PLATAFORMA DE CONTROL DE PEDIDOS MULTIPLATAFORMA

INTRODUCCIÓN Y CONTEXTO

1.1 Descripción del Proyecto

La empresa **MultiPedidos S.A.** requiere una solución de arquitectura distribuida para gestionar pedidos de distintos clientes y proveedores en varias regiones, operando con múltiples bases de datos por políticas internas y requerimientos técnicos.

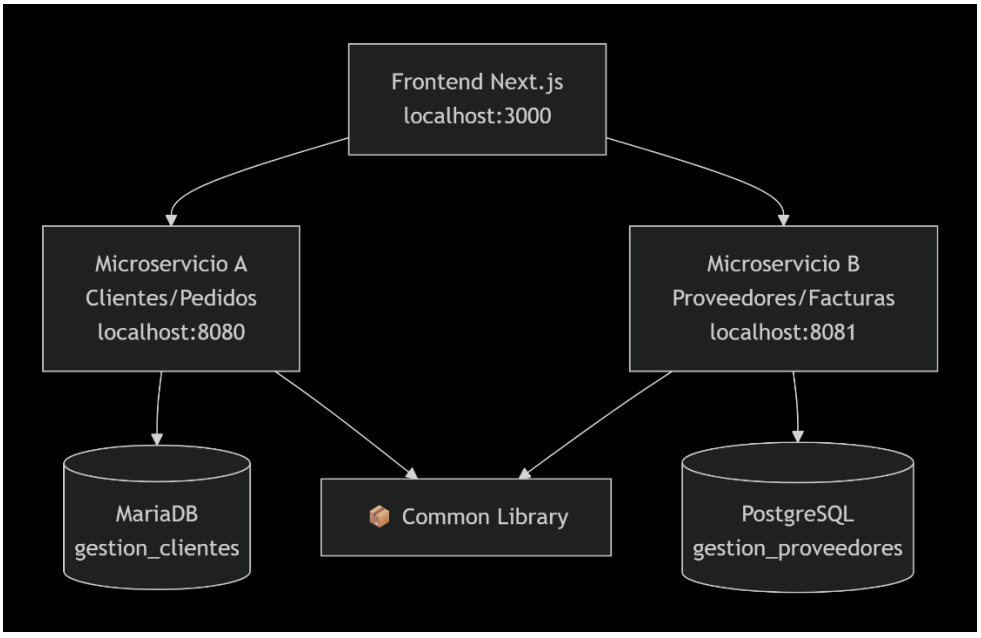
1.2 Objetivos del Sistema

- Gestionar clientes y pedidos en MariaDB
- Gestionar proveedores y facturación en PostgreSQL
- Compartir lógica de negocio común mediante librería Maven
- Proporcionar dashboard administrativo unificado en Next.js

---

2. ARQUITECTURA DEL SISTEMA

2.1 Diagrama de Arquitectura General



2.2 Especificaciones Técnicas por Componente

Componente	Tecnología	Base de Datos	Puerto	Responsabilidad
Frontend Dashboard	Next.js 14	-	3000	Interfaz administrativa unificada
Microservicio A	Spring Boot 3.x	MariaDB	8080	Gestión de Clientes y Pedidos
Microservicio B	Spring Boot 3.x	PostgreSQL	8081	Gestión de Proveedores y Facturas
Common Library	Maven 3.9+	-	-	Lógica de negocio compartida

3. INSTALACIÓN Y CONFIGURACIÓN

3.1 Prerrequisitos del Sistema

[INSERTAR SCREENSHOT AQUÍ: prerrequisitos-terminal.png]

Software Requerido:

- Java JDK 17 o superior
- Apache Maven 3.9 o superior
- Node.js 18 o superior
- npm 9 o superior
- MariaDB 10.11 o superior
- PostgreSQL 15 o superior

