Gestion de Catering

**Nome Alumno/a:**

***Rodrigo Gajardo Blanco***

**Curso: *2º DAM* Materia:** ***Proxecto Final Módulo Acceso a Datos***

**Docente:** ***Javier Feijóo López***

***🗐 Proxecto: https://github.com/RodrigoGB12/ProyectoApi***

Contido

[1. Introdución 2](#_Toc189548783)

[2. Deseño da Base de Datos – Modelo EER 2](#_Toc189548784)

[3. Tecnoloxías Empregadas e Configuración da Contorna 2](#_Toc189548785)

[4. Deseño do Backend – API REST con Spring Boot 2](#_Toc189548786)

[5. Implementación de Autenticación e Seguridade 3](#_Toc189548787)

[6. Desenvolvemento do Backend con Thymeleaf 3](#_Toc189548788)

[7. Documentación da API e Probas en Postman 3](#_Toc189548789)

[8. Conclusións e Valoración Persoal 3](#_Toc189548790)

[9. Anexos (Opcional) 3](#_Toc189548791)

# 1. Introdución

Breve presentación do proxecto, respondendo a preguntas clave:

La finalidad del sistema es gestionar un servicio de catering proporcionando informacion sobre todos los pedidos, cliente y trabajadores implicados.

Esta facilita la gestion de un restaurante y sus pedidos

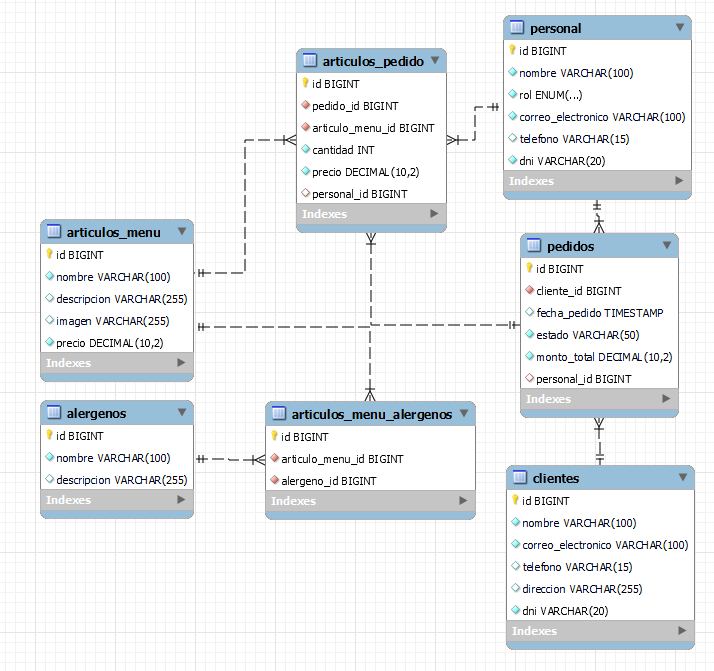
* Resumen das tecnoloxías empregadas.

Se uso Java,H2,Spring,Thymeleaf y Swagger

# 2. Deseño da Base de Datos – Modelo EER

Explicación do deseño da base de datos, incluíndo:

* Modelo Entidade-Relación Estendido (EER) cunha imaxe.



* Explicación de cada entidade e das súas relacións.

Personal: Contiene los miembros del personal, su informacion y rol.

Pedidos: Contiene el cliente del pedido y el repartidor

Clientes: Informacion de los clientes

Articulos\_pedido: Contiene los platos que tienen los pedidos y su cocinero

Articulos\_menu\_alergenos: Indica que alergenos tiene cada comida.

Alergenos: Informacion de alergenos

Articulos\_menu: Informacion y precio sobre los platos

Usuario: Tabla de usuarios.

