

## Trabajo Fin de Grado

Ampliación de una aplicación móvil, y su panel de administración web, para consultar y gestionar datos sobre los edificios de la Universidad de Zaragoza

Extension of a mobile application, and its web administration panel, for querying and managing data about the buildings of the Universidad Zaragoza

Autor

**Alejandro Royo Amondarain**

Director

**Rubén Béjar Hernández**

Escuela de Ingeniería y Arquitectura  
2017





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

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

D./D<sup>a</sup>. Alejandro Royo Amondarain,

con nº de DNI 17768640J en aplicación de lo dispuesto en el art.

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

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado \_\_\_\_\_, (Título del Trabajo)  
Ampliación de una aplicación móvil, y su panel de administración web, para  
consultar y gestionar datos sobre los edificios de la Universidad de Zaragoza  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

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

Zaragoza, 22 de junio de 2017

Fdo: Alejandro Royo Amondarain



# Ampliación de una aplicación móvil, y su panel de administración web, para consultar y gestionar datos sobre los edificios de la Universidad de Zaragoza

## Resumen

El objetivo de este proyecto es mejorar una aplicación de ayuda y guía al usuario dentro de la Universidad de Zaragoza para facilitar la gestión de los espacios de la universidad y sus modificaciones al administrador y así potenciar el uso de la aplicación.

Para facilitar la tarea del administrador respecto a la gestión de los espacios se ha definido en colaboración con la Unidad Técnica de Construcciones (UTC) un sistema de comunicación de notificaciones. Este sistema permite crear notificaciones mediante una aplicación móvil que son enviadas a un administrador. Éste las revisa y toma una decisión desde el panel de administrador, de manera que usuario y administrador pueden responderse mutuamente hasta que se considere por parte del administrador que no es necesaria información adicional.

Con la motivación de acercar el sistema a un estado en el que pueda ser usado por las personas que tienen acceso a la universidad, se ha añadido seguridad a la aplicación, de manera que el sistema soporta múltiples grupos de usuarios en función de los permisos que poseen.

La arquitectura del sistema es una arquitectura cliente-servidor de tres niveles. El cliente móvil se ha desarrollado con el framework Ionic, AngularJS, HTML5, CSS y Javascript, con el objetivo de poder generar una aplicación Android y una aplicación web fácilmente. El cliente web administrador se ha programado principalmente con *jQuery*. El servidor está desarrollado en el framework *Spring* para la plataforma *Java* y se despliega en *Tomcat 7* en un servidor propiedad de la universidad. La base de datos, situada en la misma máquina que el servidor, utiliza la tecnología *PostgreSQL*.

El proyecto ha sido realizado en dos iteraciones, con una fase inicial previa donde se han planteado las líneas de trabajo con el cliente (UTC) y se ha estudiado la tecnología sobre la que estaba desarrollada la aplicación original. Posteriormente, en la fase final se ha evaluado el trabajo realizado junto al cliente y se ha completado la documentación del proyecto así como este documento.

En el contexto de colaboración con un cliente real, se han priorizado las necesidades de la UTC respecto a otras mejoras planteadas inicialmente. El cliente ha valorado positivamente el trabajo realizado y coincide en las posibles líneas de trabajo futuro, como la generación y el pintado de rutas sobre los mapas de la aplicación o la integración de las credenciales de la universidad en el sistema para mantener un control de qué usuarios realizan determinadas acciones.



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto del proyecto . . . . .	1
1.1.1. Colaboración con UTC . . . . .	1
1.1.2. Contexto tecnológico . . . . .	1
1.2. Objetivo . . . . .	1
1.3. Motivación . . . . .	2
1.4. Estructura del documento . . . . .	2
<b>2. Análisis del problema</b>	<b>4</b>
2.1. Descripción del problema . . . . .	4
2.1.1. Sistema de comunicación de notificaciones . . . . .	4
2.2. Requisitos . . . . .	8
2.2.1. Requisitos funcionales . . . . .	8
2.2.2. Requisitos no funcionales . . . . .	9
<b>3. Diseño de la solución</b>	<b>10</b>
3.1. Arquitectura del sistema . . . . .	10
3.1.1. Cliente web administrador . . . . .	11
3.1.2. Cliente móvil . . . . .	11
3.1.3. Servidor . . . . .	13
3.2. Protocolo de seguridad . . . . .	16
3.3. Modelo de datos . . . . .	18
<b>4. Implementación de la solución</b>	<b>20</b>
4.1. Cliente admin . . . . .	20
4.1.1. Problemas encontrados . . . . .	20
4.2. Cliente móvil . . . . .	20
4.3. Servidor . . . . .	21
4.3.1. Problemas encontrados . . . . .	21
4.4. Base de datos . . . . .	21
4.5. Pruebas . . . . .	21
4.6. Despliegue . . . . .	22
4.7. Herramientas de desarrollo . . . . .	22
<b>5. Gestión del proyecto</b>	<b>24</b>
5.1. Planificación . . . . .	24
5.2. Comunicación con el cliente . . . . .	25
5.3. Control de esfuerzos . . . . .	26
5.4. Gestión de configuraciones . . . . .	27
5.5. Análisis de riesgos . . . . .	28
<b>6. Conclusiones</b>	<b>29</b>
6.1. Resumen de aportaciones . . . . .	29
6.2. Trabajo futuro . . . . .	29
6.3. Valoración personal . . . . .	30

<b>Referencias</b>	<b>31</b>
<b>Anexos</b>	<b>33</b>
<b>A. API del servidor (nuevas operaciones)</b>	<b>33</b>
A.1. Notificaciones . . . . .	33
A.2. Usuarios . . . . .	40
A.3. Comprobaciones del token . . . . .	41
A.4. Correo electrónico . . . . .	41
<b>B. API del servidor (operaciones ya existentes)</b>	<b>42</b>
B.1. Puntos de interés . . . . .	42
B.2. Fotos . . . . .	42
B.3. Usuarios . . . . .	43
B.4. Búsquedas y Estancias . . . . .	43
<b>C. Manual de usuario</b>	<b>44</b>
C.1. Cliente web administrador . . . . .	44
C.1.1. Pantalla de inicio de sesión . . . . .	44
C.1.2. Pantalla principal . . . . .	45
C.1.3. Pantalla de gestión de cambios . . . . .	46
C.1.4. Pantalla de revisión de un cambio . . . . .	47
C.1.5. Pantalla de revisión de la imagen de un cambio . . . . .	49
C.1.6. Pantalla de eliminación de un cambio . . . . .	50
C.1.7. Pantalla de revisión de una incidencia . . . . .	51
C.1.8. Pantalla de gestión de usuarios . . . . .	52
C.1.9. Pantalla de edición de usuarios . . . . .	53
C.1.10. Pantalla de eliminación de un usuario . . . . .	54
C.2. Cliente móvil y web . . . . .	55
C.2.1. Pantalla principal de la aplicación . . . . .	55
C.2.2. Pantalla de inicio de sesión . . . . .	57
C.2.3. Pantalla de visualización del mapa de un campus . . . . .	58
C.2.4. Pantalla de visualización del plano de una planta . . . . .	59
C.2.5. Pantalla de visualización de los detalles de un espacio . . . . .	60
C.2.6. Pantalla de creación de una notificación . . . . .	61
C.2.7. Pantalla de gestión de cambios . . . . .	62
C.2.8. Pantalla de actualización de cambios . . . . .	63
<b>D. Pruebas sistemáticas</b>	<b>65</b>
D.1. Cliente administrador . . . . .	65
D.2. Cliente móvil . . . . .	66
<b>E. Casos de uso</b>	<b>68</b>



# Índice de figuras

1.	Diagrama de actividad que describe el protocolo de comunicación de cambios en los espacios. . . . .	6
2.	Diagrama de actividad que describe el protocolo de comunicación de incidencias en los espacios. . . . .	6
3.	Diagrama de componentes que describe la arquitectura del sistema. . . . .	10
4.	Componentes de lógica de negocio y Spring Security detallados. . . . .	13
5.	Diagrama de paquetes que describe la estructura del código del servidor. . . . .	14
6.	Diagrama de clases y paquetes que detalla el paquete <i>com.uzapp.rest</i> . . . . .	14
7.	Diagrama de clases y paquetes que detalla el paquete <i>com.uzapp.security</i> . . . . .	15
8.	Diagrama de secuencia que describe el protocolo de seguridad de la API con el uso de JWT. . . . .	17
9.	Modelo de datos de la aplicación. . . . .	18
10.	Distribución de horas por componente desarrollado. . . . .	26
11.	Distribución de horas por componente y actividad desarrollados. . . . .	26
12.	Pantalla de inicio de sesión. . . . .	44
13.	Pantalla principal de la aplicación web de administrador. . . . .	45
14.	Pantalla de gestión de todos los cambios registrados en la aplicación. . . . .	46
15.	Pantalla de revisión de un cambio. . . . .	47
16.	Pantalla de revisión de un cambio con imagen asociada. . . . .	48
17.	Pantalla de revisión de la imagen de un cambio. . . . .	49
18.	Pantalla de eliminación de un cambio. . . . .	50
19.	Pantalla de revisión de una incidencia. . . . .	51
20.	Pantalla de gestión de usuarios. . . . .	52
21.	Pantalla de editar de usuarios. . . . .	53
22.	Pantalla de eliminación de un usuario. . . . .	54
23.	Pantalla principal del cliente de móvil. . . . .	55
24.	Pantalla principal del cliente de escritorio. . . . .	56
25.	Pantalla principal del cliente de móvil con el menú lateral desplegado. . . . .	56
26.	Pantalla de inicio de sesión de la aplicación. . . . .	57
27.	Pantalla principal de la aplicación con mensaje de bienvenida. . . . .	57
28.	Pantalla de visualización del mapa del <i>Campus Río Ebro</i> . . . . .	58
29.	Pantalla de visualización del mapa del <i>Campus Río Ebro</i> . . . . .	58
30.	Pantalla de visualización del plano de una planta. . . . .	59
31.	Pantalla de elección de una acción sobre un espacio. . . . .	59
32.	Pantalla de visualización de los detalles de un espacio. . . . .	60
33.	Pantalla de visualización de una imagen panorámica. . . . .	60
34.	Pantalla de creación de una incidencia en un espacio. . . . .	61
35.	Pantalla de creación de un cambio en un espacio. . . . .	61
36.	Pantalla de visualización del menú lateral desplegado con notificación. . . . .	62
37.	Pantalla de gestión de los cambios creados por el usuario. . . . .	62
38.	Pantalla de actualización de cambios. . . . .	63
39.	Pantalla de actualización de cambios (continuación). . . . .	63
40.	Pantalla de gestión de mis cambios. . . . .	64

# Índice de tablas

1.	Descripción de requisitos funcionales. . . . .	8
2.	Descripción de requisitos no funcionales. . . . .	9
3.	Análisis de riesgos. . . . .	28
4.	Casos de uso: Crear una incidencia con imagen y correo electrónico .	68
5.	Casos de uso: Crear un cambio con imagen . . . . .	68
6.	Casos de uso: Actualizar un cambio . . . . .	69
7.	Casos de uso: Visualizar imagen panorámica en 360° . . . . .	69
8.	Casos de uso: Visualizar todos los cambios registrados . . . . .	70
9.	Casos de uso: Visualizar todas las incidencias registradas . . . . .	70
10.	Casos de uso: Gestionar un cambio . . . . .	71
11.	Casos de uso: Gestionar una incidencia . . . . .	71
12.	Casos de uso: Modificar permisos de un usuario . . . . .	72



# 1. Introducción

Este documento describe el desarrollo del proyecto realizado para el Trabajo Fin de Grado titulado *Ampliación de una aplicación móvil, y su panel de administración web, para consultar y gestionar datos sobre los edificios de la Universidad de Zaragoza*.

## 1.1. Contexto del proyecto

### 1.1.1. Colaboración con UTC

Este proyecto ha sido desarrollado desde el mes de octubre de 2016 hasta junio de 2017. El trabajo realizado se basa en ampliar y mejorar una aplicación ya existente, en consecuencia, el desarrollo ha estado marcado por la colaboración con la Unidad Técnica de Construcciones y Energía (UTC) y con otro compañero que ha trabajado sobre la aplicación de forma paralela y que todavía no ha concluido su proyecto en el momento de redacción de este documento. Así, las contribuciones realizadas por este proyecto han sido elaboradas por Alejandro Royo Amondarain con la ayuda y supervisión de Rubén Béjar Hernández.

### 1.1.2. Contexto tecnológico

Para el desarrollo de este proyecto y dado que se trabaja sobre el resultado de proyectos previos, se han conservado las tecnologías utilizadas anteriormente y se han añadido nuevas tecnologías en función de las necesidades a cubrir. Debido al carácter multiplataforma de uno de los clientes se ha utilizado el framework *Ionic* [7] y para el desarrollo web se ha hecho uso de tecnologías como *HTML5*, *CSS* y *Javascript*. Este framework permite la generación de aplicaciones para diversas plataformas, como es el objetivo del proyecto (aplicación web y Android).

En el caso del servidor, se ha mantenido el uso del framework *Spring* [10] y se han utilizados módulos de este framework para dar soporte a algunas de las nuevas funcionalidades como es el caso con la inclusión de *Spring Security* [11].

## 1.2. Objetivo

El objetivo de este Trabajo Fin de Grado es mejorar una aplicación de ayuda y guía al usuario para facilitar la gestión de los espacios y sus modificaciones al administrador y así potenciar el uso de la aplicación.

El estado de la aplicación previo a la realización de este proyecto ofrece al usuario la posibilidad de consultar información sobre los espacios de la universidad sobre un mapa. Se busca aprovechar el sistema definido para ampliar la interacción del usuario con el mapa y permitir que pueda enviar modificaciones (asociadas a espacios concretos) a un usuario administrador de manera que éste tenga información más precisa de las modificaciones realizadas, así como de posibles eventos inesperados en los espacios.

Se quiere ampliar el panel de administrador para ofrecer al usuario la posibilidad de gestionar las modificaciones que se registren desde el cliente móvil, de manera que se produzca una conversación con el usuario creador de la notificación que facilite la tarea de gestión de los espacios al administrador.

Por otro lado y con el objetivo de acercar la aplicación a un estado en el que poder abrir su uso al público, se quiere añadir seguridad al sistema, de forma que el sistema de soporte a múltiples grupos de usuarios en función de sus permisos.

### **1.3. Motivación**

La principal motivación para llevar a cabo este proyecto surge de la posibilidad de crear una aplicación móvil que ayude a alumnos y profesores a desplazarse más fácilmente por los espacios de la universidad y así conocerlos mejor. Otro aspecto interesante que ha influido en la realización de este trabajo es el escenario de colaboración con un cliente real que tiene sus propias necesidades.

Este proyecto se centra en mejorar la gestión del administrador de los distintos espacios y las modificaciones que se realizan sobre ellos. Esto es interesante porque obliga a hablar con el cliente y discutir las diferentes posibilidades desde las que abordar el problema y así llegar a un entendimiento.

### **1.4. Estructura del documento**

Este apartado describe el esquema general seguido al redactar este documento así como una breve descripción del contenido de cada sección. El documento consta de 6 secciones, la bibliografía utilizada y los anexos incluyendo un índice de figuras y un índice de tablas.

La primera sección (*Introducción 1*) describe los aspectos generales del proyecto, como el contexto en el que se desarrolla, el objetivo y la motivación que lleva a realizar este trabajo.

En la segunda sección (Análisis del problema 2) se detalla la descripción del problema a abordar, un análisis de la problemática del cliente y los requisitos derivados de dicho análisis.

En la tercera sección (Diseño de la solución 3) se plantea la propuesta de solución elaborada por este proyecto al problema descrito, así como detalles sobre el diseño de los diferentes componentes del sistema ampliado. Esto incluye el diseño del protocolo de seguridad, del sistema de comunicación de notificaciones y el modelo de datos utilizado.

La cuarta sección (Implementación de la solución 4) realiza una descripción de las tecnologías utilizadas en el desarrollo de cada componente del sistema, los problemas encontrados y las pruebas llevadas a cabo.

La quinta sección (Gestión del proyecto 5) explica los detalles relacionados con la planificación del proyecto, el control de esfuerzos, la gestión de configuraciones o la comunicación con el cliente.

La sexta sección (Conclusiones 6) incluye un resumen de las aportaciones realizadas por este proyecto, posibles líneas de trabajo futuras y una valoración personal sobre el proyecto en general.

Por último, los anexos (A) contienen todo el detalle no incluido en las secciones anteriores de este documento. Esto es, la documentación de la API del servidor, un manual de usuario de los dos clientes sobre los que se trabaja, un conjunto de pruebas sistemáticas, los diferentes casos de uso planteados y un análisis de riesgos.

## 2. Análisis del problema

### 2.1. Descripción del problema

La Universidad de Zaragoza consta de numerosos campus, facultades y espacios en general repartidos entre Huesca, Zaragoza y Teruel. Para un alumno o profesor familiarizados con la universidad puede resultar sencillo desplazarse por algunos de los espacios, pero para una persona ajena o recién llegada a la misma supone un problema la localización de su destino. Este problema es el que intenta cubrir la aplicación original, ofreciendo una guía al usuario mediante mapas y planos de los distintos edificios y espacios de la universidad.

En el contexto de colaboración con la UTC y con el objetivo de continuar el trabajo realizado con la aplicación, se plantea un problema principal relacionado con el registro de modificaciones en los espacios.

Los espacios de la universidad se modifican con bastante frecuencia. Actualmente, desde que se realiza una modificación en un espacio hasta que se registra puede transcurrir mucho tiempo. Esto es debido a que no existe un protocolo de comunicación entre las personas que llevan a cabo las modificaciones y los responsables de la gestión de los espacios de la universidad (UTC).

Por último, con el objetivo de permitir el uso de la aplicación por parte de la universidad, resulta necesario definir una comunicación segura que abarque tanto la parte del panel de administrador como el cliente web.

#### 2.1.1. Sistema de comunicación de notificaciones

Una de las necesidades del cliente es definir un protocolo de comunicación de notificaciones entre usuario y administrador que soporte la aplicación para facilitar la comunicación de dichos eventos. A continuación se describen los problemas encontrados y las posibles soluciones planteadas.

Actualmente existe la posibilidad de comunicar cualquier tipo de información a través del correo electrónico, pero el objetivo es precisamente integrar un sistema de comunicación de notificaciones en la aplicación, de manera que sea más cómodo para administrador y usuario comunicar las modificaciones en los espacios y por lo tanto fomente su uso continuado.

Este sistema tiene que facilitar tanto la creación de las de las notificaciones por parte del usuario como la revisión de las mismas para el administrador. El objetivo es que el usuario pueda abrir la aplicación desde un espacio después de haber realizado una modificación y enviar una notificación con una descripción y posiblemente una imagen para aportar más detalle. Por otro lado, el administrador tiene que poder recibir múltiples notificaciones en un mismo lugar, revisar sus detalles y tomar una decisión.

Para facilitar una comunicación fluida de las modificaciones se propone soportar una conversación entre administrador y usuario, la cual permita corregir la modificación al usuario mientras el administrador se lo pida. De esta manera, el usuario debe mantener un registro de las notificaciones creadas. A partir de ahora este tipo de notificación se llamará **cambio**.

Por otro lado, se plantea la posibilidad de permitir a cualquier usuario enviar notificaciones a modo de aviso al administrador, de forma que éste las resuelva sin necesidad de mantener una conversación con el usuario. Este escenario no acepta un registro de las notificaciones en la aplicación móvil ya que el usuario que las crea es anónimo. A partir de ahora este tipo de notificación se llamará **incidencia**.

Respecto al método de comunicación de notificaciones, hay que diferenciar dos tipos de comunicación, la que realiza el usuario al administrador al crear la notificación y la que lleva a cabo el administrador al responder al usuario.

- **Usuario crea notificación:** el usuario utiliza la aplicación móvil para registrar una notificación en el sistema y el administrador la visualiza en su aplicación web. No existen diferencias entre cambios e incidencias.
- **Administrador resuelve notificación:** el administrador toma una decisión sobre la notificación y se lo comunica al usuario. En este punto, se diferencian dos opciones, correspondientes a los tipos de notificación definidos anteriormente:
  - **Cambio:** el administrador responde al usuario a través de la aplicación, con el objetivo de que éste pueda modificar la notificación y así mantener una conversación hasta que el administrador decida dar por resuelto el cambio.
  - **Incidencia:** el administrador resuelve la notificación y envía un correo al usuario para comunicar y explicar la decisión tomada al respecto. Se evita mantener una conversación con el objetivo de ofrecer el uso de este tipo de notificación a cualquier usuario.



A continuación se detalla el flujo de comunicación para cambios e incidencias por separado.