Comandos de Verificación:

java -version

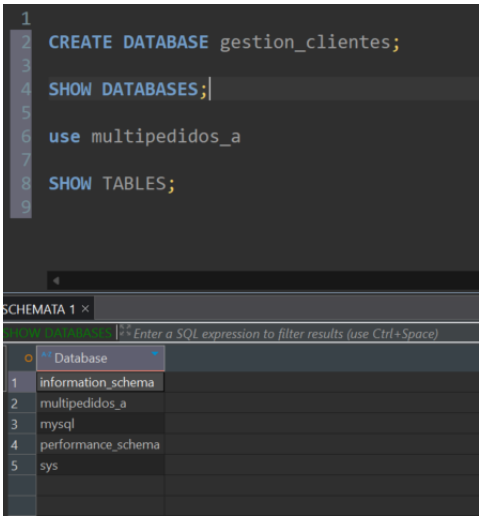
mvn -version

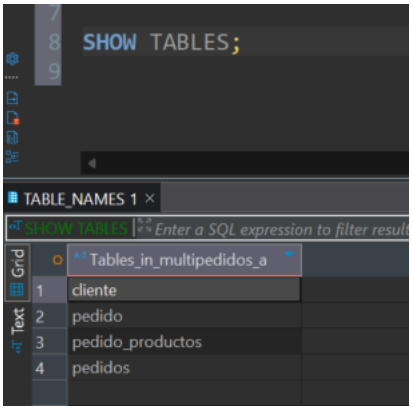
node --version

npm --version

3.2 Configuración de Bases de Datos

3.2.1 MariaDB (Microservicio A)





3.2.2 PostgreSQL (Microservicio B)

3.3 Instalación de Microservicios

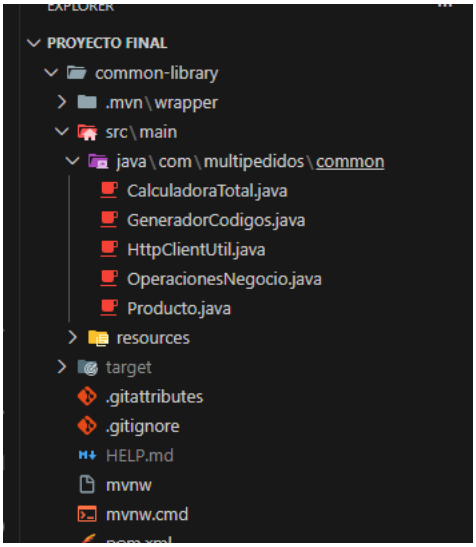
3.3.1 Common Library (Componente C)

cd common-library

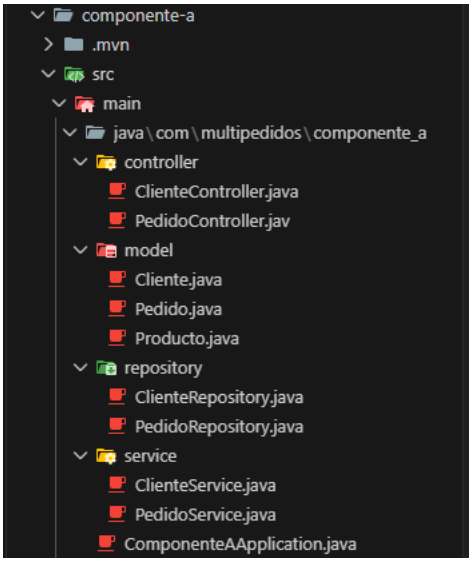
 # Compilar e instalar

mvn clean install

(se puede crear la carpeta con mkdir en la carpeta raíz)



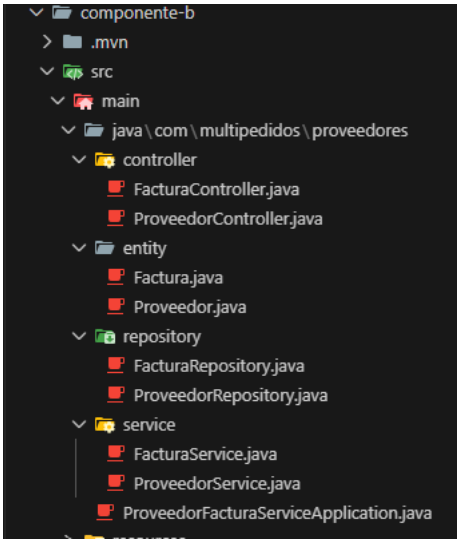
3.3.2 Microservicio A (Clientes y Pedidos)



