Ana Clara  
Bruno  
Igor  
João Nascimento  
Pedro Cruz

Projeto Integrador 5º Semestre: Media Suggester

* **Introdução**

O aplicativo “Media Suggester” é uma plataforma de recomendação de Filmes e Séries desenvolvida em Flutter para dispositivos Android. Ele oferece uma experiencia intuitiva e prática de recomendação de conteúdo para os usuários assistirem com base em seus gostos.

Principais Funcionalidades:

* Recomendação de Filmes e Séries: Os usuários recebem recomendações de filmes e séries com base em seus gostos cadastros e favoritos, utilizando inteligência artificial.
* Busca de Filmes e Séries: Os usuários podem pesquisar as informações de um filme ou série específica.
* Cadastro de Favoritos: Os usuários podem cadastrar filmes e séries como seus favoritos, e podem ser utilizados para futuras recomendações.
* Publicar Reviews: Os usuários podem deixar sua review em um filme ou série, dando sua nota e escrevendo um texto sobre o que achou do conteúdo, essa review ficará exposta para os outros usuários.
* Ler reviews de outros usuários: Os usuários podem ler as reviews deixadas por outros usuários.

Tecnologias Utilizadas:

* Flutter: O aplicativo foi desenvolvido em Flutter, permitindo a criação de uma única base de código para Android e outras plataformas.
* Firebase: O aplicativo utiliza o Firebase para armazenamento de dados.
* Google: O aplicativo utiliza a API do Google para autenticação.
* TMDB: O aplicativo utiliza a API do The Movie DataBase para carregar as informações dos filmes e séries.
* Gemini: Inteligência Artificial do Google utilizada para gerar as recomendações
* .Net: Framework de C# utilizada para desenvolver a Api que consome o Gemini
* **Metodologia**

...

Emulador Android/iOS:

* Utilizado para simular o ambiente de execução do aplicativo em dispositivos virtuais.
* Permite testar o aplicativo em diferentes versões do sistema operacional e cenários de uso.

MobSF:

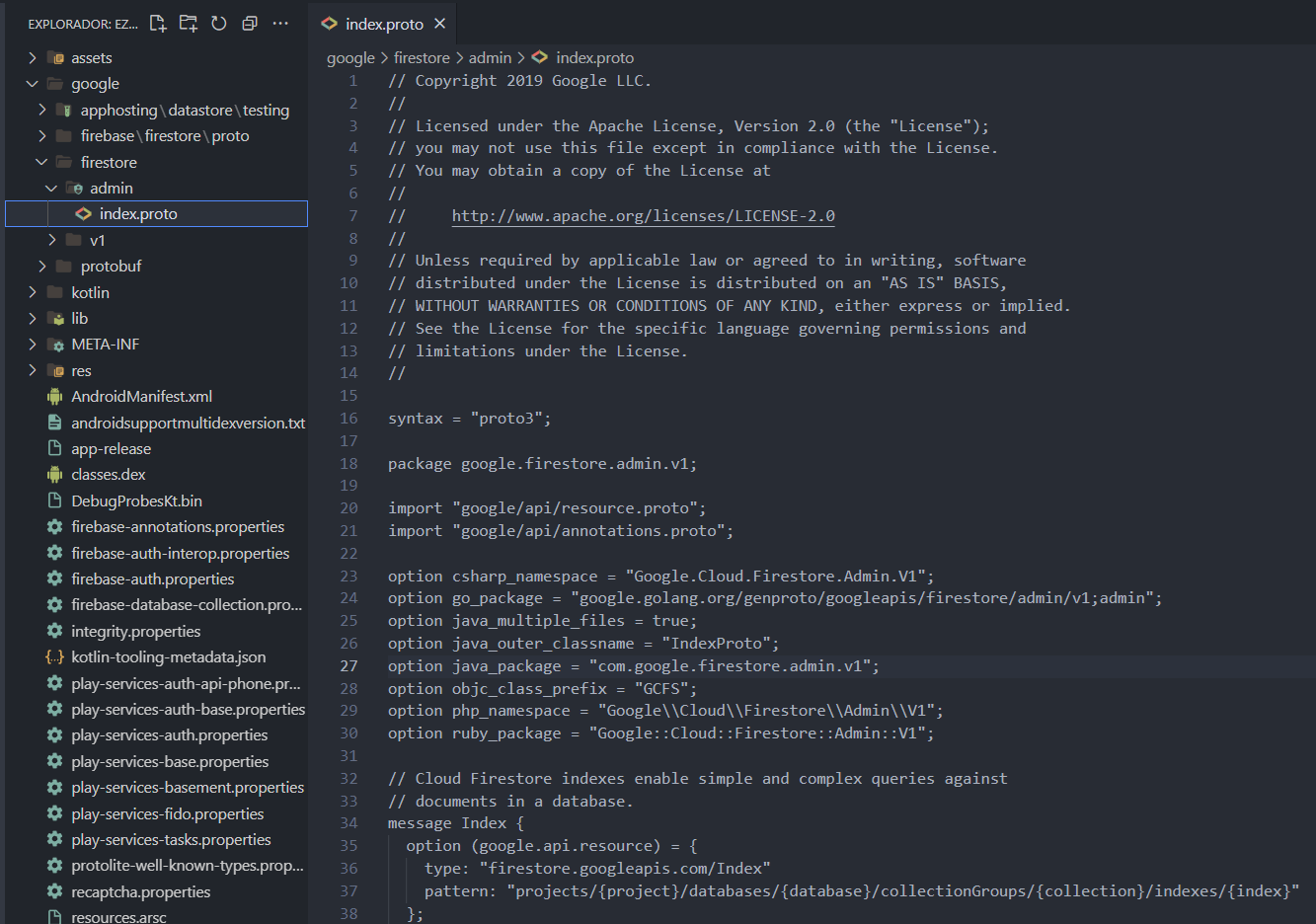
* Framework open-Source utilizado para executar Testes automatizados.
* Permite a execução de testes automatizados em busca de possíveis vulnerabilidades presentes no sistema
* **Abordagem**

...

