

FUNDACION UNIVERDITARIA DE LA RIOJA

Faculta de Ingeniería

Especialización de Ingeniería de Software

Desarrollo de Aplicaciones Web

Jose Alexis Rozo Bahamon

14136457

Profesor: Javier Díaz Díaz

17/06/2024

Ibagué – Tolima

1. Introducción

En este proyecto, se desarrolló una aplicación web CRUD (Crear, Leer, Actualizar, Eliminar) utilizando la biblioteca de JavaScript React y una API REST. La aplicación está diseñada para gestionar un inventario de mobiliarios, permitiendo a los usuarios realizar operaciones básicas de mantenimiento de datos, tales como la creación de nuevos mobiliarios, la visualización de la lista de mobiliarios existentes, la edición de información de mobiliarios y la eliminación de mobiliarios del sistema.

2. Objetivos del proyecto

- a. Desarrollar una interfaz de usuario intuitiva y funcional utilizando React:

Implementar componentes reutilizables para la gestión de mobiliarios.

Utilizar hooks de React para gestionar el estado y los efectos secundarios.

- b. Implementar un sistema CRUD completo:

Crear mobiliarios nuevos y almacenarlos en la base de datos.

Leer y mostrar la lista de mobiliarios almacenados.

Actualizar la información de los mobiliarios existentes.

Eliminar mobiliarios del sistema.

- c. Integrar la aplicación frontend con una API REST:

Realizar peticiones HTTP para interactuar con los datos del backend.

Manejar respuestas y errores de manera adecuada.

3. Alcance del proyecto

El alcance del proyecto incluye el desarrollo de las siguientes funcionalidades:

- **Listar Mobiliarios:** Una tabla que muestra todos los mobiliarios con sus respectivos detalles.
- **Crear Mobiliario:** Un formulario para agregar nuevos mobiliarios.
- **Editar Mobiliario:** Un formulario para actualizar la información de un mobiliario existente.
- **Eliminar Mobiliario:** Una función para eliminar mobiliarios de la base de datos.

Además, se incluyen características adicionales como:

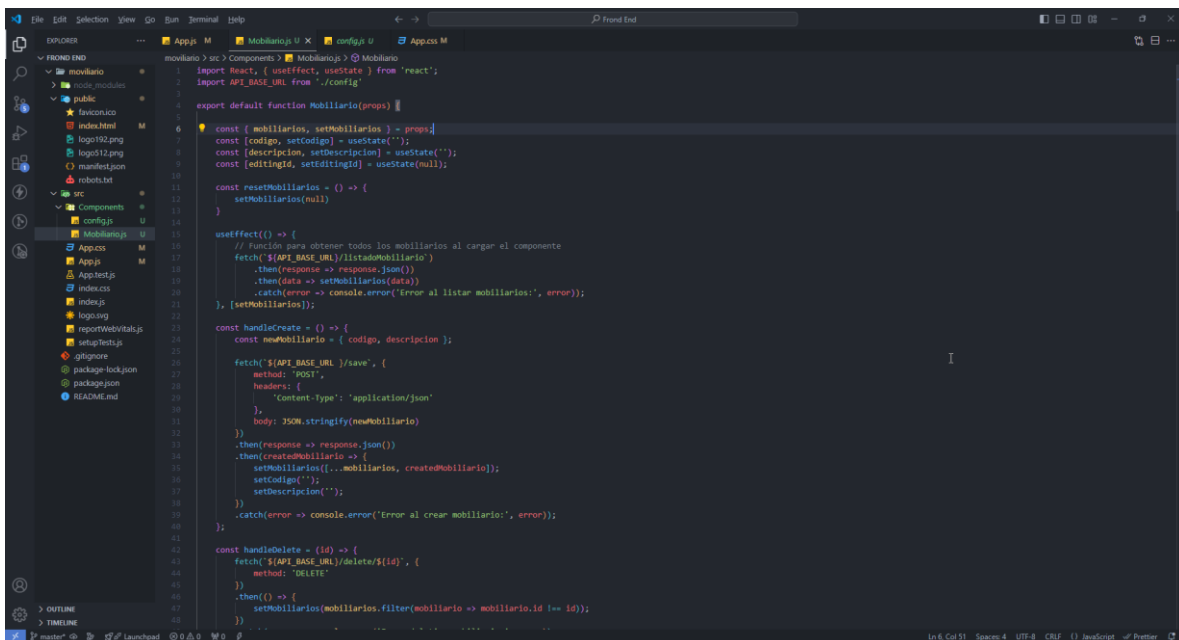
- **Validación de formularios:** Asegurando que los datos introducidos por el usuario sean correctos antes de enviarlos a la API.
- **Manejo de errores:** Notificación al usuario en caso de que ocurran errores durante las operaciones CRUD.

