## Forma Descripción generada automáticamente con confianza baja

Imagen que contiene Texto

Descripción generada automáticamente Un dibujo con letras

Descripción generada automáticamente con confianza media

**Escuela Técnica Superior de Ingeniería**

**Universidad de Huelva**

**Trabajo Fin de Grado/Máster**

Desarrollo de Aplicación Web para portal de videojuegos

Grado Ingeniería Informática

Juan Crespo Fuentes

[Fecha]

Índice

[1](#_Toc156438007)

[Resumen 5](#_Toc156438008)

[Abstract 5](#_Toc156438009)

[1. Introducción 6](#_Toc156438010)

[2. Motivación 6](#_Toc156438011)

[3. Objetivos 6](#_Toc156438012)

[4. Tecnologías 7](#_Toc156438013)

[4.1 Tecnologías para el Backend 7](#_Toc156438014)

[4.1.1 Spring Boot 7](#_Toc156438015)

[4.1.2 MySQL 7](#_Toc156438016)

[4.1.3 Eclipse 7](#_Toc156438017)

[4.1.4 Java 7](#_Toc156438018)

[4.1.5 SQL 7](#_Toc156438019)

[4.1.6 JSON 8](#_Toc156438020)

[4.2 Tecnologías para el Frontend 8](#_Toc156438021)

[4.2.1 HTML 8](#_Toc156438022)

[4.2.2 CSS 8](#_Toc156438023)

[4.2.3 Bootstrap 8](#_Toc156438024)

[4.2.4 Javascript 8](#_Toc156438025)

[4.2.5 Thymeleaf 8](#_Toc156438026)

[Postman 8](#_Toc156438027)

[5. Análisis y diseño 9](#_Toc156438028)

[5.1 Arquitectura del sistema 9](#_Toc156438029)

[5.2 Requisitos generales del sistema 9](#_Toc156438030)

[5.3 Especificaciones de Actores del Sistema 10](#_Toc156438031)

[5.4 Especificaciones de Caso de Uso del Sistema 12](#_Toc156438032)

[6. Implementación 29](#_Toc156438033)

[Dominio 29](#_Toc156438034)

[Repositorios 29](#_Toc156438035)

[Servicios 29](#_Toc156438036)

[Controladores 29](#_Toc156438037)

[Vistas 29](#_Toc156438038)

[Elementos estáticos 29](#_Toc156438039)

[Ficheros de configuración 29](#_Toc156438040)

[7. Seguridad 30](#_Toc156438041)

[8. Manual de usuario 31](#_Toc156438042)

[Todos los usuarios 31](#_Toc156438043)

[Usuarios registrados 31](#_Toc156438044)

[Utilización de la API 31](#_Toc156438045)

# Resumen

# Abstract

# Introducción

La creación de este trabajo de fin de grado, en adelante TFG, surge de mi afición por hacer uso de aplicaciones web que recopilan información sobre películas, series, libros, etc. Para la creación de mi aplicación web he tomado como referencia características comunes de varias aplicaciones web de este tipo y, en especial, de la aplicación Letterboxd. De esta aplicación he referenciado varias funcionalidades y estilos específicas de esta aplicación web.

# Motivación

Para varias de las diferentes obras artísticas existe una aplicación común para toda la comunidad de ese tipo de obra. Por ejemplo, Letterboxd, mencionado anteriormente, que es la aplicación web utilizada por la mayoría de los cinéfilos. Otros ejemplos serían Goodreads para los lectores o TV Time para los amantes de las series. En el caso de los videojuegos existen varias aplicaciones de este tipo, pero no hay un consenso general en la comunidad de jugadores.

# Objetivos

El objetivo principal es la creación de una aplicación web que recopile información sobre los videojuegos que están en el mercado, además de que sirva como aplicación web común a todos los jugadores. Los usuarios podrán llevar un seguimiento de los videojuegos que quieran probar, que estén jugando actualmente o ya hayan probado. Cada usuario podrá puntuar cada juego en el que tenga un progreso, la suma de todas estas valoraciones conseguirá una puntuación general de la comunidad para cada videojuego. El usuario también podrá escribir reseñas y crear listas.

Otro de los objetivos es prescindir de un actor administrador que vaya añadiendo los diferentes datos que refiere la aplicación web. Como sustitución se usarán los datos de una página web externa a través de una API. También, debido a esto, otra de las metas es crear nuestra propia API para que agentes externos puedan usar los datos de nuestra aplicación.

