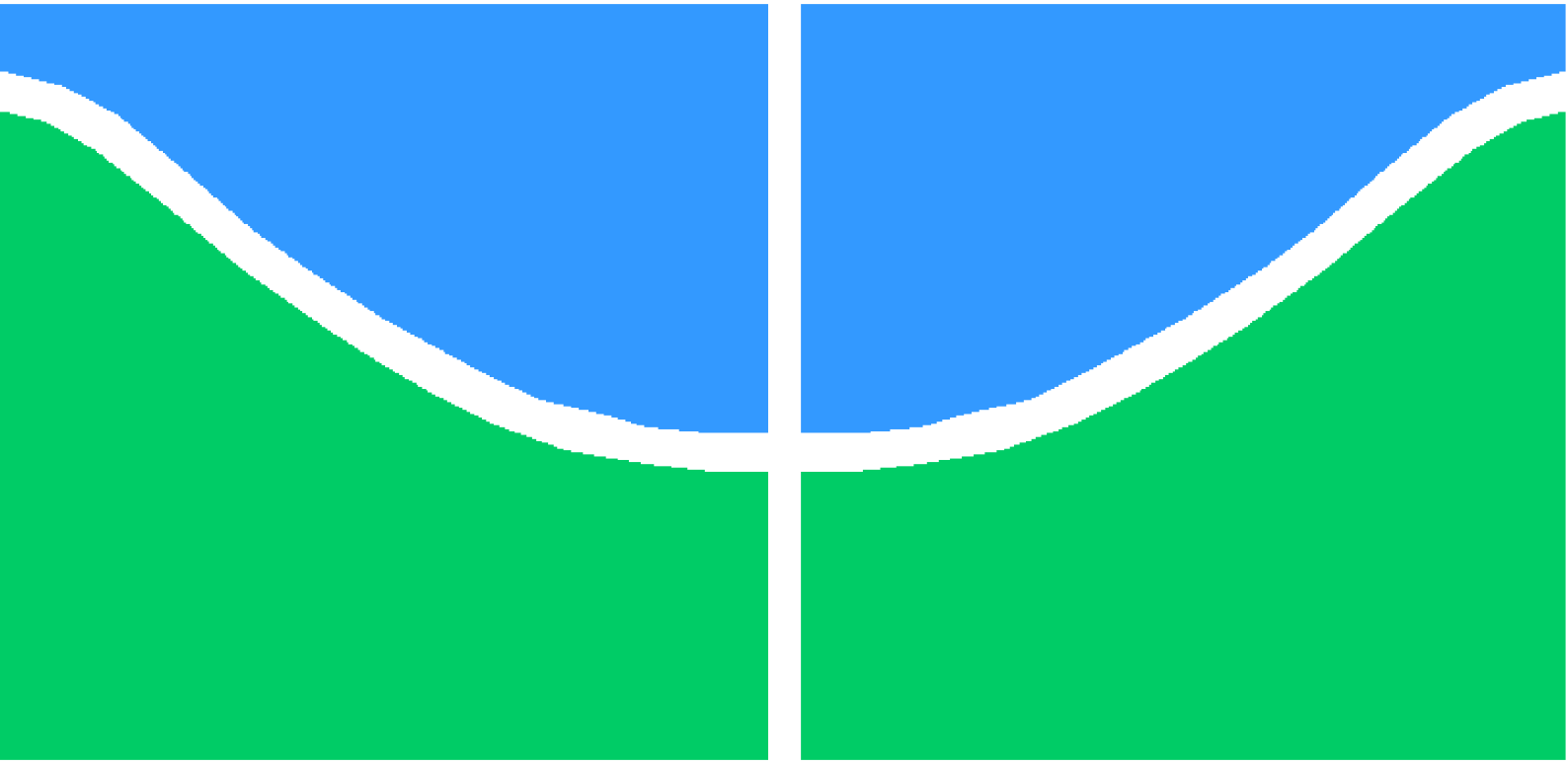

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

# Extração de informações sobre usabilidade a partir de comentários dos aplicativos feitos na *Play Store*

Autor: Augusto Moreno Vilarins Cardoso da Silva, Ícaro Pereira de Oliveira

Orientador: Msc. Cristiane Soares Ramos

Brasília, DF

2021

Augusto Moreno Vilarins Cardoso da Silva, Ícaro Pereira de Oliveira

# Extração de informações sobre usabilidade a partir de comentários dos aplicativos feitos na *Play Store*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Msc. Cristiane Soares Ramos

Coorientador: Msc. Ricardo Ajax Dias Kosloski

Brasília, DF

2021

Augusto Moreno Vilarins Cardoso da Silva, Ícaro Pereira de Oliveira

Extração de informações sobre usabilidade a partir de comentários dos aplicativos feitos na *Play Store*/ Augusto Moreno Vilarins Cardoso da Silva, Ícaro Pereira de Oliveira. – Brasília, DF, 2021-

69 p. : il. (algumas color.) ; 30 cm.

Orientador: Msc. Cristiane Soares Ramos

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2021.

1. usabilidade. 2. mineração-de-dados. I. Msc. Cristiane Soares Ramos. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Extração de informações sobre usabilidade a partir de comentários dos aplicativos feitos na *Play Store*

CDU 02:141:005.6

Augusto Moreno Vilarins Cardoso da Silva, Ícaro Pereira de Oliveira

# Extração de informações sobre usabilidade a partir de comentários dos aplicativos feitos na *Play Store*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 18 de novembro de 2021:

---

**Msc. Cristiane Soares Ramos**
Orientadora

---

**Msc. Ricardo Ajax Dias Kosloski**
Coorientador

---

**Msc. Rafael Fazzolino Pinto Barbosa**
Convidado 1

Brasília, DF
2021

*Este trabalho é dedicado à todos que participam em nossas vidas, dedicando tempo, suor e sangue pelos nossos sonhos. Nosso muito obrigado.*

*Então, cerra os punho, sorria*
*E jamais volte pra sua quebrada de mão e mente vazia*
*(Emicida)*

# Resumo

Construir um aplicativo utilizável com base nas necessidades do usuário pode se tornar um desafio para desenvolvedores de software. Mesmo que haja uma preocupação com o feedback do usuário, às vezes o grande volume de dados torna a revisão manual dos comentários uma tarefa difícil. Este trabalho fornece um processo semiautomático para extrair informações de feedback de usuários de aplicativos *mobile* na loja do Google *Play*. A proposta é desacoplar uma classificação de dados com o contexto da aplicação por meio de categorização dos dados utilizando heurísticas de usabilidade. Após as fases de extração, transformação e análise e carregamento, o resultado é uma informação e útil para monitorar a qualidade de uso do aplicativo analisado através de um dashboard que consome os dados previamente processados.

**Palavras-chave**: usabilidade. qualidade em uso. feedback do usuário. análise de dados. mobile.

# Abstract

Building a usable application based on user needs can become a challenge for software developers. Even though there is a concern for user feedback, sometimes the sheer volume of data makes manually reviewing comments a difficult task. This work provides a semi-automatic process to extract feedback information from users of mobile applications in the Google Play. The proposal is to decouple data classification from the application context through data categorization through usability heuristics. After the extraction, transformation and analysis and loading phases, the result is informative and useful for monitoring the quality of use of the analyzed application through a panel that consumes previously processed data.

**Key-words**: usability. quality in use. user feedback. data analysis. mobile.

# Lista de ilustrações

# Lista de tabelas

# Sumário

# 1 Introdução

Neste capitulo será introduzido o trabalho de concusão de curso pretendido. O capitulo apresenta as seções de contextualização, problema e justificativa, objetivos geral e específicos e organização geral do trabalho.

## 1.1 Contextualização

Durante a história da Engenharia de *Software*, entre muitas áreas, os estudos de Interação Humano Computador e de Qualidade do produto de *software* costumam atuar mutuamente na área de Usabilidade. Entre outras coisas, a usabilidade estuda formas de mensurar e melhorar a facilidade da interação entre o usuário e a ferramenta, de forma a realizar uma tarefa específica.

Problemas de usabilidade costumam causar frustração e impactam diretamente na avaliação do produto de *software* feita pelas pessoas usuárias. Portanto, há uma preocupação para que os usuários dos produtos façam parte do processo de qualidade e tenham uma boa experiência ao utilizá-los.

Atualmente, existem diversas formas, *top-down*, de coletar e analisar a qualidade do *software* em uso, sob a ótica da usabilidade. Nessas técnicas, a equipe responsável pelo produto busca o público-alvo e estuda os resultados obtidos. São algumas delas: testes de usabilidade, entrevistas, quantificação da satisfação subjetiva do usuário ao completar tarefas etc.

Além disso, também existem espaços abertos de *feedback* que são utilizados pelos usuários para relatar suas frustrações e satisfação ao utilizar a ferramenta, invertendo, assim, a forma como essa informação chega à equipe, já que é um movimento *bottom-up*. Ou seja, a iniciativa parte dos usuários e não da equipe.

Alguns desses espaços são fóruns, redes sociais ou seções de *feedbacks* dos usuários, geralmente fornecidos pelas lojas de aplicativos *mobile*, como a *Play Store* para os sistemas *Android* e *App Store* para os sistemas *iOS*.

Esse movimento *bottom-up* sugere que as frustrações que são expressadas pelo usuário resultem em iniciativas de avisar à equipe responsável e outros usuários sobre experiência malsucedida ao realizar determinada tarefa.

*Apps* com missões críticas, como fornecer um *login* único para serviços governamentais, motivam uma grande quantidade de usuários a comentar sobre sua experiência nos espaços mencionados.

Porém, sob a visão da equipe responsável pelo desenvolvimento do *software*, alcançar as informações que podem ser extraídas desses aplicativos pode não ser uma missão fácil. Dado o volume de dados ou a inexperiência da equipe, entre outros fatores, reconhecer, classificar e analisar esses comentários torna-se difícil.

## 1.2 Problema e justificativa

Em equipes que, embora se preocupem com a experiência do usuário, não coletam suficientemente métricas de usabilidade, seja por limitação técnica ou de planejamento, alguns problemas, de usabilidade acabam sendo detectados apenas em ambiente de produção, como por exemplo: *bugs* que ocorrem em dispositivos ou versões de sistema específicas, falhas de implementação, inadequações funcionais etc.

