



**salesianos**

**TRIANA**

# Alacena



**María Luisa Merón Campillos**  
**Proyecto interdisciplinar**  
**2º DAM 2024-2025**

# Índice

1. Introducción
2. Plan de Empresa
  - 2.1 Justificación
  - 2.2 Nombre y logo
  - 2.3 Producto
  - 2.4 Modelo de negocio
  - 2.5 Consumidores y posibles clientes
  - 2.6 Competencia
  - 2.7 Análisis DAFO
  - 2.8 Publicidad y promoción
3. Descripción del sistema.
  - 3.1. Listado de Historias de Usuario
4. Modelado y diseño
  - 4.1 Prototipado en Figma de la aplicación
  - 4.2 Diagrama de clases del modelo de dominio de la API
5. Diseño
  - 5.1 Diagrama de clases del diseño
6. Implementación
  - 6.1 Descripción de los paquetes y clases de cada aplicación

# 1. Introducción

Mi nombre es María Luisa Merón Campillos y estoy en segundo curso del ciclo formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma. Estoy haciendo un proyecto de Fin de Grado con la intención de simplificar la gestión de recetas y listas de compras en el ámbito doméstico. Este es un proyecto de una aplicación web llamada *Alacena*. Este proyecto ha sido desarrollado con Angular, diseñado utilizando Bootstrap y soportado por una API REST desarrollada con Java.

Esta herramienta permite a los usuarios registrar, organizar y personalizar recetas, administrar ingredientes favoritos y generar listas de compras optimizadas basadas en las necesidades culinarias. El objetivo principal de esta aplicación es optimizar la planificación de comidas, reducir el desperdicio de alimentos y facilitar la organización de compras.

## 2. Plan de Empresa

### 2.1 Justificación

#### ¿Por qué este negocio?

La planificación de comidas y la gestión de compras son tareas esenciales en los hogares modernos, pero a menudo resultan tediosas y desorganizadas, por lo que existe una creciente demanda de herramientas que faciliten estas actividades. *Alacena* responde a esta necesidad ofreciendo una solución digital que combina la gestión de recetas con la planificación de compras, todo en una plataforma accesible y fácil de usar.

#### ¿Existe competencia?

Sí, existen aplicaciones similares como *Yummly*, *Paprika*, o *Mealtime*, que también ofrecen funcionalidades de gestión de recetas y listas de compras. Sin embargo, muchas de estas soluciones son de pago, tienen interfaces complejas o no están traducidas al español, lo que limita su accesibilidad.

#### ¿Por qué motivos creo que el producto se venderá?

- **Facilidad de uso:** La interfaz basada en Bootstrap es intuitiva y atractiva.
- **Personalización:** Permite ajustar recetas según el número de porciones y clasificarlas por categorías.
- **Sostenibilidad:** Ayuda a reducir el desperdicio al optimizar listas de compras según los ingredientes ya disponibles.

### ¿Qué me diferencia de los demás?

- Uso de tecnologías como Angular, garantizando escalabilidad y facilidad de integración con otras plataformas.
- Funcionalidad de ajuste automático de cantidades según el número de porciones, algo poco común en aplicaciones gratuitas.

### ¿Qué hay de novedoso en mi producto?

- **Ajuste dinámico de porciones:** Los usuarios pueden modificar el número de comensales y la aplicación recalcula automáticamente las cantidades de ingredientes.
- **Gestión de favoritos y categorías personalizadas:** Los usuarios pueden marcar recetas e ingredientes como favoritos y organizarlas según sus preferencias.
- **Enfoque en sostenibilidad:** La generación de listas de compras optimizadas reduce el desperdicio de alimentos.

## 2.2 Nombre y Logo

**Nombre:** La palabra “alacena” proviene del árabe hispánico “alḥazāna”, que a su vez deriva del árabe clásico “ḥizānah”. Esta palabra evoca culturas cuya tradición gastronómica es rica en diversidad de ingredientes, recetas y sabores. Este nombre es corto y conciso, haciéndolo atractivo para reconocer fácilmente la aplicación.

**Logo:** El diseño es minimalista y moderno, acorde con la estética de la interfaz de Bootstrap. Los colores y elementos visuales transmiten accesibilidad, creatividad y conexión con la cocina casera.



