

CINE COLLECTION: UM APLICATIVO PARA RECOMENDAÇÕES DE FILMES

Letícia Ellen Bernardo¹
Kleber de Oliveira Andrade²

Resumo

Com o crescimento do mercado de dispositivos móveis, atualmente, o celular se tornou o meio mais popular dos brasileiros se conectarem a internet. Hoje em dia, os celulares são utilizados para diversas funções, como socializar, buscar informações e suprir passatempos, dentre eles, o mais comum é assistir a vídeos e filmes. Por meio de uma análise de viabilidade foi detectado que muitas pessoas utilizam meios manuais para lembrar-se de filmes e para procurar por novos filmes. Por este fato, o trabalho apresentado documenta o desenvolvimento de um aplicativo de filmes, *Cine Collection*, que faz uso de um sistema de recomendação de filtragem colaborativa, com a finalidade de apresentar recomendações personalizadas aos usuários que buscam por novos filmes. O trabalho apresenta, primeiramente, um estudo sobre os sistemas de recomendação e, em seguida, o processo de desenvolvimento do aplicativo utilizando diversos recursos, como, a base de dados de filmes gratuita (TMDB), recursos do Firebase, como banco de dados e autenticação do Firebase com login do Facebook, entre outros, como também, apresenta os testes realizados.

Palavras-chave: Inteligência artificial. Engenharia de software. Dispositivos móveis. Aplicativos.

Abstract

With the growth of the mobile market, today, smartphones have become the most popular way for Brazilians to connect to the Internet. Nowadays, smartphones are used for various functions, such as socializing, searching for information and supplying hobbies, among which the most common is watching videos and movies. Through a feasibility analysis it has been found that many people use manual means to remember movies and to search for new movies. For this reason, the presented work documents the development of the Cine Collection movie application that uses a collaborative filtering system, with the purpose of presenting personal recommendations to users looking for new movies. The work presents, first, a study on the recommendation systems and then the process of developing the application using several resources, such as The Movie Database (TMDB), Firebase features such as real-time database and Firebase authentication with Facebook login, among others. As well, it presents the tests performed.

Keywords: Artificial intelligence. Software engineering. Mobile. Applications.

Introdução

A origem de novas tecnologias e o aumento da acessibilidade da internet contribuiu para o crescimento exponencial no volume de dados, e seu tratamento em tempo hábil tornou-se inexecutável. Um exemplo desse crescimento pode ser encontrado na Pesquisa Nacional Por Amostra de Domicílios (Pnad) divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE)³, que apontou que, entre os anos 2013 e 2014, o índice de pessoas conectadas subiu de 48% para 54,9% do total, isto é, cerca de 36,8 milhões de casas com acesso à internet (PNAD,

¹ Tecnóloga em Análise e Desenvolvimento de Sistemas pela Fatec Americana. E-mail: leticiaellenbernardo@gmail.com

² Docente da Fatec Americana. E-mail: pdjkleber@gmail.com

³ Informações disponíveis em <http://www.ibge.gov.br>.

2014). O celular se tornou o meio mais popular dos brasileiros de se conectar à internet, ultrapassando até mesmo os computadores pessoais. Das 36,8 milhões de casas conectadas, 29,6 milhões dispõem de um telefone móvel para se conectar, representando 80,4% do total (PNAD, 2014).

O crescimento do celular foi tão notável que o Android, sistema operacional mais utilizado em *smartphones* ultrapassou o Windows, sistema operacional mais utilizado em microcomputadores pessoais, em número de dispositivos conectados à internet (STATCOUNTER, 2017). Os dispositivos móveis estão cada vez mais presentes na vida das pessoas, seja na própria casa, na rua ou até mesmo no trabalho. Suas funções variam entre socializar com outras pessoas, buscar notícias e informações, acessar sites, fazer compras, planejar atividades, entre outros. De acordo com uma pesquisa feita pelo Comitê Gestor da Internet no Brasil (CGI.BR)⁴, dentre essas atividades relacionadas a entretenimento, as mais comuns são assistir a filmes e vídeos em sites com 58% do total de usuários de Internet. O número se torna mais expressivo nas faixas de usuários entre 10 a 15 anos (68% e 66%, respectivamente) e entre 16 a 24 anos (67% e 69%) (CGI.BR, 2014).

Uma pesquisa divulgada pela Conectaí apontou que, um em cada três internautas brasileiros (34%) assistem a filmes, pelo menos, uma vez por semana, mostrando que assistir a filmes é um dos passatempos mais comuns entre os brasileiros (CONNECTAÍ, 2017). A fim de suprir as necessidades deste passatempo, existem variados serviços na web de streaming⁵ de filmes, sites de filmes, organizadores cinéfilos e sistemas de recomendação de filmes. Atualmente, os softwares mais populares que possuem recomendação de filmes são Netflix⁶, Google Play Filmes⁷, Amazon Prime Video⁸ e Youtube⁹. Esses serviços oferecem diversas funcionalidades e valores diferentes, fornecendo recomendações de filmes direta ou indiretamente (quando o usuário não se dá conta de estar recebendo uma recomendação) para seus usuários. No entanto, diversos usuários têm dificuldades para encontrar novos filmes que os agradam nesta abundância de dados. A Tabela 1 resume as principais diferenças entre esses serviços e o Cine Collection – aplicativo desenvolvido neste trabalho.

Este trabalho tem como objetivo geral apresentar o processo de desenvolvimento de um aplicativo utilizando a técnica de filtragem colaborativa como o motor de recomendação de filmes. Também pretende fundamentar que o conceito de sistemas de recomendação pode ser aplicável até mesmo no desenvolvimento de uma aplicação em um dispositivo móvel. Quanto aos objetivos específicos são: realizar um estudo sobre os sistemas de recomendação, suas características e suas técnicas; desenvolver e aplicar a técnica de recomendação baseada em filtragem colaborativa em um aplicativo de recomendação de filmes; utilizar a metodologia cascata e expor os procedimentos realizados; realizar experimentos e discutir os resultados; apresentar as conclusões finais e exibir as possibilidades para trabalhos futuros.

1 Sistemas de recomendação

⁴ CGI.BR. Para saber mais acesse: <<http://www.cgi.br>>.

⁵ *Streaming* é uma tecnologia de transmissão simultânea de dados de áudio e vídeo por meio da rede. Através do *streaming* não há necessidade de fazer o download do conteúdo, pois, na medida que, o software de *streaming* recebe os dados, eles são arquivados temporariamente na máquina e transmitidos ao usuário.

⁶ Netflix. Disponível em: <<https://www.netflix.com>>. Acesso em: 22/04/2017.

⁷ Google Play Filmes. Disponível em: <https://play.google.com/store/movies?hl=pt_BR>. Acesso em: 22/04/2017.

⁸ Amazon Prime Video. Disponível em: <<https://www.primevideo.com>>. Acesso em: 22/04/2017.

⁹ Youtube. Disponível em: <<https://www.youtube.com>>. Acesso em: 22/04/2017.

Os sistemas de recomendação (SR) são uma subárea de aprendizagem de máquina (AM), do inglês, *machine learning*, e que tem como objetivo fornecer sugestões para um usuário (IBM, 2017). Essas sugestões relacionam-se com vários processos de decisões humanas como: qual item comprar, qual notícia ler ou até mesmo qual filme assistir. Pode-se dizer que, o sistema de recomendação é um software que antecipa as necessidades do usuário antes mesmo que ele precise (RICCI *et al*, 2011).

Em geral, os sistemas de recomendação têm como objetivo fornecer as recomendações a níveis computacionais baseando-se nas preferências que coleta do usuário.

As recomendações focam normalmente em um tipo específico de itens que são considerados termos genéricos para designar o que o sistema recomenda aos seus utilizadores, podendo assumir propósitos bem distintos, tais como, livros, filmes, notícias, vídeos, anúncios, links patrocinados ou produtos de uma loja virtual (RICCI *et al*, 2011). Os itens recomendados tentam atender as necessidades de um usuário, ou seja, um bom sistema de recomendação tenta se adequar ao utilizador.

A capacidade de aprender e identificar as necessidades individuais de cada usuário é denominado Personalização em Sistemas de Recomendação. Os sistemas que possuem esta característica armazenam uma representação interna das preferências de cada usuário que é chamada de perfil do usuário. O objetivo da personalização é gerar recomendações customizadas para cada usuário, baseando-se na representação de seu perfil.

Já os Sistemas de Recomendação Não Personalizados, embora se apoiem no processo de recomendação, não são capazes de identificar as necessidades dos usuários individualmente. O papel deste tipo de sistema é fazer a coleta e a distribuição das recomendações de forma generalizada para todos os usuários.

