

HelpMyBrain

CFGS DAW

CURSO 2022/2023

ÁLVARO REBOLLO GARCÍA

COLEGIO CALASANZ SALAMANCA



1. Introducción.....	3
2. Descripción de la aplicación	4
3. Plan de Empresa	6
4. Tecnologías escogidas y justificación	11
5. Diseño de la aplicación.....	12
5.1. Diagramas y definición de casos de uso.....	12
5.2. Diagramas de clases	15
5.3. Modelo entidad-relación.....	15
6. Arquitectura de la aplicación.....	18
6.1. Estructura del proyecto.....	18
6.2. Librerías externas utilizadas	20
7. Manual de despliegue	Error!
	Marcador no definido.



1. Introducción

HelpMyBrain es una aplicación de terapia online ofrece una solución de salud mental accesible y conveniente para aquellos que no pueden asistir a sesiones presenciales con un terapeuta. La aplicación ofrece una forma sencilla y cómoda de acceder a terapia online, lo que podría ayudar a mejorar la salud mental de las personas que la utilicen. Además, al permitir que los usuarios elijan el terapeuta que prefieran, permite una mayor personalización y adaptación a las necesidades individuales de cada usuario.

Esta aplicación permitiría a los usuarios registrarse y acceder a sesiones de terapia en línea con psicólogos profesionales debidamente certificados. La aplicación podría tener una sección pública donde se describen los servicios y se muestran los psicólogos disponibles, una sección de usuario donde cada usuario puede ver sus citas y comunicarse con su terapeuta, y una sección de administración donde se pueden gestionar los psicólogos, las citas y la configuración de la aplicación.

Al iniciar la aplicación por primera vez, se le pedirá al usuario que introduzca sus datos personales y seleccione un terapeuta. Una vez finalizado este proceso, el usuario podrá acceder a la plataforma y comenzar a utilizarla.



2. Descripción de la aplicación

La aplicación cuenta con tres secciones principales: la sección pública, la sección de usuario y la sección de administración. La sección pública está disponible para todos los usuarios y ofrece información general sobre la aplicación, incluyendo una descripción de los servicios ofrecidos y la lista de terapeutas disponibles.

La sección de usuario está disponible solo para usuarios registrados y permite a los usuarios ver y modificar sus datos personales, así como reservar sesiones de terapia con un terapeuta, y comunicarse .

Por último, la sección de administración está disponible solo para los administradores de la aplicación y permite gestionar la información de los usuarios y terapeutas, así como modificar la configuración general de la aplicación.

Algunas posibles funcionalidades:

- Reserva de sesiones de terapia. Los usuarios registrados podrán reservar sesiones de terapia con el terapeuta de su elección.
- Registro de usuarios. Los usuarios podrán crear una cuenta en la aplicación para acceder a sus datos personales y reservar sesiones de terapia.
- Contacto. La aplicación debería ofrecer un formulario de contacto para que los usuarios puedan ponerse en contacto con el equipo de atención al cliente.
- Gestión de usuarios y terapeutas. Los administradores podrán gestionar la información de los usuarios y terapeutas, así como crear nuevas cuentas para ellos.
- Gestión de sesiones de terapia. Los administradores podrán gestionar las sesiones de terapia reservadas por los usuarios, así como crear nuevas sesiones para un usuario específico.
- Configuración de la aplicación. Los administradores podrán configurar la aplicación, como el nombre de la aplicación, los logos, las imágenes, la contraseña del administrador, etc.
- Configuración inicial. La primera vez que se ejecuta la aplicación se debería configurar los parámetros de la base de datos y cargar datos de muestra para facilitar la edición de la aplicación.



Lógica de negocio: Ofrecer descuentos por número de sesión.



3. Plan de Empresa

1. Clasificación y Caracterización de las Empresas del Sector

Las empresas en el sector de reserva de citas para psicólogos son en su mayoría plataformas digitales que conectan a los pacientes con profesionales de la salud mental. La estructura organizativa puede variar, pero en general, estas empresas tienen departamentos de desarrollo tecnológico, atención al cliente, marketing, y operaciones.

2. Necesidades más Demandadas

Las necesidades más demandadas por las empresas incluyen funcionalidad de alta calidad, usabilidad, accesibilidad, confidencialidad y seguridad de los datos del paciente, y cumplimiento de las regulaciones sanitarias.