# Tecnologías

## Tecnologías para el Backend

### Spring Boot

Spring Boot es un marco de desarrollo de aplicaciones de código abierto en Java. Ofrece gran facilidad en la creación y despliegue del proyecto debido a la automatización en ciertos aspectos de la configuración como por ejemplo la incorporación de su propio servidor.

### MySQL

MySQL es un gestor de base de datos relacional gratuito y de código abierto.

### Eclipse

Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment) de código abierto muy utilizado para el desarrollo software. Ofrece una amplia variedad de herramientas para el uso de diferentes lenguajes de programación, aunque el más usado en este entorno es Java.

### Java

Java es el lenguaje número uno de programación y desarrollo. Su sintaxis es clara e intuitiva, además de ser parecida a otros lenguajes de programación similares. Tiene una gran variedad de librerías. Su portabilidad la hace accesible a todo tipo de plataformas, con ello consigue que muchos de los desarrolladores elijan esta opción a la hora de crear sus aplicaciones.

### Qué es Azure SQL? - Tec InnovaSQL

SQL, o Lenguaje de Consulta Estructurada, es un lenguaje utilizado para consultar, administrar, modificar o recuperar información de un sistema gestor de base de datos relacional.

### 4.1.6 JSON

## 4.2 Tecnologías para el Frontend

### HTML5 Básico, Tu Primera Pagina Web - Ser Programador4.2.1 HTML

HTML, o Lenguaje de Marcado de Hipertexto, es el lenguaje más esencial para la creación de las vistas en una web. Define la estructura de la web a través de etiquetas.

### Logo Css Css3 - Imagen gratis en Pixabay - Pixabay4.2.2 CSS

CSS, o Hojas de Estilo en Cascada, es un lenguaje de diseño utilizado para establecer el estilo de los elementos en un documento HTML.

### 4.2.3 Bootstrap

Bootstrap es un framework front-end de uso gratuito utilizado para el diseño de las aplicaciones web o apps. Internamente utiliza tanto HTML, como CSS y Javascript. Ofrece una gran variedad de estilos predefinidos y modificables que ayudan a ahorrar tiempo y trabajo a la hora de diseñar una aplicación.

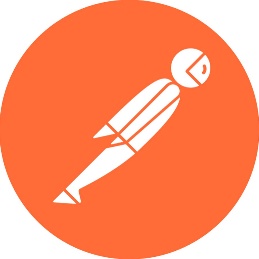
### Guía de estilo para Javascript - Manejando datos4.2.4 Javascript

Javascript es un lenguaje de programación utilizado para dotar a una página web de interactividad y dinamismo. En esta aplicación se ha usado para la automatización visual al entrar en una vista y para la actualización dinámica visual en determinados eventos.

### 4.2.5 Thymeleaf

Thymeleaf es un motor de plantillas utilizado en proyectos spring con arquitectura MVC. En este caso se ha utilizado para HTML5, pero puede ser usado para XML y XHTML también. Es una herramienta de flexible y de fácil uso con gran variedad de utilidad que ayuda a simplificar la codificación y la relación de los datos entre los componentes del MVC.

### Postman

Postman es una API utilizada para testear aplicaciones y APIs. Envía solicitudes HTTP en el que se podrá ver las respuestas en una interfaz gráfica intuitiva. En Zero se ha utilizado para testear la API que se ha creado.

# Análisis y diseño

## 5.1 Arquitectura del sistema

## 5.2 Requisitos generales del sistema

|  |  |
| --- | --- |
| **RG-000** |  |
| **Versión** |  |
| **Descripción** |  |
| **Importancia** |  |
| **Prioridad** |  |
| **Comentarios** |  |

## 5.3 Especificaciones de Actores del Sistema

|  |  |
| --- | --- |
| **AS-001** | Usuario registrado |
| **Versión** | 1 |
| **Dependencias** |  |
| **Descripción** | Este actor del sistema representa al usuario que ha sido registrado previamente mediante una verificación de inicio de sesión. Esto le permitirá tener acceso a todas las funcionalidades de la aplicación. |
| **Comentarios** |  |