# Alacena

## 2.3 Producto

### Descripción del producto:

*Alacena* es una aplicación web que permite a los usuarios:

- Registrar recetas con detalles como ingredientes, cantidades, pasos y categorías.
- Marcar recetas e ingredientes como favoritos para un acceso rápido.
- Generar listas de compras basadas en recetas seleccionadas, con ajustes automáticos según el número de porciones.
- Gestionar perfiles de usuario con autenticación segura (JWT) y roles diferenciados (usuario y administrador).
- Administrar usuarios (solo administradores).

**Finalidad:** Facilitar la planificación de comidas y la gestión de compras, optimizando el tiempo y los recursos del usuario.

### Necesidades que cubre:

- Organización de recetas y compras para usuarios ocupados.
- Reducción de desperdicio de alimentos mediante listas de compras inteligentes.
- Personalización de recetas según necesidades dietéticas o preferencias.

## 2.4 Modelo de Negocio

*Alacena* opera bajo un modelo **freemium**:

- **Gratuito:** Acceso a funciones básicas como registrar recetas, marcar favoritos y crear listas de compras.
- **Suscripción Premium (*Alacena Pro*):** Incluye funciones avanzadas como sincronización con dispositivos móviles, análisis de nutrientes, y exportación de listas de compras a aplicaciones de terceros. La suscripción se ofrecerá a través de pagos mensuales o anuales.
- **Publicidad:** Anuncios no invasivos en la versión gratuita, enfocados en productos relacionados con la cocina (utensilios, supermercados).

## 2.5 Consumidores y Posibles Clientes

### Tipo de consumidores:

- **Amas de casa y familias:** Buscan organizar comidas diarias de manera eficiente.
- **Jóvenes profesionales o estudiantes:** Necesitan soluciones rápidas para planificar

comidas saludables.

- **Entusiastas de la cocina:** Buscan inspiración y organización para recetas creativas.
- **Personas con dietas específicas:** Necesitan personalizar recetas según restricciones alimenticias.

#### **Características de los destinatarios:**

- Edad: 15-55 años.
- Interés en cocina casera, sostenibilidad y tecnología.
- Usuarios de dispositivos móviles u ordenadores con acceso a internet.

#### **Zona geográfica:**

Principalmente el mercado hispanohablante (España, México, Argentina, Colombia, etc.), con planes de expansión a otros idiomas en el futuro.

## **2.6 Competencia**

#### **Aplicaciones similares:**

- **Yummly:** Ofrece recetas personalizadas, pero su enfoque está en la recomendación más que en la gestión de compras.
- **Paprika:** Excelente para gestión de recetas, pero es de pago y no está optimizada para el mercado hispanohablante.
- **Mealime:** Centrada en planificación de comidas, pero con menos opciones de personalización de recetas.

#### **Diferenciadores de Alacena:**

- Interfaz en español adaptada culturalmente.
- Modelo freemium accesible.
- Ajuste dinámico de porciones y enfoque en sostenibilidad.

## **2.7 Análisis DAFO**

#### **Debilidades:**

- Dependencia inicial de usuarios gratuitos, lo que puede limitar ingresos.
- Competencia con aplicaciones establecidas en el mercado.
- Requerimientos técnicos para mantener la API y la base de datos.

#### **Amenazas:**

- Entrada de nuevos competidores con mayores recursos.
- Cambios en hábitos de consumo que reduzcan el interés en la cocina casera.
- Dependencia de tecnologías de terceros (Angular, Bootstrap, PostgreSQL).

#### **Fortalezas:**

- Interfaz intuitiva y responsiva gracias a Bootstrap.
- API robusta con autenticación JWT y documentación Swagger.
- Enfoque en sostenibilidad y personalización, atractivo para usuarios modernos.

#### **Oportunidades:**

- Creciente interés en cocina casera y sostenibilidad.
- Expansión a mercados internacionales con posibles traducciones.
- Posibles convenios con supermercados o marcas de utensilios de cocina.

## **2.8 Publicidad y Promoción**

#### **Soportes:**

- **Redes sociales:** Campañas en Instagram, TikTok y Pinterest con videos cortos de recetas y tutoriales de uso de la aplicación.
- **Publicidad en Google Ads:** Anuncios dirigidos a palabras clave como "gestión de recetas" o "planificación de comidas".
- **Colaboraciones con influencers:** Bloggers y creadores de contenido gastronómico para promocionar la aplicación.
- **SEO y contenido:** Blog con recetas y consejos de cocina para atraer tráfico.

