



Red Social de Ayuda a Mayores RESOAM

Memoria de Proyecto Final de Grado

Grado Multimedia

Desarrollo de aplicaciones interactivas

Antonio Jiménez Mahillo

Consultor: Kenneth Capseta Nieto
Profesor: Carlos Casado Martínez
Fecha de entrega: 18/06/2017

Créditos/Copyright

Esta obra (RESOAM) está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)

Todas las imágenes utilizadas en la aplicación web RESOAM corresponden al [Dominio Público](#) y se han obtenido de la web de pixabay (<https://pixabay.com>).



Apache struts 2
Apache License 2.0
<http://www.apache.org/licenses/LICENSE-2.0>

Apache Tiles 3
Apache License 2.0
<http://www.apache.org/licenses/LICENSE-2.0>

Bootstrap v3.3.7
Copyright 2011-2016 Twitter, Inc.
Licensed under the MIT license
<https://github.com/twbs/bootstrap/blob/master/LICENSE>

Hibernate 4. 0.1.Final
LGPL 2.1
<http://hibernate.org/community/license/>

jQuery v1.12.4
Copyright jQuery Foundation
<http://jquery.org/license/>

Oracle MySql 5.5
<https://www.mysql.com/about/legal/licensing/oem/>

OpenStreetMap
<https://www.openstreetmap.org/copyright/en>

Las imágenes que se presentan en este documento se reproducen acogiéndose al derecho de cita o reseña (art. 32 de la LPI - Ley de Propiedad Intelectual), y están, por lo tanto, excluidas por defecto de la licencia de este documento. Además, el uso del mismo tiene exclusivamente fines académicos.

Abstract

El objetivo de este proyecto es desarrollar una plataforma web interactiva con soporte multi-idioma de carácter social con la tecnología Java EE y Struts 2, que permita poner en contacto a voluntarios con personas de avanzada edad en riesgo de exclusión social. Los acuerdos entre ambos se realizan sobre un mapa dinámico en Internet que muestra la ubicación física de cada uno de ellos; para proporcionar la funcionalidad de georreferenciación sobre los mapas de OpenStreetMap se utiliza la librería Leaflet.

Para facilitar el acceso a la aplicación y favorecer la movilidad de los voluntarios se emplea un diseño responsivo basado en Bootstrap 3, que permite que las páginas se adapten automáticamente al tamaño de la pantalla del dispositivo. Toda la comunicación entre el cliente y el servidor se realiza de forma asíncrona con AJAX y jQuery para evitar recargar las páginas y mejorar la experiencia de usuario.

El desarrollo del sistema es modular y está basado en tres capas: presentación, negocio y datos. Estas capas tienen un bajo nivel de acoplamiento lo que permite que una capa pueda ser reemplazada rápidamente sin afectar al resto. Los datos que generan los usuarios se almacenan en una base de datos ágil como MySQL y se accede a ellos con el framework de Hibernate que además se encarga de la persistencia de los datos.

Dado que se intercambian datos personales, horarios y ubicaciones resulta imprescindible contar con el apoyo de una Administración Pública que se encargue de la gestión y la custodia de los datos al tiempo que aporta la seguridad y confianza necesaria para los usuarios.

Palabras clave: voluntarios, mayores, ayuda, red social, abandono, ancianos, tercera edad, trabajo final de grado, resoam.

Abstract (English version)

The objective of this project is to develop an interactive multi-language web platform with social support with Java EE technology and Struts 2, which allows to put in touch volunteers with elderly people at social exclusion risk. The agreements between both are made on a dynamic map on the Internet that shows the physical location of each one of them. The georeferencing functionality over the OpenStreetMap maps is provided by Leaflet javascript library.

To facilitate application access and to promote the mobility of volunteers, a Bootstrap 3 based responsive design is used. This design allows pages to adapt automatically at the device screen size. All communication between the client and the server is done asynchronously with AJAX and jQuery to avoid reloading the pages and improving the user experience.

The development of the system is modular and is based on three layers: presentation, business and data. These layers have a low level of coupling which allows one layer to be replaced quickly without affecting the rest. The data generated by the users are stored in MySQL and is accessed by Hibernate framework that also handles the persistence of the data.

Due to personal data, schedules and locations are exchanged, it is indispensable to have the support of a Public Administration that is responsible for the management and custody of the data. In addition, also may provide the necessary security and confidence for the users.

Keywords: volunteers, elderly, help, social network, abandonment, seniors, final grade work, resoam.

Notaciones y convenciones

Este documento utiliza exclusivamente las fuentes tipográficas Arial y Tahoma en color negro sobre fondo blanco con sutiles variaciones de tamaño y estilo.

Título de primer nivel Arial 21 puntos negrita.

Título de segundo nivel Arial 16,1 puntos negrita.

Título de tercer nivel Arial 14 puntos negrita.

Contenido Arial 11 puntos normal.

Citas Arial 11 puntos cursiva.

Cabecera de tabla Arial 11 puntos negrita.

Celda de una tabla Arial 11 puntos normal.

Cabecera y pie de página Arial 10 puntos cursiva.

Pie de tabla y de figura Arial 10 puntos normal.

Nota al pie Arial 9 puntos cursiva.

Código fuente y comandos Tahoma 9 puntos

Índice

1.Introducción.....	9
2.Descripción.....	11
3.Objetivos.....	14
3.1.Principales.....	14
3.2.Secundarios.....	14
4.Marco teórico.....	15
5.Contenidos.....	17
6.Metodología.....	21
7.Arquitectura de la aplicación.....	22
8.Plataforma de desarrollo.....	25
9.Planificación.....	27
10.Proceso de trabajo.....	29
10.1.Análisis global del proyecto.....	29
10.2.Diseño de la base de datos.....	30
10.3.Diseño del flujo de navegación por las pantallas.....	31
10.4.Elaboración de los wireframes.....	31
10.5.Instalación y configuración del servidor de pruebas.....	32
10.6.Instalación y configuración del entorno de desarrollo.....	32
10.7.Definición de la estructura del proyecto.....	32
10.8.Desarrollo del código fuente.....	32
10.9.Integración de los componentes.....	32
10.10.Pruebas.....	33
10.11.Revisión final.....	33
10.12.Empaquetado.....	33
10.13.Implantación.....	33
11.Diagramas UML.....	34
12.APIs utilizadas.....	41
13.Prototipos.....	44
13.1.Lo-Fi.....	44
13.2.Hi-Fi.....	48
14.Perfiles de usuario.....	52
15.Usabilidad.....	53
16.Seguridad.....	54
17.Test.....	56
18.Requisitos de instalación.....	57
19.Instrucciones de instalación.....	60
20.Instrucciones de uso.....	62
20.1.Acceso.....	62
20.2.Registro.....	63
20.3.Identificación.....	64

20.4.Acuerdos.....	66
21.Presupuesto.....	68
22.Análisis de mercado.....	69
22.1.Objetivos.....	70
22.2.Estrategias.....	70
23.Viabilidad.....	71
24.Bugs.....	72
25.Proyección a futuro.....	73
26.Conclusiones.....	74
Anexo 1. Entregables del proyecto.....	75
Anexo 2. Código fuente.....	76
Anexo 3. Librerías/Código externo.....	81
Anexo 4. Glosario.....	83
Anexo 5. Bibliografía.....	85
Anexo 6. Vita.....	89

Figuras y tablas

Índice de figuras

Imagen 1: Edad media de la población española.....	15
Imagen 2: Ubicación del menú en función de la anchura del dispositivo.....	18
Imagen 3: Diagrama de capas/niveles de la aplicación.....	22
Imagen 4: Diagrama de la arquitectura de la aplicación.....	24
Imagen 5: Pavilion dv6.....	25
Imagen 6: Orange Pi Plus.....	25
Imagen 7: Diagrama Gantt del proyecto.....	27
Imagen 8: Historias de usuario.....	29
Imagen 9: Modelo físico de la base de datos.....	30
Imagen 10: Mapa de configuración para tiles.....	31
Imagen 11: Mapa de navegación.....	31
Imagen 12: Empaquetando el proyecto con Maven desde Eclipse.....	33
Imagen 13: Diagrama de casos de uso.....	34
Imagen 14: Diagrama de clases de negocio.....	35
Imagen 15: Diagrama de paquetes de negocio.....	35
Imagen 16: Diagrama de paquetes de persistencia.....	36
Imagen 17: Diagrama de actividad alta solicitud voluntario.....	36
Imagen 18: Diagrama de actividad solicitud alta demandante.....	37
Imagen 19: Diagrama de actividad realizar acuerdo demandante.....	38
Imagen 20: Diagrama de secuencia para las peticiones.....	39
Imagen 21: Diagrama de despliegue.....	39
Imagen 22: Modelo entidad-relación.....	40
Imagen 23: Página principal.....	44
Imagen 24: Página de alta de solicitud para voluntarios.....	45
Imagen 25: Página de alta de solicitud para demandantes.....	45
Imagen 26: Página del tablón/dashboard.....	46
Imagen 27: Página de búsqueda de acuerdos.....	46
Imagen 28: Página genérica correspondiente a una opción de menú.....	47
Imagen 29: Página principal (Hi-Fi).....	48
Imagen 30: Página de alta de solicitud para voluntarios (Hi-Fi).....	49
Imagen 31: Página de alta de solicitud para demandantes (Hi-Fi).....	50
Imagen 32: Página del tablón/dashboard (Hi-Fi).....	50

Imagen 33: Página de búsqueda de acuerdos (Hi-Fi).....	51
Imagen 34: Página genérica correspondiente a una opción de menú (Hi-Fi).....	51
Imagen 35: Consola web de administración Apache tomEE.....	61
Imagen 36: Selección de idioma.....	62
Imagen 37: Detección de errores en la primera fase de validación.....	63
Imagen 38: Detección de errores en la segunda fase de validación.....	64
Imagen 39: Formulario de identificación de usuario.....	65
Imagen 40: Funcionalidades limitadas.....	65
Imagen 41: Búsqueda de voluntarios.....	66
Imagen 42: Selección de la fecha del acuerdo.....	67

Índice de tablas

Tabla 1: Hardware de la plataforma de desarrollo.....	25
Tabla 2: Software de la plataforma de desarrollo.....	26
Tabla 3: Desglose de tareas de los sprints.....	28
Tabla 4: Leyenda de iconos en función de la distancia.....	67
Tabla 5: Presupuesto de realización del proyecto.....	68
Tabla 6: Diagnóstico DAFO.....	69

1. Introducción

Deberíamos sentirnos afortunados por vivir en una época que ha supuesto una auténtica revolución de la sociedad, la era digital, la era de Internet. Los avances tecnológicos que se producen cada día son abrumadores, lo que ha permitido que personas de todos los estratos sociales puedan disfrutar de elementos como el teléfono móvil, el ordenador portátil, la tableta digital, e incluso el smartwatch, que ya se han convertido en habituales e incluso en amigos inseparables para la mayoría de nosotros.

La proliferación de Internet ha cambiado la forma en la que nos relacionamos con los demás bien sea por ocio o trabajo. Podemos comprar comida sin salir de casa, e incluso ir de rebajas a una tienda online que se encuentra en el otro lado del mundo y comprar unas zapatillas, más importante quizás, ya es viable realizar una gran cantidad de trámites con la Administración Pública y nuestro DNI electrónico sin necesidad de sufrir largas esperas, incluso solicitar una cita médica desde el teléfono móvil mientras viajamos en tren.

La forma de relacionarnos con las máquinas ha cambiado, ahora nos comunicamos con ellas a través de interfaces gráficas intuitivas y agradables que nos hacen la vida más sencilla. La expansión del software libre originada por movimientos como el CopyLeft y el uso de licencias Creative Commons ha permitido a todo el mundo acceder a contenido de una forma legal que años atrás parecía impensable.

Hoy en día, existen aplicaciones para casi todo lo que podamos imaginar, basta con realizar una búsqueda por Internet para encontrarlo. Se han realizado grandes esfuerzos en el sector del desarrollo informático de aplicaciones para mejorar la accesibilidad de las mismas y permitir su uso a personas con algún tipo de incapacidad. Sin embargo, he observado que estamos dejando de lado a las personas mayores y considero que es un craso error, por este motivo decidí que este trabajo giraría en torno a esta idea.

Todo el mundo quiere llegar a viejo con la mejor salud posible y ser capaz de hacer lo que ha venido haciendo durante toda su vida. Aunque la edad no perdona y, siempre llega el momento en el que podamos encontrarnos solos y necesitemos ayuda. Para estas situaciones la Red Social de Ayuda a Mayores (RESOAM en adelante) se proyecta como una aplicación web que permite que nuestros mayores sean capaces de gestionar y solicitar ayuda de forma sencilla en Internet.

El interés del proyecto reside en poner al alcance de estas personas un servicio de ayuda social gratuito prestado por voluntarios de forma desinteresada y gestionado por un organismo público que se encargue de revisar físicamente las solicitudes y proporcione mayor grado de seguridad y confidencialidad sobre la información. Ya que los datos recogidos se utilizarán con el único fin con el que se han solicitado y no se facilitarán a ninguna empresa.

En conclusión, RESOAM no puede ser una aplicación más, debe convertirse en la aplicación de referencia en su nicho de mercado y ser la semilla de un movimiento social que se vaya extendiendo por toda la geografía española, al tiempo que permite concienciar a los ciudadanos sobre la importancia de este tipo de iniciativas y solicita su colaboración desinteresada. Hay que luchar por conseguir un Internet por y para las personas.

2. Descripción

Este proyecto se emprende con un marcado fin social ya que no existe ánimo de lucro en su explotación. Además, está orientado a uno de los estratos más desfavorecidos de la sociedad, las personas de avanzada edad que padecen problemas de severos de salud y, no pueden realizar normalmente sus tareas cotidianas si no es con ayuda externa. Existen otras aplicaciones similares en el mercado pero son de pago y las gratuitas ofrecen muy pocos servicios; es por ello que RESOAM es realmente diferente y pretende ser un referente para el sector y favorecer su expansión por todo el territorio nacional.

El verdadero objetivo del proyecto es conseguir acercar y poner de acuerdo a estas personas necesitadas con otras personas que están dispuestas a ayudarlas de manera desinteresada. Básicamente la aplicación debe actuar como un intermediario que permita salvar las barreras de la distancia y la movilidad. La forma de realizar estos acuerdos es mediante un mapa visual que se puede actualizar y permite mostrar la ubicación de voluntarios y demandantes con ciertas limitaciones. Un demandante solo puede visualizar a los voluntarios dentro de un área de influencia, mientras que los voluntarios pueden visualizar a todos los demandantes de una zona.

Para realizar el desarrollo y el seguimiento del proyecto durante todas sus fases se utiliza una metodología ágil como scrum, ya que así se pueden concentrar los esfuerzos en el desarrollo de software en lugar de generar demasiada documentación como ocurre con otras metodologías. El proyecto consiste en una aplicación web 2.0 de diseño responsivo y asíncrona que sea accesible y utilizable por personas que no están acostumbradas a utilizar Internet. Por este motivo la interfaz de usuario y en especial los formularios de introducción de datos son claros y sencillos.

El sistema está organizado en tres capas: presentación, negocio y datos. La primera es la que se encarga de gestionar el aspecto visual de la aplicación, de mostrar los datos proporcionados por el servidor y enviar los datos introducidos por el usuario a la capa de negocio. La segunda es la que se encarga de la lógica de la aplicación, aunque se realizan validaciones sobre los datos en la primera capa a nivel de cliente, es en esta capa donde se vuelven a revisar los datos introducidos y se decide su idoneidad para integrarse en el sistema. Si son idóneos esta capa se comunica con la capa de datos. La tercera es la que se encarga de gestionar la persistencia de los datos y actúa una envoltorio para aislar al programador de la creación de los objetos y la elaboración de las consultas a base de datos.

La tecnología empleada es Java EE junto con el framework de desarrollo de aplicaciones web Apache Struts 2, que sigue el patrón de diseño Modelo-Vista-Controlador -MVC-. Además se utilizan otros componentes como Apache Tiles 3 para la estructuración de las páginas JSP y la librería Log4j también de Apache para la generación de los archivos de log. En menor medida también se utiliza la librería gson de Google que permite serializar y deserializar objetos y la librería de junit para realizar los test unitarios.

Los mapas que se utilizan son proporcionados por el proyecto de mapas libres OpenStreetMap. Aunque no están al nivel de los proporcionados por Google Maps resultan una buena elección ya que no tienen las restricciones de uso por accesos que impone Google -si una aplicación sobrepasa un determinado número de peticiones diarias de mapas debe abonar a Google una pequeña cuota-. Para el control y la gestión tanto de los mapas como de los elementos que aparecen sobre él se utiliza la librería javascript Leaflet. Existen otras librerías más sofisticadas para gestionar los mapas pero esta es sencilla y fácil de integrar..

El diseño responsivo es gestionado por el framework Bootstrap 3 que está compuesto por archivos en lenguaje javascript, páginas de estilo y archivos de fuentes. Para las páginas de la aplicación se utilizan algunos de los diseños que el propio framework utiliza como ejemplo, pero adaptados a las características de la aplicación. Para la validación de los formularios desde la máquina cliente se utiliza la librería jQuery junto con una serie de plugin o añadidos que aportan funcionalidad extra al framework.

Para mejorar la experiencia de usuario la aplicación utiliza AJAX para realizar peticiones al servidor y mostrar los datos recibidos por pantalla sin necesidad de recargar la página. Esta funcionalidad se vuelve especialmente útil cuando se accede desde un dispositivo móvil, ya que se reduce el consumo de datos y se agiliza el rendimiento de la aplicación.

La configuración de las dependencias del proyecto y el empaquetado del código fuente para el despliegue en el servidor web se realiza con la herramienta Maven. En la raíz del proyecto existe un archivo pom.xml que almacena la configuración. En este se pueden definir perfiles de compilación condicional para diversos entornos.

El aspecto de la seguridad es fundamental y proporcionar información abiertamente sobre cuando alguien está o no en su domicilio es un tema delicado; especialmente, si se trata de personas mayores que en la mayoría de las ocasiones suelen vivir solas. Por lo tanto, para lograr el éxito de

la plataforma y conseguir que tanto voluntarios como demandantes se inscriban, resulta imprescindible que la aplicación se encuentre bajo la protección de una Administración Pública que se encargue de velar por la custodia y la confidencialidad de la información, al mismo tiempo que proporcione un nivel de confianza y seguridad adecuado para los usuarios.

Es recomendable que los accesos a la aplicación se realicen exclusivamente con el protocolo HTTPS para asegurar que el intercambio de mensajes entre el equipo cliente y el servidor esté cifrado. Así, aunque un intruso lograra interceptar alguno de los paquetes enviados le resultaría imposible descifrarlo. Además, a nivel de usuario resulta esencial que este pueda consultar desde el navegador web el certificado digital del servidor al que se quiere conectar para comprobar que realmente se trata del servidor indicado y no de un intento de suplantación de identidad por parte de un delincuente informático.

3. Objetivos

Desarrollo de una plataforma interactiva en Internet que ofrezca un servicio gratuito de pacto de acuerdos entre voluntarios y demandantes, para ayudar a las personas de avanzada edad que por diversas circunstancias pueden encontrarse en riesgo de exclusión social.

3.1. Principales

- Evitar la exclusión social, se pretende conseguir que los ancianos con menos recursos económicos y sin ayuda familiar caigan en el autoaislamiento y terminen formando parte de alguno de los colectivos marginados por la sociedad. Además se ofrecerá una nueva oportunidad a aquellos que ya se encuentran marginados para luchar por su reinserción social.
- Mejorar la calidad de vida de las personas dependientes, para ello, los voluntarios pueden prestar ayuda no solo física sino también psicológica; ya que en muchos casos lo único que necesitan los mayores es algo de atención y compañía que ayude a mitigar su soledad y les haga sentir que siguen formando parte de la sociedad.

3.2. Secundarios

- Ofrecer un servicio de calidad que permita que una persona mayor pueda obtener ayuda con tan solo unos clics de ratón.
- Reducir los costes del servicio ya que el verdadero trabajo físico lo realizan los voluntarios de manera desinteresada.

4. Marco teórico

Desde finales del siglo XX y especialmente desde comienzos del siglo XXI, la evolución tecnológica ha favorecido la aparición en el mercado de una gran cantidad de dispositivos electrónicos. Los teléfonos móviles, las tabletas y los relojes inteligentes se han convertido en elementos de uso cotidiano para los sectores más jóvenes de la sociedad. Sin embargo, no todo es positivo y este proceso evolutivo no está exento de detractores.

Las personas mayores -con más problemas para adaptarse al movimiento tecnológico- han perdido su papel en la sociedad. La juventud parece no comprender que ellos al final también envejecerán hasta convertirse en miembros del grupo social al que rechazan. Como norma general, los jóvenes suelen considerar a las personas mayores como el lastre que les impide disfrutar del tiempo libre.

Si miramos hacia atrás podremos observar que desde el origen de las civilizaciones la figura del anciano era un símbolo de sabiduría y el referente para todos los miembros de la tribu. Infundía respeto y estaba bien considerado dentro de la sociedad. Nadie habría podido imaginar que en los últimos cincuenta años la situación hubiese cambiado tan drásticamente. Actualmente, cuando una persona se hace mayor y deja de ser útil para la sociedad, comienza una guerra por la supervivencia que no siempre termina bien.

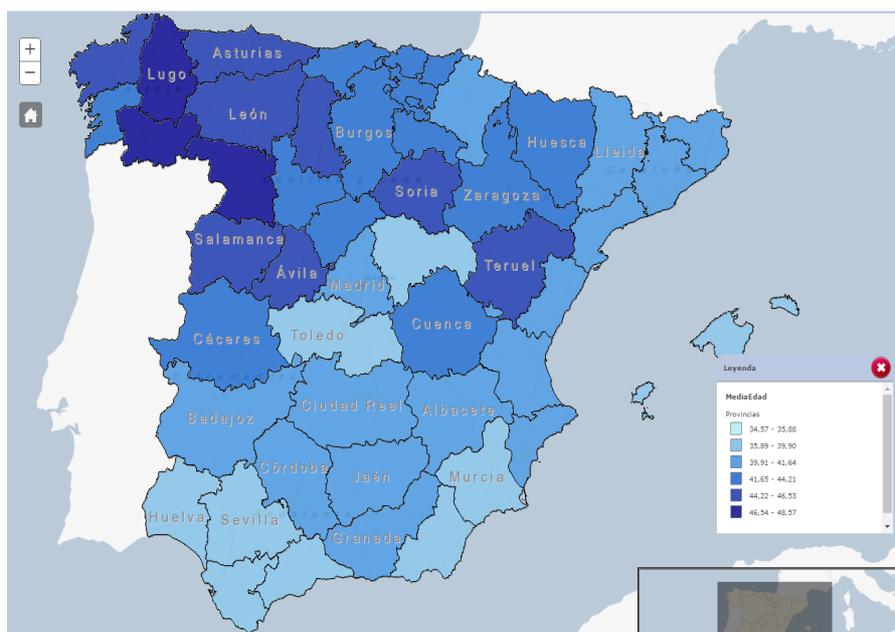


Imagen 1: Edad media de la población española (Fuente: INE)

El edadismo se ha convertido en una de las causas más importantes de discriminación por detrás del racismo y del sexismo. Se define como una forma de discriminación hacia las personas por motivo de edad. Las pirámides de población de los países desarrollados están invertidas, cada año transcurrido disminuye el número de nacimientos y se incrementa la edad media de la población. Algunos de los estudios que se han realizado indican que aproximadamente unos dos millones de personas mayores viven solas en España.

Los recursos económicos y humanos que aseguren la cobertura total de los mayores en nuestro país no son suficientes. De hecho, en los últimos años y debido a la crisis económica en la que estamos inmersos, se han recortado las ayudas a las personas dependientes en unos dos mil millones de euros. Por lo tanto, resulta evidente que es necesario buscar nuevas alternativas para intentar ayudar a las personas mayores y evitar su exclusión social.

La Unión Europea es consciente de este reto y está fomentando la creación de proyectos piloto que permitan mejorar la situación. El voluntariado es la solución, ya que es una acción que se realiza de forma altruista y cuyo coste no repercute en las arcas del Estado. Desgraciadamente, aunque existen muchos proyectos de voluntariado, estos están mayoritariamente enfocados hacia otros grupos sociales como los afectados por la pobreza, el maltrato, la xenofobia, etc., y muy pocos dirigidos a las personas de la tercera edad.

Por un lado, existen iniciativas como las realizadas por adoptaunabuelo.org, en la que los voluntarios se incorporan a las residencias de mayores con la finalidad que estos vuelvan a recuperar las relaciones sociales. Se crea un espacio intergeneracional donde jóvenes y mayores comparten sus experiencias y aprenden el uno del otro. Por el otro lado, hay empresas especializadas en ofrecen servicios de ayuda a personas mayores a cambio de una retribución económica.

En este contexto nace RESOAM, es una alternativa al resto de programas de voluntariado y destaca por ser gratuito, ágil y dinámico. Además, no corta de una forma radical la relación con las instituciones gubernamentales; al contrario, las necesita para garantizar su propio éxito. Son las personas mayores las que se aprovechan de la tecnología actual para localizar a los voluntarios que se encuentran cerca de su domicilio y solicitarles ayuda. Y no solo eso, también pretende convertirse en una red social que permita poner en contacto a personas mayores para evitar que caigan en la soledad y favorecer su reinserción social.

5. Contenidos

La aplicación utiliza diversos tipos de contenido. En primer lugar, las imágenes y recursos web -páginas de estilos y archivos .js- componen la parte estática de la aplicación y se almacenan en el servidor web Apache. En segundo lugar, los mapas que se utilizan para realizar la búsqueda de acuerdos se obtienen dinámicamente desde los servidores de OpenStreetMap. Finalmente, el resto de contenidos -que incluyen la información general y datos sensibles del usuario- se almacena en el gestor de base de datos MySQL.

A nivel de usuario los contenidos se clasifican en accesibles para todo el mundo -puesto que no requieren que el usuario se identifique en el sistema- y restringidos a un determinado rol -si requieren identificación-. La aplicación no permite que un usuario con un rol determinado pueda acceder a las pantallas o ejecutar acciones que pertenecen a otro rol, salvo excepciones controladas para los roles gestor y administrador. Esta es una medida para simplificar la interfaz y aumentar la seguridad, ya que garantiza que cada perfil solo puede ejecutar las acciones que se le asignaron en la fase de diseño del proyecto.

Contenidos de la parte pública de la web a los que cualquier usuario puede acceder sin estar registrado en la aplicación.

Menú de la parte superior de la pantalla

- Enlace a la Administración Pública: muestra una ventana emergente con la información de contacto de la administración que gestiona la aplicación.
- Inicio: permite regresar a la página de inicio de la aplicación.
- Contacto: si el usuario tiene instalada en su dispositivo una aplicación de correo electrónico permite crear un mensaje de contacto.
- Idioma: muestra un menú desplegable que permite seleccionar el idioma de la aplicación.
- Acceso: redirige la navegación a la página de identificación de usuarios.

Enlaces en el pie de página

- Alta de solicitud de voluntario: redirige la navegación a la pantalla donde se muestra el formulario de inscripción para los voluntarios.

