PAPELERÍA TECH-NO

Introducción

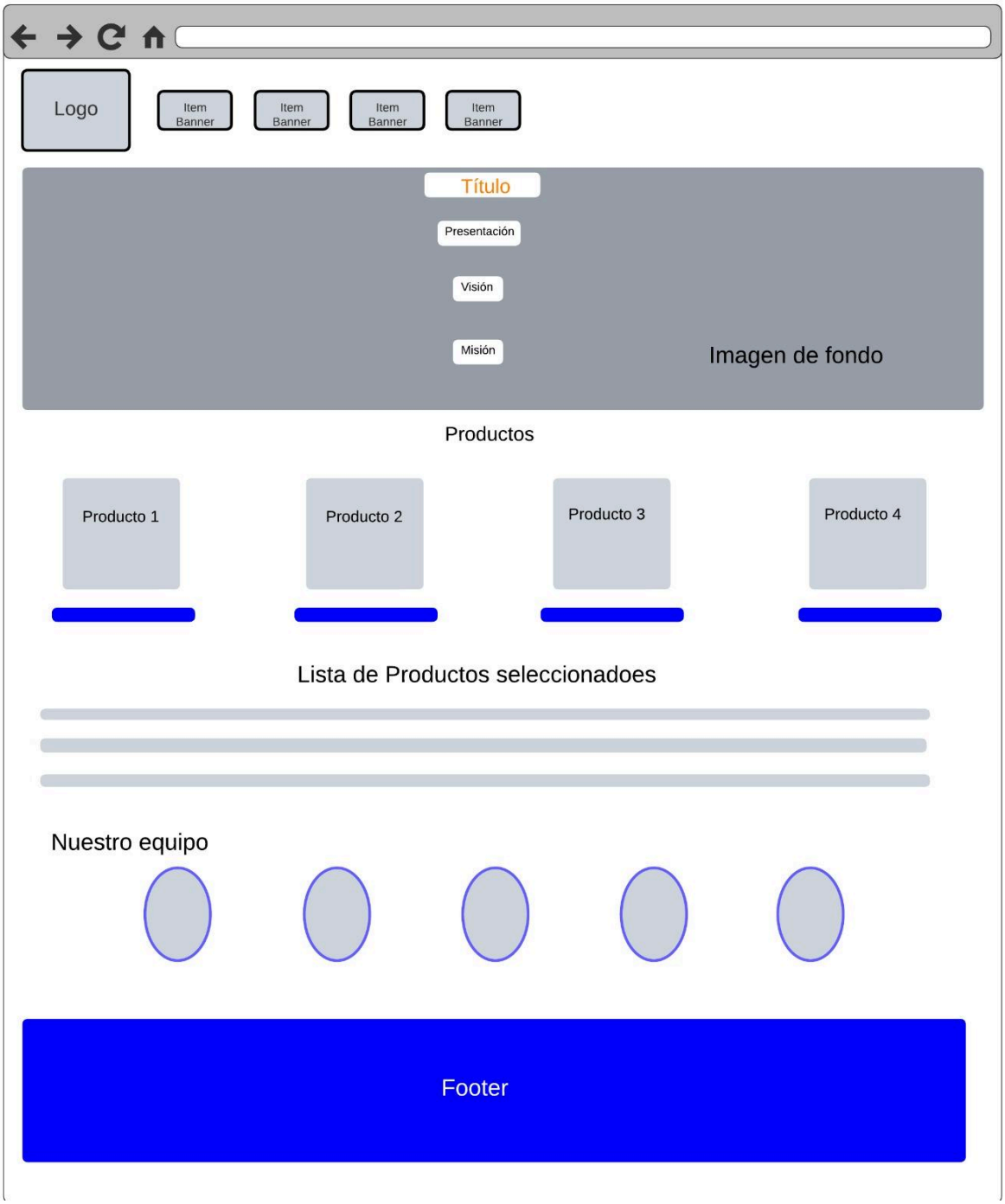
Papelería Techno es una tienda virtual que ofrece una amplia gama de productos de papelería y artículos de oficina, pensada para cubrir las necesidades de los usuarios. Permite realizar compras de manera accesible y conveniente, desde cualquier lugar y en cualquier momento, brindando una experiencia de compra rápida, personalizada y confiable.

