

Chefsito

Gestor de Recetas con Fotos de la Despensa

Alumno: Luigi Alfonso Pineda Villamizar

Ciclo Formativo: Desarrollo de Aplicaciones Web

Curso académico: 2024/2025

Índice

1. Introducción

- En qué consiste el proyecto
- Módulos implicados en el desarrollo del proyecto

1.2 Justificación del proyecto y objetivos

- Por qué has elegido este proyecto
- Estado del arte
- Objetivos del proyecto
- Perfil del público objetivo

1.3 Planificación

- Mapa de historias de usuario (Producto mínimo viable y primer sprint)

1.4 Parte experimental

1.4.1 Análisis. ¿Qué hará la aplicación?

1.4.2 Diseño. ¿Cómo se hará la aplicación?

1. Introducción

En qué consiste el proyecto

El proyecto Chefsito consiste en el desarrollo de una aplicación web que permite al usuario registrar los ingredientes disponibles en su despensa, incluyendo la posibilidad de añadir fotografías para identificarlos fácilmente. A partir de esa información, la aplicación detecta los alimentos presentes y propone recetas posibles con lo que ya se tiene en casa. El sistema también muestra el nivel de coincidencia entre lo que está disponible y lo que falta para preparar cada plato.

Módulos implicados en el desarrollo del proyecto

- Bases de Datos (BD)
- Desarrollo Web en Entorno Cliente (DWECL)
- Desarrollo Web en Entorno Servidor (DWES)
- Diseño de Interfaces Web (DIW)
- Despliegue de Aplicaciones Web (DAW)

1.2 Justificación del proyecto y objetivos

Por qué he elegido este proyecto

He elegido este proyecto porque responde a una necesidad real y cotidiana: muchas veces las personas no saben qué cocinar con lo que ya tienen en casa, lo que termina generando desperdicio de comida. Además, me interesa combinar tecnologías modernas (visión por computadora, bases de datos y aplicaciones web) con un propósito práctico que aporte valor en la vida diaria.

Estado del arte

En el mercado existen aplicaciones de recetas y gestores de cocina, pero la mayoría se enfocan en ofrecer catálogos generales o recetas predefinidas. Muy pocas permiten registrar de manera intuitiva los ingredientes disponibles en casa y, menos aún, integran el reconocimiento automático mediante fotografías. Esta carencia abre una oportunidad de innovación en la que Chefsito puede destacar, al ofrecer un sistema práctico que une visión por computadora, gestión de despensa y recomendaciones personalizadas.

Perfil del público objetivo

La aplicación está dirigida principalmente a hogares y estudiantes, usuarios que deseen organizar mejor su despensa y optimizar sus compras. También se orienta a usuarios de dispositivos móviles y aplicaciones web que buscan soluciones prácticas para el día a día.

Objetivos del proyecto

Tipo de objetivo	Descripción
General	Desarrollar una aplicación web que sugiera recetas en función de los ingredientes disponibles, ayudando a reducir el desperdicio de alimentos.
Específico 1	Permitir el registro de ingredientes mediante fotos y entrada manual.
Específico 2	Implementar un sistema de reconocimiento de alimentos con posibilidad de corrección manual.
Específico 3	Generar recomendaciones de recetas indicando nivel de coincidencia con la despensa del usuario.
Específico 4	Gestionar digitalmente la despensa con funciones de alta, edición y consumo de ingredientes.
Específico 5	Reducir el desperdicio alimentario al fomentar el uso de lo que ya se tiene en casa.
Específico 6	Crear listas de la compra automáticas con los productos faltantes para completar recetas.

Tipo de objetivo	Descripción
Específico 7	Explorar e implementar tecnologías modernas (Java, Spring Boot, Angular, SQL) para consolidar conocimientos en desarrollo web.

