



Instituto Superior de Engenharia de Lisboa  
Licenciatura Engenharia Informática e Multimédia

Ano letivo 2024/2025

## Desenvolvimento de Aplicações Móveis

Projeto Final

Turma 61D

Gabriel Rodrigues, A50787

Junho de 2025

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Interface</b>	<b>2</b>
<b>3</b>	<b>Clientes e Monetização</b>	<b>4</b>
<b>4</b>	<b>Planeamento</b>	<b>5</b>
4.1	Requisitos . . . . .	5
4.2	Casos de Utilização e Prioridade . . . . .	5
4.3	Modelo de dados . . . . .	6
<b>5</b>	<b>Implementação</b>	<b>8</b>
<b>6</b>	<b>Conclusões</b>	<b>10</b>

# Índice de figuras

1	Modelo conceptual de dados para a aplicação.	7
---	--	---

# 1 Introdução

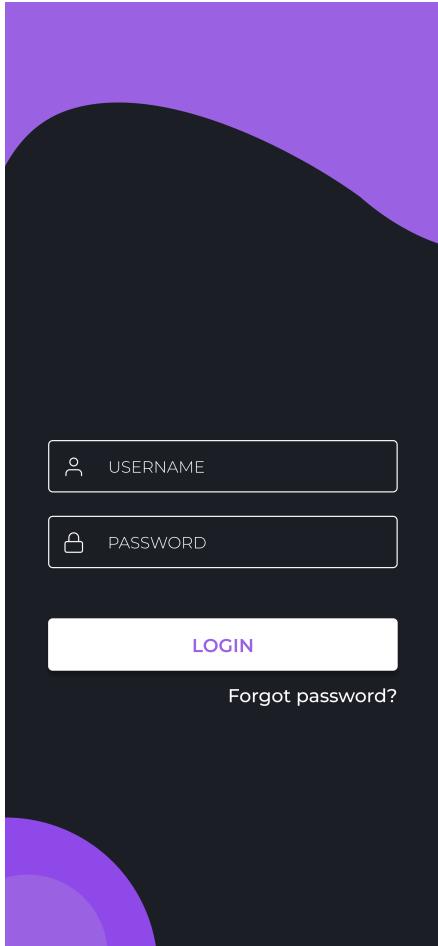
Com o crescente consumo de séries televisivas em plataformas de streaming, os utilizadores procuram cada vez mais partilhar opiniões e descobrir novos conteúdos através das recomendações de outros espetadores. Embora existam soluções bem estabelecidas para filmes, como o *Letterboxd*, verifica-se uma lacuna no mercado no que toca a plataformas dedicadas exclusivamente a séries.

A principal motivação para este projeto surge, portanto, da necessidade de uma aplicação móvel centrada em episódios e temporadas de séries, onde os utilizadores possam avaliar episódios individualmente, criar listas temáticas, seguir amigos e aceder a sugestões com base nos seus gostos.

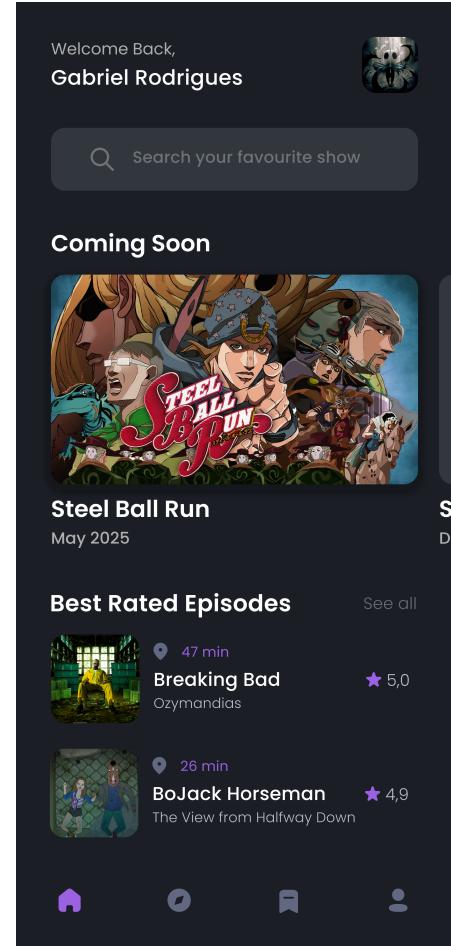
A aplicação destina-se a utilizadores que consomem séries televisivas com regularidade, em particular aqueles que valorizam a partilha de opiniões e a descoberta de novos conteúdos através de recomendações sociais. Incluem-se jovens adultos, utilizadores de plataformas de streaming e entusiastas de crítica audiovisual.

