

Trabajo Fin de Grado

Universando. El comparador de experiencias laborales

Autor

David Abad Tomás

Director

Francisco Javier López Pellicer

Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza

2015



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. DAVID ABAD TOMÁS,

con nº de DNI 17767097B en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
GRADO, (Título del Trabajo)
UNIVERSANDO. EL COMPARADOR DE EXPERIENCIAS LABORALES

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 22/08/2015

Fdo: DAVID ABAD TOMÁS

Universando. El comparador de experiencias laborales.

RESUMEN

Actualmente millones de personas comparten información sobre sus conocimientos y experiencias a través de la red. Esta tendencia ha llevado a la creación de cientos de productos web con características sociales que han conseguido un gran éxito a lo largo de los últimos años.

Este proyecto engloba el análisis, diseño, implementación y despliegue de un sistema que permite a estudiantes de las diferentes universidades españolas compartir opiniones anónimas sobre sus experiencias laborales en diferentes organizaciones, ya sea realizando proyectos, prácticas o trabajando como empleados. De este modo, los usuarios universitarios pueden contar con una guía de referencia que les ayude a decidir donde prefieren comenzar su vida laboral, mientras que las organizaciones pueden obtener gratuitamente la opinión de sus trabajadores para intentar mejorar sus competencias.

El sistema abarca aspectos como el tratamiento avanzado de información dada en lenguaje natural, utilizando técnicas de aprendizaje automático para afrontar problemas de elevada dificultad como el “análisis de sentimientos” y toda la problemática que conlleva. Este es un campo realmente complejo por lo que se ha creado un sistema capaz de aprender automáticamente de sus errores para mejorar su precisión a lo largo del tiempo.

Con el fin de enriquecer el contenido disponible, el sistema utiliza técnicas de recuperación de información para obtener, clasificar y almacenar ofertas de empleo. Para ello el sistema rastrea diferentes sitios web externos de forma concurrente y utilizando técnicas y estructuras de datos que permiten realizar el proceso de forma rápida, eficiente y obteniendo resultados de calidad.

Toda esta lógica queda expuesta al público mediante un conjunto de servicios web accesibles a través de una API RESTful, que permite una sencilla integración de la funcionalidad en cualquier tipo de cliente. Utilizando estos servicios se ha creado un sitio web con un diseño *responsive*, con el fin de proporcionar una interfaz gráfica sencilla y usable desde cualquier tipo de dispositivo, que permita a los usuarios utilizar las funcionalidades implementadas.

El sistema desarrollado se encuentra totalmente desplegado y accesible al público en la dirección <http://www.universando.es>.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivo del proyecto	1
1.3. Marco tecnológico	2
1.3.1. Tecnologías necesarias	2
1.3.2. Herramientas de desarrollo necesarias	3
1.4. Metodología	4
1.5. Estructura de la memoria	6
2. Trabajo realizado	7
2.1. Análisis	7
2.1.1. Casos de uso	7
2.1.2. Requisitos funcionales	9
2.1.3. Requisitos no funcionales	10
2.1.4. Conclusiones	11
2.2. Diseño	11
2.2.1. Arquitectura del sistema	11
2.2.2. Prototipo	13
2.2.3. Mapa de navegabilidad	14
2.2.4. Modelo de datos	15
2.2.5. Diseño de los servicios web	17
2.2.6. Conclusiones	19

2.3. Implementación	19
2.3.1. Uso del framework Spring	19
2.3.2. Búsqueda de organizaciones	20
2.3.3. Clasificación automática de opiniones	21
2.3.4. Obtención de ofertas de sitios externos	24
2.3.5. Tratamiento de opiniones	25
2.3.6. Aplicación web	26
2.3.7. Pruebas funcionales (Integración continua)	28
2.3.8. Conclusiones	28
3. Gestión del proyecto	29
3.1. Planificación	29
3.2. Tiempo dedicado	30
3.3. Herramientas de gestión	30
4. Conclusiones	31
4.1. Resultados	31
4.2. Lecciones aprendidas	31
4.3. Conclusión personal	32
4.4. Futuro del proyecto	32
BIBLIOGRAFÍA	33

Anexo A. Casos de uso detallados	37
Anexo B. Detalles de implementación	71
B.1. Detección de información en el proceso de scraping	71
B.2. Paquetes del sistema	73
B.3. Pruebas detalladas	74
B.4. Técnicas SEO utilizadas	79
B.5. Librerías utilizadas	80
Anexo C. Detalles de gestión	83
C.1. Tiempo empleado	83
C.2. Gráficos burndown	84
C.3. User stories	86
Anexo D. Manuales	93
D.1. Manual de uso (Usuario universitario)	93
D.2. Manual de uso (Usuario organización)	105
D.3. Manual de uso (Usuario administrador)	112
D.4. Manual de despliegue	119
Anexo E. Prototipo vs. Resultado final	123
Anexo F. Especificación del API RESTful	135

Índice de figuras

1.1. Metodología de desarrollo ágil	4
1.2. Lean Canvas	5
2.1. Casos de uso (Anónimo)	7
2.2. Casos de uso (Universitario)	8
2.3. Casos de uso (Organización)	8
2.4. Casos de uso (Administrador)	9
2.5. Diagrama de componentes (Alto nivel)	11
2.6. Diagrama de despliegue	12
2.7. Prototipo (<i>wireframe</i>) de la aplicación web	13
2.8. Mapa de navegabilidad (Anónimo)	14
2.9. Mapa de navegabilidad (Administrador)	14
2.10. Mapa de navegabilidad (Organización)	15
2.11. Mapa de navegabilidad (Universitario)	15
2.12. Diagrama de entidad-relación	16
2.13. Ejemplo de respuesta de servicio web	18
2.14. Comparativa de motores de búsqueda	20
2.15. Dependencia entre palabras cercanas	21
2.16. Clasificación Bayesiana	22
2.17. Regla de decisión del clasificador bayesiano	22
2.18. Aplicación del suavizado de Laplace	23
2.19. Ejemplo de escritura “Leet speak”	25

2.20. Ejemplo de palabras separadas por capitalización	25
3.1. Calendario GANTT	29
3.2. Tiempo dedicado por tipo de tarea	30
A.1. Diagrama de secuencia: Ver mejores organizaciones	38
A.2. Diagrama de secuencia: Listar organizaciones	40
A.3. Diagrama de secuencia: Buscar organizaciones	42
A.4. Diagrama de secuencia: Registrar usuario	44
A.5. Diagrama de secuencia: Identificar usuario	46
A.6. Diagrama de secuencia: Ver opiniones de organización	48
A.7. Diagrama de secuencia: Ver ofertas de organización	50
A.8. Diagrama de secuencia: Listar ofertas	52
A.9. Diagrama de secuencia: Publicar opinión	54
A.10. Diagrama de secuencia: Modificar información de usuario	56
A.11. Diagrama de secuencia: Ver novedades	57
A.12. Diagrama de secuencia: Ver opiniones sobre mi organización	59
A.13. Diagrama de secuencia: Publicar oferta	61
A.14. Diagrama de secuencia: Editar oferta	63
A.15. Diagrama de secuencia: Ver ofertas de mi organización	65
A.16. Diagrama de secuencia: Activar organización	67
A.17. Diagrama de secuencia: Ver estadísticas	68
A.18. Diagrama de secuencia: Realizar sorteo	70
B.1. Patrones HTML para el sitio “Student Job”	71
B.2. Diagrama de paquetes	73
B.3. Caja de envío de feedback	77

B.4. Prueba de carga del servidor	78
B.5. URL semántica (arriba) vs. URL común (abajo)	79
C.1. Tiempo dedicado por tipo de tarea	83
C.2. Burndown chart (Iteración 1)	84
C.3. Burndown chart (Iteración 2)	85
C.4. Burndown chart (Iteración 3)	85
C.5. Burndown chart (Iteración 4)	85
C.6. Burndown chart (Iteración 5)	86
C.7. Burndown chart (Iteración 6)	86
D.1. Manual de uso: Crear usuario	93
D.2. Manual de uso: Acceder como usuario registrado	94
D.3. Manual de uso: Recuperar contraseña olvidada	95
D.4. Manual de uso: Modificar información de usuario (1)	96
D.5. Manual de uso: Modificar información de usuario (2)	96
D.6. Manual de uso: Desconectar de cuenta de usuario	97
D.7. Manual de uso: Ver novedades	97
D.8. Manual de uso: Consultar mejores/peores organizaciones	98
D.9. Manual de uso: Buscar/listar organizaciones (1)	99
D.10. Manual de uso: Buscar/listar organizaciones (2)	99
D.11. Manual de uso: Ver detalles de valoración de una organización (1)	100
D.12. Manual de uso: Ver detalles de valoración de una organización (2)	101
D.13. Manual de uso: Publicar una opinión	102
D.14. Manual de uso: Buscar/listar ofertas (1)	103
D.15. Manual de uso: Buscar/listar ofertas (2)	103

D.16.Manual de uso: Enviar datos a una oferta	104
D.17.Manual de uso: Ver opiniones sobre mi organización	106
D.18.Manual de uso: Responder opiniones sobre mi organización	107
D.19.Manual de uso: Ver mis ofertas publicadas (1)	108
D.20.Manual de uso: Ver mis ofertas publicadas (2)	108
D.21.Manual de uso: Ver usuarios interesados en ofertas	109
D.22.Manual de uso: Publicar oferta	110
D.23.Manual de uso: Editar/Re-publicar oferta (1)	111
D.24.Manual de uso: Editar/Re-publicar oferta (2)	111
D.25.Manual de uso: Ver estadísticas	113
D.26.Manual de uso: Eliminar organización	113
D.27.Manual de uso: Eliminar opinión (1)	114
D.28.Manual de uso: Eliminar opinión (2)	115
D.29.Manual de uso: Eliminar oferta	115
D.30.Manual de uso: Banear usuario (1)	116
D.31.Manual de uso: Banear usuario (2)	116
D.32.Manual de uso: Activar organizaciones (1)	117
D.33.Manual de uso: Activar organizaciones (2)	117
D.34.Manual de uso: Realizar sorteo	118
E.1. Prototipo vs. Resultado final (Inicio)	123
E.2. Prototipo vs. Resultado final (Login)	124
E.3. Prototipo vs. Resultado final (Registro)	125
E.4. Prototipo vs. Resultado final (Notificaciones)	126
E.5. Prototipo vs. Resultado final (Ajustes de cuenta)	127

E.6. Prototipo vs. Resultado final (Lista de organizaciones)	128
E.7. Prototipo vs. Resultado final (Opiniones de organización)	129
E.8. Prototipo vs. Resultado final (Detalles de organización)	130
E.9. Prototipo vs. Resultado final (Lista de ofertas)	131
E.10. Prototipo vs. Resultado final (Publicar opinión)	132
E.11. Prototipo vs. Resultado final (Publicar oferta)	133

Índice de cuadros

2.1. Requisitos funcionales	10
2.2. Requisitos no funcionales	10
2.3. Operaciones HTTP	17
2.4. Recursos accesibles mediante la API RESTful	18
2.5. Precisión del sistema de clasificación de opiniones	23
2.6. Rendimiento del sistema de scraping	25
A.1. Caso de uso expandido: Ver mejores organizaciones	37
A.2. Caso de uso expandido: Listar organizaciones	39
A.3. Caso de uso expandido: Buscar organizaciones	41
A.4. Caso de uso expandido: Registrar usuario	43
A.5. Caso de uso expandido: Identificar usuario	45
A.6. Caso de uso expandido: Ver opiniones de organización	47
A.7. Caso de uso expandido: Ver ofertas de organización	49
A.8. Caso de uso expandido: Listar ofertas	51
A.9. Caso de uso expandido: Publicar opinión	53
A.10.Caso de uso expandido: Modificar información de usuario	55
A.11.Caso de uso expandido: Ver novedades	57
A.12.Caso de uso expandido: Ver opiniones sobre mi organización	58
A.13.Caso de uso expandido: Publicar oferta	60
A.14.Caso de uso expandido: Editar oferta	62
A.15.Caso de uso expandido: Ver ofertas de mi organización	64

A.16.Caso de uso expandido: Activar organización	66
A.17.Caso de uso expandido: Ver estadísticas	68
A.18.Caso de uso expandido: Realizar sorteo	69
B.1. Reconocimiento de información en scraping	73
B.2. Pruebas unitarias realizadas	75
B.3. Pruebas de integración realizadas	76
B.4. Equivalencias de URLs semánticas	80
C.1. Tiempo dedicado por tipo de tarea y periodo	84
C.2. Listado de user-stories	91

Capítulo 1

Introducción

Este capítulo introductorio pretende servir como prólogo para explicar en qué consiste el proyecto realizado. La sección 1.1 explica el problema que trata de cubrirse con el sistema a desarrollar. La sección 1.2 define los puntos que debe satisfacer el sistema para cumplir con los objetivos del proyecto. La sección 1.3 y 1.4 explican las tecnologías y metodologías utilizadas en el proyecto respectivamente. Finalmente la sección 1.5 resume la estructura de la memoria, explicando los diferentes capítulos.

1.1. Contexto

Actualmente los estudiantes universitarios no disponen de ningún servicio que ofrezca un catálogo sobre las condiciones laborales que se dan en las diferentes organizaciones. Por ello, en muchas ocasiones los universitarios acaban encontrando situaciones o condiciones inesperadas que no cumplen con sus expectativas y que pueden provocar enfado o frustración.

Universando pretende ser una guía de referencia para conocer cuáles son las mejores organizaciones, basándose en la opinión de sus empleados y en la oferta de empleo disponible. Por tanto, *Universando* busca ser de utilidad tanto para universitarios, que podrán conocer las condiciones laborales existentes en cada organización, como para las propias organizaciones, que podrán conocer la opinión de sus empleados para gestionar aquellos aspectos mejorables.

Este contexto es el idóneo para tratar el problema conocido como “análisis de sentimientos”, ya que la principal fuente de información de *Universando* son las opiniones de sus usuarios. Dicho problema consiste en evaluar las opiniones laborales para concluir si se tratan de comentarios positivos, negativos o neutrales. Éste es un problema realmente complejo que actualmente no tiene una solución clara. Parte de su elevada complejidad reside en que no existe una frontera bien definida entre lo que se podría clasificar como una opinión positiva, una opinión negativa o una opinión neutral, y esta supuesta frontera varía enormemente en función del usuario evaluador.

1.2. Objetivo del proyecto

El objetivo de este proyecto es el desarrollo completo de un sistema capaz de ser una guía de referencia para todos los estudiantes universitarios que quieran expresar su opinión o conocer la opinión de otros usuarios sobre las condiciones laborales existentes en organizaciones de su localidad; proporcionando además un medio para que las organizaciones conozcan las opiniones anónimas de sus empleados.

A continuación se detallan las funcionalidades básicas que debe proporcionar el sistema para considerar que los objetivos del proyecto se han cumplido:

1. El sistema debe permitir publicar, leer y contestar opiniones anónimas sobre experiencias laborales en organizaciones.
2. El sistema debe permitir publicar, leer, contestar y gestionar ofertas laborales, así como obtener ofertas de sitios externos.
3. El sistema debe permitir conocer las organizaciones más valoradas, así como buscar opiniones u ofertas de organizaciones concretas.
4. El sistema debe ofrecer mecanismos de administración para controlar los usuarios, las organizaciones, las opiniones y las ofertas existentes.

Además, el sistema debe ser capaz de afrontar el problema conocido como “análisis de sentimientos” que permite clasificar las opiniones de los usuarios como comentarios positivos, negativos o neutrales con una precisión aceptable.

El proyecto no consiste únicamente en la elaboración de un software que cumpla con una serie de requisitos. En este proyecto se trata de abordar el problema propuesto explorando cuales son las mejores soluciones para cada aspecto, realizando un diseño de la solución que no solo aporte una buena funcionalidad, sino que también permita evolucionar fácilmente el software implementado para ser utilizado como un producto capaz de rentabilizar las horas invertidas.

1.3. Marco tecnológico

En esta sección, para facilitar la comprensión del proyecto, se expone un listado con las tecnologías y herramientas que han sido necesarias en el desarrollo del sistema.

1.3.1. Tecnologías necesarias

A continuación se enumeran las principales tecnologías que se han utilizado.

1. **Java:** *Java* es un lenguaje de programación de propósito general, orientado a objetos y que cuenta con una comunidad muy activa. En este proyecto se ha utilizado *Java* como lenguaje base para implementar toda la lógica del sistema.
 - a) **Spring framework:** *Spring* [1] es un *framework* para la plataforma *Java*, es decir es una colección de módulos de *software* que sirven como base para el desarrollo de nuevos sistemas. Para este proyecto se ha elegido *Spring* ya que cuenta con módulos que facilitan la programación de servicios web, la autenticación de usuarios, la planificación de tareas, etc.

- b) **JSoup:** *JSoup* [2] es una librería para la plataforma *Java* que facilita la realización de tareas de *scraping* como la descarga y el procesamiento del contenido de páginas web. En este proyecto se ha utilizado esta librería para obtener ofertas laborales de sitios web externos.
2. **MySQL:** *MySQL* es un sistema de bases de datos relacionales. En este proyecto se ha utilizado para almacenar toda la información referente a organizaciones, usuarios, ofertas, opiniones...
 3. **MongoDB:** A diferencia de *MySQL*, *MongoDB* es un sistema de bases de datos no relacional [3], es decir los datos se almacenan sin ningún tipo de esquema. Esto favorece enormemente a la velocidad de lectura y escritura, por lo que en este proyecto se ha utilizado *MongoDB* para almacenar la información del sistema de aprendizaje automático implementado, ya que requiere de un gran número de consultas a la base de datos.
 4. **Apache Solr:** *Apache Solr* [4] es un motor de búsqueda de código abierto capaz de indexar grandes cantidades de información para posteriormente realizar búsquedas en tiempos muy reducidos. En este proyecto se ha utilizado *Apache Solr* para la búsqueda de organizaciones por nombre.
 5. **HTML5, CSS3 y JQuery:** HTML y CSS son las tecnologías más utilizadas tanto para establecer la estructura y contenido de una página como para definir su estilo visual. Por otra parte, *JQuery* [5] es una librería *Javascript* que permite dotar de interactividad a la página. En este proyecto se han utilizado estas tecnologías conjuntamente para elaborar una aplicación web *responsive* capaz de adaptarse a cualquier tipo de dispositivo, y que interactúa con los servicios web programados.

1.3.2. Herramientas de desarrollo necesarias

A continuación se explican las principales herramientas utilizadas en el desarrollo.

1. **GitHub:** *GitHub* es un sistema de control de versiones que permite gestionar los cambios que se realizan en el sistema en desarrollo, así como las tareas a realizar.
2. **Gradle:** *Gradle* [6] es una herramienta que permite automatizar los procesos de compilación, ejecución de pruebas y despliegue del sistema en un servidor embebido en el propio *script* de *Gradle*. Además incluye soporte de dependencias, pudiendo descargar las librerías necesarias.
3. **Travis:** *Travis* es un entorno de integración continua [7] que permite la ejecución automática de pruebas durante el proceso de desarrollo del sistema. Estos entornos son de gran utilidad para la detección temprana de errores, ya que si algún cambio realizado durante el desarrollo provoca un resultado inesperado en las pruebas, los errores se reportarán de forma inmediata.
4. **Eclipse:** *Eclipse* es un entorno de desarrollo que facilita las tareas de programación en diferentes lenguajes, así como la interacción con la plataforma *GitHub*.

1.4. Metodología

Este proyecto se ha gestionado mediante una metodología de desarrollo ágil [8] en la que el sistema se desarrolla de forma iterativa e incremental. Las tareas se dividen en iteraciones de duración determinada a través de las cuales se obtienen versiones crecientes del sistema como se muestra en la figura 1.1.



Figura 1.1: Metodología de desarrollo ágil

La obtención del *backlog* de tareas y su posterior división en las diferentes iteraciones requiere de un proceso previo de análisis del problema que puede resumirse en:

1. Definir los requisitos funcionales y no funcionales que se deben cumplir.
2. Partiendo de cada uno de los requisitos, elaborar una lista de *user stories* [9] que definan ejemplos de situaciones concretas de uso del sistema.
3. Cada *user story* debe ser dividido en las tareas necesarias para su implementación.
4. Las tareas listadas deben ser ordenadas según prioridad, teniendo en cuenta que algunas tareas pueden necesitar ser implementadas antes que otras debido a posibles dependencias.
5. Dividir las tareas de forma equilibrada en las diferentes iteraciones a realizar.

La metodología utilizada cuenta con elementos propios del método *Lean Startup* [10] como la elaboración de un *canvas* (figura 1.2) para plasmar los siguientes puntos de la idea del proyecto:

1. **Segmentos de clientes:** Segmentos de clientes sobre los que trabajar.
2. **Problema:** Problemas detectados que se tratarán de explotar como oportunidades de negocio.
3. **Proposición de valor única:** Declaración sobre cómo el producto resolverá los problemas.
4. **Solución:** Características del producto que ayudarán a resolver los problemas planteados.
5. **Canales:** Medios para hacer llegar la solución (el producto) a los potenciales clientes.
6. **Flujos de ingresos:** Medios de ingreso de capital.
7. **Estructura de costes:** Gasto mensual aproximado.
8. **Métricas clave:** Actividades que se deben medir para la futura toma de decisiones.
9. **Ventaja diferencial:** Aspectos en los que el producto planteado destacará sobre los demás.

PROBLEMA	SOLUCIÓN	PROPOSICIÓN DE VALOR ÚNICA	VENTAJA DIFERENCIAL	SEGMENTOS DE CLIENTES
1 – Los universitarios desconocen las condiciones y oportunidades de trabajo reales de las organizaciones. 2 – Algunos universitarios están descontentos con las condiciones encontradas en organizaciones al hacer su TFG, al hacer prácticas o en sus primeros empleos. 3 – Los universitarios no cuentan con un gran “catálogo” de los posibles TFG o prácticas que pueden realizar en organizaciones.	1 – Dar a conocer opiniones de otros universitarios sobre sus primeros empleos, prácticas o proyectos en diferentes organizaciones. 2 – Mantener un listado con las organizaciones disponibles junto con una puntuación calculada a partir de las opiniones de los usuarios. 3 – Permitir que las organizaciones puedan publicar y gestionar ofertas de TFG, prácticas o empleo. 4 – Republicar ofertas existentes dirigidas a alumnos.	La mejor manera de evitar que un amigo tenga una mala primera experiencia profesional. El comparador para elegir tu primera experiencia profesional <i>Concepto de alto nivel:</i> Medio con mayor número de opiniones constructivas y valoraciones fiables sobre las condiciones de trabajo a la hora de realizar un TFG, prácticas o al trabajar en un gran número de organizaciones, permitiendo el feedback de las organizaciones implicadas. Medio rápido y seguro para que universitarios puedan localizar posibilidades de realizar TFG, prácticas o encontrar un primer empleo.	Comunidad de usuarios. Repertorio de organizaciones/opiniones. Repertorio de ofertas.	Universitarios (visitante, usuario): <ul style="list-style-type: none">- Últimos años del grado- Durante el master- Primer o segundo empleo Organizaciones (visitante, usuario): <ul style="list-style-type: none">- Empresas- ONG- Centros de investigación Primeros usuarios (early-adopter): Universitarios de último año
	MÉTRICAS CLAVE A – Usuarios registrados. A – Organizaciones registradas (Y tipo de organización). A – Opiniones publicadas. R – Usuarios con opiniones en más de una organización. R – Organizaciones que emiten contestaciones y publican ofertas regularmente.		CANALES Estudiantes: <ol style="list-style-type: none">1. Redes sociales.2. Listas de distribución universitarias.3. Buscadores (SEO)4. Medios de comunicación.5. Boca a boca. Organizaciones: <ol style="list-style-type: none">1. Mails directos.2. Redes sociales.3. Buscadores (SEO)4. Medios de comunicación.5. Boca a boca.	
ESTRUCTURA DE COSTES Hosting: 2,5€ / mes Dominio: 4€ / año Incentivos por publicar opiniones: 10€/mes (Primeros meses)	FLUJOS DE INGRESOS Publicidad de Google Suscripción mensual para empresas. 1,99€/mes (En un futuro)			

Figura 1.2: Lean Canvas

1.5. Estructura de la memoria

La presente memoria se divide en diferentes capítulos que explican los detalles del proyecto realizado:

1. **Introducción:** Capítulo que introduce el problema a resolver, los objetivos del proyecto y las tecnologías y metodologías a utilizar.
2. **Trabajo realizado:** Capítulo que explica el análisis, diseño e implementación del sistema.
3. **Gestión del proyecto:** Capítulo que engloba la planificación y gestión de los ciclos del proyecto.
4. **Conclusiones:** Capítulo final que expone un resumen del proyecto, explicando el resultado obtenido y sus posibilidades de futuro.

De forma adicional, se han incluido diversos anexos que complementan la información explicada en los capítulos anteriores:

1. **Anexo 1. Casos de uso detallados:** Ejemplos concretos de los usos más habituales que pueden darse en el sistema.
2. **Anexo 2. Detalles de implementación:** Detalles de implementación como: paquetes implementados, pruebas realizadas, técnicas de SEO aplicadas, librerías externas utilizadas...
3. **Anexo 3. Detalles de gestión:** Información sobre la gestión del proyecto: tiempos empleados, gráficas, tareas...
4. **Anexo 4. Manuales:** Manuales sobre el despliegue y uso del sistema para los diferentes usuarios.
5. **Anexo 5. Prototipo vs. Resultado final:** Comparativa entre el prototipo generado en la fase de diseño y el resultado final.
6. **Anexo 6. Especificación del API RESTful:** Documentación sobre las operaciones disponibles en el API RESTful implementado.

Capítulo 2

Trabajo realizado

Este capítulo engloba los detalles de las diferentes etapas del desarrollo del proyecto. En las secciones 2.1 y 2.2 se exponen los elementos más importantes de las fases de análisis y diseño respectivamente, mientras que la sección 2.3 explica la implementación del proyecto.

2.1. Análisis

La fase de análisis ocupa diferentes tareas de conceptualización, como la elaboración de diagramas de casos de uso (subsección 2.1.1) y de listados de requisitos funcionales y no funcionales (subsecciones 2.1.2 y 2.1.3). Estas tareas permiten definir las funcionalidades que debe proporcionar el sistema.

2.1.1. Casos de uso

Para facilitar la comprensión, se han elaborado diagramas de casos de uso separados para cada rol de usuario. Estos diagramas se muestran en las figuras 2.1, 2.2, 2.3 y 2.4, y exponen las funcionalidades disponibles para el rol de usuario no identificado, usuario identificado como universitario, usuario identificado como organización o usuario identificado como administrador respectivamente.

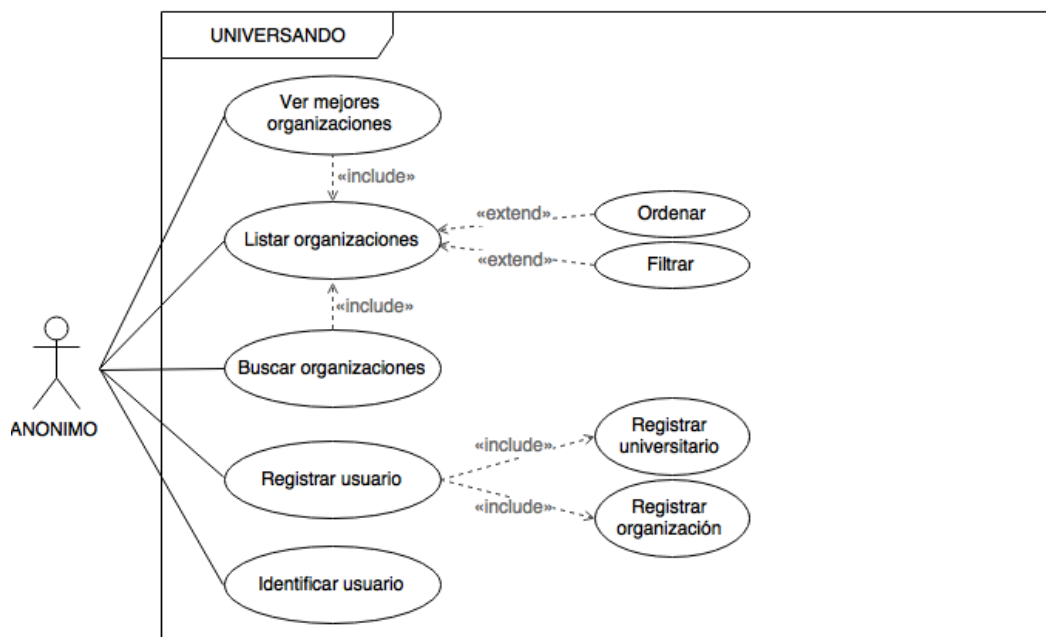


Figura 2.1: Casos de uso (Anónimo)

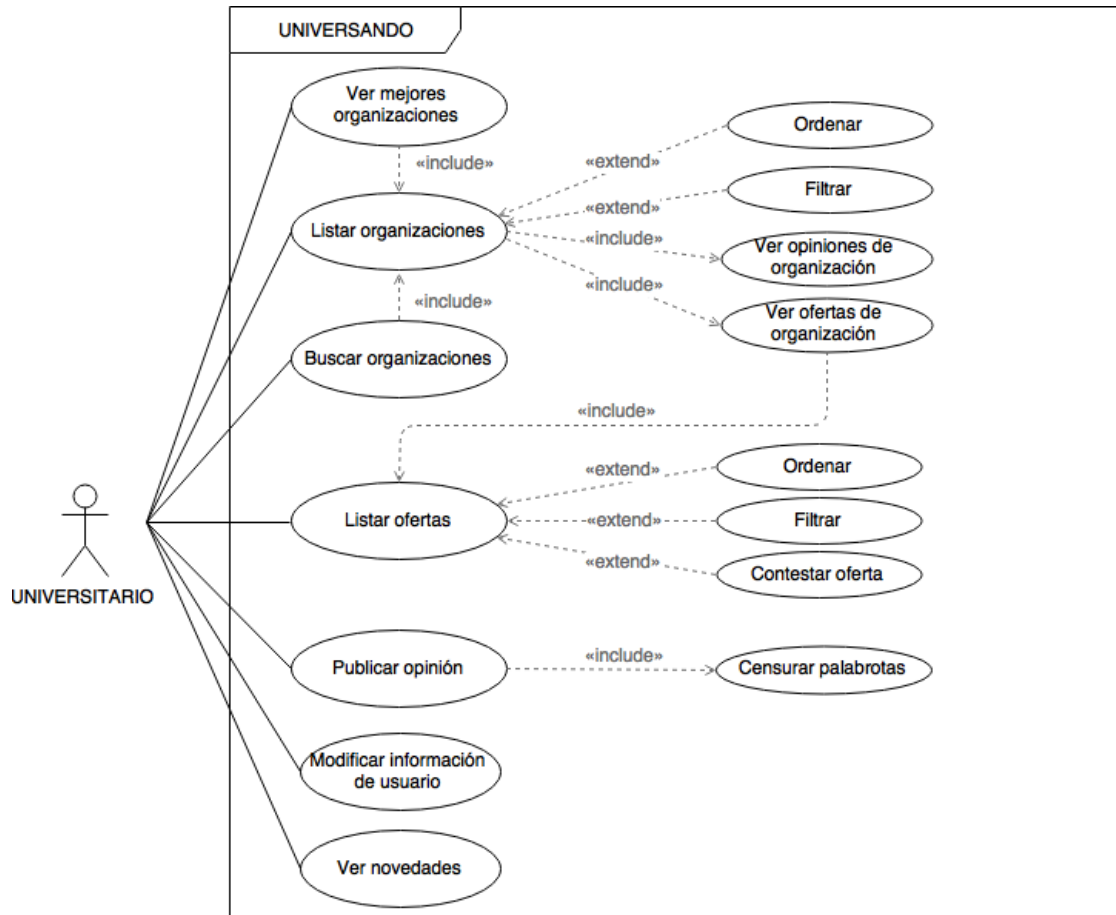


Figura 2.2: Casos de uso (Universitario)

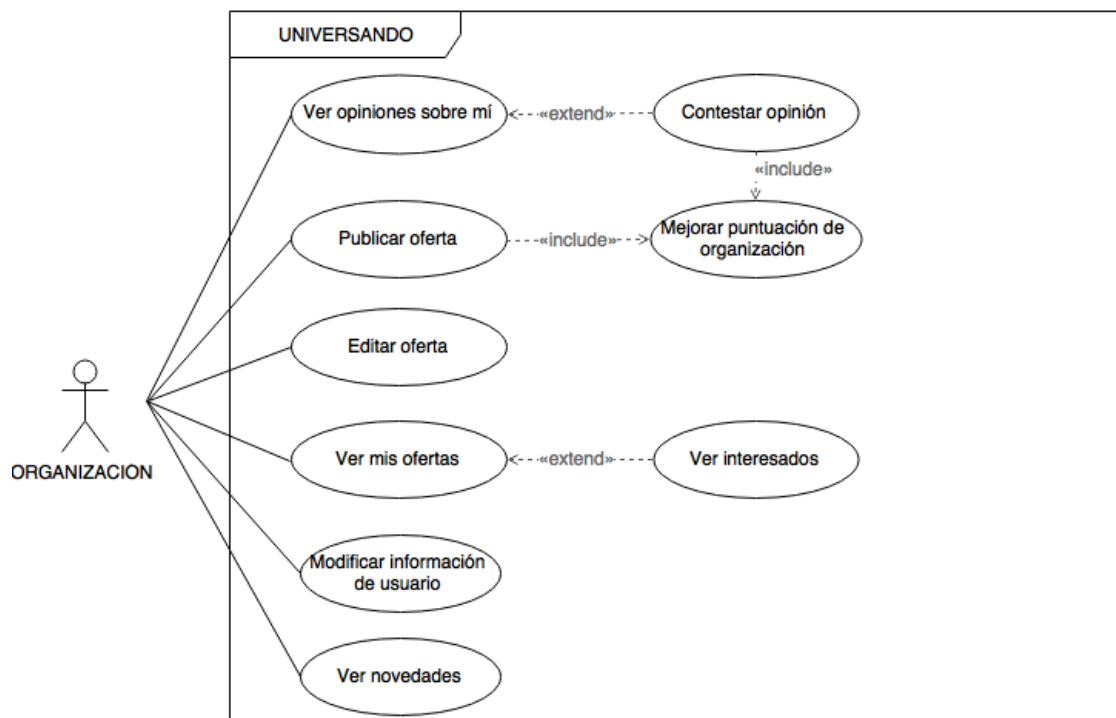


Figura 2.3: Casos de uso (Organización)

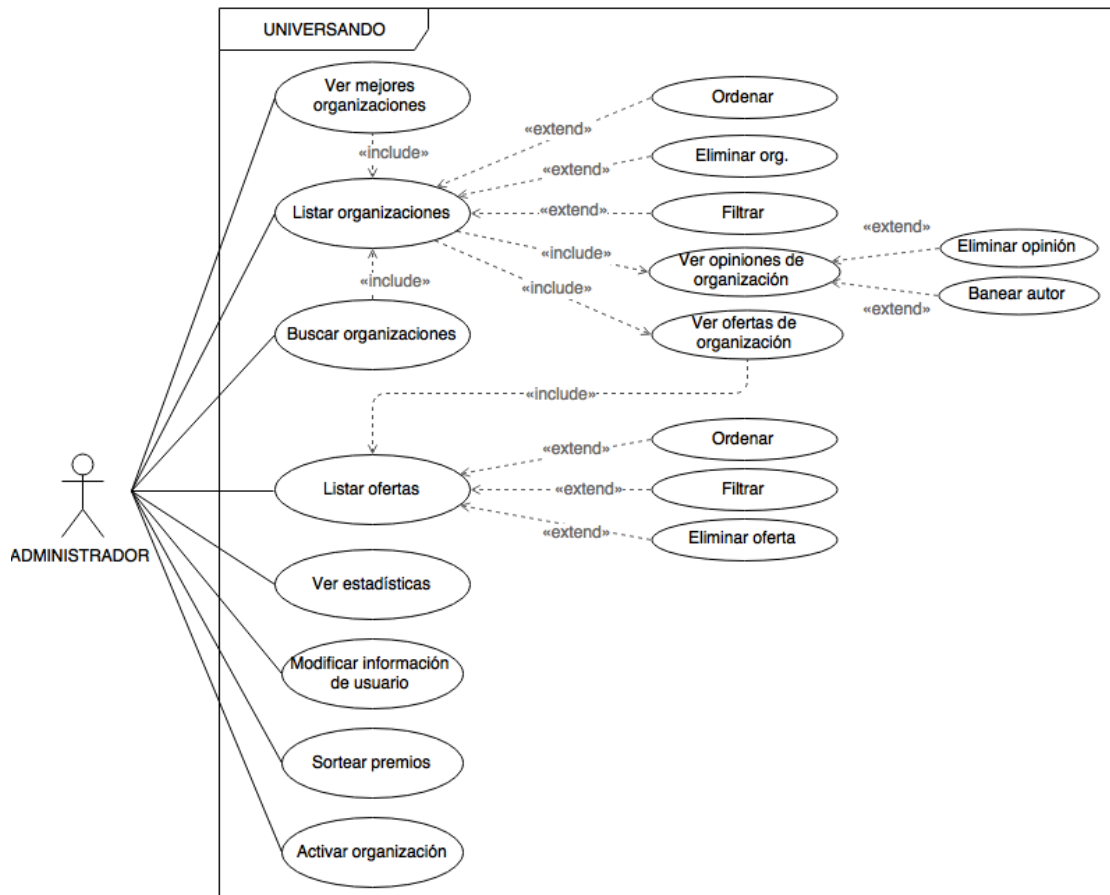


Figura 2.4: Casos de uso (Administrador)

2.1.2. Requisitos funcionales

A continuación se expone la lista de necesidades de comportamientos concretos del sistema.

RF01	El sistema permite listar las organizaciones existentes junto con sus puntuaciones medias.
RF02	El sistema permite ordenar el listado de organizaciones según sus puntuaciones.
RF03	El sistema permite filtrar el listado de organizaciones según provincia y titulación.
RF04	El sistema permite buscar organizaciones por nombre.
RF05	El sistema permite ver las mejores organizaciones de la titulación del visitante.
RF06	El sistema permite registrarse/identificarse.
RF07	El sistema permite a usuarios modificar la información de su cuenta.
RF08	El sistema permite a usuarios modificar su contraseña en caso de olvido.
RF09	El sistema permite a universitarios ver las opiniones sobre cada organización.
RF10	El sistema permite a universitarios publicar opiniones anónimas sobre organizaciones.
RF11	El sistema permite a universitarios ver las ofertas de una organización concreta.
RF12	El sistema permite a universitarios ver las ofertas disponibles para su titulación.
RF13	El sistema permite a universitarios contestar (enviar su CV) a las ofertas publicadas.

RF14	El sistema permite a universitarios ir a la página original de las ofertas recopiladas de otras web.
RF15	El sistema permite ordenar las ofertas por fecha y puntuación.
RF16	El sistema permite filtrar las ofertas según su tipo o provincia.
RF17	El sistema permite a organizaciones ver las opiniones publicadas sobre ellas.
RF18	El sistema permite a organizaciones contestar a las opiniones publicadas sobre ellas.
RF19	El sistema permite a organizaciones publicar ofertas especificando titulación, tipo y caducidad.
RF20	El sistema permite a organizaciones editar sus ofertas publicadas.
RF21	El sistema permite a organizaciones ver sus ofertas publicadas.
RF22	El sistema permite a organizaciones obtener los CV de los interesados en sus ofertas.
RF23	El sistema permite a administradores conocer las estadísticas de uso del sitio.
RF24	El sistema permite a administradores eliminar opiniones concretas.
RF25	El sistema permite a administradores eliminar organizaciones concretas.
RF26	El sistema permite a administradores banear usuarios concretos.
RF27	El sistema permite a administradores eliminar ofertas concretas.
RF28	El sistema permite a administradores activar cuentas de organizaciones.
RF29	El sistema permite a administradores realizar sorteos de premios.
RF30	El sistema envía periódicamente <i>emails</i> con la situación del <i>ranking</i> de organizaciones.
RF31	El sistema notifica a universitarios y organizaciones novedades sobre opiniones y ofertas.
RF32	El sistema censura las opiniones inadecuadas de los universitarios.
RF33	El sistema re-publica ofertas existentes en otros sitios web.
RF34	El sistema clasifica automáticamente las opiniones como positivas, neutrales, negativas.
RF35	El sistema permite recomendar el servicio a través de las redes sociales.

Cuadro 2.1: Requisitos funcionales

2.1.3. Requisitos no funcionales

A continuación se listan las necesidades del sistema que no expresan un comportamiento concreto.

RNF01	El sitio web del sistema sigue principios SEO para mejorar el posicionamiento en buscadores.
RNF02	El sitio web del sistema posee un diseño <i>responsive</i> que se adapta a todo tipo de dispositivo.
RNF03	El sistema muestra sugerencias de organizaciones mientras se escribe en el buscador.
RNF04	El sistema expone su lógica a través de una API de servicios web RESTful.
RNF05	El sistema muestra la puntuación de las nuevas opiniones mientras se escriben.

Cuadro 2.2: Requisitos no funcionales

2.1.4. Conclusiones

Esta fase de análisis es de gran importancia ya que define las funcionalidades que proporcionará el sistema y que se tomarán como referencia para las posteriores fases de diseño e implementación.

Las funcionalidades descritas permiten el control de los usuarios, organizaciones, opiniones y ofertas. Además se definen diferentes roles de usuario que podrán utilizar un subconjunto de las funcionalidades del sistema. Estos roles son: universitario, organización, administrador y usuario invitado.

2.2. Diseño

La fase de diseño pretende determinar aspectos como los componentes e infraestructura del sistema (subsección 2.2.1), el aspecto visual de la aplicación web (subsecciones 2.2.2 y 2.2.3), las estructuras de almacenamiento de datos (subsección 2.2.4) y los principios que debe seguir la interfaz de servicios web (subsección 2.2.5).

2.2.1. Arquitectura del sistema

Como puede observarse en la figura 2.5, la lógica del sistema hace uso de bases de datos relacionales y no relacionales, así como de un repositorio *Solr* para obtener el mejor rendimiento.