Tradicionalmente, os Sistemas de Recomendação podem ser classificados em três principais técnicas:

- i. **Filtragem baseada em conteúdo:** essa técnica consiste em gerar automaticamente palavras-chave/ termos dos conteúdos dos itens e comparar esses termos com os termos de interesse do usuário a fim de encontrar itens que o usuário possa ter mais interesse (SILVA, 2014).
- ii. **Filtragem colaborativa:** utiliza as avaliações de itens feitas pelos “X” usuários e as compara com as avaliações realizadas por um “Y” usuário para gerar a ele, cria-se uma lista de classificados de itens em potencial (SAMPAIO, 2006).
- iii. **Filtragem híbrida:** combinam as duas técnicas citadas acima, com a finalidade de eliminar as fraquezas de cada uma (REATEGUI, 2005).

Este trabalho utilizou-se da filtragem colaborativa para recomendar filmes e por isso, somente esta abordagem será explicada com detalhes.

1.1 Filtragem colaborativa

A filtragem colaborativa (FC) foi criada para solucionar os pontos em aberto da filtragem baseada em conteúdo e se diferencia por não exigir conhecimento do conteúdo dos itens. Sua essência é identificar perfis semelhantes, ou seja, que tenham avaliado uma quantidade relevante de itens de modo semelhante e desta forma, apresentar uns aos outros os itens distintos bem avaliados por cada um.

Em um sistema de FC, o perfil do usuário consiste em um vetor de itens avaliados. As avaliações feitas pelo usuário podem ser binárias (0 ou 1) como, por exemplo, favoritar ou não um filme. Também, é possível que a avaliação seja representada por um número real que indica o grau de preferência do usuário como classificar um filme entre 0 a 5 estrelas.

A técnica de FC pode ser dividida em duas categorias: baseado em memória e baseado em modelo (BOGERS, 2009).

- i. Os algoritmos baseados em memória foram os primeiros a serem desenvolvidos e por processarem as informações na memória são chamados de preguiçosos (*Lazy Learning*). Este método concentra-se no cálculo da similaridade entre usuários ou entre itens. São mais fáceis de implementar e apresentam bons resultados. Mas podem apresentar problemas quando há poucos itens ou poucos usuários semelhantes (SILVA, 2014).
- ii. Os algoritmos baseados em modelo foram criados com o objetivo de tentar aprimorar o desempenho dos sistemas de recomendação. Incorporam conceitos de aprendizado de máquina e de mineração de dados. Um exemplo são as redes neurais bayesianas. Conforme Silva (2014), estes algoritmos possuem como propriedade “a criação de um modelo preditivo baseado nas avaliações dos usuários como fase preliminar as recomendações”. O uso de algoritmos de Aprendizado de Máquina para a construção do modelo torna a etapa de treinamento mais árdua, por causa disto, estes algoritmos são conhecidos como “impacientes” (*Eager Learning*). Porém, esse treinamento é compensado durante a tarefa de recomendação.

A técnica de FC também pode buscar a similaridade entre dois itens ou dois usuários com a finalidade de encontrar uma lista de usuários ou itens mais similares.

1.2 Filtragem colaborativa usuário-usuário

A filtragem colaborativa usuário-usuário possui como objetivo recomendar para um usuário alvo, os itens que outros usuários, com gostos similares a ele, gostaram no passado. A similaridade entre dois usuários é calculada com base em seu histórico de avaliações. O princípio desta abordagem é que a nota do usuário alvo a para um item i é provavelmente similar à nota recebida por outro usuário b , mede-se a e b avaliaram itens no passado de forma semelhante.

Para construir um FC usuário-usuário de uma forma genérica o sistema deve executar as seguintes etapas: i) criar representações, ii) encontrar usuários “vizinhos”, iii) selecionar itens recomendáveis e iv) recomendar a lista de itens com as melhores notas.

1.3 Representação

A representação em um sistema de FC é uma matriz de utilidade $m \times n$, onde m representa os usuários e n representa os itens numa perspectiva histórica. Numa matriz usuário-item r que, por exemplo, represente os filmes favoritados, o elemento r_{ij} , pode apresentar o valor igual a 1, se o usuário favoritou ou igual a 0, se o usuário não favoritou. A Tabela 2 mostra um exemplo de uma matriz usuário-item.

Tabela 1 – Matriz usuário-item.

| | Interstellar | Batman: O Cavaleiro das Trevas | A Origem | Perdido em Marte | Procurando Nemo |
|---------|--------------|--------------------------------|----------|------------------|-----------------|
| João | 1 | | 1 | 1 | |
| Maria | | 1 | | | 1 |
| Letícia | 1 | 1 | 1 | | 1 |

| | | | | | |
|------|---|--|---|---|--|
| José | 1 | | 1 | 1 | |
|------|---|--|---|---|--|

Fonte: Elaborado pelos autores.

1.4 Encontrar usuários vizinhos

Após a composição da representação, a próxima etapa, e uma das mais importantes em FC, é identificar os usuários “vizinhos” do usuário alvo. O algoritmo para encontrar os possíveis vizinhos pode ser dividido em duas partes: (1) calcular a medida de proximidade (ou similaridade) e (2) selecionar a “vizinhança”.

Um dos métodos mais utilizados, e que foi utilizado neste projeto, para calcular a similaridade entre dois usuários e, conseqüentemente descobrir a vizinhança é o Coeficiente de Correlação de Pearson.

O Coeficiente de Correlação de Pearson (REATEGUI, 2005) é normalmente aplicado quando as avaliações dos usuários estão dispostas em uma escala numérica discreta, normalmente entre 1 até 5 ou 1 a 10. Por exemplo, avaliações com estrelas para filmes em uma escala de 1 a 5, que podem representar valores entre “ruim” e “excelente”. A Equação 1 apresenta a Correlação de Pearson (REATEGUI, 2005):

$$corr_{ab} = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2 \sum_i (r_{bi} - \bar{r}_b)^2}} \quad (1)$$

Sendo o $corr_{ab}$ a correlação do usuário alvo a com um determinado usuário b ; r_{ai} é a avaliação que o usuário ativo a atribuiu para o item i ; r_{bi} é a avaliação que o usuário ativo b atribuiu para o item i ; \bar{r}_a é a média de todas as avaliações do usuário ativo a em comum com o usuário b ; \bar{r}_b é a média de todas as avaliações do usuário ativo b em comum com o usuário a . Note que é necessária mais de uma avaliação em comum para que o índice seja útil. Ao final, os resultados são considerados pesos e variam entre 1 para similaridade total e -1 para total dissimilaridade. Um exemplo da realização do cálculo da Equação 1 é apresentado a seguir de acordo com as notas apresentadas na Tabela 3.

Tabela 2 - Matriz de notas.

| ID | Usuário | Filmes | | | |
|----|---------|--------|---------------|--------------------|---------------------|
| | | Matrix | Clube da Luta | O Jogo da Imitação | A Viagem de Chihiro |
| 1 | João | 4 | 0 | 5 | 5 |
| 2 | Maria | 4 | 1 | 3 | 5 |
| 3 | José | 3 | 0 | 2 | 4 |
| 4 | Amelie | 4 | 4 | 0 | 0 |
| 5 | Pedro | 2 | 1 | 3 | 5 |

Fonte: Elaborado pelos autores.

Dado dois usuários João e Pedro, o primeiro passo é calcular todas a média de todas as avaliações entre dois usuários. Sendo \bar{r}_a a média de todas as avaliações de filmes do usuário A (João):

$$\frac{4 + 0 + 5 + 5}{4} = 3,5 \quad (2)$$

Sendo que, \bar{r}_b é a média de avaliações de filmes do usuário B (Pedro):

$$\frac{2 + 1 + 3 + 5}{4} = 2,75 \quad (3)$$

Onde, r_{ai} é a avaliação que o usuário A deu para um item, neste caso, um filme e r_{bi} é a avaliação que o usuário B deu para o mesmo item. Sendo assim:

$$\begin{aligned} corr_{1,5} &= \frac{\sum_i (4 - 3,5) * (2 - 2,75) + (0 - 3,5) * (1 - 2,75) + (5 - 3,5) * (3 - 2,75) + (5 - 3,5) * (5 - 2,75)}{\sqrt{(4 - 3,5)^2 + (0 - 3,5)^2 + (5 - 3,5)^2 + (5 - 3,5)^2} \sqrt{(2 - 2,75)^2 + (1 - 2,75)^2 + (3 - 2,75)^2 + (5 - 2,75)^2}} \\ corr_{1,5} &= \frac{\sum_i (0,5) * (-0,75) + (-3,5) * (-1,75) + (1,5) * (1,25) + (1,5) * (2,25)}{\sqrt{(0,5)^2 + (-3,5)^2 + (1,5)^2 + (1,5)^2} \sqrt{(-0,75)^2 + (-1,75)^2 + (0,25)^2 + (-2,25)^2}} \\ corr_{1,5} &= \frac{-0,375 + 6,125 + 0,375_{3,375}}{\sqrt{17} \sqrt{8,1875}} = \frac{-0,375 + 6,125 + 0,375_{3,375}}{4,123 * 2,86} \\ corr_{1,5} &= \frac{9,5}{4,123 * 2,86} = 0,8 \end{aligned} \quad (4)$$

Os resultados dos cálculos do usuário João com os outros usuários: João x Maria: 0,3; João x José: -0,8; João x Amelie: 0,7; João x Pedro: 0,8.