## 2 Interface

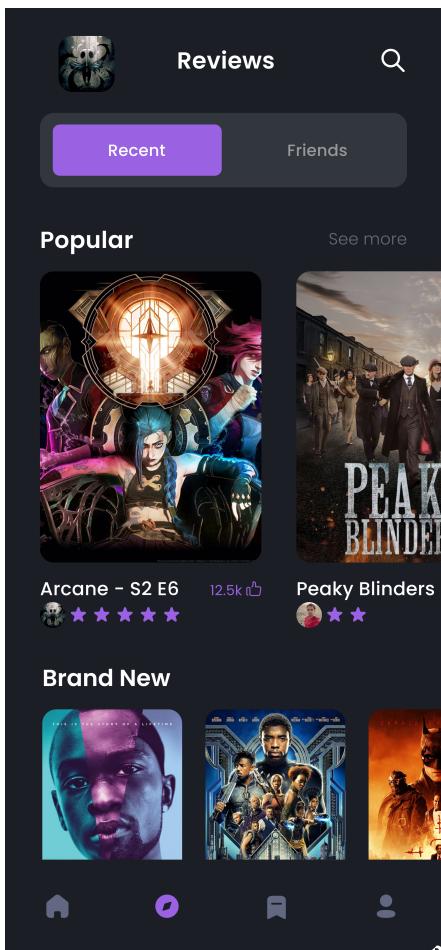
A interface foi desenhada para ser intuitiva, moderna e visualmente apelativa. Estes são os *Mockups* propostos:



Autenticação



Página inicial



Explorar

This screen shows a review for 'Arcane - S2 E6'. It includes the poster, title, subtitle ('Blisters and Bedrock'), year ('2024'), and a 2.5-star rating. Below the rating is a text input field with placeholder text: 'Ts wack asl istg it pmo sm 🤦‍♂️ RIP ISHA'. At the bottom are buttons for 'Post', 'Spoilers', and 'Friends only'.

Análise

This screen provides detailed information for 'Arcane - S2 E6'. It features a large poster of the show, the title, and a synopsis: 'Vi awakens to surprising news. An unsettling reunion isn't what it seems. Caitlyn uncovers the origins of Shimmer.' A 'Read More' link is present. Below the synopsis is a purple 'Review' button. Navigation icons are at the bottom.

Detalhes do Conteúdo

This screen shows a user profile for 'Gabriel Rodrigues'. It includes a profile picture, a 'Premium' badge, the name, the date joined ('July 14 2024'), and a follower count ('18'). Below the profile is a 'Playlists' section with a 'WatchList' entry for 'FORTNITE' and a 'Crack a Laugh' entry for 'SRIENDS'. At the bottom is a large green button with the text 'Breaking Bad' and a plus sign. Navigation icons are at the very bottom.

Perfil

### 3 Clientes e Monetização

A aplicação é direcionada principalmente a fãs de séries e televisão que valorizam a troca de opiniões, críticas e recomendações personalizadas. Os potenciais clientes incluem:

- **Utilizadores casuais:** Pessoas que acompanham algumas séries e procuram recomendações baseadas em avaliações populares ou de amigos.
- **Fãs dedicados:** Consumidores frequentes de séries que querem criar listas, partilhar críticas detalhadas e construir uma rede social de amigos com gostos semelhantes.
- **Críticos de entretenimento independentes:** Utilizadores que pretendem usar a plataforma como portefólio informal de análises e opiniões.

