Establece la URL base para las solicitudes a la API. Todas las solicitudes se harán a esta dirección base

const URL = "http://localhost/api/login";

**LOGINREQUEST**

export const loginRequest = async (data) => {

const res = await fetch(`${URL}/login.php`, {

method: "POST",

body: JSON.stringify(data),

});

const loginData = await res.json();

console.log(loginData);

return loginData;

};

* Esta función realiza una solicitud de inicio de sesión a la URL ${URL}/login.php utilizando el método HTTP POST.
* La información del usuario se envía en el cuerpo de la solicitud en formato JSON, utilizando JSON.stringify(data).
* La respuesta de la solicitud se almacena en la variable res, y luego se convierte a formato JSON utilizando res.json().

**REGISTERREQUEST**

export const registerRequest = async (data) => {

const res = await fetch(`${URL}/register.php`, {

method: "POST",

body: JSON.stringify(data),

});

const registerData = await res.json();

console.log(registerData);

return registerData;

};

Similar a loginRequest, esta función realiza una solicitud de registro a la URL ${URL}/register.php utilizando el método HTTP POST.

La información del usuario se envía en el cuerpo de la solicitud en formato JSON, al igual que en la función de inicio de sesión.

La respuesta de la solicitud se almacena en la variable res y se convierte a formato JSON.

**MOVIEDETAILS**

mostramos los detalles de una película, incluido su título, descripción, idioma, popularidad y un tráiler de YouTube. También presenta una sección de recomendaciones para películas relacionadas.

// Importación de módulos y componentes de React

import { useEffect, useState } from "react";

import { Link, useParams } from "react-router-dom";

import YouTube from "react-youtube";

import axios from "axios";

// Importación de estilos CSS específicos para este componente

import css from "./MovieDetails.module.css";

// Importación de un componente Loader y RecomendedMovie

import { Loader } from "../Loader/Loader";

import { RecomededMovie } from "../RecomendedMovie/RecomededMovie";

// Definición de constantes para la URL de la API, clave API, y URL de imágenes

const API\_URL = "https://api.themoviedb.org/3";

const API\_KEY = "4f5f43495afcc67e9553f6c684a82f84";

const URL\_IMAGE = "https://image.tmdb.org/t/p/original";

// Definición del componente funcional MovieDetails

export const MovieDetails = () => {

// Estado local para almacenar la información de la película actual

const [movie, setMovie] = useState({ title: "Loading Movies" });

// Estado local para indicar si la carga de datos está en progreso

const [isLoading, setIsLoading] = useState(true);

// Estado local para almacenar las películas recomendadas

const [moviesReccomended, setMoviesRecomended] = useState({});

// Estado local para controlar si el tráiler está reproduciéndose

const [playing, setPlaying] = useState(false);

// Estado local para almacenar la información del tráiler

const [trailer, setTrailer] = useState(null);

// Obtención de los parámetros de la URL, en este caso, el ID de la película

const params = useParams();

// Función para realizar la petición de detalles de una película y mostrar el tráiler

const fetchMovie = async () => {

setIsLoading(true);

const { data } = await axios.get(`${API\_URL}/movie/${params.id}`, {

params: {

api\_key: API\_KEY,

append\_to\_response: "videos",

},

});

setMovie(data);

setTrailer(trailer ? trailer : data.videos.results[0]);

setIsLoading(false);

};

// Efecto secundario para cargar las películas recomendadas al montar el componente

useEffect(() => {

recommendMovies();

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

// Función para obtener y almacenar las películas recomendadas

const recommendMovies = async () => {

const { data } = await axios.get(

`${API\_URL}/movie/${params.id}/recommendations?api\_key=${API\_KEY}`,

{

params: {

api\_key: API\_KEY,

},

}

);

setMoviesRecomended(data.results);

};

// Efecto secundario para cargar los detalles de la película al montar el componente

useEffect(() => {

fetchMovie();

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

// Renderizado condicional del componente Loader durante la carga de datos

if (isLoading) {

return <Loader />;

}

// Renderizado del componente MovieDetails

return (

<>

<div className={css.movies\_\_details}>

<h1>Title: {movie.original\_title}</h1>

<div className={css.movie\_\_info\_\_container}>

<img

width={"700"}

src={URL\_IMAGE + movie.backdrop\_path}

alt={movie.backdrop\_path}

/>

<div className={css.movie\_\_description}>

<h2 className={css.video\_\_title}>Video Trailer</h2>

{movie ? (

<div className={css.container\_\_video}>

{playing ? (

<>

<button

onClick={() => setPlaying(false)}

className={css.button\_\_video}

>

Close

</button>

<YouTube videoId={trailer.key} />

</>

) : (

<div className="container">

<div className="">

{trailer ? (

<button

className="boton"

onClick={() => setPlaying(true)}

type="button"

>

Play Trailer

</button>

) : (

"Sorry, no trailer available"

