

Proyecto: Plataforma Web “La Pesquera”

Integrantes:

- Samuel Díaz García – Ficha 3171084
- Diego Alejandro Benítez – Ficha 3171084

Instructor: Gian Andrei Gomez

Fecha de entrega: 3/12/2025

Repositorio GitHub:

https://github.com/jeyb747/Proyecto_Pesquera.git

https://jeyb747.github.io/Proyecto_Pesquera/

B. Descripción del Proyecto

Propósito y Objetivos

La Pesquera es un sitio web diseñado para digitalizar los servicios del restaurante. Busca mejorar la experiencia del cliente permitiendo:

- Visualizar el menú completo.
- Realizar pedidos a domicilio.
- Hacer reservas.
- Llevar un registro del historial.
- Realizar compras mediante carrito con métodos de pago.
- Facilitar el contacto con el restaurante.

El objetivo principal es ofrecer una plataforma rápida, accesible y moderna, que centralice las funciones más importantes del negocio.

Público Objetivo

- **Edad:** 15 – 65 años
- **Ubicación:** Bogotá – bomberos
- **Necesidades:**
 - Consultar menú actualizado
 - Hacer pedidos de forma rápida
 - Ahorrar tiempo en filas
 - Reservar sin llamar
 - Realizar pagos digitales
- **Contexto de uso:**

Clientes que buscan una experiencia más ágil, especialmente en fines de semana y horas pico.

Problemática que resuelve

- Altos tiempos de espera en el local.

- Falta de un canal oficial para reservas.
- Dificultad para mostrar el menú digitalmente.
- Necesidad de automatizar pruebas y soporte a clientes.

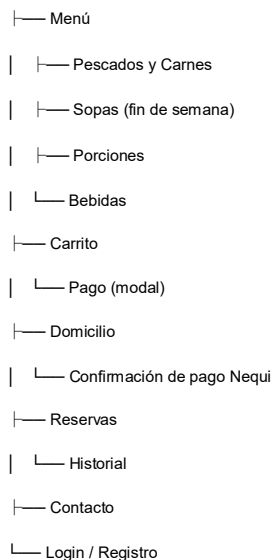
Alcance y funcionalidades principales

- Menú con categorías, imágenes, precios y descripción.
- Carrito de compras funcional.
- Métodos de pago (efectivo, tarjeta, Nequi).
- Envío de domicilio con formulario validado.
- Reservas con fecha/hora y número de personas.
- Historial de reservas o pedidos.
- Sistema básico de login y registro con localStorage.

C. Arquitectura de Información

Mapa del Sitio

Inicio



Estructura de navegación

- Barra superior fija con enlaces principales.

- Modal emergente para pagos.
- Secciones organizadas por categorías y cards.
- Formularios con validación.

Jerarquía de contenidos

1. navegación
2. Contenido principal
3. Llamados a acción (botones)
4. Formularios
5. Información secundaria
6. Pie de página

D. Decisiones de Diseño UI/UX (2–3 páginas)

Paleta de colores

Color	Código	Uso
Azul Marino	#0A3D62	Navegación, botones principales
Amarillo	#F1C40F	Acentos, resaltados, promociones
Gris Oscuro	#2C3E50	Textos principales
Blanco	#FFFFFF	Fondos

Justificación:

- El azul porque es el color del mar, que es donde sale la comida principal del
- El amarillo es el color de la playa y las costas del mar
- Los tonos neutros evitan distracción y mejoran la legibilidad.

Tipografía

- **Principal:** *Poppins*
- **Jerarquía:**

- **H1:** 32–40px, bold
- **H2:** 24–30px, semi-bold
- **H3:** 18–22px, medium
- **Párrafos:** 14–16px, regular

Componentes clave

Botones

- Normal: fondo azul, letra blanca
- Hover: azul más oscuro
- Active: presión con sombra reducida
- Disabled: gris claro

Formularios

- Bordes redondeados
- Sombras sutiles
- Indicadores de error
- Campos obligatorios marcados

Cards (Menú)

- Imagen destacada
- Nombre del plato
- Precio
- Hover con sombra elevada

Navegación

- Barra fija
- Menú hamburgués en móvil

- Enlace activo resaltado en amarillo

Principios UX aplicados

- **Consistencia:** colores, botones, tipografía uniforme.
- **Jerarquía visual:** títulos grandes → contenido → acciones.
- **Accesibilidad:** contraste alto y navegación con teclado.
- **Claridad:** textos cortos, íconos claros.
- **Retroalimentación:** mensajes de éxito/error, modal de pago.

E. Proceso de Desarrollo

Metodología

- Desarrollo iterativo.
- Construcción por módulos: Menú → Carrito → Pagos → Domicilio → Reservas.

Herramientas

- HTML5
- CSS3
- JavaScript
- GitHub
- LocalStorage
- Figma (para UI)
- VS Code

Distribución por integrante (ejemplo)

- **Integrante 1:** HTML + Estructura + CSS + UI/UX + JavaScript (carrito, login, reservas)

- **Integrante 4:** Documentación + GitHub

Desafíos enfrentados

- Guardar datos en localStorage
- Manejo del carrito y datos del menú
- Validación de formularios
- Modal de pago
- Integración entre páginas

Soluciones:

Código modular, pruebas manuales, comentarios en el JS y separación de scripts.

F. Pruebas de Usabilidad

Usuarios testeados

- 3 personas entre 18 y 40 años.

Escenarios probados

- Agregar productos al carrito
- Realizar un pago
- Completar el domicilio
- Hacer una reserva
- Navegar entre páginas

Hallazgos

- Usuarios no encontraban botón de “Pagar” → se rediseñó más visible
- Algunos botones estaban muy juntos → se separaron
- Se agregó confirmación visual en el modal

Mejoras implementadas

- Botón pagar en verde

- Validación reforzada
- Modal más grande
- Tipografías aumentadas

G. Aspectos Técnicos

Tecnologías utilizadas

- HTML5 (semántico)
- CSS3 (variables, responsive, grid/flex)
- JavaScript puro
- LocalStorage

Características especiales

- Modal dinámico de pagos
- Carrito persistente
- Validación de formularios
- Render dinámico del menú
- Subida de comprobante de pago (Nequi)

Optimización

- Minimización de imágenes
- Uso de variables CSS
- Reducción de repeticiones en código JS

Medidas de seguridad

- Validaciones en formulario
- Prevención de HTML Injection
- Sanitización de textos mostrados

H. Checklist de Accesibilidad (1 página)

- ✓ Contraste suficiente entre colores
- ✓ HTML semántico
- ✓ Formularios con etiquetas <label>
- ✓ Atributos alt en imágenes
- ✓ Navegable por teclado
- ✓ Fuentes suficientemente grandes
- ✓ Prueba con Lighthouse
- ✓ Botones accesibles con rol correcto

I. Conclusiones y Trabajo Futuro (1 página)

Conclusiones

- El proyecto permite gestionar el restaurante digitalmente.
- Se logró una plataforma funcional, intuitiva y accesible.
- Se adquirieron habilidades en UI, frontend y lógica JS.

Trabajo futuro

- Integración real con base de datos y backend.
- Panel administrativo.
- Envío de correos al registrar usuario.
- Mejorar seguridad de datos.
- Implementar pagos en línea reales.

J. Referencias y Recursos

- MDN Web Docs

- W3Schools
- Figma
- Tipografía: Google Fonts
- Documentación de HTML, CSS y JS
- Capturas propias del proyecto