

# Desarrollo de aplicaciones multiplataforma

## **Práctica Evaluación Continua 2: Explorador de películas con React**

---

### **Entregable**

1. Código subido en un comprimido .ZIP
2. Memoria en PDF (no Word) explicada brevemente  
Puedes incluir problemas y dificultades que te hayas encontrado.
3. Enlace de repositorio en Github

### **Fecha de entrega**

Domingo 2 de noviembre a las 23:59

---

### **Objetivo**

El objetivo es desarrollar una aplicación web de películas utilizando React y la API de **The Movie Database (TMDB)**. La aplicación permitirá a los usuarios buscar películas, ver películas populares y gestionar una lista de favoritos con persistencia en localStorage.

### **Requisitos funcionales**

#### **1. Página Principal (3p)**

- Mostrar películas populares al cargar la página inicialmente
- Barra de búsqueda para buscar películas por título
- Indicador de carga mientras se obtienen los datos

#### **2. Sistema de Favoritos (2p)**

- Botón en cada tarjeta de película para agregar/quitar de favoritos

- Indicador visual del estado de favorito (corazón activo/inactivo)
- Persistencia de favoritos usando localStorage
- Página dedicada para visualizar todas las películas favoritas

### **3. Navegación (2p)**

- Barra de navegación fija con logo/título de la aplicación
- Enlaces para navegar entre Home y Favorites
- Navegación mediante React Router

### **4. Tarjetas de Películas (3p)**

- Imagen del póster de la película
- Título de la película
- Año de lanzamiento
- Overlay con botón de favoritos al hacer hover

---

## **Componentes Requeridos**

### **1. App.jsx**

- Componente principal de la aplicación
- Configuración de rutas con React Router
- Envolver la aplicación con el provider del contexto

### **2. NavBar.jsx**

- Componente de navegación
- Links a Home y Favorites
- Diseño responsivo

### **3. Home.jsx (Página)**

- Lógica de búsqueda de películas
- Carga inicial de películas populares
- Formulario de búsqueda
- Grid de películas

### **4. Favorites.jsx**

- Visualización de películas favoritas
- Mensaje cuando no hay favoritos
- Grid de películas favoritas

#### 5. **MovieCard.jsx**

- Tarjeta individual de película
- Imagen del póster
- Información básica (título, año)
- Botón de favorito interactivo

#### 6. **MovieContext.jsx**

- Context API para gestión global de favoritos
- Custom hook para consumir el contexto

### **Hooks de React Utilizados**

#### 1. **useState**

- Gestión del estado de búsqueda (searchQuery)
- Lista de películas (movies)
- Estado de carga (loading)
- Mensajes de error (error)
- Lista de favoritos (favorites)

#### 2. **useEffect**

- Cargar películas populares al montar el componente Home
- Cargar favoritos desde localStorage al iniciar
- Guardar favoritos en localStorage cuando cambien

#### 3. **useContext**

- Consumir el contexto de películas (MovieContext)
- Acceso a favoritos y funciones desde cualquier componente

#### 4. **createContext**

- Crear el contexto para compartir estado de favoritos

#### 5. **Favorites.jsx**

- Visualización de películas favoritas
- Mensaje cuando no hay favoritos
- Grid de películas favoritas

## 6. **MovieCard.jsx**

- Tarjeta individual de película
- Imagen del póster
- Información básica (título, año)
- Botón de favorito interactivo

## 7. **MovieContext.jsx**

- Context API para gestión global de favoritos
- Custom hook para consumir el contexto

## **API Utilizada**

### **The Movie Database (TMDB)**

**URL Base:** <https://api.themoviedb.org/3>

**Autenticación:** API Key (gratuita, requiere registro en <https://www.themoviedb.org/>)