1.3 Planificación

Mapa de historias de usuario – Producto mínimo viable y primer sprint

- Como usuario, quiero registrar los ingredientes disponibles en mi despensa para tener un control actualizado.
- Como usuario, quiero subir una foto de mis ingredientes para que el sistema los detecte automáticamente.
- Como usuario, quiero recibir sugerencias de recetas basadas en mis ingredientes disponibles.
- Como usuario, quiero generar automáticamente una lista de la compra con los ingredientes que me faltan para completar una receta.

1.4 Parte experimental

1.4.1 Análisis. ¿Qué hará la aplicación?

La aplicación permitirá:

- Registrar ingredientes de la despensa mediante entrada manual y mediante fotografías con reconocimiento asistido.
- Revisar y corregir los ingredientes detectados para asegurar precisión (edición manual).
- Gestionar la despensa: altas, ediciones, bajas, cantidades, unidades y fechas de caducidad; descuento automático al cocinar.
- Sugerir recetas basadas en los ingredientes disponibles, mostrando el nivel de coincidencia y los faltantes.
- Filtrar recetas por tiempo de preparación, categoría y preferencias dietéticas.
- Generar listas de la compra a partir de una o varias recetas seleccionadas.
- Autenticación de usuarios y perfil básico (preferencias de idioma).
- Interfaz web responsive pensada para uso en móvil y escritorio.

Prototipado de pantallas

Chefsito

Nombre

Correo electrónico

Contraseña

Repetir contraseña

CREAR CUENTA

¿Ya tienes cuenta?

[Inicia sesión](#)

Registro

ES ▾

Mi despensa

Buscar



Despensa

Lista de compras



Añadir producto



Todos Completados Pendientes

Zanahorias

Pechuga de pollo

Tomates

Pan

Pasta

Eliminar completados

Lista de compras



Recetas que puedes preparar

Fácil Vegetariana

Pollo a la parrilla 82 %

20 min

[Ver receta](#)

Sopa de tomate 73 %

30 min

[Ver receta](#)

Pepino relleno de zanahoria 67 %

15 min

[Ver receta](#)