- Alta de solicitud de demandante: redirige la navegación a la pantalla donde se muestra el formulario de inscripción para los demandantes.
- Acceso al portal web de la AGPD: abre una nueva ventana con la página informativa de la AGPD donde se muestra información sobre la privacidad y seguridad en Internet.
- Acceso a redes sociales: permite acceder a diferentes redes sociales como Facebook, Twitter o YouTube.

Enlaces desde la página de inicio

- Alta de solicitud de voluntario: redirige la navegación a la pantalla donde se muestra el formulario de inscripción para los voluntarios.
- Alta de solicitud de demandante: redirige la navegación a la pantalla donde se muestra el formulario de inscripción para los demandantes.

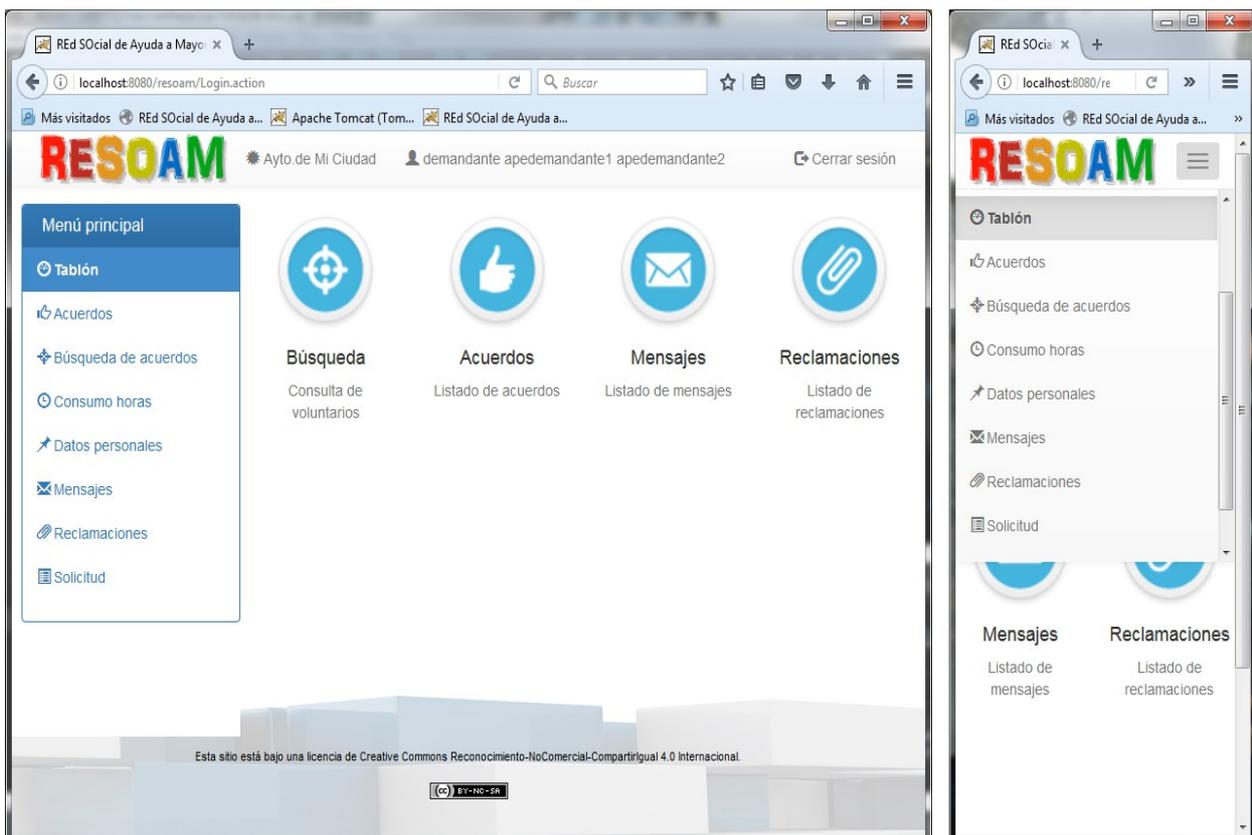


Imagen 2: Ubicación del menú en función de la anchura del dispositivo

Contenidos de la parte restringida de la web a los que únicamente pueden acceder los usuarios registrados. Se accede a los mismos utilizando el menú lateral que se muestra en la parte

izquierda de la pantalla. Si las dimensiones del dispositivo son reducidas -como en el caso de los smartphones- el menú aparecerá al pulsar el botón con las tres franjas de la parte superior derecha de la pantalla.

Rol de demandante

- Acuerdos: muestra el listado y el detalle de los acuerdos en los que interviene el demandante.
- Búsqueda de acuerdos: muestra el mapa en el que el demandante puede consultar que voluntarios se encuentran cerca del área de influencia y realizar la propuesta de un acuerdo.
- Consumo de horas: muestra el número anual de horas concedidas y las horas consumidas.
- Datos personales: muestra los datos personales que introdujo en la ficha de solicitud de registro.
- Mensajes: muestra el listado y el detalle de todos mensajes que genera el sistema. No se trata de correos electrónicos, son mensajes internos generados por la aplicación.
- Reclamaciones: muestra el listado y el detalle de las reclamaciones en las que interviene el demandante, ya sea como reclamante o como reclamado.
- Solicitud: muestra la información que introdujo en la ficha de solicitud de registro.

Rol de voluntario

- Acuerdos: muestra el listado y el detalle de los acuerdos en los que interviene el voluntario.
- Búsqueda de acuerdos: muestra el mapa en el que el demandante puede consultar que demandantes se encuentran cerca del área de influencia y realizar la propuesta de un acuerdo.
- Datos personales: muestra los datos personales que introdujo en la ficha de solicitud de registro.
- Mensajes: muestra el listado y el detalle de todos mensajes que genera el sistema.
- Reclamaciones: muestra el listado y el detalle de las reclamaciones en las que interviene el voluntario, ya sea como reclamante o como reclamado.
- Solicitud: muestra la información que introdujo en la ficha de solicitud de registro.

Rol de administrador

- Datos personales: muestra los datos personales del administrador del sistema.
- Calendario: permite crear los calendarios que permitirán que voluntarios y demandantes puedan realizar acuerdos.
- Mensajes: muestra el listado y el detalle de todos mensajes que genera el sistema.
- Usuarios: muestra el listado y el detalle de los usuarios del sistema con rol gestor, inspector y administrador.

Rol de gestor.

- Datos personales: muestra los datos personales del gestor del sistema.
- Mensajes: muestra el listado y el detalle de todos mensajes que genera el sistema.
- Solicitudes demandantes, muestra el listado y el detalle de las solicitudes de los demandantes.
- Solicitudes voluntarios, muestra el listado y el detalle de las solicitudes de los voluntarios.
- Reclamaciones: muestra el listado y el detalle de las reclamaciones.

Rol de inspector.

- Datos personales: muestra los datos personales del inspector.
- Inspecciones: muestra el listado de las inspecciones pendientes y la ficha del demandante.
- Mensajes: muestra el listado y el detalle de todos mensajes que genera el sistema.

6. Metodología

Actualmente los productos y las soluciones tecnológicas se quedan obsoletos con demasiada rapidez, por lo que resulta más adecuado abandonar el enfoque tradicional de desarrollo -en el que se planifica completamente el proyecto antes de comenzar- en beneficio de una metodología ágil como scrum¹ que permite un desarrollo incremental marcado por los denominados sprints².

En conclusión, el uso de scrum aporta una mayor flexibilidad ya que permite reaccionar rápidamente ante los cambios que se puedan producir durante el desarrollo del proyecto; lo que a su vez permite gestionar los riesgos de forma más realista. Además, se consigue incrementar la productividad al no invertir tiempo en la generación de documentación y la calidad del software generado es superior.

1 [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))

2 <http://www.obs-edu.com/es/blog-investigacion/project-management/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>

7. Arquitectura de la aplicación

Se trata de realizar una aplicación web multi-idioma y con diseño responsivo que permita ajustarse automáticamente a las dimensiones del dispositivo visualizador y mostrar correctamente los contenidos en una interfaz amigable (user-friendly). Se desarrollada estableciendo como prioridad su uso por personas mayores que no tienen demasiada experiencia en la navegación web.

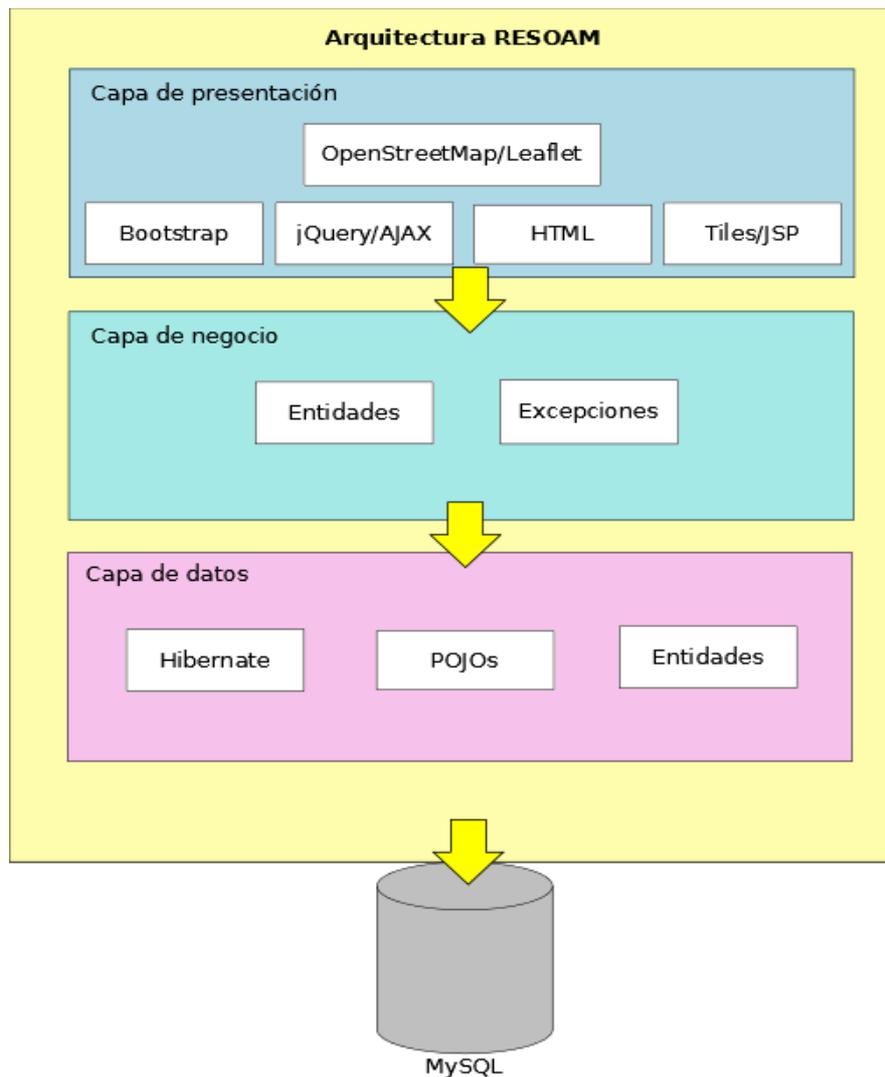


Imagen 3: Diagrama de capas/niveles de la aplicación

El desarrollo se realiza en la plataforma J2EE con el IDE de Eclipse. Se emplea el patrón MVC (Modelo/Vista/Controlador) y se realiza un diseño modular en tres capas: presentación, negocio y datos. De esta forma, se consigue que los componentes que se construyan tengan un bajo nivel

de acoplamiento y se puedan extender las funcionalidades iniciales o añadir otras nuevas de forma sencilla y sin efectos colaterales.

Como framework de desarrollo principal se utilizará Apache struts 2, que pese a no haber conseguido el éxito de su predecesor struts 1, es un framework mucho más robusto y eficiente que simplifica el proceso de desarrollo.

La capa de presentación utilizará páginas JSP en combinación con HTML5 y el framework de Bootstrap para conseguir un diseño homogéneo en todos los navegadores y, adaptable al tamaño de la pantalla del dispositivo. Para la georreferenciación se utilizarán los mapas proporcionados por OpenStreetMap y la librería Javascript Leaflet que se encargará de la manipulación y el envío de los datos georreferenciados entre el cliente y el servidor.

La capa de datos utilizará el framework de Hibernate para aislar a la aplicación del tipo de base de datos a utilizar. En este caso se emplea MySQL porque es una base de datos ligera que no consume demasiados recursos y funciona adecuadamente en entornos pequeños y medianos. Pero cuando se realice el despliegue de la aplicación en un entorno de producción es probable que la base de datos sea distinta y, es en ese momento cuando Hibernate entra en acción para permitir que el proceso de migración tenga el mínimo impacto.

Para una correcta visualización se recomienda la utilización de los siguientes navegadores ya que son los que se han utilizado para realizar las pruebas.

- Mozilla Firefox 52.0.2 (32-bit)
- Google Chrome Versión 57.0.2987.133
- Microsoft Internet Explorer 11.0.9600.18537

Para que la aplicación sea altamente escalable y multiplataforma se utilizará la combinación de un servidor web y un contenedor de servlets con soporte para páginas JSP que implemente las especificaciones Servlet 3.0 y JSP 2.2.

En el siguiente diagrama se muestra la arquitectura propuesta para desplegar la aplicación. El sistema representado no es muy complejo, de hecho, es una arquitectura bastante común en la mayoría de las Administraciones Públicas y en algunos proveedores de alojamiento web.

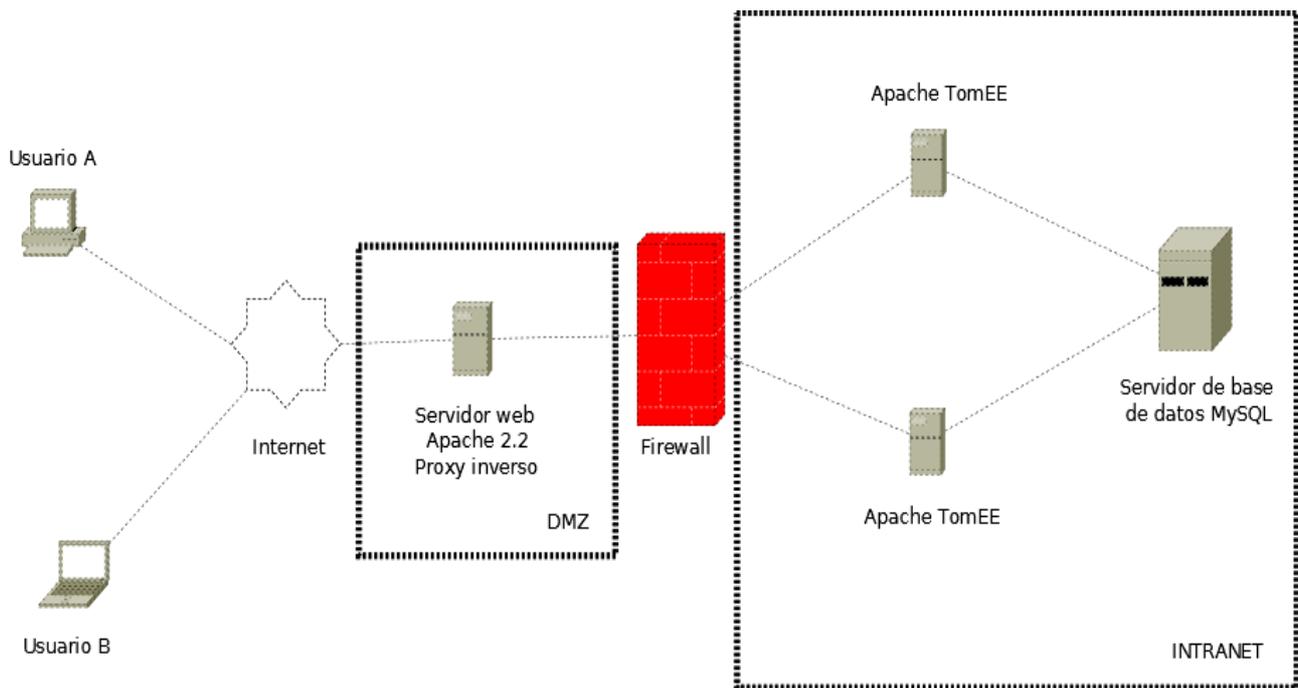


Imagen 4: Diagrama de la arquitectura de la aplicación

El servidor web se encargará de servir el contenido estático de la aplicación -archivos con extensión css, js, html, png, jpg, pdf- descargando de este trabajo al contenedor de servlets, lo que mejorará el tiempo de carga de las páginas web. También funcionará como proxy inverso (Aumentando la seguridad del sistema al ser el único equipo expuesto en la DMZ) y podrá balancear las peticiones a más de un contenedor de servlets.

El numero de servidores tomEE puede ser variable en función de la carga de trabajo que soporte el sistema. Estos servidores se encuentran en la intranet y en ellos se realiza el despliegue de la aplicación.

Aunque en el gráfico se muestran varios equipos físicos, para el desarrollo del proyecto tanto el servidor Apache 2.2 como el servidor Apache tomEE y el servidor MySQL se ejecutan en una misma máquina física.

8. Plataforma de desarrollo

Consiste en un equipo portátil destinado a las tareas de elaboración de documentación, diseño de diagramas, retoque de imágenes e implementación del código de la aplicación. Además de un servidor dedicado que incluye un conjunto de herramientas de software que proporcionan la infraestructura necesaria para hacer la aplicación accesible en Internet.

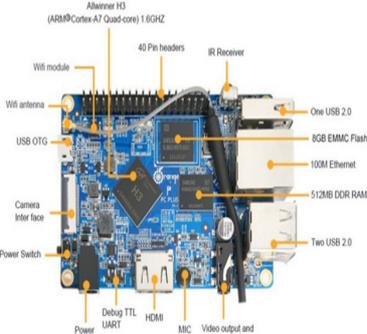
Estación de trabajo	 Imagen 5: Pavilion dv6	Marca/Modelo	HP / Pavilion dv6 Notebook PC ³
		Procesador	Intel Core I7 1,60 GHz.
		Memoria	8Gb. DDR3
		GPU	AMD Radeon HD 6500 1Gb.
		Almacenamiento	HD Sata 446 Gb.
		Conectividad	LAN Ethernet 10/100/1000 LAN inalámbrica 802.11a/b/g/n
Servidor	 Imagen 6: Orange Pi Plus	Marca/Modelo	Xunlong Software CO / Orange Pi Plus ⁴
		Procesador	H3 Quad-core Cortex-A7 H.265/HEVC 4K
		Memoria	1GB DDR3
		GPU	Mali400MP2 GPU @600MHz
		Almacenamiento	Micro SD Samsung 16 Gb.
		Conectividad	10/100/1000M Ethernet RJ45 Realtek RTL8189ETV, IEEE 802.11 b/g/n

Tabla 1: Hardware de la plataforma de desarrollo

3 <http://support.hp.com/us-en/product/HP-Pavilion-dv6-3100-Entertainment-Notebook-PC-series/4247579/model/5076680>

4 <http://www.orangepi.org/>

Estación de trabajo	Sistema operativo	Windows 7 Home Premium 64 bits
	Versión Java	JRE 1.0.8_101-b13 JDK 1.0.8_101
	IDE Desarrollo Java	Eclipse Mars Eclipse Java EE IDE for Web Developers. Version: Mars.2 Release (4.5.2) Build id: 20160218-0600 Subclipse 1.10.13 Subversion Client Adapter 1.10.3 Subversion JavaHL Native Library Adapter 1.8.15
	Herramienta de empaquetado	Maven 3.3.9
	Servidor web (Contenedor de servlets)	Apache Tomcat 8.0.41
	Navegador web	Google Chrome 57.0.2987.133 Mozilla Firefox 52.0.2 (32 bits) Internet Explorer 11.0.9600.18537
Servidor	Sistema operativo	ARMBIAN 5.25 stable Debian GNU/Linux 8 (jessie) 3.4.113-sun8i
	Versión Java	JRE 1.8.0_121-b13
	Herramienta de empaquetado	Maven 3.0.5
	Servidor web	Apache web server 2.2
	Servidor web (Contenedor de servlets)	Apache tomEE 7.0.2
	Servidor de base de datos	MySQL 5.5.54-0+deb8u1 Compilado para debian-linux-gnu (armv7l)
	Control de código fuente	Subversion server para Debian GNU/Linux 8 (jessie)
	Despliegue de aplicaciones	Jenkins 2.32.2

Tabla 2: Software de la plataforma de desarrollo

9. Planificación

El proyecto establece cuatro hitos importantes que se corresponden con las entregas intermedias que se deben realizar, que son la definición del proyecto, la fase alfa, la beta y la final 1.0. Con la realización de cada sprint se logra concluir una parte del proyecto que servirá como punto de partida para las siguientes.

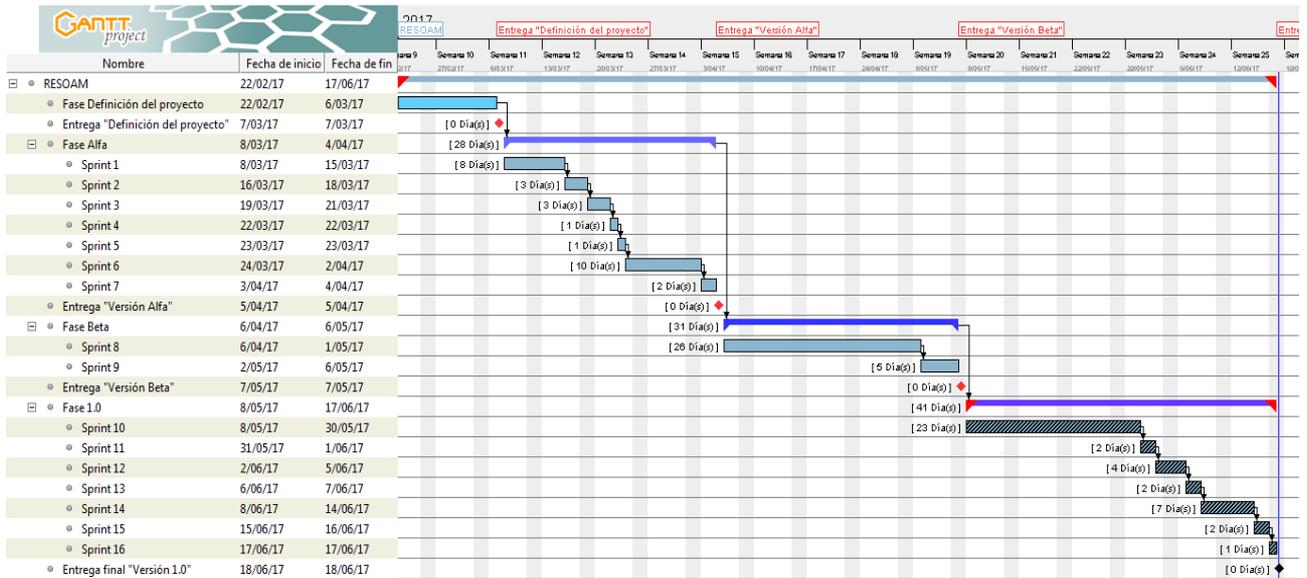


Imagen 7: Diagrama Gantt del proyecto

Fase	Sprint	Descripción	Duración (Días)
Alfa	S1	Análisis general: <ul style="list-style-type: none"> • Aplicación. • Configuración de desarrollo. • Base de datos. • Equipo de desarrollo y servidor de pruebas. • Configuración de paquetes y estructura del proyecto. 	8
	S2	Diseño de la base de datos: <ul style="list-style-type: none"> • Modelo lógico. • Modelo físico. • Realizar los scripts de creación y carga inicial de datos. 	3
	S3	Diseño de las pantallas y de la iteración: <ul style="list-style-type: none"> • Definir y diseñar las pantallas. • Definir los flujos entre las mismas. 	3
	S4	Instalación y configuración del servidor de pruebas: <ul style="list-style-type: none"> • Servidor web Apache. 	1

Fase	Sprint	Descripción	Duración (Días)
Alfa		<ul style="list-style-type: none"> • Servidor de servlets Apache TomEE. • Servidor de base de datos MySQL. • Instalar la base de datos de la aplicación. • Configurar el pool de conexiones en Apache TomEE. • Configurar Apache como proxy inverso. • Configurar software de control de versiones Subversion en el servidor Apache. 	
	S5	Instalación y configuración del entorno de desarrollo: <ul style="list-style-type: none"> • IDE de desarrollo Eclipse Mars. • Maven 3.3.9. • Struts 2.5.10.1 • Hibernate 4.0.1 • jQuery 1.12.4 • Bootstrap 3.3.7 • Integración del código fuente en el control de versiones. 	1
	S6	Desarrollo 1: <ul style="list-style-type: none"> • Pantalla principal (Home). • Pantallas de control de acceso. 	10
	S7	Actualización de la memoria.	2
Beta	S8	Desarrollo 2: <ul style="list-style-type: none"> • Pantallas para el perfil de gestor. • Pantallas para el perfil voluntario. 	26
	S9	Actualización de la memoria.	5
1.0	S10	Desarrollo 3: <ul style="list-style-type: none"> • Pantallas para el perfil solicitante. • Pantallas para los perfiles inspector y administrador. • Exportación del plan de trabajo para el voluntario. • Exportación del plan de ayuda para el solicitante. • Exportación de informes para el gestor. 	23
	S11	Revisión final de la aplicación: <ul style="list-style-type: none"> • Control de acceso. • Gestión de perfiles. • Navegación y geolocalización. 	2
	S12	Actualización de la memoria.	4
	S13	Elaboración de la presentación.	2
	S14	Realización del video de defensa del proyecto.	7
	S15	Revisión global del proyecto.	2
	S16	Cambios menores.	1

Tabla 3: Desglose de tareas de los sprints

10. Proceso de trabajo

Una buena planificación del proceso de trabajo es clave para lograr el éxito ante cualquier reto al que nos enfrentemos. Por este motivo, se recomienda descomponer las tareas complejas en unidades más pequeñas denominadas subtareas, que resultan más manejables y fáciles de gestionar. Los siguientes apartados describen brevemente las subtareas del proceso de trabajo de esta aplicación.

10.1. Análisis global del proyecto

Es la primera subtask en realizarse y la más crítica de todas, ya que de ella dependen las demás. Se realiza una primera aproximación hacia el objetivo que se quiere lograr y se valoran las necesidades y requisitos que se deben satisfacer. Existen distintos tipos de metodologías a aplicar en el proceso de desarrollo de software. En la Administración Pública se suele utilizar METRICA3, sin embargo, esta requiere elaborar una gran cantidad de documentos que cubran todos los aspectos técnicos del desarrollo del proyecto antes de empezar a programar; así que no se ajusta a un proyecto en el que el factor tiempo es vital. Por lo tanto, se empleará una metodología ágil como scrum que permite concentrar los esfuerzos en el desarrollo para cumplir con el cronograma del proyecto.