**Claves Primarias (Primary Keys):**

Todas las claves primarias son BIGINT AUTO\_INCREMENT y se llaman “id”

**Tabla clientes:**

**Descripción:** La columna id es un número único que identifica de manera exclusiva a cada cliente.

**Tabla personal:**

**Descripción:** La columna id es un número único que identifica a cada miembro del personal (cocinero o conductor).

**Tabla alergenos:**

**Descripción:** La columna id es un número único que identifica a cada alérgeno.

**Tabla articulos\_menu:**

**Descripción:** La columna id es un número único que identifica cada artículo del menú.

**Tabla pedidos:**

**Descripción:** La columna id es un número único que identifica cada pedido realizado por un cliente

**Tabla articulos\_pedido:**

**Descripción:** La columna id es un número único que identifica cada registro dentro de la relación entre los artículos y los pedidos.

**Tabla articulos\_menu\_alergenos:**

**Descripción**: La columna id es un número único que identifica la relación entre un artículo del menú y un alérgeno.

**Tabla usuarios**:

**Descripción:** La columna id es un número único que identifica a cada usuario (admin o usuario común) en el sistema. Es un campo BIGINT AUTO\_INCREMENT.Restricións e regras de integridade.

Claves Foráneas (Foreign Keys)

**Tabla pedidos:**

**Clave foránea: cliente\_id**

**Descripción:** La columna cliente\_id en la tabla pedidos es una clave foránea que hace referencia a la columna id de la tabla clientes. Esta relación indica que cada pedido debe estar asociado con un cliente específico.

**Clave foránea: personal\_id**

**Descripción:** La columna personal\_id en la tabla pedidos es una clave foránea que hace referencia a la columna id de la tabla personal. Esto indica que cada pedido puede estar asociado con un miembro del personal (por ejemplo, un cocinero o conductor).

**Tabla articulos\_pedido:**

**Clave foránea: pedido\_id**

**Descripción:** La columna pedido\_id en la tabla articulos\_pedido es una clave foránea que hace referencia a la columna id de la tabla pedidos. Esta relación establece que cada artículo del pedido pertenece a un pedido específico.

**Clave foránea: articulo\_menu\_id**

**Descripción:** La columna articulo\_menu\_id en la tabla articulos\_pedido es una clave foránea que hace referencia a la columna id de la tabla articulos\_menu. Cada artículo en el pedido corresponde a un artículo específico del menú.

**Clave foránea: personal\_id**

**Descripción:** La columna personal\_id en la tabla articulos\_pedido es una clave foránea que hace referencia a la columna id de la tabla personal. Esto indica que un miembro del personal es responsable de un artículo específico en un pedido.

**Tabla articulos\_menu\_alergenos:**

**Clave foránea: articulo\_menu\_id**

**Descripción:** La columna articulo\_menu\_id en la tabla articulos\_menu\_alergenos es una clave foránea que hace referencia a la columna id de la tabla articulos\_menu. Esto establece una relación entre los artículos del menú y los alérgenos asociados con ellos.

**Clave foránea: alergeno\_id**

Descripción: La columna alergeno\_id en la tabla articulos\_menu\_alergenos es una clave foránea que hace referencia a la columna id de la tabla alergenos. Esto indica que cada artículo del menú puede tener uno o más alérgenos asociados.

# 3. Tecnoloxías Empregadas e Configuración da Contorna

Explicación das ferramentas empregadas:

