

# **Proyecto de Desarrollo de Aplicaciones Multiplataforma**



Autor: Aarón Cutillas Ruz

Tutor: Dimas Asencio

IES Macià Abela (Crevillent)

Curso 2023-2024

## **Abstract**

This project, developed as part of the Higher Technician in Multiplatform Application Development (DAM) program, focuses on the creation of a mobile application designed to manage orders for a single restaurant. The app allows users to add products to their carts and place orders, while restaurant employees can accept or reject these orders. Additionally, administrators have the same capabilities as employees, but they can also add, delete, and update products. The system is centralized through an Android application developed in Java, with different activities adapted to the three roles defined in the database: user, worker and administrator. The backend is built using Spring Boot and Java, ensuring secure and efficient data management.

The application aims to manage orders for a single restaurant. This project meets the requirements of the specified modules. This initial version includes basic functionality for order management and is designed to be easily extensible to accommodate future needs and improvements.

# Índice

1. Abstract
2. Introducción
3. Funcionalidad
  1. Aplicación para dispositivos Android
  2. Operaciones
    1. Listado de productos
    2. Listado de Pedidos
    3. Modificación de los productos
    4. Borrado de productos
    5. Creación de productos
    6. Modificación de pedidos
    7. Creación de usuario trabajador
4. API REST
5. Arquitectura
  1. Componentes
  2. Aplicación para dispositivos Android
  3. API REST
6. Satisfacción de requisitos
  1. Acceso a datos
  2. Desarrollo de interfaces
  3. Inglés técnico
  4. Programación multimedia y dispositivos móviles
7. Conclusiones y trabajo futuro
8. Fuentes consultadas

# Introducción

En el contexto de la gestión de un restaurante, es crucial contar con un sistema eficiente para manejar los pedidos de los clientes. El presente proyecto del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma (DAM) tiene como objetivo el desarrollo de una aplicación móvil destinada a la gestión de pedidos de un restaurante. Esta aplicación permite a los usuarios añadir productos a sus carritos y realizar pedidos, mientras que los trabajadores y administradores pueden aceptar o rechazar dichos pedidos. Además, los administradores tienen la capacidad de añadir, eliminar y actualizar los productos disponibles en el restaurante.

La aplicación está diseñada para dispositivos Android y está desarrollada en Java. La base de datos que respalda el sistema está implementada con Spring Boot y Java, garantizando así un manejo eficiente y seguro de la información. La aplicación contempla diferentes roles de usuario: cliente, trabajador y administrador, cada uno con actividades específicas y permisos adecuados a sus funciones.

El sistema permite que los usuarios realicen pedidos de manera rápida y sencilla, mejorando así la experiencia del cliente. Por su parte, los trabajadores pueden gestionar los pedidos de forma organizada, mientras que los administradores tienen un control completo sobre el catálogo de productos del restaurante. Esta estructura garantiza una operación fluida y coordinada dentro del restaurante.

A través del desarrollo y puesta en marcha de esta aplicación, se satisfacen los requisitos definidos en varios módulos del segundo curso de DAM. El sistema desarrollado no solo proporciona la funcionalidad esencial para la gestión de pedidos, sino que también es escalable y puede adaptarse para incorporar futuras necesidades y mejoras.

# Funcionalidad

## Aplicación para dispositivos Android

### Operaciones

En función de su perfil, los usuarios de la aplicación para dispositivos Android pueden realizar algunas de las siguientes operaciones:

#### Listado de productos

Los usuarios pueden visualizar todos los productos ofrecidos por el restaurante. Cada producto se muestra con su id, nombre, precio.

#### Listado de pedidos

Los trabajadores y administradores pueden acceder a una lista de todos los pedidos realizados por los clientes. Esta lista incluye detalles como el estado del pedido (pendiente, aceptado, rechazado), los productos solicitados y la información del cliente.

#### Modificación de productos

Los administradores tienen la capacidad de modificar los detalles de los productos existentes. Esto incluye actualizar el nombre, la descripción, el precio de los productos.

#### Borrado de productos

Los administradores pueden eliminar productos del catálogo del restaurante. Esta operación asegura que el menú del restaurante esté siempre actualizado y libre de productos obsoletos.

#### Creación de productos

Los administradores pueden añadir nuevos productos al menú del restaurante. Al crear un nuevo producto, se deben proporcionar detalles completos, como el nombre, la descripción, el precio.

#### Modificación de pedidos

Los trabajadores y administradores pueden cambiar el estado de los pedidos, aceptándolos o rechazándolos según corresponda.

#### Creación de usuario trabajador

Los administradores tienen la capacidad de crear nuevos usuarios con el rol de trabajador. Esto facilita la incorporación de nuevo personal y la asignación de permisos adecuados dentro del sistema.

