



# StackWise

28/04/2025

—

Juan Mauricio Manzano Pachón

CESUR, Sevilla

2º DAM

## ÍNDICE

<b>1. Introducción</b>	<b>2</b>
1.1 Objetivo	2
1.2 Tecnologías Utilizadas	2
<b>2. Funcionalidades Principales</b>	<b>2</b>
2.1 Gestión de Empleados	2
2.1.1 Registro Seguro	2
2.1.2 Autenticación	3
2.2 Gestión de Inventario Inteligente	3
2.2.1 CRUD de Productos	3
2.2.2 Actualización Automatizada	3
2.3 Sistema de Pedidos con QR	4
2.3.1 Generación Dinámica de Menú	4
2.3.2 Procesamiento de Pedidos	4
<b>3. Estructura del Proyecto</b>	<b>5</b>
<b>4. Seguridad</b>	<b>5</b>
<b>5. Base de Datos (MariaDB)</b>	<b>5</b>
<b>6. Posibles Mejoras Futuras</b>	<b>6</b>
<b>7. Conclusiones</b>	<b>6</b>

# 1. Introducción

## 1.1 Objetivo

Se va a detallar el desarrollo de un sistema integral para la automatización de procesos en restaurantes, enfocado en tres pilares fundamentales: gestión de empleados, control de inventario y procesamiento de pedidos mediante tecnología QR. Esta solución tecnológica busca optimizar las operaciones diarias, reducir errores humanos y mejorar la experiencia del cliente.

## 1.2 Tecnologías Utilizadas

Se va a hacer uso de diferentes programas vistos durante el curso. Estos son los siguientes:

- **Backend**: Java + Spring Boot + Spring Security
- **Base de datos**: MariaDB
- **ORM**: Hibernate (JPA)
- **Seguridad**: BCrypt PasswordEncoder y FIREBASE
- **API REST + Spring Actuator** para monitorización

# 2. Funcionalidades Principales

## 2.1 Gestión de Empleados

### 2.1.1 Registro Seguro

Proceso de registro con validación en dos pasos:

- a. Registro inicial:
  - i. Campos obligatorios: nombre completo, correo electrónico, contraseña, restaurante.
  - ii. La contraseña se hashea inmediatamente con BCrypt (cost=12)
  - iii. Generación automática de código de validación alfanumérico de 5 caracteres
  - iv. Envío automático de correo electrónico con el código (implementado con JavaMail Sender)
- b. Validación de cuenta:
  - i. El empleado ingresa el código recibido
  - ii. Verificación contra base de datos con timeout de 24 horas

### *2.1.2 Autenticación*

- Flujo de login seguro:
  - Validación de credenciales (email + contraseña)
  - Verificación de cuenta activa
  - Generación de token JWT con roles (válido por 8 horas)
  - Auditoría de accesos (registro de IP, hora y dispositivo)

## 2.2 Gestión de Inventario Inteligente

### *2.2.1 CRUD de Productos*

- Modelo completo de productos:
  - Nombre (único, con búsqueda fuzzy)
  - Cantidad actual (decimal para unidades fraccionarias)
  - Unidad de medida (enum: kg, gr, lt, un, etc.)
  - Fecha de caducidad (con alertas automáticas)
  - Nivel mínimo de stock (para notificaciones)

### *2.2.2 Actualización Automatizada*

- Entradas:
  - API REST para registrar nuevas compras

- Actualización batch mediante archivos CSV
- Salidas:
  - Descuento automático al procesar pedidos
  - Sistema de reserva temporal de ingredientes
- Alertas:
  - Notificaciones push cuando stock < mínimo
  - Reporte semanal de productos próximos a caducar

## 2.3 Sistema de Pedidos con QR

### 2.3.1 Generación Dinámica de Menú

- Código QR vinculado a:
  - Mesa/cliente específico
  - Menú del día (considera disponibilidad de ingredientes)
  - Promociones personalizadas
- Endpoint móvil optimizado con:
  - Carga diferida de imágenes
  - Filtros por categorías
  - Búsqueda instantánea

### 2.3.2 Procesamiento de Pedidos

- Flujo completo:
  1. Cliente escanea QR y selecciona items
  2. Validación en tiempo real de disponibilidad
  3. Confirmación con tiempo estimado
  4. Actualización inmediata de inventario
  5. Notificación a cocina (integración con WebSockets)

### 3. Estructura del Proyecto

La estructura del proyecto va a seguir la de Spring Boot. Con las carpetas correspondientes como entidades, repositorios, servicios y controladores. Además, haremos una parte de testeo para verificar que todo funciona. Tendremos que configurar el pom.xml para usar las dependencias correspondientes.

### 4. Seguridad

- **Encriptación de Contraseñas:** BCrypt
- **Acceso protegido:**
  - Empleados no validados no pueden iniciar sesión.
- **Spring Security Configuración:**
  - Usuario y contraseña para `/actuator/health`
  - Seguridad personalizada para la API REST

### 5. Base de Datos (MariaDB)

La base de datos va a ser en MariaDB, por su forma estructural y la obtención de grandes informaciones. El acceso a la base de datos se va a hacer con JPA.

## 6. Posibles Mejoras Futuras

- Notificaciones por correo electrónico para envío del código de validación.
- Gestión de roles de usuarios (administrador, camarero, cocinero).
- Dashboard en frontend para ver pedidos e inventario en tiempo real.
- App móvil nativa para pedidos internos y gestión de stock.

## 7. Conclusiones

Este sistema proporciona una solución completa para la digitalización de restaurantes, ofreciendo:

- **Reducción de costos:** Minimizar mermas por caducidad o robo
- **Eficiencia operativa:** Automatiza procesos manuales propensos a errores
- **Experiencia cliente:** Pedidos rápidos sin intermediarios
- **Escalabilidad:** Arquitectura preparada para integraciones futuras (pagos digitales, fidelización)