Tabla AS-001

|  |  |
| --- | --- |
| **AS-002** | Usuario no registrado |
| **Versión** | 1 |
| **Dependencias** |  |
| **Descripción** | Este actor del sistema representa al usuario que no ha sido registrado en el sistema. |
| **Comentarios** | Solo tendrá acceso a las funcionalidades que impliquen visualizar. |

Tabla AS-002

|  |  |
| --- | --- |
| **AS-003** | API externa |
| **Versión** | 1 |
| **Dependencias** |  |
| **Descripción** | Servicio externo que proporciona todos los datos relacionados con los videojuegos, plataformas, desarrolladores y géneros. |
| **Comentarios** | En este caso se utiliza la API de RAWG. |

Tabla AS-003

## 5.4 Especificaciones de Caso de Uso del Sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Iniciar sesión | | | |
| **Versión** | 1 | | | |
| **Precondición** | El usuario deberá estar registrado en el sistema. | | | |
| **Dependencias** |  | | | |
| **Descripción** | El sistema pedirá a un usuario que no ha iniciado sesión que introduzca sus credenciales. | | | |
| **Secuencia normal** | **Paso** | **Acción** | | |
| 1 | El sistema le pide al usuario que introduzca su nombre de usuario y contraseña. | | |
| 2 | El usuario introduce los datos. | | |
| 3 | El sistema comprueba si los datos introducidos son correctos. | | |
| 4 | El inicio de sesión habrá finalizado con éxito. | | |
| **Postcondición** | Acceso a las funcionalidades que pueden realizar los usuarios registrados. | | | |
| **Excepciones** | **Paso** | | **Acción** | |
| 3 | | Si los datos introducidos no son correctos. | |
| 3.1 | El sistema lanza un mensaje de error informando que los datos introducidos no son correctos. |
| 3.2 | Volverá al paso 1, pedirá de nuevo que el usuario introduzca sus credenciales. |
| **Importancia** | Muy alta | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Registrarse | | | |
| **Versión** | 1 | | | |
| **Precondición** | No tener una sesión iniciada | | | |
| **Dependencias** |  | | | |
| **Descripción** | El sistema pedirá a un usuario no registrado que introduzca los datos de su nueva cuenta a través de un formulario. | | | |
| **Secuencia normal** | **Paso** | | **Acción** | |
| 1 | | El sistema le pide al usuario no registrado rellenar un formulario sobre los siguientes datos: nombre de usuario, email, nombre, apellidos, contraseña y país (opcional). | |
| 2 | | El usuario introduce los datos. | |
| 3 | | El sistema comprueba que los datos introducidos son correctos. | |
| 4 | | El registro de un nuevo usuario habrá finalizado con éxito. | |
| **Postcondición** |  | | | |
| **Excepciones** | **Paso** | **Acción** | | |
| 3 | Si el correo o el nombre de usuario ya está registrado. | | |
| 3.1 | | El sistema lanza un mensaje de error informando que el correo o el nombre de usuario ya están en uso. |
| 3.2 | | El sistema pide de nuevo los datos al usuario. |
| 3 | Si alguno de los campos obligatorios está vacío. | | |
| 3.1 | | El sistema avisa al usuario qué campo está vacío. |
| **Importancia** | Muy alta | | | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Cerrar sesión | |
| **Versión** | 1 | |
| **Precondición** | Tener una sesión iniciada. | |
| **Dependencias** |  | |
| **Descripción** | Un usuario solicita que se cierra la sesión de su cuenta. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita al sistema que se cierra la sesión de su cuenta. |
| 2 | El sistema cierra la sesión. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Muy Alta | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CU-000** | Añadir progreso | | |
| **Versión** | 1 | | |
| **Precondición** | El usuario tiene que haber iniciado sesión y no tener ningún progreso con el videojuego seleccionado. | | |
| **Dependencias** |  | | |
| **Descripción** | El usuario seleccionará una puntuación y un estado de progreso en un videojuego. | | |
| **Secuencia normal** | **Paso** | **Acción** | |
| 1 | El usuario selecciona una puntuación (entre 0,5 y 5) y selecciona un estado de progreso (podrá elegir entre: Jugando, Completado, Abandonado, Planeo jugarlo). | |
| 2 | El sistema procesa los datos introducidos. | |
| 3 | El sistema guarda los datos introducidos en la base de datos y se actualiza la puntuación general del videojuego. | |
| 4 | La operación habrá sido un éxito, se mostrará un mensaje por pantalla. | |
| **Postcondición** | El usuario tendrá la opción de añadir una reseña al videojuego. | | |
| **Excepciones** | **Paso** | **Acción** | |
| 2 | Si el usuario no ha seleccionado una puntuación. | |
| 2.1 | El sistema marca a 0 la puntuación. |
| 2 | Si el usuario no ha seleccionado un estado de progreso. | |
| 2.1 | El sistema marca ‘Jugando’ como estado de progreso. |
| **Importancia** | Alta | | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Actualizar progreso | |
| **Versión** | 1 | |
| **Precondición** | El usuario registrado deberá tener un progreso con el videojuego seleccionado. | |
| **Dependencias** |  | |
| **Descripción** | El usuario actualizará la puntuación o el estado de progreso de un videojuego. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario cambia la puntuación o el estado de progreso de un videojuego. |
| 2 | El sistema procesa los cambios. |
| 3 | El sistema actualiza los cambios en la base de datos. |
| 4 | La operación habrá sido un éxito, se mostrará un mensaje por pantalla. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Alta | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Crear lista | | | |
| **Versión** | 1 | | | |
| **Precondición** | El usuario deberá haber iniciado sesión. | | | |
| **Dependencias** |  | | | |
| **Descripción** | El usuario creará una nueva lista de videojuegos. | | | |
| **Secuencia normal** | **Paso** | **Acción** | | |
| 1 | El usuario solicita crear una nueva lista. | | |
| 2 | El sistema muestra un formulario en el que se pide introducir: Título, Estado (Público o Privado) y Descripción (opcional). | | |
| 3 | El usuario rellena el formulario. | | |
| 4 | El sistema procesa la información introducida y la guarda en la base de datos. | | |
| 5 | La lista se habrá creado con éxito. | | |
| **Postcondición** | El usuario tendrá la opción de añadir juegos a esa lista y si es pública podrá ser visitada por cualquier usuario. | | | |
| **Excepciones** | **Paso** | | **Acción** | |
| 4 | | Si el campo ‘Título’ está vacío. | |
| 4.1 | El sistema informará que se debe rellenar ese campo. |
| **Importancia** | Media | | | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Modificar lista | |
| **Versión** |  | |
| **Precondición** | El usuario deberá tener alguna lista creada. | |
| **Dependencias** |  | |
