

Evidencia de desempeño: GA8-220501096-AA1-EV01 desarrollar software a partir de la integración de sus módulos componentes:

Proyecto de Api seleccionado: tienda virtual

A continuación, veremos un paso a paso de cómo crear una API para una tienda virtual.

1. Codificación en Spring Boot:

- Codificar cada módulo específico en Spring Boot para el backend, implementando la lógica de negocio relacionada con la gestión de tareas.
 - Empezamos definiendo la estructura del proyecto en Spring Boot utilizando Spring Initializr, haciendo las configuraciones necesarias para crear un nuevo proyecto. Iniciamos encogiendo el tipo del proyecto es Maven, lenguaje Java, el Spring Boot ponemos la última versión 3.2.0, ponemos los metadatos del proyecto, el tipo de Packaging es Jar y la versión del Java será 17.

2. Control de Versiones:

- Utilizar un repositorio de control de versiones como GIT para gestionar y rastrear el código fuente del backend.
 - En este punto descargamos e instalamos el Git en el equipo, iniciamos un nuevo repositorio Git utilizando el comando "Git Init". Esto crea un repositorio local en el equipo.

3. Gestión de Dependencias:

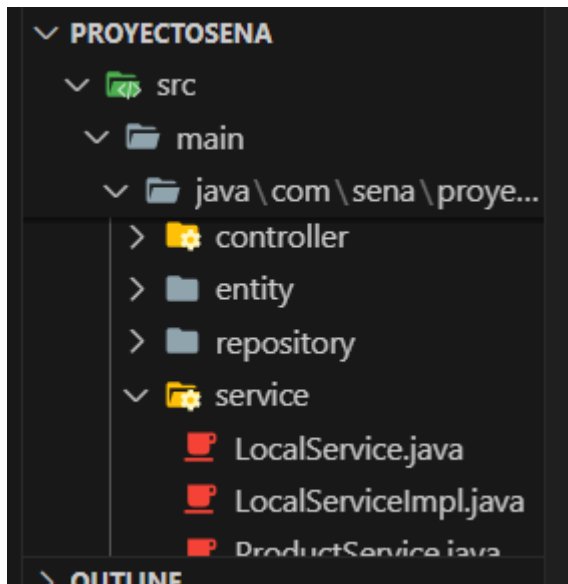
- Determinar las librerías necesarias para el backend, como Hibernate para la capa de persistencia en Spring Boot.
 - Creamos las dependencias en el Spring Boot Initializr, My SQL Driver, Spring Data JPA, Lombok, Spring Boot Dev Tools y Spring Web, estas serían todas las librerías necesarias para comenzar nuestro proyecto. A continuación, las descargamos y las guardamos en donde las tengamos a la mano.

4. Frameworks en el Backend:

- Utilizar Spring Boot como framework principal para el desarrollo del backend.
 - Utilizar Spring Boot como Framework principal para nuestro proyecto de desarrollo es una excelente elección debido a sus amplios beneficios: Productividad y facilidad de uso, configuración simplificada, soporte para microservicios y amplia comunidad y recursos.

5. Componentes Reutilizables:

- Diseñar y desarrollar componentes reutilizables en el backend, especialmente aquellos relacionados con la gestión de tareas y asignación.
 - Se crearon los componentes reutilizables



```
server.port=8090
spring.sql.init.platform=mysql
spring.datasource.url=jdbc:mysql://localhost:3306/bd_proyecto_sena
spring.datasource.username=root
spring.datasource.password=
spring.jpa.show-sql=true
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update
```

6. Buenas Prácticas de Codificación:

- Aplicar buenas prácticas de escritura de código específicamente para el backend, siguiendo estándares y convenciones.

```
import java.util.List;

@RestController
@RequestMapping("api/local")
public class LocalController {

    @Autowired
    LocalService localService;

    @GetMapping("/findAll")
    public List<Local> findAll(){
        return localService.findAll();
    }
}
```

```

package com.sena.proyectoSena.controller;

import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import com.sena.proyectoSena.entity.Product;

import com.sena.proyectoSena.service.ProductService;

import java.util.List;

@RestController
@RequestMapping("api/Product")
public class ProductController {

    @Autowired
    ProductService productService;

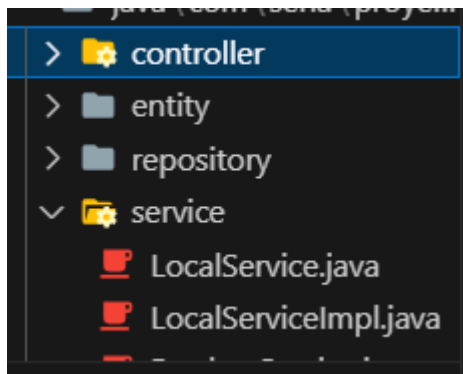
    @GetMapping("/findAll")
    public List<Product> findAll(){
        return productService.findAll();
    }
}

```

Aplicamos buenas prácticas de escritura en nuestro proyecto.

