

NotePost++

CFGs DAW

CURSO 2020/2021

CAROLINA VANESSA ZAPATA CARDOZO

COLEGIO CALASANZ SALAMANCA



1. Introducción	4
2. Descripción de la aplicación	5
3. Tecnologías escogidas y justificación	8
4. Diseño de la aplicación	9
4.1. Diagramas y definición de casos de uso.....	9
4.1.1. <u>CU01 Iniciar sesión</u>	9
4.1.2. <u>CU02 Buscar Proyecto</u>	10
4.1.3. <u>CU03 Mostrar Proyectos</u>	10
4.1.4. <u>CU04 Crear Proyecto</u>	11
4.1.5. <u>CU05 Eliminar Proyecto</u>	11
4.1.6. <u>CU06 Dar de alta usuario</u>	12
4.1.7. <u>CU07 Cerrar sesión</u>	13
4.1.8. <u>CU08 Datos cuenta</u>	13
4.1.9. <u>CU09 Modificar datos</u>	14



4.1.10. <u>CU10 Eliminar cuenta</u>	14
4.1.11. <u>CU11 Exportar a pdf</u>	15
4.1.12. <u>CU12 Invitar usuarios</u>	15
4.1.13. <u>CU13 Editar</u>	16
4.1.14. <u>CU14 Marcar tarea</u>	16
4.2. Diagramas de clases	18
4.3. Modelo de dominio	23
5. Arquitectura de la aplicación	25
5.1. Estructura del proyecto	25
5.2. Librerías externas utilizadas	27
6. Manual de despliegue	28



1. Introducción

NotePost++ es una aplicación web para el seguimiento de las tareas a realizar en un proyecto donde se refleja el progreso del mismo a medida que se completan dichas tareas. Permite a usuarios registrados en la aplicación crear nuevos proyectos en los que se podrán añadir tareas sobre el trabajo pendiente relacionado con el mismo. Permite modificar y eliminar las tareas, así como eliminar completamente un proyecto. Además, existe la posibilidad de invitar como colaboradores a otros usuarios, los cuales podrán marcar tareas realizadas. Solo el creador del proyecto podrá eliminar proyectos. Para poder hacer uso de la aplicación es necesario estar registrado. También existe la opción de exportar a pdf los detalles de cada proyecto.



2. Descripción de la aplicación

La página inicial de la aplicación web permite al usuario iniciar sesión. Consta de dos campos para introducir nombre de usuario y contraseña.

En caso de no estar registrado, puede darse de alta pulsando en el enlace que se indica debajo del formulario de login, lo cual muestra una nueva página con un formulario. Los campos que debe introducir el usuario serán nombre, apellidos, nombre de usuario, contraseña, correo electrónico y si quiere estar visible o no a otros usuarios. Cuando se pulse el botón enviar, una vez validados los datos, se procederá a registrar al nuevo usuario. En caso de existir previamente, se muestra un mensaje indicándolo y no se registrará. En caso contrario, se indica con un mensaje de éxito si se ha podido registrar y se redirige al usuario a la página de login. Si se produce algún tipo de error durante el registro se indicará con un mensaje de error.

Una vez el usuario se haya identificado, se muestra la página principal de la aplicación. La parte superior contiene el encabezado (el cual aparece en todas las vistas de la aplicación una vez iniciado sesión) con el logo de la página a la izquierda y a la derecha un menú desplegable que, al pasar el ratón por encima de las opciones y pulsar, se muestran las diferentes opciones. Éstas son: inicio, donde se muestra información de los últimos 2 proyectos; mis proyectos, de los cuales se puede listar, crear y eliminar un proyecto; y usuario, desde donde es posible ver los detalles de la cuenta del usuario logeado, editar sus datos, eliminar su cuenta, y cerrar la sesión actual. Si se pulsa sobre el logo, se dirige a la página de inicio. Debajo del encabezado se muestran todos los proyectos del usuario ordenados por fecha con el rol de casa usuario (propietario o colaborador), el título, la fecha de creación, una breve descripción, la barra de progreso, las tareas realizadas y las que no, el botón para exportar a pdf, otro botón para invitar colaboradores y la opción de editar el proyecto. Algunos de estos contenidos se muestran o no dependiendo de la página donde se visualicen los proyectos.

Inicio:

- En esta página aparecen los dos últimos proyectos de cada usuario. De los mismos se muestra el nombre, fecha, descripción y progreso, además de si el usuario es propietario o colaborador.

Proyectos:

- Mis proyectos: cuando el usuario pulsa esta opción, se le muestra una página en la que puede visualizar sus proyectos. La diferencia con la página de inicio es que le permite filtrarlos a través de un formulario con los campos nombre, descripción y fecha de creación. Si no indica ningún criterio, se mostrarán todos los proyectos. De cada proyecto se visualizan todos los aspectos anteriormente mencionados en la descripción general de los proyectos.