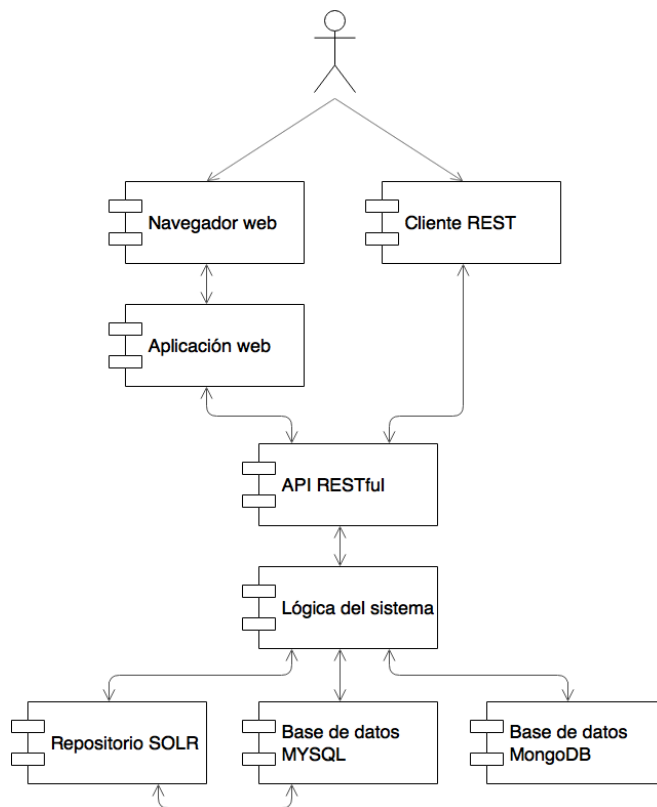


Figura 2.5: Diagrama de componentes (Alto nivel)

Por otro lado, los usuarios tienen dos modos de interactuar con el sistema:

1. Accediendo a la aplicación web a través de un navegador, la cual utilizará la interfaz de servicios web para hacer uso de la lógica del sistema.
2. Utilizando un cliente REST para interactuar directamente con la interfaz de servicios.

En la figura 2.6 se muestran los requisitos de infraestructura necesarios para desplegar el sistema. Este sistema requiere de un único servidor que será el encargado de almacenar y ejecutar los diferentes componentes, así como de atender las peticiones de todos los clientes.

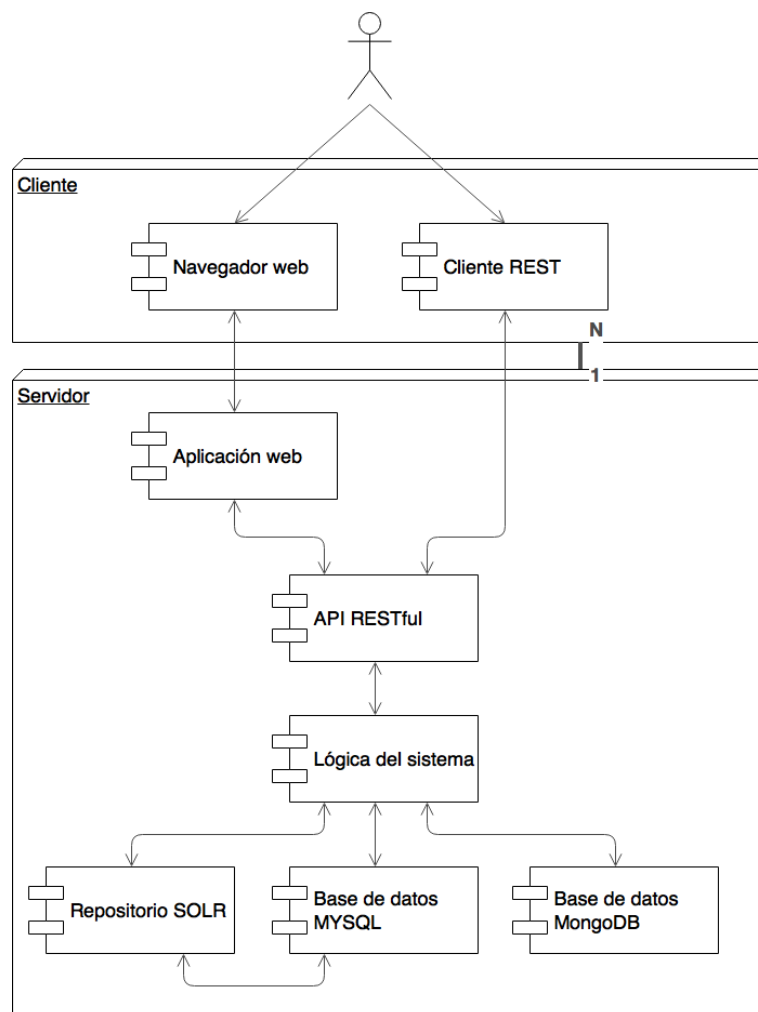


Figura 2.6: Diagrama de despliegue

2.2.2. Prototipo

El *wireframe* mostrado en la figura 2.7 muestra parte del prototipo inicial de la aplicación web del sistema. El prototipo completo junto con el aspecto final de la aplicación web se muestran detalladamente en el Anexo E, donde se incluyen las siguientes páginas:

- 1. **Inicio:** Pantalla inicial de la aplicación con el *ranking* de organizaciones.
- 2. **Login:** Formulario de identificación.
- 3. **Registro:** Formulario de registro de nuevos usuarios.
- 4. **Notificaciones:** Novedades del usuario.
- 5. **Ajustes de cuenta:** Formulario de modificación de la información del usuario.
- 6. **Lista de organizaciones:** Listado con las diferentes organizaciones existentes.
- 7. **Opiniones de organización:** Listado con opiniones de una organización concreta.
- 8. **Detalles de organización:** Detalles de valoración de una organización concreta.
- 9. **Lista de ofertas:** Listado con las ofertas existentes.
- 10. **Publicar opinión:** Formulario de nueva opinión.
- 11. **Publicar oferta:** Formulario de nueva oferta.

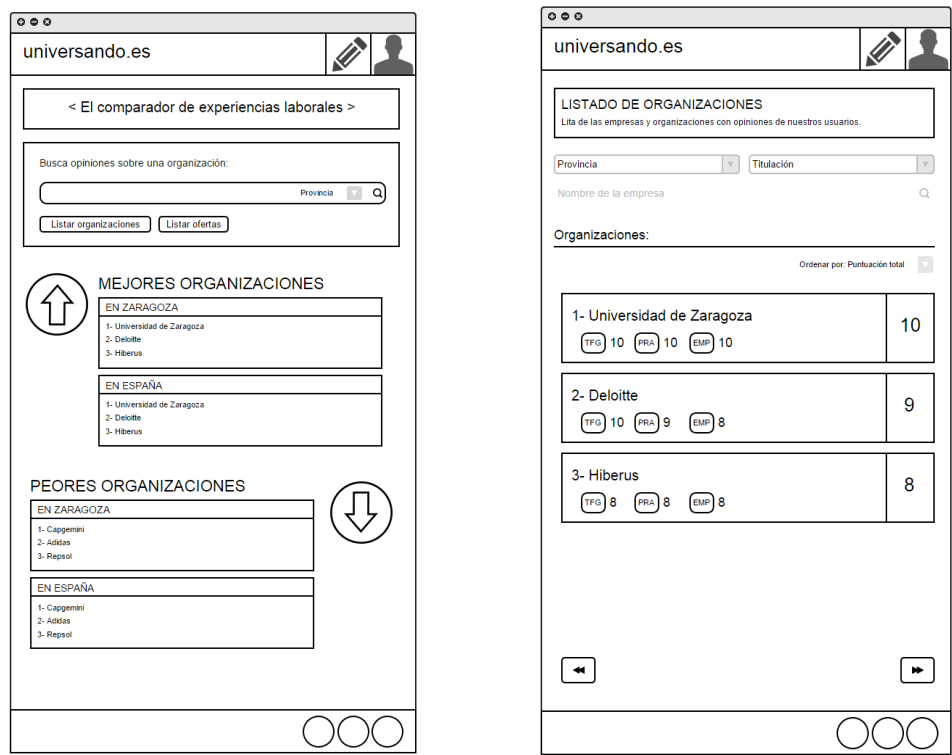


Figura 2.7: Prototipo (*wireframe*) de la aplicación web

2.2.3. Mapa de navegabilidad

El mapa de navegabilidad da una visión de las páginas a las que pueden acceder los usuarios. Para facilitar la comprensión de estos mapas, se han realizado diagramas separados para los diferentes roles de usuario. De modo que las figuras 2.8, 2.9, 2.10 y 2.11 representan el mapa de navegabilidad de los roles de usuario anónimo, administrador, organización y universitario respectivamente.

Los usuarios tienen acceso a un conjunto de páginas que varía dependiendo de su rol, ya que algunas funcionalidades solo están disponibles para un grupo de usuarios concreto. Además, se ha incorporado un menú desde el que, entre otras funciones, los usuarios pueden identificarse para alterar su rol.

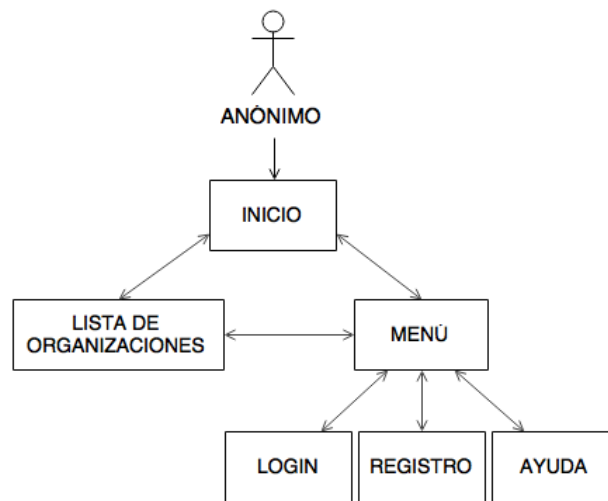


Figura 2.8: Mapa de navegabilidad (Anónimo)

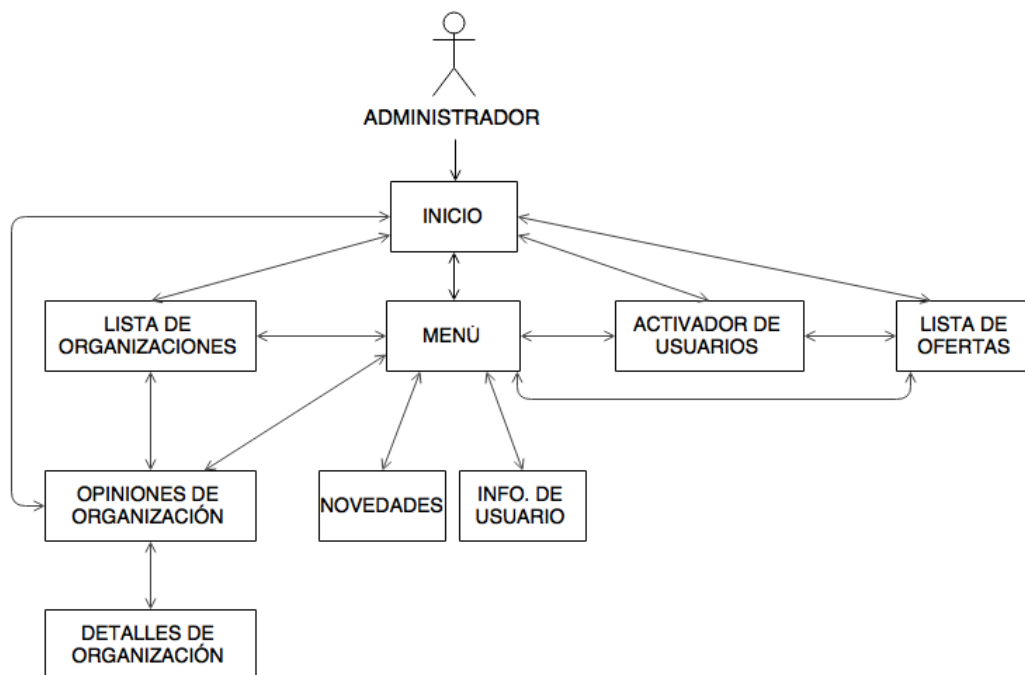


Figura 2.9: Mapa de navegabilidad (Administrador)

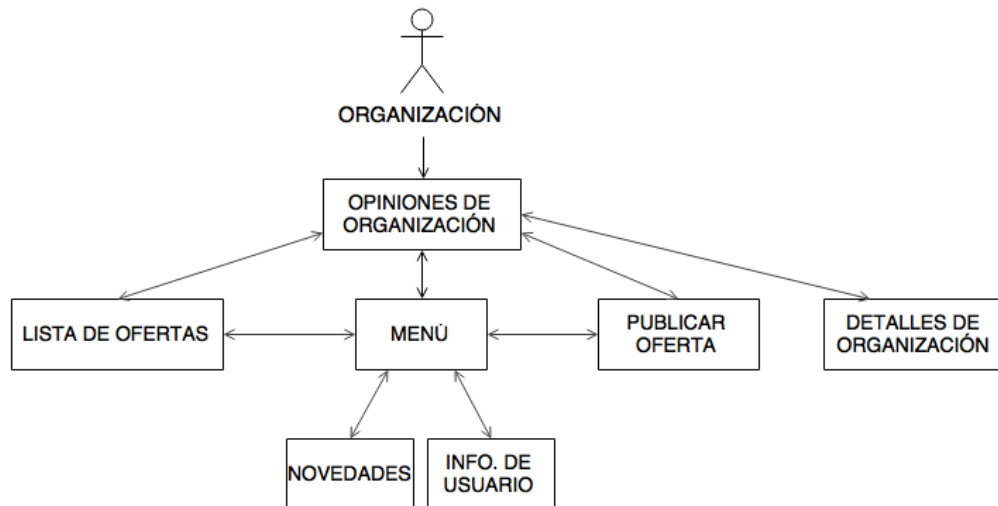


Figura 2.10: Mapa de navegabilidad (Organización)

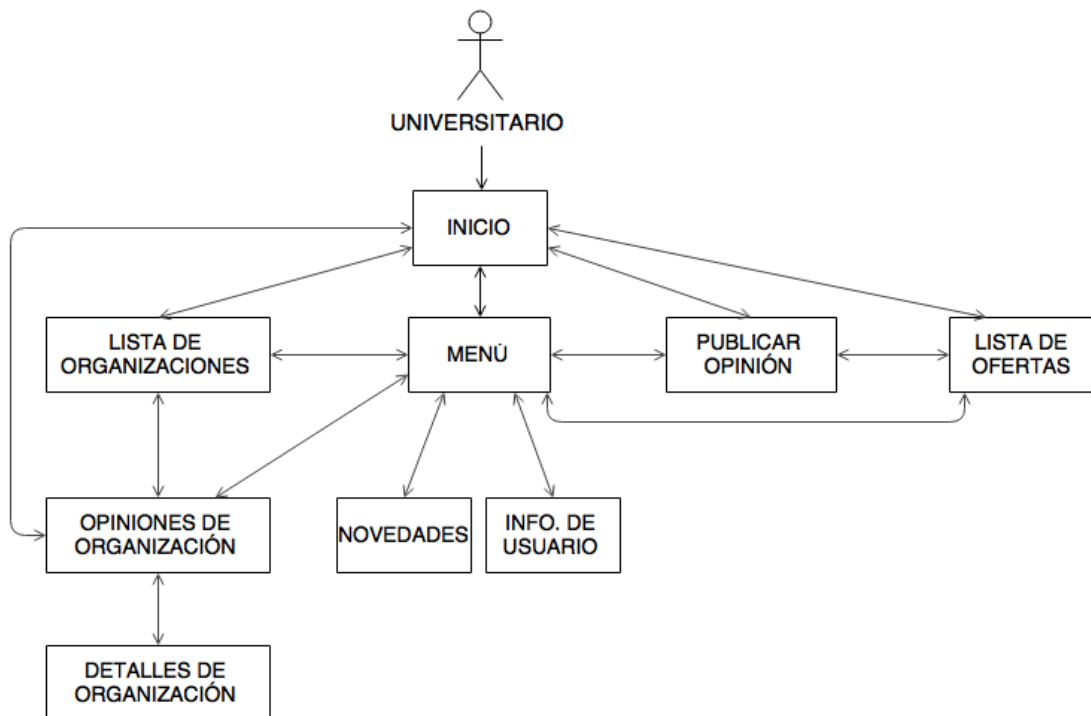


Figura 2.11: Mapa de navegabilidad (Universitario)

2.2.4. Modelo de datos

La figura 2.12 muestra el esquema de la base de datos *MySQL* del sistema, la cual almacena la gran mayoría de la información sobre usuarios, organizaciones, ofertas, opiniones, currículos, novedades, provincias, universidades, sorteos y titulaciones. Se ha elegido una base de datos relacional como medio de almacenamiento principal ya que la información posee un gran número de relaciones.

Por otro lado, el sistema de aprendizaje automático utiliza una base de datos no relacional que almacena colecciones de datos estadísticos sobre las opiniones y las palabras que aparecen en ellas. En este caso se ha elegido una base de datos no relacional ya que los datos almacenados no poseen fuertes relaciones entre sí y se requieren tiempos de lectura y escritura muy eficientes.

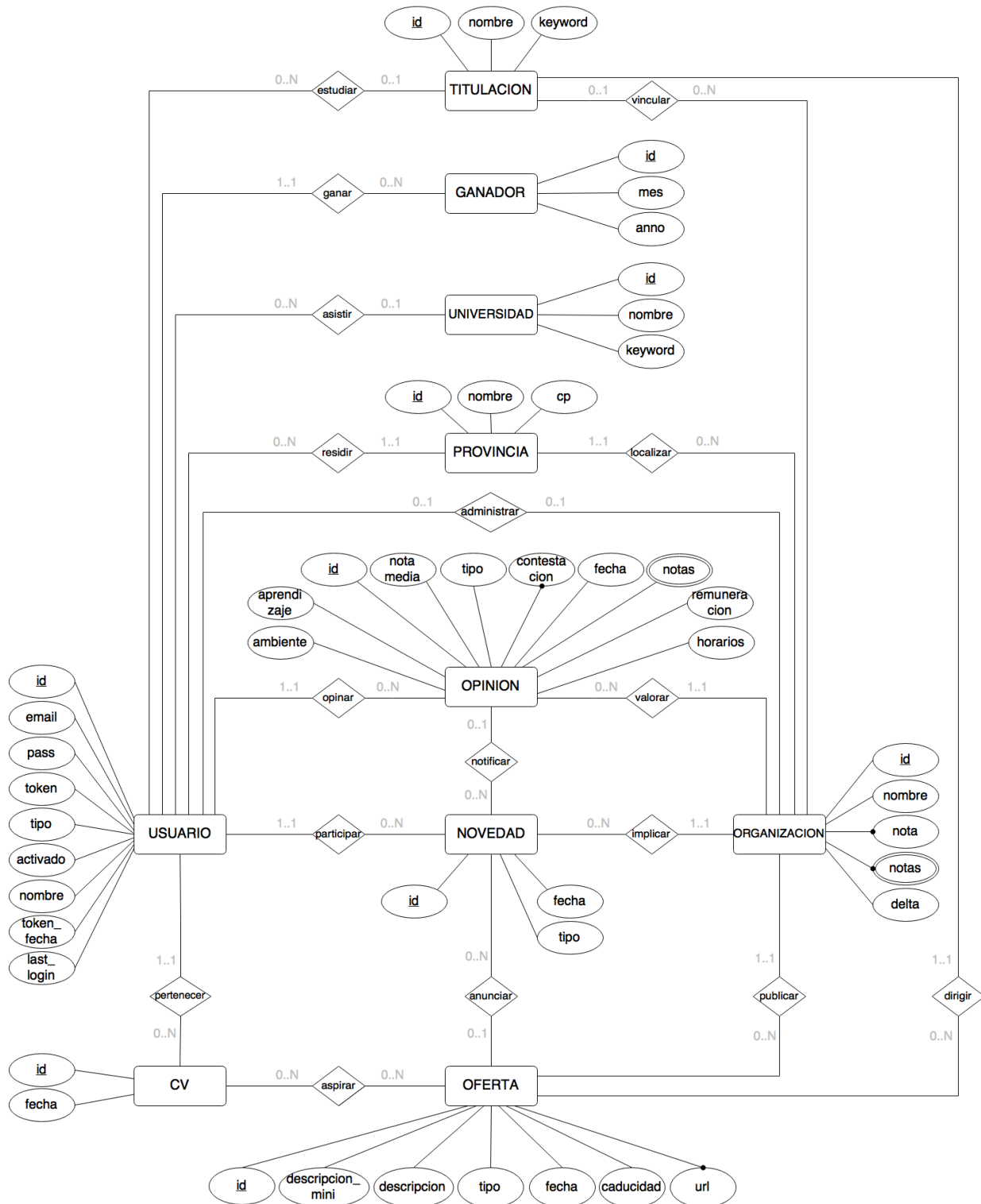


Figura 2.12: Diagrama de entidad-relación

2.2.5. Diseño de los servicios web

Con el fin de simplificar el diseño y favorecer la interoperabilidad se ha diseñado una API RESTful (*Application Programming Interface*) [12] que expone una serie de servicios web capaces de ejecutar funciones concretas de la lógica del sistema. Estos servicios web permiten desacoplar la parte frontal de la parte lógica del sistema, permitiendo que incluso aplicaciones externas creadas por terceros puedan hacer uso de la lógica implementada.

Para mejorar la experiencia de uso de la API se siguen los siguientes principios básicos [13]:

1. **Recursos direccionables:** Cada recurso es diferenciable del resto y posee al menos una URI que actúa como identificador.
2. **Interfaz uniforme:** Se utiliza el conjunto de operaciones proporcionado por HTTP para aplicar acciones concretas sobre recursos. En el cuadro 2.3 se listan las diferentes operaciones permitidas en la API y sus acciones equivalentes.
3. **HATEOAS (*Hypermedia as the Engine of Application State*):** Los mensajes de respuesta se emiten en formato JSON (*JavaScript Object Notation*) e incluyen enlaces a recursos relacionados. Por ejemplo, si se solicita un recurso “organización”, en la respuesta se incluye un enlace a su recurso “provincia” asociado. De este modo, cualquier cambio en las URIs de acceso a los recursos relacionados sería transparente para los clientes de la API.
4. **Interacción sin estado:** Cada petición HTTP se realiza de forma aislada, de modo que el encargado de gestionar el estado de la interacción es el cliente, el cual debe almacenar los *tokens* correspondientes.

GET	Obtiene el recurso asociado a la URI dada
POST	Crea o actualiza un recurso
PUT	Crea o actualiza el recurso asociado a la URI dada
DELETE	Elimina el recurso asociado a la URI dada

Cuadro 2.3: Operaciones HTTP

El sistema de autenticación de usuarios se basa en un modelo de *tokens* dinámicos que se envían y devuelven en la cabecera “X-CSRF-TOKEN” de cada petición. Este método es gestionado por el *framework Spring* y proporciona protección ante ataques CSRF (*Cross Site Request Forgery*) [14].

El cuadro 2.4 muestra los tipos de recurso con los que se puede interactuar utilizando la API. Los detalles de dicha interacción se recogen en el Anexo F.

CVs	Currículos de los universitarios
Estadísticas	Estadísticas de uso del sitio web
Ganadores	Usuarios ganadores de sorteos
Novedades	Notificaciones de nuevos eventos
Ofertas	Ofertas de empleo, prácticas, etc.
Opiniones	Opiniones de los universitarios sobre organizaciones
Organizaciones	Organizaciones con ofertas o sobre las que opinar
Provincias	Provincias españolas
Titulaciones	Titulaciones permitidas por el sistema
Universidades	Universidades permitidas por el sistema
Usuarios	Usuarios del sistema (universitarios, organizaciones o administradores)

Cuadro 2.4: Recursos accesibles mediante la API RESTful

En la figura 2.13 se muestra un ejemplo de respuesta JSON devuelta por uno de los servicios web disponibles, donde se pueden diferenciar varios elementos:

1. **code:** Código numérico de respuesta que sigue el estándar de códigos HTTP.
2. **status:** Estado de la operación realizada que puede variar entre “success” o “error”.
3. **message:** Mensaje descriptivo del error (Solo si se ha producido un error).
4. **data:** Incluye la información propia del recurso solicitado en formato JSON junto con enlaces a recursos relacionados (.links).

```

code: 200
status: "success"
message: ""
data: {
  nombre: "Universidad de Zaragoza"
  provincia: 50
  titulacion: 0
  idRecurso: 5
  _links: {
    -self: {
      href: "http://universando.es/organizaciones/5"
    }
    -provincia: {
      href: "http://universando.es/provincias/50"
    }
  }
}

```

Figura 2.13: Ejemplo de respuesta de servicio web

2.2.6. Conclusiones

El sistema cuenta con varias bases de datos y componentes que requieren de un único servidor para ser desplegados. Este servidor será el encargado de atender las peticiones de los clientes realizadas a través de la aplicación web o de cualquier otra aplicación que utilice la API RESTful implementada.

Por otro lado, los datos del sistema se distribuyen entre una base de datos relacional, que almacenará la mayor parte de la información del sistema, y una base de datos no relacional utilizada para el almacenamiento de los datos usados por el sistema de aprendizaje automático.

2.3. Implementación

Esta sección se centra en los aspectos más relevantes de la fase de implementación del proyecto. En la subsección 2.3.1 se explica el papel que ha tenido el *framework Spring* en el desarrollo del sistema. En las subsecciones 2.3.2, 2.3.3, 2.3.4 y 2.3.5 se explican detalles sobre la implementación de las funcionalidades que se han considerado más relevantes, como la búsqueda de organizaciones, la clasificación y tratamiento de opiniones y la obtención de ofertas de sitios externos. La explicación de las técnicas utilizadas en la creación del sitio web se detallan en la subsección 2.3.6. Y finalmente, en la subsección 2.3.7 se especifican los diferentes tipos de prueba que verifican la corrección del sistema, las cuales se han ejecutado en un entorno de integración continua.

2.3.1. Uso del framework Spring

Se ha utilizado el *framework Spring* [1] para facilitar las tareas de implementación, ya que proporciona una serie de módulos que incluyen la lógica de los aspectos más habituales en este tipo de proyecto:

1. **Servicios web:** El *framework Spring* permite definir controladores REST concisos y simples que atienden las peticiones HTTP de los clientes y emiten una respuesta en formato JSON. [15]
2. **Seguridad:** Determinadas funcionalidades del sistema solo pueden ser accedidas por algunos usuarios con un determinado rol. *Spring* proporciona el control de seguridad necesario para permitir identificar a los usuarios y determinar si cuentan con los privilegios suficientes para acceder a los diferentes servicios web. El sistema implementado utiliza una política de seguridad basada en *tokens* que incluye protección ante ataques CSRF. [16] [17]
3. **Control de excepciones:** *Spring* permite definir controladores que gestionan las excepciones del sistema, permitiendo enviar mensajes de error predefinidos como respuesta a las peticiones que han producido el error. [18]
4. **Interacción con bases de datos:** El sistema utiliza bases de datos relacionales (*MySQL*) y no relacionales (*MongoDB*). *Spring* incluye mecanismos de interacción con bases de datos que facilitan el proceso. [19] [20]

5. **Tareas programadas:** Existen determinadas tareas que deben ejecutarse de forma periódica para cumplir con la funcionalidad deseada y mantener el correcto funcionamiento del sistema. *Spring* permite llevar a cabo una planificación de tareas para que determinadas funciones se ejecuten con una periodicidad establecida. [21]
6. **Envío de correos electrónicos:** El sistema envía correos electrónicos para informar de determinados eventos a los usuarios. *Spring* facilita estas tareas de envío de emails, así como el uso de plantillas *Apache Velocity*. [22]
7. **Integración con *Apache Solr*:** *Spring* permite una sencilla integración del motor de búsqueda *Apache Solr*, permitiendo realizar búsquedas y gestionar el repositorio de forma fácil. [23]

2.3.2. Búsqueda de organizaciones

El sistema proporciona la funcionalidad de buscar organizaciones concretas escribiendo su nombre. Para implementar esta característica se ha utilizado el motor de búsqueda *Apache Solr* [4] que crea un índice sobre la colección de organizaciones para posteriormente ejecutar búsquedas en un tiempo óptimo.

Como se muestra en la figura 2.14 [24], *Apache Solr* proporciona unos tiempos de búsqueda realmente buenos. Existen otras soluciones igualmente válidas como *Sphinx*, sin embargo se ha optado por utilizar *Apache Solr* ya que el *framework Spring* utilizado para la implementación del sistema incluye módulos que facilitan su integración.

	<i>build</i>	<i>insert</i>	<i>storage</i>	<i>query</i>	<i>solution</i>
LIKE expression	0	0	0	49k-399k ms	SQL
FULLTEXT MyISAM	31:18	33:28	2382MiB	16-200ms	MySQL
FULLTEXT InnoDB	25:27	55:46	?	350-740ms	MySQL 5.6
Apache Solr	n/a	14:28	2766MiB	79ms	Java
Sphinx Search	n/a	8:20	3487MiB	13ms	C++
Trigraphs	n/a	116:50	16.2 GiB	13,800ms	SQL

Figura 2.14: Comparativa de motores de búsqueda

Por tanto, la información de las organizaciones está almacenada tanto en la base de datos *MySQL* como en el repositorio de *Apache Solr*. Para mantener la consistencia de esta información, las operaciones que modifican estos datos lo hacen de forma paralela en ambas estructuras. Además se ha programado un proceso nocturno que sincroniza la información.

Las consultas de búsqueda se realizan como una conjunción de las diferentes palabras dadas, es decir los resultados buscados deben contar con todas las palabras incluidas en la consulta. Sin embargo, con el fin de mejorar los resultados devueltos, se realiza el siguiente tratamiento al texto de la consulta:


1. **Separación de las palabras (Tokenización):** El primer paso es obtener las diferentes palabras existentes en la consulta, separando por espacios, símbolos de puntuación, etc.
2. **Normalización:** Las palabras obtenidas deben ser convertidas a minúsculas y se deben eliminar sus tildes, símbolos especiales (paréntesis, interrogantes, exclamaciones, ...), etc.
3. **Eliminación de *stop-words*:** Algunas palabras como artículos, preposiciones, etc. no aportan información relevante a la consulta y pueden empeorar los resultados obtenidos. Por ello, antes de realizar la consulta deben eliminarse estas palabras.
4. **Obtener raíces de las palabras (*Stemming*):** Con el fin de mejorar los resultados devueltos, se ha utilizado *Snowball Token Filter* para aplicar un proceso de *stemming* [25] a la consulta, es decir para obtener las raíces de las palabras (lexemas). De este modo se aceptan resultados en los que aparezcan palabras que aunque sean diferentes a las solicitadas, poseen la misma raíz. Esto permite que el sistema sea parcialmente insensible a formas plurales, conjugaciones verbales, etc.

2.3.3. Clasificación automática de opiniones

A la hora de publicar una nueva opinión, el sistema es capaz de clasificar automáticamente los mensajes escritos por el usuario como opiniones positivas, neutrales o negativas. El resultado de esta clasificación se muestra al usuario en tiempo real, mientras éste escribe su opinión. Por ello es necesario obtener un sistema capaz de realizar clasificaciones multi-clase de forma rápida.

En este problema, las diferentes palabras de las opiniones son utilizadas como parámetros. Estos parámetros son generalmente independientes entre sí, ya que la existencia de determinadas palabras en una zona del texto no afecta al significado de las palabras escritas en otra zona. Sin embargo, puede darse el caso de que una palabra cercana a otra altere su significado (Figura 2.15). Debido a esto, no solo se contemplan las palabras de las opiniones individualmente, sino que también se utilizan N-gramas de palabras.

La organización no tramaba **nada bueno**.



El diagrama muestra una línea horizontal que comienza con una vertical descendente a la izquierda y una vertical ascendente a la derecha, formando un U invertido. Debajo de esta estructura, la palabra "NEGACIÓN" está centrada.

Figura 2.15: Dependencia entre palabras cercanas

Para implementar esta funcionalidad, se ha programado un sistema de aprendizaje automático basado en Bayes Ingenuo [26]. Se ha elegido este tipo de sistema debido a la independencia de los parámetros y a la facilidad de realizar un aprendizaje supervisado mediante modelos probabilistas.

$$P(y_i|X) = \frac{P(X|y_i)P(y_i)}{P(X)}$$
$$P(X) = \sum_{i=1}^N P(X|y_i)P(y_i)$$
$$P(X|y_i) = \prod_{j=1}^M P(x_j|y_i)$$

Figura 2.16: Clasificación Bayesiana

Como se muestra en la figura anterior, el sistema considera varios datos para realizar sus cálculos:

1. Y : Posibles resultados de la clasificación (Opinión positiva, opinión neutral, opinión negativa).
2. X : Conjunto de N-gramas que conforman la opinión a clasificar.
3. $P(y_i)$: Probabilidad a priori de que un texto sea una opinión del tipo y_i .
4. $P(X)$: Probabilidad de aparición de los N-gramas recogidos en X , sea cual sea el tipo de opinión.
5. $P(y_i|X)$: Probabilidad a posteriori. Probabilidad de que un texto sea del tipo y_i si posee los N-gramas recogidos en X .

Una vez realizados estos cálculos, puede aplicarse la siguiente regla de decisión para determinar a que tipo pertenece una opinión.

$$\hat{y} = \operatorname{argmax}_i P(y_i|X)$$

Figura 2.17: Regla de decisión del clasificador bayesiano

Para realizar estos cálculos y clasificar las opiniones, se necesitan los siguientes datos:

1. Número total de opiniones de cada tipo.
2. Número de apariciones de cada N-grama en cada tipo de opinión.
3. Probabilidad de aparición de cada N-grama en cada tipo de opinión.

Esta información es almacenada en una base de datos no relacional (*MongoDB*), ya que una base de datos relacional ocasionaría un retardo elevado en los procesos de entrenamiento y clasificación debido al gran número de consultas necesarias.

Este sistema tiene un inconveniente, ya que si se dan N-gramas con apariciones en un solo tipo de opinión, su probabilidad de aparición en el resto sería 0 y por tanto su probabilidad $P(X|y_i)$ es 0.

Esto da lugar a un gran número de clasificaciones erróneas, por lo que para solventarlo se utiliza el suavizado de Laplace que realiza un incremento ficticio de K unidades tanto en las apariciones de cada N-grama para cada tipo de opinión $P(x_j|y_i)$, como en las apariciones de cada tipo de opinión $P(y_i)$.

$$Probabilidad = \frac{Num. apariciones + K}{Total elementos + K \cdot Num. clases}$$

Figura 2.18: Aplicación del suavizado de Laplace

Para realizar el entrenamiento del sistema de aprendizaje se han recopilado un total de 999 opiniones (333 de cada tipo) de sitios como *TripAdvisor*, *Google Play* o *Ciao*. De este total de 999 opiniones se ha destinado un 80 % para el entrenamiento del sistema y un 20 % para probar la precisión adquirida.

Para obtener unos resultados óptimos, se han probado varios pre-tratamientos de texto para las opiniones, como aplicar una normalización, obtener las raíces de las palabras (*stemming*) o eliminar los *stop-words* (palabras que no aportan información útil). En el cuadro 2.5 se muestra la precisión obtenida al utilizar estas técnicas sobre un diferente número de N-gramas.

N-Gramas	Stemming	Tratamiento stop-words	Precisión (Laplace=0)	Precisión (Laplace=1)
1	0	0	44 % - 96 %	63 % - 94 %
1	0	1	44 % - 96 %	63 % - 93 %
1	1	0	47 % - 97 %	62 % - 93 %
1	1	1	49 % - 97 %	65 % - 93 %
2	0	0	39 % - 99 %	61 % - 92 %
2	0	1	43 % - 98 %	64 % - 93 %
2	1	0	38 % - 100 %	59 % - 93 %
2	1	1	45 % - 99 %	70 % - 95 %
3	0	0	38 % - 99 %	57 % - 93 %
3	0	1	43 % - 98 %	65 % - 92 %
3	1	0	37 % - 100 %	56 % - 93 %
3	1	1	45 % - 99 %	70 % - 95 %
4	0	0	38 % - 99 %	56 % - 93 %
4	0	1	43 % - 98 %	65 % - 92 %
4	1	0	37 % - 100 %	54 % - 94 %
4	1	1	44 % - 99 %	69 % - 95 %

Cuadro 2.5: Precisión del sistema de clasificación de opiniones

Debido a que no existe una frontera clara entre las opiniones neutrales y las opiniones de otro tipo, se han considerado dos niveles de precisión. Por ello, en el cuadro se muestran dos precisiones separadas por un guión, a la izquierda la precisión real y a la derecha la precisión considerando únicamente fallos críticos, es decir clasificar como negativa una opinión positiva (o viceversa).

El sistema es capaz de clasificar correctamente entre un 70 % y un 95 % de las opiniones. Éstas son cifras bastante aceptables considerando la dificultad del problema. Sin embargo hay margen de mejora, por lo que el sistema ejecuta un proceso semanal que continúa con el aprendizaje utilizando las nuevas opiniones de los usuarios, los cuales pueden modificar la clasificación de sus opiniones si no están conformes con la calculada. De este modo, el sistema puede aprender de sus errores y mejorar su precisión.

2.3.4. Obtención de ofertas de sitios externos

El sistema implementa técnicas de recuperación de información para obtener ofertas de proyectos, empleo o prácticas publicadas en sitios web externos como:

1. **StudentJob:** *Studentjob* es un portal de empleo especializado en ofertas de trabajo para estudiantes y jóvenes recién titulados.
2. **EINA:** La página web de la Escuela de ingeniería y arquitectura perteneciente a la Universidad de Zaragoza cuenta con diversas ofertas de proyectos para las diferentes titulaciones impartidas.
3. **Universia:** *Universia* es un portal dedicado a estudiantes universitarios que ofrece numerosas ofertas de empleo y prácticas.

Esta obtención de ofertas se realiza mediante un proceso de *scraping* [27] ejecutado semanalmente. Este proceso utiliza la librería *JSoup*, que facilita las tareas de detectar las ofertas de cada web analizada y procesar su contenido para obtener título, descripción, tipo, organización, provincia y titulación.

En este proceso la red es un “cuello de botella” inevitable, ya que se descarga mucha información de sitios web externos. Para optimizar el rendimiento, cada sitio se analiza en un hilo de ejecución independiente que descarga y procesa la información de las ofertas encontradas. Como resultado, el sistema inserta en la base de datos las nuevas ofertas y actualiza las ya existentes. En caso de detectar una oferta perteneciente a una organización que aun no existe en el sistema, ésta también es almacenada.

El procedimiento para detectar si una oferta obtenida ya se encuentra almacenada no es trivial, ya que la URL de origen puede haber cambiado, puede encontrarse la misma oferta en diferentes sitios web, etc. Para ello se obtienen todas las ofertas que coinciden con ella en provincia, organización, tipo y titulación. Tras ello, se comprueba si dichas ofertas cumple alguna de las siguientes condiciones:

1. Poseen la misma URL
2. Sus títulos y descripciones, tras una normalización, eliminación de *stop-words* y *stemming*, poseen una similaridad del 80 % y del 70 % respectivamente (según la distancia de Levenshtein [28]).

En el cuadro 2.6 se muestra el rendimiento del sistema a la hora de obtener y procesar las ofertas.

	Ofertas detectadas	Titulación detectada	Provincia detectada	Tipo detectado	Ofertas almacenadas
Student Job	1435	1288 (89,7 %)	1390 (96.8 %)	1390 (96.8 %)	1246 (86.8 %)
EINA	306	300 (98,0 %)	306 (100 %)	306 (100 %)	300 (98,0 %)
Universia	1753	1098 (62,6 %)	1519 (86.6 %)	1519 (86.6 %)	955 (54.4 %)
TOTAL	3494	2686 (76,8 %)	3215 (92,0 %)	3215 (92,0 %)	2501 (71.5 %)

Cuadro 2.6: Rendimiento del sistema de scraping

2.3.5. Tratamiento de opiniones

El sistema realiza ciertas validaciones en el proceso de publicación de opiniones para evitar comentarios inadecuados, como se explica a continuación:

1. **Tratamiento de *Leet Speak*:** *Leet Speak* es uno de los métodos más utilizados en foros y *chats* para escribir palabras inapropiadas evitando los filtros. Esta escritura sustituye las letras por números o símbolos equivalentes, como se muestra en la figura 2.19. Para evitar estas prácticas, se aplican una serie de transformaciones con el fin de recuperar las palabras originales. [29]

Original:
ESPERO QUE OS GUSTE EL PROYECTO

Leet Speak:
35p3r0 qu3 05 6u573 3l pr0y3c70

Figura 2.19: Ejemplo de escritura “Leet speak”

2. **Separación de palabras según *Letter Case*:** Otro método para introducir palabras inapropiadas consiste en unir palabras utilizando la capitalización de los caracteres iniciales para indicar su separación (figura 2.20). Por tanto, se aplican transformaciones para deshacer dichas palabras.

Original:
ESPERO QUE OS GUSTE EL PROYECTO

Texto transformado:
EsperoQueOsGusteElProyecto

Figura 2.20: Ejemplo de palabras separadas por capitalización

3. **Normalización:** El texto es sometido a un proceso de normalización en el que sus caracteres se convierten a minúsculas y se eliminan tildes, caracteres especiales, etc.

4. **Separación de palabras (Tokenización):** Tras normalizar el texto, se separan sus palabras para procesarlas una a una.
5. **Obtención de raíces de palabras (*Stemming*):** Con el fin de detectar las palabras inadecuadas independientemente de su forma plural, conjugación verbal, etc. Se obtiene la raíz (lexema) de cada una de ellas, utilizando un proceso de *stemming* mediante el filtro *Snowball*.
6. **Búsqueda de palabras inapropiadas:** Este proceso revisa si alguna de las palabras que forman el texto tratado se encuentra en una lista de palabras inapropiadas y, en caso afirmativo, las sustituye por la cadena “***”.

La lista de palabras inapropiadas utilizada en el último paso del proceso, es cargada en memoria al inicio de la ejecución del sistema, utilizando una estructura hash. Por lo que el coste de buscar una palabra concreta en dicha estructura es constante. Por tanto, el coste de ejecución total de este proceso es lineal en el número de palabras del texto a tratar, puesto que se deben revisar una a una.

Además de este proceso de censura, el sistema realiza otras verificaciones antes de permitir la publicación de una nueva opinión:

1. **Censura de URLs:** Por motivos de seguridad, no se permite la publicación de URLs en las opiniones, por lo que se utilizan expresiones regulares para buscar y censurar dichos elementos.
2. **Longitud del texto:** La longitud del texto a publicar debe estar entre 40 y 1000 caracteres.
3. **Idioma del texto:** Solo se permite la publicación de opiniones en español, por lo que el sistema verifica el idioma del texto a publicar, denegando otros idiomas.
4. **Aleatoriedad del texto:** El sistema aplica ciertas comprobaciones para determinar si un texto tiene sentido o se ha escrito de forma aleatoria. El coste de esta comprobación es lineal en el número de caracteres del texto, ya que se comprueba si se cumple alguno de los siguientes puntos:
 - a) Las palabras que componen el texto tienen menos de 3 caracteres de media o más de 10.
 - b) El texto contiene más de 6 vocales, consonantes o caracteres especiales seguidos.
 - c) El texto contiene más de 9 dígitos seguidos.
 - d) El texto contiene menos letras que dígitos y caracteres especiales juntos.

2.3.6. Aplicación web

La aplicación web se ha diseñado utilizando las tecnologías HTML y CSS a partir del prototipo realizado en la etapa de diseño. Por otra parte, su lógica se ha implementado en lenguaje *javascript* utilizando la librería *jQuery* [5].

