

Documentación del Proyecto Netflix

1. Descripción General

Este proyecto es una aplicación estilo Netflix que permite a los usuarios explorar películas y series, iniciar sesión, suscribirse, y gestionar su contenido favorito. El sistema está dividido en frontend y backend, conectados mediante llamadas HTTP.

2. Tecnologías Utilizadas

Frontend:

- React con Vite JS
- Visual Studio Code
- Hooks: useState, useEffect, useRef
- Zustand
- React Router DOM

Backend:

- Node.js con Express
- Conexión a base de datos (en config/db.js)
- Estructura con carpetas routes y controllers
- Nodemon

3. Estructura del Proyecto

Frontend (src/):

- routes/AppRoutes.jsx: Define las rutas principales
- pages/: Login.jsx, Suscripcion.jsx, InicioSesion.jsx (usa Email.js); Series.jsx, Peliculas.jsx (con carruseles)
- components/: Footer.jsx, Mains.jsx, Header.jsx; MiLista.jsx (sin repeticiones, permite eliminar)
- store/useStore.js: Maneja películas, series, favoritos, búsqueda
- css/: Estilos personalizados

Backend:

- config/db.js: Conexión a la base de datos

Documentación del Proyecto Netflix

- routes/: Rutas para las entidades
- controllers/: Lógica y consultas de base de datos

4. Ejemplo de Código Backend

Archivo config/db.js:

```
const mysql = require('mysql2');
const db = mysql.createConnection({
  host: 'localhost',
  port: 3306,
  user: 'root',
  password: '29deMayo.',
  database: 'netflix'
});
module.exports = db;
```

Archivo routes/roles.js:

```
const express = require("express");
const router = express.Router();
const { getAllRoles, getOneRol, createRol, updateRol, deleteRol, getEspecificRol } =
require("../controllers/roles");
router.get("/roles", getAllRoles);
router.get("/roles/:id", getOneRol);
router.post("/roles", createRol);
router.put("/roles/:id", updateRol);
router.delete("/roles/:id", deleteRol);
router.get("roles/buscar/especifico", getEspecificRol);
module.exports = router;
```

Documentación del Proyecto Netflix

Archivo controllers/roles.js (fragmento):

```
const getAllRoles = (req, res) => {  
  const consulta = "SELECT * FROM roles;";  
  db.query(consulta, (err, results) => {  
    if (err) throw err;  
    res.json(results);  
  });  
};
```

5. Funcionalidades Principales

- Carruseles de películas y series
- Login, registro y suscripción con Email.js
- Página Mi Lista: contenido favorito sin duplicados
- Footer con íconos, enlaces, código de servicio y derechos
- Interfaz de ABM (Alta, Baja, Modificación) para rol de administrador

6. Dependencias

Backend:

```
"cors": "^2.8.5",  
"express": "^5.1.0",  
"mysql": "^2.18.1",  
"mysql2": "^3.14.1"  
"nodemon": "^3.1.10"
```

Frontend:

```
"axios", "react-router-dom", "react-icons", "zustand", "emailjs-com"
```

7. Consideraciones Finales

Documentación del Proyecto Netflix

- La base de datos está conectada desde el backend, pero no se detalla aquí
- Interfaz atractiva y navegación clara
- Carruseles adaptados por categorías
- Administración de contenido restringida al rol admin