* **Testes de Vulnerabilidade**

Vulnerabilidade 1: Extração do arquivo .APK

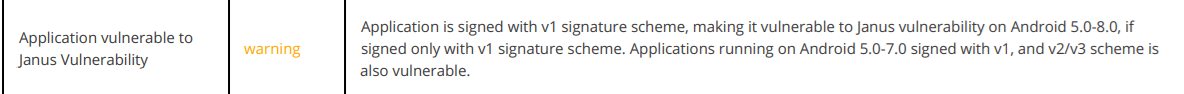
* Foi extraído o arquivo .APK (aplicativo compilado para Android), com a intenção de encontrar a chave da API do TMDB ou dados para a integração com o Firebase. Porém nenhum dos dois foi encontrado.



* Caso as chaves fossem encontradas facilmente com uma extração seria uma grande vulnerabilidade, pois qualquer pessoa teria acesso ao nosso banco de dados no Firebase, ou a nossa API do TMDB.

Vulnerabilidade 2: Esquema de Assinatura V1(MobSF)

* A aplicação está assinada utilizando o esquema de assinatura v1 o que permite que a mesma seja instalada em dispositivos que possuem o Android 5.0-8.0 e abaixo



* Este esquema de assinatura abre portas para uma vulnerabilidade conhecida como Janus que permite a inserção de código malicioso em uma aplicação sem que a assinatura da mesma seja alterada, assim podendo ser distribuído sem que o Android perceba qualquer alteração maliciosa
* Solução: Assinar a aplicação com um esquema de assinatura superior

Vulnerabilidade 3: Api em C# não possui autenticação

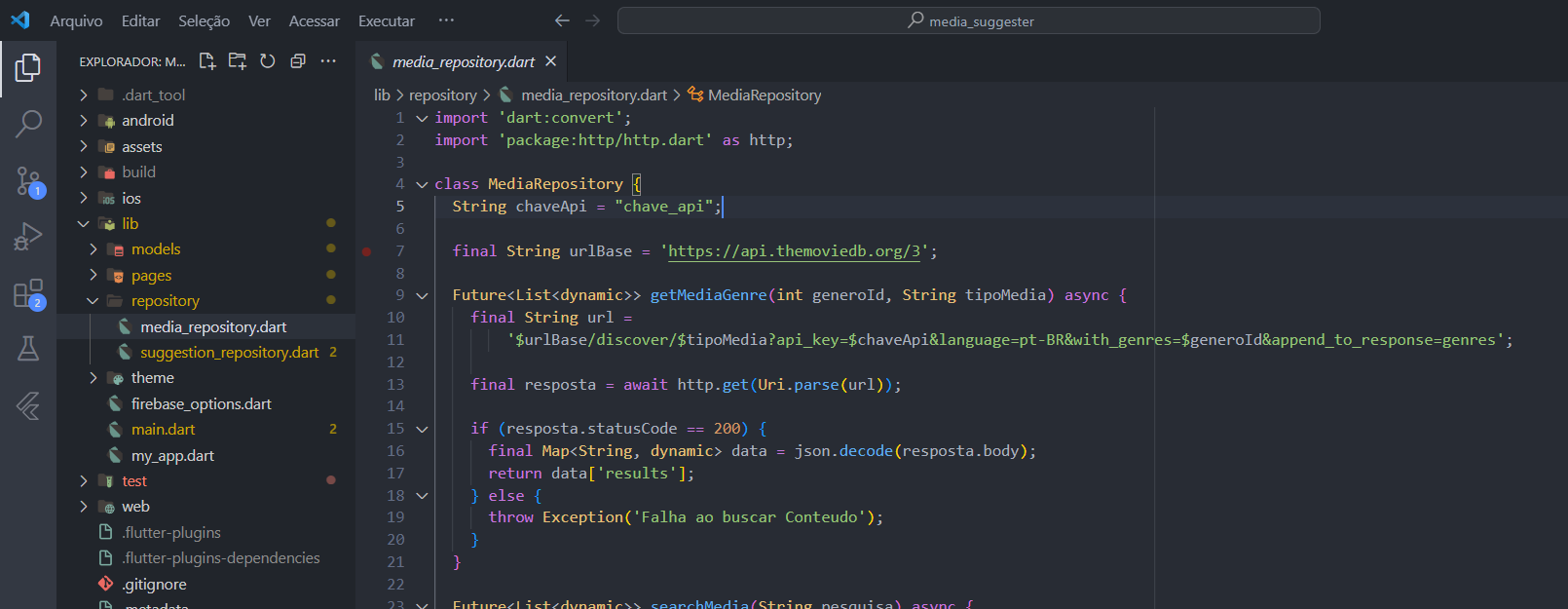
* A api não possui autenticação de usuário permitindo com que qualquer pessoa seja capaz de enviar requisições para a mesma fazendo com que o limite de uso seja atingido

(Inserir Imagem)

* Solução: Implementar autenticação de usuário

Vulnerabilidade 4: Projeto não possui .env

* O projeto não possui um arquivo que contem a chave da api, sendo necessário inserir manualmente(hardcoded) a chave no arquivo.
* O nosso projeto está armazenado no GitHub fazendo com que se algum membro do grupo fazer um commit com a chave a mesma ficara publica para qualquer usuario



Solução: Adicionar um arquivo .env ao projeto para armazenar a chave da api

* **Conclusão**

...