

PLATAFORMA DE CONTROL DE PEDIDOS MULTIPLATAFORMA

INTRODUCCIÓN Y CONTEXTO

1.1 Descripción del Proyecto

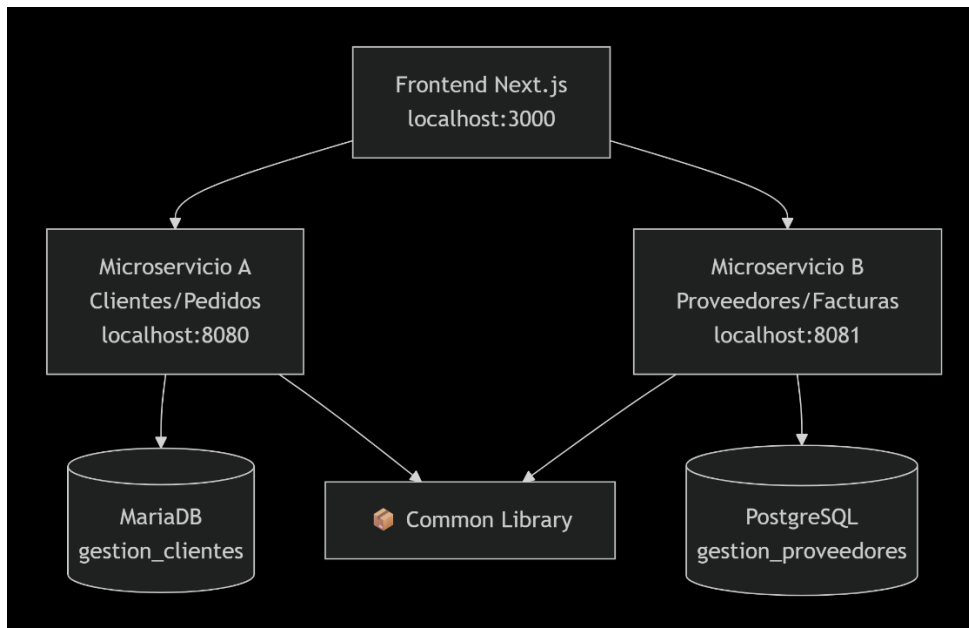
La empresa **MultiPedidos S.A.** requiere una solución de arquitectura distribuida para gestionar pedidos de distintos clientes y proveedores en varias regiones, operando con múltiples bases de datos por políticas internas y requerimientos técnicos.

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

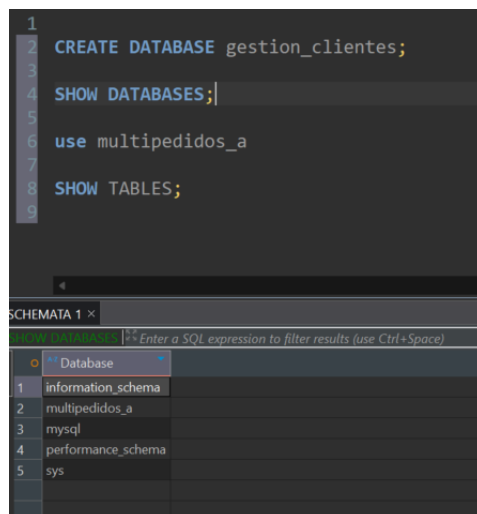
node --version

npm --version

3.2 Configuración de Bases de Datos

3.2.1 MariaDB (Microservicio A)

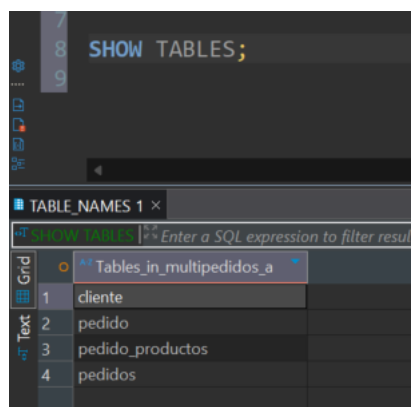
```
1
2 CREATE DATABASE gestion_clientes;
3
4 SHOW DATABASES;
5
6 use multipedidos_a
7
8 SHOW TABLES;
9
```



The screenshot shows a MySQL/MariaDB command-line interface. The user has entered the following commands: `CREATE DATABASE gestion_clientes;`, `SHOW DATABASES;`, `use multipedidos_a`, and `SHOW TABLES;`. The output of `SHOW DATABASES;` is displayed in a table below the command prompt.