A continuación se explican los puntos mas relevantes del funcionamiento de la aplicación web:

1. **Llamadas a los servicios web:** La aplicación web no implementa ningún aspecto de la lógica de negocio, simplemente se limita a realizar llamadas *Ajax* a los servicios web del sistema.
2. **Gestión de estado:** La aplicación web es la encargada de gestionar el estado de la comunicación cliente-servidor, almacenando los *tokens* correspondientes en una *cookie* de duración limitada.
3. **Principios SEO:** Como se explica en el Anexo B, en la aplicación web se utilizan ciertas técnicas de SEO (*Search Engine Optimization*) como el uso de *snapshots* HTML, URLs semánticas... Con esto se trata de mejorar el posicionamiento del sitio en los diferentes buscadores con el fin de llegar a un público más amplio a través de estos medios.

La aplicación cuenta con diferentes páginas y elementos que exponen las distintas funcionalidades:

1. **Página de inicio (*Ranking* de organizaciones):** Página principal con un buscador de organizaciones y el *ranking* con las mejores y peores organizaciones de la provincia y país del usuario.
2. **Panel de usuario anónimo:** Panel desplegable desde la cabecera del sitio que permite a los usuarios no identificados identificarse, registrarse o recuperar una contraseña olvidada.
3. **Panel de usuario identificado:** Panel desplegable desde la cabecera del sitio que permite a los usuarios identificados realizar acciones como ver las novedades del sitio o cambiar los ajustes de su cuenta.
4. **Listado de organizaciones:** Página que muestra un listado con los nombres y puntuaciones de las organizaciones solicitadas por el usuario, así como diferentes filtros y opciones para alterar la visualización de dicha lista.
5. **Listado de ofertas:** Página que muestra un listado con las ofertas laborales solicitadas por el usuario, así como diferentes filtros y opciones para alterar la visualización de dicha lista.
6. **Opiniones de organización:** Página que muestra las puntuaciones y las opiniones laborales publicadas por los usuarios sobre una organización concreta.
7. **Detalles de valoración de organización:** Página que muestra una serie de gráficos que representan el estado actual y la evolución de la valoración de una organización concreta.
8. **Formulario de publicación de opinión:** Página que incluye un formulario que permite la publicación de nuevas opiniones laborales sobre organizaciones.
9. **Formulario de publicación de oferta:** Página que incluye un formulario que permite la publicación de nuevas ofertas laborales.
10. **Listado de usuarios pendientes de activar:** Página que incluye un listado desde el que los administradores pueden activar las cuentas de los usuarios registrados como organización.

2.3.7. Pruebas funcionales (Integración continua)

Con el fin de garantizar el correcto funcionamiento del sistema, se ha elaborado una colección de pruebas que cubre una gran cantidad de casos de uso. A continuación se explican los tipos de prueba más importantes que se han realizado:

1. **Pruebas unitarias:** Las pruebas unitarias [30] proporcionan mecanismos para probar el correcto funcionamiento de módulos concretos de código. En este proyecto se ha utilizado la tecnología *JUnit* para elaborar un total de 93 pruebas unitarias que verifican la corrección de los diferentes métodos y funciones implementados.
2. **Pruebas de integración:** Las pruebas de integración [31] se encargan de comprobar la corrección de varios métodos o funciones ejecutados conjuntamente. En este proyecto se ha utilizado la tecnología *JUnit* junto con el entorno de pruebas facilitado por *Spring* para realizar un total de 49 pruebas que verifican el correcto funcionamiento de los diferentes servicios web implementados.
3. **Pruebas de sistema:** Las pruebas del sistema consisten en utilizar la aplicación web para probar casos de uso concretos que podrían realizar los usuarios reales del sistema. Los casos de uso probados se detallan en el Anexo A.

Tanto las pruebas unitarias como las pruebas de integración se han ido elaborando a lo largo del proceso de implementación y se han ejecutado en el entorno de integración continua *Travis-CI*. Este entorno es capaz de sincronizarse con el sistema de control de versiones utilizado, en este caso *GitHub*, de modo que cada vez que se registra una nueva versión del sistema en desarrollo, se realiza la ejecución automática de las pruebas programadas. Esto permite una detección temprana de los posibles fallos existentes en el sistema.

En el Anexo B se explican con mayor detalle las diferentes pruebas realizadas, incluyendo otros tipos de prueba como pruebas *cross-browser*, pruebas de carga y pruebas *beta*.

2.3.8. Conclusiones

La implementación del sistema se ha realizado utilizando lenguaje *Java* junto al *framework Spring*. Mientras que la lógica de la aplicación web utiliza la librería *JQuery* basada en *Javascript*.

Para poder proporcionar las funcionalidades previstas se han usado técnicas de recuperación de información (*scraping*) y de aprendizaje automático (Bayes Ingenuo). Además se han utilizado técnicas de tratamiento de información en lenguaje natural que permiten su normalización y procesamiento.

Finalmente, con el fin de comprobar el correcto funcionamiento del sistema se han elaborado una serie de pruebas que permiten determinar si el funcionamiento es el esperado.

Capítulo 3

Gestión del proyecto

Como ya se ha explicado, en este proyecto se ha utilizado una metodología de desarrollo ágil [8] en la que las tareas de desarrollo se dividen en iteraciones. En este capítulo se explican aspectos como la planificación de dichas iteraciones, en la sección 3.1, las horas dedicadas a la realización del proyecto, detalladas en la sección 3.2, y las herramientas de gestión utilizadas, explicadas en la sección 3.3.

3.1. Planificación

Como puede observarse en la figura 3.1, el proyecto comienza el día 15 de Enero con una fase preliminar destinada al análisis del problema y el diseño de la solución. Tras esta fase, comienza el desarrollo del proyecto dividido en seis iteraciones de 26 días de duración, en las que se realiza la implementación y testeo de todas las características del sistema. Finalmente se procede a la puesta en producción del sistema y a la elaboración de la documentación del proyecto, con el fin de cumplir los plazos previstos en los días 15 de Agosto y 1 de Septiembre respectivamente.

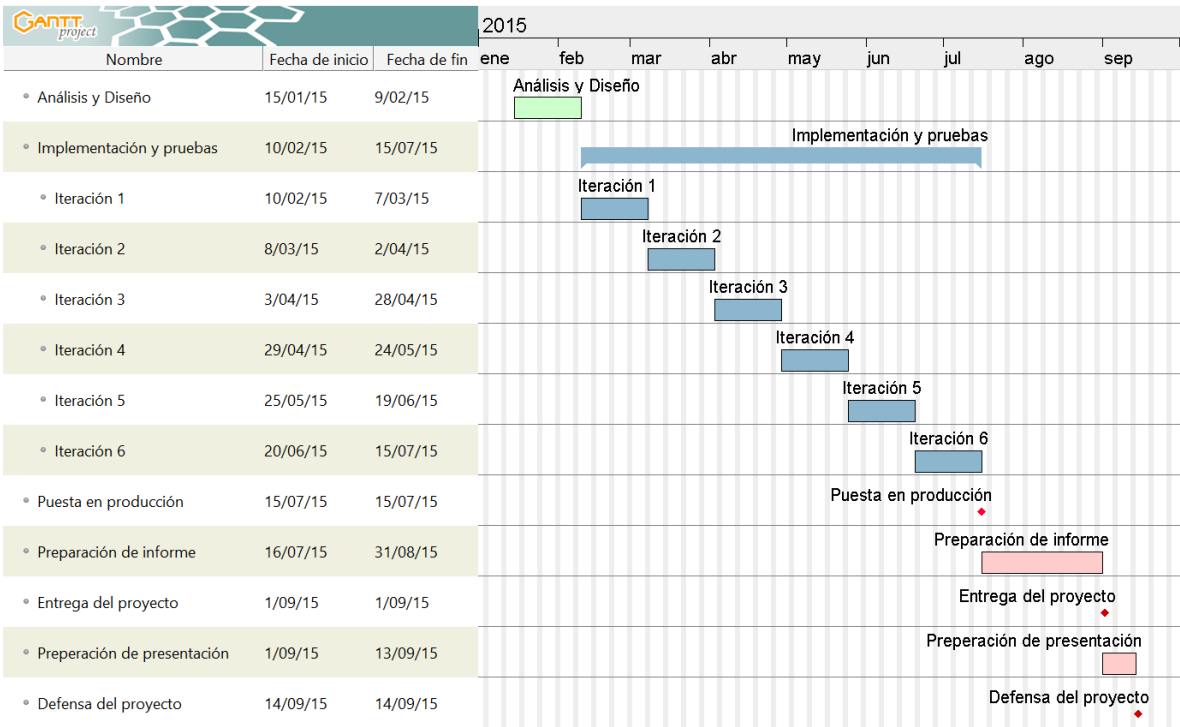


Figura 3.1: Calendario GANTT

3.2. Tiempo dedicado

En la realización de este proyecto se han dedicado un total de 612 horas repartidas entre tareas de análisis, diseño, implementación, pruebas, aprendizaje, documentación y otros tipos de actividades, como se muestra en la figura 3.2.

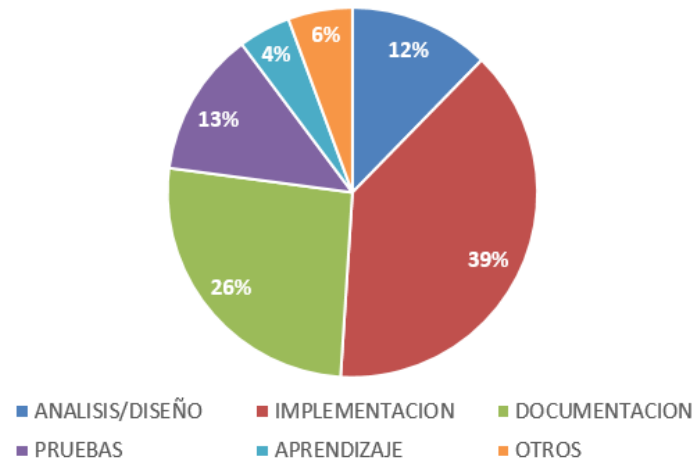


Figura 3.2: Tiempo dedicado por tipo de tarea

3.3. Herramientas de gestión

Para una correcta gestión del proyecto se han utilizado las siguientes herramientas:

1. **Github:** *Github* ha sido la plataforma elegida para mantener el control tanto de las diferentes versiones del sistema desarrollado, como de las tareas a realizar, *bugs* detectados, etc.
2. **Google Keep:** *Google Keep* es una sencilla herramienta para tomar notas rápidas que se ha utilizado en el día a día del proceso de desarrollo.
3. **GanttProject:** *GanttProject* es un software utilizado para la elaboración de diagramas de tipo GANTT que permiten plasmar el calendario que se debe seguir en el proceso de desarrollo.
4. **Dropbox:** *Dropbox* es una herramienta de almacenamiento en la nube que se ha utilizado para mantener un directorio de trabajo compartido con el tutor del proyecto.
5. **Procesador de textos:** Se han utilizado diferentes programas de procesamiento de texto, por ejemplo *Microsoft Word* o *Texmaker*, para elaborar y mantener documentos como manuales de despliegue, documentos de gestión de configuración, etc.
6. **Procesador de hojas de cálculo:** Se han utilizado diferentes programas de procesamiento de hojas de cálculo como *Microsoft Excel*, para elaborar documentos como listas de *user stories*, listas de tareas, listas de requisitos, gráficos *burndown* o documentos de gestión de esfuerzos.

Capítulo 4

Conclusiones

Este capítulo expone una valoración del proyecto realizado. En la sección 4.1 se revisa el cumplimiento de requisitos por parte del producto. En la sección 4.2 se exponen las lecciones aprendidas. Y en las secciones 4.3 y 4.4 se realiza una valoración personal del proyecto y de sus posibilidades de futuro.

4.1. Resultados

Como resultado del proyecto se ha obtenido un sistema totalmente funcional, accesible a través de la dirección <http://www.universando.es>. El proyecto cumple con todos los requisitos y permite:

1. Publicar, leer y contestar opiniones anónimas sobre experiencias laborales en organizaciones.
2. Publicar, leer, contestar y gestionar ofertas, así como obtenerlas de sitios externos.
3. Conocer las organizaciones más valoradas y buscar opiniones u ofertas de organizaciones.
4. Administrar usuarios, organizaciones, opiniones y ofertas existentes.

Además utiliza un sistema basado en Bayes Ingenuo que clasifica las opiniones como positivas, negativas o neutrales, con una precisión del 70 % y que aprende de sus errores, mejorando la precisión.

4.2. Lecciones aprendidas

A continuación se explican las lecciones más relevantes aprendidas durante la realización del proyecto.

1. **Los *frameworks* agilizan el desarrollo y mejoran la calidad del resultado:** *Frameworks* como *Spring* proporcionan una base de lógica ya implementada que facilita la implantación de aspectos básicos. Permitiendo centrar el esfuerzo en la parte propia de la lógica de negocio.
2. **La integración continua permite la detección temprana de fallos:** Plataformas como *Travis* permiten la detección inmediata de fallos mediante la ejecución automática de pruebas.
3. **Los servicios web favorecen la interoperabilidad:** Los servicios web separan la capa lógica de la capa frontal, reduciendo la complejidad y permitiendo el uso de la lógica por terceras partes.
4. **Para implementar búsquedas de texto es recomendable usar motores de búsqueda:** Sistemas como *MySQL* no ofrecen tiempos de respuesta óptimos. Se recomiendan motores como *Apache Solr* que crean índices para realizar búsquedas eficientes y mejoran la escalabilidad.
5. ***Gradle* permite automatizar el proceso de compilación y ejecución:** *Gradle* gestiona las dependencias del sistema, su compilación y la ejecución automática de pruebas. Además permite ejecutar el sistema en servidores web embebidos en el propio *script* de *Gradle*.

6. **Las aplicaciones web *responsive* permiten llegar a un público mas amplio:** Las aplicaciones web *responsive* adaptan su aspecto según la resolución, garantizando la mejor experiencia desde cualquier dispositivo y pudiendo llegar a un mayor público.
7. **Los sitios web basados en *Ajax* requieren tratamiento especial para ser procesados por buscadores:** *Google* es incapaz de recuperar información de páginas que obtienen y escriben su contenido de forma dinámica en el cliente. Es necesario crear *snapshots* con contenido fijo.
8. **El procesamiento de grandes cantidades de información persistente requiere de bases de datos no relacionales:** Usar bases de datos relacionales en procesos con un gran número de operaciones de lectura/escritura, puede ocasionar un cuello de botella que ralentice el sistema. Se recomienda el uso de una base de datos no relacional como *MongoDB*.
9. **El patrón de diseño DAO permite encapsular el acceso a la base de datos:** Este patrón separa toda interacción con la base de datos del resto de la lógica. De este modo el acceso a la base de datos se encapsula en una API usada desde el resto del código.

4.3. Conclusión personal

La realización del proyecto ha sido una buena experiencia de aprendizaje. Ha conllevado una carga de trabajo y una dificultad considerables ya que engloba la planificación y desarrollo de un sistema complejo utilizando una gran diversidad de tecnologías junto a técnicas de recuperación de información, aprendizaje automático y servicios web. Se ha dedicado una gran cantidad de tiempo a la exploración de soluciones y al aprendizaje de nuevas tecnologías, con el fin de llegar a un producto de calidad. Por tanto, considero que ha sido una experiencia muy útil que me ha permitido aprender de las dificultades y me será de gran utilidad para agilizar y mejorar el desarrollo de futuros proyectos.

4.4. Futuro del proyecto

El objetivo final es obtener un producto rentable. Para lograrlo y garantizar la continuidad del proyecto, se han definido varios puntos a realizar una vez terminada la naturaleza académica del proyecto:

1. **Captación de potenciales usuarios:** Para fomentar la afluencia de usuarios, se realizarán tareas de difusión en redes sociales, medios de comunicación...
2. **Actuar según el *feedback* recibido:** Una vez desarrollada la versión mínima viable del producto, se debe atender al *feedback* de los usuarios y adaptar el producto a sus necesidades.
3. **Mejorar sistema de obtención de ofertas:** El sistema de *scraping* implementado rastrea una cantidad reducida de sitios web en busca de ofertas, por lo que conviene ampliar esta lista de sitios para ofrecer una mayor diversidad en cuanto a lugar, tipo y titulación de las ofertas.
4. **Rentabilizar sitio web:** Con el fin de obtener beneficios, se debe incluir un medio para generar ingresos como la incorporación de *banners* de publicidad en el sitio web.

Bibliografía

- [1] Spring. Spring framework. <http://projects.spring.io/spring-framework>.
[Fecha de acceso: 01-Agosto-2015].
- [2] JSoup. Jsoup: Java html parser. <http://jsoup.org>.
[Fecha de acceso: 03-Agosto-2015].
- [3] Wikipedia. Nosql. <https://es.wikipedia.org/wiki/NoSQL>.
[Fecha de acceso: 01-Agosto-2015].
- [4] Apache. Apache solr. <http://lucene.apache.org/solr>.
[Fecha de acceso: 01-Agosto-2015].
- [5] JQuery. What is jquery? <https://jquery.com>.
[Fecha de acceso: 01-Agosto-2015].
- [6] Gradle. Gradle. <https://gradle.org>.
[Fecha de acceso: 01-Agosto-2015].
- [7] Wikipedia. Integración continua. https://es.wikipedia.org/wiki/Integraci%C3%B3n_continua.
[Fecha de acceso: 01-Agosto-2015].
- [8] Wikipedia. Desarrollo ágil de software. https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software.
[Fecha de acceso: 01-Agosto-2015].
- [9] Wikipedia. User story. https://en.wikipedia.org/wiki/User_story.
[Fecha de acceso: 01-Agosto-2015].
- [10] Ash Maurya. O'Reilly. *Running LEAN. Iterate from Plan A to a Plan That Works*. 2012.
- [11] Javier Megias. Lean canvas, un lienzo de modelos de negocio para startups. <http://javiermegias.com/blog/2012/10/lean-canvas-lienzo-de-modelos-de-negocio-para-startups-emprendedores>, 2012.
[Fecha de acceso: 01-Agosto-2015].
- [12] Sam Ruby. O'Reilly Leonard Richardson. *RESTful Web Services*. 2007.
- [13] Stefan Tilkov. Principios de rest. <http://www.dosideas.com/noticias/java/332-principios-de-rest.html>, 2008.
[Fecha de acceso: 02-Agosto-2015].

- [14] Wikipedia. Cross site request forgery. https://es.wikipedia.org/wiki/Cross_Site_Request_Forgery.
[Fecha de acceso: 02-Agosto-2015].
- [15] Spring. Building a restful web service. <https://spring.io/guides/gs/rest-service>.
[Fecha de acceso: 02-Agosto-2015].
- [16] Study Trails. Spring security: Securing methods using secured annotation. <http://www.studytrails.com/frameworks/spring/spring-security-method-level.jsp>.
[Fecha de acceso: 02-Agosto-2015].
- [17] Codesandnotes. Spring security's csrf protection for rest services: the client side and the server side. <http://www.codesandnotes.be/2015/02/05/spring-securitys-csrf-protection-for-rest-services-the-client-side-and-the-server-side>, 2015.
[Fecha de acceso: 02-Agosto-2015].
- [18] Paul Chapman. Exception handling in spring mvc. <https://spring.io/blog/2013/11/01/exception-handling-in-spring-mvc>, 2013.
[Fecha de acceso: 02-Agosto-2015].
- [19] Mkyong. Spring + jdbc example. <http://www.mkyong.com/spring/maven-spring-jdbc-example>, 2010.
[Fecha de acceso: 02-Agosto-2015].
- [20] Mkyong. Spring data mongodb hello world example. <http://www.mkyong.com/mongodb/spring-data-mongodb-hello-world-example>, 2011.
[Fecha de acceso: 02-Agosto-2015].
- [21] Spring. Task execution and scheduling. <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/scheduling.html>.
[Fecha de acceso: 02-Agosto-2015].
- [22] Yashwant Chavan. Spring email velocity template example. <http://www.technicalkeeda.com/spring/spring-email-velocity-template-example>, 2014.
[Fecha de acceso: 02-Agosto-2015].
- [23] Petri Kainulainen. Spring data solr tutorial. <http://www.petrikainulainen.net/spring-data-solr-tutorial>, 2010.
[Fecha de acceso: 02-Agosto-2015].
- [24] Karwin Software Solutions LLC. Full text search throwdown. <http://www.slideshare.net/billkarwin/practical-full-text-search-with-my-sql>, 2009.
[Fecha de acceso: 03-Agosto-2015].
- [25] Wikipedia. Stemming. <https://en.wikipedia.org/wiki/Stemming>.
[Fecha de acceso: 03-Agosto-2015].

- [26] University of Toronto. Naive bayes classification. <http://www.cs.toronto.edu/~bonner/courses/2007s/csc411/lectures/03bayes.zemel.pdf>, 2007.
[Fecha de acceso: 04-Agosto-2015].
- [27] Wikipedia. Web scraping. https://es.wikipedia.org/wiki/Web_scraping.
[Fecha de acceso: 03-Agosto-2015].
- [28] Wikipedia. Levenshtein distance. https://en.wikipedia.org/wiki/Levenshtein_distance.
[Fecha de acceso: 03-Agosto-2015].
- [29] Kim Togo. L33t tr4nsl4t0r (leet translator). <http://www.codeproject.com/Tips/207582/L-t-Tr-nsl-t-r-Leet-Translator>, 2011.
[Fecha de acceso: 03-Agosto-2015].
- [30] Felipe Martínez D. Junit en eclipse. pruebas unitarias en java. <http://felinfo.blogspot.com.es/2013/06/junit-en-eclipse-pruebas-unitarias-en.html>, 2013.
[Fecha de acceso: 03-Agosto-2015].
- [31] Wikipedia. Integration testing. https://en.wikipedia.org/wiki/Integration_testing.
[Fecha de acceso: 03-Agosto-2015].

Anexo A

Casos de uso detallados

En este anexo se muestran una serie de casos de uso expandidos que recogen las acciones más comunes que pueden realizarse con el sistema, junto con los diagramas de secuencia que representan dichas acciones.

Caso de uso	Ver mejores organizaciones
Nº de CU	1
Actores	Usuario anónimo, universitario o administrador
Propósito	Obtener el listado de las mejores organizaciones en la provincia del usuario.
Descripción breve	Un usuario anónimo o identificado como universitario o administrador accede a la página inicial del sitio web. El sistema obtiene la provincia del usuario, recupera el <i>ranking</i> de mejores organizaciones para dicha provincia y lo muestra.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede a la página inicial del sitio web.	2 Muestra el contenido del sitio web junto con el <i>ranking</i> de mejores organizaciones para la provincia del usuario.
Cursos de eventos alternativos	

Cuadro A.1: Caso de uso expandido: Ver mejores organizaciones

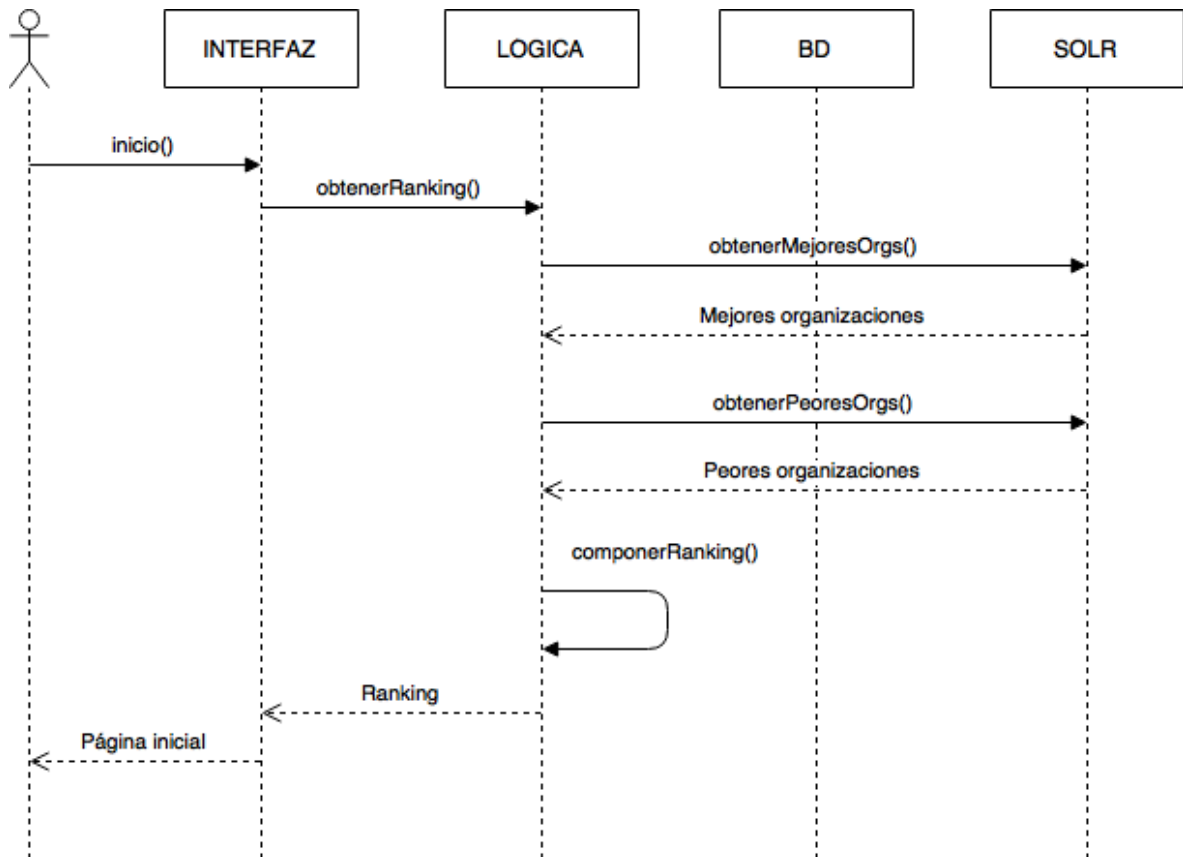


Figura A.1: Diagrama de secuencia: Ver mejores organizaciones

Caso de uso	Listar organizaciones
Nº de CU	2
Actores	Usuario anónimo, universitario o administrador
Propósito	Obtener un listado con las organizaciones almacenadas en el sistema.
Descripción breve	Un usuario anónimo o identificado como universitario o administrador accede al sitio web y solicita el listado de todas las organizaciones. Posteriormente aplica filtros y reordenaciones para ajustar el resultado. El sistema devuelve dicho listado y lo muestra por pantalla.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede a la página inicial del sitio web.	2 Muestra el contenido del sitio web.
3 Accede al listado de organizaciones (Botón “Listar organizaciones”).	4 Muestra la lista de organizaciones.
5 Modifica los filtros del listado.	6 Muestra las organizaciones que cumplen los criterios dados.
7 Solicita una reordenación del listado.	8 Muestra las organizaciones que cumplen los criterios dados, ordenadas según la característica elegida.
Cursos de eventos alternativos	

Cuadro A.2: Caso de uso expandido: Listar organizaciones

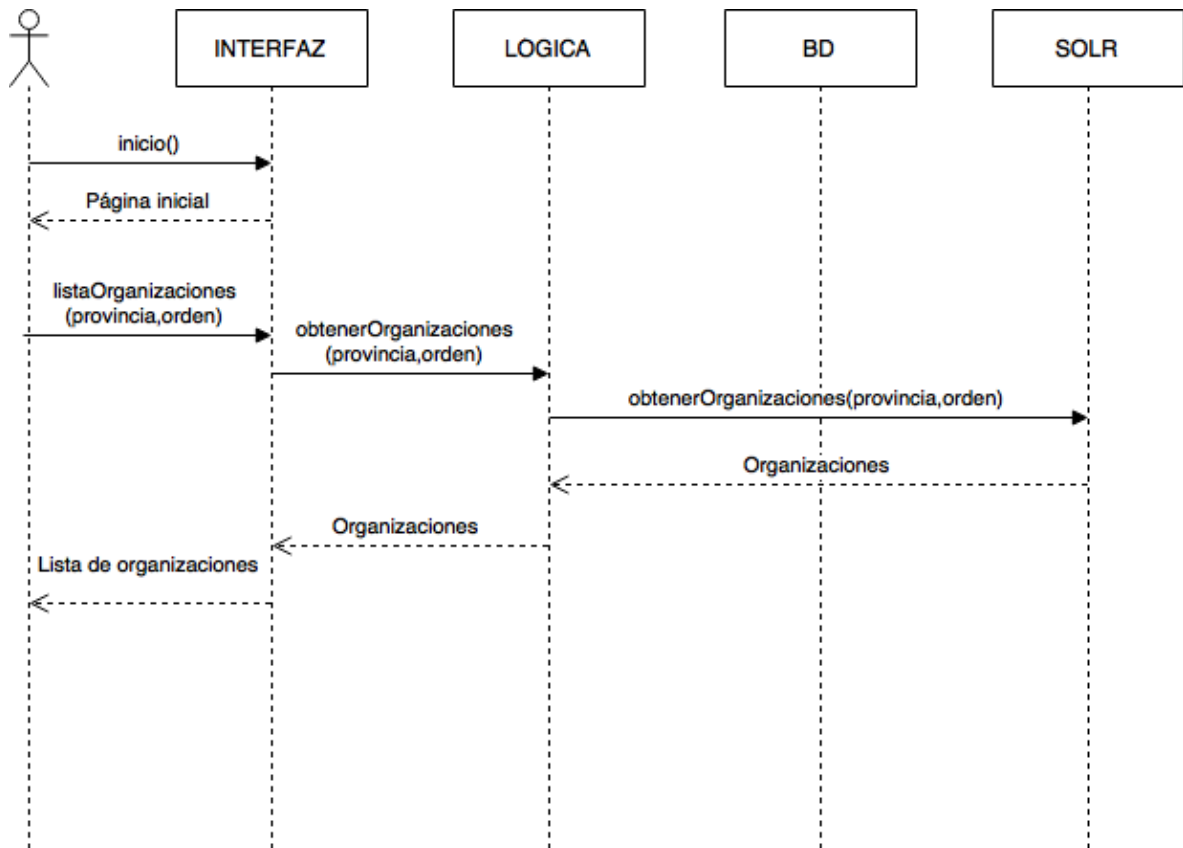


Figura A.2: Diagrama de secuencia: Listar organizaciones

Caso de uso	Buscar organizaciones
Nº de CU	3
Actores	Usuario anónimo, universitario o administrador
Propósito	Obtener listado de organizaciones que cumplen una serie de criterios.
Descripción breve	Un usuario anónimo o identificado como universitario o administrador realiza una búsqueda de organizaciones para unos criterios determinados. Posteriormente aplica filtros y reordenaciones para ajustar el resultado. El sistema obtiene y devuelve las organizaciones que cumplen las condiciones.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede a la página inicial del sitio web. 3 Completa la información del formulario de búsqueda. 5 Modifica los criterios de búsqueda usando los filtros. 7 Solicita una reordenación de los resultados.	2 Muestra el contenido del sitio web. 4 Muestra las organizaciones que cumplen los criterios dados. 6 Muestra las organizaciones que cumplen los criterios dados. 8 Muestra las organizaciones que cumplen los criterios dados, ordenadas según la característica elegida.
Cursos de eventos alternativos	

Cuadro A.3: Caso de uso expandido: Buscar organizaciones

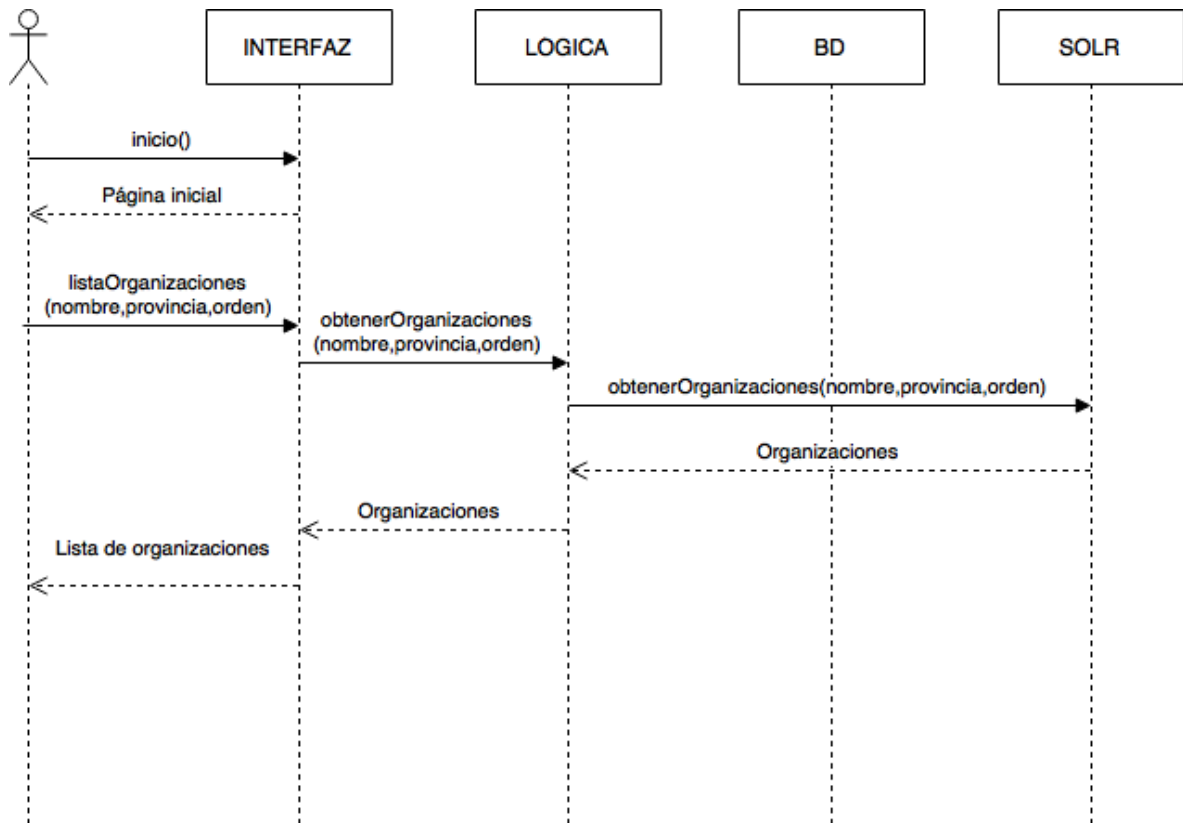


Figura A.3: Diagrama de secuencia: Buscar organizaciones

Caso de uso	Registrar usuario
Nº de CU	4
Actores	Usuario anónimo
Propósito	Crear una nueva cuenta de usuario del tipo universitario u organización.
Descripción breve	Un usuario anónimo rellena el formulario de registro de usuario. El sistema procesa la información y crea un nuevo usuario.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al formulario de nuevo registro. 5 Completa el formulario de registro.	2 Muestra el contenido del sitio web. 4 Muestra el formulario de registro. 6 Valida la información de registro y almacena el nuevo usuario. 7 Activa los mecanismos de confirmación del registro y muestra las instrucciones del proceso.
Cursos de eventos alternativos	
Línea 6: La información proporcionada es incorrecta. Se muestra un error y se cancela el proceso.	

Cuadro A.4: Caso de uso expandido: Registrar usuario

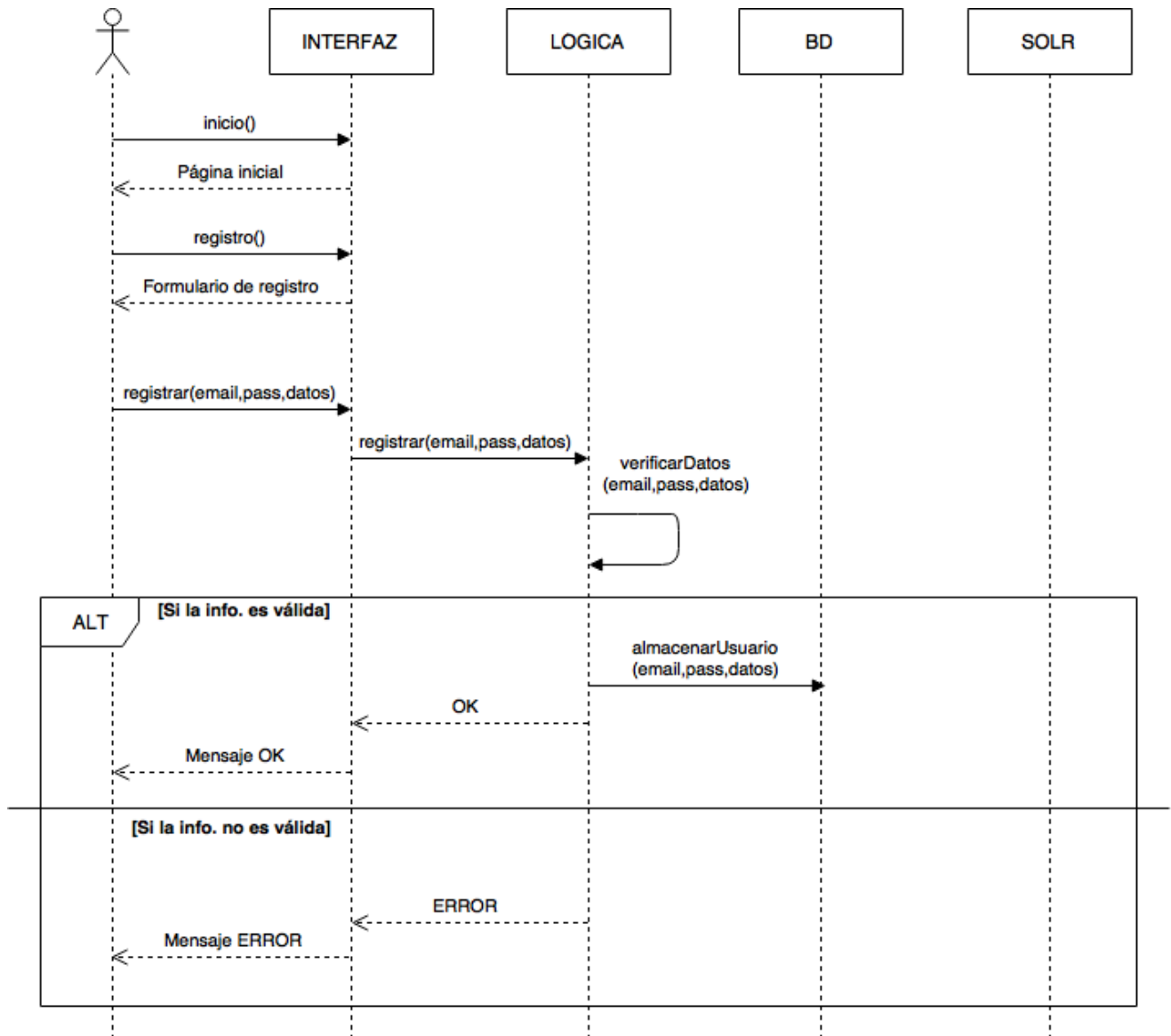


Figura A.4: Diagrama de secuencia: Registrar usuario

Caso de uso	Identificar usuario
Nº de CU	5
Actores	Usuario anónimo
Propósito	Acceder al sistema como un usuario registrado.
Descripción breve	Un usuario anónimo rellena la información de identificación para acceder al sistema como usuario universitario, organización o administrador. El sistema valida esta información y da acceso al usuario.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web.	2 Muestra el contenido del sitio web.
3 Accede al formulario de identificación.	4 Muestra el formulario de identificación.
5 Completa la información de identificación (email y contraseña).	6 Valida la información del usuario y concede el acceso.
Cursos de eventos alternativos	
Línea 6: La información proporcionada es incorrecta. Se muestra un error y se cancela el proceso.	

Cuadro A.5: Caso de uso expandido: Identificar usuario

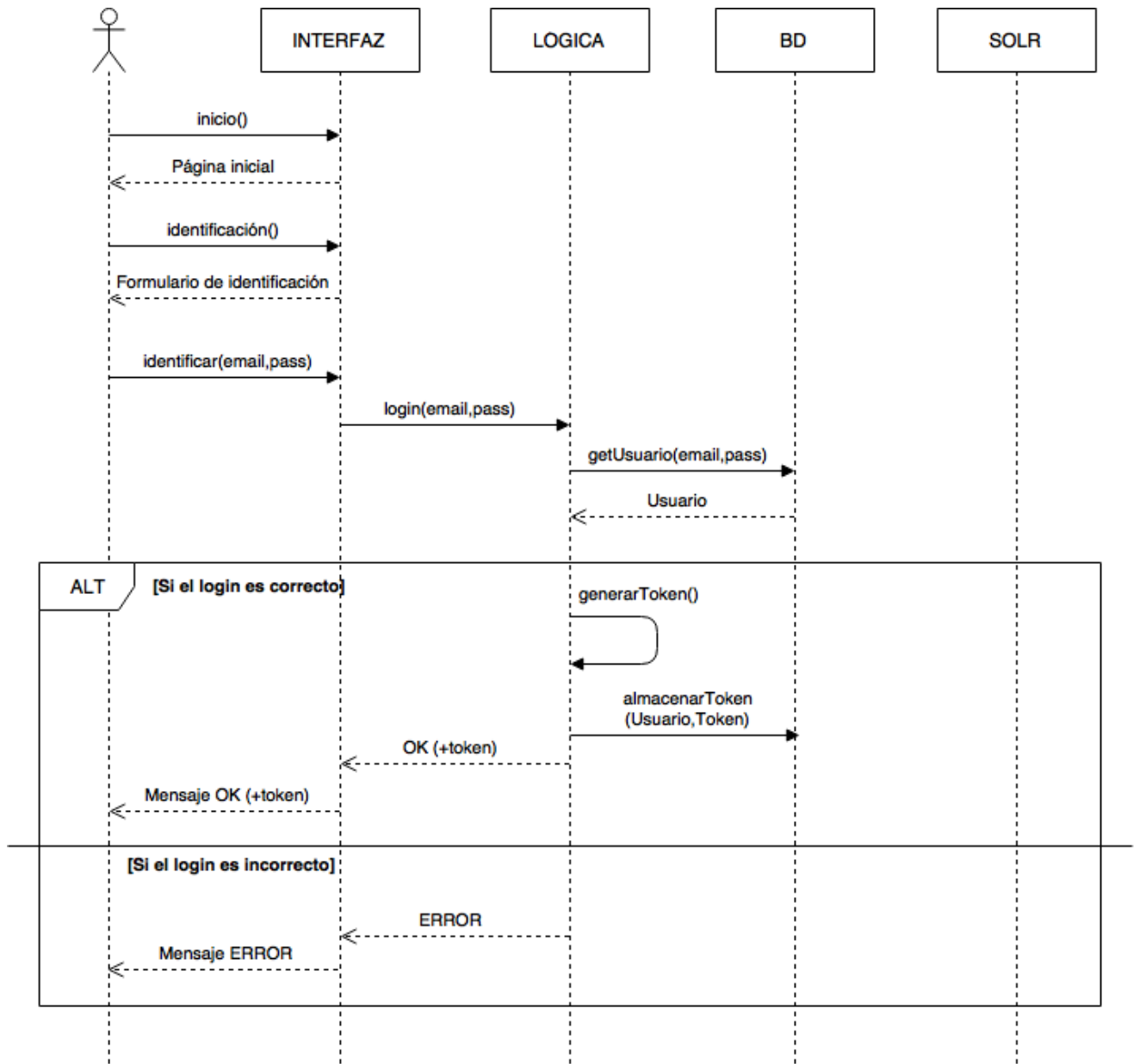


Figura A.5: Diagrama de secuencia: Identificar usuario

Caso de uso	Ver opiniones de organización
Nº de CU	6
Actores	Usuario universitario o administrador
Propósito	Obtener las opiniones que los usuarios han publicado sobre una organización concreta.
Descripción breve	Un usuario identificado como universitario o administrador accede al listado de organizaciones y solicita las opiniones publicadas sobre una ellas. El sistema procesa la solicitud y devuelve el listado de opiniones solicitado.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al listado de organizaciones. 5 Accede a la página de opiniones de una de las organizaciones.	2 Muestra el contenido del sitio web. 4 Muestra el listado de organizaciones. 6 Recupera y muestra las opiniones de la organización indicada.
Cursos de eventos alternativos	
Línea 3: Realiza una búsqueda de organizaciones estableciendo diferentes criterios.	