Os dados extraídos desses comentários podem ajudar na priorização e no planejamento de melhorias e correções do produto. Porém, analisar essa quantidade de dados apresenta dificuldades, como realizar a coleta de comentários, extrair sobre quais funcionalidades cada comentário trata, agrupar os comentários extraídos dentro de métricas de usabilidade e ter uma visão histórica de quais versões apresentam certo problema.

Porém, ultrapassando essas barreiras e utilizando as informações obtidas, a equipe pode obter vários resultados, como: satisfação dos usuários, mais usuários utilizando a plataforma, melhora na reputação do aplicativo perante o público-alvo, engajamento dos usuários em forma de tempo de uso, visualização de anúncios ou compras dentro do aplicativo etc.

Para realizar tal feito, um escopo de pesquisa precisa ser bem definido. Por isso, a elaboração dos objetivos de pesquisa foi precedida por uma pesquisa exploratória, que serviu como base elucidativa para as questões de trabalho.

Na seção de trabalhos relacionados, explicada no Capítulo 2, as propostas de estavam focadas em agrupar os comentários baseados em requisitos funcionais da aplicação ou em palavras-chave pré-definidas, que tinham a ver com a temática do aplicativo.

A maior parte dessas abordagens, porém, não utilizaram como foco padrões de usabilidade, que podem ser desacopladas do contexto em que se inseriram, conforme proposto por Nielsen.

Motivado pelos objetivos descritos no capítulo introdutório, este trabalho sugere, portanto, uma forma de análise automatizada que possa ser aplicada em comentários de aplicativos de diferentes contextos, diminuindo a dependência da identificação de problemas com a temática do *software* estudado.

## 1.3 Objetivo geral

O objetivo geral de um trabalho é de certa forma um insumo para a compreensão detalhada do tema específico. Para (MARCONI; LAKATOS, 2003), o objetivo relaciona-se com o conteúdo intrinseco dos fenomenos, eventos e das idéias estudadas. Portanto, vincula-se diretamente a própria significação da abordagem proposta pelo projeto.

O objetivo geral do presente estudo é identificar problemas de usabilidade de aplicativos móveis a partir da análise de comentários dos usuários na loja de aplicativos *Google Play*.

## 1.4 Objetivos específicos

- **OE1** - Compreender quais características tornam um produto de *software* usável.
- **OE2** - Compreender quais são as técnicas e métodos utilizados para classificar a qualidade de um produto de software em relação à usabilidade.
- **OE3** - Identificar quais são métodos, técnicas e ferramentas que podem ser utilizadas para análise automatizada de textos e clusterização dos dados.
- **OE4** - Interpretar os problemas relatados e catalogá-los segundo os métodos utilizados para classificação de características de usabilidade.

## 1.5 Organização do trabalho

Este capítulo apresenta os problemas e objetivos que motivaram a existência deste trabalho, além da organização geral do documento.

O capítulo 2 trata do Referencial Teórico, no qual são apresentados os conceitos que sustentam a proposta de solução e metodologia do trabalho, trazendo as definições de qualidade de *software*, características de usabilidade, Heurísticas de Nielsen, contextualização de usabilidade e experiência do usuário no contexto *mobile*, além da definição de planejamento orientado a dados e métodos de análise textual, para as *reviews* dos usuários, e de trabalhos relacionados.

No capítulo 3, são tratados os aspectos metodológicos de desenvolvimento deste trabalho e, consequentemente, como se deram os procedimentos de pesquisa, coleta de dados para a análise e plano de desenvolvimento da proposta de trabalho, além dos cronogramas de entregas das Fases 1 e 2.

No capítulo 4, é apresentada a proposta de trabalho, na qual são explicadas as fases e abordagens propostas para a solução do problema, assim como o suporte tecnológico e o

detalhamento das etapas para a realização desta pesquisa. Também são expostos resultados preliminares, proveniente de análises feitas durante as fases de inserção e exploração do contexto e definição do escopo.

Por fim, no capitulo 5, discutimos as considerações finais do trabalho, onde apresentamos um breve resumo da problemática e da solução proposta.

# 2 Referencial teórico

Este capítulo abordará definições e conceitos utilizados para orientar a proposta feita, tratando desde considerações sobre o que é qualidade em *software*, quais as definições vigentes sobre usabilidade, dimensão de qualidade escolhida para guiar o trabalho, além de técnicas de análise e coleta de dados de usabilidade e de opiniões de usuários em lojas de aplicativos, que podem ser utilizadas em processos de desenvolvimento orientados a dados.

## 2.1 Qualidade em Software

*Systems and Software Quality Requirements and Evaluation* (SQuaRE) é um modelo de qualidade que define quais características devem ser avaliadas em um produto de *software*. Essa norma, estabelecida na ISO/IEC 25010, define, na Divisão de Modelo de Qualidade (ISO/IEC, 2011) que qualidade é "a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas". Então, nesse contexto, a qualidade de *software* mede quão bem um *software* é projetado e quão bem o *software* entra em conformidade com o seu *design* inicial (PRESSMAN, 2005).

Essas definições elaboram, em alto nível, qual deve ser a orientação principal ao medir a qualidade em uso de um produto de *software*. Porém, é necessário sair de níveis de abstração maiores, tornando mais palpável quais características e dimensões de qualidade devem ser observadas e quais critérios devem ser levados em conta durante o processo de garantia da qualidade.

A norma SQuaRE é composta por cinco principais características, podendo ser detalhada em mais subcaracterísticas. As principais características são:

- **Adequação funcional**: grau em que um produto ou sistema fornece funções que atendem às necessidades declaradas e implícitas quando usado sob condições especificadas. Essa característica está preocupada apenas em saber se as funções atendem às necessidades declaradas e implícitas, não à especificação funcional;
- **Eficiência de desempenho**: desempenho em relação à quantidade de recursos usados nas condições estabelecidas;
- **Compatibilidade**: grau em que um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes e/ou executar suas funções necessárias, enquanto compartilha o mesmo ambiente de *hardware* ou *software*;

- **Confiabilidade**: grau em que um sistema, produto ou componente executa funções especificadas sob condições especificadas por um período de tempo especificado;
- **Usabilidade**: grau em que um produto ou sistema pode ser usado por usuários especificados para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado.

Assim como todas as outras quatro características definidas pela SQuaRE (ISO/IEC, 2011) citadas anteriormente, a usabilidade também possui subcaracterísticas que são definidas em diversas dimensões que abrangem como as especificações de eficácia, eficiência e satisfação do usuário devem ser avaliadas.

- **Reconhecimento de adequabilidade**: grau em que os usuários podem reconhecer se um produto ou sistema é apropriado para suas necessidades. Nesse caso, essa capacidade está relacionada com o reconhecimento de funções do sistema a partir das impressões iniciais do produto ou sistema e/ou qualquer documentação associada, seja através de informações fornecidas pelo sistema, documentação, tutoriais etc.
- **Aprendizagem**: grau em que um produto ou sistema pode ser usado por usuários especificados para atingir objetivos específicos de aprender a usar o produto ou sistema com eficácia, eficiência, isenção de risco e satisfação em um contexto de uso especificado.
- **Operacionalidade**: grau em que um produto ou sistema tem atributos que o tornam fácil de operar e controlar.
- **Proteção contra erros do usuário**: grau em que um sistema protege os usuários contra cometer erros.
- **Estética da interface do usuário**: grau em que uma interface de usuário permite uma interação agradável e satisfatória para o usuário.
- **Acessibilidade**: grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades para atingir uma meta especificada em um contexto de uso especificado.

Embora essas subcaracterísticas avaliem múltiplas dimensões de usabilidade, em certo nível, ainda podem ser consideradas abstratas e podem ser refinadas ainda mais com o auxílio de outros métodos de avaliação, categorização e contextualização. Dessa forma, a próxima seção tratará de definições de Heurísticas de usabilidade de Nielsen, que embasam e detalham mais algumas das características citadas nesta seção.

## 2.2 Heurísticas de usabilidade de Nielsen

O termo heurística possui origem grega e significa "que serve para descobrir ou encontrar". Esse termo também é definido como "uma estratégia que ignora parte da informação, com o objetivo de tomar decisões de forma mais rápida, econômica e/ou precisa do que métodos mais complexos" (GIGERENZER; GAISSMAIER, 2010).

Considerando a definição dada de usabilidade, um atributo de qualidade que avalia quão fácil uma interface é de ser utilizada (DOURADO; CANEDO, 2018), e aplicando a definição de Gigerenzer e Gaissmaier de heurística, entende-se que o usuário utiliza esses mecanismos também no produto de *software*, tomando as decisões que aparentam ser mais fáceis, intuitivas e cognitivamente econômicas.

Jakob Nielsen definiu dez heurísticas de usabilidade (NIELSEN; MOLICH, 1990) que servem como princípios gerais de *design* de interação entre o usuário e a interface. Elas são chamadas dessa forma pois são "regras de ouro" e não diretrizes específicas de usabilidade. São elas:

- **Visibilidade do *status* do sistema:** A interface deve sempre manter os usuários informados sobre o que está acontecendo, através de um *feedback* apropriado dentro de um período de tempo razoável.
- **Combinação entre o sistema e o mundo real:** A interface deve falar a língua do usuário. Devem ser utilizadas palavras, símbolos e conceitos que sejam familiares aos usuários, e as informações devem ser apresentadas de forma natural e lógica.
- **Controle e liberdade do usuário:** Os usuários precisam de uma "saída de emergência" claramente indicada para deixar a ação indesejada sem ter que passar por um processo extenso.
- **Consistência e padrões:** Os usuários não devem se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. Devem ser seguidas as convenções da plataforma e do setor.
- **Prevenção de erros:** Devem ser eliminadas as condições sujeitas a erros, e deve ser apresentada aos usuário uma condição de confirmação antes de eles se comprometerem com a ação. É importante notar a diferença entre enganos e deslizes: deslizes são erros inconscientes causados por desatenção. Enganos são erros conscientes baseados em uma incompatibilidade entre o modelo mental do usuário e a interface.
- **Reconhecimento em vez de recordação:** O usuário não deve ter que se lembrar de informações enquanto navega entre uma parte da interface e outra. As informações devem ser visíveis ou fáceis de recuperar quando necessárias.