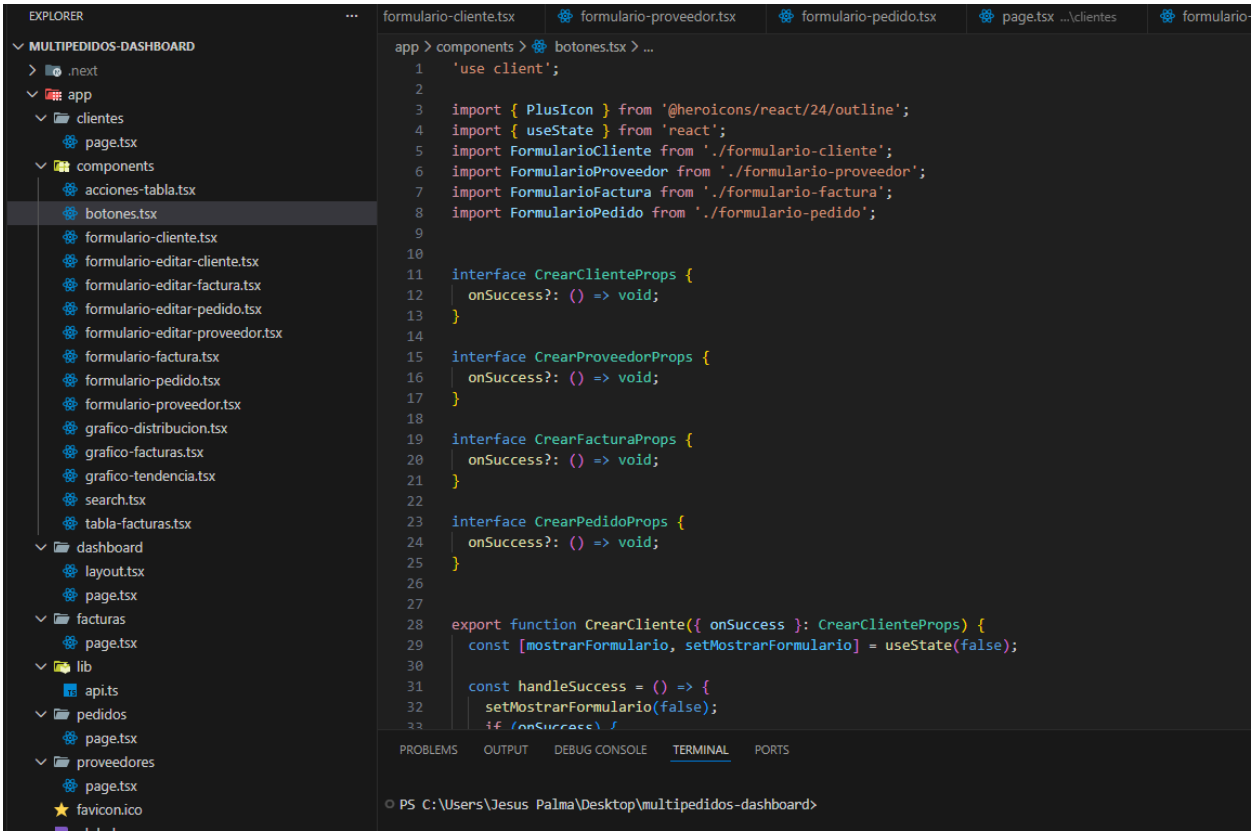
Ejecutar la aplicación

mvn spring-boot:run

3.3.3 Microservicio B (Proveedores y Facturas)



3.4 Instalación del Frontend



Instalar dependencias

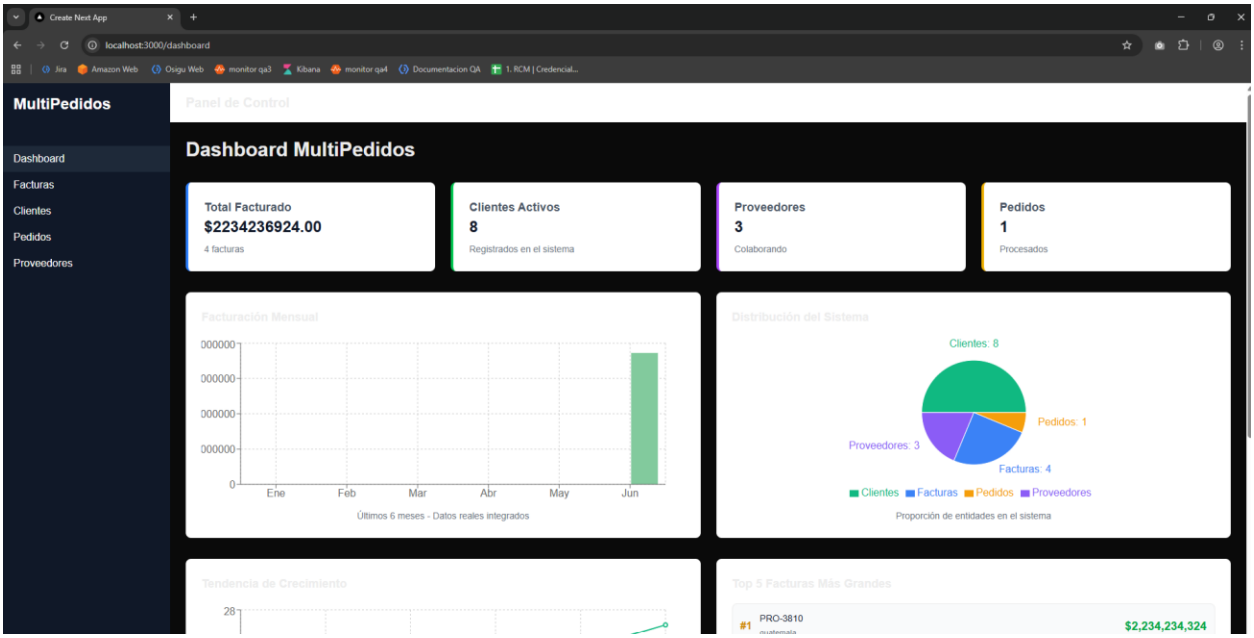