Cuadro A.6: Caso de uso expandido: Ver opiniones de organización

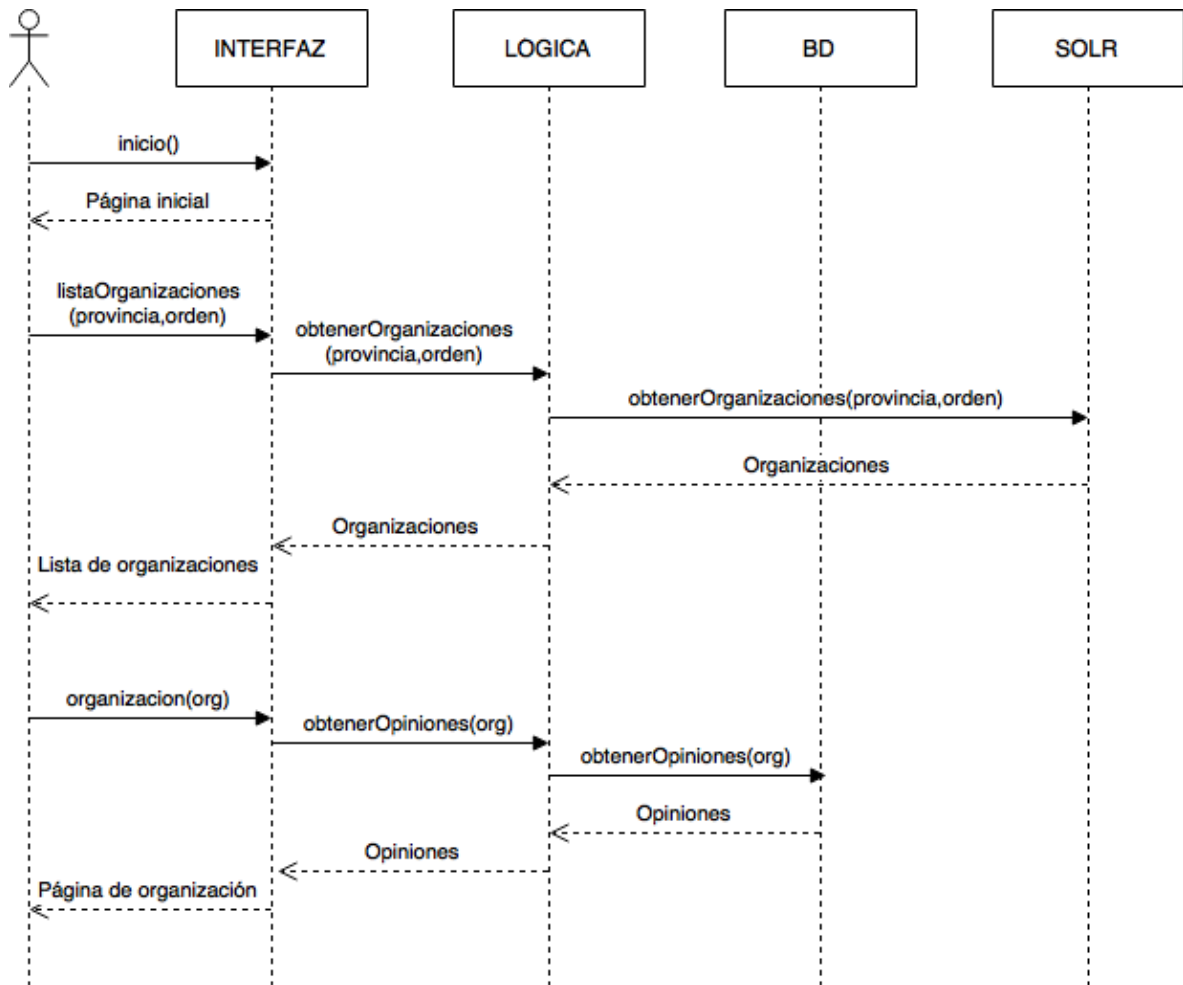


Figura A.6: Diagrama de secuencia: Ver opiniones de organización

Caso de uso	Ver ofertas de organización
Nº de CU	7
Actores	Usuario universitario o administrador
Propósito	Obtener las ofertas publicadas por una organización concreta.
Descripción breve	Un usuario identificado como universitario o administrador accede al sitio web, localiza una organización concreta y obtiene el listado de sus ofertas. El sistema procesa la solicitud y devuelve las ofertas disponibles.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al listado de organizaciones. 5 Accede a la página de opiniones de una de las organizaciones. 7 Accede a la página de ofertas de la organización (Botón “Ver ofertas”).	2 Muestra el contenido del sitio web. 4 Muestra el listado de organizaciones. 6 Recupera y muestra las opiniones de la organización indicada. 8 Recupera y muestra las ofertas de la organización indicada.
Cursos de eventos alternativos	
Linea 3: Realiza una búsqueda de organizaciones estableciendo diferentes criterios.	

Cuadro A.7: Caso de uso expandido: Ver ofertas de organización

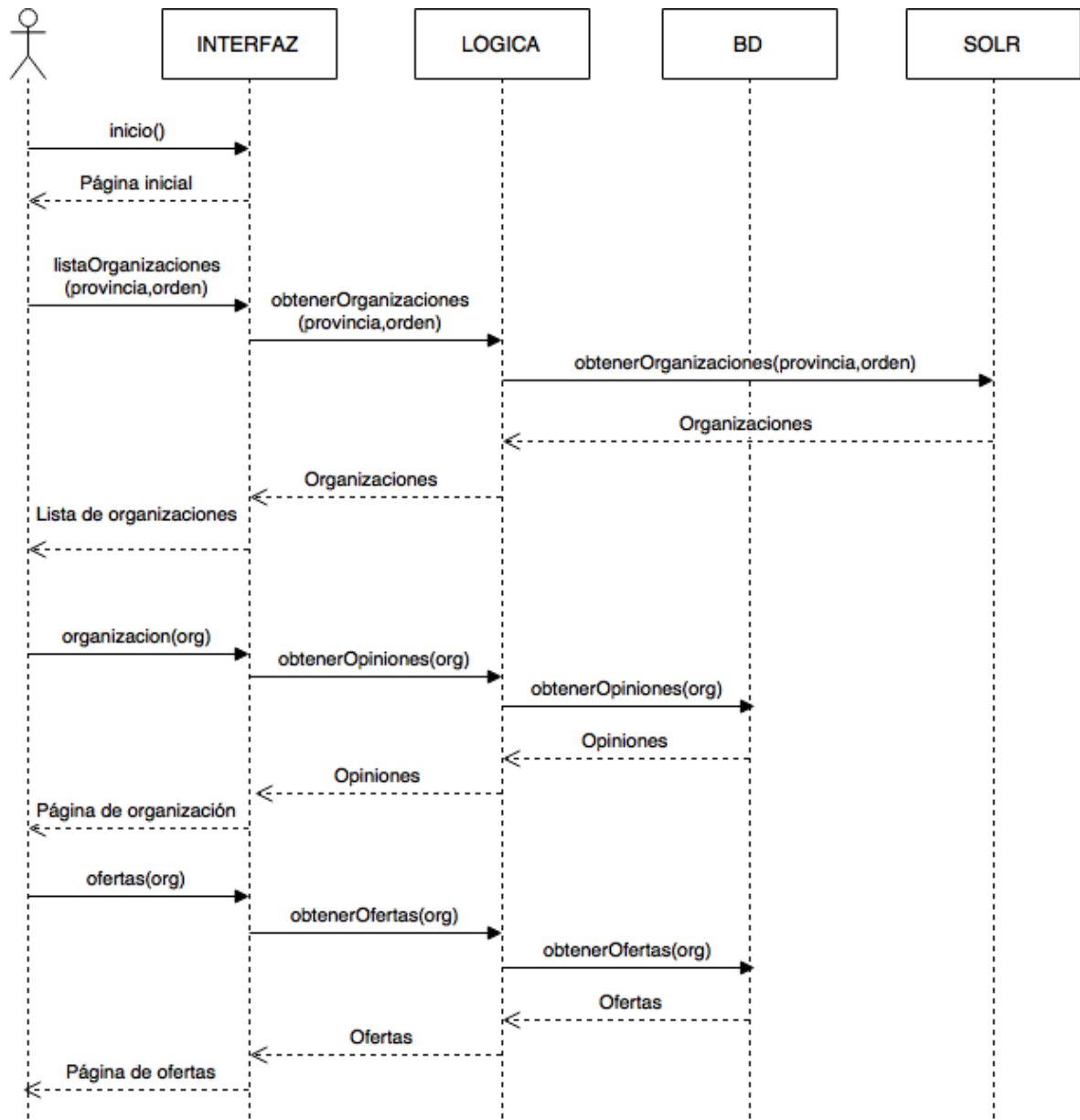


Figura A.7: Diagrama de secuencia: Ver ofertas de organización

Caso de uso	Listar ofertas
Nº de CU	8
Actores	Usuario universitario o administrador
Propósito	Obtener el listado de ofertas disponibles en el sistema.
Descripción breve	Un usuario identificado como universitario o administrador accede al sitio web y solicita el listado de las ofertas disponibles. Posteriormente aplicar filtros y reordenaciones para ajustar el resultado. El sistema procesa la petición y devuelve la lista de ofertas.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web.	2 Muestra el contenido del sitio web.
3 Accede al listado de ofertas (Botón “Listar ofertas”).	4 Muestra el listado de ofertas.
5 Aplica filtros sobre el listado de ofertas.	6 Muestra el listado de ofertas que cumplen los criterios dados.
7 Aplica una reordenación sobre las ofertas del listado.	8 Muestra el listado de ofertas que cumplen los criterios dados, ordenadas por la característica indicada.
Cursos de eventos alternativos	

Cuadro A.8: Caso de uso expandido: Listar ofertas

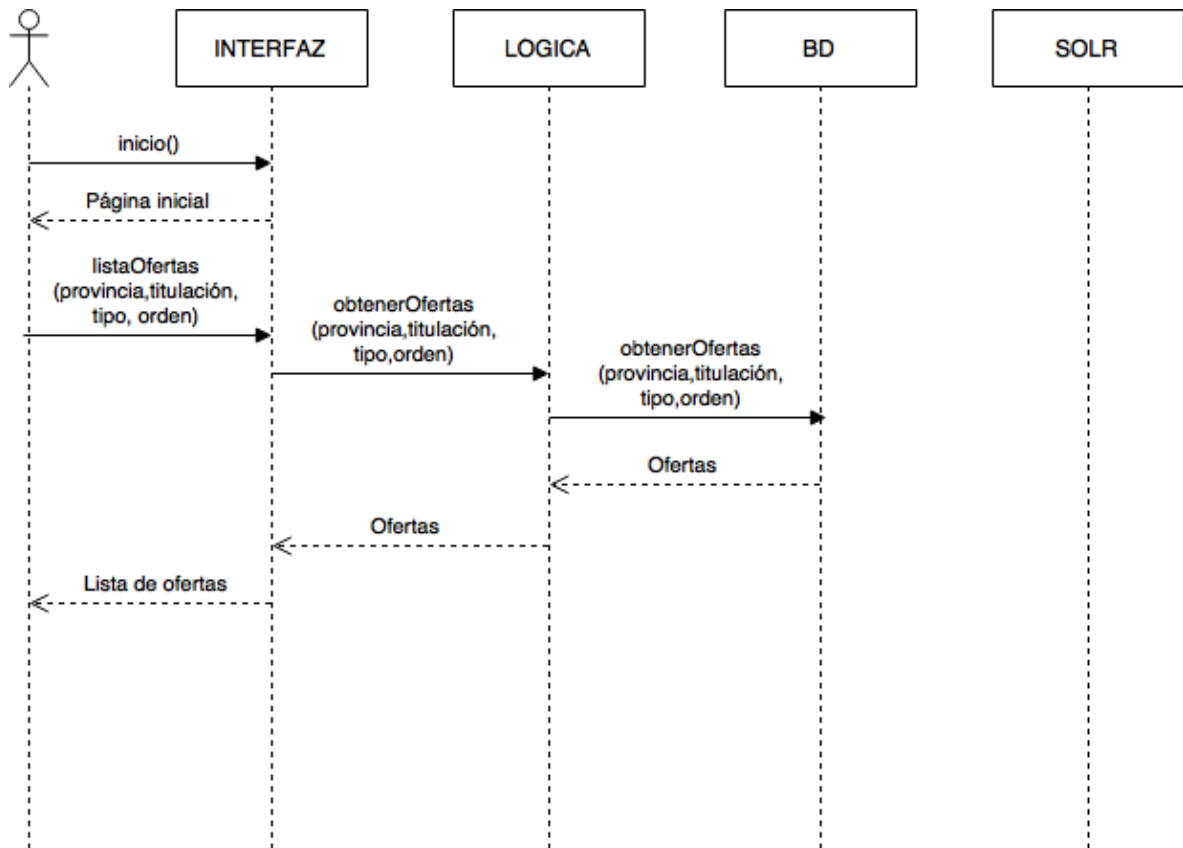


Figura A.8: Diagrama de secuencia: Listar ofertas

Caso de uso	Publicar opinión
Nº de CU	9
Actores	Usuario universitario
Propósito	Publicar una nueva opinión laboral sobre una organización concreta.
Descripción breve	Un usuario identificado como universitario escribe una opinión laboral sobre una organización concreta. El sistema valida esta opinión, calcula su puntuación y la publica.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al formulario de publicar opinión. 5 Completa la información del formulario.	2 Muestra el contenido del sitio web. 4 Muestra el formulario de publicación de opinión. 6 Valida la información dada. 7 Calcula la puntuación de la opinión. 8 Publica la opinión. 9 Actualiza la puntuación de la organización. 10 Envía notificación a la organización.
Cursos de eventos alternativos	
Línea 6: Si la opinión contiene palabras inadecuadas, se censuran y se continua con el proceso. Línea 6: Si la opinión no es apropiada, se muestra un error y se cancela el proceso. Línea 9: Si la organización aun no existe en el sistema, se crea.	

Cuadro A.9: Caso de uso expandido: Publicar opinión

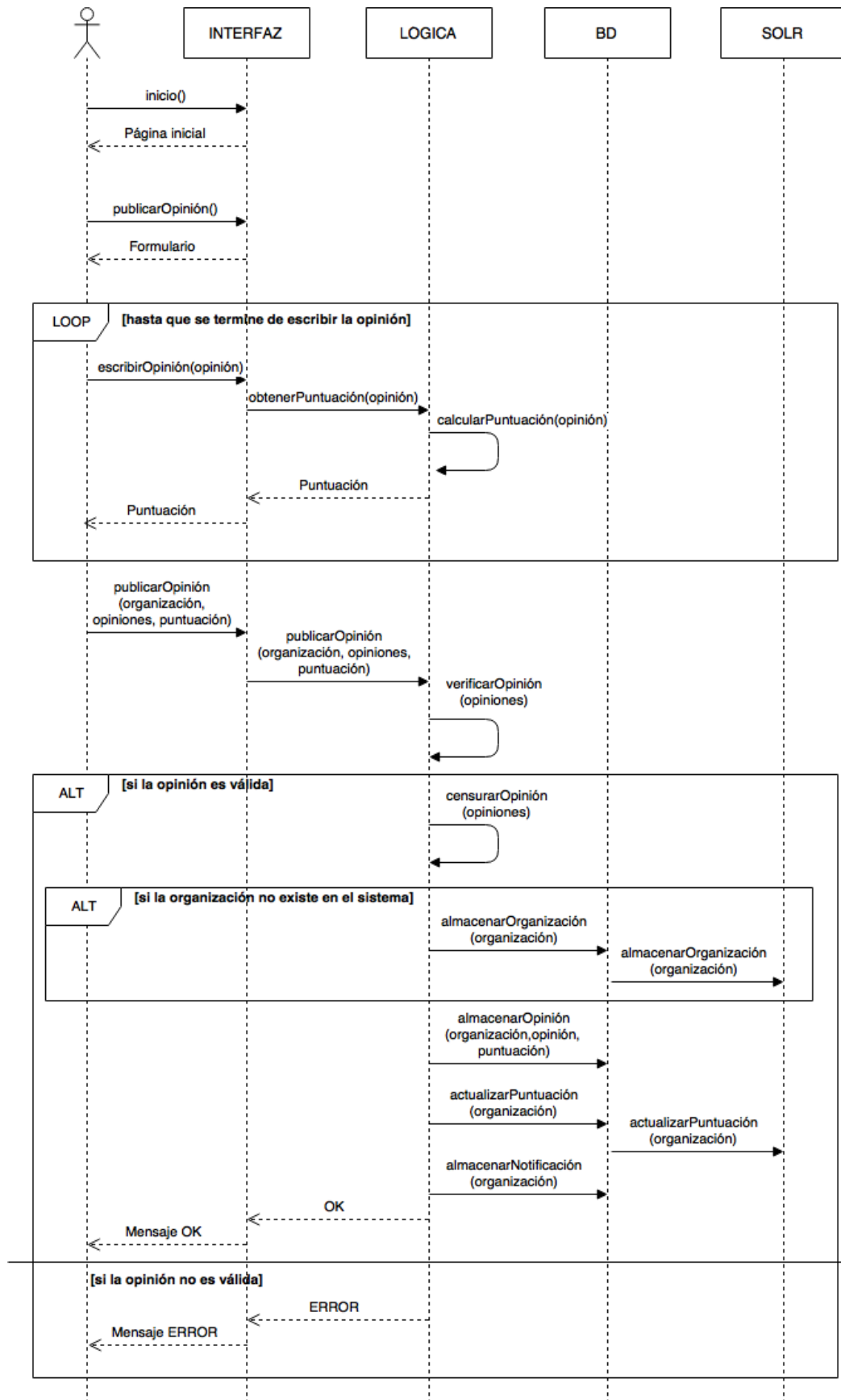


Figura A.9: Diagrama de secuencia: Publicar opinión

Caso de uso	Modificar información de usuario
Nº de CU	10
Actores	Usuario universitario, organización o administrador
Propósito	Editar la información vinculada a una cuenta de usuario.
Descripción breve	Un usuario identificado como universitario, organización o administrador modifica los datos de su cuenta de usuario. El sistema valida esta información y la almacena.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al formulario de “Ajustes de usuario”. 5 Completa la información del formulario.	2 Muestra el contenido del sitio web. 4 Muestra el formulario de “Ajustes de usuario”. 6 Valida la información dada. 7 Almacena la nueva información del usuario.
Cursos de eventos alternativos	
Línea 6: Si la información no es válida, muestra un error y se cancela el proceso.	

Cuadro A.10: Caso de uso expandido: Modificar información de usuario

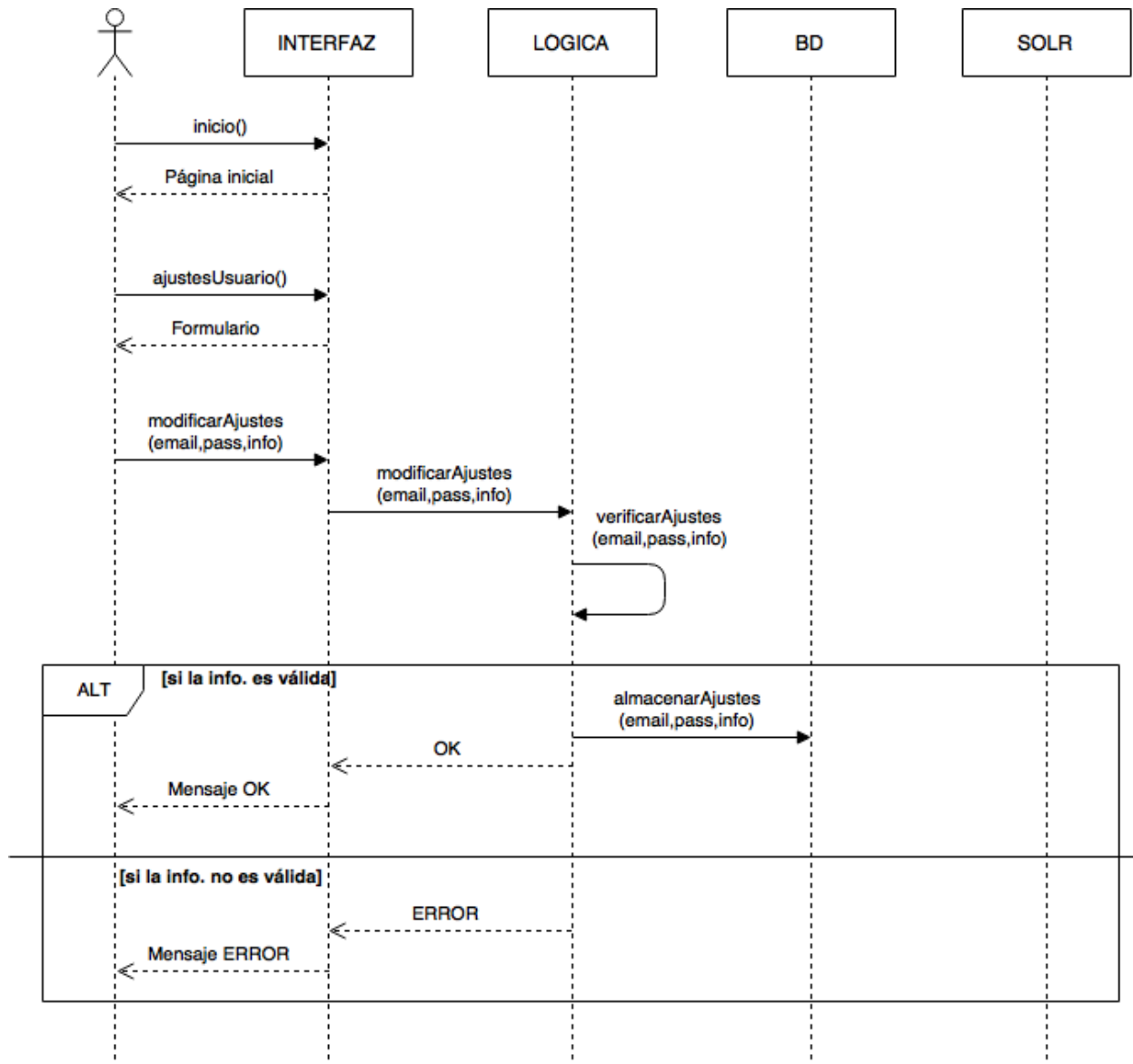


Figura A.10: Diagrama de secuencia: Modificar información de usuario

Caso de uso	Ver novedades
Nº de CU	11
Actores	Usuario universitario u organización
Propósito	Obtener una lista con las novedades.
Descripción breve	Un usuario identificado como universitario u organización solicita las novedades ocurridas desde el último acceso al sitio. El sistema recupera y devuelve la lista de novedades.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al panel de novedades.	2 Muestra el contenido del sitio web. 4 Muestra el panel con las novedades del usuario.
Cursos de eventos alternativos	

Cuadro A.11: Caso de uso expandido: Ver novedades

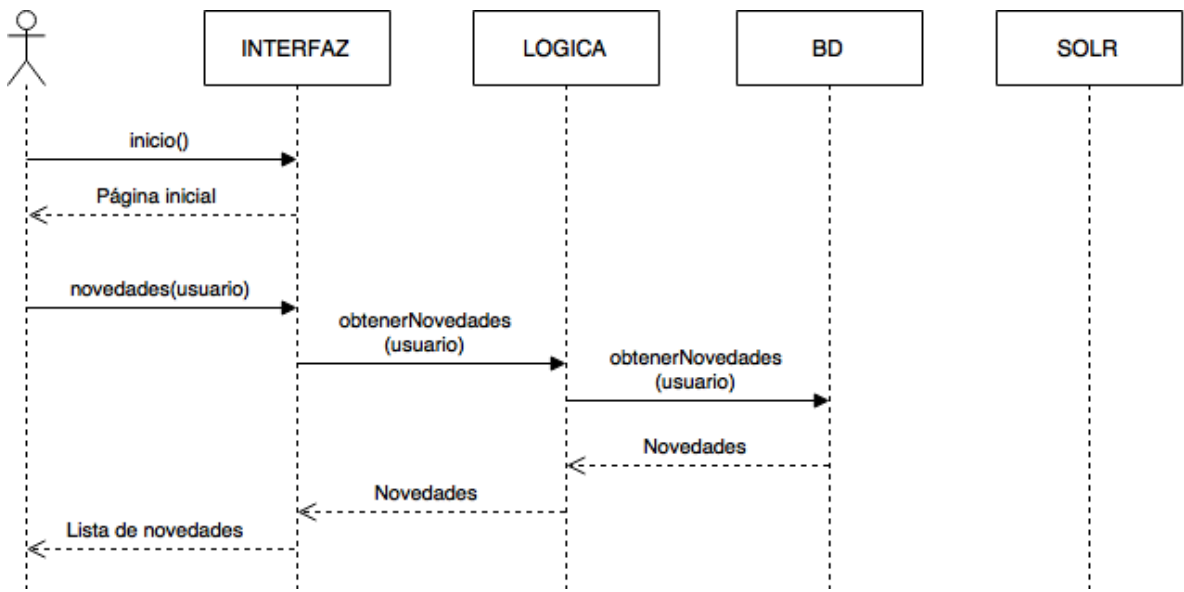


Figura A.11: Diagrama de secuencia: Ver novedades

Caso de uso	Ver opiniones sobre mi organización
Nº de CU	12
Actores	Usuario organización
Propósito	Listar las opiniones almacenadas para la organización del usuario.
Descripción breve	Un usuario identificado como organización solicita la lista de opiniones publicadas sobre dicha organización. Posteriormente aplica filtros y reordenaciones para ajustar el resultado. El sistema procesa la petición y muestra las opiniones.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede a la página inicial del sitio web. 3 Aplica un filtro al listado de opiniones. 5 Aplica una reordenación al listado de opiniones.	2 Muestra el contenido del sitio, junto con las opiniones almacenadas para la organización. 4 Muestra las opiniones que cumplen los criterios dados. 6 Muestra las opiniones que cumplen los criterios dados, ordenadas según la característica indicada.
Cursos de eventos alternativos	

Cuadro A.12: Caso de uso expandido: Ver opiniones sobre mi organización

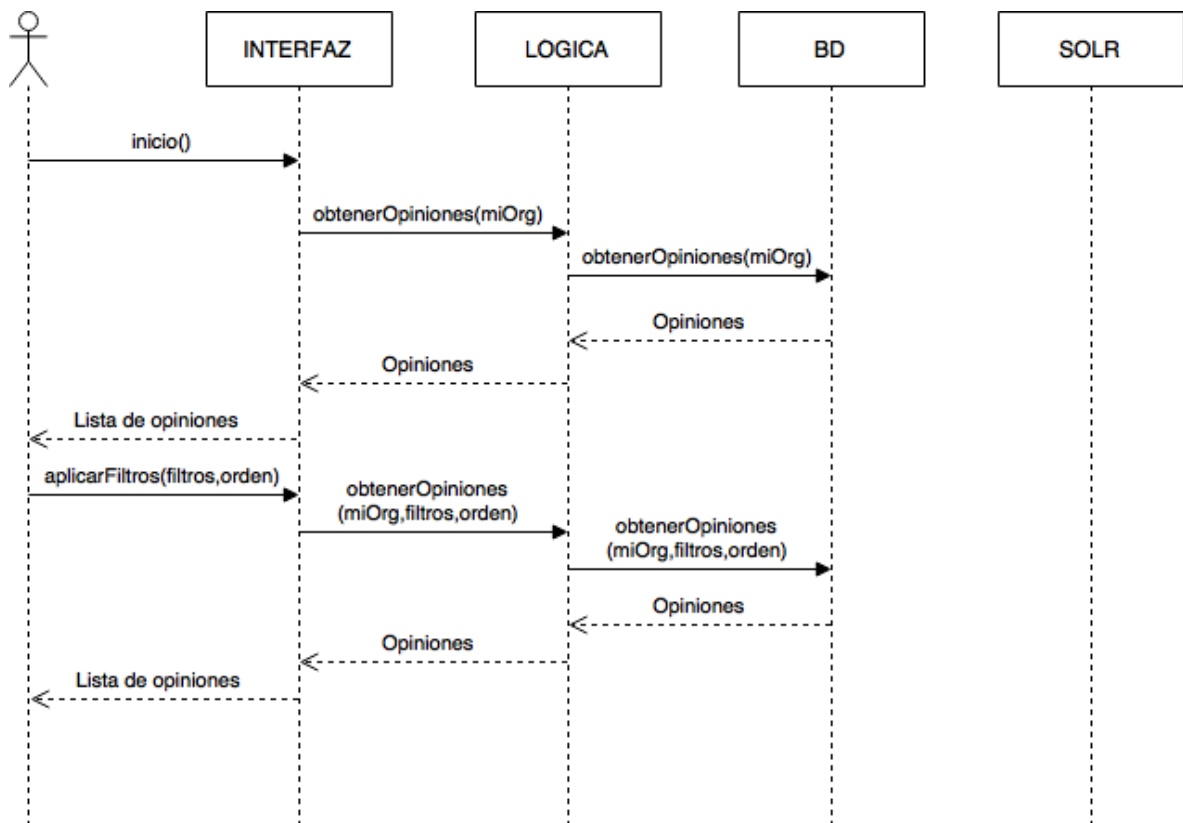


Figura A.12: Diagrama de secuencia: Ver opiniones sobre mi organización

Caso de uso	Publicar oferta
Nº de CU	13
Actores	Usuario organización
Propósito	Publicar una nueva oferta de empleo, tfg o prácticas.
Descripción breve	Un usuario identificado como organización accede al formulario de publicación de ofertas, rellena la información necesaria y publica una nueva oferta. El sistema procesa esta información, almacena la oferta y mejora la valoración de la organización ofertante.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al formulario de “nueva oferta”. 5 Completa la información del formulario de publicación.	2 Muestra el contenido del sitio web. 4 Muestra el formulario de publicación de ofertas. 6 Valida la información dada. 7 Publica la nueva oferta. 8 Mejora la valoración de la organización ofertante. 9 Notifica el evento a los usuarios universitarios interesados.
Cursos de eventos alternativos	
Línea 6: Si la información no es válida, muestra un error y se cancela el proceso.	

Cuadro A.13: Caso de uso expandido: Publicar oferta

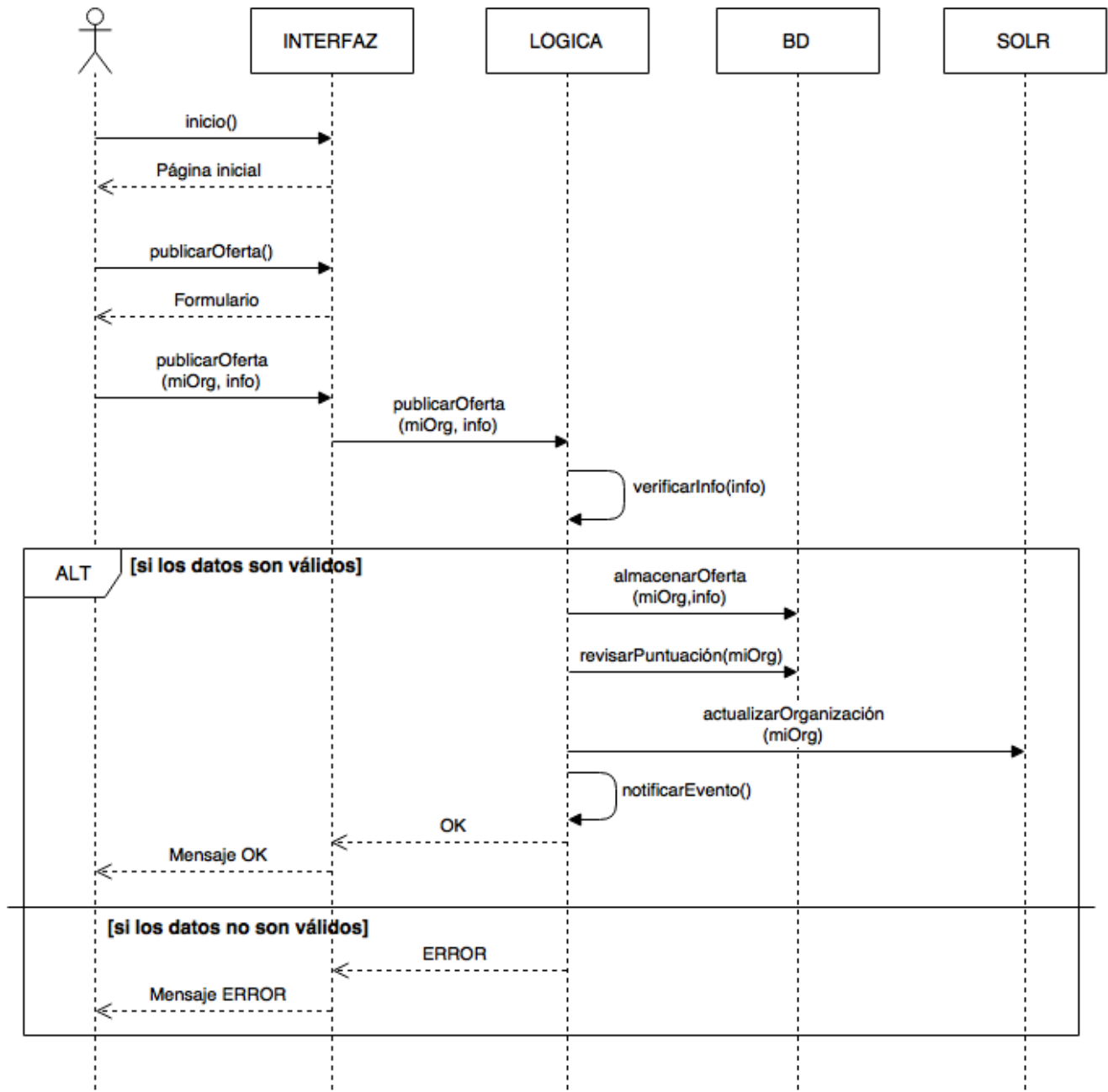


Figura A.13: Diagrama de secuencia: Publicar oferta

Caso de uso	Editar oferta
Nº de CU	14
Actores	Usuario organización
Propósito	Editar los datos de una de las ofertas publicadas por la organización del usuario.
Descripción breve	Un usuario identificado como organización obtiene la lista de sus ofertas, elige una de ellas y edita los datos que la forman. El sistema procesa y almacena la nueva información.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al listado de ofertas de la organización. 5 Solicita la edición de una oferta concreta. 7 Completa los datos del formulario de edición de oferta.	2 Muestra el contenido del sitio web. 4 Muestra el listado de ofertas de la organización. 6 Muestra el formulario de edición de oferta. 8 Valida la información dada. 9 Actualiza la oferta. 10 Revisa la valoración de la organización ofertante. 11 Notifica el evento a los usuarios universitarios interesados.
Cursos de eventos alternativos	
Línea 8: Si la información no es válida, muestra un error y se cancela el proceso.	

Cuadro A.14: Caso de uso expandido: Editar oferta

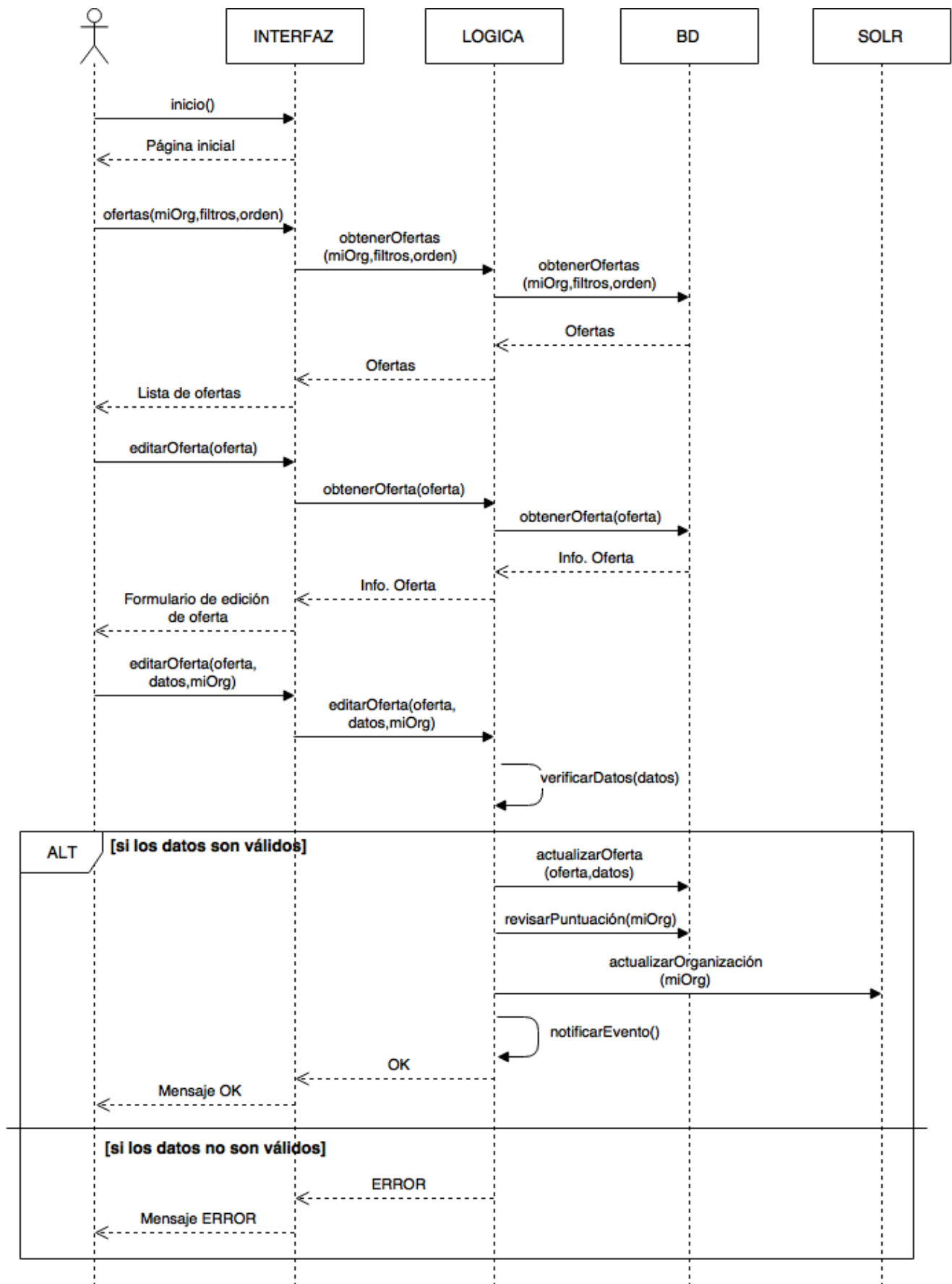


Figura A.14: Diagrama de secuencia: Editar oferta

Caso de uso	Ver ofertas de mi organización
Nº de CU	15
Actores	Usuario organización
Propósito	Listar las ofertas que ha publicado la organización del usuario.
Descripción breve	Un usuario identificado como organización solicita la lista de sus ofertas publicadas. Posteriormente aplica filtros y reordenaciones para ajustar el resultado. El sistema recupera las ofertas y las muestra al usuario.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al listado de ofertas de la organización (Botón “Ver ofertas”). 5 Aplica un filtro al listado de las ofertas. 7 Aplica una reordenación al listado de las ofertas.	2 Muestra el contenido del sitio web. 4 Muestra el listado de ofertas de la organización. 6 Muestra el listado de ofertas de la organización que cumplen las condiciones dadas. 8 Muestra el listado de ofertas de la organización que cumplen las condiciones dadas, ordenado según la característica indicada.
Cursos de eventos alternativos	

Cuadro A.15: Caso de uso expandido: Ver ofertas de mi organización

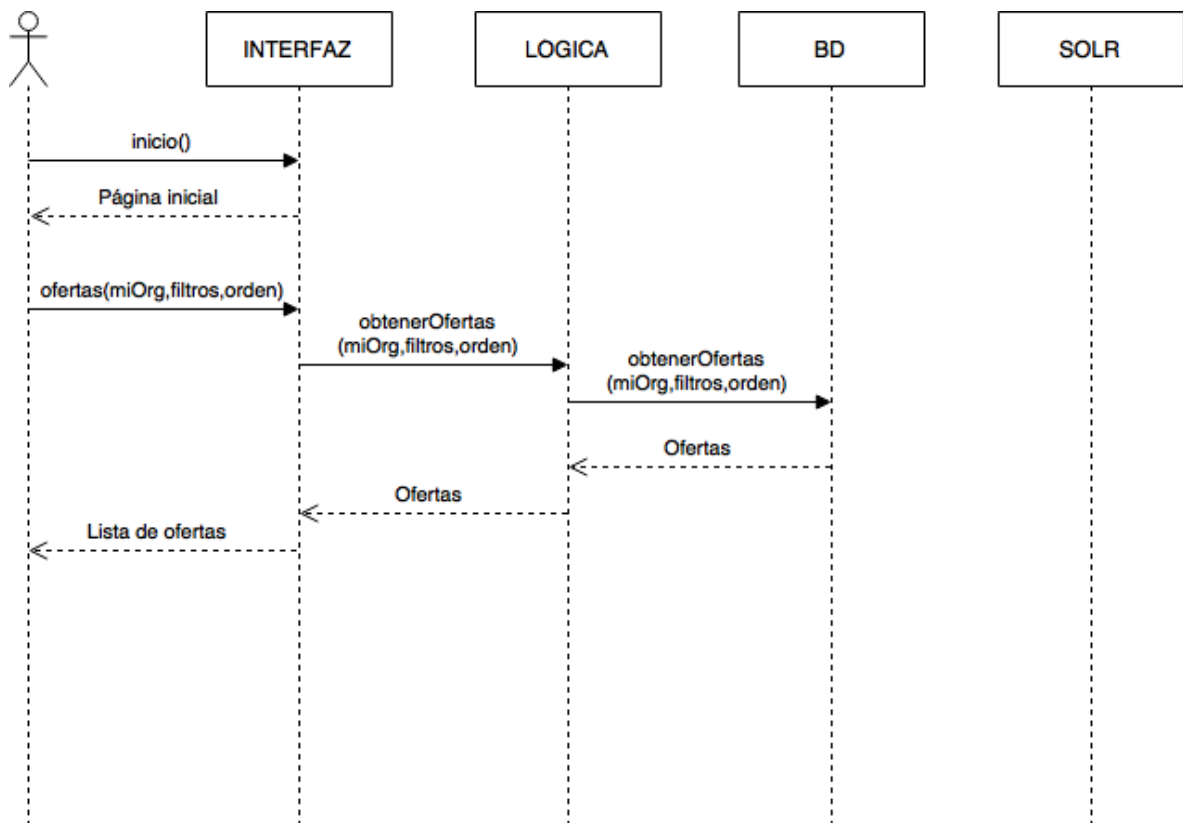


Figura A.15: Diagrama de secuencia: Ver ofertas de mi organización

Caso de uso	Activar organización
Nº de CU	16
Actores	Usuario administrador
Propósito	Activar la cuenta de un usuario registrado como organización.
Descripción breve	Un usuario identificado como administrador solicita la lista de usuarios registrados como organización que aun no han sido activados y procede a la activación de uno de ellos. El sistema devuelve la lista de usuarios sin activar y procesa la activación solicitada, notificando por email al usuario activado.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Accede al listado de usuarios pendientes de activar. 5 Solicita la activación de un usuario concreto.	2 Muestra el contenido del sitio web. 4 Muestra el listado de usuarios pendientes de activar. 6 Activa el usuario indicado. 7 Asocia el usuario con la organización correspondiente. 8 Envía una notificación al usuario activado.
Cursos de eventos alternativos	
Línea 5: El usuario rechaza un usuario concreto. El sistema procesa la solicitud, informa al usuario rechazado y finaliza el proceso.	

