

Estrutura Analítica do Projeto

Objetivo Geral

Desenvolver um aplicativo móvel para a plataforma Android utilizando o framework **ionic**, com integração a APIs REST externas para oferecer notícias personalizadas. O app permitirá que usuários busquem, visualizem e personalizem artigos de diversas categorias (como tecnologia, esportes e entretenimento), com base em uma API de notícias pública.

Objetivos Específicos

1. Implementar um sistema de autenticação seguro para acesso ao aplicativo.
2. Integrar o consumo de dados de uma API REST externa para exibição de notícias atualizadas.
3. Fornecer uma interface de usuário (UI) intuitiva e responsiva.
4. Oferecer funcionalidades de personalização, como filtros e favoritos.
5. Melhorar a experiência do usuário com notificações push e otimização de performance.

Funcionalidades Principais

1. Autenticação de Usuário

- Implementação de login e cadastro.
- Integração com serviços de autenticação (Firebase Authentication ou JWT).
- Recuperação de senha via e-mail.

2. Integração com API de Notícias

- Consumo de uma API REST (como NewsAPI) para buscar notícias em tempo real.

- Parametrização de requisições com base em filtros selecionados pelo usuário.

3. Interface de Usuário Responsiva

- Design e prototipação com componentes nativos do Ionic.
- Implementação de navegação eficiente entre telas, utilizando roteamento nativo.

4. Filtros e Personalização

- Seleção de categorias de interesse (ex.: esportes, política, entretenimento).
- Configuração de preferências para ajustar o layout ou a ordem de exibição.

5. Exibição de Notícias

- Listagem das notícias em formato compacto (título, imagem e resumo).
- Visualização detalhada de uma notícia, incluindo conteúdo completo e fonte.

6. Favoritos

- Funcionalidade para salvar e gerenciar notícias favoritas.
- Armazenamento local (cache) ou em banco de dados remoto.

7. Notificações Push

- Implementação de alertas para novas notícias em categorias selecionadas.
- Integração com serviços como Firebase Cloud Messaging (FCM).

8. Cache e Otimização de Performance

- Armazenamento de dados no dispositivo para acesso offline.
- Minimização de requisições à API, reduzindo latência e melhorando a eficiência.

Estrutura Técnica

Tecnologias Utilizadas

- **Frontend:** Ionic Framework (HTML, CSS, TypeScript).
- **Backend:** Integração com APIs REST externas.
- **Banco de Dados:** Firebase Realtime Database ou Firestore (opcional para favoritos).
- **Autenticação:** Firebase Authentication ou JWT.
- **Notificações:** Firebase Cloud Messaging.

Arquitetura

- **Modelo de Consumo de Dados:** RESTful APIs.
- **Gestão de Estados:** Utilização de frameworks de gerenciamento como RxJS.

Resultados Esperados

- Entrega de um aplicativo funcional, seguro e responsivo.
- Experiência personalizada para o usuário, com acesso rápido e confiável a notícias.
- Retenção de usuários por meio de notificações relevantes e desempenho aprimorado.