- **Flexibilidade e eficiência de uso:** Permita que os usuários personalizem ações frequentes.

- **Design estético e minimalista:** As interfaces não devem conter informações irrelevantes ou raramente necessárias.

- **Ajude os usuários a reconhecer, diagnosticar e se recuperar de erros:** As mensagens de erro devem ser expressas em linguagem simples (sem códigos de erro), indicando precisamente o problema e sugerindo uma solução de forma construtiva.

- **Ajuda e documentação:** É melhor se o sistema não precisar de nenhuma explicação adicional. No entanto, pode ser necessário fornecer documentação para ajudar os usuários a entender como concluir suas tarefas.

## 2.3 Medição e planejamento da *User eXperience* orientadas a dados

A experiência do usuário, no contexto de Engenharia de Software, é uma relação contínua com eventos que acontecem entre o usuário e o sistema em que o foco não é a experiência em si, mas a relação estabelecida entre o usuário e o sistema e as resultantes que podem ser obtidas dessa interação (HASSENZAHL, 2008).

Portanto, a experiência do usuário aparece como um complemento à usabilidade, que auxilia essa relação sendo uma visão mais ampla, focado na relação individual do usuário, como sentimentos, percepções e intenções resultantes desta interação (NIELSEN; MOLICH, 1990).

A experiência do usuário está diretamente ligada à capacidade do usuário reconhecer e executar as tarefas de um sistema com eficiência e satisfação, em um contexto de uso específico (MacDonald; ATWOOD, 2014). Para uma análise dos parâmetros de *User exPerience*, deve-se observar critérios que estão atrelados à definição de usabilidade e suas variantes. As métricas de usabilidade mais amplamente utilizadas são: tempo de conclusão da tarefa, número de erros, precisão e taxa de conclusão da tarefa e satisfação. Essas métricas também mapeiam diretamente atributos de experiência de usuário (MacDonald; ATWOOD, 2014).

Apesar de métricas bem definidas para serem analisadas durante o uso do sistema, a User eXperience (UX) também envolve as etapas que antecedem e sucedem o contato direto do usuário com o sistema (OHASHI et al., 2018). E dada a diversidade de perspectivas, é importante direcionar também a análise para essa parte do processo (PETTERSSON et al., 2018),

No cenário atual de desenvolvimento de *software*, usuários podem enviar facilmente *feedback* sobre os produtos em lojas de aplicativos, mídias sociais ou grupos de usuários. Além disso, os fornecedores de *software* estão coletando grandes quantidades de *feedback* implícitos na forma de dados de uso, registros de erros e dados de sensores (MAALEJ et al., 2016).

As equipes de gerência e/ou desenvolvimento podem ser capazes de, a partir dos requisitos obtidos pelas massas de usuários, decidir o que desenvolver e quando lançar uma nova funcionalidade, assim, utilizando sistematicamente dados explícitos e implícitos do usuário de uma forma agregada para apoiar as decisões de projeto (MAALEJ et al., 2016).

Essas tendências sugerem uma mudança na área de desenvolvimento de *software*, que tende a ir em direção à identificação, à priorização e gerenciamento de Requisitos de *Software* de forma mais centrada no usuário e que seja orientada por dados de forma que os engenheiros conheçam melhor os usuários finais, embora existam limitações nessa interação.

Embora essa prática possa trazer bons resultados na melhoria da experiência em uso, a coleta de dados pode ser difícil de várias formas isso acontece. Seja porque os usuários não são capazes de identificar quais subsistemas são responsáveis por um comportamento específico do *software* seja porque o funil de entrada de dados das opiniões dos usuários, como comentários nas lojas dos aplicativos, exige muito recurso manual e a análise desses dados ocupa muito tempo da equipe responsável, tornando a tarefa imanejável (GÄRTNER; SCHNEIDER, 2012).

Alguns dos principais desafios de um time que possui foco contínuo no usuário são coletar informações suficientes para mostrar que o esforço despendido gera valor, manter a confiança nas ações tomadas e construir consenso através de opiniões e informações coletadas (FEHNERT; KOSAGOWSKY, 2008).

A medição empírica da usabilidade, que em um processo iterativo poderá servir como apoio para a priorização de tarefas com foco na entrega de valor para o usuário, pode ser medida de duas formas (BUIDU, 2017): (a) *Qualitativamente*: consiste em achados observacionais que identificam quais características do *design* são difíceis ou fáceis de usar; e (b) *Quantitativamente*: consiste em oferecer uma avaliação indireta da usabilidade de um *design*. Medindo, por exemplo, a taxa de sucesso na execução de uma tarefa ou o tempo de execução destas.

Embora os dados quantitativos ajudem a referenciar quais *features* possuem pontos de atenção, eles não explicam, por si só, quais são os problemas que os usuários encontraram de fato, sendo esse então o papel dos dados qualitativos coletados (BUIDU, 2017).

Em outras palavras, os dados quantitavos nem sempre podem ser encarados como

um resultado indicativo de um problema, enquanto o qualitativo entrega mais valor quando complementado por análises quantitativas (FEHNERT; KOSAGOWSKY, 2008).

## 2.4 Análise textual de *reviews* dos usuários

Os dados qualitativos ou quantitativos podem ser coletados de inúmeras fontes e ocasiões, sejam elas de opiniões de usuários durante sessões de testes de usabilidade, *trackers* de análise de fluxo do usuário dentro do sistema, avaliações através de caixas de comentários em redes sociais, canais de ouvidoria ou de avaliações em texto nas lojas de aplicativos móveis disponibilizadas pelas fabricantes etc.

As avaliações dos aplicativos fornecidas pelos usuários nas lojas de aplicativos móveis possuem dados valiosos que podem ser convertidos em dados quantitativas e qualitativos oferecendo novas formas de analisar a satisfação destes em diversas perspectivas, sendo uma delas sob a ótica da usabilidade.

Como citado anteriormente, alguns funis de entrega desses dados possuem problemas de alto volume de entrada que demanda muitas vezes recursos escassos, como mão de obra e tempo.

Existem inúmeras abordagens para mineração de dados que podem ser utilizadas no contexto de extração de informações das *reviews* dos usuários (TAVAKOLI et al., 2018). Cabe citar: aprendizado de máquina supervisionado (*supervised machine learning*); processamento de linguagem natural (*neural language processing*); e extração de características (*feature extraction*).

Cada uma dessas abordagens utiliza também suas próprias técnicas para classificação: manual, automatizada ou ambas. Também podem utilizar-se de adição de *features* para os dados, como análise de sentimentos e técnicas estatísticas que identificam e adicionam mais importância às palavras frequentes, como a técnica da frequência do termo–inverso da frequência nos documentos (*term frequency–inverse document frequency* ou TF-IDF, em inglês).

O tópicos de análise das ferramentas ou abordagens propostas também pode variar. Algumas, por exemplo, focam apenas em comentários negativos que podem indicar desde problemas funcionais quanto não funcionais (muita utilização de recursos, problemas de conexão, tempo de resposta, combatibilidade etc). Outras, podem focar em aspectos de pedidos por novas funcionalidades, melhoria estética ou pedidos gerais dos usuários.

Há também ferramentas que focam na experiência do usuário e usabilidade, investigando desde comentários que questionam como utilizar uma *feature*, cenários de uso e informações sobre dificuldades na utilização.

## 2.5 Métricas para avaliação de um modelo de processamento de linguagem natural

A acurácia geral é a razão entre quantidade de previsões corretas e quantidade de previsões totais. O problema em analisar apenas essa métrica é que ela pode não revelar desbalanceamento entre as etiquetas. Ou seja, ela pode acertar muito para uma heurística, mas errar bastante para outra e, no fim, ter uma acurácia geral alta.

O *recall* tenta responder à seguinte questão: qual proporção de positivos reais foi identificada corretamente? O valor resultante é o número de resultados positivos verdadeiros dividido pelo número de todos os resultados positivos, incluindo aqueles não identificados corretamente. Ao não produzir nenhum falso positivo, pode-se chegar a um valor máximo de 1.0 (PEDREGOSA et al., 2011).

A *precision* tenta responder à seguinte questão: qual proporção de identificações positivas estava realmente correta? O valor resultante é o número de resultados positivos verdadeiros dividido pelo número de todas as amostras que deveriam ter sido identificadas como positivas. Ao não produzir nenhum falso negativo, pode-se chegar a um valor máximo de 1.0 (PEDREGOSA et al., 2011).

Porém, essas duas métricas são, de certa forma, conflitantes, uma vez que, ao melhorar a *precision*, o *recall* diminui. O contrário também é válido, e o ideal é analisar as métricas em conjunto e modificar o limite mínimo de acurácia aceitável (PEDREGOSA et al., 2011).

O *F-score* é a média harmônica entre a *precision* e o *recall*. O maior valor possível é 1.0, que indica que *precision* e *recall* são perfeitos. O menor valor possível, inversamente, é 0.0 (PEDREGOSA et al., 2011).

## 2.6 Trabalhos relacionados

A ferramenta MARK utiliza uma classificação semi-automatizada, onde um analista pode inserir palavras-chave de seu interesse e o algoritmo encontra e classifica reviews que contenham o conteúdo indicado (PHONG et al., 2015).

O artigo *User Feedback in the AppStore: An Empirical Study* (PAGANO; MAALEJ, 2013) conduz um estudo utilizando mais de um milhão de reviews de usuários da *AppStore*, que investiga padrões de com qual frequência e como os usuários dão feedback, além de definir alguns tópicos são os mais abordados pelos usuários, dentre eles: pedidos de *features*, *bug reports* ou experiência de usuário.

O trabalho *Extracting Usability and User Experience Information from Online User Reviews* (HEDEGAARD; SIMONSEN, 2013) analista mais de cinco mil sentenças de cerca

de 3 mil *reviews* de *softwares* e *video games* para investigar a distribuição entre diferentes dimensões de usabilidade e experiências relatadas pelos usuários. O estudo indica que existe uma oportunidade para entender mais sobre os usuários a partir de uma análise mais aprofundada do tema. Por fim, sugere como trabalhos futuros a análise utilizando diferentes contextos de *reviews*, não apenas os dois utilizados, além de entender a relação entre as sentenças pertencentes à alguma dimensão de experiência de usário e a análise de sentimento das *reviews* analisadas.

