MO 6 Desenvolupament Web Client

UF2 - Projecte React

Mohamed Bouirig Ait Bourig
27-05-2024

INDICE

1 - Objetivos:

(1 punt) La informació de les pel·lícules s'ha d'agafar d'un fitxer local i emmagatzemar-se en un context. I ser accessible des de tots els components de l'aplicació.

• (1 punt) S'ha d'utilitzar un Router per a definir les rutes de la nostra aplicació.

• (1 punt) Ha d'haver-hi una barra de navegació que tingui almenys enllaços a la pàgina principal, mostrar pel·lícules, inserir una pel·lícula i nosaltres.

• (2 punts) La pàgina principal (Catalog) ha de mostrar totes les pel·lícules en format targeta (PeliculaCard) i en cada targeta de pel·lícula hi haurà un botó per veure la informació detallada de cada pel·lícula.

• (2 punts) En la vista individual de cada pel·lícula (PeliculaDetall) ha d'haver-hi la informació completa de la pel·lícula i botons per llogar (i retornar), modificar i eliminar cada pel·lícula.

• (2 punts) Per a inserir i modificar les pel·lícules s'ha de fer servir el Hook useForm per crear els formularis.

• L'estructura mínima de components ha de ser: ◦ Calatog: Component que obtindrà l'array de pel·lícules del context i les mostrarà en format targeta. ◦ PeliculaCard: Component que mostrarà la informació bàsica d'una pel·lícula amb un botó per mostrar la informació detallada de cada pel·lícula. ◦ PeliculaDetall: Component per mostrar tota la informació completa de cada pel·lícula amb els botons per llogar (i retornar), modificar i eliminar la pel·lícula en concret.

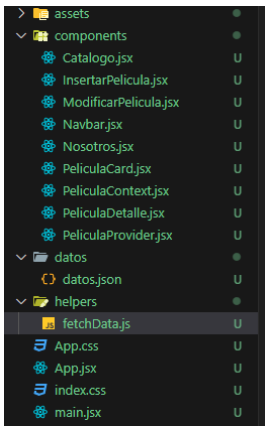
• (1 punt) Memòria.

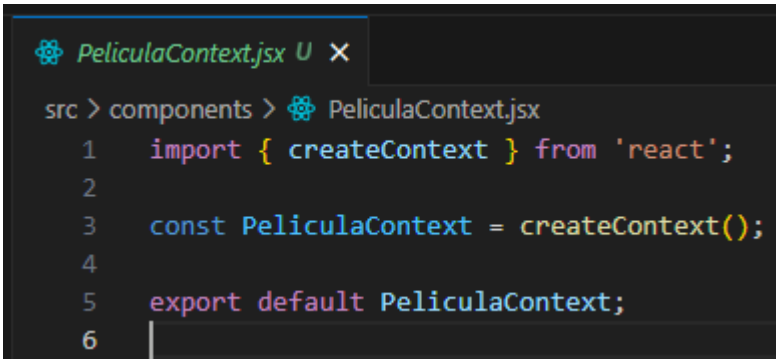
2 - Estructura del proyecto:

El proyecto está estructurado en varios componentes React, cada uno encargado de una funcionalidad específica.

Además, se utiliza un contexto (PeliculaContext) para compartir el estado global de las películas entre los componentes.

El archivo fetchData.js contiene funciones para obtener y procesar los datos de las películas.





```
fetchData.js U
src > helpers > fetchData.js
1  import data from "../datos/datos.json";
2
3  export const rebreInformacio = () => {
4      return new Promise((resolve) => {
5          setTimeout(() => resolve(data), 1500);
6      });
7  };
8
9  export const obtenerPeliculas = async () => {
10     try {
11         const response = await fetch(data);
12         if (!response.ok) {
13             throw new Error('No se pudo obtener la lista de peliculas');
14         }
15         const data = await response.json();
16         return data;
17     } catch (error) {
18         console.error('Error al obtener peliculas:', error);
19     }
20 };
21
```

3 - Componentes:

App.jsx: Este es el componente principal que define las rutas y renderiza los demás componentes. Aquí se configura el enrutamiento de la aplicación utilizando React Router, se importan los otros componentes y se envuelven en el proveedor de contexto PeliculaProvider.

```
App.jsx U
src > App.jsx
1  import { useState } from 'react'
2  import { BrowserRouter, Route, Routes } from "react-router-dom"
3  import reactLogo from './assets/video.png'
4  import PeliculaProvider from './components/PeliculaProvider'
5  import Navbar from './components/Navbar'
6  import Catalog from "./components/Catalogo"
7  import InsertarPelicula from './components/InsertarPelicula'
8  import PeliculaCard from './components/PeliculaCard'
9  import PeliculaDetalle from './components/PeliculaDetalle'
10 import ModificarPelicula from './components/ModificarPelicula'
11 import Nosotros from './components/Nosotros'
12 import { ToastContainer, toast } from 'react-toastify';
13 import 'react-toastify/dist/ReactToastify.css';
```

```
function App() {
  // const [count, setCount] = useState(0)

  return (
    <>
    <BrowserRouter>
      <PeliculaProvider>
          <Navbar />
          <Routes>
            <Route path="/" element={<Catalog />} />
            <Route path="/peliculas" element={<Catalog />} />
            <Route path="/pelicula/:id" element={<PeliculaDetalle />} />
            <Route path="/insertar" element={<InsertarPelicula />} />
            <Route path="/modificar/:id" element={<ModificarPelicula />} />
            <Route path="/nosotros" element={<Nosotros />} />
          </Routes>
          <ToastContainer />
        </PeliculaProvider>
      </BrowserRouter>
    </>
  )
}

export default App
```

Catalog.jsx: Este componente muestra el catálogo de películas. Utiliza el contexto PeliculaContext para obtener la lista de películas y las renderiza usando el componente PeliculaCard.

Catalogo.jsx U

src > components > Catalogo.jsx

```jsx
import React, { useContext } from 'react';
import PeliculaContext from './PeliculaContext';
import PeliculaCard from './PeliculaCard';

const Catalog = () => {
    const { peliculas } = useContext(PeliculaContext);

    return (
        <div className="container my-5">
            <h1 className="mb-4 text-center">Catálogo de Películas</h1>
            <div className="row row-cols-1 row-cols-md-3 g-4">
                {peliculas.map((pelicula) => (
                <div className="col" key={pelicula.id}>
                    <PeliculaCard pelicula={pelicula} />
                </div>
                ))}
            </div>
        </div>
    );
};

export default Catalog;
```

InsertarPelicula.jsx: Este componente permite a los usuarios insertar una nueva película. Utiliza el hook useForm de la biblioteca react-hook-form para manejar el formulario y el contexto PeliculaContext para actualizar la lista de películas.

InsertarPelicula.jsx U

src > components > InsertarPelicula.jsx

```jsx
import React, { useContext } from 'react';
import { useNavigate } from 'react-router-dom';
import PeliculaContext from './PeliculaContext';
import { useForm } from 'react-hook-form';
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

const InsertarPelicula = () => {
    const { peliculas, setPeliculas } = useContext(PeliculaContext);
    const navigate = useNavigate();
    const { register, handleSubmit, reset } = useForm();

    const guardarPeli = (data) => {
        const nuevaPelicula = {
          id: peliculas.length + 1,
          ...data,
          rented: false,
        };
        setPeliculas([...peliculas, nuevaPelicula]);
        toast.success('Película insertada', {
          position: 'top-right',
          autoClose: 3000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: undefined,
        });
        reset();
        navigate('/peliculas');
      };

    return (
      <div className="container my-5">
        <h1 className="mb-4 text-center">Insertar Nueva Película</h1>
        <form onSubmit={handleSubmit(guardarPeli)}>
          <div className="mb-3">
            <label htmlFor="title" className="form-label">
              Título
            </label>
            <input
              type="text"
              className="form-control"
              id="title"
              {...register('title', { required: true })}
            />
          </div>
          <div className="mb-3">
            <label htmlFor="year" className="form-label">
              Año
            </label>
            <input
              type="text"
              className="form-control"
              id="year"
              {...register('year', { required: true })}
            />
          </div>
          <div className="mb-3">
            <label htmlFor="director" className="form-label">
              Director
```

```
            <label htmlFor="director" className="form-label">
              Director
            </label>
            <input
              type="text"
              className="form-control"
              id="director"
              {...register('director', { required: true })}
            />
          </div>
          <div className="mb-3">
            <label htmlFor="poster" className="form-label">
              URL del Póster
            </label>
            <input
              type="text"
              className="form-control"
              id="poster"
            />
          </div>
```

```
            <input
              type="text"
              className="form-control"
              id="poster"
            />
          </div>
          <div className="mb-3">
            <label htmlFor="synopsis" className="form-label">
              Sinopsis
            </label>
            <textarea
              className="form-control"
              id="synopsis"
              rows="3"
            ></textarea>
          </div>
          <button type="submit" className="btn btn-primary">
            Insertar Película
          </button>
        </form>
      </div>
    );
  };

  export default InsertarPelicula;
```

