

# Universidade de Vigo

## ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

**D. Antonio Neira Valiñas**

para a obtención do Título de Graduado en Enxeñaría Informática

**Sistema de información de planes de prevención**



Setembro, 2021

**Traballo de Fin de Grao N°:** EI 20/21-64

**Titor/a:** Javier Rodeiro Iglesias

**Área de coñecemento:** Linguaxes e Sistemas Informáticos

**Departamento:** Informática



## Tabla de contenido

|         |   |    |
|---------|---|----|
| 1       | Introducción .....  | 5  |
| 2       | Objetivos .....   | 6  |
| 3       | Resumen de la solución propuesta .....                    | 7  |
| 3.1     | Metodología .....   | 7  |
| 3.2     | Solución Propuesta.....                                   | 9  |
| 4       | Planificación y Seguimiento.....                          | 11 |
| 4.1     | Planificación .....                                       | 11 |
| 4.2     | Puntos Críticos.....                                      | 12 |
| 4.3     | Ejecución .....   | 12 |
| 4.4     | Desviaciones.....   | 12 |
| 5       | Arquitectura .....  | 14 |
| 6       | Tecnologías e integración de productos de terceros.....   | 15 |
| 6.1     | Tecnologías Utilizadas .....                              | 15 |
| 6.2     | Productos de Terceros .....                               | 17 |
| 7       | Especificación y Análisis de Requisitos .....             | 18 |
| 7.1     | Requisitos No Funcionales .....                           | 18 |
| 7.2     | Requisitos Funcionales .....                              | 19 |
| 7.2.1   | Iteración 6. Gestión de Responsables.....                 | 19 |
| 7.2.2   | Iteración 7. Gestión de Edificios .....                   | 20 |
| 7.2.2.1 | Gestión de Edificios (RF2) .....                          | 20 |
| 7.2.2.2 | Gestión de Plantas (RF3) .....                            | 20 |
| 7.2.2.3 | Gestión de Espacios (RF4) .....                           | 22 |
| 7.2.3   | Iteración 8. Gestión de Planes de Prevención .....        | 23 |
| 7.2.3.1 | Definición de Planes (RF5) .....                          | 23 |
| 7.2.3.2 | Definición de Documentos (RF6) .....                      | 24 |
| 7.2.3.3 | Definición de Procedimientos (RF7).....                   | 24 |
| 7.2.3.4 | Definición de Rutas (RF8).....                            | 25 |
| 7.2.3.5 | Definición de Formaciones (RF9) .....                     | 25 |
| 7.2.3.6 | Definición de Simulacros (RF10) .....                     | 26 |
| 7.2.4   | Iteración 9. Implementación de Planes de Prevención. .... | 26 |
| 7.2.4.1 | Implementación de Documentos en Edificios (RF11) .....    | 26 |
| 7.2.4.2 | Implementación de Procedimientos en Edificios (RF12)..... | 27 |
| 7.2.4.3 | Implementación de Rutas en Plantas (RF13) .....           | 28 |
| 7.2.4.4 | Implementación de Formaciones en Edificios (RF14) .....   | 29 |
| 7.2.4.5 | Implementación de Simulacros en Edificios (RF15) .....    | 30 |
| 7.2.4.6 | Verificación de Plan Implementado (RF16).....             | 30 |
| 7.2.5   | Iteración 10. Sistema de Notificaciones. ....             | 31 |
| 7.3     | Casos de Uso .....  | 31 |
| 7.3.1   | Diagramas de Casos de Uso.....                            | 31 |
| 7.3.2   | Descripción de Casos de Uso.....                          | 33 |
| 7.3.2.1 | Iteración 6. Gestión de Usuarios .....                    | 33 |
| 8       | Diseño del Software (Estático y Dinámico).....            | 37 |

|       |                                       |                              |
|-------|---------------------------------------|------------------------------|
| 8.1   | Diseño del Software Estático.....     | 37                           |
| 8.2   | Diseño del Software Dinámico .....    | 38                           |
| 8.2.1 | Iteración 6. Gestión de Usuarios..... | 38                           |
|       | Login .....                           | Error! Marcador no definido. |
|       | Consultar Usuarios.....               | Error! Marcador no definido. |
|       | Consultar Detalles del Usuario.....   | Error! Marcador no definido. |
|       | Eliminar Usuario.....                 | Error! Marcador no definido. |
|       | Añadir Usuario .....                  | Error! Marcador no definido. |
|       | Editar Usuario .....                  | Error! Marcador no definido. |
| 9     | Gestión de Datos e Información.....   | 43                           |

# 1 Introducción

La prevención de riesgos engloba a todas aquellas actividades o medidas de seguridad adoptadas por una empresa u organización con el objetivo de evitar (o reducir en la medida de lo posible) los riesgos derivados de la actividad laboral. Aunque actualmente en España la prevención de riesgos se haya convertido en un punto básico y fundamental en la actividad de cualquier organización, no fue hasta el año 1995 con la entrada en vigor de la Ley 31/1995 que la sociedad española empezó a tomar verdadera conciencia de su importancia.

La Ley 31/1995 constituye la base legislativa de la prevención de riesgos, y en ella se define un cuerpo base de garantías y responsabilidades necesarias para garantizar la seguridad de los trabajadores en el desempeño de sus actividades. Entre estas responsabilidades se encuentra la obligación empresarial de elaborar y aplicar un plan de prevención de riesgos laborales, a través del cual se integre la prevención de riesgos en el funcionamiento general de la empresa. Este plan debe ser recogido en un documento, y la empresa deberá mantenerlo a disposición de las autoridades laborales y sanitarias, actualizarlo cuando se produzcan cambios importantes en la estructura organizativa de la empresa o en los riesgos existentes que puedan afectar a la seguridad de los empleados, y difundirlo entre los trabajadores para que estos puedan cumplir con sus funciones y responsabilidades.

El plan de prevención es de obligatorio cumplimiento para cualquier empresa o persona autónoma que tenga al menos un trabajador, pero en función de las características de la organización o del tipo de actividad que lleve a cabo puede no ser el único. Por ejemplo, el Real Decreto 393/2007 define el plan de autoprotección con la finalidad de prevenir los riesgos que puedan afectar a personas o bienes en centros, establecimientos y dependencias donde se puedan producir situaciones de emergencia. Por otro lado, del artículo 20 de la Ley 31/1995 deriva el plan de emergencia y evacuación, documento en el que se especifican las medidas de protección y prevención previstas que tienen la finalidad de evitar accidentes en el entorno laboral. Por lo tanto, el proceso que debe llevar a cabo la empresa para el mantenimiento de la distinta documentación u otros elementos asociados a la prevención de riesgos se torna complejo, motivando deficiencias o malas praxis como por ejemplo implementar varios planes en el mismo documento, o ubicar los documentos de los planes en lugares poco accesibles por los miembros de la organización u a otros interesados. Este proceso puede llegar a ser incluso mucho más complejo en el caso de empresas u organizaciones que cuenten con

varias sedes, ya que se puede necesitar mantener planes específicos para cada una de las ubicaciones (cómo en el caso de los planes de evacuación).

El objetivo de este proyecto es apoyar al empresario y/o a los responsables de la organización en el proceso de mantenimiento de planes de prevención mediante el desarrollo de una aplicación que permita gestionar los documentos y otros elementos (como formaciones u simulacros) asociados a los distintos planes de prevención de la empresa, organizarlos en función de las distintas sedes, y hacerlos accesibles a los posibles interesados a través de portales dedicados.

## 2 Objetivos

El objetivo principal del proyecto es el diseño e implementación de una aplicación web que permita, a los responsables de una organización, centralizar y dar soporte al proceso de desarrollo y de mantenimiento de sus planes de prevención. Para ello, la aplicación debe cumplir con los siguientes objetivos.

- **Objetivo 1.** El sistema debe reflejar la estructura de la organización. En su forma más genérica la organización se visualizará como un conjunto de edificios, estando estos agrupados por ciudad y compuestos por plantas y espacios. De esta forma, la navegación por parte del usuario empezará por los ítems más genéricos (edificio de una ciudad) hasta llegar a los más concretos (espacio de una planta).
- **Objetivo 2.** El responsable de la organización tendrá la capacidad para gestionar a los responsables de los edificios, pudiendo crearlos, eliminarlos, modificarlos y asociarlos a edificios. La aplicación contará con un usuario de tipo responsable de organización (administrador) predefinido con el que poder empezar las gestiones.
- **Objetivo 3.** El responsable de la organización podrá crear y/o modificar cualquier edificio, planta o espacio, mientras que los responsables de los edificios sólo podrán crear/modificar las plantas y espacios de los edificios a los que están asignados.

- **Objetivo 4.** La aplicación debe permitir al responsable de la organización definir planes de prevención. Para definir un plan se adjuntará un nombre, y se especificarán los elementos (documentos, procedimientos y/o rutas de evacuación) necesarios para su implementación.
- **Objetivo 5.** El responsable de la organización podrá asignar los planes definidos a los distintos edificios que necesiten cumplimentarlos. La aplicación notificará, a los responsables de estos edificios, de los planes que han sido asignados y de los elementos necesarios para implementarlos.
- **Objetivo 6.** Cada responsable de edificio realizará una implementación específica de los planes asignados. El responsable de un edificio podrá cargar documentos, procedimientos y rutas de evacuación en formato PDF, ODT o DOC y asociarlos a un plan. Una vez adjuntados todos los elementos necesarios a un plan, el responsable del edificio podrá marcarlo como implementado, lo que provocará que la aplicación notifique del hecho al responsable de la organización.
- **Objetivo 7.** El responsable del edificio podrá crear formaciones y simulacros para preparar los planes de prevención, registrando para ello la fecha en la que se llevará a cabo. Las formaciones y simulacros con fecha futura aparecerán en la sección de eventos del edificio.

## 3 Resumen de la solución propuesta

### 3.1 Metodología

Para el desarrollo de este proyecto se ha elegido seguir la **metodología RUP (Rational Unified Process)**. Esta metodología soporta los modelos en Cascada y por Componentes, y presenta **tres características principales**:

- **Dirigido por casos de uso:** Se emplean casos de uso para capturar los requisitos funcionales del sistema. Un caso de uso representa un fragmento de funcionalidad que aporta valor al usuario, y además de especificar dicho requisito, también sirve como guía para su posterior diseño, implementación y pruebas.

- **Centrado en la arquitectura:** Se busca establecer una buena arquitectura en las etapas iniciales del desarrollo la cual no se vea demasiado afectada ante cambios posteriores. De esta forma, al principio del desarrollo la arquitectura se empezará a consolidar por medio de baselines, y esta se irá evolucionando en función de las necesidades del proyecto hasta obtener una arquitectura más robusta en las etapas finales.

La arquitectura y los casos de uso deben evolucionar en paralelo, de forma que los casos de uso deben encajar en la arquitectura cuando se implementan, y la arquitectura debe permitir el desarrollo de los casos de uso requeridos, tanto actuales como futuros.

- **Iterativo e incremental:** La aplicación se divide en subproyectos más pequeños, los cuales incorporan una parte de las especificaciones. Cada uno de estos subproyectos constituyen una iteración, cuyo resultado se traduce en un incremento que se añade al producto.

**RUP** se repite durante una serie de **ciclos**, donde cada ciclo constituye una **versión del producto** para los clientes. Cada uno de estos ciclos se dividen en **cuatro fases**, dentro de las cuales se realizan varias iteraciones en función del proyecto. Estas cuatro fases se describen a continuación.

- **Fase de Inicio:** Se establece el caso del negocio y se identifican las entidades externas que interactuarán con el sistema.
- **Fase de Elaboración:** Tiene como objetivo definir el dominio del problema, eliminar riesgos críticos y establecer la baseline de la arquitectura.
- **Fase de Construcción:** Conlleva la implementación del diseño realizado y su integración con el resto de componentes del sistema.
- **Fase de Transición:** Se entrega el sistema a los usuarios finales para realizar pruebas en un entorno real.



## 3.2 Solución Propuesta

La solución que se propone es una aplicación web que facilite los procesos de desarrollo, mantenimiento y seguimiento de los planes de prevención de una empresa u organización, organizando la documentación asociada a los planes de un modo en que esta sea fácilmente accesible a las personas o entidades interesadas, proporcionando un sistema de feedback que permita al responsable hacer un seguimiento del nivel de implementación de un plan en cada una de las sedes, y permitiendo registrar eventos que la empresa vaya a llevar a cabo para difundir el contenido de los planes entre los empleados.

El desarrollo de la aplicación se estructurará en las distintas fases de la metodología RUP tal y como se detalla a continuación.

### Fase de Inicio

Esta fase corresponde a la Iteración 0, y en ella es donde se llevará a cabo el estudio del caso de negocio, dando como resultado los siguientes apartados en este documento:

1 INTRODUCCIÓN, 2 OBJETIVOS, 3 RESUMEN DE LA SOLUCIÓN PROPUESTA, 4 PLANIFICACIÓN Y SEGUIMIENTO y 7 ESPECIFICACIÓN Y ANÁLISIS DE REQUISITOS.