Com base nos resultados, pode-se confirmar que dentro desta lista de usuários, Pedro e José são os vizinhos mais próximos de João.

Outros métodos utilizados são: Similaridade Cosseno, a Distância Euclidiana (MELO, 2016, p. 40), *k-nearest neighbors* (CABRÉ, 2011), a Correlação de Ranking Spearman (FAZIO, 2013, p. 40), Correlação de Kendall (LIRA, 2004, p. 106), entre outras que são pouco descritas na literatura, uma vez que a maioria dos testes com filtros colaborativos utilizam Correlação de Pearson ou Cosseno.

1.5 Selecionar itens recomendáveis

Após achar os vizinhos mais próximos, o próximo passo é recomendar os itens dos usuários vizinhos que o usuário alvo. O requisito principal é que os itens têm que ser novos, ou seja, por exemplo, em um sistema de recomendação de filmes, encontrar filmes de usuários vizinhos que o usuário alvo ainda não tenha assistido. Logo, seguindo o exemplo, os itens recomendáveis podem ser: (1) todos os filmes não assistidos, (2) selecionar aleatoriamente entre todos os filmes não assistidos até encontrar os Top-N filmes, ou (3) pode-se restringir a busca apenas aos filmes assistidos pelos k usuários mais semelhantes (k-vizinhos) e que ainda não foram assistidos pelo usuário alvo.

Para encontrar recomendações de itens podemos utilizar diversos métodos como médias, *kNN*, métodos híbridos, entre outros. Para exemplificar, será utilizada como método a média ponderada das avaliações de filmes com boas notas entre os vizinhos mais próximos. A Equação 5 corresponde ao cálculo da predição por média ponderada (REATEGUI, 2005):

$$p_{ai} = \bar{r}_a + \frac{\sum_{b=1}^n (r_{bi} - \bar{r}_b) * corr_{ab}}{\sum_{b=1}^n |corr_{ab}|} \quad (5)$$

Sendo que $corr_{ab}$ corresponde à similaridade entre o usuário alvo a com um determinado usuário b ; p_{ai} corresponde à predição de um item i para um usuário a ; \bar{r}_a é a média de todas as avaliações do usuário alvo a aos itens que foram computados por todos os seus usuários similares; r_{bi} é a avaliação que o usuário alvo b atribuiu ao item i ; \bar{r}_b é a média de todas as avaliações do usuário b em comum com o usuário a . Este cálculo deve percorrer todos os itens de um determinado usuário b que não tenham sido adquiridos pelo usuário alvo a . E assim, ao final, terá uma lista de itens com suas notas (frequências), sendo que as maiores notas têm que mais chance de o usuário alvo gostar.

1.6 Recomendar itens

De acordo com o método utilizado na seleção de itens, alguns critérios podem ser levados em consideração na hora de apresentar os itens ao usuário. No caso de uma seleção aleatória, pode-se utilizar como critério “itens que obtiveram nota maior que x serão recomendados” e também, “itens que obtiveram notas superiores à média do usuário serão recomendados”.

1.7 Vantagens e desvantagens

A popularização dos algoritmos de recomendação baseados na filtragem colaborativa se deve as suas vantagens, que são:

- Não depender da análise do conteúdo.
- Apresentar recomendações inesperadas, ou seja, que não possuam características semelhantes aos itens de seu perfil e que não foram pesquisados de forma ativa.
- Possibilidade de formação de comunidades de usuários pela identificação de seus gostos e interesses similares.

Embora, a propagação de filtragem colaborativa ajuda desenvolver melhorias nos algoritmos, ainda existem muitos desafios pertinentes:

- Problemas de escalabilidade já que muitos sistemas possuem muito mais usuários do que itens.
- Quando o usuário possui preferências muito específicas, sendo pouco semelhantes aos demais usuários (*Black Sheep*). Por conseguinte, este usuário não se beneficiará das recomendações.

1. Projeto *Cine Collection*

Este capítulo detalha o processo do desenvolvimento de um aplicativo de recomendação de filmes por meio da engenharia de software, a ciência capaz de representar todos os aspectos de produção de software. Também descreve o processo de desenvolvimento do aplicativo.

Segundo Pressman (2007, p. 31) “os métodos de engenharia de software proporcionam os detalhes de “como fazer” para construir o software”. Ao longo dos anos, foram concebidas várias metodologias de desenvolvimento de software, sendo que uma metodologia conhecida como o modelo cascata serviu e ainda serve de base para grande parte das metodologias

modernas, como também é muito utilizada até hoje e foi à metodologia escolhida para desenvolver este projeto.

O Modelo Cascata também chamado de ciclo de vida clássico, foi descrito por Winston W. Royce em 1970. Até meados da década de 1980 foi o único modelo com aceitação geral. Este modelo propõe uma abordagem sequencial e sistemática para o desenvolvimento, sendo que a próxima etapa só pode ser iniciada depois que sua predecessora for finalizada (PRESSMAN, 2007), com o fim de estabelecer ordem no desenvolvimento de grandes produtos de software.

A metodologia Cascata possui cinco importantes etapas: comunicação, planejamento, modelagem, construção e implantação. Nos próximos subcapítulos será detalhada cada atividade de engenharia de software baseando-se nas etapas fundamentais sugeridas por Pressman e aplicadas no contexto deste projeto.

2.1 Comunicação

A comunicação é primeira etapa de um projeto de software, onde são executadas tarefas como: a definição dos objetivos do software, a análise de viabilidade e o levantamento de requisitos.

O objetivo de um estudo de viabilidade, como o próprio nome já diz, é avaliar sob o ponto de vista técnico, operacional, econômico e organizacional se o projeto é viável. Para Sommerville (2007) todo sistema novo deve realizar um estudo de viabilidade:

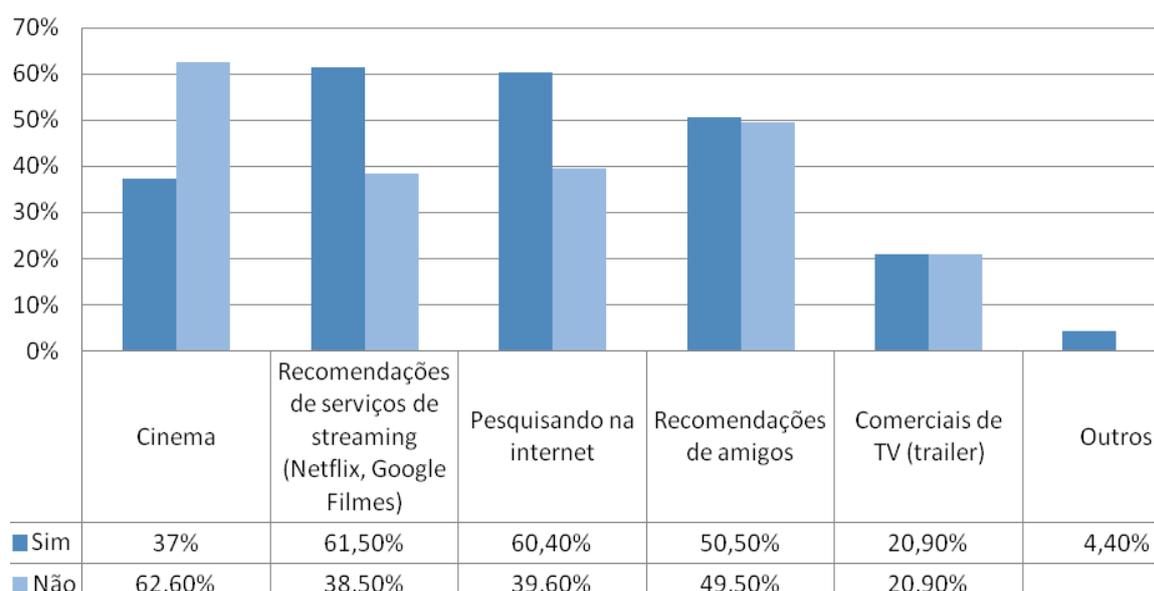
“Para todos os sistemas novos, o processo de requisitos deve começar com um estudo de viabilidade. A entrada para o estudo de viabilidade é uma descrição geral do sistema e de como ele será utilizado” (SOMMERVILLE, 2007, p. 103).

A pesquisa de viabilidade analisou a proposta do aplicativo de filmes Cine Collection e, por meio de uma pesquisa avaliou sua viabilidade.

A proposta do aplicativo de filmes nasceu pela dificuldade de recordar-se de uma lista de filmes assistidos e de achar boas recomendações. Após uma pesquisa, constatou-se que, para celular não há aplicativos eficientes para a realização destas tarefas. Para testar a viabilidade de um aplicativo de filmes, foi executada uma pesquisa no qual as perguntas levariam ao resultado se o aplicativo de filmes Cine Collection é viável ou não.

