

TPI - 2025 - Programación III - Tema 5

Objetivo:

Desarrollar un sistema web que tenga administración de usuarios y perfiles. Debe gestionar reservas de distintos tipos de negocios, un ABM de los datos de dominio y un dashboard.

Especificaciones generales:

El sitio web debe tener 2 (dos) roles de usuario, uno ADMIN y el otro USUARIO.

Desde el rol ADMIN se debe poder crear los usuarios y cambiar la password.

Desde el rol de usuarios login y registro (si no ha iniciado sesión) y el menú con las opciones correspondientes y la opción de cerrar sesión).

Dado que no se dispone de un sistema de backend tipo API REST, este se emulará utilizando el servicio gratuito de terceros mockapi.io. Por lo tanto las entidades se deben definir y crear en [mockapi](https://mockapi.io).

El material de trabajo es HTML5, CSS3 y JavaScript.

No hay plantilla inicial con lo cual cada grupo tiene la libertad de utilizar alguna plantilla si lo desea o algún framework para el uso de los estilos (ej: Bootstrap).

El trabajo se debe realizar en grupos de 2 a 4 integrantes y el código debe estar subido a un repositorio Git para que podamos acceder al código fuente. Dicho repositorio debe ser público, no privado.

Criterios de Evaluación:

Maquetación: Armado correcto del sitio, uso de los TAGS.

Estilos: CSS prolijos y diseño responsivo (flex, grid, distintos dispositivos).

Codificación JavaScript: Manejo correcto de los eventos, asincronismo, arrow functions, uso de objetos y arrays.

Manejo del DOM: Lectura, escritura, sobre el árbol del DOM.

Seguridad: Correcto manejo de los roles y correcta asignación del menú de cada rol.

Dominio Tema 5: Sistema de Pedidos Online (Pizzería)

Contexto:

Una pizzería desea permitir a los clientes hacer pedidos desde la web.

Roles:

- **ADMIN:** gestiona el menú (pizzas, bebidas) y los pedidos.
- **USUARIO:** se registra, arma su pedido y lo envía.

Entidades MockAPI:

- `users: { id, nombre, email, password, role }`
- `menu: { id, nombre, tipo, precio, disponible }`
- `orders: { id, userId, items[], total, estado }`

Requisitos:

- Catálogo de productos (con cards dinámicas).
- Carrito de compras (array en JS).
- Envío del pedido (POST a MockAPI).
- Admin puede ver y actualizar el estado (pendiente, en preparación, entregado).
- Un dashboard con un gráfico con el estado de los pedidos.