ModificarPelicula.jsx: Este componente permite a los usuarios modificar una película existente. Al igual que InsertarPelicula.jsx, utiliza useForm y PeliculaContext para actualizar la información de la película.

ModificarPelicula.jsx U

src > components > ModificarPelicula.jsx

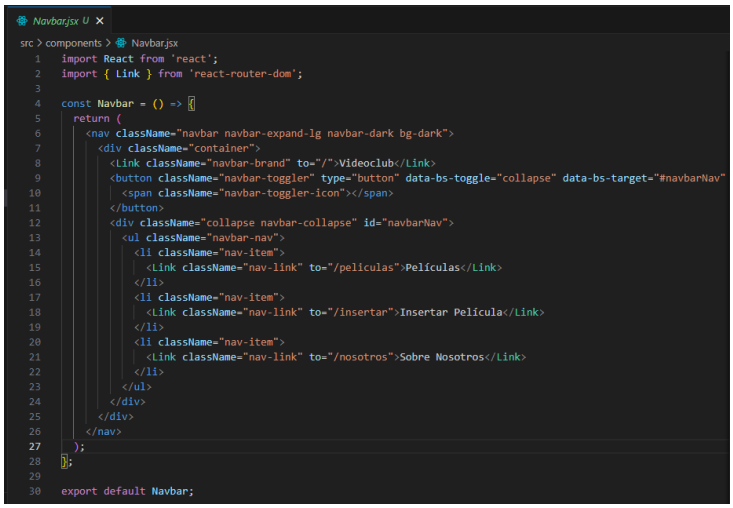
```
import React, { useContext } from 'react';
import { useNavigate, useParams } from 'react-router-dom';
import PeliculaContext  from './PeliculaContext';
import { useForm } from 'react-hook-form';
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

const ModificarPelicula = () => {
  const { id } = useParams();
  const { peliculas, setPeliculas } = useContext(PeliculaContext);
  const navigate = useNavigate();
  const { register, handleSubmit } = useForm();

  const pelicula = peliculas.find((p) => p.id === parseInt(id));

  const guardarPeli = (data) => {
    const updatedPelicula = {
      id: pelicula.id,
      ...data,
      rented: pelicula.rented,
    };
    const updatedPeliculas = peliculas.map((p) =>
      p.id === pelicula.id ? updatedPelicula : p
    );
    setPeliculas(updatedPeliculas);
    toast.success('Pelicula modificada', {
      position: 'top-right',
      autoClose: 3000,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
```

```
    setPeliculas(updatedPeliculas);
    toast.success('Pelicula modificada', {
      position: 'top-right',
      autoClose: 3000,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      progress: undefined,
    });
    navigate('/peliculas');
  };

  return (
    <div className="container my-5">
      <h1 className="mb-4 text-center">Modificar Película</h1>
      <form onSubmit={handleSubmit(guardarPeli)}>
        <div className="mb-3">
          <label htmlFor="title" className="form-label">
            Título
          </label>
          <input
            type="text"
            className="form-control"
            id="title"
            defaultValue={pelicula.title}
            {...register('title', { required: true })}
          />
        </div>
```

```
        <div className="mb-3">
          <label htmlFor="year" className="form-label">
            Año
          </label>
          <input
            type="text"
            className="form-control"
            id="year"
            defaultValue={pelicula.year}
            {...register('year', { required: true })}
          />
        </div>
        <div className="mb-3">
          <label htmlFor="director" className="form-label">
            Director
          </label>
          <input
            type="text"
            className="form-control"
            id="director"
            defaultValue={pelicula.director}
            {...register('director', { required: true })}
          />
        </div>
```