### Fase de Elaboración

En la fase de elaboración se desarrollará la base de la aplicación. Las distintas funcionalidades identificadas en la fase de inicio serán representadas en un diagrama de casos de uso (Iteración 2), se modelará e implementará la base de datos (Iteración 3), se desarrollará la arquitectura inicial (Iteración 4) y se diseñarán las interfaces necesarias (Iteración 5). Con el diseño de las interfaces se cumple el Objetivo 1.

### Fase de Construcción

En la fase de construcción se llevará a cabo la implementación de la aplicación de forma iterativa. En cada iteración se seleccionarán un conjunto de casos de uso (funcionalidades) y para cada uno de los casos de uso seleccionados se diseñará su diagrama de secuencia, se llevará a cabo la implementación y se validará mediante pruebas.

Para el desarrollo de esta aplicación se ha dividido la fase de construcción en 6 iteraciones, las cuales se detallan a continuación.

Iteración 6. Gestión de Responsables. Iteración en la cual se implementa la gestión de los usuarios del sistema (responsable de la organización y responsable del edificio) y el proceso de autenticación en la aplicación. Al final de esta iteración se habrá cumplido el Objetivo 2.

Iteración 7. Gestión de Edificios. En esta iteración se implementan la gestión de los edificios de la organización y de los elementos de los que están compuestos (plantas y espacios), y se añade la funcionalidad de asignar edificios a responsables. Se añadirán restricciones para que un edificio sólo pueda ser gestionado por los responsables que tenga asignados. Con esta iteración se cumple el Objetivo 3.

Iteración 8. Gestión de Planes de Prevención. Se implementará la gestión de planes de prevención y la funcionalidad de asociar los planes a los distintos edificios de la organización. La gestión de planes de prevención sólo estará disponible para los usuarios definidos como responsable de la organización. Se cumple de esta forma el Objetivo 4 y parcialmente el Objetivo 5.

Iteración 9. Implementación de Planes de Prevención. En esta iteración se añadirán las funcionalidades para gestionar documentos, procedimientos y rutas, así como la funcionalidad para asociar estos elementos a los planes asignados al edificio. Al conjunto de documentos, procedimientos y rutas asociados a un plan se le considera como implementación del plan, y cada implementación será específica de un edificio. El responsable del edificio asociará los elementos necesarios para la implementación y podrá marcar el plan a completado. Esta iteración cumple parcialmente con el Objetivo 6.

Iteración 10. Sistema de Notificaciones. Se implementará un sistema de notificaciones para favorecer el seguimiento de los planes por parte de los distintos tipos de responsables de la organización. A través de un sistema de notificaciones, el sistema enviará una notificación al responsable de un edificio en el momento en el que el responsable de la organización asocie un plan a un edificio, mientras que el responsable de la organización recibirá una notificación cuando se haya completado la implementación de un plan en un edificio. Los responsables podrán modificar el estado de las notificaciones recibidas a “leídas”. Las notificaciones leídas serán eliminadas de la base de datos tras un período de tiempo. Al completar esta iteración, se cumplen completamente el Objetivo 5 y el Objetivo 6.

Iteración 11. Gestión de Eventos. Se implementará la gestión de formaciones y simulacros. Podrán gestionar eventos el responsable de la organización y los responsables cuyos edificios estén asociados al evento en cuestión. La consulta de

eventos estará accesible a cualquier tipo de usuario (identificado o no). Formaciones y simulacros planificados a fecha futura aparecerán en el panel de eventos dentro de los portales de los edificios involucrados. Finalmente se cumple el Objetivo 7.

#### Fase de Transición

En la fase de Transición se llevarán a cabo las pruebas de validación del sistema (Iteración 12) y se completará la documentación restante (Iteración 13).

## 4 Planificación y Seguimiento

### 4.1 Planificación

En la siguiente tabla se muestra la planificación inicial del proyecto, especificando un tiempo estimado a cada iteración y agrupándolas por fases. Se planifica dedicar 5 horas diarias al desarrollo del proyecto, con fecha de inicio **XX de XX del 2021** y fecha de finalización prevista **XX de XX del 2021**.

| ETAPAS   | TIEMPO ESTIMADO (horas) |
|--|-------------------------|
| <b>Fase de Inicio</b>                                | <b>15</b>               |
| Iteración 1. Análisis del problema.                  | 15                      |
| <b>Fase de Elaboración</b>                           | <b>60</b>               |
| Iteración 2. Diseño e implementación de la BD.       | 15                      |
| Iteración 3. Diseño del diagrama de casos de uso.    | 10                      |
| Iteración 4. Desarrollo de la arquitectura inicial.  | 15                      |
| Iteración 5. Prototipo de interfaces.                | 20                      |
| <b>Fase de Construcción</b>                          | <b>190</b>              |
| Iteración 6. Gestión de Responsables.                | 25                      |
| Iteración 7. Gestión de Edificios.                   | 30                      |
| Iteración 8. Gestión de Planes de Prevención.        | 35                      |
| Iteración 9. Implementación de Planes de Prevención. | 35                      |
| Iteración 10. Sistema de notificaciones.             | 40                      |
| Iteración 11. Gestión de eventos.                    | 25                      |
| <b>Fase de Transición</b>                            | <b>35</b>               |
| Iteración 12. Pruebas de validación del sistema.     | 20                      |
| Iteración 13. Documentación final.                   | 15                      |
| <b>TOTAL</b>   | <b>300</b>              |

(PENDIENTE DIAGRAMA DE GANTT)

## 4.2 Puntos Críticos

Se han identificado dos puntos críticos en la planificación. El primer punto crítico corresponde con la Implementación de Planes de Prevención (Iteración 9). En esta iteración se implementarán las funcionalidades de gestionar Documentos, Procedimientos y Rutas, incluyendo la posibilidad de asociarlos a Planes de Prevención y así como asociar Rutas a Plantas. Se debe permitir asociar estos elementos a los Planes de Prevención que se encuentren en estado “Pendiente” y modificar el estado del plan a “Implementado” una vez se hayan asociados los Documentos, Procedimientos y Rutas necesarios. El cambio de un estado a otro deberá hacerlo manualmente el responsable a través de la aplicación, y este cambio podrá ser reversible.

El segundo punto crítico identificado se corresponde con la implementación del Sistema de Notificaciones (Iteración 10). Esta iteración consiste en el desarrollo de un mecanismo que permita a la aplicación enviar periódicamente consultas en segundo plano (mediante Ajax) a la Base de Datos en busca de notificaciones “no leídas” dirigidas al usuario de la sesión. En caso de encontrar algún resultado, se deberá actualizar la vista para que las nuevas notificaciones aparezcan en el panel correspondiente. El usuario podrá marcar estas notificaciones como “leídas”, lo que provocará que sean eliminadas de la Base de Datos tras el transcurso de un período de tiempo.

## 4.3 Ejecución

## 4.4 Desviaciones



## 5 Arquitectura

Para este proyecto se ha optado por desarrollar una aplicación web siguiendo la arquitectura cliente-servidor. Esta arquitectura es una de las más populares a la hora de implementar sistemas distribuidos, y se basa en el reparto de las tareas de la aplicación entre dos componentes: el servidor, el cual ofrece servicios y/o recursos, y el cliente, el cual demanda estos servicios al servidor mediante protocolos de petición-respuesta. De esta forma, el servidor ofrece servicios de forma concurrente a varios a clientes mientras que los clientes acceden a ellos de forma transparente para los usuarios. La interacción entre cliente y servidor se describe a continuación:

1. Un cliente realiza una petición al servidor. En una aplicación web como este caso, el cliente es un navegador web y la petición es desencadenada, normalmente, debido a una acción del usuario.
2. El servidor recibe la petición, realiza las verificaciones oportunas y, si no se producen errores, la procesa. En este caso, el servidor es un Apache HTTP Server.
3. Cuando el servidor obtenga el resultado de procesar la petición, la envía al cliente. En un entorno web, la comunicación entre cliente y servidor se realiza sobre los protocolos TCP/IP (capa de transporte) y HTTP/HTTPS (capa de aplicación).
4. El cliente recibe el resultado.

En este proyecto la mayor parte de las responsabilidades (manipulación de datos, lógica de negocio, generación de vistas HTML) recaen en el lado servidor, quedando el cliente relegado a únicamente enviar peticiones y presentar al usuario las vistas que recibe del servidor.

Para el desarrollo de la aplicación en el lado servidor, se ha seguido el patrón de diseño Modelo-Vista-Controlador (MVC). MVC está basado en los conceptos de reutilización de código y separación de responsabilidades, y tiene como objetivo facilitar las tareas de desarrollo y mantenimiento de código. Para alcanzar este objetivo, MVC propone separar los datos, la presentación y la lógica de control de la aplicación en tres componentes:

1. Modelos: Representan los datos de la aplicación. Son los responsables de acceder a la información y de implementar la lógica de negocio.

2. Vistas: Tienen la responsabilidad de presentar el modelo en un formato adecuado para la interacción con el usuario.
3. Controladores: Actúan como intermediarios entre los Modelos y las Vistas. Tienen la responsabilidad de capturar los eventos procedentes de las Vistas y de realizar peticiones a los Modelos cuando se haga alguna solicitud sobre la información.

## 6 Tecnologías e integración de productos de terceros

### 6.1 Tecnologías Utilizadas

#### PHP

PHP, acrónimo recursivo de “PHP: Hypertext Preprocessor”, es un lenguaje interpretado de código abierto utilizado para desarrollar aplicaciones en el lado servidor. Aunque se trata de un lenguaje de propósito general, su capacidad de ser incrustado en documentos html lo hace destacar especialmente en el ámbito del desarrollo web para crear páginas web dinámicamente, las cuales son interpretadas por el navegador cliente como una página estática normal. Debido a esto y a su integración con las principales bases de datos (Oracle, MySQL, SQLite...) se ha elegido PHP para el desarrollo de esta aplicación.

#### HTML5

HTML, acrónimo de HyperText Markup Language, es un lenguaje de marcado que permite estructurar el contenido de un documento (como por ejemplo página web) mediante el uso de etiquetas. HTML5 es la última versión de este lenguaje, el cual incluye mejoras como nuevas etiquetas semánticas para estructurar documentos, etiquetas para insertar contenido multimedia, o almacenamiento local en el lado del cliente. El estándar HTML, a cargo de la W3C, es el que ha impuesto para elaborar páginas web, siendo adoptado por todos los navegadores. Por este motivo, se utilizará HTML5 para desarrollar las vistas de la aplicación.

#### CSS

CSS, acrónimo de Cascading Style Sheets, es un lenguaje de hojas de estilo desarrollado en 1996 por la W3C que permite añadir estilos a documentos escritos con un lenguaje de marcado, como es el caso de HTML. Las reglas CSS utilizadas para aplicar estos

estilos pueden ser incrustadas directamente en el documento o bien pueden ser escritas en un fichero independiente, permitiendo con esta última opción separar el contenido de la representación visual. Por este motivo se utilizará CSS para definir la presentación de las vistas.

## JavaScript

JavaScript, creado por Brendan Eich en Netscape en 1995 y estandarizado en 1996 dentro de ECMAScript, es un lenguaje de programación orientado a objetos, basado en prototipos, débilmente tipado e interpretado usado principalmente en los navegadores web para definir comportamiento e interactividad a un sitio web. Se empleará JavaScript para validar formularios y modificar elementos (Dom Scripting) en el lado cliente sin necesidad de enviar una petición al servidor.

## Ajax

Ajax (Asynchronous JavaScript and XML) es una técnica que permite, mediante programas escritos en JavaScript, que un navegador y un servidor intercambien información en formato XML, JSON, HTML... de forma asíncrona y actualizar elementos específicos de una página web sin que esta se tenga que recargar completamente. Se empleará Ajax para el sistema de notificaciones en tiempo real.

## SQL

SQL, acrónimo de Structured Query Language, es un lenguaje estándar de programación creado en la década de 1970 para administrar bases de datos relacionales y operar con sus datos. Se ha elegido este lenguaje por su facilidad para integrarse en lenguajes de programación como PHP.

## UML

UML (Unified Modeling Language) es un lenguaje de modelado visual estándar que se utiliza para describir, especificar, diseñar y documentar la estructura y el comportamiento de un sistema software y de los objetos (artefectos) que lo componen. A pesar de tratarse de un lenguaje aplicable en las principales metodologías de desarrollo de software, encaja mejor en metodologías de desarrollo iterativos, incrementales y dirigidos por casos de uso, como es el caso de la metodología RUP. UML cuenta con varios tipos de diagramas. En este proyecto se usarán tres tipos: el diagrama de casos de uso para la definición de funcionalidades y requisitos, el diagrama de clases para representar la información estática del sistema, y el diagrama de secuencia del sistema para representar la interacción entre los distintos artefactos.



## 6.2 Productos de Terceros

### XAMPP

XAMPP es una distribución de Apache, fácilmente instalable en Linux, Windows y Mac OS, en donde se incluye varios softwares libres como Apache, MySQL/MariaDB, PHP, o phpMyAdmin. De esta forma, XAMPP nos permite contar con la mayor parte del software necesario para desarrollar aplicaciones web localmente en nuestro equipo.

