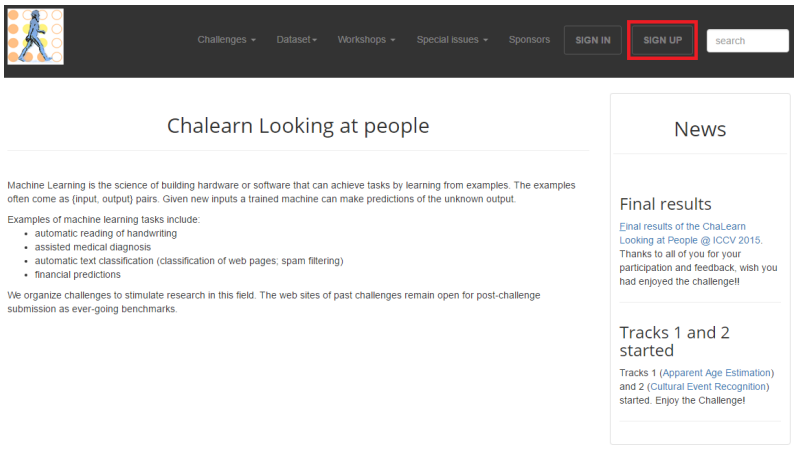

Figura 50: Página principal.

Seguidamente aparece un formulario como en la figura 51, que se ha de rellenar con la información del usuario y finalmente seleccionar el botón *Register*.

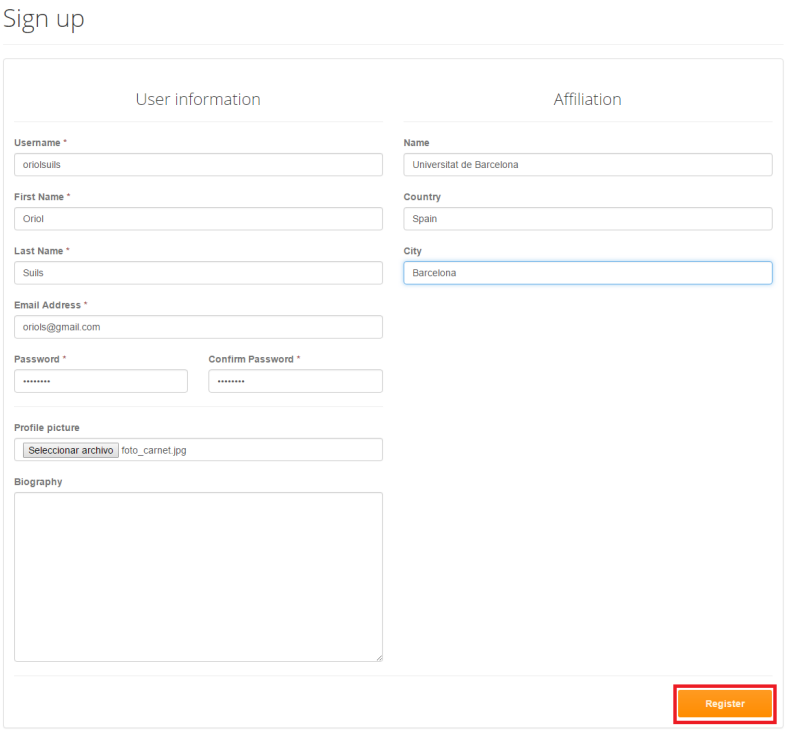

Figura 51: Página de registro.

Inicio de sesión

El primer paso consiste en seleccionar el botón *SIGN IN*, como refleja la figura 52.