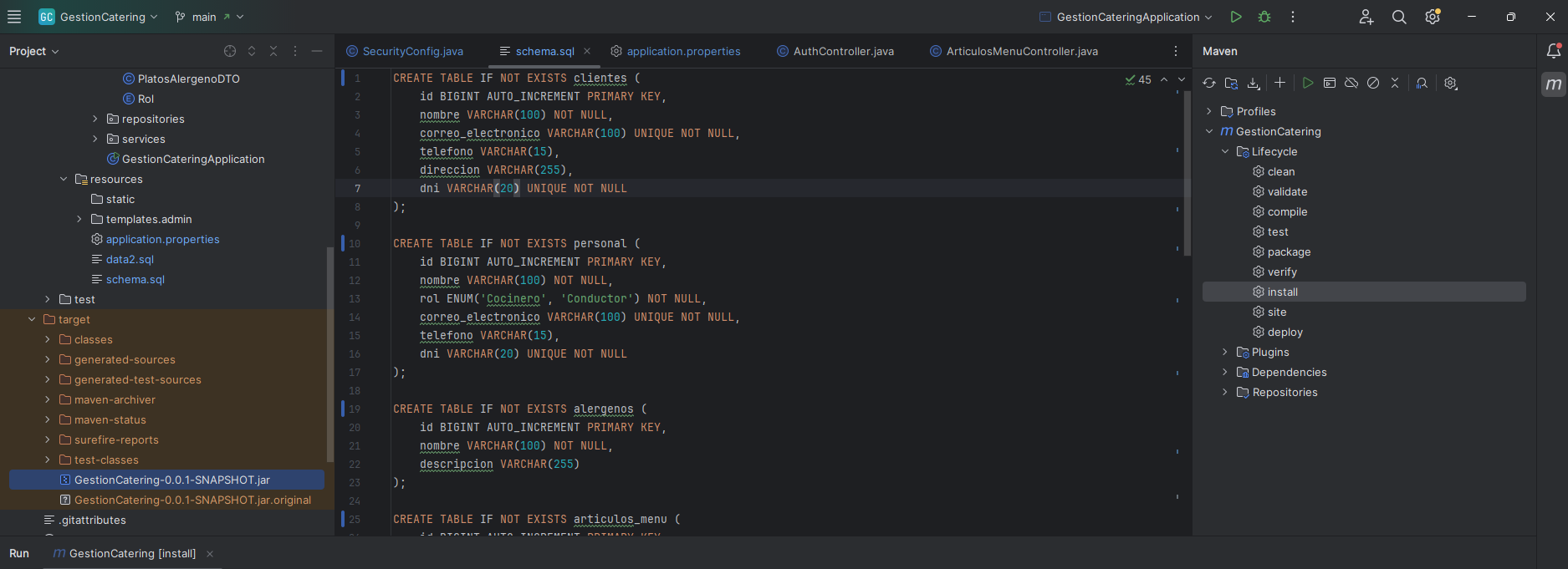
La aplicacion fue programada en Java 17, usando el framework Spring Boot 3, se uso H2 como base de datos para simplificar la aplicacion, las dependencias se usaron las de Maven, la autentificacion se hizo con Spring Security y JWT, y el frontend con Thymeleaf y Bootstrap.

La documentacion esta hecha con Swagger y Postman

De IDE recomiendo usar Intelij y para ejecutarlo haremos los siguientes pasos:

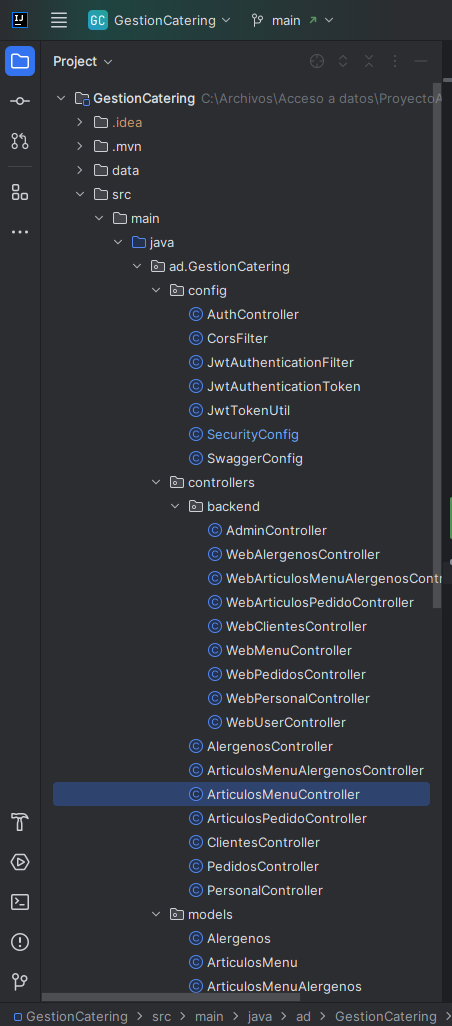
Para ejecutarlo lo primero tenemos 2 opciones, descargar el proyecto y ejecutarlo directamente, o crear el .jar para ello haremos lo siguiente:

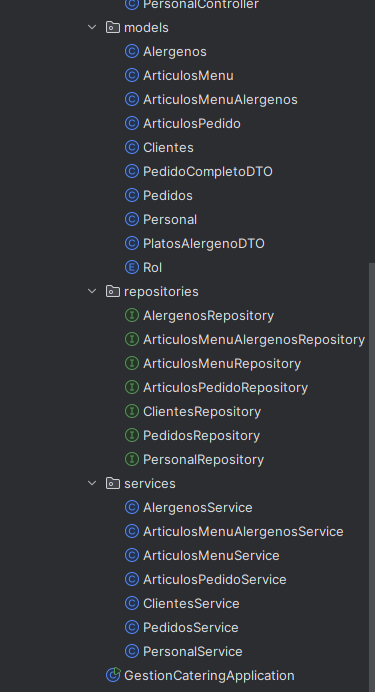
Abriremos el proyecto y a la derecha clickaremos la “M” y en el nombre del proyecto le daremos a install y nos llevaremos el .jar, despues tenemos otras dos opciones o nos llevamos la carpeta data o en caso de no llevarnosnola con el proyecto en la consola de h2 al ejecutarlo introducimos lo que hay en data2.sql

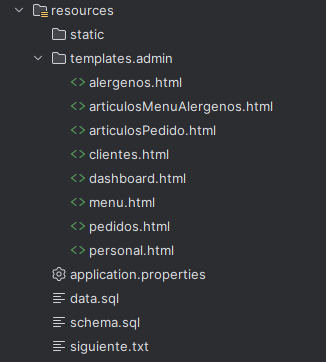


# 4. Deseño do Backend – API REST con Spring Boot

Explicación da arquitectura do backend:

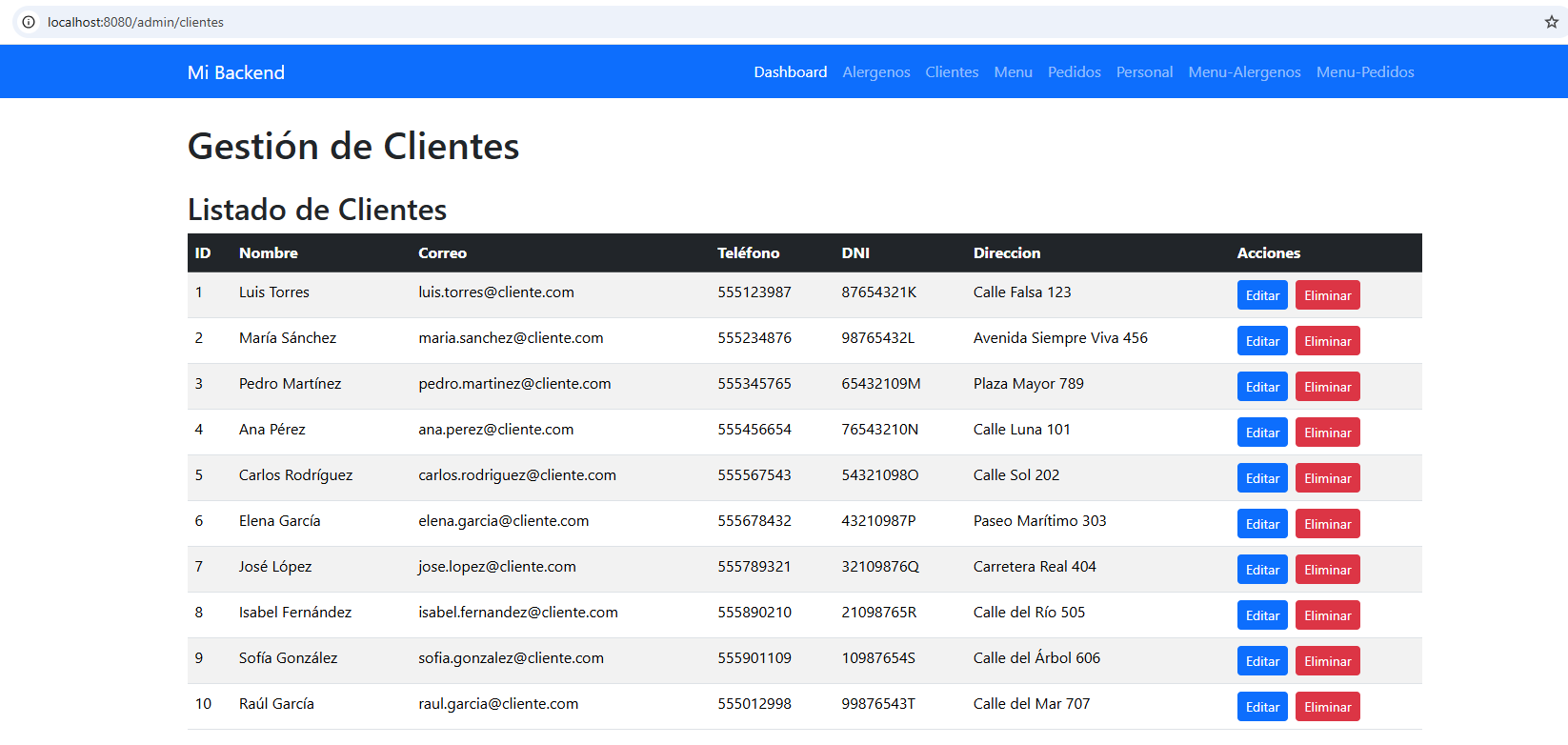


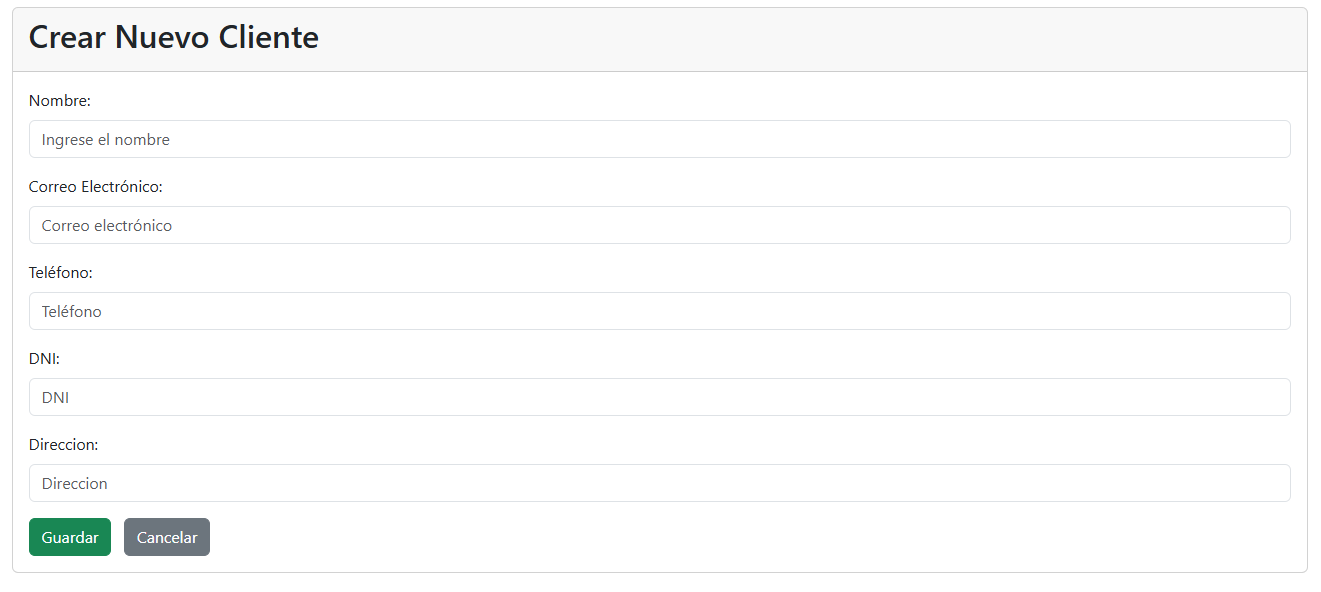




Esta es la estructura del codigo, que esta dividido en config(archivos de seguridad y configuracion),models(Los modelos con un par de DTOs),repositories,services y los controladores que tienen su “version web” y su version normal.