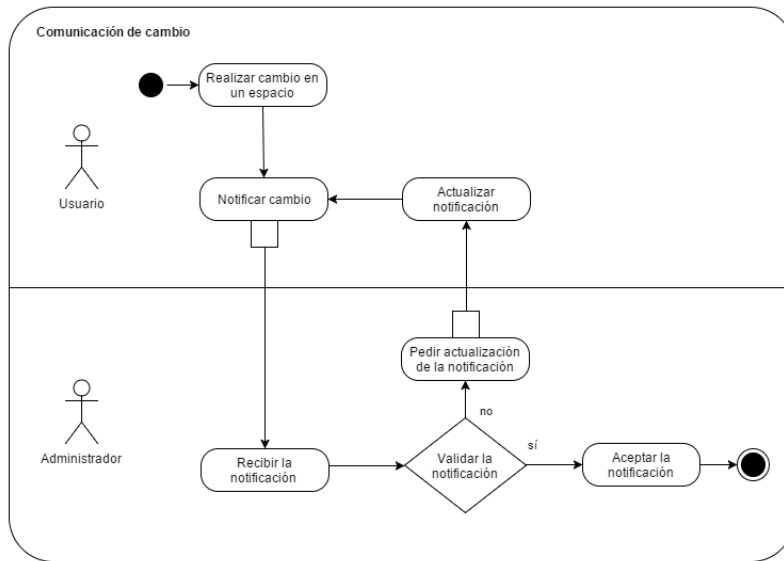


Figura 1: Diagrama de actividad que describe el protocolo de comunicación de cambios en los espacios.

Un cambio es una notificación creada por usuarios registrados que tiene como objetivo comunicar una modificación física de un espacio determinado. Consta de localización, descripción y opcionalmente una imagen.

En el flujo de comunicación destaca que una vez el usuario ha notificado el cambio, es el administrador quien debe decidir el siguiente paso. El usuario sólo puede volver a interactuar si el administrador pide una revisión de la información proporcionada. En caso de aprobación o rechazo del cambio, la interacción se termina y el usuario puede visualizar la decisión tomada y la justificación. El único que puede terminar la comunicación es el administrador.

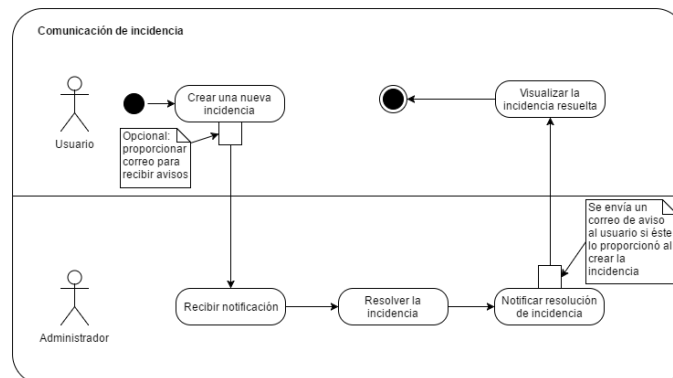


Figura 2: Diagrama de actividad que describe el protocolo de comunicación de incidencias en los espacios.

Una incidencia es una notificación creada por cualquier tipo de usuario que tiene como objetivo comunicar un evento inesperado en un espacio determinado. Contiene una descripción y opcionalmente una imagen y una dirección de correo electrónico para poder recibir una confirmación de que la incidencia ha sido resuelta.

En el diagrama se observa una diferencia principal con la comunicación de cambios y es que en el caso de las incidencias no existe un bucle interacciones entre usuario y administrador. Esto es debido a que el propósito de las incidencias es avisar de un evento inesperado producido en algún espacio de la universidad de manera que el responsable de dicho espacio pueda actuar en consecuencia. Así, cuando el administrador recibe la notificación de la incidencia debe resolverla y comunicar al usuario, si así lo desea, las razones de su decisión o las medidas llevadas a cabo para resolver dicha incidencia.

## 2.2. Requisitos

A continuación se detallan los requisitos funcionales y no funcionales que deben cubrir las ampliaciones y mejoras realizadas sobre la aplicación. Los casos de uso que cubren los requisitos definidos se encuentran detallados en el anexo E.

### 2.2.1. Requisitos funcionales

Código	Descripción del requisito
<b>RF1</b>	La aplicación móvil permitirá a un usuario cualquiera notificar una incidencia en el espacio seleccionado.
<b>RF2</b>	La aplicación móvil permitirá a un usuario cualquiera enviar una imagen al notificar una incidencia.
<b>RF3</b>	La aplicación móvil permitirá a un usuario cualquiera adjuntar un correo al notificar una incidencia para recibir un correo cuando ésta sea resuelta.
<b>RF4</b>	La aplicación móvil permitirá a un usuario registrado notificar un cambio en el espacio seleccionado.
<b>RF5</b>	La aplicación móvil permitirá a un usuario registrado enviar una imagen al notificar un cambio.
<b>RF6</b>	La aplicación móvil permitirá a un usuario registrado visualizar los cambios notificados y sus estados.
<b>RF7</b>	La aplicación móvil permitirá a un usuario registrado actualizar un cambio notificado anteriormente.
<b>RF8</b>	La aplicación móvil permitirá a un usuario cualquiera visualizar imágenes panorámicas de los espacios que dispongan de ellas.
<b>RF9</b>	El panel de administrador permitirá al usuario visualizar los cambios e incidencias notificados por el resto de usuarios.
<b>RF10</b>	El panel de administrador permitirá al usuario aprobar, rechazar o pedir una revisión de un cambio.
<b>RF11</b>	El panel de administrador permitirá al usuario aprobar o rechazar una incidencia.
<b>RF12</b>	El panel de administrador enviará un correo al usuario creador de la incidencia cuando ésta sea resuelta (si el usuario ha proporcionado un correo).
<b>RF13</b>	El panel de administrador permitirá al usuario visualizar un listado de todos los usuarios registrados en la aplicación.
<b>RF14</b>	El panel de administrador permitirá al usuario modificar los permisos de un usuario registrado en la aplicación.

Tabla 1: Descripción de requisitos funcionales.

### 2.2.2. Requisitos no funcionales

<b>Código</b>	<b>Descripción del requisito</b>
<b>RNF1</b>	La aplicación móvil y el panel de administrador se comunicarán de forma segura con el servidor.
<b>RNF2</b>	La aplicación móvil ofrecerá una interfaz de visualización de mapas más usable.
<b>RNF3</b>	La aplicación móvil mejorará el rendimiento en el pintado de mapas.

Tabla 2: Descripción de requisitos no funcionales.

### 3. Diseño de la solución

En este apartado se describe el diseño de la solución propuesta realizando una comparativa entre el diseño de la aplicación original y las ampliaciones y mejoras llevadas a cabo. En primer lugar se explica el diseño de la arquitectura del sistema original y los cambios realizados. Posteriormente se describe el protocolo de seguridad diseñado.

#### 3.1. Arquitectura del sistema

Se ha diseñado sobre la arquitectura de la aplicación original, con el objetivo de aprovechar el trabajo realizado en proyectos previos. Esto es, se mantiene una arquitectura de cliente-servidor de tres niveles, con dos clientes claramente diferenciados (cliente web administrador y cliente móvil), dos servidores (el servidor de la aplicación con el que se comunican ambos y *GeoServer* que actúa como servidor de mapas) y la base de datos donde se almacena la información.

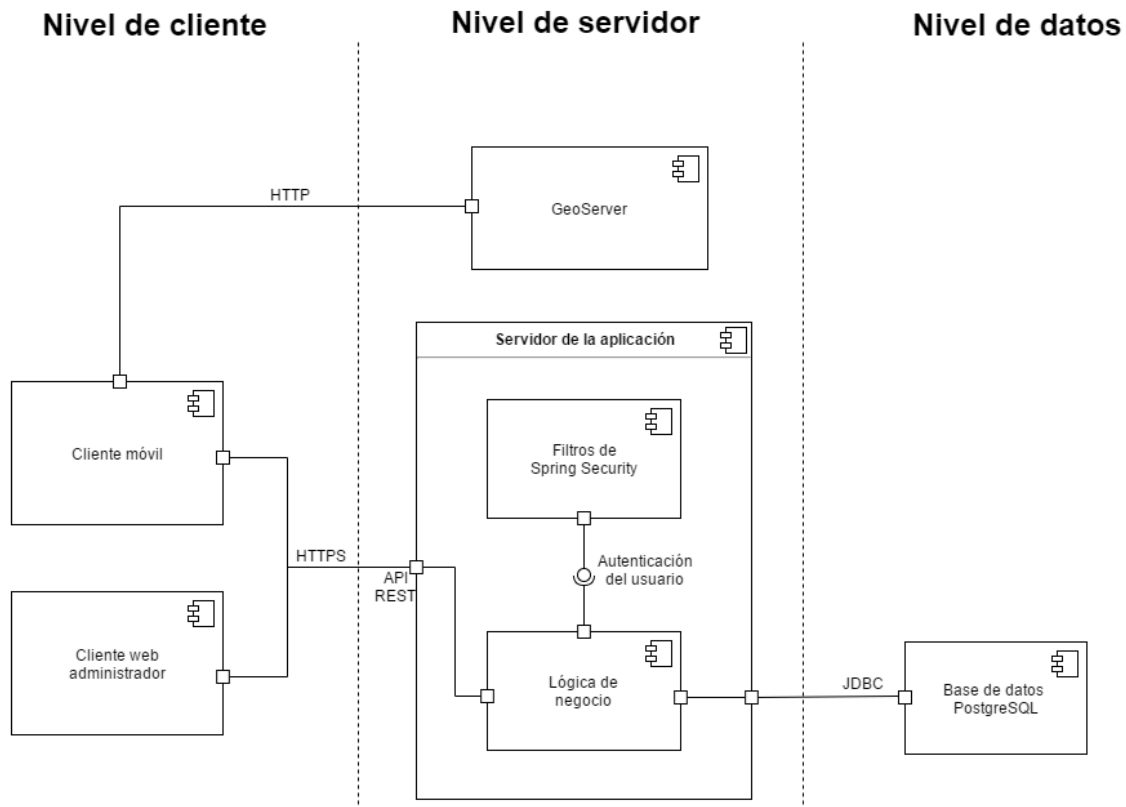


Figura 3: Diagrama de componentes que describe la arquitectura del sistema.

En la aplicación original el esquema es el mismo, pero las comunicaciones entre los clientes y el servidor de la aplicación se hacen sobre el protocolo *HTTP* y no *HTTPS* como es ahora. Esto es debido a que es necesario hacer uso de un protocolo seguro para que la aplicación se pueda utilizar de manera pública.

Destaca la inclusión del componente de *Filtros de Spring Security* en el diagrama, que se encarga de filtrar las peticiones al servidor en función de los permisos que posea cada usuario. El diseño de estos filtros se detalla más adelante.

### 3.1.1. Cliente web administrador

En el caso de la aplicación dirigida al administrador se ha modificado el tipo de comunicación con el servidor y se ha mantenido la misma estructura de código que había previamente. A continuación se describe el estado de la aplicación original así como los cambios realizados durante este proyecto.

La aplicación original tiene un componente de seguridad que controla el acceso de los usuarios. Este componente valida las credenciales cuando el usuario inicia sesión en la aplicación y cada vez que se carga una nueva vista se comprueba la existencia del objeto representante de los permisos del usuario en el cliente obtenido en el inicio de sesión. Sin embargo, las peticiones al servidor no están controladas, el nivel de seguridad se limita al cliente.

Debido a los nuevos filtros incluidos en la API del servidor, el componente de seguridad del cliente funciona de manera diferente. Por un lado, el inicio de sesión exitoso proporciona un *token* que se almacena en el cliente. Este *token* contiene la identidad del usuario y los permisos asociados. En cada petición a la API del servidor se adjunta el *token* para validar la identidad y permisos del usuario administrador. Adicionalmente, mientras se navega por la aplicación también se realizan comprobaciones automáticas de la validez del *token*. Si el *token* resulta modificado en el cliente o caduca y se vuelve a enviar al servidor, éste no lo validará y el cliente verá expirada su sesión inmediatamente.

En cuanto al sistema de comunicación de notificaciones, se ha usado un método similar al ya utilizado para la gestión de *pois* e *imágenes* en la aplicación original. Se ofrece una vista de todos los cambios o incidencias que permite realizar acciones como la revisión, aprobación o rechazo de los mismos. Por otro lado, se ha añadido la gestión de los usuarios con el objetivo de que el administrador pueda crear y modificar los permisos asociados a cada usuario con mayor comodidad.

### 3.1.2. Cliente móvil

La aplicación móvil se ha desarrollado con el framework *Ionic* [7], con el objetivo de obtener tanto una aplicación web como una móvil con el mismo código. En este caso, la exportación del código a distintos dispositivos no se limita a su uso, sino que ambas versiones están diseñadas teniendo en cuenta diferentes dimensiones de las pantallas.

Entre los añadidos más relevantes destaca la necesidad de crear notificaciones asociadas a un espacio, para lo cual se ha modificado sobre la vista del mapa para permitir al usuario la notificación de cambios e incidencias dependiendo de los permisos disponibles. Sólo un usuario registrado puede notificar cambios.

Para soportar la conversación necesaria entre administrador y usuario registrado acerca de los cambios notificados se ha definido una nueva vista que permita al usuario visualizar sus cambios según su estado actual. Se pueden diferenciar cuatro estados: *pendientes* (el usuario debe actualizar la información), *esperando* (el administrador tiene que revisar el cambio), *aprobados* y *rechazados*.

Cuando el usuario crea un cambio desde el mapa, éste pasa a aparecer en su lista de cambios a la espera de una respuesta del administrador. Una vez revisado el cambio, el usuario puede apreciar una alerta visual que le avisa del número de cambios pendientes que tiene que resolver.

Para obtener los nuevos cambios del servidor y conocer si se han producido modificaciones, se ha planteado la posibilidad de crear un sistema de notificaciones *push* en el servidor que permita avisar a los clientes suscritos pero se ha descartado por resultar demasiado costoso. De esta manera, se ha optado por actualizar periódicamente la información sobre los cambios asociados al usuario que esta usando la aplicación. Así, aunque el usuario esté visualizando el mapa o una pantalla distinta a la correspondiente a sus cambio, la información se actualiza y se notifica visualmente de las novedades.

Por último, la estructura de clases y paquetes se mantiene respecto a la aplicación original. Destaca un servicio que añade el *token* a cada petición del usuario al servidor (de forma similar al cliente web administrador), pero el resto de adiciones siguen el mismo patrón de diseño proporcionado por *AngularJS* [2] e *Ionic*, el patrón **Modelo Vista VistaModelo (MVVM)**. Esto es, cada modificación sobre los datos, tanto desde la vista como desde el modelo, se ve reflejada en la otra parte.

### 3.1.3. Servidor

Uno de los objetivos principales del proyecto era añadir un módulo de seguridad a la aplicación. De esta manera, se ha revisado toda la estructura del servidor para incluir un servicio de autenticación que pudiera ser utilizado por cualquiera de los clientes actualmente desarrollados para la aplicación.

Dado que se trata de modificar una aplicación ya existente se ha buscado un mecanismo de seguridad que cubra el objetivo propuesto sin necesidad de realizar grandes modificaciones sobre la estructura original de la aplicación. En este aspecto, se ha aprovechado que el servidor estaba desarrollado en *Spring* para hacer uso del módulo *Spring Security*, lo que facilita la integración con el código original.

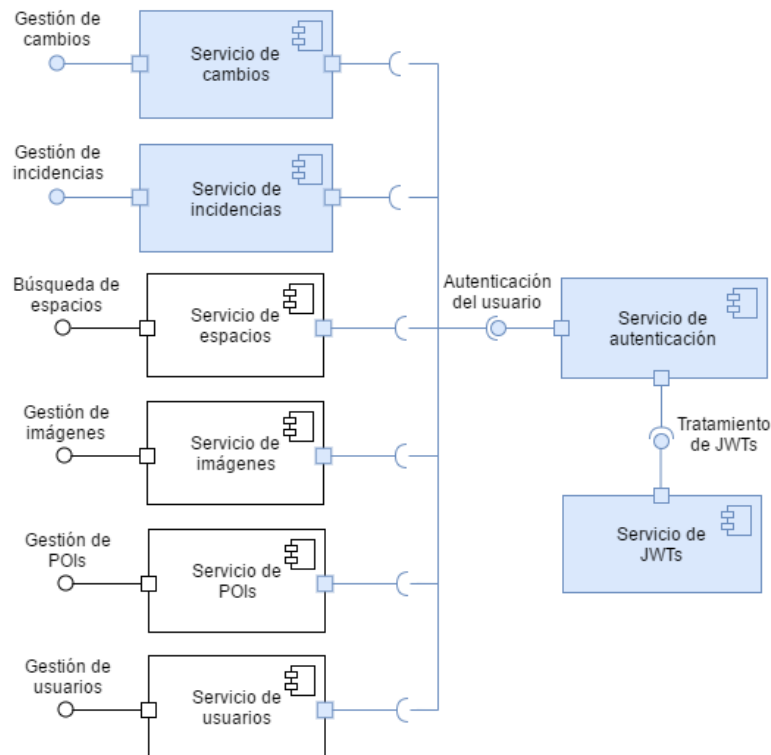


Figura 4: Componentes de lógica de negocio y Spring Security detallados.

Destacan en azul los componentes incluidos en el desarrollo de este proyecto. Por un lado se encuentran los componentes *Servicio de autenticación* y *Servicio de JWTs* (que representan el componente de *Filtros de Spring Security* (figura 3), que se encarga de filtrar todas las peticiones al servidor y comprobar si el usuario posee los permisos adecuados para llevar a cabo cada tipo de petición. Como se puede observar, los filtros afectan tanto a los nuevos componentes de cambios e incidencias como a los componentes que ya estaban presentes en la aplicación.

A continuación se describe la estructura del código del servidor con las ampliaciones realizadas:



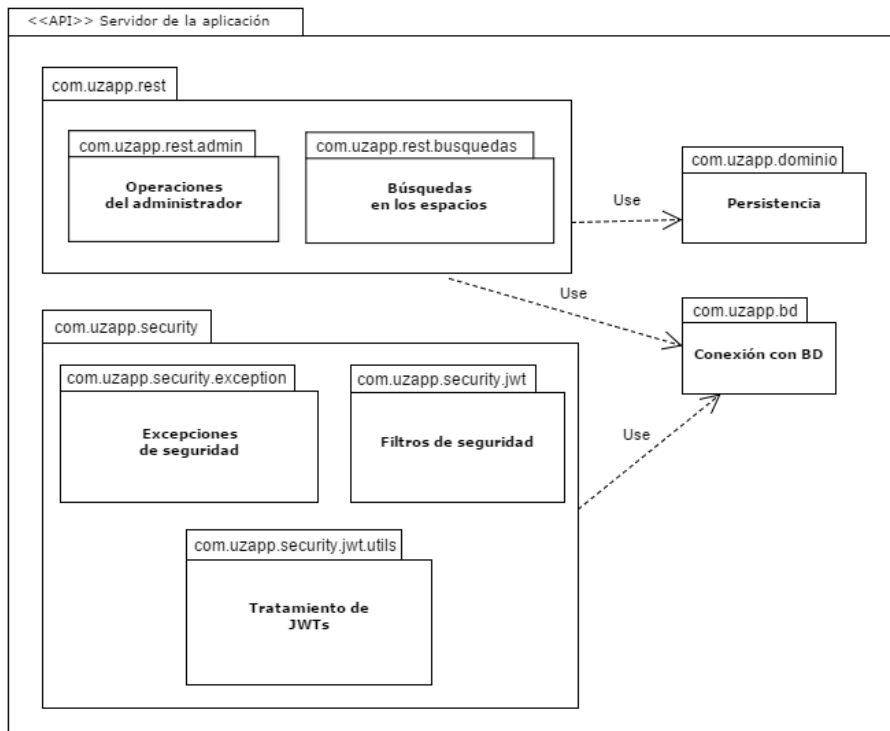


Figura 5: Diagrama de paquetes que describe la estructura del código del servidor.

Se observa la inclusión del paquete *com.uzapp.security* que contiene el código correspondiente a los filtros personalizados de *Spring Security*. Además, se han realizado modificaciones en el resto de paquetes para incluir las nuevas funcionalidades como las notificaciones. A continuación se detalla el contenido de los paquetes más complejos.

