1. Overview

Name: Geny Marcela Vargas Suarez
Name: Santiago Martínez Yara
Name: Miguel Ánjel Salazar Jaramillo
Name: Angela Consuelo torres Trujillo
Name: Esteban Henao Sánchez

1.1 Project Name: Estramipyme
1.2 Description: Estramipyme is a web platform designed to help entrepreneurs identify and work on areas for improvement in their businesses through a diagnostic test and subsequent follow-up. The test results highlight strengths and weaknesses, aiding in better business decisions and management. The platform also distinguishes between Administrator and User roles, allowing action-taking based on the user type accessing the platform.

2. Minimum Viable Product (MVP) Objectives:

2.1 Business Diagnostic: offers a test that enables entrepreneurs to get an initial evaluation of their strengths and areas for improvement.
2.2 User and Role Management: a role system for administrators and users, allowing interactions based on permissions.
2.3 User Tracking: provides statistical data about registered users, tests taken, and other actions performed within the platform.

3.Key MVP Features

3.1 Registration and Login: provides registration and login, distinguishing between Users and Administrators.
3.2 Diagnostic Test: a qualitative exam that highlights aspects to improve for greater business efficiency and guides users through different learning plans based on their results.
3.3 Progress Tracking: user counter for those who complete the test.
3.4 Creation of Teachers and Students: on the Back End, provides functionality to create Teachers and Students and modify their information as needed.

4. Request Flow and Spring Boot Architecture

Estramipyme uses a chain-type approach to manage and process requests made to the platform, employing CRUD (Create, Read, Update, Delete) principles in each interaction with the database.

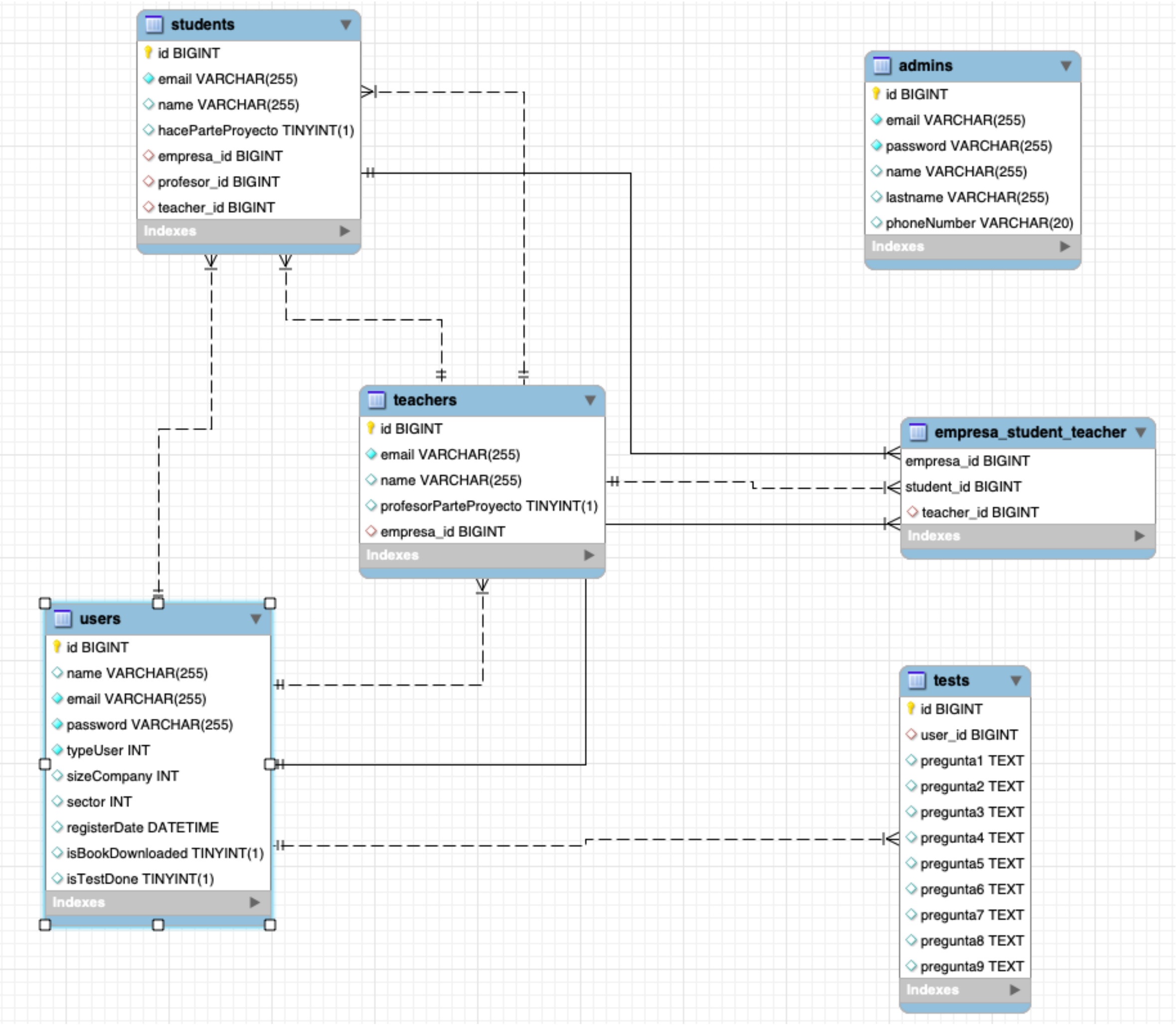
4.1 Controllers: Controllers are the entry points for HTTP requests. These access points receive requests and delegate processing to the service layer. Each controller is associated with a route and offers specific methods for each CRUD operation, allowing users to perform tasks such as:

- Creating new users, projects, and test records.
- Reading data on assigned projects, test results, and user profiles.
- Updating user information or project assignments.
- Deleting records based on user permissions.

4.2 Services: The main business logic resides in the services that control CRUD operations, call repositories to access or modify database data, and implement verifications. Services receive requests from controllers and execute the necessary logic to complete the required operation.

4.3 Models: Models represent system entities, such as User, Project, and Test. Each model is mapped to a specific table in the MySQL database via JPA (Java Persistence API) annotations, facilitating data persistence.

4.4 Repositories: Repositories are interfaces that manage CRUD operations directly with the database. They use JPA and Spring Data tools to simplify queries and transactions with MySQL, allowing efficient data access and manipulation.



5. Database

5.1 Data Management with MySQL: Data is stored and managed in MySQL, where each platform entity (such as Users, Projects, and Tests) has its corresponding table.

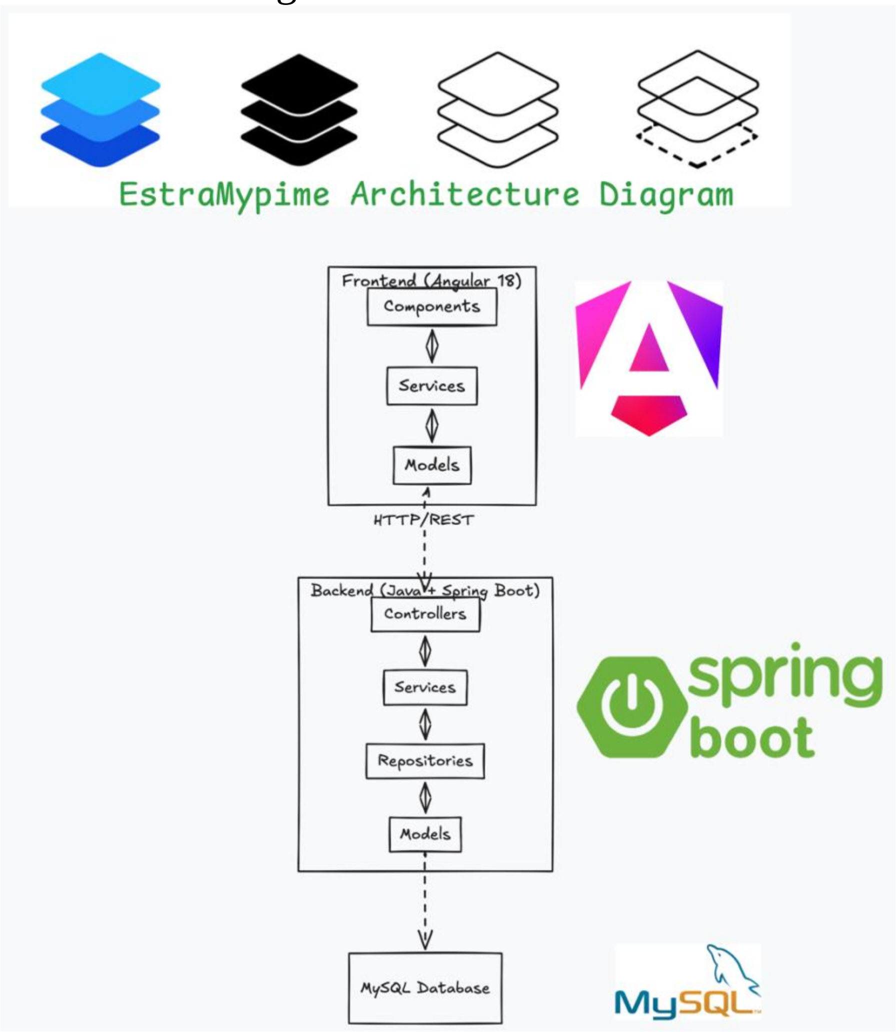
6. Technologies Used

6.1 Front End: User interfaces were built on Angular 18, with TypeScript as the programming language, Tailwind CSS as the style library, SCSS as the style pre-processor, ChartJS for dynamic charts, and HTML5 for layout.
6.2 Back End: Java 17 was used as the programming language with the Spring Boot Framework to make a more robust application and speed up development. Dependencies included Lombok to reduce code repetition, JPA, and the MySQL Driver for database connections and data management with MySQL.
6.3 Version Control: Git was used for version management, with GitHub as the repository.