| **Descripción** | El usuario modificará los datos y los juegos introducidos de una de sus listas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita la modificación de una de sus listas. |
| 2 | El sistema muestra un formulario con los datos de la lista y la lista completa de los juegos. |
| 3 | El usuario modifica los datos de la lista y/o eliminará videojuegos de la lista. |
| 4 | El sistema procesa las modificaciones y actualiza la base de datos. |
| 5 | La modificación de la lista se habrá realizado con éxito. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Media | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Eliminar lista | |
| **Versión** | 1 | |
| **Precondición** | El usuario deberá tener una lista creada. | |
| **Dependencias** |  | |
| **Descripción** | El usuario eliminará una de sus listas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita la eliminación de una de sus listas. |
| 2 | El sistema pregunta si está seguro en un desplegable. |
| 3 | El usuario confirmar su decisión. |
| 4 | El sistema borra la lista de la base de datos. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Media | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Ver lista | |
| **Versión** | 1 | |
| **Precondición** | Si la lista es de otro usuario, esta deberá ser pública. | |
| **Dependencias** |  | |
| **Descripción** | El usuario solicitara ver la información y los videojuegos de una lista. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita ver la información de una lista. |
| 2 | El sistema devuelve una vista con un listado de videojuegos y la información de la lista seleccionada. |
| 3 | Finaliza con éxito. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Media | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Añadir un videojuego a una lista | | | |
| **Versión** | 1 | | | |
| **Precondición** | El usuario registrado deberá tener una lista creada. | | | |
| **Dependencias** |  | | | |
| **Descripción** | El usuario añadirá un videojuego concreto a una o varias de sus listas. | | | |
| **Secuencia normal** | **Paso** | | **Acción** | |
| 1 | | El usuario solicita añadir el juego que está observando en una de sus listas. | |
| 2 | | El sistema muestra una ventana emergente donde mostrará un listado de todas las listas creadas por el usuario. | |
| 3 | | El usuario selecciona una o más de sus listas. | |
| 4 | | El sistema añadirá la relación entre el videojuego y la/s lista/s en la base de datos. | |
| 5 | | El videojuego se habrá añadido correctamente. | |
| **Postcondición** | El videojuego podrá eliminarse de la lista en ‘Modificar lista’. | | | |
| **Excepciones** | **Paso** | **Acción** | | |
| 4 | Si el usuario no ha seleccionado ninguna lista donde no estuviera introducido el videojuego previamente | | |
| 4.1 | | El sistema muestra un mensaje de aviso indicando que el videojuego no ha sido incluido en ninguna lista. |
| **Importancia** | Media | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Añadir reseña | | | |
| **Versión** | 1 | | | |
| **Precondición** | Estar previamente registrado y tener un progreso en el videojuego en el que se quiere hacer la reseña. | | | |
| **Dependencias** |  | | | |
| **Descripción** | El usuario escribirá una reseña de un videojuego en concreto. | | | |
| **Secuencia normal** | **Paso** | | **Acción** | |
| 1 | | El usuario solicita escribir una reseña. | |
| 2 | | El sistema muestra un formulario a rellenar. | |
| 3 | | El usuario escribe la reseña sobre el videojuego. | |
| 4 | | El sistema procesa la información y guarda los nuevos datos en la base de datos. | |
| 5 | | Reseña creada con éxito. | |
| **Postcondición** |  | | | |
| **Excepciones** | **Paso** | **Acción** | | |
| 4 | Si el texto se la reseña está vacía. | | |
| 4.1 | | El sistema avisa que no puede estar vacía. |
| **Importancia** | Media | | | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Ver perfil | |
| **Versión** | 1 | |
| **Precondición** |  | |
| **Dependencias** |  | |
| **Descripción** | El sistema mostrará la información sobre un usuario registrado. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita ver un perfil de un usuario registrado (puede ser él mismo). |
| 2 | El sistema muestra la información del usuario requerido (datos personales, los últimos 4 videojuegos donde ha añadido progreso, la última reseña escrita, un contador de los videojuegos y un contador del número de listas). |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Media | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Modificar perfil | | | |
| **Versión** | 1 | | | |
| **Precondición** | El usuario debe estar registrado | | | |
| **Dependencias** |  | | | |
| **Descripción** | El usuario cambiará los datos de su perfil. | | | |
| **Secuencia normal** | **Paso** | | **Acción** | |
| 1 | | El usuario solicita modificar los datos de su perfil. | |
| 2 | | El sistema devuelve un formulario con los datos del usuario. Solo podrán modificarse nombre, apellidos y país. Ni el correo ni el nombre de usuario podrán ser modificados. | |
| 3 | | El usuario modifica los datos pertinentes. | |
| 4 | | El sistema procesa los cambios y modifica la base de datos. | |
| 5 | | Los datos se habrán modificado con éxito. | |
| **Postcondición** |  | | | |
| **Excepciones** | **Paso** | **Acción** | | |
| 4 | Si campo ‘nombre’ o el campo ‘apellidos’ están vacíos. | | |
| 4.1 | | El sistema reclamará que los campos no pueden estar vacíos. |
| **Importancia** | Baja | | | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Listar juegos, desarrolladores, géneros, plataformas y listas | |
| **Versión** | 1 | |
| **Precondición** |  | |
| **Dependencias** |  | |
| **Descripción** | El sistema mostrará una lista de desarrolladores, plataformas, géneros o listas. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita un listado. |
| 2 | El sistema devuelve una vista con el listado de la petición. |
| 3 |  |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Alta | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Búsqueda de videojuegos | |
| **Versión** | 1 | |
| **Precondición** |  | |
| **Dependencias** |  | |
| **Descripción** | El sistema mostrará un listado teniendo en cuenta la búsqueda de un usuario. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario el campo del buscador. |
| 2 | El sistema procesará el texto escrito por el usuario y buscará en la base de datos resultados que coincidan con la búsqueda. |
| 3 | El sistema mostrará un listado de videojuegos con los resultados de la búsqueda. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** | Alta | |