### MySQL

MySQL es un sistema de gestión de bases de datos relacionales que cuenta con dos licencias: una de código abierto y otra comercial gestionada por Oracle. Al ser código abierto y, por lo tanto, fácilmente accesible, ganó una gran popularidad en el ámbito de desarrollo web. MySQL destaca por su facilidad de instalación y configuración, por su velocidad en las operaciones y su alto rendimiento, y por su entorno de seguridad y encriptación.

### Apache HTTP Server

Apache es un software de servidor web gratuito y de código abierto desarrollado por la Apache Software Foundation con una estructura basada en módulos. Su función es establecer conexiones entre los navegadores y el servidor (arquitectura cliente-servidor) mientras se transmiten los ficheros a través del protocolo HTTP.

### Bootstrap

Bootstrap es un framework de desarrollo web para la elaboración de proyectos responsivos en HTML, CSS y JavaScript. Cuenta con un sistema de cuadrícula que nos permite colocar fácilmente los elementos de la página web, con componentes prediseñados como formularios o modales, y con complementos creados en JQuery para añadir interactividad a las páginas. Todo esto hace que Bootstrap nos permita diseñar sitios web muy rápidamente y consistentes entre navegadores. En este proyecto, se empleará Bootstrap en el diseño de las interfaces.

### PhpStorm

PhpStorm es un IDE desarrollado por JetBrains y basado en Java para el desarrollo de aplicaciones web en PHP. Soporta frameworks PHP como Laravel, WordPress o CakePHP, posee un asistente de código inteligente para PHP y para lenguajes típicos en

el desarrollo web, como HTML, CSS y JavaScript, proporciona soportes para bases de datos e integra Git y GitHub.

### Visual Paradigm

Visual Paradigm es una herramienta de diseño de software para modelar sistemas de información empresarial y gestionar procesos de desarrollo de software. Soporta lenguajes de modelo como UML, BPM o SysML y provee un conjunto de herramientas software para la captura de requisitos, diseño de sistemas y bases de datos, etc. Se utilizará este software para el desarrollo de los diagramas de casos de uso, de clases y de secuencia.

### VISIO

VISIO es una herramienta de diagramación desarrollada por Microsoft para crear diagramas e imágenes vectoriales. Permite desarrollar diagramas de flujo, organigramas, diagramas UML, de bases de datos, etc. Se utilizará VISIO para crear el modelo entidad-relación.

## 7 Especificación y Análisis de Requisitos

### 7.1 Requisitos No Funcionales

Los requisitos no funcionales son aquellos que describen propiedades (de seguridad, de rendimiento, de apariencia...) del sistema. No se centran en qué debe hacer la aplicación, sino en cómo debe hacerlo. Los requisitos no funcionales que debe cumplir la aplicación se detallan a continuación:

- **RNF1:** El diseño de las interfaces debe ser responsive, de forma en que el contenido de la web se adapte, en la medida de lo posible, al tamaño de pantalla del dispositivo cliente.
- **RNF2:** El diseño de la aplicación debe ser sencillo, de forma en que su uso sea fácilmente entendible para el usuario.
- **RNF3:** Las contraseñas se enviarán y almacenarán en el servidor cifradas empleando un algoritmo hash. De esta forma, para validar la identidad de un usuario en el sistema, se comparará el hash recibido con el hash almacenado en base de datos.

- **RNF4:** La aplicación debe ser multiidioma, permitiendo al usuario elegir entre el castellano y el inglés. La traducción se hará dinámicamente en el front de la aplicación, y el idioma elegido por el usuario debe mantenerse cuando este cambie de página.

## 7.2 Requisitos Funcionales

Los requisitos funcionales son aquellos que describen servicios del sistema, es decir, describen cómo se comporta la aplicación ante determinadas acciones.

A continuación, se describirán los requisitos funcionales a considerar en las distintas iteraciones de la fase de construcción. La aplicación interactuará con cinco tipos de actores: El administrador, el responsable de la organización, el responsable del edificio, el usuario registrado y el usuario público. El administrador tendrá acceso a todas las funcionalidades, por lo que a la hora de describirlas no se mencionará al administrador a no ser que sea el único usuario capaz de realizarla.

### 7.2.1 Iteración 6. Gestión de Usuarios

En esta iteración se implementará la **gestión de usuarios (RF1)**. Las funcionalidades que se deben incluir para esta gestión son las siguientes:

- **RF1.1. Listar Usuarios:** Cualquier usuario identificado en la aplicación podrá, a través del panel de gestión, listar los usuarios registrados en el sistema.
- **RF1.2. Consultar Detalles del Usuario:** Cualquier usuario identificado en la aplicación podrá consultar los datos (excepto la contraseña) de un usuario registrado en el sistema.
- **RF1.3. Añadir Usuario:** El administrador podrá crear nuevos usuarios en el sistema con los roles de administrador, responsable de organización o usuario registrado. La aplicación asignará automáticamente el rol de responsable de edificio a un usuario en el momento en que se le asigne un edificio, por lo que no se podrá crear un usuario desde cero con este rol. Para llevar a cabo el registro será necesario informar dni, nombre de usuario, contraseña, nombre, apellidos, rol, email, teléfono y, opcionalmente, se podrá adjuntar una foto de perfil.
- **RF1.4. Login:** Proceso de inicio de sesión en sistema mediante dni y password.
- **RF1.5. Eliminar Usuario:** El administrador podrá eliminar, previa confirmación, a otros usuarios del sistema. El usuario a eliminar no podrá ser

responsable de ningún edificio, y para eliminar un responsable de organización o un administrador se deberá cumplir que exista al menos otro usuario con ese rol en el sistema.

- **RF1.6. Editar Usuario:** El administrador podrá editar cualquier dato de los usuarios, exceptuando el nombre de usuario. Se podrá cambiar el rol de un usuario administrador, responsable de organización (cumpliendo las mismas restricciones que las mencionadas en RF1.5. Eliminar Usuario), usuario registrado y responsable de edificio, siempre que este tenga/no tenga edificios asignados.
- **RF1.7. Editar Perfil:** Los usuarios identificados en la aplicación podrán modificar los campos contraseña, email, teléfono y foto de perfil de su propio usuario (Editar Perfil).
- **RF1.8. Buscar Usuarios:** Los usuarios identificados en la aplicación podrán buscar a otros usuarios informando de uno o varios campos. Los campos por los que se permite realizar la búsqueda son DNI, nombre de usuario, nombre, apellidos, teléfono, email y rol.

## 7.2.2. Iteración 7. Gestión de Edificios

En esta iteración se implementarán las funcionalidades necesarias para la **gestión de edificios (RF2), de plantas (RF3) y de espacios (RF4)**. A continuación, se detallarán los requisitos considerados en cada una de estas gestiones.

### 7.2.2.1. Gestión de Edificios (RF2)

- **RF2.1. Listar Edificios:** El responsable de la organización podrá listar todos los edificios registrados en el sistema. El responsable del edificio sólo podrá listar aquellos edificios de los que sea responsable.
- **RF2.2. Añadir Edificio:** El responsable de la organización podrá definir edificios en el sistema. Para ello debe adjuntar el nombre del edificio, la calle, la ciudad, la provincia, el código postal, el número de teléfono, el número de fax y, opcionalmente, adjuntar una foto del edificio. También deberá asignar un responsable de edificio. Los usuarios candidatos para ser el responsable de un edificio serán aquellos con el rol de Usuario Registrado o Responsable de Edificio. Al asignar un edificio a un usuario con el rol de Usuario Registrado, la

aplicación modificará automáticamente el rol de este usuario al de Responsable del Edificio.

- **RF2.3. Eliminar Edificio:** El responsable de la organización podrá eliminar, previa confirmación, los edificios del sistema. Al eliminar un edificio, automáticamente se eliminarán las plantas y espacios asociados. Para eliminar un edificio, este no podrá tener asignado ningún plan. Si al eliminar el edificio su responsable no tiene más edificios asignados, la aplicación modificará automáticamente su rol a Usuario Registrado.
- **RF2.4. Editar Edificio:** El responsable de la organización podrá modificar cualquiera de los datos del edificio, excepto el ID del edificio. También podrá cambiar el responsable del edificio. Si el nuevo usuario asignado tiene el rol de Usuario Registrado, la aplicación modificará su rol automáticamente a Responsable del Edificio. Si el antiguo responsable, tras el cambio de responsable, no tiene más edificios asignados, la aplicación modificará automáticamente su rol a Usuario Registrado.
- **RF2.5. Buscar Edificio:** El responsable de la organización y el responsable del edificio podrán buscar edificios informando uno o varios campos. Los campos por los que se permite buscar son el nombre, la calle, la ciudad, la provincia, el código postal, el teléfono y el fax. A mayores, el responsable de la organización podrá buscar por responsable de edificio. La aplicación mostrará al responsable de la organización todos los edificios que coinciden con los campos especificado, mientras que el responsable del edificio sólo obtendrá resultados de aquellos edificios que coincidan con los criterios de búsqueda y de los que sea responsable.
- **RF2.6. Consultar Detalles del Edificio:** El responsable de la organización podrá consultar los detalles de cualquiera de los edificios, mientras que el responsable del edificio sólo podrá consultar los detalles de aquellos edificios de los que sea responsable.
- **RF2.7. Acceder al Portal del Edificio:** El usuario público podrá acceder al portal de un edificio. Para ello, la aplicación mostrará las distintas ciudades en las que haya edificios registrados. Al seleccionar una ciudad, se mostrará un listado de los edificios existentes en esa ciudad. Seleccionar un edificio de este listado dará acceso a su portal.
- **RF2.8. Consultar el responsable de un Portal:** El usuario público podrá acceder, desde el portal de un edificio, a la información de su responsable.

#### 7.2.2.2. Gestión de Plantas (RF3)

- **RF3.1. Consultar Plantas:** El responsable de la organización podrá consultar todas las plantas registradas en la aplicación. El responsable del edificio sólo podrá consultar las plantas de los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las plantas asociadas a ese edificio.
- **RF3.2. Añadir Planta:** El responsable de la organización podrá añadir plantas en el sistema. Para ello deberá informar de los campos nombre, número y descripción de la planta, así como el edificio al que pertenece.
- **RF3.3. Eliminar Planta:** El responsable de la organización podrá eliminar, previa confirmación, las plantas registradas en el sistema. Para eliminar una planta, se deben eliminar previamente las implementaciones de rutas asociadas.
- **RF3.4. Editar Planta:** El responsable de la organización podrá modificar cualquiera de los campos de las plantas registradas en el sistema.

#### 7.2.2.3. Gestión de Espacios (RF4)

- **RF4.1. Consultar Espacios:** El responsable de la organización podrá consultar todos los espacios registrados en la aplicación. El responsable del edificio sólo podrá consultar los espacios asociados a las plantas de los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar los espacios asociados a las plantas de ese edificio.
- **RF4.2. Añadir Espacio:** El responsable de la organización podrá añadir espacios en el sistema. Para ello deberá informar de los campos nombre, dimensiones, descripción, e informar de la planta a la que pertenece. Opcionalmente, también se podrá añadir una foto del espacio.
- **RF4.3. Eliminar Espacio:** El responsable de la organización podrá eliminar, previa confirmación, los espacios registrados en el sistema.
- **RF4.4. Editar Espacio:** El responsable de la organización podrá modificar cualquiera de los campos de los espacios registrados en el sistema.

### 7.2.3. Iteración 8. Gestión de Planes de Prevención

En esta iteración se implementarán las funcionalidades necesarias para **definir planes (RF5), documentos (RF6), procedimientos (RF7), rutas (RF8), formaciones (RF9) y simulacros (RF10)**. Los requisitos considerados son los siguientes.

#### 7.2.3.1. Definición de Planes (RF5)

- **RF5.1. Consultar Planes:** El responsable de la organización podrá consultar todos los planes registrados en la aplicación. El responsable del edificio sólo podrá consultar los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar los planes en asignados a ese edificio que se encuentren en estado “implementado”.
- **RF5.2. Añadir Plan:** El responsable de la organización podrá añadir planes en el sistema. Para ello deberá informar de los campos nombre, descripción e instrucciones del plan.
- **RF5.3. Eliminar Plan:** El responsable de la organización podrá eliminar, previa confirmación, los planes registrados en el sistema. Para eliminar un plan, se deberán eliminar previamente las asignaciones que tenga a los edificios y las asociaciones con los documentos, procedimientos, rutas, formaciones y simulacros.
- **RF5.5. Editar Plan:** El responsable de la organización podrá modificar cualquiera de los campos de un plan.
- **RF5.6. Asignar Plan a Edificios:** El responsable de la organización podrá asignar planes a edificios. Para ello, el plan en cuestión debe tener asociado al menos un elemento de tipo documento. Al crear la asignación, esta se marcará por defecto con el estado “pendiente” y se insertará una notificación a los responsables de ese edificio.
- **RF5.7. Eliminar Plan de Edificio:** El responsable de la organización podrá eliminar la asociación de un plan con un edificio. Esta acción provocará que las implementaciones en ese edificio de los documentos, procedimientos, rutas, formaciones y simulacros asociados al plan pasen a estado “vencido”. Las implementaciones en este estado siguen estando disponibles para consultas, pero no podrán ser modificadas.
- **RF5.8. Reasignar Plan a Edificios:** El responsable de la organización podrá reasignar planes que se encuentren en estado “implementado” a edificios. Esta

acción provocará que el estado de la asignación cambie a estado “pendiente” (generando una nueva notificación a los responsables asignados a ese edificio), y que el estado de las implementaciones de documentos, procedimientos, rutas, formaciones y simulacros asociados al plan pasen a estado “vencido”.