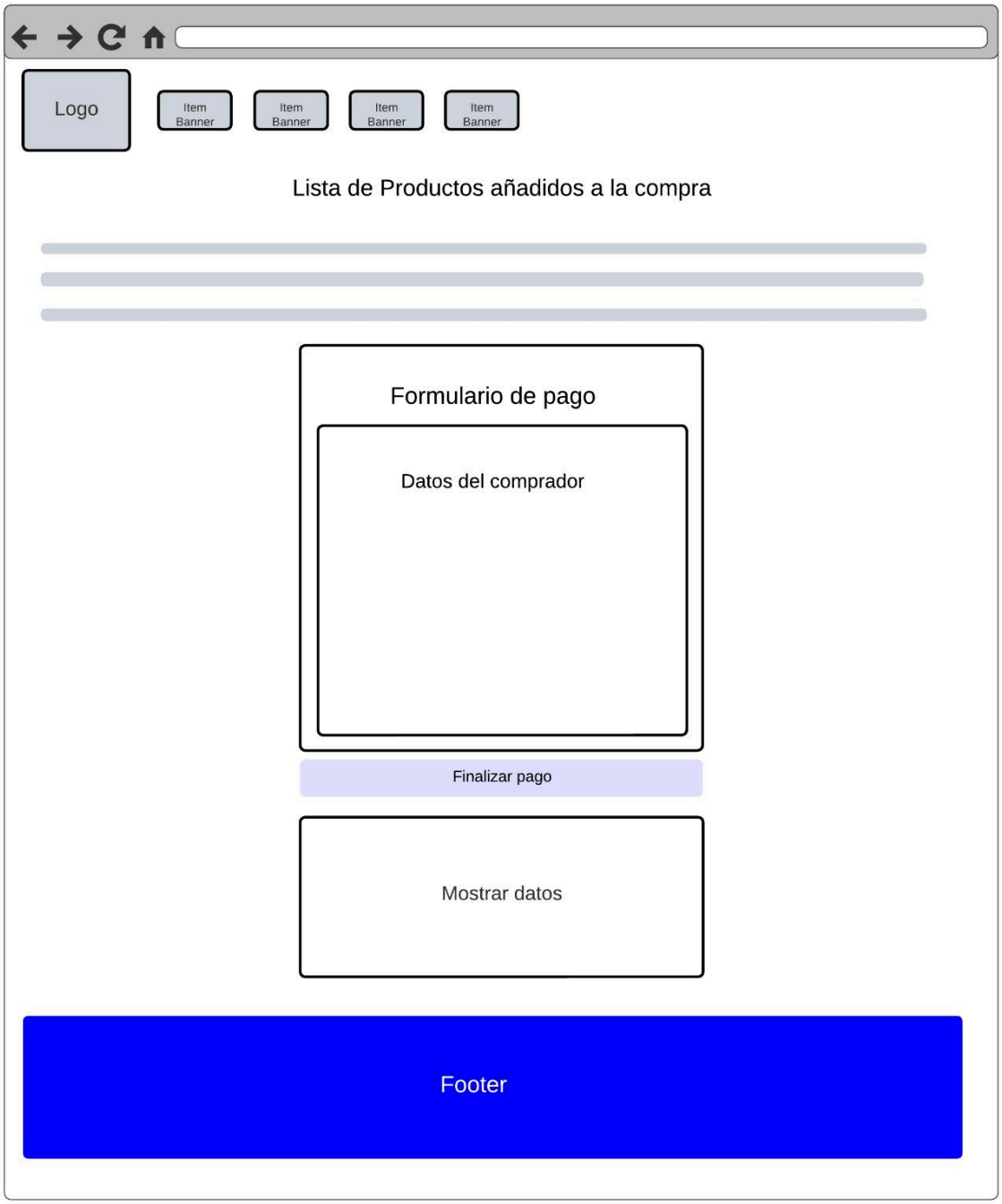
Para optimizar la gestión de ventas y operaciones, **JavaScript** juega un papel clave al agregar interactividad y funcionalidad dinámica a la página web. **Javascript** crea herramientas que facilitan tareas como la gestión de inventarios, el cálculo de precios, la generación de facturas y el manejo de la tienda en línea. De esta manera, JavaScript mejora la eficiencia y la experiencia tanto para el vendedor como para el cliente.