- Nuevo proyecto: al pulsar esta opción, el usuario podrá crear un nuevo proyecto. Para ello se carga una nueva página que dispone de un formulario para introducir el nombre del proyecto, una breve descripción y las tareas a realizar. Para las tareas, dispondrá de un campo de texto y un botón para ir añadiendo todas las tareas que precise. En la parte inferior del formulario habrá un botón para crear el nuevo proyecto. Del mismo se almacenarán su id, su nombre, las tareas asociadas y la fecha actual del sistema como fecha de creación.
- Eliminar proyecto: si se pulsa esta opción, se muestra una página que permite eliminar un proyecto. Al igual que en el apartado “Mis proyectos”, se podrá filtrar a través del mismo formulario de búsqueda. En caso de haber resultados, debajo del formulario aparecen los proyectos con un botón para eliminarlos. La aplicación pedirá confirmación y en caso de ser afirmativa, se procederá a eliminar el proyecto y a mostrar un mensaje de éxito. Solo se podrán eliminar los proyectos que haya creado el usuario.
- Botón exportar a pdf: el lateral derecho de la tarjeta de información de cada proyecto, hay un botón para exportar los datos a pdf. El sistema recopilará la información del proyecto y lo devolverá a la página en formato pdf.
- Botón colaborador: justo debajo del botón de exportar a pdf, se encuentra un botón para invitar colaboradores. Cuando se pulse, se carga un pop-up en el que se muestran los usuarios colaboradores que tiene el proyecto y permite añadir más a través de un desplegable que contienen los usuarios que hayan elegido ser visibles a otros usuarios. Al pulsar el botón invitar, se añadirá el usuario colaborador al proyecto.
- Botón editar: se encuentra situado en la misma línea que los dos anteriores. Cuando se pulsa, aparece un pop-up con un formulario que contiene toda la información del proyecto permitiendo editarla, así como añadir o eliminar tareas. Cuando se pulsa guardar, los cambios serán guardados en la base de datos. En cambio, si se pulsa el botón cancelar, no se producirá ningún cambio.

Usuario:

- Mis datos: cuando el usuario pulse esta opción puede ver los datos almacenados sobre su perfil. Se muestra una página con un formulario de solo lectura con su nombre de usuario, nombre, apellidos, correo electrónico, contraseña y visibilidad. Seguidamente, hay un botón para editar. Si el usuario lo pulsa, se habilitarán los campos del formulario y dicho botón es reemplazado por un botón para guardar los cambios y otro para cancelar. El sistema comprobará que los datos son válidos y los actualizará en la base de datos en caso de pulsar



guardar o volverá a deshabilitar el formulario en caso de pulsar cancelar. En la esquina opuesta de donde se encuentran dichos botones, hay otro para borrar la cuenta del usuario. Si se pulsa, aparece un pop-up de confirmación que en caso de ser afirmativo, se procederá a eliminar toda la información del usuario de la base de datos, incluidos sus proyectos.

- Cerrar sesión: al pulsar esta opción se cierra la sesión del usuario actual y se muestra la página de login.



3. Tecnologías escogidas y justificación

Las tecnologías que se mencionan a continuación han sido escogidas en base a los conocimientos adquiridos en el transcurso de las clases teóricas del ciclo superior Desarrollo de Aplicaciones Web.

En este proyecto se utiliza para el backend java como lenguaje de programación con el framework Spring. Spring, ofrece como elemento clave el soporte de infraestructura a nivel de aplicación, brindando un completo modelo tanto para la configuración como para la programación de aplicaciones desarrolladas en Java, sin discriminación en cuanto al despliegue de la plataforma.

Para almacenar los datos se utiliza una base de datos relacional con un sistema de gestión MySQL.

En cuanto al frontend, el framework escogido de Javascript es Angular, ya que, en Indra, centro de trabajo donde realizo prácticas, es el más utilizado.



4. Diseño de la aplicación

4.1. Diagramas y definición de casos de uso

4.1.1. CU01 Iniciar sesión

Nombre: Iniciar sesión Id: CU01
Actores: usuario
Descripción: el usuario podrá iniciar sesión para ingresar en la aplicación
Precondiciones: el usuario tendrá que estar registrado en el sistema
Pasos: <ol style="list-style-type: none">1. El usuario accederá a la página web.2. El sistema mostrará el formulario de datos.3. El usuario introducirá su nombre de usuario y contraseña y pulsará Iniciar Sesión.4. El sistema verifica que el usuario se encuentre registrado.5. El sistema verifica que la contraseña es correcta.6. El sistema da acceso a la aplicación.
Alternativa: <ol style="list-style-type: none">5. El usuario no se encuentra registrado y el sistema muestra un mensaje.6. Vuelve al paso 2.
<ol style="list-style-type: none">6. La contraseña introducida por el usuario no es correcta. El sistema muestra un mensaje.7. Vuelve al paso 2.



4.1.2. CU02 Buscar Proyecto

Nombre: Buscar Proyecto Id: CU02
Actores: usuario
Descripción: el usuario podrá visualizar sus proyectos
Precondiciones: el usuario tendrá que estar registrado en el sistema
Pasos: <ol style="list-style-type: none">2. El sistema mostrará un formulario de búsqueda con los campos nombre, descripción y fecha de creación.3. El usuario introducirá los datos por lo que quiera filtrar.4. Si se encuentran datos<ol style="list-style-type: none">4.1 El sistema mostrará los proyectos.5. SI NO<ol style="list-style-type: none">5.1 El sistema informará que no se han encontrado resultados.6. El caso de uso finaliza.

4.1.3. CU03 Mostrar Proyectos

Nombre: Mostrar Proyectos Id: CU03
Actores: usuario
Descripción: el usuario podrá visualizar sus proyectos
Precondiciones: el usuario tendrá que estar registrado en el sistema
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción "Mis Proyectos"2. El sistema ejecutará el caso de uso "Buscar Proyectos"6. El caso de uso finaliza.



4.1.4. CU04 Crear Proyecto

Nombre: Crear Proyecto Id: CU04
Actores: usuario
Descripción: el usuario podrá crear un proyecto
Precondiciones: el usuario tendrá que estar registrado en el sistema
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Crear Proyecto”2. El sistema mostrará un formulario para introducir los datos del proyecto.3. El sistema valida los datos4. Si los datos son correctos<ol style="list-style-type: none">4.1 El sistema guardará los datos en la base de datos5. Si NO<ol style="list-style-type: none">5.1 El sistema informará que los datos no son correctos.5.2 Vuelve al paso 2.6. El caso de uso finaliza.

