



BrisaacFlix

Películas Series Populares Categorías

Iniciar sesión

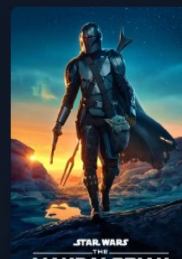
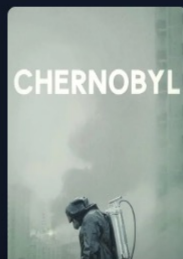
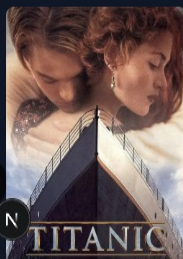
Bienvenido a BrisaacFlix

Millones de películas, series y contenido por descubrir. ¡Explora ya!

Buscar una película, serie...

Buscar

🔥 Tendencias



Brian Suarez Product Owner

Isaac Medina Scrum Master

Brian Suarez Isaac Medina Developers



> ¿Qué problema resuelve BrisaacFlix?

El consumo de contenido (películas, series, anime) en la comunidad hispanohablante ha crecido de forma sostenida. Existen múltiples plataformas de streaming y redes sociales, pero la información útil para decidir qué ver, es inconsistente y suele estar mezclada con ruido.

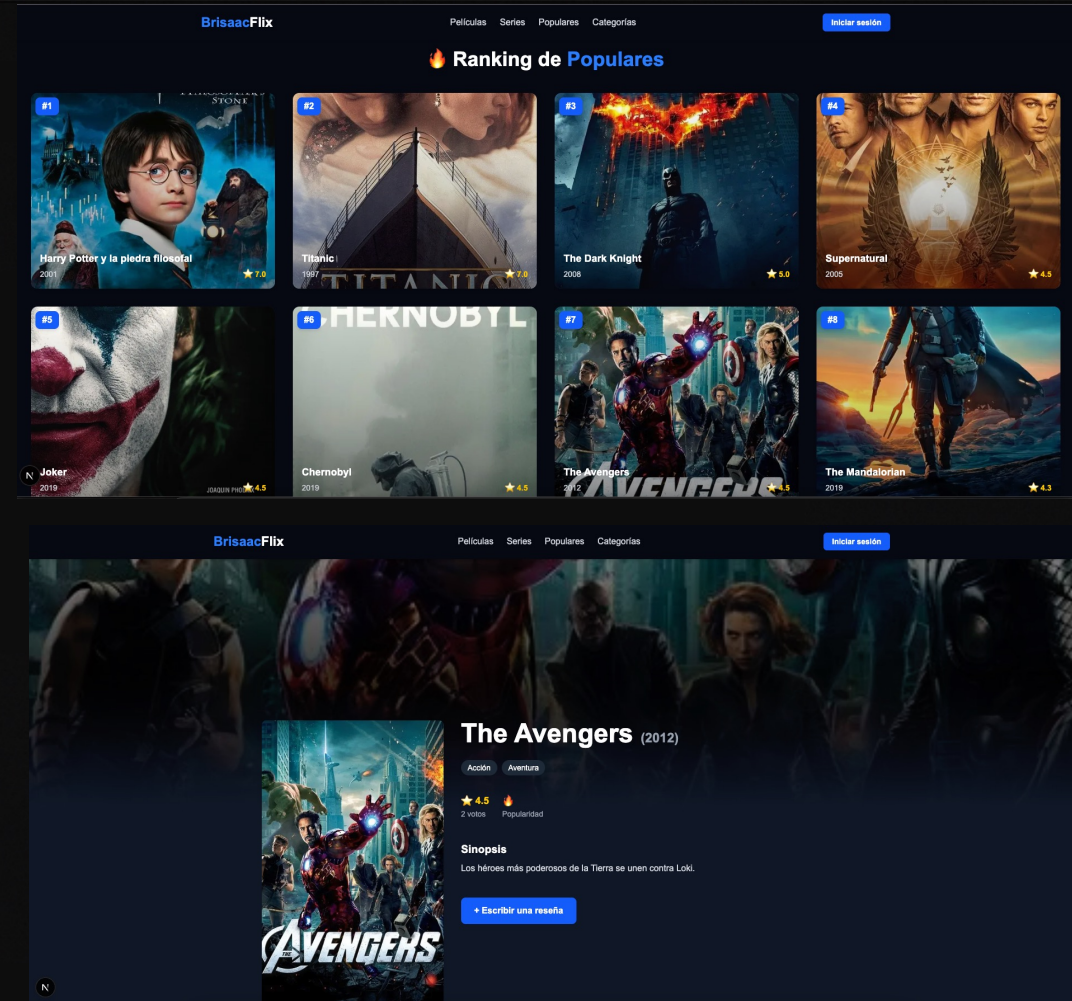




> ¿Cuál es el objetivo de BrisaacFlix?