|  |  |  |
| --- | --- | --- |
| **CU-000** | Ver detalle de un videojuego | |
| **Versión** | 1 | |
| **Precondición** |  | |
| **Dependencias** |  | |
| **Descripción** | El sistema mostrará la información completa de un videojuego. | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El usuario solicita ver los detalles de un videojuego. |
| 2 | El sistema devuelve una vista con toda la información de un videojuego (puntuación general, desarrolladores, géneros, plataformas en la que se puede jugar, descripción, nº de comentarios, nº de usuarios que tienen progreso, descripción y 3 reseñas. Si el usuario está registrado y tiene un progreso con ese videojuego también se mostrará). |
| 3 | Operación realizada con éxito. |
| **Postcondición** |  | |
| **Excepciones** |  | |
| **Importancia** |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CU-000** | Recoger datos de API externa | | | |
| **Versión** | 1 | | | |
| **Precondición** | El sistema deberá tener una clave de acceso para la API | | | |
| **Dependencias** |  | | | |
| **Descripción** | El sistema obtendrá datos de una web externa a través de una API. Los datos obtenidos de la web externa serán procesados y almacenados en la base de datos de nuestro sistema. | | | |
| **Secuencia normal** | **Paso** | **Acción** | | |
| 1 | El sistema hará una petición HTTP parametrizada a la web externa. | | |
| 2 | La web externa procesará la petición y devolverá los datos que se han solicitado en formato JSON. | | |
| 3 | El sistema procesará los datos JSON y los transformará para guardarlos en la base de datos del sistema. | | |
| **Postcondición** |  | | | |
| **Excepciones** | **Paso** | | **Acción** | |
| 3 | | Si alguno dato procesado ya está en la base de datos del sistema. | |
| 3.1 | El sistema descartará ese dato y no lo volverá a meter en la base de datos del sistema. |
| **Importancia** | Muy Alta | | | |