A	B	C	D	E	F	G	
Identificado	Historia	Puntos	Prioridad	Riesgo	Usuario	Descripción	
20	HU-01	Alta y modificación de usuarios	3	2	1	Administrador	Como usuario administrador debo dar de alta nuevos usuarios de tipo gestor y administrador, así como realizar la baja lógica de aquellos que desaparezcan de la plataforma.
21	HU-02	Creación y configuración del calendario	3	3	2	Administrador	Como usuario administrador debo poder configurar el calendario anual para que voluntarios y demandantes puedan ponerse de acuerdo en la selección de días y rangos.
22	HU-03	Realizar inspección	3	1	1	Inspector	Como usuario inspector debo poder consultar el listado de inspecciones asignadas por el gestor así como su detalle para poder realizar las visitas y asignar la valoración pertinente en función de la veracidad de los datos proporcionados por el demandante.
23	HU-04	Realizar registro de voluntariado	4	2	2	Voluntario	Como usuario voluntario debo poder realizar el registro de mi solicitud en la aplicación, e indicar mis preferencias horarias.
24	HU-05	Consulta de los datos de solicitud	2	1	1	Voluntario	Como usuario voluntario debo poder consultar los datos de la solicitud realizada para solicitar al gestor la modificación pertinente si quiero añadir nueva experiencia laboral.
25	HU-06	Realizar acuerdos	3	3	3	Voluntario	Como usuario voluntario debo poder realizar un acuerdo de servicio con un demandante.
26	HU-07	Realizar reclamación	3	2	1	Voluntario	Como usuario voluntario debo poder registrar una reclamación a un demandante para notificar que ha incumplido su acuerdo y no estaba en el domicilio cuando se realizó la visita.
27	HU-08	Busqueda de demandantes	5	3	4	Voluntario	Como usuario voluntario debo poder consultar en un mapa la ubicación de los demandantes y consultar sus rangos horarios para determinar si conciden con los míos y enviarles una solicitud de acuerdo.
28	HU-09	Consultar listado de los acuerdos realizados	3	1	1	Voluntario	Como usuario voluntario debo poder obtener un listado de los acuerdos que he realizado a lo largo de un día, de una o varias semanas, de uno o varios meses, o a lo largo de todo el año para poder exportarlos a un archivo e imprimirlos.
29	HU-10	Realizar solicitud	4	2	2	Demandante	Como usuario demandante tengo que cumplimentar una solicitud para poder acceder al servicio de ayuda.
30	HU-11	Consulta de los datos de solicitud	2	1	1	Demandante	Como usuario demandante debo poder consultar los datos de la

Imagen 8: Historias de usuario

10.2. Diseño de la base de datos

La siguiente tarea crítica es la elección de la base de datos que se va a utilizar. Se descartan los productos comerciales como Oracle 11g y Microsoft SQL Server debido a su elevado coste. Afortunadamente, existen otras alternativas igualmente válidas en el mundo del software libre como PostgreSQL y MySQL -la opción elegida-.

Una vez seleccionado el SGBD, se realiza el diagrama conceptual de base de datos en el que se por primera vez se definen las posibles entidades que formarán parte del modelo así como sus relaciones y atributos más importante. Mediante el refinamiento progresivo del modelo se llega hasta el modelo lógico donde de detallan todos los atributos de las entidades y tras refinar nuevamente el modelo se consigue elaborar el diagrama físico de base de datos.

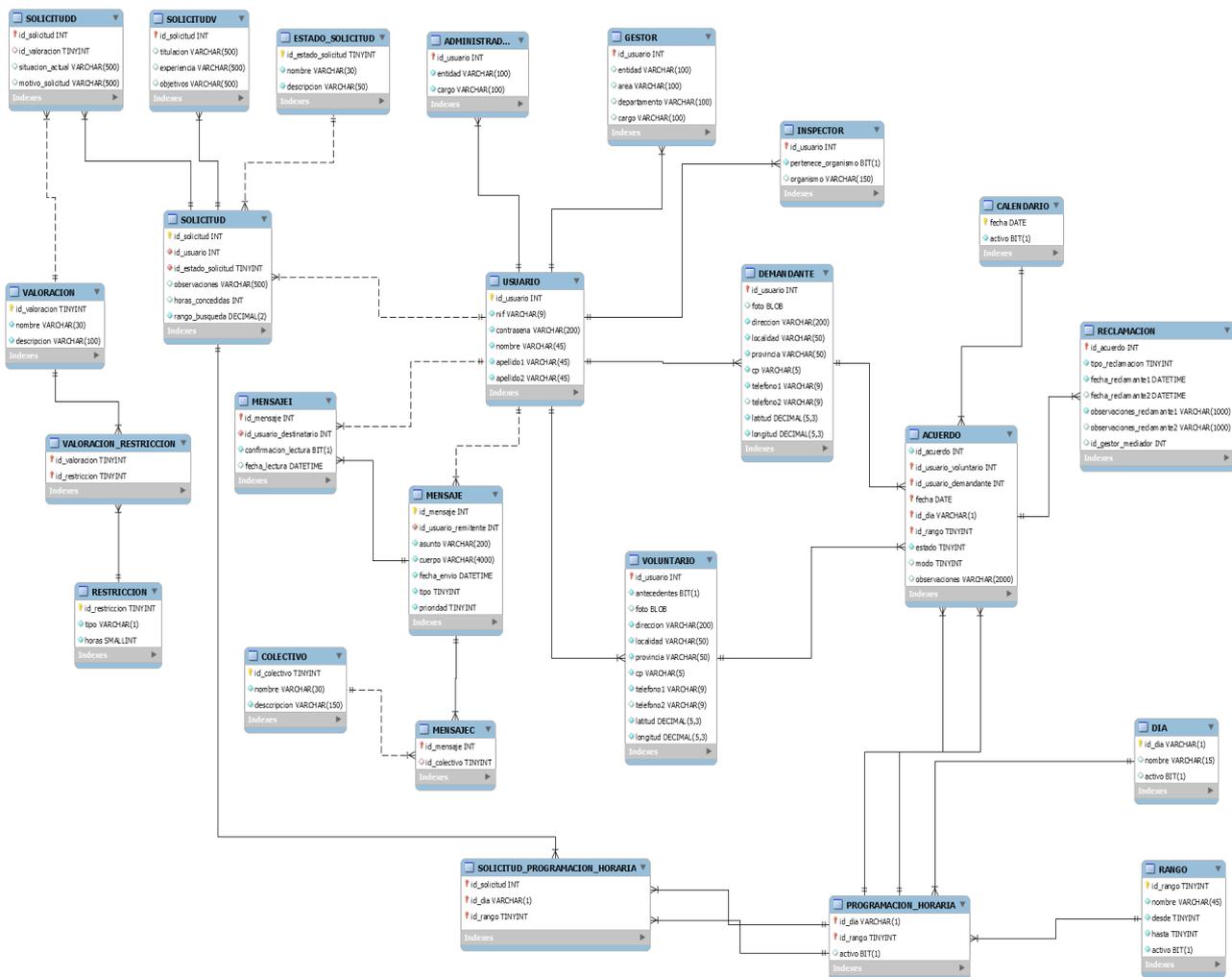


Imagen 9: Modelo físico de la base de datos

10.3. Diseño del flujo de navegación por las pantallas

Una vez definido el modelo físico de datos y utilizando diferentes técnicas se van diseñando las páginas que se prevé serán necesarias para gestionar los datos del modelo, así como la navegación entre ellas. A menudo cuando diseña una aplicación es habitual que las páginas de un sitio web tengan el mismo aspecto, es lo que se denomina branding o imagen de marca. Para conseguirlo y facilitar el desarrollo se utilizará el framework Apache Tiles, el cuál permite componer una página rápidamente utilizando pequeños fragmentos de código reutilizable.

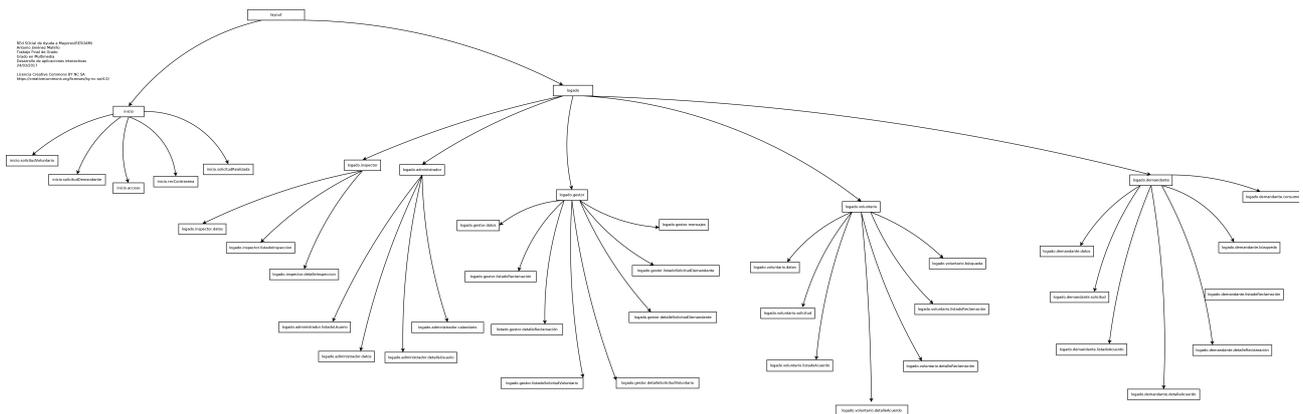


Imagen 10: Mapa de configuración para tiles

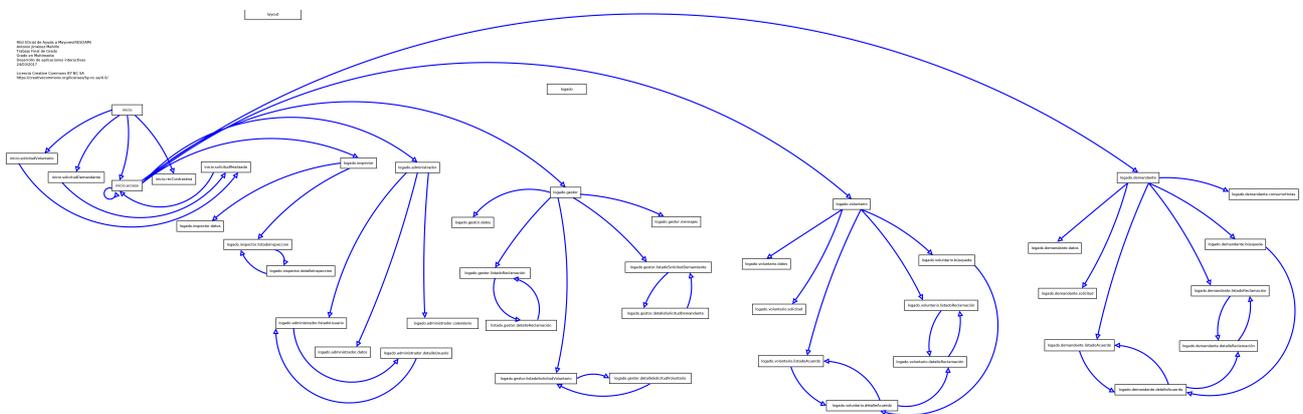


Imagen 11: Mapa de navegación

10.4. Elaboración de los wireframes

A partir del diagrama generado en el punto anterior se realizan diseños específicos del aspecto final que debería tener cada una de las páginas. Los wireframes sirven para mostrar al cliente donde van a estar colocados los elementos dentro de la página y determinar si se ajusta al diseño pretendido. Durante todo el ciclo de desarrollo puede ser necesario realizar algún pequeño ajuste; por ejemplo, para añadir un nuevo botón o mostrar el nombre del usuario en una página.

10.5. Instalación y configuración del servidor de pruebas

Como servidor de pruebas se ha recurrido a utilizar una solución muy económica -alrededor de 18 euros- una pequeña placa destinada al Internet de las cosas⁵ que integra el servidor web Apache, el contenedor de servlets tomEE, el motor de base datos MySQL y un sistema de control de versiones para gestionar y salvaguardar el código fuente de la aplicación. En un entorno de producción real, todo este proceso resultaría más complejo ya que lo habitual es que cada servidor se instale en una máquina física o virtual independiente.

10.6. Instalación y configuración del entorno de desarrollo

La instalación del entorno de desarrollo es una de las subtareas más importantes y a menudo olvidada por los desarrolladores. Consiste en instalar y configurar en el equipo todo el software necesario para acometer el proyecto antes de empezar a programar.

10.7. Definición de la estructura del proyecto

Estructurar adecuadamente el código fuente es vital durante todo el ciclo de vida del proyecto. Los errores que se comentan durante esta subtarea se arrastrarán en las siguientes. Del mismo modo, cuanto más se tarde en corregir los errores de diseño relativos a la organización del código en clases y demás, más tiempo se deberá emplear para solucionarlo.

Para esta aplicación se utiliza un diseño multiproyecto basado en arquetipos de Maven; que está compuesto por un proyecto padre que actúa como contenedor de los módulos del programa; la aplicación web, la librería de negocio y la librería de acceso a datos.

10.8. Desarrollo del código fuente

Esta es sin lugar a dudas la subtarea más entretenida para el desarrollador, ya que a partir de la información recolectada, las historias de usuario y el análisis realizado, se procede a la implementación código fuente de la aplicación.

10.9. Integración de los componentes

Conseguir agrupar en un mismo proyecto software de diverso tipo desarrollado por terceros y que funcione a la perfección es una utopía. En la mayoría de las ocasiones es necesario realizar multitud de pruebas para asegurar que la integración es satisfactoria y que no se producen efectos colaterales. En función de la naturaleza del proyecto y de los componentes utilizados el nivel de complejidad puede aumentar.

⁵ https://es.wikipedia.org/wiki/Internet_de_las_cosas

10.10. Pruebas

Aunque cada porción de código se prueba mientras se desarrolla, existe una fase destinada especialmente a realizar pruebas que permiten comprobar el comportamiento de la aplicación bajo diversos escenarios. Estas pruebas pueden estar orientadas a comprobar el funcionamiento de un módulo concreto, la integración entre varios módulos e incluso probar el rendimiento y la seguridad de la misma.

10.11. Revisión final

Esta es la fase previa al empaquetado, y en ella se vuelve a revisar completamente la funcionalidad y apariencia de la aplicación para detectar errores que pudieran haber pasado desapercibidos en fases anteriores.

10.12. Empaquetado

Una vez finalizado el desarrollo y terminadas las pruebas es necesario generar el código binario del proyecto y empaquetarlo en formato war para el contenido dinámico y tar.gz para el contenido estático; la herramienta Maven facilita enormemente esta labor y, puede ser invocada desde línea de comandos o integrada en un IDE.

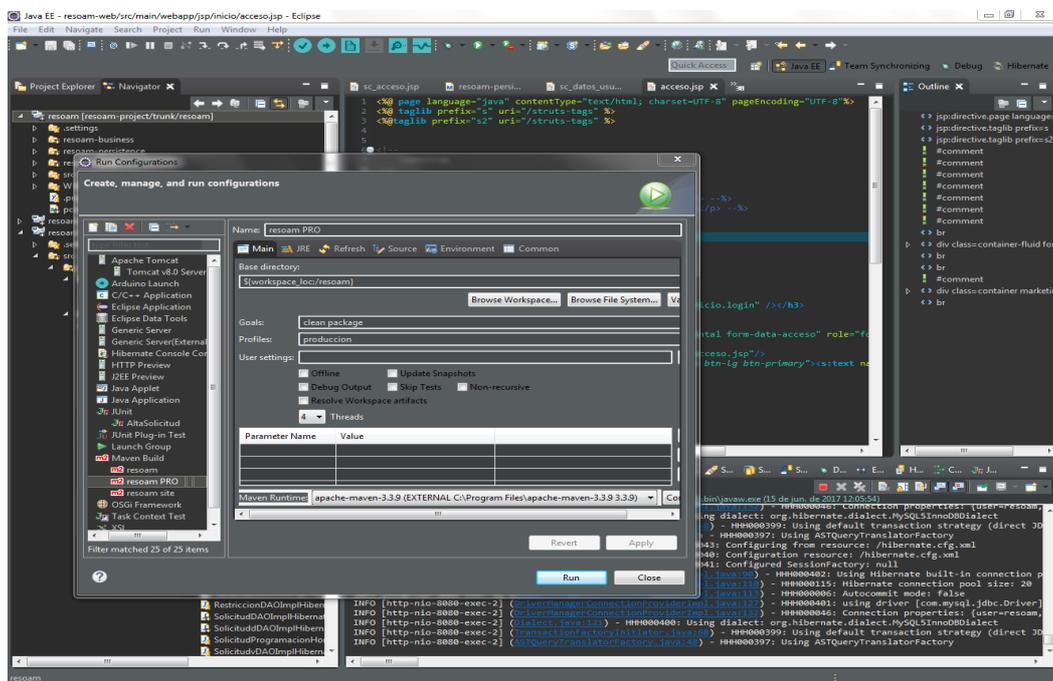


Imagen 12: Empaquetando el proyecto con Maven desde Eclipse

10.13. Implantación

En esta fase se crea y se configura la base de datos de la aplicación, se despliega el archivo war en el contenedor de servlets y, se descomprime el archivo tar.gz en el servidor web. Posteriormente se accede a la aplicación y se verifica que el funcionamiento es el esperado.

11. Diagramas UML

En esta sección se incluyen algunos de los diagramas en formato UML utilizados durante las fases de análisis y construcción del proyecto.