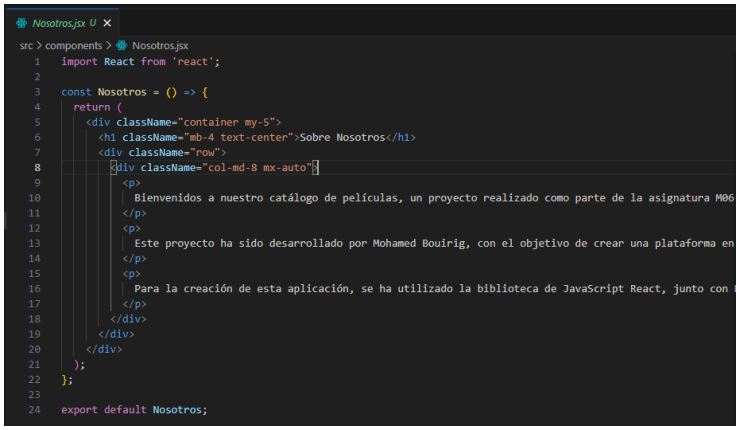
```
          <input
            type="text"
            className="form-control"
            id="poster"
            defaultValue={pelicula.poster}
          />
        </div>
        <div className="mb-3">
          <label htmlFor="synopsis" className="form-label">
            Sinopsis
          </label>
          <textarea
            className="form-control"
            id="synopsis"
            defaultValue={pelicula.synopsis}
            rows="3"
          ></textarea>
        </div>
        <button type="submit" className="btn btn-primary">
          Guardar Cambios
        </button>
      </form>
    </div>
  );
};

export default ModificarPelicula;
```

Navbar.jsx: Este componente proporciona la barra de navegación de la aplicación, con enlaces a las diferentes secciones (Peliculas “Catálogo”, Insertar Película, Sobre Nosotros).



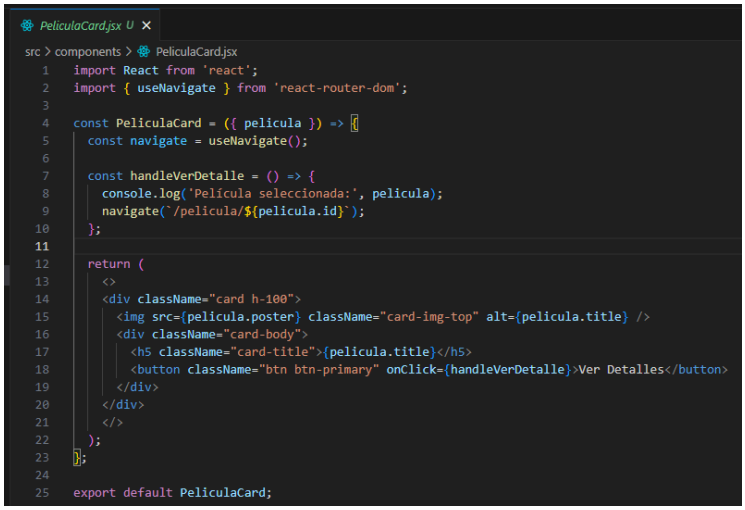
```jsx
import React from 'react';
import { Link } from 'react-router-dom';

const Navbar = () => {
  return (
    <nav className="navbar navbar-expand-lg navbar-dark bg-dark">
      <div className="container">
        <Link className="navbar-brand" to="/">Videoclub</Link>
        <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarNav">
          <ul className="navbar-nav">
            <li className="nav-item">
              <Link className="nav-link" to="/peliculas">Películas</Link>
            </li>
            <li className="nav-item">
              <Link className="nav-link" to="/insertar">Insertar Película</Link>
            </li>
            <li className="nav-item">
              <Link className="nav-link" to="/nosotros">Sobre Nosotros</Link>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  );
};

export default Navbar;
```

Nosotros.jsx: Este componente muestra información sobre el proyecto y los desarrolladores.