Outros trabalhos, porém, focam em algum aspecto mais específico de informações contidas nas *reviews*, como problemas emergentes (novos *bugs* ou refinar e/ou adicionar novas *features*). Esse é o caso do artigo *Online App Review Analysis for Identifying Emerging Issues* de (GAO et al., 2018), que apresenta o *framework IDentify Emerging App* (IDEA), que analisa os textos em *near-realtime*, ou seja: sempre que um comentário novo chega, ele passa por uma série de transformações e classificações automátizadas e é disponibilizado para o monitoramento da equipe responsável pela análise.

Uma das formas apresentadas para visualizar as análises é relacionar um número específico de problemas emergentes ao longo de cada versão nova do *software*. É interessante notar que também foi realizada uma pesquisa para coletar métricas e o impacto do seu uso em produção em vários produtos da empresa chinesa *Tencent*, que possuem centenas de milhões de usuários.

# 3 Metodologia

Neste capítulo, será apresentada a metodologia de desenvolvimento do presente trabalho e consequentemente como se deram os procedimentos de pesquisa, a coleta de dados para a análise e o plano metodológico.

## 3.1 Tipificação da pesquisa

Quanto a metodologia, classificou-se em relação à abordagem, à natureza, aos objetivos e aos procedimentos técnicos realizados.

- **Objetivos**: pretende-se observar e identificar problemas de usabilidade atrelados ao contexto *mobile* e suas dinâmicas subsequentes. Por isso, o objetivo da pesquisa é gerar conhecimentos dirigidos para aplicação de práticas à solução de problemas específicos (GIL; VERGARA, 2015). A pesquisa caracteriza-se como descritiva, pois envolve análise, técnicas de coletas e observação de dados e fenômenos (GIL; VERGARA, 2015).

- **Abordagem**: uma pesquisa de abordagem qualitativa é a opção que mostrou-se mais viável para a realização do estudo, pois os métodos qualitativos são apropriados quando o fenômeno em estudo é complexo, de natureza social e não tende à quantificação. Normalmente, são usados quando o entendimento do contexto social e cultural é um elemento importante para a pesquisa. Para aprender métodos qualitativos é preciso aprender a observar, registrar e analisar interações reais entre pessoas, e entre pessoas e sistemas (LIEBSCHER, 1998).

- **Natureza**: define-se como pesquisa aplicada, pois objetiva gerar conhecimentos para a aplicação prática dirigida à solução de problemas (LANDIM et al., 2006).

- **Procedimentos**: O meio utilizado para a realização da pesquisa foi Pesquisa bibliográfica , pois trata-se de uma pesquisa que se embasou em "uma investigação científica de obras já publicadas"(MARCONI; LAKATOS, 2003). A pesquisa bibliográfica é primordial para a construção da pesquisa científica, pois nos ajuda a reconhecer melhor o fenomeno de estudo.

- **Técnica de coleta de dados**: Como técnica de coleta de dados, define-se como análise de conteúdo, pois o objetivo é coletar o máximo de informações para não faltá-las no momento de elaborar o trabalho. Qualquer tipo de levantamento de dados deve ter o máximo de cuidado durante a coleta e análise de informaçoes.(SOARES, 2007)

## 3.2 Plano metodológico do trabalho

A elaboração e execução de uma metodologia de trabalho são importantes para que a análise e os resultados sejam satisfatórios e apresentados de maneira correta (SILVA; MENEZES, 2001). Também, é importante seguir o plano elaborado para que as etapas não sejam realizadas de forma equivocada e não haja impacto negativo no resultado do trabalho.

O processo de elaboração, portanto, foi organizado de acordo com o diagrama da Figura 1.

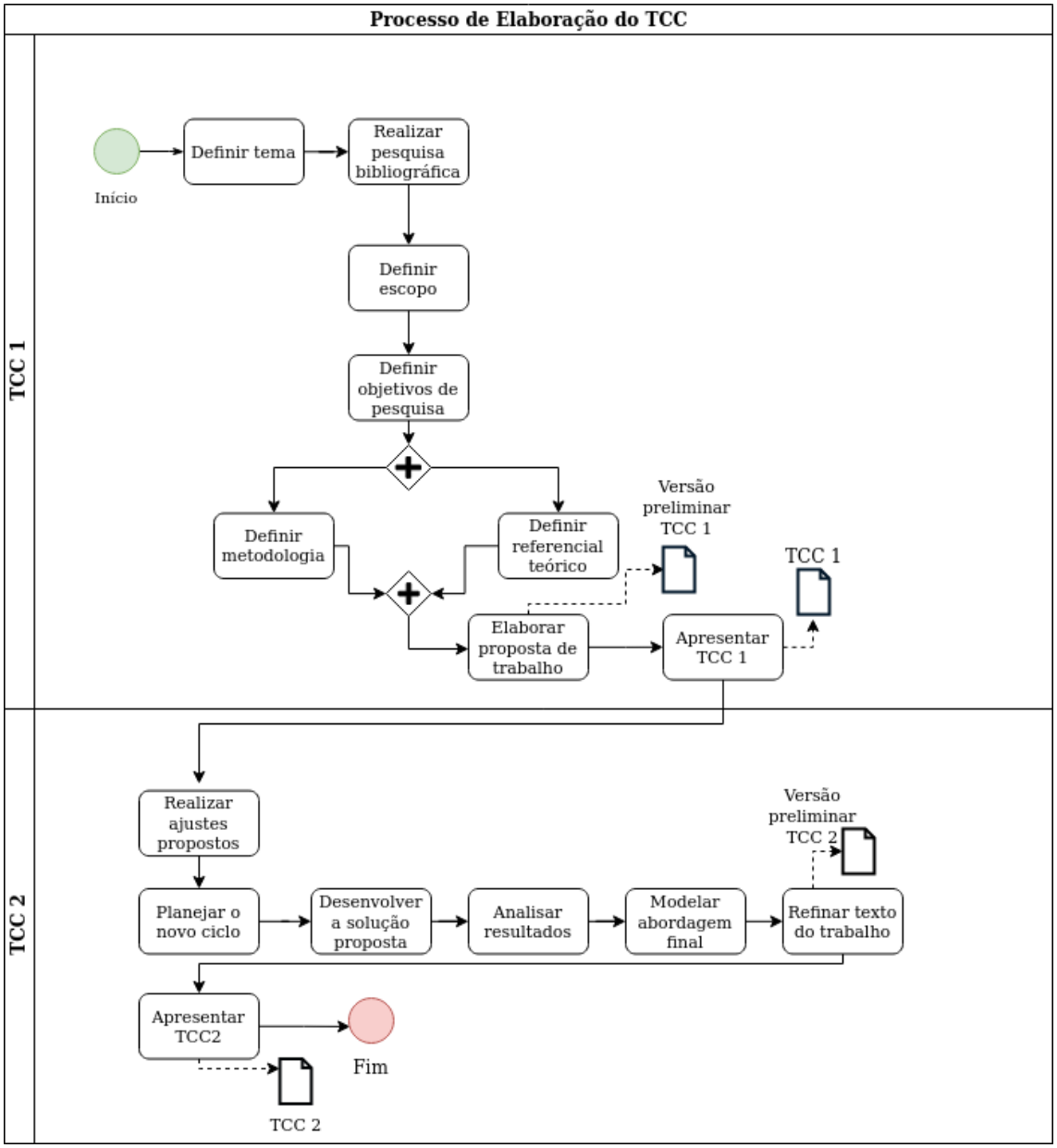


Figura 1 – Processo metodológico de elaboração do TCC.

Sendo assim, o diagrama do processo metodológico foi elaborado tendo como base duas entregas principais, que foram divididas em duas raias: Trabalho de conclusão de

curso (TCC) 1 e TCC 2. A primeira raia, referente ao TCC1, tem como foco a fundamentação teórica e estruturação da pesquisa. A segunda raia, refere-se ao TCC2, que tem como foco principal a execução da solução proposta , a análise e discussão das informações obtidas.

As atividades relacionadas ao TCC1 são listadas nas subseções a seguir.

### 3.2.1 Trabalho de Conclusão de Curso 1

1. Na primeira etapa desta pesquisa (*Definição do tema*), realiza-se a definição do tema do trabalho. Baseada nos interesses de pesquisa dos estudantes e da professora orientadora, da análise da literatura cientifica e da identificação do problema a ser abordado.

2. A *pesquisa bibliografica* é realizada com o objetivo de adquirir embasamento teórico para a elaboração do trabalho. Nesta etapa, por meio de uma pesquisa exploratória, pesquisas similares são identificadas e categorizadas a fim de enriquecer o conhecimento sobre o tema.

3. Na etapa de *definição de escopo*, os assuntos que serão abordados no trabalho são restringidos de acordo com a disponibilidade e relevância do tema para a pesquisa.

4. Na *definição da metodologia*, acontece a estruturação metodológica do trabalho, bem como a tipificação da pesquisa, definição de cronograma e organização do trabalho.

5. Nesta etapa, é realizado o embasamento teórico do trabalho. O *referencial teórico* serve como fundamentação para o trabalho, além de garantir qualidade científica para a pesquisa.

6. Na etapa de *proposta de trabalho*, é definida inicialmente qual a abordagem de resolução do problema identificado pela pesquisa será utilizada como produto do trabalho.

7. Esta etapa consiste no *refinamento do trabalho* e na realização das alterações propostas pela professora orientadora antes que o trabalho seja entregue para a banca avaliadora.

8. Por fim, para a conclusão das atividades listadas na primeira raia, o *trabalho é apresentado para a banca avaliadora* e o *feedback* para os ajustes é obtido.

### 3.2.2 Trabalho de Conclusão de Curso 2

O inicio das atividades a serem realizadas que foram listadas na segunda raia, referentes ao TCC2, serão iniciadas imediatamente após o fim da apresentação da primeira fase do trabalho.