Imagen 13: Diagrama de casos de uso

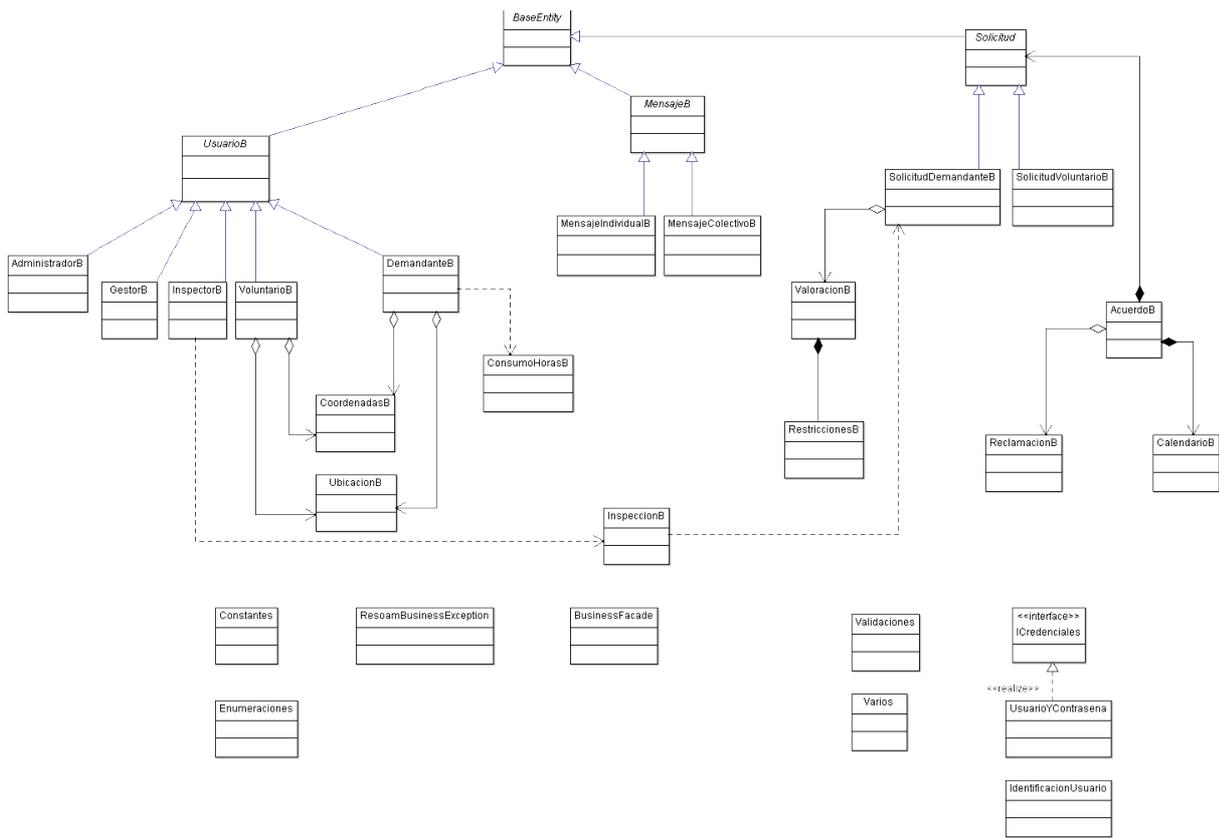


Imagen 14: Diagrama de clases de negocio

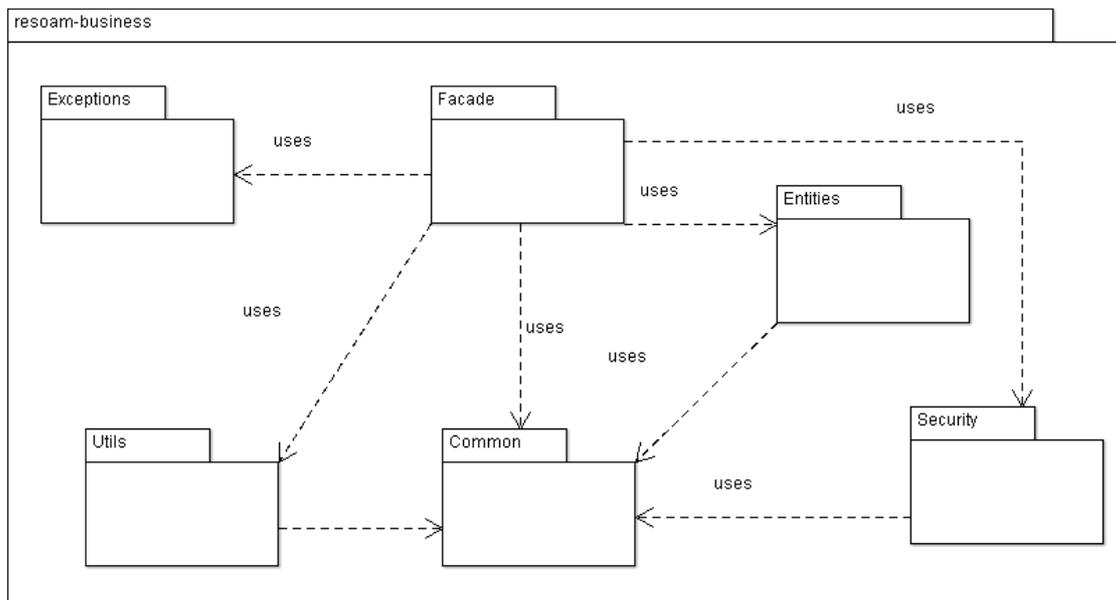


Imagen 15: Diagrama de paquetes de negocio

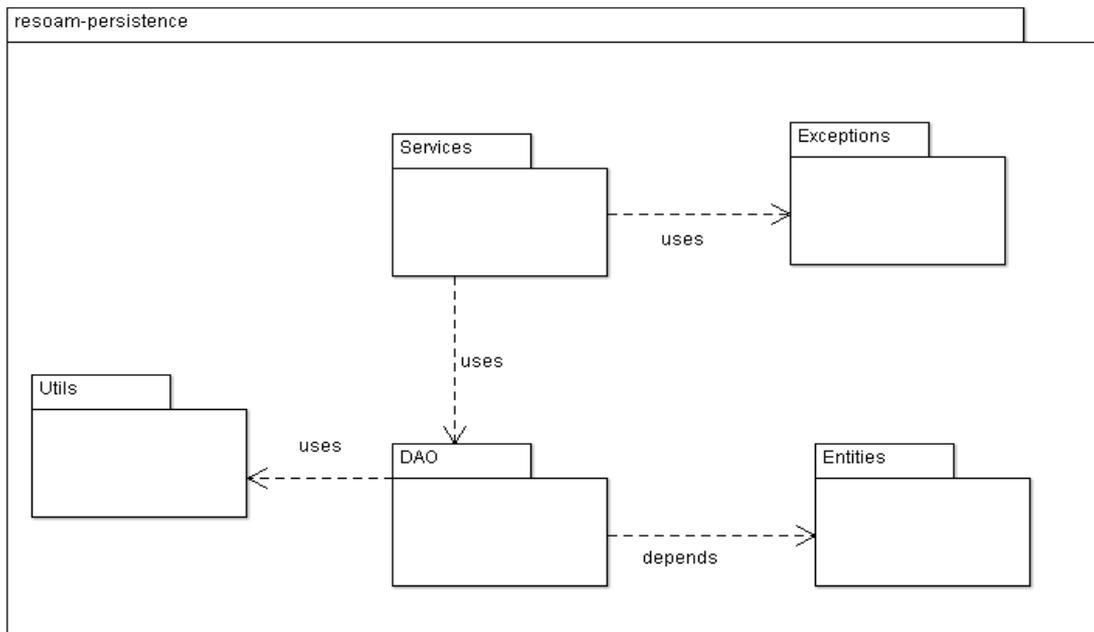


Imagen 16: Diagrama de paquetes de persistencia

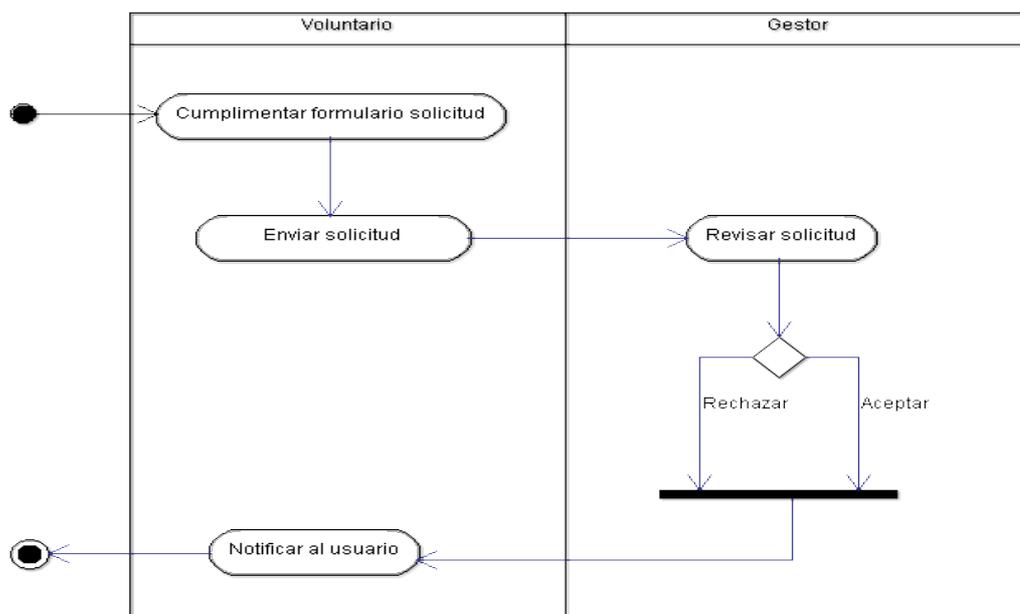


Imagen 17: Diagrama de actividad alta solicitud voluntario

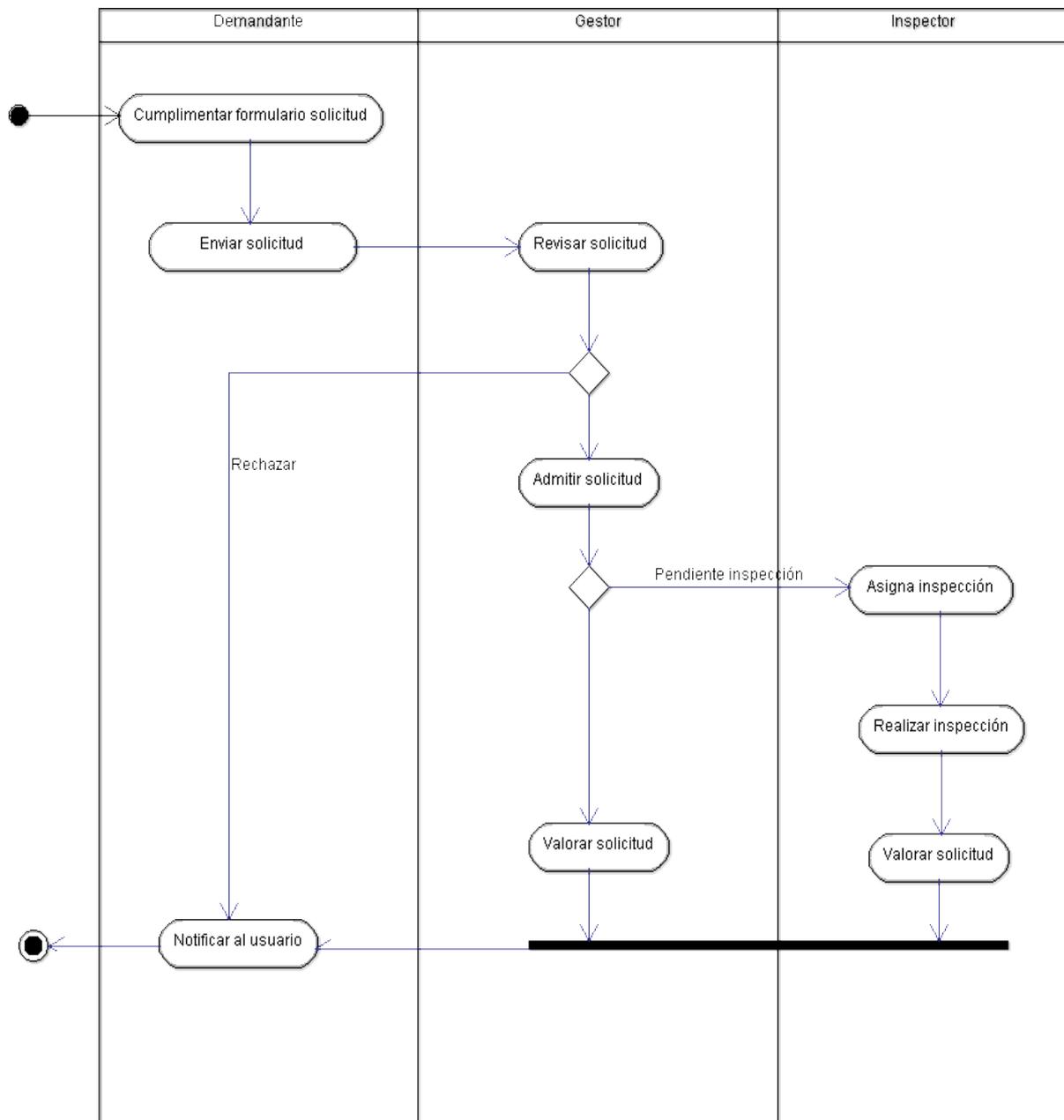


Imagen 18: Diagrama de actividad solicitud alta demandante

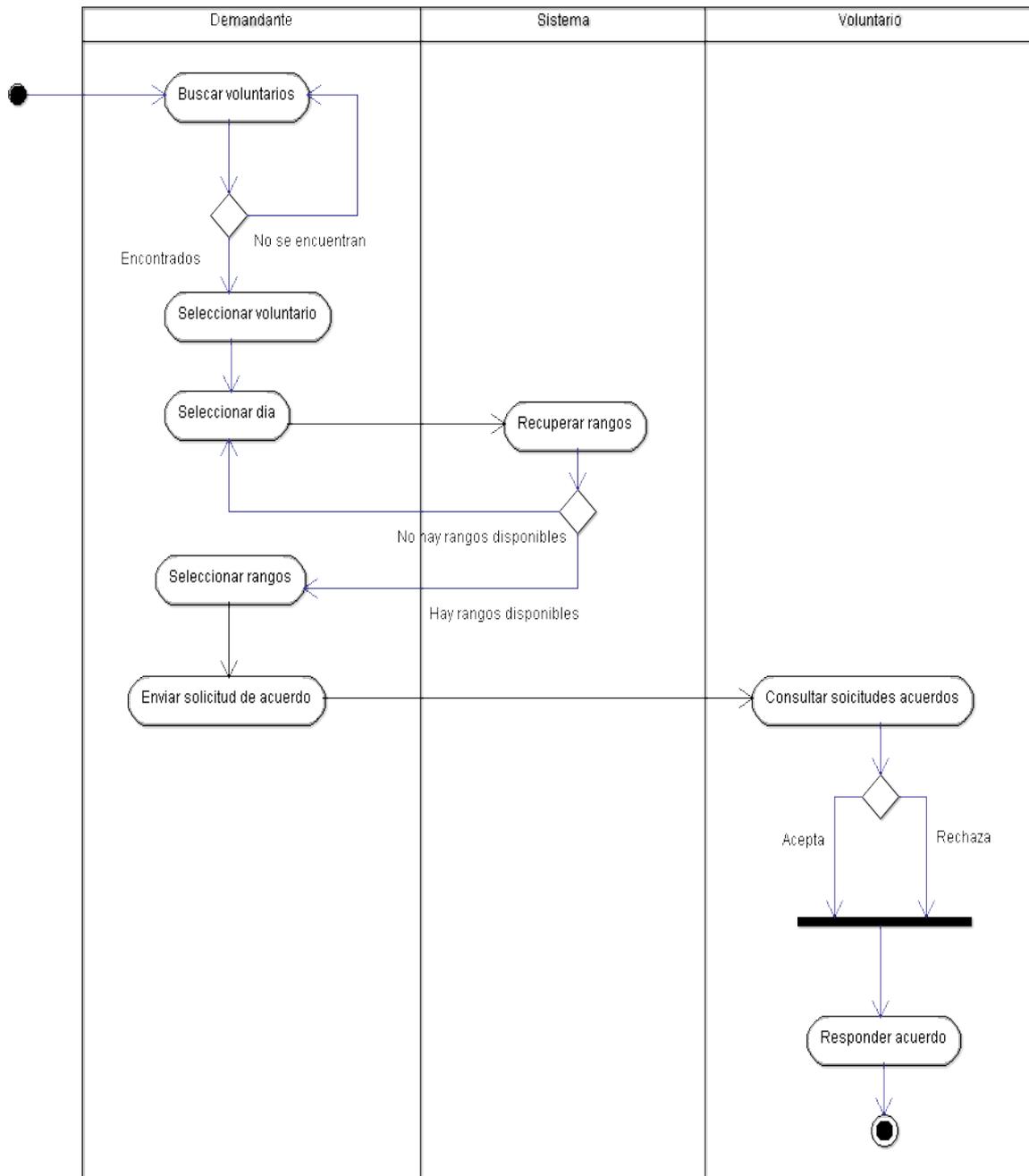


Imagen 19: Diagrama de actividad realizar acuerdo demandante

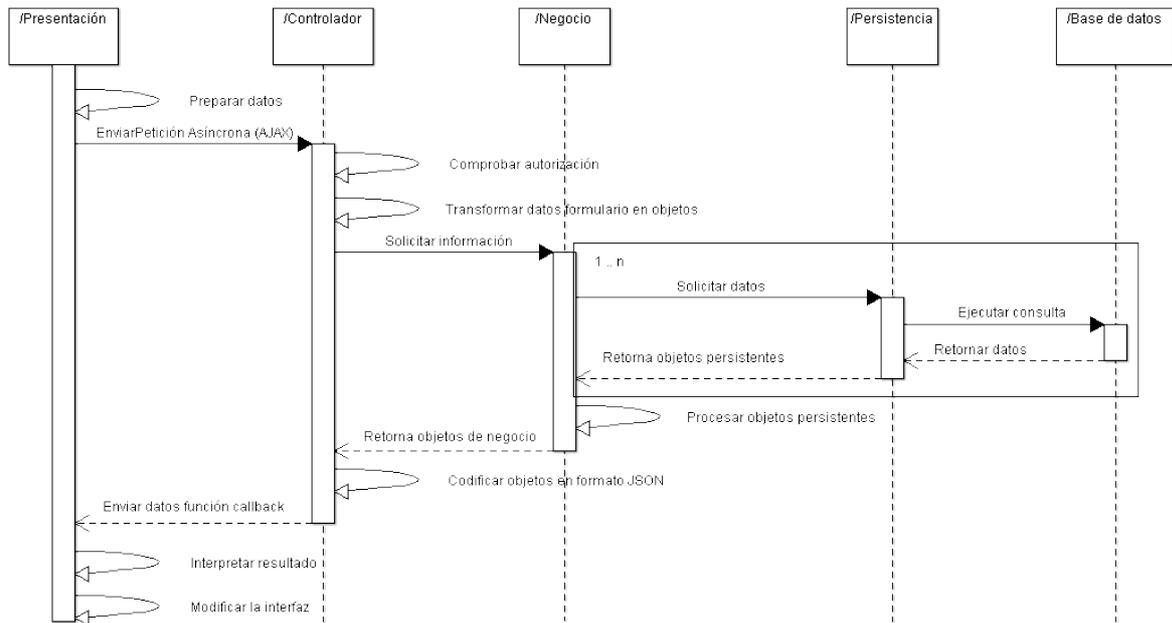


Imagen 20: Diagrama de secuencia para las peticiones

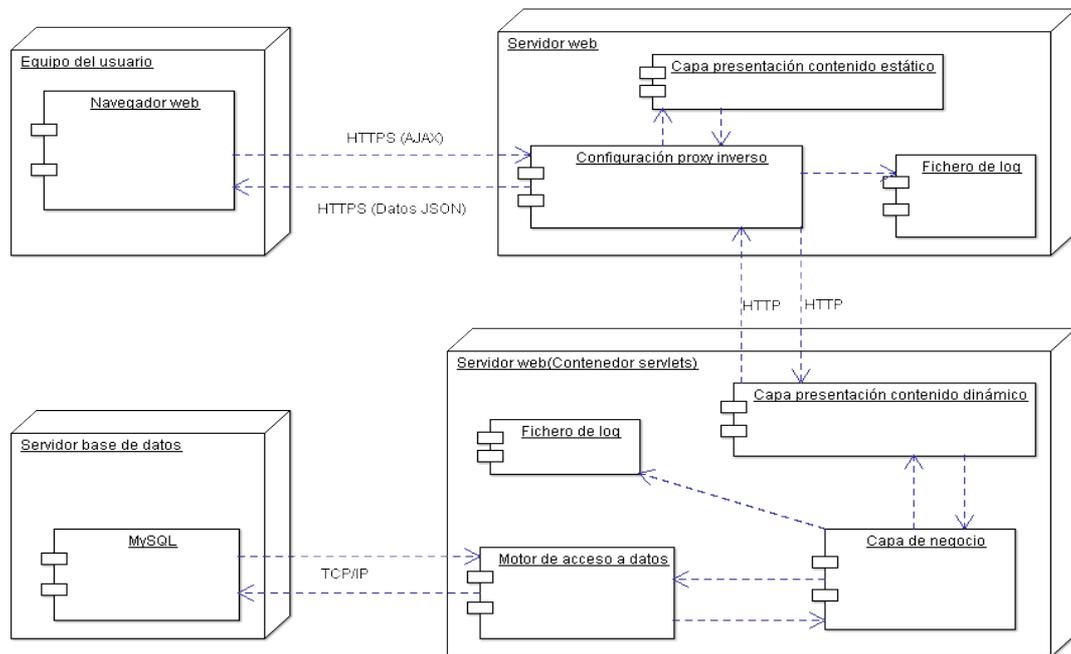


Imagen 21: Diagrama de despliegue

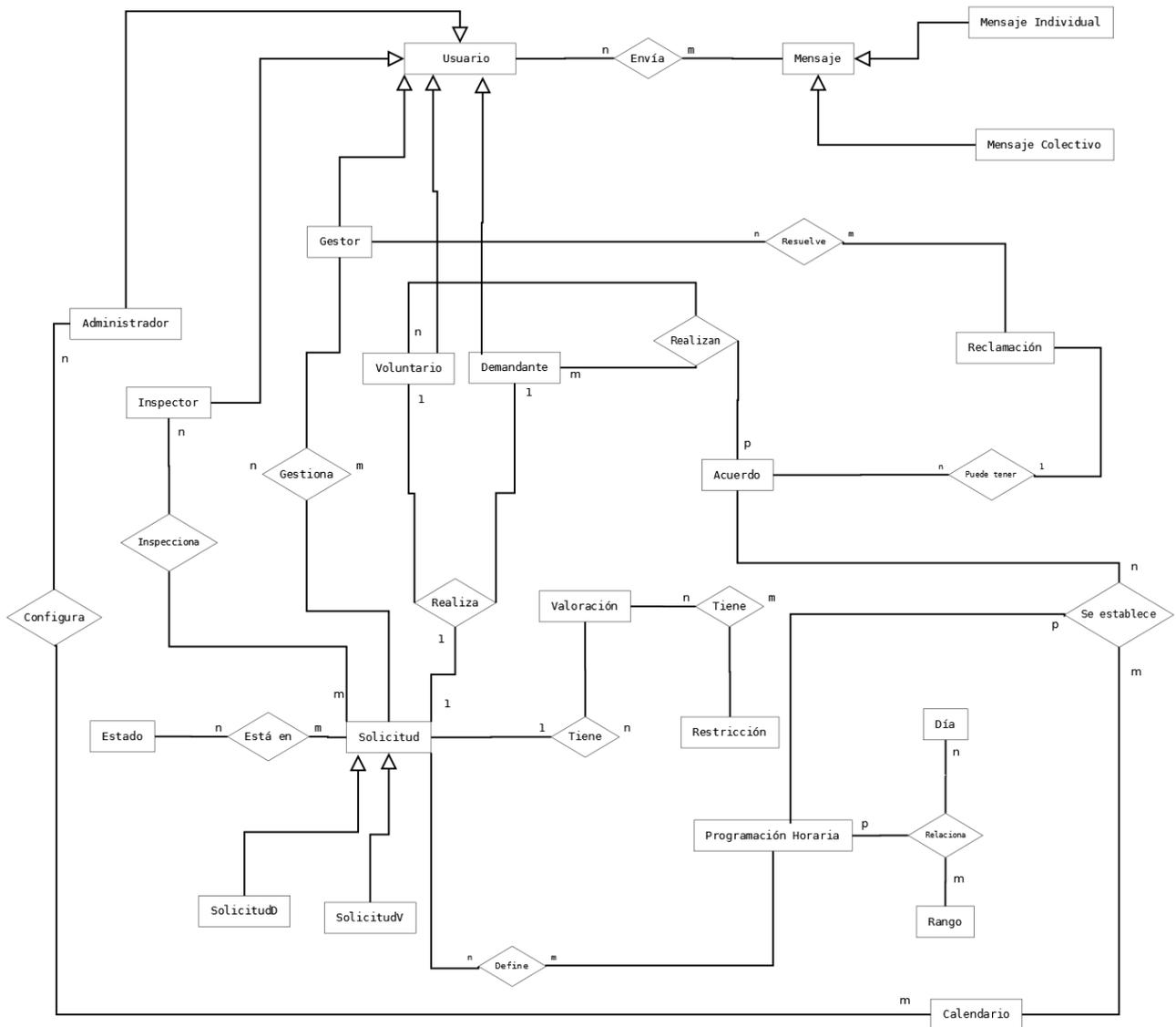


Imagen 22: Modelo entidad-relación

12. APIs utilizadas

Para el desarrollo de la aplicación se utilizan varias librerías proporcionadas de manera altruista por terceros. Estas librerías permiten agilizar el proceso de desarrollo, ya que en lugar de reinventar la rueda y programar desde cero, se pueden integrar fácilmente en el código fuente del proyecto.

- [Apache Struts 2.5.10.1](http://struts.apache.org/)⁶: es un conjunto de librerías para la plataforma java que componen un framework para el desarrollo de aplicaciones web interactivas siguiendo el patrón MVC.
- [Apache Tiles 2.5.10.1](https://tiles.apache.org/)⁷: es una librería java que complementa a la anterior y permite ahorrar tiempo en el diseño y composición de las páginas web de la aplicación. Se basa en la definición de plantillas de páginas y la utilización de pequeños archivos a modo de recursos que pueden ser combinados en diferentes páginas.
- [Apache Struts json plugin](https://struts.apache.org/docs/json-plugin.html)⁸ 2.5.10.1: es un plugin java que permite que Struts pueda enviar contenido en formato JSON al cliente en lugar de invocar a un elemento de Apache Tiles.
- [Apache log4j 1.2.17](https://logging.apache.org/log4j/2.x/)⁹: es una librería para java que permite la generación de archivos de log con una serie de características personalizables como el rotado diario de los archivos, la configuración del formato de salida y el nivel de detalle a mostrar.
- [Google gson 2.2.2](https://google.github.io/gson/apidocs/com/google/gson/Gson.html)¹⁰: es una librería de Google para java que permite leer y escribir cadenas de texto en formato JSON. La aplicación la utiliza únicamente para mapear los datos recibidos desde el formulario cliente a objetos java.
- [Hibernate 4.0.1.Final](http://hibernate.org/orm/downloads/)¹¹: es un conjunto de librerías para java que se encargan de gestionar la persistencia de los datos. Para ello, define una serie de objetos Java a partir de la aplicación de ingeniería inversa a la base de datos, que permiten abstraer todo el proceso de establecimiento de conexiones, gestión de transacciones y manipulación de datos a una simple interacción con objetos.
- [Lombok 1.16.14](https://projectlombok.org/download.html)¹²: es una librería para java que libera al programador de escribir los métodos getter y setter de cada uno de los atributos de la clase. De esta manera, el código generado es mucho más limpio y si se añade, se elimina o se modifica un atributo automáticamente volverá a generar los métodos correspondientes. Es importante recordar

6 <http://struts.apache.org/>

7 <https://tiles.apache.org/>

8 <https://struts.apache.org/docs/json-plugin.html>

9 <https://logging.apache.org/log4j/2.x/>

10 <https://google.github.io/gson/apidocs/com/google/gson/Gson.html>

11 <http://hibernate.org/orm/downloads/>

12 <https://projectlombok.org/download.html>

que los métodos no aparecen físicamente en las clases; sin embargo, cuando desde el IDE se accede por reflexión a los miembros de una clase si aparecen.

- JUnit 4.11¹³: es una librería para java que permite realizar test unitarios al código fuente de la aplicación para comprobar si un elemento funciona como le corresponde.
- Bootstrap 3.3.7¹⁴: es una librería javascript que permite realizar aplicaciones web responsivas y minimalistas con un aspecto visual agradable. Para su correcto funcionamiento requiere de unas páginas de estilos y unos archivos de fuentes.
- jQuery 1.12.4¹⁵: es la librería javascript que proporciona la funcionalidad básica para acceder y manipular una página web. Permite el acceso al DOM de la página y sirve como base para otros plugin o librerías de terceros.
- jQuery UI 1.12.1¹⁶: es una librería javascript que proporciona funcionalidad para manejar elementos visuales de una página web. Al igual que Bootstrap requiere de una página de estilos que puede ser generada desde su web.
- jqBootstrapValidation 1.3.6¹⁷: es una librería javascript que permite realizar validaciones en caliente sobre los elementos de un formulario y mostrar mensajes de error bajo ellos cuando la información introducida no cumple los criterios de validación.
- jQuery-confirm 3.2.0¹⁸: es una librería javascript que permite mostrar mensajes de confirmación y alerta personalizados. Para su correcta visualización es necesario incluir su propia página de estilos.
- jQuery.base64 0.1¹⁹: es una librería javascript que permite codificar y decodificar cadenas de texto en formato base 64.
- jQuery MsgBox 0.8.1²⁰: es una librería javascript que aporta funcionalidad para mostrar mensajes emergentes en aplicaciones web sin realizar llamadas a los métodos nativos confirm y alert.
- Respond.js 1.4.2: es una librería javascript requerida por Bootstrap para garantizar cierta compatibilidad con navegadores de la familia Internet Explorer.

13 <http://junit.org/junit4/>

14 <http://getbootstrap.com/>

15 <https://code.jquery.com/jquery/>

16 <https://jqueryui.com/download/all/>

17 <https://reactiveraven.github.io/jqBootstrapValidation/>

18 <https://craftpip.github.io/jquery-confirm/>

19 <https://plugins.jquery.com/base64/>

20 <https://plugins.jquery.com/msgbox/>

- HTML5 Shiv 3.7.3: es una librería javascript que proporciona compatibilidad HTML con navegadores de la familia Internet Explorer para Bootstrap.
- IE10 viewport hack: es una librería javascript que proporciona compatibilidad al navegador Internet Explorer 10 para interpretar correctamente los estilos aplicados por Bootstrap.
- Leaflet 1.0.3²¹: es una librería javascript que permite mostrar mapas en páginas web así como otros tipos de datos georreferenciados. Esta API solo incluye la funcionalidad no los mapas, estos son proporcionados por la plataforma OpenStreetMap.
- Leaflet.FullScreen²²: es un plugin para Leaflet que permite visualizar el mapa a pantalla completa en el dispositivo.
- Leaflet.awesome-markers plugin v 2.0²³: es un plugin para Leaflet que permite personalizar los iconos que se utilizan en el mapa como marcadores.

21 <http://leafletjs.com/>

22 <https://github.com/brunob/leaflet.fullscreen>

23 <https://github.com/lvoogdt/Leaflet.awesome-markers>

13. Prototipos

Internamente las pantallas de la aplicación se dividen en dos grandes grupos: de acceso público y de acceso restringido. Las primeras permiten que cualquier usuario pueda visualizarlas y realizar la solicitud de acceso, mientras que las segundas solo son visibles para los usuarios que han realizado el proceso de autenticación.

13.1. Lo-Fi

Estos prototipos de baja calidad sirven como punto de partida para configurar la disposición de los distintos elementos que componen cada página. A partir de refinamientos sucesivos se obtienen los modelos Hi-Fi.

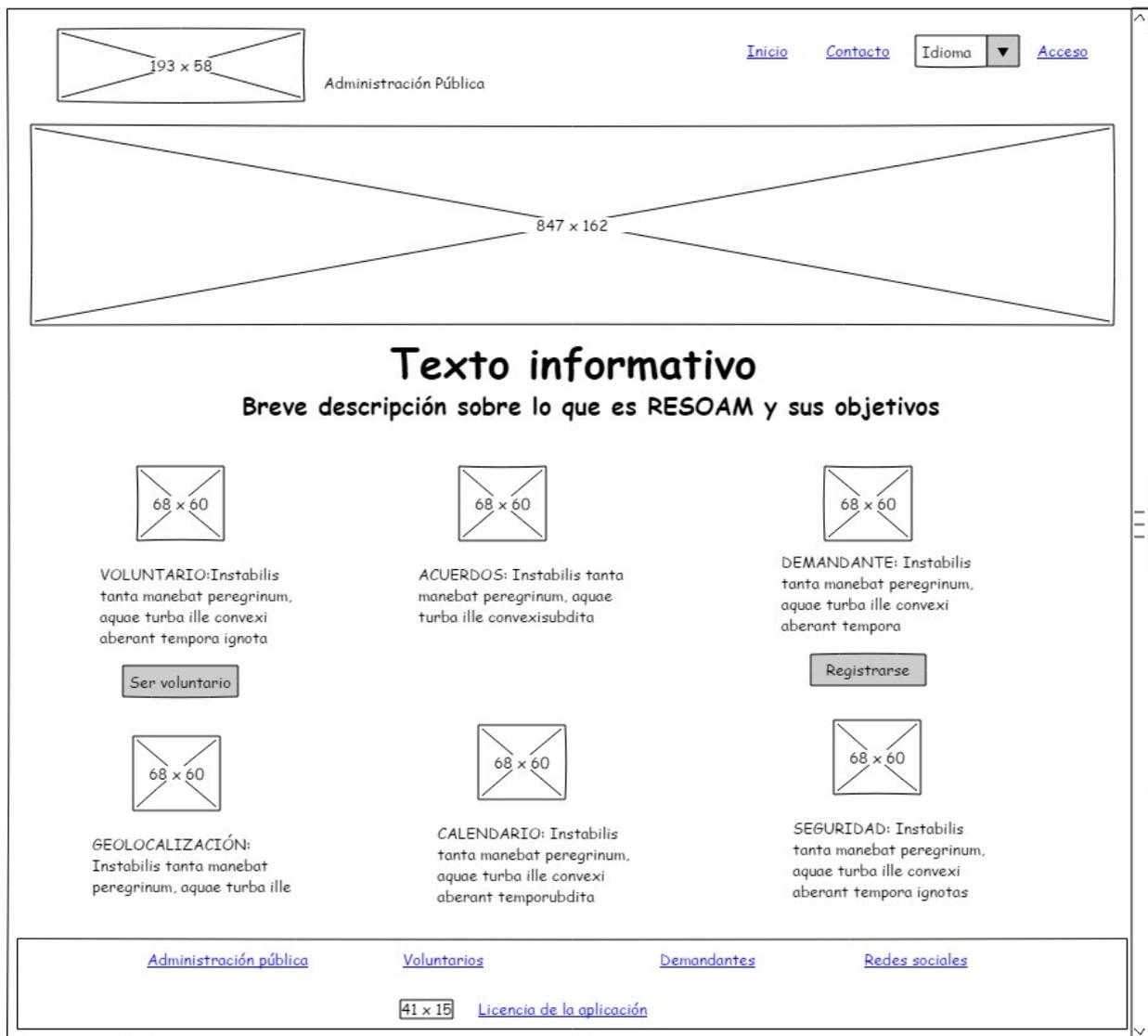


Imagen 23: Página principal

193 x 60

Administración Pública

Inicio Contacto Idioma Acceso

Alta solicitud voluntario

Label text text

Label text text

Label text text goes here

Label text text

Label text

<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					

Limpiar Aceptar

Administración pública Voluntarios Demandantes Redes sociales

41 x 15 Licencia de la aplicación

Imagen 24: Página de alta de solicitud para voluntarios

193 x 58

Administración Pública

Inicio Contacto Idioma Acceso

Alta solicitud demandante

Label text text

Label text text

Label text text goes here

Label text text

Label text

<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					
<input checked="" type="checkbox"/> Día					