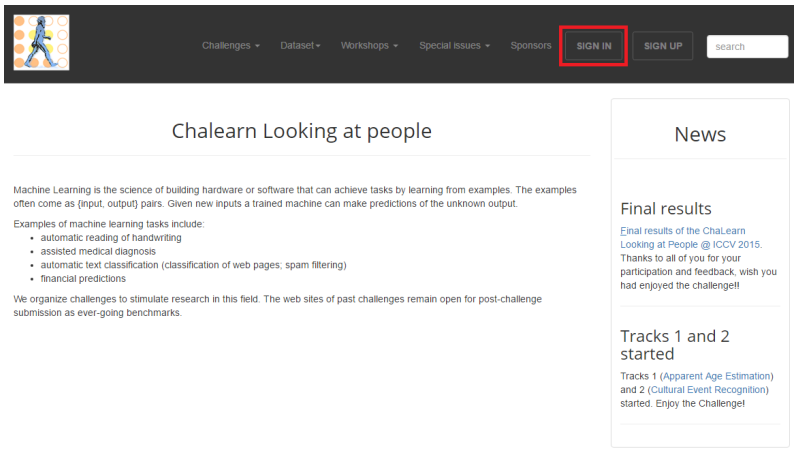

Figura 52: Página de principal.

El siguiente paso consiste en introducir los datos del usuario en el formulario de la figura 53, y seleccionar el botón *Sign in*. El usuario ha de registrarse en la plataforma antes de realizar esta acción.

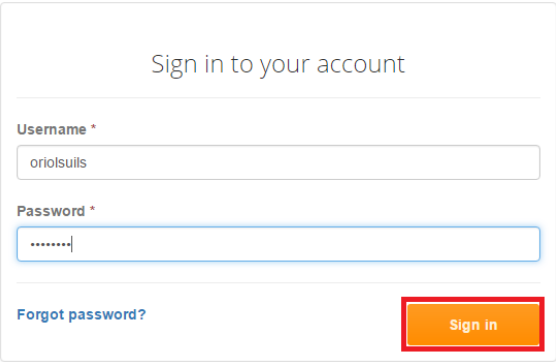

Figura 53: Página de inicio de sesión.

Proponer evento

Antes de poder realizar esta acción es necesario iniciar sesión. El primer paso consiste en seleccionar la opción *Propose event* en el menú de usuario como se muestra en la figura 54.

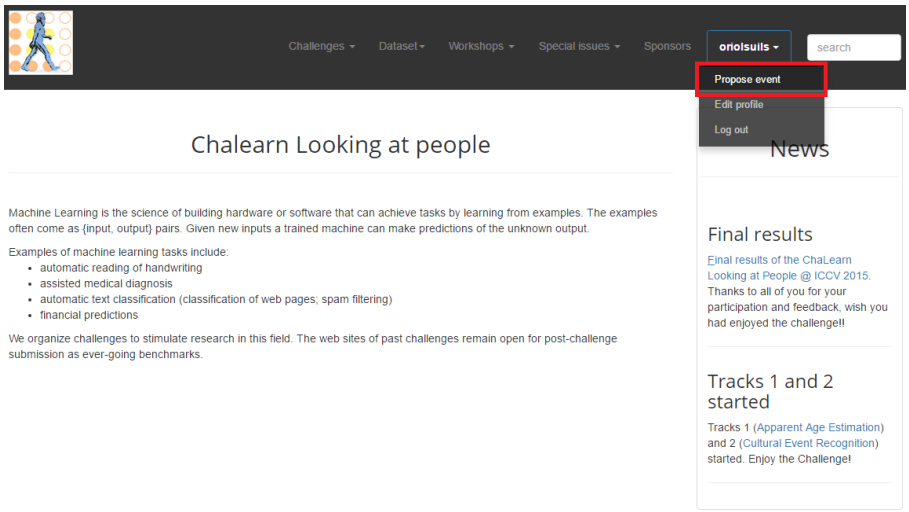

Figura 54: Página de principal.

El siguiente paso consiste en rellenar todos los campos del formulario de la figura 55 y seleccionar el botón *Save*. El administrador analizará la propuesta recibida y si lo cree necesario creará el evento, dando la oportunidad de editarlo al usuario.