#### 7.2.3.2. Definición de Documentos (RF6)

- **RF6.1. Consultar Documentos:** El responsable de la organización podrá consultar todos los documentos registrados en la aplicación. El responsable del edificio sólo podrá consultar los documentos de los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar los documentos de los planes asignados a ese edificio que se encuentren en estado “implementado”.
- **RF6.2. Añadir Documento:** El responsable de la organización podrá añadir documentos al sistema. Para ello deberá informar de los campos título, descripción, comentario y el plan al que se va a asociar.
- **RF6.3. Eliminar Documento:** El responsable de la organización podrá eliminar, previa confirmación, los documentos registrados en el sistema. Para eliminar un documento, se deberán eliminar previamente las implementaciones existentes en los distintos edificios.
- **RF6.4. Editar Documento:** El responsable de la organización podrá modificar cualquiera de los campos de un documento.

#### 7.2.3.3. Definición de Procedimientos (RF7)

- **RF7.1. Consultar Procedimientos:** El responsable de la organización podrá consultar todos los procedimientos registrados en la aplicación. El responsable del edificio sólo podrá consultar los procedimientos de los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar los procedimientos de los planes asignados a ese edificio que se encuentren en estado “implementado”.
- **RF7.2. Añadir Procedimiento:** El responsable de la organización podrá añadir procedimientos al sistema. Para ello deberá informar de los campos título, descripción, comentario y el plan al que se va a asociar.
- **RF7.3. Eliminar Procedimiento:** El responsable de la organización podrá eliminar, previa confirmación, los procedimientos registrados en el sistema. Para



eliminar un procedimiento, se deberán eliminar previamente las implementaciones existentes en los distintos edificios.

- **RF7.8. Editar Procedimiento:** El responsable de la organización podrá modificar cualquiera de los campos de un procedimiento.

#### 7.2.3.4. Definición de Rutas (RF8)

- **RF8.1. Consultar Rutas:** El responsable de la organización podrá consultar todas las rutas registradas en la aplicación. El responsable del edificio sólo podrá consultar las rutas de los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las rutas de los planes asignados a ese edificio que se encuentren en estado “implementado”.
- **RF8.2. Añadir Ruta:** El responsable de la organización podrá añadir rutas al sistema. Para ello deberá informar de los campos título, descripción, comentario y el plan al que se va a asociar.
- **RF8.3. Eliminar Ruta:** El responsable de la organización podrá eliminar, previa confirmación, las rutas registradas en el sistema. Para eliminar una ruta, se deberán eliminar previamente las implementaciones existentes en las distintas plantas.
- **RF8.4. Editar Ruta:** El responsable de la organización podrá modificar cualquiera de los campos de una ruta.

#### 7.2.3.5. Definición de Formaciones (RF9)

- **RF9.1. Consultar Formaciones:** El responsable de la organización podrá consultar todas las formaciones registradas en la aplicación. El responsable del edificio sólo podrá consultar las formaciones de los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las formaciones de los planes asignados a ese edificio que se encuentren en estado “implementado”.
- **RF9.2. Añadir Formación:** El responsable de la organización podrá añadir formaciones al sistema. Para ello deberá informar de los campos título, descripción, comentario y el plan al que se va a asociar.
- **RF9.3. Eliminar Formación:** El responsable de la organización podrá eliminar, previa confirmación, las formaciones registradas en el sistema. Para eliminar una

formación, se deberán eliminar previamente las implementaciones existentes en los distintos edificios.

- **RF9.4. Editar Formación:** El responsable de la organización podrá modificar cualquiera de los campos de una formación.

#### 7.2.3.6. Definición de Simulacros (RF10)

- **RF10.1. Consultar Simulacros:** El responsable de la organización podrá consultar todos los simulacros registrados en la aplicación. El responsable del edificio sólo podrá consultar los simulacros de los planes asociados a los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar los simulacros de los planes asignados a ese edificio que se encuentren en estado “implementado”.
- **RF10.2. Añadir Simulacro:** El responsable de la organización podrá añadir simulacros al sistema. Para ello deberá informar de los campos título, descripción, comentario y el plan al que se va a asociar.
- **RF10.3. Eliminar Simulacro:** El responsable de la organización podrá eliminar, previa confirmación, los simulacros registrados en el sistema. Para eliminar un simulacro, se deberán eliminar previamente las implementaciones existentes en los distintos edificios.
- **RF10.4. Editar Simulacro:** El responsable de la organización podrá modificar cualquiera de los campos de un simulacro.

#### 7.2.4. Iteración 9. Implementación de Planes de Prevención.

En esta iteración se desarrollarán las funcionalidades necesarias para **implementar documentos (RF11), procedimientos (RF12), rutas (RF13), formaciones (RF14) y simulacros (RF15)**. Los requisitos considerados son los siguientes.

##### 7.2.4.1. Implementación de Documentos en Edificios (RF11)

- **RF11.1. Consultar implementación de Documentos:** El responsable de la organización podrá consultar todas implementaciones de documentos registrados en la aplicación. El responsable del edificio sólo podrá consultar las implementaciones de documentos en los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las implementaciones de documentos en estado “vigente” en ese edificio.

- **RF11.2. Implementar Documento.** El responsable de la organización podrá añadir una implementación de un documento en cualquier edificio que tenga asignado el plan al que pertenece el documento, mientras que el responsable del edificio sólo podrá hacerlo en los edificios de los que sea responsable. Para implementar un documento en un edificio, se debe añadir un documento en formato pdf. Al añadir una implementación de un documento, esta se almacena con estado “vigente”. Sólo puede existir una implementación de un documento en un edificio en estado “vigente”.
- **RF11.3. Editar Implementación de Documento.** Se podrá modificar el documento pdf adjuntado (sustituyéndolo por otro) en aquellas implementaciones de documentos que se encuentre en estado “vigente”. El responsable de la organización podrá realizar esta acción en cualquier edificio, mientras que el responsable del edificio sólo podrá modificar implementaciones de documentos en los edificios que tenga asignados.
- **RF11.4. Eliminar implementación de Documento:** El responsable de la organización podrá eliminar implementaciones de documentos en cualquier edificio, independientemente del estado en que se encuentre. El responsable del edificio podrá realizar la misma gestión en los edificios que tenga asignados. Para eliminar una implementación de un documento, el plan al que pertenece no puede estar asignado al edificio donde se está borrando la implementación.

#### 7.2.4.2. Implementación de Procedimientos en Edificios (RF12)

- **RF12.1. Consultar implementación de Procedimientos.** El responsable de la organización podrá consultar todas implementaciones de procedimientos registrados en la aplicación. El responsable del edificio sólo podrá consultar las implementaciones de procedimientos en los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las implementaciones de procedimientos en estado “vigente” en ese edificio.
- **RF12.2. Implementar Procedimiento.** El responsable de la organización podrá añadir una implementación de un procedimiento en cualquier edificio que tenga asignado el plan al que pertenece el procedimiento, mientras que el responsable del edificio sólo podrá hacerlo en los edificios de los que sea responsable. Para implementar un procedimiento en un edificio, se debe añadir un documento en formato pdf. Al añadir una implementación de un documento, esta se almacena con estado “vigente”. Sólo puede existir una implementación de un procedimiento en un edificio en estado “vigente”.

- **RF12.3. Editar implementación de Procedimiento:** Se podrá modificar el documento pdf adjuntado (sustituyéndolo por otro) en aquellas implementaciones de procedimientos que se encuentre en estado “vigente”. El responsable de la organización podrá realizar esta acción en cualquier edificio, mientras que el responsable del edificio sólo podrá modificar implementaciones de procedimientos en los edificios que tenga asignados.
- **RF12.4. Eliminar implementación de Procedimiento:** El responsable de la organización podrá eliminar implementaciones de procedimientos en cualquier edificio, independientemente del estado en que se encuentre. El responsable del edificio podrá realizar la misma gestión en los edificios que tenga asignados. Para eliminar una implementación de un procedimiento, el plan al que pertenece no puede estar asignado al edificio donde se está borrando la implementación.

#### 7.2.4.3. Implementación de Rutas en Plantas (RF13)

- **RF13.1. Consultar implementación de Rutas en Plantas.** El responsable de la organización podrá consultar todas implementaciones de rutas registrados en la aplicación. El responsable del edificio sólo podrá consultar las implementaciones de rutas en las plantas de los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las implementaciones de rutas en estado “vigente” en las plantas de ese edificio.
- **RF13.2. Implementar Ruta.** El responsable de la organización podrá añadir una implementación de una ruta en las plantas de cualquier edificio que tenga asignado el plan al que pertenece la ruta, mientras que el responsable del edificio sólo podrá hacerlo en las plantas de los edificios de los que sea responsable. Para implementar una ruta en una planta, se debe añadir un documento en formato pdf. Al añadir una implementación de una ruta en una planta, esta se almacena con estado “vigente”. Sólo puede existir una implementación de una ruta en una planta en estado “vigente”. La aplicación sólo considerará una ruta implementada en un edificio cuando exista una implementación de esa ruta en cada una de las plantas.
- **RF13.3. Editar implementación de Ruta:** Se podrá modificar el documento pdf adjuntado (sustituyéndolo por otro) en aquellas implementaciones de rutas que se encuentren en estado “vigente”. El responsable de la organización podrá realizar esta acción en cualquier planta, mientras que el responsable del edificio sólo podrá modificar implementaciones de rutas en las plantas de los edificios que tenga asignados.

- **RF13.4. Eliminar implementación de Ruta:** El responsable de la organización podrá eliminar implementaciones de rutas en cualquier planta, independientemente del estado en que se encuentre. El responsable del edificio podrá realizar la misma gestión en las plantas de los edificios que tenga asignados. Para eliminar una implementación de una ruta, el plan al que pertenece no puede estar asignado al edificio de las plantas donde se está borrando la implementación.

#### 7.2.4.4. Implementación de Formaciones en Edificios (RF14)

- **RF14.1. Consultar implementación de Formaciones:** El responsable de la organización podrá consultar todas implementaciones de formaciones registrados en la aplicación. El responsable del edificio sólo podrá consultar las implementaciones de formaciones en los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las implementaciones de formaciones en estado “vigente” en ese edificio.
- **RF14.2. Implementar Formación:** El responsable de la organización podrá añadir una implementación de una formación en cualquier edificio que tenga asignado el plan al que pertenece la formación, mientras que el responsable del edificio sólo podrá hacerlo en los edificios de los que sea responsable. Para implementar una formación en un edificio, se debe añadir la fecha en la que se quiere planificar la formación, los destinatarios de la misma y, opcionalmente, un documento pdf y una url a un recurso. Al añadir una implementación de una formación, esta se almacena con estado “vigente”. Sólo puede existir una implementación de una formación en un edificio en estado “vigente”.
- **RF14.3. Editar implementación de Formación:** Se podrá modificar la fecha de planificación, el documento pdf, o la url del recurso en aquellas implementaciones de formaciones que se encuentre en estado “vigente”. El responsable de la organización podrá realizar esta acción en cualquier edificio, mientras que el responsable del edificio sólo podrá modificar implementaciones de formaciones en los edificios que tenga asignados.
- **RF14.4. Eliminar implementación de Formación:** El responsable de la organización podrá eliminar implementaciones de formaciones en cualquier edificio, independientemente del estado en que se encuentre. El responsable del edificio podrá realizar la misma gestión en los edificios que tenga asignados. Para eliminar una implementación de una formación, el plan al que pertenece no puede estar asignado al edificio donde se está borrando la implementación.

#### 7.2.4.5. Implementación de Simulacros en Edificios (RF15)

- **RF15.1. Consultar implementación de Simulacros:** El responsable de la organización podrá consultar todas implementaciones de simulacros registrados en la aplicación. El responsable del edificio sólo podrá consultar las implementaciones de simulacros en los edificios que tenga asignados. El usuario público podrá, desde el portal de un edificio, consultar las implementaciones de simulacros en estado “vigente” en ese edificio.
- **RF15.2. Implementar Simulacro:** El responsable de la organización podrá añadir una implementación de un simulacro en cualquier edificio que tenga asignado el plan al que pertenece el simulacro, mientras que el responsable del edificio sólo podrá hacerlo en los edificios de los que sea responsable. Para implementar un simulacro en un edificio, se debe añadir la fecha en la que se quiere planificar el simulacro, los destinatarios de la misma y, opcionalmente, una url a un recurso y el resultado del simulacro. Al añadir una implementación de un simulacro, esta se almacena con estado “vigente”. Sólo puede existir una implementación de una formación en un edificio en estado “vigente”.
- **RF15.3. Editar implementación de Simulacro:** Se podrá modificar la fecha de planificación, la url del recurso, los destinatarios o el resultado en aquellas implementaciones de simulacros que se encuentren en estado “vigente”. El responsable de la organización podrá realizar esta acción en cualquier edificio, mientras que el responsable del edificio sólo podrá modificar implementaciones de simulacros en los edificios que tenga asignados.
- **RF15.4. Eliminar implementación de Simulacro:** El responsable de la organización podrá eliminar implementaciones de simulacros en cualquier edificio, independientemente del estado en que se encuentre. El responsable del edificio podrá realizar la misma gestión en los edificios que tenga asignados. Para eliminar una implementación de un simulacro, el plan al que pertenece no puede estar asignado al edificio donde se está borrando la implementación.