Limpiar Aceptar

Administración pública Voluntarios Demandantes Redes sociales

41 x 15 Licencia de la aplicación

Imagen 25: Página de alta de solicitud para demandantes

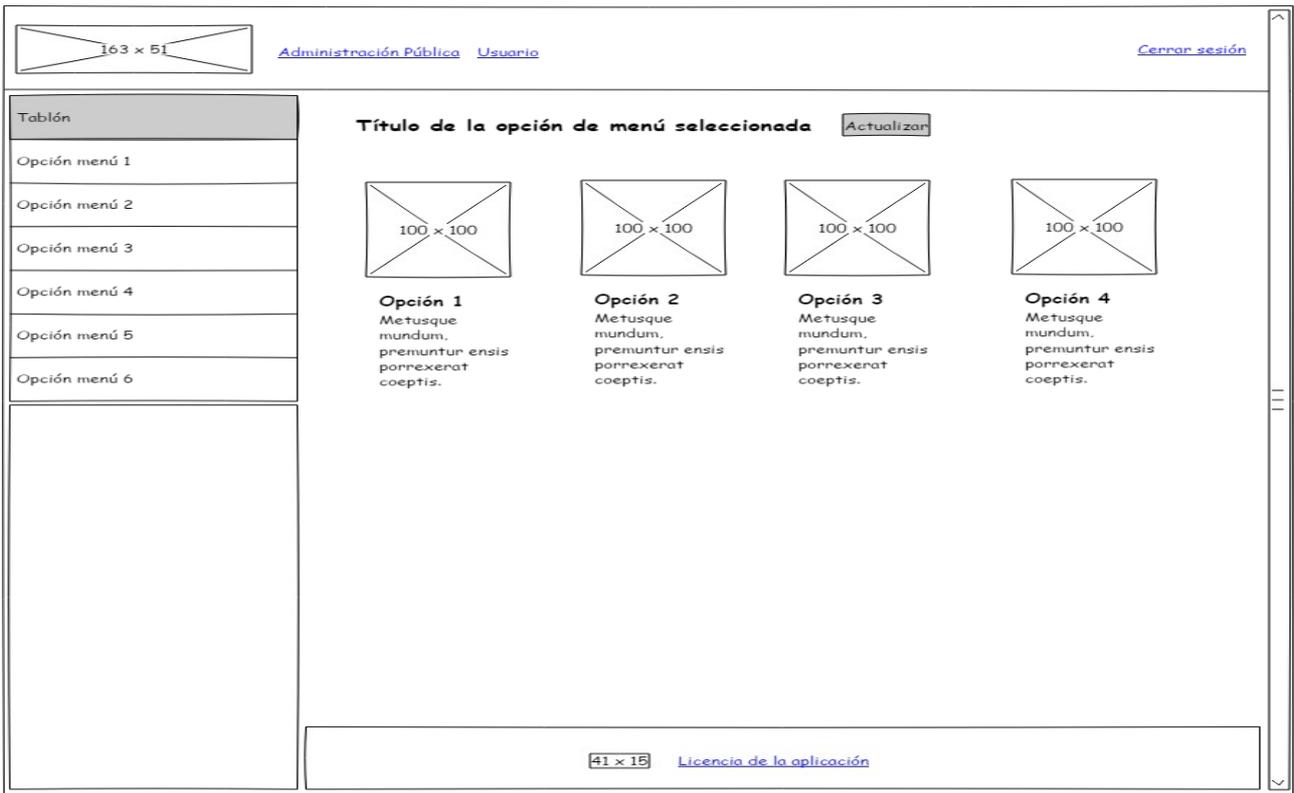


Imagen 26: Página del tablón/dashboard

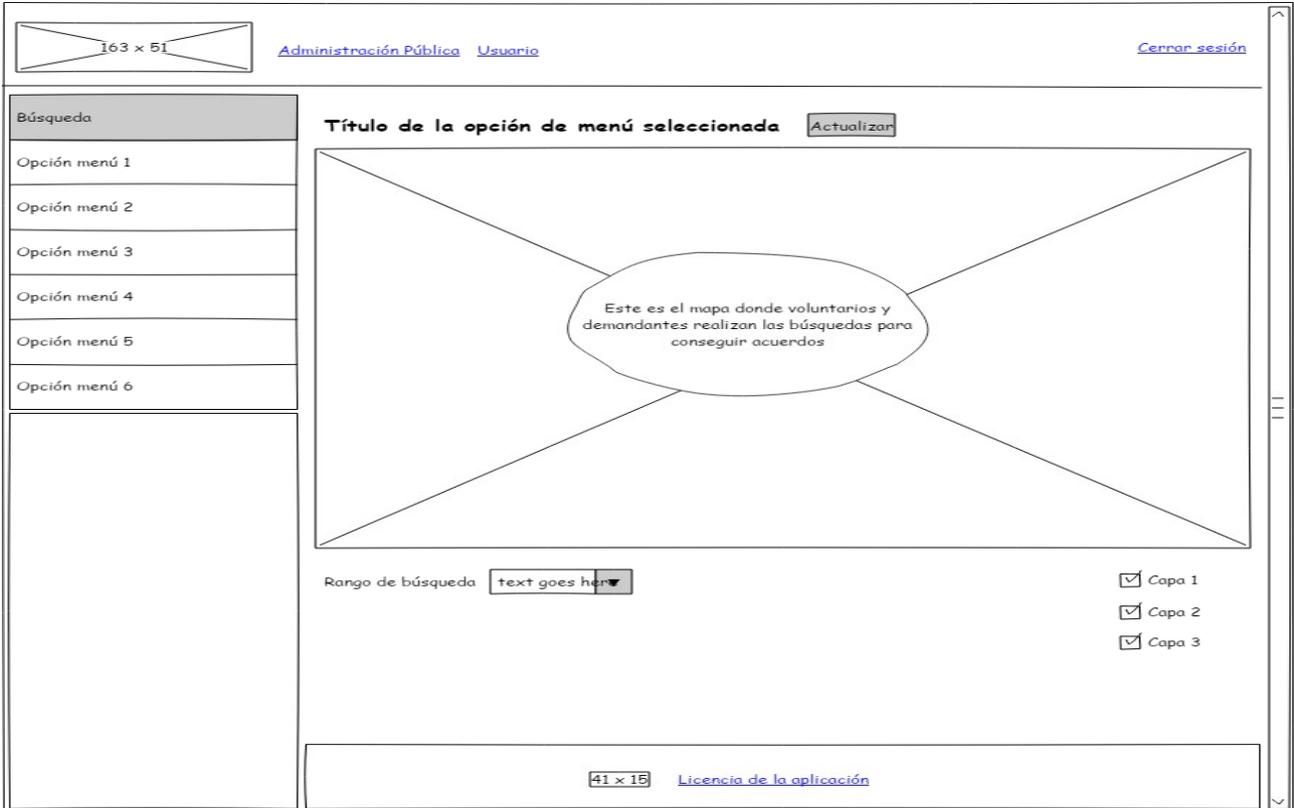


Imagen 27: Página de búsqueda de acuerdos

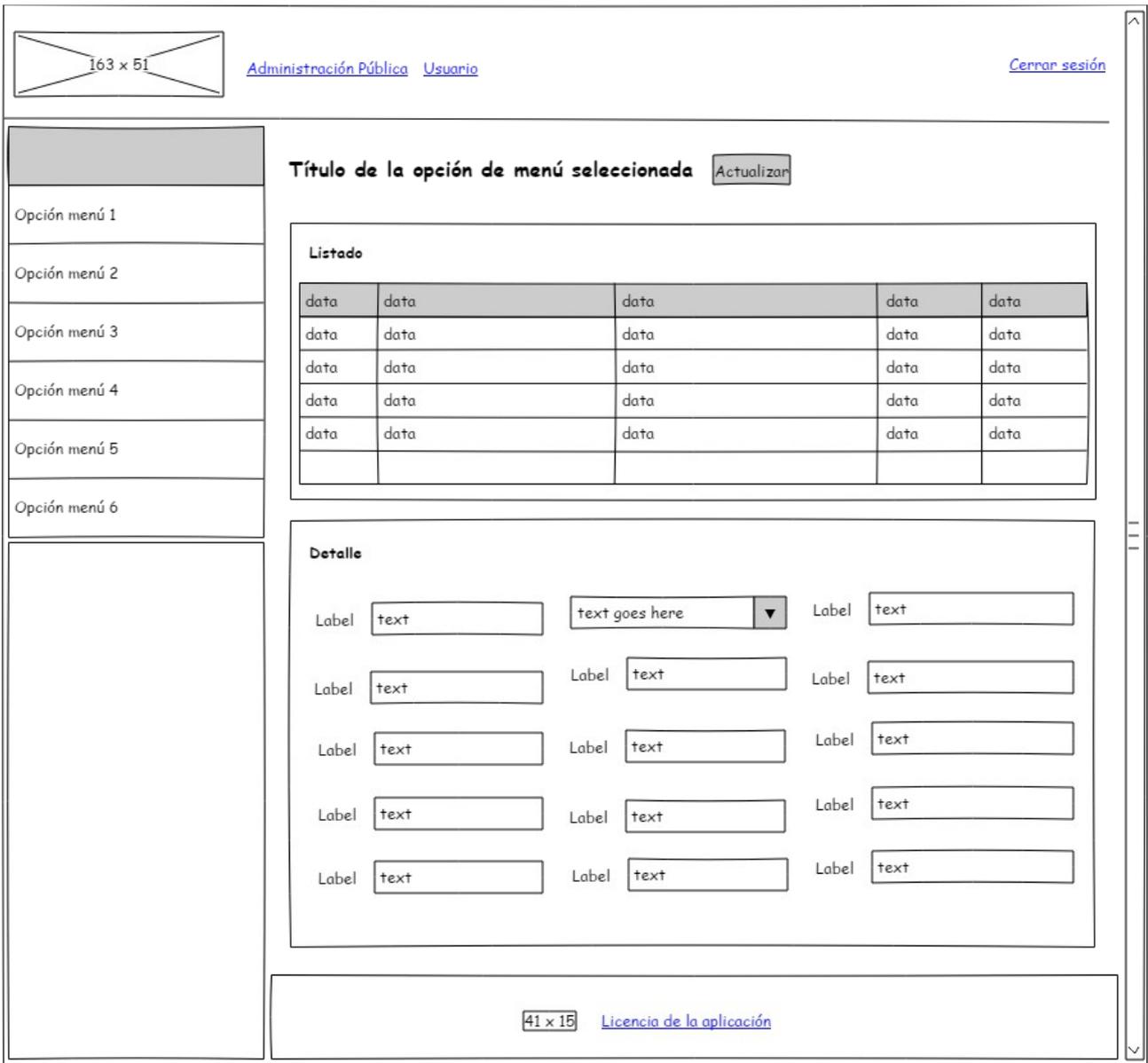


Imagen 28: Página genérica correspondiente a una opción de menú

13.2. Hi-Fi

Estos prototipos de alta calidad intentan ser un fiel reflejo del aspecto definitivo de las diferentes páginas de la web. No obstante, es posible que no coincidan exactamente con la versión final de la misma.

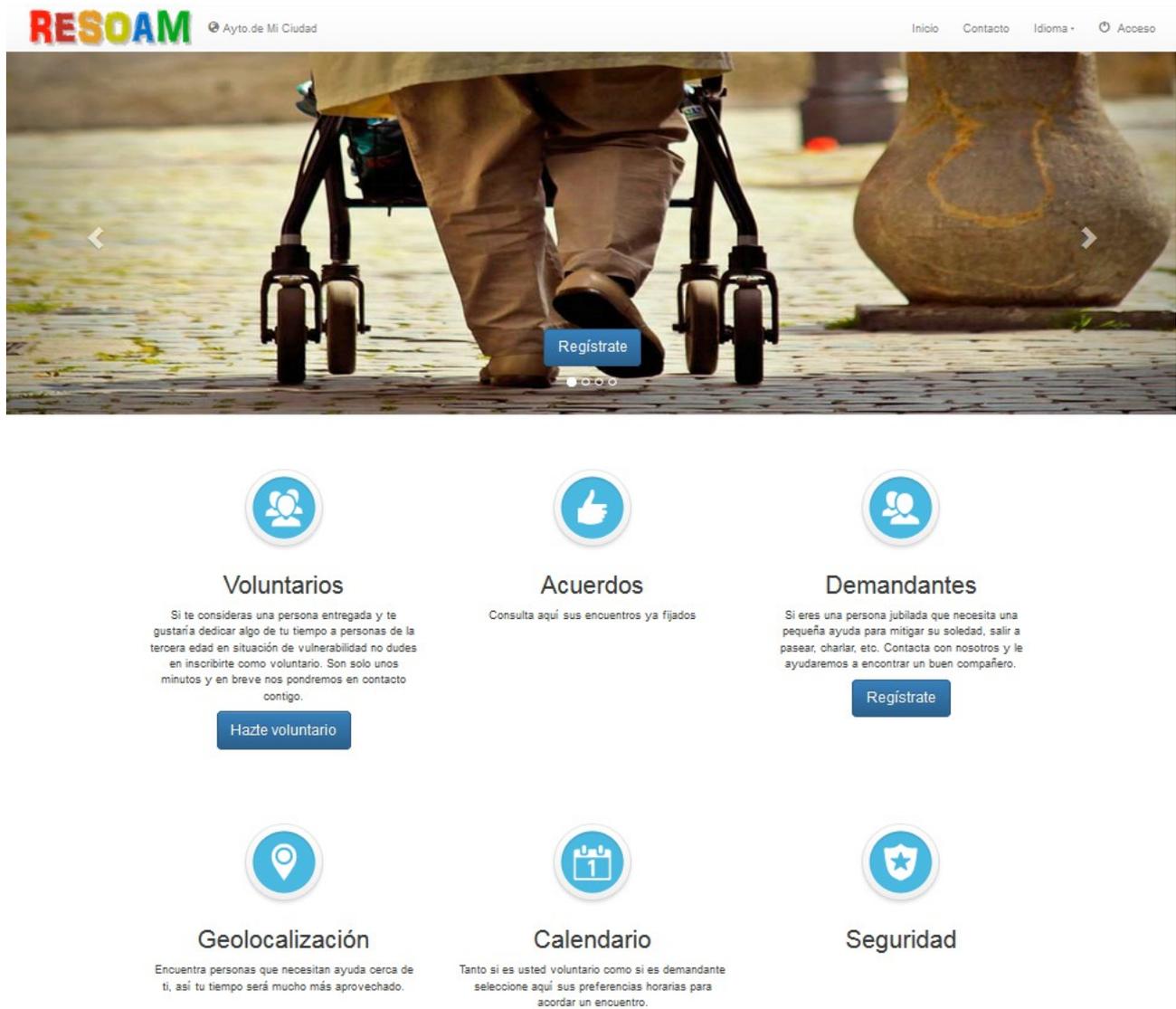


Imagen 29: Página principal (Hi-Fi)

RESOAM Ayto. de Mi Ciudad Inicio Contacto Idioma Acceso

Solicitud voluntario

Datos personales

Nif	<input type="text" value="Nif"/>				
Nombre	<input type="text" value="Nombre"/>	Apellido1	<input type="text" value="Apellido1"/>	Apellido2	<input type="text" value="Apellido2"/>
Mail	<input type="text" value="Mail"/>	Mail	<input type="text" value="Mail"/>		
Contrasena	<input type="text" value="Contrasena"/>	Contrasena	<input type="text" value="Contrasena"/>		
Foto	<input type="text" value="Foto"/>				
Direccion	<input type="text" value="Direccion"/>	Localidad	<input type="text" value="Localidad"/>	Provincia	<input type="text" value="Seleccione la prov"/>
Cp	<input type="text" value="Cp"/>	Latitud	<input type="text" value="Latitud"/>	Longitud	<input type="text" value="Longitud"/>
Fijo	<input type="text" value="Número de fijo"/>	Móvil	<input type="text" value="Número de móvil"/>		

Datos solicitud

Titulacion	<input type="text" value="Titulacion"/>
Experiencia	<input type="text" value="Experiencia"/>
Objetivos	<input type="text" value="Objetivos"/>
Distancia	<input type="text" value="Seleccione el rango"/>
Otros	<input type="text" value="Observaciones"/>



Imagen 30: Página de alta de solicitud para voluntarios (Hi-Fi)



RESOAM Ayto.de Mi Ciudad Inicio Contacto Idioma Acceso

Solicitud Demandante

Datos personales

Nif

Nombre **Apellido1** **Apellido2**

Mail **Mail**

Contrasena **Contrasena**

Foto

Direccion **Localidad** **Provincia**

Cp **Latitud** **Longitud**

Fijo **Móvil**

Datos solicitud

Situacion actual

Motivo solicitud

Distancia

Otros

Intervalos horarios disponibles

	Lu	Ma	Mi	Ju	Vi	Sá	Do
De 08:00 a 10:00	<input type="checkbox"/>						
De 10:00 a 12:00	<input type="checkbox"/>						

Imagen 31: Página de alta de solicitud para demandantes (Hi-Fi)



RESOAM Ayto.de Mi Ciudad Antonio Jiménez Mahillo Cerrar sesión

Tablón por defecto Actualizar

- Tablón por defecto
- Acuerdos
- Búsqueda de acuerdos
- Consumo horas
- Datos personales
- Mensajes
- Reclamaciones
- Solicitud

Búsqueda
Consulta de demandantes

Acuerdos
Listado de acuerdos

Mensajes
Listado de mensajes

Reclamaciones
Listado de reclamaciones

Imagen 32: Página del tablón/dashboard (Hi-Fi)

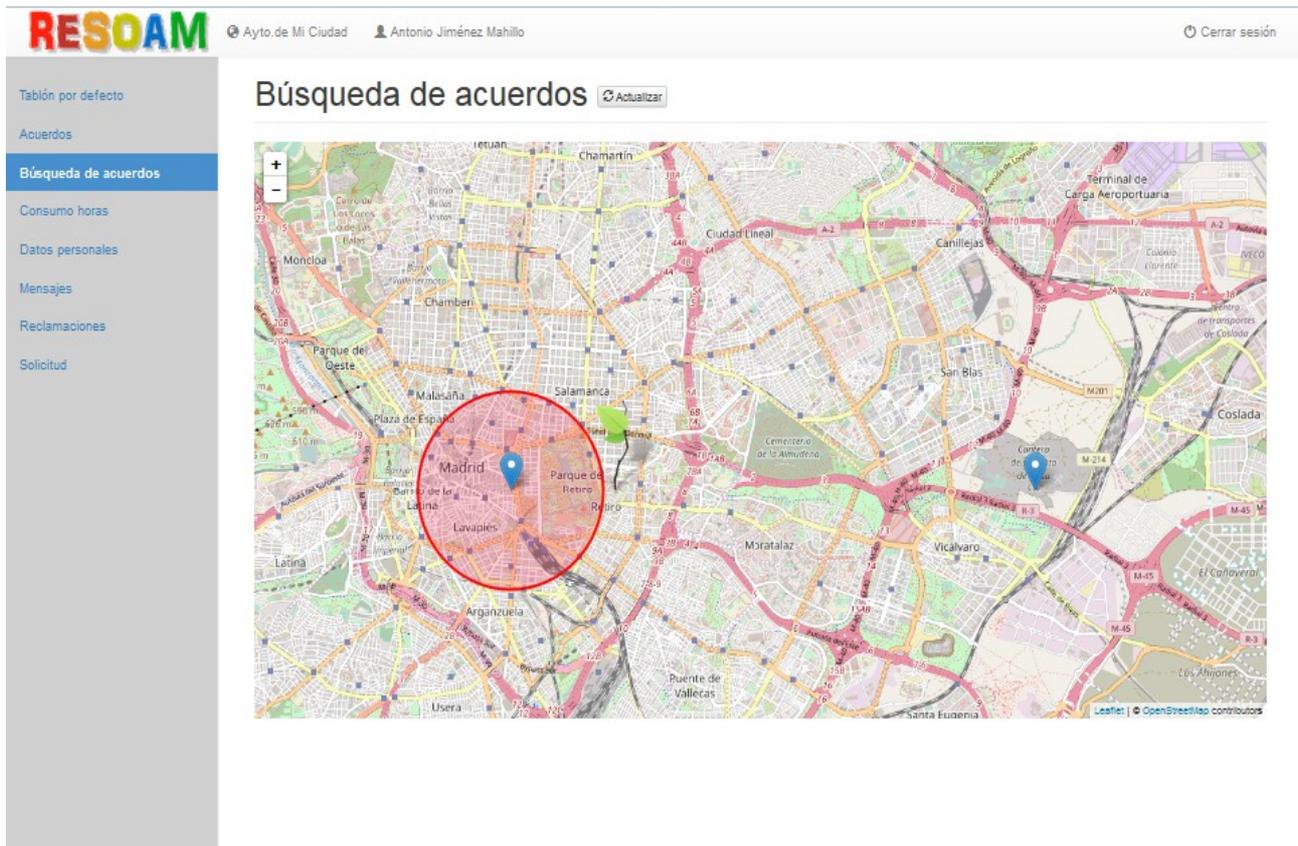


Imagen 33: Página de búsqueda de acuerdos (Hi-Fi)

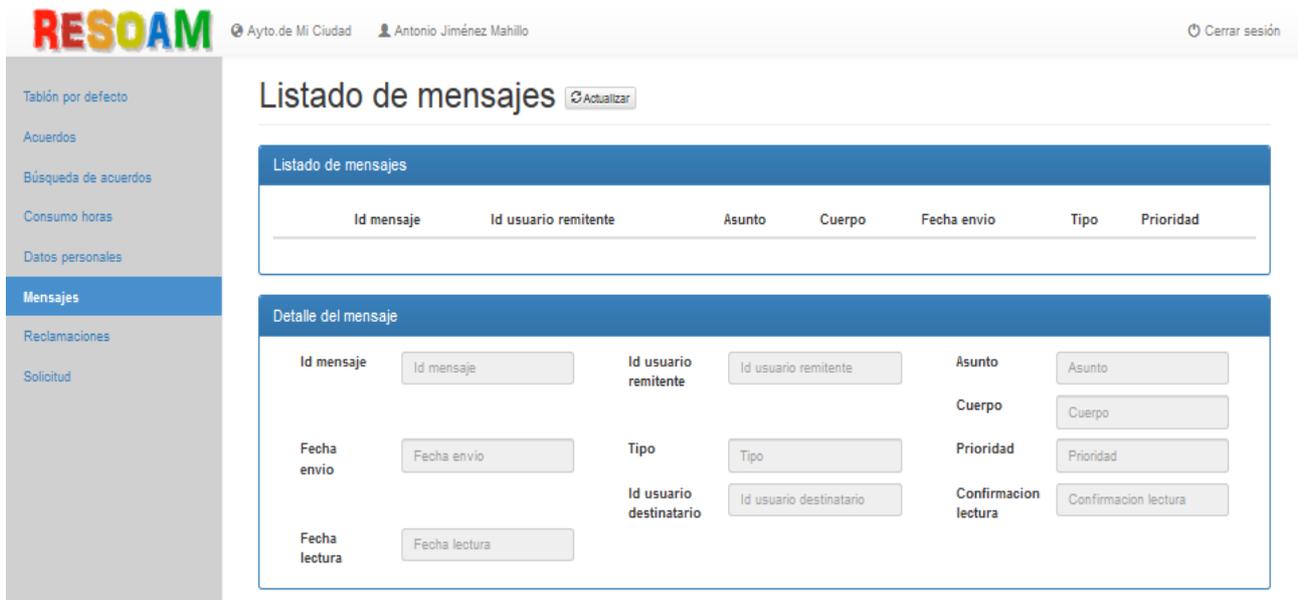


Imagen 34: Página genérica correspondiente a una opción de menú (Hi-Fi)

14. Perfiles de usuario

Los usuarios que acceden a la aplicación tienen motivaciones distintas en función de sus necesidades. Por un lado, están las personas mayores que necesitan ayuda y los voluntarios que ofrecen su tiempo de forma desinteresada. Mientras que por el otro, están las personas pertenecientes a la administración que se encargan de la gestión y el mantenimiento del sistema. Por lo tanto, el sistema requiere los siguientes perfiles de usuario:

- Demandante: es el conjunto de personas mayores con algún tipo de problema físico, mental o en situación de exclusión social, que requieren de ayuda externa a su ámbito familiar para realizar las tareas cotidianas. Representan el objetivo primario de la aplicación.
- Voluntario: es el conjunto de personas que de forma desinteresada se inscriben en el sistema y ofrecen su tiempo y sus habilidades para ayudar a personas más necesitadas. Es el segundo perfil más importante del sistema.
- Gestor: es el perfil más importante que proporciona la administración ya que se encarga de la gestión del día a día junto con sus implicaciones, como pueden ser, la revisión de solicitudes, la mediación en las disputas entre demandantes y voluntarios, así como el envío de mensajes a colectivos de usuarios.
- Administrador: es el encargado de la gestión del calendario de la aplicación y el responsable de la creación de nuevos usuarios gestores e inspectores en el sistema.
- Inspector: es el perfil con menor relevancia en el sistema ya que su número es reducido y, a nivel de la aplicación, se limita a verificar la autenticidad de los datos proporcionados por los demandantes y asignarles una valoración.

15. Usabilidad

La aplicación está especialmente diseñada para ser utilizada por personas mayores que no están habituadas al uso del ordenador y/o del teléfono móvil. Debido a esta restricción, se han realizado grandes esfuerzos a nivel de la arquitectura de la información para estructurar los contenidos y conseguir una interfaz clara y sencilla que evite ambigüedades y, permita al usuario encontrar lo que busca en apenas unos clics de ratón.

A nivel visual se ha empleado un diseño de páginas homogéneo con fuentes en color azul y negro sobre fondo blanco, lo que favorece la lectura de los textos al aumentar el contraste natural entre el fondo y el primer plano. Además, se eliminan los elementos superfluos que podrían distraer o confundir al usuario.

Los botones utilizados están sobredimensionados para favorecer su pulsación desde dispositivos móviles y se han empleado elementos visuales -iconos- que permiten reforzar el mensaje que se quiere transmitir. Por lo tanto, un icono nunca estará asociado a más de un significado, si lo estuviera sería confuso y podría ensombrecer los esfuerzos realizados.

A nivel funcional se han marcado con un asterisco de color rojo aquellos campos que son obligatorios en el momento de cumplimentar la solicitud. También se ha adaptado de forma individual todos los controles que permiten la inserción de texto para evitar que el usuario pueda introducir por error un carácter no permitido; por ejemplo, al introducir un valor numérico en un control destinado a introducir el nombre de una localidad.

La aplicación utiliza un diseño responsivo basado en Bootstrap que le permite adaptarse a la superficie de visualización del dispositivo móvil, favoreciendo de esta manera la movilidad de los usuarios y la conectividad; ya que estos pueden acceder a la aplicación en cualquier momento y desde el lugar más inverosímil.

Así mismo, se han aplicado otras medidas correctoras para permitir la fluidez de la aplicación en dispositivos móviles como la desaparición del menú lateral de la parte izquierda de la pantalla y su integración en el botón de las tres franjas horizontales situado en la parte superior de la pantalla. Igualmente, la caja de texto que aparece en la pantalla de búsqueda de acuerdos y que muestra información del voluntario/demandante se ha reducido de tamaño.