#### **Coste estimado:**

- Campañas en redes sociales: \$500-\$1,000/mes.
- Google Ads: \$300-\$600/mes.
- Colaboraciones con influencers: \$200-\$500 por publicación.
- Blog y SEO: Costes internos de creación de contenido (~\$200/mes).

### 3. Descripción detallada del sistema y listado de historias de usuario:

Al abrir Alacena en un navegador o dispositivo compatible, el usuario llega a la pantalla de **Login** con un diseño centrado, con una paleta de colores cálidos y el logo de Alacena, que incluye el nombre en tipografía manuscrita y un ícono de una alacena estilizada. La interfaz muestra un formulario para el inicio de sesión. El usuario ingresa sus credenciales. Al hacer clic en "Iniciar sesión", la aplicación envía una solicitud al backend para autenticar las credenciales. Si son correctas, se redirige al usuario al panel principal.

Al elegir "Regístrate aquí", el usuario es redirigido a la **pantalla de registro (sign in)**. La interfaz es similar a la pantalla de login, la pantalla de registro incluye un formulario para introducir sus credenciales. El usuario completa los campos y hace clic en "Crear cuenta". La aplicación valida los datos (que las contraseñas coincidan y el correo sea válido) y envía una solicitud al backend para registrar al usuario. Tras un éxito, el usuario es redirigido a la pantalla de login para iniciar sesión con sus nuevas credenciales.

Una vez autenticado, el usuario accede al **panel principal (home)**, una interfaz personalizada que actúa como el corazón de Alacena. El diseño utiliza un layout flexible con un encabezado (conteniendo el logo y el "hamburger menu"), un área principal y un pie de página. Las funcionalidades disponibles incluyen:

- **Creación de Recetas:** El usuario selecciona "Crear receta" desde el menú y se le redirige a una pantalla de formulario. El usuario completa los campos y al guardar, la receta se envía al backend, donde se almacena en una base de datos asociada al usuario. La receta aparecerá en su listado de favoritos.
- **Búsqueda de Recetas:** Una barra de búsqueda en el panel principal permite filtrar recetas por "Nombre" o "Ingredientes". El usuario escribe un término o selecciona ingredientes disponibles y la aplicación consulta la base de datos para mostrar una lista de recetas coincidentes. Al seleccionar una, se muestra la pantalla **del detalle de la receta** completa con todos los datos de la misma.
- **Añadir Ingredientes a la Cesta de la Compra:** Desde la vista de una receta, el usuario ve un botón con forma de cesta para añadir todos los ingredientes de una receta automáticamente. En la sección "Cesta de la compra", el usuario puede revisar, editar cantidades o eliminar ítems. Además, al actualizar los datos de los ingredientes recién comprados, esos mismos ingredientes se añaden automáticamente a la lista de ingredientes de la **despensa**.
- **Añadir receta a Favoritos:** De la misma manera, también puede añadirse una receta a la lista de "Favoritos" por medio del icono con forma de corazón. El usuario puede consultar su lista accediendo desde su perfil o directamente en el menú lateral del panel principal.

Podemos acceder al **perfil del usuario** por ese mismo menú lateral, y en él podemos consultar y editar los datos del usuario. Además podemos encontrar accesos directos a las listas de favoritos, cesta de la compra y despensa (pantalla que nos muestra un listado de ingredientes que el usuario tiene disponibles).