A pesquisa entrevistou em grande parte do público, jovens entre 19 a 25 anos, como resultado obteve-se que a maioria dos entrevistados não utiliza meios de organizar filmes já assistidos e que a procura por novos filmes para assistir é feita por recomendações feita por uma pesquisa na internet ou de acordo com recomendações de serviços de *streaming*. Porém, os resultados obtidos também apontam que, 48,2% dos entrevistados não acreditam ter controle dos filmes que assiste. A Figura 1 apresenta como os usuários procuram por novos filmes para assistir sendo que o usuário pode escolher mais de uma opção.

Figura 1 – Como os usuários procura por novos filmes para assistir.



Fonte: Elaborado pelos autores.

De acordo com os resultados obtidos, foi decidido prosseguir com o desenvolvimento do projeto. Após feita a análise de viabilidade, o próximo estágio do processo de engenharia de software é o levantamento e análise de requisitos. Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 4 apresenta os requisitos funcionais deste projeto.

Tabela 3 – Requisitos funcionais do projeto.

| Identificação | Requisito Funcional | Categoria | Prioridade |
|---------------|--|----------------------|------------|
| RF001 | O sistema deve ter uma tela de carregamento. | Design e Programação | Baixa |
| RF002 | O sistema deve ter login com Facebook | Design e Programação | Alta |
| RF003 | O sistema deve exibir uma lista de filmes recomendados personalizado para cada usuário. | Programação | Alta |
| RF004 | O sistema deve ter um motor de busca de filmes que exibirá a foto e o nome do filme procurado. | Programação | Alta |
| RF005 | Cada filme deverá conter ao menos uma foto, o título, a sinopse do filme, o elenco e a equipe técnica. | Design e Programação | Alta |
| RF006 | O sistema deve exibir um botão de “Favoritar” | Design e Programação | Média |
| RF007 | O sistema deve ter a possibilidade de o usuário avaliar um filme | Programação | Alta |

| | | | |
|-------|--|----------------------|-------|
| RF008 | O usuário deverá ter um perfil com uma lista dos filmes avaliados. | Design e Programação | Baixa |
|-------|--|----------------------|-------|

Fonte: Elaborado pelos autores.

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 5 apresenta os requisitos não funcionais deste projeto.

Tabela 4 – Requisitos não funcionais do projeto.

| Identificação | Requisito não funcional | Categoria | Prioridade |
|---------------|--|-------------|------------|
| RNF001 | Portabilidade: o aplicativo deve funcionar na maioria dos celulares. | Programação | Alta |
| RNF002 | Requisitos Técnicos: o aplicativo não funcionará sem rede Wifi ou 3G. | Programação | Muito alta |
| RNF003 | Usabilidade: oferecer botões e links de tamanhos razoáveis facilitando a navegação. | Design | Alta |
| RNF004 | Design: o aplicativo deve conter um design criativo e autêntico, mas que seja fácil de utilizar na visão dos usuários. | Design | Média |
| RNF005 | Desempenho: Tempo limite de processamento de todas as recomendações não deve ultrapassar de 2000 milissegundos. | Programação | Alta |
| RNF006 | Segurança: O servidor deve realizar um backup semanal do banco de dados. | Programação | Alta |

Fonte: Elaborado pelos autores.

3.2 Planejamento

Os riscos podem afetar o cronograma de projeto ou a qualidade do software. Os riscos podem ser relacionados: ao projeto: afetando a programação ou os recursos do projeto; ao produto: afetando a qualidade ou o desempenho do software que está em desenvolvimento e ao negócio: afetando a quem está desenvolvendo ou adquirindo o software (SOMMERVILLE, 2007, p. 69). A Tabela 6 apresenta os riscos encontrados que podem afetar este projeto.

Tabela 5 – Riscos encontrados que podem afetar este projeto.

| Risco | Tipo de Risco | Descrição |
|------------------------------------|---------------|---|
| Indisponibilidade da API de filmes | Produto | A API de filmes que será utilizada para ler uma biblioteca de filmes pode ficar indisponível. |

| | | |
|-------------------------|-------------------|--|
| Alteração de requisitos | Projeto | Podem haver mais requisitos ou mudança nos requisitos do que o previsto. |
| Mudança de tecnologia | Negócio | A tecnologia básica a qual o software opera pode ser superada por uma nova tecnologia. |
| Tamanho subestimado | Projeto e Produto | O tamanho ou a complexidade do sistema pode ser subestimado. |
| Prazo subestimado | Projeto e Negócio | Não entregar o projeto no prazo estipulado. |

Fonte: Elaborado pelos autores.

O aplicativo foi desenvolvido para celulares *Android*, utilizando o ambiente de desenvolvimento Android Studio. Pela facilidade de implementação foi empregado uma base de dados de filmes gratuita denominada The Movie Database (TMDB)¹⁰ e, os dados dos usuários e outros dados do aplicativo serão armazenados em um banco de dados Firebase. O sistema possui um *login* com o Facebook, portanto também comportará a biblioteca do Facebook.

3.3 Modelagem

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

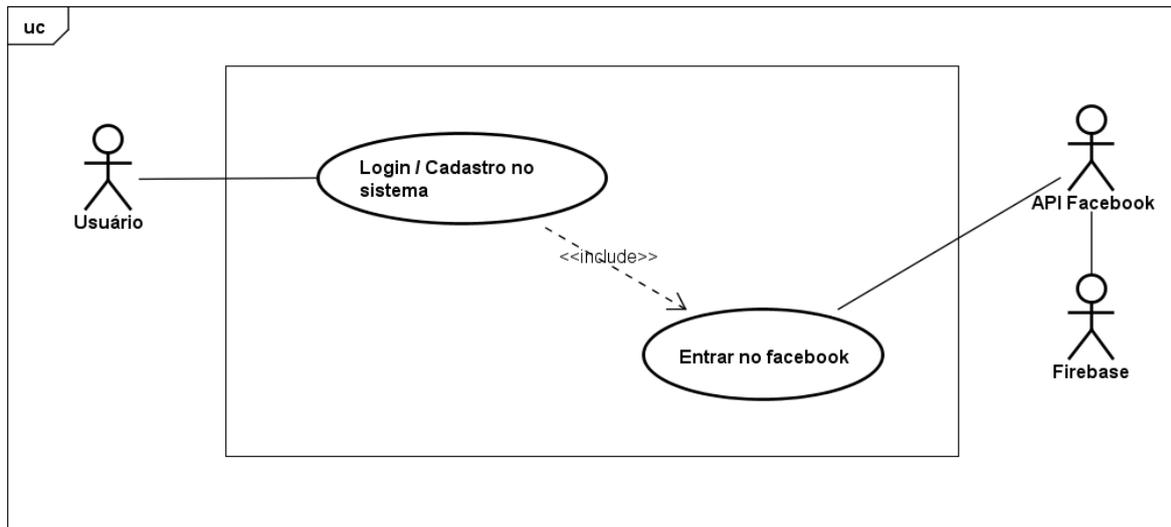
Os atores que interagem com o sistema são: o Usuário, API Facebook, Firebase, API TMDB e o Sistema de recomendação. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

- Usuário é o ator que representa os utilizadores deste aplicativo. Um ator pode, por exemplo, buscar filmes, avaliar, favoritar, entre outros.
- API Facebook representa o ator da API que permite a interação entre o aplicativo e o Facebook, por exemplo, o login com as credenciais da rede social.
- Firebase representa o banco de dados em tempo real, onde o sistema armazena as informações de usuários e de filmes.
- API TMDB representa a API de filmes que faz requisições e recebe os dados utilizando uma REST.
- Sistema de Recomendação representa a parte do sistema responsável por gerar recomendações ao usuário de modo automático.

A Figura 2 apresenta o caso de uso para a entrada do usuário no sistema.

¹⁰ The Movie Database. Disponível em: <<https://www.themoviedb.org/?language=pt-BR>>.