Cuadro A.16: Caso de uso expandido: Activar organización

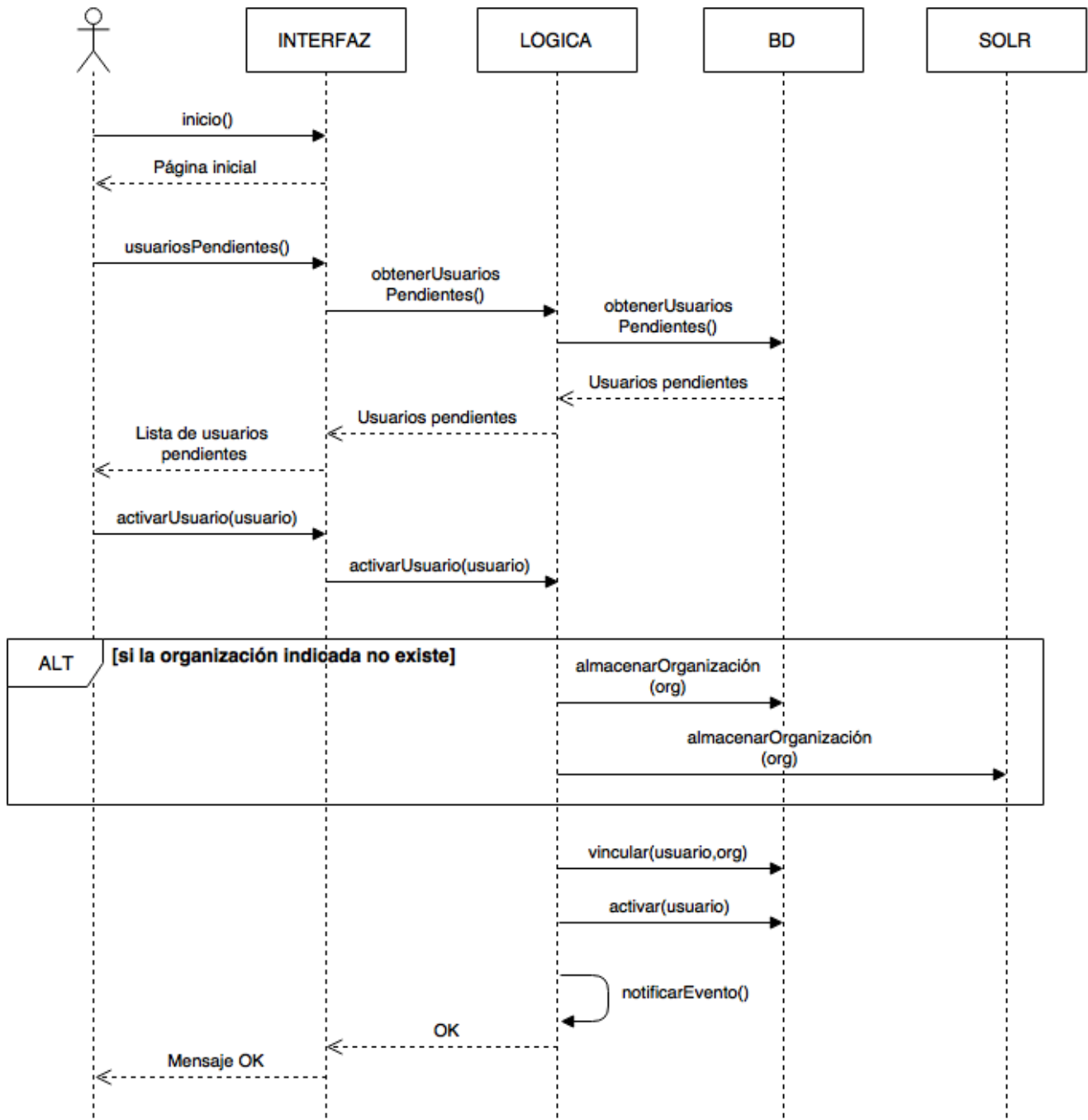


Figura A.16: Diagrama de secuencia: Activar organización

Caso de uso	Ver estadísticas
Nº de CU	17
Actores	Usuario administrador
Propósito	Ver estadísticas sobre usuarios, organizaciones, opiniones y ofertas.
Descripción breve	Un usuario identificado como administrador solicita estadísticas de uso del sitio web. El sistema obtiene dichas estadísticas y las muestra al usuario.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede a la página inicial del sitio web.	2 Muestra el contenido del sitio web, junto con las estadísticas de uso del sitio.
Cursos de eventos alternativos	

Cuadro A.17: Caso de uso expandido: Ver estadísticas

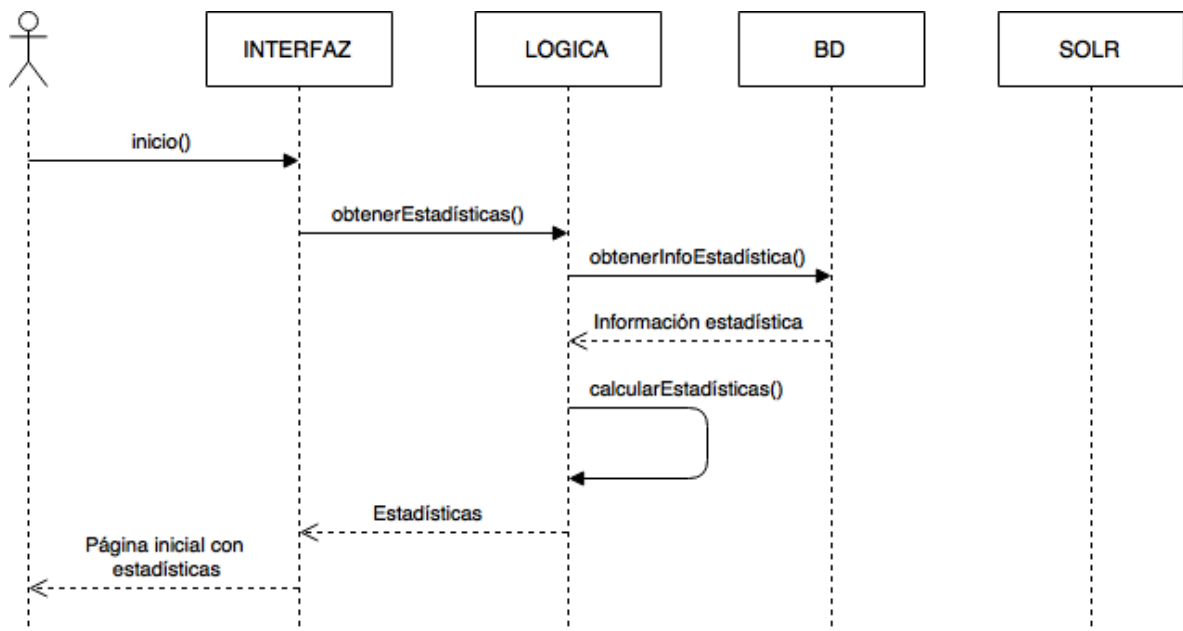


Figura A.17: Diagrama de secuencia: Ver estadísticas

Caso de uso	Realizar sorteo
Nº de CU	18
Actores	Usuario administrador
Propósito	Calcular el ganador del sorteo para el último mes vencido.
Descripción breve	Un usuario identificado como administrador solicita la realización del sorteo de premios del último mes vencido. El sistema calcula el usuario ganador, le envía un correo electrónico e informa al administrador sobre el ganador.
Curso típico de eventos	
Acciones del actor	Respuesta del sistema
1 Accede al sitio web. 3 Solicita la realización del sorteo mensual.	2 Muestra el contenido del sitio web. 4 Calcula y muestra el ganador del sorteo. 5 Envía una notificación al usuario ganador.
Cursos de eventos alternativos	
Línea 4: Si ya hay un ganador para el mes del sorteo, se devuelve un error y finaliza la ejecución.	

Cuadro A.18: Caso de uso expandido: Realizar sorteo

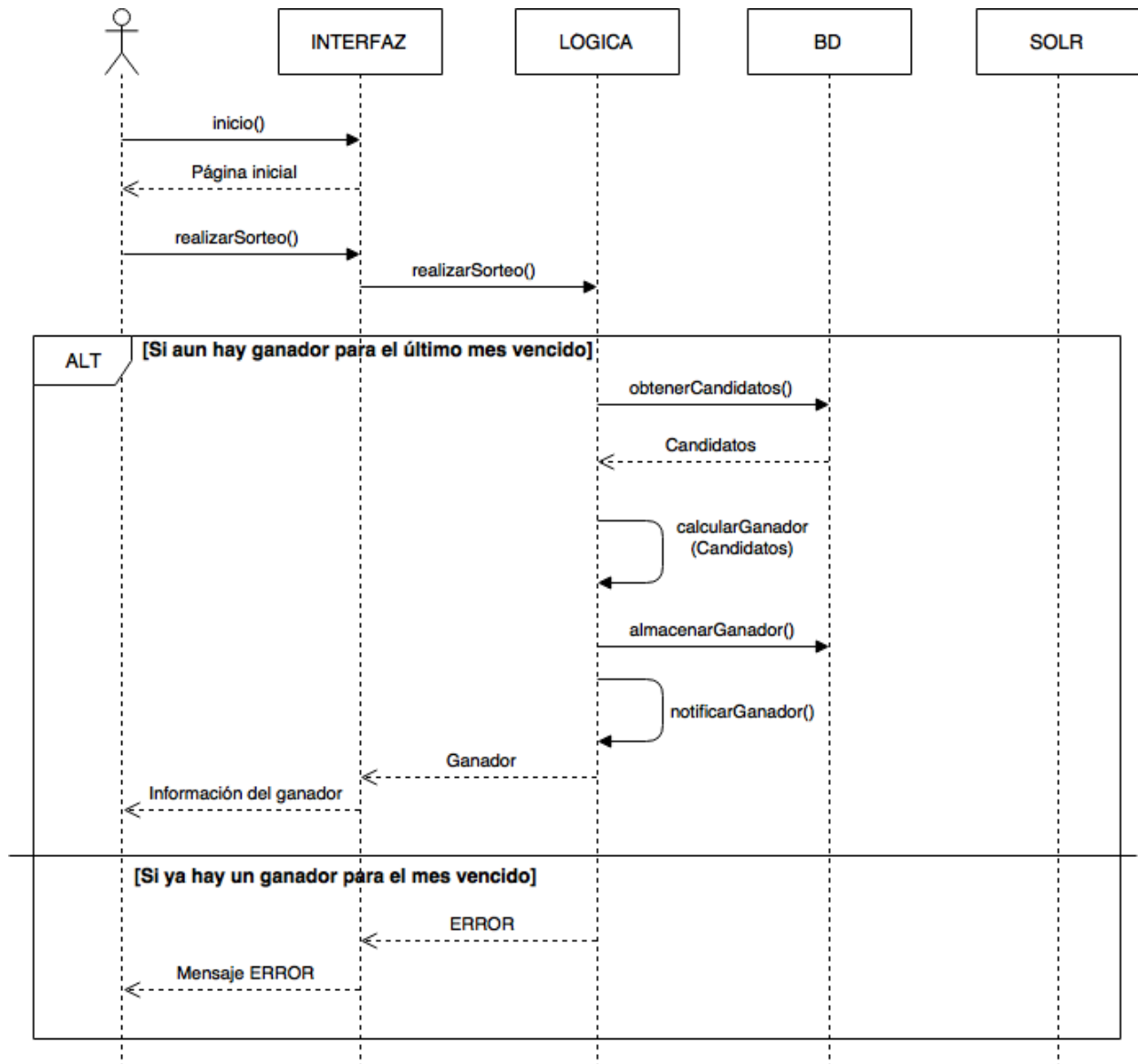


Figura A.18: Diagrama de secuencia: Realizar sorteo

Anexo B

Detalles de implementación

Este anexo extiende la información proporcionada en capítulos anteriores sobre la implementación del sistema, explicando en detalle aspectos como las técnicas y herramientas utilizadas en la implementación, así como las pruebas realizadas para verificar el correcto funcionamiento del sistema.

B.1. Detección de información en el proceso de scraping

El proceso de *scraping* permite la obtención de ofertas laborales de sitios web externos. En esta sección se explican las técnicas utilizadas para detectar información como el título, descripción, tipo, organización, provincia y titulación de cada una de las ofertas obtenidas.

Para determinar qué páginas se deben rastrear, qué elementos de dichas páginas se deben identificar como ofertas y qué partes de estos elementos corresponden al título, a la descripción, etc. se ha creado un fichero en formato JSON que engloba los patrones HTML que identifican cada tipo de dato.

```
{
  "urls": [
    {
      "url": "http://www.studentjob.es/ofertas?page={pagina}&search%5Bjob_types_id_equals%5D=9",
      "tipo": "2"
    }
  ],
  "id": "STUDENTJOB",
  "verificador": "li.active",
  "oferta": ".list-group-item",
  "provincia": "dd[itemprop=jobLocation]",
  "organizacion": ".job-opening-name h4",
  "titulacion": "h2:contains(Área de estudio preferida) + p",
  "titulo": "h1",
  "descripcion": ".job-opening-description[itemprop=description]"
}
```

Figura B.1: Patrones HTML para el sitio “Student Job”

Tanto el título como la descripción de cada oferta obtenida no requieren de ningún tratamiento especial, ya que se almacenan como texto plano. Sin embargo, se requieren mecanismos capaces de asociar la provincia, titulación y organización de las ofertas con las provincias, titulaciones y organizaciones ya existentes en la base de datos.

Reconocimiento de provincia

Para llevar a cabo el reconocimiento de provincias se construye una tabla *hash* cuya clave corresponde a la palabra más larga del nombre de cada provincia, y cuyo valor es el identificador que dichas provincias poseen en el sistema.

Una vez construida esta tabla, el reconocimiento de provincias se realiza comprobando si alguna de las palabras que forman el texto encontrado existe en la tabla.

Reconocimiento de titulación

Para llevar a cabo el reconocimiento de titulaciones se construye una tabla *hash* cuya clave corresponde a la *keyword* asociada a cada una de las titulaciones en la base de datos, y cuyo valor es el identificador que dichas titulaciones posee en el sistema.

Una vez construida esta tabla, el reconocimiento de titulaciones se realiza comprobando si alguna de las palabras que forman el texto encontrado existe en la tabla.

Reconocimiento de organización

Para llevar a cabo el reconocimiento de organizaciones se construye una tabla *hash* cuya clave corresponde al nombre de cada organización, y cuyo valor es el identificador que dichas organizaciones poseen en el sistema. Una vez construida esta tabla, el reconocimiento de organizaciones se realiza comprobando si el conjunto de palabras que forman el texto encontrado existe en la tabla.

Como se ha explicado, la información referente a las provincias, titulaciones y organizaciones existentes en la base de datos se mapea en memoria usando estructuras de datos de tipo *hash* que permiten el acceso a los datos en un tiempo constante $O(1)$. Estas estructuras se construyen al comienzo del proceso de *scraping* y se mantienen en memoria hasta la conclusión del mismo.

Para mejorar la capacidad de detección se aplica un tratamiento de texto tanto a las palabras con las que se forman las tablas *hash*, como a las palabras obtenidas de las ofertas encontradas. En el cuadro B.1 se resumen los métodos de tratamiento de texto utilizados en cada caso.

	Fuente	Normalización	Stemming	Eliminar stop-words
Provincia	Palabra más larga del nombre de cada provincia	Sí	No	Sí
Titulación	Palabras clave de cada titulación	Sí	Sí	Sí
Organización	Palabras del nombre de cada organización	Sí	No	Sí

Cuadro B.1: Reconocimiento de información en scraping

B.2. Paquetes del sistema

El código implementado del sistema se reparte entre diferentes clases *Java* que se agrupan en paquetes. En esta sección se listan dichos paquetes y se explican sus funcionalidades.

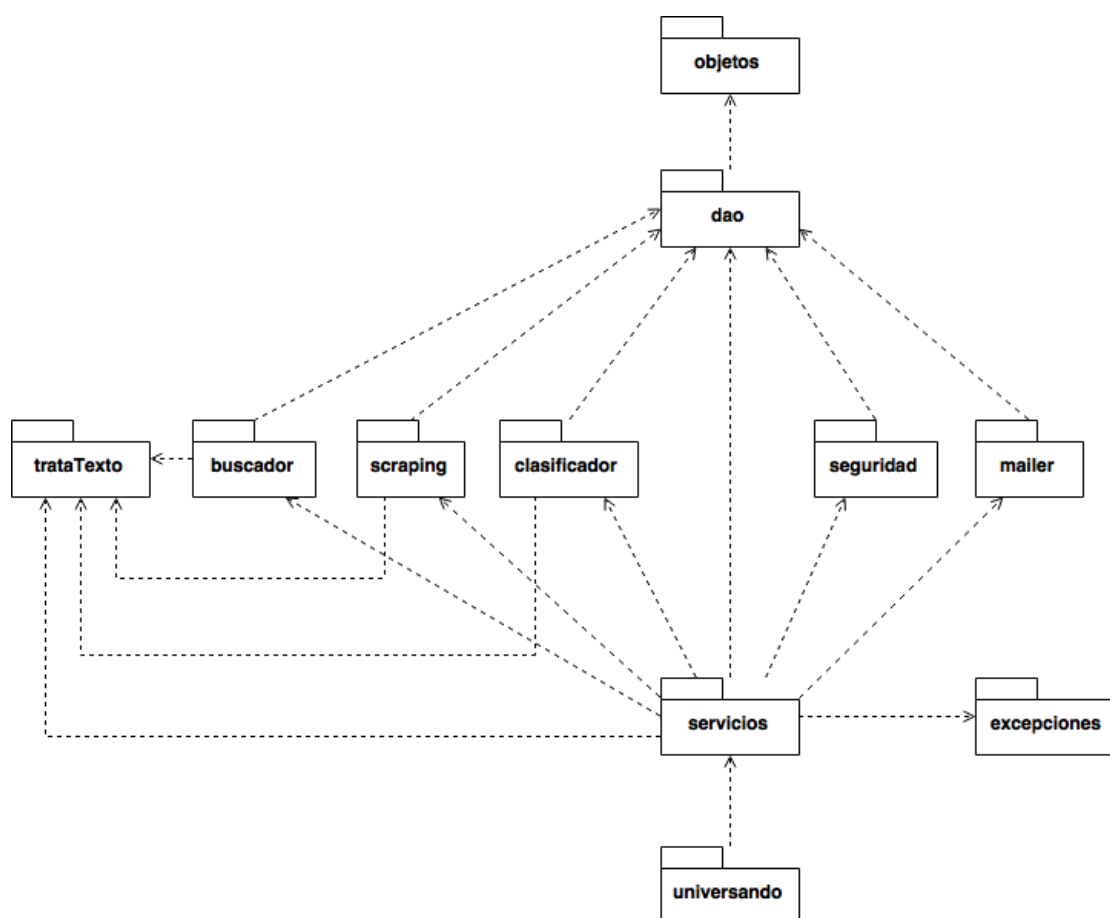


Figura B.2: Diagrama de paquetes

En la figura B.2 se muestra un diagrama que permite conocer los diferentes paquetes que componen el sistema y las relaciones de uso entre ellos. Estos paquetes son estructuras que engloban clases con características comunes. A continuación se explican las características de cada paquete del sistema:

1. **buscador:** Paquete encargado de la interacción con el repositorio de organizaciones gestionado por *Apache Solr*.
2. **clasificador:** Paquete encargado del entrenamiento del sistema de aprendizaje automático y de la clasificación de opiniones, incluyendo la interacción con la base de datos *MongoDB*.
3. **dao:** Paquete que compone una capa de interacción con la base de datos relacional *MySQL*. Este paquete engloba todas las operaciones que el sistema realiza sobre la base de datos.
4. **excepciones:** Paquete que incluye tipos de excepciones personalizados, creados para una correcta gestión de errores en los servicios web implementados.
5. **mailer:** Paquete encargado de la elaboración y envío de correos electrónicos a los usuarios.
6. **objetos:** Paquete que incluye diferentes tipos de objeto sobre los que se mapea la información obtenida de la base de datos *MySQL*, con el fin de facilitar el tratamiento de dicha información.
7. **scraping:** Paquete que proporciona los mecanismos necesarios para llevar a cabo el proceso de *scraping* que obtiene ofertas laborales de sitios web externos.
8. **seguridad:** Paquete que compone la capa de seguridad del sistema, la cual permite gestionar los diferentes roles de usuario y la identificación de los mismos.
9. **servicios:** Paquete que incluye todos los servicios web implementados a través de los que se puede acceder a la lógica de negocio.
10. **trataTexto:** Paquete encargado del procesamiento de información en lenguaje natural.
11. **universando:** Paquete principal que incluye el punto inicial de ejecución del sistema, además de otros elementos destinados a la planificación de tareas y a la configuración del entorno del sistema.

B.3. Pruebas detalladas

Para garantizar el correcto funcionamiento del sistema es necesario aplicar una serie de *tests*. En este capítulo se explican las pruebas realizadas más importantes que han servido para probar la corrección de las funcionalidades del sistema.

Pruebas unitarias

Las pruebas unitarias permiten verificar la corrección de módulos concretos de código. Estas pruebas se han elaborado durante el proceso de implementación del sistema, utilizando la tecnología *JUnit*. Su

ejecución se ha gestionado con el entorno de integración continua *Travis-CI* que permite su lanzamiento automático cada vez que se detecta una nueva versión del sistema.

En el cuadro B.2 se muestra el número de pruebas unitarias elaboradas para verificar el correcto funcionamiento de las diferentes partes del sistema. Cada una de estas pruebas engloba varios casos a verificar, de modo que si uno de ellos no devuelve el resultado esperado, el *test* será erróneo.

Clase	Paquete	Nº de pruebas unitarias
DaoAspirante	dao	2
DaoCv	dao	4
DaoEstadisticas	dao	1
DaoGanador	dao	3
DaoNovedad	dao	4
DaoOferta	dao	4
DaoOpinion	dao	6
DaoOrganizacion	dao	7
DaoProvincia	dao	3
DaoTitulacion	dao	3
DaoUniversidad	dao	4
DaoUsuario	dao	14
MailManager	mailer	6
RepositorioSolrImp	buscador	3
Entrenamiento	clasificador	1
Clasificador	clasificador	1
MongoDao	clasificador	9
DetectorPalabras	trataTexto	1
DetectorStop	trataTexto	1
TratamientoTexto	trataTexto	12
Scraper	scrapping	3
Otros	universando	1

Cuadro B.2: Pruebas unitarias realizadas

Pruebas de integración

Las pruebas de integración permiten verificar la corrección de varios módulos concretos de código ejecutados de forma secuencial. En este proyecto se ha utilizado la tecnología *JUnit* y el *framework Spring* para elaborar pruebas de integración capaces de comprobar la corrección de los servicios web desarrollados. Su ejecución, al igual que la de las pruebas unitarias, se ha gestionado con el entorno

de integración continua *Travis-CI* que permite su lanzamiento automático cada vez que se detecta una nueva versión del sistema.

En el cuadro B.3 se muestra el número de pruebas de integración elaboradas para verificar el correcto funcionamiento de las operaciones de la API RESTful implementada. Cada una de estas pruebas engloba varios casos a verificar, de modo que si uno de ellos no devuelve el resultado esperado, el *test* será erróneo.

Objeto (API)	Nº de pruebas de integración
Cv	2
Estadísticas	1
Ganador	1
Novedad	5
Oferta	6
Opinión	7
Organización	6
Provincia	3
Titulación	4
Universidad	4
Usuario	10

Cuadro B.3: Pruebas de integración realizadas

Pruebas de sistema

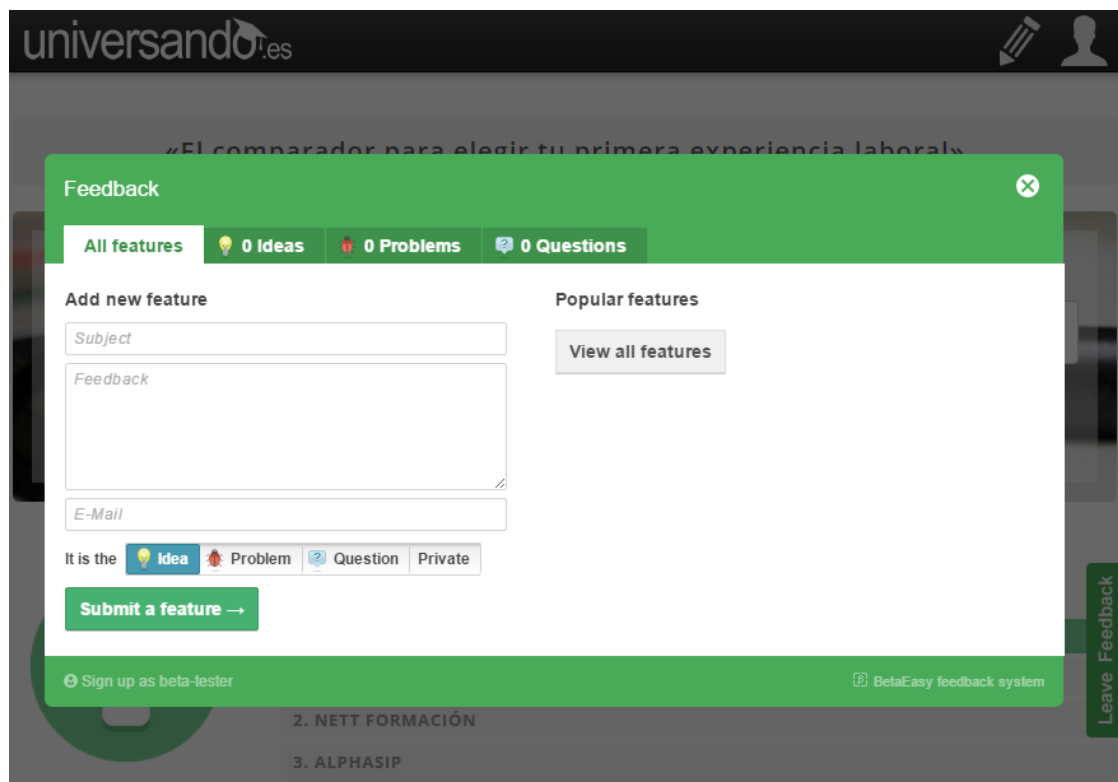
Las pruebas del sistema consisten en utilizar la aplicación web para probar casos de uso concretos que podrían realizar los usuarios reales del sistema. Los casos de uso probados se detallan en el Anexo A y verifican las siguientes acciones:

1. Ver mejores organizaciones
2. Listar organizaciones
3. Buscar organizaciones
4. Registrar usuario
5. Identificar usuario
6. Ver opiniones de organización
7. Ver ofertas de organización
8. Listar ofertas
9. Publicar opinión
10. Modificar información de usuario

11. Ver novedades
12. Ver opiniones sobre mi organización
13. Publicar oferta
14. Editar oferta
15. Ver ofertas de mi organización
16. Activar organización
17. Ver estadísticas
18. Realizar sorteo

Pruebas beta

Una vez finalizada la implementación del sistema, se han incluido los mecanismos necesarios para que los primeros usuarios reales puedan enviar su *feedback* sobre el funcionamiento de la aplicación web. Para ello se ha utilizado el servicio proporcionado por *BetaEasy* que permite incluir una caja de envío de opiniones como la mostrada en la figura B.3.



The image shows a screenshot of a web application interface for 'universando.es'. A green feedback modal is open, titled 'Feedback'. It features a navigation bar with 'All features' (selected), '0 Ideas', '0 Problems', and '0 Questions'. The main area is divided into 'Add new feature' and 'Popular features'. The 'Add new feature' section includes input fields for 'Subject', 'Feedback', and 'E-Mail', along with radio buttons for 'Idea', 'Problem', 'Question', and 'Private'. A green 'Submit a feature →' button is at the bottom. The 'Popular features' section has a 'View all features' button. At the bottom of the modal, there is a green bar with 'Sign up as beta-tester' and 'BetaEasy feedback system' links. A vertical 'Leave Feedback' button is on the right side of the modal.

Figura B.3: Caja de envío de feedback

Pruebas cross-browser

Se ha dado una gran importancia a que el sitio web se visualice correctamente desde cualquier tipo de navegador y dispositivo. Por ello, se han realizado pruebas manuales utilizando los siguientes navegadores:

1. Google Chrome v40 y Google Chrome (Android)
2. Mozilla Firefox v34
3. Internet Explorer v11
4. Opera v27
5. Microsoft Edge v20

Pruebas de carga del servidor

Para comprobar la resistencia del servidor que aloja el sistema ante una situación de elevada carga se ha utilizado el servicio proporcionado por *LoadImpact*. Este servicio realiza varias peticiones simultaneas al servidor para analizar su respuesta en situaciones de estrés.

La figura B.4 muestra el resultado de este *test*, donde se puede observar que para un número creciente de peticiones (línea azul), el tiempo de carga se mantiene aproximadamente constante (línea verde).

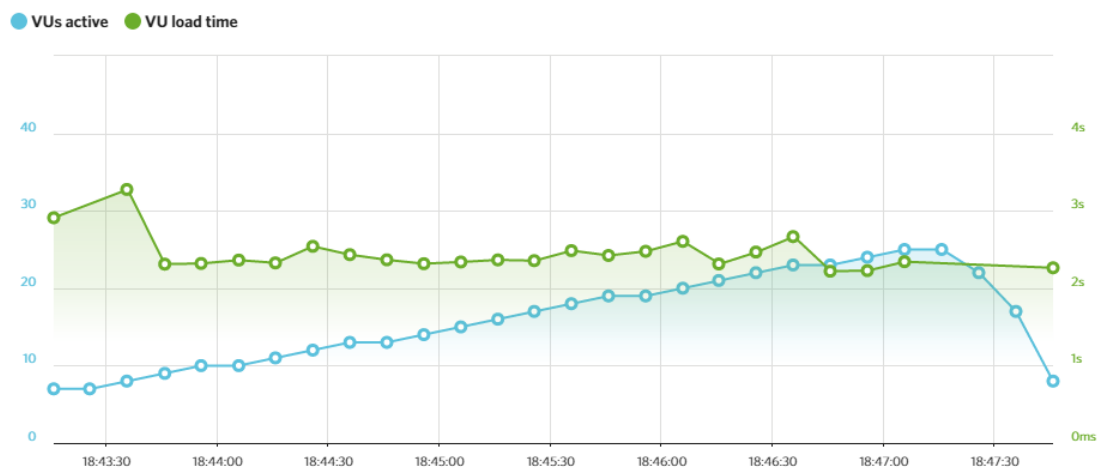


Figura B.4: Prueba de carga del servidor

B.4. Técnicas SEO utilizadas

Como ya se ha explicado en la memoria, el sistema cuenta con una aplicación web que realiza peticiones *Ajax* a la API de servicios implementada. El principal problema con este tipo de aplicaciones es que los buscadores como *Google* no son capaces de indexar el contenido que se recupera mediante dichas peticiones *Ajax*.

A continuación se explican las medidas adoptadas para solucionar el problema descrito y para mejorar el posicionamiento del sitio web en los buscadores.

Snapshots HTML

Para permitir que buscadores como *Google* puedan indexar el contenido del sitio web, se han creado una serie de *snapshots* que poseen contenido HTML estático, es decir no obtienen su contenido a través de peticiones *Ajax*. A continuación se listan los tipos de *snapshots* creados, que solo son devueltos cuando los buscadores acceden al sitio para realizar su proceso de *crawling*, por lo que nunca serán visibles para los usuarios:

1. Página principal
2. Listado de organizaciones para cada provincia
3. Listado de ofertas para cada titulación

Finalmente, las URLs de todos los *snapshots* creados (que serán las mismas que las URLs de las páginas habituales del sitio) se incluyen en un fichero *Sitemap*, con el fin de que los buscadores puedan conocer que páginas deben recorrer.

URLs semánticas

Se ha adaptado el sitio web para permitir la navegación mediante URLs “SEO friendly”, de modo que en las propias URLs se indica de forma semántica el tipo de contenido al que enlazan. Para implementar esta característica se ha utilizado la librería *URL Rewrite Filter* que permite convertir las peticiones HTTP realizadas a una URL semántica concreta en peticiones equivalentes a una URL común.

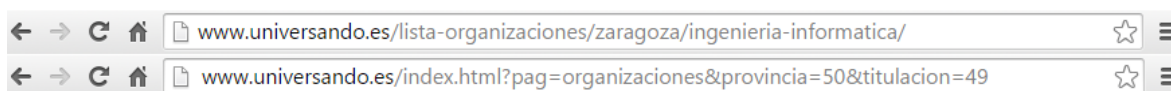


Figura B.5: URL semántica (arriba) vs. URL común (abajo)

En el cuadro B.4 se muestran los diferentes tipos de URL semántica permitidos junto con sus equivalentes no semánticos.

URL Semántica	URL Equivalente
/escritor/...	/index.html?pag=escritor...
/organizacion/?org={organizacion}...	/index.html?pag=organizacion &org={organizacion}...
/lista-ofertas/{provincia}/{titulacion} /{tipo}/...	/index.html?pag=ofertas&provincia= {provincia}&titulacion={titulacion}&tipo= {tipo}...
/lista-organizaciones/{provincia} /{titulacion}/...	/index.html?pag=organizaciones&provincia= {provincia}&titulacion={titulacion}...
/detalles/?organizacion={organizacion}	/detalles.html?organizacion={organizacion}
/terminos/	/terminos.html
/contacta/	/contacta.html

Cuadro B.4: Equivalencias de URLs semánticas

Etiquetas HTML apropiadas

El correcto uso de las etiquetas HTML esta valorado positivamente por los buscadores, por lo que el sitio web utiliza etiquetas como *meta-description*, *title*, *h1*, *h2*, etc. de la forma más correcta y coherente.

B.5. Librerías utilizadas

Además del *framework Spring*, se han utilizado una serie de librerías para facilitar el desarrollo del sistema. En este anexo se detallan las librerías utilizadas tanto para implementar la lógica del sistema como para crear la aplicación web.

Lógica del sistema

1. **JavaMail:** Librería requerida por *Spring* para realizar envíos de correos electrónicos.
2. **Apache Velocity:** Librería que permite el uso de la tecnología *Apache Velocity* para utilizar plantillas en el envío de correos electrónicos.
3. **MaxMind GeoIP:** Librería que permite la geolocalización de los usuarios a partir de su IP.
4. **Cybozu Language Detection:** Librería que permite determinar el idioma de un texto dado.
5. **Quartz Scheduler:** Librería requerida por *Spring* para la planificación de tareas automáticas.
6. **Gson:** Librería que permite mapear objetos *Java* en formato *JSON*.
7. **Snowball Stemmer:** Librería que aplica el algoritmo *Snowball Stemmer* para obtener las raíces (lexemas) de las palabras de un texto.

8. **JSoup:** Librería pensada para facilitar las tareas de *scraping* como obtener el contenido de páginas web, procesarlo, etc.
9. **GreenMail:** Librería que proporciona mecanismos para verificar el correcto envío y recepción de correos electrónicos en la ejecución de pruebas unitarias y de integración.
10. **Swagger:** Librería que permite generar documentación de calidad sobre los servicios de la API RESTful implementada.

Aplicación web

1. **JQuery:** Librería basada en *Javascript* que facilita la implementación del comportamiento dinámico del sitio web y permite la realización de llamadas *Ajax*.
2. **JQuery Cookie:** Librería que extiende la funcionalidad de *JQuery* para permitir el control de *cookies*.
3. **Select2:** Librería que permite utilizar listas desplegables con un buscador integrado.
4. **Bootstrap:** *Framework* con plantillas y estilos prediseñados que facilitan la elaboración del diseño y la estructura del sitio web.
5. **Bootstrap DatePicker:** Librería que permite mostrar calendarios para cumplir la función de selección de fechas.
6. **Chart.js:** Librería que permite generar gráficos a partir de colecciones de datos.
7. **URL Rewrite Filter:** Librería para *Java* que permite modificar el direccionamiento de determinadas URLs, facilitando el uso de URLs semánticas.

Anexo C

Detalles de gestión

Este anexo expone algunos detalles obtenidos del proceso de gestión del proyecto. En la sección C.1 se explica el tiempo empleado en la realización del proyecto según diferentes criterios. En la sección C.2 se muestran los gráficos *burndown* que recogen el progreso del desarrollo a lo largo del tiempo. Y por último, en la sección C.3 se muestran los *user stories* obtenidos a partir de los requisitos y desde los que se obtienen las diferentes tareas a realizar.

C.1. Tiempo empleado

Durante el transcurso del proyecto se ha mantenido una contabilidad del tiempo empleado en la realización de cada tarea. Gracias a esto, se ha podido calcular el tiempo total dedicado a cada actividad.

El cuadro C.1 y la figura C.1 muestran las horas empleadas durante las diferentes fases del proyecto, considerando los siguientes tipos de tarea:

1. **Análisis y diseño:** Tareas dedicadas a la conceptualización del proyecto que tratan de determinar las funcionalidades, roles, tecnologías, infraestructura, modelo de datos, aspecto visual...
2. **Implementación:** Tareas dedicadas a programar las diferentes funcionalidades del sistema.
3. **Documentación:** Elaboración de la memoria del proyecto, así como de cualquier otro documento.
4. **Pruebas:** Diseño y ejecución de pruebas unitarias, pruebas de integración, pruebas del sistema...
5. **Aprendizaje:** Tareas dedicadas al aprendizaje de nuevos conceptos y herramientas.
6. **Otros:** Otras tareas como reuniones, puesta en producción, preparación de entorno de trabajo...

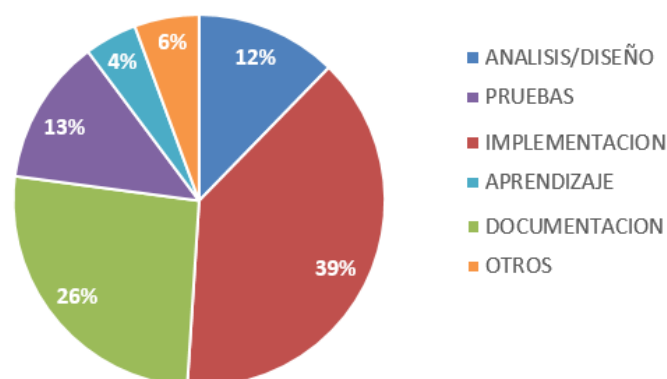


Figura C.1: Tiempo dedicado por tipo de tarea

	Fase Previa	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	Fase Final	Total
Análisis y dis.	24,0	46,8	0,3	1,0	-	-	3,2	-	75,3 horas
Implementación	-	-	36,5	44,5	26,7	61,7	67,6	-	237,0 horas
Documentación	-	-	13,0	0,3	-	7,5	-	139,0	159,8 horas
Pruebas	-	3,2	14,0	14,0	26,0	6,0	14,5	-	77,7 horas
Aprendizaje	4,5	8,0	1,5	-	4,5	-	9,5	-	28,0 horas
Otros	3,5	8,0	-	2,0	-	9,5	11,5	-	34,5 horas
Total	32,0 horas	66,0 horas	65,3 horas	61,8 horas	57,2 horas	84,7 horas	106,3 horas	139,0 horas	612,3 horas

Cuadro C.1: Tiempo dedicado por tipo de tarea y periodo

C.2. Gráficos burndown

A continuación se muestran los gráficos *burndown* utilizados como herramienta para medir el esfuerzo realizado frente al esfuerzo necesario para completar las iteraciones del desarrollo en el plazo estimado.

En estos gráficos, el eje horizontal representa el tiempo y el eje vertical representa el esfuerzo restante. Como se puede observar, la línea de color rojo simboliza el esfuerzo restante ideal para finalizar la iteración en el plazo estimado, mientras que la línea azul simboliza el esfuerzo restante real.

