



## **Ciclo 1 Universidad Don Bosco**

### **Materia:**

Diseño y Programación de Software Multiplataforma G01T

### **Nombre del trabajo:**

Proyecto de Carrito de Compras en JavaScript con Facturación

### **Estudiante:**

Guillermo Salvador Cartagena Mejia CM213015

Alan Anderson Vasquez Leiva VL221407

Kelly Abigail Vásquez Rodríguez VR220916

Julio César Posada Ramírez PR222058

### **Docente:**

Ing. Alexander Alberto Sigüenza Campos

**Universidad Don Bosco, 23 de Febrero de 2024**

**Programa:** <https://alanleiva30.github.io/Carrito/>

**Código de GIT:** <https://github.com/AlanLeiva30/Carrito.git>

# Índice

<b>Introducción</b>	<b>3</b>
<b>Estructura de código</b>	<b>4</b>
<b>Carrito de compras</b>	<b>6</b>
<b>conclusión</b>	<b>9</b>

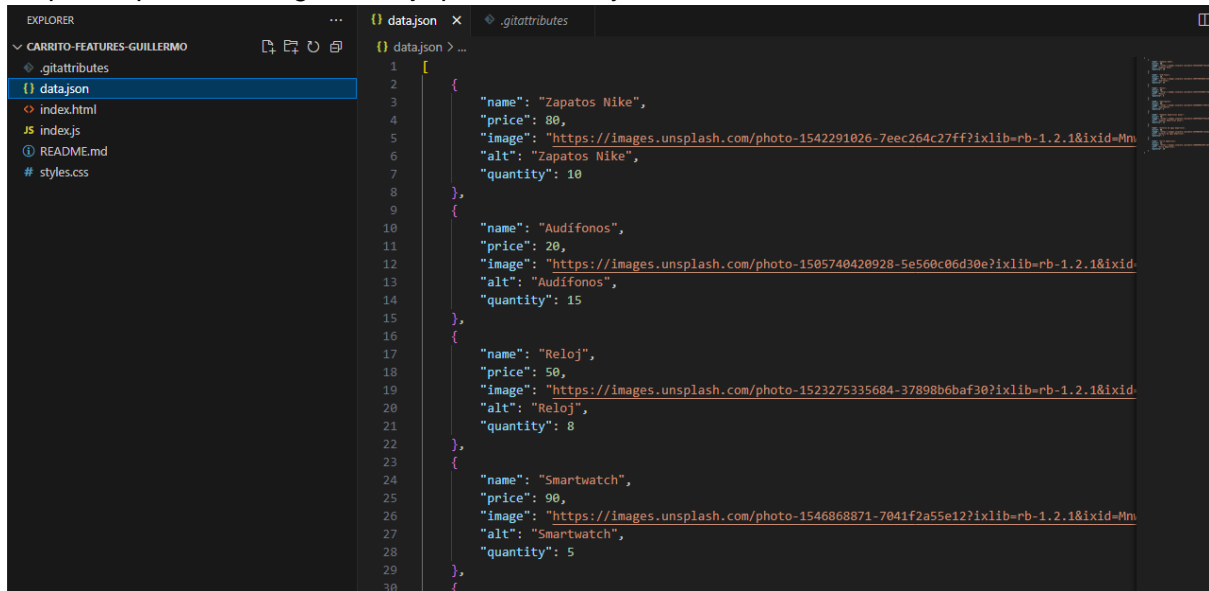
# Introducción

El proyecto tiene como objetivo el desarrollo de un carrito de compras dinámico, utilizando JavaScript como lenguaje principal. A lo largo del proyecto se emplearán conceptos fundamentales del DOM ya que este nos facilitara una representación estructurada del documento y define de que maneras el programa pueda acceder, modificar, tanto su estructura estilo y contenido y la implementación de funciones, lo que permitirá crear una experiencia interactiva y amigable para el usuario. La tienda virtual se enfocará en la venta de productos como zapatos, relojes y ropa, buscando ofrecer una plataforma fácil de usar y funcional.