Desarrollar una aplicación full-stack con Node.js, Express, MongoDB y un frontend, que permita a los usuarios registrar, calificar y rankear películas, animes y series geek, ofreciendo una plataforma segura, escalable y responsiva.

BrisaacFlix





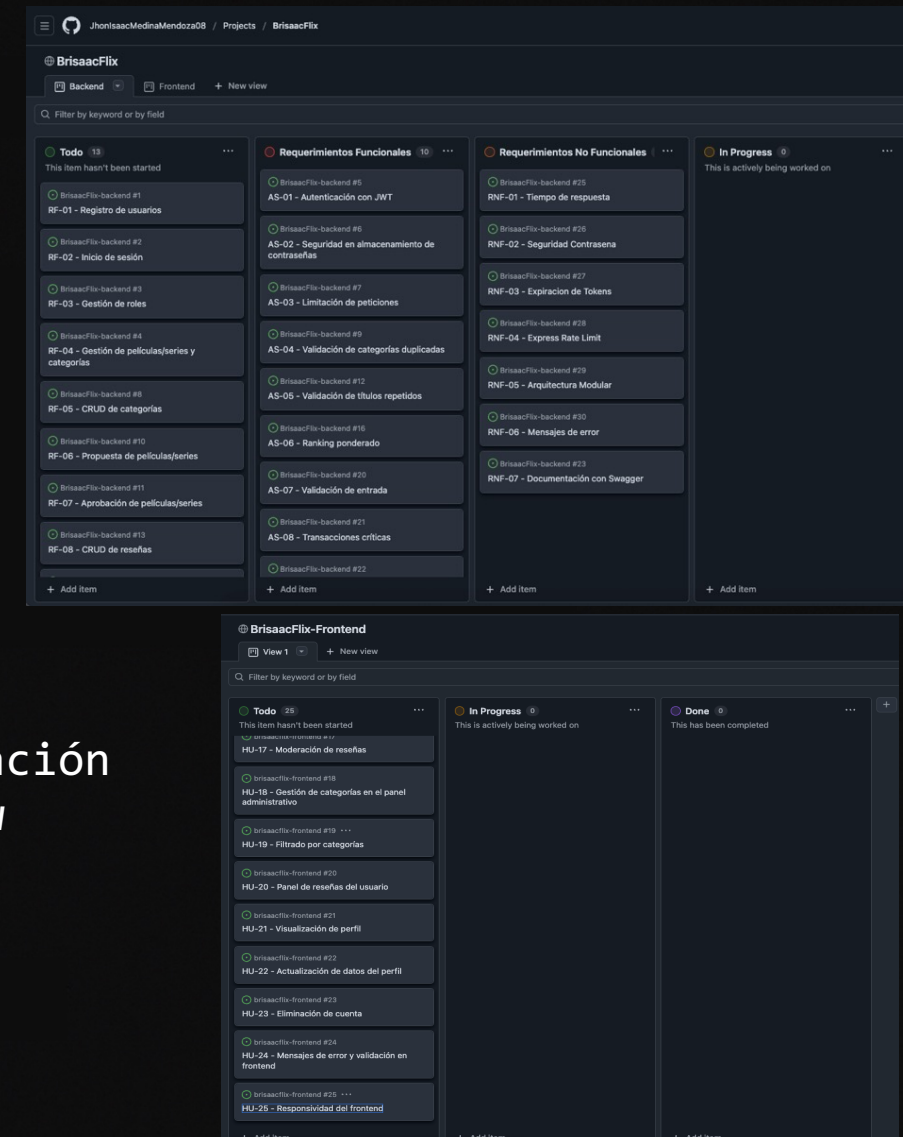
> Organización del equipo

Roles asignados:

- > Brian Suarez ➤ Product Owner
- > Isaac Medina ➤ Scrum Master
- > Brian Suarez y Isaac Medina. ➤ Developer

Metodología:

- > ✓ SCRUM aplicado con sprints por entregables
- > ✓ GitHub Projects como tablero de tareas
- > ✓ Historias de usuario con criterios de aceptación
- > ✓ Reuniones: Planning – Daily – Retro – Review





> Diseño del sistema

Principios SOLID aplicados:

S - **Single Responsibility Principle** (Responsabilidad Única)

O - **Open/Closed Principle** (Abierto/Cerrado)

L - **Liskov Substitution Principle** (Sustitución de Liskov)

I - **Interface Segregation Principle** (Segregación de Interfaces)

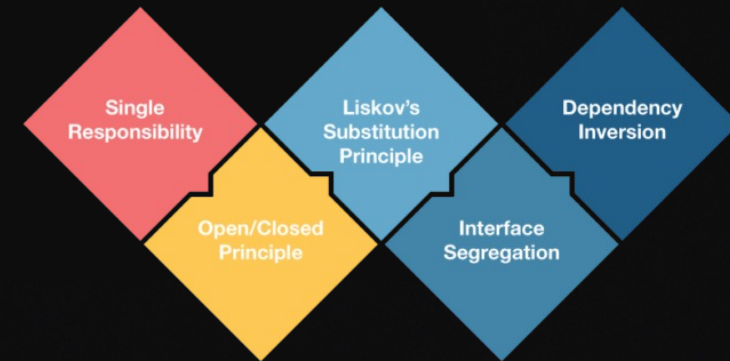
D - **Dependency Inversion Principle** (Inversión de Dependencias)