16. Seguridad

Actualmente, la seguridad de las aplicaciones web se ha convertido en un tema crítico. Es habitual que para que un usuario pueda acceder a un sitio web haya tenido que realizar previamente un proceso de registro en el que ha proporcionado información personal -que en ocasiones puede ser bastante sensible- para obtener un identificador de usuario y una contraseña. Sin embargo, las personas no siempre son conscientes del uso que las empresas hacen de sus datos. Por este motivo, la aplicación necesita estar bajo la protección de una Administración Pública que la registre en la AEPD²⁴ y se encargue de la custodia de la información.

La seguridad en este tipo de aplicaciones se implementa en varios niveles con el fin de evitar ataques informáticos, el robo de información y, el acceso no autorizado debido a fallos de programación. Para prevenir y evitar en la medida de lo posible los problemas de este tipo la plataforma RESOAM implementa los siguientes mecanismos:

- Comunicación: para evitar que un atacante pueda visualizar la información que se intercambia entre el equipo del cliente y el servidor, la navegación se puede cifrar si se utiliza el protocolo HTTPS. Para ello, es necesario obtener un certificado válido que se deberá instalar en el servidor web.

La aplicación hace uso intensivo de la tecnologías AJAX y jQuery desde la parte cliente para realizar las peticiones de datos o solicitar al servidor que realice una determinada acción. Existen dos métodos para el envío de datos GET y POST; el primero es el más inseguro porque plantea numerosos problemas de seguridad y por ello no se utiliza.

Si se intenta invocar a alguno de los actions de la aplicación utilizado GET el servidor lo detectará y lanzará una excepción de seguridad que provocará que la navegación se redirija hacia la página de error interno del servidor (500.jsp).

- Identificación: para acceder a la aplicación se debe proporcionar un identificador y una contraseña que solo el usuario conoce; por lo que las contraseñas tienen que cumplir unas características de seguridad como contener caracteres en mayúsculas, en minúsculas, números y caracteres especiales, con el fin de evitar que puedan ser descubiertas fácilmente mediante ataques por fuerza bruta.

24 <https://www.agpd.es>

Además, las contraseñas no se almacenan en texto claro en la base de datos. Lo que se almacena es una cadena de texto que ha sido obtenida con un algoritmo de cifrado asimétrico y que es una representación de la contraseña real. El algoritmo utilizado es MD5 aunque se puede reemplazar de forma sencilla por otro que proporcione un nivel superior de seguridad.

- Responsabilidad: desde el momento en el que un usuario accede a la aplicación y se identifica se le asocia un perfil. Así que las páginas a las que puede acceder y las funcionalidades que puede ejecutar están limitadas. Por ejemplo, si accede como un usuario demandante no podrá configurar el calendario del sistema que solo está disponible para el administrador de la aplicación.
- Formularios: todos los campos de los formularios realizan validaciones automáticas en la parte del cliente que impiden la introducción de datos incorrectos o caracteres no permitidos. Posteriormente estos se codifican en base64 en dos fases y se envían al servidor donde se descodifican y de forma individual se vuelven a revisar para eliminar los caracteres no permitidos y evitar la inyección de código maligno.
- Autoría: aunque realmente en esta versión no se implementa, es importante mencionarla. Básicamente el proceso de autoría consiste en mantener un registro histórico de las acciones que ejecuta cada usuario con el fin de que puedan ser analizadas posteriormente si se produce algún incidente.

17. Test

La capa de representación de la aplicación utiliza un proceso estandarizado de creación de objetos y de llamada a las funciones de la capa de negocio para realizar las acciones requeridas. Mediante la herramienta junit es factible programar la ejecución de una serie de test unitarios que permitan verificar si las funciones de la capa de datos realizan su tarea de forma adecuada.

```
package Usuario;

import org.junit.Test;

import com.ajmahillo.resoam.persistence.entities.Usuario;
import com.ajmahillo.resoam.persistence.entities.Voluntario;
import com.ajmahillo.resoam.persistence.services.IUsuarioService;
import com.ajmahillo.resoam.persistence.services.impl.hibernate.UsuarioServiceImplHibernate;
import com.ajmahillo.resoam.persistence.utils.HibernateConnector;
import java.math.BigDecimal;

public class UsuarioTest {

    @Test
    public void test() {

        HibernateConnector.createSession();
        IUsuarioService service=new UsuarioServiceImplHibernate();
        HibernateConnector.getSession().beginTransaction();
        Usuario usuario=new Usuario();
        usuario.setApellido1("Pérez");
        usuario.setApellido2("López");
        usuario.setCorreo("correo@server.com");
        usuario.setContrasena("12345AvB");
        usuario.setEstado((byte)1);
        usuario.setNif("57678895M");
        usuario.setNombre("Susana");
        service.altaUsuario(usuario);

        Voluntario voluntario=new Voluntario();
        voluntario.setCp("11111");
        voluntario.setDireccion("Calle fraguas, 23");
        voluntario.setAntecedentes(false);
        voluntario.setLatitud(new BigDecimal(0.0));
        voluntario.setLongitud(new BigDecimal(0.0));
        voluntario.setLocalidad("Madrid");
        voluntario.setProvincia("Madrid");
        voluntario.setTelefono1("916836555");
        voluntario.setTelefono2("661787656");
        voluntario.setUsuario(service.recuperarUsuarioPorNIF("57678895M"));
        service.altaVoluntario(voluntario);

        HibernateConnector.getSession().getTransaction().rollback();
        HibernateConnector.close();
    }
}
```

18. Requisitos de instalación

La instalación se realiza en una única máquina física que alberga un servidor web, un contenedor de servlets y un sistema gestor de base de datos. En un entorno real de producción, lo habitual es que se utilicen varios servidores y el acceso a los mismos esté restringido. Generalmente, algún miembro del departamento de sistemas de la administración donde se implante la aplicación, se encargará de configurar los equipos y revisar que se cumplen los siguientes requisitos:

- Apache web server 2.2
 - Java Runtime Environment 1.0.8_101-b13
 - Instalar el certificado digital que permite cifrar las conexiones e identificar al servidor.
 - Crear la carpeta resoam en documentRoot.
 - Configurar el acceso por FTP a la carpeta resoam para un usuario con permisos de lectura y escritura.
 - Verificar la comunicación con el contenedor de servlets.
 - Habilitar los módulos:
 - core_module (static).
 - so_module (static).
 - watchdog_module (static).
 - http_module (static).
 - log_config_module (static).
 - logio_module (static).
 - version_module (static).
 - unixd_module (static).
 - access_compat_module (shared).
 - alias_module (shared).
 - auth_basic_module (shared).
 - authn_core_module (shared).
 - authn_file_module (shared).
 - authz_core_module (shared).
 - authz_host_module (shared).
 - authz_user_module (shared).
 - autoindex_module (shared).
 - deflate_module (shared).

- dir_module (shared).
 - env_module (shared).
 - filter_module (shared).
 - headers_module (shared).
 - mime_module (shared).
 - mpm_prefork_module (shared).
 - negotiation_module (shared).
 - proxy_module (shared).
 - proxy_http_module (shared).
 - remoteip_module (shared).
 - reqtimeout_module (shared).
 - rewrite_module (shared).
 - setenvif_module (shared).
 - socache_shmcb_module (shared).
 - ssl_module (shared).
 - status_module (shared).
-
- Apache tomEE 7.0.2
 - Java Runtime Environment 1.0.8_101-b13.
 - Crear un usuario con permiso para acceder a la consola web.
 - Instalar las librerías:
 - javax.servlet-api
 - jstl
 - log4j
 - junit
 - lombok
 - gson
 - hibernate-core 4.0.1.Final
 - hibernate-validator 4.2.0.Final
 - hibernate-commons-annotations 4.0.1.Final
 - hibernate-entitymanager 4.1.8.Final
 - Verificar la comunicación con el servidor de base de datos.
 - Configurar el pool de conexiones que estará disponible para la aplicación vía JNDI. Se debe revisar el contenido de los siguientes archivos:

```

Archivo context.xml del servidor Apache tomEE
...
<Resource auth="Container"
  driverClassName="com.mysql.jdbc.Driver"
  factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
  type="javax.sql.DataSource"
  url="jdbc:mysql://192.168.1.150:3306/resoam"
  name="jdbc/mysql_resoam"
  username="resoam"
  password="resoam"
  maxActive="50"
  maxIdle="20"
  maxWait="3000"
  initialSize="20"
  defaultAutoCommit="false"
  removeAbandoned="true"
  removeAbandonedTimeout="60"
/>
...

```

```

Archivo web.xml del proyecto resoam-web
...
<!-- Configuración de acceso al pool de conexiones del Apache TomEE -->
<resource-ref>
  <description>Pool de conexiones resoam</description>
  <res-ref-name>jdbc/mysql_resoam</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
...

```

```

Archivo hibernate.cfg.xml del proyecto resoam-persistence
...
<!-- Conexión mediante JNDI -->
<property name="connection.datasource">java:comp/env/jdbc/mysql_resoam</property>
...

```

- MySQL 5.5.54
 - Java Runtime Environment 1.0.8_101-b13.
 - Configurar el acceso por SSH.

```

Archivo my.cnf
...
[mysqld]
lower_case_table_names=1
...

```

19. Instrucciones de instalación

Para realizar la instalación se deben seguir los siguientes pasos de forma secuencial y sin alteraciones para asegurar el correcto funcionamiento de la aplicación.

1. Conectar con el servidor de base de datos por SSH y ejecutar los scripts `creacion_bbdd.sql` y `carga_datos_inicial.sql`.

```
mysql -u root -p resoam < archivo.sql
```

2. Conectar por SSH con el servidor Apache. Si el servicio `apache2` está corriendo se debe detener.

```
service apache2 stop
```

3. Crear un nuevo `virtualHost`, el proceso puede ser algo distinto en función de la distribución de linux utilizada y la configuración del servidor. En este caso, se crea el archivo en la ruta `/etc/apache2/sites-available`.

```
<VirtualHost *:443>
    ServerName ajmahillo.sytes.net
    DocumentRoot /var/www3/ajmahillo.sytes.net

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/ajmahillo.sytes.net.crt
    SSLCertificateKeyFile /etc/apache2/ssl/ajmahillo.sytes.net.key
    SSLCertificateChainFile /etc/apache2/ssl/ajmahillo.sytes.net.crt

    ProxyRequests Off
    ProxyPreserveHost On
    RewriteEngine On

    ProxyPass /resoam/images/ !
    ProxyPass /resoam/css/ !
    ProxyPass /resoam/js/ !
    ProxyPass /resoam/fonts/ !
    ProxyPass /resoam/index.html !

    ProxyPass /resoam/ http://192.168.1.150:8080/resoam/
    ProxyPassReverse /resoam/ http://192.168.1.150:8080/resoam/

    ErrorLog ${APACHE_LOG_DIR}/ajmahillo.sytes.net_log
    CustomLog ${APACHE_LOG_DIR}/ajmahillo.sytes.net_access_log combined

    DirectoryIndex index.html
</VirtualHost>
```

- Descomprimir el contenido estático del archivo resoam.tar.gz. Al igual que en el punto anterior esta carpeta puede ser diferente en función de la configuración del servidor. En este caso la ruta utilizada es /var/ww3/ajmahillo.sytes.net/resoam.

```
tar -zxvf resoam.tar.gz
```

- Iniciar el servicio apache2.

```
service apache2 start
```

- Si no está activo el servidor Apache tomEE se debe iniciar.

```
bash /opt/apache-tomee-webprofile-7.0.2/bin/catalina.sh run
```

- Acceder a la consola web del servidor tomEE con el usuario proporcionado. Pulsar el botón seleccionar archivo para localizar resoam.war y, finalmente pulsar el botón desplegar que provocará la instalación de la aplicación en el servidor.

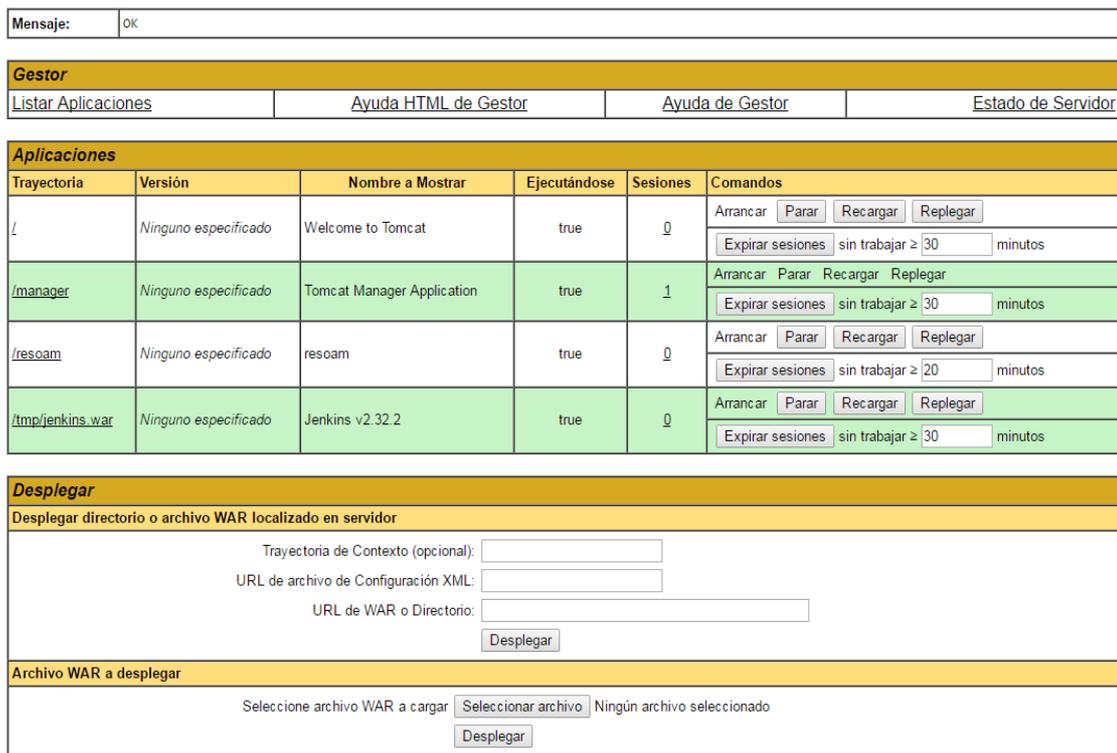


Imagen 35: Consola web de administración Apache tomEE

- Acceder a la página principal de la aplicación (la del servidor Apache no la del tomEE) y verificar que funciona. La página se debería presentar normalmente aunque los recursos estáticos los está proporcionando un servidor y los dinámicos otro.

20. Instrucciones de uso

La aplicación permite el acceso de usuarios hasta con cinco tipos de roles diferentes. Cada uno de estos roles tiene permitida la realización de unas acciones específicas -consultar el punto 5-. Indicar las instrucciones necesarias para cada una de las acciones que ejecuta cada rol sería un trabajo que se escapa del ámbito de este documento. Por lo tanto, solo se considerará a modo de ejemplo el rol de demandante.

20.1. Acceso

El primer paso para un demandante consiste en llegar a la página de inicio de la aplicación. Para ello tendría que conocer la dirección de la misma o acceder de forma indirecta a través de algún enlace en otra página web o en un mensaje de correo electrónico.

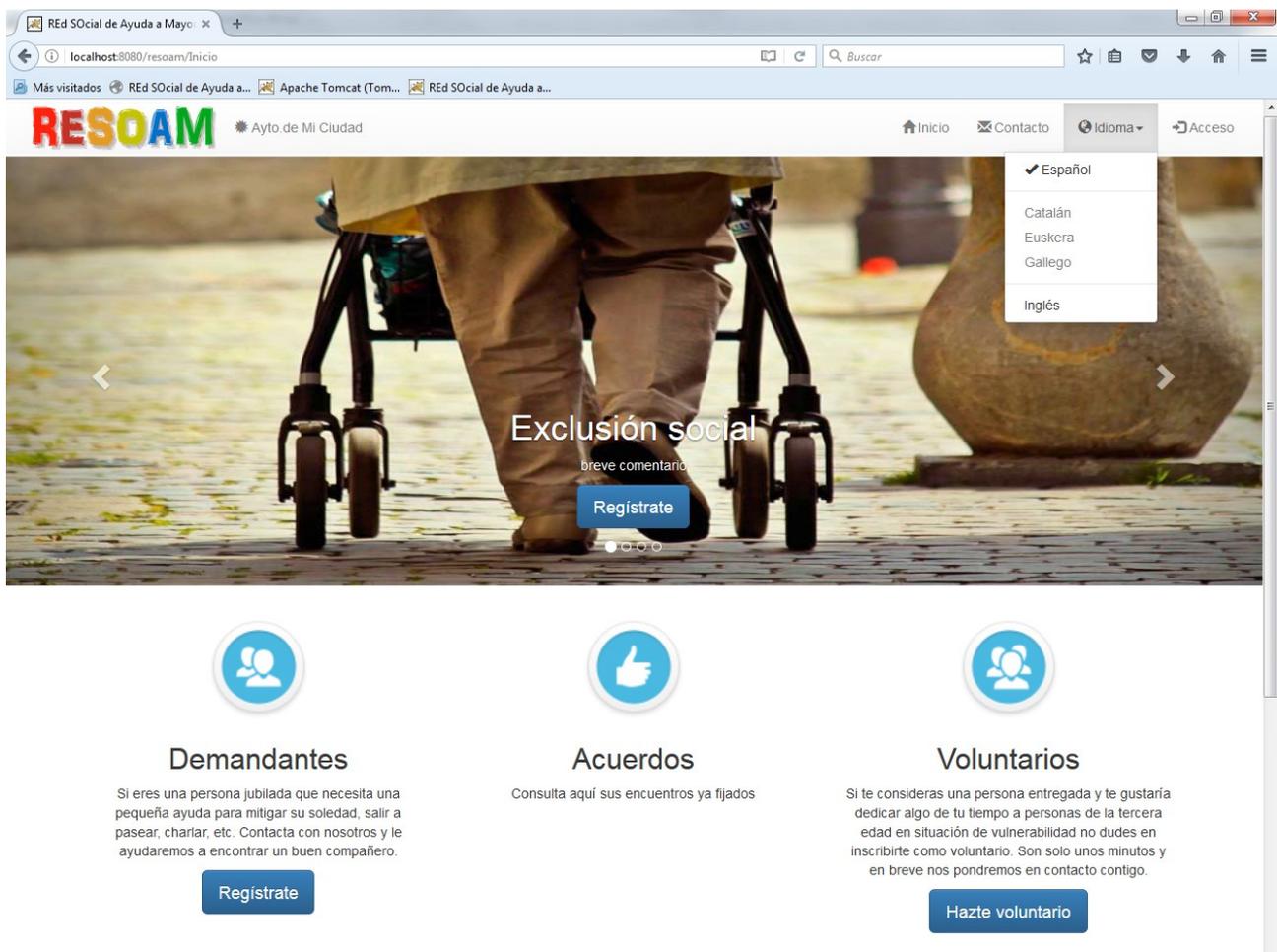


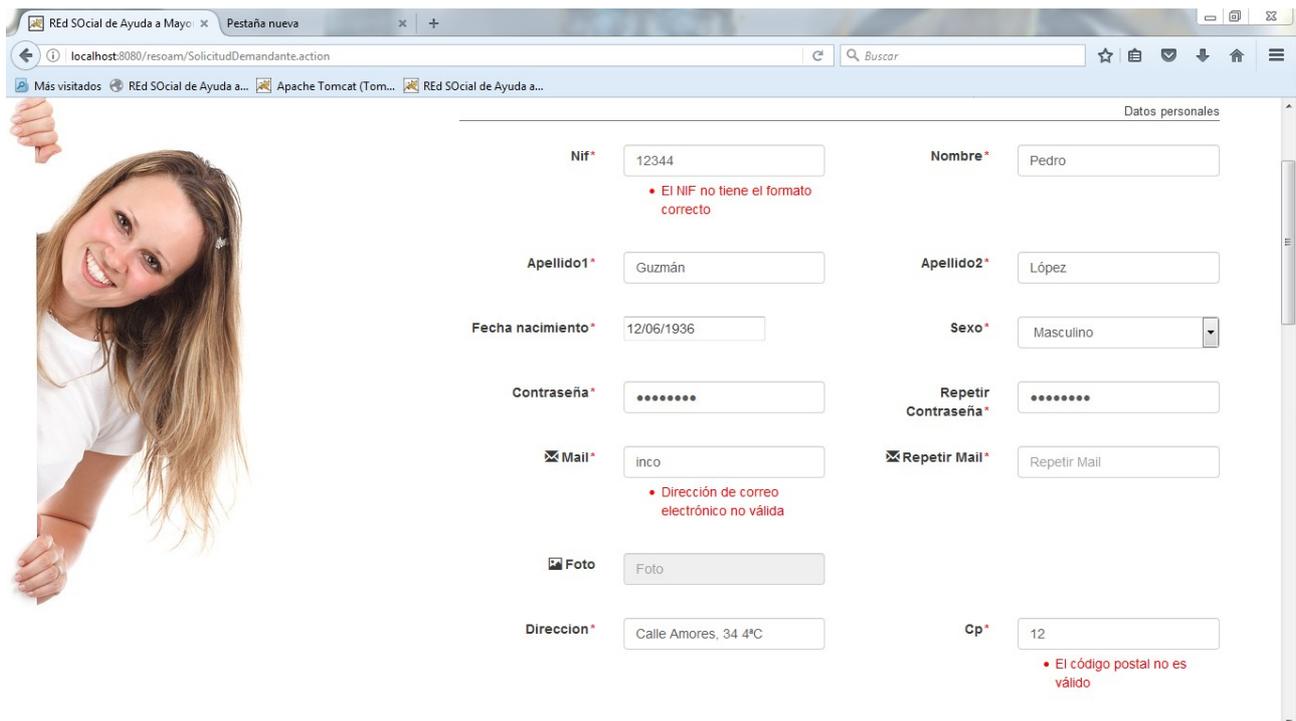
Imagen 36: Selección de idioma

Este es el aspecto de la página principal de la web. En la parte superior de la pantalla se encuentra el menú donde el usuario podrá establecer el idioma o lengua cooficial en el que quiera que se muestren los textos. Inicialmente solo están disponibles las opciones español e inglés.

El siguiente paso consiste en pulsar en uno de los botones de registro que aparecen tanto en el carrusel como en el cuerpo y en el pie de página y que realizará la navegación hacia la pantalla de registro de solicitante.

20.2. Registro

En esta pantalla el demandante debe cumplimentar de forma obligatoria todos los campos que aparecen marcados con un asterisco de color rojo y seleccionar los rangos horarios disponibles. Las cajas de texto están configuradas para admitir únicamente caracteres del tipo permitido y, en muchos casos, también se valida de forma automática que dichos caracteres estén colocados en su posición correcta.



The screenshot shows a web browser window displaying a registration form titled "Datos personales". The form contains several input fields with red error messages:

- Nif***: 12344. Error: "El NIF no tiene el formato correcto".
- Nombre***: Pedro.
- Apellido1***: Guzmán.
- Apellido2***: López.
- Fecha nacimiento***: 12/06/1936.
- Sexo***: Masculino.
- Contraseña***: [Redacted].
- Repetir Contraseña***: [Redacted].
- Mail***: inco. Error: "Dirección de correo electrónico no válida".
- Repetir Mail***: Repetir Mail.
- Foto**: Foto.
- Direccion***: Calle Amores, 34 4ºC.
- Cp***: 12. Error: "El código postal no es válido".

Imagen 37: Detección de errores en la primera fase de validación

El proceso de validación de datos consta de tres fases. La primera se lleva a cabo mientras el usuario introduce información en los controles, algunos de ellos muestran un pequeño texto en color rojo bajo el control informando al usuario que los datos introducidos no son correctos. En esta fase se consigue que la mayoría de los datos introducidos tengan el formato correcto.

La segunda fase se lleva a cabo cuando el usuario pulsa el botón de Aceptar. En ese momento se comprueba que haya rellenado todos los campos de texto obligatorios y que haya seleccionado opciones válidas para los controles desplegable. Igualmente se verifica que el correo y la contraseña -se introducen dos veces- coincidan con sus respectivas comprobaciones. La contraseña debe cumplir unos criterios mínimos de seguridad como tener una longitud obligatoria de ocho posiciones y, contener al menos un carácter en mayúsculas, otro en minúsculas y un número. Adicionalmente están permitidos los siguientes caracteres - _ . , ! \$ % & = @] * \$.

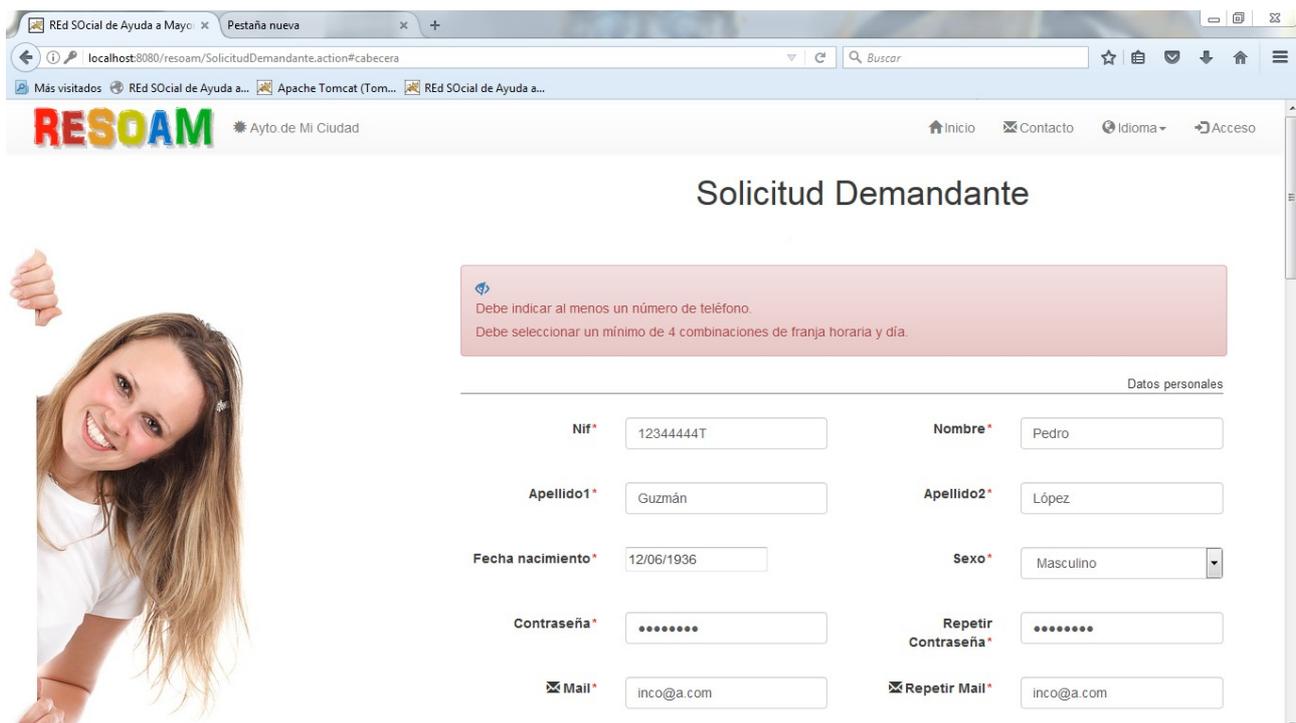


Imagen 38: Detección de errores en la segunda fase de validación

La tercera fase se ejecuta en el servidor web justo antes de realizar el registro en la base de datos de la aplicación. Nuevamente se verifica la integridad de los datos proporcionados por el demandante. Si se produce algún error en el servidor durante el proceso de registro, este se indicará en un recuadro de color rojo en la parte superior de la pantalla. Por el contrario, si el proceso finaliza correctamente, se redirige al usuario a una página que se lo indica de una manera más visual.