4.1.5. CU05 Eliminar Proyecto

Nombre: Eliminar proyecto Id: CU05
Actores: usuario
Descripción: el usuario podrá eliminar un proyecto
Precondiciones: el usuario tendrá que estar registrado en el sistema
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Eliminar Proyecto”.2. El sistema ejecuta el caso de uso “Buscar Proyecto”.



3. El usuario pulsa el botón eliminar del proyecto que escoja.
4. El sistema pedirá confirmación.
5. SI confirma que quiere eliminar el proyecto
 - 5.1 El sistema eliminará el proyecto.
 - 5.2 El caso de uso finaliza.
5. SI NO
 - 5.1 El caso de uso finaliza.

4.1.6. CU06 Dar de alta usuario

Nombre: Dar de alta usuario Id: CU06
Actores: usuario
Descripción: el usuario se dará de alta a un usuario
Precondiciones:
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Registrarse”2. El sistema mostrará un formulario para introducir los datos del usuario.3. El sistema valida los datos4. SI los datos son correctos y el usuario no está registrado<ol style="list-style-type: none">4.1 El sistema guardará los datos del usuario en la base de datos5. SI NO<ol style="list-style-type: none">5.1 El sistema informará del error.5.2 Vuelve al paso 2.6. El caso de uso finaliza.



4.1.7. CU07 Cerrar sesión

Nombre: Cerrar sesión Id: CU07
Actores: usuario
Descripción: el usuario cerrará la sesión actual
Precondiciones: el usuario debe haber iniciado sesión
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción "Cerrar sesión"2. El sistema cerrará la sesión actual3. El sistema mostrará la página de login.4. El caso de uso finaliza.

4.1.8. CU08 Datos cuenta

Nombre: Datos cuenta Id: CU08
Actores: usuario
Descripción: el usuario podrá visualizar los datos almacenados sobre él
Precondiciones: el usuario debe haber iniciado sesión
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción "Cuenta"2. El sistema mostrará la información almacenada de ese usuario.3. Si el usuario pulsa el botón "Modificar"<ol style="list-style-type: none">3.1 Se ejecuta el caso de uso "Modificar datos"4. El caso de uso finaliza



4.1.9. CU09 Modificar datos

Nombre: Modificar datos Id: CU09
Actores: usuario
Descripción: el usuario podrá modificar
Precondiciones: el usuario debe haber iniciado sesión
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Modificar”2. El sistema habilitará los campos del formulario que se puedan modificar.3. El usuario pulsará el botón guardar cambios.4. El sistema pedirá confirmación5. Si el usuario confirma que quiere guardar los cambios<ol style="list-style-type: none">3.1 El actualizará los cambios en la base de datos4. Si NO<ol style="list-style-type: none">4.1 El caso de uso finaliza.

4.1.10. CU10 Eliminar cuenta

Nombre: Eliminar cuenta Id: CU10
Actores: usuario
Descripción: el usuario podrá eliminar su cuenta
Precondiciones: el usuario debe haber iniciado sesión
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Mis datos”.2. El usuario seleccionará la opción “Eliminar cuenta”.



3. El sistema mostrará un pop-up de confirmación
4. SI el usuario confirma que quiere continuar
 - 4.1 El sistema borrará los datos del usuario en la base de datos
4. SI NO
 - 4.1 El caso de uso finaliza.

4.1.11. CU11 Exportar a pdf

Nombre: Exportar a pdf Id: CU11
Actores: sistema, usuario, colaborador
Descripción: el sistema exportará a pdf la información de un proyecto seleccionado
Precondiciones: el usuario debe haber iniciado sesión
Pasos: <ol style="list-style-type: none">1. El usuario seleccionará la opción “Exportar a pdf”2. El sistema buscará los datos almacenados en la base de datos y los almacenará en un pdf.3. El sistema devolverá el documento a la vista.

4.1.12. CU12 Invitar usuarios

Nombre: Modificar datos Id: CU12
Actores: usuario
Descripción: el usuario podrá invitar usuarios para que colaboren en su proyecto
Precondiciones: el usuario debe haber iniciado sesión



Pasos:

1. El usuario seleccionará la opción “Invitar”
2. El sistema cargará un pop-up con un desplegable con usuarios disponibles.
3. El usuario seleccionará el usuario que quiera invitar y pulsará el botón invitar.
4. El sistema añadirá como colaborador al usuario invitado.
5. El sistema cerrará el pop-up.

4.1.13. CU13 Editar

Nombre: Editar

Id: CU13

Actores: usuario

Descripción: el usuario propietario podrá editar un proyecto

Precondiciones: el usuario debe haber iniciado sesión

Pasos:

1. El usuario seleccionará la opción editar
2. El sistema mostrará un pop-up con la información del proyecto
3. El usuario podrá editar la información del proyecto así como añadir o eliminar tareas
4. El caso de uso finaliza

4.1.14. CU14 Marcar tarea

Nombre: Marcar tarea

Id: CU13

Actores: usuario, colaborador

Descripción: el usuario propietario o colaborador podrá marcar una tarea como realizada

Precondiciones: el usuario debe haber iniciado sesión

Pasos:

1. El usuario marcará el check de la tarea que quiera
2. El sistema guardará en la base de datos el nuevo estado de la tarea
3. El caso de uso finaliza