### **3.1. Listado de Historias de Usuario**

HU64 - shared (Front)  
HU63 - Detalle de receta (Front)  
HU62 - Subida de imágenes en el post  
HU61 - Nueva Receta (Front)  
HU60 - Favoritos (Front)  
HU59 - Shopping Cart (Front)  
HU58 - Despensa (Front)  
HU57 - Profile (Front)  
HU56 - Home (Front)  
HU55 - Registro de usuario (Front)  
HU54 - Login (Front)  
HU53 - Añadir Tests  
HU52 - Registro de Admin  
HU51 - Vaciar Carrito de Compra  
HU50 - Eliminar Ingrediente del Carrito de Compra  
HU49 - Actualizar Cantidad de Ingrediente en el Carrito de Compra  
HU48 - Añadir Todos los Ingredientes de una Receta al Carrito de Compra  
HU47 - Añadir Ingrediente Individual al Carrito de Compra  
HU46 - Ver Carrito de Compra  
HU45 - Crear Carrito de Compra al Registro de Usuario  
HU44 - Reasignar Categoría Padre  
HU43 - Crear subcategoría  
HU42 - Asignar Categoría a Receta  
HU41 - Asignar Categoría a Ingrediente  
HU40 - Eliminar Categoría  
HU39 - Ver Detalles de una Categoría  
HU38 - Listar Todas las Categorías  
HU37 - Editar Categoría  
HU36 - Crear Categoría  
HU35 - Eliminar Ingrediente de Despensa  
HU34 - Actualizar Cantidad de Ingrediente en Despensa  
HU33 - Listar Ingredientes en Despensa  
HU32 - Añadir Ingrediente a Despensa  
HU31 - Eliminar Ingrediente  
HU30 - Detalle de Ingrediente  
HU29 - Listar Ingredientes  
HU28 - Editar Ingrediente

HU27 - Crear Ingrediente  
HU26 - Listar Recetas Favoritas de un Usuario  
HU25 - Eliminar Receta de Favoritos  
HU24 - Añadir Receta a Favoritos  
HU23 - Eliminar Receta  
HU22 - Editar Receta  
HU21 - Detalle de Receta  
HU20 - Listar Recetas  
HU19 - Crear Receta  
HU18 - Eliminar Foto de Perfil  
HU17 - Actualizar Foto de Perfil  
HU16 - Crear README  
HU15 - Actualizar Rol de Usuario  
HU14 - Obtener Usuario por ID  
HU13 - Obtener Todos los Usuarios  
HU12 - Borrar Usuario  
HU11 - Cambiar contraseña  
HU10 - Editar perfil de usuario  
HU09 - Recuperación de Contraseña  
HU08 - Verificación por email  
HU07 - Logout de Usuario  
HU06 - Login de Usuario  
HU05 - Registro de Usuario  
HU04 - Dockerfile  
HU03 - Docker compose  
HU02 - Prototipo en Figma  
HU01 - Modelo de datos