Assim, listamos a seguir as atividades subsequentes:

1. Os *ajustes recomentados* pela banca avaliadora serão realizados para a continuidade do trabalho.
2. O *planejamento de um novo ciclo* de desenvolvimento será elaborado afim de construir continuamente o produto de *software* que será a entrega final.
3. Quanto ao *desenvolvimento da solução proposta*, baseando-se nos pilares da metodologia ágil, será realizada a etapa de desenvolvimento do produto.
4. Na etapa de *análise dos resultados*, os dados coletados e a execução do ciclo de vida do *software* servirão de insumos para a análise crítica dos resultados obtidos.
5. Por fim, a *apresentação final do trabalho* marcará o fim da apresentação dos resultados e de toda a pesquisa.

## 3.3 Cronogramas

Para garantir que o processo seja seguido e o trabalho entregue conforme proposto, foi elaborado um cronograma de atividades, exposto nas Tabelas 1 e 2.

| **Atividades** | **Fevereiro** | **Março** | **Abril** | **Maio** |
|---|---|---|---|---|
| Definir tema | x | | | |
| Realizar pesquisa bibliográfica | x | | | |
| Definir escopo | | x | | |
| Definir metodologia | | x | | |
| Definir referencial teórico | | x | | |
| Definir objetivos de pesquisa | | | x | |
| Elaborar proposta de trabalho | | | x | |
| Realizar revisão e refinamento | | | | x |
| Apresentar TCC 1 | | | | x |

Tabela 1 – Cronograma de atividads para o TCC 1.

| **Atividades** | **Junho** | **Julho** | **Agosto** | **Setembro** | **Outubro** | **Novembro** |
|---|---|---|---|---|---|---|
| Realizar ajustes propostos | x | | | | | |
| Planejar o novo ciclo | | x | | | | |
| Desenvolver a solução proposta | | x | x | | | |
| Analisar resultados | | | | x | | |
| Modelar abordagem final | | | | x | x | |
| Refinar texto do trabalho | | | | | x | |
| Apresentar TCC2 | | | | | | x |

Tabela 2 – Cronograma de atividades para o TCC 2.

## 3.4 Gestão de desenvolvimento do TCC

Além do cronograma e processo metodológico da elaboração do documento, é necessário seguir ciclos para o desenvolvimento do trabalho. Existem as atividades, com as entregas definidas de acordo com o cronograma. Para cada atividade, são definidas tarefas a serem realizadas para concluir a atividade.

As tarefas a serem feitas para finalizar uma atividade descrita na organizados utilizando a metodologia do Kanban, onde as tarefas a serem realizadas são organizadas entre as fases: para fazer, fazendo, em revisão e feito.

As atividades são definidas junto à orientadora. Quando uma atividade em revisão é aprovada pela orientadora, a atividade é dada como feita.

Os ciclos maiores, para a finalização das atividades são definidos de acordo com as datas estabelecidas no cronograma. As tarefas necessárias para finalizá-las sendo feitas e apresentadas em ciclos quinzenais com a orientadora.

# 4 Proposta de trabalho

Este capítulo tratará da proposta de solução do problema observado, bem como das tecnologias que serão utilizadas como suporte para o processo de desenvolvimento apresentado na seção anterior.

## 4.1 Processo geral em alto nível

As atividades de extração, persistência, tratamento, padronização e análise de dados dos comentários para avaliação da qualidade do aplicativo a partir dos resultados é inspirada no processo de *Extract, Transform, Load* (ETL). Essas etapas foram divididas nas fases a seguir, realizadas em ordem:

1. **Extrair**: obtêm-se os dados de uma fonte externa;
2. **Transformar**: os dados passam por um processo de transformação (ex.: tradução, junção com outros dados, classificação, etiquetados etc.) para atender o contexto especificado. Nesta proposta, também foi adicionada a classificação dos e análise dos dados;
3. **Carregar**: os dados transformados e analisados podem ser carregados em um banco de dados, planilhas ou outros *softwares* de armazenamento de dados.

O processo, em uma visão de alto nível, segue as etapas apresentadas na Figura 2.

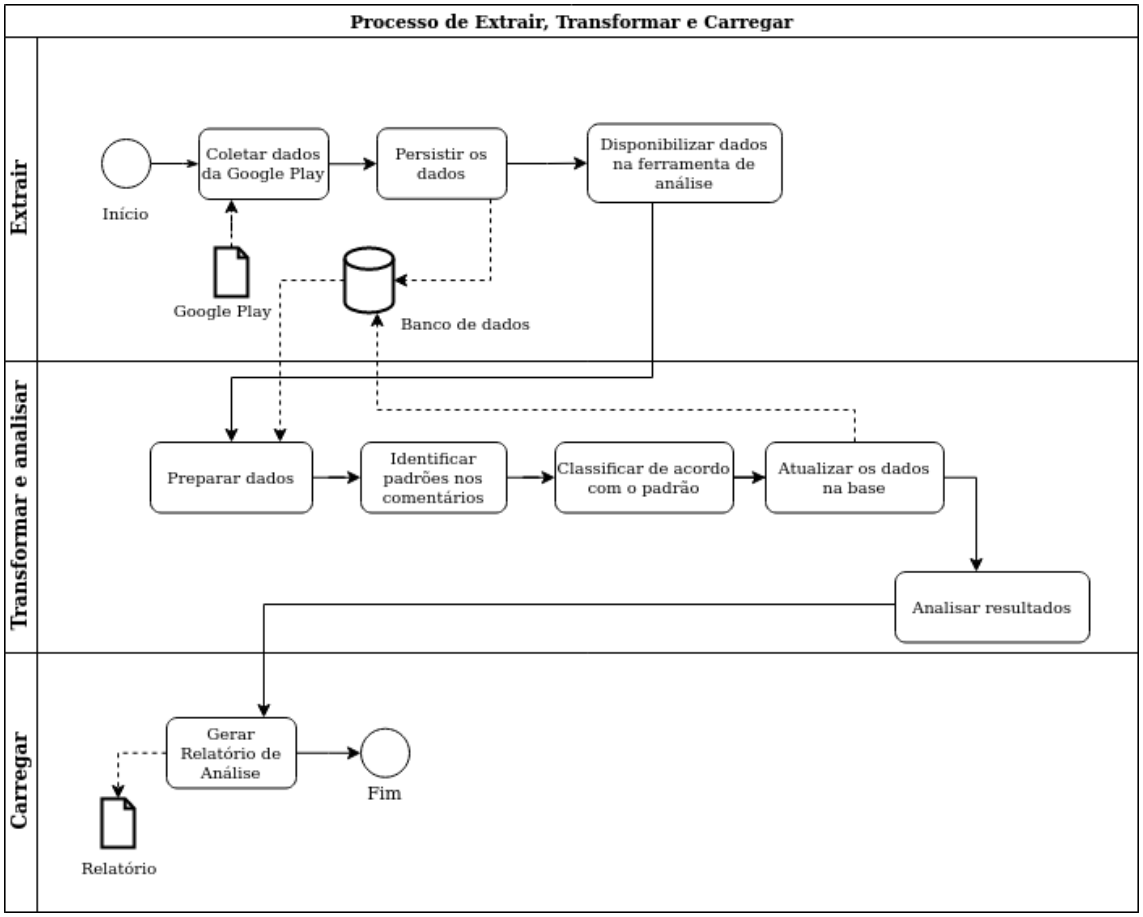


Figura 2 – Modelagem do processo proposto.

Esse processo é comumente utilizado para transferir dados de um sistema para outro, aplicando tratamentos necessários para o contexto e utilizando-os para análise posterior. Nesse trabalho, por exemplo, transferindo os comentários de uma lista de avaliações para uma análise mais aprofundada sobre as informações que indicam problemas de qualidade em uso.

Cada uma das raias (etapas) possui tarefas específicas, que serão detalhadas nas subseções a seguir.

### 4.1.1 Etapa 1 - Extrair

Esta primeira etapa tem como objetivo fazer a extração dos dados da loja. Além dos textos das *reviews* dos usuários, são coletados dados como: versão do aplicativo, data do comentário, quantidade de estrelas dada pelo usuário (entre 1 e 5. Quanto maior, melhor). Suas atividades são:

- **Coletar dados da *Google Play*:** Nessa etapa, o *script* elaborado em Node.JS irá coletar os dados de um ou mais aplicativos pré-definidos pela equipe.

- **Persistir os dados no banco de dados:** Após a coleta de dados, o *script* persiste-os no banco de dados PostgreSQL.

- **Disponibilizar dados na ferramenta de análise:** Assim que um dado novo é salvo no banco de dados, o Metabase, ferramenta de análise adotada, sincroniza e disponibiliza os dados para visualização, filtragem e análise.

### 4.1.2 Etapa 2 - Transformar

Algumas tarefas de transformação tratam apenas de limpeza dos dados. Porém, em algumas tarefas de análise, também pode ser feita a transformação desses dados. A identificação e categorização dos comentários em possíveis padrões de usabilidade, por exemplo, ocorrem durante essa etapa. Suas atividades são:

- **Preparar dados**: Os dados disponíveis no banco de dados passam por limpeza de caracteres indesejados, correção ortográfica, etiquetagem, entre outros.

- **Identificar padrões nos comentários**: Identificar como um comentário se relaciona com um padrão de usabilidade, dentro de um grupo de padrões pré-definidos. Essa tarefa é um exemplo de análise que resulta em uma tarefa de transformação.

- **Classificar de acordo com o padrão**: Ao relacionar um ou mais padrões com o comentário, o comentário é classificado de acordo. Nessa fase, também ocorre o estudo e desenvolvimento de algoritmos para automatizar essa tarefa.

- **Atualizar os dados na baase**: Os dados classificados são atualizados na base, para posterior análise.

- **Análise dos resultados**: Nessa tarefa, são realizadas as análises quantitativas e qualitativas. É possível identificar métricas importantes sobre quais são os problemas de usabilidade mais recorrentes nos comentários, como quais são os principais grupos de problemas que os usuários avaliam no aplicativo, qual o impacto desse padrão na nota do aplicativo ou qual o histórico de aumento desse padrão de acordo com as versões dos aplicativos.