Para garantir a sustentabilidade da aplicação, foram identificadas as seguintes estratégias de monetização:

- **Plano Premium:** Um modelo freemium onde os utilizadores podem subscrever para desbloquear funcionalidades adicionais, como:
  - Criação ilimitada de listas personalizadas
  - Personalização do perfil (fundos e bordas do avatar)
- **Parcerias com serviços de streaming:** Possibilidade futura de incluir links afiliados ou recomendações patrocinadas com base nos hábitos de visualização dos utilizadores.

Estas abordagens visam equilibrar a experiência do utilizador com a viabilidade financeira da aplicação.

## 4 Planeamento

### 4.1 Requisitos

O levantamento de requisitos foi realizado com base em observação de plataformas semelhantes (como o *Letterboxd* e o *MyAnimeList*) e na análise do comportamento esperado dos utilizadores.

#### Requisitos Funcionais

- Registo e autenticação de utilizadores;
- Pesquisa e visualização de séries e episódios;
- Publicação de avaliações (com ou sem *spoilers*), associadas a episódios ou séries;
- Definição da visibilidade das avaliações (pública ou apenas para amigos);
- Consulta de críticas por popularidade, novidade e rede de amigos;
- Gestão de amizades (pedido, aceitação, remoção);
- Criação e gestão de listas temáticas personalizadas;
- Personalização do perfil (imagem de fundo e borda do avatar, para contas premium).

#### Requisitos Não Funcionais

- A aplicação deverá ser desenvolvida para dispositivos Android;
- Deverá apresentar tempos de resposta reduzidos nas operações principais (idealmente inferiores a 500ms);
- Os dados deverão estar persistidos em tempo real, com sincronização automática;
- A arquitetura da aplicação deve permitir escalabilidade e manutenibilidade.

### 4.2 Casos de Utilização e Prioridade

Caso de Utilização	Prioridade	Justificação
Registo/Autenticação de utilizador	Alta	Base de toda a experiência de utilização.
Pesquisa de séries/episódios	Alta	Essencial para explorar o conteúdo.
Publicar avaliação	Alta	Funcionalidade central da aplicação.
Adicionar amigos	Média	Amplifica a componente social da experiência.
Criar lista temática	Média	Permite personalização da navegação.
Editar perfil premium	Baixa	Funcionalidade exclusiva a contas pagas.

Tabela 1: Matriz de prioridade dos casos de utilização

### 4.3 Modelo de dados

A definição do modelo de dados é uma etapa crítica para garantir a robustez e escalabilidade da aplicação. Partindo dos requisitos funcionais identificados (cf. Secção 4.1), o sistema requer um conjunto de entidades capazes de representar utilizadores, séries, episódios e avaliações, assegurando ao mesmo tempo funcionalidades sociais (amizades) e personalização (listas de interesse, visibilidade de conteúdos).

De forma geral, o sistema é composto pelas seguintes entidades:

- **Utilizador (*User*):** representa o utilizador registado na plataforma. Armazena dados de identificação, preferências visuais e referências a amizades e listas criadas. Cada utilizador pode ter amigos e publicar críticas.
- **Série (*Show*):** corresponde ao conteúdo audiovisual de longa duração. Armazena dados descritivos, imagem de capa e géneros associados. Cada série está associada a múltiplos episódios.
- **Episódio (*Episode*):** representa uma unidade de conteúdo dentro de uma série. Inclui informações como o nome, duração, data de lançamento e descrição.
- **Avaliação (*Review*):** contém a opinião de um utilizador sobre um episódio ou uma série, composta por uma classificação numérica (0-5), uma descrição textual e uma opção de marcação de *spoiler*. Um aspeto relevante é a introdução do campo *visibility*, que permite ao utilizador escolher se a crítica é pública ou visível apenas para amigos.
- **Lista (*Playlist*):** permite ao utilizador agrupar séries de acordo com interesses pessoais. O número de listas disponíveis pode variar conforme o tipo de conta (normal ou premium).