3. Oportunidades de Negocio

Existen oportunidades de negocio en la expansión a nuevas regiones geográficas, en la inclusión de servicios adicionales como la telemedicina, y en la adaptación a las necesidades de nichos específicos de pacientes.

4. Tipo de Proyecto Requerido

Se requiere un proyecto de desarrollo de software que construya una plataforma web y móvil intuitiva y segura para la reserva de citas de psicólogos con descuentos.

5. Características Específicas del Proyecto

El proyecto debe incluir funciones como la reserva y cancelación de citas, perfiles de psicólogos, revisiones de usuarios, pagos seguros y un sistema de descuentos.



6. Obligaciones Fiscales, Laborales, y de Prevención de Riesgos

Esta sección requiere más investigación. Sin embargo, es importante tener en cuenta que las aplicaciones de salud pueden tener requisitos regulatorios específicos.

7. Posibles Ayudas o Subvenciones

Existen diversas subvenciones y ayudas disponibles para empresas tecnológicas y startups, especialmente aquellas que se centran en la salud digital. Este tema necesita una investigación más detallada.

8. Guión de Trabajo

La elaboración del proyecto puede seguir las siguientes fases: Investigación de mercado, Diseño y desarrollo, Pruebas, Lanzamiento y Mejora continua.

9. Estudio de Viabilidad Técnica

Esta sección requiere más investigación. Sin embargo, es fundamental contar con un equipo de desarrollo con experiencia en tecnologías web y móviles, y conocimiento en seguridad de datos.

10. Recursos Materiales y Personales Necesarios

Se necesitará un equipo de desarrollo de software, un equipo de marketing, un equipo de soporte al cliente y un equipo de operaciones. Además, se necesitarán servidores, software y hardware para el desarrollo y el mantenimiento de la plataforma.

11. Presupuesto Económico



Descripción	Costo mínimo (EUR)	Costo máximo (EUR)
Investigación y planificación	4,250	8,500
Diseño	8,500	12,750
Desarrollo de la aplicación	42,500	127,500
Pruebas	4,250	8,500
Lanzamiento y marketing	8,500	42,500
Mantenimiento y actualizaciones (anual)	6,375 (15% del costo de desarrollo)	25,500 (20% del costo de desarrollo)
Operaciones y administración (anual)	8,500	17,000



Descripción	Costo mínimo (EUR)	Costo máximo (EUR)
Presupuesto total	82,875	216,250

12. Necesidades de Financiación

La financiación inicial probablemente necesitará cubrir el desarrollo de la plataforma, marketing y operaciones, y puede provenir de inversores, préstamos o fondos propios.

13. Documentación para el Diseño

Se necesitará documentación técnica detallada, incluyendo requisitos de software, arquitectura del sistema, diagramas de flujo y especificaciones de interfaz de usuario.

14. Control de Calidad

El control de calidad puede realizarse a través de pruebas de software, revisiones de código y pruebas de usabilidad.

15. Necesidades de Formación

El equipo de desarrollo necesitará formación en tecnologías específicas utilizadas en el proyecto, y el equipo de atención al cliente necesitará formación en el manejo de consultas de usuarios y problemas técnicos.

16. Calendario de Ejecución



El calendario de ejecución dependerá de la complejidad del proyecto, pero puede esperarse que tome varios meses desde la concepción hasta el lanzamiento.

17. Mantenimiento

El mantenimiento incluirá la corrección de errores, la actualización de software y hardware, y la mejora continua basada en el feedback de los usuarios.



4. Tecnologías escogidas y justificación

En este proyecto, para el backend se utiliza Java como lenguaje de programación con el framework Spring Boot. Spring Boot ofrece una forma rápida y accesible de desarrollar aplicaciones Spring, simplificando la configuración y el despliegue. Además, se utiliza Spring Data JPA para facilitar el acceso a los datos en las aplicaciones JPA y Spring Security para la autenticación y el control de acceso.

Para almacenar los datos se utiliza una base de datos relacional con un sistema de gestión MySQL, utilizando el conector MySQL Connector/J para la comunicación con la base de datos. Además, se utiliza Thymeleaf, un motor de plantillas de Java, para procesar y generar HTML, XML, JavaScript, CSS y otros formatos de texto.

