

UNIVERSIDAD DON BOSCO
ESCUELA DE COMPUTACION
DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA



TITULO:

Guia #03

DOCENTE:

Ing. Emerson Francisco Cartagena Candelario

PRESENTADO POR:

Alvarenga Romero, Josías Isaac #AR200778,

Antiguo Cuscatlán, 23 de agosto de 2023

DESARROLLO

1. Creación de proyecto React, cambiando estilos y creando archivo data.js (Literales 1,2,3,4,5).

```
export const data = [
  {
    id:1,
    title:'Cien años de soledad',
    price:100,
    urlImage:'https://images.penguinrandomhouse.com/cover/9780525562443',
    quantity:1
  },
  {
    id:2,
    title:'El señor de los anillos(Trilogía)',
    price:190,
    urlImage:'https://proassetspdlcom.cdnstatics2.com/usuaris/libros/fotos/358/original/portada_1',
    quantity:1
  },
  {
    id:3,
    title:'Cuentos de Barro',
    price:30,
    urlImage:'https://www.librosdelaballena.com/wpcontent/uploads/2020/05/cuentos-barro-244x300.',
    quantity:1
  },
  {
    id:4,
    title:'Tierra de Infancia',
    price:30,
    urlImage:'https://marketsvstg.blob.core.windows.net/marketsv/0011705_tierra-de-infancia_510.',
    quantity:1
  },
]
```

2. Creando componentes y utilizándolos (Literales 6,7,8,9,10).

```
import React from "react";

export const Headers = () => {
  return (
    <header>
      <h1>Tienda de Libros</h1>
      <div className="container-icon">
        <div className="container-cart-icon">
          
          <div className="count-products"></div>
          <span id="contador-productos">0</span>
        </div>
      </div>
      <div className="container-cart-products hidden-cart">
        <div className="row-product hidden">
          <div className="cart-product">
            <div className="info-cart-product">
              <span className="cantidad-producto-carrito">1</span>
              <p className="titulo-producto-carrito">Don Quijote</p>
              <span className="precio-producto-carrito">$80</span>
            </div>
            
            </div>
          </div>
          <div className="cart-total hidden">
            <h3>Total:</h3>
            <span className="total-pagar">$200</span>
          </div>
          <p className="cart-empty">El carrito está vacío</p>
        </div>
      </div>
    </header>
  );
}
```

```

import React from "react";
import { data } from "../data";

export const ProductList = () => {

  return (
    <div className="container-items">
      {data.map(product => (
        <div className="item" key={product.id}>
          <figure>
            <img src={product.urlImage} alt={product.title}/>
          </figure>
          <div className="info-product">
            <h2>{product.title}</h2>
            <p className="price">${product.price}</p>
            <button className="btn-add-cart">Añadir al carrito</button>
          </div>
        </div>
      ))}
    </div>
  );
}

```

```

✓ import logo from './logo.svg';
  import './App.css';
  import { Headers } from './components/Headers';
  import { ProductList } from './components/ProductList';

✓ function App() {
✓   return (
✓     <>
      <Headers/>
      <ProductList/>
    </>
  );
}

export default App;

```

3. Modificación de Componentes (Literales 11,12,13,14).

Headers.jsx

```
export const Headers = ({
  allProducts,
  setAllProducts,
  total,
  countProducts,
  setCountProducts,
  setTotal,
}) => {

  const [active, setActive] = useState(false);
  const onDeleteProduct = product => {
    const results = allProducts.filter(
      item => item.id !== product.id
    );
    setTotal(total - product.price * product.quantity);
    setCountProducts(countProducts - product.quantity);
    setAllProducts(results);
  };
  const onCleanCart = () => {
    setAllProducts([]);
    setTotal(0);
    setCountProducts(0);
  };
};
```

```
return (
  <header>
    <h1>Tienda de Libros</h1>
    <div className="container-icon">
      <div className="container-cart-icon"
        onClick={() => setActive(!active)}
      >
        
        <div className="count-products">
          <span id="contador-productos">{countProducts}</span>
        </div>
      </div>
    </div>
    <div className={`container-cart-products ${
      active ? '' : 'hidden-cart'
    }`}
    >
      {allProducts.length ? (
        <>
          <div className="row-product">
            {allProducts.map(product => (
              <div className="cart-product"
                key={product.id}>
                <div className="info-cart-product">
                  <span className="cantidad-productocarrito">
                    {product.quantity}
                  </span>
                  <p className="titulo-productocarrito">
                    {product.title}
                  </p>
                  <span className="precio-productocarrito">
                    ${product.price}
                  </span>
                </div>
                <img
```

```

        </span>
      </div>
       onDeleteProduct(product)}
      />
    </div>
  )}
</div>
<div className='cart-total'>
  <h3>Total:</h3>
  <span className='total-pagar'>${total}</span>
</div>
<button className='btn-clear-all'
  onClick={onCleanCart}>
  Vaciar Carrito
</button>
</>
) : (
  <p className='cart-empty'>El carrito está vacío</p>
)
</div>
</header>
);