4. Tecnologías utilizadas:

- React: Biblioteca de JavaScript para la construcción de interfaces de usuario.
- Bootstrap: Framework de CSS para estilizar la aplicación y hacerla responsiva.
- Fetch API: Para realizar peticiones HTTP a la API REST.
- API REST: Backend que proporciona los endpoints para realizar las operaciones CRUD.

5. Operación:

Creamos el proyecto de mobiliario en React para hacer el consumo de la API REST antes trabajada con Java.



```
import React, { useState, useEffect } from 'react';
import API_BASE_URL from './config';

export default function Mobiliario(props) {
  const [mobiliarios, setMobiliarios] = useState([]);
  const [codigo, setCodigo] = useState('');
  const [descripcion, setDescripcion] = useState('');
  const [editingId, setEditingId] = useState(null);

  const resetMobiliarios = () => {
    setMobiliarios(null);
  }

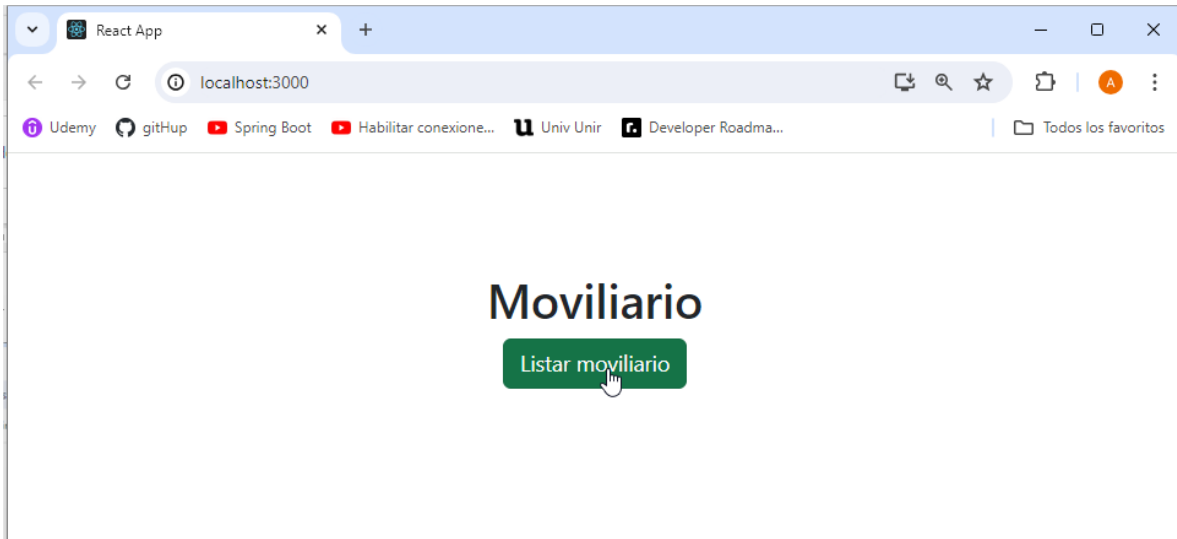
  useEffect(() => {
    // Función para obtener todos los mobiliarios al cargar el componente