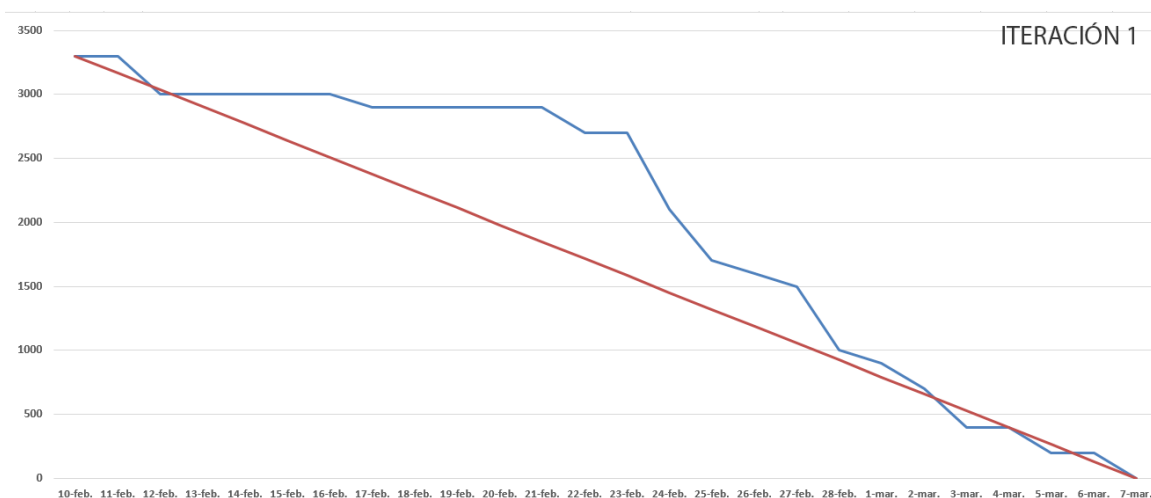


Figura C.2: Burndown chart (Iteración 1)

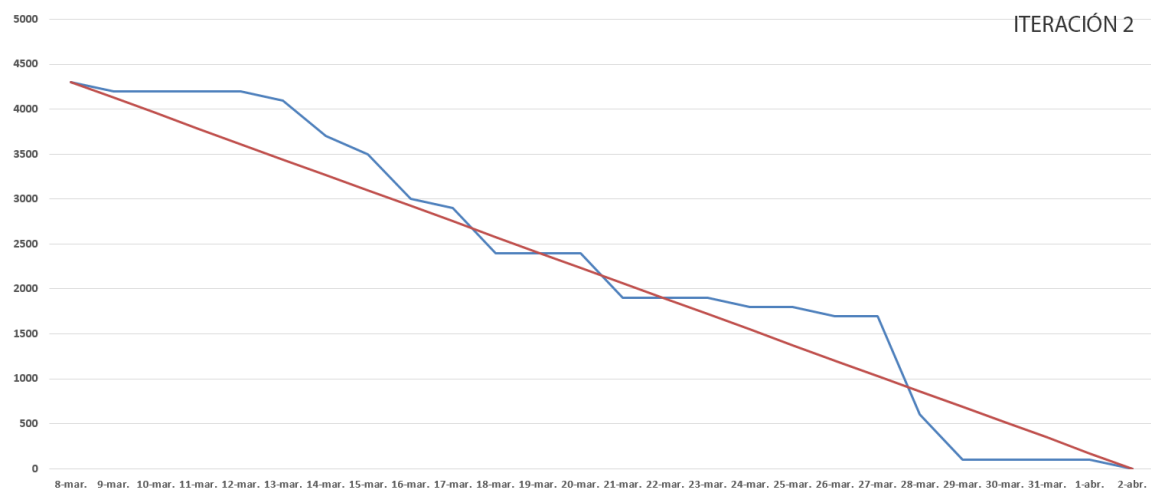


Figura C.3: Burndown chart (Iteración 2)



Figura C.4: Burndown chart (Iteración 3)

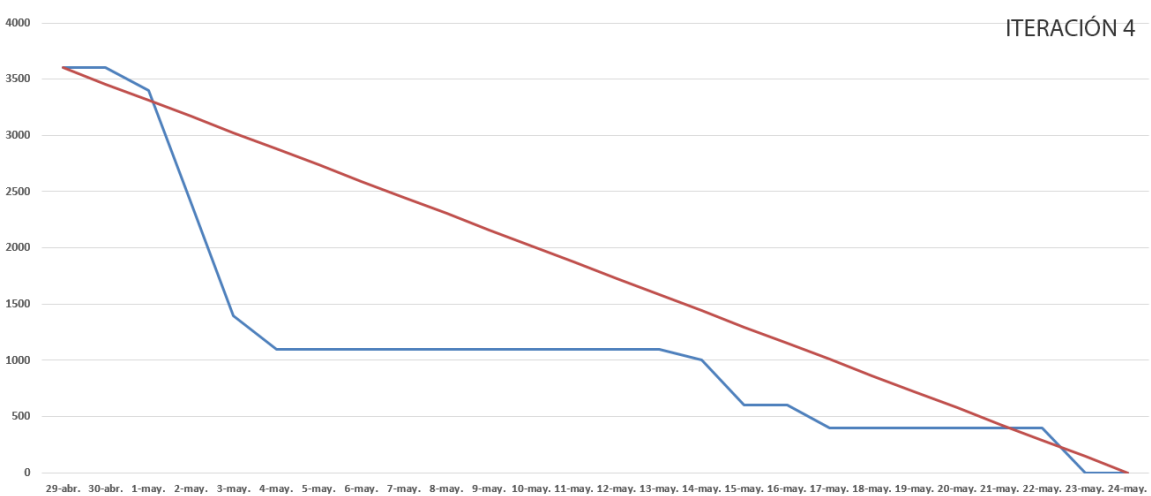


Figura C.5: Burndown chart (Iteración 4)



Figura C.6: Burndown chart (Iteración 5)

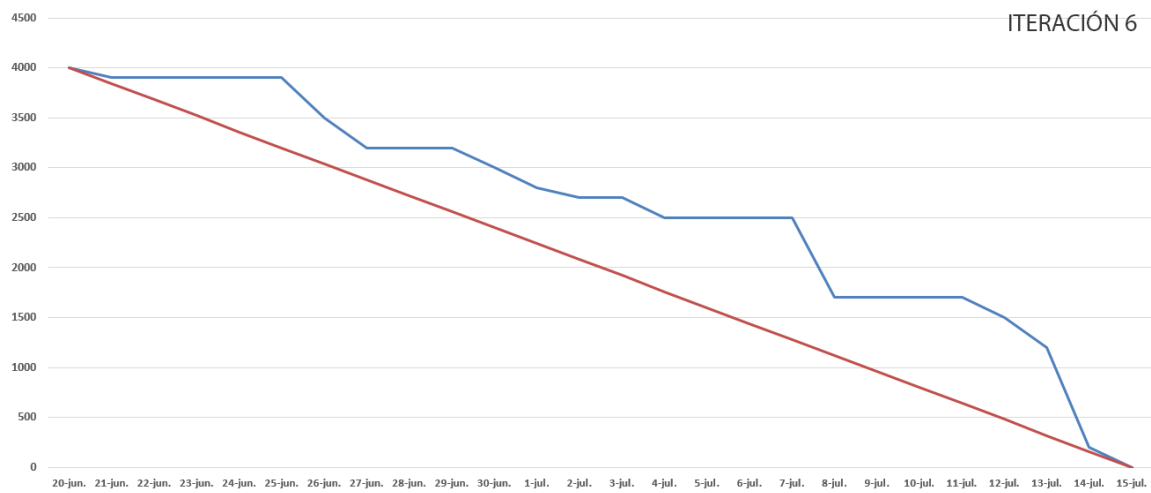


Figura C.7: Burndown chart (Iteración 6)

C.3. User stories

Como se ha explicado anteriormente, la planificación del proyecto se ha realizado obedeciendo el siguiente proceso:

1. Especificar la idea
2. Obtener requisitos
3. Elaborar *user-stories*
4. Listar tareas

ANEXO C. DETALLES DE GESTIÓN

En esta sección se expone la lista de *user-stories* (cuadro C.2) creada a partir de los requisitos y utilizada para elaborar el listado de tareas. Estos *user-stories* describen usos concretos que los diferentes tipos de usuario pueden realizar con el sistema.

ID	Requisito	User Story
1.1	RF01	Como universitario, quiero ver los datos de cada organización (nombre, localidad, nº de opiniones, puntuaciones medias...) en el listado.
1.2	RF01	Como universitario, quiero acceder a una lista de las opiniones de una organización concreta del listado.
1	RF01	Como universitario, quiero obtener un listado con todas las organizaciones disponibles.
2	RF02	Como universitario, quiero ordenar el listado de organizaciones para que se muestren primero aquellas con mayor puntuación.
3	RF03	Como universitario, quiero filtrar las organizaciones según su provincia para visualizar sólo las organizaciones cuyas localizaciones las hagan interesantes.
4	RF03	Como universitario, quiero filtrar las organizaciones según su titulación.
5	RF04	Como universitario, quiero filtrar las organizaciones escribiendo total o parcialmente su nombre.
6.1	RF05	Como universitario, quiero conocer las organizaciones de mi localidad con mejor valoración para mi titulación.
6.2	RF05	Como universitario, quiero conocer las organizaciones de mi país con mejor valoración para mi titulación.
6.3	RF05	Como universitario, quiero acceder rápidamente a las opiniones de una de las organizaciones del <i>ranking</i> .
6	RF05	Como universitario, quiero conocer las organizaciones con mejor valoración para mi titulación.
7.1	RF06	Como usuario, quiero registrarme como universitario proporcionando: Nombre, titulación, <i>email</i> , contraseña y localización.
7.2	RF06	Como universitario, quiero que sea necesario registrarse utilizando un <i>email</i> asociado a una universidad, para evitar usuarios indeseados.
7.3	RF06	Como universitario, quiero confirmar mi registro pulsando un enlace enviado en un <i>mail</i> de confirmación, para evitar suplantaciones.
7	RF06	Como usuario quiero registrarme como universitario.
8.1	RF06	Como organización, quiero registrarme proporcionando: Nombre de organización, localidad, sitio web, <i>email</i> , contraseña.
8.2	RF06	Como organización, quiero que mi cuenta se active de forma manual para evitar suplantaciones.
8	RF06	Como usuario, quiero registrarme como organización.
9	RF06	Como usuario quiero identificarme escribiendo mi <i>email</i> y mi contraseña en el formulario de <i>login</i> .
10	RF06	Como usuario, quiero desconectarme de mi cuenta de usuario.

11.1	RF07	Como universitario, quiero modificar la contraseña de mi cuenta de usuario (introduciendo mi anterior contraseña, por seguridad).
11.2	RF07	Como universitario, quiero modificar mi localización, nombre o <i>email</i> en mi cuenta de usuario.
11	RF07	Como universitario, quiero modificar mi información de usuario.
12.1	RF07	Como organización, quiero modificar la contraseña de mi cuenta de usuario (introduciendo mi anterior contraseña, por seguridad).
12.2	RF07	Como organización, quiero modificar mi localización, nombre o <i>email</i> en mi cuenta de usuario.
12	RF07	Como organización quiero modificar mi información de usuario.
13.1	RF07	Como administrador, quiero modificar la contraseña de mi cuenta de usuario (introduciendo mi anterior contraseña, por seguridad).
13	RF07	Como administrador, quiero modificar mi información de usuario.
14	RF08	Como universitario, he olvidado la contraseña, por lo que quiero recibir un <i>mail</i> que me lleve a un formulario que permita modificar mi contraseña.
15	RF08	Como organización, he olvidado la contraseña, por lo que quiero recibir un <i>mail</i> que me lleve a un formulario que permita modificar mi contraseña.
16	RF08	Como administrador, he olvidado la contraseña, por lo que quiero recibir un <i>mail</i> que me lleve a un formulario que permita modificar mi contraseña.
17.1	RF09	Como universitario, quiero ver los comentarios sobre Aprendizaje, ambiente de trabajo, remuneración y horarios de una organización.
17.2	RF09	Como universitario, quiero ver las puntuaciones sobre Aprendizaje, ambiente de trabajo, remuneración y horarios de una organización.
17.3	RF09	Como universitario, quiero ver un gráfico sobre el nivel de satisfacción del resto de universitarios en una organización de mi interés.
17.4	RF09	Como universitario, quiero ver gráficos sobre la evolución de las opiniones/valoración de una organización de mi interés.
17.5	RF09	Como universitario, quiero que solo usuarios universitarios (identificados) puedan ver las opiniones publicadas, por privacidad.
17	RF09	Como universitario, quiero ver las opiniones de usuarios en una organización de mi interés.
18.1	RF10	Como universitario, quiero escribir un comentario sobre Aprendizaje, ambiente de trabajo, remuneración y horarios de una organización.
18.2	RF10	Como universitario, quiero asociar una puntuación (positiva, neutral, negativa) a cada comentario escrito.
18.3	RF10	Como universitario, quiero que cada vez que publique una opinión, se recalcule la valoración de la organización en cuestión.
18.4	RF10	Como universitario, quiero que cada vez que opine sobre una organización, mi titulación se asocie al perfil de contratación de la organización.
18.5	RF10	Como universitario, quiero que no se permita la publicación de comentarios con caracteres aleatorios o en idiomas diferentes al español.

18	RF10	Como universitario, quiero publicar una opinión totalmente anónima sobre una organización.
19.1	RF11	Como universitario, quiero ver una lista con las ofertas disponibles y la información básica de cada oferta.
19.2	RF11	Como organización, quiero que mis ofertas dejen de aparecer en el listado cuando se cumple la fecha de caducidad indicada.
19.3	RF11	Como universitario, quiero ver la información detallada de una oferta concreta.
19	RF11	Como universitario, quiero listar las ofertas de una organización concreta.
20.1	RF12	Como universitario, quiero ver una lista con las ofertas disponibles y la información básica de cada oferta.
20.2	RF12	Como organización, quiero que mis ofertas dejen de aparecer en el listado cuando se cumple la fecha de caducidad indicada.
20.3	RF12	Como universitario, quiero ver la información detallada de una oferta concreta.
20	RF12	Como universitario, quiero listar las ofertas de mi titulación.
21.1	RF13	Como universitario, quiero contestar a una oferta adjuntado mi CV.
21.2	RF13	Como organización, no quiero que el mismo usuario pueda contestar más de una vez a la misma oferta.
21	RF13	Como universitario, quiero contestar a una oferta en la que estoy interesado.
22	RF14	Como universitario, quiero ir a la página web original de donde se ha obtenido una oferta de mi interés.
23	RF15	Como universitario, quiero ordenar las ofertas del listado para que se muestren primero aquellas cuya organización tenga una mayor puntuación en el aspecto que ofrecen en la oferta.
24	RF15	Como universitario, quiero ordenar las ofertas del listado para que se muestren primero las más nuevas.
25	RF16	Como universitario, quiero filtrar la lista de ofertas disponibles según el tipo de oferta.
26	RF16	Como universitario, quiero filtrar la lista de ofertas disponibles según la provincia
27	RF16	Como organización, quiero filtrar mi lista de ofertas según el tipo de oferta.
28.1	RF17	Como organización, quiero ver los comentarios registrados sobre mi organización.
28.2	RF17	Como organización, quiero ver las puntuaciones registradas sobre mi organización.
28.3	RF17	Como organización, quiero ver mis puntuaciones medias.
28.4	RF17	Como organización, quiero ver un gráfico sobre el nivel de satisfacción de los usuarios universitarios en mi organización.
28.5	RF17	Como organización, quiero ver gráficos sobre la evolución de las opiniones/valoración de mi organización.
28	RF17	Como organización, quiero ver las opiniones registradas sobre mi organización.
29.1	RF18	Como organización, quiero escribir un texto en respuesta de una opinión concreta de mi organización.

29.2	RF18	Como organización, quiero que cada vez que publique una contestación, se recalculen las puntuaciones de mi organización.
29	RF18	Como organización, quiero emitir una contestación oficial sobre la opinión publicada por un usuario respecto a mi organización.
30.1	RF19	Como organización, quiero escribir una oferta para una titulación concreta, indicando: Tipo de oferta, título corto, descripción y caducidad.
30.2	RF19	Como organización, quiero que se recalcule mi puntuación cada vez que publico una oferta.
30.3	RF19	Como organización, quiero que cada una de mis ofertas pueda cerrarse automáticamente al llegar a una fecha de caducidad fijada.
30	RF19	Como organización, quiero publicar una oferta.
31	RF19	Como organización, quiero re-publicar una oferta anterior que ya había caducado.
32	RF20	Como organización, quiero editar la información de una de mis ofertas publicadas.
33.1	RF21	Como organización, quiero ver la información resumida de cada una de mis ofertas en el listado: Título, especialidad, caducidad, estado (abierta/cerrada).
33.2	RF21	Como organización, quiero ver la información detallada de una de las ofertas de mi listado de ofertas.
33	RF21	Como organización, quiero ver una lista con las ofertas que he publicado.
34	RF22	Como organización, quiero ver los cv de los universitarios que han contestado a una de mis ofertas.
35.1	RF23	Como administrador, quiero conocer el número de opiniones publicadas y contestadas.
35.2	RF23	Como administrador, quiero conocer el número de organizaciones almacenadas.
35.3	RF23	Como administrador, quiero conocer el número de universitarios y de organización registradas.
35.4	RF23	Como administrador, quiero conocer el número de ofertas publicadas y recuperadas, diferenciando las ofertas vigentes de las caducadas.
35.5	RF23	Como administrador, quiero conocer el número de ofertas contestadas y el número de contestaciones a ofertas.
35	RF23	Como administrador, quiero conocer las estadísticas del sitio (número de opiniones, número de organizaciones, número de universitarios, número de ofertas).
36	RF24	Como administrador, quiero eliminar una opinión inadecuada.
37	RF25	Como administrador, quiero eliminar una organización inadecuada.
38	RF26	Como administrador, quiero banear al usuario que ha publicado una opinión inadecuada.
39	RF27	Como administrador, quiero eliminar una oferta inadecuada.
40.1	RF28	Como administrador, quiero ver la información de las organizaciones pendientes de activar.
40.2	RF28	Como administrador, quiero activar la cuenta de un usuario vinculándolo a una organización nueva.
40.3	RF28	Como administrador, quiero activar la cuenta de un usuario vinculándolo a una organización existente.

40	RF28	Como administrador, quiero activar la cuenta de un usuario organización.
41.1	RF29	Como administrador, quiero realizar un sorteo automático.
41.2	RF29	Como administrador, quiero conocer el nombre y <i>email</i> del ganador del sorteo.
41	RF29	Como administrador, quiero realizar un sorteo de premios.
42	RF30	Como universitario, quiero recibir un informe mensual del <i>ranking</i> de las mejores organizaciones.
43	RF30	Como organización, quiero recibir un informe mensual del <i>ranking</i> de las mejores organizaciones y la situación de mi organización en dicho <i>ranking</i> .
44.1	RF31	Como universitario, quiero recibir una notificación cuando se publiquen ofertas de mi titulación.
44.2	RF31	Como universitario, quiero recibir una notificación cuando una organización conteste a una de mis opiniones.
44	RF31	Como universitario, quiero recibir notificaciones cuando se produzcan novedades.
45.1	RF31	Como organización, quiero recibir una notificación cuando un usuario publica una opinión sobre mi organización.
45.2	RF31	Como organización, quiero recibir una notificación cuando un usuario envíe su CV a una de mis ofertas.
45	RF31	Como organización, quiero recibir notificaciones cuando se produzcan novedades.
46	RF32	Como usuario, no quiero encontrar opiniones ofensivas o inadecuadas.
47	RF33	Como universitario, quiero encontrar una recopilación de ofertas publicadas en otros sitios web.
48	RF33	Como organización, quiero gestionar las ofertas de mi organización recuperadas de otros sitios web.
49	RF34	Como universitario, quiero que se detecte automáticamente la puntuación de la opinión que estoy escribiendo.
50.1	RF35	Como usuario, quiero recomendar el sitio web a mis amigos a través de <i>Facebook</i> .
50.2	RF35	Como usuario, quiero recomendar el sitio web a mis amigos a través de <i>Twitter</i> .
50.3	RF35	Como usuario, quiero recomendar el sitio web a mis amigos a través de <i>Google+</i> .
50	RF35	Como usuario, quiero recomendar el sitio web a mis amigos.

Cuadro C.2: Listado de user-stories

Anexo D

Manuales

En este anexo se explica como utilizar y desplegar el sistema desarrollado. En las secciones D.1, D.2 y D.3 se explica como utilizar las funcionalidades del sistema diseñadas para usuarios universitarios, organizaciones y administradores, respectivamente. Por otro lado, la sección D.4 engloba los pasos a seguir para desplegar el sistema en un servidor.

D.1. Manual de uso (Usuario universitario)

Este manual explica las funcionalidades que el sistema proporciona a los usuarios universitarios.

Crear usuario

Para comenzar a utilizar *Universando* se debe crear una nueva cuenta de usuario.

The screenshot shows the 'universando.es' website header with a user icon (1) and a navigation bar with 'IDENTIFICATE', 'REGISTRATE' (2), and 'AYUDA'. A registration form (3) is displayed with fields for Email, Contraseña, Repetir contraseña, Tipo de cuenta, and Provincia. A green 'CREAR USUARIO' button (4) is at the bottom. A disclaimer at the bottom states: 'Registrándote aceptas las condiciones expuestas en la sección de AYUDA.'

Figura D.1: Manual de uso: Crear usuario

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.1).
2. En el *popup* desplegado, pulsar la opción “Registrate” (señalada con un 2 en la figura D.1).
3. En el formulario mostrado (señalado con un 3 en la figura D.1) se debe rellenar toda la información solicitada. El registro de usuarios universitarios debe realizarse utilizando una dirección de correo universitario. Posteriormente se podrá editar esta dirección de correo.

4. Una vez completado el formulario de registro, se debe pulsar el botón “Crear usuario” (señalado con un 4 en la figura D.1).
5. Finalmente, se emitirá un mensaje con las instrucciones necesarias para activar la nueva cuenta.

Acceder como usuario registrado

Para acceder a *Universando* como usuario registrado, una vez completado el proceso de registro y activación de la cuenta, se debe realizar el siguiente proceso:

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.2).
2. En el *popup* desplegado, completar el formulario de identificación (señalado con un 2 en la figura D.2).
3. Una vez completado el formulario de identificación, se debe pulsar el botón “Identificar” (señalado con un 3 en la figura D.2).



Figura D.2: Manual de uso: Acceder como usuario registrado

Recuperar contraseña olvidada

En caso de haber olvidado la contraseña asociada a tu cuenta de usuario, puedes modificarla mediante el siguiente proceso:

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.3).
2. En el *popup* desplegado, completar el campo “email” del formulario mostrado (señalado con un 2 en la figura D.3).

3. Pulsar el botón “He olvidado la contraseña” (señalado con un 3 en la figura D.3).
4. Tras esto, se emitirá un correo electrónico con las instrucciones para restablecer la contraseña olvidada.



Figura D.3: Manual de uso: Recuperar contraseña olvidada

Modificar información de usuario

Una vez identificado como usuario, es posible modificar la información vinculada a dicha cuenta.

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.4).
2. En el *popup* desplegado, pulsar la opción “Ajustes de cuenta” (señalada con un 2 en la figura D.4).
3. En el formulario mostrado (señalado con un 1 en la figura D.5) se debe escribir la contraseña actual en el campo situado en la parte superior del formulario.
4. Si se desea modificar la contraseña, puede escribirse una nueva en los campos “Nueva contraseña” y “Repetir contraseña”. Si se desea mantener la contraseña actual, se deben dejar estos campos vacíos.
5. Si se desea modificar el nombre, provincia o email de la cuenta de usuario, se pueden editar los campos inferiores del formulario.
6. Una vez editada la información del formulario, se debe pulsar el botón “Actualizar información” (señalado con un 2 en la figura D.5).



Figura D.4: Manual de uso: Modificar información de usuario (1)

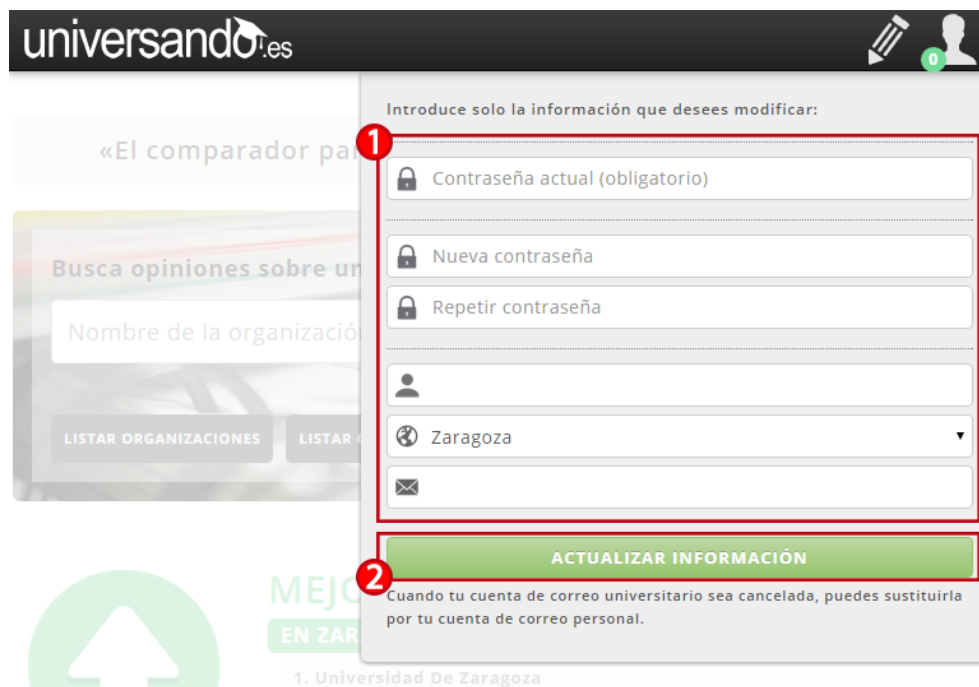


Figura D.5: Manual de uso: Modificar información de usuario (2)

Desconectar de cuenta de usuario

Una vez identificado como usuario, para salir manualmente de la cuenta puede seguirse este proceso:

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.6).
2. En el *popup* desplegado, pulsar la opción “Cerrar sesión” (señalada con un 2 en la figura D.6).



Figura D.6: Manual de uso: Desconectar de cuenta de usuario

Ver novedades

Una vez identificado como usuario, en el icono de usuario de la cabecera de la página (señalado con un 1 en la figura D.7) aparecerá un círculo verde indicando el número de novedades disponibles. Pueden consultarse las novedades mediante el siguiente proceso:

1. Pulsar el icono de usuario situado en la cabecera de la página (señalado con un 1 en la figura D.7).
2. En el *popup* desplegado, se pueden observar las novedades disponibles (señaladas con un 2 en la figura D.7).
3. Pulsando sobre estas novedades, serás redirigido a la página correspondiente.



Figura D.7: Manual de uso: Ver novedades

Consultar mejores/peores organizaciones

Desde la página inicial del sitio web es posible ver un *ranking* con las mejores y peores organizaciones de tu provincia (Figura D.8). Pulsando sobre el nombre de dichas organizaciones es posible acceder a la página con las opiniones de las organizaciones en cuestión.



universando.es

MEJORES ORGANIZACIONES

EN ZARAGOZA

1. Universidad De Zaragoza
2. HAILO. La App Para Taxis
3. Valencia Lounge Hostel

EN ESPAÑA

1. Universidad De Zaragoza (Zaragoza)
2. HAILO. La App Para Taxis (Madrid)
3. Valencia Lounge Hostel (Valencia)

PEORES ORGANIZACIONES

EN ZARAGOZA

1. HAILO. La App Para Taxis
2. Valencia Lounge Hostel
3. Talent Search People

EN ESPAÑA

1. HAILO. La App Para Taxis (Madrid)
2. Valencia Lounge Hostel (Valencia)
3. Talent Search People (Barcelona)

Figura D.8: Manual de uso: Consultar mejores/peores organizaciones

Buscar/listar organizaciones

Universando permite la búsqueda de organizaciones según diferentes criterios.

1. Acceder a la página principal.
2. Si se desean listar todas las organizaciones, se debe pulsar el botón “Listar organizaciones” (señalado con un 1 en la figura D.9).

3. Si se desean buscar organizaciones con un nombre o provincia concretos, se debe utilizar el buscador principal (señalado con un 2 en la figura D.9).
4. En el listado de organizaciones (Figura D.10) aparecerán los nombres y puntuaciones de las organizaciones disponibles para los criterios dados.
5. Si se desean modificar los criterios utilizados para buscar/listar las organizaciones, se pueden utilizar las opciones situadas en la parte superior de la página (señalados con un 1 en la figura D.10).
6. Si se desea reordenar el listado según un criterio específico, puede realizarse mediante el seleccionable etiquetado con el texto “Ordenar por” (señalado con un 2 en la figura D.10).

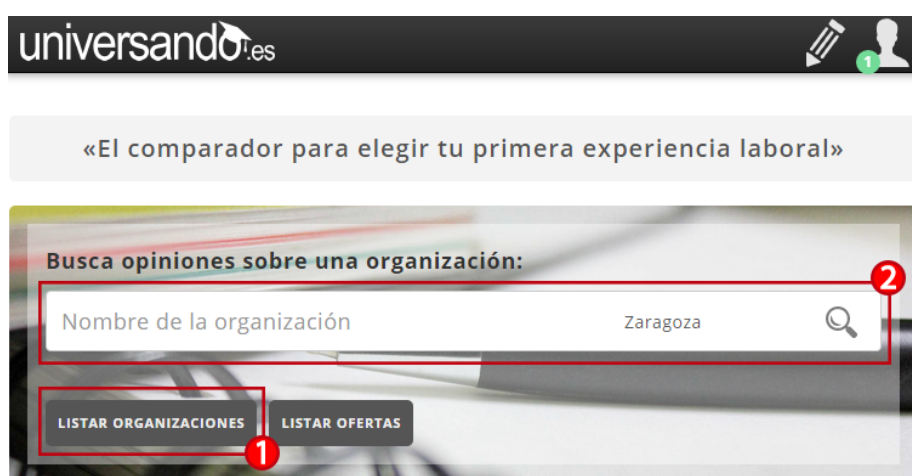


Figura D.9: Manual de uso: Buscar/listar organizaciones (1)



Figura D.10: Manual de uso: Buscar/listar organizaciones (2)

Ver puntuaciones y opiniones de una organización

Una vez identificado como usuario, es posible ver las opiniones publicadas sobre una organización concreta.

1. Acceder al listado de organizaciones, mediante el método explicado anteriormente.
2. Pulsar el nombre de la organización que se desea consultar.

También es posible acceder directamente a las opiniones de las organizaciones del *ranking* existente en la página inicial, pulsando sobre los nombres de dichas organizaciones.

Ver detalles de valoración de una organización

Una vez identificado como usuario, es posible acceder a los detalles de la valoración de una organización concreta.

1. Acceder a la página de opiniones de una organización concreta.
2. Pulsar el botón “Ver detalles” localizado en la parte superior de la página (señalado con un 1 en la figura D.11).
3. Aparecerá una ventana emergente con la información detallada de la valoración de la organización (Figura D.12).



Figura D.11: Manual de uso: Ver detalles de valoración de una organización (1)

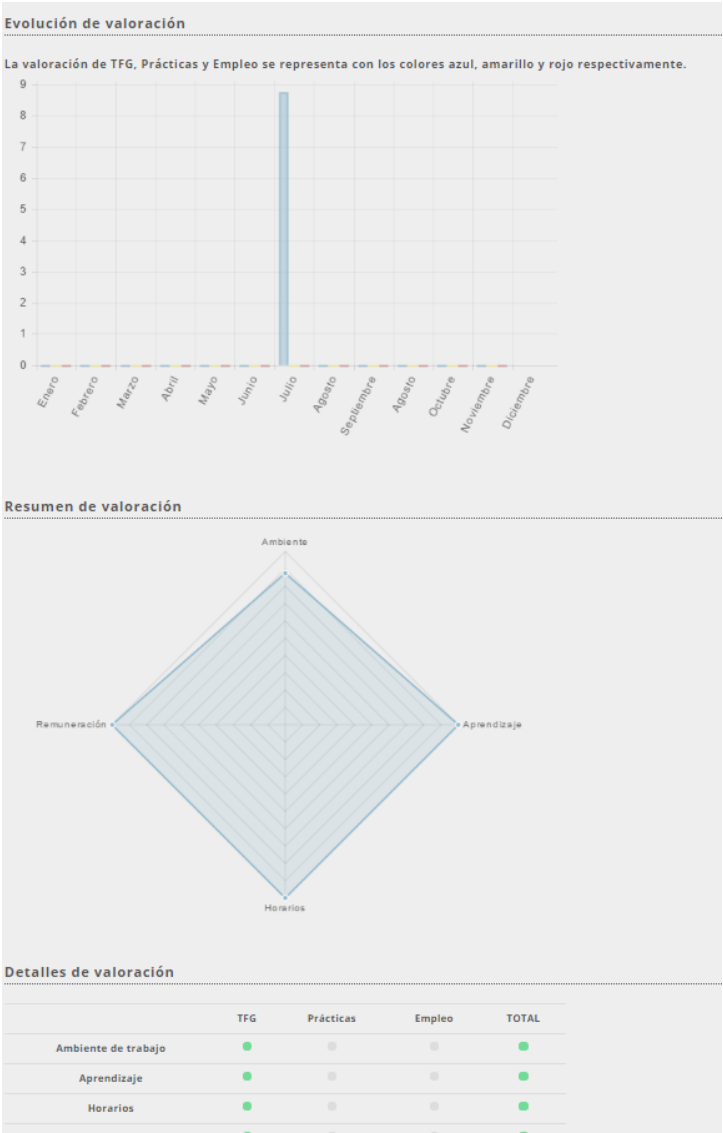


Figura D.12: Manual de uso: Ver detalles de valoración de una organización (2)

Publicar una opinión

Una vez identificado como universitario, es posible publicar nuevas opiniones sobre organizaciones.

1. Pulsar el botón de escritura localizado en la cabecera (señalado con un 1 en la figura D.13).
2. Seleccionar la provincia de la organización sobre la que opinar. Tras ello, aparecerá un desplegable para seleccionar la organización concreta (señalado con un 2 en la figura D.13).
3. Seleccionar la organización sobre la que opinar. Si la organización no se encuentra en la lista, se debe pulsar el enlace “¿No está la organización que buscas?” y escribir el nombre de dicha organización en el cuadro de texto que aparecerá.
4. Seleccionar el tipo de opinión (TFG, prácticas, empleo).

5. Escribir la opinión sobre los aspectos “Ambiente de trabajo”, “Aprendizaje”, “Horarios” y “Remuneración”. Mientras se escriben estos textos, se calculará automáticamente su puntuación, activándose uno de los iconos situados a la derecha (señalados con un 3 en la figura D.13).
6. Si no estás conforme con alguna de las puntuaciones asignadas automáticamente, puedes modificarla pulsando sobre el icono correspondiente (señalados con un 3 en la figura D.13).
7. Pulsar el botón “Publicar opinión” situado al final de la página.

universando.es

PUBLICA TU OPINIÓN. ¡Es anónimo!

Si has realizado un proyecto, prácticas o has tenido un primer empleo en una empresa u organización. ¡Cuéntanos qué tal!

Nombre de la organización

Organización

¿No está la organización que buscas?

Provincia de la organización

Provincia

Tipo de trabajo

Tipo de trabajo

Danos tu opinión sobre...

Escribe una valoración textual y selecciona un indicador de satisfacción.

Ambiente de trabajo

¿Qué tal con tus compañeros? ¿Y con tus jefes?

3

Figura D.13: Manual de uso: Publicar una opinión

Buscar/listar ofertas

Una vez identificado como usuario, es posible obtener las ofertas disponibles.

1. Acceder a la página principal.
2. Pulsar el botón “Listar ofertas” (señalado con un 1 en la figura D.14).

3. Se mostrará el listado con las ofertas disponibles para la titulación correspondiente. A la izquierda de cada opinión, se mostrará el tipo de la oferta junto con la puntuación de la organización.
4. Si se desea reordenar o filtrar el listado de ofertas, es posible hacerlo mediante los filtros situados en la parte superior de la página (señalados con un 1 en la figura D.15).
5. Si se desea leer el contenido completo de una oferta, se debe pulsar el botón “Leer oferta completa” situado en la caja de la oferta en cuestión (señalado con un 2 en la figura D.15).
6. Si se desean ver las opiniones de la organización que ha publicado una oferta, se debe pulsar el botón “Ver opiniones” situado en la caja de la oferta en cuestión.

Otra forma de obtener las ofertas de una organización concreta es acceder a la página de opiniones de dicha organización, mediante los métodos ya explicados, y pulsar el botón “Ver ofertas”.



Figura D.14: Manual de uso: Buscar/listar ofertas (1)



Figura D.15: Manual de uso: Buscar/listar ofertas (2)

Enviar datos a una oferta

Una vez identificado como usuario universitario, es posible enviar un currículum en formato PDF a una oferta de tu interés.

1. Acceder a la página donde se encuentra la oferta de interés, utilizando los métodos ya explicados.
2. Pulsar el botón “Leer oferta completa” situado en la caja de la oferta de interés (señalado con un 1 en la figura D.16).
3. Pulsar el botón “Me interesa” (señalado con un 2 en la figura D.16).
4. Si la oferta ha sido publicada manualmente por una organización, aparecerá un formulario (señalado con un 3 en la figura D.16) donde se puede elegir enviar un nuevo currículum en formato PDF o enviar el mismo currículum que enviaste a la última oferta de tu interés (si se da el caso).
5. Si la oferta ha sido publicada automáticamente por el sistema, serás redirigido a la página externa de donde se obtuvo dicha oferta.

universando.es

Creación: 22-07-2015 · Caducidad: 24-08-2015

TFG 8.8

UNIVERSIDAD DE ZARAGOZA (ZARAGOZA) - INGENIERÍA INFORMÁTICA

Herramienta de Generación Automática de Mapas Conceptuales

Los mapas conceptuales son una herramienta para representar de forma gráfica, sencilla e intuitiva el conocimiento. Permiten transmitir información de forma simple y concisa, y son muy usados en distintas disciplinas para expresar diseños, comunicar ideas, formar, evaluar, etc. En este proyecto se plantea la creación de una herramienta que a partir de un texto en lenguaje natural sea capaz de crear un mapa conceptual que represente el conocimiento. Para ello se combinarán técnicas basadas en aprendizaje automático, procesamiento del lenguaje natural e ingeniería ontológica. El proyecto se realizará en colaboración con una conocida empresa de comunicación, existiendo la posibilidad de proporcionar al alumno beca o ayuda al estudio asociada a la realización del proyecto.

VER OPINIONES LEER OFERTA COMPLETA ¡ME INTERESA!

Envía tu último CV utilizado o carga uno nuevo.

Seleccionar archivo Ningún archivo seleccionado

ENVIAR NUEVO CV

ENVIAR ÚLTIMO CV

Figura D.16: Manual de uso: Enviar datos a una oferta

D.2. Manual de uso (Usuario organización)

El siguiente manual explica las diferentes funcionalidades que el sistema proporciona a los usuarios vinculados a una organización.

Crear usuario

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Acceder como usuario registrado

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Recuperar contraseña olvidada

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Modificar información de usuario

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Desconectar de cuenta de usuario

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Ver novedades

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Ver opiniones sobre mi organización

Una vez identificado como organización, es posible leer las opiniones que los usuarios han publicado sobre tu organización.

1. Acceder a la página principal.
2. En la página mostrada se visualizarán las puntuaciones de la organización en la parte superior (señaladas con un 1 en la figura D.17) y las opiniones de los usuarios en la parte inferior. A la izquierda de cada opinión se mostrará una caja indicando el tipo, fecha y puntuación de dicha opinión.

3. Si se desea mostrar únicamente un tipo de opinión (TFG, prácticas o empleo), es posible hacerlo mediante el filtro de tipo (señalado con un 2 en la figura D.17).
4. Si se desea leer el contenido completo de una opinión, es posible hacerlo pulsando el botón “Leer opinión completa” de la opinión en cuestión (señalado con un 3 en la figura D.17).



Figura D.17: Manual de uso: Ver opiniones sobre mi organización

Ver detalles de valoración de mi organización

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Responder opiniones sobre mi organización

Una vez identificado como organización, es posible contestar a las opiniones que los usuarios han publicado sobre tu organización, lo cual influirá positivamente en tu puntuación de “Feedback”.

1. Acceder a la página principal, que muestra las opiniones sobre tu organización.

2. Pulsar el botón “Contestar” de la opinión que se desea responder (señalado con un 1 en la figura D.18).
3. Se mostrará un nuevo formulario (señalado con un 2 en la figura D.18) donde escribir una respuesta a dicha opinión. Tras escribir la respuesta, se debe pulsar el botón “Publicar respuesta”.

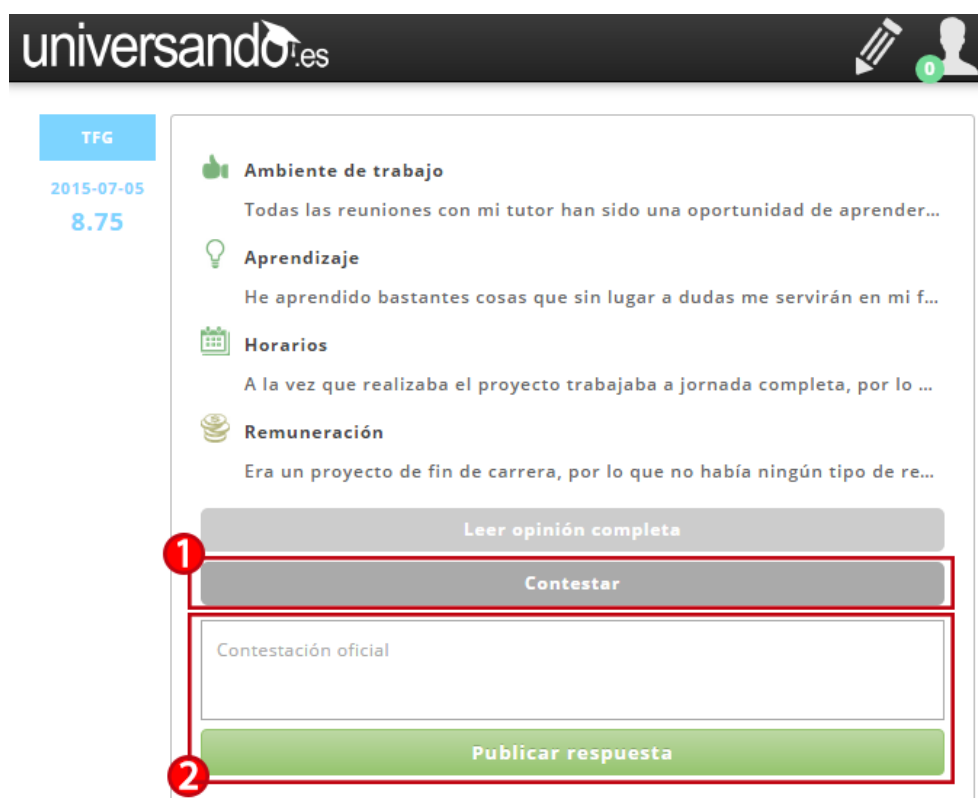


Figura D.18: Manual de uso: Responder opiniones sobre mi organización

Ver mis ofertas publicadas

Una vez identificado como organización, es posible ver y gestionar las ofertas publicadas.

1. Acceder a la página principal.
2. En la página mostrada, se debe pulsar el botón “Ver ofertas” (señalado con un 1 en la figura D.19).
3. Se mostrará una página con las ofertas publicadas a lo largo del tiempo (Figura D.20). En la parte izquierda de cada oferta se puede observar su tipo, mientras que en la parte superior se puede observar su fecha de creación y de caducidad, mostrándose en rojo si la oferta ya ha caducado.
4. Si se desean ordenar o filtrar las ofertas según un criterio concreto, se pueden utilizar los filtros localizados en la parte superior de la página (señalados con un 1 en la figura D.20).
5. Si se desea ver el contenido completo de una oferta, se debe pulsar el botón “Leer oferta completa” situado en la caja de la oferta en cuestión (señalado con un 2 en la figura D.20).



universando.es

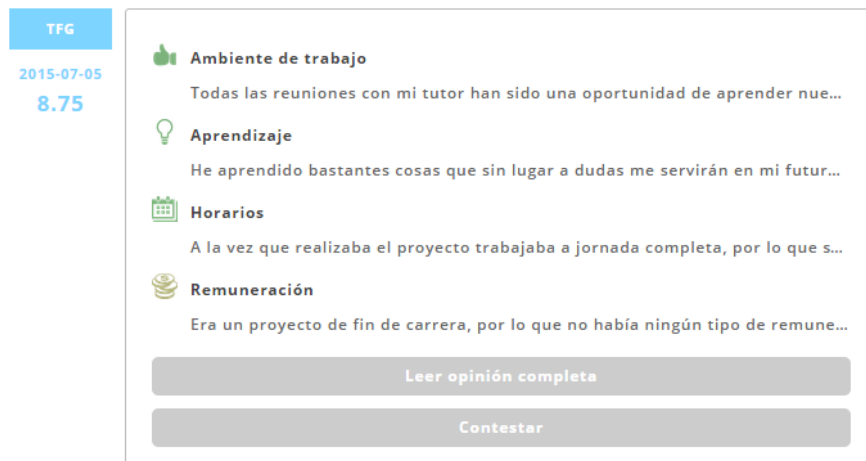
UNIVERSIDAD DE ZARAGOZA
Zaragoza. Ingeniería informática

8.9

TFG	8.75	PRÁCTICAS	7.77	EMPLEO	7.77	FEEDBACK	10.00
VER DETALLES VER OFERTAS							

1 opiniones - Página 1

Mostrar todo ▼



TFG

2015-07-05
8.75

Ambiente de trabajo
Todas las reuniones con mi tutor han sido una oportunidad de aprender nue...