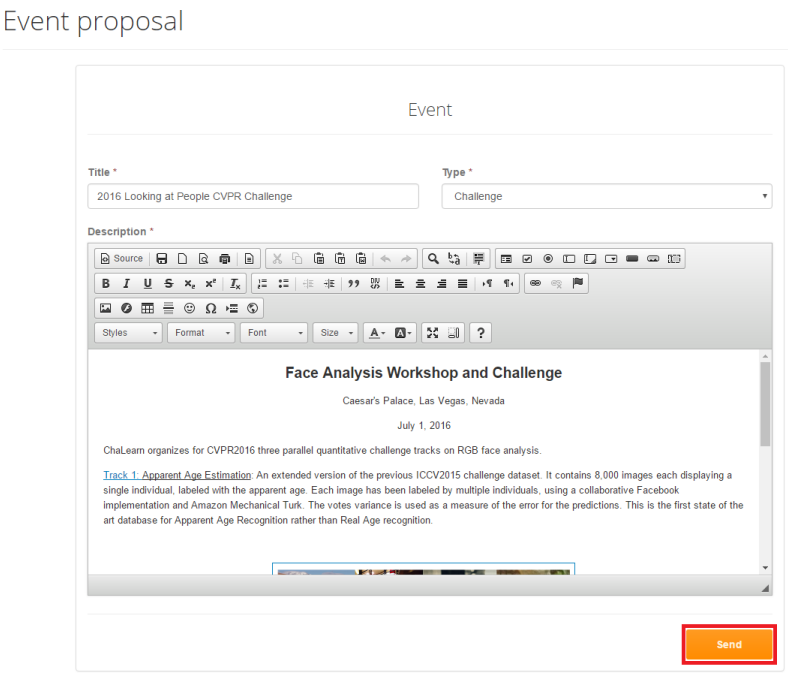

Figura 55: Página de creación de propuesta.

Descargar ficheros de datos y subir resultados

Al finalizar una competición se publican los datos utilizados en ésta, dando la posibilidad a todos los visitantes de la plataforma de descargar el conjunto de datos. El primer paso es acceder a la información de un *dataset* concreto como se muestra en la figura 56.

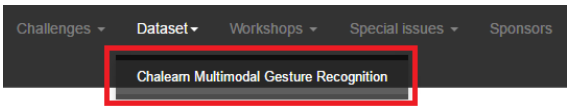

Figura 56: Página principal, selección de un *dataset*.

Seguidamente se accede a un *data* concreto como en la figura 57.

Figura 57: Página de descripción de un *dataset*.

Finalmente se selecciona la opción *Files* y se descargan los ficheros necesarios, como se muestra en la figura 58.

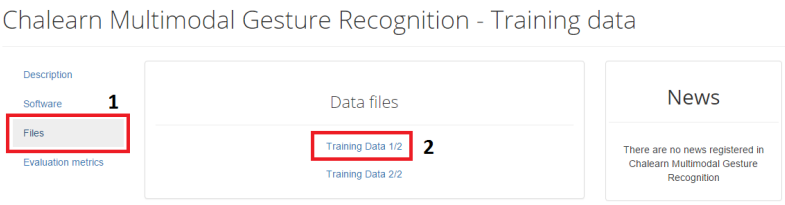

Figura 58: Página de ficheros de un *data*.

Después de descargar los datos de la competición el usuario puede desarrollar su código y subir sus resultados para compararlos. Primero de todo ha de iniciar sesión en la web. Más tarde ha de acceder a la información del dataset en cuestión, y seleccionar la opción *Results* donde ha de hacer clic sobre el botón *Create Submission*, como en la figura 59.