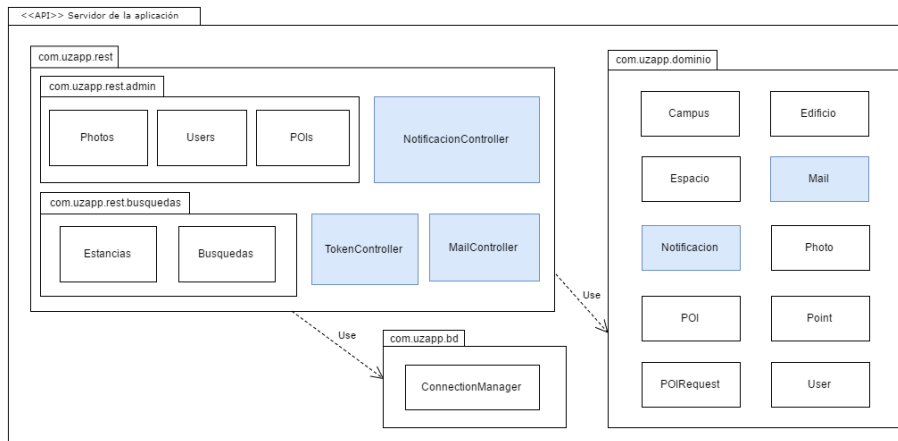


Figura 6: Diagrama de clases y paquetes que detalla el paquete *com.uzapp.rest*

Todas las clases incluidas en el paquete *com.uzapp.rest* contienen el código referente a la lógica de la API. No se detallan las operaciones implementadas en cada clase porque se corresponden directamente con los *endpoints* soportados por la propia API que se define en los anexos A y B. Las clases representadas en azul corresponden a las nuevas funcionalidades. Son las siguientes:

- **NotificacionController**: contiene el código de las operaciones de la API que soportan el sistema de comunicación de cambios e incidencias.
- **TokenController**: ofrece operaciones de validación y comprobación del JWT (Json Web Token [14]) que proporciona el usuario al realizar una petición.
- **MailController**: permite enviar un correo electrónico.

Respecto al paquete *com.uzapp.dominio* destacan las clases *Notificacion* y *Mail* que son nuevas en este proyecto. Las clases de este paquete hacen referencia a las entidades del modelo de datos descrito más adelante, en concreto son las siguientes: *Edificio*, *Espacio*, *Notificación*, *Photo*, *POI*, y *User*.

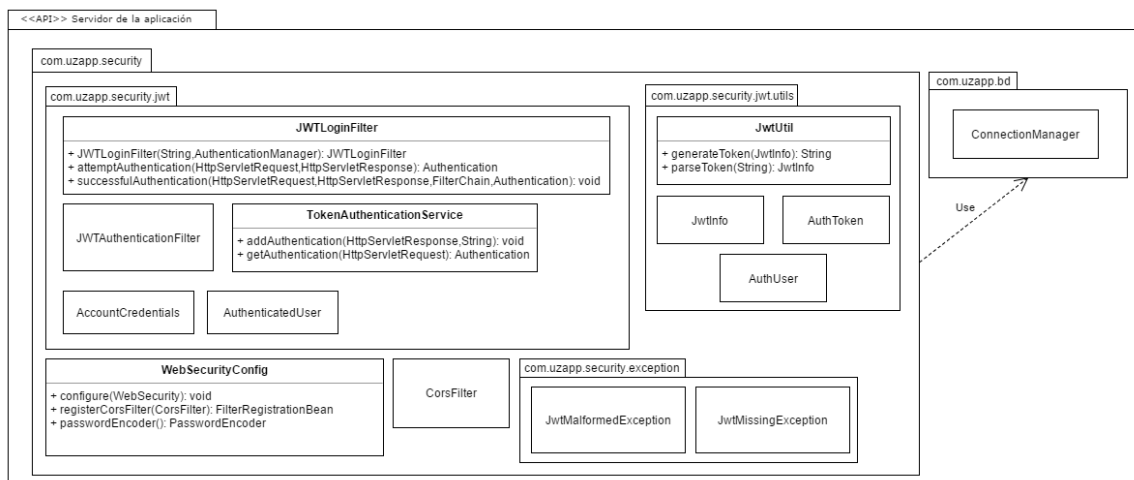


Figura 7: Diagrama de clases y paquetes que detalla el paquete *com.uzapp.security*

Todas las clases y paquetes contenidos en el paquete *com.uzapp.security* han sido desarrollados dentro de este proyecto. En el paquete *com.uzapp.security.jwt* se encuentran los filtros personalizados que permiten la autenticación del usuario de dos maneras distintas:

- **JWTLoginFilter**: esta clase se encarga de comprobar las credenciales del usuario si éste no ha proporcionado un *JWT* que le identifique.
- **JWTAuthenticationFilter**: este filtro es responsable de comprobar si el usuario ha proporcionado un *JWT* y autenticar a dicho usuario si el token es válido.

El paquete *com.uzapp.security.jwt.utils* que contiene todo el código relacionado con el tratamiento de los *JWTs*, es decir, generación, validación y acceso a dichos *tokens*. Por otro lado, la clase *WebSecurityConfig* es la encargada de definir la configuración de la seguridad en la aplicación. Contiene desde la configuración del protocolo *CORS* añadiendo un filtro personalizado, hasta la definición de los puntos de acceso al servidor detallando los permisos necesarios para cada uno.

## 3.2. Protocolo de seguridad

El objetivo es facilitar una comunicación segura entre cliente y servidor. Se han tenido en cuenta las siguientes alternativas para construir el protocolo de comunicación:

- **Autenticación basada en cookies:** Este tipo de autenticación está basada en guardar la sesión. El usuario introduce sus credenciales y el servidor guarda un objeto de sesión que se almacena en la base datos. Posteriormente, el cliente envía con cada petición al servidor la *cookie* con el identificador del usuario. Esto hace que para cada petición del cliente, el servidor realice peticiones a la base de datos, lo que aumenta el tiempo de respuesta.
- **Autenticación basada en JSON Web Tokens (JWTs):** Este tipo de autenticación no guarda la sesión. Una vez el usuario introduce sus credenciales, el servidor genera un token que envía en la respuesta al usuario para que este lo guarde. Este token contiene información no sensible sobre el usuario (identificador, permisos...) y será enviado en cada petición al servidor.

La principal ventaja es que se trata de un mecanismo que no guarda el estado, lo que permite soportar un gran número de usuario sin necesidad de almacenar más información, simplemente genera un nuevo token para cada nuevo usuario. Esto es, es escalable.

Por otro lado, dado que el token puede guardar información como los permisos del usuario, el servidor puede detectar rápidamente qué tipo de usuario está realizando la petición y actuar en consecuencia (sin tener que pedir más información a la base de datos).

Adicionalmente, la comunicación de *tokens* en las peticiones realizadas al servidor es un sistema fácil de implementar para cualquier tipo de dispositivo, ya que solamente es necesario incluirlos en las cabeceras de dichas peticiones.

Así, teniendo en cuenta que la aplicación tiene que dar soporte a dos clientes y a distintos tipos de usuario con diferentes niveles de privilegios, se ha optado por utilizar la autenticación basada en *JSON Web Tokens*. A continuación, se muestra un diagrama del protocolo de comunicación diseñado con este mecanismo de seguridad que describe el procesamiento del servidor con cada petición del usuario.

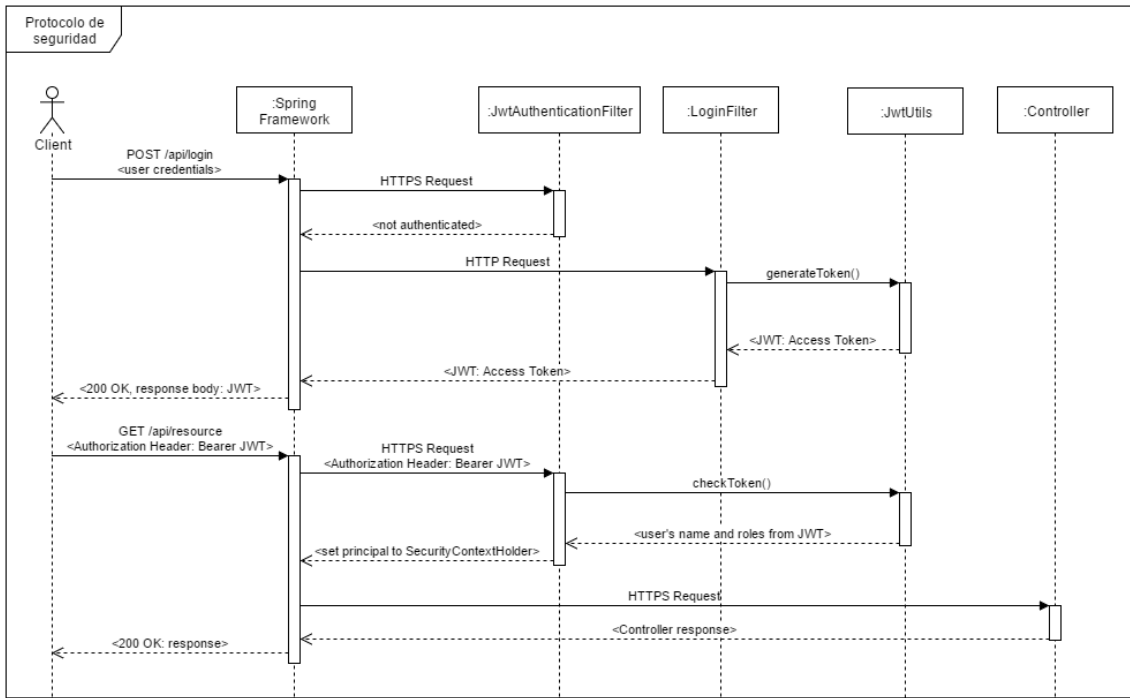


Figura 8: Diagrama de secuencia que describe el protocolo de seguridad de la API con el uso de JWT.

Este diagrama detalla el nuevo flujo de comunicación con los filtros de seguridad añadidos y con cada petición al servidor. Cuando el usuario proporciona sus credenciales (nombre de usuario y contraseña) correctamente, el servidor genera un *JWT* que devuelve en la respuesta. De esta manera, el cliente añade a cada petición posterior dicho token para validar sus permisos.

Respecto al servidor, se ha situado en primer lugar el filtro correspondiente a la validación del *JWT* y en segundo lugar el filtro que comprueba las credenciales (en caso de que el cliente no proporcione un token). La definición de estos filtros personalizados se integra fácilmente con el framework utilizado, que además ofrece soporte para utilizar la validación de los permisos del usuario en el resto de componentes del servidor.

### 3.3. Modelo de datos

Se han mantenido las entidades presentes en la aplicación original, esto es, los espacios, puntos de interés (pois), imágenes y usuarios. Además, se han incluido nuevas entidades al modelo como son las notificaciones, que incluyen cambios e incidencias. A continuación se describe el modelo de datos propuesto en esta solución:

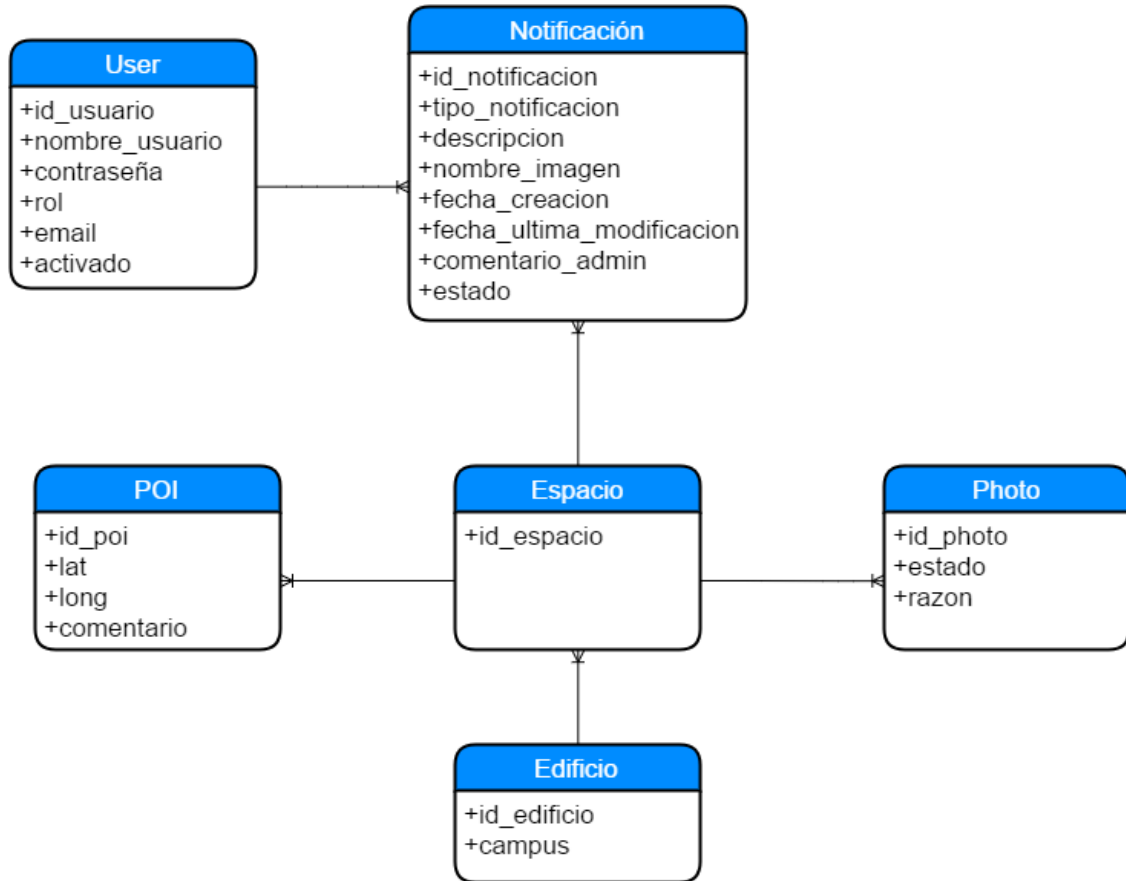


Figura 9: Modelo de datos de la aplicación.

Se observa la inclusión de la entidad *Notificación* que representa los cambios e incidencias utilizados en el sistema de comunicación propuesto.

- **Notificación**: consta de varios atributos relevantes. En primer lugar, *id\_notificacion* actúa a modo de clave. Se utiliza el atributo *tipo\_notificacion* para diferenciar los cambios y las incidencias. Además, aparecen los atributos *descripcion* y *nombre\_imagen*, éste último se utiliza para asociar una imagen a la notificación. Esto es debido a que las imágenes no se almacenan en la base de datos, sino en el propio servidor, de manera que se utiliza el nombre asociado para obtener el contenido de las imágenes.

Este método de almacenamiento es el mismo utilizado con las imágenes asociadas a la entidad *Photo*, aunque éstas representan un espacio (*Espacio*), no un cambio o incidencia sobre el mismo.

- **User:** Esta entidad contiene la información relacionada con los usuarios registrados en la aplicación. Entre sus atributos se encuentran las credenciales (nombre de usuario y contraseña) que utiliza para iniciar sesión en la aplicación. Además, describe el rol del usuario (registrado o administrador) que sirve para determinar a qué tipo de recursos tiene acceso un usuario y almacena un correo electrónico por entidad.

Destaca el atributo *activado* incluido en esta entidad que representa el estado del usuario, de manera que sólo en caso de estar activado un usuario puede utilizar sus credenciales normalmente.

- **Punto de interés (POI):** representa la información asociada a un *poi*, esto es, a un punto en el mapa concreto (latitud y longitud definidas) que puede resultar especialmente relevante para el usuario. Aunque un *poi* tiene coordenadas que lo definen, se asocia a un espacio por el hecho de estar contenido en el mismo, de manera que resulta más sencillo representar la información asociada a un espacio. La creación de un punto de interés es llevada a cabo por el usuario.
- **Espacio:** esta es la entidad que más información almacena, dado que representa toda la información asociada a una estancia de cualquier edificio de la universidad. Posee un identificador del espacio que determina su localización y numerosos atributos que describen las diferentes características de dicho espacio. Algunos ejemplos pueden ser las dimensiones del espacio, el tipo de calefacción, el número de ventanas o su orientación principal.
- **Edificio:** se utiliza para almacenar la información de los edificios de la universidad, como su nombre o el campus al que pertenecen.
- **Photo:** almacena la información relacionada con las imágenes de cada espacio. Dado que un usuario puede proporcionar imágenes sobre un espacio, la entidad consta de un *estado* que determina la validez de la imagen y de una *razón* que soporta la explicación del administrador a su decisión de aprobar o no una imagen aportada por un usuario. Además, tiene un nombre que permite recuperar el contenido de la imagen del servidor donde están guardada.

Las entidades *Punto de interés*, *Espacio*, *Edificio* y *Photo* se mantienen respecto a la aplicación original. Su inclusión en el diagrama tiene como motivación la visualización de las nuevas relaciones creadas y la comparación de éstas entidades con las adiciones. Así, sóloamente se describen los atributos más relevantes de cada entidad ya existente.

## 4. Implementación de la solución

En este apartado se presentan los detalles de la implementación de la solución en sus diferentes componentes como son el cliente web administrador, el cliente móvil y el servidor.

### 4.1. Cliente admin

La tecnología utilizada en el desarrollo del cliente web administrador se ha mantenido respecto a la que ya se había usado para construir la aplicación original. Este cliente es una aplicación web basada en *jQuery* que hace uso principalmente de un *plug-in* (DataTables [6]) para representar la información en un formato de tabla. Esto es especialmente útil para un administrador ya que permite ordenar y buscar los elementos deseados en dicha tabla con mucha flexibilidad.

Se ha conservado el formato de panel de control para acceder a la gestión de los distintos tipos de información a los que tiene acceso el administrador.

El resultado de la implementación del sistema de notificaciones en el cliente web administrador y el funcionamiento de las distintas pantallas se detalla en el anexo C.1.

#### 4.1.1. Problemas encontrados

El problema más relevante ha sido uno relacionado con el *plug-in* ya mencionado de *DataTables* [6]. Al inicio del proyecto y mientras se preparaba el entorno de desarrollo se detectó que el *plug-in* *DataTables* no funcionaba con ciertas versiones de *jQuery*, en concreto la versión de *jQuery* debe ser igual o mayor que la *1.9.1* y menor que la versión 3.

### 4.2. Cliente móvil

En el caso del cliente móvil, se ha mantenido la tecnología ya utilizada. Así, se ha desarrollado con el framework *Ionic* con el objetivo de soportar una aplicación web y principalmente una aplicación móvil.

Para la visualización de imágenes panorámicas se ha utilizado la librería *Photo Sphere Viewer* [8] de Javascript que permite tanto la visualización de este tipo de imágenes en distintos dispositivos como la interacción con las mismas, ofreciendo además cierto nivel de personalización.

El resultado de la implementación del sistema de notificaciones en el cliente móvil y el funcionamiento de las distintas pantallas se detalla en el anexo C.2.

### 4.3. Servidor

Para desarrollar el servidor de la aplicación se ha utilizado el framework *Spring* [10] para la plataforma *Java* [9], en concreto se ha trabajado sobre *Spring Boot* [12], que simplifica la puesta en marcha de la aplicación. Se ha aprovechado el módulo de seguridad de este framework *Spring Security* [11] para integrar fácilmente el protocolo de seguridad planteado en apartados anteriores.

Como tecnología de gestión de dependencias se utiliza *Gradle* [4], que se integra fácilmente con *Spring Boot*.

#### 4.3.1. Problemas encontrados

Al intentar utilizar el framework *Spring Security* para implementar el protocolo de seguridad se ha observado que este módulo no soporta por defecto la gestión de *JWT*. De esta manera, se han creado los filtros personalizados ya mencionados.

### 4.4. Base de datos

Se ha utilizado la misma base de datos de la aplicación original con tecnología *PostgreSQL* [13] y se han incluido entre las entidades almacenadas las nuevas notificaciones, así como la modificación de los usuarios. Además, las contraseñas de los usuarios están ahora encriptadas.

### 4.5. Pruebas

Se han realizado principalmente test de la API del servidor haciendo uso de la herramienta *Postman* [23]. Además de las pruebas realizadas durante el desarrollo para construir la API, se ha definido una batería de test para probar conjuntos de operaciones de manera automática [15].

Esta batería de test se puede probar pero hace falta definir un *environment* que no se ha publicado por motivos de seguridad. Un *environment* es un elemento de la herramienta *Postman* que define el entorno de pruebas necesario. En este caso es necesario definir cuatro variables para poder ejecutar la batería de test:

- *url*: es una cadena que representa la dirección donde se encuentra la API.
- *username*: es una cadena que debe contener el nombre de usuario.
- *password*: es una cadena que debe contener la contraseña del usuario.
- *email*: es una cadena que debe contener el correo electrónico al que se enviará un correo de prueba.