Database
information_schema
multipedidos_a
mysql
performance_schema
sys

```
8 SHOW TABLES;
9
```



The screenshot shows the MySQL/MariaDB command-line interface with the command `SHOW TABLES;` entered. The output is displayed in a table below the command prompt.

Tables_in_multipedidos_a
cliente
pedido
pedido_productos
pedidos

3.2.2 PostgreSQL (Microservicio B)

3.3 Instalación de Microservicios

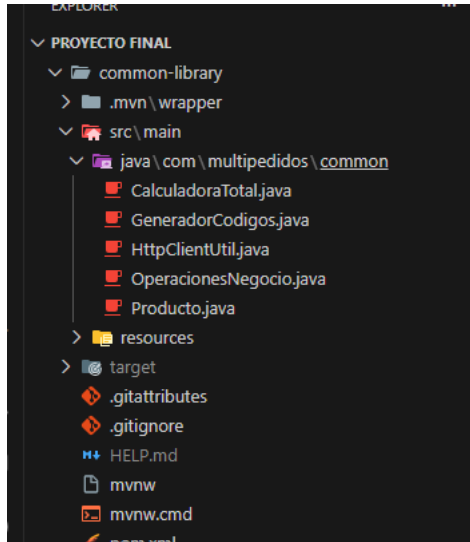
3.3.1 Common Library (Componente C)

cd common-library

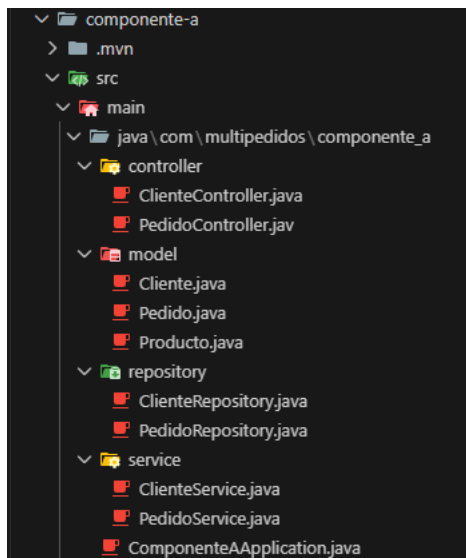
Compilar e instalar

mvn clean install

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



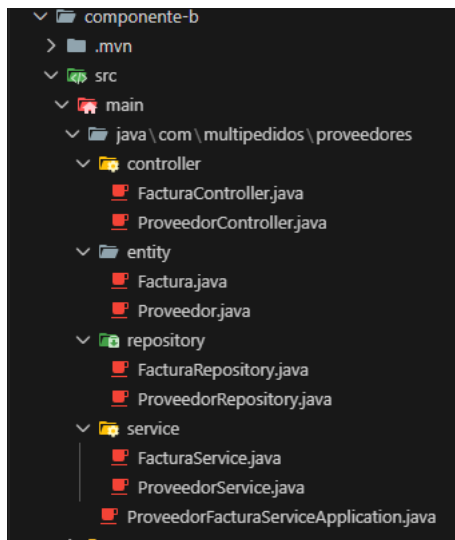
3.3.2 Microservicio A (Clientes y Pedidos)