20.3. Identificación

El acceso a la aplicación se realiza pulsando sobre la opción de menú Acceso ubicada en la parte superior de la pantalla. Aparecerá un formulario de identificación que el usuario deberá cumplimentar para acceder a la aplicación. Como identificador podrá utilizar indistintamente su

NIF o su dirección de correo electrónico. La contraseña será la proporcionada en la ficha de registro. Una vez completado el formulario debe pulsar el botón Aceptar.

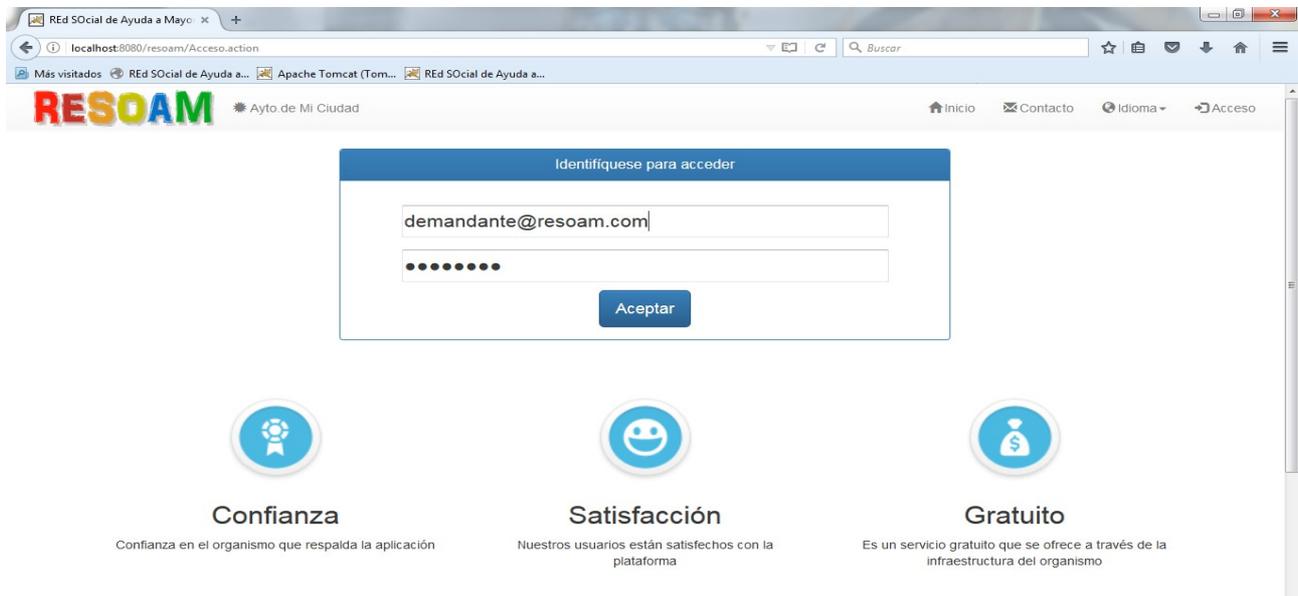


Imagen 39: Formulario de identificación de usuario

Aunque el demandante ya puede acceder a la aplicación, este tiene la mayor parte de la funcionalidad deshabilitada. Solo puede acceder a la consulta de mensajes -en el caso que el gestor quiera comunicarse con él- y a los datos de la solicitud. Cuando el gestor valide la solicitud aparecerán todas las opciones disponibles en el menú

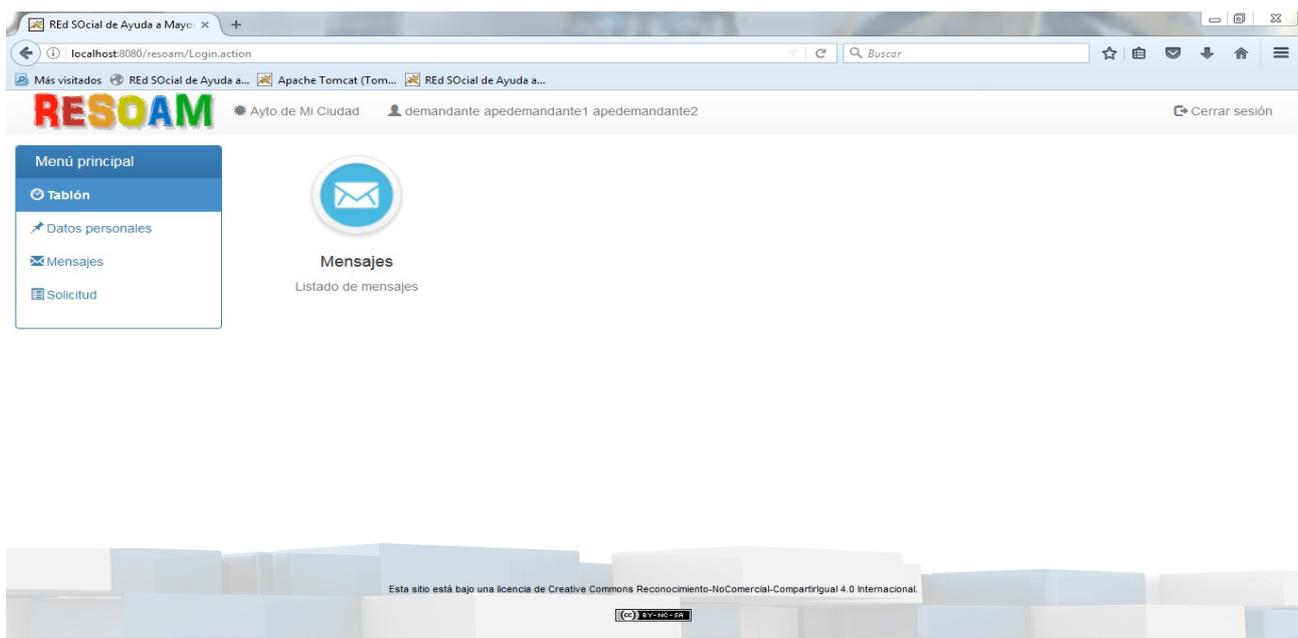


Imagen 40: Funcionalidades limitadas

20.4. Acuerdos

Esta es la funcionalidad más importante de la aplicación y la que permite a los demandantes cumplir sus expectativas. Consta de un mapa georreferenciado en el que primeramente se establece la posición del usuario mediante un icono color púrpura. Este se ubica en el centro de un círculo semitransparente que indica el área de influencia -no tiene ningún impacto funcional-.

A continuación, se recuperan del servidor los datos de los voluntarios que se encuentran hasta un máximo de 10 kilómetros de distancia del demandante y se ubican en las coordenadas correspondientes.

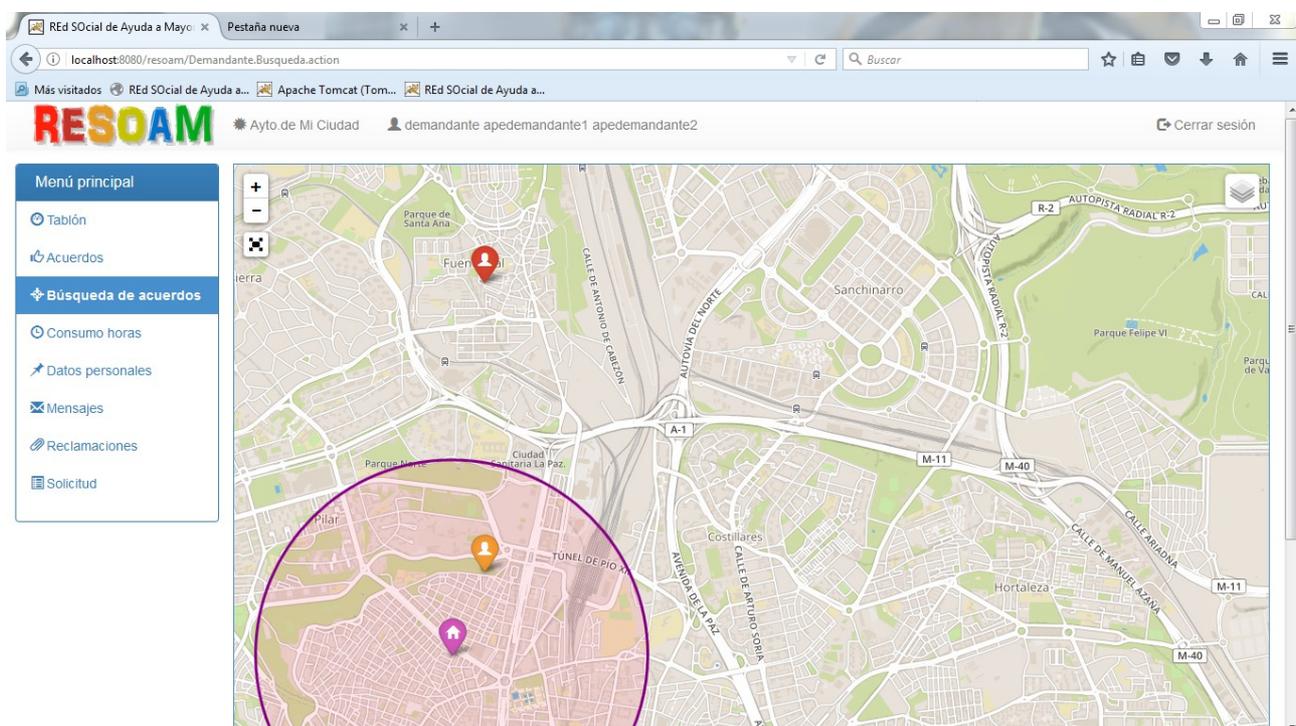


Imagen 41: Búsqueda de voluntarios

En la parte superior del mapa se ubican los controles que añaden cierta funcionalidad. A la izquierda los que permiten al usuario aumentar o disminuir el nivel de detalle del mapa y a la derecha el que permite mostrar u ocultar las capas de los usuarios. En función de la distancia que separa al demandante de los voluntarios.

El demandante puede pulsar sobre cualquiera de los iconos de los voluntarios para consultar la información que proporcionaron en la ficha de solicitud. Cuando considera que un voluntario es de su agrado inicia el proceso de tramitación de un acuerdo. Es un proceso sencillo que se realiza en

tres clics; primero debe seleccionar el día, luego la franja horaria y finalmente pulsar el botón Enviar. Cuando el voluntario entre en la aplicación y consulte la pantalla de mensajes podrá decidir si acepta o no la solicitud de acuerdo. Independientemente de la opción elegida se enviará un mensaje al demandante con la respuesta.

Los iconos de estos se muestran en un color diferente para facilitar visualmente su ordenación espacial en el mapa. En la tabla número 4 se detallan los valores utilizados.

Distancia	Color
Hasta 250m.	Verde
Entre 250m y 500m.	Verde oscuro
Entre 500m y 1000m.	Azul claro
Entre 1000m y 1500m.	Naranja
Entre 1500m y 2000m.	Rojo
Más de 2000m.	Rojo oscuro

Tabla 4: Leyenda de iconos en función de la distancia

Existe una restricción a la hora de realizar un acuerdo, este no se puede llevar a cabo para el día actual ni con posterioridad a un mes y medio. De esta forma se evita generar acuerdos que pueden ser difíciles de realizar.

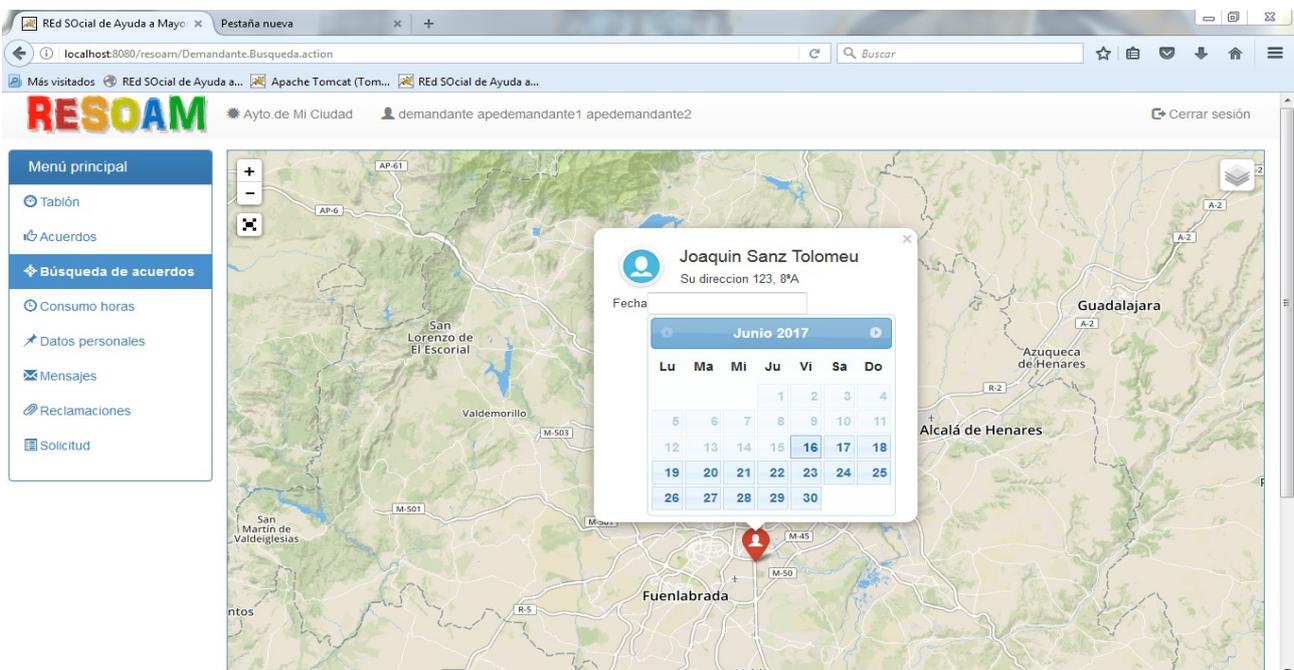


Imagen 42: Selección de la fecha del acuerdo

21. Presupuesto

El siguiente presupuesto incluye los costes derivados de los recursos humanos y materiales. Estos últimos no serían necesarios si el proyecto es gestionado por una Administración Pública.

Tipo	Descripción		Importe	
Software	Contratar dominio web		14	
Hardware	Alojamiento web anual		150	
Mantenimiento	De tipo anual		300	
Otros	Contingencias		600	
Total costes recursos materiales (€)			1.064	
Descripción	Categoría	Total horas	Coste €/H	Importe
Fase de análisis	JP	24	37	888
	AF	80	30	2.400
	TBD	40	17,5	700
	TS	10	17,5	175
Fase de diseño	AT	160	25	4.000
	DS	60	17,5	1.050
	JP	16	37	592
Fase de construcción	AT	80	25	2.000
	P	300	20	6.000
	JP	8	37	296
Fase de implantación	TS	8	17,5	140
	JP	8	37	296
Total costes recursos humanos (€)				18.537
Subtotal (Costes recursos materiales + Costes recursos humanos) (€)				19,601,00
IVA (23%) (€)				4,508,23
Total (€)				24.109,23
<p>Relación coste €/H JP: 37 AF:30 AT:25 P:20 TS:17,5 TBD:17,5 DS:17,5</p> <p>JP-Jefe de Proyecto AF-Analista Funcional AT-Analista Técnico P-Programador TS-Técnico de Sistemas TBD-Técnico de base de datos DS-Diseñador gráfico</p>				

Tabla 5: Presupuesto de realización del proyecto.

22. Análisis de mercado

El desarrollo de este producto tiene un fin puramente social y se asienta en dos grandes pilares: las personas mayores en riesgo de exclusión y los voluntarios; ambos componen el target de la aplicación. Esta se desarrolla por y para ellos. Además, en ningún caso se pretende conseguir el más mínimo beneficio económico ni recibir ningún tipo de contraprestación. Una vez terminado el desarrollo de la aplicación se ofrecerá de forma gratuita a las Administraciones Públicas para que puedan evaluarla.

La plataforma RESOAM no es única en su nicho de mercado, existen otras aplicaciones similares que utilizan diversos sistemas de negocio como el modelo de pago por servicio, la franquicia y el servicio gratuito. Sin embargo, esta tiene una serie de características que la hacen diferente y le aportan un valor añadido sobre sus competidores. Lo que a corto plazo será valorado muy favorablemente por los usuarios.

Análisis interno	Fortalezas	Debilidades
	<p>Amplios conocimientos en el desarrollo de aplicaciones web para las Administraciones Públicas.</p> <p>Interés por desarrollar una plataforma que pueda convertirse en una referencia para el resto.</p> <p>No existe ánimo de lucro.</p>	<p>Tiempo y dedicación limitados.</p> <p>No se puede ofrecer el servicio sin colaboración externa.</p>
Análisis externo	Oportunidades	Amenazas
	<p>Se ofrece de forma gratuita y no tiene coste para los usuarios.</p> <p>Si las Administraciones van aceptando su puesta en marcha de forma progresiva, puede servir como la semilla para la creación de un gran ecosistema de servicios a nivel nacional.</p>	<p>Existen empresas que ofrecen servicios más detallados y específicos pero no a coste cero.</p> <p>Hay Administraciones Públicas que están apoyando otras webs similares por lo que podría ser difícil desbancarlas.</p>

Tabla 6: Diagnóstico DAFO.

22.1. Objetivos

- Poner en funcionamiento una plataforma social para ayudar a las personas mayores que necesitan ayuda y no disponen de medios suficientes para contratar los servicios de un profesional.
- Mejorar la calidad de vida de las personas mayores en la medida de lo posible y evitar su exclusión social.

22.2. Estrategias

- En una primera fase negociar con las Administraciones Públicas de pequeñas y medianas localidades la prestación del servicio.
- En una segunda fase negociar con las Administraciones Públicas de las grandes ciudades.
- Contactar de forma preferente con ayuntamientos y consejerías.
- Emplear voluntarios en lugar de profesionales para reducir los costes del servicio.
- Totalmente gratuito para demandantes y voluntarios.

23. Viabilidad

El éxito del proyecto depende de su relación con una Administración Pública que se encargue de las labores diarias de gestión y mantenimiento junto con la salvaguarda de la información. Será la propia administración la que se encargue de publicitar la aplicación en diferentes medios. No hay que olvidar que la mayoría de los demandantes que accederán a la plataforma serán personas mayores, que acuden con frecuencia a centros médicos y hospitalarios, por lo que una publicidad tradicional basada en carteles podría resultar efectiva.

Otro factor importante que influye directamente en la viabilidad del proyecto es el trabajo que realizan los voluntarios. Estos proporcionan un servicio vocacional sin coste alguno para la administración. Por lo tanto, sería conveniente realizar una campaña publicitaria destinada principalmente a Internet y a la televisión para atraer al mayor número de voluntarios posible.

24. Bugs

Cuando se utilizan librerías de terceros que aportan diversa funcionalidad sobre una misma base común, es frecuente que se produzcan efectos colaterales que pueden manifestarse de forma distinta en cada una de las capas que componen la aplicación. Los bugs detectados en la versión final del aplicativo del proyecto son los siguientes:

- Bug-1: En la pantalla de búsqueda de acuerdos tanto para el rol demandante como voluntario, cuando se muestra el mapa a pantalla completa no se visualiza el calendario que permite seleccionar la fecha del acuerdo. Es muy probable que este error se produzca por una superposición de capas o valores incorrectos del atributo z-order en las capas en el interior de alguno de los archivos de estilo.
- Bug-2: En la pantalla de búsqueda de acuerdos cuando el usuario pulsa sobre la caja de texto para seleccionar la fecha, se muestra el calendario de forma no modal y si en ese instante se desplaza la rueda del ratón, el nivel del zoom correspondiente al mapa cambia y el calendario no se posiciona en la ubicación que le correspondería dentro de la ventana de información.
- Bug-3: En las pantallas en las que el voluntario o el demandante puede introducir su fecha de nacimiento no se queda seleccionado ni el mes ni el año elegido, aunque el cambio de selección de mes y año funciona correctamente. Al pulsar en uno de los días del calendario la fecha se almacena correctamente.

25. Proyección a futuro

Inexorablemente la aplicación necesita estar vinculada a una Administración Pública que se encargue de la custodia de la información y, proporcione el nivel de seguridad y confianza que tanto los demandantes como los voluntarios requieren. Normalmente, serán los ayuntamientos de cada localidad los encargados de la gestión y el mantenimiento de la aplicación.

En un futuro no muy lejano podrían ser decenas o cientos las instalaciones de RESOAM desplegadas en la red. Sin embargo, cada una de ellas estaría aislada de las demás. Una mejora que incrementaría de manera notable la funcionalidad de la aplicación, consistiría en implementar un sistema de comunicación entre las diversas instalaciones que permitiera el intercambio de información entre ellas. De esta forma se favorece la movilidad y se permite que un voluntario de una ciudad se desplace a otra ciudad distinta para prestar sus servicios.

Al ser una aplicación que facilita el encuentro físico entre personas resulta de vital importancia aumentar la seguridad. Por este motivo, sería recomendable sustituir el formulario de autenticación por la gestión de acceso basada en certificados digitales y DNI electrónico, así como implementar un sistema de auditoría que permita almacenar datos en referencia a quien y en qué momento ha consultado una determinada información.

26. Conclusiones

La realización de este proyecto me ha permitido poner en práctica la mayor parte de las competencias adquiridas durante los últimos años que he cursado el grado Multimedia, especialmente las relacionadas con la programación web y la arquitectura de la información. Sin olvidar las relativas al aspecto visual como el diseño gráfico, la composición digital y el diseño de interfaces. El producto final se ha desarrollado con la premisa de disponer de una base sólida que permitiera añadir en el futuro nuevas funcionalidades de forma rápida y sencilla.

La apuesta de configurar un servidor -físico- doméstico ligero que pudiera correr software de servidores web y servidores de aplicaciones, con solo 1Gb. de memoria, ha sido un verdadero éxito. Es obvio que nunca llegará a convertirse en un servidor de producción y, además, en las pruebas realizadas la concurrencia de cientos de peticiones no ha sido testada. Sin embargo, como servidor de pruebas es una solución muy económica y realmente útil para cualquier tipo de desarrollo, las opciones que ofrece son infinitas.

Funcionalmente la aplicación no tiene un nivel de complejidad elevado, pero el hecho de incrementar el número de perfiles dificultó el desarrollo y tuve que realizar un mayor número de páginas. Igualmente, el no disponer de un servidor SMTP que pudiera enviar correos fuera de la red local me obligó a que todo el sistema de mensajería estuviera integrado en la aplicación. Lo que a su vez supuso habilitar el acceso a la aplicación para demandantes y voluntarios, aunque el gestor no hubiera validado su solicitud.

La integración en la aplicación de los archivos en lenguaje Javascript procedentes de terceros ha resultado satisfactoria, aunque no por ello, libre de efectos colaterales. Algunos de los plugin que tenía pensado utilizar para dotar el mapa de nuevas funcionalidades colisionaba con otros y decidí abandonarlos. Además, con el compendio de archivos utilizados no es factible que la interfaz web obtuviese las calificaciones AA ni CSS. Estas solo se llegaron a cumplir en los primeros momentos del desarrollo, sin embargo también tuve que abandonarlas en pro de un mejor aspecto visual.

El componente que más inconvenientes me ha causado ha sido Hibernate, desconozco realmente los motivos pero he tenido que cambiar de versión hasta en tres ocasiones. Cabe la posibilidad que los problemas tuvieran origen en la configuración del equipo de desarrollo, puesto que este tiene instalado una gran cantidad de software y más de una versión de Eclipse y varios JDK y JRE.

Anexo 1. Entregables del proyecto

La aplicación debe ejecutarse sobre un servidor web con capacidad para manejar servlets. Entre los más destacados están Tomcat, Weblogic y Jetty; aunque también existe la opción de adicionar un servidor web como Apache web server o Nginx que actúe como proxy inverso y se encargue de servir el contenido estático y redirigir las peticiones de ejecución de contenido dinámico al contenedor de servlets.

Cuando se empaqueta el proyecto con la herramienta Maven las librerías `javax.servlet-api`, `jstl`, `log4j`, `junit`, `lombok`, `gson` no se incluyen en el archivo con extensión `.war` debido a que están definidas con ámbito `provided/test` en el archivo `pom`. Por lo tanto, el contenedor de servlets es el responsable de proporcionarlas.

- La versión "reverse proxy" se utiliza en el entorno de producción. Incluye un archivo `.tar.gz` con el contenido estático de la aplicación, archivos javascript, páginas de estilo e imágenes. La herramienta Maven excluye automáticamente el contenido indicado del archivo `.war`.
 - `resoam.war`²⁵
 - `resoam.tar.gz`²⁶
 - `creacion_bbdd.sql`
 - `carga_datos_inicial.sql`
- La versión "standalone" se utiliza para el entorno local de desarrollo. Tanto el contenido dinámico como estático se almacena en el archivo `.war`.
 - `resoam.war`
 - `creacion_bbdd.sql`
 - `carga_datos_inicial.sql`

²⁵ Esta versión del archivo no incluye contenido estático.

²⁶ Este archivo contiene todo el contenido estático que se debe desplegar en el servidor web.

Anexo 2. Código fuente

Todas las peticiones que realiza el cliente al servidor se ejecutan invocando a los actions proporcionados por el framework struts 2. En el archivo struts.xml se configura el tipo de respuesta utilizada, que podrá ser tiles o json.

```

...
<!-- Internacionalización -->
<constant name="struts.custom.i18n.resources" value="{struts.languageFolder__}/{struts.languageFilePrefix__}"/>

<constant name="struts.locale" value="{struts.locale__}" />

<!-- Package por defecto -->
<package name="default" extends="struts-default, json-default">
  <result-types>
    <result-type name="tiles" class="org.apache.struts2.views.tiles.TilesResult"/>
  </result-types>

  <!-- Acciones JSON de recuperación de datos-->
  <action name="RecuperarDatosGestor" class="com.ajmahillo.resoam.action.JSONRecuperarAction"
  method="recuperarDatosGestor">
    <result type="json"/>
  </action>
...

```

La función javascript getJson ubicada en el archivo resoam-core.js es la encargada de realizar las peticiones AJAX, a excepción del cambio de idioma y del inicio y cierre de la sesión de usuario.

```

...
function getJson(actionName, data, okFunction, errorFunction){
  if(!isNullOrBlank(actionName) && !isNullOrBlank(data) && !isNullOrBlank(okFunction)){
    $.ajax({
      url: actionName+ ".action",
      data: {"jsonData":data},
      method: AJAX_POST_METHOD,
      success: function(data){
        var uncoded=$.base64('decode',data);
        var invoke = okFunction + "(" + JSON.stringify(uncoded,null) + ")";
        jQuery.globalEval(invoke);
      },
      error: function(jqXHR, textStatus){
        if(errorFunction!=null){
          var invoke = errorFunction + "(" + textStatus + ")";
          jQuery.globalEval(invoke);
        }else{
          alert(textStatus);
        }
      }
    });
  }
}
...