El resto de variables se utilizan para ejecutar diferentes acciones sobre el recurso creado recientemente o para adjuntar siempre el *token* obtenido al iniciar sesión con las credenciales adecuadas.



Con el objetivo de llevar a cabo test en los clientes se han definido guiones de prueba que se utilizan de forma manual y sistemática para probar el correcto funcionamiento de las nuevas interfaces gráficas, aunque estos test no han llegado a ser automatizados. Todos los test de este tipo utilizados durante el desarrollo del proyecto se describen en el anexo D.

## 4.6. Despliegue

La configuración del despliegue del sistema ha sido resuelta por el proyecto realizado en paralelo por otro compañero que todavía no ha concluido. Se ha programado un *script* (accesible desde el repositorio donde se encuentra la aplicación [16]) que sigue el siguiente esquema:

1. Obtiene la última versión del repositorio de *GitHub* que contiene todo el código de la aplicación.
2. Asigna las variables del sistema necesarias para su uso público.
3. Instala las dependencias de los clientes y los publica.
4. Inicia el servidor.

## 4.7. Herramientas de desarrollo

En este apartado se describen las tecnologías utilizadas para desarrollar la aplicación.

### Cliente administrador

- Para programar el cliente: Atom IDE [17]
- Para lanzar el cliente: XAMPP [18]
- Para acceder al cliente: Google Chrome [19]

### Cliente móvil

- Para programar el cliente: Atom IDE
- Para acceder al cliente: Google Chrome

### Servidor

- Para programar el servidor: Spring Tool Suite [20]
- Para programar el servidor: Atom IDE
- Para acceder y realizar pruebas: Postman

### Base de datos

- Para gestionar la base de datos: pgAdmin [21]

## Otros

- Para conectar con la servidor situado en la red de la universidad: Cisco AnyConnect Security Mobile Client [22]
- Para redactar la memoria: Overleaf [24]
- Para mantener el control de versiones del código: GitHub [25]
- Para mantener el control de versiones de la documentación: [28]

## 5. Gestión del proyecto

En esta sección se explica cómo se ha llevado a cabo el proyecto, teniendo en cuenta aspectos como la planificación inicial, la comunicación con el cliente, el control de esfuerzos o la distribución de tiempo por tareas.

### 5.1. Planificación

A continuación se detalla la planificación inicial realizada posteriormente a la propuesta del proyecto y a partir de las ideas debatidas inicialmente con el cliente. El desarrollo del proyecto se extiende desde septiembre de 2016 hasta junio de 2017.

- **Fase previa** (septiembre-octubre):
  - Definición de requisitos
  - Estudio de las tecnologías presentes en la aplicación
  - Preparación del entorno de desarrollo
- **Primera iteración** (noviembre-enero):
  - Estudio sobre las alternativas para el protocolo de seguridad
  - Desarrollo del protocolo de seguridad
  - Integración del protocolo en la aplicación para usuarios con distintos permisos según su grupo
- **Segunda iteración** (febrero-abril):
  - Diseño del sistema de comunicación de notificaciones
  - Ampliación de la API del servidor para gestionar las notificaciones (cambios e incidencias)
  - Inclusión de la gestión de notificaciones en el cliente web administrador
  - Modificación del cliente móvil para añadir la notificación de cambios e incidencias para usuarios registrados y anónimos
  - Ampliación del cliente móvil para soportar el sistema de comunicación de cambios para usuarios registrados
- **Fase final** (mayo-junio):
  - Reunión con el cliente para valorar la aplicación completa
  - Documentación de las mejoras realizadas
  - Redacción de la memoria del proyecto

## 5.2. Comunicación con el cliente

El proyecto se inicia a partir de la primera reunión con el cliente (UTC) en septiembre de 2016. En ese momento se ponen en común ideas desde las dos partes acerca de posibles mejoras en las que trabajar. De esta reunión se desprenden varias necesidades, entre las cuales destaca la comunicación de las modificaciones en los espacios.

Bajo el criterio de satisfacer las necesidades del cliente se decide enfocar el proyecto a dos mejoras principales para la aplicación, un nuevo sistema de comunicación de notificaciones y un protocolo de seguridad que permita un uso seguro de la aplicación por parte de usuarios con diferentes permisos. Además, se decide incluir en el proyecto la posibilidad de visualizar imágenes panorámicas de los espacios, con el objetivo de mostrar una vista más intuitiva del aspecto de los mismos.

En cuanto al sistema de comunicación de notificaciones se prepara un prototipo que permite al cliente comprobar el funcionamiento del sistema completo, tanto la parte de administrador que gestiona todas las notificaciones como la del usuario que las envía. De esta manera, se recibe una valoración positiva del cliente que propone algunos cambios, como añadir algún tipo de aviso visual que permita al usuario detectar cuándo el administrador ha respondido a cualquiera de sus cambios notificados, sin importar en qué parte de la aplicación se encuentre éste.

Por otro lado, en una reunión posterior se decide revisar la vista principal de la aplicación para adaptarse a las distintas dimensiones de pantalla soportadas.

Además, se propone la posibilidad de restringir la subida de imágenes por parte de usuarios anónimos (al notificar incidencias) para evitar un uso inadecuado de dicho servicio. Se toma la decisión de posponer esta restricción a falta de estudiar otras posibilidades antes de ofrecer la aplicación al público.

Por último, se muestra la aplicación completa al cliente, que valora positivamente el sistema de comunicación de notificaciones implementado, así como las mejoras añadidas a la aplicación. Se discute la posibilidad de revisar la interfaz general de la aplicación para mejorar la usabilidad, pero se decide mantener esta línea de trabajo fuera del proyecto actual debido a la dimensión que ya alcanza el mismo.

### 5.3. Control de esfuerzos

Desde el inicio del proyecto se ha guardado registro del tiempo dedicado a cada parte del desarrollo. A continuación se detalla la distribución de horas en función de la parte de la aplicación realizada.

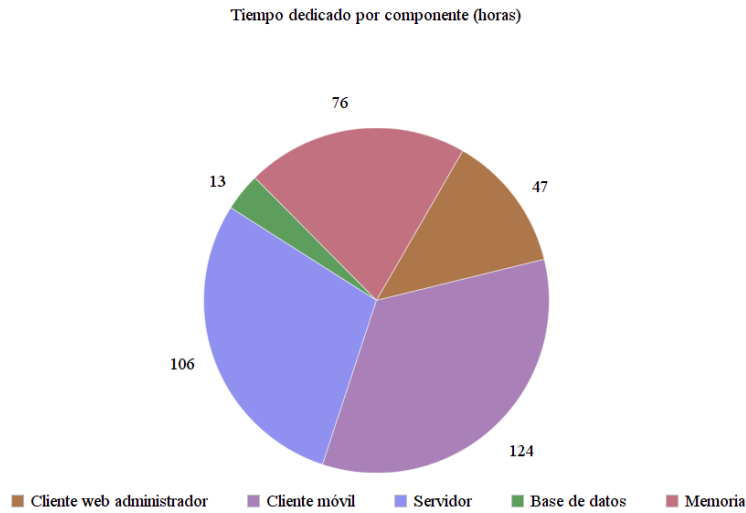


Figura 10: Distribución de horas por componente desarrollado.

Este gráfico detalla la distribución de tiempo en función del componente desarrollado. Destaca el cliente móvil como el componente en el que más tiempo se ha invertido. Esto es debido a que se ha rediseñado parte de la interfaz trabajando con tecnología nueva y se ha necesitado más tiempo para obtener el conocimiento necesario.

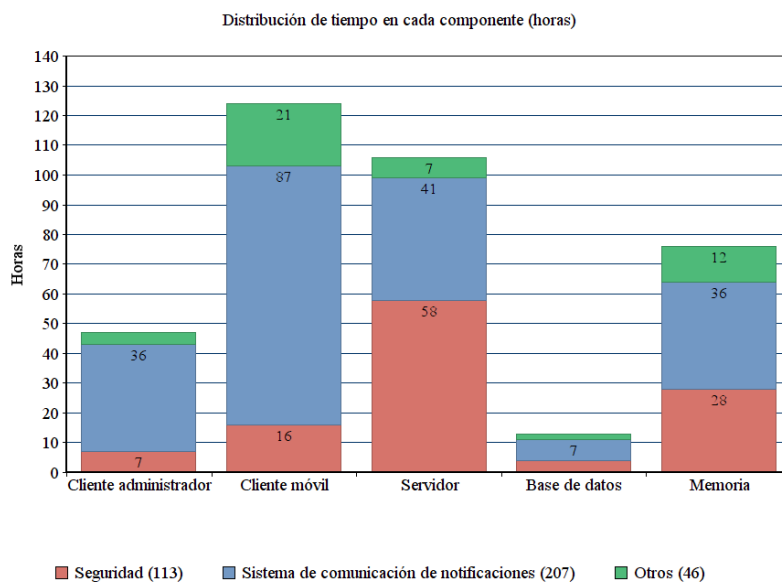


Figura 11: Distribución de horas por componente y actividad desarrollados.

Se observa como las tareas relacionadas con la implementación del sistema de comunicación de notificaciones han ocupado la mayor parte del tiempo, especialmente en el cliente móvil, aunque han representado una parte muy importante del desarrollo a lo largo de todo el proyecto.

Por otro lado, las tareas relacionadas con la seguridad han tenido más relevancia en el servidor, ya que es el encargado de filtrar todas las peticiones y donde la configuración cobra una especial importancia.

El proyecto completo ha tenido una duración de 366 horas de trabajo. La mayor parte del tiempo ha sido invertido en el sistema de comunicación de notificaciones, la necesidad principal del cliente. También ha sido relevante el tiempo invertido en la implementación de un protocolo de seguridad para todo el sistema. Por último, se ha dedicado el tiempo necesario a la memoria para poder documentar todo el proceso con suficiente detalle.

## 5.4. Gestión de configuraciones

A lo largo del proyecto se ha mantenido un control de versiones del código desarrollado, así como de la documentación relacionada con el diseño del protocolo de seguridad o con el sistema de comunicación de notificaciones, para lo cual se ha hecho uso de *Dropbox*.

Se ha utilizado la herramienta *GitHub* para mantener el control de versiones de todo el código de la aplicación, esto es, ambos clientes y el servidor están incluidos en el mismo repositorio [1].

Al inicio del proyecto y dado que durante la realización del mismo otro compañero ha estado desarrollando en paralelo sobre la misma aplicación, se decidió realizar un *fork* [26] del repositorio principal de la aplicación para obtener una copia del código sobre la que poder trabajar sin generar conflictos. Una vez implementadas las mejoras se ha realizado un *pull request* [27] al repositorio original para incluirlas. Debido a una estructura modular del código, las tareas a realizar en cada proyecto afectan a componentes diferentes, de manera que no se han generado conflictos al incluir el código correspondiente.

## 5.5. Análisis de riesgos

A continuación se presenta una tabla de análisis de riesgos definida al inicio del proyecto. Destaca el riesgo relacionado con el desconocimiento de la tecnología ya que se trata de un trabajo de ampliación de una aplicación ya existente, y por lo tanto es necesario aprender la tecnología antes de poder implementar cualquier adición.

<b>Riesgo</b>	<b>Impacto</b>	<b>Plan de mitigación</b>
Pérdida de datos de código o documentación.	Alto	Utilizar herramientas online como GitHub o Dropbox para mantener accesibles los datos.
Falta de respuesta del cliente.	Medio	Crear una propuesta y definir una fecha de reunión con el cliente.
Desconocimiento de la tecnología a utilizar y del sistema a ampliar.	Alto	Dedicar más tiempo al estudio de la tecnología y el sistema al inicio del proyecto.
No llegar a tiempo a una entrega.	Medio	Entregar en la siguiente fecha disponible.

Tabla 3: Análisis de riesgos.

## 6. Conclusiones

### 6.1. Resumen de aportaciones

A continuación se resumen las contribuciones de este proyecto al sistema objetivo de ampliación:

- **Seguridad:** se ha incluido un protocolo de seguridad que soporta la comunicación de ambos clientes con el servidor de manera segura. Se han creado permisos para los distintos tipos de usuarios que acceden a cada aplicación.
- **Sistema de comunicación de notificaciones:** se ha definido un sistema de comunicación de notificaciones que facilita la gestión de los espacios por parte del administrador. Esto supone una mejora importante, porque actualmente no se cuenta con un método de comunicación que permita mantener actualizada la información de los espacios de la universidad.

Este sistema también soporta el uso por parte de usuario anónimos para que notifiquen eventos inesperados directamente al administrador, de manera que éste pueda reaccionar rápidamente al saber de qué se trata y dónde se ha producido.

- **Visualización de imágenes panorámicas en 360°:** se ha incluido la posibilidad de visualizar imágenes panorámicas en 360° de los espacios que dispongan de ellas. Aunque actualmente no se cuenta con imágenes de todos los espacios, esta funcionalidad abre la puerta a futuras mejoras que permitan mayor interacción y conocimiento de los espacios por parte del usuario.

### 6.2. Trabajo futuro

Una de las principales conclusiones que se ha obtenido al terminar el proyecto y hablar con el cliente, es que será necesario trabajar en la interfaz general de la aplicación para hacerla más atractiva al usuario.

Por otro lado, una de las ideas que se ha planteado al inicio del proyecto se decidió no llevar a cabo es la posibilidad de generar rutas, ya sea estáticas o dinámicas, que guíen al usuario de un espacio a otro de la universidad.

Durante el desarrollo del proyecto y en relación con la seguridad de la aplicación, se ha planteado la posibilidad de integrar las credenciales de profesores y alumnos de la universidad como usuarios de la misma con la aplicación. De esta manera, se soluciona el problema que puede surgir al permitir que personas anónimas envíen imágenes a la aplicación sin ningún tipo de filtro, ya que los alumnos tendrían que identificarse para hacer uso de esta funcionalidad.



### **6.3. Valoración personal**

Durante el transcurso del proyecto se ha puesto de manifiesto la importancia y el valor que tiene el trabajo en equipo. Aunque el desarrollo ha sido guiado por un profesor y se han producido reuniones con el cliente, el trabajo solitario sobre un sistema ya construido y con varias aplicaciones diferentes puede llegar a ser muy costoso y complejo. Al comparar el trabajo realizado con cualquiera de los proyectos llevados a cabo durante el grado de Ingeniería Informática destaca la gran diferencia de complejidad y dedicación.

El trabajo diario con compañeros de equipo, aunque cada uno se encargue de una parte del desarrollo es mucho más fluido y suaviza muchos momentos de bloqueo en el diseño e implementación de determinadas partes, lo que favorece el correcto desarrollo y corrige muchos errores antes de que influyan en otros momentos del proceso.

En cuanto a la tecnología utilizada, se valora positivamente haber aprendido a utilizar múltiples nuevas tecnologías desconocidas antes del inicio del desarrollo de este proyecto.

Por último, la comunicación con el cliente ha resultado una experiencia gratificante y muy útil ya que ha permitido mostrar los avances en la aplicación y obtener reacciones a los mismos para mejorar el sistema. Ha resultado especialmente gratificante en la medida en la que ha sido un cliente con necesidades reales el que ha valorado positivamente el trabajo realizado.

## Referencias

- [1] Repositorio de la aplicación. <https://github.com/IAAA-Lab/UNIZAR-BuildingsApp>.
- [2] AngularJS. <https://angularjs.org/>.
- [3] Node JS. <https://nodejs.org/es/>.
- [4] Gradle. <https://gradle.org/>.
- [5] jQuery Javascript library. <https://jquery.com/>.
- [6] DataTables jQuery plug-in. <https://datatables.net/>.
- [7] Ionic framework. <https://ionicframework.com/>.
- [8] Photo Sphere Viewer Javascript library. <http://photo-sphere-viewer.js.org/>.
- [9] Java. <https://www.java.com/es/>.
- [10] Spring framework. <https://spring.io/>.
- [11] Spring Security. <https://projects.spring.io/spring-security/>.
- [12] Spring Boot. <https://projects.spring.io/spring-boot/>.
- [13] PostgreSQL. <https://www.postgresql.org/>.
- [14] Json Web Token (JWT). <https://jwt.io/introduction/>.
- [15] Batería de test en Postman.  
<https://documenter.getpostman.com/view/497974/uzapp-tests/6Z6qA9D>.
- [16] Script de despliegue.  
<https://github.com/IAAA-Lab/UNIZAR-BuildingsApp/blob/master/scripts/deploy.sh>.
- [17] Atom. <https://atom.io/>.
- [18] XAMPP. <https://www.apachefriends.org/es/index.html>.
- [19] Google Chrome. <https://www.google.es/chrome/browser/desktop/index.html>.
- [20] Spring Tool Suite. <https://spring.io/tools/sts/all>.
- [21] pgAdmin. <https://www.pgadmin.org/>.
- [22] Cisco AnyConnect Security Mobile Client. <https://tinyurl.com/yau2uwqt>.
- [23] Postman. <https://www.getpostman.com/>.
- [24] Overleaf. <https://www.overleaf.com/>.
- [25] GitHub. <https://github.com/>.
- [26] Operación *fork* (GitHub). <https://help.github.com/articles/fork-a-repo/>.

- [27] Operación *pull request* (GitHub).  
<https://help.github.com/articles/creating-a-pull-request/>.
- [28] Dropbox. <https://www.dropbox.com/>.
- [29] Unizar. <https://www.unizar.es/>.
- [30] Jorge Garuz Sánchez. Aplicación móvil para la consulta y geolocalización de los espacios de los edificios de la Universidad de Zaragoza a partir de los datos de la Unidad Técnica de Construcciones y Energía. <https://zagan.unizar.es/record/47896/files/TAZ-TFG-2015-3244.pdf>.

# Anexos

## A. API del servidor (nuevas operaciones)

A continuación se detallan todas las operaciones que ofrece la API del servidor ampliada en este proyecto. En primer lugar se documentan las nuevas operaciones resultantes del trabajo realizado. La *url* base sobre la que se ofrece la API es: *http://155.210.14.31:8080*.

### A.1. Notificaciones

#### ■ Creación de un cambio

*Endpoint:* /notificacion/cambio

*Método:* POST

*Parámetros:* Notificación

*Permisos:* Usuario registrado

El parámetro Notificación se corresponde con la entidad *Notificacion* definida en el modelo de datos y contiene los siguientes atributos:

- *id.espacio:* es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *descripcion:* es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *id.usuario:* no es obligatorio. Puede tener cualquier valor.
- *estado:* no es obligatorio. Puede tener cualquier valor.
- *foto:* no es obligatorio. Puede tener cualquier valor.
- *email.usuario:* es obligatorio. Puede tener cualquier valor.
- *comentario\_admin:* es obligatorio. Puede tener cualquier valor.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con el identificador del cambio creado. Ej: 347.

#### ■ Actualización de un cambio

*Endpoint:* /notificacion/cambio/id\_cambio

*Método:* PUT

*Parámetros:* id\_cambio, Notificación

*Permisos:* Usuario registrado

El parámetro `id_cambio` se corresponde con un número natural que identifica de manera única al cambio. El parámetro `Notificación` se corresponde con la entidad *Notificación* definida en el modelo de datos y contiene los siguientes atributos:

- `id_espacio`: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- `descripcion`: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- `id_usuario`: no es obligatorio. Puede tener cualquier valor.
- `estado`: no es obligatorio. Puede tener cualquier valor.
- `foto`: no es obligatorio. Puede tener cualquier valor.
- `email_usuario`: es obligatorio. Puede tener cualquier valor.
- `comentario_admin`: es obligatorio. Puede tener cualquier valor.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con el identificador del cambio actualizado. Ej: 347.

#### ■ Obtención de todos los cambios

*Endpoint*: `/notificacion/cambio`

*Método*: GET

*Parámetros*: Ninguno

*Permisos*: Usuario administrador

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con la lista de cambios solicitados en formato *JSON*:

```
[
  {
    "id_notificacion": 347,
    "tipo_notificacion": 1,
    ...
    "id_usuario": "usuario"
  },
  {
    "id_notificacion": 348,
    "tipo_notificacion": 1,
    ...
    "id_usuario": "usuario2"
  },
  ...
]
```

## ■ Obtención de un cambio

*Endpoint:* /notificacion/cambio/id\_cambio

*Método:* GET

*Parámetros:* id\_cambio

*Permisos:* Usuario registrado

El parámetro id\_cambio se corresponde con un número natural que identifica de manera única al cambio.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con el cambio solicitado en formato *JSON*:

```
{
  "id_notificacion": 347,
  "tipo_notificacion": 1,
  "id_espacio": "espacio",
  "descripcion": "descripcion",
  "fecha": {
    "date": {
      "year": 2017,
      "month": 6,
      "day": 18
    },
    "time": {
      "hour": 18,
      "minute": 5,
      "second": 31,
      "nano": 322000000
    }
  },
  "fechaUltimaModificacion": {
    "date": {
      "year": 2017,
      "month": 6,
      "day": 18
    },
    "time": {
      "hour": 18,
      "minute": 27,
      "second": 44,
      "nano": 985000000
    }
  },
  "id_usuario": "usuario"
}
```

## ■ Obtención de todos los cambios de un usuario

*Endpoint:* /notificacion/cambio/user

*Método:* GET

*Parámetros:* Ninguno

*Permisos:* Usuario registrado

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con la lista de cambios solicitados en formato *JSON*:

```
[
  {
    "id_notificacion": 348,
    "tipo_notificacion": 1,
    ...
    "id_usuario": "usuario2"
  },
  {
    "id_notificacion": 349,
    "tipo_notificacion": 1,
    ...
    "id_usuario": "usuario2"
  },
  ...
]
```

## ■ Eliminación de un cambio

*Endpoint:* /notificacion/cambio/id\_cambio

*Método:* DELETE

*Parámetros:* id\_cambio

*Permisos:* Usuario administrador

El parámetro id\_cambio se corresponde con un número natural que identifica de manera única al cambio.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito.