Recetas sugeridas



Cocinaste: Pollo a la parrilla

Receta elegida

Ingredientes usados

Faltantes comprados

Tiempo total: 35 min

Desperdicio evitado

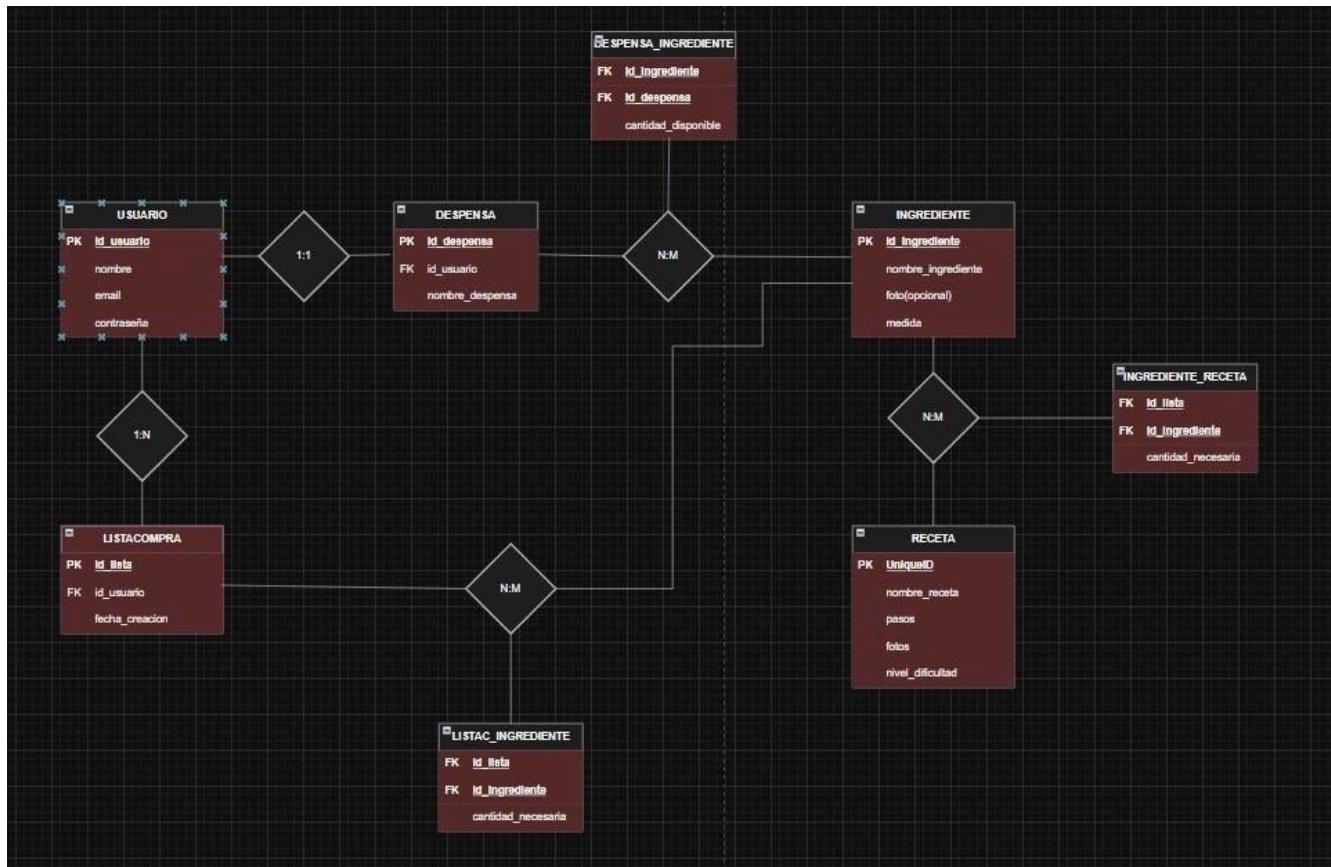
[Marcar como cocinada](#)

[Compartir](#)

[Volver a inicio](#)

Resumen final

Diagrama Entidad-Relación (E/R)



1.4.2 Diseño. ¿Cómo se hará la aplicación?

Estructura de la base de datos: se utilizarán tablas en PostgreSQL para gestionar usuarios, recetas ingredientes y la despensa.

Arquitectura del proyecto: aplicación de página única (SPA) en Angular, comunicada con una API REST desarrollada en Spring Boot

Seguridad: autenticación mediante tokens JWT, control de accesos con roles diferenciados (usuario/administrador) y limitación básica de peticiones (rate limiting).