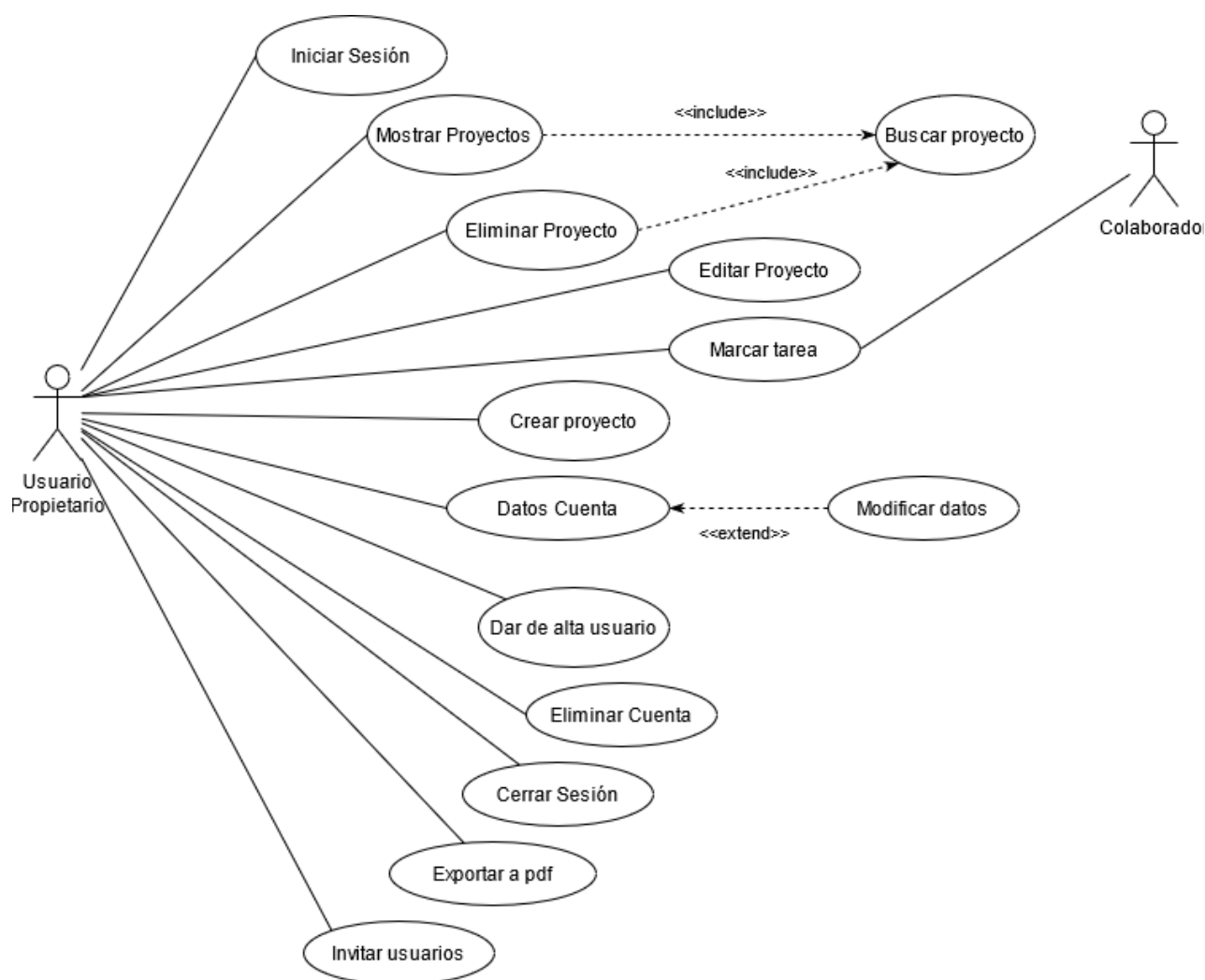


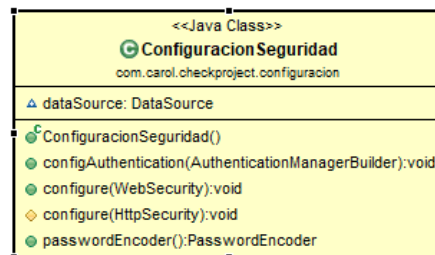
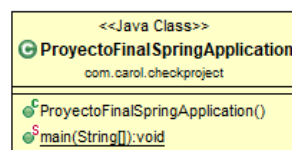
Diagrama de casos de uso de la aplicación



4.2. Diagramas de clases

A Continuación se muestran los diagramas de clases por módulos para simplificar su visualización:

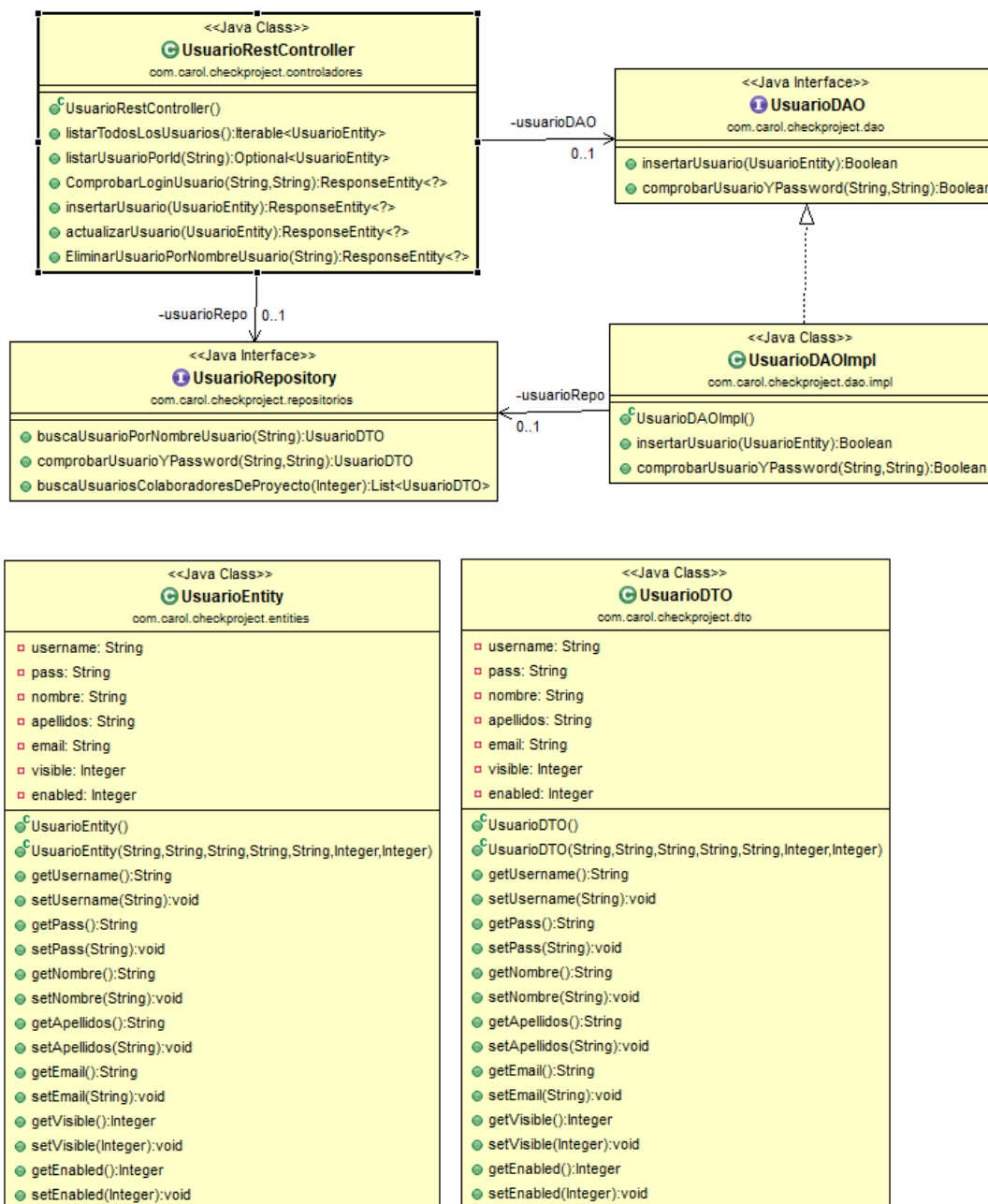
- Clase principal del proyecto y configuración:



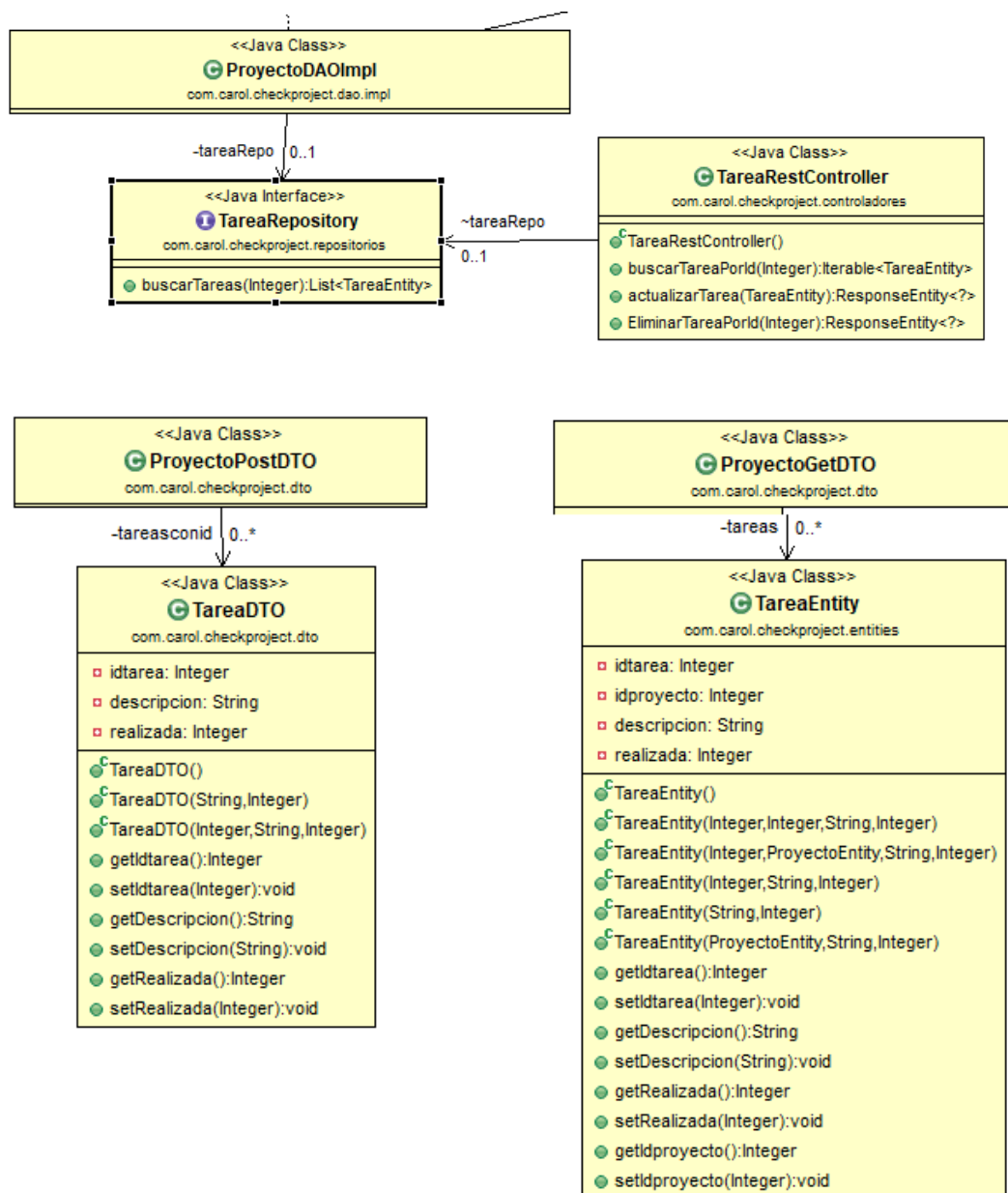
– Clases relacionadas con proyecto:



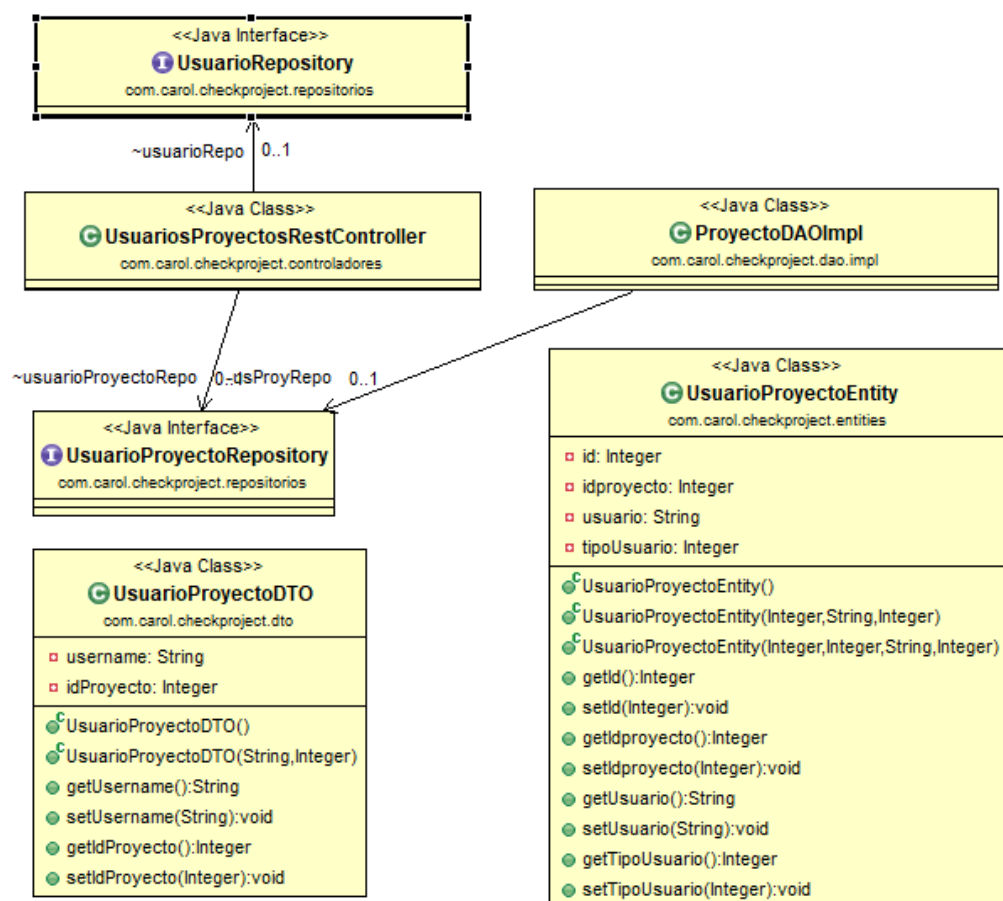
– Clases relacionadas con usuarios:



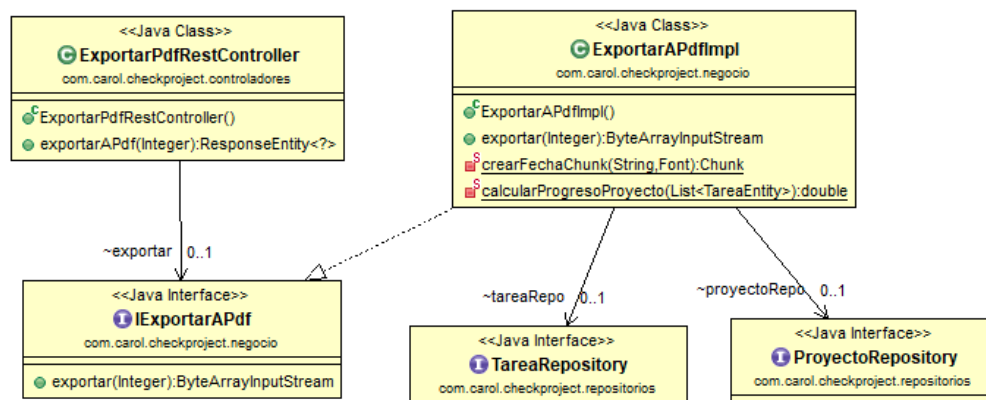
– Clases relacionadas con tareas:



- Tablas relacionadas con usuarios-proyectos:



- Clases relacionadas con la capa de negocio (exportar a pdf):



4.3. Modelo de dominio

En este apartado se detallan los datos que se almacenarán en la base de datos:

- Usuarios: tendrán un identificador único que será su nombre de usuario, único e ineditable, su nombre y apellidos, su correo electrónico, su contraseña y un campo que será cero o uno para indicar si el usuario quiere estar visible para otros usuarios.
- Proyecto: se almacenará un identificador único autogenerated, el identificador del propietario (el usuario que lo creó), el nombre del proyecto, la fecha de creación y una breve descripción.
- Usuario-Proyecto: se utilizará esta tabla intermedia para saber qué usuario creó el proyecto y almacenar los posibles colaboradores. Tendrá un identificador único autogenerated, el identificador del proyecto, el identificador del usuario y el identificador del tipo de usuario del mismo. Por ejemplo, si el usuario 10 crea un proyecto con identificador 33 en esta tabla se guardará que el usuario 10 ha creado el proyecto 33, por lo que su identificador de tipo de usuario será 1 (el correspondiente al propietario del proyecto).
- Tarea: se almacenará un identificador único autogenerated, el identificador del proyecto al que pertenece, la descripción y un 0 o un 1 para indicar si la tarea está realizada.



- Tipo-usuario: en esta tabla se almacenarán los tipos de usuarios que habrá en la aplicación, como creador o colaborador. De esta manera, si en un futuro hay más tipos es más fácil su escalabilidad.

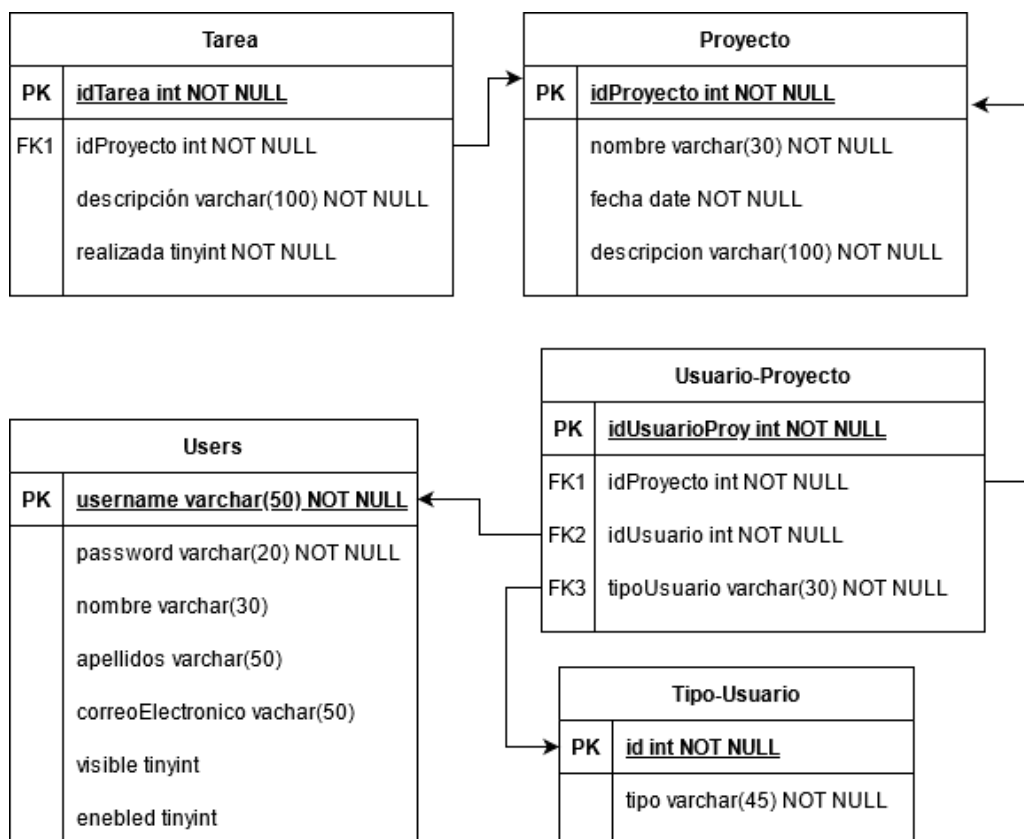


Diagrama relacional de la base de datos



5. Arquitectura de la aplicación

El patrón de arquitectura elegido para este proyecto es el Modelo-Vista-Controlador puesto que separa los datos y lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

5.1. Estructura del proyecto

El backend presenta una arquitectura REST por lo que dispone de diferentes controladores dependiendo del recurso. La estructura de paquetes es la siguiente:

- Controladores: paquete donde se encuentran todos los controladores rest para los recursos.
- Configuración: paquete con clases de configuración como puede ser la de seguridad.
- Dtos: paquete que contiene las clases dtos necesarias para el funcionamiento de la aplicación como usuario, proyecto, etc.
- Entities: paquete con las entidades que permiten el manejo de los datos y la relación con las tablas de la base de datos.
- DAO y DAOImpl: paquetes con las interfaces y sus respectivas implementaciones.
- Negocio: representa la capa de negocio. En este paquete se almacena la interfaz de negocio y su implementación que permite exportar a pdf la información el proyecto seleccionado.
- Repositorios: paquete con los repositorios que necesitamos para poder gestionar operaciones sobre datos fundamentales que nos proporciona Spring, como guardar, modificar, eliminar o leer datos.