## ■ Creación de una incidencia

*Endpoint:* /notificacion/incidencia

*Método:* POST

*Parámetros:* Notificación

*Permisos:* Usuario registrado

El parámetro Notificación se corresponde con la entidad *Notificacion* definida en el modelo de datos y contiene los siguientes atributos:

- *id\_espacio*: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *descripcion*: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *id\_usuario*: no es obligatorio. Puede tener cualquier valor.
- *estado*: no es obligatorio. Puede tener cualquier valor.
- *foto*: no es obligatorio. Puede tener cualquier valor.
- *email\_usuario*: es obligatorio. Puede tener cualquier valor.
- *comentario\_admin*: es obligatorio. Puede tener cualquier valor.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con el identificador de la incidencia creada. Ej: 347.

## ■ Actualización de una incidencia

*Endpoint:* /notificacion/incidencia/id\_incidencia

*Método:* PUT

*Parámetros:* id\_incidencia, Notificación

*Permisos:* Usuario administrador

El parámetro *id\_incidencia* se corresponde con un número natural que identifica de manera única a la incidencia. El parámetro Notificación se corresponde con la entidad *Notificacion* definida en el modelo de datos y contiene los siguientes atributos:

- *id\_espacio*: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *descripcion*: es obligatorio. Puede tener cualquier valor, excepto *null* o *undefined*.
- *id\_usuario*: no es obligatorio. Puede tener cualquier valor.
- *estado*: no es obligatorio. Puede tener cualquier valor.
- *foto*: no es obligatorio. Puede tener cualquier valor.
- *email\_usuario*: es obligatorio. Puede tener cualquier valor.
- *comentario\_admin*: es obligatorio. Puede tener cualquier valor.



Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con el identificador de la incidencia creada. Ej: 347.

- **Obtención de todas las incidencias**

*Endpoint:* /notificacion/incidencia

*Método:* GET

*Parámetros:* Ninguno

*Permisos:* Usuario administrador

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto con la lista de incidencias solicitadas en formato *JSON*:

```
[
  {
    "id_notificacion": 347,
    "tipo_notificacion": 2,
    ...
    "id_usuario": "usuario"
  },
  {
    "id_notificacion": 348,
    "tipo_notificacion": 2,
    ...
    "id_usuario": "anonimo"
  },
  ...
]
```

- **Eliminación de una incidencia**

*Endpoint:* /notificacion/incidencia/id\_incidencia

*Método:* DELETE

*Parámetros:* id\_incidencia

*Permisos:* Usuario administrador

El parámetro id\_incidencia se corresponde con un número natural que identifica de manera única a la incidencia.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito.

## ■ Subida de una imagen

*Endpoint:* /notificacion/photo

*Método:* POST

*Parámetros:* name, file, id\_notificacion

*Permisos:* Usuario anónimo

El parámetro *name* es una cadena que contiene el nombre de la imagen. El parámetro Notificación se corresponde con la entidad *Notificacion* definida en el modelo de datos y contiene los siguientes atributos:

- *name*: es obligatorio. Es una cadena que contiene el nombre de la imagen.
- *file*: es obligatorio. Es un objeto de tipo *MultipartFile* donde se encuentra el contenido de la imagen.
- *id\_notificacion*: se corresponde con un número natural que identifica de manera única al cambio.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito.

## ■ Descarga de una imagen

*Endpoint:* /notificacion/imagen/id\_imagen

*Método:* GET

*Parámetros:* id\_imagen

*Permisos:* Usuario anónimo

El parámetro *id\_imagen* es una cadena y corresponde con el nombre de la imagen.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito junto a la imagen solicitada en formato de cadena de bytes codificada en base 64.

En este proyecto se ha añadido una operación a este recurso que se describe a continuación:

## A.2. Usuarios

- Obtención de la información del usuario

*Endpoint:* /users/info

*Método:* GET

*Parámetros:* Ninguno

*Headers:* Authorization: 'Bearer token'

*Permisos:* Usuario registrado

El contenido del *header* Authorization es una cadena que representa el *JWT* que contiene la identidad y permisos del usuario.

Devuelve un código 500 si se ha producido algún error, en caso contrario devuelve un código 200 junto a la información del usuario :

```
{
  "id": 40,
  "username": "username",
  "password": null,
  "email": "email",
  "name": "name",
  "surnames": "surnames",
  "birthDate": "aaaa-mm-dd",
  "role": "ADMIN"
}
```

### A.3. Comprobaciones del token

- **Comprobación de la validez del token**

*Endpoint:* /checkToken

*Método:* POST

*Parámetros:* Ninguno

*Headers:* Authorization: 'Bearer token'

*Permisos:* Usuario anónimo

El contenido del *header* Authorization es una cadena que representa el *JWT* que contiene la identidad y permisos del usuario.

Devuelve un código 400 si no se ha proporcionado *token*, 401 si el *token* no es válido y 200 si es válido.

- **Comprobación del permiso de administrador**

*Endpoint:* /checkAdmin

*Método:* POST

*Parámetros:* Ninguno

*Headers:* Authorization: 'Bearer token'

*Permisos:* Usuario anónimo

El contenido del *header* Authorization es una cadena que representa el *JWT* que contiene la identidad y permisos del usuario.

Devuelve un código 400 si no se ha proporcionado *token*, 401 si el *token* no es válido y 200 si es válido y el usuario tiene permisos de administrador.

### A.4. Correo electrónico

- **Envío de un correo electrónico**

*Endpoint:* /mail

*Método:* POST

*Parámetros:* Ninguno

*Headers:* Authorization: 'Bearer token'

*Permisos:* Usuario administrador

El contenido del *header* Authorization es una cadena que representa el *JWT* que contiene la identidad y permisos del usuario.

Devuelve un código 500 si se produce algún error y 200 si ha habido éxito.

## B. API del servidor (operaciones ya existentes)

En este anexo se incluyen las operaciones de la API del servidor que se han definido en otros proyectos.

### B.1. Puntos de interés

Las operaciones definidas sobre los puntos de interés han sido desarrolladas por otro proyecto, que en el momento de la redacción de este documento no ha concluido. A continuación se resumen las operaciones disponibles:

- `/pois/` (GET): devuelve una lista con todos los POIs
- `/pois/id` (GET): devuelve la información de un POI con identificador igual al parámetro *id*
- `/pois/building/floor` (GET): devuelve todos los POIs de una planta
- `/pois/` (POST): crea un POI
- `/pois/` (PUT): actualiza un POI
- `/pois/id` (DELETE): elimina un POI
- `/pois/request/pending` (GET): devuelve los POIs pendientes
- `/pois/request` (POST): crea una petición para modificar o eliminar un POI
- `/pois/request/requestId/verb/action` (POST): aprueba o rechaza una petición para editar o eliminar un POI

### B.2. Fotos

Las operaciones definidas sobre las fotos han sido desarrolladas por otro proyecto, que en el momento de la redacción de este documento no ha concluido. A continuación se resumen las operaciones disponibles:

- `/photos/` (GET): devuelve todas las fotos
- `/photos/approved/roomID` (GET): devuelve todas las fotos de un espacio
- `/photos/upload` (POST): sube una foto al servidor
- `/photos/` (PUT): actualiza una foto
- `/photos/id` (DELETE): elimina una foto
- `/photos/insert` (POST): inserta información asociada a una foto (no el contenido de la imagen)

### **B.3. Usuarios**

Las operaciones definidas sobre los usuarios han sido desarrolladas por otro proyecto, que en el momento de la redacción de este documento no ha concluido. A continuación se resumen las operaciones disponibles:

- `/users/create` (POST): crea un usuario
- `/users/edit` (PUT): modifica un usuario
- `/users/login` (POST): inicia la sesión de un usuario

### **B.4. Búsquedas y Estancias**

Las operaciones definidas sobre las búsquedas y estancias han sido desarrolladas por otro proyecto y están definidas en la memoria del mismo, en el anexo 1 (Diseño detallado [30]).

## C. Manual de usuario

En este anexo se ofrece una guía de uso para ambos clientes que incluye una explicación de las nuevas operaciones.

### C.1. Cliente web administrador

#### C.1.1. Pantalla de inicio de sesión

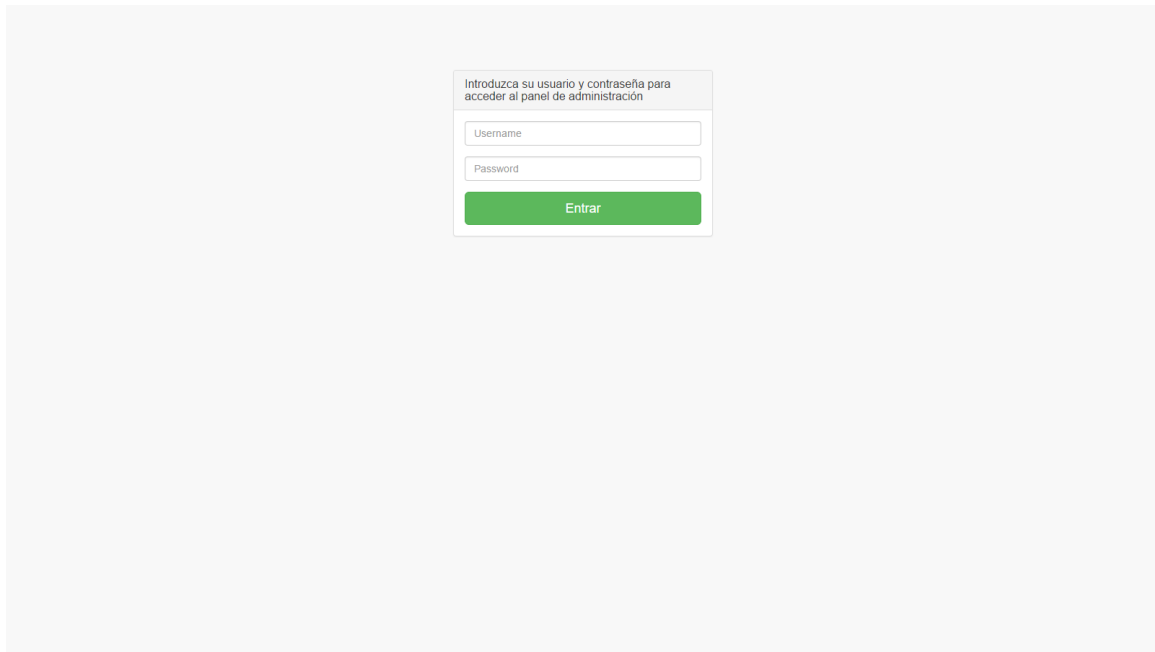


Figura 12: Pantalla de inicio de sesión.

Esta pantalla es la primera que ve el usuario al acceder a la aplicación web de administrador. Si se introducen las credenciales adecuadas, nombre de usuario y contraseña, de un usuario con permisos de administrador y se presiona sobre el botón *Entrar* la aplicación continuará a la pantalla principal. En caso de proporcionar unas credenciales erróneas aparecerá un mensaje de error y la aplicación se mantendrá en esta pantalla.

## C.1.2. Pantalla principal

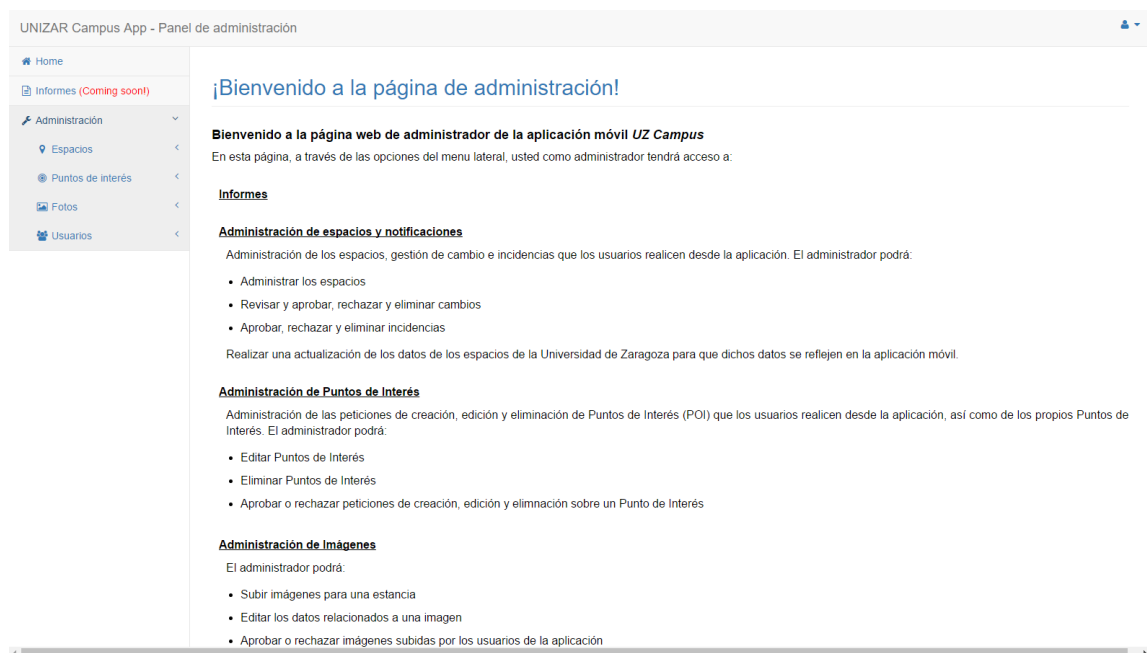


Figura 13: Pantalla principal de la aplicación web de administrador.

Esta pantalla muestra un resumen de todas las acciones que puede realizar el usuario administrador dentro de la propia aplicación. En la parte superior izquierda se observa el nombre de la aplicación justo encima de un menú lateral desplegable. Este menú, situado permanentemente en la izquierda de la pantalla permite acceder a los diferentes paneles de gestión de cada elemento (espacios, pois, fotos y usuarios).

Por otro lado, en la parte superior derecha se puede observar un icono de usuario que permite acceder al perfil del administrador, al menú de creación de nuevos usuarios o salir de la sesión.



### C.1.3. Pantalla de gestión de cambios

UNIZAR Campus App - Panel de administración

Home

Informes (Coming soon)

Administración

- Espacios
- Puntos de interés
- Fotos
- Usuarios

#### Administración de cambios en los espacios

En la tabla inferior se muestran los datos de todos los cambios realizados en los espacios. Se podrán llevar a cabo las siguientes acciones seleccionando una fila de la tabla:

- Revisar los datos relativos a un cambio
- Aprobar/rechazar un cambio pendiente
- Solicitar la revisión de un cambio
- Añadir una aclaración acerca de la decisión tomada sobre un cambio
- Eliminar un cambio

Show 10 rows Refresh Revisar Eliminar

Buscar:

ID	Estado	Descripción	Fecha de creación	Última modificación	Ciudad	Campus	Edificio	Planta	Espacio	Espacio ID	Dirección
142	Pendiente	Esto es una breve descripción de un cambio.	2017-06-19 13:17:11	2017-06-19 13:17:11	Zaragoza	Campus Río Ebro	ADA BYRON	02	SEMINARIO_A24	CRE.1200.02.070	C/Maria de Luna, s/n
143	Pendiente	Esto es un cambio importante.	2017-06-19 13:17:30	2017-06-19 13:17:30	Zaragoza	Campus Río Ebro	ADA BYRON	02	SEMINARIO_A22	CRE.1200.02.040	C/Maria de Luna, s/n

Showing 1 to 2 of 2 entries

Previous 1 Next

Figura 14: Pantalla de gestión de todos los cambios registrados en la aplicación.

Esta pantalla es accesible a través del menú lateral izquierdo mediante la opción *Espacios/Listado de cambios*. Se observa un listado de todos los cambios de la aplicación en formato de tabla con toda la información asociada a ellos. Estos cambios se muestran paginados y se pueden filtrar por valor de cualquiera de sus atributos incluyendo la cadena deseada en la caja de búsqueda situada en la parte derecha encima de la tabla.

En la parte superior aparece una breve descripción de las acciones disponibles para el administrador a realizar sobre un cambio de la tabla. Para revisar un cambio, hay que seleccionar su fila en la tabla y presionar el botón *Revisar* de la barra de acciones situada encima de la tabla. Esto abre una ventana modal cuyo contenido se explica más adelante. En el caso de pulsar sobre el botón *Eliminar* se accede a otra ventana modal que se detalla más adelante.

### C.1.4. Pantalla de revisión de un cambio

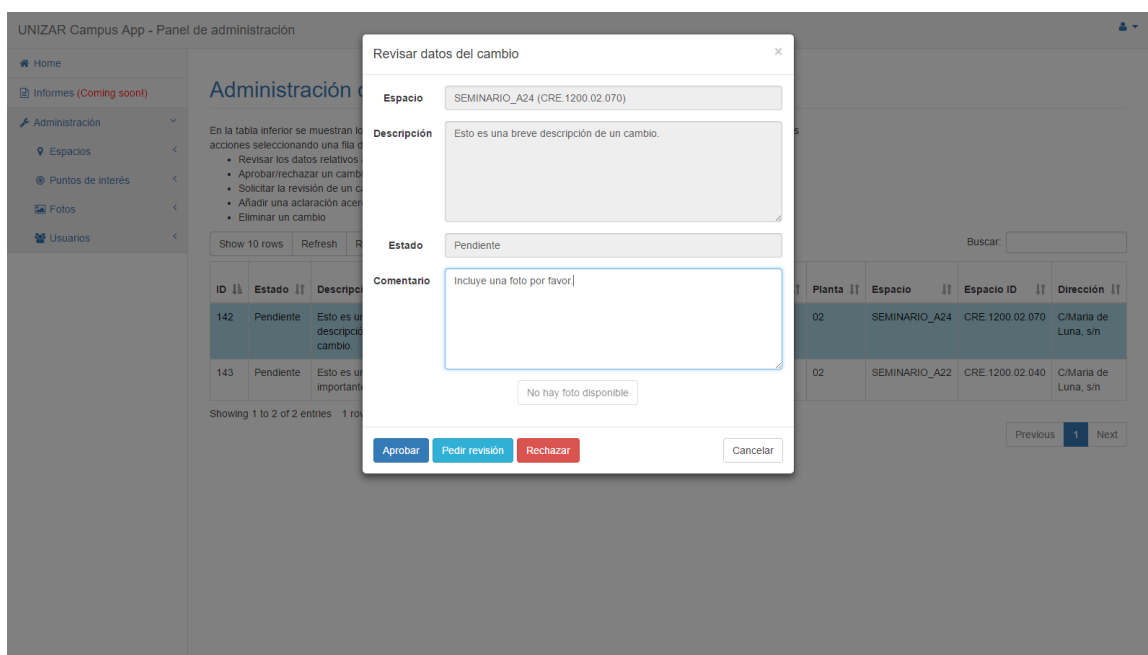


Figura 15: Pantalla de revisión de un cambio.

Esta pantalla ofrece la posibilidad de revisar un cambio registrado por un usuario de la aplicación. Se muestra el identificador del espacio donde ha tenido lugar el cambio, la descripción de la modificación realizada, el estado actual del cambio y una opción editable que permite al administrador redactar un mensaje a modo de explicación sobre la decisión que va a tomar acerca del cambio. Adicionalmente, se observa un botón desactivado con el texto 'No hay imagen disponible'. Este botón se activa cuando el cambio tiene una imagen asociada (proporcionada por el usuario) y modifica su texto a 'Ver foto', el cual permite visualizar la imagen en una nueva modal como se ve en la siguiente imagen.