    fetch(`${API_BASE_URL}/listadoMobiliario`)
      .then(response => response.json())
      .then(data => setMobiliarios(data))
      .catch(error => console.error('Error al listar mobiliarios:', error));
  }, [setMobiliarios]);

  const handleCreate = () => {
    const nuevoMobiliario = { codigo, descripcion };

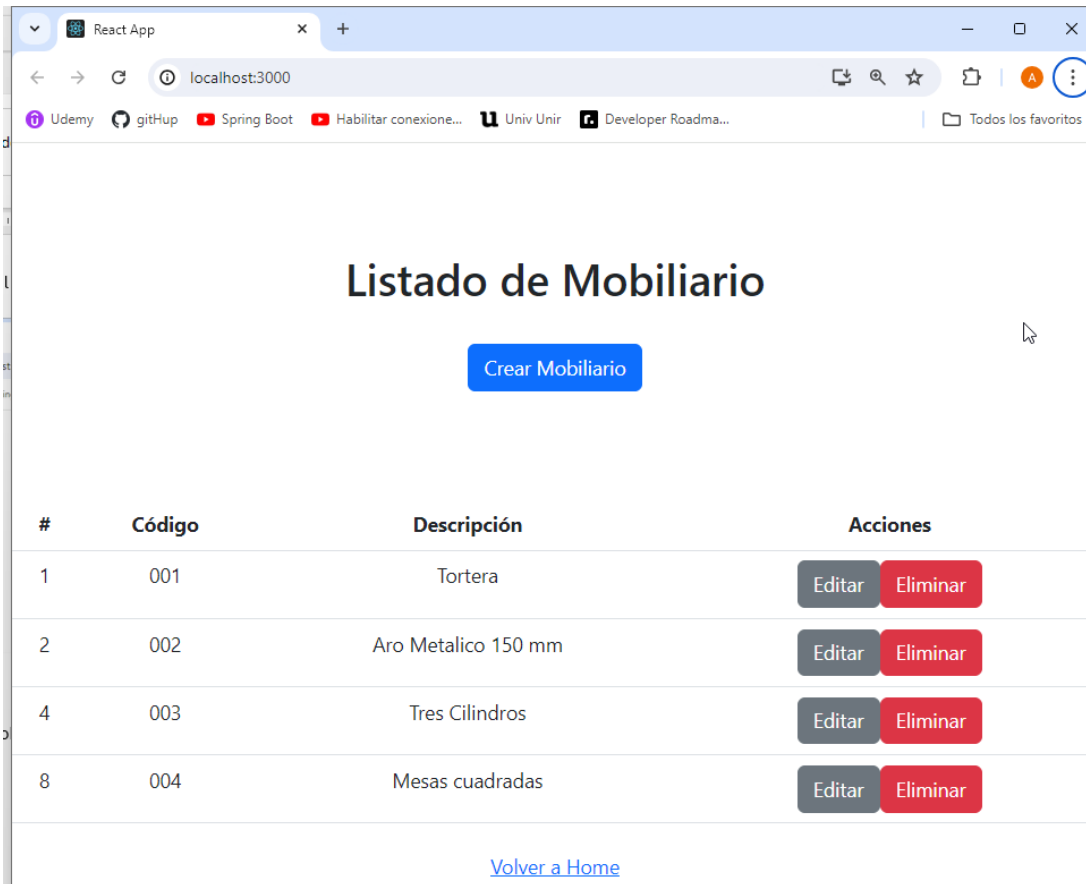
    fetch(`${API_BASE_URL}/save`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(nuevoMobiliario)
    })
      .then(response => response.json())
      .then(createMobiliario => {
        setMobiliarios([...mobiliarios, createMobiliario]);
        setCodigo('');
        setDescripcion('');
      })
      .catch(error => console.error('Error al crear mobiliario:', error));
  };

  const handleDelete = (id) => {
    fetch(`${API_BASE_URL}/delete/${id}`, {
      method: 'DELETE'
    })
      .then(() => {
        setMobiliarios(mobiliarios.filter(mobiliario => mobiliario.id !== id));
      })
  };
}
```