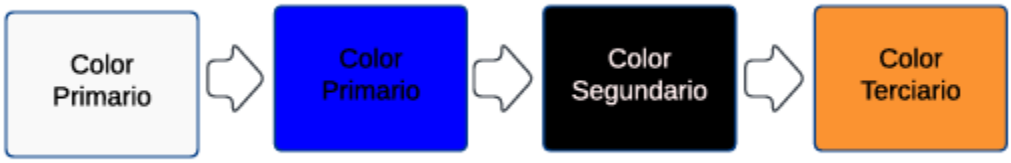
Objetivos

- Agregar los productos al carrito de compras.
- Proporcionar de manera detallada los productos seleccionados por el usuario.
- Realizar la compra y conectar con el formulario de pago.
- Registrar y validar los datos del usuario.
- Imprimir la información de la venta y del usuario.

Boceto / Wireframe


Logo
Item Banner
Item Banner
Item Banner
Item Banner
Título
Presentación
Visión
Misión
Imagen de fondo
Productos
Producto 1
Producto 2
Producto 3
Producto 4
Lista de Productos seleccionadoes
Nuestro equipo
Footer


Logo
Item Banner
Item Banner
Item Banner
Item Banner
Lista de Productos añadidos a la compra
Formulario de pago
Datos del comprador
Finalizar pago
Mostrar datos
Footer


Color Primario
Color Primario
Color Segundario
Color Terciario

Roles asignados

Alejandro Romero: Líder del proyecto, responsable de supervisar y coordinar el proyecto que está desarrollando, busca y gestiona que las metas se cumplan en en los plazos acordados dentro del equipo, monitorea el avance del proyecto. También es el tester, es decir, es responsable de asegurarse de que el software desarrollado funcione correctamente y cumpla con los requisitos especificados. También cumple rol como desarrollador.

Cristian Quintero: Diseñador, encargado de darle la estructura y apariencia del software una vez acordado en grupo. También es el principal desarrollador del grupo, encargado de escribir, implementar y mantener el código que constituye el software de un proyecto.

Marco Lagarez: Diseñador, encargado de la creación de wireframes y la paleta de colores del proyecto. También cumple rol de desarrollador secundario.

Jhonatan Ramirez: Tester, encargado de probar el código y se asegura de que el software funcione correctamente, aportando además su punto de vista que ayudan a mejorar el proyecto.

Oscar Vera: Tester, encargado de probar el código y se asegura de que el software funcione correctamente, aportando además su punto de vista que ayudan a mejorar el proyecto.

Desarrollo de funcionalidades Javascript

Las funcionalidades implementadas con el lenguaje Javascript fueron las siguientes:

- **Función: addToCart(productName, productPrice)**

Descripción: Añade un producto al carrito. Si el producto ya existe, incrementa su cantidad y actualiza su subtotal. Si no, crea un nuevo objeto y lo añade al carrito.

- **Función: updateCart()**

Descripción: Actualiza la interfaz del carrito, mostrando los productos en una tabla y calculando el total.

- **Función: updateQuantity(productName, action)**

Descripción: Permite aumentar o disminuir la cantidad de un producto en el carrito.

- **Función: removeFromCart(productName)**

Descripción: Elimina un producto del carrito.

- **Función: checkout()**

Descripción: Confirmar la compra.

- **Función: loadCart()**

Descripción: Carga el carrito desde localStorage al abrir la página y muestra los productos en una tabla de la página de ventas.

- **Función principal: validateAll()**

Descripción: Válida todos los campos relacionados con el pago (número de tarjeta, fecha de expiración y CVV).

- **Función: generateInvoice()**

Descripción: Genera una factura mostrando los detalles del cliente y del pago.