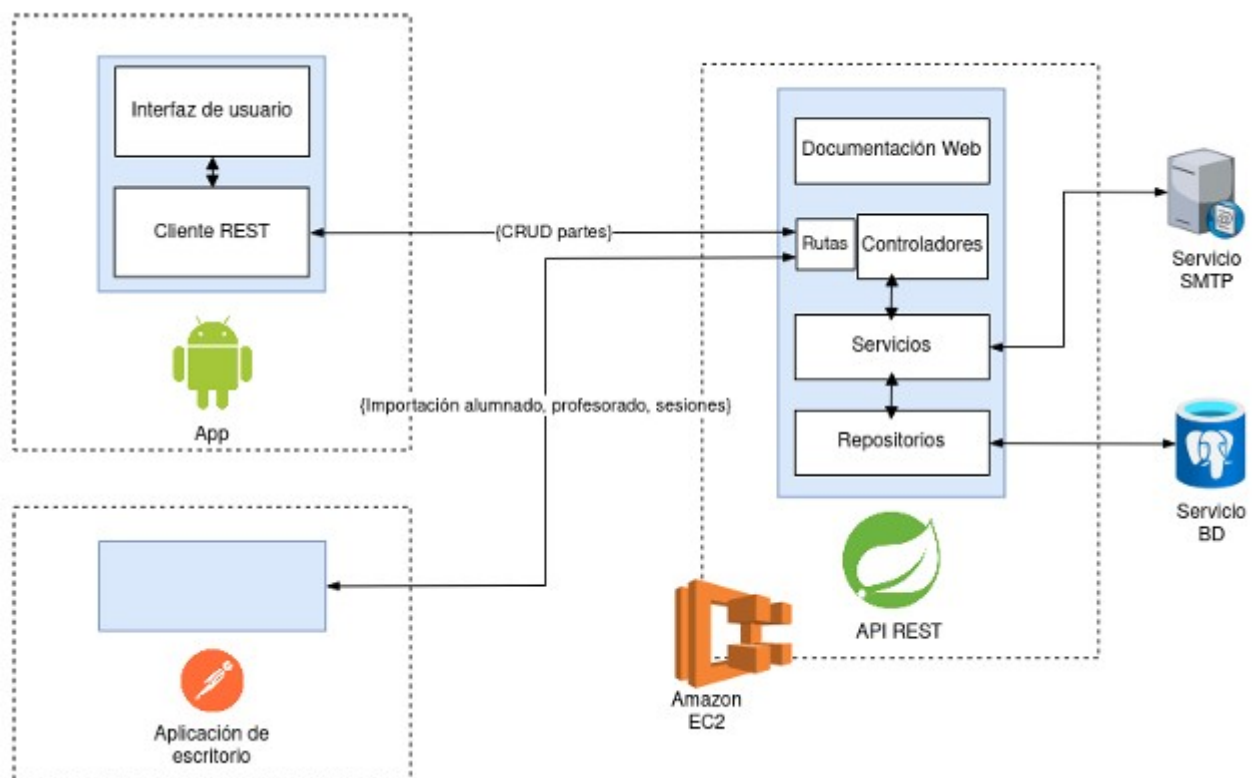
## API REST

Aparte de los endpoints conectados con las operaciones descritas anteriormente (creación, listado, modificación y borrado), la API REST ofrece endpoint de autenticación de usuarios.

## Arquitectura

Componentes del sistema

- Una aplicación para dispositivos Android que ofrece una interfaz gráfica a un restaurante.
- Una API REST que ofrece endpoints para la creación, listado, modificación y borrado de productos y pedidos, también dispone de autenticación de usuario.



## Aplicación para dispositivos Android

La aplicación es el punto de entrada al sistema para usuarios con perfiles de cliente, trabajador y administrador. Ofrece una interfaz gráfica para que los clientes puedan realizar pedidos y ve el estado de los mismos, los trabajadores pueden gestionar pedidos realizados, y los administradores pueden gestionar tanto pedidos como productos, así como crear trabajadores desde la interfaz.

Está desarrollada con Android SDK y la forman dos componentes principales:

- Interfaz de usuario: Es la interfaz gráfica que permite a los usuarios interactuar con el sistema.
- Cliente REST: Es el módulo que se comunica con la API REST a través de peticiones HTTP para acceder a los servicios de autenticación de usuarios y creación, listado, modificación y borrado de productos y pedidos.

El código de la aplicación está disponible en el repositorio de GitHub

[TFGAaron/ANDROID at main · aaroncr6/TFGAaron \(github.com\)](https://github.com/aaroncr6/TFGAaron)

## API REST

La API ofrece servicios de autenticación de usuarios, creación, listado, modificación y borrado de partes, e importación de datos de alumnado, profesorado y sesiones. Sus dos consumidores son la aplicación Android para la gestión de partes y la aplicación de escritorio Postman para la importación de datos.

La API está desarrollada con Spring Boot y contiene los siguientes módulos:

- Controladores: Procesan las peticiones de los clientes, transformándolas en los Data Transfer Objects (DTOs) correspondientes y ayudándose de los servicios para realizar las operaciones necesarias. Devuelven el resultado de la petición a los clientes.
- Servicios: Implementan las operaciones relacionadas con los datos del sistema, representados como entidades que se persisten y recuperan a través de los repositorios.
- Repositorios: Conectan las entidades con un servicio de base de datos PostgreSQL utilizando Hibernate. Puede consultarse el diagrama entidad-relación del modelo de datos en el anexo.