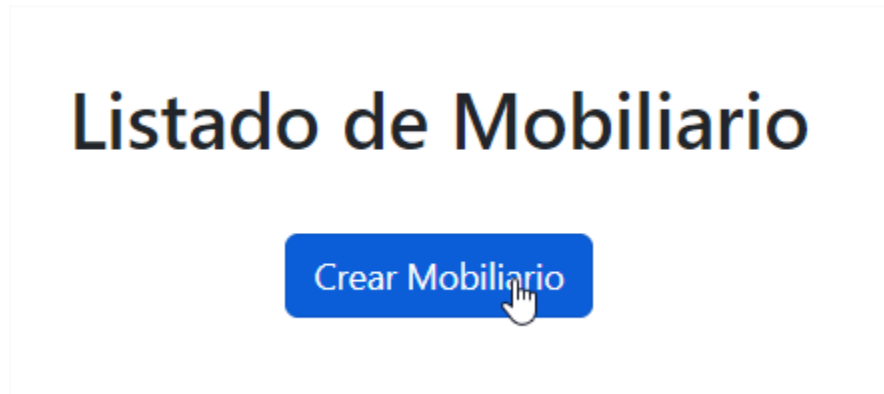
- Pagina Inicial



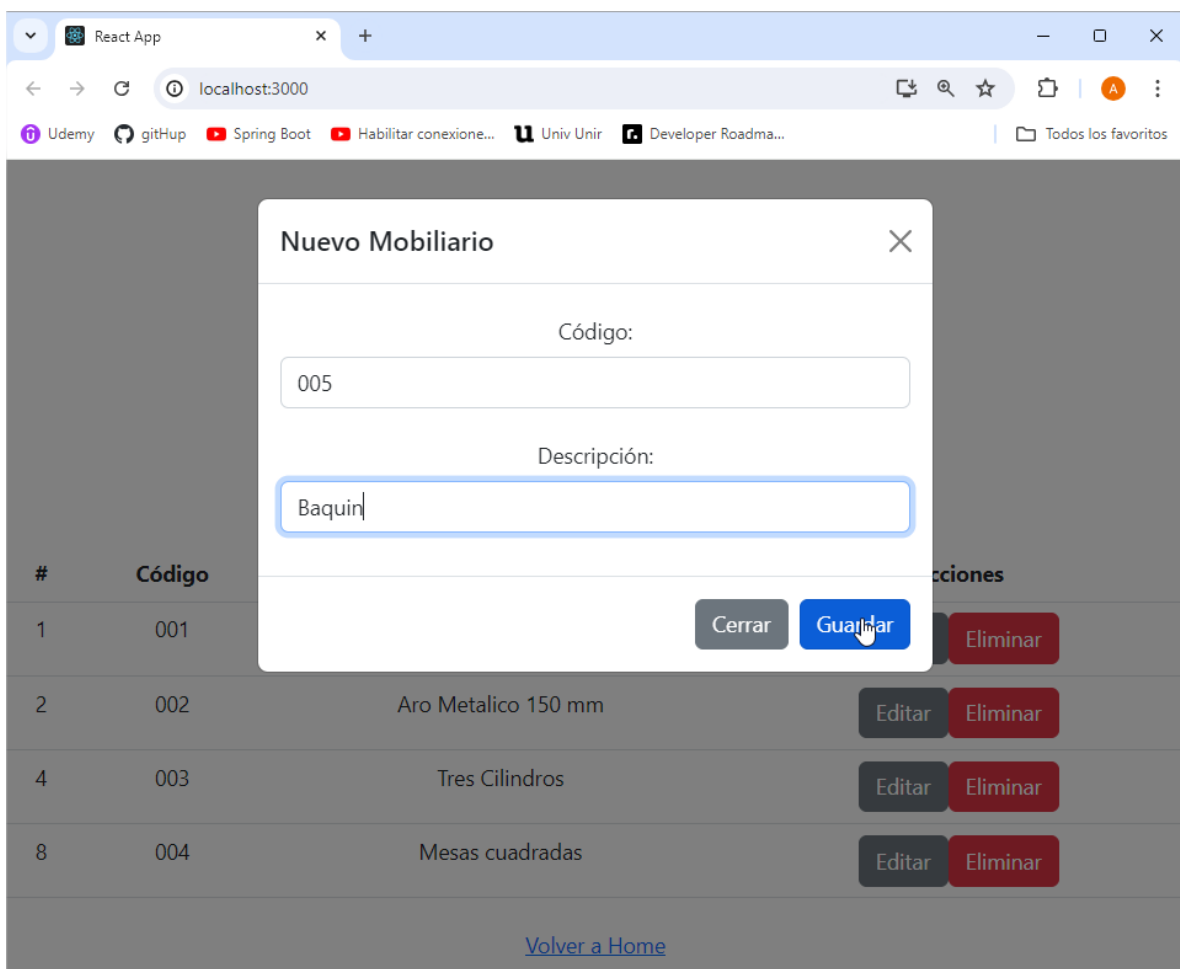
- Página de Mobiliario