#### 7.2.4.6. Verificación de Plan Implementado (RF16)

Siempre que se ejecuten con éxito los casos de uso **Implementar Documento (RF11.2)**, **Implementar Procedimiento (RF12.2)**, **Implementar Ruta (RF13.2)**,

**Implementar Formación (RF14.2) o Implementar Simulacro (RF15.2)**, la aplicación comprobará si el plan asociado al elemento se encuentra completamente implementado en ese edificio. Para ello, se comprobarán si existe una implementación en estado “vigente” para cada uno de los documentos, procedimientos, rutas, formaciones y simulacros que componen el plan (en caso de las rutas, debe existir una implementación en estado “vigente” para cada una de las plantas del edificio). Si se cumple esta condición, el estado de la asignación entre el plan y el edificio pasará a estado “implementado”, y se insertará una notificación a los responsables de tipo organización. Los planes asignados en estado “implementado” se mostrarán en el portal del edificio.

#### 7.2.5. Iteración 10. Sistema de Notificaciones.

En esta iteración se implementará la **gestión de Notificaciones (RF17)**. Los requisitos funcionales que se consideran en esta gestión son los siguientes.

- **RF17.1. Consultar Notificaciones.** Los usuarios registrados en el sistema podrán listar las notificaciones que tengan asignadas.
- **RF17.2. Eliminar Notificaciones.** Los usuarios registrados en el sistema podrán eliminar las notificaciones que tengan asignadas.

### 7.3 Casos de Uso

#### 7.3.1 Diagramas de Casos de Uso

En los siguientes apartados se mostrarán las figuras correspondientes a los diagramas de casos de uso del sistema separados por iteraciones. Cada figura representa los casos de uso que participan en esa iteración.

##### 7.3.1.1 Iteración 6 – Gestión de Usuarios

Los siguientes casos de uso representan las funcionalidades derivadas del requisito funcional RF1 (Gestión de Usuarios).

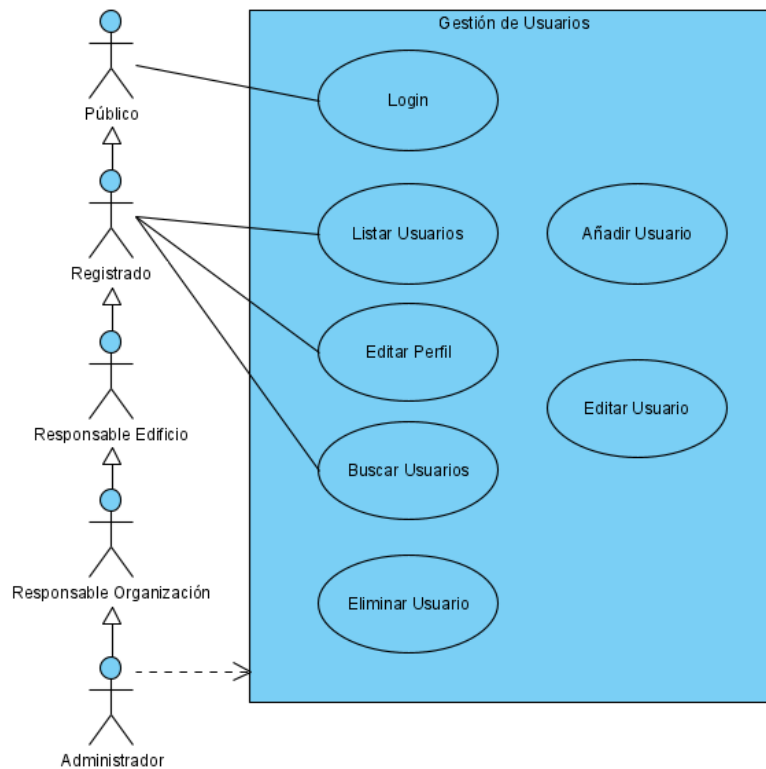


Figura 1 Casos de Uso - Gestión de Usuarios

### 7.3.1.2 Iteración 7 – Gestión de Edificios

Los siguientes casos de uso representan las funcionalidades derivadas de los requisitos funcionales RF2 (Gestión de Edificios), RF3 (Gestión de Plantas) y RF4 (Gestión de Espacios)

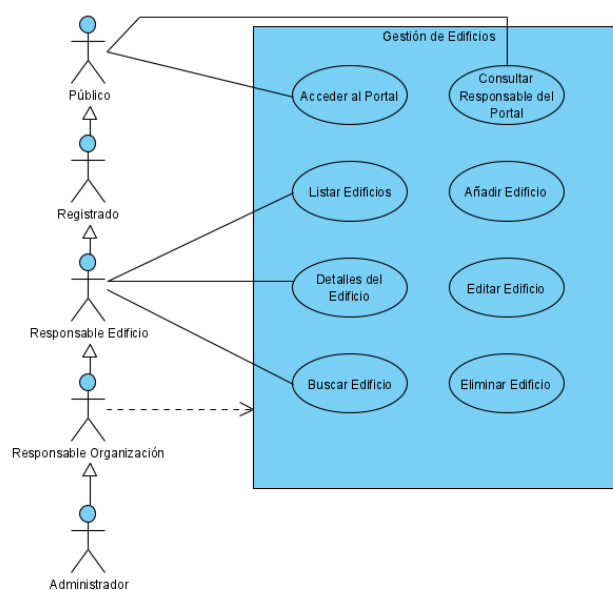


Figura 2 Casos de Uso - Gestión de Edificios



## 7.3.2 Descripción de Casos de Uso

La especificación de los casos de uso se detallará por iteraciones y por gestiones respectivamente.

### 7.3.2.1 Iteración 6. Gestión de Usuarios

- Listar Usuarios

**ID Caso de Uso:** Listar Usuarios.

**Actor:** Registrado, Responsable Edificio, Responsable Organización, Administrador

**Breve descripción:** Los usuarios identificados en la aplicación podrán consultar los usuarios registrados. Cumple el requisito funcional RF1.1

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Ninguna.

**Flujo normal de eventos:**

**Usuario:** Solicita consultar usuarios.

**Aplicación:** Recupera los usuarios registrados y los muestra al usuario.

- Consultar Detalles del Usuario

**ID Caso de Uso:** Consultar Detalles del Usuario.

**Actor:** Registrado, Responsable Edificio, Responsable Organización, Administrador

**Breve descripción:** Los usuarios que hayan iniciado sesión podrán acceder a los detalles de un usuario para consultar todos sus datos, excepto la contraseña. Cumple el requisito funcional RF1.2

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Ninguna.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita acceder a los detalles de un usuario.

**Aplicación:** 2. Recupera información del usuario solicitado y lo muestra al usuario (2a).

**Flujo alternativo de eventos:**

2a. Si el usuario del que se solicitan los detalles no existe.

1. Se muestra mensaje indicando el usuario solicitado no existe.

- Añadir Usuario

**ID Caso de Uso:** Añadir Usuario.

**Actor:** Administrador

**Breve descripción:** El Administrador podrá añadir nuevos usuarios al sistema. Cumple con el requisito funcional RF1.3

**Precondiciones:** El Administrador podrá crear nuevos usuarios en el sistema con los roles de administrador, responsable de organización o usuario registrado. La aplicación asignará automáticamente el rol de responsable de edificio a un usuario en el momento en que se le asigne un edificio, por lo que no se podrá crear un usuario desde cero con este rol

**Postcondiciones:** Se añade el nuevo usuario al sistema.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita añadir un nuevo usuario.

**Aplicación:** 2. Muestra la vista de añadir usuario.

**Usuario:** 3. Introduce los datos del nuevo usuario.

**Aplicación:** 4. Valida los datos introducidos, verifica que los atributos únicos no existan en el sistema, comprueba que el rol introducido no sea el de responsable de edificio y sube la foto de perfil en caso de que haya sido informada (4a).

**Aplicación:** 5. Almacena los datos del nuevo usuario y envía un mensaje confirmando la acción.

**Flujo alternativo de eventos:**

4a. Los datos introducidos no son válidos, alguno de los atributos únicos ya existe en el sistema, el rol especificado es el de responsable de edificio o se produce un error al subir la foto de perfil al servidor.

1. Se muestra mensaje informando del error.

#### ▪ Login

**ID Caso de Uso:** Login.

**Actor:** Público

**Breve descripción:** Un usuario público puede identificarse en la aplicación. Cumple con el requisito funcional RF1.4

**Precondiciones:** Ninguna.

**Postcondiciones:** El usuario se identifica en la aplicación.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita iniciar sesión.

**Aplicación:** 2. Muestra la vista de login.

**Usuario:** 3. Introduce el nombre de usuario y la contraseña.

**Aplicación:** 4. Valida que los datos introducidos corresponden con un usuario registrado en sistema (4a).

**Aplicación:** 5. El usuario inicia sesión en la aplicación.

**Flujo alternativo de eventos:**

4a. Los datos introducidos no se corresponden con ningún usuario registrado.

1. Se muestra mensaje informando de que los datos introducidos no corresponden con ningún usuario.

2. Vuelve al paso 2.

#### ▪ Eliminar Usuario

**ID Caso de Uso:** Eliminar Usuario.

**Actor:** Administrador.

**Breve descripción:** El administrador podrá eliminar, previa confirmación, a otros usuarios del sistema. El usuario a eliminar no podrá tener edificios asignados, y para eliminar un responsable de organización o un administrador se deberá cumplir que exista al menos otro usuario con ese rol en el sistema. Cumple el requisito funcional RF1.5

**Precondiciones:** El Administrador debe haber iniciado sesión.

**Postcondiciones:** Se elimina el usuario solicitado del sistema.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita eliminar un usuario.

**Aplicación:** 2. Recupera los datos del usuario, pide confirmación de la acción (2a).

**Usuario:** 3. Confirma la acción (3a).

**Aplicación:** 4. Verifica que el usuario no sea responsable de algún edificio, comprueba en el caso de que el usuario a eliminar sea responsable de organización o administrador que existan más usuarios con ese rol y elimina la foto de perfil si es que existe (4a).

**Aplicación:** 4. Elimina al usuario y muestra mensaje informando al usuario.

**Flujo alternativo de eventos:**

2a. El usuario no que se solicita eliminar no existe.

1. Muestra mensaje informando de que el usuario no existe.

3a. El usuario cancela la acción.

1. La aplicación redirige al usuario al listado de usuarios.

4a. El usuario que se quiere eliminar es responsable de un edificio, es responsable de organización o administrador y no existen más usuarios con ese rol, o se produce error al eliminar la foto de perfil.

2. La aplicación muestra un mensaje indicando el error.

## ▪ Editar Usuario

**ID Caso de Uso:** Editar Usuario.

**Actor:** Administrador.

**Breve descripción:** El administrador podrá editar cualquier dato de los usuarios, exceptuando el nombre de usuario. A la hora de modificar un rol, el cambio no puede provocar que no queden más administradores o responsables de organización en la aplicación. En el caso del rol de responsable de edificio, este sólo podrá ser asignado a usuarios que tengan edificios asignados, y desasignado a usuarios que no los tengan. Cumple el requisito funcional RF1.6

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Se modifican los datos del usuario solicitado.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita editar un usuario.

**Aplicación:** 2. Recupera los datos del usuario y los muestra en la vista de editar usuario (2a).

**Usuario:** 3. Introduce los nuevos datos.

**Aplicación:** 4. Valida los datos introducidos, comprueba que no se intenta asignar/desasignar el rol de responsable de edificio, en caso de que el rol a modificar sea responsable de organización o administrador, comprueba que existan más usuarios con ese rol, y modifica la foto de perfil en caso de que se haya adjuntado una nueva (4a).

**Aplicación:** 5. Modifica el usuario solicitado con los nuevos datos y muestra mensaje confirmando la operación.

**Flujo alternativo de eventos:**

2a. El usuario que se solicita editar no existe.

1. Muestra mensaje indicando que el usuario no existe.

4a. Los datos introducidos no son válidos, se solicita asignar/desasignar el rol de responsable de edificio, se intenta modificar el rol de responsable de organización o administrador y no hay más usuarios con ese rol, o se produce un error al modificar la foto de perfil.

1. Muestra mensaje informando del error.

## ▪ Editar Perfil

**ID Caso de Uso:** Editar Perfil.

**Actor:** Registrado, Responsable de Edificio, Responsable de Organización, Administrador.

**Breve Descripción:** Los usuarios identificados en la aplicación podrán modificar los campos contraseña, email, teléfono y foto de perfil de su propio usuario. Cumple el requisito funcional RF1.7.