```

ProductList.jsx

```

export const ProductList = ([
  allProducts,
  setAllProducts,
  countProducts,
  setCountProducts,
  total,
  setTotal,
]) => {
  const onAddProduct = product => {
    if (allProducts.find(item => item.id === product.id)) {
      const products = allProducts.map(item =>
        item.id === product.id
        ? { ...item, quantity: item.quantity + 1 }
        : item
      );
      setTotal(total + product.price * product.quantity);
      setCountProducts(countProducts + product.quantity);
      return setAllProducts([...products]);
    }
    setTotal(total + product.price * product.quantity);
    setCountProducts(countProducts + product.quantity);
    setAllProducts([...allProducts, product]);
  };
};

```



```

return (
  <div className="container-items">
    {data.map(product =>(
      <div className="item" key={product.id}>
        <figure>
          <img src={product.urlImage} alt={product.title}/>
        </figure>
        <div className="info-product">
          <h2>{product.title}</h2>
          <p className="price">${product.price}</p>
          <button className="btn-add-cart"onClick={() => onAddProduct(product)}>
            Añadir al carrito
          </button>
        </div>
      </div>
    )
    )}
  </div>
);

```

App.js

```

function App() {

  const [allProducts, setAllProducts] = useState([]);
  const [total, setTotal] = useState(0);
  const [countProducts, setCountProducts] = useState(0);

  return (
    <>
      <Headers
        allProducts={allProducts}
        setAllProducts={setAllProducts}
        total={total}
        setTotal={setTotal}
        countProducts={countProducts}
        setCountProducts={setCountProducts}
      />
      <ProductList
        allProducts={allProducts}
        setAllProducts={setAllProducts}
        total={total}
        setTotal={setTotal}
        countProducts={countProducts}
        setCountProducts={setCountProducts}
      />
    </>
  );
}

export default App;

```

Resultados:

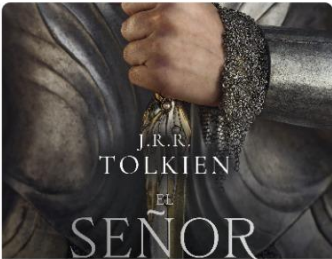
Tienda de Libros



Cien años de soledad

\$100

Añadir al carrito



El señor de los
anillos(Trilogía)

\$190

1 El señor de los anillos(Trilogía) \$190

1 Cien años de soledad \$100

1 Cuentos de Barro \$30

Total: \$320

Vaciar Carrito

Cuentos de Barro

\$30

Añadir al carrito

Tienda de Libros



Cien años de soledad

\$100

Añadir al carrito



El señor de los
anillos(Trilogía)

\$190

1 El señor de los anillos(Trilogía) \$190

1 Cien años de soledad \$100

Total: \$290

Vaciar Carrito

Cuentos de Barro

\$30

Añadir al carrito

El carrito está vacío



Discusión:

Modificaremos el ejercicio de la practica para que cuando se agreguen los productos también se pueda apreciar una imagen junto al nombre del producto.

```
<div className='info-cart-product'>
  <span className='img-producto'>
    <img src={product.urlImage} alt="image" />
  </span>

```

```
.img-producto img{
  width: 100px;
  height: 100px;
  font-size: 10px;
}
```