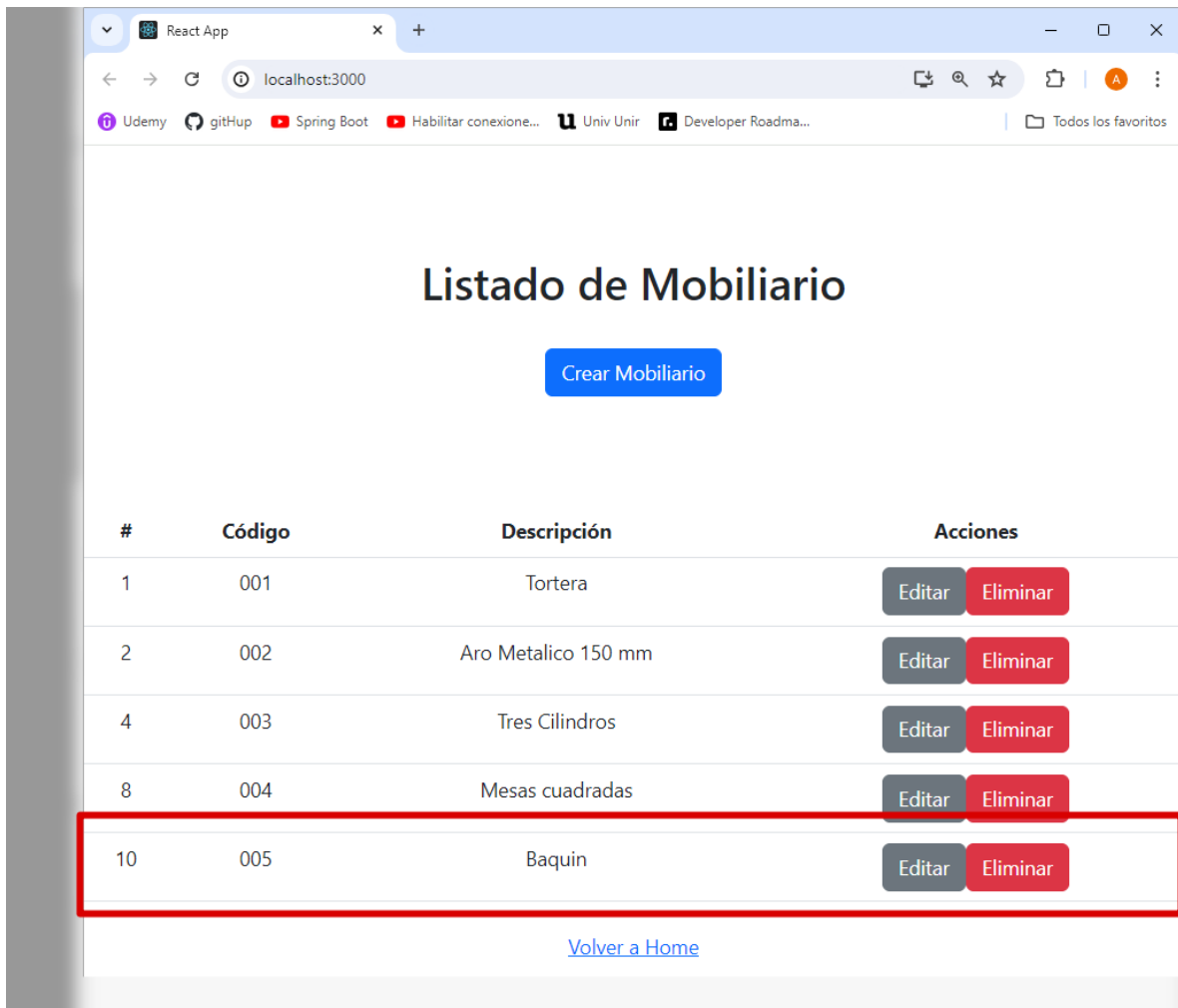
- Creamos vemos que esta listando los mobiliarios para fiestas, ahora creamos un nuevo mobiliario