npm install

Ejecutar en modo desarrollo

npm run dev

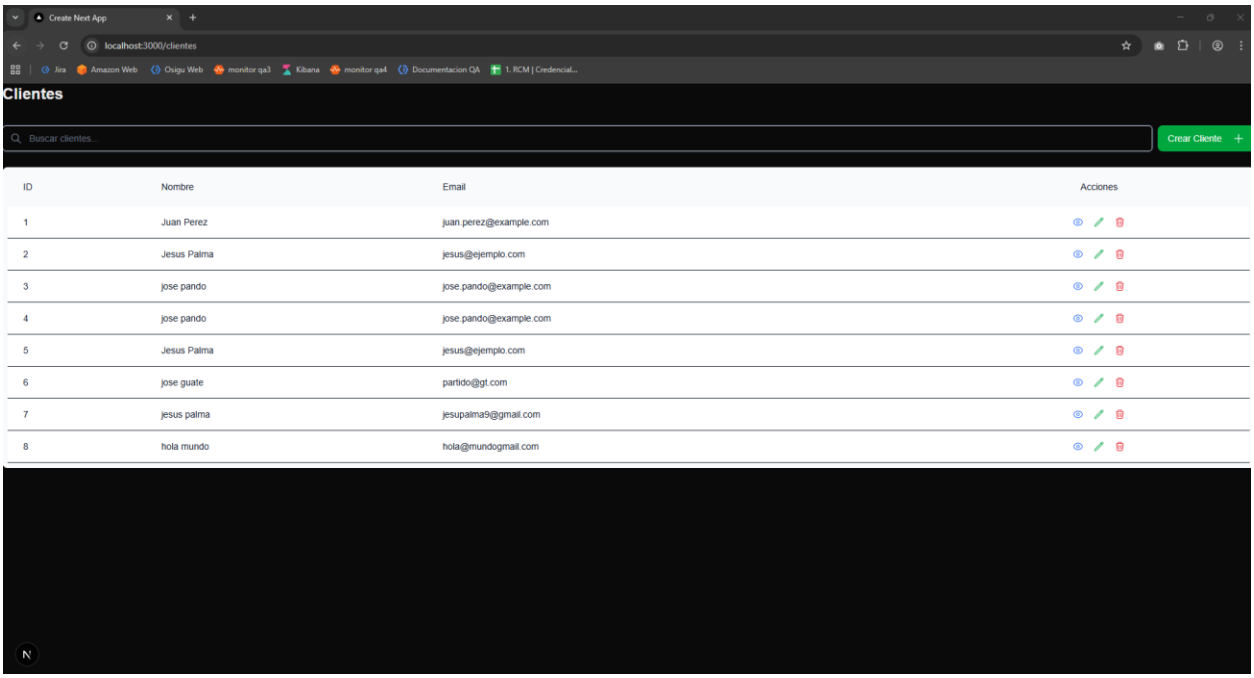
4. MANUAL DE USUARIO

4.1 Dashboard Principal



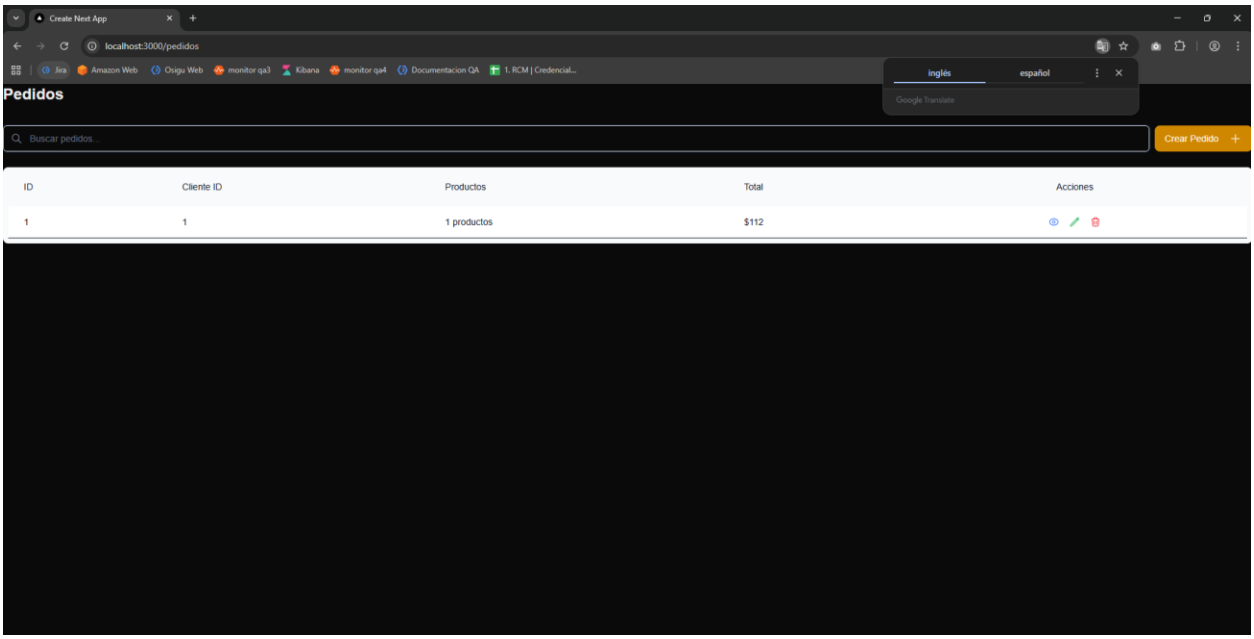