En cuanto al frontend, el framework escogido de Javascript es Vue.js, que es fácil de integrar con otros proyectos y bibliotecas y es capaz de potenciar las aplicaciones de una sola página. Se utiliza Vue Router para la construcción de aplicaciones de una sola página y Axios para hacer solicitudes HTTP desde el navegador. Además, se utiliza Bootstrap para el diseño responsivo y Vue Carousel para la implementación de carruseles.

Estas tecnologías se han seleccionado no solo por su relevancia en el ámbito educativo, sino también por su amplio uso en la industria, lo que aumenta la empleabilidad de los estudiantes una vez finalizado el curso.



5. Diseño de la aplicación

5.1. Diagramas y definición de casos de uso

Caso de uso 1: Registro de usuario

- **Actor:** Usuario
- **Descripción:** El usuario se registra en la aplicación.
- **Precondiciones:** El usuario no está registrado en la aplicación.
- **Pasos:**
 1. El usuario selecciona la opción de registro.
 2. El usuario introduce la información requerida para el registro.
 3. El sistema valida la información y registra al usuario.
 4. El sistema notifica al usuario que el registro fue exitoso.

Caso de uso 2: Inicio de sesión de usuario

- **Actor:** Usuario
- **Descripción:** El usuario inicia sesión en la aplicación.
- **Precondiciones:** El usuario está registrado en la aplicación.
- **Pasos:**
 1. El usuario introduce su nombre de usuario y contraseña.
 2. El sistema valida la información y permite al usuario iniciar sesión.
 3. El sistema redirige al usuario a su panel personal.

Caso de uso 3: Actualizar información personal

- **Actor:** Usuario
- **Descripción:** El usuario actualiza su información personal.
- **Precondiciones:** El usuario ha iniciado sesión en la aplicación.
- **Pasos:**
 1. El usuario selecciona la opción para actualizar su información personal.
 2. El usuario introduce la nueva información.
 3. El sistema valida y actualiza la información en la base de datos.
 4. El sistema notifica al usuario que la actualización fue exitosa.



Caso de uso 4: Reservar cita con psicólogo

- **Actor:** Usuario
- **Descripción:** El usuario reserva una cita con un psicólogo.
- **Precondiciones:** El usuario ha iniciado sesión en la aplicación, el psicólogo está disponible para citas.
- **Pasos:**
 1. El usuario selecciona la opción para reservar una cita.
 2. El usuario selecciona un psicólogo y una fecha y hora para la cita.
 3. El sistema valida la disponibilidad del psicólogo y reserva la cita.
 4. El sistema aplica cualquier descuento disponible basado en el historial de citas del usuario.
 5. El sistema notifica al usuario que la reserva de la cita fue exitosa.

Caso de uso 5: Registro de psicólogo

- **Actor:** Psicólogo
- **Descripción:** El psicólogo se registra en la aplicación.
- **Precondiciones:** El psicólogo no está registrado en la aplicación.
- **Pasos:**
 1. El psicólogo selecciona la opción de registro.
 2. El psicólogo introduce la información requerida para el registro.
 3. El sistema valida la información y registra al psicólogo.
 4. El sistema notifica al psicólogo que el registro fue exitoso.

Caso de uso 6: Actualizar información personal y disponibilidad

- **Actor:** Psicólogo
- **Descripción:** El psicólogo actualiza su información personal y su disponibilidad para citas.
- **Precondiciones:** El psicólogo ha iniciado sesión en la aplicación.
- **Pasos:**
 1. El psicólogo selecciona la opción para actualizar su información personal y disponibilidad.
 2. El psicólogo introduce la nueva información y disponibilidad.
 3. El sistema valida y actualiza la información en la base de datos.



4. El sistema notifica al psicólogo que la actualización fue exitosa.

Caso de uso 7: Establecer precio por hora de la cita

- **Actor:** Psicólogo
- **Descripción:** El psicólogo establece su tarifa por hora para las citas.
- **Precondiciones:** El psicólogo ha iniciado sesión en la aplicación.
- **Pasos:**
 1. El psicólogo selecciona la opción para establecer su tarifa por hora.
 2. El psicólogo introduce su tarifa por hora.
 3. El sistema valida y actualiza la tarifa en la base de datos.
 4. El sistema notifica al psicólogo que la actualización fue exitosa.