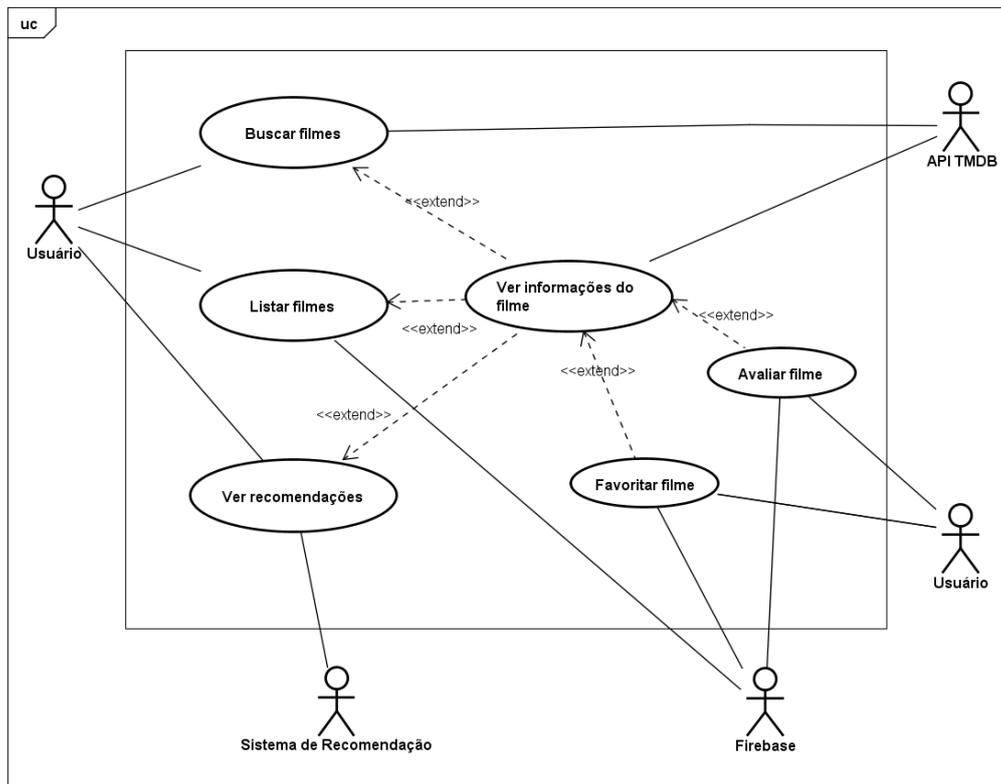
Figura 2 - Diagrama de caso de uso do login de usuário.



Fonte: Elaborado pelos autores.

A Figura 3 apresenta o caso de uso que representa as funcionalidades relacionadas ao usuário após a entrada no sistema.

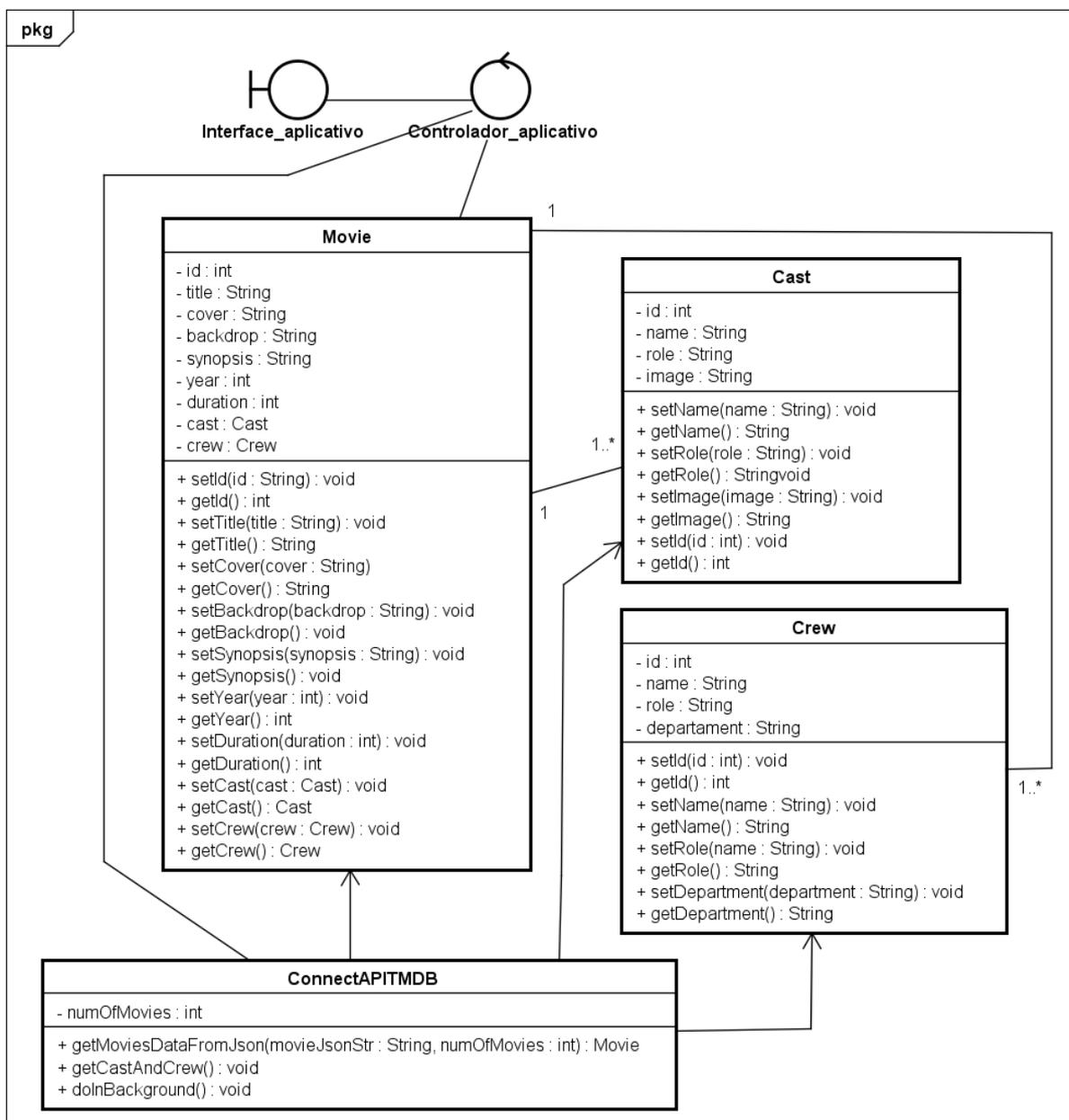
Figura 31 - Caso de uso de funcionalidades do aplicativo.



Fonte: Elaborado pelos autores.

O diagrama de classe é responsável por oferecer uma representação da estrutura e relações das classes como também as operações solicitadas pelos atores que servem de modelo para os objetos. Para entender melhor a estrutura funcional do projeto, a Figura 4 apresenta as principais classes do aplicativo Cine Collection.

Figura 2 – Diagrama de classe de filmes.



Fonte: Elaborado pelos autores.

A classe *ConnectAPITMDB* é responsável por criar e gerenciar a conexão com a API. Possuindo dois métodos que são:

- *doInBackground*: este método inicia a *Thread* ¹¹ que acessa a API TMDb, retorna os dados como um objeto JSON e utiliza este objeto no método *getMovieDataFromJson*.
- *getMovieDataFromJson*: retorna os dados de um filme específico ou de um grupo de filmes, de acordo com o requerimento feito pela interface.
- *getCastAndCrew*: obtém os dados do Elenco e da Equipe técnica de um filme em específico.

O método *getCastAndCrew* só é acessado pelo método *getMovieDataFromJson* e quando a interface necessita de dados específicos de um filme.

A classe *Movie* é responsável por armazenar todos os dados de um filme. Além disso, fornece acesso aos seus atributos por meio de métodos *getters* e *setters*. Dentro os dados do filme estão as classes *Cast* e *Crew*.

A classe *Cast* é encarregada de armazenar os dados do elenco de um filme, já a classe *Crew* guarda os dados de equipe técnica de um filme. Ambas possuem os métodos assessores *getters* e *setters*.

Outras classes que compõem o aplicativo são as classes controladoras, ou seja, classes denominadas como *Activity* ou *Fragments*. A Figura 5 apresenta um conjunto de classes de visualização e controle utilizados no projeto.