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Se actualiza el usuario con los nuevos datos.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita editar su perfil.

**Aplicación:** 2. Recupera los datos del usuario y los muestra en la vista de editar perfil (2a).

**Usuario:** 3. Introduce los nuevos datos.

**Aplicación:** 4. Valida los nuevos datos, verifica que el usuario que se solicita editar corresponde con el registrado en sesión y modifica la foto de perfil en caso de que sea necesario (4a).

**Aplicación:** 5. Modifica los datos del usuario y envía un mensaje confirmando la acción.

**Flujo alternativo de eventos:**

2a. El usuario que se solicita editar no existe.

1. Muestra mensaje indicando que el usuario no existe.

4a. Los datos introducidos no son válidos, el usuario que se está intentando editar no corresponde con el que está registrado en sesión, o se produce un error al modificar la foto de perfil.

1. Se muestra al usuario un mensaje indicando el error.

## ▪ Buscar Usuarios

**ID Caso de Uso:** Buscar Usuarios.

**Actor:** Registrado, Responsable de Edificio, Responsable de Organización, Administrador.

**Breve Descripción:** Los usuarios identificados en la aplicación podrán buscar a otros usuarios informando de uno o varios campos. Los campos por los que se permite realizar la búsqueda son DNI, nombre de usuario, nombre, apellidos, teléfono, email y rol. Cumple con el requisito funcional RF1.8.

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Ninguna.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita buscar usuarios.

**Aplicación:** 2. Muestra la vista de buscar usuarios.

**Usuario:** 3. Introduce los datos sobre los cuales quiere realizar la búsqueda.

**Aplicación:** 4. Valida los datos introducidos, recupera los usuarios que coincidan con la información especificada y los muestra al usuario.

### 7.3.2.2 Iteración 7. Gestión de Sedes

#### 7.3.2.2.1 Gestión de Edificios

- Listar Edificios

**ID Caso de Uso:** Listar Edificios

**Actor:** Responsable de Edificio, Responsable de Organización, Administrador.

**Breve Descripción:** El responsable de la organización podrá listar todos los edificios registrados en el sistema. El responsable del edificio sólo podrá listar aquellos edificios de los que sea responsable. Cumple con el requisito funcional RF2.1.

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Ninguna.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita consultar Edificios.

**Aplicación:** 2. Recupera la información de los edificios y los muestra al usuario. En caso de que el usuario sea Responsable de Edificio, sólo se recuperará y mostrará la información de los edificios que tenga asignados.

- Añadir Edificio

**ID Caso de Uso:** Añadir Edificio.

**Actor:** Responsable de Organización, Administrador.

**Breve Descripción:** El responsable de la organización podrá definir edificios en el sistema y asignarlos a un usuario el cual se convertirá en su responsable. Los usuarios candidatos para ser el responsable de un edificio serán aquellos con el rol de Usuario Registrado o Responsable de Edificio. Al asignar un edificio a un usuario con el rol de Usuario Registrado, la aplicación modificará automáticamente el rol de este usuario al de Responsable del Edificio. Cumple con el requisito funcional RF2.2.

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Se añaden los datos del edificio al sistema. El rol del usuario responsable se modifica.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita añadir un Edificio.

**Aplicación:** 2. Muestra la vista de añadir edificio, incluyendo los usuarios candidatos a ser el responsable del nuevo edificio (Usuarios Registrados y Responsables de Edificio) (2a).

**Usuario:** 3. Añade los datos del edificio y selecciona a un responsable.

**Aplicación:** 4. Se validan los datos introducidos, comprueba que el usuario introducido es válido para ser el responsable de ese edificio, se sube la foto del edificio en caso de haber sido informada (4a).

**Aplicación:** 5. Se añade el edificio al sistema y se modifica el rol del nuevo responsable a Responsable de Edificio.

**Flujo alternativo de eventos:**

2a. No hay ningún usuario en sistema candidato a ser responsable de edificio.

1. Muestra mensaje informando al usuario.

4a. Los datos del edificio no son válidos, el usuario que se quiere asignar como responsable no cumple con los requisitos para serlo o se produce un error al subir la foto del edificio.

1. Muestra mensaje informando del error.

- Eliminar Edificio

**ID Caso de Uso:** Eliminar Edificio.

**Actores:** Responsable de Organización, Administrador.

**Breve Descripción:** El responsable de la organización podrá eliminar, previa confirmación, los edificios del sistema. Al eliminar un edificio, automáticamente se eliminarán las plantas y espacios asociados. Para eliminar un edificio, este no podrá tener asignado ningún plan. Si al eliminar el edificio su responsable no tiene más edificios asignados, la aplicación modificará automáticamente su rol a Usuario Registrado. Cumple con el requisito funcional RF2.3.

**Precondiciones:** El usuario debe haber iniciado sesión.

**Postcondiciones:** Se elimina el edificio del sistema. Se modifica el rol del responsable si no tiene más edificios asignados. Se eliminan las plantas y espacios asociados.

**Flujo normal de eventos:**

**Usuario:** 1. Solicita eliminar un edificio.

**Aplicación:** 2. Recupera los datos del edificio, solicita confirmación de la acción (2a).

**Usuario:** 3. Confirma la acción (3a).

**Aplicación:** 4. Comprueba que el edificio no tenga planes asignados (4a).

**Aplicación:** 5. Elimina los datos del edificio, elimina la foto del edificio y modifica el rol del usuario responsable en caso de que este no tenga otros edificios asignados.

**Flujo alternativo de eventos:**

2a. El edificio que se intenta eliminar no existe.

1. Muestra mensaje informando del error.

3a. El usuario cancela la acción.

1. Redirige a la vista de mostrar edificios.

4a. El edificio tiene planes asignados.

1. Muestra mensaje informando del error.

## 8 Diseño del Software (Estático y Dinámico)

### 8.1 Diseño del Software Estático

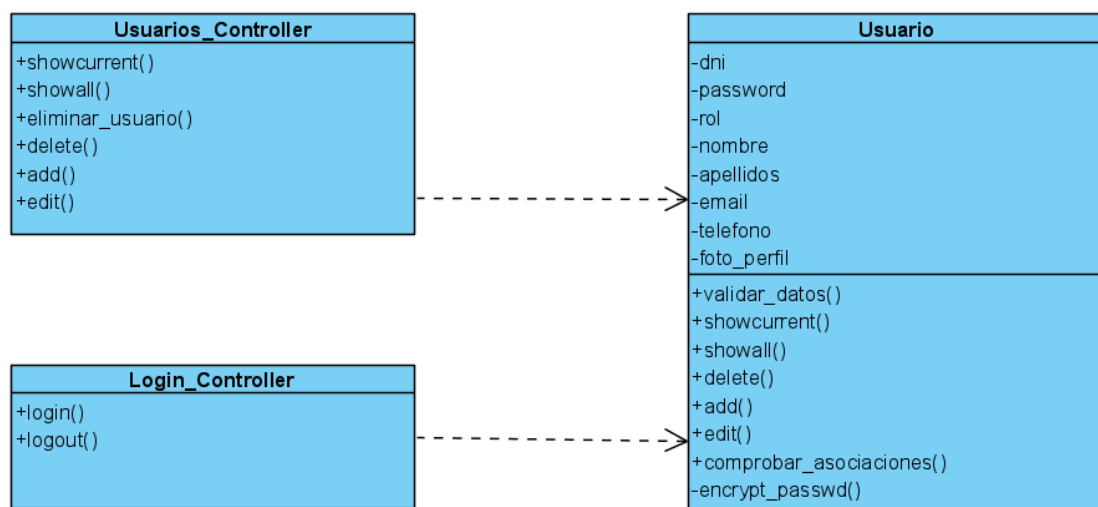


Diagrama de Clases - Iteración 6

## 8.2 Diseño del Software Dinámico

En este apartado se hace uso de los diagramas de secuencia para modelar cómo deben interactuar los objetos del sistema para cumplir los requisitos capturados en los distintos casos de uso del apartado 7. De la misma forma en que se hizo con los casos de uso, se agruparán los diagramas de secuencia por iteraciones.

