UNIVERSIDAD PRIVADA DEL VALLE

INGENIERIA EN SISTEMASESTUDIANTES:

MANUAL TÉCNICO Y DE MANTENIMIENTO: SPORTAPZ

SIXTO ALEX APAZA FLORES

DOCENTE:

ING. HENRY MIRANDA ORDOÑEZ

MATERIA:

PROGRAMACIÓN WEB I

GESTION: 2025

La paz – Bolivia

1. Introducción y Propósito del Documento

Este manual está dirigido al personal técnico y a los desarrolladores responsables del mantenimiento, actualización y resolución de problemas de la plataforma web **SportApz**.
Aquí detallo la estructura de la interfaz de usuario, las tecnologías utilizadas y el funcionamiento interno del carrito de compras, con el objetivo de facilitar futuras modificaciones al sistema.

2. Estructura del Proyecto y Tecnologías Utilizadas

La plataforma está construida con tecnologías web estándar (HTML, CSS y JavaScript nativo), lo que permite mantener el proyecto de forma sencilla sin depender de frameworks externos.

2.1 Tecnologías Empleadas

- **HTML5:** Manejo semántico de secciones como productos, navegación y carrito.
- **CSS3 (con Flexbox):** Control del diseño horizontal de los productos y alineación del encabezado.
- **JavaScript (ES6+):** Implementación de la lógica del carrito de compras y persistencia de información en el navegador.

2.2 Clases CSS Más Importantes para el Layout

Clase	Propósito	Regla Clave	Efecto
.header-container	Contenedor principal del encabezado	display: flex; justify-content: space-between;	Separa el logo (izquierda) del título (derecha)
.productos	Contenedor de las tarjetas de producto	display: flex; flex-wrap: nowrap; overflow-x: auto;	Muestra los productos en una fila horizontal con scroll
.card	Tarjeta de producto individual	flex-shrink: 0; width: 280px;	Evita que la tarjeta se deforme y mantiene su tamaño fijo

3. Lógica del Carrito de Compras (JavaScript)

Toda la funcionalidad del carrito está construida en JavaScript puro, utilizando **localStorage** para mantener los productos aunque el usuario cierre la página.

3.1 Persistencia de Datos con localStorage

El carrito se almacena como un arreglo de objetos, por ejemplo:

[{ nombre: "Polera", precio: 120 }, ...]

Inicialización al cargar la página

let carrito = JSON.parse(localStorage.getItem("carrito")) || [];

Guardado después de cada modificación

localStorage.setItem("carrito", JSON.stringify(carrito));

Con esto garantizo que el carrito se mantenga actualizado entre sesiones.

3.2 Flujo de Añadir un Producto

(Acción del botón **.boton-comprar**)

1. Se obtienen los valores desde los atributos data-nombre y data-precio.
2. Se crea un objeto { nombre, precio } y se agrega al arreglo carrito.
3. Se ejecuta actualizarCarrito() para refrescar la interfaz.
4. Se guarda el nuevo estado en localStorage.

3.3 Flujo de Eliminación de un Producto

(Acción del botón **.boton-eliminar**)

1. El botón envía el índice correspondiente mediante data-indice.
2. La función elimina el producto de la lista usando:

carrito.splice(indice, 1);

3. Se actualiza el carrito en la interfaz y en localStorage.

4. Mantenimiento y Actualización del Contenido

4.1 Agregar Nuevos Productos

Para añadir un producto nuevo, basta con duplicar un bloque:

 ...

Luego modificar:

Elemento	Atributo	Descripción
<img>	src="nueva_ruta.png"	Imagen del producto
<h3>	Texto	Nombre del producto
<p>	Precio visible	Precio en bolivianos
<a class="boton-comprar">	data-nombre	Nombre que se guarda en el carrito
<a class="boton-comprar">	data-precio	Precio usado para cálculos

4.2 Solución de Problemas Comunes

• El carrito no guarda productos al recargar

- Revisar si localStorage está habilitado.
- Verificar en la consola que localStorage.setItem() y getItem() se ejecutan sin errores.

• El botón “Inicio” no funciona

- Confirmar que el enlace sea exactamente: href="index1.html"
- Verificar que el archivo *index1.html* exista en la carpeta principal.