Descripción: Vista general del dashboard de Next.js mostrando métricas y accesos rápidos

4.2 Gestión de Clientes



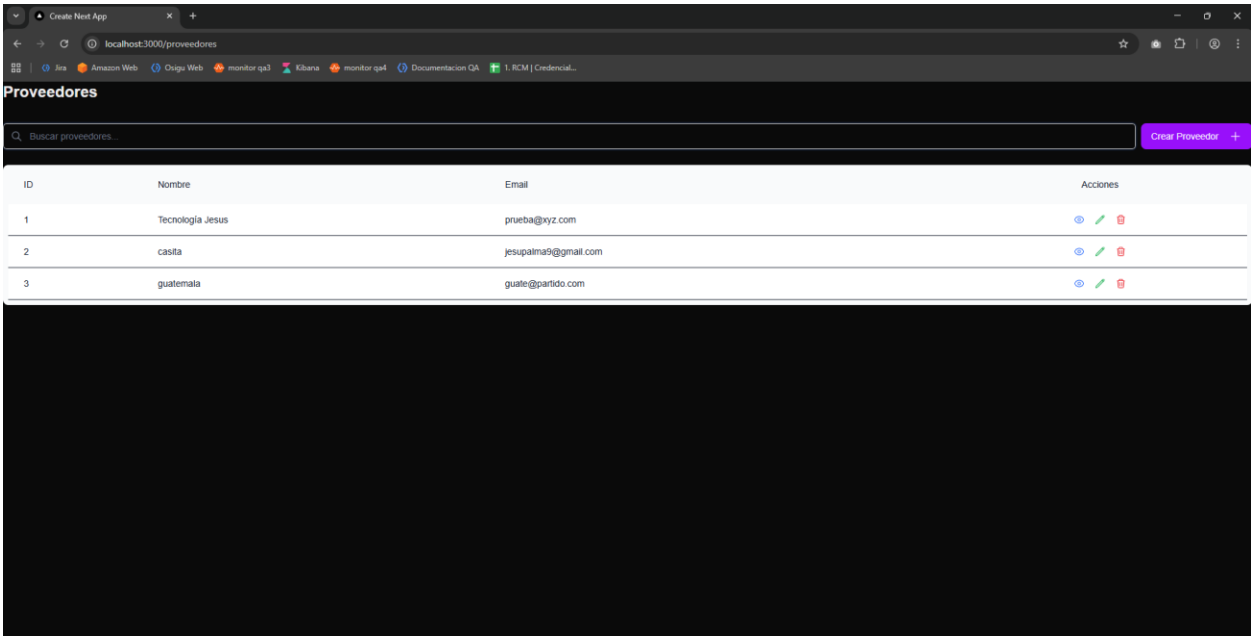
Descripción: Interfaz para crear, editar y listar clientes