Se usara la clase cliente para explicar la implementacion del crud:





Aqui podemos crear y cambiar todos los campos, menos el id que lo gestionara el propio programa,pordremos crear, editar, consultar y eliminar.

# 5. Implementación de Autenticación e Seguridade

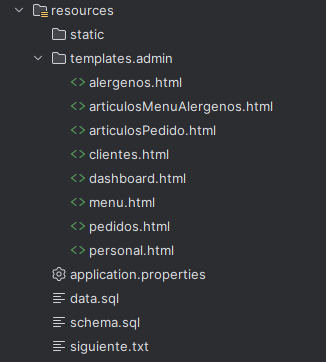
Explicación da seguridade no sistema:

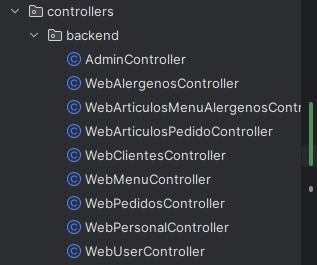
Los endpoints protegidos son los normales(no los web) y e dejado sin proteccion el /menu, esta hecho usando tokens y hay un solo rol que seria el de admin son usuario/contraseña: usuario/123456

# 6. Desenvolvemento do Backend con Thymeleaf

Explicación do panel de administración con Thymeleaf:

* Vistas e plantillas HTML.
* Formularios de inicio de sesión e xestión de datos.
* Controladores MVC e servizos para a web.

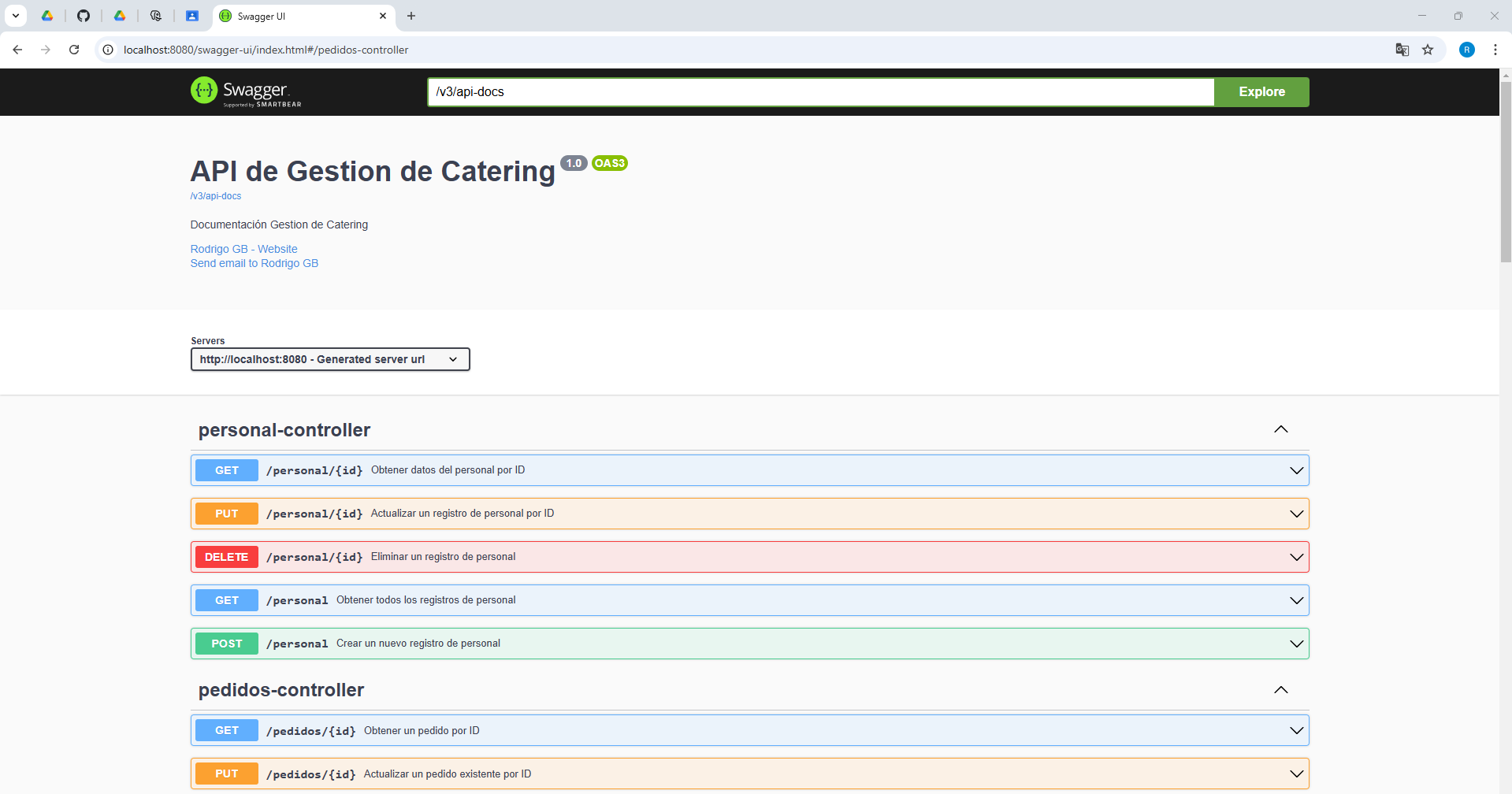
Estas son las plantillas y los controladores web:  