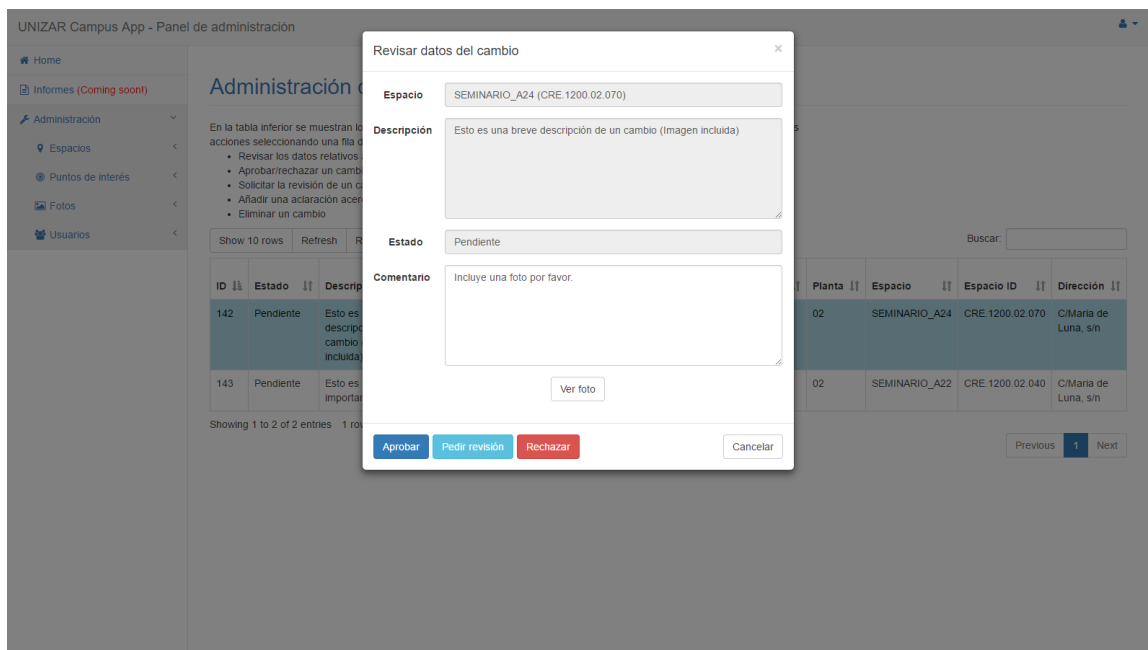


Figura 16: Pantalla de revisión de un cambio con imagen asociada.

Las decisiones disponibles sobre el cambio están representadas en los tres botones (*Aprobar*, *Pedir revisión* y *Rechazar*) que aparecen en la parte inferior izquierda de la ventana modal. Todas las opciones adjuntan el mensaje escrito por el administrador en el espacio con la etiqueta *Comentario*.

- *Aprobar*: cambia el estado del cambio a 'Aprobado'. Finaliza la comunicación con el cliente.
- *Pedir revisión*: cambia el estado del cambio a 'Pendiente del usuario'. En este momento el cambio no es revisable por el administrador, hasta que el usuario vuelva a responder.
- *Rechazar*: cambia el estado del cambio a 'Rechazado'. Finaliza la comunicación con el cliente.

Cabe destacar que ninguna de las tres acciones permitidas requiere de confirmación posterior. Una vez se pulsa el botón correspondiente se lleva a cabo la operación, la ventana se cierra y se muestra un mensaje con el resultado de la dicha operación.

Situado en la parte inferior derecha de la ventana modal se encuentra el botón de cancelar, que cierra la modal sin realizar ninguna modificación sobre el cambio.

### C.1.5. Pantalla de revisión de la imagen de un cambio

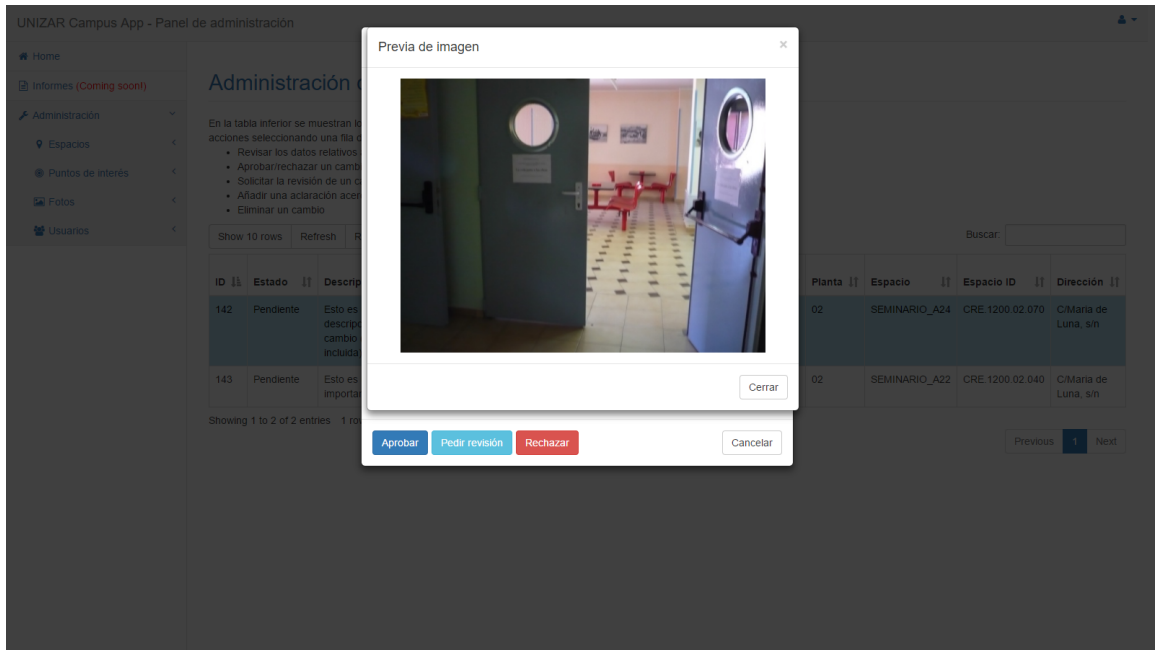


Figura 17: Pantalla de revisión de la imagen de un cambio.

Esta pantalla muestra la imagen asociada a un cambio al presionar el botón 'Ver foto' de un cambio con imagen. En la parte inferior derecha de la ventana modal que contiene la imagen se sitúa el botón *Cerrar* que permite ocultar esta modal y volver a la de revisión del cambio.

### C.1.6. Pantalla de eliminación de un cambio

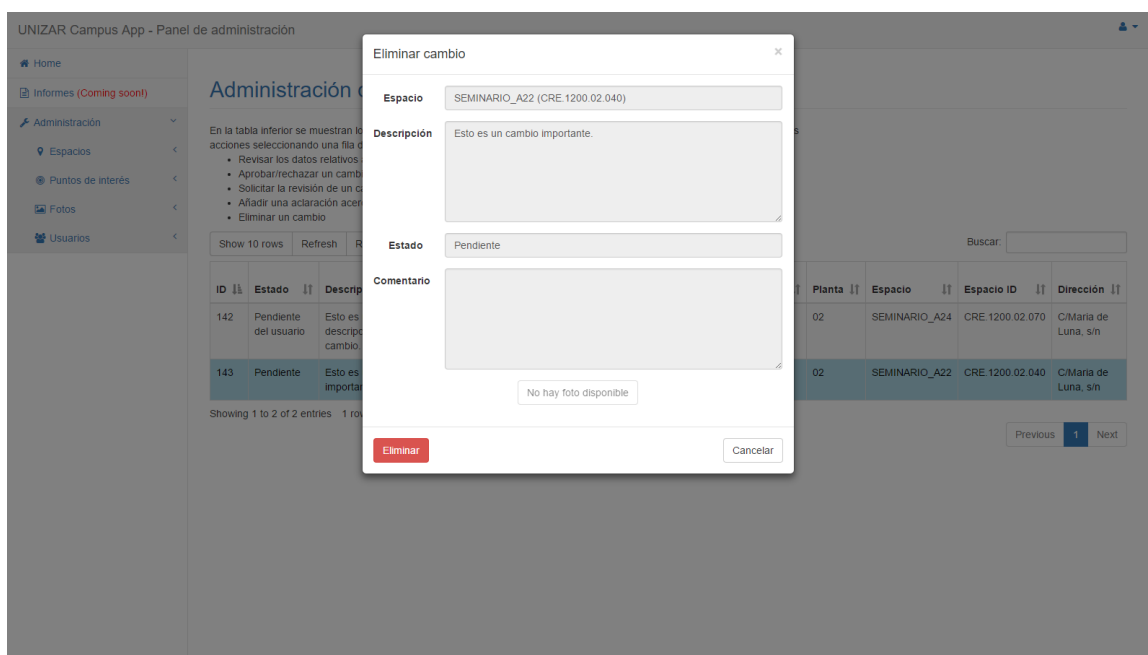


Figura 18: Pantalla de eliminación de un cambio.

Esta pantalla ofrece una ventana modal que muestra un resumen de la información asociada al cambio similar al que contiene la pantalla de revisión. En la parte inferior izquierda de la ventana se observa el botón *Eliminar* que permite borrar un cambio y su imagen asociada si la tiene. También consta en la parte inferior derecha de un botón *Cancelar* que permite cerrar la modal sin eliminar el cambio actual.

Cabe destacar que la acción de eliminar no espera confirmación del administrador. Una vez el botón de *Eliminar* de color rojo que aparece en esta pantalla, el cambio es eliminado. La ventana se cierra y se muestra un mensaje con el resultado de la operación.

### C.1.7. Pantalla de revisión de una incidencia

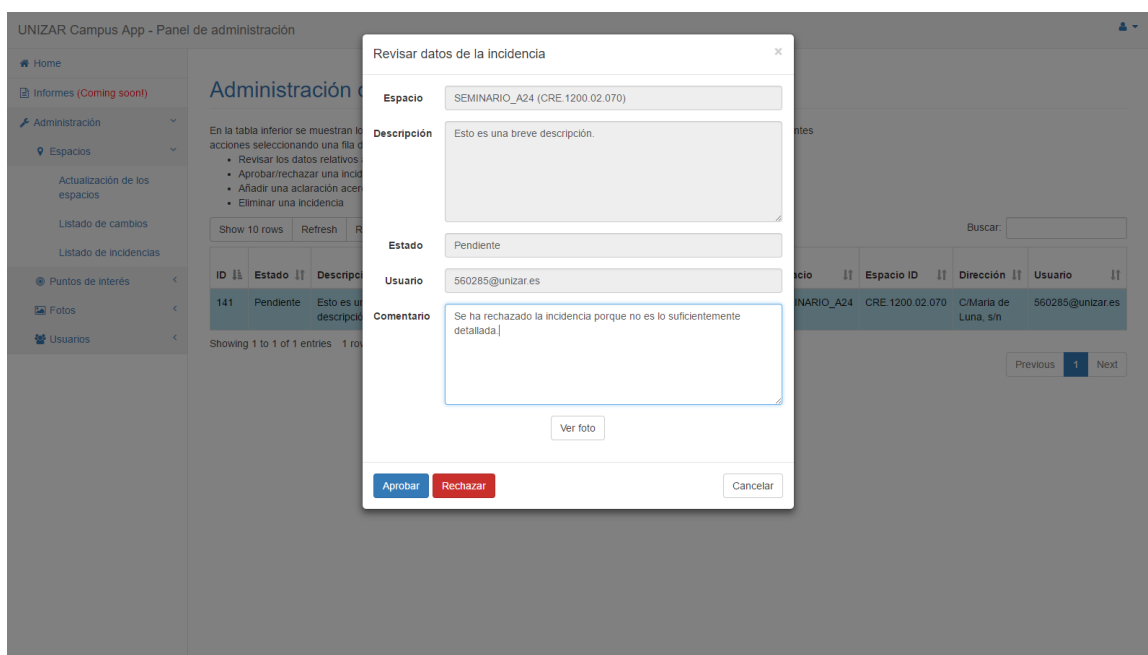


Figura 19: Pantalla de revisión de una incidencia.

De manera similar a la pantalla de revisión de un cambio representada en la figura 15, se muestra la información de la incidencia. La principal diferencia se observa en que en este caso se muestra el correo electrónico del usuario junto a la etiqueta *Usuario*. Esto es debido a que el contenido de la opción editable *Comentario* será enviado en un correo a la dirección proporcionada por el usuario al realizar una de las dos acciones disponibles.

- *Aprobar*: cambia el estado de la incidencia a 'Aprobada' y envía el comentario del administrador en un correo electrónico a la dirección proporcionada por el usuario (etiqueta *Usuario*), si se ha proporcionado alguna.
- *Rechazada*: cambia el estado de la incidencia a 'Rechazada' y envía el comentario del administrador en un correo electrónico a la dirección proporcionada por el usuario (etiqueta *Usuario*), si se ha proporcionado alguna.

Se observan los botones *Ver Foto* y *Cancelar* que llevan a cabo las mismas acciones que en la pantalla de revisión de cambios representada en la figura 16.

## C.1.8. Pantalla de gestión de usuarios

UNIZAR Campus App - Panel de administración

Home

Informes (Coming soon)

Administración

Espacios

Puntos de interés

Fotos

Usuarios

Listado usuarios

### Administración de los usuarios de la aplicación

En la tabla inferior se muestran los datos de todos los usuarios registrados en la aplicación. Se podrán llevar a cabo las siguientes acciones seleccionando una fila de la tabla:

- Revisar la información de un usuario
- Modificar los permisos de un usuario
- Eliminar un usuario

Show 10 rows Refresh Editar Eliminar

Buscar:

Usuario	Rol	Email	Nombre	Apellidos	Fecha de nacimiento
alejandro	ADMIN	560265@unizar.es	Alejandro	Royo	1994-10-26
daniegarcia254	ADMIN	daniegarcia254@gmail.com	Daniel	García Páez	1987-04-25
maxbolea	ADMIN	maxbolea@unizar.es	Máximo	Bolea	No disponible
rbejar	ADMIN	rbejar@unizar.es	Rubén	Bejar	No disponible
userPrueba	USER	user@user.com	user	user	2017-06-03

Showing 1 to 5 of 5 entries 1 row selected

Previous 1 Next

Figura 20: Pantalla de gestión de usuarios.

Esta pantalla muestra de manera análoga a la pantalla de gestión de cambios (figura 14) un listado de los usuarios registrados en la aplicación. En la parte superior se describen también las acciones que puede realizar el administrador con cada usuario. Se presenta toda la información asociada a cada usuario excepto su contraseña.

Para editar o eliminar un usuario, es necesario seleccionar la fila en la que se encuentra el usuario dentro de la tabla que se verá resaltada de azul tal y como se aprecia en la imagen con la primera fila. Una vez seleccionado, se pueden realizar las siguientes acciones.

- *Editar*: abre una ventana modal que permite editar información del usuario, incluyendo la creación de una nueva contraseña.
- *Eliminar*: abre una ventana modal que permite eliminar cualquier usuario registrado en la aplicación.

### C.1.9. Pantalla de edición de usuarios

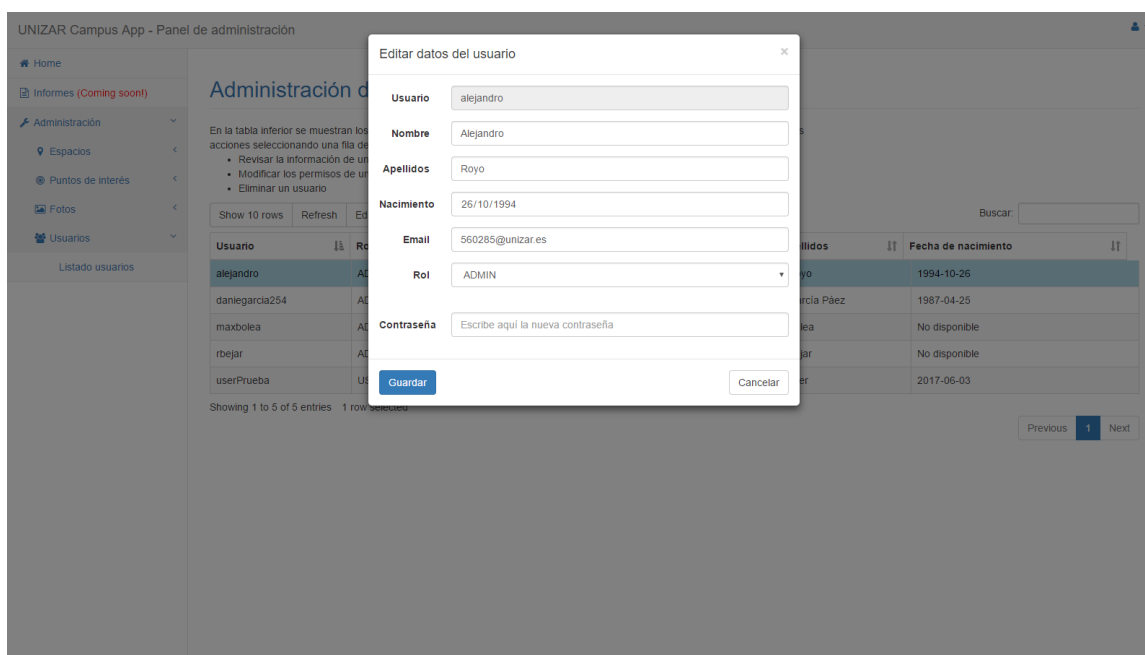


Figura 21: Pantalla de editar de usuarios.

Esta pantalla ofrece una ventana modal con todos los datos modificables de un usuario:

- *Nombre*: contiene el nombre real del usuario.
- *Apellidos*: contiene los apellidos reales del usuario.
- *Nacimiento*: contiene la fecha de nacimiento del usuario.
- *Email*: contiene el correo electrónico del usuario.
- *Rol*: contiene los permisos asignados al usuario. Los posibles valores son:
  - *ADMIN*: el usuario puede entrar en el cliente web administrador y realizar todas las operaciones que soporta la aplicación. El usuario tiene acceso a todas las operaciones correspondientes a los permisos *USER*.
  - *USER*: el usuario tiene acceso al cliente móvil y por lo tanto a la creación de incidencias y notificaciones desde el mismo. No puede acceder al cliente web administrador.
- *Contraseña*: define la nueva contraseña del usuario. Si se completa con alguna cadena, ésta será la nueva contraseña. Si se deja en blanco, la contraseña del usuario seguirá siendo la misma que antes.

En la parte inferior izquierda de la ventana modal, aparece el botón *Guardar*, que pulsado almacena todas las modificaciones realizadas sobre el formulario presentado. Después, cierra la modal y muestra un mensaje con el resultado de la operación.



### C.1.10. Pantalla de eliminación de un usuario

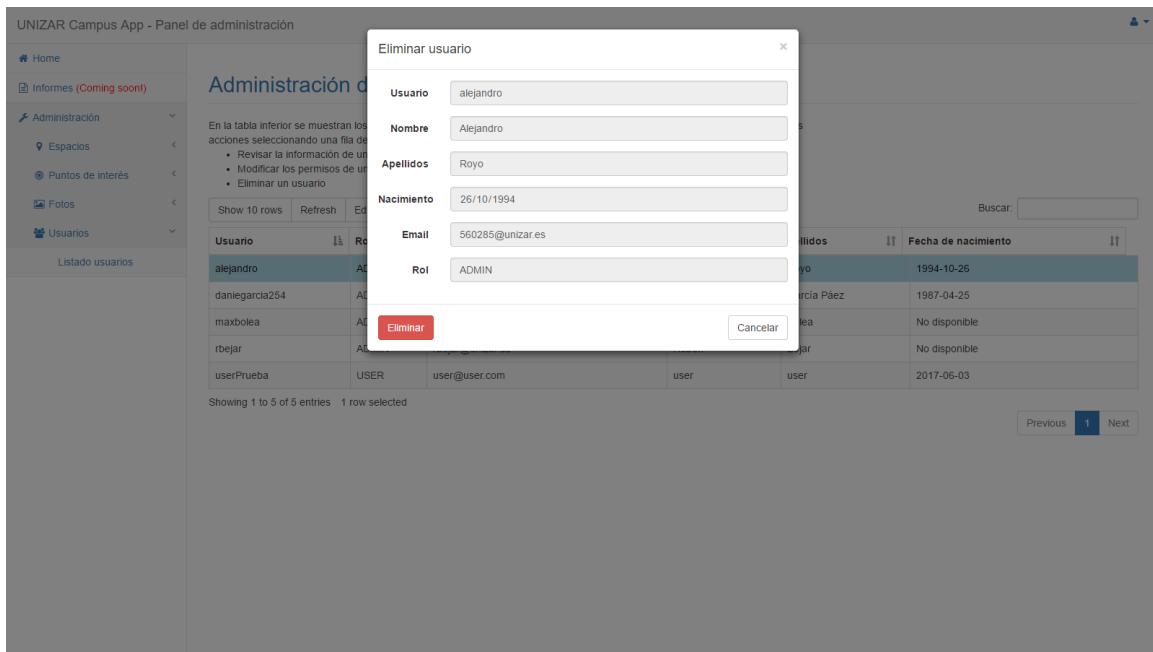


Figura 22: Pantalla de eliminación de un usuario.

