

# Análise dos fatores e relações que levam aplicativos de ensino de programação ao sucesso na Google Play Store

Jonathan Dias Rodrigues, Luiz Fernando de Oliveira Macedo, Lesandro Ponciano

PUC Minas em Contagem

Bacharelado em Sistemas de Informação

jonathandias96@gmail.com, luizfernando1996@yahoo.com.br,  
[lesandrop@pucminas.br](mailto:lesandrop@pucminas.br)

**Abstract.** *The growth in the use of smartphones in recent years has led to increased competition between application developers, and a concern about quality to achieve success (number of stars) in the app store environment. In this context, several apps have been developed with the more varied objectives, among them, we highlight in this study those apps that aim to support the study of computer programming. The computer programming is an area of prominence for the growing demand for people who have this knowledge. This study investigates which factors and their relations are important for an app reach success in the app store environment, particularly in the Google Play Store. In the analysis, by using k-means algorithm, the 433 applications are clustered into four groups: unpopular, low popularity, medium and popular popularity. The analysis also includes the correlation of the groups with the polarity of the 78.727 comments provided by the users. It is observed that applications that make up the medium popularity group tend to be successful .*

**Resumo.** *O crescimento no uso de smartphones nos últimos anos tem levado ao aumento da concorrência entre os desenvolvedores de aplicativos, e a preocupação com a qualidade para o alcance de sucesso (número de estrelas) no ambiente de loja de aplicativos. Nesse contexto, têm sido desenvolvidos diversos aplicativos com os mais variados objetivos, dentre eles, destacam-se neste estudo os aplicativos que têm o objetivo de auxiliar no aprendizado de programação de computadores. O ensino de programação é uma área de importância devido à crescente demanda por pessoas que tenham esse conhecimento. Este estudo avalia quais fatores e suas relações que levam um aplicativo de ensino de programação ao sucesso, em lojas de aplicativos, em particular na Google Play Store. Na análise, usando o algoritmo K-means, 433 aplicativos são agrupados em 4 categorias, que são: impopular, popularidade baixa, popularidade média e popular. A análise contempla também a correlação dos grupos com a polaridade de 78.727 comentários providos pelos usuários. É observado que aplicativos que compõem o grupo de popularidade média tendem a obtenção de sucesso.*

## 1. Introdução

Atualmente, tem ocorrido um aumento da concorrência entre desenvolvedores de aplicativos móveis (ou *apps*) ocasionado, principalmente, pelo crescimento da

quantidade de aplicativos disponíveis nas lojas de aplicativos e pelo aumento no uso de *smartphones* [Chen et al. 2014]. Em consequência disso, a preocupação com a qualidade dos aplicativos tem aumentado [Khalid et al. 2015]. Isso torna a análise da atual popularidade dos aplicativos e das avaliações dos usuários (reclamações ou sugestões) fatores de importância para a manutenção do sucesso dos aplicativos [McIlroy et al. 2017]. Logo, é importante avaliar *feedbacks* dos usuários e outras métricas para saber quais fatores e quais relações entre fatores fazem com que um aplicativo obtenha sucesso (número de estrelas) nas lojas de aplicativos. Um exemplo de lojas de aplicativos é a Google Play Store<sup>1</sup>.

Há diferentes fatores que influenciam na popularidade de um aplicativo [Ouyang et al. 2018)]. Para compreendê-los, é necessário a realização de um estudo multivariado. Os estudos de Ouyang et al. (2018) como os de Fu et al. (2013), Sarro et al. (2015) e Lee e Raghu (2014) foram realizados considerando diferentes categorias de aplicativos, mas sem levar em conta o domínio para o qual o aplicativo foi desenvolvido. O foco desse trabalho está em um domínio específico que é o ensino de programação. Então, o problema tratado neste trabalho é **quais fatores e relações que levam os aplicativos relacionados ao ensino de programação a obterem sucesso na Google Play Store?**

O ensino de programação é um domínio importante para o desenvolvimento de software, e aprender tal conhecimento não é uma tarefa simples [Shein 2014]. Logo cada vez mais é importante a existência de aplicativos que deem suporte ao seu aprendizado para o benefício dos futuros ingressantes na área de TI [Oliveira et al. 2014]. Ao determinar os fatores e relações que levam um aplicativo de ensino de programação ao sucesso no Google Play Store, busca-se prover recomendações de como desenvolver aplicativos de sucesso. Busca-se também auxiliar os desenvolvedores em quais fatores priorizar, no processo de desenvolvimento, para aumentar as chances de êxito de seus aplicativos. Por fim, o estudo busca investigar quais são os maiores problemas que desenvolvedores e usuários experimentam nesse contexto.

O principal **objetivo deste trabalho** é descobrir os principais fatores e relações que levam aplicativos de ensino de programação a obter destaque na Google Play Store. Para atingir tal objetivo geral propõem-se os seguintes **objetivos específicos**: (1) obter métricas para avaliação de aplicativos de sucesso no ensino de programação; (2) propor uma abordagem para a identificação das características comuns em aplicativos de sucesso no domínio estudado; (3) compreender atributos de aplicativos de sucesso.

Neste trabalho, a popularidade e o sucesso de aplicativos são estudados através da análise do perfil de aplicativos de acordo com os dados coletados do repositório Play Store do Google, para aplicativos de ensino de programação. Usando o algoritmo de agrupamento k-means, são encontrados quatro diferentes perfis baseados em três diferentes aspectos: número de instalações; número de comentários; número de avaliações. Os perfis são nomeados da seguinte maneira: Popular; Popularidade média; Popularidade baixa; Impopular. A maior porcentagem de aplicativos aparece no perfil popular (91%), enquanto o menor número de aplicativos aparece no perfil popularidade baixa (1%). Em seguida, o estudo da relação dos centróides de cada perfil com o

---

<sup>1</sup> Google Play Store - Acesso disponível em: <<https://play.google.com/store>>

número de estrelas dos aplicativos, demonstra que o perfil de popularidade média possui maior tendência a obtenção de sucesso.

## **2. Referencial Teórico**

Nesta seção são apresentados os principais conceitos relacionados a aplicativos e lojas de aplicativos. A Google Play Store, exemplo de loja de aplicativos, é apresentada em duas perspectivas: a de usuários e a de desenvolvedores.

### **2.1. Aplicativo (*App*)**

Aplicativos, *apps*, ou também aplicações, são softwares que se caracterizam como computação móvel, porque podem ser instalados em um dispositivo eletrônico móvel, como um *smartphone*, leitor de MP3 (do inglês, *MPEG Audio Layer-3*), Assistente Digital Pessoal (PDA, do inglês *Personal Digital Assistant*), entre outros dispositivos, e assim assistir o usuário para cumprir determinados objetivos de forma eficaz e eficiente [Floriano et al. 2011]. Emerge desses objetivos o paradigma de *mobile learning* que é caracterizado pela utilização da tecnologia móvel no modelo de aprendizado [Marçal et al. 2005].