Eventos del DOM:

- **DOMContentLoaded:** Configura las validaciones en tiempo real y maneja el evento de envío del formulario de pago.

- **window.onload:** Llama a loadCart para cargar el carrito cuando la página termine de cargar.

Persistencia con localStorage:

- **localStorage.setItem:** Guarda el carrito en formato JSON.
- **localStorage.getItem:** Recupera el carrito almacenado.

Validaciones especificas:

- **validateCardNumber():**
    - Verifica que el número de tarjeta tenga 16 dígitos (usando una expresión regular).
    - Cambia el color del borde del campo según sea válido o no.

- **validateExpiryDate():**
    - Valida que la fecha de expiración esté en el formato MM/AA.
    - Comprueba que la fecha no sea anterior a la actual.

- **validateCVV():**
    - Verifica que el CVV sea un número de 3 o 4 dígitos.

Revisión y pruebas del proyecto

La etapa de revisión y pruebas es fundamental para garantizar que el proyecto funcione de manera adecuada y sin errores, ofreciendo una experiencia óptima al usuario. Para ello, se han definido tres áreas clave: revisión del código, pruebas en diferentes navegadores y análisis de feedback para realizar mejoras. A continuación, se describen los pasos realizados:

Revisión del código

Durante esta etapa, se llevó a cabo una inspección exhaustiva del código JavaScript desarrollado, evaluando tanto su funcionalidad como su estructura. Se verificaron las siguientes funciones principales:

- **addToCart(productName, productPrice):** Se comprobó que esta función añadiera productos correctamente al carrito y que actualizara las cantidades y subtotales cuando un producto ya existiera.
- **updateCart():** Se aseguró que la interfaz del carrito se actualizara dinámicamente, reflejando los cambios en tiempo real en la tabla de productos y en el cálculo del total.
- **validateAll():** Se validaron todos los campos del formulario de pago (número de tarjeta, fecha de expiración y CVV), comprobando que se aplicaran correctamente las validaciones específicas.

El código fue optimizado para mantener la legibilidad y facilitar futuros mantenimientos.

Pruebas en diferentes navegadores

La funcionalidad del proyecto fue probada en los navegadores más comunes (Chrome, Firefox, Edge y Safari) para garantizar la compatibilidad y una experiencia uniforme para los usuarios. Durante esta fase, se evaluaron los siguientes aspectos:

- Eventos del DOM, como DOMContentLoaded y window.onload, asegurando que las funciones como loadCart se ejecutaran correctamente al cargar la página.

- La persistencia de datos en el carrito mediante el uso de localStorage, verificando que los productos se guardaran y cargaran sin problemas en todos los navegadores.
- La validación de formularios en tiempo real, comprobando que el borde de los campos cambiara de color dependiendo de la validez del dato ingresado.

Feedback y mejoras

Finalmente, se realizó una fase de recopilación de feedback, tanto de usuarios finales como de desarrolladores. Este proceso permitió identificar áreas de mejora, como:

- Ajustes en la interfaz del carrito para hacerla más intuitiva y clara. Por ejemplo, el botón de eliminación del carrito asociado a removeFromCart(productName) se rediseñó para mayor visibilidad.
- Refinamiento de las validaciones, incluyendo mejoras en las expresiones regulares de funciones como validateCardNumber() y validateExpiryDate().
- Mejora de la experiencia de compra con la función generateInvoice(), optimizando el diseño de la factura para incluir más detalles sobre los productos y el cliente.

En esta fase también se documentaron todas las mejoras implementadas y se actualizaron los casos de prueba para futuras iteraciones del proyecto.

Anexos de documentación

- La documentación referente a todo el esquema html incluído en la página web (‘/documentacion pagina web.docx’).
- La documentación referente a todo el esquema CSS incluído en la página web (‘/documentacion Hoja de estilos css.docx’).
- La documentación referente a todo el esquema Javascript incluído en la página web (‘/definicion de las funciones de javascript.docx’).
- La presentación del proyecto. (‘/Entregable 2 - Presentación.pdf’)