# Implementación

Texto

Descripción generada automáticamenteEn este apartado se describirá los diferentes módulos y paquetes del proyecto de una manera más detallada. Como se ha mencionado antes, se ha usado Eclipse como IDE para realizar la aplicación. Cada uno de estos elementos forman una parte crucial en el proyecto.

El proyecto está dividido en dos partes fundamentales. La primera es la carpeta *src/main/java,* donde está codificada toda la parte del Back-end y la segunda es la carpeta *src/main/resources* donde está la parte del cliente, el Front-end.

Es la estructura común de un proyecto *Maven,* que nos permite gestionar las dependencias y nos facilita la gestión y construcción del proyecto.

A continuación, se describirá cada módulo o paquete esencial del proyecto:

## Dominio

*com.zero.domain*

Texto

Descripción generada automáticamente con confianza mediaDentro de este paquete se encuentran todas las entidades que encapsulan la lógica de negocio de la aplicación, como hemos visto anteriormente en el Diagrama de clases. Cada entidad recoge tanto los atributos requeridos como las relaciones entre ellas. Todas las entidades tendrán su correspondiente tabla en la base de datos, al igual que las relaciones creadas. Para poder identificar con facilidad cada objeto creado, se ha añadido un atributo *id* autogenerado a cada una de las entidades.

## Repositorios

*Com.zero.repository*

Texto

Descripción generada automáticamenteLos repositorios son los encargados la persistencia de los datos en nuestra base de datos de MySQL. Gracias al framework de **hibernate** tenemos un fácil acceso a los datos de nuestra base de datos, debido a que es capaz de acceder a una base de datos relacional con objetos Java. En este proyecto se ha usado en los repositorios la notación @Query, perteneciente a Spring Data JPA (Java Persistence API), que nos permite poder hacer consultas SQL y orientarla a objetos.

## Servicios

Texto

Descripción generada automáticamente*com.zero.servicies*

Los servicios contienen funciones y el resto de las lógicas de negocio que no han podido ser definidas en el dominio especialmente en la creación, actualización y eliminación de objetos. Funciona como intermediario entre los controladores y los repositorios.

## Texto Descripción generada automáticamenteControladores

*com.zero.controllers*

Los controladores actúan como intermediarios entre el cliente y el servidor. Recibirá y enviará solicitudes orquestadas por el usuario. Es la parte del Back-end que se comunica con las vistas y el encargado de la gestión de las URL. En este caso se han añadido dos subpaquetes (*com.zero.controllers.user* y *com.zero.controllers.API*) con la intención de especificar su función, en el caso de los usuarios por motivos de seguridad.

## Vistas

## Elementos estáticos

## Ficheros de configuración

# Seguridad

# Manual de usuario

## Todos los usuarios

## Usuarios registrados

## Utilización de la API