

Prueba práctica Java

Una vez descargado el proyecto encontraremos dos carpetas, ValidBackend con los desarrollos de Backend, realizado en Java con Spring Boot y ValidFrontend realizado con Angular.

Proyecto Backend

Para desplegar el proyecto Java y observar la funcionalidad, se debe importar el proyecto maven adjunto al correo en el IDE recomendado (STS) y desplegar (<http://localhost:8080>).

Aplicación Backend desarrollada en Spring Boot Framework, diseñada con el patrón de diseño MVC, DTO, Facade para la exposición de servicios web de tipo REST. Los paquetes de domain, contiene las entidades del dominio del problema con sus respectivos DTO, el cual nos facilita la creación del objeto plano y él envío o recuperación de datos del servidor en una sola invocación.G

Los paquetes core contienen el aspect el cual nos ayuda en el control de los logs de la aplicación, en config están las configuraciones para validar os datos nulos de los Json, y las configuraciones de documentación para el uso de Open api (<http://localhost:8080/swagger-ui.html>) y las clases que nos permiten el manejo de excepciones de Rest.

Para probar la funcionalidad de open api agregar a pom.xml la siguiente dependencia, actualizar maven y des comentar las líneas de la clase PersonDTO en el paquete domain.dto y la clase OpenApiConfig en el paquete core.config.documentation (<https://github.com/Andres2512/Valid>)

```
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-ui</artifactId>
  <version>${springdoc-openapi-ui.version}</version>
</dependency>
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-webmvc-core</artifactId>
  <version>${springdoc-openapi-ui.version}</version>
</dependency>
<dependency>
  <groupId>io.swagger.core.v3</groupId>
  <artifactId>swagger-annotations</artifactId>
  <version>2.1.1</version>
</dependency>
```

Los paquetes de repositorio, contienen la abstracción de Spring Framework empleando JPA Data para administrar la capa de persistencia de la aplicación incluyendo un método que emplea queries con jpql y la implementación del patrón de diseño Facade el cual nos ayuda a reducir la complejidad del sistema con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos.

En el paquete service, se encuentran las clases que orquestan el negocio de la aplicación, al ser un CRUD, la lógica de la aplicación está alineada a las operaciones básicas sobre una base de datos.

En el paquete de web api, se encuentran las clases que realizan la administración de los endpoints de la aplicación, exponiéndose en webservices de tipo REST con manejo de excepciones por medio de REST.

En la carpeta test encontraremos las clases correspondientes a las pruebas, estas se realizaron utilizando la herramienta Junit.

Proyecto Frontend

Para desplegar el proyecto Angular, se debe abrir la consola en la dirección del proyecto (carpeta ValidFrontendP/Valid-app) y ejecutar el comando `ng update --all --force`, el cual permitirá actualizar las dependencias relacionadas al proyecto. Luego se desplegará con el comando `ng serve -o`, lo que permitirá visualizar el aplicativo en el navegador (<http://localhost:4200>).

El proyecto angular tiene una estructura de módulos, separándolos de los componentes genéricos como el header o el footer, en donde encontraremos los respectivos componentes específicos relacionados al módulo, lo cual facilitara el mantenimiento de la aplicación.