### **2.2. Loja de Aplicativos**

Loja de aplicativos é uma plataforma de distribuição de aplicativos móveis [Panichella et al. 2015]. Segundo Lee e Raghu (2014), lojas de aplicativos são acessadas por um público heterogêneo. Em alguns casos, como a Apple App Store, trata-se de um canal exclusivo para dispositivos de marca. As lojas de aplicativos além de disponibilizar aplicativos para *download*, também fornecem informações relativas a eles, dentre os quais: categoria, número de *downloads*, descrição e classificações [Ouyang et al. 2018]. Segundo Finkelstein et al. (2017), essas informações podem ser classificadas em três categorias: relacionadas aos clientes (classificação), ao comércio (número de *downloads* e preço) e informações técnicas (descrição). Além disso, as lojas de aplicativos armazenam uma lista de recursos técnicos que são informados pelos desenvolvedores.

Essas lojas são canais que permitem aos usuários baixar, avaliar um aplicativo e tecer comentários relacionados a experiência de uso, evidenciando aspectos da qualidade do aplicativo. Tais revisões dos usuários servem tanto para desenvolvedores encontrarem erros no código e novos requisitos, quanto para novos usuários na decisão de compra [Chen et al. 2014]. Portanto, em estudos realizados em lojas de aplicativos, é necessário considerar que nesse ambiente existem duas perspectivas: a do usuário de aplicativos e a do desenvolvedor de aplicativos.

### **2.3. Google Play Store na Perspectiva do Usuário**

Google Play Store é uma loja de aplicativos amplamente conhecida. Ela permite que os usuários baixem aplicativos para seu dispositivo Android. Após o *download* e uso eles podem avaliar os aplicativos atribuindo-lhes uma classificação e incluindo um comentário [Ouyang et al. 2018]. Com as avaliações, os usuários expressam suas percepções contribuindo para o aumento ou diminuição da pontuação geral de um aplicativo de acordo com a classificação informada [Sarro et al. 2015]. O usuário também pode atribuir uma classificação para atributos ou características específicas dos

aplicativos no momento da avaliação. A pontuação geral permite que usuários possam decidir quanto à adotar ou não um aplicativo sem a necessidade de leitura de todos os comentários [Fu et al. 2013]. Quando o usuário não deseja mais utilizar o aplicativo, ele pode abandoná-lo removendo-o do seu dispositivo. Todos esses fatores são levados em consideração no estudo de popularidade de um aplicativo [Lu et al. 2017].

#### **2.4. Google Play Store na Perspectiva dos Desenvolvedores**

A loja Google Play Store, além de permitir que os desenvolvedores insiram seus aplicativos no mercado, também permite que eles acessem a uma série de dados sobre seus aplicativos. A Google Play Store disponibiliza um histograma de classificações do aplicativo, além da descrição de funcionalidades que são incluídas pelos próprios desenvolvedores. Dessa forma, por meio do histograma, os desenvolvedores podem se informar quanto à adoção de seus próprios aplicativos ou de aplicativos concorrentes que contenham em sua descrição as mesmas funcionalidades [Fu et al. 2013].

As avaliações dos usuários podem conter em seus comentários dados importantes, os quais permitem a descoberta de necessidades dos usuários para realização de ações corretivas ou implementação de novas funcionalidades nas próximas atualizações por parte dos desenvolvedores [Sarro et al. 2015]. Quando os desenvolvedores lançam uma atualização de aplicativo, há a intenção de que os usuários como um todo realizem a atualização de seus softwares, o que os leva a empregar estratégias de liberação e lançamento de aplicativos para maximizar a adoção por parte dos usuários [Ouyang et al. 2018].

### **3. Trabalhos Relacionados**

Nesta seção são apresentados trabalhos relacionados à avaliação de aspectos ligados à popularidade de aplicativos em lojas de aplicativos. Embora os trabalhos não tratem diretamente da avaliação de aplicativos do domínio de ensino de programação, eles apresentam diversas informações que norteiam a abordagem proposta neste trabalho.

Com o objetivo de obter uma previsão de popularidade (número de *downloads*) de certos aplicativos, Ouyang et al. (2018) realizaram uma análise em lojas de aplicativos. O trabalho mostra que aplicativos de categorias mais populares recebem um maior número de atualizações. Mas, no geral, aplicativos populares podem apresentar uma evolução de popularidade totalmente distinta uns dos outros. Isto é explicado pelos diferentes estímulos que os mesmos recebem, o que leva a necessidade da realização de uma análise multivariada. O trabalho de Ouyang et al. motiva o presente trabalho no entendimento de que existe uma relação entre os diversos fatores que deve ser analisada como um fator para que o aplicativo alcance o sucesso. O trabalho de Ouyang et al. se difere do atual, pois ele se limita ao estudo de popularidade, enquanto o presente trabalho esta relacionado a sucesso e popularidade, e tem uma categoria definida que é o ensino de programação.

Em Finkelstein et al. (2017), foi realizada uma análise da correlação entre preço, classificações dos usuários e popularidade de aplicativos em lojas de aplicativos. A abordagem utilizada consistiu em quatro etapas: (1) coleta de dados de uma loja de aplicativos; (2) extração de atributos de aplicativos (preço, avaliações e descrição); (3) mineração de informações técnicas a partir das descrições textuais; e, por último, (4)

computação de métricas técnicas (requisitos funcionais), métricas de negócio (preço) e métricas de usuários (comentários). O estudo mostra forte correlação entre as avaliações de usuários e a popularidade, uma leve correlação inversa entre preços de aplicativos e avaliações de usuários. Além disso, aplicativos gratuitos geralmente são melhores avaliados que aqueles que são pagos. Desse modo é reforçado a necessidade de um estudo da relação entre os fatores e é apresentado a necessidade de uma análise dos comentários. O trabalho de Finkelstein et al. se difere do trabalho atual, pois ele não se baseia em um domínio específico, além de as métricas de aplicativos apresentadas estarem voltadas somente a popularidade e no trabalho atual além da análise de popularidade, tem se interesse em medir o sucesso através do número de estrelas.