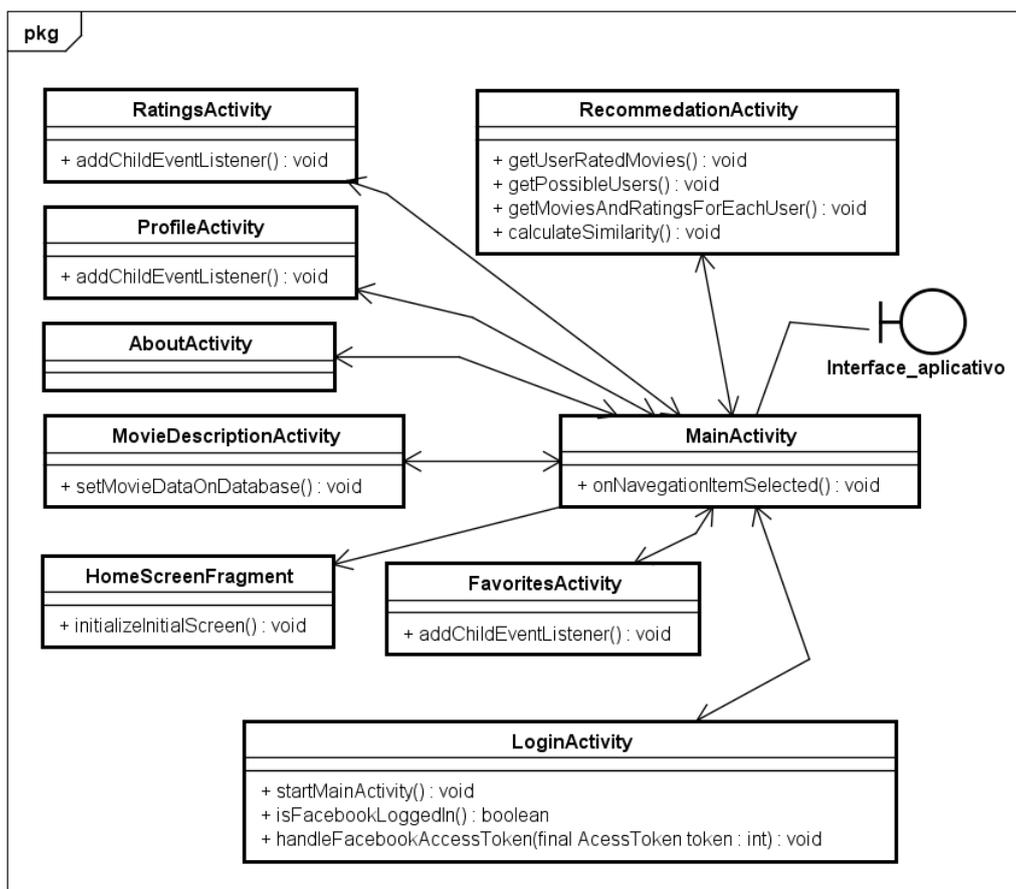
As classes controladoras têm como objetivo fazer a ligação entre o modelo (Firebase e APIs) e a View (Layouts em XML), como também controla o que está sendo visualizado pelo usuário.

A classe *LoginActivity* possui os seguintes métodos:

- *isFacebookLoggedIn*: este método checa se objeto *AccessToken* do Facebook não é um valor nulo, caso não veja nulo, retorna *true*.
- *handleFacebookAccessToken*: este método é responsável por pegar o *token* da pessoa que fez o login do Facebook pelo aplicativo e checar se já existe ou não sua conta no banco de dados do Firebase, se não existir, ele adiciona o usuário a base e faz o login, caso já exista o usuário, o banco loga com a conta do usuário.
- *startMainActivity*: este método inicia a *MainActivity* que leva para a página principal do aplicativo.

¹¹ *Thread* ou encadeamento de execução é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas em paralelo.

Figura 3 – Diagrama de classe de visualização e controle.



Fonte: Elaborado pelos autores

A classe *MainActivity* possui como métodos:

- *onNavigationItemSelected*: implementado através de um *NavigationView*, este método tem como objetivo fornecer a navegação entre as páginas do aplicativo.
- A classe *AboutActivity* retorna para a view as informações sobre o desenvolvedor.
- A classe *MovieDescriptionActivity* é responsável por carregar todos os dados de um determinado filme e possibilitar o usuário Favoritar ou Avaliar um filme. Possui como método:
- *setMovieDataOnDatabase*: este método persiste o filme no banco de dados Firebase para que possa utilizado em outras partes do aplicativo com o intuito de agilizar a busca dos filmes.
- *OnTaskCompleted*: é executada assim que a classe *ConnectAPITMDB* termina sua execução.

A classe *ProfileActivity* é responsável por apresentar uma pequena mostra de filmes que o usuário já avaliou e já favoritou, já as classes *FavoritesActivity* e *RatingActivity* retornam uma listagem de filmes a partir de uma requisição ao Firebase. A classe *RecommendationActivity* apresenta ao usuário as recomendações geradas pelo Sistema de Recomendação. O método *getUserRatedMovies()* é responsável por buscar todos os filmes bem avaliados do usuário e a partir disto, é chamado o método *getPossibleUsers()*, que busca os usuários que também avaliaram os filmes que o usuário alvo avaliou e chama o método

getMoviesAndRatingsForEachUser() que monta a matriz de *usuários* x *itens* com suas respectivas avaliações e, após isto, o método *calculateSimilarity()* é chamado no qual é calculado a similaridade entre o usuário alvo e os outros usuários, encontrado os usuários vizinhos, é encontrado os filmes que os usuários vizinhos avaliaram com boas notas e que o usuário alvo ainda não tenha avaliado e é recomendado estes itens.

3.4 Avaliação

Por meio de uma avaliação pode-se verificar se o aplicativo é funcional em todos os aspectos, apresentando “qualidade”. A qualidade de software é a resolução total de requisitos funcionais e não funcionais atingindo o resultado esperado a fim de satisfazer as necessidades do cliente (ISO 9126).

A ISO 9126 representa a atual padronização mundial para a qualidade de software. A norma propõe que um software para ter qualidade deve possuir as seguintes características: Funcionalidade; Confiabilidade; Usabilidade; Eficiência; Manutenibilidade e Portabilidade.

As características de qualidade de software implementadas no aplicativo Cine Collection são:

- **Funcionalidade:** o software possui um sistema recomendador de filmes baseado em filtragem colaborativa, de acordo com a proposta do aplicativo.
- **Confiabilidade:** toda vez que ocorre um erro inesperado, o software mostra o erro que ocorreu informando o seu motivo.
- **Usabilidade:** o software possui um sistema intuitivo e fácil de ser utilizado, pois este aplicativo segue os padrões estabelecidos pelo Material Design¹² do Google.
- **Eficiência:** o software se mantém estável durante toda sua utilização, quando há um processamento demorado a ser carregado, o software informa que está carregando o conteúdo.
- **Manutenibilidade:** Quando o aplicativo está em uma nova versão na loja, o usuário é notificado pela loja do Google Play Store sobre as atualizações do software.
- **Portabilidade:** foi priorizado a renderização do software em smartphones, sendo compatível com 76,9% dos smartphones Android utilizados no mundo (ANDROID STUDIO, 2017).

De acordo com a avaliação final dos usuários conclui-se que o aplicativo Cine Collection atingiu seus objetivos em fornecer recomendações de filmes com base nos filmes que o usuário avaliou, baseando-se na filtragem colaborativa, o sistema buscou os vizinhos mais próximos (Seção 2.1) e forneceu as recomendações.

2. Resultados e discussão

Nesta seção é apresentado o resultado do aplicativo desenvolvido seguindo o modelo cascata. Este aplicativo cumpre os resultados apresentados na seção 3.5. Pode-se observar, na Figura 6a, a tela de abrir o aplicativo no qual é exibido uma tela de abertura, além de exibir uma imagem ao usuário, também tem a função de carregar o botão do login e fazer com que redirecione para a tela inicial do aplicativo. A Figura 6b apresenta a tela de login com o Facebook no aplicativo. Caso o retorno da tela inicial seja bem-sucedido, ele redireciona para tela inicial do aplicativo. A tela inicial do aplicativo exibe os filmes populares e as recomendações em caso que o login da tela inicial seja bem-sucedido, além de incluir o botão de menu e de busca. A Figura 6c apresenta a tela inicial do aplicativo quando o usuário está “logado”.

¹² Informações disponíveis em <https://material.io>.

Os testes realizados com os usuários detectaram uma facilidade de acessar as funcionalidades do aplicativo, tais como as recomendações, lista de filmes favoritos e minha conta. As recomendações quando a base de usuários é pequena ou quando o usuário ainda não favoritou ou avaliou filmes se torna um tanto imprecisa, sendo necessário que o usuário avalie, pelo menos, três filmes para as avaliações começarem a fazer sentido.

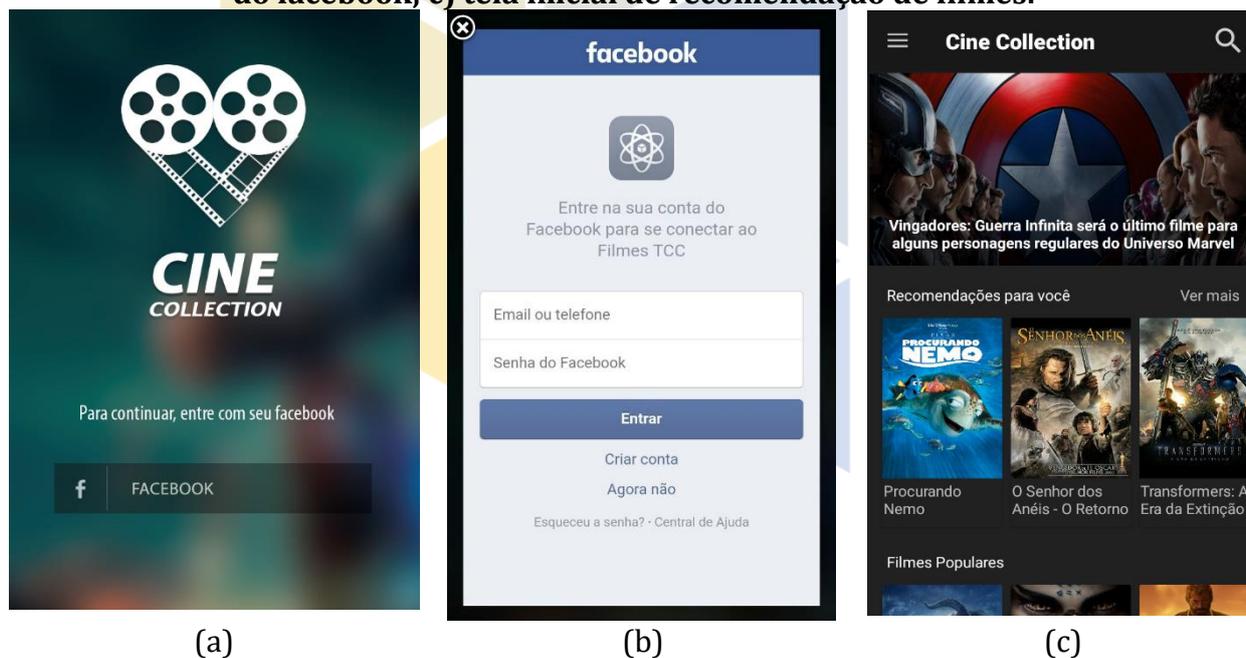
Os usuários também detectaram a necessidade de criar listas de filmes, também como históricos de filmes avaliados. Sentiram a falta de botões de compartilhamento para compartilhar os filmes nas redes sociais, deram a sugestão de as listas de filmes também possam ser compartilhadas.

