

Ambrosia – Sistema Web de Gestión de Restaurante

ICES - Los mas duros del sistema

2025-05-29

Tabla de contenidos

1	Especificación de Requisitos de Software (ERS)	2
1.1	Introducción	2
1.1.1	Propósito	2
1.1.2	Alcance	2
1.1.3	Definiciones, acrónimos y abreviaturas	2
1.1.4	Referencias	2
1.1.5	Visión general	3
1.2	Descripción general	3
1.2.1	Perspectiva del producto	3
1.2.2	Funciones del producto	3
1.2.3	Características de los usuarios	3
1.2.4	Suposiciones y dependencias	3
1.3	Requisitos específicos	4
1.3.1	Requisitos funcionales	4
1.3.2	Requisitos no funcionales	4
1.4	Casos de uso	5
1.4.1	UC-01 Toma de pedido	5
1.4.2	UC-02 Preparación en cocina	6
1.4.3	UC-03 Notificación de orden lista	6
1.4.4	UC-04 Generación de factura	6
1.5	Modelo de datos	6
1.6	Apéndices	7

1. Especificación de Requisitos de Software (ERS)

1.1. Introducción

1.1.1. Propósito

Este documento define los requisitos funcionales y no funcionales del sistema **Ambrosia**, una aplicación web para la gestión de pedidos en restaurantes. Su objetivo principal es optimizar el flujo de trabajo entre meseros y cocina, permitiendo registrar pedidos en tiempo real, enviarlos automáticamente a cocina y notificar cuando estén listos.

1.1.2. Alcance

La aplicación será utilizada exclusivamente por el personal del restaurante. Las funcionalidades previstas incluyen:

- Toma de pedidos por parte de meseros.
- Envío automático de órdenes a cocina.
- Visualización del estado de las órdenes.
- Notificación a meseros cuando la comida esté lista.
- Gestión de mesas y generación de facturas.

No se contempla el uso por parte de clientes ni la generación de facturación electrónica en esta fase.

1.1.3. Definiciones, acrónimos y abreviaturas

- **ERS**: Especificación de Requisitos de Software
- **RF**: Requisito Funcional
- **RNF**: Requisito No Funcional
- **UC**: Caso de Uso
- **SRS**: Software Requirements Specification

1.1.4. Referencias

- IEEE 830-1998 – Recommended Practice for Software Requirements Specifications.
- Manual de estilo de software del equipo ICES.

1.1.5. Visión general

El resto del documento está estructurado en: - Descripción general del sistema (Sección General). - Requisitos específicos, incluyendo funcionales, no funcionales y casos de uso. - Modelos de datos. - Apéndices y anexos para futuras extensiones.

1.2. Descripción general

1.2.1. Perspectiva del producto

Ambrosia será una aplicación web cliente-servidor desarrollada con:

- **Frontend:** React (diseño responsivo para tablets).
- **Backend:** Django.
- **Base de datos:** PostgreSQL (sujeto a evaluación).
- **Despliegue:** Red local del restaurante o red Wi-Fi interna.

1.2.2. Funciones del producto

- Gestión de mesas.
- Registro de pedidos y órdenes.
- Visualización del estado de preparación.
- Generación de facturas.
- Autenticación de usuarios con roles definidos.

1.2.3. Características de los usuarios

- **Mesero:** toma pedidos, consulta órdenes y genera facturas.
- **Cocina:** recibe órdenes y actualiza su estado.
- **Caja:** consulta las órdenes para agilizar el proceso de facturación manual.
- **Administrador (futuro):** gestión de usuarios, reportes, configuración general.

1.2.4. Suposiciones y dependencias

- Se requiere conexión a red local para el funcionamiento del sistema.
 - Dispositivos móviles tipo tablet serán utilizados por los meseros.
 - Podría emplearse un dispositivo central por zona.
 - No se requiere conexión a internet para operar (modo offline en evaluación).
-

1.3. Requisitos específicos

1.3.1. Requisitos funcionales

1.3.1.1. RF-01 Gestión de mesas

- Listar mesas con su estado (disponible, ocupada, reservada, en limpieza).
- Cambio automático del estado según el flujo del pedido.