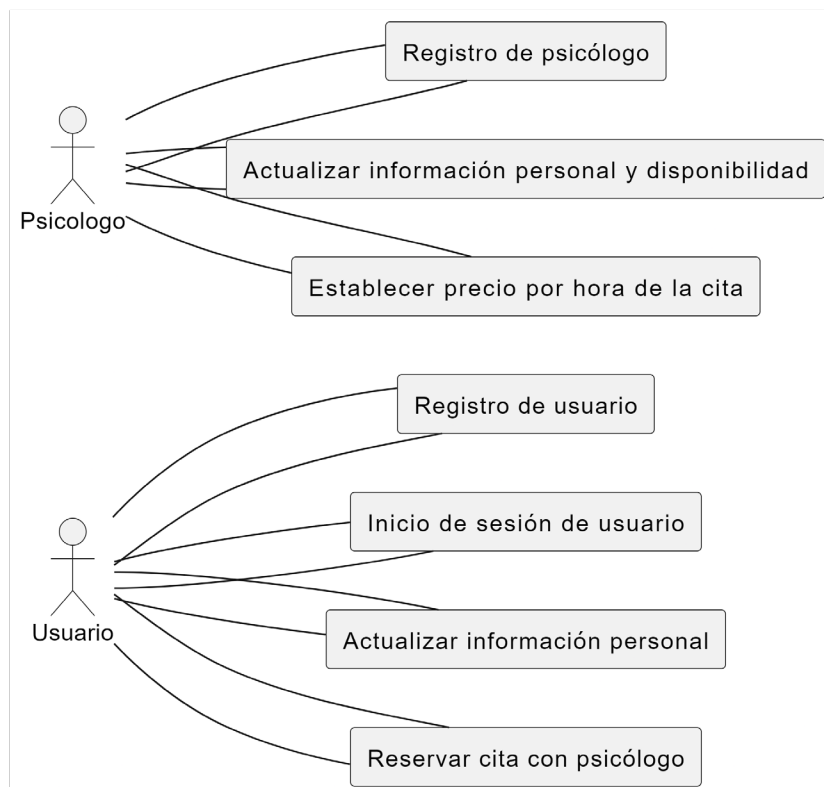


Diagrama de Casos de Uso 1



5.2. Diagramas de clases