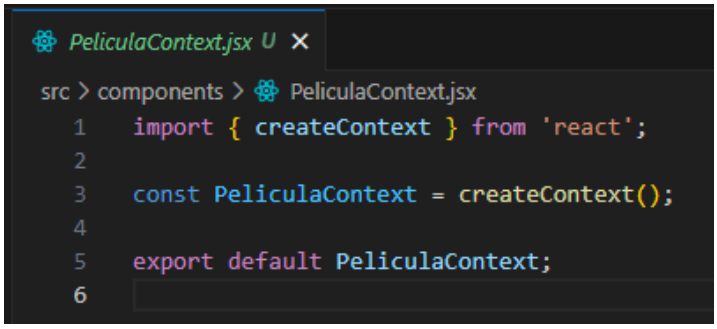
```jsx
import React from 'react';

const Nosotros = () => {
  return (
    <div className="container my-5">
      <h1 className="mb-4 text-center">Sobre Nosotros</h1>
      <div className="row">
        <div className="col-md-8 mx-auto">
          <p>
            Bienvenidos a nuestro catálogo de películas, un proyecto realizado como parte de la asignatura M06
          </p>
          <p>
            Este proyecto ha sido desarrollado por Mohamed Bouirig, con el objetivo de crear una plataforma en
          </p>
          <p>
            Para la creación de esta aplicación, se ha utilizado la biblioteca de JavaScript React, junto con
          </p>
        </div>
      </div>
    </div>
  );
};

export default Nosotros;
```

PeliculaCard.jsx: Este componente representa una tarjeta individual de película en el catálogo. Cuando se hace clic en el botón "Ver Detalles", redirige al usuario a la página de detalles de la película.



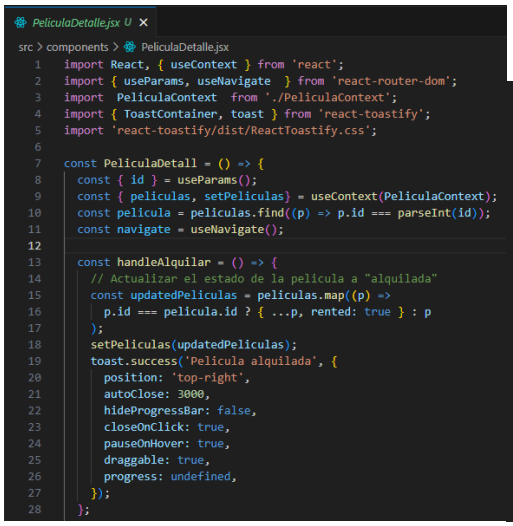
```jsx
import React from 'react';
import { useNavigate } from 'react-router-dom';

const PeliculaCard = ({ pelicula }) => {
  const navigate = useNavigate();

  const handleVerDetalle = () => {
    console.log('Película seleccionada:', pelicula);
    navigate(`/pelicula/${pelicula.id}`);
  };

  return (
    <>
    <div className="card h-100">
      <img src={pelicula.poster} className="card-img-top" alt={pelicula.title} />
      <div className="card-body">
        <h5 className="card-title">{pelicula.title}</h5>
        <button className="btn btn-primary" onClick={handleVerDetalle}>Ver Detalles</button>
      </div>
    </div>
    </>
  );
};

export default PeliculaCard;
```

PeliculaContext.jsx: Este archivo define el contexto PeliculaContext, que se utiliza para compartir el estado de las películas entre los diferentes componentes de la aplicación.



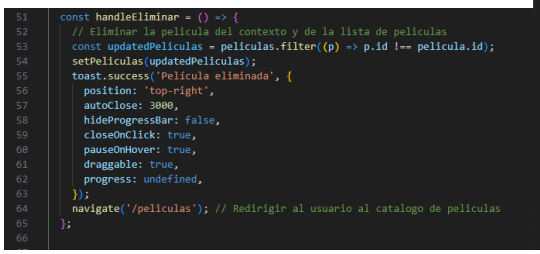
```jsx
import { createContext } from 'react';

const PeliculaContext = createContext();

export default PeliculaContext;
```

PeliculaDetalle.jsx: Este componente muestra los detalles de una película seleccionada, incluyendo la posibilidad de alquilar, devolver, modificar y eliminar la película.