1.3.1.2. RF-02 Gestión de pedidos

- Seleccionar mesa.
- Registrar productos, cantidades y notas.
- Enviar orden a cocina.
- Cocina puede marcar orden como “en preparación” o “lista”.
- Notificar al mesero automáticamente cuando el pedido esté listo.

1.3.1.3. RF-03 Gestión de facturación

- Mostrar resumen de órdenes por mesa.
- Permitir cierre manual de cuenta.

1.3.1.4. RF-04 Gestión de usuarios

- Roles: Mesero y Cocina.
- Autenticación por credenciales.
- Rol de Administrador a incluir en versiones futuras.

1.3.1.5. RF-05 Integración con caja

- Enviar datos de órdenes para facilitar facturación manual.
- No incluye facturación electrónica por el momento.

1.3.2. Requisitos no funcionales

1.3.2.1. RNF-01 Arquitectura tecnológica

- React + Django.
- Diseño responsive para tablets.

1.3.2.2. RNF-02 Seguridad

- Autenticación y control de acceso por roles.
- Comunicación segura en red interna.

1.3.2.3. RNF-03 Rendimiento

- Latencia inferior a 1 segundo para operaciones críticas (envío de órdenes, notificaciones).

1.3.2.4. RNF-04 Escalabilidad

- El sistema debe poder expandirse para incluir futuras funcionalidades como reportes y reservas.

1.3.2.5. RNF-05 Soporte offline

- Posibilidad de operación desconectada mediante caché local (PWA o similar, en evaluación).

1.3.2.6. RNF-06 Dispositivos compartidos

- Soporte para operación desde un dispositivo central por zona.
-

1.4. Casos de uso

1.4.1. UC-01 Toma de pedido

- **Actor principal:** Mesero
- **Precondición:** Mesero ha iniciado sesión
- **Flujo principal:**
 1. Selecciona mesa.
 2. Añade productos.
 3. Envía orden a cocina.
- **Resultado esperado:** Orden registrada y enviada.

1.4.2. UC-02 Preparación en cocina

- **Actor principal:** Cocina
- **Flujo principal:**
 1. Visualiza orden nueva.
 2. Marca como “en preparación”.
 3. Marca como “lista”.

1.4.3. UC-03 Notificación de orden lista

- **Actor principal:** Mesero
- **Flujo principal:**
 1. Recibe notificación.
 2. Entrega comida.

1.4.4. UC-04 Generación de factura

- **Actor principal:** Mesero
 - **Flujo principal:**
 1. Consulta órdenes asociadas a la mesa.
 2. Genera factura.
 3. Cierra cuenta.
-

1.5. Modelo de datos

El sistema contará con las siguientes entidades:

Entidad	Descripción
Table	Representa mesas físicas del restaurante con su estado actual
Product	Elementos disponibles en el menú con sus precios
Order	Pedido asociado a una mesa con su estado de preparación
OrderItem	Productos individuales de una orden con cantidades y notas especiales
Bill	Factura generada al finalizar la atención en una mesa

Entidad	Descripción
CustomUser	Usuarios del sistema con roles (mesero, admin, chef) y datos de acceso
Notification	Notificaciones del sistema para diferentes eventos importantes