Na descrição, os usuários gostaram do design, mas argumentaram que o tamanho das fontes estava muito grande, perdendo espaço para novas informações. Outras sugestões foram mostrar informações de atores e diretores de filmes e “se tornar fã”.

Na tela de busca, os usuários deram a sugestão de trazer mais informações como o ano do filme. E também, em ver perfis de outros usuários e buscar amigos pelo Facebook para compartilhar seus gostos cinéfilos.

A avaliação em geral foi boa, apesar de se tratar de um MVP, os usuários apresentaram aceitação ao aplicativo, sugerindo novas funcionalidades, que devem ser implementadas antes do lançamento final.

Figura 5 - Telas iniciais do aplicativo: a) tela de abertura do aplicativo; b) tela de login do facebook; c) tela inicial de recomendação de filmes.

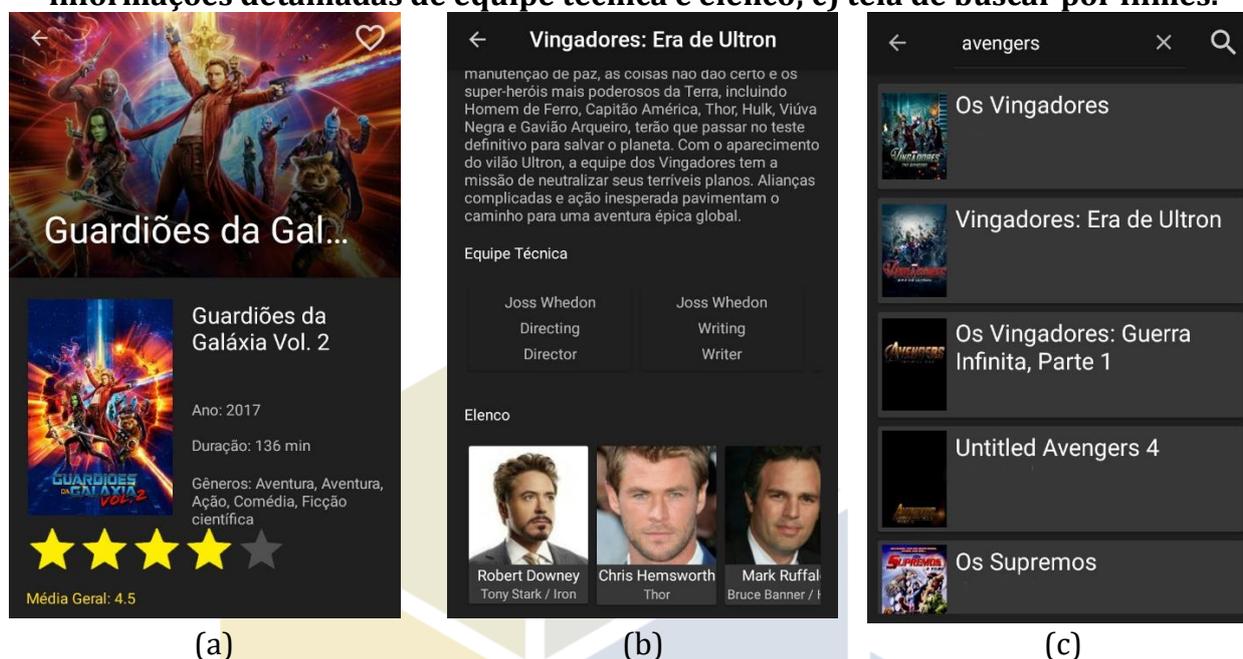


Fonte: Elaborado pelos autores

A tela de informações do filme mostra todas as informações do filme, como título, ano, sinopse, duração. Note que esta tela também traz as informações sobre os votos e os botões de Favoritar ou não, como também, há um elemento chamado “Rating Bar”. A Figura 7a apresenta a tela de informações detalhadas sobre o filme. Ainda na tela de informações do filme é possível ver as informações sobre a equipe técnica e sobre o elenco. A Figura 7b apresenta como as informações adicionais são apresentadas. A tela de busca de filmes mostra os filmes de acordo

com o resultado de uma requisição através do campo de texto solicitando a pesquisa por alguma informação do filme. A Figura 7c apresenta a tela de busca por filmes.

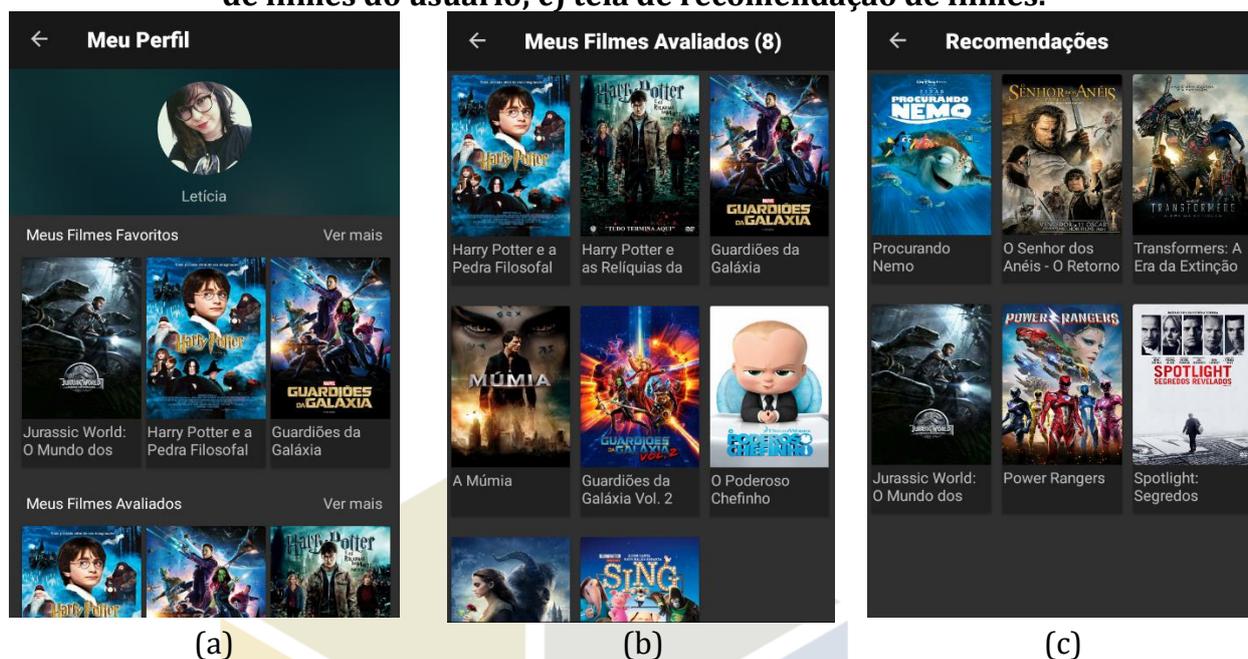
Figura 6 – Telas do aplicativo: a) tela de informações detalhadas do aplicativo; b) tela de informações detalhadas de equipe técnica e elenco; c) tela de buscar por filmes.



Fonte: Elaborado pelos autores

A tela de perfil mostra um resumo dos filmes favoritos e dos avaliados bem como a quantidade de filmes de cada uma dessas categorias. Após clicar no botão “Ver mais” o aplicativo irá para a listagem completa da categoria selecionada. A Figura 8a apresenta a tela de perfil do usuário. A tela de filmes favoritos e avaliados mostra uma listagem de todos os filmes que o usuário favoritou ou avaliou de acordo com a que o usuário solicitou, no qual ao clicar em uma miniatura do filme, o sistema leva para a página de informações do filme. A Figura 8b apresenta a tela de listagem de filmes favoritos ou avaliados pelo usuário. A tela de recomendações de filmes se trata das recomendações personalizadas que o sistema de recomendação do aplicativo o processou. As recomendações são carregadas de forma automática quando a tela de recomendação é solicitada. A Figura 8c apresenta a tela de recomendação de filmes.

