

Proyecto: Minisistema de Gestión para Spa de Mascotas

Tecnologías permitidas: HTML + CSS + JavaScript puro

Tipo de proyecto: Frontend modular y funcional (simulación de sistema real)

🎯 Objetivo General del Proyecto

Desarrollar un **Frontend funcional y organizado en módulos**, que simule un sistema básico de gestión para un **Spa de Mascotas**, aplicando apropiadamente:

- **HTML** → estructura, formularios, navegación
 - **CSS** → estilos, layout, responsividad
 - **JavaScript** → manipulación del DOM, sesiones simples
-

⚠️ Restricciones Fundamentales

- ✓ **Solo HTML, CSS y JavaScript (sin librerías externas)**
 - ✓ Todo el sistema debe funcionar en el navegador.
 - ✓ No se permite backend.
-

🔑 Módulo 0 — Inicio de Sesión y Gestión de Sesión

ID Requerimiento

- RF-L01** Mostrar una **vista de Login** con formulario (Usuario + Contraseña).
- RF-L02** Validar datos con JS. Si son correctos, mostrar los módulos del sistema (simulación de sesión).
- RF-L03** Incluir un **botón de “Cerrar Sesión”** visible en la barra de navegación.
-

🧱 Módulos Funcionales del Sistema

El sistema debe incluir una **barra de navegación** para acceder a los cuatro módulos principales.

1 Módulo de Registro — Dueños y Mascotas 🐶 🐾

📌 Funcionalidad

Permite registrar al dueño y su mascota por separado.

ID Requerimiento

RF-R01 Formulario de Dueño con campos: Nombre, Teléfono, Correo.

RF-R02 Formulario de Mascota con campos: Nombre, Especie, Raza.

2 Módulo de Agenda de Servicios

ID Requerimiento

RF-A01 Crear un formulario para **agendar un servicio** (fecha, hora, mascota, servicio).

RF-A02 Al enviar, la información se debe guardar.

3 Módulo de Carrito de Compras

El catálogo puede contener productos o servicios sin hora (baños, comida, juguetes, etc.).

ID Requerimiento

RF-C01 Mostrar un **catálogo dinámico** en pantalla usando JavaScript.

RF-C02 Cada ítem debe tener un botón "**Comprar**" para agregarlo al carrito.

RF-C03 Mostrar en el **carrito de compras, los productos que se van comprando**

RF-C04 Mostrar **Subtotal y Total** calculados automáticamente.

Entregables Mínimos Recomendados

El proyecto debe estar organizado con **separación de responsabilidades**:

Archivo Contenido

index.html Estructura principal, navegación, secciones de módulos, formularios.

style.css Diseño, colores, layout, estilos del catálogo y formularios.

app.js Lógica de sesión, registros, agenda, carrito, DOM dinámico.

Estructura sugerida del proyecto

- **index.html** → Página principal.
- **/css/style.css** → Todos los estilos del proyecto.
- **/js/app.js** → Toda la lógica JavaScript (login, validaciones, etc.).
- **/imagenes/** → Aquí colocas *todas* las imágenes: logo, mascotas, íconos.