### 4.1.3 Etapa 3 - Carregar

Esta etapa consiste na geração de relatórios com as métricas identificadas pela fase anterior. Os relatórios podem ser gerados pela plataforma Metabase, quanto utilizando outras plataformas. Essa fase também poderá ser automatizada.

## 4.2 Suporte tecnológico e fonte de dados

Esta seção detalha as linguagens, bibliotecas e fontes de dados principais para a coleta, tratamento e análise dos dados.

- **Python**: será utilizada para o tratamento dos dados, correção ortográfica dos textos, remoção de caracteres indesejados etc.

- **Node.JS**: será utilizada para coletar os dados da *Google Play* juntamente com a biblioteca *google-play-scraper* e salvá-los no banco de dados.

- **Loja de aplicativos *Google Play***: será a fonte dos dados que serão estudados. Ela fornece os comentários dos usuários dos aplicativos, juntamente com metadados como data do comentário, versão do app, nota dada pelo autor etc.

- **Biblioteca *google-play-scraper***: é uma biblioteca que auxilia na conexão com a *Google Play* através do Node.JS, permitindo coletar os comentários dos usuários. Disponibilizada sob a Licença MIT, uma licença permissiva utilizada em softwares livres desenvolvida pelo Instituto de Tecnologia de Massachusetts (MIT).

- **PostgreSQL**: é um sistema gerenciador de banco de dados relacional, disponibilizado como projeto de código aberto sob uma licença própria. Será utilizado para armazenar os dados coletados da loja.

- **Metabase**: é uma ferramenta de análise de dados, que permite a construção de gráficos e *dashboards* através de consultas SQL. Essa ferramenta permite que sejam feitas exportações desses dados conforme a necessidade de maior detalhamento em alguma análise. Também é possível entender o crescimento da base, filtrar os resultados por versão e utilizar apenas esse segmento dos dados, caso seja necessário. Essa ferramenta é disponibilizada em versão gratuita, sob a licença *GNU Affero General Public License* (AGPL).

## 4.3 Resultados preliminares

Durante a fase inicial deste trabalho, foi feita uma prova de conceito, com o objetivo de entender melhor o contexto e auxiliar na definição de escopo.

Essa prova de conceito foi realizada conforme as fases do processo proposto, utilizando como objeto de exemplo um aplicativo de utilização de serviços públicos do governo brasileiro para exemplo.

### 4.3.1 Etapa 1 - Extrair

Nesta fase, foram extraídos 1032 comentários, utilizando a ferramenta de *scraping*, descrita anteriormente. As datas dos comentários coletados correspondem a um intervalo de tempo de dois meses.

Um exemplo de um comentário extraído e estruturado no formato *JavaScript Object Notation* (JSON) é apresentado na Figura 3. As informações contidas nesse exemplo foram detalhadas anteriormente, durante a descrição das etapas na seção anterior.

```
{
  id: 'gp:...',
  userName: 'Usuário descontente',
  score: 3,
  title: 'Muito confuso',
  text: 'não entendi como funciona o reconhecimento facial',
  replyDate: '2020-11-10T18:31:42.174Z',
  replyText: 'Olá! Verifique o nosso FAQ em (...)',
  version: '1.0.2',
  thumbsUp: 29,
}
```


Figura 3 – Exemplo de dado coletado da loja *Google Play*.

*Fonte: Elaborada pelos autores*

Os dados foram persistidos no banco de dados PostgreSQL, prontos para a fase seguinte. Além disso, o banco de dados foi sincronizado com uma instância do Metabase.

### 4.3.2 Etapa 2 - Transformar

Essa etapa apresentou a maior parte de resultados, novas ideias e também dificuldades, para serem resolvidas na implementação real do projeto.

- **Preparar dados**

  Nessa fase, foi feito um pré-processamento nos textos dos comentários, com o propósito de transferi-los de uma linguagem humana para um formato mais "legível" pelos algoritmos de análise de dados.

- **Identificar padrões nos comentários**

  Durante essa fase, pôde-se perceber que existem inúmeras formas para classificar, organizar e padronizar os comentários. Algumas dessas formas já haviam sido identificadas na leitura dos trabalhos relacionados, como aglomerar por problemas (por exemplo: aplicativo drenando muita bateria, lentidão, formulário confuso etc) ou

| Comentário | Classificação |
|---|---|
| Perdi a senha, nao tenho cnh nem conta em banco. Ficou impossivel recuperar meu acesso | Usabilidade |
| Não consigo validar o identidade facial, dá erro ao final | Usabilidade |
| Muito importante | Não-usabilidade |
| Não tá digitando nada. Não acesso ele | Usabilidade |
| Sim facilita muito em vários serviços e o mais importante, sem sair de casa | Usabilidade |
| Eu não tenho CNH é mesmo assim quando coloco meu CPF pede CNH não sei o que faço | Usabilidade |
| Bom | Não-usabilidade |
| App horroroso | Não-usabilidade |

Tabela 3 – Exemplos da classificação de comentários.

*Fonte: Elaborada pelos autores*

por funcionalidades (por exemplo: login, carrinho de compras etc). Como o objetivo da prova de conceito era validar o processo como um todo e não definir qual era a estratégia de classificação de fato, foi utilizada uma classificação simples baseada nas heurísticas de Nielsen.

- **Classificar de acordo com o padrão**

  O processo utilizado foi uma rotulação manual dos comentários que indicavam, de forma binária, se cada um se encaixava ou não dentro do contexto de usabilidade segundo alguma heurística de Nielsen.

  Por exemplo, um comentário que diz algo como “quando vou fazer login, sempre dá erro e nunca me diz o motivo”, pode indicar alguma falha na heurística de visibilidade do status do sistema ou ajuda de reconhecimento, diagnóstico e recuperação de erros. Portanto, o comentário seria assinalado como “usabilidade”.

  De outra forma, comentários sem contexto suficiente, dizendo “app horrível”, “app muito bom” ou “app não funciona” foram assinalados como “não usabilidade”.

  Em seguida, foi desenvolvido um algoritmo classificador baseado em ocorrência, utilizando parte desses dados para treinamento e outra para teste de acurácia.

  Na última fase, os comentários classificados como positivos para violação nas heurísticas de usabilidade são agrupados utilizando a técnica de clusterização hierárquica aglomerativa e cada grupo (*cluster*) criado é nomeado conforme o contexto representado. Essa técnica permite que as palavras sejam agrupadas conforme a

sua proximidade. Por exemplo, comentários contendo as palavras “reconhecimento” e/ou “facial” terão uma maior proximidade do que comentários contendo “senha” e/ou “recuperação”.

| **Classificação** | **Utilização** | **Quantidade** |
|---|---|---|
| Usabilidade | Treino | 497 |
| Não Usabilidade | Treino | 153 |
| Usabilidade | Teste | 214 |
| Não Usabilidade | Teste | 66 |

Tabela 4 – Enumeração da utilização e quantidade dos comentários classificados.

*Fonte: Elaborada pelos autores*

- **Atualizar os dados na base**

  Os dados classificados foram atualizados na base, para posterior análise.

- **Análise dos resultados**

  Os resultados podem ser divididos entre análise quantitativa e análise qualitativa.

  A análise quantitativa traz informações sobre a quantidade dos dados, relação entre quantidade de comentários e notas dadas ao aplicativo, contagem da classificação se pertencem à classe de “usabilidade” etc.

  A análise qualitativa trata dos resultados obtidos através do algoritmo classificador, incluindo os resultados de acurácia dele próprio, além dos dados obtidos através do agrupamento dos comentários de acordo com os seus semelhantes.

  Quantitativamente, pôde-se observar que a maioria dos comentários é feito com a classificação de uma estrela abrangendo mais da metade do total de comentários extraídos da loja.

| **Estrelas** | **Porcentagem** |
|---|---|
| 1 estrela | 52% |
| 2 estrelas | 7% |
| 3 estrelas | 14% |
| 4 estrelas | 7% |
| 5 estrelas | 20% |

Tabela 5 – Porcentagem de comentários por nota.

*Fonte: Elaborada pelos autores*

Apesar de "1 estrela" ser o menor valor possível para ser atribuído a um comentário, isso não significa que todos os comentários dessa categoria são necessariamente negativos, pois alguns usuários inadvertidamente podem errar ao enviar a nota. De qualquer maneira, o número de comentários negativos ainda é um número expressivo e chegam a ser mais da metade do número total de comentários.

Sobre o algoritmo de classificação em ocorrência, foi possível identificar quando um comentário feito é relacionado à usabilidade com aproximadamente 97% de acurácia, e 35% caso não seja. O resultado baixo relacionado à "não usabilidade" ocorre pela falta de comentários suficientes para manter a base de treinamento (comentários classificados) balanceada.

Para melhor visualização, comentários selecionados dentro do assunto "usabilidade" foram agrupados em um dendograma e resultaram em quatro *clusters*, podendo ser visualizados na Figura 4.

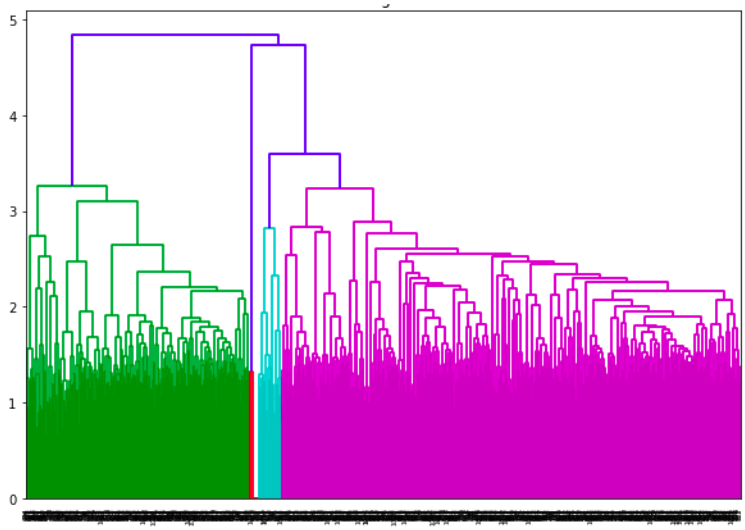


Figura 4 – *Clusters* de comentários. Quanto mais largo, mais palavras relacionadas.