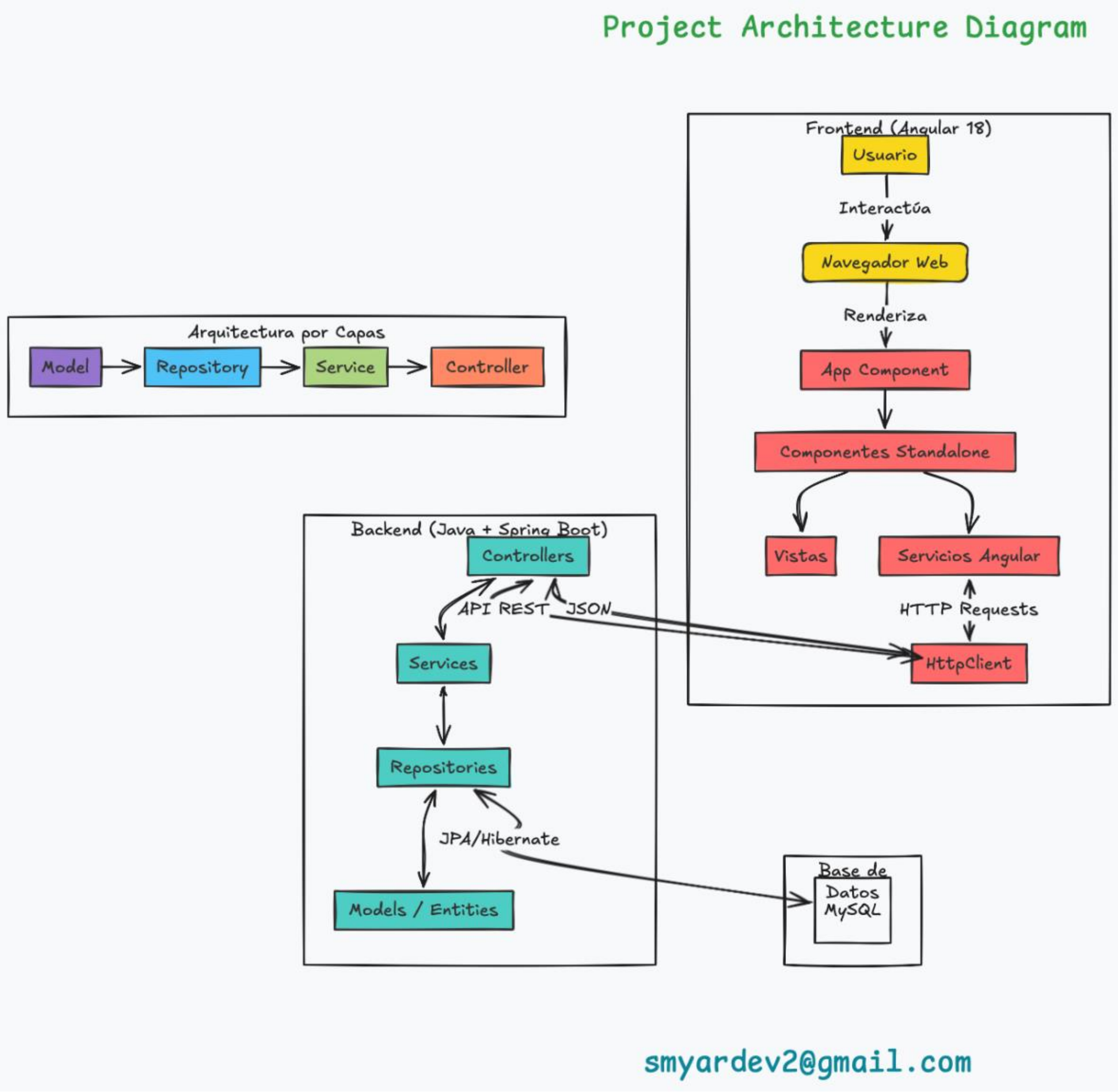
Postman Collections

https://www.postman.com/gold-escape-545722/estramypimegrupo3/folder/jzukjtx/teachers?action=share&creator=28945019&ctx=documentation

Architecture diagram



Detailed flowchart


Project Architecture Diagram
Frontend (Angular 18)
Usuario
Interactúa
Navegador Web
Renderiza
App Component
Componentes Standalone
Vistas
Servicios Angular
HTTP Requests
HttpClient
Arquitectura por Capas
Model
Repository
Service
Controller
Backend (Java + Spring Boot)
Controllers
API REST
JSON
Services
Repositories
JPA/Hibernate
Models / Entities
Base de Datos MySQL
smyardev2@gmail.com