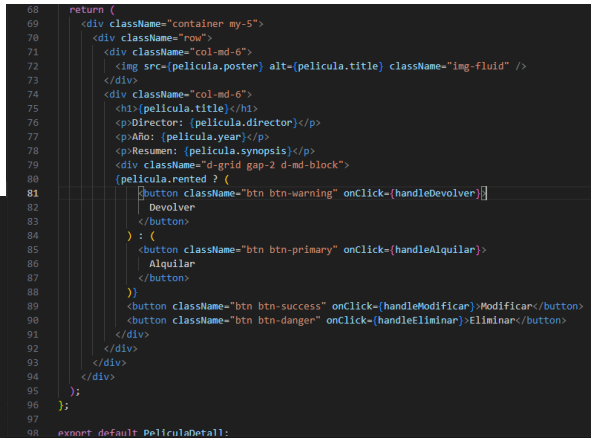


```jsx
import React, { useContext } from 'react';
import { useParams, useNavigate } from 'react-router-dom';
import PeliculaContext from './PeliculaContext';
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

const PeliculaDetall = () => {
  const { id } = useParams();
  const { peliculas, setPeliculas} = useContext(PeliculaContext);
  const pelicula = peliculas.find((p) => p.id === parseInt(id));
  const navigate = useNavigate();

  const handleAlquilar = () => {
    // Actualizar el estado de la película a "alquilada"
    const updatedPeliculas = peliculas.map((p) =>
      p.id === pelicula.id ? { ...p, rented: true } : p
    );
    setPeliculas(updatedPeliculas);
    toast.success('Película alquilada', {
      position: 'top-right',
      autoClose: 3000,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      progress: undefined,
    });
  };
```

```jsx
const handleDevolver = () => {
  // Actualizar el estado de la película a "devuelta"
  const updatedPeliculas = peliculas.map((p) =>
    p.id === pelicula.id ? { ...p, rented: false } : p
  );
  setPeliculas(updatedPeliculas);
  toast.success('Película devuelta', {
    position: 'top-right',
    autoClose: 3000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
  });
};

const handleModificar = () => {
  navigate(`/modificar/${pelicula.id}`);
};
```



```jsx
const handleEliminar = () => {
  // Eliminar la película del contexto y de la lista de películas
  const updatedPeliculas = peliculas.filter((p) => p.id !== pelicula.id);
  setPeliculas(updatedPeliculas);
  toast.success('Película eliminada', {
    position: 'top-right',
    autoClose: 3000,
    hideProgressBar: false,
    closeOnClick: true,
    pauseOnHover: true,
    draggable: true,
    progress: undefined,
  });
  navigate('/peliculas'); // Redirigir al usuario al catálogo de películas
};
```



```jsx
return (
  <div className="container my-5">
    <div className="row">
      <div className="col-md-6">
        <img src={pelicula.poster} alt={pelicula.title} className="img-fluid" />
      </div>
      <div className="col-md-6">
        <h1>{pelicula.title}</h1>
        <p>Director: {pelicula.director}</p>
        <p>Año: {pelicula.year}</p>
        <p>Resumen: {pelicula.synopsis}</p>
        <div className="d-grid gap-2 d-md-block">
        {pelicula.rented ? (
          <button className="btn btn-warning" onClick={handleDevolver}>
            Devolver
          </button>
        ) : (
          <button className="btn btn-primary" onClick={handleAlquilar}>
            Alquilar
          </button>
        )}
        <button className="btn btn-success" onClick={handleModificar}>Modificar</button>
        <button className="btn btn-danger" onClick={handleEliminar}>Eliminar</button>
        </div>
      </div>
    </div>
  </div>
);
};

export default PeliculaDetall;
```

PeliculaProvider.jsx: Este componente proporciona el proveedor de contexto para las películas, cargando los datos iniciales desde un archivo JSON gracias a las funciones de fetchData.js.