A ligação entre episódios e séries é feita através de um identificador de série (`showId`) presente na entidade *Episode*. De modo semelhante, cada crítica possui uma referência ao conteúdo a que se refere, podendo ser um `episodeId` ou um `showId`, mas nunca ambos em simultâneo. Esta abordagem permite manter flexibilidade e coerência na estrutura dos dados.

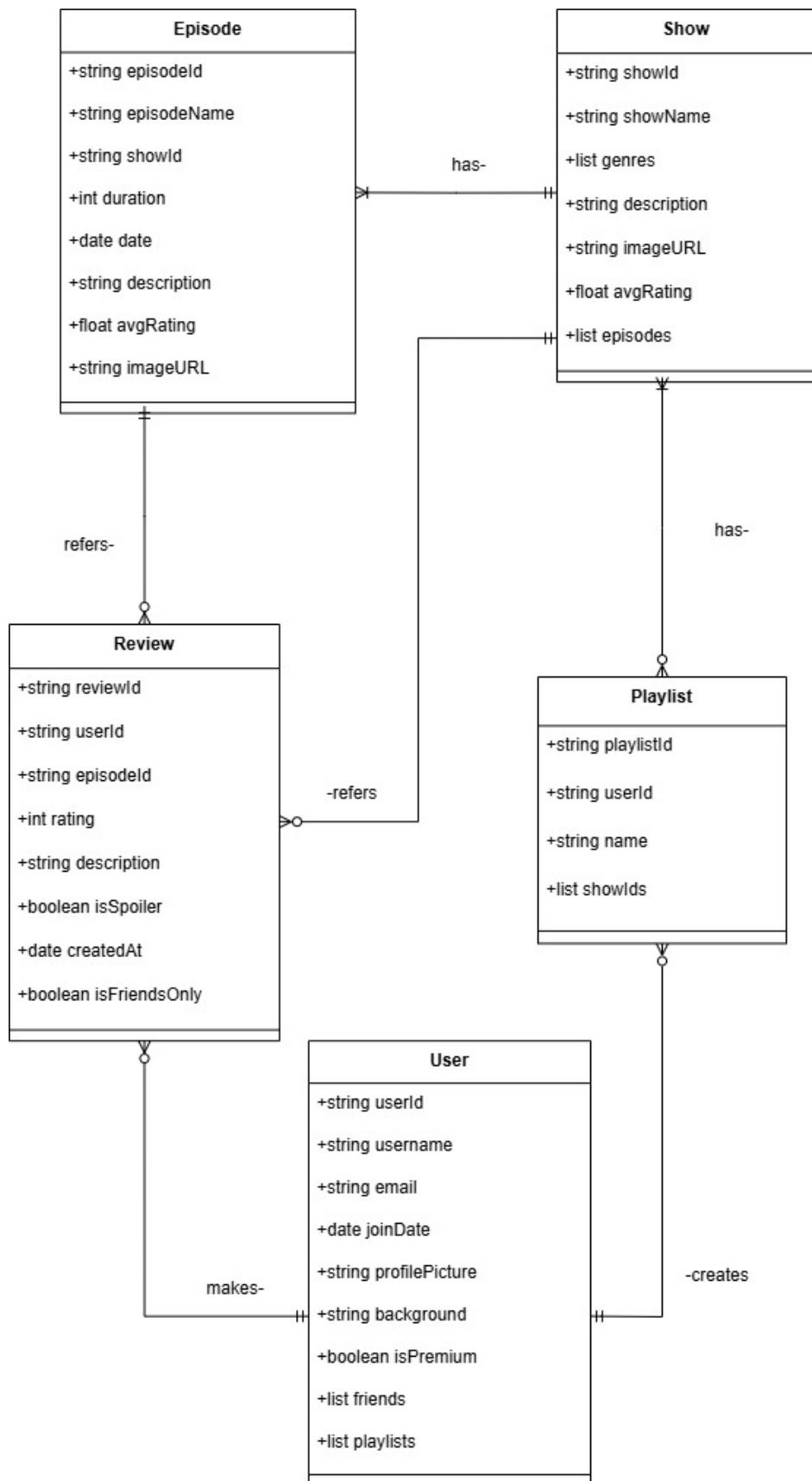


Figura 1: Modelo conceptual de dados para a aplicação.

