

PopCornTime

Aplicación desarrollada por Miguel Antonio Peraza Peña



Miguel Antonio Peraza Peña



INTRODUCCIÓN

PopcornTime es una aplicación innovadora diseñada para gestionar tu biblioteca de películas, desarrollada en Flutter para garantizar una experiencia nativa en diversas plataformas. La necesidad de este proyecto radica en tener unificada para el control de contenido multimedia, aprovechando las ventajas de Flutter en términos de rendimiento y desarrollo multiplataforma. El software es relevante para usuarios que buscan una alternativa de código abierto y multiplataforma para el control de películas y series.

ANÁLISIS DEL PROBLEMA

2.a Problemática

Muchos usuarios enfrentan la limitación de aplicaciones de streaming que son exclusivas para ciertas plataformas o que no ofrecen una experiencia de usuario consistente. Además, las opciones disponibles pueden no ser de código abierto, lo que limita la capacidad de personalización y auditoría por parte de la comunidad.

2.b Clientes potenciales

Usuarios que buscan una aplicación de control de contenido multimedia que sea:

- Multiplataforma (disponible en Android, iOS, Linux, macOS, Windows y web).
- De código abierto, permitiendo personalización y contribuciones de la comunidad.
- Desarrollada con tecnologías modernas que aseguren un rendimiento óptimo y una experiencia de usuario coherente.



2.c Análisis DAFO

- Fortalezas:
 - Desarrollo en Flutter, lo que permite una base de código única para múltiples plataformas.
 - Código abierto, fomentando la colaboración y transparencia.
- Debilidades:
 - Dependencia de la comunidad para el mantenimiento y de una api para actualización del proyecto.
 - Posibles desafíos en la integración de funcionalidades específicas de cada plataforma.
- Oportunidades:
 - Creciente adopción de Flutter en la industria, lo que facilita la incorporación de nuevas funcionalidades y mejoras.
 - Demanda de aplicaciones de control multiplataforma y personalizables.
- Amenazas:
 - Competencia con aplicaciones de streaming ya establecidas y con grandes bases de usuarios.

2.d Monetización y beneficios

Aunque el proyecto es de código abierto, existen varias estrategias de monetización posibles:

- **Donaciones:** Implementar un sistema de donaciones para apoyar el desarrollo continuo.
- **Servicios premium:** Ofrecer características adicionales o contenido exclusivo a través de una suscripción.
- **Publicidad:** Integrar anuncios no intrusivos dentro de la aplicación.



DISEÑO DE LA SOLUCIÓN

3.a Tecnologías elegidas

El proyecto está desarrollado en Flutter, un framework de código abierto de Google para la creación de aplicaciones nativas compiladas para móvil, web y escritorio desde una única base de código. Esto permite una experiencia de usuario coherente y un desarrollo más eficiente.

Base de datos local: Se usa `sqflite` para almacenamiento de datos persistentes.

API externa: Se usa `http` o `dio` para obtener datos de una API como TMDB.