Esta pantalla presenta una modal con la información asociada a un usuario de forma similar a lo visto en la pantalla de edición de un usuario (figura 21). En la parte inferior izquierda se observa el botón *Eliminar* (de color rojo), que si pulsado procede sin confirmación al borrado del usuario actual.

## C.2. Cliente móvil y web

En esta sección del anexo se detallan todas las pantallas añadidas en el transcurso de este proyecto a la aplicación. También se muestra alguna pantalla ya existente, con motivo de seguir el flujo de ejecución de la aplicación que permita al usuario obtener una visión más precisa del funcionamiento del programa. La apariencia de algunas pantallas o el proceso seguido en algunos casos puede variar debido a alguna actualización de la aplicación.

### C.2.1. Pantalla principal de la aplicación

Esta pantalla muestra imágenes de los campus más relevantes de cada ciudad (Huesca, Zaragoza y Teruel). Cuando se pulsa sobre cualquiera de las imágenes se obtiene acceso al mapa del campus principal en cada uno de los casos.

Se han realizado modificaciones respecto de la pantalla principal original, con el objetivo de ofrecer una navegación más intuitiva y que mantenga un aspecto similar a la versión web de la aplicación que también ha sido modificada como se explica a continuación.

En la parte superior izquierda de la pantalla y en todas las vistas de la aplicación, se observa un icono de navegación, que si pulsado despliega un menú lateral desde la izquierda que permite acceder a las distintas pantallas de la aplicación, así como iniciar sesión como usuario registrado.

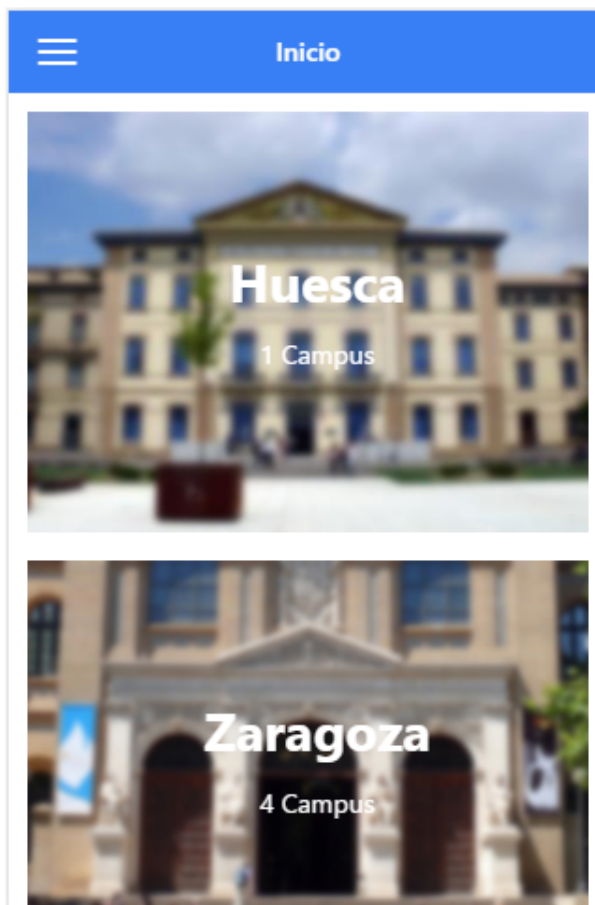


Figura 23: Pantalla principal del cliente de móvil.

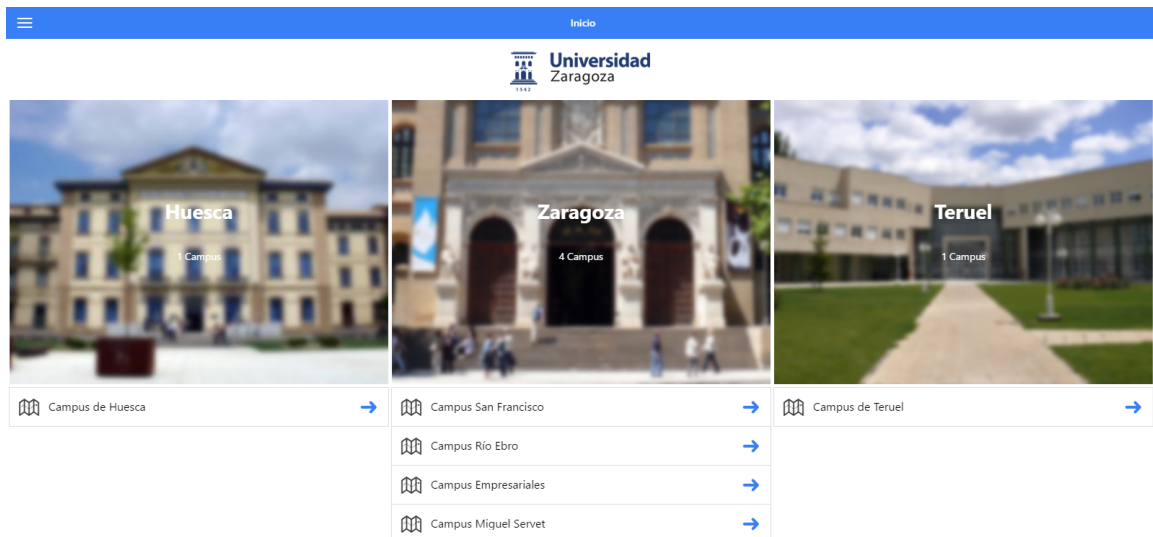


Figura 24: Pantalla principal del cliente de escritorio.

Esta pantalla ha sido rediseñada por completo, ya que la anterior no se ajustaba correctamente a tamaños de pantalla grandes como es el caso de la versión web de este cliente.

En primer lugar se observan tres imágenes correspondientes a las tres ciudades que tienen algún campus perteneciente a *Unizar* [29]. Debajo de cada imagen se encuentran los botones que llevan a los distintos campus situados en cada ciudad. De esta manera, pulsando sobre el botón (representado por la caja rectangular entera) de *Campus Río Ebro* se accede a la vista del mapa de dicho campus (figura 28).

Con el menú lateral desplegado se observa que existen diferentes opciones para seleccionar. El elemento nuevo es el denominado como *Iniciar sesión* que al pulsar lleva al usuario a una nueva pantalla con un pequeño formulario (figura 26).

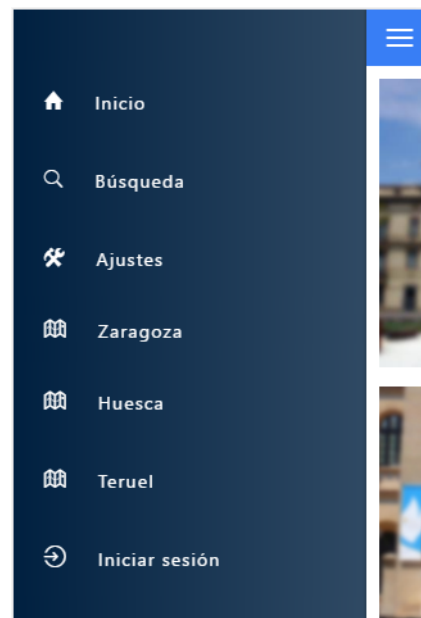


Figura 25: Pantalla principal del cliente de móvil con el menú lateral desplegado.

## C.2.2. Pantalla de inicio de sesión

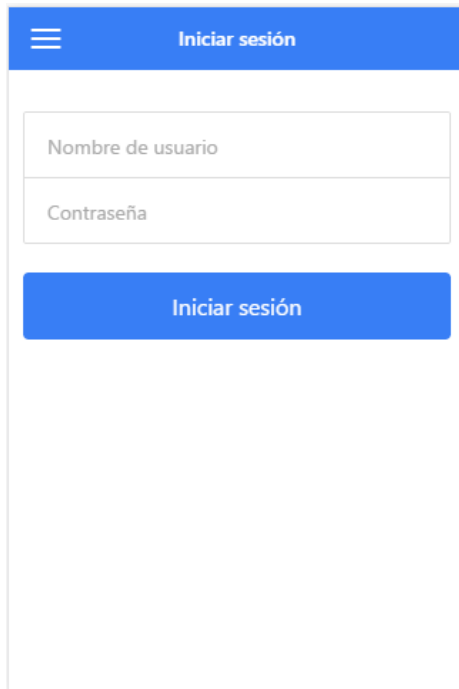


Figura 26: Pantalla de inicio de sesión de la aplicación.

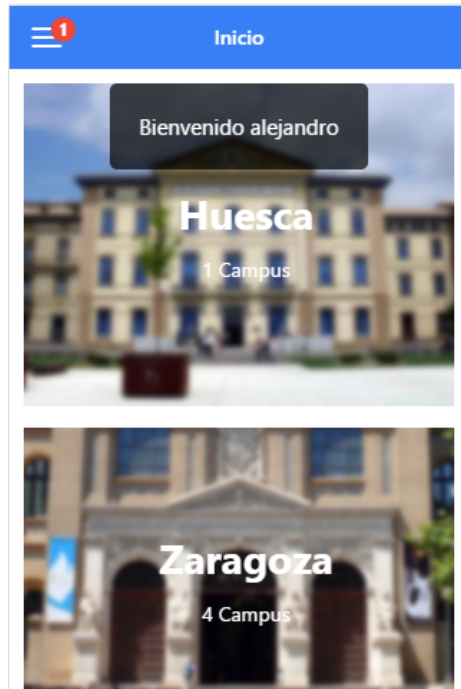


Figura 27: Pantalla principal de la aplicación con mensaje de bienvenida.

La pantalla de la izquierda (figura 26) muestra un pequeño formulario donde introducir las credenciales del usuario, esto es, nombre de usuario y contraseña (figura 26). En caso de ser correctas y tras pulsar el botón de *Iniciar sesión* la aplicación lleva directamente la pantalla principal con un mensaje de bienvenida (figura 27).

En caso de error, la aplicación notifica al usuario para que vuelva a introducir sus credenciales.

### C.2.3. Pantalla de visualización del mapa de un campus

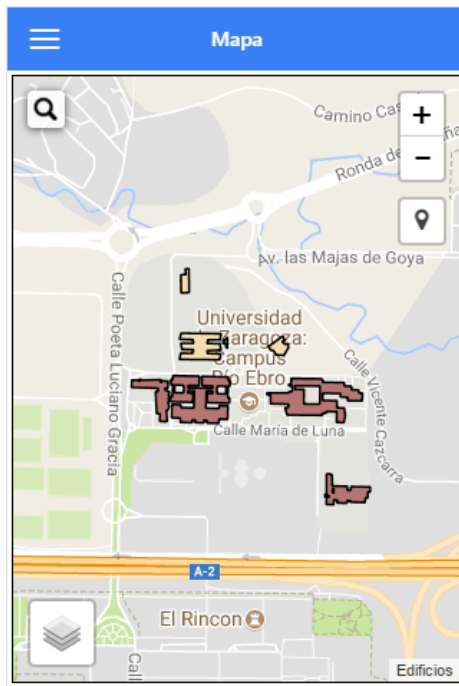


Figura 28: Pantalla de visualización del mapa del *Campus Río Ebro*.

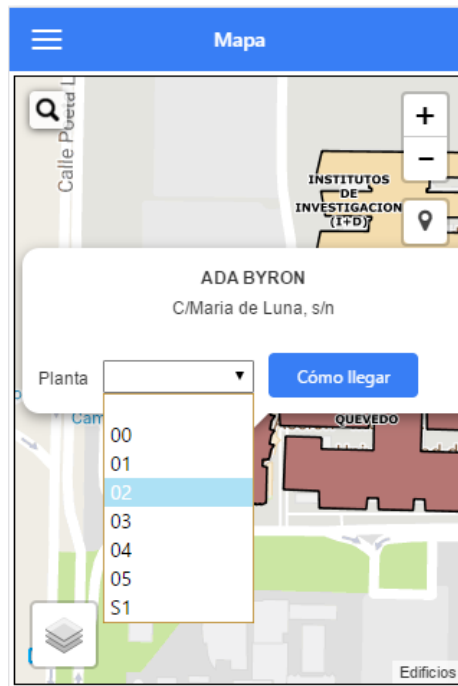


Figura 29: Pantalla de visualización del mapa del *Campus Río Ebro*.

En la pantalla de la izquierda (figura 28) se muestra la vista de un mapa interactivo sobre un campus de la Universidad de Zaragoza. Entre otras funcionalidades ya existentes se puede seleccionar un edificio pulsando dentro de su contorno para desplegar un *pop-up* que permite realizar varias acciones como se aprecia en la imagen de la derecha (figura 29).

Al seleccionar la planta deseada de un edificio la aplicación muestra al usuario en una nueva pantalla la vista del plano de la planta seleccionada (30).

#### C.2.4. Pantalla de visualización del plano de una planta

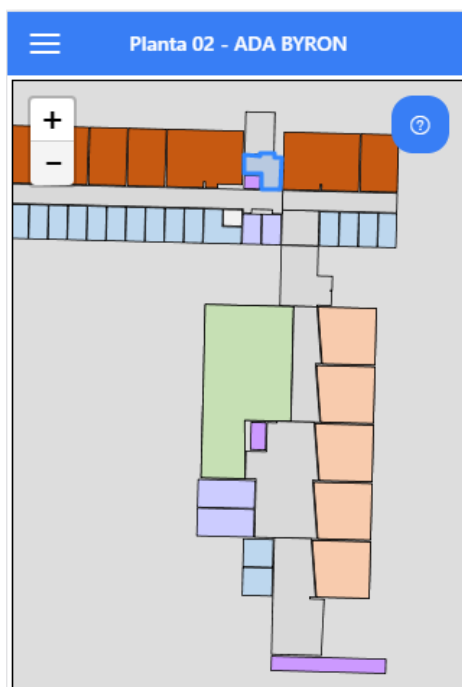


Figura 30: Pantalla de visualización del plano de una planta.



Figura 31: Pantalla de elección de una acción sobre un espacio.

En la vista de la izquierda (figura 30) se puede seleccionar cualquier espacio de la planta pulsando sobre él, lo cual activará la posibilidad de realizar varias acciones.

Este menú (figura 31) aparece como uno de los añadidos del proyecto a la aplicación. Se ha modificado la apariencia respecto *pop-up* original para integrar de manera más intuitiva la creación de notificaciones. En la parte superior del menú se observa el identificador del espacio así como su nombre. Se aprecian cuatro acciones a realizar:

- *Más información*: lleva a una pantalla adicional que muestra la información detallada sobre el espacio (32).
- *Notificar incidencia*: esta opción aparece para todo tipo de usuarios y abre una modal para crear la incidencia (34).
- *Notificar cambio*: esta opción sólo aparece para los usuarios registrados o administradores que hayan iniciado sesión en la aplicación (35).
- *Cancelar*: cierra el menú (similar a pulsar sobre un punto fuera del menú) y vuelve a la vista de la planta anterior.

### C.2.5. Pantalla de visualización de los detalles de un espacio

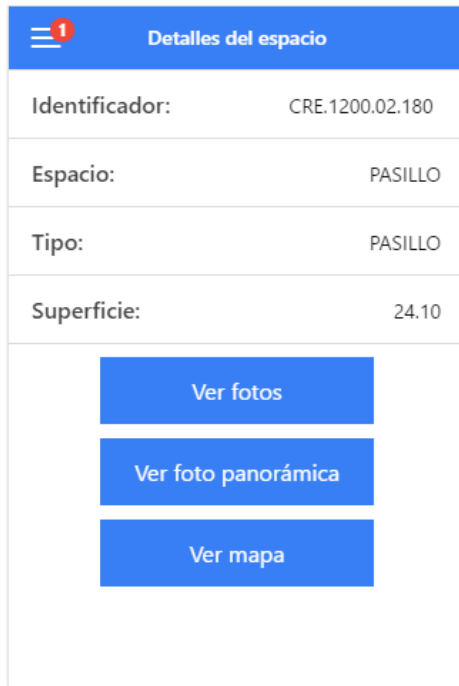


Figura 32: Pantalla de visualización de los detalles de un espacio.




Figura 33: Pantalla de visualización de una imagen panorámica.

Se ha mantenido la interfaz de la pantalla de la izquierda (figura 32), con la inclusión de la opción *Ver foto panorámica* (sólo si el espacio consta de una imagen de este tipo para visualizar) que permite al usuario acceder a una nueva vista donde visualizar e interactuar con la imagen panorámica (33).

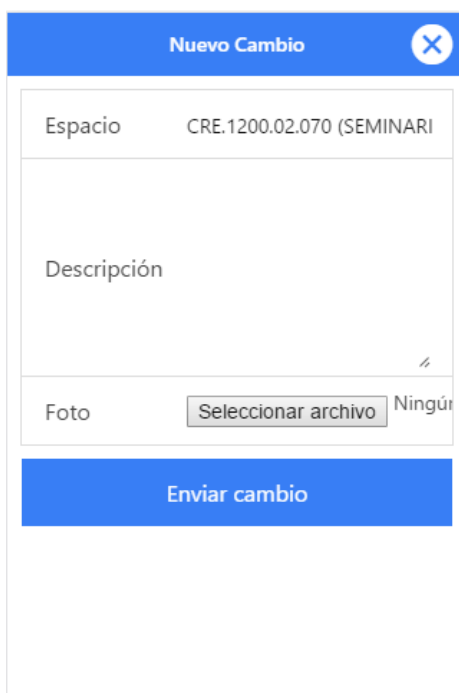
La pantalla situada en la derecha muestra una imagen panorámica del espacio seleccionado al usuario. Se puede interactuar con la imagen pulsando sobre un punto cualquiera de la misma y arrastrando el cursor en cualquier dirección para desplazarla y así girar entorno al punto desde el que fue tomada.

## C.2.6. Pantalla de creación de una notificación



The screenshot shows a mobile application form titled "Nueva Incidencia" with a close button (X) in the top right corner. The form contains the following fields: "Espacio" with the value "CRE.1200.02.080 (SEMINARI)", "Email" with the value "560285@unizar.es", and a large "Descripción" text area. Below the description is a "Foto" field with a "Seleccionar archivo" button and the text "Ningún". At the bottom of the form is a blue button labeled "Enviar incidencia".

Figura 34: Pantalla de creación de una incidencia en un espacio.



The screenshot shows a mobile application form titled "Nuevo Cambio" with a close button (X) in the top right corner. The form contains the following fields: "Espacio" with the value "CRE.1200.02.070 (SEMINARI)", a large "Descripción" text area, and a "Foto" field with a "Seleccionar archivo" button and the text "Ningún". At the bottom of the form is a blue button labeled "Enviar cambio".

Figura 35: Pantalla de creación de un cambio en un espacio.

En la imagen de la izquierda (figura 34) se observa la pantalla definida para la creación de una incidencia. Se presenta un formulario con varios campos de los cuales el usuario tiene que rellenar como mínimo la *descripción*. Adicionalmente se puede proporcionar un *email* para recibir un aviso cuando el administrador resuelva la incidencia que se va a enviar y por último, de manera opcional, se puede proporcionar una *foto* relacionada con la incidencia. El usuario debe tener la imagen almacenada en su dispositivo previamente.

Por otro lado, en la imagen de la derecha (figura 35) se observa el formulario necesario para la creación de un cambio. Para acceder a esta pantalla es necesario haber iniciado sesión en la aplicación previamente (desde el menú lateral). Respecto a los campos presentes, para poder enviar el cambio es necesario rellenar la *descripción* y opcionalmente se puede adjuntar una *foto*.

En ambos casos, al enviar la notificación se muestra un *pop-up* con el resultado de la operación que el usuario debe aceptar para volver a la vista previa de la planta correspondiente al espacio sobre el que se ha creado la notificación.