Aprendizaje
He aprendido bastantes cosas que sin lugar a dudas me servirán en mi futur...

Horarios
A la vez que realizaba el proyecto trabajaba a jornada completa, por lo que s...

Remuneración
Era un proyecto de fin de carrera, por lo que no había ningún tipo de remune...

[Leer opinión completa](#)

[Contestar](#)

Figura D.19: Manual de uso: Ver mis ofertas publicadas (1)



universando.es

Ofertas de Universidad de Zaragoza

Página 1

Ordenar por fecha ▼ Todas las provincias ▼ Tipo de oferta ▼

Creación: 02-08-2015 · Caducidad: 24-08-2015

TFG
8.8

UNIVERSIDAD DE ZARAGOZA (ZARAGOZA) - INGENIERÍA INDUSTRIAL
APLICACIÓN BIOMÉDICA DE LA ELECTROPORACIÓN PARA TRATAMIENTO DE TUMORES MEDIANTE UN SISTEMA ELECTRÓNICO

[EDITAR/REPUBLICAR](#) [VER OPINIONES](#) [VER CANDIDATOS \(0\)](#)

[LEER OFERTA COMPLETA](#)

Figura D.20: Manual de uso: Ver mis ofertas publicadas (2)

Ver usuarios interesados en ofertas

Una vez identificado como usuario, es posible ver que universitarios han enviado sus currículos a una de tus ofertas.

1. Acceder a la página donde se encuentra la oferta en cuestión.
2. Pulsar el botón “Ver candidatos” (señalado con un 1 en la figura D.21).
3. Se mostrará un listado con los usuarios que han enviado su currículum. Dichos currículos pueden obtenerse en formato PDF pulsando sobre el nombre de los usuarios.



Figura D.21: Manual de uso: Ver usuarios interesados en ofertas

Publicar oferta

Una vez identificado como organización, es posible publicar nuevas ofertas. Al publicar nuevas ofertas, tu puntuación de “Feedback” se verá incrementada.

1. Pulsar el botón de escritura situado en la cabecera (señalado con un 1 en la figura D.22).
2. Rellenar el formulario que aparecerá (Figura D.22), especificando una titulación, tipo de trabajo, fecha de caducidad, descripción corta (título) y descripción larga.

3. Pulsar el botón “Publicar oferta” (señalado con un 2 en la figura D.22).



universando.es

PUBLICA UNA NUEVA OFERTA
Publica ofertas para la realización de TFG, prácticas o primeros empleos en tu empresa/organización e incrementa tus puntos de feedback.

Titulación
Titulación

Tipo de trabajo
Tipo de trabajo

Fecha de caducidad
dd/mm/yyyy

Descripción
Descripción corta
Descripción larga

PUBLICAR OFERTA

Figura D.22: Manual de uso: Publicar oferta

Editar/Re-publicar oferta

Una vez identificado como organización, es posible editar o republicar las ofertas ya existentes.

1. Acceder a la página donde se encuentra la oferta que deseas editar o republicar, mediante los métodos ya explicados.
2. Pulsar el botón “Editar/Republicar” situado en la caja de la oferta en cuestión (señalado con un 1 en la figura D.23).
3. En el formulario que aparecerá (Figura D.24) se podrán modificar los datos y ampliar la fecha de caducidad de la oferta, pudiendo republicar ofertas ya caducadas.
4. Pulsar el botón “Publicar oferta” (señalado con un 1 en la figura D.24).



universando.es

Ofertas de Universidad de Zaragoza

Página 1

Ordenar por fecha ▼ Todas las provincias ▼ Tipo de oferta ▼

Creación: 02-08-2015 · Caducidad: 24-08-2015

TFG 8.8

UNIVERSIDAD DE ZARAGOZA (ZARAGOZA) - INGENIERÍA INDUSTRIAL

APLICACIÓN BIOMÉDICA DE LA ELECTROPORACIÓN PARA TRATAMIENTO DE TUMORES MEDIANTE UN SISTEMA ELECTRÓNICO

EDITAR/REPUBLICAR VER OPINIONES VER CANDIDATOS (0)

LEER OFERTA COMPLETA

Figura D.23: Manual de uso: Editar/Re-publicar oferta (1)



universando.es

PUBLICA UNA NUEVA OFERTA

Publica ofertas para la realización de TFG, prácticas o primeros empleos en tu empresa/organización e incrementa tus puntos de feedback.

Titulación

Ingeniería industrial ▼

Tipo de trabajo

TFG ▼

Fecha de caducidad

24/08/2015

Descripción

APLICACIÓN BIOMÉDICA DE LA ELECTROPORACIÓN PARA TRATAMIENTO DE TUMORES MEDIANTE UN SISTEMA ELECTRÓNICO

de tumores mediante el uso de un generador de alta tensión ($E > 1000$ V/cm) y frecuencia (pulsos de μs). El objetivo de este trabajo es el estudio de los efectos en células vegetales de diferentes protocolos de aplicación de la electroporación, y la extracción de conclusiones que permitan diseñar y optimizar futuras terapias de tratamiento de tumores en seres humanos. Para la realización de este trabajo se requieren conocimientos básicos de electrónica digital y de potencia. Además, se valorará la especial motivación en aplicaciones biomédicas y/o conocimientos de...

PUBLICAR OFERTA

Figura D.24: Manual de uso: Editar/Re-publicar oferta (2)

D.3. Manual de uso (Usuario administrador)

El siguiente manual explica las diferentes funcionalidades que el sistema proporciona a los usuarios administradores.

Acceder como usuario registrado

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Recuperar contraseña olvidada

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Modificar información de usuario

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Desconectar de cuenta de usuario

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Consultar mejores/peores organizaciones

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Buscar/listar organizaciones

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Ver puntuaciones y opiniones de una organización

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Ver detalles de valoración de una organización

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

Buscar/listar ofertas

(Ver funcionalidad equivalente para usuarios universitarios en D.1)

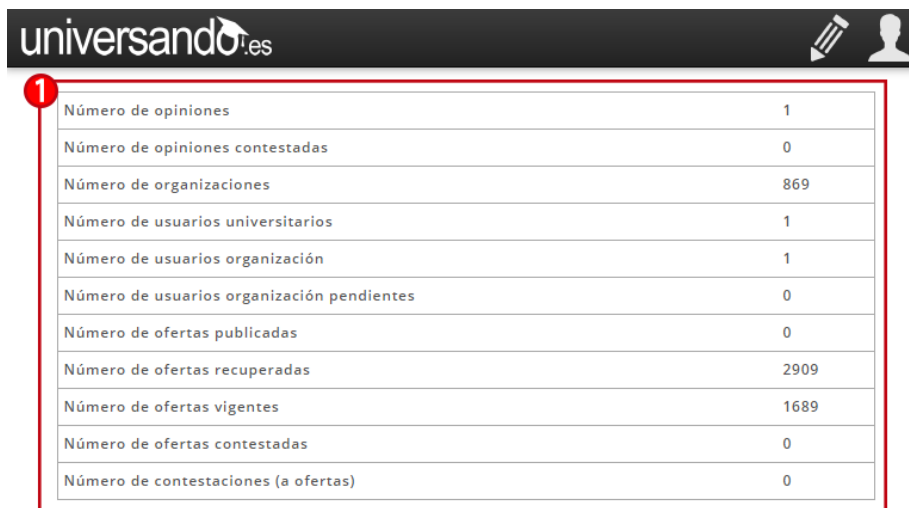
Ver usuarios interesados en ofertas

(Ver funcionalidad equivalente para usuarios de organizaciones en D.2)

Ver estadísticas

Una vez identificado como administrador, es posible visualizar las estadísticas de uso del sistema.

1. Acceder a la página principal de *Universando*.
2. En la parte inferior de la página principal se encuentran las estadísticas del sistema (señaladas con un 1 en la figura D.25).



universando.es	
Número de opiniones	1
Número de opiniones contestadas	0
Número de organizaciones	869
Número de usuarios universitarios	1
Número de usuarios organización	1
Número de usuarios organización pendientes	0
Número de ofertas publicadas	0
Número de ofertas recuperadas	2909
Número de ofertas vigentes	1689
Número de ofertas contestadas	0
Número de contestaciones (a ofertas)	0

Figura D.25: Manual de uso: Ver estadísticas

Eliminar organización

Una vez identificado como usuario administrador, es posible eliminar una organización concreta.



universando.es

UNIVERSIDAD DE ZARAGOZA
Zaragoza. Ingeniería informática

8.2

TFG 8.75 PRÁCTICAS ?..? EMPLEO ?..? FEEDBACK 5.00

ELIMINAR ORGANIZACIÓN VER DETALLES VER OFERTAS

Figura D.26: Manual de uso: Eliminar organización

1. Acceder a la página de opiniones de la organización a eliminar, mediante cualquiera de los métodos ya explicados.
2. Pulsar el botón “Eliminar organización” (señalado con un 1 en la figura D.26). Se debe tener en cuenta que al eliminar una organización, se eliminarán todas sus opiniones y ofertas.

Eliminar oferta

Una vez identificado como usuario administrador, es posible eliminar una oferta concreta.

1. Acceder a la página donde se encuentra la oferta a eliminar.
2. Pulsar el botón “Eliminar” localizado en la caja de la oferta en cuestión (señalado con un 1 en la figura D.29).

Eliminar opinión

Una vez identificado como administrador, es posible eliminar una opinión concreta.

1. Acceder a la página donde se encuentra la opinión a eliminar.
2. Pulsar el botón “Leer opinión completa” localizado en la caja de dicha opinión (señalado con un 1 en la figura D.27).
3. Pulsar el botón “Eliminar” (señalado con un 1 en la figura D.28) situado en la parte inferior del *popup* abierto que muestra el contenido completo de la opinión.

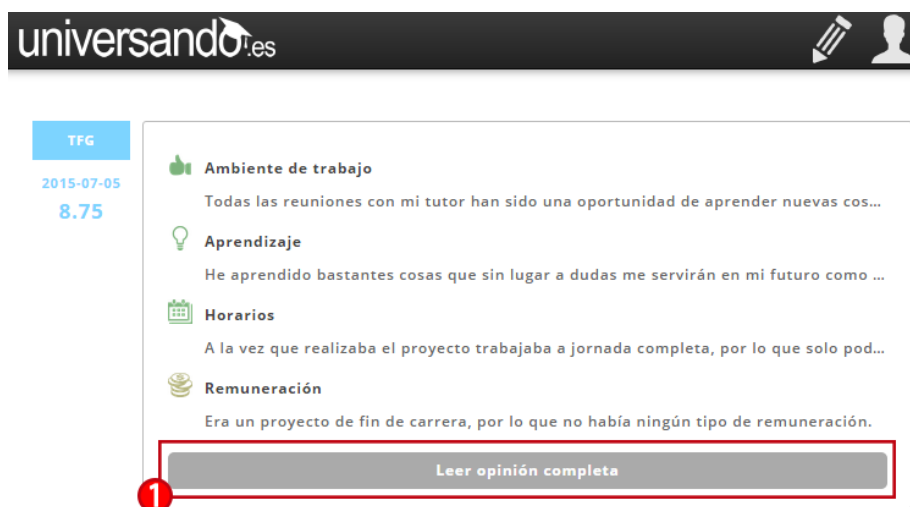


Figura D.27: Manual de uso: Eliminar opinión (1)

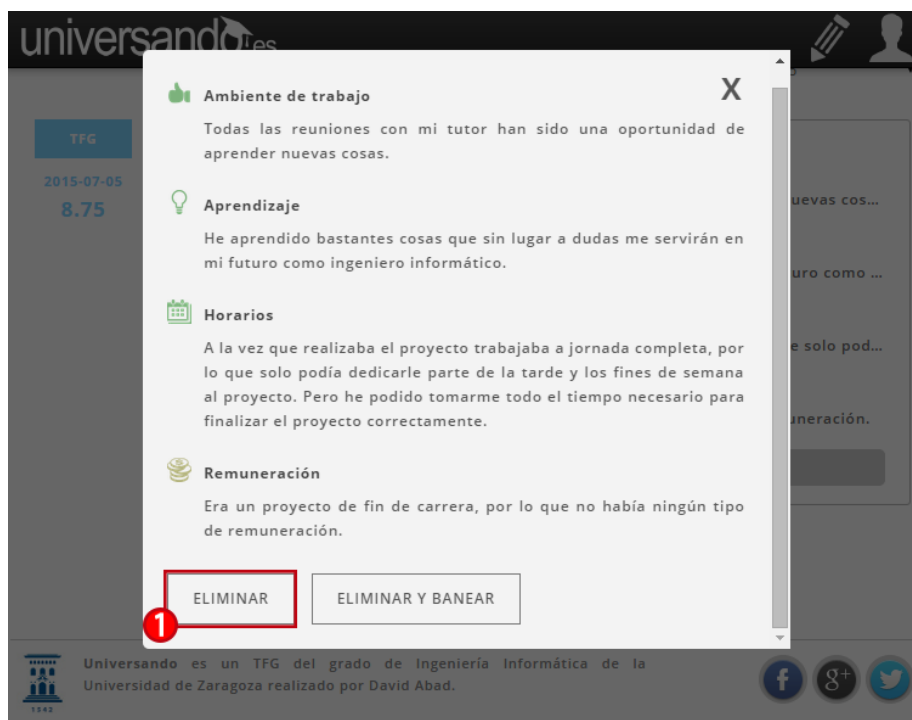


Figura D.28: Manual de uso: Eliminar opinión (2)



Figura D.29: Manual de uso: Eliminar oferta

Banear usuario

Una vez identificado como usuario administrador, es posible banear al usuario que ha escrito una opinión concreta.

1. Acceder a la página donde se encuentra la opinión a eliminar.

2. Pulsar el botón “Leer opinión completa” localizado en la caja de la opinión en cuestión (señalado con un 1 en la figura D.30).
3. Pulsar el botón “Eliminar y banear” (señalado con un 1 en la figura D.31) situado en la parte inferior del *popup* abierto que muestra el contenido completo de la opinión. Se debe tener en cuenta que además de bloquear al usuario, también se eliminará la opinión implicada en este proceso.

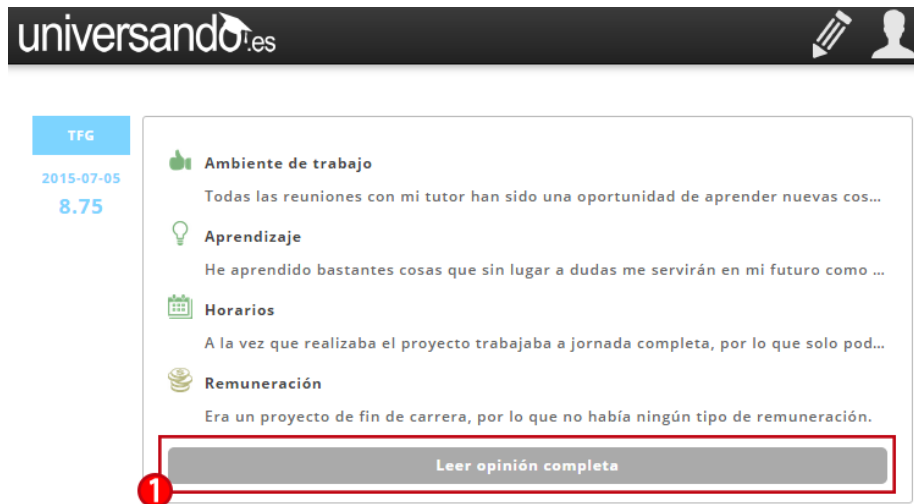


Figura D.30: Manual de uso: Banear usuario (1)

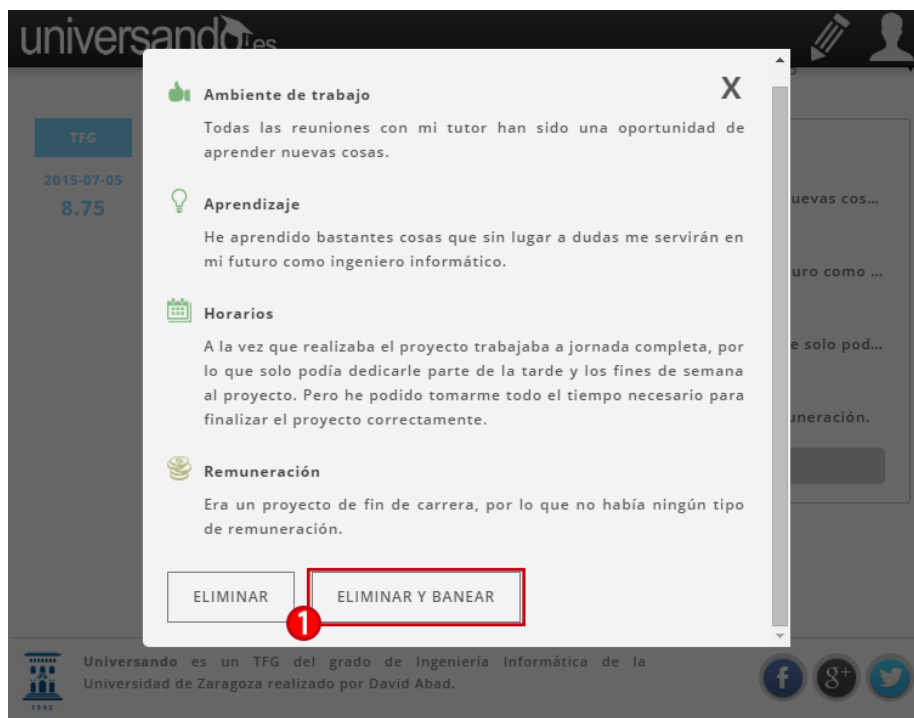


Figura D.31: Manual de uso: Banear usuario (2)

Activar organizaciones

Una vez identificado como usuario administrador, se pueden listar y activar los usuarios registrados como organización que están pendientes de activación.

- 1. Pulsar el botón de escritura situado en la cabecera del sitio web (señalado con un 1 en la figura D.32).
- 2. Se mostrará un nuevo formulario con los usuarios pendientes de activar.
- 3. Si se desea rechazar un usuario, se debe pulsar el botón “Rechazar” (señalado con un 2 en la figura D.32).
- 4. Si se desea activar un usuario vinculándolo a una organización nueva en el sistema, se debe pulsar el botón “Enviar” (señalado con un 3 en la figura D.32). Este proceso creará una nueva organización con el nombre indicado por el usuario.
- 5. Si se desea activar un usuario vinculándolo a una organización ya existente en el sistema, se debe escribir el ID de dicha organización en la caja “ID de la org.” y pulsar el botón “Enviar” (señalado con un 3 en la figura D.32). El ID correspondiente a dicha organización aparece en la URL de su página de opiniones (señalado con un 1 en la figura D.33).



Figura D.32: Manual de uso: Activar organizaciones (1)

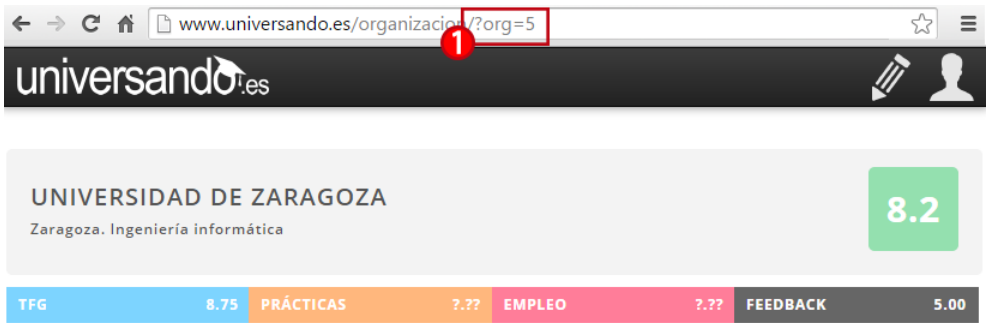


Figura D.33: Manual de uso: Activar organizaciones (2)

Realizar sorteo

Una vez identificado como usuario administrador, es posible realizar el sorteo de premios correspondiente al último mes vencido.

1. Acceder a la página principal de *Universando*.
2. Pulsar el botón “Realizar sorteo” situado en la parte inferior de la página principal (señalado con un 1 en la figura D.34).
3. El nombre y *email* del ganador aparecerá justo encima del botón “Realizar sorteo” (señalado con un 2 en la figura D.34), aunque esta información solo será visible completamente para los usuarios administradores.



Figura D.34: Manual de uso: Realizar sorteo

D.4. Manual de despliegue

En este manual se proceden a explicar los pasos necesarios para desplegar el sistema desarrollado en un servidor con un sistema operativo *Debian* de 64 bits.

1. Actualizar los repositorios del sistema:

```
apt-get update
```

2. Instalar *Apache*:

```
apt-get install apache2
```

3. Configurar *Apache*:

Cambiar el puerto por defecto en el fichero `/etc/apache2/ports.conf`, modificando el puerto 80 por el 8080. Tras esto, se reinicia *Apache*:

```
service apache2 restart
```

4. Instalar *MySQL*:

```
apt-get install mysql-server mysql-client
```

5. Instalar *PHP*:

```
apt-get install php5
```

6. Instalar *PhpMyAdmin*:

Durante la instalación se debe seleccionar el servidor *apache2* y la configuración *dbconfig-common*.

```
apt-get install phpmyadmin
echo "Include /etc/phpmyadmin/apache.conf" >> /etc/apache2/apache2.conf
service apache2 restart
```

7. Crear una base de datos *MySQL*:

A través de *PhpMyAdmin* se debe crear una base de datos con el nombre “universando”. Además, se debe crear un usuario con todos los permisos sobre la base de datos creada y con acceso solo local.

8. Instalar *Java*:

```
apt-get install openjdk-7-jdk
```

9. Instalar *Apache Solr*:

```
cd ~
wget http://apache.rediris.es/lucene/solr/5.1.0/solr-5.1.0.tgz
tar xvzf solr-5.1.0.tgz
rm solr-5.1.0.tgz
```

10. Configurar *Apache Solr*:

Se deben transferir por SFTP los ficheros de configuración de *Solr* proporcionados. Tras esto, se debe modificar el fichero `~/solr-5.1.0/server/solr/universando/conf/data-config.xml` para indicar los datos de conexión con la base de datos *MySQL* creada.

11. Ejecutar *Apache Solr*:

```
cd ~/solr-5.1.0/server
nohup java -jar start.jar&
```

12. Instalar *MongoDB*:

```
echo "deb http://repo.mongodb.org/apt/debian wheezy/mongodb-org/3.0 main"
    | tee /etc/apt/sources.list.d/mongodb-org-3.0.list
apt-get update
apt-get install -y mongodb-org
```

13. Crear una base de datos *MongoDB*:

```
nohup mongod --dbpath ~/mongo&
mongo
use universando
db.createUser(
{
    user: "NOMBRE_DE_USUARIO",
    pwd: "CONTRASEÑA_DE_USUARIO",
    roles: [
        { role: "readWrite", db: "universando" }
    ]
})
exit
```

14. Ejecutar *MongoDB*:

```
kill $(ps -A | grep mongo | sed 's/^ *//g' | cut -d' ' -f1)
nohup mongod --auth --dbpath ~/mongo&
```

15. Instalar *Gradle*:

```
cd ~
apt-get install unzip
wget https://services.gradle.org/distributions/gradle-2.5-all.zip
unzip -d /tmp/ gradle-2.5-all.zip
rm gradle-2.5-all.zip
mv /tmp/gradle* /opt/
echo 'GRADLE_HOME=/opt/gradle-2.5' >> $HOME/.bashrc
echo 'export PATH=$PATH:$GRADLE_HOME/bin' >> $HOME/.bashrc
source $HOME/.bashrc
```

16. Crear directorio de almacenamiento de *logs*:

```
mkdir /logs
chmod -R 777 /logs
```

17. Transferir ficheros del sistema:

Se deben transferir los ficheros del sistema (vía SFTP) al directorio *~/Universando*.

18. Modificar configuración del sistema:

Se debe modificar el fichero *~/Universando/src/main/resources/application.properties* para adaptarlo a la configuración del servidor.

19. Iniciar pruebas automáticas:

```
cd ~/Universando
gradle build
```

20. Iniciar la aplicación:

```
nohup gradle bootRun&
```

21. Entrenar sistema de aprendizaje:

A través de un cliente REST (p.e. *Advanced REST Client*) se debe realizar la petición HTTP *POST /opiniones/entrenamiento*, con rol de usuario administrador.

22. (Opcional) Crear el siguiente *script* para facilitar futuros lanzamientos del sistema:

```
service apache2 start
service mysql start
cd ~/solr-5.1.0/server; nohup java -jar start.jar&
kill $(ps -A | grep mongo | sed 's/^ *//g' | cut -d' ' -f1)
cd ~/; nohup mongod --auth --dbpath ~/mongo&
cd ~/Universando; nohup gradle bootRun&
```


Anexo E

Prototipo vs. Resultado final

A continuación se muestra el prototipo completo del sistema (*wireframe*) elaborado en la fase de diseño, junto con el aspecto final obtenido al convertir dicho prototipo a las tecnologías HTML y CSS.

La figura E.1 muestra la pantalla inicial de la aplicación, donde se ha colocado un buscador de organizaciones y un listado con las mejores y peores organizaciones de la provincia y país del usuario.

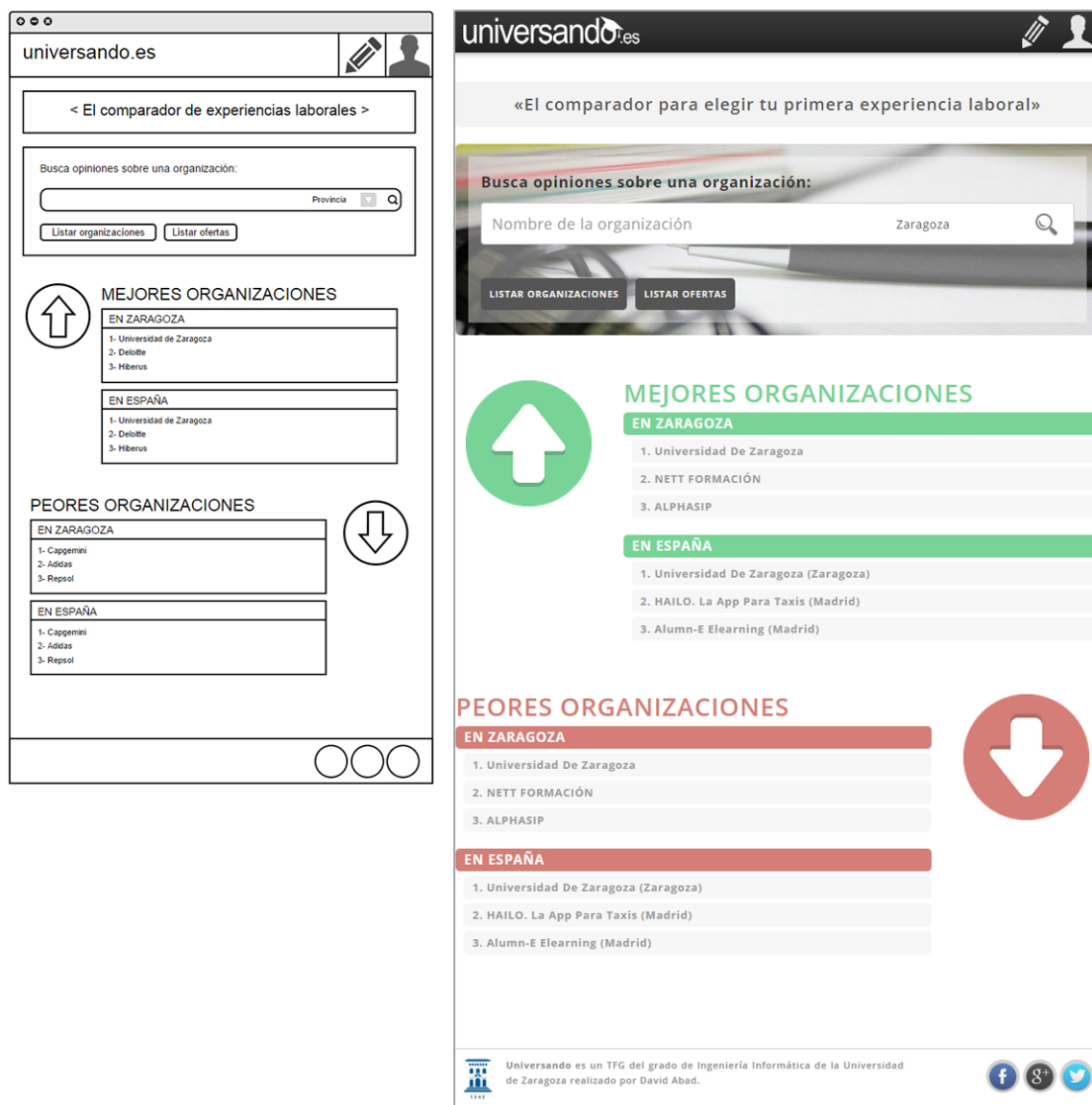


Figura E.1: Prototipo vs. Resultado final (Inicio)

En la figura E.2 se muestra un formulario desplegable desde el *header* que permite a los usuarios identificarse escribiendo su *email* y su contraseña, así como solicitar una recuperación de contraseña en caso de olvido.

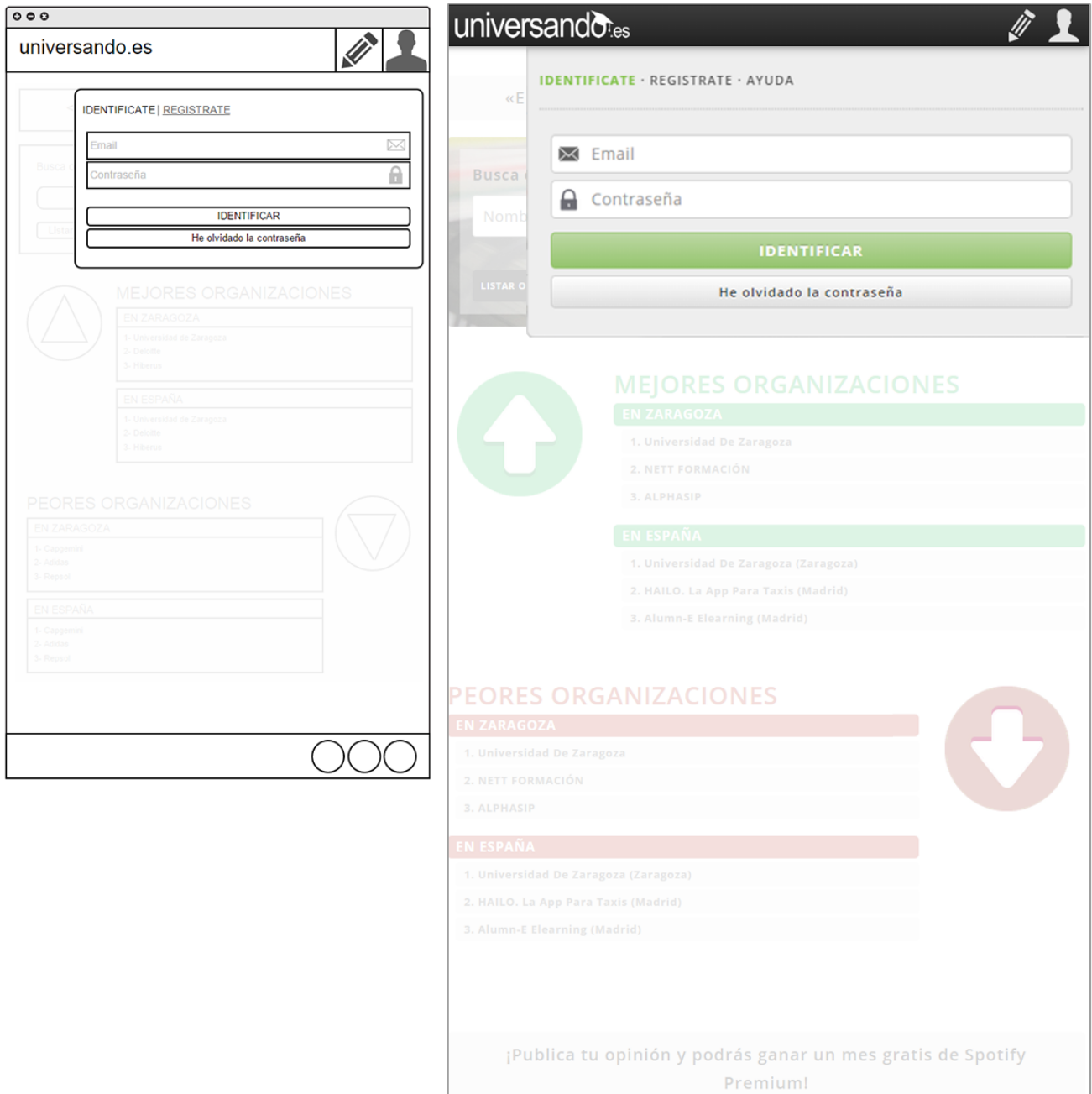


Figura E.2: Prototipo vs. Resultado final (Login)

En la figura E.3 se muestra otro formulario desplegable desde el *header*. Este formulario permite a los nuevos usuarios registrarse como universitarios u organizaciones. En el formulario se muestran diferentes campos que el usuario debe rellenar de forma obligatoria para completar su registro.

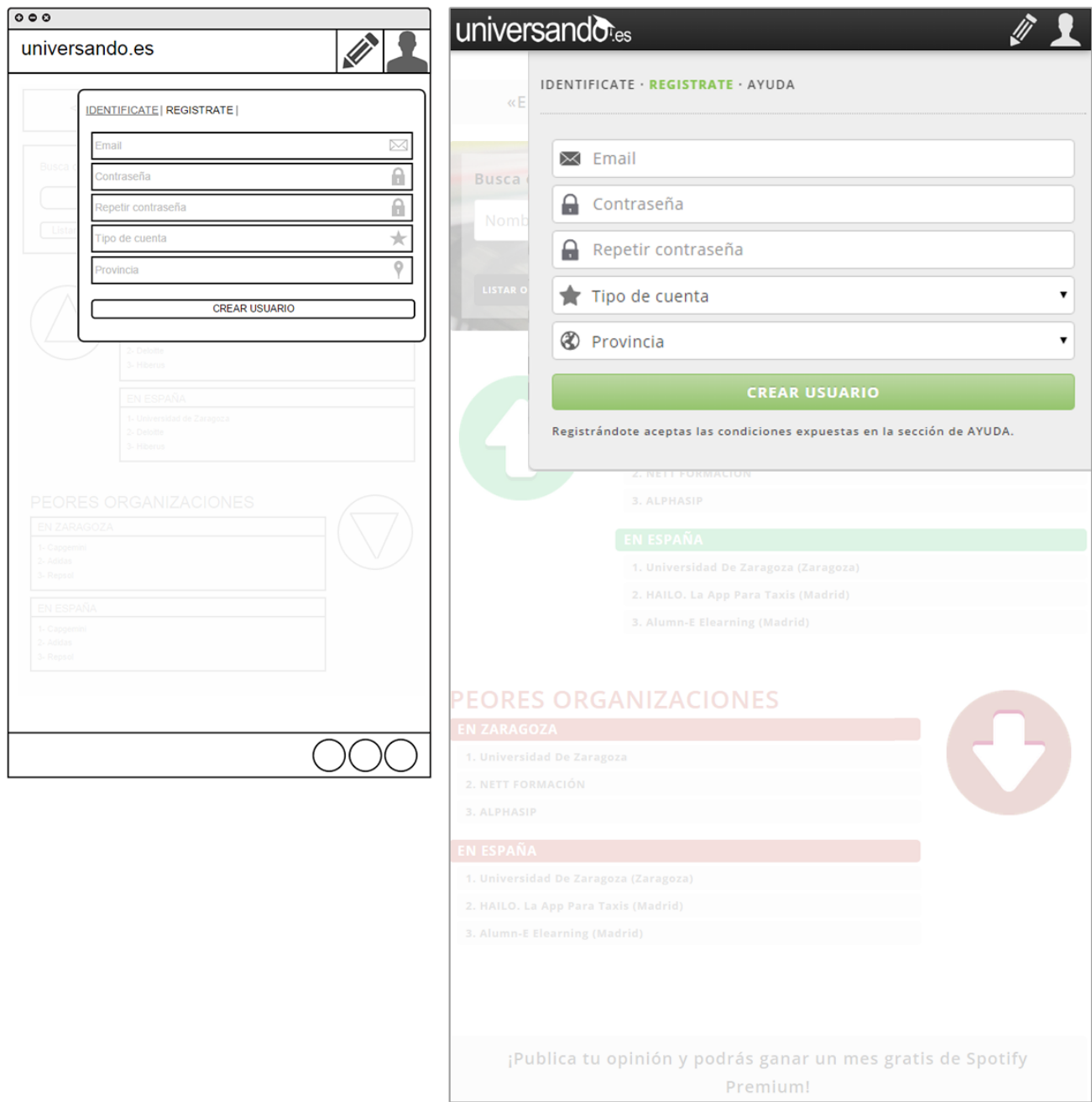


Figura E.3: Prototipo vs. Resultado final (Registro)

Cuando el usuario ya se ha identificado, puede acceder desde el *header* a un panel de notificaciones como el de la figura E.4, donde se muestran novedades relacionadas con ofertas y opiniones. Además, desde este panel se tiene acceso a la sección de “Ajustes de cuenta”.

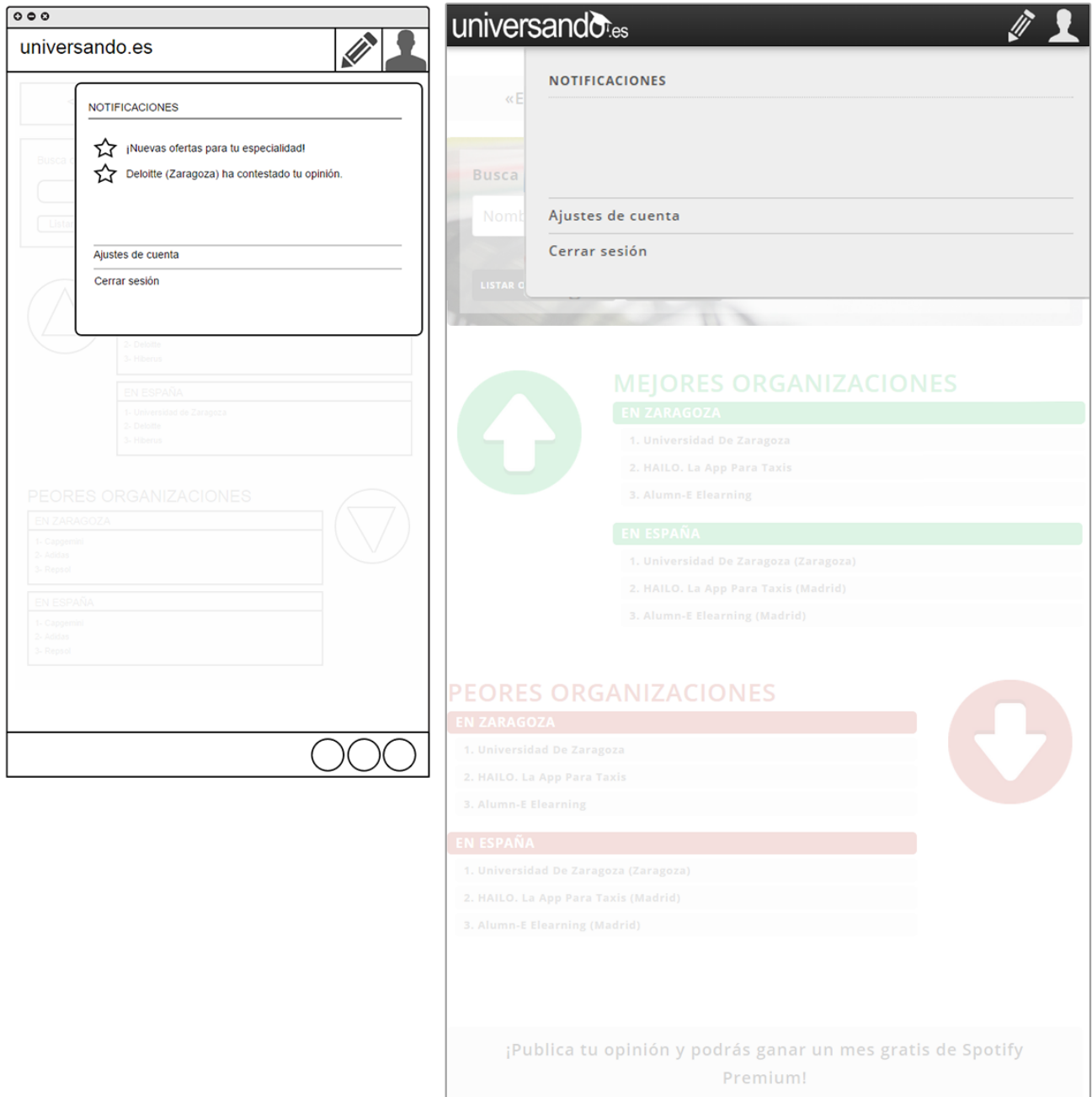


Figura E.4: Prototipo vs. Resultado final (Notificaciones)

El formulario mostrado en la figura E.5 permite a los usuarios registrados modificar la información o contraseña de su cuenta, para lo cual se deberá indicar la contraseña actual en el campo superior del formulario.

universando.es

Introduce solo la información que desees modificar:

Contraseña

Nueva contraseña

Repetir contraseña

Nombre y apellidos

Provincia

Email

ACTUALIZAR INFORMACIÓN

1- Universidad de Zaragoza

2- Deloitte

3- Hiberna

PEORES ORGANIZACIONES

EN ZARAGOZA

1- Capgemini

2- Adidas

3- Repsol

EN ESPAÑA

1- Capgemini

2- Adidas

3- Repsol

universando.es

Introduce solo la información que desees modificar:

Contraseña actual (obligatorio)

Nueva contraseña

Repetir contraseña

David Abad

Zaragoza

abad023@gmail.com

ACTUALIZAR INFORMACIÓN

Cuando tu cuenta de correo universitario sea cancelada, puedes sustituirla por tu cuenta de correo personal.