### 8.2.1 Iteración 6. Gestión de Usuarios

#### ▪ Listar Usuarios

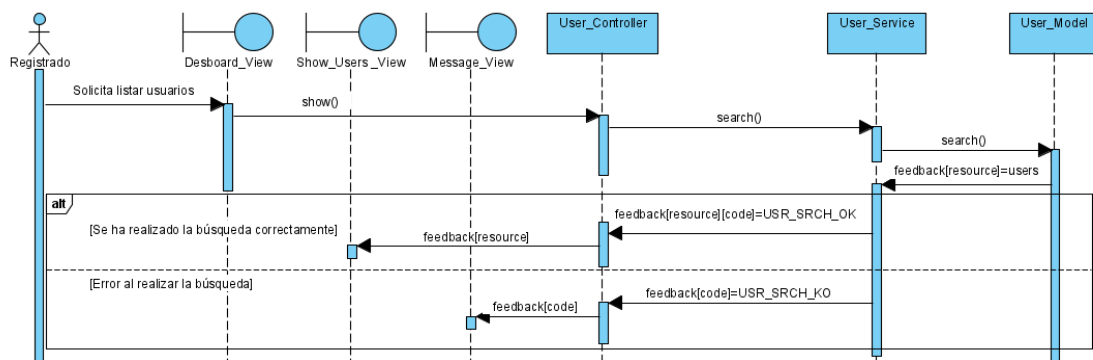


Figura 3 Diagrama de Secuencia - Listar Usuarios

#### ▪ Consultar Detalles del Usuario

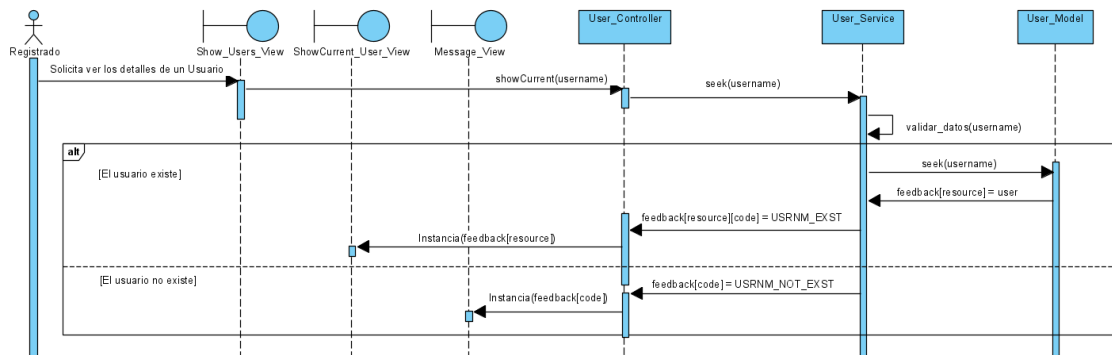


Figura 4 Diagrama de Secuencia - Consultar Detalles del Usuario

#### ▪ Añadir Usuario

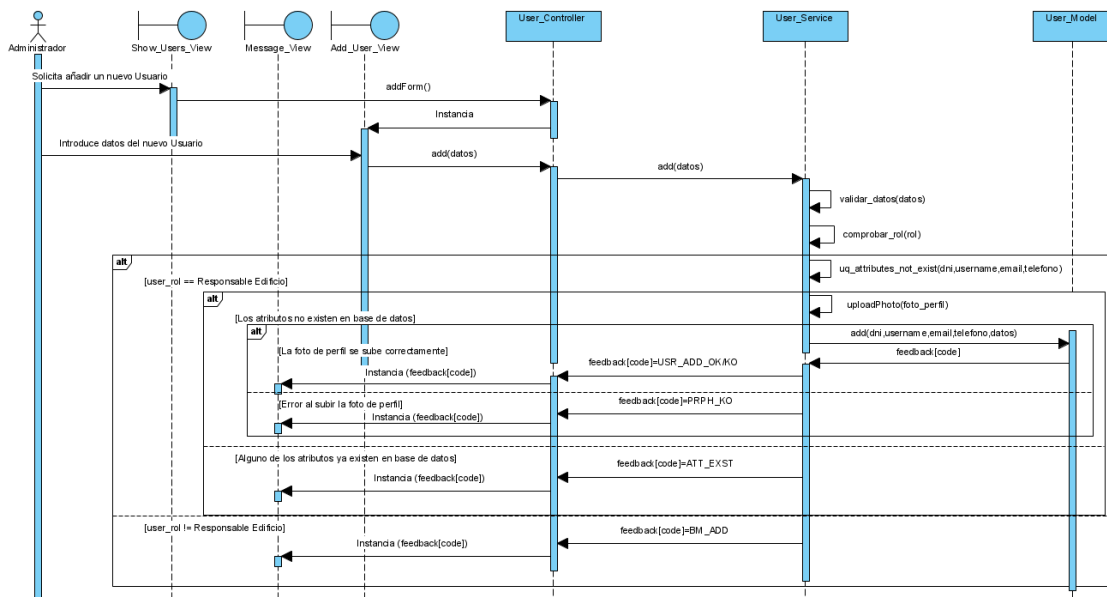


Figura 5 Diagrama de Secuencia - Añadir Usuario

## ■ Login

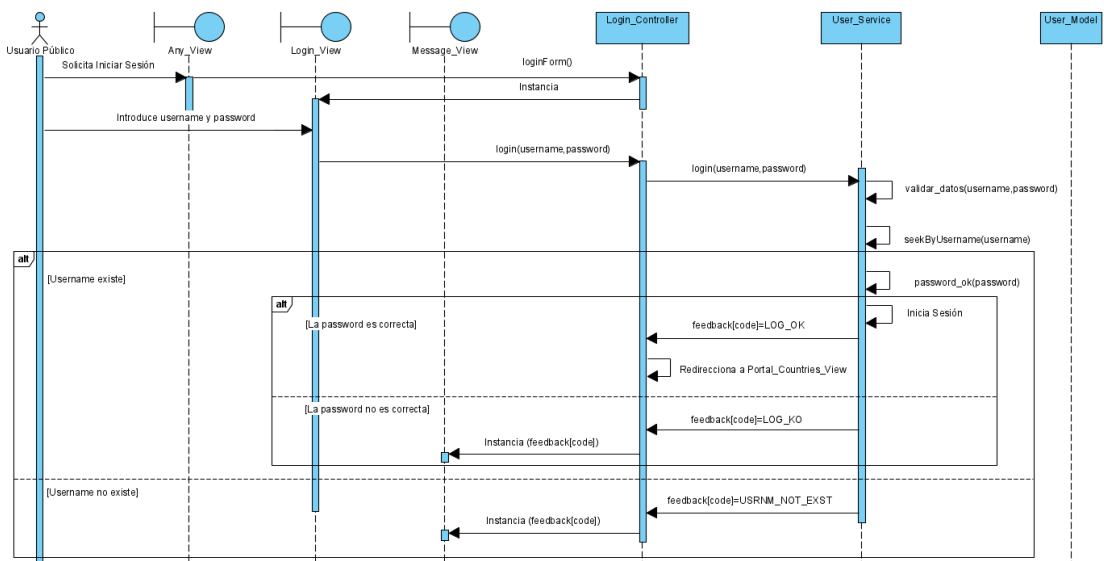


Figura 6 Diagrama de Secuencia - Login

## ■ Eliminar Usuario

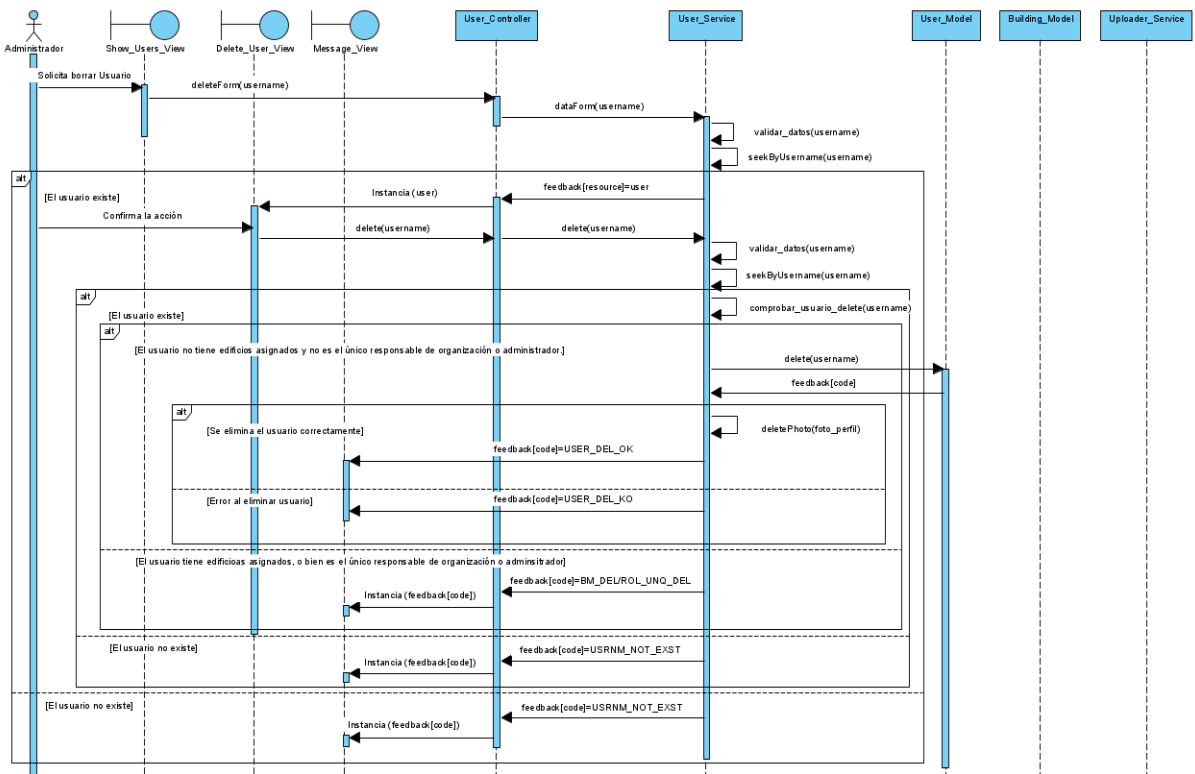


Figura 7 Diagrama de Secuencia - Eliminar Usuario

## ■ Editar Usuario

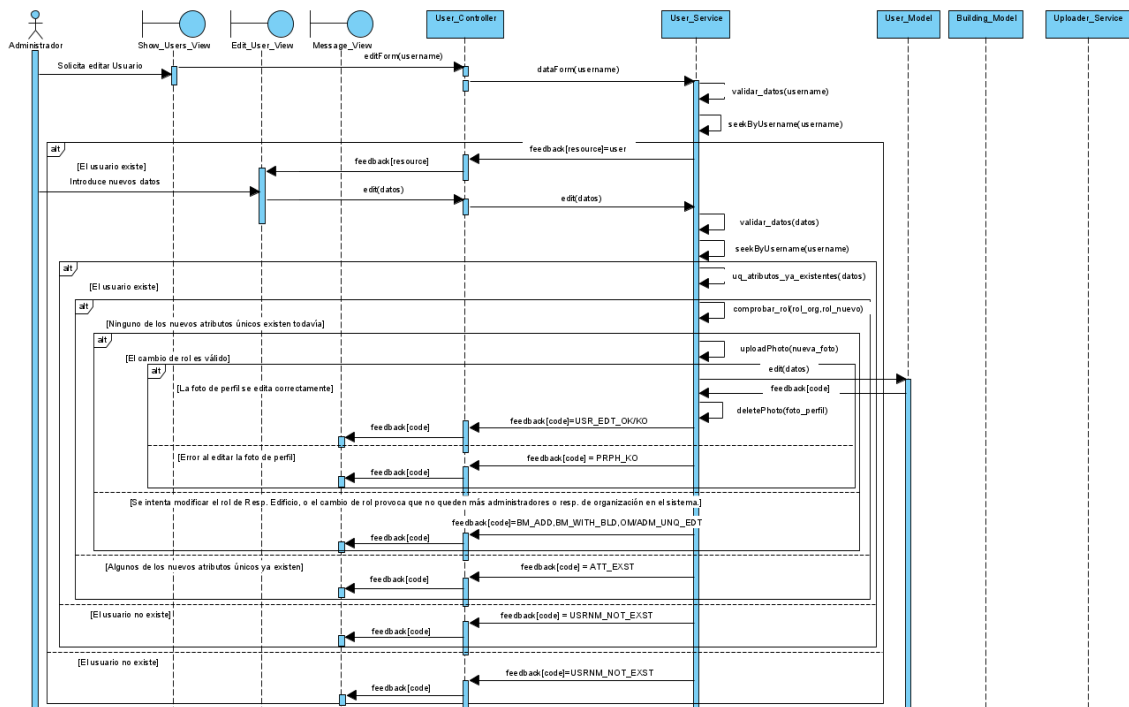


Figura 8 Diagrama de Secuencia - Editar Usuario



## ■ Buscar Usuarios

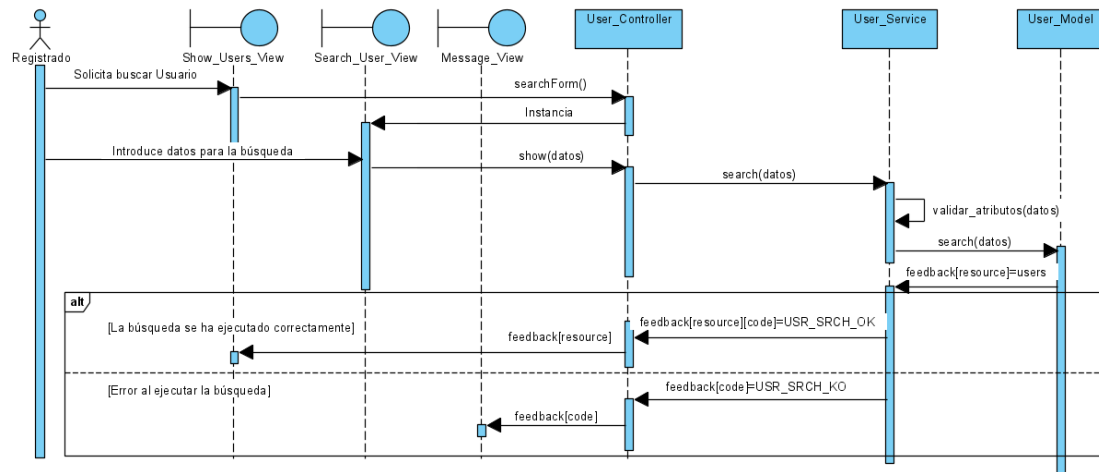


Figura 9 Diagrama de Secuencia - Buscar Usuarios

## ■ Editar Perfil

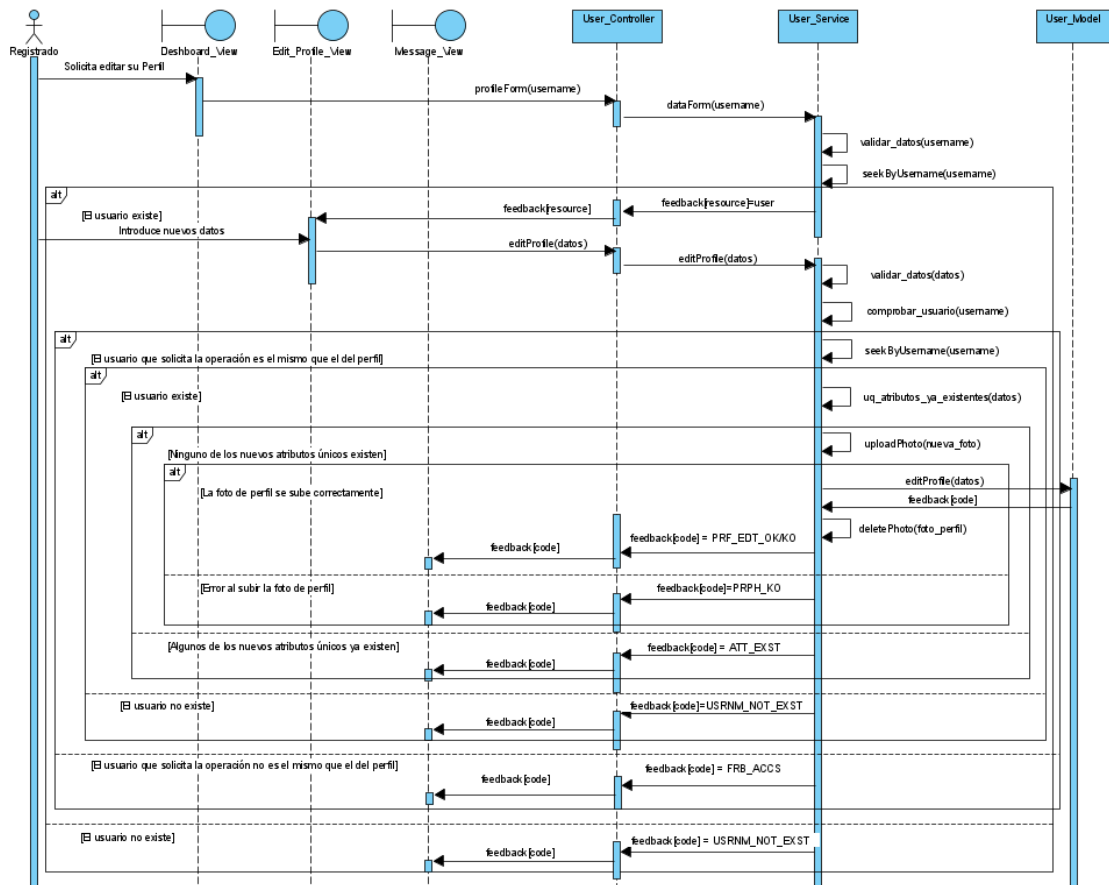


Figura 10 Diagrama de Secuencia - Editar Perfil