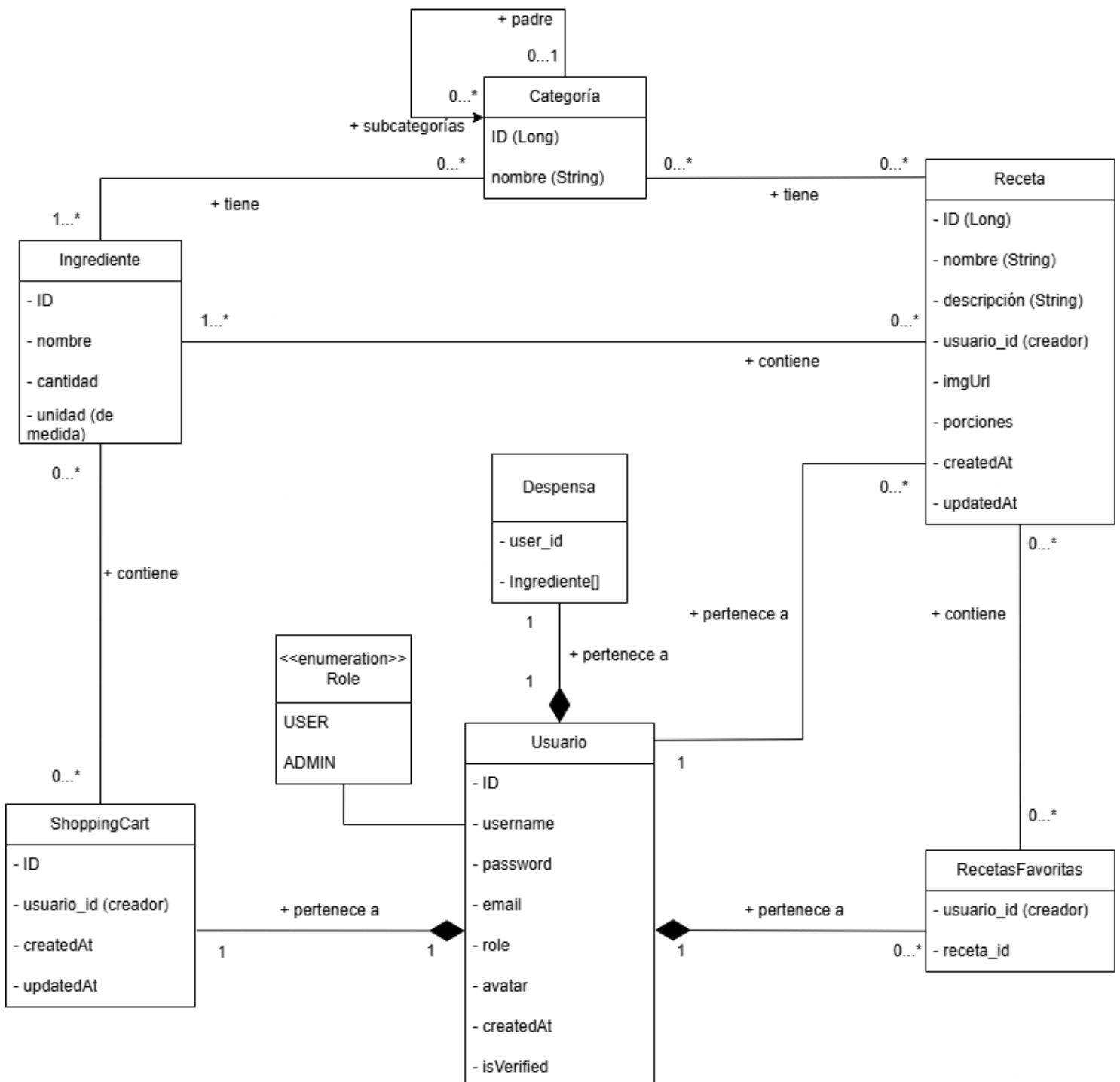
## 4. Modelado y diseño

### 4.1 Prototipado en Figma de la aplicación

El prototipo completo de esta aplicación puede consultarse en los siguientes enlaces:

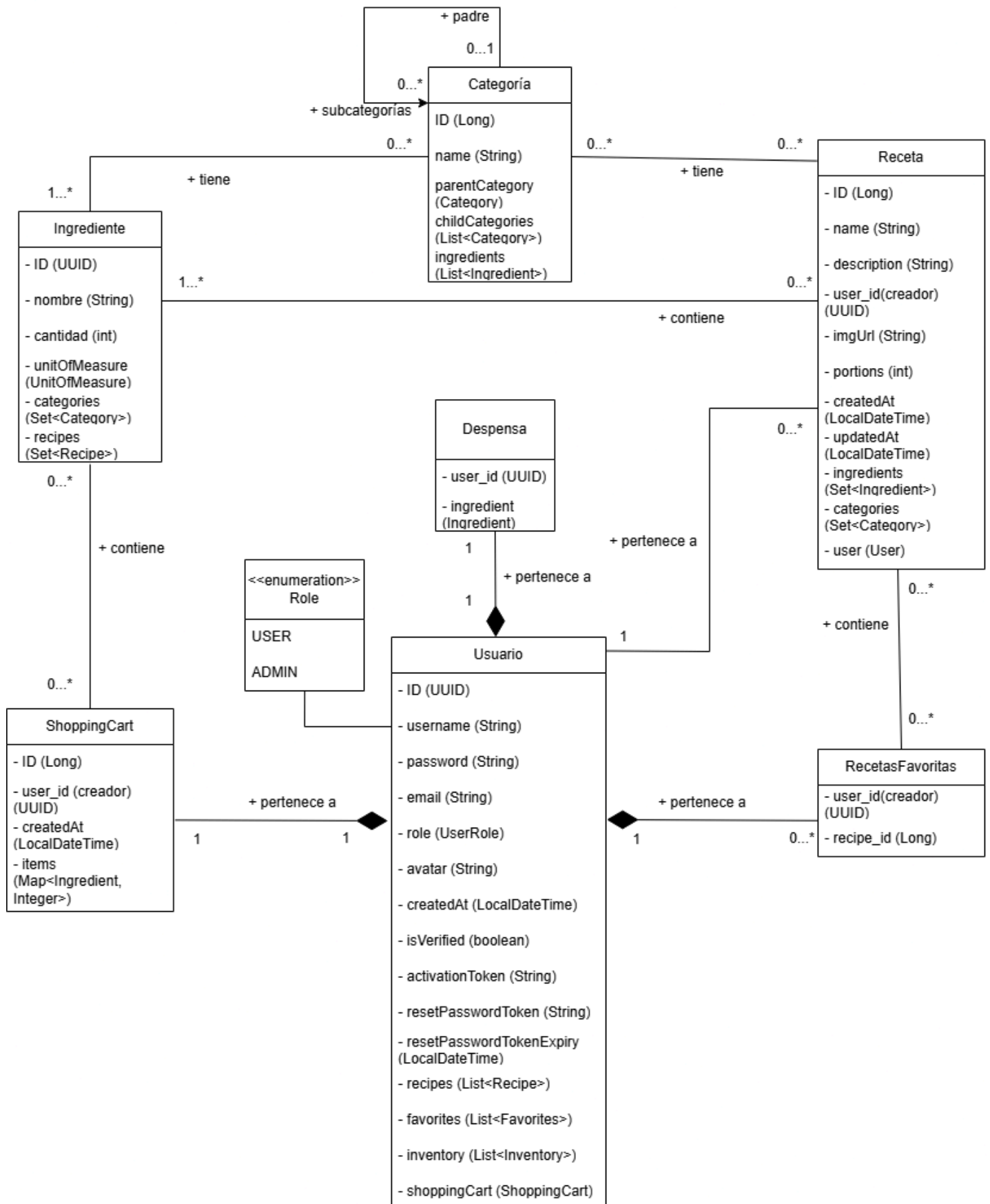
- [Enlace del archivo en modo edición.](#)
- [Enlace del archivo en "dev mode".](#)
- [Enlace del archivo en modo reproducción del prototipo.](#)

