

Práctica Guiada 3: Interactividad en la tienda de productos con Hooks

Objetivo: Ampliar la aplicación de productos informáticos integrando Hooks de React para manejar interactividad:

- Filtrar productos por nombre.
- Añadir un contador de productos en el carrito.
- Usar para mejorar la experiencia del usuario (ej. enfocar el input de búsqueda).

Paso 0: Requisitos previos

- Proyecto React creado y funcionando (Prácticas 1 y 2).
- Bootstrap ya instalado e importado.

Paso 1: Crear el componente de búsqueda de productos (SearchBar.js)

Ubicación: src/components/SearchBar.js

```
// src/components/SearchBar.js
import React from 'react';

export default function SearchBar({
  query,
  onChange,
  onClear,
  placeholder = 'Buscar productos por nombre, código o descripción...',
}) {
  const handleInput = (e) => onChange(e.target.value);
  const handleClear = () => onClear?.();

  return (
    <div className="container mt-3">
      <div className="row justify-content-center">
        <div className="col-12 col-md-10 col-lg-8">
          <div className="input-group">
            <span className="input-group-text" id="search-label"> 🔍 </span>
            <input
              type="text"
              value={query}
              onChange={handleInput}
              className="form-control"
            />
          </div>
        </div>
      </div>
    </div>
  );
}
```

```

placeholder={placeholder}
aria-label="Barra de búsqueda de productos"
aria-describedby="search-label"
/>
{query && (
  <button
    type="button"
    className="btn btn-outline-secondary"
    onClick={handleClear}
    title="Limpiar búsqueda">Limpiar</button>
  )}
</div>
<small className="text-muted d-block mt-1">
  Filtra en tiempo real. Ejemplos: <code>"SSD"</code>, <code>"P-
1001"</code>, <code>"Lenovo"</code>.
</small>
</div>
</div>
</div>
);
}


```

Paso 2: Crear componente de carrito con contador

Ubicación: src/components/Cart.js

```

//Src/components/Cart.js
import React from 'react';

export default function Cart({ items = [] }) {
  const total = items.length;
  return (
    <div className="cart d-flex align-items-center">
      <span className="me-2">  </span>
      <span>
        Carrito ({total} {total === 1 ? 'producto' : 'productos'})
      </span>
    </div>
  );
}

```

}Paso 3: Modificar (ProductsList.js) para aplicar el filtro

Ubicación: src/components/ProductsList.js

```
// src/components/ProductList.js
import React from 'react';

export default function ProductList({ products, onAdd }) {
  if (!products || products.length === 0) {
    return <p className="text-center text-muted mt-3">No hay productos disponibles.</p>;
  }

  return (
    <div className="container mt-4">
      <h2 className="mb-3">Listado de Productos Informáticos</h2>

      <table className="table table-striped table-bordered align-middle text-center">
        <thead className="table-dark">
          <tr>
            <th>Código</th>
            <th>Imagen</th>
            <th>Descripción</th>
            <th>Precio (€)</th>
            <th>Disponible</th>
            <th>Añadir</th>
          </tr>
        </thead>

        <tbody>
          {products.map((p) => (
            <tr key={p.id}>
              <td>{p.id}</td>

              <td>
                <img
                  src={p.imagen}
                  alt={p.nombre}
                  className="product-img img-thumbnail"
                  style={{ width: "80px" }}
                />
              </td>

              <td>{p.nombre}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}
```

```

    {/* Precio */}
    <td>{p.precio.toFixed(2)}</td>

    {/* Disponible */}
    <td>
      <span className={`badge ${p.enStock ? 'bg-success' : 'bg-danger'}`}>
        {p.enStock ? 'Sí' : 'No'}
      </span>
    </td>

    {/* Botón Añadir */}
    <td>
      <button
        className="btn btn-primary btn-sm"
        disabled={!p.enStock}
        onClick={() => onAdd?.(p)}
      >
        Añadir
      </button>
    </td>
  </tr>
)}
</tbody>
</table>
</div>
);
}

```

Paso 4: Integrar todo en App.js

Ubicación: src/App.js

```

// src/App.js
import React, { useState } from "react";

// Componentes
import Header from "./components/Header";
import Footer from "./components/Footer";
import ProductList from "./components/ProductList";
import SearchBar from "./components/SearchBar";
import Cart from "./components/Cart";

```

```

// Datos
import { catalog } from "../data/catalog";

export default function App() {

  // Estado de búsqueda
  const [query, setQuery] = useState("");

  // Estado del carrito
  const [cartItems, setCartItems] = useState([]);

  // Añadir producto al carrito
  const addToCart = (product) => {
    setCartItems([...cartItems, product]);
  };

  // Filtrado en tiempo real
  const filteredProducts = catalog.filter((p) => {
    const text = query.toLowerCase();
    return (
      p.nombre.toLowerCase().includes(text) ||
      p.id.toLowerCase().includes(text) ||
      String(p.precio).includes(text)
    );
  });

  return (
    <>
      { /* Cabecera */ }
      <Header />

      { /* Carrito */ }
      <div className="container mt-3">
        <Cart items={cartItems} />
      </div>

      { /* Barra de búsqueda */ }
      <SearchBar
        query={query}
        onChange={setQuery}

```

```

        onClear={() => setQuery("")}
      />

      { /* Listado filtrado + botón Añadir */}
      <ProductList
        products={filteredProducts}
        onAdd={addToCart}
      />

      { /* Pie de página */}
      <Footer />
    </>
  );
}

```

Paso 5: Ejecutar y verificar

Entregable

- Capturas de pantalla mostrando:
 - Filtro funcionando.
 - Carrito incrementando productos.
 - Input enfocado con useRef.
- Código de los nuevos componentes.