# Estructura de código

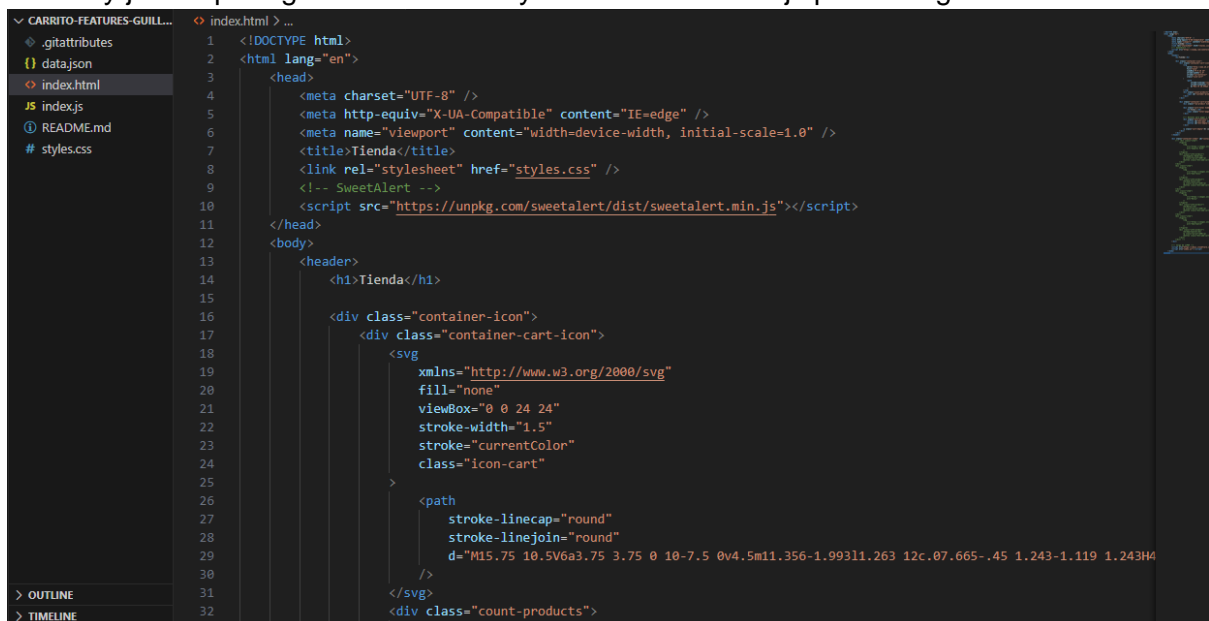
data.json

esta parte del código es el que nos permite almacenar la lista de productos en formato JSON, lo que da paso al código index.js pueda leerlo y mostrarlo en nuestra tienda.



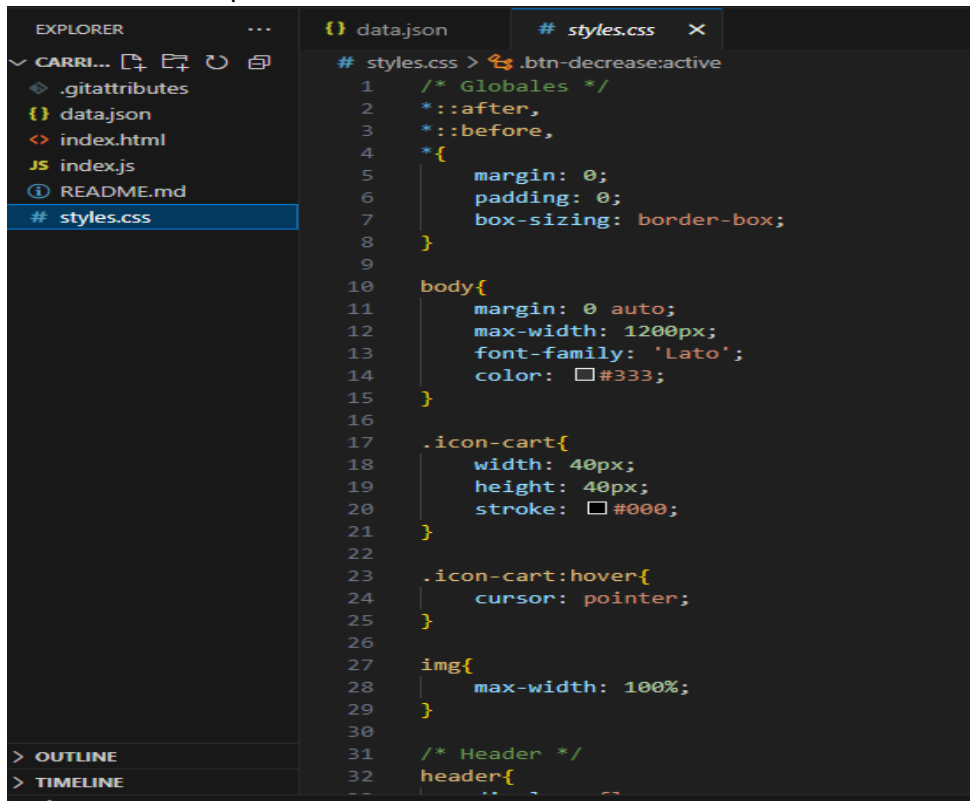
```
1  [
2
3  {
4    "name": "Zapatos Nike",
5    "price": 80,
6    "image": "https://images.unsplash.com/photo-1542291026-7eec264c27ff?ixlib=rb-1.2.1&ixid=Mm",
7    "alt": "Zapatos Nike",
8    "quantity": 10
9  },
10 {
11   "name": "Audifonos",
12   "price": 20,
13   "image": "https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-1.2.1&ixid-",
14   "alt": "Audifonos",
15   "quantity": 15
16 },
17 {
18   "name": "Reloj",
19   "price": 50,
20   "image": "https://images.unsplash.com/photo-1523275335684-37898b6baf30?ixlib=rb-1.2.1&ixid-",
21   "alt": "Reloj",
22   "quantity": 8
23 },
24 {
25   "name": "Smartwatch",
26   "price": 90,
27   "image": "https://images.unsplash.com/photo-1546868871-7041f2a55e12?ixlib=rb-1.2.1&ixid=Mm",
28   "alt": "Smartwatch",
29   "quantity": 5
30 },
31 ]
```