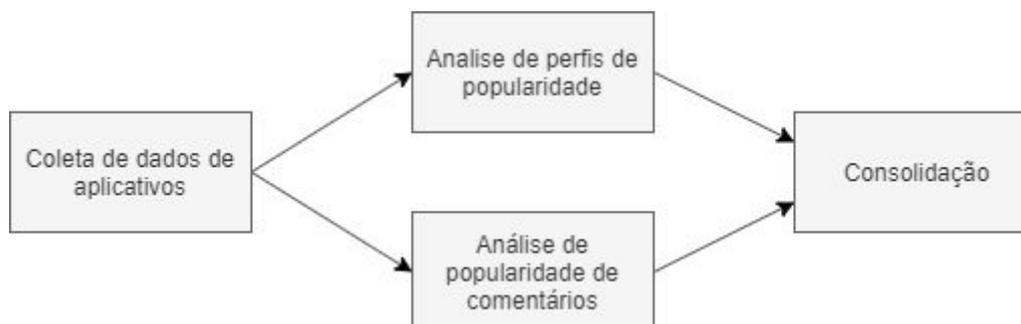
Lee e Raghu (2014) examinaram as principais características do desenvolvedor e do próprio aplicativo que implicam o sucesso em lojas de aplicativos. O estudo investiga como o portfólio de um desenvolvedor na loja de aplicativos está associado ao seu desempenho de vendas ao longo do tempo. Uma análise longitudinal durante 39 semanas, é realizada com o objetivo de analisar o ciclo de vida de aplicativos em espaços semanais. Os resultados mostram a importância de uma constante atualização dos aplicativos disponíveis no portfólio como fator crítico para o sucesso de aplicativos. Sendo tais atualizações facilitadas pela busca de redução de custo de atualizações. Desse modo, esse trabalho colabora para o presente apresentando que não apenas os comentários implicam no sucesso dos aplicativos mas outros fatores também como as características do desenvolvedor, tornando assim necessário a análise de outros fatores para o alcance do sucesso. Diferentemente do trabalho atual, o estudo de Lee e Raghu está voltado a características do desenvolvedor que levam ao sucesso, enquanto o trabalho atual está voltado a características dos usuários e do próprio aplicativo que o leva ao sucesso.

Neste contexto, existem outros estudos que analisaram aplicativos, por meio de uma análise de avaliações de usuários. Chen et al. (2014) propuseram uma ferramenta chamada AR-Miner para filtrar avaliações informativas das que não são informativas para os desenvolvedores, através de análise sentimentos e aprendizagem de máquina. Panichella et al. (2015) combinaram Processamento de Linguagem Natural, Análise de Sentimentos, Análise de Texto e Aprendizagem de Máquina, para extração de comentários que apontem erros ou novas funcionalidades. Fu et al. (2013) propuseram uma ferramenta para avaliar comentários e investigar fatos que levam a má avaliação de um aplicativo. Na análise dos fatores que levam um aplicativo ao sucesso, os estudos contribuem para o presente trabalho por reforçarem a importância das informações contidas nos comentários. Os estudos apresentados se diferem do atual, por estarem voltados a análise de aspectos dos comentários, sendo que o trabalho atual busca apenas a análise de polaridade dos comentários.

#### **4. Metodologia**

Esta seção está dividida em quatro subseções. Na primeira seção são apresentados as ferramentas e métodos utilizados para a coleta de dados de aplicativos de ensino de programação. Na segunda são apresentados as estratégias de análise de métricas quantitativas e agrupamento por valores de métricas para identificação e análise de perfis de popularidade de aplicativos. Na terceira seção são apresentados os métodos e ferramentas utilizados para a obtenção de dados de comentários de usuários em

aplicativos de ensino de programação. Por fim, serão apresentados os métodos de correlação entre o resultados dos estudos de agrupamento e análise de comentários que representa a consolidação do estudo realizado. A Figura 1 demonstra as etapas e a ordem em que o estudo é realizado para a obtenção dos resultados.



**Figura 1. Esquema metodológico**

#### 4.1. Coleta de Dados de Aplicativos

Na coleta dos dados de aplicativos, são buscados e selecionados apenas os aplicativos do domínio de ensino de programação na Google Play Store. Os filtros estabelecidos para esse objetivo são: 1) **palavras-chave:** *programming*, programação, *learn programming*, ensino de programação, *learn code*. 2) **categorias:** educação, livros e referências. É relevante lembrar que os aplicativos selecionados por tais palavras-chave devem estar em alguma das duas categorias para serem categorizados como aplicativos do domínio de ensino de programação. São coletados dados de aplicativos gratuitos, que estejam na língua portuguesa e inglesa, e que retornem valor para todas as métricas descritas.

Dados dos aplicativos de ensino de programação estão disponíveis na Google Play Store e podem ser extraídos através da ferramenta *Google Play Scraper*<sup>2</sup> que a partir de determinados parâmetros retorna um conjunto de informações. A ferramenta *Google Play Scraper* é de código aberto (*open source*) e foi desenvolvida na tecnologia NodeJS<sup>3</sup>. Ela permite que se realize diversas pesquisas na Google Play Store exigindo apenas parâmetros necessários e execuções de métodos de pesquisa.

Para a obtenção dos aplicativos para a coleta de dados para as métricas, é utilizado o método *search* da ferramenta *Google Play Scraper* que a partir de cada palavra chave definida retorna uma lista de aplicativos, o que resulta em cinco listas de aplicativos. A partir das listas obtidas é efetuado chamadas ao método *app* que é responsável por obter atributos que representam as métricas quantitativas dos aplicativos coletados, e também para o método *reviews* que retorna diversos detalhes relacionados aos comentários de cada aplicativo obtido.

Os dados utilizados neste trabalho foram coletados no dia 13 de Outubro de 2019. Ao todo, foram coletados dados de popularidade de 433 aplicativos com idioma

<sup>2</sup> *GooglePlayScraper* - Acesso disponível em: <<https://github.com/facundooolano/google-play-scraper>>

<sup>3</sup> Sobre o NodeJs - Acesso disponível em: <<https://nodejs.org/en/about/>>

em inglês e português. Também foram coletados os *reviews* desses aplicativos, que são comentários deixados pelos usuários na página do aplicativo na Google Play Store. Ao todo foram coletados 78.727 *reviews*. Todos os dados coletados, assim como os códigos produzidos nas análises realizadas neste trabalho, estão disponíveis na plataforma GitHub, no seguinte link: <https://github.com/luizfernando1996/TCC>.

## 4.2. Análise de perfis de popularidade

Na análise de perfis de popularidade são utilizadas informações a partir dos atributos quantitativos que podem estar associados ao sucesso dos aplicativos. Essas informações, em sua maioria, são valores numéricos usados para medir os níveis de cada uma delas que implica no sucesso do aplicativo.

1. **Número de avaliações:** Essa métrica informa a quantidade de usuários que avaliaram o aplicativo e não teceram comentários.
2. **Número de instalações:** Essa métrica representa o número aproximado de instalações que o aplicativo obteve até o momento da análise.
3. **Número de comentários:** Número de comentários realizados pelos usuários no aplicativo.

Para realização da análise da influência de cada uma das métricas no número de estrelas dos aplicativos que compõem a categoria de ensino de programação, é aplicada a correlação de Spearman. Dessa forma, a partir do resultado do teste de correlação é possível descobrir se as métricas influenciam individualmente no sucesso dos aplicativos que compõem tal categoria. Não sendo encontrada uma correlação do valor das métricas com o número de estrelas, torna-se necessário a realização do agrupamento dos aplicativos de modo a verificar se, no conjunto, as métricas de popularidade explicam o sucesso dos aplicativos.