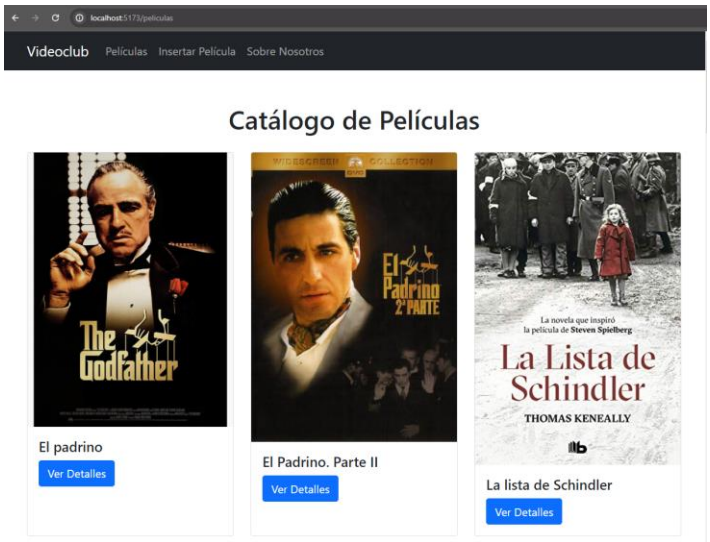
PeliculaProvider.jsx U

src > components > PeliculaProvider.jsx

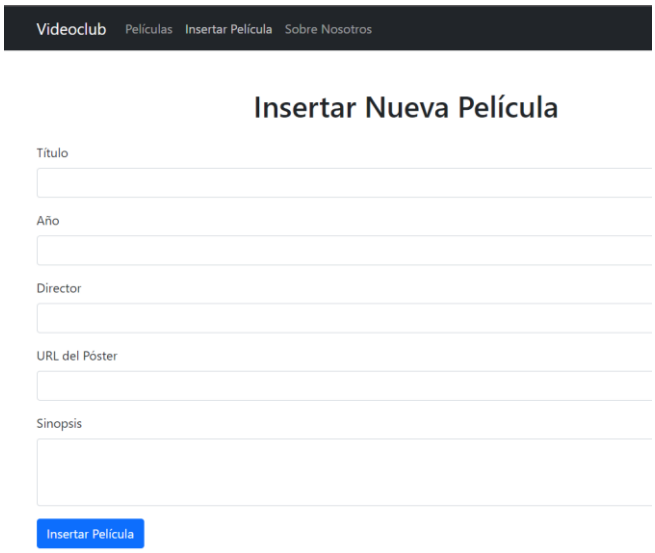
```jsx
import React, { useState, useEffect } from 'react';
import PeliculaContext from '../components/PeliculaContext';
import { rebreInformacio, obtenerPeliculas } from '../helpers/fetchData.js';

const PeliculaProvider = ({ children }) => {
    const [peliculas, setPeliculas] = useState([]);

    useEffect(() => {
        const fetchData = async () => {
            try {
                const data = await rebreInformacio();
                setPeliculas(data);
            } catch (error) {
                console.error('Error al obtener peliculas:', error);
            }
        };
        fetchData();
    }, []);

    return (
        <PeliculaContext.Provider value={{ peliculas, setPeliculas }}>
            {children}
        </PeliculaContext.Provider>
    );
};

export default PeliculaProvider;
```

4 - Funcionalidades:

Visualización del catálogo de películas: El usuario puede ver la lista de películas disponibles en el catálogo.