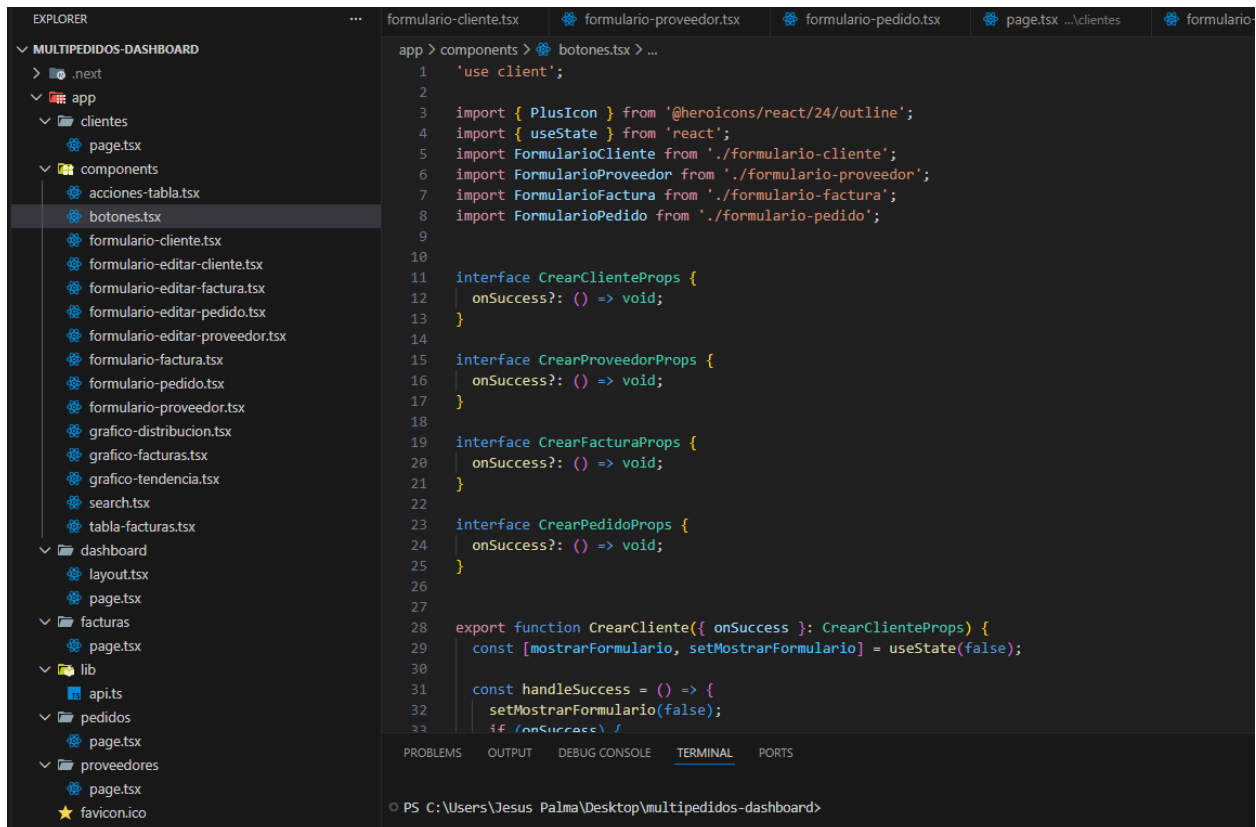
Ejecutar la aplicación

mvn spring-boot:run

3.3.3 Microservicio B (Proveedores y Facturas)



3.4 Instalación del Frontend



Instalar dependencias

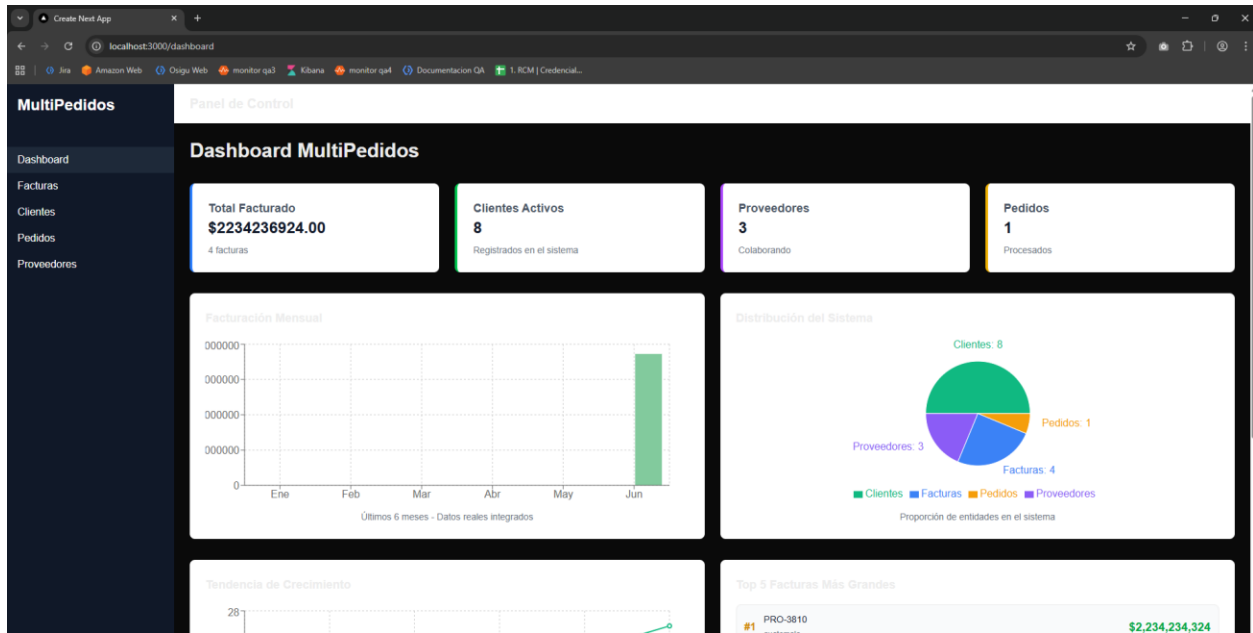
npm install

Ejecutar en modo desarrollo

npm run dev

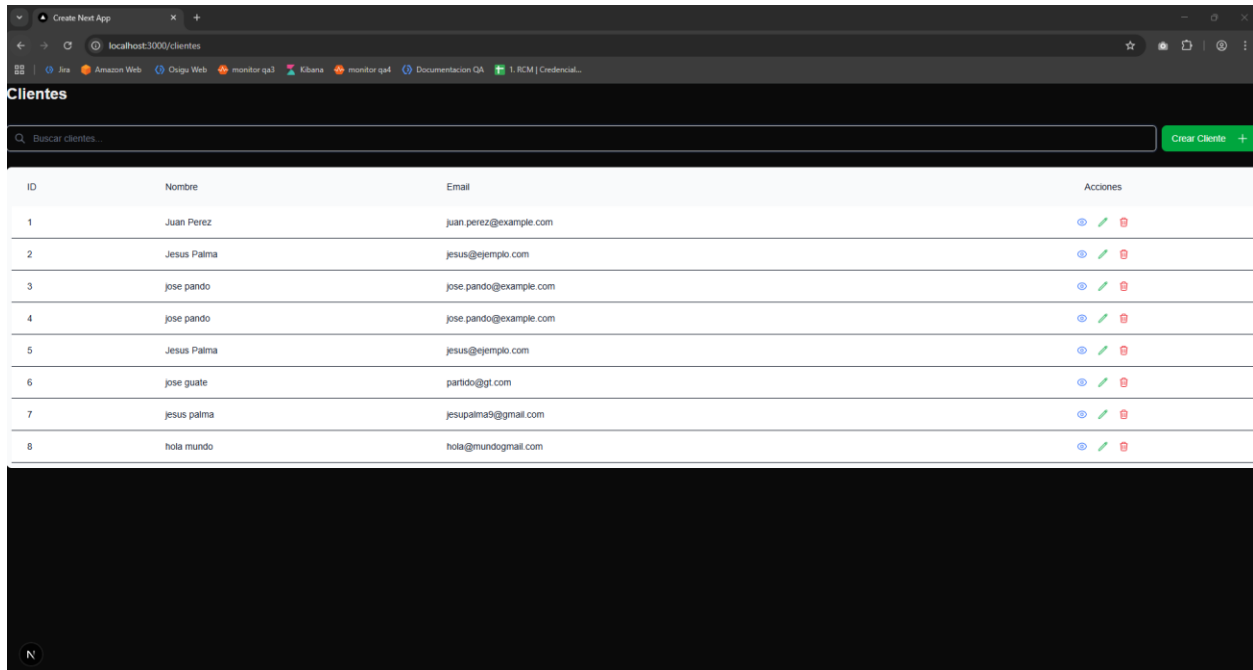