*Fonte: Elaborada pelos autores*

Nota-se que os *clusters* possuem tamanho diferenciado, sendo o Verde e o Rosa os dois maiores. A quantidade de comentários de cada *cluster* é detalhada na Tabela 6.

Cada *cluster* foi nomeado de acordo com o seu respectivo contexto, objetivando evidenciar quais os principais problemas do aplicativo que estão afetando a experiência dos usuários. Esses dados, conforme esperado, podem servir de insumo para a equipe de desenvolvimento, em forma pedidos de novas *features*, correções de erros e falhas etc.

Os resultados também indicaram que é possível obter, tratar e extrair informações sobre a experiência em uso dos usuários.

| Cluster | Assunto | Quantidade |
|---|---|---|
| Verde | Dificuldade de cadastro | 324 |
| Rosa | Problemas na recuperação de senha | 662 |
| Vermelho | Problemas genéricos de execução | 13 |
| Azul | Problemas com reconhecimento facial | 33 |

Tabela 6 – Distribuição dos comentários por *cluster*.

*Fonte: Elaborada pelos autores*

### 4.3.3 Etapa 3 - Carregar

Nessa etapa, foi desenvolvida uma dashboard que formaliza os resultados das metricas encontradas na fase anterior. Dessa forma, sempre que os dados passarem por esse tratamento, classificação e análise da Etapa 2, os dados estão disponíveis para consulta e distribuição posterior.

## 4.4 Considerações finais do capítulo

Este capítulo conclui a apresentação da proposta de trabalho, apresentando as ferramentas utilizada bem como um modelo de obtenção, categorização e análise dos dados. Foi apresentado uma prova de conceito do processo proposto, além da análise dos resultados obtidos. Esses resultados foram de suma importancia para complementar as ideias iniciais do projeto, além de indicarem pontos de atenção e de análise para o prosseguimento do produto final.

Percebeu-se também que há a possibilidade de utilização de outras arquiteturas e algoritmos classificadores, que passarão por uma etapa maior de implementação e verificação. Algumas dessas ideias são:

- Quais algoritmos mais sofisticados podem ser utilizados para classificar comentários que tratam de usabilidade ou não?
- Quais algoritmos mais sofisticados podem ser utilizados para classificar, a partir de um *set* de dimensões e/ou heurísticas de usabilidade, os comentários que tratam de usabilidade?
- Como os critérios de classificação podem ser melhor definidos para categorizar os comentários desse *dataset*, que servirá para o treinamento do algoritmo?
- Como extrair a informação de qual funcionalidade se trata, evoluindo a partir da ideia obtida após a clusterização no dendograma?

# 5 Resultados

Este capítulo tem como objetivo a apresentação dos resultados obtidos com a proposta de execuçao do modelo de identificação de problemas de usabilidade, que teve como base a execução do processo de trabalho apresentado no capítulo anterior.

## 5.1 Etapa 1 - Extrair

Os dados foram coletados utilizando a biblioteca *Google Play Scraper*, durante o período de maio a dezembro de 2020. Os comentários foram coletados a partir da página de avaliação do aplicativo na *Play Store*.

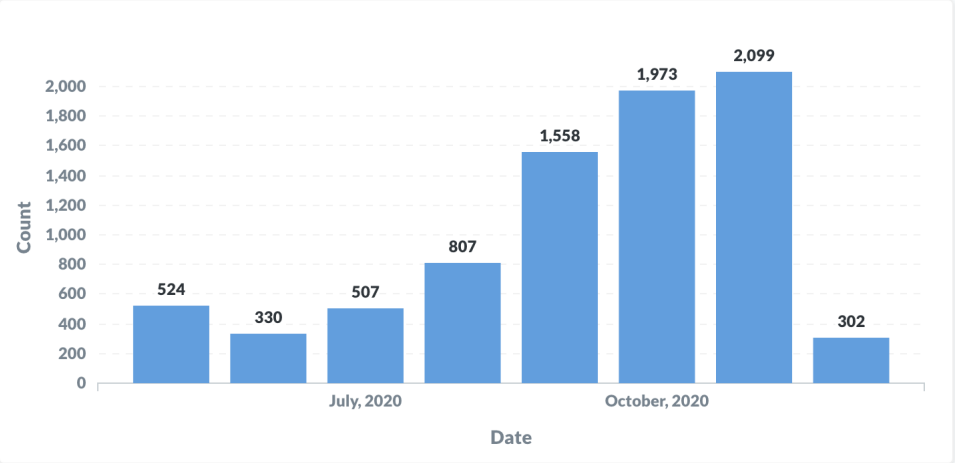


Figura 5 – Distribuição das *reviews* coletadas entre Maio e Dezembro de 2020.

*Fonte: Elaborada pelos autores*

Inicialmente, foram coletados todos os aplicativos que pertenciam à conta do Governo Brasileiro. Porém, devido ao volume, resolvemos continuar apenas com os comentários do aplicativo previamente estudado nos resultados preliminares.

| **Coletados** | **Selecionados** |
|---|---|
| 48100 | 8100 |

Tabela 7 – Total de comentários coletados e comentários do aplicativo selecionado.

*Fonte: Elaborada pelos autores*

Devido à natureza do aplicativo selecionado para estudo, que tem como objetivo servir de interface de autenticação para serviços digitais do governo brasileiro, pudemos

ter acesso a uma gama de comentários diversificada, uma vez que o público-alvo é todo cidadão brasileiro e não um nicho de usuário específico (como *hard-users* de tecnologia, por exemplo).

Os dados são persistidos durante todo o fluxo de extração por meio de um *script* que os armazena no *Postgres*. Assim, imediatamente após a extração dos dados, eles são armazenados e disponibilizados para manipulação e análise.

Foi realizada uma pequena modelagem para englobar os dados coletados e as futuras classificações a serem realizadas.

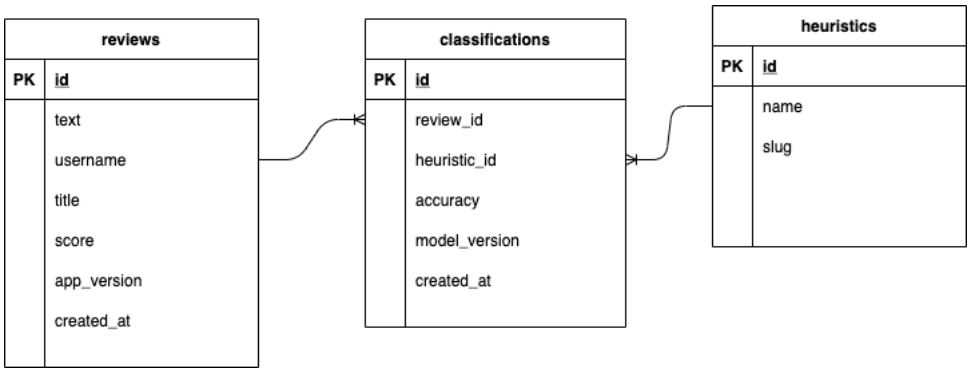


Figura 6 – Modelo físico do banco de dados elaborado para armazenar os comentários e classificações.

*Fonte: Elaborada pelos autores*

Na tabela de *reviews*, referente aos comentários coletados, são salvos os seguintes dados: o texto (*text*), o nome do usuário (*username*), o título do comentário (*title*), a classificação do aplicativo dada pelo usuário ao comentar (*score*), a versão do aplicativo em que o comentário foi feito (*app_version*) e a data de criação do comentário (*created_at*).

Na tabela de heurísticas, são salvos: o nome (*name*) e a sigla da heurística (*slug*) (referentes a cada uma que será utilizada na etapa de classificação manual e que servirá posteriormente como insumo para o classificador automatizado).

Por fim, na tabela de *classifications*, referente às classificações realizadas, são persistidas as chaves primárias do comentário (*comment_id*) e da heurística (*heuristic_id*) que o classificador automatizado classificou, além da acurácia resultante, a versão do modelo do classificador utilizado para a classificação e a data em que a classificação foi realizada.

Assim que os dados são salvos no banco de dados, uma instância do *Metabase* que foi sicronizada exibe as informações atualizadas.

O *Metabase* sincroniza regularmente a base de dados. Assim, a medida que novos comentários ou novas classificações são persistidas, ficam disponíveis logo após o próximo ciclo de sincronização da ferramenta.

## 5.2 Etapa 2 - Transformar

Esta seção trata dos resultados obtidos com a transformação dos dados. Trata desde os subprodutos obtidos com a identificação dos padrões nos textos de forma manual, a partir de uma categorização multietiquetas pré-definidas, até a elaboração e treinamento do modelo automatizado.

Também são feitas considerações sobre o processo de várias iterações de treinamento e classificação manual e automatizada do modelo.

### 5.2.1 Higienização dos dados

A princípio, a preparação dos dados obtidos na etapa de extração se daria por um processo de lematização, redução de radicais, remoção de caracteres especiais e *emojis*.

Porém, percebeu-se que, embora não possa haver mudança representativa na acurácia desse modelo específico, em alguns casos essas informações podem até mesmo melhorar os resultados (SURIKOV; EGOROVA, 2020). Os *emojis* e palavras que indicassem algum tipo de entonação, como caracteres repetidos (exemplo: “muuuuitooo”), podem servir, por exemplo, para processamentos de dados futuros como classificação por sentimentos, mesmo que estejam fora do escopo desse trabalho. Por isso, decidimos não realizar transformações durante limpeza dos dados.

### 5.2.2 Classificação manual da base de treinamento

Conforme decidido anteriormente, foram escolhidas as heurísticas de Nielsen para classificar os comentários. Essa primeira fase do subprocesso tem como produto um grupo de comentários classificados manualmente a partir dessas heurísticas.

Essa é a grande diferença entre o modelo apresentado nos resultados preliminares: cada comentário pode receber uma ou mais etiquetas ao mesmo tempo; diferente do primeiro modelo, no qual a classificação era binária, apenas para usabilidade ou não.

Além da possibilidade de ser relacionado com uma ou mais heuristicas de Nielsen, um comentário também pode ser categorizado como *OTHER*, que é uma classificação designada a comentários que não se encaixam em nenhuma heurística.