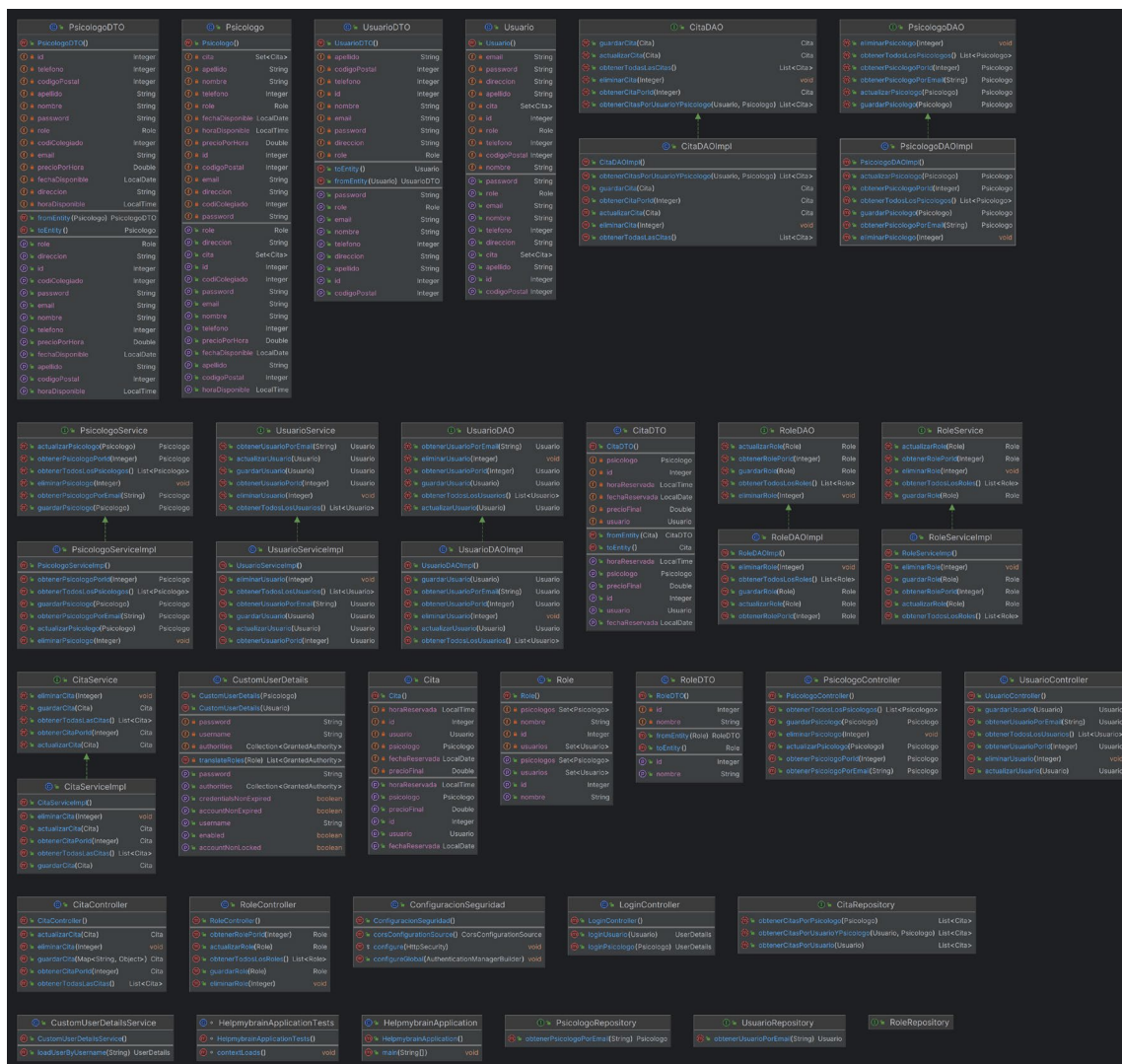
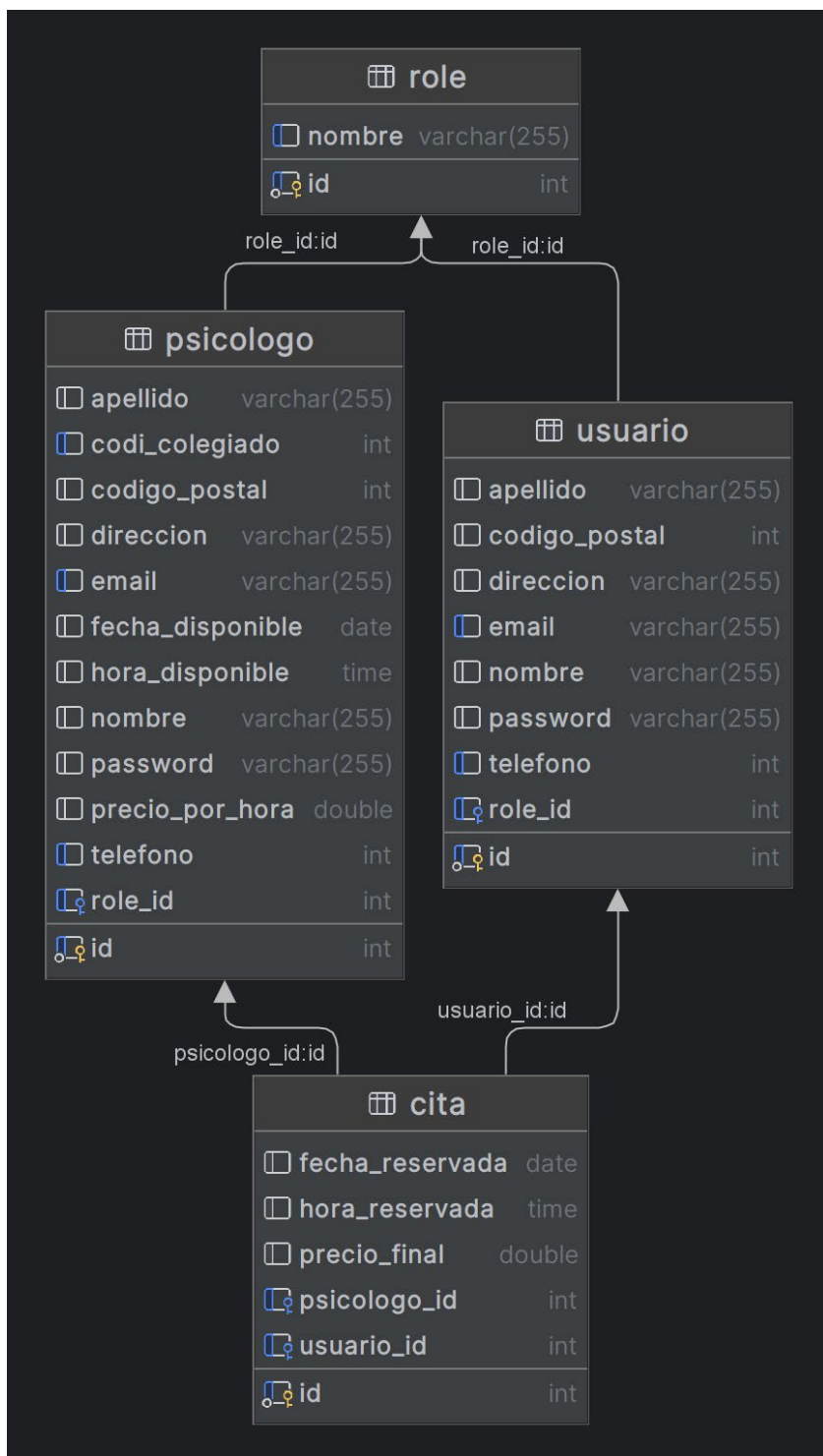