3.b Diagrama de clases

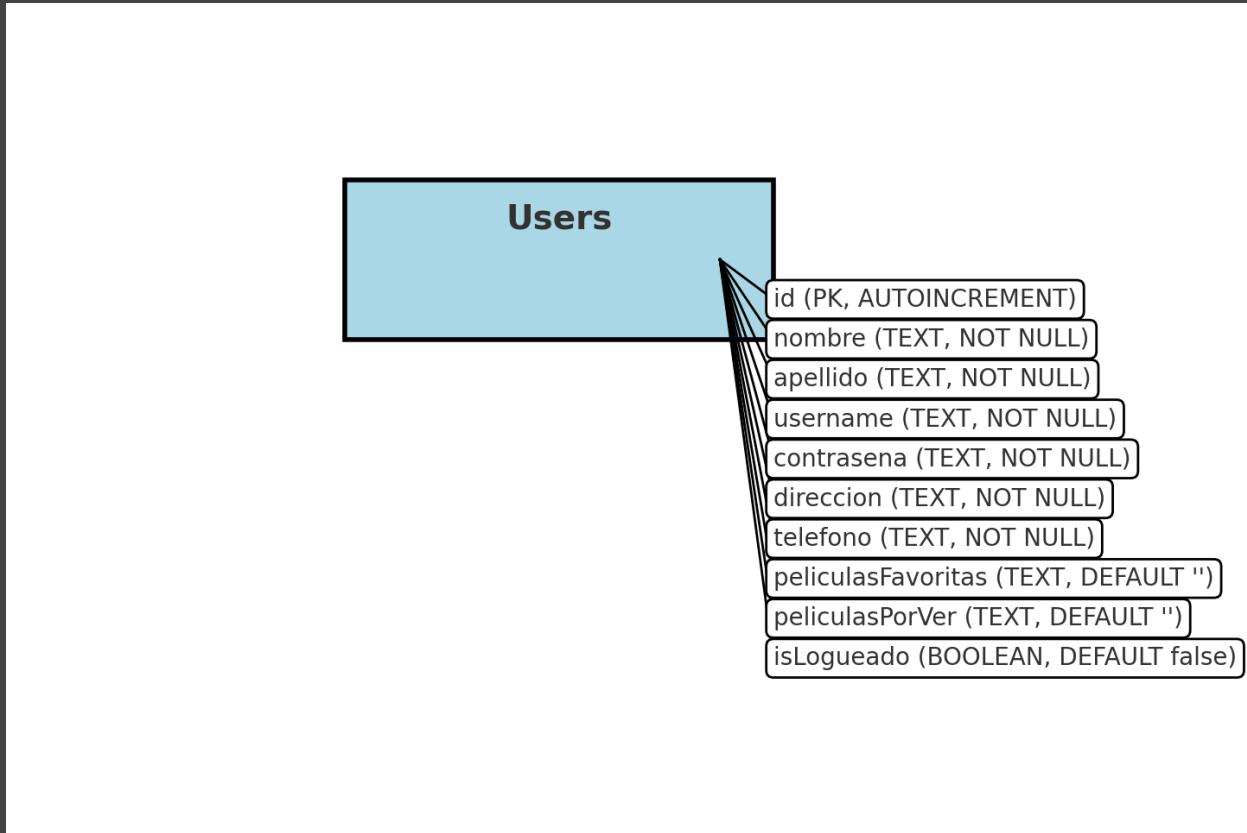
El código de desarrollo de la aplicación se encuentra en la carpeta lib, la cual a su vez se divide en cuatro subcarpetas

- **La carpeta lib/config:** contiene archivos esenciales para la configuración de la aplicación. En esta se puede encontrar la constante de al API key, el router de la aplicación, el theme el acceso a la base de datos y el tratamiento de datos para hacerlos más entendibles para los usuarios.
- **La carpeta lib/domain:** contiene los modelos de la aplicación, las interfaces y/o implementaciones que gestionan la comunicación entre la aplicación y las fuentes de datos y los componentes que encapsulan la lógica de negocio de la aplicación.
- **La carpeta lib/infrastructure:** contiene la fuente de datos, ya sean remotas o locales, componentes que actúan como intermediarios entre la aplicación y diferentes servicios o fuentes de datos, asegurando una integración coherente y eficiente y funciones y herramientas auxiliares que apoyan las operaciones de infraestructura, como la serialización/deserialización de datos, gestión de configuraciones, y manejo de dependencias.
- **La carpeta lib/presentation:** contiene componentes relacionados con la interfaz de usuario y la experiencia del usuario, las pantallas de la aplicación, los widgets usados por las distintas pantallas controladores de estado .

Estos componentes en la carpeta presentation son fundamentales para definir cómo los usuarios interactúan con la aplicación y cómo se presenta la información, asegurando una experiencia de usuario intuitiva y atractiva.



3.d Diagrama E/R



3.e Consideraciones técnicas

Integración de APIs: Para obtener información sobre contenido multimedia y gestionar el streaming.

Compatibilidad multiplataforma: Asegurarse de que la aplicación funcione correctamente en todas las plataformas objetivo, manejando diferencias específicas de cada una.



DOCUMENTACIÓN DE LA SOLUCIÓN

El código fuente está disponible en el repositorio mapp2308/PopcornTime. Para obtener acceso, es necesario clonar el repositorio utilizando Git:

```
git clone https://github.com/mapp2308/PopcornTime.git
```

Donde cada clase está comentada de forma que al principio de la clase se encuentra un resumen general de la función de la clase y luego en el código está comentado las partes de mayor importancia

ENLACES DE INTERÉS

Repositorio oficial de PopcornTime: [repositorio oficial](#).