4. MANUAL DE USUARIO

























4.1 Dashboard Principal



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

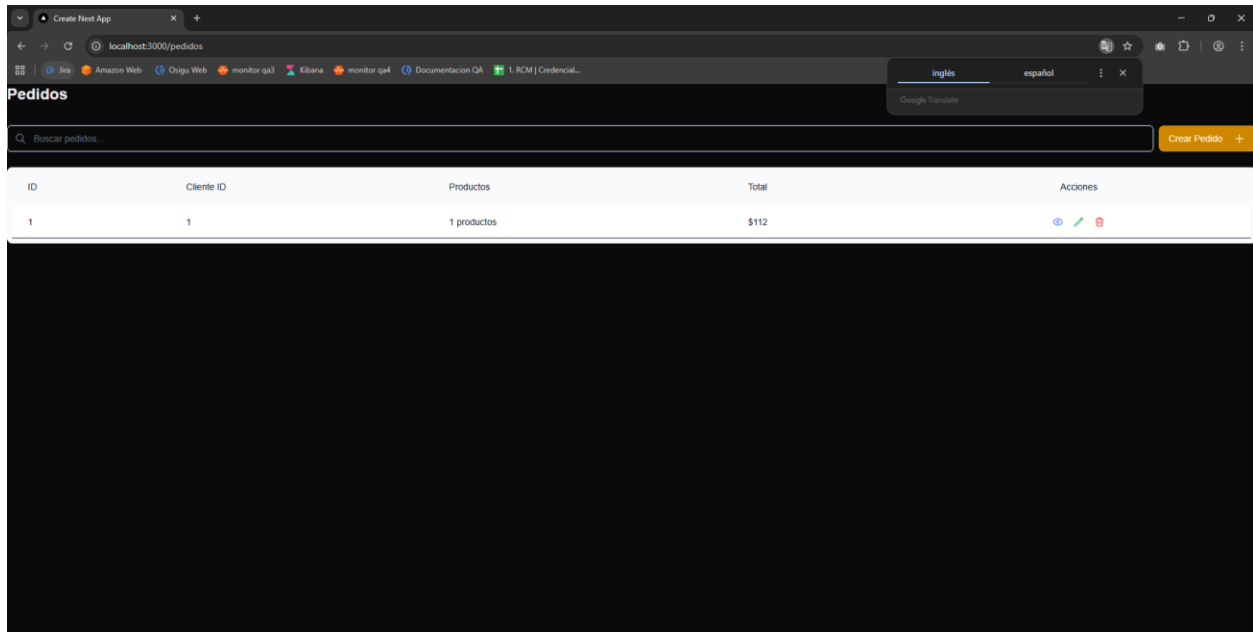
4.2 Gestión de Clientes



ID	Nombre	Email	Acciones
1	Juan Perez	juan.perez@example.com	  
2	Jesus Palma	jesus@ejemplo.com	  