### 4.2 Diagrama de clases del modelo de dominio de la API



## 5. Diseño

### 5.1 Diagrama de clases del diseño:



## 6. Implementación

### 6.1 Descripción de los diferentes paquetes y clases de cada aplicación.

#### Aplicación Frontend (Angular)

##### Paquete: app

El paquete raíz de la aplicación Angular, responsable de inicializar la aplicación, configurar los proveedores globales y definir las rutas principales. Contiene los componentes y configuraciones esenciales para arrancar la aplicación. Clases/Componentes:

- **AppComponent:** Componente raíz que sirve como punto de entrada de la aplicación. Renderiza el enrutador (<router-outlet>) y puede incluir componentes compartidos como la barra de navegación. Responsabilidades: Proporciona el contenedor principal para la aplicación y carga el enrutador.
- **AppConfig:** Archivo de configuración que define los proveedores globales (por ejemplo, HTTP, Angular Material, interceptores). Responsabilidades: Configura los servicios y módulos necesarios para la aplicación.
- **AppRoutes:** Archivo que define las rutas principales de la aplicación. Responsabilidades: Mapear URLs a componentes de características (por ejemplo, /profile, /inventory).

##### Paquete: core

Contiene servicios, interceptores y guardias utilizados de manera transversal en la aplicación. Este paquete centraliza la lógica de negocio y las integraciones con el backend. Clases/Servicios:

- **UserService:** Servicio que gestiona operaciones relacionadas con los usuarios, como obtener y editar perfiles, cambiar contraseñas y actualizar fotos de perfil. Responsabilidades: Interactúa con los endpoints de usuario del backend (/user/profile, /user/change-password, etc.).
- **InventoryService:** Servicio que gestiona el inventario de ingredientes del usuario, permitiendo obtener, añadir, editar y eliminar elementos. Responsabilidades: Interactúa con los endpoints de inventario (/user/{id}/inventory).
- **IngredientService:** Servicio que gestiona los ingredientes, permitiendo obtener listas, detalles y realizar operaciones CRUD. Responsabilidades: Interactúa con los endpoints de ingredientes (/ingredient, /ingredient/{id}).
- **RecipeService** (Inferido): Servicio que gestiona recetas, permitiendo obtener listas, detalles y realizar operaciones CRUD. Responsabilidades: Interactúa con endpoints de recetas (por ejemplo, /recipe).
- **AuthInterceptor** (Inferido): Interceptor HTTP que añade tokens JWT a las solicitudes. Responsabilidades: Autentica solicitudes al backend.