)}

<h1>{movie.title}</h1>

<p>{movie.overview}</p>

</div>

</div>

)}

</div>

) : null}

</div>

</div>

<p>

<strong>Languaje</strong>: <span>{movie.original\_language}</span>{" "}

</p>

<p>

<strong>Description</strong>: {movie.overview}

</p>

<p>

<strong>Popularity</strong>: {movie.popularity}

</p>

</div>

<section>

<h2 className={css.recomended\_\_title}>Recomendaciones</h2>

<RecomededMovie moviesList={moviesReccomended} />

</section>

<footer className={css.button\_\_movie}>

<Link to={"/"}>

<button>Volver a las peliculas</button>

</Link>

</footer>

</>

);

};

Este componente utiliza efectos secundarios (useEffect) para cargar datos al montar el componente y para recomendar películas. También utiliza estados locales (useState) para manejar la información de la película, el estado de carga, las películas recomendadas, el estado de reproducción del tráiler y la información del tráiler.

**MOVIELIST**

/\* eslint-disable react/prop-types \*/

// Importación del hook useNavigate desde react-router-dom

import { useNavigate } from "react-router-dom";

// Importación de estilos CSS específicos para este componente

import css from "./MovieList.module.css";

// Definición de la URL base para las imágenes de las películas

const URL\_IMAGE = "https://image.tmdb.org/t/p/original";

// Definición del componente funcional MovieList

export const MovieList = ({ movies = [] }) => {

// Uso del hook useNavigate para la navegación programática

const navigate = useNavigate();

// Renderizado del componente MovieList

return (

<div className={css.movies}>

{movies.map((movie) => (

<div

key={movie.imdbID}

onClick={() => navigate(`/movie-details/${movie.id}`)}

>

{/\* Mostrar la imagen de la película \*/}

<img src={`${URL\_IMAGE + movie.poster\_path}`} alt="" width="300" />

{/\* Mostrar el título de la película \*/}

<h4 className="text-center">{movie.title}</h4>

{/\* Mostrar la fecha de lanzamiento de la película \*/}

<p>{movie.release\_date}</p>

</div>

))}

</div>

);

};

Desactivación de las advertencias de prop-types:

jsx

/\* eslint-disable react/prop-types \*/

Se desactivan las advertencias de prop-types para este archivo específico. Esto permite que el código omita las verificaciones de tipo de las propiedades recibidas.

Importación de módulos y estilos:

jsx

// Importación del hook useNavigate desde react-router-dom

import { useNavigate } from "react-router-dom";

// Importación de estilos CSS específicos para este componente

import css from "./MovieList.module.css";

Se importa el hook useNavigate de react-router-dom para la navegación programática.

Se importan los estilos CSS del archivo MovieList.module.css.

Definición de constantes:

jsx

// Definición de la URL base para las imágenes de las películas

const URL\_IMAGE = "https://image.tmdb.org/t/p/original";

Definición del componente funcional MovieList:

jsx

export const MovieList = ({ movies = [] }) => {

// Uso del hook useNavigate para la navegación programática

const navigate = useNavigate();

// Renderizado del componente MovieList

return (

<div className={css.movies}>

{movies.map((movie) => (

<div

key={movie.imdbID}

onClick={() => navigate(`/movie-details/${movie.id}`)}

>

{/\* Mostrar la imagen de la película \*/}

<img src={`${URL\_IMAGE + movie.poster\_path}`} alt="" width="300" />

{/\* Mostrar el título de la película \*/}

<h4 className="text-center">{movie.title}</h4>

{/\* Mostrar la fecha de lanzamiento de la película \*/}

<p>{movie.release\_date}</p>

</div>

))}

</div>

);

};

El componente MovieList recibe un arreglo de películas como propiedades (movies).

Utiliza el hook useNavigate para la navegación programática cuando se hace clic en una película.

**RECOMEDEDMOVIE**

/\* eslint-disable react/prop-types \*/

// Importación del componente Loader y estilos CSS específicos para este componente

import { Loader } from "../Loader/Loader";

import css from "./RecomendedMovie.module.css";

// Definición de la URL base para las imágenes de las películas

const URL\_IMAGE = "https://image.tmdb.org/t/p/original";

// Definición del componente funcional RecomededMovie

export const RecomededMovie = ({ moviesList = [] }) => {

// Comprobación: Si la lista de películas recomendadas está vacía, muestra el componente Loader

if (moviesList.length <= 0) {

return <Loader />;

}

// Renderizado del componente RecomededMovie

return (

<div className={css.movies\_\_recomended\_\_grid}>

{/\* Mapeo sobre la lista de películas recomendadas y muestra la información de cada una \*/}

{moviesList.map((movie) => (

<div key={movie.id}>

{/\* Mostrar el título de la película \*/}

<h3>{movie.title}</h3>

{/\* Mostrar la imagen de la película \*/}

<img

width={"200"}

src={URL\_IMAGE + movie.poster\_path}

alt={movie.poster\_path}

/>

{/\* Mostrar el título original de la película \*/}

<p>{movie.original\_title}</p>

{/\* Mostrar la popularidad de la película \*/}

<p>{movie.popularity}</p>

</div>

))}

</div>

);

};