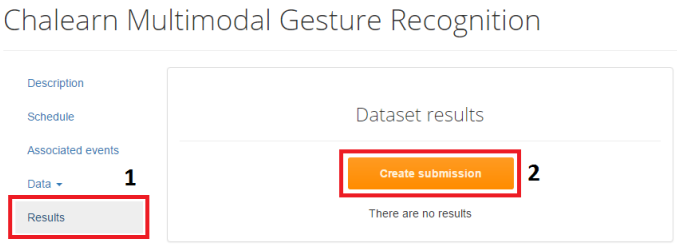

Figura 59: Página de resultados de un *dataset*.

Seguidamente el usuario ha de rellenar el formulario de la figura 60, indicando un enlace al repositorio del código, otro enlace al *paper*, y los resultados numéricos obtenidos. Finalmente seleccionar el botón *Save*.

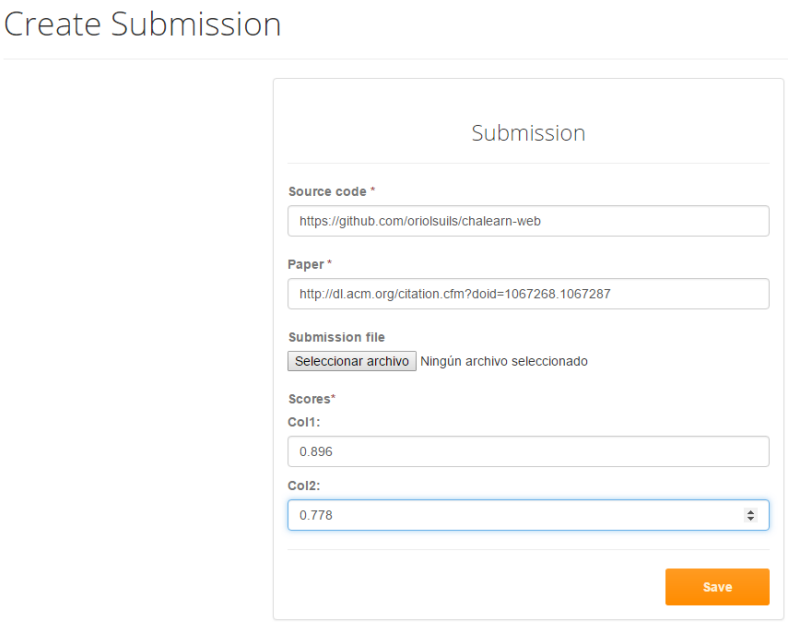

Figura 60: Página de creación de un resultado.

C. Manual de administrador

Las siguientes funcionalidades solo están permitidas para los usuarios de tipo administrador.

Crear nuevo evento

Antes de poder realizar esta acción es necesario haber iniciado sesión y ser un usuario de tipo administrador. Para la creación de un nuevo evento existen dos vías, la primera consiste en crear un nuevo evento mediante las propuestas realizadas por los usuarios, y la segunda consiste en crearlo desde cero.
La primera opción se realiza accediendo al botón *Event Proposals list* dentro del menú de usuario, como se muestra en la figura 61.

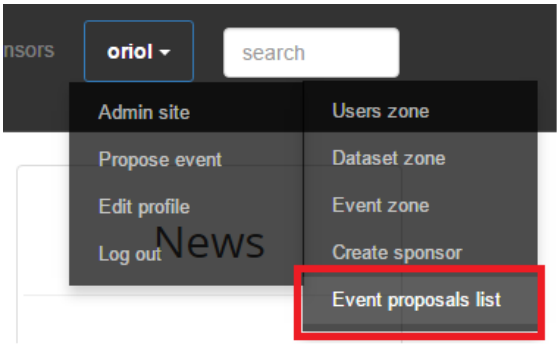

Figura 61: Página principal, opciones del menú de usuario.

Más tarde aparece una lista con las propuestas realizadas por los usuarios, como en la figura 62. Se selecciona una propuesta para observar los detalles.