1. Universidad De Zaragoza (Zaragoza)

2. HAILO. La App Para Taxis (Madrid)

3. Alumn-E Elearning (Madrid)

PEORES ORGANIZACIONES

EN ZARAGOZA

1. Universidad De Zaragoza

2. HAILO. La App Para Taxis

3. Alumn-E Elearning

EN ESPAÑA

1. Universidad De Zaragoza (Zaragoza)

2. HAILO. La App Para Taxis (Madrid)

3. Alumn-E Elearning (Madrid)

Figura E.5: Prototipo vs. Resultado final (Ajustes de cuenta)

La figura E.6 muestra el aspecto del listado de organizaciones, que incluye los nombres y calificaciones de las organizaciones almacenadas y permite acceder a las opiniones de cada una de ellas. Este listado proporciona una serie de filtros que permiten mostrar solamente un subconjunto de las organizaciones atendiendo a su provincia, titulación y nombre, así como ordenar el listado obedeciendo a diferentes criterios.

universando.es

LISTADO DE ORGANIZACIONES

Lista de las empresas y organizaciones con opiniones de nuestros usuarios.

Provincia: Titulación:

Nombre de la empresa:

Organizaciones:

Ordenar por: Puntuación total

Organización	Puntuación total
1- Universidad de Zaragoza	10
2- Deloitte	9
3- Hiberus	8

← →

○ ○ ○

universando.es

LISTADO DE ORGANIZACIONES

Lista de las empresas y organizaciones con opiniones de nuestros usuarios.

Zaragoza Todas las titulaciones

Nombre de organización:

Organizaciones:

Ordenar por: Puntuación total

Organización	Puntuación total
UNIVERSIDAD DE ZARAGOZA - ZARAGOZA	10
NETT FORMACIÓN - ZARAGOZA	9
ALPHASIP - ZARAGOZA	8
KÜHNEL ESCUELA DE NEGOCIOS - ZARAGOZA	7

TFG: Trabajos/Proyectos de fin de grado (TFG), de fin de master (TFM), de fin de carrera (PFC)...

PRA: Prácticas en la organización

EMP: (Primeros) empleos en la organización

FEED: Feedback de la organización (La organización responde las opiniones de los usuarios, publica ofertas, etc.)

La nota global se obtiene realizando una media ponderada entre la media aritmética de TFG, PRA y EMP (85%) y FEED (15%).

Universando es un TFG del grado de Ingeniería Informática de la Universidad de Zaragoza realizado por David Abad.

f g+ t

Figura E.6: Prototipo vs. Resultado final (Lista de organizaciones)

En la figura E.7 se muestra la página que expone las diferentes puntuaciones y opiniones de una organización concreta. Estas opiniones pueden leerse completamente al pulsar el botón que acompaña a cada opinión, el cual muestra el texto a pantalla completa. La página también permite filtrar las opiniones, así como acceder a las ofertas y detalles de la organización en cuestión.

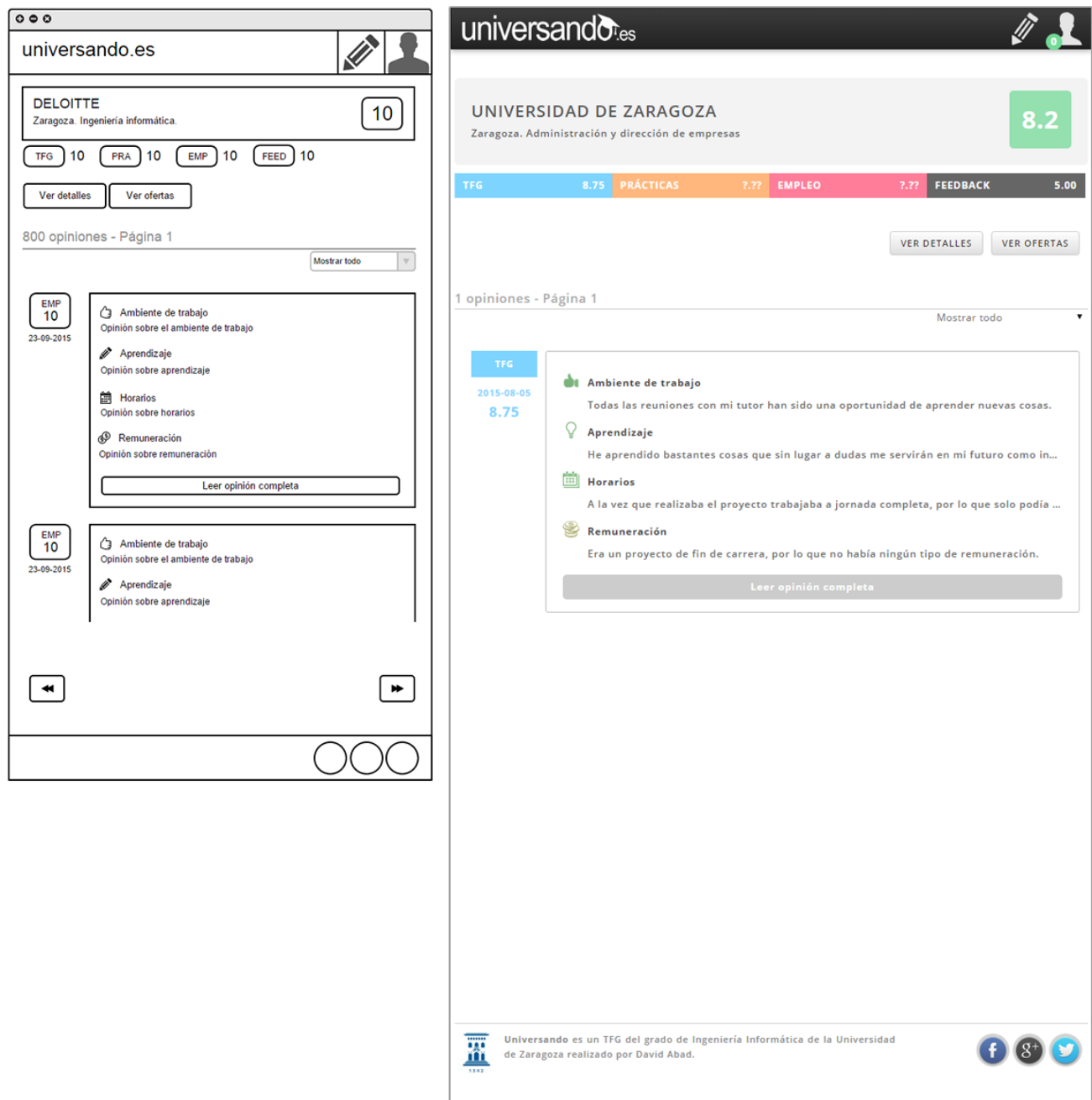


Figura E.7: Prototipo vs. Resultado final (Opiniones de organización)

Los detalles de una organización concreta se abren en un *popup* como el mostrado en la figura E.8. Estos detalles incluyen un gráfico en la parte superior que expone los datos relativos a las puntuaciones que ha obtenido la organización a lo largo del tiempo. Un segundo gráfico resume la valoración actual de la organización. Y finalmente, en la parte inferior se muestra una tabla con la valoración de la organización más detallada y mostrada según un código de colores.

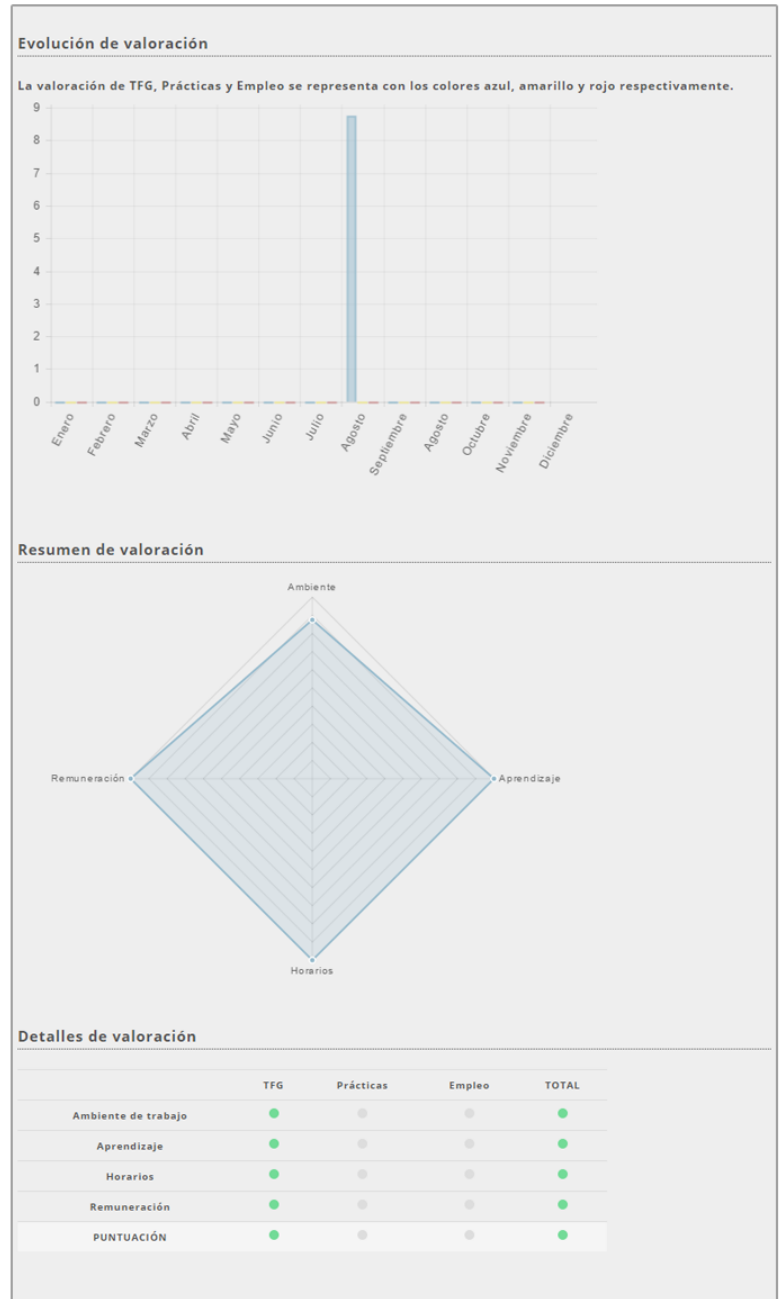
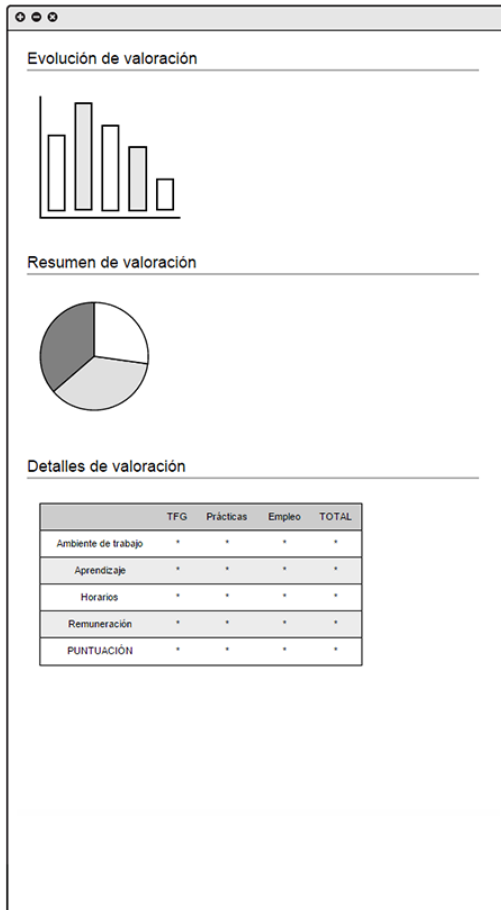


Figura E.8: Prototipo vs. Resultado final (Detalles de organización)

La página mostrada en la figura E.9 lista las diferentes ofertas disponibles, mostrando su provincia, titulación, organización, tipo, nota y título. Estas ofertas pueden ser desplegadas para acceder a una descripción más detallada. Además, los usuarios universitarios pueden mostrar un formulario para enviar su CV en cada oferta, mientras que las organizaciones pueden acceder a un listado con los usuarios que han enviado su CV.

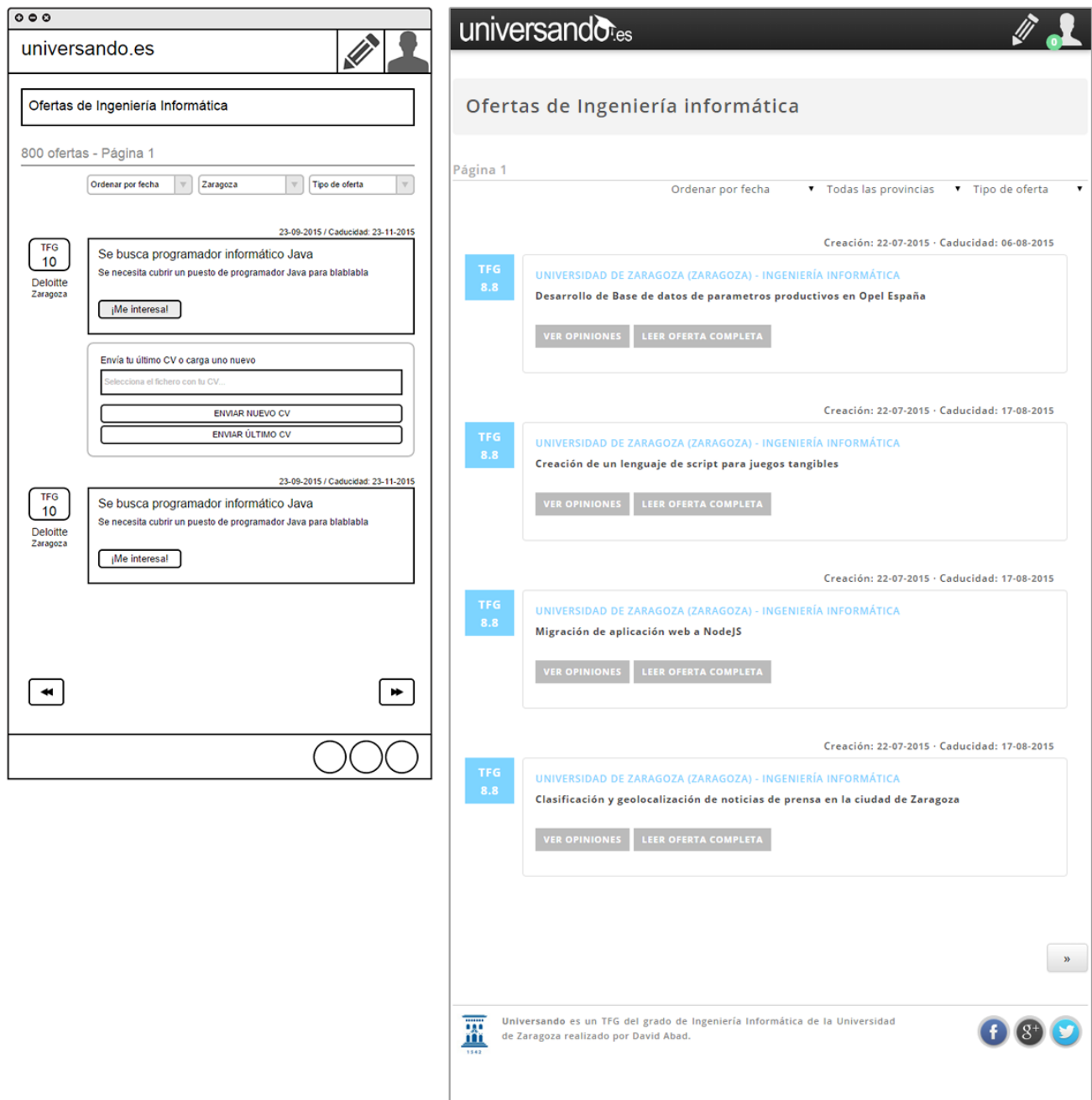


Figura E.9: Prototipo vs. Resultado final (Lista de ofertas)

En la figura E.10 se muestra el formulario al que pueden acceder los universitarios para publicar nuevas opiniones. Una vez seleccionada una provincia, el usuario podrá elegir una organización existente en la base de datos o una organización nueva. La opinión de un usuario consta de cuatro comentarios sobre el ambiente, el aprendizaje, los horarios y la remuneración. Mientras estos textos están siendo escritos, se activará uno de los iconos situados a la derecha de cada caja, mostrando la calificación calculada a partir del texto, aunque el usuario podrá modificar esta calificación si no está de acuerdo con el icono activado.

The image shows two versions of a web form for publishing an opinion on 'universando.es'. The left version is a prototype, and the right version is the final result.

Prototipo (Left):

- Header: 'universando.es' with a pencil icon and a user profile icon.
- Title: 'PUBLICA TU OPINIÓN. ¡Es anónimo!' with a subtitle: 'Si has realizado un proyecto, prácticas o has tenido un primer empleo en una...'
- Form fields:
 - 'Nombre de la organización' with a placeholder 'Empresa/Organización'.
 - 'Provincia de la organización' with a dropdown menu showing 'Provincia'.
 - 'Tipo de trabajo' with a dropdown menu showing 'Tipo de trabajo'.
- Section: 'Danos tu opinión sobre...' with the instruction 'Escribe una valoración textual y selecciona un indicador de satisfacción.'
 - 'Ambiente de trabajo': Text area with a placeholder and three smiley face icons (happy, neutral, sad).
 - 'Aprendizaje': Text area with a placeholder and three smiley face icons.
 - 'Horarios': Text area with a placeholder and three smiley face icons.
 - 'Remuneración': Text area with a placeholder and three smiley face icons.
- Buttons: 'PUBLICAR OPINIÓN' and three circular icons at the bottom right.

Resultado final (Right):

- Header: 'universando.es' with a pencil icon and a user profile icon.
- Title: 'PUBLICA TU OPINIÓN. ¡Es anónimo!' with a subtitle: 'Si has realizado un proyecto, prácticas o has tenido un primer empleo en una empresa u organización. ¡Cuéntanos qué tal!'.
- Form fields:
 - 'Provincia de la organización' with a dropdown menu showing 'Provincia'.
 - 'Tipo de trabajo' with a dropdown menu showing 'Tipo de trabajo'.
- Section: 'Danos tu opinión sobre...' with the instruction 'Escribe una valoración textual y selecciona un indicador de satisfacción.'
 - 'Ambiente de trabajo': Text area with a placeholder '¿Qué tal con tus compañeros? ¿Y con tus jefes?' and three smiley face icons.
 - 'Aprendizaje': Text area with a placeholder '¿Aprendiste cosas interesantes?' and three smiley face icons.
 - 'Horarios': Text area with a placeholder '¿Qué tal el horario? ¿Trabajabas más horas de las previstas?' and three smiley face icons.
 - 'Remuneración': Text area with a placeholder '¿Conforme con la remuneración? ¿Tu trabajo merecía más?' and three smiley face icons.
- Buttons: 'PUBLICAR OPINIÓN' and a disclaimer at the bottom: 'Una vez publicada, tu opinión no podrá ser modificada. Te recomendamos revisar el contenido de los mensajes antes de...'.

Figura E.10: Prototipo vs. Resultado final (Publicar opinión)

En la figura E.11 se muestra el formulario al que pueden acceder las organizaciones para publicar nuevas ofertas de empleo, TFG o prácticas. Todos los campos mostrados son obligatorios y el campo “caducidad” permite desplegar un calendario para facilitar la selección de una fecha.

The image displays two versions of a web form for publishing a job offer on the website 'universando.es'. The left version is a wireframe prototype, and the right version is the final implemented design.

Prototype (Left):

- Header: 'universando.es' with a pencil icon and a user profile icon.
- Title: 'PUBLICA UNA NUEVA OFERTA' with a subtitle 'Publica ofertas para la realización de TFG, prácticas o primeros empleos....'.
- Form fields: 'Nueva oferta:', 'Titulación' (dropdown), 'Tipo de trabajo' (dropdown), 'Fecha de caducidad' (text input with placeholder 'dd/mm/yyyy'), 'Descripción' (with 'Descripción corta' and 'Descripción larga' sub-headers), and a 'PUBLICAR OFERTA' button.
- Footer: Three empty circles.

Final Result (Right):

- Header: 'universando.es' with a pencil icon, a user profile icon, and a green notification bubble.
- Title: 'PUBLICA UNA NUEVA OFERTA' with a subtitle 'Publica ofertas para la realización de TFG, prácticas o primeros empleos en tu empresa/organización e incrementa tus puntos de feedback.'.
- Form fields: 'Titulación' (dropdown), 'Tipo de trabajo' (dropdown), 'Fecha de caducidad' (text input with placeholder 'dd/mm/yyyy'), 'Descripción' (with 'Descripción corta' and 'Descripción larga' sub-headers), and a green 'PUBLICAR OFERTA' button.
- Text below button: 'Una vez publicada la oferta, ésta podrá ser editada pero no eliminada. La oferta dejará de ser visible para los usuarios en la fecha indicada como "caducidad".'
- Footer: Universando logo, text 'Universando es un TFG del grado de Ingeniería Informática de la Universidad de Zaragoza realizado por David Abad.', and social media icons for Facebook, Google+, and Twitter.

Figura E.11: Prototipo vs. Resultado final (Publicar oferta)

Anexo F

Especificación del API RESTful

A continuación se especifican las diferentes operaciones que pueden realizarse utilizando la API implementada. Algunas operaciones requieren un rol de usuario concreto, por lo que se debe utilizar el servicio de *login* para identificarse como usuario. Tras ello, se debe enviar y recibir el *token* “X-CSRF-TOKEN” en la cabecera de las peticiones HTTP.

GET /cvs

Devuelve la lista de CVs del usuario (Requiere rol de Universitario)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "fecha": "2015-08-05T04:55:49.059Z",
    "usuario": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

POST /cvs

Sube un nuevo CV para el usuario (Requiere rol de Universitario)

Parámetros

1. file (formData): Fichero a subir
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "fecha": "2015-08-05T04:55:49.075Z",
    "usuario": 0,
    "idRecurso": 0,
  }
}
```

```
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  }  
}
```

DELETE /cvs/{id}

Elimina el CV indicado (Requiere rol de Administrador)

Parámetros

1. id (integer): ID del CV a eliminar

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": null  
}
```

GET /cvs/{id}

Devuelve el CV indicado (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID del CV a obtener

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": "string"  
}
```

GET /estadisticas

Devuelve estadísticas del sitio (Requiere rol de Administrador)

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": {  
    "numOpiniones": 0,  
  }  
}
```

```
"numOpinionesContestadas": 0,
"numOrganizaciones": 0,
"numUsuariosUniversitario": 0,
"numUsuariosOrganizacion": 0,
"numUsuariosOrganizacionPendientes": 0,
"numOfertasPublicadas": 0,
"numOfertasRecuperadas": 0,
"numOfertasVigentes": 0,
"numOfertasContestadas": 0,
"numContestacionesOferta": 0,
"links": [ {
    "rel": "string",
    "href": "string"
  } ]
}
```

GET /ganadores

Devuelve lista de ganadores del sorteo (Requiere rol de Administrador)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "mes": 0,
    "anno": 0,
    "email": "string",
    "nombre": "string",
    "provincia": 0,
    "provinciaNombre": "string",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

POST /ganadores

Calcula y devuelve un nuevo ganador de sorteo para el mes actual (Requiere rol de Administrador)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "mes": 0,
    "anno": 0,
```

```

    "email": "string",
    "nombre": "string",
    "provincia": 0,
    "provinciaNombre": "string",
    "idRecurso": 0,
    "links": [ {
        "rel": "string",
        "href": "string"
    } ]
}

```

GET /ganadores/ultimo

Devuelve el último ganador del sorteo

Respuesta

```

{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "mes": 0,
    "anno": 0,
    "email": "string",
    "nombre": "string",
    "provincia": 0,
    "provinciaNombre": "string",
    "idRecurso": 0,
    "links": [ {
        "rel": "string",
        "href": "string"
    } ]
  }
}

```

GET /ofertas

Devuelve lista de ofertas (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. tipo (integer): Tipo de ofertas a recuperar (0:Todas; 1:TFG; 2:Prácticas; 3:Empleo)
2. titulacion (integer): ID de la titulación de las ofertas
3. provincia (integer): ID de la provincia de las ofertas (0: Todas)
4. organizacion (integer): ID de la organización de las ofertas (0: Todas)
5. caducado (boolean): Recuperar ofertas caducadas
6. orden (integer): Orden de las ofertas (0: Fecha; 1: Nota)
7. desde (integer): Paginador “desde” (0: Mostrar todo)
8. hasta (integer): Paginador “hasta” (0: Mostrar todo)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "descripcionMini": "string",
    "descripcion": "string",
    "titulacion": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.106Z",
    "caducidad": "2015-08-05T04:55:49.106Z",
    "organizacion": 0,
    "url": "string",
    "organizacionNombre": "string",
    "notaOferta": 0,
    "titulacionNombre": "string",
    "localidad": 0,
    "localidadNombre": "string",
    "aspirantes": [ {
      "oferta": 0,
      "cv": 0,
      "usuarioNombre": "string",
      "links": [ {
        "rel": "string",
        "href": "string"
      } ]
    } ]
  } ],
  "idRecurso": 0,
  "links": [ {
    "rel": "string",
    "href": "string"
  } ]
}
```

POST /ofertas

Publica una nueva oferta (Requiere rol de Organización)

Parámetros

1. descripcionMini (string): Texto de título
 2. descripcion (string): Texto de descripción
 3. titulacion (integer): ID de titulación
 4. tipo (integer): Tipo de oferta (1:TFG; 2:Prácticas; 3:Empleo)
 5. caducidad (long): Fecha de caducidad
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
```

```

"data": {
  "descripcionMini": "string",
  "descripcion": "string",
  "titulacion": 0,
  "tipo": 0,
  "fecha": "2015-08-05T04:55:49.122Z",
  "caducidad": "2015-08-05T04:55:49.122Z",
  "organizacion": 0,
  "url": "string",
  "organizacionNombre": "string",
  "notaOferta": 0,
  "titulacionNombre": "string",
  "localidad": 0,
  "localidadNombre": "string",
  "aspirantes": [ {
    "oferta": 0,
    "cv": 0,
    "usuarioNombre": "string",
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ],
  "idRecurso": 0,
  "links": [ {
    "rel": "string",
    "href": "string"
  } ]
}

```

POST /ofertas/scrap

Obtiene nuevas ofertas de sitios web externos (Requiere rol de Administrador)

Respuesta

```

{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {}
}

```

DELETE /ofertas/{id}

Elimina la oferta indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de la oferta a eliminar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /ofertas/{id}

Devuelve la oferta indicada (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID de la oferta a recuperar
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "descripcionMini": "string",
    "descripcion": "string",
    "titulacion": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.137Z",
    "caducidad": "2015-08-05T04:55:49.137Z",
    "organizacion": 0,
    "url": "string",
    "organizacionNombre": "string",
    "notaOferta": 0,
    "titulacionNombre": "string",
    "localidad": 0,
    "localidadNombre": "string",
    "aspirantes": [ {
      "oferta": 0,
      "cv": 0,
      "usuarioNombre": "string",
      "links": [ {
        "rel": "string",
        "href": "string"
      } ]
    } ],
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

POST /ofertas/{id}

Contesta (envía CV) a la oferta indicada (Requiere rol de Universitario)

Parámetros

1. id (integer): ID de la oferta
2. cv (integer): ID del CV

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "descripcionMini": "string",
    "descripcion": "string",
    "titulacion": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.137Z",
    "caducidad": "2015-08-05T04:55:49.137Z",
    "organizacion": 0,
    "url": "string",
    "organizacionNombre": "string",
    "notaOferta": 0,
    "titulacionNombre": "string",
    "localidad": 0,
    "localidadNombre": "string",
    "aspirantes": [ {
      "oferta": 0,
      "cv": 0,
      "usuarioNombre": "string",
      "links": [ {
        "rel": "string",
        "href": "string"
      } ]
    } ],
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /ofertas/{id}

Edita la oferta indicada (Requiere rol de Organización)

Parámetros

1. id (integer): ID de la oferta
 2. body (Model): Información a editar (JSON con los datos descripcionMini, descripcion, titulacion, tipo y caducidad)
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "descripcionMini": "string",
    "descripcion": "string",
    "titulacion": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.137Z",
    "caducidad": "2015-08-05T04:55:49.137Z",
    "organizacion": 0,
    "url": "string",
    "organizacionNombre": "string",
    "notaOferta": 0,
    "titulacionNombre": "string",
    "localidad": 0,
    "localidadNombre": "string",
    "aspirantes": [ {
      "oferta": 0,
      "cv": 0,
      "usuarioNombre": "string",
      "links": [ {
        "rel": "string",
        "href": "string"
      } ]
    } ]
  },
  "idRecurso": 0,
  "links": [ {
    "rel": "string",
    "href": "string"
  } ]
}
```

GET /novedades

Devuelve lista de novedades del usuario (Requiere rol de Universitario u Organización)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "fecha": "2015-08-05T04:55:49.106Z",
    "tipo": 0,
```

```

        "oferta": 0,
        "opinion": 0,
        "organizacion": 0,
        "usuario": 0,
        "organizacionNombre": "string",
        "usuarioNombre": "string",
        "idRecurso": 0,
        "links": [ {
            "rel": "string",
            "href": "string"
        } ]
    } ]
}

```

GET /novedades/{id}

Devuelve la novedad indicada (Requiere rol de Universitario u Organización)

Parámetros

1. id (integer): ID de la novedad a obtener

Respuesta

```

{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "fecha": "2015-08-05T04:55:49.106Z",
    "tipo": 0,
    "oferta": 0,
    "opinion": 0,
    "organizacion": 0,
    "usuario": 0,
    "organizacionNombre": "string",
    "usuarioNombre": "string",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}

```

DELETE /novedades/{id}

Elimina la novedad indicada (Requiere rol de Universitario u Organización)

Parámetros

1. id (integer): ID de la novedad a eliminar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /organizaciones

Devuelve lista de organizaciones

Parámetros

1. provincia (integer): ID de provincia
2. titulacion (integer): ID de titulación
3. nombre (string): Nombre de la organización
4. orden (integer): Orden de aparición (0:Similitud del nombre; 1:Nota media; 2:Nota TFG; 3:Nota Prácticas; 4:Nota Empleo; 5: Alfabético [-1..-5: Equivalentes ascendentes])
5. desde (integer): Paginador “desde” (0: Mostrar todo)
6. hasta (integer): Paginador “hasta” (0: Mostrar todo)
7. usuario (integer): ID del usuario al que está asignada la organización

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "nombre": "string",
    "provincia": 0,
    "titulacion": 0,
    "usuario": 0,
    "nota": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "titulacionNombre": "string",
    "provinciaNombre": "string",
    "totalOpiniones": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

GET /organizaciones/{id}

Devuelve la organización indicada

Parámetros

1. id (integer): ID de organización a obtener

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "nombre": "string",
    "provincia": 0,
    "titulacion": 0,
    "usuario": 0,
    "nota": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "titulacionNombre": "string",
    "provinciaNombre": "string",
    "totalOpiniones": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

POST /organizaciones

Crea una nueva organización (Requiere rol de Administrador)

Parámetros

1. provincia (integer): ID de provincia
2. nombre (string): Nombre de la organización

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "nombre": "string",
    "provincia": 0,
    "titulacion": 0,
    "usuario": 0,
  }
}
```

```
"nota": 0,
"nota1": 0,
"nota2": 0,
"nota3": 0,
"nota4": 0,
"titulacionNombre": "string",
"provinciaNombre": "string",
"totalOpiniones": 0,
"idRecurso": 0,
"links": [ {
    "rel": "string",
    "href": "string"
  } ]
}
```

GET /organizaciones/detalles/{id}

Devuelve los detalles de la organización (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID de organización
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "notasFecha": [{}],
    "notasCategoria": [{}],
    "organizacion": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

POST /organizaciones/{id}

Asocia la organización indicada a un usuario concreto (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de organización
 2. usuario (integer): ID del usuario
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "nombre": "string",
    "provincia": 0,
    "titulacion": 0,
    "usuario": 0,
    "nota": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "titulacionNombre": "string",
    "provinciaNombre": "string",
    "totalOpiniones": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

DELETE /organizaciones/{id}

Elimina la organización indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de organización a eliminar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /provincias

Devuelve lista de provincias

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
```

```
    "cp": "string",
    "nombre": "string",
    "idRecurso": 0,
    "links": [ {
        "rel": "string",
        "href": "string"
    } ]
} ]
}
```

GET /provincias/{id}

Devuelve la provincia indicada

Parámetros

1. id (integer): ID de provincia

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "cp": "string",
    "nombre": "string",
    "idRecurso": 0,
    "links": [ {
        "rel": "string",
        "href": "string"
    } ]
  }
}
```

GET /provincias/geoloc

Devuelve la provincia en la que se encuentra el cliente

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "cp": "string",
    "nombre": "string",
    "idRecurso": 0,
    "links": [ {
        "rel": "string",
        "href": "string"
    } ]
  }
}
```

GET /opiniones

Devuelve lista de opiniones (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. tipo (integer): Tipo de opiniones a recuperar (0:Todas; 1:TFG; 2:Prácticas; 3:Empleo)
2. organizacion (integer): ID de la organización de las opiniones
3. orden (integer): Criterio de ordenación (0:Fecha; 1:Nota)
4. desde (integer): Paginador “desde” (0: Mostrar todo)
5. hasta (integer): Paginador “hasta” (0: Mostrar todo)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "ambiente": "string",
    "aprendizaje": "string",
    "horarios": "string",
    "remuneracion": "string",
    "contestacion": "string",
    "notaMedia": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.153Z",
    "organizacion": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

POST /opiniones

Publica una nueva opinión (Requiere rol de Universitario)

Parámetros

1. organizacion (integer): ID de la organización sobre la que se opina
2. organizacionnombre (integer): Nombre de organización (Solo si no se especifica ID)
3. provincia (integer): ID de la provincia a la que pertenece la organización (Solo si no se especifica ID)
4. ambiente (integer): Opinión sobre ambiente
5. aprendizaje (string): Opinión sobre aprendizaje
6. horarios (string): Opinión sobre horarios
7. remuneracion (string): Opinión sobre remuneración

8. nota1 (integer): Nota sobre ambiente (0,1,2)
9. nota2 (integer): Nota sobre aprendizaje (0,1,2)
10. nota3 (integer): Nota sobre horarios (0,1,2)
11. nota4 (integer): Nota sobre remuneración (0,1,2)
12. tipo (integer): Tipo de opinión (1:TFG; 2:Prácticas; 3:Empleo)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "ambiente": "string",
    "aprendizaje": "string",
    "horarios": "string",
    "remuneracion": "string",
    "contestacion": "string",
    "notaMedia": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.169Z",
    "organizacion": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /opiniones/baneados/{id}

Banea al usuario autor de la opinión indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de la opinión publicada por el usuario a banear
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "ambiente": "string",
    "aprendizaje": "string",
    "horarios": "string",
    "remuneracion": "string",
    "contestacion": "string",
  }
}
```

```
"notaMedia": 0,
"nota1": 0,
"nota2": 0,
"nota3": 0,
"nota4": 0,
"tipo": 0,
"fecha": "2015-08-05T04:55:49.180Z",
"organizacion": 0,
"idRecurso": 0,
"links": [ {
  "rel": "string",
  "href": "string"
} ]
}
```

GET /opiniones/clasificador

Devuelve una puntuación calculada a partir del texto dado (0:Descontento; 1:Neutro; 2:Contento)

Parámetros

1. opinion (string): Texto de la opinión a clasificar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": 0
}
```

POST /opiniones/entrenamiento

Entrena el sistema de clasificación de opiniones (Requiere rol de Administrador)

Parámetros

1. reset (integer): Elimina el entrenamiento actual y usa el fichero CSV (1) o mantiene el entrenamiento actual y utiliza las opiniones nuevas (0)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": "string"
}
```

DELETE /opiniones/{id}

Elimina la oferta indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de la opinión a eliminar
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /opiniones/{id}

Devuelve la opinión indicada (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID de la opinión a obtener
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "ambiente": "string",
    "aprendizaje": "string",
    "horarios": "string",
    "remuneracion": "string",
    "contestacion": "string",
    "notaMedia": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.194Z",
    "organizacion": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /opiniones/{id}

Emite un mensaje de respuesta para la opinión indicada (Requiere rol de Organización)

Parámetros

1. id (integer): ID de la opinión a contestar
2. body (string): Texto de contestación

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "ambiente": "string",
    "aprendizaje": "string",
    "horarios": "string",
    "remuneracion": "string",
    "contestacion": "string",
    "notaMedia": 0,
    "nota1": 0,
    "nota2": 0,
    "nota3": 0,
    "nota4": 0,
    "tipo": 0,
    "fecha": "2015-08-05T04:55:49.197Z",
    "organizacion": 0,
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

GET /titulaciones/{id}

Devuelve la titulación indicada

Parámetros

1. id (integer): ID de la titulación a obtener

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "nombre": "string",
    "keyword": "string",
    "idRecurso": 0,
  }
}
```

```
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  }  
}
```

GET /titulaciones

Devuelve lista de titulaciones

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": [ {  
    "nombre": "string",  
    "keyword": "string",  
    "idRecurso": 0,  
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  } ]  
}
```

POST /titulaciones

Crea una nueva titulación (Requiere rol de Administrador)

Parámetros

1. nombre (string): Nombre de la nueva titulación
 2. keyword (string): Keyword de la nueva titulación
-

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": {  
    "nombre": "string",  
    "keyword": "string",  
    "idRecurso": 0,  
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  }  
}
```

DELETE /titulaciones/{id}

Elimina la titulación indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de la titulación a eliminar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /universidades

Devuelve lista de universidades

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "nombre": "string",
    "dominio": "string",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

POST /universidades

Crea una nueva universidad (Requiere rol de Administrador)

Parámetros

1. nombre (string): Nombre de la nueva universidad
2. dominio (string): Dominio de la nueva universidad

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
```

```
"nombre": "string",
"dominio": "string",
"idRecurso": 0,
"links": [ {
    "rel": "string",
    "href": "string"
} ]
}
```

DELETE /universidades/{id}

Elimina la universidad indicada (Requiere rol de Administrador)

Parámetros

1. id (integer): ID de la universidad a eliminar
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /universidades/{id}

Devuelve la universidad indicada

Parámetros

1. id (integer): ID de la universidad a obtener
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "nombre": "string",
    "dominio": "string",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

POST /usuarios

Crea un nuevo usuario (Requiere rol de ANÓNIMO)

Parámetros

1. email (string): Email del usuario
2. pass (string): Contraseña del usuario
3. provincia (integer): ID de la provincia del usuario
4. tipo (integer): Tipo de usuario
5. nombre (string): Nombre del usuario
6. titulacion (integer): ID de la titulación del usuario

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.209Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /usuarios/activados/{id}

Activa el usuario indicado (Requiere rol de Administrador)

Parámetros

1. id (integer): ID del usuario a activar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
```



```
"tipo": 0,
"titulacion": 0,
"universidad": 0,
"provincia": 0,
"nombre": "string",
"email": "string",
"lastLogin": "2015-08-05T17:42:02.216Z",
"idRecurso": 0,
"links": [ {
  "rel": "string",
  "href": "string"
} ]
}
```

PUT /usuarios/activados/{id}/{token}

Activa la cuenta del usuario indicado

Parámetros

1. id (integer): ID del usuario a activar
2. token (string): Token de activación

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.220Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /usuarios/baneados/{id}

Banea el usuario indicado (Requiere rol de Administrador)

Parámetros

1. id (integer): ID del usuario a banear

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.223Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

DELETE /usuarios/{id}

Elimina el usuario indicado (Requiere rol de Administrador)

Parámetros

1. id (integer): ID del usuario a eliminar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": null
}
```

GET /usuarios/email/{email}

Devuelve el usuario indicado a partir de su email (Requiere rol de Universitario, Organización o Admin.)

Parámetros

1. email (string): Email del usuario a recuperar

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.227Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

PUT /usuarios/olvidados/{email}/

Solicita recuperar la contraseña del usuario indicado (Requiere rol de ANÓNIMO)

Parámetros

1. email (string): Email del usuario
-

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.230Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  }
}
```

GET /usuarios/pendientes

Devuelve el listado de usuarios del tipo organización que están pendientes de activar (Requiere rol de Admin)

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": [ {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.233Z",
    "idRecurso": 0,
    "links": [ {
      "rel": "string",
      "href": "string"
    } ]
  } ]
}
```

PUT /usuarios/{id}/{token}

Modifica la contraseña del usuario indicado (Requiere rol de ANÓNIMO)

Parámetros

1. id (integer): ID del usuario a modificar
2. token (string): Token de modificación
3. body (string): Nueva contraseña

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": {
    "activado": 0,
    "ban": 0,
    "tipo": 0,
    "titulacion": 0,
    "universidad": 0,
    "provincia": 0,
    "nombre": "string",
    "email": "string",
    "lastLogin": "2015-08-05T17:42:02.247Z",
    "idRecurso": 0,
  }
}
```

```
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  }  
}
```

PUT /usuarios/{id}

Modifica la información del usuario indicado (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID del usuario a modificar
 2. body (Model): JSON con email, pass, passOriginal, provincia y nombre
-

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": {  
    "activado": 0,  
    "ban": 0,  
    "tipo": 0,  
    "titulacion": 0,  
    "universidad": 0,  
    "provincia": 0,  
    "nombre": "string",  
    "email": "string",  
    "lastLogin": "2015-08-05T17:42:02.243Z",  
    "idRecurso": 0,  
    "links": [ {  
      "rel": "string",  
      "href": "string"  
    } ]  
  }  
}
```

GET /usuarios/{id}

Devuelve el usuario indicado (Requiere rol de Universitario, Organización o Administrador)

Parámetros

1. id (integer): ID del usuario a recuperar
-

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",
```

```
"data": {
  "activado": 0,
  "ban": 0,
  "tipo": 0,
  "titulacion": 0,
  "universidad": 0,
  "provincia": 0,
  "nombre": "string",
  "email": "string",
  "lastLogin": "2015-08-05T17:42:02.239Z",
  "idRecurso": 0,
  "links": [ {
    "rel": "string",
    "href": "string"
  } ]
}
```

POST /login

Identifica al usuario mediante username y password. Puedes identificarte con rol anónimo con los datos anonimo:anonimo

Parámetros

1. username (string): Email del usuario
2. password (string): Contraseña del usuario

Respuesta

```
{}
```

GET /login/info

Devuelve la información del usuario identificado.

Respuesta

```
{
  "code": 0,
  "status": "string",
  "message": "string",
  "data": []
}
```

GET /logout

Desconecta al usuario

Respuesta

```
{  
  "code": 0,  
  "status": "string",  
  "message": "string",  
  "data": null  
}
```