

# Estructura general del sitio

Yo diseñé el sitio como una web interactiva para un restaurante ficticio. Tiene varias secciones: **Inicio**, **Menú**, **Reservas**, **Opiniones**, **Acerca**, **Contacto** y **Galería**. Algunas están en la misma página principal (`INICIO.html`) y otras tienen su propia página (como `menu.html`, `acerca.html`, etc.).

Usé HTML5 para la estructura, CSS externo para los estilos (`styles.css`) y JavaScript para la interactividad (como el saludo personalizado, el cambio de tema, y la carga dinámica del menú).

## Encabezado y navegación

En el `<header>` puse el nombre del restaurante y un menú de navegación con enlaces a todas las secciones. Algunos enlaces usan anclas (`#reservas`, `#opiniones`) para ir a partes de la misma página, y otros abren páginas nuevas como `menu.html` o `galeria.html`.

También incluí un botón para cambiar entre modo claro y oscuro. Ese botón tiene el ícono  y lo programé para que al hacer click cambie el tema del sitio usando JavaScript y `localStorage`, así el usuario puede elegir el modo que prefiera.

## Portada y bienvenida personalizada

En la sección de inicio (`#inicio`) puse un mensaje de bienvenida con una descripción del restaurante y un botón que lleva directamente al menú.

Justo después, agregué una sección donde el usuario puede escribir su nombre. Cuando lo guarda, el sitio lo saluda con un “¡Bienvenido, [nombre]!” en la parte superior. Esto lo hice con un formulario que guarda el nombre en `localStorage` y actualiza el DOM con JavaScript.

## Página del menú

En `menu.html` diseñé una sección para mostrar los platos del restaurante. Incluye:

- Un buscador por nombre
- Un filtro por categoría
- Un área donde se muestran los platos

La idea era cargar los platos desde un archivo `data.json` usando `fetch`, pero si ese archivo no está disponible o hay un error, el sitio muestra un mensaje: “Error cargando datos. Comprueba `data.json` o añade platos manualmente.”

Para que el usuario no se quede sin opciones, también programé un formulario para añadir platos manualmente. Se puede escribir el nombre, precio, categoría y una imagen opcional. Al guardar, el plato se agrega a la lista y se muestra en pantalla.

## Reservas

En la sección de reservas (`#reservas`) puse un formulario donde el usuario puede:

- Escribir su nombre
- Elegir fecha y hora
- Seleccionar cuántos invitados

Cuando se envía, la reserva se guarda y aparece en una lista de “Reservas próximas”. Esto lo hice para simular cómo funcionaría un sistema de reservas básico.

## Opiniones

En la sección de opiniones (`#opiniones`) el usuario puede dejar su nombre y una opinión de hasta 500 caracteres. Hay un contador que se actualiza en tiempo real para que no se pase del límite. Las opiniones se guardan y se muestran debajo del formulario.

## Acerca del equipo

En `acerca.html` presenté al equipo del restaurante con fotos y descripciones:

- **Sofía Morales** – Chef Ejecutiva
- **Andrés Ruiz** – Pastelero
- **Jose Carlos** – Sous-chef

Esto lo hice para darle personalidad al sitio y mostrar quién está detrás del restaurante , aunque no estoy seguro si a Gordon Ramsay le gusta el aceite.

También agregué una sección que explica qué distingue al restaurante de los demás, como el servicio directo desde la cocina y la atención personalizada.

## Contacto

En `contacto.html` hay un formulario con campos para nombre, email y mensaje (máximo 800 caracteres). También tiene un contador que se actualiza mientras el usuario escribe. El botón “Enviar” está preparado para enviar el mensaje

# Galería

En `galeria.html` puse una imagen del restaurante con vista a la Torre Eiffel, como si estuviera en París. Usé flechas para simular un carrusel de imágenes. Esto lo hice para mostrar el ambiente del local y atraer visualmente al usuario.

## Fecha y pie de página

En todas las páginas agregué la fecha actual al final (“Fecha: 18 de noviembre de 2025”) para que el sitio parezca más vivo y actualizado.

## Accesibilidad y buenas prácticas

Me aseguré de usar etiquetas semánticas (`<header>`, `<main>`, `<section>`, etc.), atributos `aria-label` para accesibilidad, y `label` en los formularios para que sean fáciles de usar con lectores de pantalla.

También separé los estilos en un archivo CSS externo para mantener el código limpio y facilitar cambios de diseño.

## Problemas que resolví

- Cuando el archivo `data.json` no se carga, el sitio muestra un mensaje y permite añadir platos manualmente.
- El botón de tema guarda la preferencia del usuario y cambia el diseño del sitio.

## Mejoras que tengo pensadas pero no tube tiempo

- Añadir estrellas en las opiniones.
- menu hamburguesa
- darle mas personalidad con estilos

## donde más me sostuve con IA

realmente siendo sincero me sostuve más en IA en los siguientes apartados.

- modo oscuro- realmente fue uno de mis principales problemas
- algún js se lo he pedido a copilot pero lo fui modificando a mi gusto

un ejemplo de una cosa que puso copilot pero quise dejarlo por estética fue el icono de modo oscuro, quería cambiarlo no para que no se notara la ia ya que en este punto todos la usamos en esta asignatura , pero no lo cambie porque me gusto como quedo , y me justifico y lo demuestro en los comentarios del código donde hago referencia a esto de lo que hablo

## **conclusiones**

aprecio mucho aprender y sinceramente yo valoro sus explicaciones profesor y el apoyo que ofrece por educamos con sus videos pero está pero su asignatura me da ganas de salirme del módulo, con respeto.