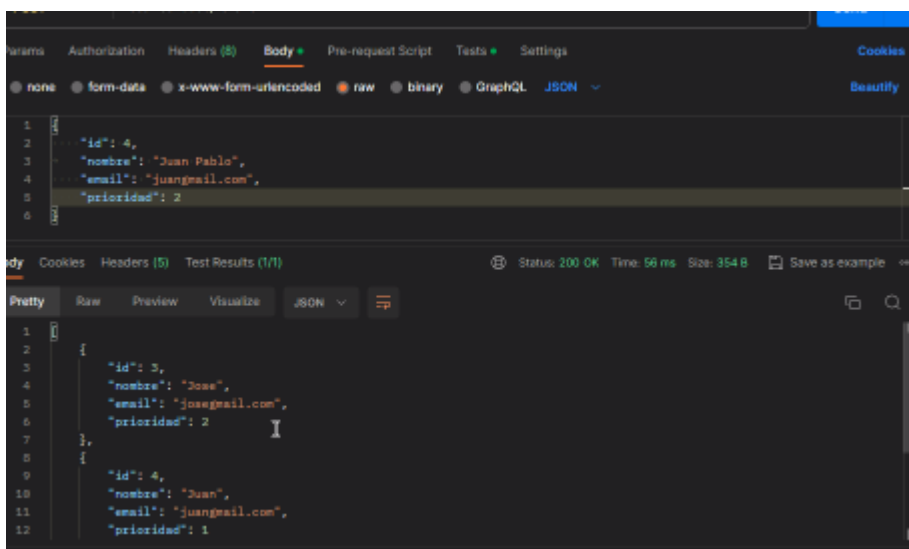
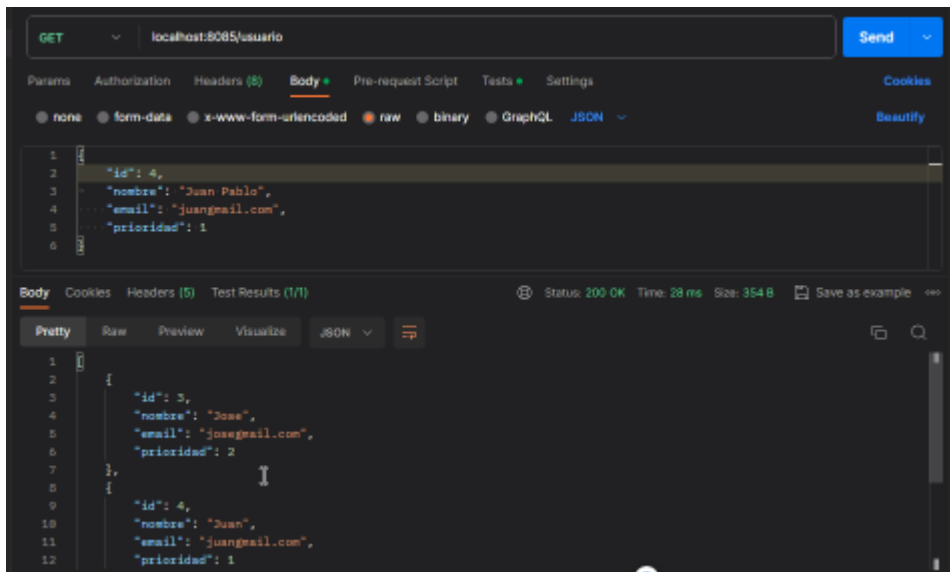
7. Patrones de Diseño:

- Aplicar patrones de diseño, como el patrón MVC, adaptados a la arquitectura del software en el backend.
 - Aplicamos patrones de diseño (MVC) es una práctica común y beneficiosa en el desarrollo de aplicaciones web.
 -



8. Pruebas Unitarias:

- Realizar pruebas unitarias exhaustivas para cada módulo del backend, garantizando la funcionalidad y la integridad de las operaciones con POSTMAN.
 - Para garantizar la funcionalidad y la integridad de las operaciones en el backend, es crucial realizar pruebas unitarias exhaustivas.



9. Configuración del Servidor:

- Configurar el servidor para el backend (Spring Boot), asegurando su correcto funcionamiento y rendimiento.
 - Seleccionar un Servidor de Aplicaciones: Spring Boot incluye un servidor web embebido por defecto, puedes utilizar uno de los servidores para ejecutar la aplicación sin necesidades de configuraciones adicionales
 - Optimizar la Configuración del Servidor Embebido: si decides utilizar el servidor embebido proporcionado por Spring Boot, puedes optimizar su configuración para mayor rendimiento

10. Documentación del Ambiente:

- Documentar detalladamente el ambiente de desarrollo y pruebas del backend, incluyendo configuraciones y requisitos específicos del backend.
 - Configuración del Entorno de Desarrollo: descargamos e instalamos el JDK, para esto se accede al sitio web de Oracle u Open JDK, se descarga la versión más reciente del JDK compatible con el sistema operativo.
 - Instalación del JDK: Ejecuta el instalador descargado y sigue las instrucciones del sistema para completar la instalación del JDK en el pc.