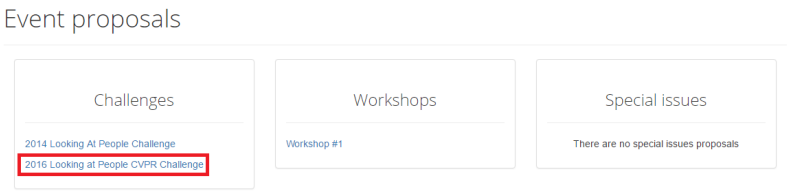

Figura 62: Página del listado de propuestas.

En la figura 63 se puede observar la información de la propuesta realizada por un usuario. Para crear el evento se selecciona el botón *Create event*.

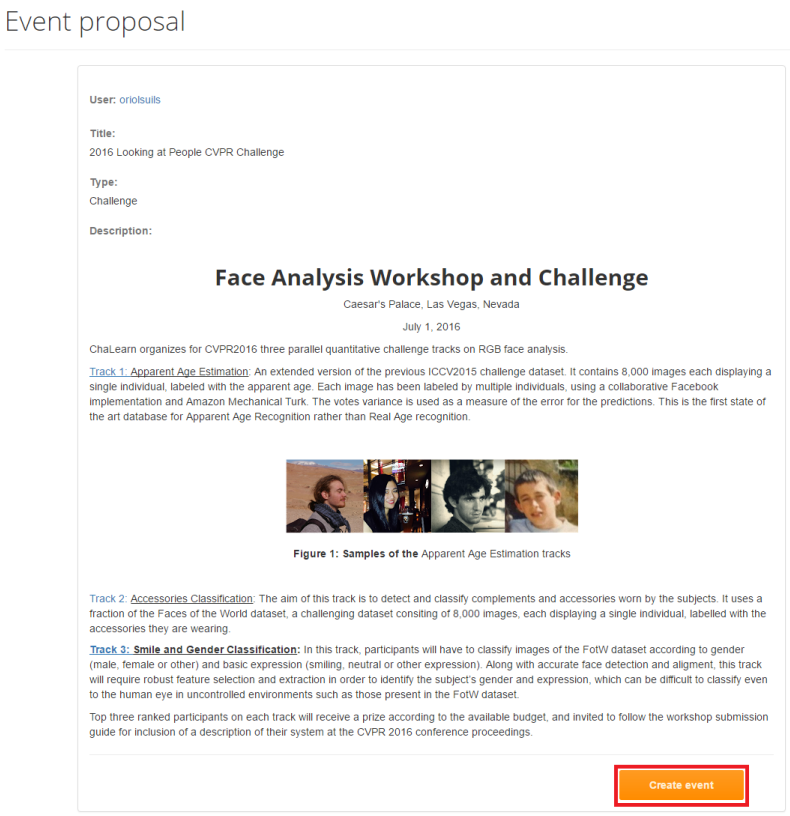

Figura 63: Página de la información de una propuesta.

La segunda opción para crear un evento consiste en seleccionar la opción *Event zone* como en la figura 64.

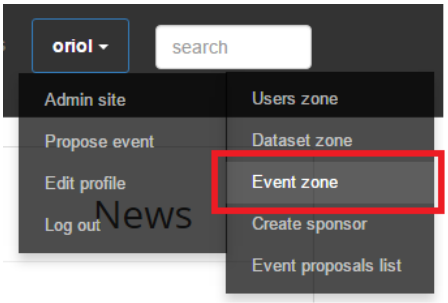

Figura 64: Página principal, menú de usuario.

Seguidamente se selecciona el botón para crear un nuevo evento, como se muestra en la figura 65.

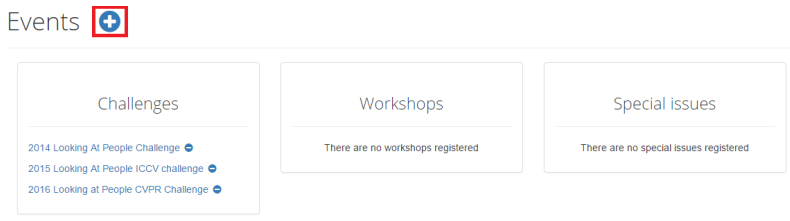

Figura 65: Página del listado de eventos creados.