Patrones de diseño aplicados:

1. MVC (Model – View – Controller)

2. Repository Pattern

3. Factory Pattern





> Tecnologías y herramientas usadas Backend

Tecnología principal:

> Node.js 20.x

Base de datos:

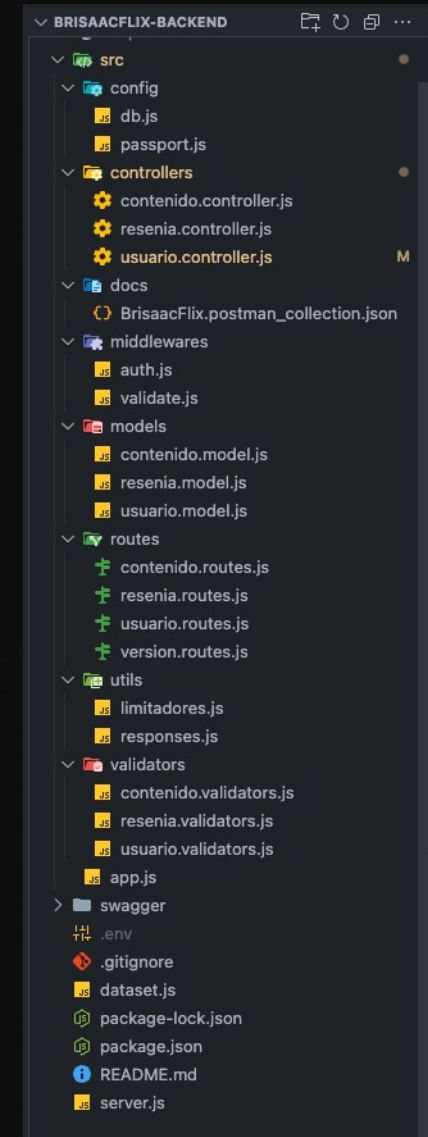
> MongoDB (driver oficial, sin Mongoose)

Librerías:

> "bcrypt": "^6.0.0", "cors": "^2.8.5", "dotenv": "^17.2.1",
> "express": "^5.1.0", "express-rate-limit": "^8.0.1", "express-validator": "^7.2.1",
> "js-yaml": "^4.1.0", "jsonwebtoken": "^9.0.2", "mongodb": "^6.18.0",
> "nodemon": "^3.1.10", "passport": "^0.7.0", "passport-jwt": "^4.0.1",
> "semver": "^7.7.2", "swagger-ui-express": "^5.0.1"

Estructura modular:

> /models /config /controllers /utils /middlewares /routes
/validators





> Tecnologías y herramientas usadas Frontend

Tecnología principal:

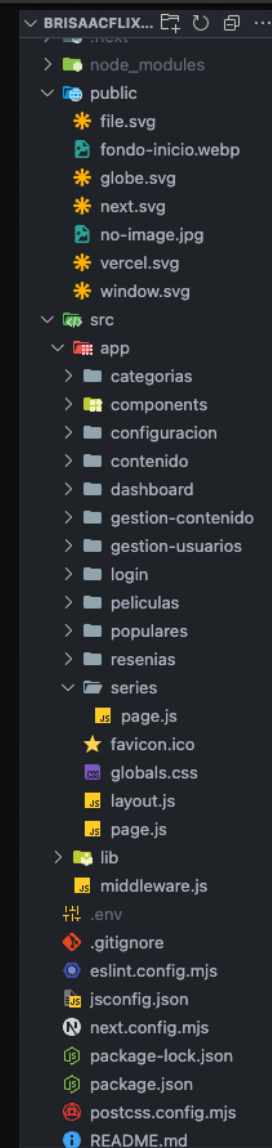
> React

Librerías:

> ""axios": "^1.11.0",
> "lucide-react": "^0.542.0",
> "next": "15.5.1",
> "react": "19.1.0",
> "react-dom": "19.1.0"

Estructura modular:

> /public /components /configuracion /dashboard /login





> Funcionalidad principal

- Explorar contenido: listado de películas y series con filtros y ranking.
- Reseñas y calificaciones: los usuarios pueden escribir, editar y valorar reseñas.
- Likes y dislikes: interacción entre usuarios sobre las reseñas.
- Gestión de catálogo: los administradores aprueban, eliminan películas y series (Gestión de contenido).
- Panel de usuario: cada usuario gestiona su perfil, reseñas, cuenta y solicita la aprobación de un contenido que él crea.