Diagrama de Clases 1

5.3. Modelo entidad-relación



Modelo Entidad Relación 1



La estructura de helpmybrain que consta de cuatro tablas: cita, psicólogo, role, y usuario. Aquí está una descripción breve de cada tabla:

cita: Esta tabla representa una cita reservada por un usuario con un psicólogo. Contiene información sobre la fecha y hora de la cita, el precio final de la cita, y las identificaciones del psicólogo y del usuario que están involucrados en la cita.

psicólogo: Esta tabla representa a un psicólogo en el sistema. Contiene información personal y de contacto del psicólogo, como el nombre, apellido, dirección, email, y teléfono. También incluye información relevante para el sistema de citas, como el código de colegiado, el precio por hora de las citas, y la disponibilidad del psicólogo.

usuario: Esta tabla representa a un usuario del sistema. Contiene información personal y de contacto del usuario, similar a la tabla Psicólogo. Los usuarios pueden ser clientes que reservan citas o pueden tener otros roles en el sistema.

role: Esta tabla representa el rol que un usuario o psicólogo puede tener en el sistema. Por ejemplo, un usuario puede ser un cliente, un administrador, o un psicólogo.

Las relaciones entre las tablas se establecen mediante claves foráneas. Por ejemplo, la tabla cita tiene una clave foránea psicologo_id que hace referencia a la tabla psicólogo, indicando que cada cita está asociada con un psicólogo específico. De manera similar, cada cita está asociada con un usuario específico, y cada usuario y psicólogo tiene un rol específico en el sistema.



6. Arquitectura de la aplicación

6.1. Estructura del proyecto

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

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 aplicar los descuentos pertinentes.
- 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 Vue Cli para Vue, 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:



App.vue: Este es el componente raíz de la aplicación Vue.js. Todos los demás componentes se anidan dentro de este componente.

assets: Este directorio contiene los archivos estáticos como imágenes, estilos, etc.

components: Este directorio contiene los componentes Vue.js que se utilizan en la aplicación. Los componentes son bloques de código reutilizables que pueden tener su propio estado, marcas y comportamientos.

main.js: Este es el punto de entrada de la aplicación Vue.js. Aquí es donde se crea la instancia de Vue y se monta en el DOM.

router: Este directorio contiene la configuración del enrutador Vue. El enrutador Vue permite navegar entre páginas en la aplicación de una sola página.

services: Este directorio contiene los servicios que se utilizan en la aplicación. Los servicios pueden incluir la lógica para llamar a las APIs, realizar cálculos, etc.

views: Este directorio contiene los componentes Vue que representan las diferentes vistas o páginas de la aplicación.

En la carpeta frontend:

package.json: configuración necesaria para poder ejecutar la aplicación. Describe, entre otras cosas, las dependencias que necesita el proyecto para funcionar.



6.2. 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
- spring-security

Vue:

- Bootstrap 5
- Axios 1.4.0
- Vue-router
- Vue-carousel



colegio
CALASANZ
SALAMANCA

Paseo de Canalejas, 139-159
37001 Salamanca
Tel.: 923 26 79 61 | Fax: 923 27 04 54
www.calasanzsalamanca.es
CENTRO PRIVADO CONCERTADO