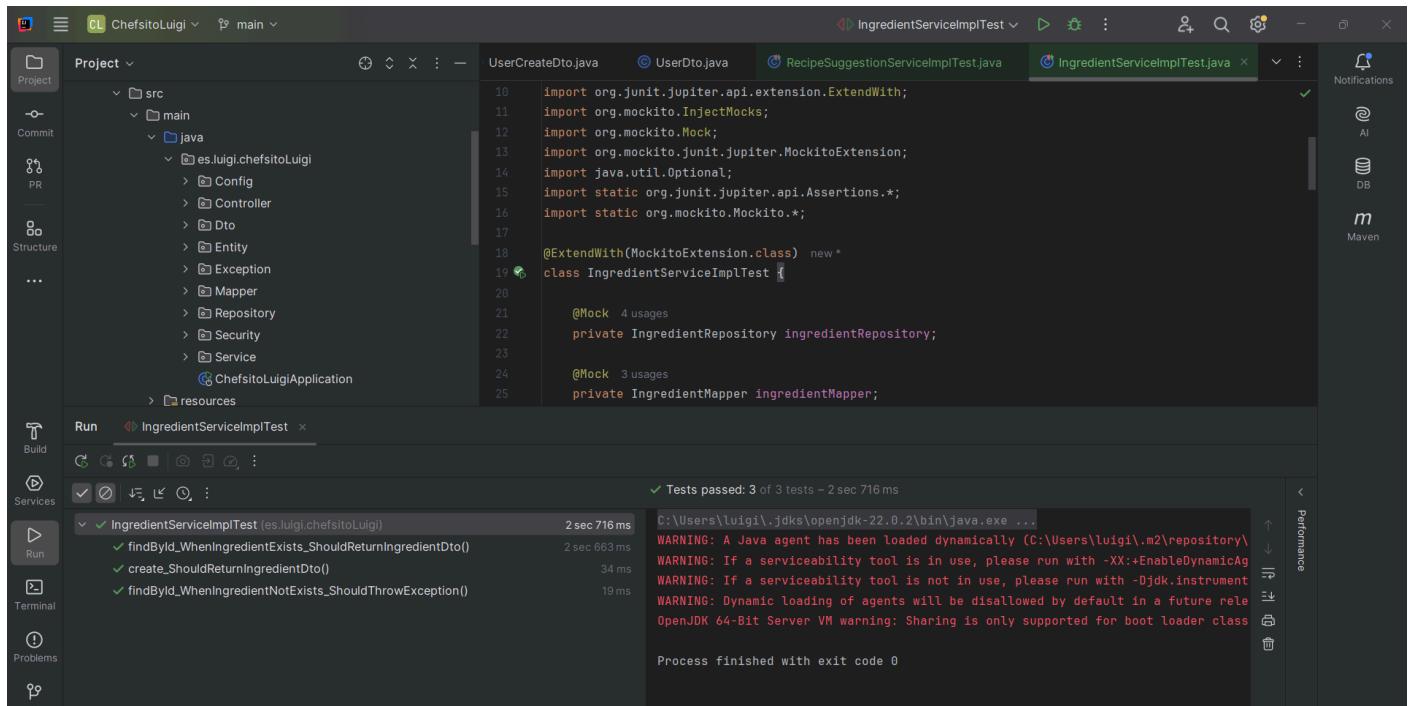
- **Internacionalización:** integración de ngx-translate para soportar varios idiomas, inicialmente español e inglés.

1.4.3 Implementación y pruebas

Tecnologías y estructura

Backend con Java 22 y Spring Boot 3, organizado en capas: Controller (REST), Service (lógica), Repository (JPA), DTO/Mapper (transformación) y Security (JWT y filtros). Base de datos: PostgreSQL 16. Pruebas unitarias con JUnit 5 y Mockito.