Para o agrupamento dos aplicativos, é utilizado o algoritmo k-means. A escolha desse método se dá pela sua ampla utilização na análise de perfis [Ponciano e Brasileiro 2014]. No agrupamento são utilizadas as métricas: número de instalações, número de comentários e número de avaliações. Sendo a entrada de dados uma matriz  $|N| \times 3$  onde cada linha se refere a um aplicativo e as colunas as métricas analisadas. Como o resultado dos agrupamentos requer que os valores relativos dos parâmetros sejam agrupados, é necessária uma normalização dos parâmetros antes do agrupamento. Para a normalização dos valores em um intervalo entre 0 e 1, é utilizada a fórmula  $x_i = (x_i - x_{min}) / (x_{max} - x_{min})$ , onde  $x$  representa as métricas e  $i$  os aplicativos. Para a definição do número de grupos a serem analisados é considerado o valor que otimiza o *trade-off* entre o máximo valor da silhueta média e o menor valor da soma dos quadrados dentro do grupo.

Para entendermos melhor os grupos descobertos pelo algoritmo, foram analisados dois itens importantes. Primeiro, os centróides que representam cada grupo. Segundo, como as diferentes métricas agrupadas explicam o grupo. Isto é, como os centróides dos grupos interferem no número de estrelas alcançado pelos aplicativos.

## 4.3. Análise de Polaridade dos Comentários

Para medir sucesso de um aplicativo, além de considerar o número de estrelas que o aplicativo recebeu, também considera-se a polaridade dos comentários deixados pelos usuários. A análise de polaridade dos comentários é feita por meio da ferramenta *Sentiment*<sup>4</sup>. A ferramenta *Sentiment* é um módulo do Node.js. que usa a lista de palavras AFINN-165<sup>5</sup> e o Ranking de Sentimentos Emoji<sup>6</sup> para executar a análise de sentimentos em blocos arbitrários de texto de entrada. A *AFINN* é uma lista de palavras classificadas para valência com um número inteiro entre menos cinco (negativo) e mais cinco (positivo). O *Ranking de Sentimentos Emoji* é um recurso que permite a análise automatizada de emoticons.

O resultado da análise da ferramenta *Sentiment* é um valor de polaridade positiva, neutra ou negativa na escala de 5 a -5. Neste presente trabalho tal valor auxilia a obter métricas da quantidade de comentários positivas (notas acima de 0), negativos (nota abaixo de 0) e neutras (nota igual a 0) de cada um dos aplicativos. A partir de tais métricas se efetua o cálculo da polaridade de cada aplicativo por meio da Equação 1

$$Polaridade\ do\ aplicativo = \frac{comentário\ positivos}{(comentários\ positivos + comentários\ negativos)} \quad (1)$$

O resultado da equação, como se pode observar, é um valor entre 0 e 1. Quando o resultado é um valor igual a 1, isso indica que o aplicativo possui apenas comentários positivos. Quando o resultado é um valor igual a 0, isso indica que o aplicativo possui apenas comentários negativos. Valores intermediários indicam a proporção de comentários positivos em relação à total de comentários nas duas classes. Naturalmente, aplicativos neutros não entram nessa análise de polaridade.

#### 4.4. Consolidação

Na consolidação dos resultados, é analisado o relacionamento entre os resultados da análise de perfis de popularidade e os resultados da análise de polaridade dos comentários a fim de identificar os fatores que levam aplicativos de ensino de programação ao sucesso na Google Play Store. A análise consiste em observar se existe um perfil de aplicativos que tende a alcançar mais estrelas ou a ter mais comentários de polaridade positiva. Também analisa-se, usando o coeficiente de Spearman, a correlação existente entre número de estrelas recebidas pelos aplicativos e polaridade dos comentários deixados pelos usuários.

### 5. Resultados

Nesta seção, apresentam-se os resultados obtidos. Primeiro são apresentados os resultados obtidos na correlação entre as variáveis com o número de estrelas, segundo são apresentados os centróides obtidos no agrupamento, depois os dados relativos a análise dos comentários, e por último a interrelação entre estes dados a fim de obter as características próprias dos aplicativos que compõem a categoria estudada.

#### 5.1 Análise de Correlação entre Métricas de Popularidade e Número de Estrelas

<sup>4</sup> *Sentiment* - Acesso disponível em: <<https://www.npmjs.com/package/sentiment>>

<sup>5</sup>AFINN-165 - Acesso disponível em: <[http://www2.imm.dtu.dk/pubdb/views/publication\\_details.php?id=6010](http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010)>

<sup>6</sup>Ranking de Sentimentos Emoji - Acesso disponível em: <<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0144296>>

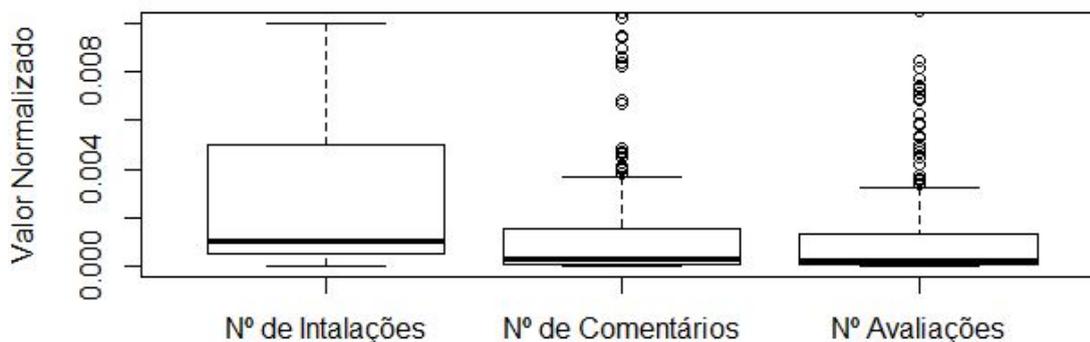
Na análise de correlação entre as métricas com o número de estrelas é observado que ao se aplicar a correlação de Spearman, não há uma forte correlação entre o número de estrelas e as demais métricas individualmente. Isso é demonstrado na Tabela 1 que representa a matriz de correlação de Spearman.

**Tabela 1. Matriz de correlação de Spearman**

	Número de Estrelas	Número de Instalações	Número de comentários	Número de avaliações
Número de Estrelas	1	-0.0561	0.0809	0.0503
Número de Instalações	-0.0561	1	0.8792	0.9151
Número de comentários	0.0809	0.8792	1	0.9476
Número de avaliações	0.0503	0.9151	0.9476	1

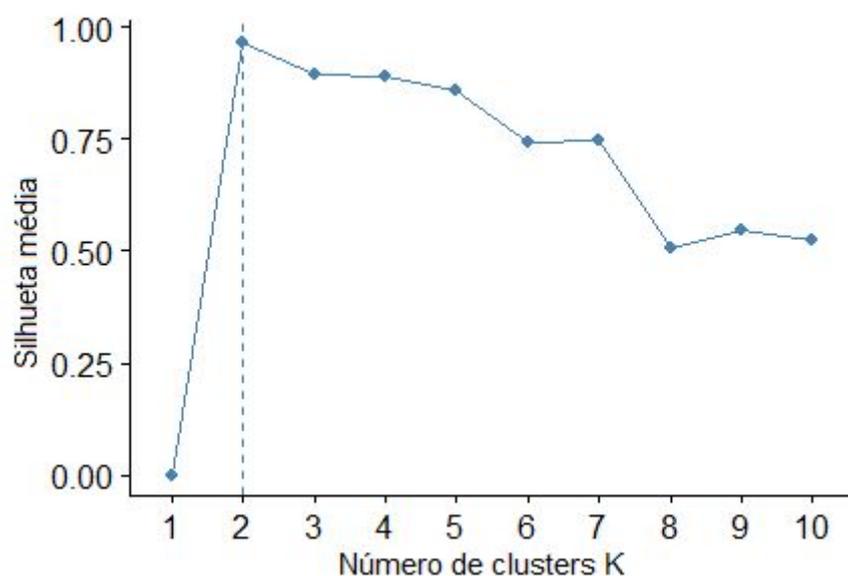
## 5.2 Normalização e Agrupamento

Na Figura 2 pode-se verificar uma análise de dispersão dos valores das métricas usadas no agrupamento dos aplicativos, utilizando Boxplot. Os valores das métricas encontram-se normalizados conforme descrito na seção de metodologia.