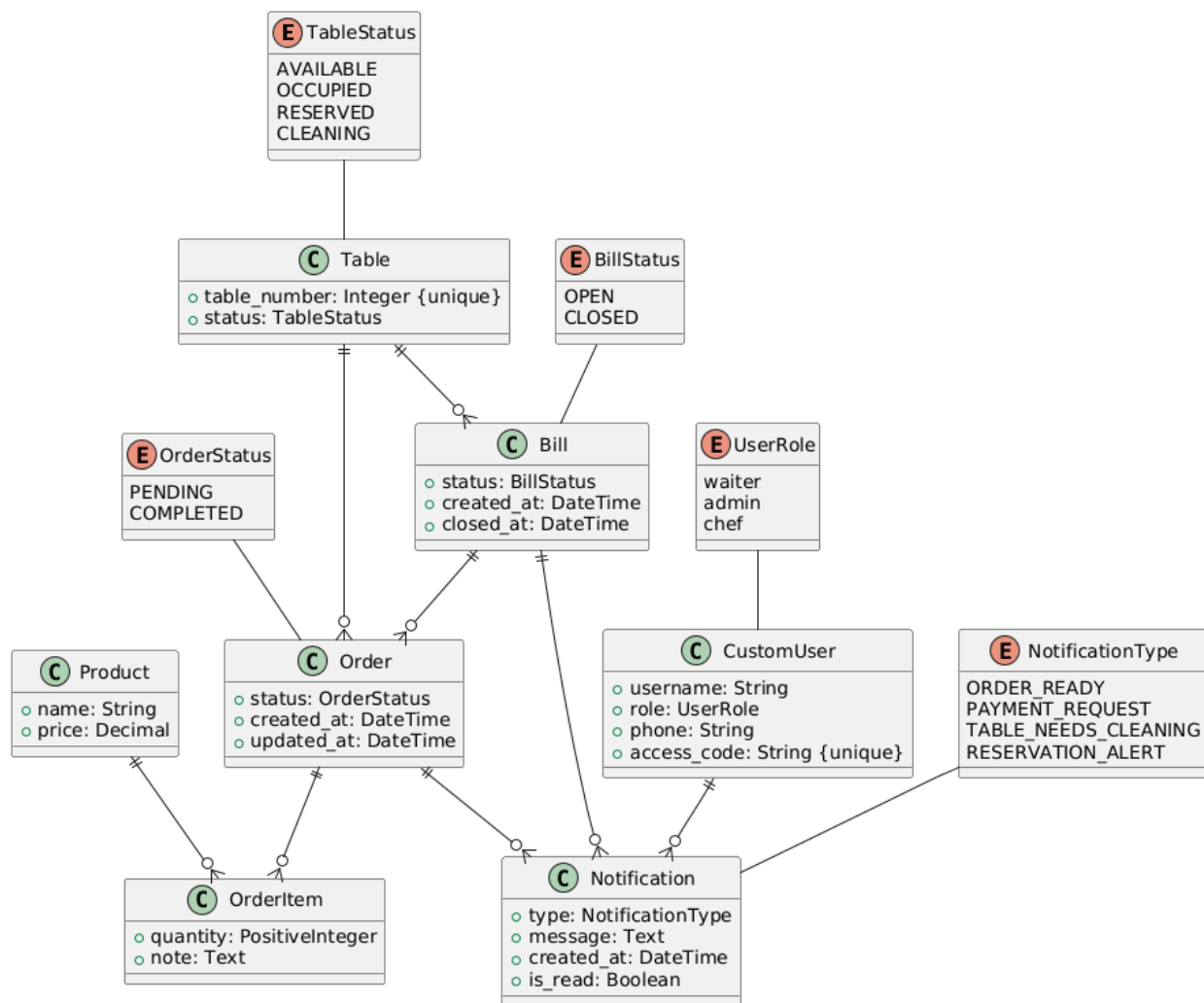


Figura 1: Modelo de datos

1.6. Apéndices

■ Futuras funcionalidades:

- Módulo de reservas.
- Administración avanzada de usuarios.
- Soporte a inventario.
- Reportes de ventas y rendimiento.

- Integración con sistemas de fidelización.
- Soporte offline y sincronización.
- Módulo administrativo completo.

■ **Tecnologías sugeridas:**

- Docker para despliegue local.
 - PostgreSQL como base de datos principal.
 - WebSocket o polling para notificaciones en tiempo real.
-