



## Reto Extra — Mini Catálogo de Productos

**Objetivo:** practicar **componentes reutilizables**, **props**, **eventos**, **renderizado de listas** y **organización**.

### ♦ Instrucciones

1. Crea un componente **ProductCard** en `src/components/Products/ProductCard/`.
  2. Crea un componente **ProductButton** en `src/components/Products/ProductButton/`.
  3. La `ProductCard` debe mostrar:
    - **Imagen** (prop)
    - **Nombre** (prop)
    - **Precio** (prop)
    - **Texto condicional:** si `precio > 10`, mostrar "Producto premium"
    - **Botón** "Añadir al carrito" → **es el componente ProductButton**
      - Al hacer clic, `console.log("Añadiste <nombre> al carrito")`
      - El botón acepta una **prop color** para el fondo
  4. En `App.jsx`, renderiza **3 ProductCard** con datos distintos.
- 
5. **Nivel 3 (opcional - NO OBLIGATORIO):** guarda los productos en un **array** y píntalos con `map()` (usa `key={id}`).

### ♦ Pistas

- `props.color` → estilo del botón (inline o clase CSS).
- Condicional:

```
{precio > 10 && <p className="premium">Producto premium</p>}
```

- Renderizado de lista con `map()` y key única:

```
{productos.map(p => <ProductCard key={p.id} ... />)}
```

---

### 🧩 Estructura sugerida

src/

└─ components/

    └─ Cards/

        └─ ProductCard/

            └─ ProductCard.jsx

            └─ ProductCard.css

└─ Buttons/

    └─ ProductButton/

        └─ ProductButton.jsx

        └─ ProductButton.css

---

## Ejemplo (base de referencia)

Trata de investigar por tí mismo cada paso previamente y sino aquí tienes un ejemplo de referencia que puede facilitarte.

### ProductButton.jsx

```
import "../ProductButton.css";

const ProductButton = ({ color = "lightblue", onClick, children }) => {
  return (
    <button style={{ backgroundColor: color }} onClick={onClick}>
      {children}
    </button>
  );
};

export default ProductButton;
```

### ProductCard.jsx

```
import "../ProductCard.css";
import ProductButton from "../ProductButton/ProductButton";

const ProductCard = ({ nombre, precio, imagen, colorBoton }) => {
  const handleAdd = () => console.log(`Añadiste ${nombre} al carrito`);

  return (
    <div className="product-card">
      <img src={imagen} alt={nombre} />
      <h3>{nombre}</h3>
      <p>{precio} €</p>

      {precio > 10 && <p className="premium">Producto premium</p>}

      <ProductButton color={colorBoton} onClick={handleAdd}>
        Añadir al carrito
      </ProductButton>
    </div>
  );
};
```

```

    </div>
  );
};

export default ProductCard;

```

### App.jsx (con array + map + key)

```

import ProductCard from
"./components/Products/ProductCard/ProductCard";

function App() {
  const productos = [
    { id: 1, nombre: "Manzana", precio: 2, imagen:
"https://via.placeholder.com/150", colorBoton: "green" },
    { id: 2, nombre: "Café", precio: 12, imagen:
"https://via.placeholder.com/150", colorBoton: "saddlebrown" },
    { id: 3, nombre: "Chocolate", precio: 15, imagen:
"https://via.placeholder.com/150", colorBoton: "orange" },
  ];

  return (
    <div>
      <h1>Catálogo</h1>
      <h2>Total: {productos.length}</h2>

      {productos.map(({ id, nombre, precio, imagen, colorBoton })
=> (
        <ProductCard
          key={id}
          nombre={nombre}
          precio={precio}
          imagen={imagen}
          colorBoton={colorBoton}
        />
      ))}
    </div>
  );
}

```

```
export default App;
```

### **OBJETIVOS:**

- Crear componentes reutilizables.
- Pasar props dinámicas (nombre, precio, color, imagen).
- Practicar eventos (onClick).
- Introducir renderizado condicional (precio > 10).
- Renderizar una lista con map() y key.
- Mantener estructura limpia de carpetas.