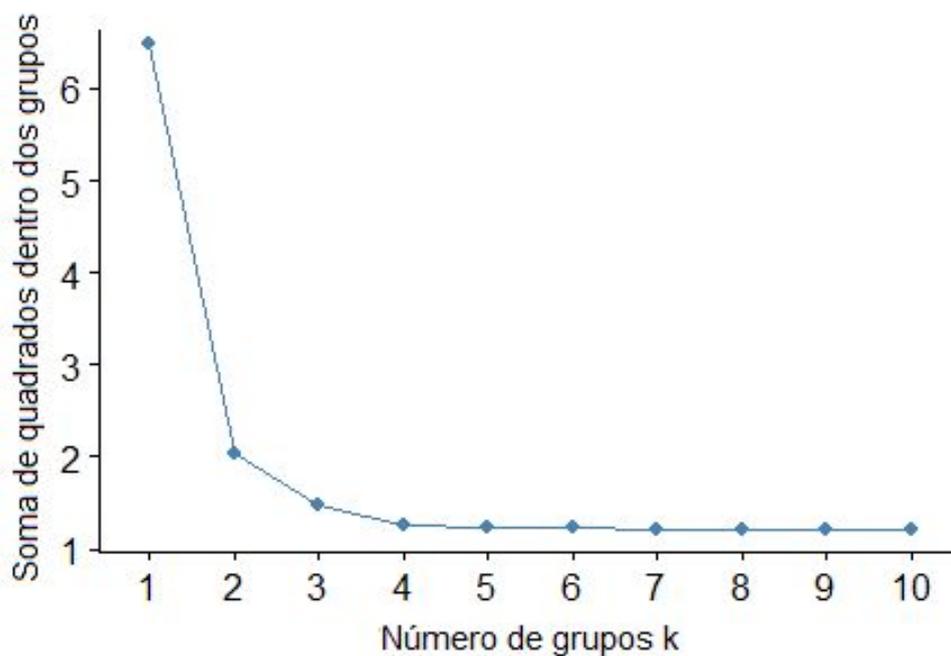


**Figura 2. Boxplot mostrando de forma normalizada a distribuição das métricas consideradas para agrupamento.**

Para a realização do agrupamento das variáveis, é realizada a análise de dois gráficos para encontrar o melhor número de grupos para as variáveis, onde avaliamos a qualidade do agrupamento calculando a soma dos quadrados dentro do grupo na Figura 4 e a largura média da silhueta na Figura 3. O resultado da qualidade dos agrupamentos, quando o número de grupos varia entre 2 e 10 pode ser visto nas Figuras 3 e 4. Esses números mostram que 4 é o número de grupos que melhor otimiza o *trade-off* entre o número de grupos e a soma dos quadrados dentro do grupo.



**Figura 3. Análise de agrupamento K-means, silhueta média.**

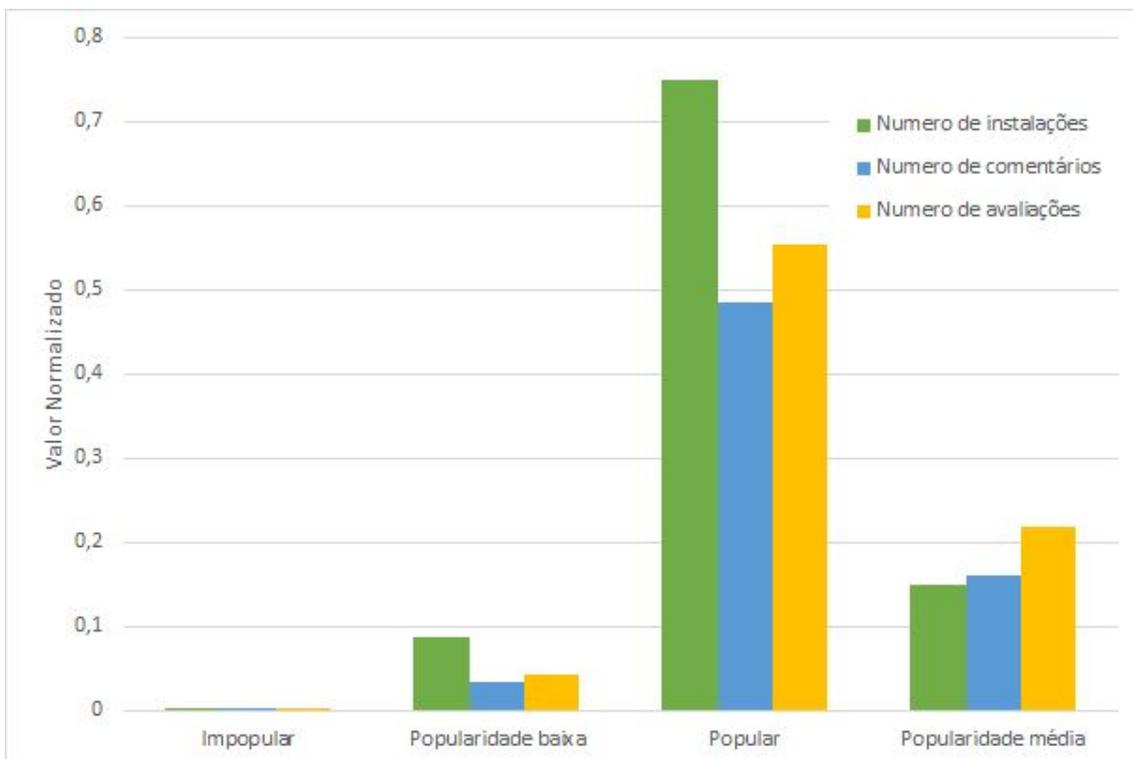


**Figura 4. Análise do agrupamento K-Means, soma dos quadrados no grupo.**

Ao observar os gráficos, é possível definir que o melhor número de grupos é 4, pois ao avaliar a soma dos quadrados na Figura 4, percebe-se que a reta se estabiliza no menor valor do intervalo de grupos após quatro grupos, e na Figura 3 o número de grupos 4 é o que tem a terceira maior silhueta média, o que reforça este número de grupos como o ideal para a análise realizada.