En cuanto al frontend, está representado por diferentes componentes que, a su vez, hacen uso de componentes. Esta estructura es la misma proporcionada por la herramienta Angular Cli para Angular, por lo que nos encontramos con las siguientes carpetas:



- node_modules: Carpeta en la que se incluyen todos los node modules instalados al hacer npm install.
- Src: carpeta en la que se realizará el desarrollo de la aplicación. Consta de:
 - Carpeta app:
 - Componentes: carpeta que incluye componentes individuales que pueden ser utilizados en cualquier página.
 - Páginas: carpeta que contiene los componentes principales de cada página. Cada uno de ellos representa una página de la aplicación.
 - Shared: carpeta donde se agrupan recursos compartidos como los servicios, pipes o modelos de datos.
 - routing.module.ts: archivo donde se declaran las rutas de navegación de la aplicación.
 - app.module.ts: archivo nos permite importar todos los elementos que usaremos en una aplicación angular, como componentes, módulos y servicios.
 - Archivos app.components.*: archivos que forman el componente principal de la aplicación.
 - Carpeta assets: está dedicada a guardar los diferentes recursos que se necesitan en la aplicación como pueden ser imágenes, iconos, tipos de fuentes o archivos de traducciones.
 - Carpeta environments: en esta carpeta se guardan los archivos de configuración de los diferentes entornos que se vayan a utilizar.
 - Archivo index.xml: es el html principal, a partir del cual se cargará el resto de la aplicación.
 - Archivo styles.scss: es el archivo de estilos principal de la aplicación

Existen varios archivos con diferentes extensiones. Los más importantes son:



- package.json: configuración necesaria para poder ejecutar la aplicación. Describe, entre otras cosas, las dependencias que necesita el proyecto para funcionar.
- Browserslist: es un archivo de configuración que permite indicar para qué navegadores estará disponible la aplicación que vamos a desarrollar

5.2. Librerías externas utilizadas

En este proyecto se usan las siguientes librerías externas:

Spring:

- spring-boot-starter-data-jpa
- spring-boot-starter-web
- spring-boot-devtools
- tomcat-embed-jasper
- mysql-connector-java
- spring-boot-starter-test
- itext (librería para exportar a pdf)

Angular:

- Bootstrap 4
- Ng-bootstrap
- Angular-fontawesome
- Moment

6. Manual de despliegue

El despliegue de esta aplicación se realiza a través de Docker. De esta manera, basta con ejecutar una serie de instrucciones para tener funcionando la página web independientemente de la máquina o el sistema operativo.

Previamente es necesario crear las imágenes de las diferentes partes que conforman la aplicación. Para ello, se han de seguir los siguientes pasos:

- Generar la imagen de la base de datos: ejecutar la instrucción “docker build -t carovz/img_bbdd .” en la carpeta donde se encuentre el Dockerfile, el cual se detalla a continuación:

```
Dockerfile M X
D: > FP > 2º > PROYECTO > proyectoFinalDAW > Dockerfile > ...
You, seconds ago | 1 author (You)
1 #Dockerfile para crear una imagen basada en mysql
2 FROM mysql
3
4 #Se copian los scripts desde la carpeta en la que estén al directorio dentro del contenedor
5 COPY ./scriptSql/ /docker-entrypoint-initdb.d/
```

- Generar la imagen para el backend: en la carpeta del proyecto ejecutar “docker build -t carovz/img_back .”. Es necesario que el Dockerfile y el war del proyecto se encuentren en la misma altura en la jerarquía de directorios. El Dockerfile del que se parte es el siguiente:

```
Dockerfile M X
D: > FP > 2º > PROYECTO > proyectoFinalDAW > proyectoFinalSpring > Dockerfile > ...
You, a day ago | 1 author (You)
1 FROM tomcat:9.0.24
2
3 EXPOSE 8080
4
5 RUN rm -rf /usr/local/tomcat/webapps/
6
7 #incluimos el war generado en la carpeta correspondiente del servidor tomcat
8 ADD ./checkproject.war /usr/local/tomcat/webapps/
9
10 CMD ["/usr/local/tomcat/bin/catalina.sh", "run"]
11
```



Generar la imagen para el frontend: en la carpeta del proyecto de Angular, donde se encuentra el Dockerfile, ejecutar el comando “docker build -t carovz/img_front .”

```
Dockerfile
proyectoFinalAngular > Dockerfile > ...
1 FROM node:12-alpine as build-step
2
3 RUN mkdir -p /app
4
5 WORKDIR /app
6
7 COPY package.json /app
8
9 RUN npm install
10
11 COPY . /app
12
13 RUN npm run build --prod
14
15 #Segunda Etapa
16 FROM nginx:1.17.1-alpine
17 COPY --from=build-step /app/dist/proyectoFinalAngular /usr/share/nginx/html
18
```

Una vez creadas las imágenes, el siguiente paso es crear una red propia de Docker y los contenedores con los comandos que se muestran a continuación:

- Red:

```
docker network create miRed
```

- Contenedor para la base de datos:

```
docker run -d --network miRed --name mi_mysql -e MYSQL_ROOT_PASSWORD=root -e
MYSQL_DATABASE=proyectoFinal -e MYSQL_PASSWORD=root -p 3307:3306
carovz/img_bbdd
```

- Contenedor frontend:

```
docker run -d --name mi_front -p 7200:80 carovz/img_front
```



- Contenedor backend:

```
docker run --network miRed --name mi_back -p 8080:8080 carovz/img_back
```

Por último, introducir en el navegador de preferencia la dirección “<http://localhost:7200>”