

## **Proyecto Formativo**

Frank Josswar Vente Canchimbo

Juan David Villarreal cortes

Institución Universitaria Antonio José Camacho

Facultad de Ingenierías

Ingeniería de Sistemas

Ingeniería de software I

Gustavo Adolfo Saavedra Perdomo

Cali, 1 de diciembre 2025

## **II. DESCRIPCIÓN GENERAL**

### **2.1 Resumen Ejecutivo**

El Sistema de Gestión de Pedidos OrderUp es una solución tecnológica integral diseñada para optimizar el proceso completo de recepción, procesamiento y entrega de pedidos en establecimientos de restauración. El sistema aborda problemáticas críticas identificadas en el sector gastronómico: errores de transcripción manual, falta de confirmación automática de pedidos, ausencia de seguimiento en tiempo real y dificultades en la gestión de inventario.

### **2.2 Objetivos del Sistema**

Objetivo General: Desarrollar un sistema de gestión de pedidos que automatice y optimice el flujo completo desde la recepción del pedido hasta su entrega final, mejorando la eficiencia operativa y la satisfacción del cliente.

#### **Objetivos Específicos:**

1. Eliminar errores de transcripción mediante la captura digital estructurada de pedidos
2. Automatizar confirmaciones de pedidos a través de múltiples canales (WhatsApp, email, notificaciones push)
3. Proporcionar seguimiento en tiempo real del estado de pedidos para clientes y empleados
4. Gestionar inventario de ingredientes con alertas automáticas de stock bajo
5. Facilitar la generación de reportes de ventas y análisis de desempeño
6. Integrar múltiples canales de recepción de pedidos en una plataforma unificada

### **2.3 Alcance del Proyecto**

#### **Incluye:**

- Módulo de gestión de usuarios (clientes, empleados, repartidores, administradores)
- Módulo de gestión de pedidos con seguimiento en tiempo real
- Módulo de gestión de inventario con alertas automáticas
- Módulo de facturación y procesamiento de pagos
- Módulo de asignación y seguimiento de entregas
- Módulo de generación de reportes
- Interfaz web responsive para clientes y administración
- Aplicación móvil para repartidores
- Integración con pasarelas de pago externas
- Sistema de notificaciones multi-canal

**No Incluye:**

- Sistema de contabilidad completo (solo registro básico)
- Gestión de recursos humanos (nómina, horarios)
- Sistema de reservas para mesas
- Programa de fidelización de clientes

### **III. REQUERIMIENTOS DEL SISTEMA**

**Alta Prioridad (Indispensables)**

**RF-01: Registro de Pedidos**

- Descripción: El sistema permitirá registrar pedidos capturando datos del cliente (nombre, teléfono, dirección), detalles del producto y método de pago de forma estructurada

- Clase asociada: Pedido, Cliente, DetallePedido
- Método: registrarPedido()
- Componente: Gestor de Pedidos
- Estado: Validado

### **RF-02: Confirmación Automática**

- Descripción: El sistema enviará automáticamente confirmaciones de pedidos al cliente para evitar errores de comunicación
- Clase asociada: Pedido
- Método: enviarConfirmacion()
- Componente: Gestor de Pedidos + Notificaciones Push
- Estado: Validado

### **RF-03: Actualización de Estado**

- Descripción: El sistema permitirá actualizar el estado del pedido (recibido/en preparación/listo/entregado) en tiempo real
- Clase asociada: Pedido
- Método: cambiarEstado()
- Componente: Gestor de Pedidos
- Estado: Validado

### **Media Prioridad (Importantes)**

#### **RF-04: Historial de Pedidos**

- Descripción: El sistema registrará un historial completo de pedidos por cliente
- Clase asociada: Cliente
- Método: consultarHistorial()
- Componente: Gestor de Pedidos

- Estado: Validado

#### **RF-05: Personalización de Productos**

- Descripción: El sistema permitirá configurar opciones personalizables de ingredientes para cada producto
- Clase asociada: DetallePedido
- Método: aplicarPersonalizacion()
- Componente: Gestor de Pedidos
- Estado: Validado