luego tenemos estructura de la tienda index.html el cual nos sirve para definir la interfaz de usuario de nuestra tienda y del carrito de compras, carga la libreria SweetAlert para mostrar alertas y jsPDF para generar la factura y se enlaza al index.js para la logica



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7    <title>Tienda</title>
8    <link rel="stylesheet" href="styles.css" />
9    <!-- SweetAlert -->
10   <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
11 </head>
12 <body>
13   <header>
14     <h1>Tienda</h1>
15
16     <div class="container-icon">
17       <div class="container-cart-icon">
18         <svg
19           xmlns="http://www.w3.org/2000/svg"
20           fill="none"
21           viewBox="0 0 24 24"
22           stroke-width="1.5"
23           stroke="currentColor"
24           class="icon-cart">
25
26         <path
27           stroke-linecap="round"
28           stroke-join="round"
29           d="M15.75 10.5V6a3.75 3.75 0 0 1 7.5 0V4.5a1.356 1.356 0 0 1 1.993 1.263 12c.07 6.65-.45 1.243-1.119 1.243H4"
30         />
31       </svg>
32       <div class="count-products">
```

por otro lado tenemos Styles.css el cual se encarga de los diseños y estilos de la tienda define la apariencia de la tienda por ejemplo la interacción en los botones, animacion al pasar el mouse sobre los productos etc.



```
# styles.css > .btn-decrease:active
1  /* Globales */
2  *::after,
3  *::before,
4  *{
5      margin: 0;
6      padding: 0;
7      box-sizing: border-box;
8  }
9
10 body{
11     margin: 0 auto;
12     max-width: 1200px;
13     font-family: 'Lato';
14     color: #333;
15 }
16
17 .icon-cart{
18     width: 40px;
19     height: 40px;
20     stroke: #000;
21 }
22
23 .icon-cart:hover{
24     cursor: pointer;
25 }
26
27 img{
28     max-width: 100%;
29 }
30
31 /* Header */
32 header{
```

el archivo index.js es el que maneja prácticamente toda la interacción de nuestra tienda en otras palabras maneja la interacción del usuario, carga los productos el cual los obtiene de data.json también se encarga de verificar las cantidades disponibles antes de agregar un producto, actualiza el carrito, elimina productos y realiza la acción de pago y generación de facturas en pdf

```
EXPLORER
...
CARRI...
  .gitattributes
  {} data.json
  {} index.html
  JS index.js
  README.md
  # styles.css

