



Universidad San Carlos de Guatemala

Centro Universitario de Occidente

División Ciencias de la Ingeniería

Ingeniería en Ciencias y Sistemas

Laboratorio de Manejo e Implementación de Archivos

Estudiante: Diego José Avila Estrada

Carnet: 201731120

Proyecto 1

Manual Técnico

- **Tecnologías usadas:**

- PostgreSQL
- Java
- Spring Boot
- Html
- Css
- Javascript
- Fetch API



PostgreSQL PostgreSQL: Es la base de datos relacional que se usó para el almacenamiento de los datos de la aplicación.



Java: Se usó el lenguaje de programación Java para realizar la parte backend de la aplicación, todo lo relacionado con la comunicación con la Base de Datos.



Spring Boot: Es un framework de Java que se usó para la facilitación del desarrollo de la aplicación en la parte backend y para la creación de una API REST, usando las dependencias de JPA para la base de datos , SPRING WEB para poder crear la API REST, etc.



HTML 5 : Se usó para el front de la aplicación, ya que por ser una aplicación web, definitivamente html 5 debía usarse para la estructura básica de cada ventana a mostrar.



JavaScript: Se usó para la parte front de la aplicación, y su función es darle dinamismo a la interacción del usuario con la aplicación, al igual que validar formularios y consumir la API REST de la aplicación creada por Java Spring Boot.



Fetch API JavaScript: Fetch es una interfaz de JavaScript que nos ayuda a realizar peticiones HTTP a algún servidor o API, funciona de manera asíncrona y por ello trabaja bajo promesas, lo que hace que permite hacer peticiones sin necesidad de recargar la página.

Paquetes creados en Java (BackEnd):

Controllers: Contiene las clases que representan los controladores que son los endpoints para que el usuario se comuniquen con mi API REST.

Service: Contiene las clases que se comunican con los controladores y con la base de datos.

Repository: Contiene las clases que se comunican directamente con la base de datos, en este paquete guardo mis entidades que representan las tablas creadas en mi base de datos, así como las interfaces que me permiten realizar los CRUD de cada entidad de mi base de datos.

DTO: Contiene las clases DTO y estas me permiten transferir datos tanto para enviar como para recibir, sus siglas significan Data Transfer Object, es una manera de mapear, en este caso, objetos JSON que recibo y envío a través de mi API.

Paquetes creados en JavaScript (FrontEnd):

Vistas: Contiene todas las estructuras HTML de cada ventana que el usuario va a visualizar a la hora de interactuar con la aplicación.

CSS: Contiene todos los archivos CSS que le da estilos a cada estructura HTML.

JS: Contiene los archivos JavaScript que se usan en cada ventana o en cada estructura HTML, para darle dinamismo a la misma, y también contiene la lógica para recibir y enviar peticiones HTTP a la API creada en Java.