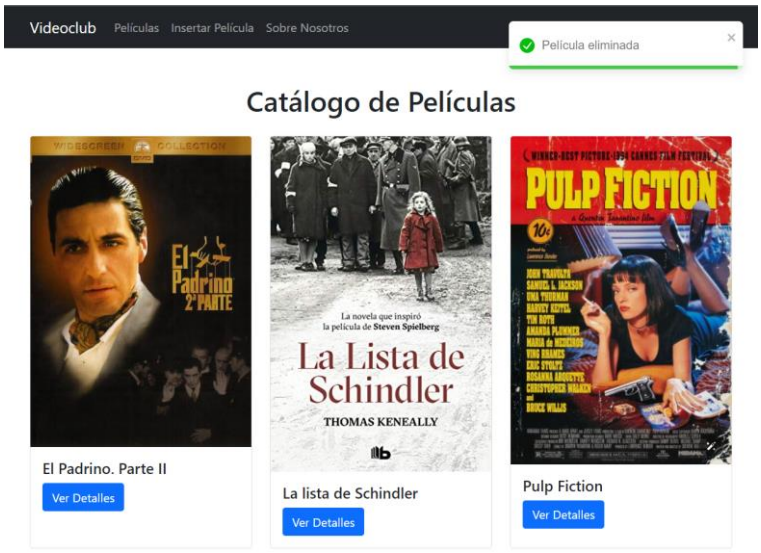
Inserción de nuevas películas: El usuario puede agregar una nueva película al catálogo a través del formulario de inserción.



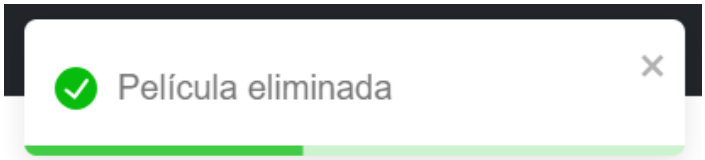
Modificación de películas existentes: El usuario puede editar la información de una película existente a través del formulario de modificación.

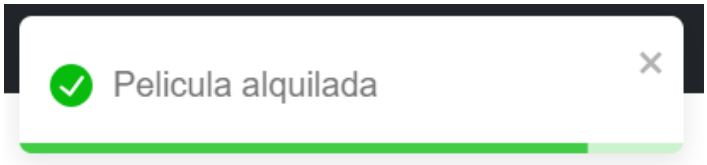
Alquiler y devolución de películas: El usuario puede alquilar o devolver una película desde la página de detalles de la película.

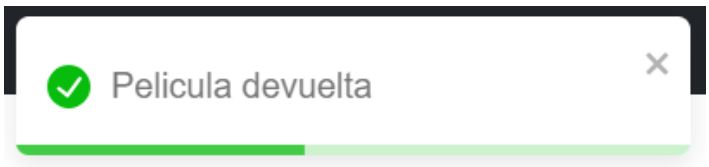
Eliminación de películas: El usuario puede eliminar una película desde la página de detalles de la película.

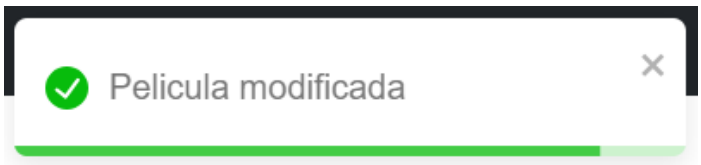


Notificaciones de éxito: Se utilizan notificaciones (toasts) para informar al usuario sobre las operaciones exitosas, como la inserción, modificación, alquiler o eliminación de una película.









5 - Tecnologías utilizadas:

React: Librería de JavaScript para la construcción de interfaces de usuario.

React Router: Biblioteca para el manejo de enrutamiento en aplicaciones React.

React Hook Form: Biblioteca para la gestión de formularios en React.

React Toastify: Biblioteca para mostrar notificaciones. (https://www.npmjs.com/package/react-toastify)

Bootstrap: Framework CSS para el diseño y la interfaz de usuario.

JSON: Formato de intercambio de datos utilizado para almacenar y transmitir la información de las películas.

6 - Conclusión:

Ha sido un proyecto bastante interesante que me ha ayudado a comprender mucho mejor React y de sus ventajas. Al principio he tenido dificultades a la hora de realizar la lógica para alquilar, modificar y eliminar, pero con algo de investigación he encontrado varias formas de hacerlo.

También he encontrado una librería toastify (parecida o igual a toastr, que te enseñé en la anterior uf pero que no pude implementar) para React para el tema de las notificaciones y no hacerlo con alert ya que era demasiado intrusivo.