### 5.3 Perfis

Nesta análise, estabelecemos nomes para os grupos que representa a principal característica de cada um. Estes grupos representam diferentes tipos de aplicativos, nomeados como: Impopular; Popularidade Baixa; Popularidade Média; Popular. Na Figura 5, são mostrados os centróides que representam cada um dos perfis e como eles se diferem. O eixo horizontal apresenta cada grupo de aplicativos e cada barra representa cada métrica analisada, o eixo vertical indica como os perfis pontuam em cada uma das métricas.



**Figura 5. Medidas de cada métrica em cada grupo de aplicativos de ensino de programação. Os grupos são representados pelos seus centróides gerado pelo K-Means.**

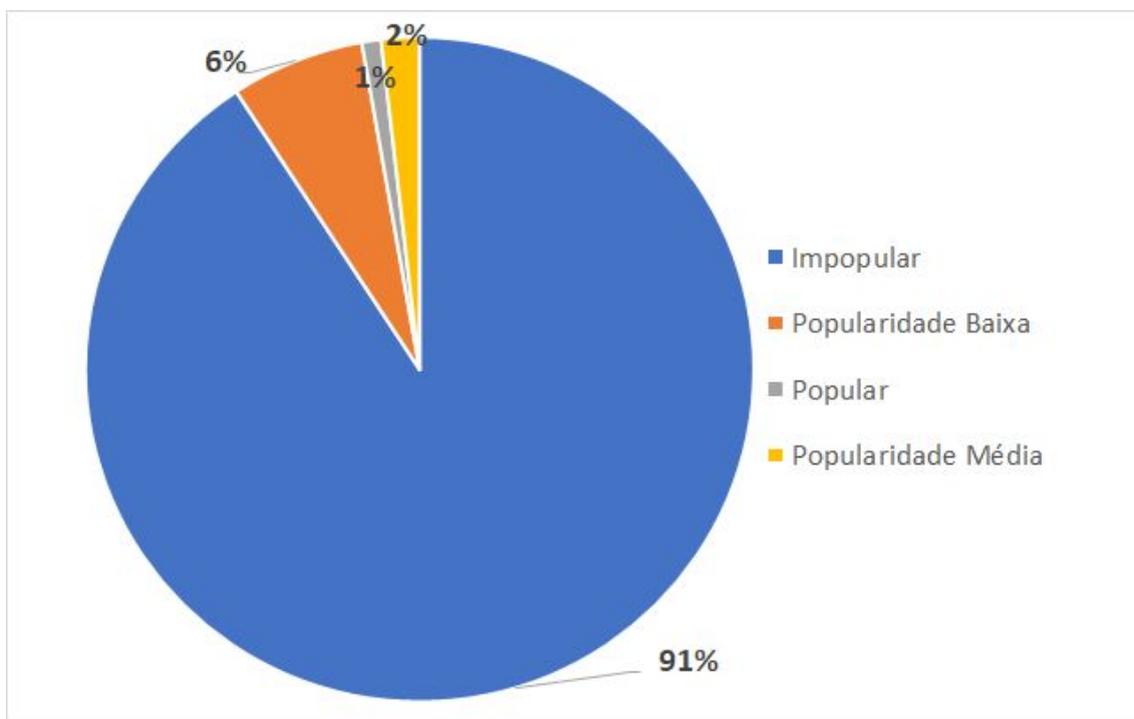
Perfil **Impopular**. Este Perfil se caracteriza pelo baixo número de instalações, baixo número de comentários e baixo número de avaliações. Pode-se verificar que estes aplicativos não despertaram interesse de usuários o que configura a impopularidade dos mesmos. Existe uma baixa porcentagem de aplicativos para este perfil sendo 393 aplicativos que representam 91% da amostra (Figura 6).

Perfil **Popularidade Baixa**. Este Perfil se caracteriza por um número maior de instalações do que de avaliações e comentários. Pode-se verificar que estes aplicativos apesar do número de instalações apresentado, não despertaram o interesse dos usuários para tecer comentários e realizar avaliações. Existe uma baixa porcentagem de aplicativos para este perfil sendo 28 aplicativos que representam 6% da amostra (Figura 6).

Perfil **Popularidade Média**. Este Perfil se caracteriza pelo número de instalações menor que o número de comentários e avaliações em relação aos valores individualmente comparados aos aplicativos que compõem a amostra. Pode-se verificar que para estes aplicativos os usuários comentam e avaliam, mas o número de instalações está entre os menores valores da amostra. Existe uma baixa porcentagem de aplicativos para este perfil sendo 8 aplicativos que representam 2% da amostra (Figura 6).

Perfil **Popular**. Este Perfil se caracteriza pelos maiores valores para as métricas na amostra analisada. Pode-se verificar que estes aplicativos despertaram interesse dos usuários em downloads e avaliações o que configura a popularidade dos mesmos. Existe uma alta porcentagem de aplicativos para este perfil sendo 4 aplicativos que representa 1% da amostra (Figura 6).

A Figura 6 apresenta como cada um dos perfis se diferem em termos de números de aplicativos, conforme descrito nos perfis acima.



**Figura 6. Porcentagem da quantidade de aplicativos por perfil**

#### 5.4. Análise de Comentários

A Tabela 2 apresenta os valores das métricas da quantidade de comentários positivas, negativas e neutras obtidas a partir do resultado da ferramenta *Sentiment* para uma pequena amostra de aplicativos utilizadas no presente trabalho. Esta tabela também apresenta o número de estrelas de cada aplicativo e o valor final da fórmula da polaridade do aplicativo que se baseia no número de comentários positivos e negativos.

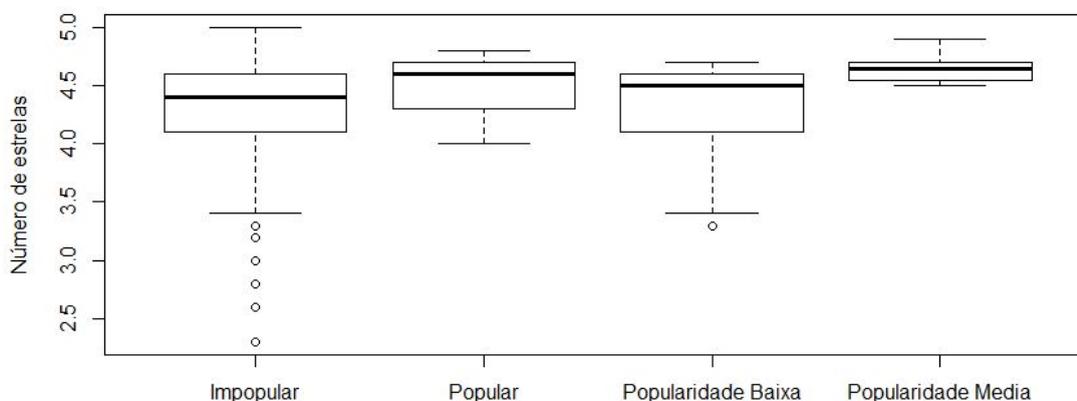
**Tabela 2. Análise de Sentimento dos aplicativos**

Bundle id	Número de comentários positivos	Número de comentários negativos	Número de comentários neutros	Número de comentários totais	Polaridade do aplicativo	Número de estrelas do app
wscubetech.android	0	0	8	8	0	4
com.aniketo.ds	44	0	8	52	1	4.8
com.infoland.oops	8	4	4	16	0.67	3.7

Na análise de correlação entre a polaridade do aplicativo com o número de estrelas é observado que ao se aplicar a correlação de Spearman, o coeficiente resultante é 0.39. Isso indica uma correlação fraca entre as métricas, levando a necessidade de uma análise da correlação dos perfis com a polaridade do aplicativo e outra a análise da correlação dos perfis com o número de estrelas.