### C.2.7. Pantalla de gestión de cambios

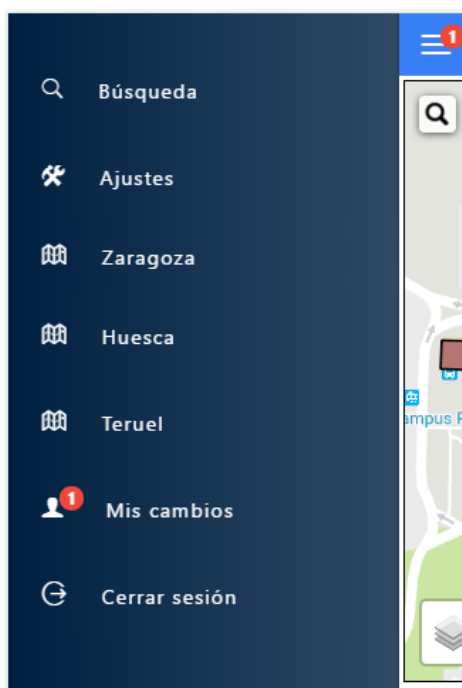


Figura 36: Pantalla de visualización del menú lateral desplegado con notificación. Figura 37: Pantalla de gestión de los cambios creados por el usuario.

Si un usuario ha iniciado sesión y ha creado algún cambio es probable que aparezca un icono rojo en la parte superior izquierda de la pantalla. Esto quiere decir que el administrador ha respondido a su cambio y ahora el usuario tiene que realizar alguna modificación. Si se pulsa sobre el icono rojo se despliega un menú (figura 36) que permite acceder a la pantalla de *Mis cambios* (figura 37) donde se gestionan los cambios de un usuario. Se dividen en cuatro estados:

- *Pendientes*: se listan los cambios que ya han sido respondidos por el administrador y deben ser actualizados por el usuario (la suma de todos ellos corresponde al número que aparece como notificación dentro del círculo rojo, tanto en el menú desplegable como en el icono de navegación a lo largo de toda la aplicación).
- *Esperando*: muestra una lista de los cambios creados o modificados por el usuario y que el administrador todavía no ha resuelto. El número total de cambios incluidos en esta categoría aparece junto al nombre de la misma en la parte superior de la pantalla.
- *Aprobados*: muestra una lista de los cambios resueltos por el administrador positivamente, es decir los cambios aprobados. El número total de cambios incluidos en esta categoría aparece junto al nombre de la misma en la parte superior de la pantalla.
- *Rechazados*: muestra una lista de los cambios resueltos por el administrador negativamente. El número total de cambios incluidos en esta categoría aparece junto al nombre de la misma en la parte superior de la pantalla.

## C.2.8. Pantalla de actualización de cambios



Figura 38: Pantalla de actualización de cambios.

Esta pantalla permite al usuario modificar la información de un cambio al que el administrador ha respondido. De esta manera, aparecen los siguientes elementos:

- *Imagen del cambio*: es la última imagen proporcionada por el usuario, ya sea en la creación del cambio o en alguna modificación previa.
- *Nombre del espacio*: representa el nombre natural del espacio.
- *Fecha de última modificación*: situada a la derecha del nombre del espacio y justo debajo de la imagen. Es el criterio mediante el cual se ordenan las listas de cambios. De más reciente a más antiguo.
- *Cambiar imagen*: es un botón que permite modificar la imagen actual del cambio. Una vez seleccionada una nueva imagen se puede ver la previsualización de la misma donde se encontraba la anterior.
- *Localización*: detalla el lugar exacto en el que se encuentra el espacio modificado.
- *Descripción*: describe la modificación realizada sobre el espacio. Sólo la puede modificar el usuario creador del cambio.
- *Comentario del administrador*: contiene la respuesta del administrador al cambio notificado por el usuario. Sólo la puede modificar el administrador.

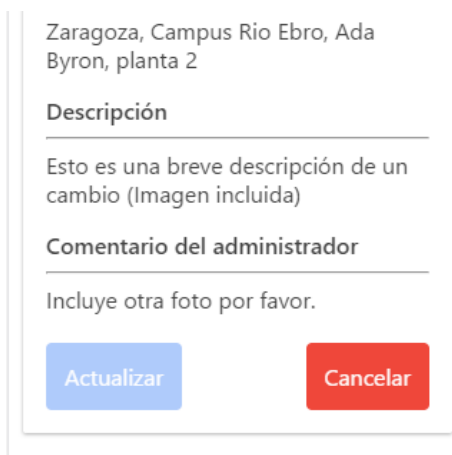


Figura 39: Pantalla de actualización de cambios (continuación).

- *Actualizar*: este botón se activa cuando uno de los dos campos imagen o descripción es modificado, de manera que se aporta nueva información. Si se pulsa guarda los cambios y los envía al administrador. Además, modifica inmediatamente el estado del cambio a *Esperando*.
- *Cancelar*: este botón siempre está activo y permite deshacer todos los cambios y cerrar la vista de los detalles del cambio.

Para representar esta pantalla en un dispositivo de dimensiones mayores se ha diseñado una vista diferente con otra distribución de la información, pero con el mismo conjunto de operaciones.

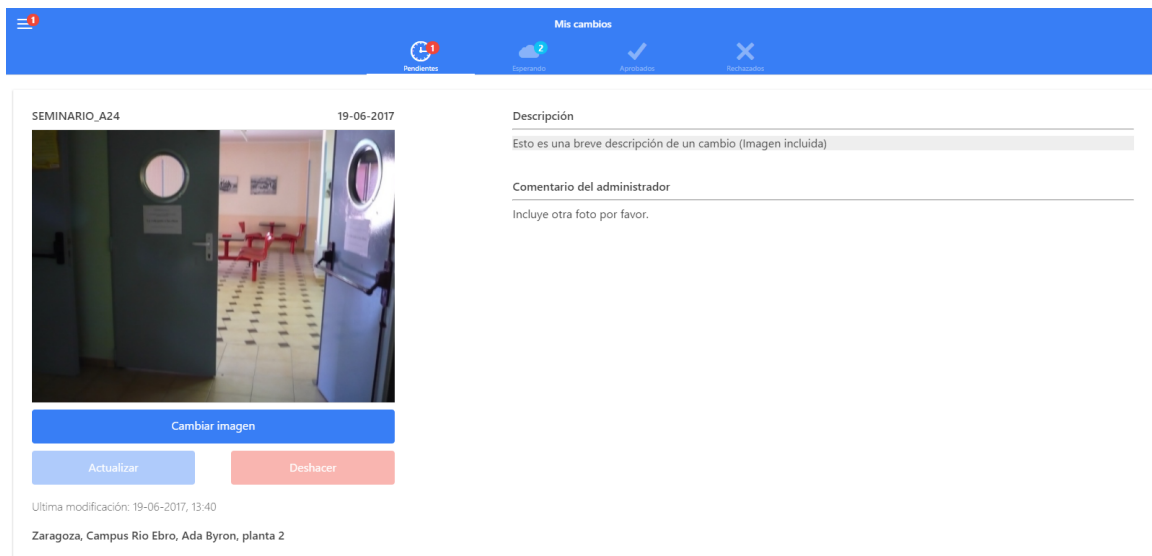


Figura 40: Pantalla de gestión de mis cambios.

## D. Pruebas sistemáticas

### D.1. Cliente administrador

#### Test de revisión de un cambio

1. Acceder a la pantalla principal de la aplicación.
2. Proporcionar credenciales inválidas y comprobar que resulta en error.
3. Proporcionar credenciales válidas y comprobar que la aplicación redirige a la pantalla principal.
4. Desde la pantalla principal seleccionar la opción *Espacios* del menú lateral izquierdo y en su desplegable pulsar sobre *Listado de cambios*.
5. Comprobar que aparece una lista en formato tabla con múltiples cambios. Seleccionar una fila y pulsar sobre el botón *Revisar*.
6. Comprobar que aparece una modal con la información del cambio correctamente rellena (identificador del espacio, descripción del cambio, estado del cambio y comentario del administrador).
7. Pulsar sobre el botón *Cancelar* y comprobar que se cierra la modal y la aplicación vuelve a la vista completa de la tabla.
8. Comprobar el valor del estado del cambio seleccionado.
9. Con el cambio seleccionado, pulsar sobre el botón *Revisar* y comprobar que se abre una modal como la anterior.
10. Comprobar que el estado del cambio dentro de la modal se corresponde con el mostrado en su fila de la tabla anterior.
11. Rellenar el campo de *comentario del administrador* y pulsar sobre *Pedir revisión*.
12. Comprobar que el estado del cambio en la tabla se ha modificado a *Pendiente del usuario*.

#### Test de creación y eliminación de un usuario

1. Acceder a la pantalla principal de la aplicación.
2. Proporcionar credenciales válidas y comprobar que la aplicación redirige a la pantalla principal.
3. Desde la pantalla principal, en la parte superior derecha seleccionar el icono de usuario y en su desplegable pulsar sobre *Crear usuario*.
4. Comprobar que la aplicación redirige a la pantalla de creación de usuario con el formulario correspondiente.

5. Rellenar todos los campos del formulario y pulsar sobre *Guardar*.
6. En el menú lateral izquierdo, seleccionar la opción *Usuarios* y en su desplegable pulsar sobre *Listado de usuarios*.
7. Comprobar que aparece una lista en formato tabla con múltiples usuarios.
8. Comprobar que el usuario creado aparece en esa lista.
9. Selecciona la fila del usuario creado y pulsar sobre el botón *Eliminar*.
10. Comprobar que aparece una modal con la información del usuario correctamente rellena.
11. Pulsar sobre el botón *Eliminar* y comprobar que se cierra la modal y la aplicación vuelve a la vista completa de la tabla.
12. Comprobar que el usuario eliminado no aparece en la lista de usuarios.

## D.2. Cliente móvil

### Test de creación de un cambio

1. Acceder a la pantalla principal de la aplicación.
2. Abrir el menú lateral y seleccionar la opción *Iniciar sesión*.
3. Proporcionar credenciales inválidas y comprobar que resulta en error.
4. Proporcionar credenciales válidas y comprobar que la aplicación redirige a la pantalla principal.
5. Desde la pantalla principal seleccionar un campus para acceder a su mapa.
6. Una vez cargado el mapa, seleccionar un edificio y elegir una planta para visualizar su plano.
7. Desde la vista del plano de una planta seleccionar un espacio y comprobar que aparece la opción para *Notificar cambio* (exclusiva de usuarios registrados) junto a *Notificar incidencia*.
8. Pulsar sobre *Notificar cambio*. Comprobar que se abre una modal con un formulario para proporcionar información sobre el cambio.
9. Rellenar los campos de *descripción* e *imagen* del formulario y pulsar sobre el botón de *enviar*.
10. Comprobar que aparece un mensaje de éxito al enviar el cambio, aceptar y comprobar que la aplicación vuelve al plano de la planta actual.
11. Abrir el menú lateral y seleccionar la opción de *Mis cambios*.
12. Desde la vista de *Mis cambios* cambiar a la pestaña de *Esperando* y comprobar que el cambio creado anteriormente se encuentra el primero de la lista.

## Test de actualización de un cambio

1. Acceder a la pantalla principal de la aplicación.
2. Abrir el menú lateral y seleccionar la opción *Iniciar sesión*.
3. Proporcionar credenciales inválidas y comprobar que resulta en error.
4. Proporcionar credenciales válidas y comprobar que la aplicación redirige a la pantalla principal.
5. Abrir el menú lateral y seleccionar la opción de *Mis cambios*.
6. Desde la vista de *Mis cambios*, en la pestaña de *Pendientes*, sin modificar ningún campo pulsar sobre el botón *Actualizar*. Comprobar que está desactivado y no se produce ningún efecto.
7. Modificar el campo descripción del primer cambio de la lista y comprobar que el botón *Actualizar* de ese cambio se habilita.
8. Pulsar el botón *Deshacer* del primer cambio y comprobar que la *descripción* recupera su valor original y que el botón *Actualizar* ahora está deshabilitado.
9. Pulsar sobre el botón *Cambiar Imagen*, seleccionar una imagen distinta a la ya asociada a ese cambio y comprobar que se modifica sin interacción adicional con la aplicación.
10. Comprobar que el valor del número que aparece al lado del nombre de la pestaña actual y el de la parte superior izquierda de la pantalla junto al icono de navegación son el mismo.
11. Pulsar sobre el botón *Actualizar* y comprobar que aparece un *pop-up* indicando el resultado de la operación.
12. Aceptar el mensaje del *pop-up* y comprobar que la pantalla se mantiene en la vista de *Pendientes* y que el contador de esta pestaña ha descendido en una unidad respecto a la comprobación anterior. Si antes valía *1*, comprobar que ahora no aparece el icono rojo de notificación en ninguna de las dos posiciones anteriores.
13. Comprobar que el valor del número que aparece al lado del nombre de la pestaña *Esperando* ha incrementado en una unidad.
14. Seleccionar la pestaña *Esperando* y comprobar que el cambio actualizado se encuentra en la primera posición y tiene la imagen añadida anteriormente.

## E. Casos de uso

Tabla 4: Casos de uso: Crear una incidencia con imagen y correo electrónico

<b>Nombre</b>	Crear una incidencia con imagen y correo electrónico
<b>Descripción</b>	Se notifica una incidencia al administrador. Cuando éste la resuelva se envía un correo electrónico a la dirección proporcionada por el usuario.
<b>Actor principal</b>	Usuario anónimo en aplicación móvil o cliente web.
<b>Precondición</b>	El usuario se encuentra en la pantalla de visualización del plano de una planta de un edificio de la universidad. El usuario no ha iniciado sesión en la aplicación.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de visualización del plano de una planta de un edificio de la universidad y ha recibido una confirmación del resultado de la creación de la incidencia.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un espacio de la planta.</li> <li>2. El usuario selecciona la opción <i>Notificar incidencia</i></li> <li>3. El usuario rellena el campo de <i>Email</i> y selecciona la imagen que desea añadir.</li> <li>4. El usuario pulsa sobre enviar y aparece un <i>pop-up</i> con la confirmación del resultado de la operación.</li> </ol>

Tabla 5: Casos de uso: Crear un cambio con imagen

<b>Nombre</b>	Crear un cambio con imagen
<b>Descripción</b>	Se crea un cambio que inicia el proceso de comunicación con el administrador.
<b>Actor principal</b>	Usuario registrado en aplicación móvil o cliente web.
<b>Precondición</b>	El usuario se encuentra en la pantalla de visualización del plano de una planta de un edificio de la universidad. El usuario ha iniciado sesión en la aplicación.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de visualización del plano de una planta de un edificio de la universidad y ha recibido una confirmación del resultado de la creación del cambio.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un espacio de la planta.</li> <li>2. El usuario selecciona la opción <i>Notificar cambio</i></li> <li>3. El usuario rellena el campo de <i>Descripción</i> y selecciona la imagen que desea añadir.</li> <li>4. El usuario pulsa sobre enviar y aparece un <i>pop-up</i> con la confirmación del resultado de la operación.</li> </ol>

Tabla 6: Casos de uso: Actualizar un cambio

<b>Nombre</b>	Actualizar un cambio
<b>Descripción</b>	Se actualiza la información de un cambio que estaba pendiente de modificación por parte del usuario para que el administrador pueda obtener información más detallada sobre el cambio.
<b>Actor principal</b>	Usuario registrado en aplicación móvil o cliente web.
<b>Precondición</b>	El usuario ha iniciado sesión en la aplicación. El usuario se encuentra en la pantalla de visualización de <i>Mis cambios</i> en la pestaña de <i>Pendientes</i> y hay al menos un cambio listado.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de visualización de <i>Mis cambios</i> en la pestaña de <i>Pendientes</i> y el cambio actualizado ya no se encuentra listado.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario modifica el campo <i>descripción</i> de un cambio.</li> <li>2. El usuario selecciona la opción <i>Actualizar</i> y aparece un <i>pop-up</i> de confirmación de la operación realizada.</li> <li>4. El usuario pulsa acepta el mensaje y vuelve al listado de cambios de la pestaña <i>Pendientes</i> donde ya no está el cambio actualizado.</li> </ol>

Tabla 7: Casos de uso: Visualizar imagen panorámica en 360°

<b>Nombre</b>	Visualizar imagen panorámica en 360°
<b>Descripción</b>	Se visualiza una imagen panorámica en 360° de un espacio de la universidad.
<b>Actor principal</b>	Usuario anónimo en aplicación móvil o cliente web.
<b>Precondición</b>	El usuario se encuentra en la pantalla de detalles de un espacio.
<b>Postcondición</b>	El usuario visualiza una imagen panorámica de un espacio.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre la opción <i>Ver imagen panorámica</i>.</li> <li>2. El usuario visualiza la imagen panorámica del espacio seleccionado.</li> </ol>



Tabla 8: Casos de uso: Visualizar todos los cambios registrados

<b>Nombre</b>	Visualizar todos los cambios registrados
<b>Descripción</b>	Se visualiza una lista en formato tabla con todos los cambios registrados por todos los usuarios en la aplicación.
<b>Actor principal</b>	Usuario administrador en cliente web administrador.
<b>Precondición</b>	El usuario se encuentra en la pantalla principal de la aplicación.
<b>Postcondición</b>	El usuario visualiza una lista con todos los cambios registrados.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre la opción <i>Espacios y Listado de cambios</i>.</li> <li>2. El usuario visualiza una tabla con la información de todos los cambios registrados en la aplicación hasta ese momento.</li> </ol>

Tabla 9: Casos de uso: Visualizar todas las incidencias registradas

<b>Nombre</b>	Visualizar todas las incidencias registradas
<b>Descripción</b>	Se visualiza una lista en formato tabla con todas las incidencias registradas por todos los usuarios en la aplicación hasta ese momento.
<b>Actor principal</b>	Usuario administrador en cliente web administrador.
<b>Precondición</b>	El usuario se encuentra en la pantalla principal de la aplicación.
<b>Postcondición</b>	El usuario visualiza una lista con todas las incidencias registradas.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa sobre la opción <i>Espacios y Listado de incidencias</i>.</li> <li>2. El usuario visualiza una tabla con la información de todas las incidencias registradas en la aplicación hasta ese momento.</li> </ol>

Tabla 10: Casos de uso: Gestionar un cambio

<b>Nombre</b>	Gestionar un cambio
<b>Descripción</b>	Se revisa un cambio y se toma una decisión sobre él. Se puede elegir entre <i>Aprobar</i> , <i>Rechazar</i> o <i>Pedir revisión</i> .
<b>Actor principal</b>	Usuario administrador en cliente web administrador.
<b>Precondición</b>	El usuario se encuentra en la pantalla de listado de cambios.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de listado de cambios. El cambio gestionado ha modificado su estado según la modificación del usuario y es visible en la tabla que lista los cambios.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un cambio y pulsa sobre el botón <i>Revisar</i>.</li> <li>2. El usuario escribe un comentario y pulsa sobre uno de los tres botones disponibles (<i>Aprobar</i>, <i>Pedir revisión</i>, <i>Rechazar</i>).</li> <li>3. El usuario visualiza el nuevo estado del cambio gestionado en la tabla acorde con la decisión que ha tomado.</li> </ol>

Tabla 11: Casos de uso: Gestionar una incidencia

<b>Nombre</b>	Gestionar una incidencia
<b>Descripción</b>	Se revisa una incidencia y se toma una decisión sobre ella. Se puede elegir entre <i>Aprobar</i> o <i>Rechazar</i> .
<b>Actor principal</b>	Usuario administrador en cliente web administrador.
<b>Precondición</b>	El usuario se encuentra en la pantalla de listado de incidencias.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de listado de incidencias. La incidencia gestionada ha modificado su estado según la modificación del usuario y es visible en la tabla que lista las incidencias.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona una incidencia y pulsa sobre el botón <i>Revisar</i>.</li> <li>2. El usuario escribe un comentario y pulsa sobre uno de los dos botones disponibles (<i>Aprobar</i>, <i>Rechazar</i>).</li> <li>3. Se envía un correo al usuario creador de la incidencia con el comentario del usuario como mensaje.</li> <li>4. El usuario visualiza el nuevo estado de la incidencia gestionada en la tabla acorde con la decisión que ha tomado.</li> </ol>

Tabla 12: Casos de uso: Modificar permisos de un usuario

<b>Nombre</b>	Modificar permisos de un usuario
<b>Descripción</b>	Se selecciona un usuario y se modifican sus permisos de manera que pase de administrador a usuario registrado o viceversa.
<b>Actor principal</b>	Usuario administrador en cliente web administrador.
<b>Precondición</b>	El usuario se encuentra en la pantalla de listado de usuarios.
<b>Postcondición</b>	El usuario se encuentra en la pantalla de listado de usuarios. El usuario modificado ha cambiado su rol según la modificación del usuario y es visible en la tabla que lista los usuarios.
<b>Escenario de éxito</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un usuario y pulsa sobre el botón <i>Editar</i>.</li> <li>2. El usuario modifica el rol del usuario seleccionado a <i>ADMIN</i>.</li> <li>3. El usuario pulsa sobre el botón <i>Guardar</i> y se cierra la modal.</li> <li>4. El usuario visualiza que el nuevo rol del usuario modificado es ahora <i>ADMIN</i> .</li> </ol>