## 5 Implementação

A implementação da aplicação foi desenvolvida com recurso à framework **Jetpack Compose**, que permite a construção de interfaces modernas e declarativas em Android. Foi adotada uma arquitetura baseada no padrão **MVVM (Model-View-ViewModel)**, promovendo uma clara separação de responsabilidades e facilitando a escalabilidade e testabilidade do projeto. Para suporte a base de dados e autenticação usou-se a Firebase da Google.

### Camadas da Arquitetura

- **Model:** Representa os dados da aplicação, incluindo entidades como `User`, `Playlist`, `Review`, entre outras. Estes dados são obtidos a partir do *backend* (Firebase Firestore e Firebase Storage).
- **ViewModel:** Responsável pela lógica de negócio e gestão de estado. Cada `ViewModel` expõe os dados e ações necessárias à UI através de `StateFlow` ou `MutableState`, garantindo que a interface reage automaticamente a alterações.
- **View (UI):** Composta por funções `@Composable` que consomem os estados fornecidos pelos `ViewModels` e apresentam os dados ao utilizador. Estas funções são reativas e recompõem-se automaticamente sempre que os dados mudam.

### Gestão de Estado e Reatividade

O estado da aplicação é mantido utilizando os mecanismos nativos do Compose, como `remember`, `mutableStateOf` e `collectAsState()`. Isto permite uma gestão eficiente e fluida do ciclo de vida da interface, sem necessidade de recorrer a chamadas manuais para atualizar a UI.

Como exemplo, na apresentação do perfil do utilizador, o carregamento da imagem de avatar é feito de forma assíncrona através da função `getDownloadUrlFromStorage`, que obtém o URL a partir do Firebase Storage. Este URL é então armazenado num estado reativo, garantindo que a imagem é apresentada assim que estiver disponível:

```
var resolvedUrl by remember { mutableStateOf<String?>(null) }
LaunchedEffect(downloadUrl) {
    if (!downloadUrl.isNullOrEmpty()) {
        getDownloadUrlFromStorage(downloadUrl) { url ->
            resolvedUrl = url
        }
    }
}
```

A abordagem reativa permite que a imagem se atualize automaticamente, sem necessidade de lógica adicional para recompor a UI.

### Integração com Firebase

A persistência de dados e ficheiros é feita com recurso ao Firebase Firestore (dados estruturados) e Firebase Storage (imagens de perfil, capas de playlists, etc.). A autenticação de utilizadores é

também assegurada pelo Firebase Auth. A aplicação foi desenhada para sincronizar alterações em tempo real, refletindo imediatamente atualizações no perfil, listas ou avaliações.

## Composição da Interface

Todas as ecrãs são compostos por funções `@Composable`, organizadas em colunas e linhas (`Column` e `Row`) e estilizadas com base na `Material3`. Elementos como botões, ícones, listas e imagens adaptam-se dinamicamente ao conteúdo e ao estado da aplicação.

A utilização de componentes personalizados, como cartões de playlist ou imagens com borda colorida (para contas premium), assegura consistência visual e modularidade na construção da UI.

## **6 Conclusões**

Este projeto permitiu dar resposta a uma lacuna existente no ecossistema digital: a ausência de uma plataforma dedicada à avaliação e descoberta social de séries televisivas. Através da combinação de funcionalidades de crítica detalhada (a episódios ou séries), listas personalizadas e uma componente social centrada em amizades, a aplicação oferece uma experiência de utilização completa, personalizada e orientada para o público-alvo identificado.