### 5.5 Relação entre as Medidas de Sucesso

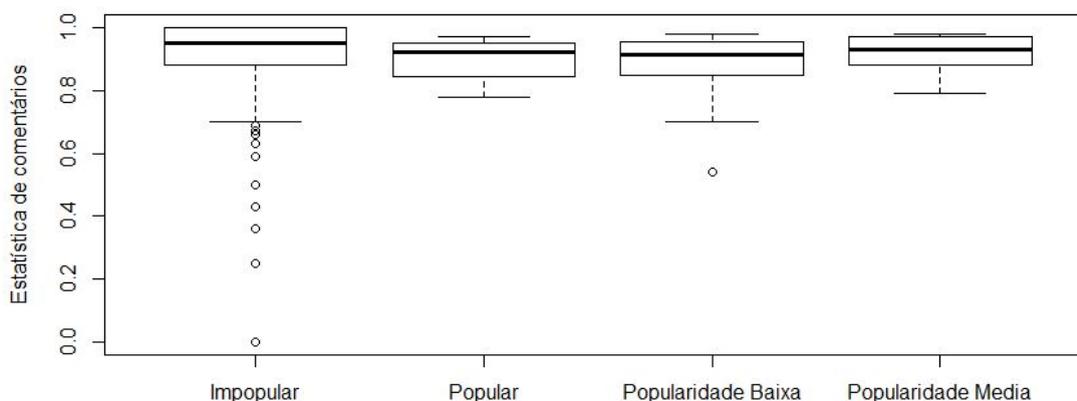
Relacionando os grupos com o número de estrelas como pode ser observado na Figura 7, entende-se que o grupo de popularidade média apresenta um maior número de estrelas variando menos que os demais grupos. Dessa forma este grupo apresenta uma maior tendência a obtenção de sucesso, apesar de não ser o de maior popularidade.



**Figura 7. Boxplot da distribuição do número de estrelas para cada grupo**

Ao observar o gráfico da Figura 8, é possível verificar que para a polaridade do aplicativo o grupo popular se destaca em relação a maior aceitação representada pelos usuários, mas o grupo de popularidade média possui uma menor variação dos valores para a polaridade, e representa o segundo maior valor. Dessa maneira é possível

verificar que o grupo de popularidade média possui um boa aceitação dos usuários, reforçando a sua tendência a obtenção de sucesso.



**Figura 8. Boxplot da distribuição da polaridade dos comentários para cada grupo**

## 6. Conclusões

Neste estudo foram abordadas as seguintes questões: Primeiro, quais informações pode-se obter sobre aplicativos disponíveis no Storage da Google Play para avaliação de sucesso de aplicativos de ensino de programação. Segundo, quais as similaridades entre os aplicativos de sucesso na categoria estudada. Por fim, como os grupos de aplicativos com suas similaridades podem ser utilizados para identificar quais modelos de aplicativos tendem a obter uma melhor aceitação por parte dos usuários desta categoria.

Nossos resultados mostram que os aplicativos abordados neste trabalho compartilham várias similaridades e diferenças em termos de valores das métricas estudadas. Eles foram rotulados de acordo com o número de instalações em Impopular; Popularidade baixa; Popularidade Média e Popular. Entretanto, conforme pode se observar na figura 5 a utilização de outras variáveis (número de comentários, número de avaliações) em detrimento da variável número de instalações para a atividade de rotulação dos grupos, não mudaria o resultado final da atividade. Sendo assim, a definição dos rótulos dos grupos de acordo com o critério de popularidade pode ter ocultado características individuais de cada grupo. Isto é, o grupo rotulado como popularidade média para qualquer análise por meio das variáveis número de instalações, número de comentários ou número de avaliações será o terceiro grupo em ordem crescente de valor para tal variável, entretanto, este grupo é o único com o maior valor para a variável número de avaliações do que os outros.

A respeito da distribuição dos aplicativos pelos grupos, a maior porcentagem de aplicativos encontra-se no grupo de aplicativos impopulares. Tal grupo conforme pode se observar também é responsável por possuir uma maior quantidade de aplicativos outliers. A menor porcentagem de aplicativos encontra-se no grupo de popularidade que é responsável pelos valores mais altos das métricas normalizadas.

Sobre o resultado da análise de comentários, pode-se perceber que o número de estrelas não está fortemente correlacionado a qualidade dos comentários segundo a correlação de Spearman de 0,39 efetuados pelos usuários. Também foi observado que para os aplicativos que compõem o domínio estudado, os valores das métricas número de estrelas e polaridade de comentários, tem seus valores altos para a maior parte da amostra. Logo, é possível inferir que a maior parte dos aplicativos tem um número entre 4 e 5 estrelas. Questões que surgem disto e que futuros trabalhos deveriam explorar são: i) O domínio estudado é que influenciou nisso? ii) Os aplicativos serem de ensino é que influenciam o resultado?

## **6.1 Limitações**

O estudo realizado apresentou diversas limitações, algumas delas durante todo o estudo e outras em apenas algumas fases. É interessante que futuros trabalhos entendam tais limitações descritas e extingam elas em seus trabalhos ou as explorem mais para que outros trabalhos possam usufruir de tais colaborações.

Limitações transversais tais como em relação ao: i) tempo de realização do trabalho, pois análises mais aprofundadas exigem maior tempo de análise, para se atingir resultados de maior expressão; ii) domínio estudado, isto é, nossas conclusões são válidas para o domínio estudado, entretanto, não se conhece se são válidas também para outros domínios; iii) tamanho da amostra de aplicativos que compõem o domínio de ensino de programação, o que conseqüentemente dificulta análises mais aprofundadas sobre as características comuns e singulares de cada perfil.

A limitação existente na primeira atividade da metodologia que tratou a coleta dos dados de aplicativos de programação, foi a coleta assertiva de aplicativos de ensino de programação. Como a Google Play Store não possui tal categoria de aplicativos expressamente visível em sua plataforma, foi necessário a definição de diversos filtros para se obter apenas os aplicativos nesta categoria. Logo, foi inferido que tais filtros obtêm todos os aplicativos de ensino de programação. Entretanto, o quanto tais limitações impactaram na redução da amostra de aplicativos e na melhor expressão de nossos resultados é algo a ser investigado.