Y finalmente se rellena el formulario de la figura 66 indicando la información básica del evento y se selecciona *Save*.

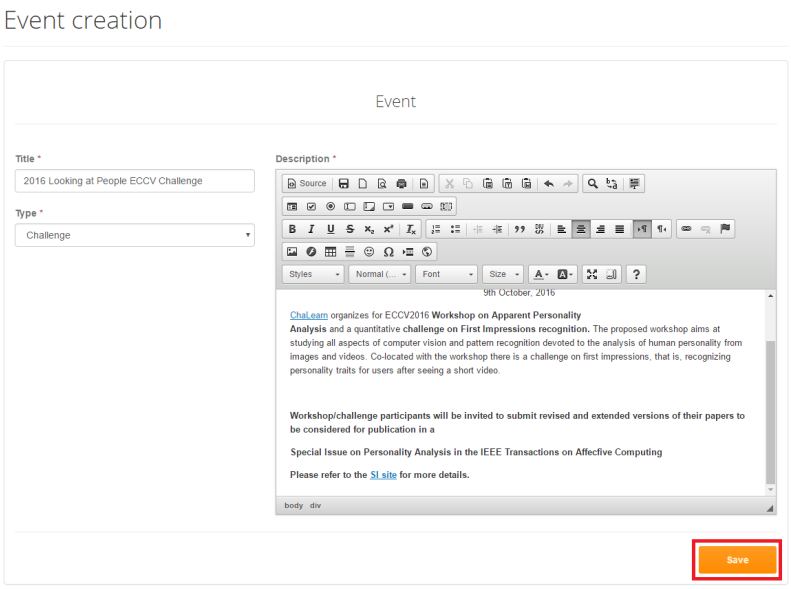

Figura 66: Página de creación de un evento.

Editar evento

El administrador puede editar cualquier evento que se haya creado en la plataforma. Para poder realizar esta acción se ha de seguir los siguientes pasos.
El usuario ha de iniciar sesión en la plataforma, siendo administrador. Seguidamente ha de seleccionar la opción *Event zone*, como la figura 64. Como se muestra en la figura 67, el resultado de esta selección es una lista de los eventos creados en la plataforma, para editar uno de ellos se selecciona el evento deseado.

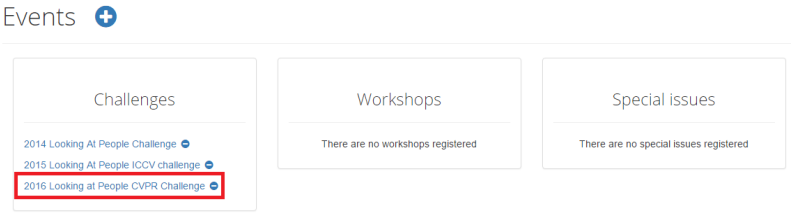

Figura 67: Página del listado de eventos creados.

En la figura 68 se muestran todas las opciones de edición de un evento, en este caso un *challenge*. Podemos dividir la página en cuatro zonas de posibles modificaciones.

- La zona número uno sirve para modificar el título y la descripción del evento en cuestión. Si se modifica alguno de estos campos es necesario seleccionar el botón *Save* posteriormente.
- La segunda zona tiene múltiples opciones, que sirven para añadir, editar y eliminar diferentes elementos relacionados con el evento.

- La zona número tres contiene un campo para incluir un fichero de tipo CSV con los resultados de la competición. Si se modifica este campos también es necesario seleccionar el botón *Save* posteriormente.
- La cuarta zona contiene un botón para modificar la visibilidad del evento hacia los visitantes y usuarios de la plataforma.