La API utiliza los siguientes servicios externos:

- Gmail SMTP: Se encarga del envío de los correos electrónicos que contienen las notificaciones que debe recibir el equipo directivo y la familia cada vez que se registra un parte.
- PostgreSQL: Gestiona la persistencia de los datos a través de un motor de bases de datos relacionales.

La API está desplegada en una instancia t2.micro de Amazon Elastic Compute Cloud (EC2) sobre una imagen Amazon Linux AMI.

Los endpoints de la API están documentados utilizando la herramienta Swagger. La documentación está accesible a través de la ruta /v1/docs/.

El código de la API está disponible en el repositorio de GitHub

[TFGAaron/SPRINGBOOT/gestorRestaurante at main · aaroncr6/TFGAaron \(github.com\)](https://github.com/aaroncr6/TFGAaron/SPRINGBOOT/gestorRestaurante)

## Satisfacción de requisitos

El proyecto cumple los requisitos de los cuatro módulos implicados tal y como se detalla a continuación.

### Acceso a datos

1. Se ha desarrollado una API REST utilizando Spring Boot. Esta API:

a. Accede a una base de datos PostgreSQL.

b. Se estructura en tres capas: controladores, servicios y repositorios.

c y d. La capa de controladores transforma las peticiones en DTOs y estos en entidades que puedan ser procesadas por los servicios, usando mapeadores para definir la correspondencia entre DTOs y entidades.

e. Define excepciones propias para identificar errores a nivel de datos.

f. Registra situaciones de error y advertencia a través de un mecanismo de logging.

g. La capa de repositorios utiliza consultas de CRUD y personalizadas con JPA.

h. Expone la documentación Swagger de los endpoints a través de la ruta /v3/api-docs/.

i. Ofrece un endpoint de autenticación y define tres roles de autorización: direcapición, profesorado y familia.

j. Está desplegada en una instancia t2.micro de Amazon Elastic Compute Cloud (EC2), de Amazon Web Services.



## Desarrollo de interfaces

1. Se ha desarrollado una aplicación para dispositivos Android que ofrece una interfaz gráfica de usuario:
  - a. Desarrollada con el editor gráfico Layout Editor de Android Studio.
  - b. Diseñada para que se puedan realizar las operaciones definidas (eficacia) con el menor número de interacciones posibles (eficiencia), que además son fáciles de aprender.
- 2.b. En el código de la API REST se han implementado pruebas unitarias.
3. Se ha creado un enlace del cual se puede descargar la apk, este estará en los manuales.
4. Junto al enlace de descarga de la apk, se ha incluido un fichero PDF con un manual de usuario de la aplicación y de administrador.

## Inglés técnico

Al principio de este documento se ha incluido un abstract de esta memoria escrito en inglés.

## Programación multimedia y dispositivos móviles

1. Se ha desarrollado una aplicación para dispositivos Android que ofrece una interfaz gráfica interactiva eficaz:
  - a. Incluye listas para representar los resultados de las búsquedas y elementos textView, editText, botones, check box,etc.
  - b. Notifica al usuario con una notificación simple cuando este realiza un pedido con éxito.
  - c. Incluye una acción para acceder al código de la aplicación que está en github.
  - d. Incluye variedad de menús en las diferentes actividades.
  - e. Incluye una opción para que la aplicación recuerde tu usuario y contraseña.
  - f. Está disponible en castellano e inglés.
  - g. Ofrece una actividad con contenido multimedia.
2. Se ha desarrollado una aplicación para dispositivos Android que accede a una base de datos PostgreSQL a través de una API REST:
  - a. Es capaz de mostrar, insertar, borrar y modificar información..
  - b. Se conecta a la API REST a través de un cliente HTTP.
  - c. El código se estructura en una capa de la interfaz de usuario y una de cliente HTTP REST.
  - d. Se incluye un manual de uso.

## Conclusiones y trabajo futuro

Tras presentar la propuesta formal he completado el desarrollo de la primera versión de este proyecto.

La primera versión incluye las funcionalidades especificadas, exceptuando la eliminación de usuarios por parte del administrador desde la aplicación android, la cuál no incluí ya que ir creando la aplicación no me terminó de convencer la idea-

El desarrollo de la aplicación siempre ha estado enfocado a cumplir los requisitos planteados por los diferentes módulos del curso implicados en el mismo.

Durante este proceso he encontrado diferentes problemas que he ido resolviendo a través de investigación y análisis del código desde diferentes fuentes. Por ejemplo a la hora de crear Pedidos y sus Detalle Pedido en base de datos, añadiendo todos los datos necesarios de las relaciones en su formato correcto.

Una propuesta de mejora sería añadir perfiles en la aplicación para los diferentes roles, añadir imágenes a los productos, mejorar la visualización de las diferentes listas y que la aplicación pueda gestionar más de un único restaurante.

## Fuentes Consultadas

<https://www.youtube.com>

<https://www.stackoverflow.com>

Chat GPT

Documentación oficial de Spring Boot.

Documentación oficial de Android.

## Diagrama entidad relación del modelo de datos