pruebas desarrolladas

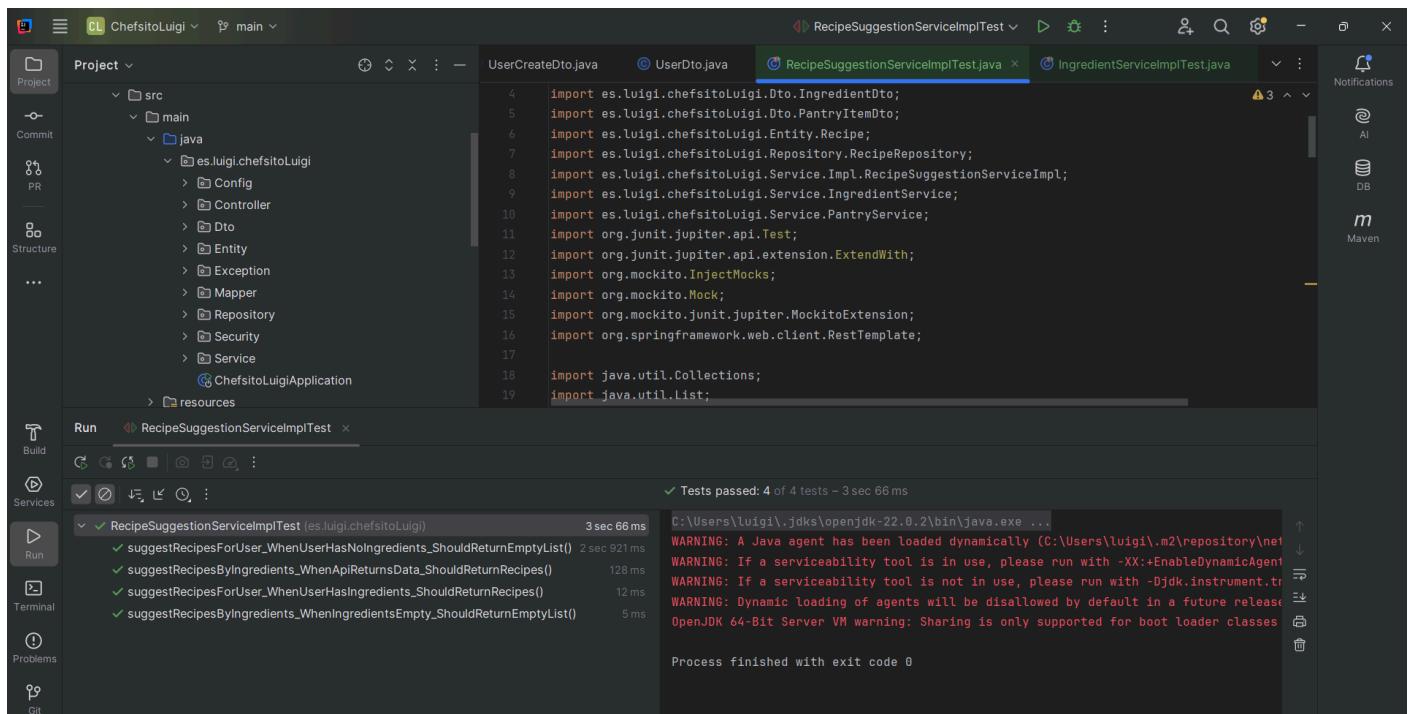


```
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.InjectMocks;
import org.mockito.Mockito;
import org.mockito.junit.jupiter.MockitoExtension;
import java.util.Optional;
import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.*;  
  
@ExtendWith(MockitoExtension.class) new *  
class IngredientServiceImplTest {  
  
    @Mock 4 usages  
    private IngredientRepository ingredientRepository;  
  
    @Mock 3 usages  
    private IngredientMapper ingredientMapper;  
  
    ...  
}
```

Tests passed: 3 of 3 tests – 2 sec 716 ms

C:\Users\luigi\.jdks\openjdk-22.0.2\bin\java.exe ...
WARNING: A Java agent has been loaded dynamically (C:\Users\luigi\.m2\repository)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgent
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader class

Process finished with exit code 0



```
import es.luigi.chefsitoLuigi.Dto.IngredientDto;
import es.luigi.chefsitoLuigi.Dto.PantryItemDto;
import es.luigi.chefsitoLuigi.Entity.Recipe;
import es.luigi.chefsitoLuigi.Repository.RecipeRepository;
import es.luigi.chefsitoLuigi.Service.Imp.RecipeSuggestionServiceImpl;
import es.luigi.chefsitoLuigi.Service.PantryService;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.mockito.InjectMocks;
import org.mockito.Mockito;
import org.mockito.junit.jupiter.MockitoExtension;
import org.springframework.web.client.RestTemplate;
import java.util.Collections;
import java.util.List;
```

Tests passed: 4 of 4 tests – 3 sec 66 ms

C:\Users\luigi\.jdks\openjdk-22.0.2\bin\java.exe ...
WARNING: A Java agent has been loaded dynamically (C:\Users\luigi\.m2\repository)\net
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgent
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes

Process finished with exit code 0

Repositorio del proyecto

El código fuente y avances de Chefsito se encuentran disponibles en el siguiente repositorio de GitHub:
<https://github.com/LUIGI071/ChefsitoLuigi>