##### Paquete: features

Contiene componentes específicos de las funcionalidades de la aplicación, organizados por módulo de característica. Cada subpaquete representa una sección de la interfaz de

usuario. Subpaquetes y Clases:

Subpaquete: profile

Gestiona la visualización y edición del perfil del usuario.

Clases:

- **ProfileComponent**: Muestra los datos del perfil (avatar, nombre, correo, contraseña) y permite abrir diálogos para editarlos.
- **EditProfileDialogComponent**: Diálogo para editar nombre y correo.
- **EditPasswordDialogComponent** y **EditAvatarDialogComponent**: Diálogos para cambiar contraseña y actualizar la foto de perfil, respectivamente.

Subpaquete: inventory

Gestiona el inventario de ingredientes del usuario.

Clases:

- **InventoryComponent**: Lista los ingredientes en tarjetas con botones para editar/eliminar y un botón para añadir nuevos ingredientes.
- **AddInventoryDialogComponent**: Diálogo para añadir/editar ingredientes con autocompletado.

Subpaquete: home (Inferido)

Página principal que muestra una lista de recetas.

Clases:

- **HomeComponent**: Lista recetas con opciones de filtrado/paginación.

Subpaquetes: favorites, shopping-cart, recipes (Inferidos)

Gestionan favoritos, carrito de compra y detalles/creación de recetas, respectivamente.

Clases:

- **FavoritesComponent**: Lista recetas marcadas como favoritas.
- **ShoppingCartComponent**: Muestra y gestiona el carrito de ingredientes.
- **RecipeListComponent**, **RecipeDetailComponent**, **RecipeCreateComponent**: Gestionan la visualización y creación de recetas.

Paquete: shared

Contiene componentes, tuberías y directivas reutilizables en toda la aplicación. Clases:

- **HeaderComponent**: Barra de navegación con enlaces a las secciones principales.
- **TruncatePipe** (Inferido): Tubería para truncar texto largo.
- **AutofocusDirective** (Inferido): Directiva para enfocar automáticamente inputs.

Paquete: models

Contiene interfaces TypeScript que definen los modelos de datos para la comunicación con el backend. Clases/Interfaces:

- **UserInterface**: Define modelos como UserProfile, EditProfileRequest,

EditPasswordRequest.

- **InventoryInterface**: Define modelos como InventoryItem, InventoryListResponse.
- **IngredientInterface**: Define modelos como Ingredient, IngredientList.

#### Paquete: environments

Contiene configuraciones de entorno. Clases:

- **Environment**: Configuración para desarrollo.
- **EnvironmentProd**: Configuración para producción.

#### Paquete: assets

Contiene archivos estáticos (no clases, pero incluido para completitud).

Contenidos: img/icons/arrow-icon.svg, img/icons/edit-icon.svg, img/icons/trash-icon.svg, images/avatar-placeholder.png., etc.

### **Aplicación Backend (Spring Boot, Inferida)**

#### Paquete: com.luisa.iAlacena

Paquete raíz del backend, contiene configuraciones globales y la clase de arranque.

Clases:

- **IAlacenaApplication**: Clase principal que inicia la aplicación Spring Boot.

#### Paquete: com.luisa.iAlacena.user

Gestiona usuarios, perfiles y autenticación. Clases:

- **User**: Entidad JPA que representa un usuario.
- **UserController**: Controlador REST para operaciones de usuario.
- **UserService**: Servicio que implementa la lógica de negocio para usuarios.

#### Paquete: com.luisa.iAlacena.inventory

Gestiona el inventario de ingredientes. Clases:

- **InventoryItem**: Entidad JPA para elementos del inventario.
- **InventoryController**: Controlador REST para operaciones de inventario.

#### Paquete: com.luisa.iAlacena.ingredient

Gestiona ingredientes y sus categorías. Clases:

Ingredient: Entidad JPA para ingredientes.

IngredientController: Controlador REST para operaciones de ingredientes.

#### Paquete: com.luisa.iAlacena.recipe, com.luisa.iAlacena.category, com.luisa.iAlacena.auth

Gestionan recetas, categorías y autenticación, respectivamente (inferidos, con estructuras similares).

Diagrama uml de paquetes (adjunto el enlace para mayor legibilidad).