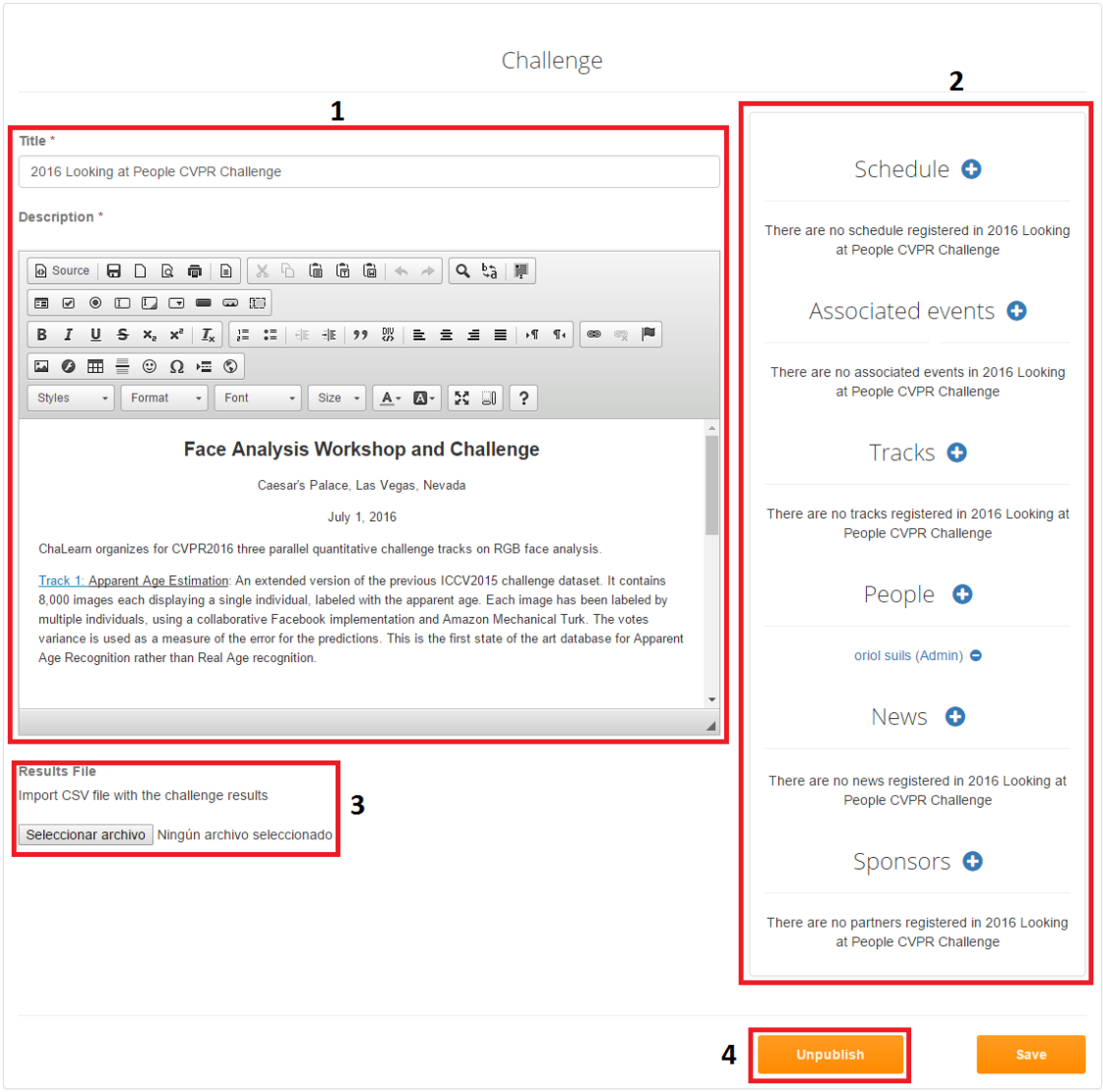

Figura 68: Página de edición de un evento.