```

Las clases action java encargadas de gestionar las peticiones en formato JSON como JSONActualizarAction, JSONEliminarAction, JSONInsertarAction y JSONRecuperarAction. Todas heredan de la clase JSONAction que define los objetos necesarios para llevar a cabo las tareas a realizar. En concreto, jsonData almacena los datos recibidos de la petición AJAX, jsonResponse permite configurar la respuesta que se envía al cliente y businessFacade representa el punto de entrada a las clases de negocio de la aplicación. La función requestFormToJson parsea los datos recibidos realizando una llamada a la función parsearDeRequestAJson de la clase estática ManipuladorCadenas.

```
package com.ajmahillo.resoam.action;

import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;
import org.apache.struts2.interceptor.ServletRequestAware;
import org.apache.struts2.interceptor.ServletResponseAware;

import com.ajmahillo.resoam.business.facade.BusinessFacade;
import com.ajmahillo.resoam.common.JSONResponse;
import com.ajmahillo.resoam.utils.ManipuladorCadenas;
import com.google.gson.Gson;
import lombok.Data;
import lombok.EqualsAndHashCode;

@EqualsAndHashCode(callSuper=true)
@Data public class JSONAction extends AutorizacionAction implements ServletRequestAware, ServletResponseAware{

    private static final long serialVersionUID = 1L;
    private static Logger log = Logger.getLogger(JSONAction.class);

    protected HttpServletRequest request;
    protected HttpServletResponse response;
    protected ServletOutputStream outputStream;
    protected Gson gson;
    protected String jsonData;
    protected JSONResponse jsonResponse;
    protected BusinessFacade businessFacade;

    protected String requestFormToJson(){
        return (ManipuladorCadenas.parsearDeRequestAJson(jsonData));
    }

    public void setServletRequest(HttpServletRequest hsr) {
        this.request = hsr;
    }

    public void setServletResponse(HttpServletResponse hsr) {
        this.response = hsr;
    }
}
```

Los datos de los formularios enviados en la petición están codificados en base64. En realidad, son codificados dos veces; en la primera fase solo se codifican solo los valores de los elementos y en la segunda fase se codifica la cadena completa que se va a enviar.

```

...
function getFormData(formName){
    var data="";
    $.base64.utf8encode = true;
    $("#"+formName).find(':input').each(function() {
        if(this.type=="checkbox"){
            data+=this.id + SEP_1 + $.base64('encode', this.checked.toString()) + SEP_2;
        }else if(this.type=="radio"){
            data+=this.id + SEP_1 + $.base64('encode', this.selected.toString()) + SEP_2;
        }else if(this.type=="submit" || this.type=="reset"){
            //No incluirlos en la cadena
        }else{
            data+=this.id + SEP_1 + $.base64('encode', this.value) + SEP_2;
        }
    });

    if(data.length>0){
        data=data.substr(0,data.length-1);
        data=$.base64('encode',data);
    }
    return data;
}
...

```

La función parsearDeRequestAJson se encarga de descodificar la cadena y parsea los datos para evitar que pudieran incluir código malicioso; por ejemplo, mediante la inyección de código.

```

public static String parsearDeRequestAJson(String cad){
    String aux=null;
    String primeraFase=null;

    if(cad!=null){
        primeraFase=base64ATexto(cad);
        aux="";
        String[] arrData1=primeraFase.split(SEP_2);

        for(String s: arrData1){
            String[] arrData2=s.split(SEP_1);
            aux+="\" + arrData2[0] + "\":\"";
            if(NULL_STRING.equalsIgnoreCase(arrData2[1])){
                aux+="\"\",\"";
            }else{
                aux+="\" + limpiaParametroRequest(base64ATexto(arrData2[1])) + "\",\"";
            }
        }
        aux=aux.substring(0,aux.length()-1);
        aux="{\" + aux + "\"}";
    }
    return aux;
}

```

La lógica encargada de resolver las peticiones es muy similar en todos los actions. Se trata de un proceso que engloba desde la recuperación de los datos enviados, pasando por la creación de los objetos de negocio hasta realizar el envío de la respuesta al cliente. Se utilizan controladores de excepciones para asegurar que la aplicación informe debidamente al usuario si se produce algún error no controlado.

La clase BusinessFacade es una fachada para todas las clases de la librería de negocio de la aplicación. De esta manera se puede simplificar la lógica de la capa de la vista, ya que esta se limitará a crear una serie de objetos simples que se pasan como parámetro a la función de la fachada.

```

...
public String altaSolicitudVoluntario(){
    try {
        lanzaExcepcionSiMetodoGet();
        jsonResponse=new JSONResponse();
        businessFacade=new BusinessFacade(this.identificacionUsuario);
        gson=new Gson();
        FrmSolicitudVoluntario frm=gson.fromJson(requestFormToJJson(), FrmSolicitudVoluntario.class);

        if(frm!=null){
            SolicitudVoluntario sol=new SolicitudVoluntario();
            ...
            businessFacade.altaSolicitudVoluntario(sol);
            jsonResponse.setResult(JSONResponseResultEnum.OK);
        }else{
            jsonResponse.getErrors().add(ACTION_ERROR_RECUPERAR_FORMULARIO);
        }
    }catch(Exception e) {
        jsonResponse.getErrors().add(e.getMessage());
    }

    //Escribir la respuesta
    try {
        outputStream = response.getOutputStream();
        outputStream.print(jsonResponse.toString());
    } catch (IOException e) {
        log.error(e.getMessage());
    }
    return null;
}
...

```

Ejemplo de llamada a una función de la fachada de negocio para el proceso de autenticación.

```

...
identificador=ManipuladorCadenas.limpiaParametroRequest(identificador);
contrasena=ManipuladorCadenas.limpiaParametroRequest(contrasena);

try{
    BusinessFacade businessFacade=new BusinessFacade();
    IdentificacionUsuario identificacion=businessFacade.autenticarUsuario(identificador, contrasena);
}
...

```

La clase JSONResponse contiene entre otros atributos una serie de arrays de cadenas que permiten almacenar los errores y los datos que se deben retornar al cliente.

```

...
import com.ajmahillo.resoam.common.Enumeraciones.JSONResponseResultEnum;
import com.ajmahillo.resoam.utils.ManipuladorCadenas;

import lombok.Data;

@Data public class JSONResponse {

    private static Logger log = Logger.getLogger(JSONResponse.class);
    private static final String CODIGO_ERROR_POR_DEFECTO="-1";

    private JSONResponseResultEnum result;
    private String token;
    private ArrayList<String> errors = new ArrayList<String>();

    private String nombre;
    private String tipo;
    private ArrayList<String> registros = new ArrayList<String>();

    ...

```

Ejemplo de respuesta JSON estandarizada para todos los actions.

```

{
  "respuesta": [{
    "resultado": "ok",
    "token": "178327483",
    "errores": [{
      "id": "0",
      "descripción": "Error 1"
    },
    {
      "id": "1",
      "descripción": "Error 2"
    }
  ]
}],
  "datos": [{
    "nombre": "el nombre",
    "tipo": "el tipo",
    "registros": [{
      "id": "1",
      "campo1": "valor1",
      "campo2": "valor2"
    },
    {
      "id": "2",
      "campo1": "valor1",
      "campo2": "valor2"
    }
  ]
}]
}

```

Anexo 3. Librerías/Código externo

- Java
 - Apache Struts 2.5.10.1
 - Apache Tiles 2.5.10.1
 - Apache Struts json plugin 2.5.10.1
 - Apache log4j 1.2.17
 - Google gson 2.2.2
 - Hibernate-core 4.0.1.Final
 - Hibernate-validator 4.2.0.Final
 - Hibernate-commons-annotations 4.0.1.Final
 - Hibernate-entitymanager 4.1.8.Final
 - Lombok 1.16.14
 - junit 4.11
 - jboss-logging 3.1.0.CR2
 - slf4j-log4j12 1.6.4
 - mysql-connector-java 5.1.41

- Javascript
 - Bootstrap 3.3.7
 - jQuery 1.12.4
 - jQuery UI 1.12.1
 - jqBootstrapValidation 1.3.6
 - jQuery-confirm 3.2.0
 - jQuery.base64 0.1
 - jQuery MsgBox 0.8.1
 - Respond.js 1.4.2
 - HTML5 Shiv 3.7.3
 - IE10 viewport hack
 - Leaflet 1.0.3
 - Leaflet.FullScreen
 - Leaflet.awesome-markers plugin v 2.0

- CSS
 - Bootstrap 3.3.7
 - IE10 viewport hack
 - jQuery-confirm 3.2.0
 - jQuery UI 1.12.1
 - Leaflet 1.0.3
 - Leaflet.awesome-markers.css
 - Control.FullScreen.css

- Fonts
 - glyphicons-halflings-regular.eot
 - glyphicons-halflings-regular.svg
 - glyphicons-halflings-regular.ttf
 - glyphicons-halflings-regular.woff
 - glyphicons-halflings-regular.woff2

Anexo 4. Glosario

- Apache tomEE/Tomcat: es un contenedor de servlets muy ágil y sencillo de configurar e integrar en entornos de desarrollo como Eclipse o Netbeans.
- Apache web server: es un servidor web ligero y altamente configurable. Se hizo muy popular desde los inicios de Internet hasta que llegó a convertirse en el número uno indiscutible de la web.
- Array: en programación también se conoce como matriz y, representa un conjunto de valores a los que se puede acceder de forma secuencial.
- Contenedor de servlets: es un software Java que puede recibir peticiones de una página web y es capaz de reenviarlas a un servlet específico.
- Contenido dinámico: es aquel contenido variable que necesita ser preprocesado en el servidor y, por lo tanto, está personalizado para cada una de las peticiones que realiza el usuario.
- Contenido estático: es aquel contenido que no sufre variaciones y el servidor lo devuelve de manera inmediata. Un claro exponente de este tipo de contenidos sería un archivo de imagen o una página de estilos css.
- GPU: es la unidad de procesamiento gráfico de un equipo.
- HTTPS: es el acrónimo de Hiper Text Transfer Protocol Secure, es uno de los principales protocolos de Internet. Se diferencia del HTTP porque la comunicación entre el cliente y el servidor está cifrada, de esta manera se consigue aumentar el nivel de seguridad y confidencialidad de los datos que envía el usuario.
- IDE: es un entorno de desarrollo integrado que permite a los programadores desarrollar el código fuente del proyecto.
- Jetty: es un servidor web ligero basado en Java que además puede actuar como contenedor de servlets.
- JDK: es el acrónimo de Java Development Kit, un conjunto de herramientas de desarrollo y librerías para aplicaciones Java
- JRE: es el acrónimo de Java Runtime Environment, está formado por el conjunto mínimo de librerías necesarias para ejecutar aplicaciones Java en un ordenador.
- JVM: es el acrónimo de Java Virtual Machine, está formado por un conjunto de archivos denominados máquina virtual que permiten la ejecución de código Java en plataformas como Linux/Unix, Windows, Solaris, Android, etc.
- Maven: es una herramienta de software para la plataforma Java que permite la creación y el empaquetado de los productos finales en formatos jar, war y ear a partir del código

fuelle del proyecto.

- Nginx: es un servidor web ligero de alto rendimiento, que ha experimentado un crecimiento significativo en Internet durante los últimos años.
- Pom: es el acrónimo de Project Object Model, es un archivo en formato XML que utiliza Maven para confeccionar la estructura del proyecto así como las dependencias externas y los repositorios desde donde descargará los archivos fuente necesarios para la ejecución del proyecto.
- Provided: se suele utilizar en los archivos POM para indicar al compilador que no incluya la librería en el producto final cuando se realice el ensamblado de los paquetes puesto que esa librería la proporciona el servidor en el que se va a instalar la aplicación.
- Proxy inverso (Reverse proxy): es un servidor que se despliega en la DMZ de una empresa y actúa de intermediario entre los usuarios que acceden desde Internet y los servidores de la organización que se encuentran dentro de su red Interna. Un proxy inverso añade seguridad, ya que establece un único lugar en el que verificar la identidad del usuario y determinar los recursos a los que puede acceder.
- Servidor web: es un software que permite la comunicación bidireccional síncrona como asíncrona entre el cliente y el servidor.
- Servlet: es una clase en lenguaje Java que permite ampliar las funcionalidades de un servidor web. Es la base que permite el intercambio dinámico de información entre el cliente y el servidor.
- SGBD: sistema gestor de base de datos, es un software que facilita al usuario experto la creación y gestión de bases de datos.
- SSH: es un intérprete de órdenes seguro que permite manejar servidores de forma remota bajo entorno Linux/Unix y Windows.
- Standalone: suele utilizarse para designar sistemas o elementos que componen un sistema y que pueden funcionar de forma autónoma.
- Test: conjunto de pruebas que se realizan para asegurar que el software desarrollado cumple las especificaciones requeridas.
- User-friendly: interfaz amigable, puede resumirse como las directivas aplicadas durante todo el ciclo de vida del proyecto para conseguir una interfaz clara e intuitiva que facilite al usuario la interacción con el sistema.
- Weblogic: es un servidor muy robusto de aplicaciones Java y contenedor EJB que cumplen con el estándar J2EE.
- War: formato de archivo para el despliegue de aplicaciones en servidores Java 2EE.

Anexo 5. Bibliografía

Rumbaugh, James, Ivar Jacobson, and Grady Booch. *Lenguaje Unificado De Modelado: Manual De Referencia*. Madrid: Addison Wesley, 2002. Print.

Horine, Gregory M. *Gestión De Proyectos*. Madrid: Anaya Multimedia, 2010. Print.

Pressman, Roger S, Martín R. Ojeda, and Aguilar L. Joyanes. *Ingeniería Del Software: Un Enfoque Práctico*. Madrid: McGraw-Hill Interamericana de España, 1999. Print.

Northcutt, Stephen, and Judy Novak. *Detección De Intrusos: Guía Avanzada*. Madrid [etc.]: Prentice Hall, 2001. Print.

Sturm, Jake, Blázquez A. Moreno, and de F. M. López. *Desarrollo De Soluciones Xml*. Madrid: McGraw Hill, 2001. Print.

McClelland, Deke. *La Biblia De Photoshop 5 Para Windows*. Madrid: Anaya Interactiva, 1998. Print.

Diseño De Interfaz De Usuarios Para Aplicaciones Windows. Madrid: McGraw-Hill, 2000. Print.

"1- Introduction." *Install JBoss Tools into Eclipse*. N.p., 2 2017. Web. 5 May 2017. <<http://o7planning.org/en/10119/install-jboss-tools-into-eclipse>>.

"6.1. Iconos (glyphicons)." *6.1. Iconos (glyphicons) (Bootstrap 3, El Manual Oficial)*. N.p., 1 2017. Web. 5 May 2017. <http://librosweb.es/libro/bootstrap_3/capitulo_6/iconos_glyphicons.html>.

"7.3. Métodos Ajax De JQuery." *7.3. Métodos Ajax De JQuery (Fundamentos De JQuery)*. N.p., 1 Mar. 2017. Web. 5 May 2017. <http://librosweb.es/libro/fundamentos_jquery/capitulo_7/metodos_ajax_de_jquery.html>.

"8 Funciones Para La Validación De Formularios Con Expresiones Regulares." *8 Funciones Para La Validación De Formularios Con Expresiones Regulares - MichuBlog*. N.p., 4 Feb. 2017. Web. 5 May 2017. <<http://www.michublog.com/informatica/8-funciones-para-la-validacion-de-formularios-con-expresiones-regulares>>.

"Ajaxload." *Ajaxload - Ajax Loading Gif Generator*. N.p., 2 2017. Web. 5 May 2017. <<http://www.ajaxload.info/>>.

Arenzana, David. "Principios De Usabilidad Web De Jakob Nielsen: Diseño UX." *Principios De Usabilidad Web De Jakob Nielsen: Diseño UX*. N.p., 3 Mar. 2017. Web. 5 May 2017. <<https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>>.

"Base64 Decode and Encode Plugin - Base64.js." *Base64 Decode and Encode Plugin - Base64.js | Free JQuery Plugins*. N.p., 5 Apr. 2017. Web. 5 May 2017. <<http://www.jqueryscript.net/text/Base64-Decode-Encode-Plugin-base64-js.html>>.

- "Bootply." *Bootstrap 3 Responsive Form JrNvaahcOh*. N.p., 1 2017. Web. 5 May 2017. <<http://www.bootply.com/jrNvaahcOh>>.
- Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework..* N.p., 5 July 2017. Web. 5 May 2017. <<http://getbootstrap.com/>>.
- "Bootstrap-datepicker." *Bootstrap-datepicker — Bootstrap-datepicker Documentation*. N.p., 1 Feb. 2017. Web. 5 May 2017. <<https://bootstrap-datepicker.readthedocs.io/en/latest/>>.
- "Composite/jQuery.MsgBox." *Home · Composite/jQuery.MsgBox Wiki · GitHub*. N.p., 4 2017. Web. 5 May 2017. <[https://github.com/composite/jquery.MsgBox/wiki#confirmmessagecallback--msgbox-confirm](https://github.com/composite/jquery.msgbox/wiki#confirmmessagecallback--msgbox-confirm)>.
- Cruz, Juan Carlos Reyes. "Manual Struts2 Espanol." *Manual Struts2 Espanol | Juan Carlos Reyes Cruz - Academia.edu*. N.p., 1 Feb. 2017. Web. 5 May 2017. <http://www.academia.edu/11560148/Manual_Struts2_Espanol>.
- "David González." *Ficheros .war Sin Las Librerías Del Proyecto Con Maven Y Tomcat – Cuaderno De Software*. N.p., 3 2017. Web. 5 May 2017. <<http://www.programania.net/disenio-de-software/ficheros-war-sin-las-librerias-del-proyecto-con-maven-y-tomcat/>>.
- "El Voluntariado De UDP." *El Voluntariado De UDP*. N.p., 14 Jan. 2015. Web. 5 May 2017. <<https://www.mayoresudp.org/el-voluntariado-de-udp/>>.
- "Examples." *Examples - Bootstrap-select*. N.p., 1 Dec. 2017. Web. 5 May 2017. <<https://silviomoreto.github.io/bootstrap-select/examples/>>.
- "Expresiones Regulares Útiles." *Expresiones Regulares Útiles | DevTroce.com*. N.p., 3 2017. Web. 5 May 2017. <<http://devtroce.com/2012/12/10/expresiones-regulares-utiles/>>.
- "Instituto De Mayores Y Servicios Sociales." *Imserso. Instituto De Mayores Y Servicios Sociales :: Personas Mayores*. N.p., 3 2017. Web. 5 May 2017. <http://www.imserso.es/imserso_01/per_mayores/index.htm>.
- "Intervención Social." *Intervención Social - Mayores - Cruz Roja*. N.p., 4 2017. Web. 5 May 2017. <<http://www.cruzrojajuventud.org/principal/web/intervencion-social/mayores>>.
- "ISO 639-1." *ISO 639-1 - Wikipedia, La Enciclopedia Libre*. N.p., n.d. Web. 5 May 2017. <https://es.wikipedia.org/wiki/ISO_639-1>.
- "Java SE 8: Creating a Web App with Bootstrap and Tomcat Embedded." *Java SE 8: Creating a Web App with Bootstrap and Tomcat Embedded*. Oracle.net, 3 Mar. 2017. Web. 5 May 2017. <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/basic_app_embedded_to_mcat/basic_app-tomcat-embedded.html>.

- "JqBootstrapValidation." *JqBootstrapValidation by ReactiveRaven*. N.p., 4 2017. Web. 5 May 2017. <<http://reactiveraven.github.io/jqBootstrapValidation/>>.
- "jQuery Manual." *Datepicker En Formato Español Y Restricción De Fechas JQuery UI | JQuery Manual*. N.p., 2 Dec. 2017. Web. 5 May 2017. <<http://jquery-manual.blogspot.com.es/2013/02/datepicker-en-formato-espanol-y.html>>.
- "jQuery Manual." *Jquery Ui - Cambiar Propiedades CSS Del Campo De Fecha - Datepicker | JQuery Manual*. N.p., 2 2017. Web. 5 May 2017. <http://jquery-manual.blogspot.com.es/2013/10/jquery-ui-cambiar-propiedades-css-del_6831.html>.
- Jquery-confirm.js | The Multipurpose Alert & Confirm*. N.p., 2 2017. Web. 5 May 2017. <<https://craftpip.github.io/jquery-confirm/>>.
- "JSONLint - The JSON Validator." *JSONLint - The JSON Validator*. N.p., 1 2017. Web. 5 May 2017. <<http://jsonlint.com/>>.
- "Jugando Con JSON En Java Y La Librería Gson." *Jugando Con JSON En Java Y La Librería Gson | Adictosaltrabajo*. N.p., 1 2017. Web. 5 May 2017. <<https://www.adictosaltrabajo.com/tutoriales/gson-java-json/>>.
- "Las 5 Etapas En Los "Sprints" De Un Desarrollo Scrum." *Las 5 Etapas En Los "Sprints" De Un Desarrollo Scrum | OBS Business School*. N.p., 5 2017. Web. 5 May 2017. <<http://www.obs-edu.com/es/blog-investigacion/project-management/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>>.
- "List of XML and HTML Character Entity References." *List of XML and HTML Character Entity References - Wikipedia*. N.p., 4 2017. Web. 5 May 2017. <https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references>.
- "Métodos Y Procesos En la Metodología Ágil." *Beneficios De Aplicar Metodologías Ágiles En El Desarrollo De Software*. N.p., 2 Oct. 2017. Web. 5 May 2017. <<http://www.i2btech.com/blog-i2b/tech-deployment/5-beneficios-de-aplicar-metodologias-agiles-en-el-desarrollo-de-software/>>.
- Oficial, Zhonne. "Aplicaciones Web." *Aplicaciones Web: Select Provincias España*. N.p., 5 May 2017. Web. 5 May 2017. <<http://phpciones.blogspot.com.es/2013/07/select-provincias-espana.html>>.
- "Pantalla De Espera En Peticiones AJAX Con JQuery." *Pantalla De Espera En Peticiones AJAX Con JQuery*. N.p., 29 Apr. 2011. Web. 5 May 2017. <<http://www.linuxhispano.net/2011/04/29/pantalla-de-espera-en-peticiones-ajax-con-jquery/>>.

"Reviblog." *JQuery UI Datepicker, Como Poner El Calendario En Español, Euskera O Cualquier Otro Idioma.* | *Reviblog.* N.p., 7 Jan. 2014. Web. 5 May 2017.
<<https://reviblog.net/2014/01/07/jquery-ui-datepicker-poner-el-calendario-en-espanol-euskera-o-cualquier-otro-idioma/>>.

"Scrum (desarrollo De Software)." *Scrum (desarrollo De Software) - Wikipedia, La Enciclopedia Libre.* N.p., 5 Jan. 2017. Web. 5 May 2017.
<[https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))>.

"Struts 2 - Guía Rápida." *Struts 2 - Guía Rápida.* N.p., 5 Feb. 2017. Web. 5 May 2017.
<http://www.w3ii.com/es/struts_2/struts_quick_guide.html>.

"Tip Struts 2 #11: I18n (internacionalización)." *Tip Struts 2 #11: I18n (internacionalización) | Danielme.com.* N.p., 8 Jan. 2014. Web. 5 May 2017. <<https://danielme.com/tip-struts-2-11-i18n-internacionalizacion/>>.

"La España Interior Se Queda Vieja Y Sin Habitantes (mientras Las Capitales Engordan)." *La España Interior Se Queda Vieja Y Sin Habitantes (mientras Las Capitales Engordan). Noticias De España.* N.p., n.d. Web. 3 June 2017.
<http://www.elconfidencial.com/espana/2017-01-06/espana-pierde-habitantes-despoblacion_1310809/>.

"Pirámides De Población Del Mundo Desde 1950 a 2100." *Población: España 2017 - PopulationPyramid.net.* N.p., n.d. Web. 3 June 2017.
<<http://www.populationpyramid.net/es/espa%C3%B1a/2017/>>.

CSIC, Envejecimiento En Red. *Pirámide De Población De España. Gráficos Dinámicos. Estadísticas. Envejecimiento En Red.* CSIC. Envejecimiento En Red. CSIC. C/ Albasanz, 26-28, 28037 Madrid, España. envejecimientoenred@cchs.csic.es, n.d. Web. 3 June 2017.
<<http://envejecimiento.csic.es/estadisticas/graficos-dinamicos/graficos/piramide-espanya.html>>.

Anexo 6. Vita

Hola, mi nombre es Antonio Jiménez Mahillo y aunque mis raíces son extremeñas me trajeron al mundo un cálido domingo 30 de mayo de 1976 en el hospital La Paz de Madrid. Como anécdota ese día se celebraba en Madrid el desfile de las Fuerzas Armadas y, tal vez por ello, siempre he sentido mucha fascinación por los aviones.

Mis padres emigraron desde un pequeño pueblo de la provincia de Cáceres, por aquel entonces éramos una familia numerosa y, como es habitual, los comienzos en la nueva ciudad fueron bastante difíciles. La casa no era muy grande y teníamos que comer por turnos puesto que no había sitio físico para sentarnos todos juntos. Unos años después mi padre tuvo que jubilarse por motivos de salud y mis hermanas mayores comenzaron a formar sus propias familias. Al final pasamos de ser nueve en casa a ser tan solo tres, así que mucha parte de la infancia la viví como hijo único. En alguna ocasión mi madre me ha contado que de pequeño era callado y tímido, y que siempre estaba centrado en mis cosas.

La etapa escolar fue entretenida, hice buenos amigos y siempre llegaba temprano para jugar antes de entrar a clase. Cuando terminé la EGB fui al instituto, lamentablemente mi elección no fue acertada, puesto que lo habían abierto el año anterior y la mayoría de los días los profesores no se presentaban; así que terminé por dedicar ese tiempo a jugar al ajedrez con otros compañeros. En los últimos años de instituto todavía era obligatorio realizar el servicio militar así que decidí alistarme a los boinas verdes. Fue una experiencia muy dura pero me aportó mucha satisfacción, incluso pensé en alistarme pero lo de estar todo el día a las órdenes de otros no me terminaba de convencer.

Cuando terminé el instituto estuve realizando algunos trabajos para conseguir algo de dinero y estudiar informática por mi cuenta. Realicé bastantes cursos que me abrieron las puertas de muchas empresas y rápidamente encontré trabajo. Comencé mi carrera como programador hasta convertirme en jefe de proyecto, pero siempre echaba en falta no haber realizado la carrera en su momento. Un día mi mujer vio en Internet el grado Multimedia de la UOC y me propuso retomar los estudios. Estos años que he estado cursando el grado Multimedia han sido realmente duros, ya que además de trabajar y estudiar he formado mi propia familia y en muchas ocasiones no he podido dedicarles todo el tiempo que me hubiera gustado.