3	jose pando	jose.pando@example.com	  
4	jose pando	jose.pando@example.com	  
5	Jesus Palma	jesus@ejemplo.com	  
6	jose guate	partido@gt.com	  
7	jesus palma	jesupalma9@gmail.com	  
8	hola mundo	hola@mundogmail.com	  

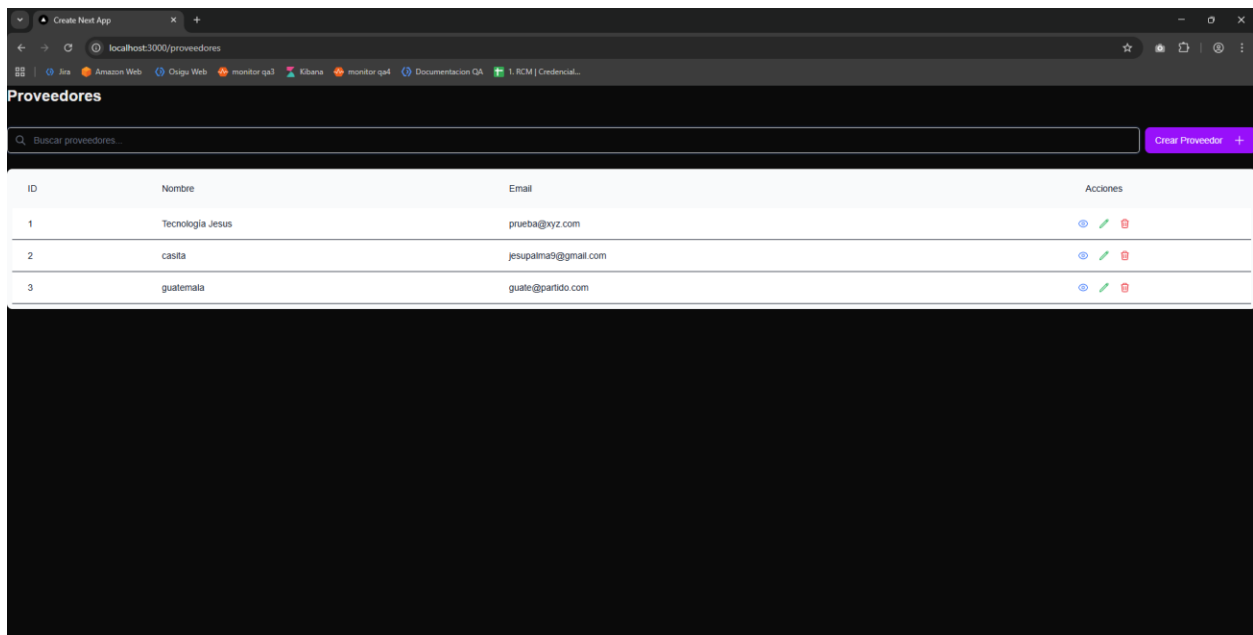
Descripción: Interfaz para crear, editar y listar clientes