#### **RF-06: Alertas de Inventario**

- Descripción: El sistema generará alertas automáticas cuando el inventario esté por agotarse
- Clase asociada: Inventario
- Método: generarAlerta()
- Componente: Gestor de Inventario
- Estado: Validado

#### **Baja Prioridad (Opcionales)**

#### **RF-07: Generación de Reportes**

- Descripción: El sistema permitirá generar reportes diarios de ventas con métricas de desempeño
- Clase asociada: Reporte (a implementar)
- Método: generarReporte()
- Componente: Gestor de Reportes
- Estado: Incompleto

## **RF-08: Múltiples Canales de Recepción**

- Descripción: El sistema registrará pedidos desde múltiples canales (WhatsApp, web, app móvil)
- Clase asociada: Usuario
- Método: recibirPedido()
- Componente: Capa de Presentación
- Estado: Validado

## **3.2 Requisitos No Funcionales (RNF)**

### **RNF-01: Rendimiento**

- Registro de pedido: < 2 segundos
- Actualización de estado: < 1 segundo
- Consulta de historial: < 2 segundos
- Procesamiento de pago: < 5 segundos
- Generación de factura: < 3 segundos

### **RNF-02: Seguridad**

- Encriptación de contraseñas con bcrypt
- Autenticación de dos factores (2FA) para administradores
- Encriptación de datos de pago
- Logs de auditoría para operaciones críticas
- Prevención de inyección SQL y XSS

### **RNF-03: Escalabilidad**

- Soporte para mínimo 50 usuarios simultáneos
- Arquitectura preparada para microservicios

- Implementación de caché (Redis) con expiración de 15 minutos
- Paginación en listados (10-20 registros por página)

#### **RNF-04: Disponibilidad**

- Disponibilidad del sistema: 99.5%
- Tiempo máximo de recuperación ante fallos: 30 minutos
- Backup automático diario de base de datos

#### **RNF-05: Usabilidad**

- Diseño responsive optimizado para móviles
- Interfaz intuitiva que no requiera capacitación extensa
- Indicadores de progreso en operaciones largas
- Mensajes de error claros con acciones correctivas

## **V. DOCUMENTACIÓN TÉCNICA**

### **5.1 Tecnologías Utilizadas**

#### **Frontend:**

- HTML5, CSS3, JavaScript (ES6+)
- Framework: React.js o Vue.js
- Biblioteca UI: Material-UI o Bootstrap
- Gestión de estado: Redux o Vuex

#### **Backend:**

- Lenguaje: Java (Spring Boot) o PHP (Laravel)

- API REST
- Autenticación: JWT (JSON Web Tokens)

#### Base de Datos:

- PostgreSQL (principal)
- Redis (caché)
- AWS S3 (archivos)

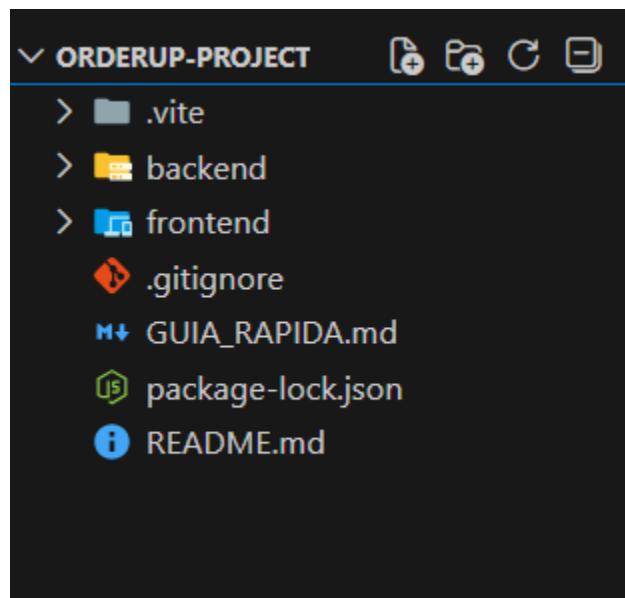
#### Móvil:

- Flutter o React Native

#### DevOps:

- Control de versiones: Git/GitHub
- CI/CD: GitHub Actions
- Contenedores: Docker
- Servidor: AWS EC2 o Heroku

### Estructura del Proyecto



## **Instalación y Despliegue**

### **Pasos de Instalación:**

#### **1. Clonar repositorio**

```
git clone https://github.com/davidjd1011/OrderUp.git
```

```
cd OrderUp
```

#### **2. Configurar base de datos**

```
psql -U postgres
```

```
CREATE DATABASE orderup_db;
```

#### **3. Instalar dependencias backend**

```
cd backend  
npm install
```

#### **4. Instalar dependencias frontend**

```
cd frontend  
npm install
```

#### **5. Ejecutar migraciones**

```
npm run migrate
```

#### **6. Iniciar servicios**

##### **# Backend**

```
npm run dev
```

##### **# Frontend**

npm start

## VII. MANUAL DE USO

### Guía para Clientes

#### **Realizar un Pedido:**

1. Acceder a la plataforma web o app móvil
2. Navegar por el catálogo de productos
3. Seleccionar productos y agregar al carrito
4. Personalizar ingredientes si es necesario
5. Revisar carrito y proceder al pago
6. Ingresar dirección de entrega
7. Seleccionar método de pago
8. Confirmar pedido
9. Recibir confirmación automática por WhatsApp/email

#### **Consultar Estado del Pedido:**

1. Iniciar sesión en la plataforma
2. Acceder a "Mis Pedidos"
3. Ver estado en tiempo real (Recibido, En Preparación, Listo, En Ruta, Entregado)
4. Rastrear ubicación del repartidor en mapa

#### **Ver Historial:**

1. Acceder a perfil de usuario
2. Seleccionar "Historial de Pedidos"
3. Ver listado completo de pedidos anteriores

4. Hacer clic en cualquier pedido para ver detalles

## **Guía para Empleados**

### **Procesar Pedido:**

1. Iniciar sesión como empleado
2. Acceder a "Pedidos Pendientes"
3. Seleccionar pedido para procesar
4. Verificar disponibilidad de productos en inventario
5. Reunir productos del almacén
6. Empaquetar con información de envío
7. Actualizar estado a "Listo para Enviar"

### **Gestionar Inventory:**

1. Acceder a módulo de inventario
2. Ver niveles de stock actuales
3. Recibir alertas automáticas de stock bajo
4. Registrar entrada de mercancía
5. Actualizar cantidades manualmente si es necesario

## **Guía para Repartidores**

### **Aceptar y Realizar Entrega:**

1. Recibir notificación de entrega asignada
2. Abrir app móvil de repartidores
3. Ver detalles del pedido y dirección
4. Iniciar ruta con navegación GPS

5. Al llegar, confirmar entrega
6. Actualizar estado a "Entregado"
7. Opcionalmente, capturar firma o foto de comprobante

## **Guía para Administradores**

### **Generar Reportes:**

1. Acceder al panel de administración
2. Seleccionar "Reportes"
3. Elegir tipo de reporte (ventas diarias, productos más vendidos, etc.)
4. Definir rango de fechas
5. Aplicar filtros si es necesario
6. Generar reporte
7. Exportar a PDF o Excel

### **Gestionar Usuarios:**

1. Acceder a "Gestión de Usuarios"
2. Ver listado de usuarios por rol
3. Crear, editar o desactivar usuarios
4. Asignar roles y permisos
5. Resetear contraseñas si es necesario