Na atividade de análise da polaridade dos comentários, o estudo se limitou a avaliar e quantificar os comentários como bom, neutro ou ruim. Problemas conseqüentes são o número de comentários classificados como positivos, negativos ou neutros mas sem relação a um aplicativo, tais como “Que dia belo” (hipótese), e que foram relacionados às métricas de quantidade de comentários positivos, negativos ou neutros de um aplicativo. Logo, tais comentários podem ter influenciado nos resultados do estudo, e futuros trabalhos devem adotar ações para minimizar este risco. Outra questão advinda desta limitação, é que a análise de polaridade dos comentários não validou se o comentário faz sentido em relação ao número de estrelas atribuído a ele. Essa questão é um risco, porque os comentários irônicos podem ser melhor classificados com polaridades opostas de acordo com o comentário se a atividade de classificação das polaridades dos comentários avaliasse além do texto escrito também o número de estrelas. O quanto comentários foram incorretamente classificados é uma segunda questão desta etapa, que pode ter impactado nos resultados do estudo.

## 6.2 Trabalhos Futuros

Como trabalhos futuros, em particular, seria relevante conduzir estudos que abordem: i) A extensão desse estudo a domínios diferentes do ensino de programação, para verificar se os resultados obtidos neste trabalho também ocorrem em outros domínios; ii) Analisar o motivo de grande parte dos aplicativos de ensino de programação, terem um alto número de estrelas e polaridade de comentários, investigando aspectos como por exemplo, o perfil dos usuários que avaliam tais aplicativos; iii) Investigar melhor os grupos definidos neste trabalho a fim de se definir rótulos que representam melhor as particularidades de cada grupo; iv) Investigar as particularidades dos aplicativos que compõem o grupo de popularidade média; v) Realizar uma análise de comentários que levem em consideração os aspectos que são abordados pelos usuários, e não apenas a polaridade dos comentários, a fim de entender melhor os requisitos do grupo de ensino de programação. Esses estudos propostos podem trazer conhecimento novo acerca de informações que rodeiam o escopo abordado por este estudo.

## Referências Bibliográficas

- CHEN, Ning; LIN, Jialiu; HOI, Steven C. H.; XIAO, Xiaokui; ZHANG, Boshen (2014). "AR-miner: mining informative reviews for developers from mobile app marketplace". In: Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), p.767-778. DOI: 10.1109/RE.2015.7320414
- FINKELSTEIN, Anthony; HARMAN, Mark; JIA, Yue; MARTIM, William; SARRO, Frederica; ZHANG, Yuanyuan (2017). "Investigating the relationship between price, rating, and popularity in the Blackberry World App Store". In: IEEE Software, v. 87, n. 0, p. 119-139. DOI: 10.1016/j.infsof.2017.03.002
- FLORIANO, Diogo; KAHL, Marcelo (2012). "Computação ubíqua, tecnologia sem limites." In: IV Semana Acadêmica de Sistemas de Informação da UDESC/CEAVI. Disponível em: <[http://www.ceavi.udesc.br/arquivos/id\\_submenu/387/diogo\\_floriano\\_marcelo\\_kahl\\_computacao\\_ubiqua.pdf](http://www.ceavi.udesc.br/arquivos/id_submenu/387/diogo_floriano_marcelo_kahl_computacao_ubiqua.pdf)>. Acesso em: 25 maio. 2019.
- FU, Bin; LIN, Jialiu; LI, Lei; FALOUTSOS, Christos; HONG, Jason; SADEH, Normant (2013). "Why people hate your app: making sense of user feedback in a mobile app store". In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13), p. 1276-1284. DOI: 10.1145/2487575.2488202
- KHALID, Hammad; NAGAPPAN, Meiyappan; HASSAN, Ahmed E. (2016). "Examining the relationship between FindBugs Warnings and App Ratings?". In: IEEE Software, v. 33, n. 4, p. 34-39. DOI: 10.1109/MS.2015.29
- KHALID, Hammad; SHINAB, Emad; NAGAPPAN, Meiyappan e HASSAN, Ahmed E. (2015). "What Do Mobile App Users Complain about?". In: IEEE Software, vol. 32, no. 3, p. 70-77. DOI: 10.1109/MS.2014.50
- LEE, Gunwoong; RAGHU, T. S. (2014). "Determinants of Mobile Apps Success: Evidence from App Store Market". In: Journal of Management Information Systems, v.31, n.2, p. 133-170. DOI: 10.2753/MIS0742-1222310206

- LU, Xuan; CHEN, Zhenpeng; LIU, Xuanzhe; LI, Huoran; XIE, Tao; MEI, Qiaozhu (2017). "Predicting App Adoption by Learning the Correlation between Developer - Controllable Properties and User Behaviors". In: Journal Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, v. 1, n. 3, p. 79-109. DOI: 10.1145/3130944
- MARÇAL, Edgar; ANDRADE, Rossana; RIOS, Riverson. (2005). "Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual". RENOTE - Revista Novas Tecnologias na Educação, CINTED-UFRGS . v. 3, n.1. DOI: 10.22456/1679-1916.13733
- MCLLROY, Stuart; SHANG, Weiyi; ALI, Nasir (2017). "Is It Worth Responding to Reviews?". In: IEEE Software, v. 34, n. 3, p. 64-71. DOI: 10.1109/MS.2015.149
- OUYANG, Yi; GUO, Bin; GUO, Tong; CAO, Longbing; YU, Zhiwen (2018). "Modeling and Forecasting the Popularity Evolution of Mobile Apps: A Multivariate Hawkes Process Approach". In: Proc. ACM Interact. Mob. Wearable Ubiquitous Technol, p.182-205. DOI: 10.1145/3287060
- PANICHELLA, Sebastiano; SORBO, Andreia D.; GUZMAN, Emitza; VISAGGIO, Corrado. A.; CANFORA, Gerardo; GALL, Harald C. (2015). "How can i improve my app? Classifying user reviews for software maintenance and evolution". In: IEEE International Conference on Software Maintenance and Evolution (ICSME), Bremen, p. 281-290. DOI: 10.1109/ICSM.2015.7332474
- PONCIANO, Lesandro; BRASILEIRO, Francisco (2014). Finding Volunteers' Engagement Profiles in Human Computation for Citizen Science Projects. Human Computation, v. 1, n. 2.
- SARRO, Federica; ALSUBAIHIN, Afnan; HARMAN, Mark; JIA, Yue; MARTIN, William; ZHANG, Yuanyuan (2015). "Feature lifecycles as they spread, migrate, remain, and die in App Stores". In: IEEE 23rd International Requirements Engineering Conference (RE), p. 76-85. DOI: 10.1109/RE.2015.7320410
- SHEIN, Esther (2014). "Should Everybody Learn to Code?". In: Communications of the ACM, v. 57, n. 2, p. 16-18. DOI: 10.1145/2557447
- OLIVEIRA, Millena; SOUZA, Anderson; BARBOSA, Aline; BARREIROS, Emanuel (2014). "Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência". In: XXXIV Congresso da Sociedade Brasileira de Computação, p. 1525-1534.