4.3 Gestión de Pedidos



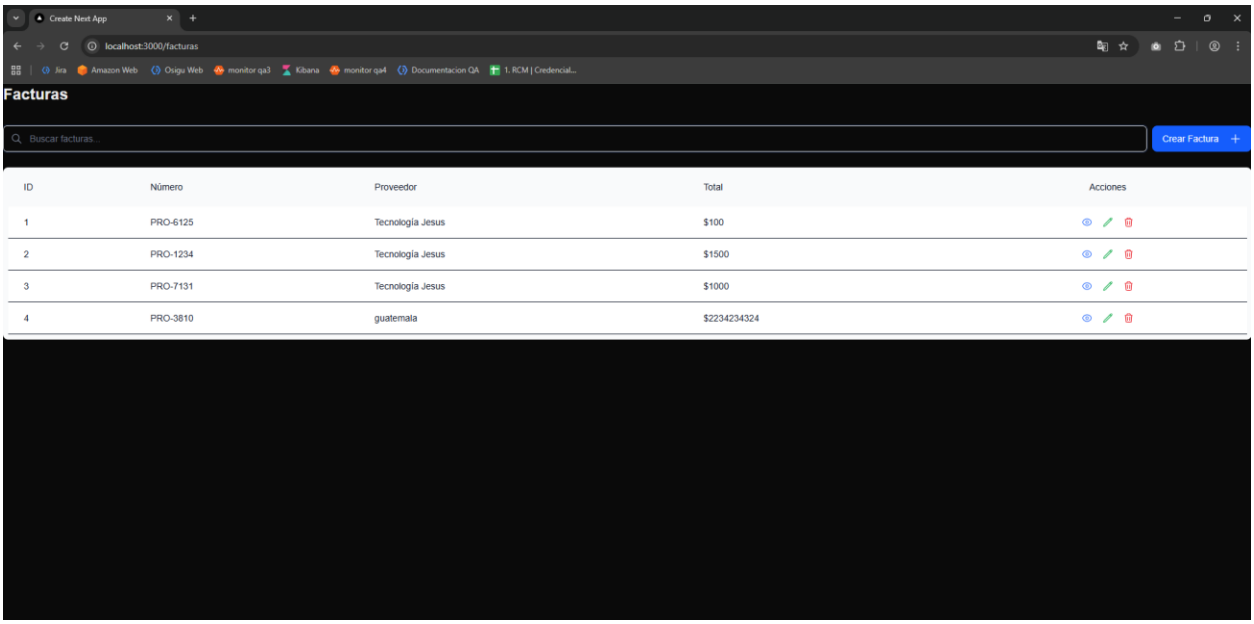
Descripción: Interfaz para crear y visualizar pedidos con cálculos automáticos