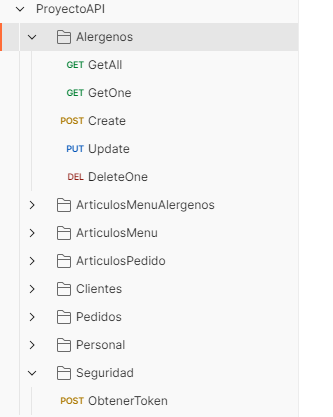
A continuacion ponemos unas capturas de ejemplo del backend:

# 7. Documentación da API e Probas en Postman

Incluir exemplos de endpoints documentados con Swagger e Postman:



Esta es una captura del Swagger y abajo vemos los endpoints del postman:



# 8. Conclusións e Valoración Persoal

Reflexión sobre o desenvolvemento do proxecto:

El mayor problema para el desarollo de esta aplicacion fue probablemente la seguridad, para implementar la seguridad necesite bastante tiempo, dando esta bastantes problemas.

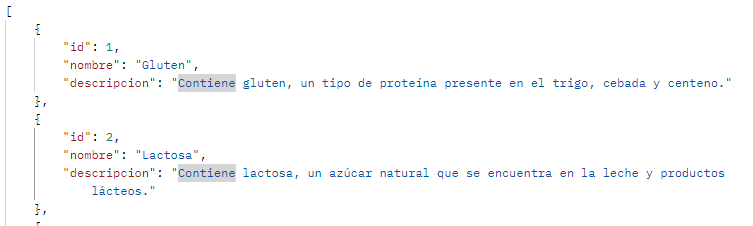
En versiones superiores se podria darle un vistazo al tema de la seguridad.

Spring boot me parecio un framework bastante util, que simplifica bastante el desarollo de una api.

# 9. Anexos (Opcional)

Se o proxecto inclúe documentación adicional, achegar:

* Estrutura JSON de respostas da API.

La estructura del JSON seran los atributos de la clase como se ve a continuacion:

El codigo sql esta en el programa, en resources es el schema.sql

Para la instalacion en un servidor se hara igual que arriba descargando el archivo y colocando la carpeta data en el mismo nivel y ejecutandolo.