

Rapport Complet de Développement – Application FilmEra

1. Introduction

FilmEra est une application mobile développée avec Flutter permettant aux utilisateurs de découvrir, rechercher et gérer leurs films et séries favoris grâce à l'API TMDB. Ce rapport détaille les choix techniques, l'architecture, les difficultés rencontrées et les solutions apportées tout au long du projet.

2. Description fonctionnelle

L'application propose les fonctionnalités suivantes :

- Découverte de films et séries populaires
- Recherche par titre
- Consultation des détails d'un média (synopsis, casting, note, etc.)
- Ajout et gestion des favoris
- Navigation fluide entre les différentes sections

3. Architecture et choix techniques

L'architecture repose sur le pattern BLoC/Cubit pour la gestion d'état, assurant une séparation claire entre la logique métier et l'interface utilisateur. L'organisation du code est la suivante :

- `lib/cubits/` : gestion d'état (Cubits et états)
- `lib/models/` : modèles de données
- `lib/repositories/` : accès aux données et logique métier
- `lib/screens/` : écrans de l'application
- `lib/services/` : services externes (API TMDB)

L'UI est conçue pour être responsive et s'adapter à différentes tailles d'écran.

4. Packages et dépendances

Les principaux packages utilisés sont :

- `flutter_bloc` : gestion d'état réactive
- `http` : requêtes réseau
- `equatable` : comparaison d'objets
- `cached_network_image` : gestion du cache des images
- `sqflite` ou `hive` : stockage local des favoris
- `provider` : injection de dépendances (si utilisé)
- `url_launcher` : ouverture de liens externes

5. Problèmes rencontrés

- Intégration de l'API TMDb (authentification, pagination, gestion des quotas)
- Gestion des états de chargement et d'erreur dans l'UI
- Optimisation des performances lors du chargement d'images
- Adaptation de l'interface pour les petits écrans
- Gestion de la persistance des favoris

6. Solutions apportées

- Centralisation de la logique métier dans les Cubits pour une meilleure maintenabilité
- Utilisation de widgets dédiés pour l'affichage des états de chargement et d'erreur
- Mise en cache des images avec `CachedNetworkImage`
- Utilisation de `sqflite`/`hive` pour la sauvegarde locale des favoris

7. Gestion de projet

Le projet a été géré en suivant une méthodologie agile, avec des itérations courtes et des points de revue réguliers. Les tâches ont été suivies à l'aide d'un tableau Kanban (Jira).

8. Tests et validation

- Tests unitaires sur les Cubits et les repositories
- Tests manuels sur les principales fonctionnalités (recherche, ajout aux favoris, navigation)
- Validation sur différents appareils et émulateurs pour garantir la compatibilité

9. Conclusion et perspectives

L'application Filmera répond aux besoins initiaux en offrant une expérience utilisateur fluide et complète pour la découverte de films et séries. Des améliorations futures pourraient inclure :

- Ajout de notifications personnalisées
- Intégration d'un mode hors-ligne

- Amélioration de l'accessibilité
- Extension à d'autres plateformes (web, desktop)