**Prueba Técnica Backend LINKTIC**

**Arquitectura**

**Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.**

**Componentes:**

* **Cliente (navegador o Postman): Interactúa con el sistema.**
* **Gateway (microservicio-gateway):** Enrutador de peticiones hacia los microservicios.
* **Eureka Server (microservicio-eureka):** Registro y descubrimiento de servicios.
* **Microservicio de Productos (microservicio-productos):** Gestiona productos.
* **Microservicio de Inventarios (microservicio-inventarios):** Gestiona inventarios y consulta productos

**Tecnologías usadas:**

* **Spring Boot**
* **Spring Cloud Eureka**
* **Spring Cloud Gateway**
* **Docker & Docker Compose**
* **MySQL**
* **RestTemplate o FeignClient**

**Endpoints:**

**Microservicio-Productos**

* **Consutar productos**

[**http://localhost:8090/Swagger/OpenAPI/productos/**](http://localhost:8090/Swagger/OpenAPI/productos/)

* **Consultar productos por id**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

[**http://localhost:8090/Swagger/OpenAPI/productos/2**](http://localhost:8090/Swagger/OpenAPI/productos/2)

[**http://localhost:8090/Swagger/OpenAPI/productos/3**](http://localhost:8090/Swagger/OpenAPI/productos/3)

[**http://localhost:8090/Swagger/OpenAPI/productos/4**](http://localhost:8090/Swagger/OpenAPI/productos/4)

* **Editar producto por id**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

**ejemplo en json**

**{**

**"nombre": "MOUSE ZOWIE EC3",**

**"precio": 880000**

**}**

**Desde Postman**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

* **Eliminar producto**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Microservicio-Inventarios**

* **Consultar cantidad**

[**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1**](http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1)

* **Actualizar cantidad por id de producto**

[**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1?nuevaCantidad=70**](http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1?nuevaCantidad=70)

**Requisitos previos en otro computador**

**Instalar Git**

**https://git-scm.com/downloads**

**Instalar Docker Desktop**

**https://www.docker.com/products/docker-desktop/**

**Para windows instale DOCKER AMD64**

**Asegúrate de que Docker esté corriendo o ejecutándose en el computador.**

**No es necesario logearse con cuenta de gmail**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Ingresar al repositorio mio**

**https://github.com/Loping08/microservicios-proyecto**

**PASO A PASO desde PowerShell**

**1. Clonar el repositorio**

**- Ingresar a PowerShell desde windows**

**- ubicarse en la raiz por ejemplo PS C:\Users\juli\Documents>**

**- ejecutar la siguiente linea de comando para clonar el proyecto:**

**git clone** [**https://github.com/Loping08/microservicios-proyecto**](https://github.com/Loping08/microservicios-proyecto)

**Texto

El contenido generado por IA puede ser incorrecto.**

**- verificar la clonacion en la raiz ejemplo:**

**C:\Users\julia\Documents\microservicios-proyecto**

**- ejecutar cd microservicios-proyecto ejemplo:**

**PS C:\Users\julia\Documents> cd microservicios-proyecto**

**- estando alli**

**2. Verifica que existe un docker-compose.yml**

**-dir**

**-ejemplo: PS C:\Users\julia\Documents\microservicios-proyecto> dir**

**Texto

El contenido generado por IA puede ser incorrecto. - (Si ves el archivo docker-compose.yml, todo está bien)**

**3. Levantar los servicios con Docker Compose**

**-docker-compose up --build**

**-ejemplo: PS C:\Users\julia\Documents\microservicios-proyecto> docker-compose up –build**

**Texto

El contenido generado por IA puede ser incorrecto.**

**4. Esperar a que se levanten los Docker compose**

**- debe salir algo como esto:**

**Texto

El contenido generado por IA puede ser incorrecto.**

**En el programa DOCKER debe verse de esta forma**

**Interfaz de usuario gráfica, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

**6. comenzar a ejecutar desde Postman o el navegador los EndPoints**

**- Nota: véase el documento Prueba Técnica Backend**

**EndPoints:**

**MICROSERVICIO PRODUCTOS**

**- Consutar productos(GET)**

**http://localhost:8090/Swagger/OpenAPI/productos/**

**- Consultar productos por id(GET)**

**http://localhost:8090/Swagger/OpenAPI/productos/1**

**http://localhost:8090/Swagger/OpenAPI/productos/2**

**http://localhost:8090/Swagger/OpenAPI/productos/3**

**http://localhost:8090/Swagger/OpenAPI/productos/4**

**- Editar producto por id (PUT)**

**http://localhost:8090/Swagger/OpenAPI/productos/1**

**ejemplo en json**

**{**

**"nombre": "MOUSE ZOWIE EC3",**

**"precio": 880000**

**}**

**MICROSERVICIO INVENTARIOS**

**- Consultar cantidad(GET)**

**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1**

**- Actualizar cantidad por id de producto(PUT)**

**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1?nuevaCantidad=70**

**NOTA:** producto/id modificar id por el número del producto y la variable nuevaCantidad=x, x por el número de cantidad que quiere modificar

**NOTA:** LO PUEDE HACER DESDE POSTMAN O DESDE EL NAVEGADOR

**LEER:** en la carpeta README de este repositorio se encuentra la documentación con diagrama de arquitectura y ejemplos en postman con sus respectivos endpoints

**RECOMENDACION:** DESDE POSTMAN PUEDE EJECUTAR MEJOR LOS PUT con el body/raw formato en JSON aplica para editar de producto pero para actualizar la cantidad no es necesario note que puede hacerlo enviando la variable nuevaCantidad=x

**BASE DE DATOS**

En el repositorio podrá observar una carpeta mysql-init la cual contiene el archivo [init.sql](https://github.com/Loping08/microservicios-proyecto/blob/main/mysql-init/init.sql) en el cual esta previamente la creación de la base de datos y tablas con sus respectivos insert de prueba.

microservicios-proyecto/mysql-init/init.sql

Esto con el fin de tener data previa al momento de levantar los Docker Compose

Nota: no es necesario que tenga usted MySQL en su computador ya que gracias a la configuración de **docker-compose.yml se realiza la persistencia de datos en una instancia mysql-db**

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.

**PRUEBAS CON POSTMAN**

**Microservicio-productos**

* **Consultar todos los productos**

[**http://localhost:8090/Swagger/OpenAPI/productos/**](http://localhost:8090/Swagger/OpenAPI/productos/)

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

* **Consultar producto por id**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

* **Crear producto**

[**http://localhost:8090/Swagger/OpenAPI/productos/**](http://localhost:8090/Swagger/OpenAPI/productos/)

**en el body seleccionar raw y formato JSON y enviar los parámetros necesarios para el producto.**

**Ejemplo de JSON:**

**{**

**"nombre": "MOUSE ZOWIE EC3",**

**"precio": 750000**

**}**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

* **Editar Producto por id**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

* **Eliminar Producto**

[**http://localhost:8090/Swagger/OpenAPI/productos/1**](http://localhost:8090/Swagger/OpenAPI/productos/1)

**Microservicio-Inventarios**

* **Consultar cantidad de producto por id**

[**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1**](http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1)

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

[**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/3**](http://localhost:8090/Swagger/OpenAPI/inventarios/producto/3)

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

* **Actualizar cantidad de inventario por producto id y variable**

[**http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1?nuevaCantidad=70**](http://localhost:8090/Swagger/OpenAPI/inventarios/producto/1?nuevaCantidad=70)

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**