4.3 Gestión de Pedidos



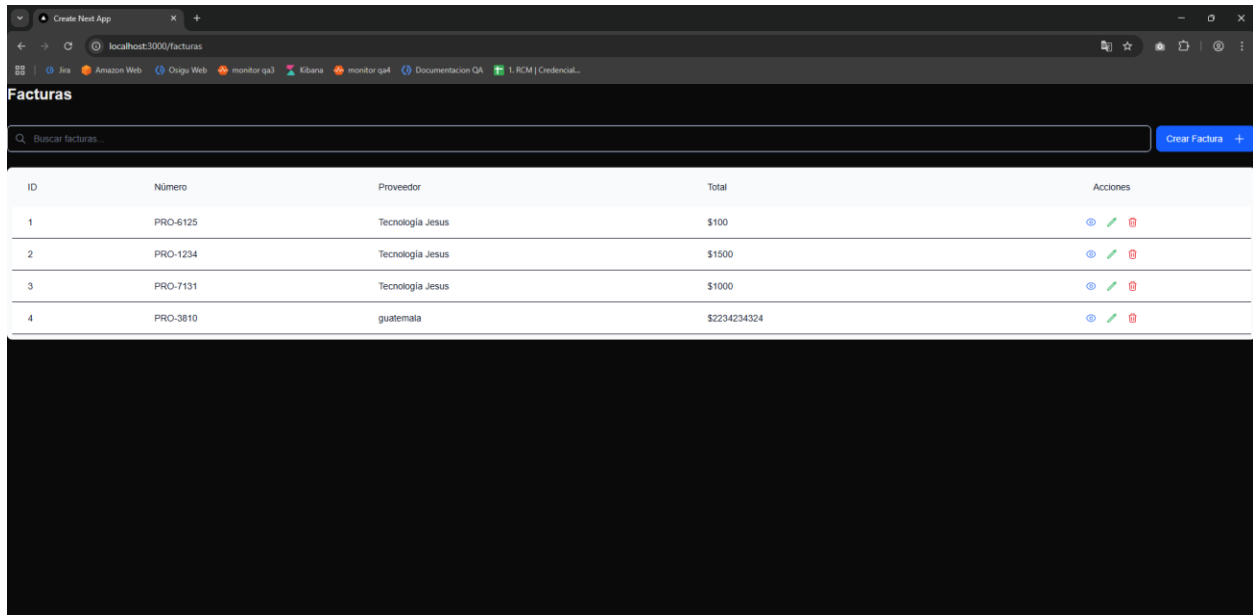
Descripción: Interfaz para crear y visualizar pedidos con cálculos automáticos













4.4 Gestión de Proveedores



Descripción: Interfaz para administrar proveedores

4.5 Gestión de Facturas




ID	Número	Proveedor	Total	Acciones
1	PRO-6125	Tecnología Jesus	\$100	  
2	PRO-1234	Tecnología Jesus	\$1500	  
3	PRO-7131	Tecnología Jesus	\$1000	  
4	PRO-3810	guatemala	\$2234234324	  