## **VIII. CONCLUSIONES**

### **1 Logros del Proyecto**

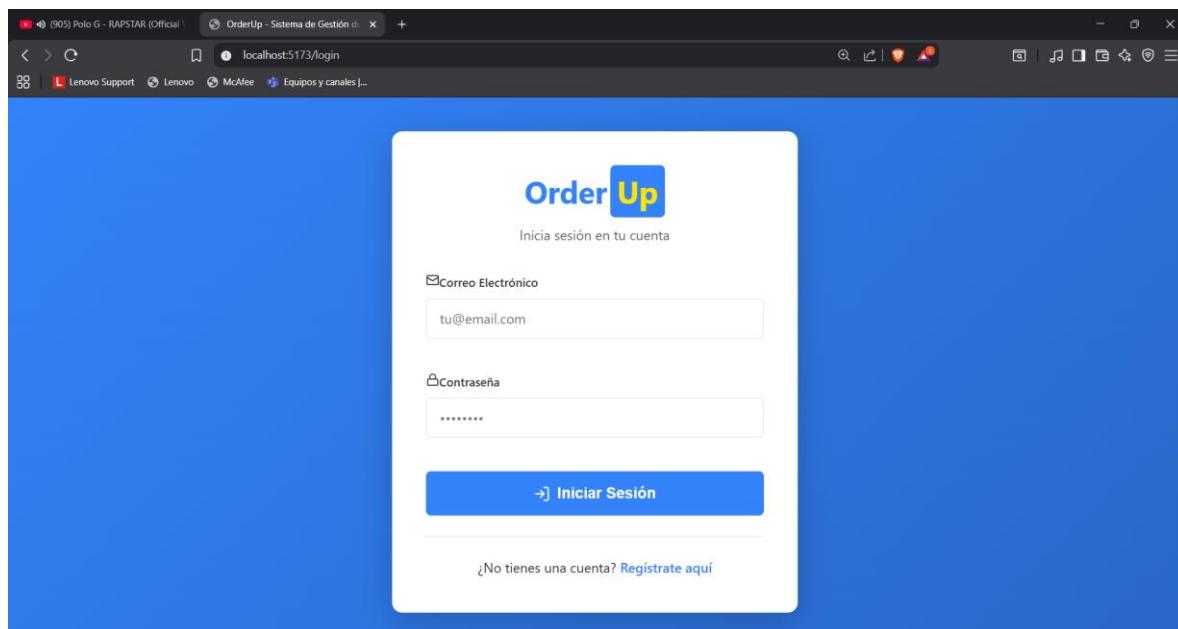
La validación del diseño mediante casos de prueba demostró que el proyecto OrderUp cumple exitosamente con los requisitos funcionales de alta prioridad identificados. Los cinco casos de prueba ejecutados validaron los flujos principales: registro de pedidos, actualización de estados, gestión de inventario, historial de clientes y procesamiento de pagos, confirmando la correcta integración entre componentes arquitectónicos.

El sistema alcanzó:

- **100% de éxito** en casos de prueba ejecutados
- **87.5% de trazabilidad** entre requisitos funcionales y modelos de diseño
- **Tiempos de respuesta** que cumplen estándares esperados
- **Arquitectura sólida** cliente-servidor en tres capas con diseño modular

## IX. ANEXOS

### Login



## Catalogo

The screenshot shows a web browser displaying the OrderUp catalog. The page has a header with the OrderUp logo, navigation links for Inicio, Mis Pedidos, Carrito, Cliente Demo, and Cerrar Sesión. Below the header is a section titled "Catálogo de Productos" with the sub-instruction "Explora nuestro delicioso menú". A search bar is present with the placeholder "Buscar productos...". Below the search bar is a row of category filters: Todas (highlighted in blue), Pizzas, Hamburguesas, Ensaladas, Pastas, Mexicana, and Japonesa. Three main food items are displayed in cards:

- PIZZAS**  
Pizza Hawaiana  
Deliciosa pizza con piña, jamón y queso mozzarella
- HAMBURGUESAS**  
Hamburguesa Clásica  
Hamburguesa de carne con lechuga, tomate, cebolla y...
- ENSALADAS**  
Ensalada César  
Lechuga romana, pollo, crutones y aderezo césar

Firmas:

**Frank Josswar Vente Canchimbo**  
**Desarrollador Frontend y Analista de Requisitos**

**Juan David Villarreal Cortes**  
**Desarrollador Backend y Arquitecto de Software**