Figura 7 – Telas iniciais do aplicativo: a) tela de perfil do usuário; b) tela de avaliações de filmes do usuário; c) tela de recomendação de filmes.



Fonte: Elaborado pelos autores

4 Considerações finais

Este trabalho teve como objetivo final, realizar um estudo sobre sistemas de recomendação e suas técnicas mais conhecidas. Para consolidar esse estudo foi realizado o desenvolvimento do aplicativo Cine Collection utilizando a linguagem Java para Android no qual foi implementado um Sistema de Recomendação baseada em filtragem colaborativa que garantiu os resultados propostos.

O trabalho inicialmente explorou o que é um Sistema de Recomendação e trouxe o conceito das técnicas de filtragem baseada em conteúdo, filtragem colaborativa e filtragem híbrida. Foi escolhido a filtragem colaborativa como sistema recomendador para a recomendação de filmes, pela facilidade de gerar as informações necessárias, pelo fato de poder existir uma comunidade colaborativa.

Na sequência, apresentou-se todo o processo de desenvolvimento do aplicativo, ministrando todo o planejamento desde a escolha da metodologia de desenvolvimento cascata, a escolha das tecnologias e das ferramentas utilizadas, plano de requisitos, diagramas e por fim o desenvolvimento, no qual foi desenvolvido as partes fundamentais do software.

Durante o desenvolvimento do aplicativo, houveram algumas dificuldades dado, principalmente, pela curva de aprendizado de algumas ferramentas utilizadas. Os principais pontos foram a inexperiência com bancos de dados não relacionais e, também, o modelo de desenvolvimento de aplicativos Android.

Contudo, o desenvolvimento do aplicativo ainda foi facilmente implementado, graças a documentação do Android e a outros pequenos projetos feitos no passado, que serviram de base para implementação de várias partes do código como, as requisições da API do Facebook, as requisições utilizando uma REST API, entre outras.

A implementação ainda pode ser melhorada em alguns pontos, como, a conexão com o banco de dados do Firebase, o cálculo do sistema de recomendação e a organização geral do projeto.

A interface do usuário foi construída a fim de manter a experiência do mesmo concisa, seguindo os padrões da plataforma Android para facilitar a navegação e manter a familiaridade do usuário em relação a outros aplicativos do gênero. O aplicativo é simples de ser utilizado e pode ser utilizado por usuários de qualquer nível de experiência, mesmo sendo focado para o público jovem.

O aplicativo cumpre o que promete, gerando recomendações de novos filmes aos usuários, porém, para ser funcional necessita de uma base de usuários maior para ser mais efetivo.

Como possíveis trabalhos futuros, pode-se apontar: um sistema recomendador de filmes com base no humor do usuário ou em bases comportamentais; o desenvolvimento do aplicativo também para iPhone, oferecendo a mesma estrutura que da plataforma Android e, até mesmo, um sistema com recomendação híbrida para suprir as fraquezas de ambas às técnicas de recomendação.

Referências

- AMAZON PRIME VIDEO. Disponível em: <<https://www.primevideo.com>>. Acesso em: 22 maio 2017.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126: Engenharia de software – Qualidade de Produto**, 1994.
- BOGERS, T. **Recommender Systems for Social Bookmarking**. Tese (PhD), Tilburg University, 2009.
- CABRÉ, G.J. **Filtragem Colaborativa Aplicada à Recomendação Musical**. Dissertação (Mestrado em Informática Aplicada) - Fundação Edson Queiroz, Universidade de Fortaleza, Fortaleza, 2011. Disponível em: <https://upcommons.upc.edu/bitstream/handle/2099.1/13071/Monografia_FINAL.pdf>. Acesso em: 19 de mar. 2017.
- CAZELLA, S.C.; NUNES, M.A.S.; REATEGUI, E.B. **A Ciência da opinião: estado da arte em Sistemas de Recomendação**. CSBC – XXX Congresso da SBC – Jornada de Atualização de Informática-JAI, p. 161-216, 2010.
- CGI.BR – **Comitê Gestor da Internet no Brasil**. TIC Domicílios 2014. Disponível em: <http://www.cgi.br/media/docs/publicacoes/2/TIC_Domicilios_2014_livro_eletronico.pdf>. Acesso em: 07 de mar. 2017.
- CONNECTAÍ. **34% dos internautas brasileiros assistem conteúdo on demand uma vez por semana**. Disponível em: <<http://conecta-i.com/?q=pt-br/34-dos-internautas-brasileiros-assistem-conte%C3%BAdo-demand-uma-vez-por-semana>>. Acesso em: 09 de mar. 2017.
- FAZIO, M.R. **Previsão de Avaliações em Sistemas de Recomendação para Nichos de Mercado**. Dissertação (Pós-Graduação em Engenharia de Sistemas) - Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013. Disponível em: <<http://www.cos.ufrj.br/uploadfile/1365598708.pdf>>. Acesso em: 05 de maio 2017.

- IBGE. Instituto Brasileiro de Geografia e Estatística.** Disponível em: <<http://www.ibge.gov.br>>. Acesso em: 07 de mar. 2017.
- PNAD. Pesquisa Nacional por Amostra de Domicílios (PNAD) 2014.** Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv94935.pdf>>. Acesso em: 22 de maio 2017.
- IBM. Sistemas de Recomendação.** Disponível em: <https://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao>. Acesso em: 11 de mar. 2017.
- LIRA, S.A. Análise de Correlação: Abordagem Teórica e de Construção dos Coeficientes com Aplicações.** Dissertação (Pós-Graduação em Métodos Numéricos) - Engenharia, Universidade de Federal do Paraná, 2004. Disponível em: <http://www.ipardes.gov.br/biblioteca/docs/dissertacao_sachiko.pdf>. Acesso em: 23 maio 2017.
- MELO, E.V. Sistema de Recomendação de Imagens Baseado em Atenção Visual.** Tese (Doutorado em Ciência da Computação) - Faculdade de Computação, Universidade Federal de Uberlândia, Minas Gerais, 2016. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/17824/1/SistemaRecomendacaoImagens.pdf>>. Acesso em: 11 de maio 2017.
- PAZZANI, M.J. A Framework for Collaborative, Content-Based and Demographic Filtering.** *Artificial Intelligence Review*, p. 393-408, 1999
- PRESSMAN, R.S. Engenharia de Software.** 7ª Edição, São Paulo, Editora: Makron Books, 2007.
- REAL, R.; VARGAS, J. The probabilistic basis of jaccard's index of similarity.** *Systematic Biology*. p. 380-385, 1996.
- REATEGUI, E.B.; CAZELLA, S.C. Sistemas de recomendação.** *Anais do XXV Congresso da Sociedade Brasileira de Computação*, p. 306-348, 2005. Disponível em: <<https://pdfs.semanticscholar.org/ff37/56d0e7d480dd098b334df5006a740d11ce06.pdf>>. Acesso em: 20 de set. 2016.
- RICCI, F., ROKACH, L., SHAPIRA, B. Introduction to recommender systems handbook.** *Recommender Systems Handbook*. Boston, MA: Springer, 2011.
- ROZENDO, R.G. Avaliação de Sistemas de Recomendação com uma Proposta de um Algoritmo Híbrido.** Dissertação (Graduação em Engenharia de Computação e Informação) - Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017. Disponível em: <<http://monografias.poli.ufrj.br/monografias/monopoli10019604.pdf>>. Acesso em 13 abr. 2017.
- SAMPAIO, I.A. Aprendizagem Ativa em Sistemas de Filtragem Colaborativa.** Tese (Mestrado em Ciência da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife, 2006. Disponível em: <http://repositorio.ufpe.br/bitstream/handle/123456789/2608/arquivo5346_1.pdf>. Acesso em: 11 de mar. 2017.
- SILVA, R.G.N. Sistema de Recomendação baseado em conteúdo textual: avaliação e comparação.** Dissertação (Mestrado Multiinstitucional em Ciência da Computação) -

Universidade Federal da Bahia, Universidade Estadual de Feira de Santana, 2014. Disponível em:

<https://repositorio.ufba.br/ri/bitstream/ri/19281/1/dissertacao_mestrado_ciencia_computacao_rafael_glauber.pdf>. Acesso em: 22 de maio 2017.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Edição. Editora: Pearson Addison-Wesley. São Paulo, 2007.

STATCOUNTER. **Android overtakes Windows for first time**. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv94935.pdf>>. Acesso em: 08 de abr. 2017.

THE MOVIE DATABASE. Disponível em: <<https://www.themoviedb.org>>. Acesso em: 09 de mar. 2017.