## 8.2.2 Iteración 7. Gestión de Edificios

### 8.2.2.1 Gestión de Edificios

#### ▪ Listar Edificios

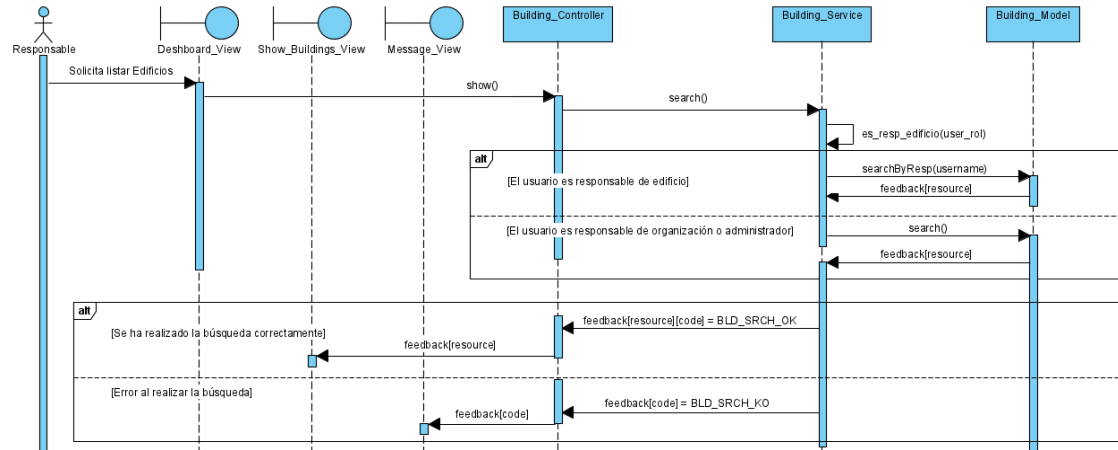


Figura 11 Diagrama de Secuencia - Listar Edificios

#### ▪ Añadir Edificio

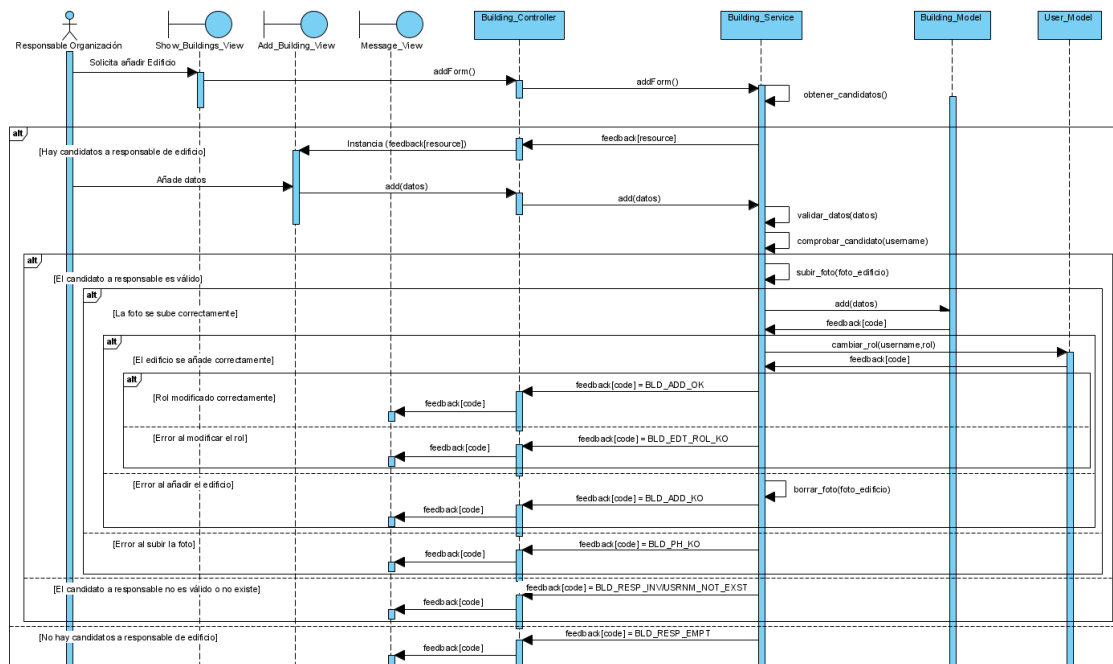
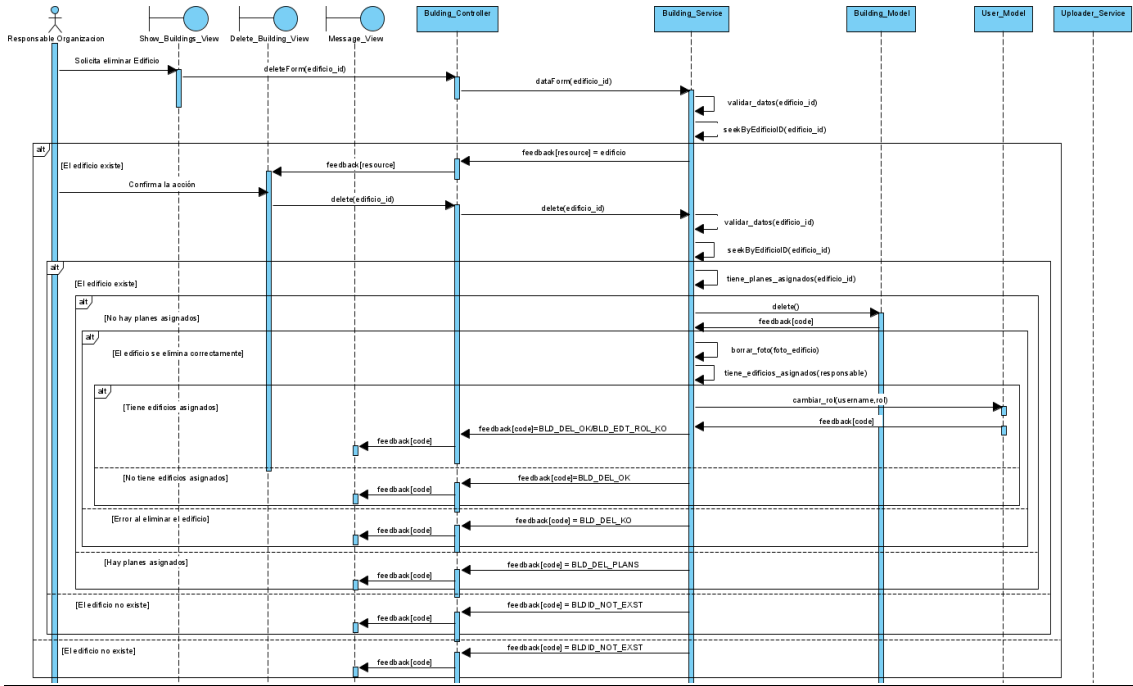


Figura 12 Diagrama de Secuencia - Añadir Edificio

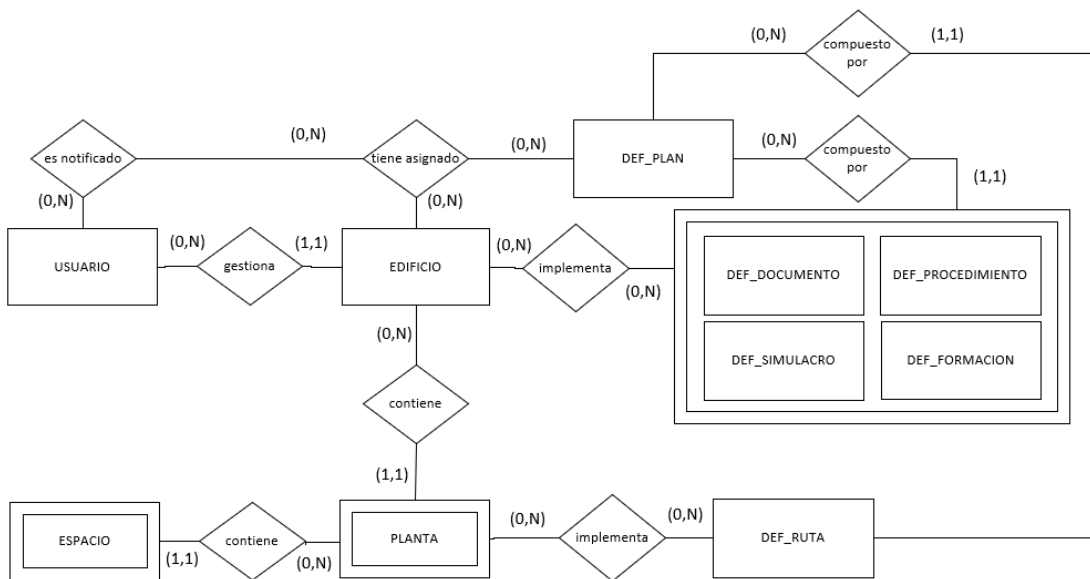
- Eliminar Edificio



*Figura 13 Diagrama de Secuencia - Eliminar Edificio*

## 9 Gestión de Datos e Información

A continuación se detalla el diagrama Entidad/Relación (Figura 4), en la que se puede observar mucho más claramente las relaciones entre las distintas entidades de la BD, y el diagrama de modelado de la base de datos (Figura 5), en el que se incluye los atributos de cada entidad.



*Figura 14. Modelo Entidad/Relación*



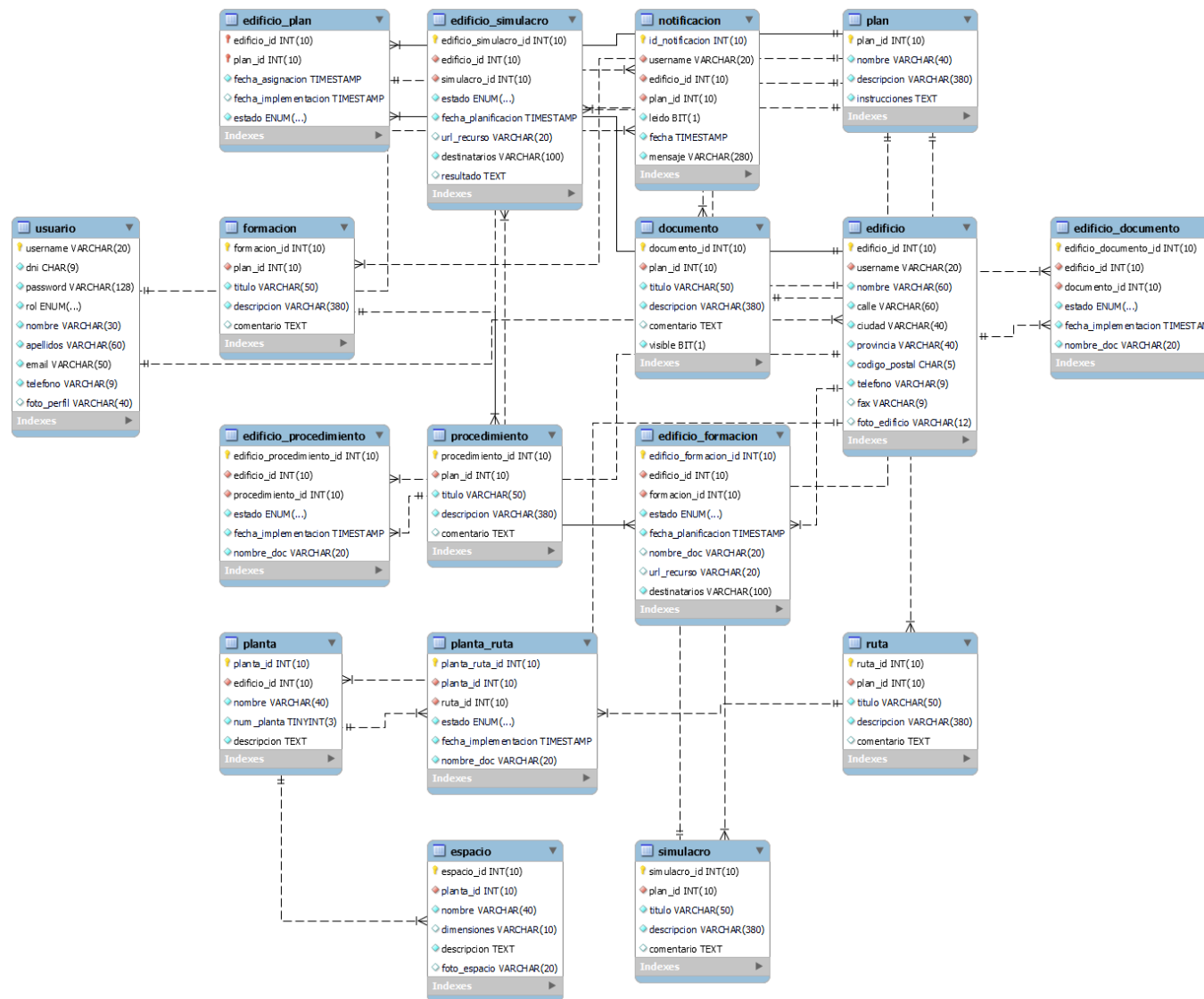


Figura 15. Diagrama de modelado de la Base de Datos