# TRABAJO DE ALMACENAMIENTO DE PRODUCTOS

**CODIFICACIÓN:**

```javascript
JS index.js M ✕
src > pages > JS index.js > ⊗ ProductManagement
 1   import React, { useState } from 'react';
 2
 3   export default function ProductManagement() {
 4     const [products, setProducts] = useState([]);
 5     const [newProduct, setNewProduct] = useState({
 6       id: 1,
 7       name: '',
 8       description: '',
 9       price: '',
10       image: null,
11     });
12     const [editingIndex, setEditingIndex] = useState(null);
13
14     const addProduct = () => {
15       if (editingIndex !== null) {
16         const updatedProducts = [...products];
17         updatedProducts[editingIndex] = newProduct;
18         setProducts(updatedProducts);
19         setEditingIndex(null);
20       } else {
21         setProducts([...products, { ...newProduct, id: products.length + 1 }]);
22       }
23       setNewProduct({ id: newProduct.id + 1, name: '', description: '', price: '', image: null });
24     };
25
26     const updateProduct = (index, product) => {
27       setNewProduct(product);
28       setEditingIndex(index);
```

```javascript
31     const deleteProduct = (index) => {
32       const updatedProducts = products.filter((_, i) => i !== index);
33       setProducts(updatedProducts.map((product, index) => ({ ...product, id: index + 1 })));
34       setEditingIndex(null);
35     };
36
37     const handleInputChange = (e) => {
38       const { name, value } = e.target;
39
40       if (name === 'price') {
41         const nonNegativePrice = Math.max(0, parseFloat(value));
42         setNewProduct({ ...newProduct, [name]: nonNegativePrice });
43       } else {
44         setNewProduct({ ...newProduct, [name]: value });
45       }
46     };
47
48     const handleImageChange = (e) => {
49       setNewProduct({ ...newProduct, image: e.target.files[0] });
50     };
51
52     return (
53       <div>
54         <h1>ALMACENAMIENTO DE PRODUCTOS</h1>
```

```jsx
        <input
          type="text"
          name="name"
          placeholder="Nombre del producto"
          value={newProduct.name}
          onChange={handleInputChange}
        />
        <input
          type="text"
          name="description"
          placeholder="Descripción"
          value={newProduct.description}
          onChange={handleInputChange}
        />
        <input
          type="number"
          name="price"
          placeholder="Precio en soles"
          value={newProduct.price}
          onChange={handleInputChange}
        />
        <input type="file" name="image" onChange={handleImageChange} />
        <button onClick={addProduct}>
          {editingIndex !== null ? 'Guardar Cambios' : 'Agregar Producto'}
        </button>

        <table>
              <th>Descripción</th>
              <th>Precio S/.</th>
              <th>Imagen</th>
              <th>Modificación</th>
            </tr>
          </thead>
          <tbody>
            {products.map((product, index) => (
              <tr key={index}>
                <td>{product.id}</td>
                <td>{product.name}</td>
                <td>{product.description}</td>
                <td>{product.price}</td>
                <td>
                  {product.image && <img src={URL.createObjectURL(product.image)} alt="Product" />}
                </td>
                <td>
                  <button onClick={() => updateProduct(index, product)}>Editar</button>
                  <button onClick={() => deleteProduct(index)}>Eliminar</button>
                </td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
  );
}
```

# RESULTADOS:



## ALMACENAMIENTO DE PRODUCTOS

Nombre del producto

Descripción

Precio en soles

Seleccionar archivo | laptop.jpg

Agregar Producto

| ID | Producto | Descripción | Precio S/. | Imagen | Modificación |
|----|----------|-------------|-----------|--------|--------------|
| 1 | Carro | De juguete | 350 |  | Editar  Eliminar |
| 2 | Laptop | Gamer de 9na Generación | 4500 |  | Editar  Eliminar |