4.4 Gestión de Proveedores



Descripción: Interfaz para administrar proveedores

4.5 Gestión de Facturas

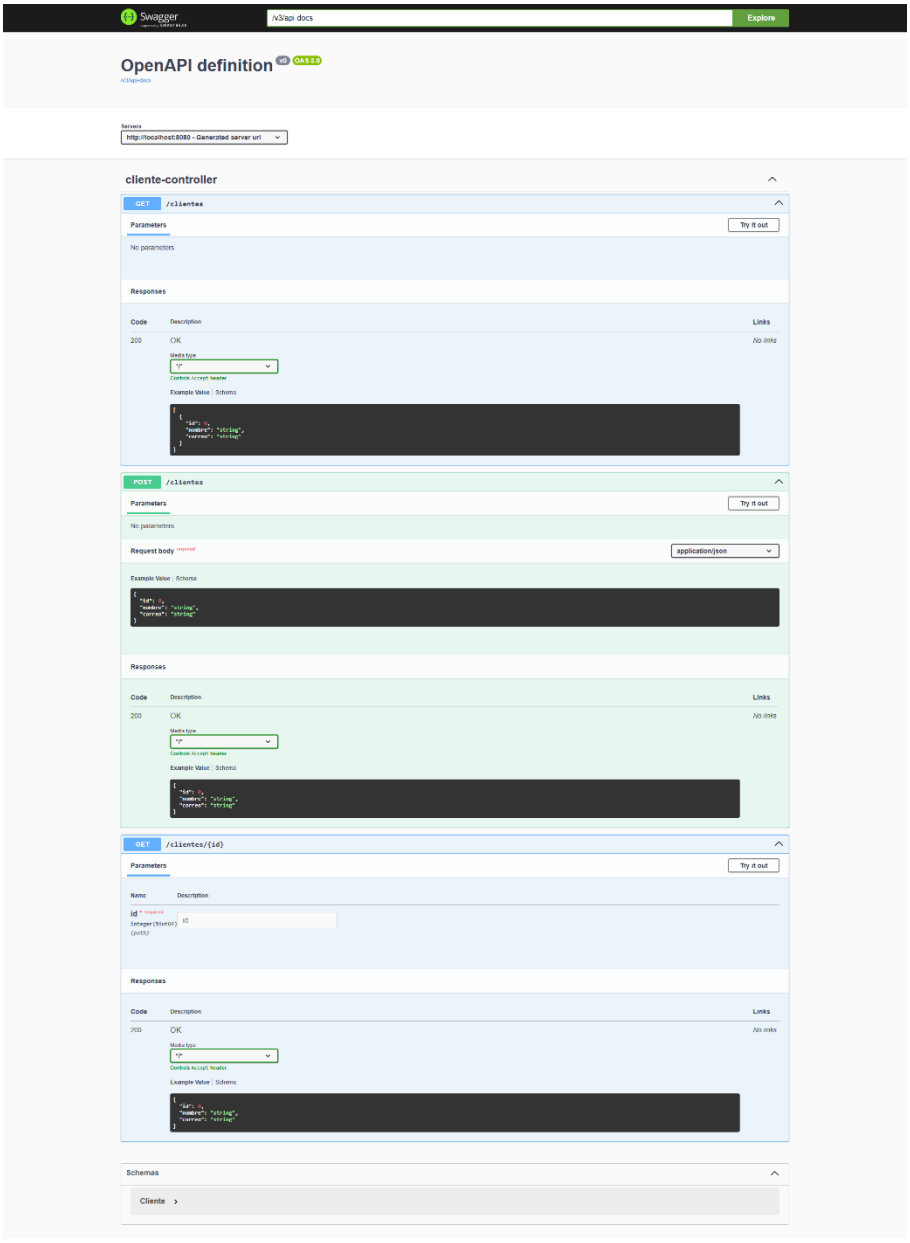


Descripción: Interfaz para generar y consultar facturas

5. API REFERENCE

5.1 Microservicio A - Clientes y Pedidos

URL Base: http://localhost:8080



Endpoints de Clientes:

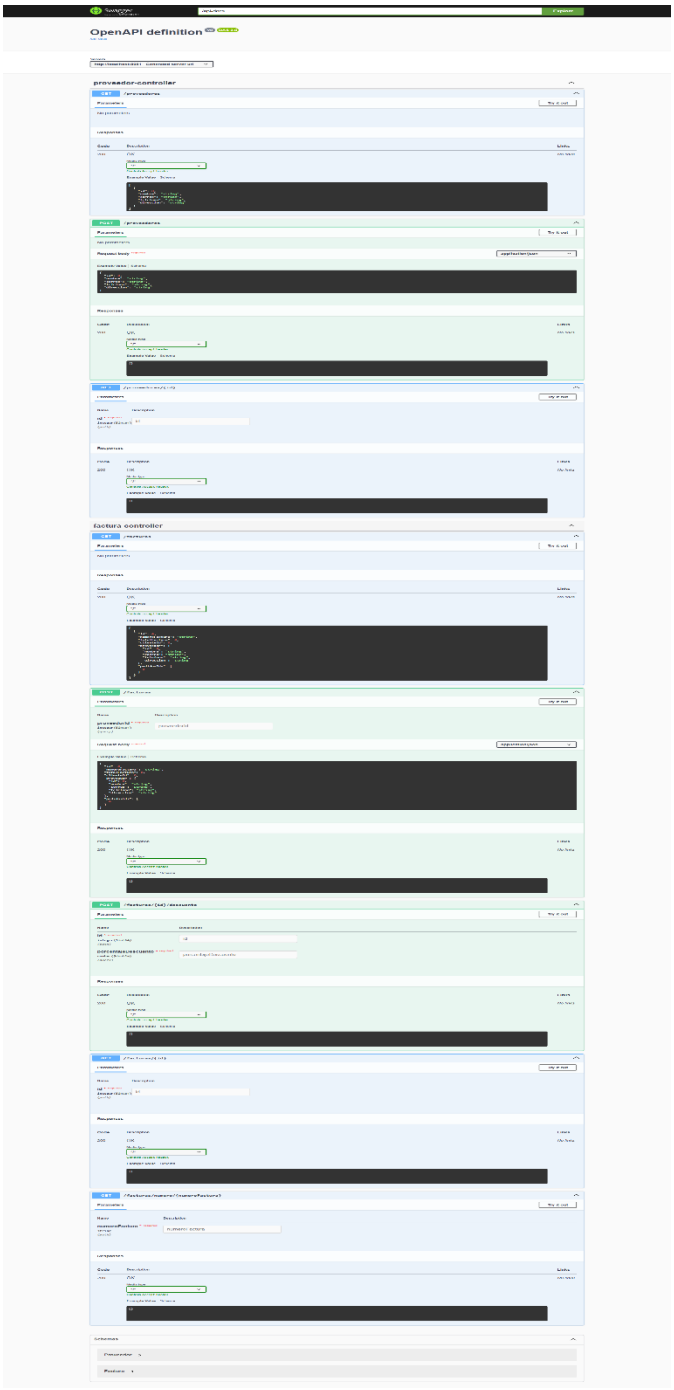
Método	Endpoint	Descripción	Body Request
POST	/clientes	Crear nuevo cliente	{ "nombre": "string", "correo": "string" }
GET	/clientes	Listar todos los clientes	-
GET	/clientes/{id}	Obtener cliente por ID	-

Endpoints de Pedidos:

Método	Endpoint	Descripción	Body Request
POST	/pedidos	Crear nuevo pedido	{ "clienteId": "number", "productos": [...] }
GET	/pedidos	Listar todos los pedidos	-
GET	/pedidos/{id}	Obtener pedido por ID	-

5.2 Microservicio B - Proveedores y Facturas

URL Base: http://localhost:8081



Endpoints de Proveedores:

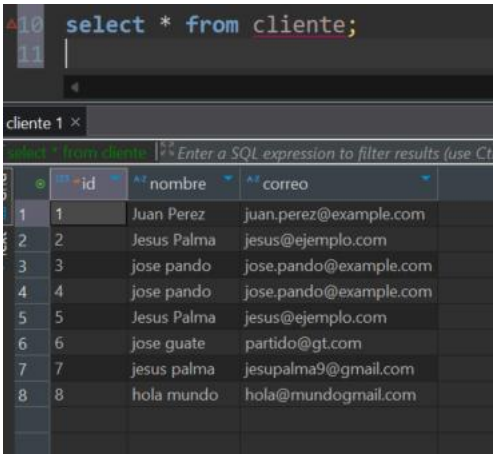
Método	Endpoint	Descripción	Body Request
POST	/proveedores	Registrar proveedor	{ "nombre": "string", "correo": "string" }
GET	/proveedores	Listar proveedores	-

Endpoints de Facturas:

Método	Endpoint	Descripción	Body Request
POST	/facturas	Registrar factura	{ "proveedorId": "number", "pedidos": [...] }
GET	/facturas	Listar facturas	-
GET	/facturas/{id}	Obtener factura por ID	-

6. ESTRUCTURA DE BASES DE DATOS

6.1 MariaDB - Esquema de Clientes y Pedidos



CONCLUSIÓN

Esta plataforma representa una solución robusta y escalable para la gestión de pedidos multi-plataforma, implementando mejores prácticas de arquitectura de microservicios y desarrollo full-stack.