



Simulacro prueba de desempeño

Proyecto: Sistema web de gestión de pedidos (restaurante)

Modalidad: Frontend con API simulada

Diseños: Proporcionados en Figma

1. Planteamiento del Problema

Una cadena de comida rápida desea digitalizar su sistema de pedidos mediante una aplicación web.

El sistema debe permitir:

- A los **clientes (usuarios)** registrarse, iniciar sesión y realizar pedidos.
- A los **administradores** gestionar productos y supervisar las órdenes.

Actualmente no existe un sistema digital, por lo que se requiere desarrollar una **aplicación web completa** que simule el flujo real de pedidos usando una API falsa con **JSON Server**.

El diseño visual, estructura de pantallas y componentes UI ya están definidos en **Figma** (login, menú, órdenes, dashboard admin).

2. Alcance del Proyecto

El coder deberá construir:



- ✓ Sistema de autenticación simulado
 - ✓ Manejo de roles (user / admin)
 - ✓ Consumo de API falsa con JSON Server
 - ✓ Gestión de pedidos
 - ✓ Panel administrativo con métricas
 - ✓ Persistencia de sesión
 - ✓ Separación clara entre vistas según rol
- ✗ No se requiere backend real.
 - ✗ No se requiere pasarela de pagos
 - ✗ No se requiere despliegue productivo (solo local)

3. Roles del sistema

Rol	Descripción
Usuario (User)	Cliente que compra productos
Administrador (Admin)	Gestiona productos y supervisa pedidos

Regla de negocio obligatoria:

Un usuario con rol **user** NO puede acceder a vistas de **admin**, y un admin no usa vistas de cliente.

4. Tecnologías Obligatorias

- HTML5
- CSS3
- Bootstrap 5 o Tailwind CSS
- JavaScript (Vanilla, sin frameworks)
- JSON Server (API falsa)
- LocalStorage o SessionStorage (manejo de sesión)



5. Módulo Usuario

Funcionalidades obligatorias

1. **Registro**
 - a. Crear cuenta nueva
 - b. Rol asignado automáticamente: user
2. **Login**
 - a. Validar credenciales contra JSON Server
 - b. Guardar sesión
3. **Menú de Productos**
 - a. Listar productos
 - b. Agregar al pedido
4. **Crear Orden**
 - a. Generar orden con estado inicial: pending
 - b. Asociar orden al usuario logeado
5. **Mis Pedidos**
 - a. Ver solo sus órdenes
 - b. Mostrar estado (pending, preparing, delivered, cancelled)
6. **Cerrar Sesión**

6. Módulo Administrador

Funcionalidades obligatorias

1. **Login (mismo formulario)**
 - a. Detectar rol admin
 - b. Redirigir a dashboard
2. **Dashboard**

Debe mostrar:

 - a. Total de órdenes realizadas
 - b. Órdenes en estado **pending**
 - c. Total de dinero del día actual



d. Lista de órdenes recientes

3. Gestión de Órdenes

- a. Ver detalle de la orden
- b. Cambiar estado:
 - i. pending
 - ii. preparing
 - iii. delivered
 - iv. cancelled

4. Gestión de Productos

- a. Crear nuevo producto
- b. Listar productos

7. Reglas de Seguridad y Lógica

- Rutas protegidas por rol
- Si no hay sesión → redirigir a login
- Validar que:
 - User solo vea sus órdenes
 - Admin pueda ver todas
- No se permite manipular datos directamente desde el navegador sin validación de rol

8. Diseño UI

- Seguir exactamente lo definido en **Figma**
- Diseño responsivo

Vistas esperadas según imagen:

- Login
- Menú usuario
- Pedidos usuario
- Dashboard admin



<https://www.figma.com/design/1AZ7oz3fIZXSZWA2wWDdcs/simulacro?node-id=0-1&t=A7IO3mlucMECv704-1>