Descripción: Interfaz para generar y consultar facturas


5. API REFERENCE

5.1 Microservicio A - Clientes y Pedidos

URL Base: <http://localhost:8080>

 Swagger
API docs

Explore

OpenAPI definition  0.45.0

Services

http://localhost:8080 - Generated server url

cliente-controller

GET /clientes

Parameters

No parameters


Responses

Code	Description	Links
200	OK Media type: <input type="text" value="application/json"/> Click on left header Example Value Schema <pre>{ "id": 1, "nombre": "Juan", "apellido": "Perez" }</pre>	API links

POST /clientes

Parameters

No parameters

Request body 

application/json

Example Value | Schema


```
{
  "id": 1,
  "nombre": "Juan",
  "apellido": "Perez"
}
```

Responses

Code	Description	Links
200	OK Media type: <input type="text" value="application/json"/> Click on left header Example Value Schema <pre>{ "id": 1, "nombre": "Juan", "apellido": "Perez" }</pre>	API links

GET /clientes/{id}

Parameters

Name	Description
id 	ID Integer (int32) Path

Responses

Code	Description	Links
200	OK Media type: <input type="text" value="application/json"/> Click on left header Example Value Schema <pre>{ "id": 1, "nombre": "Juan", "apellido": "Perez" }</pre>	API links

Schemas

Cliente

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>

The screenshot displays the Swagger UI for an OpenAPI definition. The interface is organized into sections for different API controllers. The 'proveedor-controller' section is expanded, showing its endpoints. The 'factura-controller' section is also visible below it. Each endpoint is detailed with its HTTP method, path, parameters, and response schema. The UI includes a search bar at the top and a sidebar on the left for navigation. The 'proveedor-controller' endpoints include 'getProveedores', 'getProveedorById', 'createProveedor', 'updateProveedor', and 'deleteProveedor'. The 'factura-controller' endpoints include 'getFacturas', 'getFacturaById', 'createFactura', 'updateFactura', and 'deleteFactura'. The response schemas are defined in the 'Responses' tab for each endpoint.

OpenAPI definition

proveedor-controller

Endpoints:

- GET /proveedores
- GET /proveedores/{id}
- POST /proveedores
- PUT /proveedores/{id}
- DELETE /proveedores/{id}

factura-controller

Endpoints:

- GET /facturas
- GET /facturas/{id}
- POST /facturas
- PUT /facturas/{id}
- DELETE /facturas/{id}

Endpoints de Proveedores:

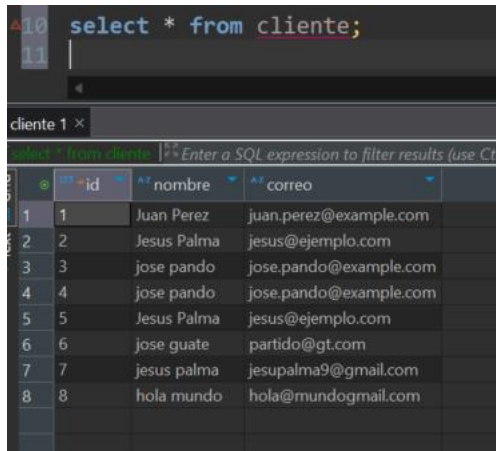
Método	Endpoint	Descripción	Body Request
POST	/proveedores	Registrar proveedor	{ "nombre": "string", "correo": "string" }
GET	/proveedores	Listar proveedores	-

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



The screenshot shows a SQL client interface with a query window and a results window. The query window contains the SQL statement `select * from cliente;`. The results window displays a table with 8 rows and 3 columns: `id`, `nombre`, and `correo`. The data is as follows:

	id	nombre	correo
1	1	Juan Perez	juan.perez@example.com
2	2	Jesus Palma	jesus@ejemplo.com
3	3	jose pando	jose.pando@example.com
4	4	jose pando	jose.pando@example.com
5	5	Jesus Palma	jesus@ejemplo.com
6	6	jose guate	partido@gt.com
7	7	jesus palma	jesupalma9@gmail.com
8	8	hola mundo	hola@mundogmail.com

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.