



Objetivo

Desplegar en línea el proyecto recomendaciones-ia, que permite mostrar y sugerir películas usando inteligencia artificial (IA), MongoDB y React, para que esté disponible desde cualquier dispositivo conectado a internet.

Práctica 4.3: Despliegue en la nube con Render (Backend y Frontend) de CECYFLYX

- ¿Qué es Render?
- Render es una plataforma gratuita y fácil de usar para [subir tu proyecto completo a internet](#), con soporte para backend, frontend, bases de datos y más.
- Ventajas:
 - ➔ Sin tarjeta de crédito
 - ➔ Integración directa con GitHub
 - ➔ Fácil configuración
 - ➔ Dominio gratuito
- Requisitos previos
- Antes de continuar, asegúrate de tener:
 - ➔ Cuenta en GitHub
 - ➔ Proyecto recomendaciones-ia funcionando localmente
 - ➔ MongoDB Atlas ya conectado
 - ➔ API de OpenRouter funcionando
 - ➔ Cuenta en Render.com



Paso 1: Subir tu proyecto a GitHub.

- Crea un nuevo repositorio llamado CECYFLIX. Una vez que se haya creado el repositorio, dentro de nuestro proyecto en VS Code realizaremos un push.

```
git init
git remote add origin "LINK DE TU REPOSITORIO"
git add .
git commit -m "Proyecto recomendaciones-ia"
git push -u origin main
```

Paso 2: Desplegar el BACKEND en Render.

- Estructura esperada del proyecto

```
recomendaciones-ia/
  backend/
    index.js
    .env
    routes/
    models/
  frontend/
    src/
    public/
```

- Agregar configuración render.yaml
- Esto le dice a Render cómo construir tu backend. Creamos un archivo llamado **render.yaml** dentro de **backend**.

```
services:
  - type: web
    name: recomendaciones-backend
    env: node
    plan: free
    region: oregon
    buildCommand: npm install
    startCommand: node index.js
    rootDir: backend
    envVars:
      - key: MONGO_URI
        value: <tu URI de MongoDB Atlas>
      - key: OPENROUTER_API_KEY
        value: <tu clave API de OpenRouter>
```

- Debemos reemplazar **value** con nuestra URI de **MongoDB Atlas** y nuestra clave API de OpenRouter.

Paso 3: Crear el backend en Render.

- ➔ Entra a <https://render.com>
- ➔ Inicia sesión con tu cuenta de GitHub
- ➔ Haz clic en “New > Web Service”
- ➔ Autoriza el acceso a tu repositorio si no lo has hecho antes
- ➔ Selecciona el repositorio recomendaciones-ia
- ➔ En Root Directory, escribe: backend
- ➔ Configura lo siguiente:

Opción	Valor
Name	recomendaciones-backend
Environment	Node
Build Command	npm install
Start Command	node index.js

- En la sección Environment Variables, agrega:

Key	Value
MONGO_URI	tu URI completa de MongoDB Atlas
OPENROUTER_API_KEY	tu clave de OpenRouter

- Haz clic en Create Web Service. Render empezará a construir tu backend. Cuando termine, te mostrará una URL como:

<https://recomendaciones-backend.onrender.com> (URL EJEMPLO)

Paso 4: Desplegar el FRONTEND (React) en Render.

- Preparar el archivo App.js y modifica las URLs de tus fetch():
- En lugar de fetch('/api/peliculas') usa:

```
fetch('https://recomendaciones-backend.onrender.com/api/peliculas')
```

- En lugar de `fetch('/api/recomendaciones')` usa:

```
fetch('https://recomendaciones-backend.onrender.com/api/recomendaciones', {
  method: 'POST',
  //No borres el código que se encuentra aquí dentro.
})
```

Paso 5: Crear sitio estático en Render.

- ➔ Ve a Dashboard > **New** > **Static Site**
- ➔ Elige el mismo repositorio
- ➔ Configura:

Opción	Valor
Name	recomendaciones-frontend
Root Directory	frontend
Build Command	npm install && npm run build
Publish Directory	build

- Haz clic en Create Static Site
- Espera unos minutos y Render te dará una URL como:

<https://recomendaciones-frontend.onrender.com>

- ¡Despliegue completo!
- Tendrás algo así como se muestra en la tabla

Elemento	URL Pública Render
 Backend	https://recomendaciones-backend.onrender.com
 Frontend	https://recomendaciones-frontend.onrender.com
 Base de datos	En MongoDB Atlas, accesible desde el backend

- Las URLs son solo ejemplos, las correctas son las que obtendrás mediante realices esta actividad.