JS index.js
1  const btnCart = document.querySelector('.container-cart-icon');
2  const containerCartProducts = document.querySelector('.container-cart-products');
3  const container_items = document.querySelector('.container-items');
4
5  //llama al archivo data.json y renderiza los productos en el DOM
6  fetch('data.json')
7    .then(response => response.json())
8    .then(data => {
9      console.log(data);
10     data.forEach(product => {
11       const item = document.createElement('div');
12       item.classList.add('item');
13
14       item.innerHTML = `
15         <figure>
16           
17         </figure>
18         <div class="info-product">
19           <h2>${product.name}</h2>
20           <p class="price">${product.price}</p>
21           <p class="quantity">Cantidad disponible: <span class="available-quantity">${product.quantity}</span></p>
22           <button class="btn-add-cart">Añadir al carrito</button>
23         </div>
24       `;
25
26       container_items.appendChild(item);
27     });
28   });
29
30 btnCart.addEventListener('click', () => {
31   containerCartProducts.classList.toggle('hidden-cart');
32 });
```

# Carrito de compras

Tienda



Zapatos Nike

\$80

Cantidad disponible: 10

Añadir al carrito



Audifonos

\$20

Cantidad disponible: 15

Añadir al carrito



Reloj

\$50

Cantidad disponible: 8

Añadir al carrito



Smartwatch

\$90

Cantidad disponible: 5

Añadir al carrito



Zapatos deportivos mujer

\$70

Cantidad disponible: 12

Añadir al carrito



botella de agua deportiva

\$60

Cantidad disponible: 7

Añadir al carrito

seleccionamos añadir al carrito

**Zapatos Nike**

**\$85**

Cantidad disponible: 15

Añadir al carrito

nos aparecera la opcion de agregar otro smartawatch



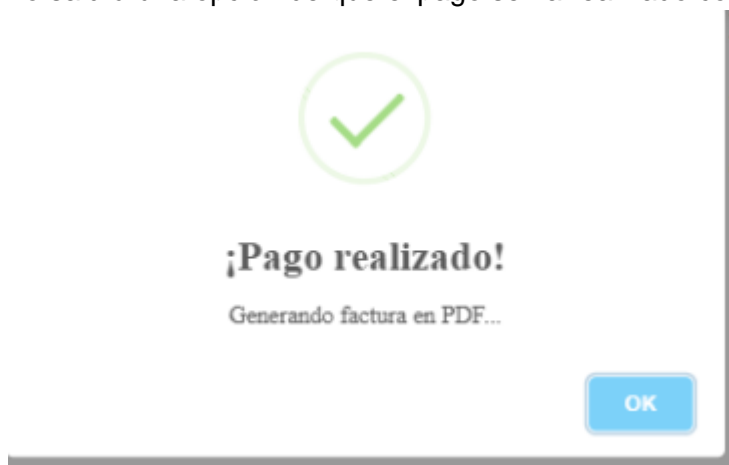
o también podemos agregar mas productos



podemos vaciar el carrito al no querer la compra ya.

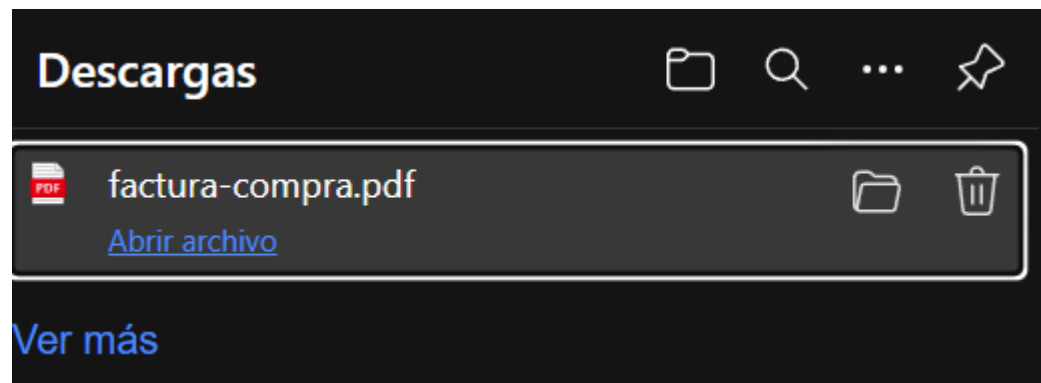
si seleccionamos pagar.

no saldrá una opción de que el pago se ha realizado con éxito.



y generará un PDF al seleccionar OK.





## Factura de Compra

1 x Smartwatch - \$90

1 x Zapatos Nike - \$85

Total a pagar: \$175

## conclusión

de este pequeño proyecto se puede verificar como el uso del DOM nos permite un uso más dinámico de una interfaz web, a través del uso de javascript se logra modificar y actualizar los elementos de HTML. lo cual hace que los cambios sean inmediatos, aparte que también en nuestro proyecto se integra la lectura del archivo para que los productos se puedan generar dinámica, lo cual nos facilita mucho a nosotros trabajar con la actualización del inventario sin necesidad de modificar el código fuente. por lo tanto aqui tenemos un buen ejemplo como poder mejorar la experiencia del usuario en las app web, sin depender de tecnologías avanzadas