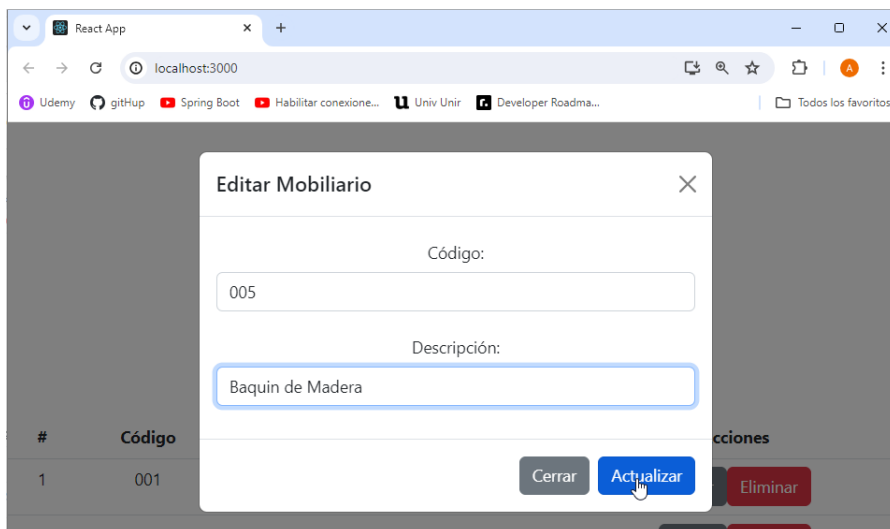
- Llenamos el formulario y guardamos



- Vemos que se guardó correctamente:



- Ahora damos clic para Editar y le agregamos el nombre de Baquin de Madera



- Verificamos en la grilla que se actualice el nombre.

React App

localhost:3000

Listado de Mobiliario

Crear Mobiliario

#	Código	Descripción	Acciones
1	001	Tortera	Editar Eliminar
2	002	Aro Metalico 150 mm	Editar Eliminar
4	003	Tres Cilindros	Editar Eliminar
8	004	Mesas cuadradas	Editar Eliminar
10	005	Baquin de Madera	Editar Eliminar

[Volver a Home](#)

- Por ultimo procederemos a Eliminar.

10	005	Baquin de Madera	Editar Eliminar
----	-----	------------------	-----------------

[Volver a Home](#)

#	Código	Descripción	Acciones
1	001	Tortera	Editar Eliminar
2	002	Aro Metalico 150 mm	Editar Eliminar
4	003	Tres Cilindros	Editar Eliminar
8	004	Mesas cuadradas	Editar Eliminar
Volver a Home			

6. Conclusión

Este proyecto demuestra cómo combinar tecnologías modernas para desarrollar una aplicación web funcional y mantenible. Utilizando React para el frontend y una API REST para el backend, se ha construido una aplicación que permite gestionar eficientemente un inventario de mobiliarios, proporcionando una base sólida para futuros desarrollos y mejoras.