Para classificar cada comentário de acordo com uma ou mais heurística, a pessoa responsável pela classificação respondia a seguinte pergunta, após a leitura de cada comentário: “Sobre quais heurísticas este comentário indica tratar?”.

A relação entre as etiquetas e seu significado é descrita na Tabela 8.

| Sigla | Descrição |
|---|---|
| H1 | Visibilidade do *status* do sistema |
| H2 | Correpondência entre o sistema e o mundo real |
| H3 | Controle e liberdade para o usuário |
| H4 | Consistência e padronização |
| H5 | Prevenção de erros |
| H6 | Reconhecimento em vez de memorização |
| H7 | Eficiência e flexibilidade de uso |
| H8 | Estética e *design* minimalista |
| H9 | Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros |
| H10 | Ajuda e documentação |
| OTHER | Comentário não se encaixa em nenhuma heurística |

Tabela 8 – Etiquetas criadas para a classificação.

*Fonte: Elaborada pelos autores*

Em termos práticos, cada texto recebe o valor verdadeiro (1.0) ou falso (0.0) para cada etiqueta que a pessoa classificadora julgar necessária, conforme exemplificado na Figura 7.

```
{
  "cats": {
      "H1": 0.0,
      "H2": 0.0,
      "H3": 0.0,
      "H4": 0.0,
      "H5": 1.0,
      "H6": 0.0,
      "H7": 1.0,
      "H8": 0.0,
      "H9": 1.0,
      "H10": 1.0,
      "OTHER": 0.0
  },
  "text": "pessimo reconhecimento facial nao funciona nunca"
}
```


Figura 7 – Exemplo de classificação manual que servirá como *input* no classificador.

*Fonte: Elaborada pelos autores*

### 5.2.3 Arquitetura do algoritmo e treinamento do modelo

Os comentários classificados manualmente foram divididos de forma aleatória, 80% para treino e 20% para teste, uma técnica de *machine learning* conhecida como *hold-out* (CLAUDE, 2011).

A ferramenta escolhida para processar e treinar o modelo foi o *framework spaCy*, que oferece *pipelines* pré-treinadas e pré-configuradas para processamento de linguagem natural.

As *pipelines* de treinamento são responsáveis por receber os dados (aqui, os textos das reviews) e transformá-los para os algoritmos de *machine learning* também disponibilizados pela ferramenta.

O categorizador de textos *spaCy* é alimentados por modelos estatísticos. Cada estimativa feita por esse componente é uma predição baseada nos pesos atuais do modelo. Esses pesos são estimados durante a fase de treinamento.

Por exemplo, se palavras como “dificuldade“ e “cadastrar“ aparecem muito em frases etiquetadas com H7, o algoritmo considera que essas palavras possuem um peso maior ao calcular a estimativa final do que comparadas com outras palavras como “terrível“ ou “obrigado“.

Em linhas gerais, o treinamento passa por um processo iterativo, no qual, em cada iteração, os resultados da previsão do modelo são comparados com os resultados de referência (fornecido pelo *input* inicial) para reajustar os valores dos pesos utilizados para o cálculo estatístico da solução.

Dessa forma, pode-se reduzir o gradiente, vetor que indica sentido e direção para qual o peso pode ser ajustado para atingir o maior valor possível, o que resulta em previsões mais próximas das etiquetas de referência.

Essa técnica feita para comparar os resultados e reajustar os pesos é conhecida como *backpropagation*.

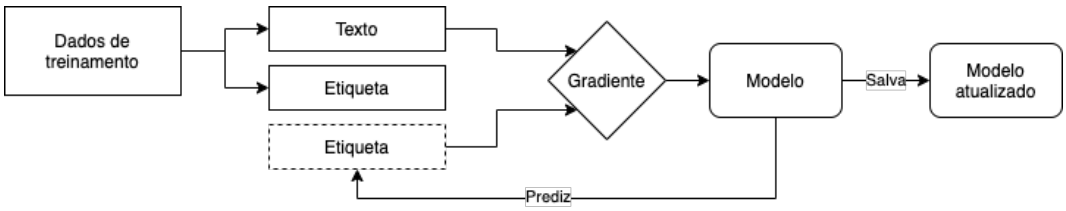


Figura 8 – Organização da arquitetura da *pipeline* de classificação.

*Fonte: Adaptado da documentação do spaCy*

Depois dessa etapa o modelo de avaliação é criado, empacotado e carregado em um *script* que recebe um comentário e retorna as heurísticas identificadas pelo modelo e sua respectiva acurácia.

O formato da resposta é idêntico ao da entrada, com a diferença de que os resultados não são binários (1.0 ou 0.0), mas transitam entre esses dois valores.


Figura 9 – Exemplo de *output* retornado pelo modelo.

*Fonte: Elaborada pelos autores*

Desse resultado, pode-se definir um valor mínimo para considerar ou não o resultado fornecido. Por exemplo, caso o valor seja de 75% (ou 0.75 na figura), heurística com acurácia maior que 0.75 é considerada válida. Assim, na Figura 9, as etiquetas válidas seriam: H7, H9 e H10.

Caso nenhuma acurácia atinja o valor mínimo para ser aceito, o comentário é classificado como *OTHER*. Caso o algoritmo identifique *OTHER* com acurácia maior ou igual que o valor mínimo (no caso dessa etiqueta ser inserida no modelo, como na terceira versão), todas as outras etiquetas são ignoradas, e o comentário é classificado apenas como *OTHER*.

### 5.2.4 Atualização da base de dados

Por fim, toda a base de dados de comentários é classificada, e é realizada a atualização no banco de dados e, consequentemente, a sincronização com o *Metabase* para análise posterior.

Após cada classificação, é montado um objeto JSON similar ao descrito na Figura 10 com os dados agregados e prontos para serem salvos. Esse documento é desestruturado, e os dados são relacionados e salvos nas respectivas tabelas, conforme a Figura 6.


Figura 10 – Exemplo do documento final gerado pela *pipeline* de tratamento e classificação dos textos.

*Fonte: Elaborada pelos autores*

### 5.2.5 Analisar resultados

Nessa etapa do processo, além da atualização da base de dados com os comentários classificados pelo modelo, realizaram-se as análises qualitativas e quantitativas, que servem como insumo para a elaboração dos relatórios de análise e novas iterações. Para estas análises, são utilizadas as métricas de *recall*, *precision* e *F-score*.

- **Primeira versão**

O valor mínimo de acurácia, considerando a regra descrita na seção de arquitetura, foi de 75%.

A Figura 11 indica o número total de classificações acima do valor mínimo de acurácia feita para cada heurística.

Em relação as métricas, o algoritmo obteve um *score* geral de 68%. Ou seja, acertou 68% das previsões que fez. A Tabela 9 mostra as métricas obtidas para análise detalhada para cada etiqueta.

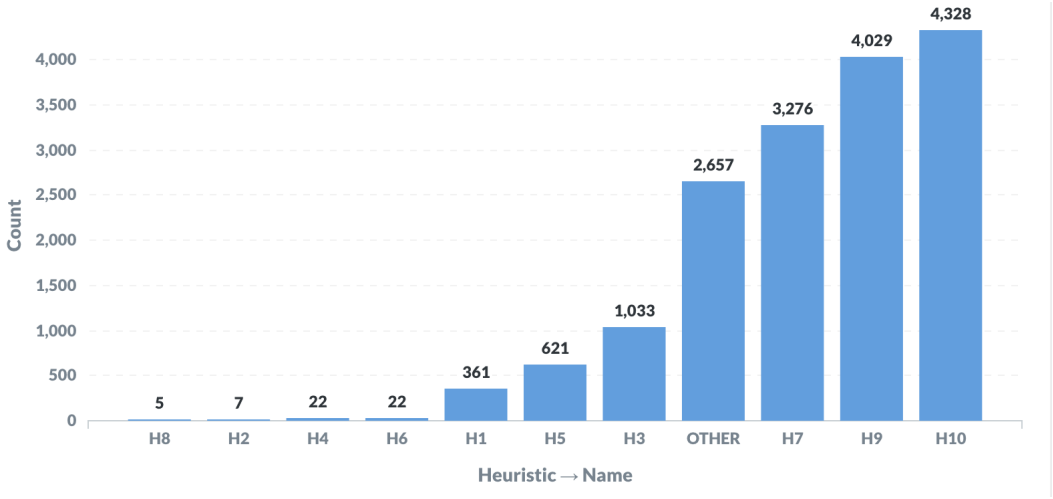


Figura 11 – Distribuição de classificações por heurística na primeira versão do modelo.

*Fonte: Elaborada pelos autores*

| **Etiqueta** | ***Precision*** | ***Recall*** | ***F-score*** |
|---|---|---|---|
| H1 | 0.2 | 0.034 | 0.06 |
| H2 | 0.0 | 0.0 | 0.0 |
| H3 | 0.67 | 0.67 | 0.67 |
| H4 | 0.0 | 0.0 | 0.0 |
| H5 | 0.3 | 0.3 | 0.3 |
| H6 | 0.0 | 0.0 | 0.0 |
| H7 | 0.73 | 0.91 | 0.81 |
| H8 | 0.0 | 0.0 | 0.0 |
| H9 | 0.9 | 0.95 | 0.93 |
| H10 | 0.88 | 0.95 | 0.91 |

Tabela 9 – Métricas de acurácia para a primeira versão do modelo.

*Fonte: Elaborada pelos autores*

Ao analisar os resultados da primeira iteração, percebeu-se que boa parte dos comentários estavam sendo categorizados simultaneamente com as etiquetas H7, H9 e H10, evidenciando, portanto, a possibilidade de existir um acoplamento entre essas três heurísticas.

De outra forma, as heurísticas H2, H4, H6 e H8 não obtiveram bons indicadores, uma vez que o classificador falhou na maioria dos testes. Isso indica que os resultados obtidos para essas etiquetas na Figura 11 podem ser de *outliers*, ou seja, que foram feitas classificações possivelmente indevidas utilizando aquelas etiquetas. Isso também indica que a base de treinamento não possuía classificações suficientes dessas heurísticas para treinar o modelo adequadamente.

Para investigar a possibilidade de enviesamento por parte da dupla ao classificar as

heurísticas no objeto de estudo, foi proposta uma rodada de categorização dos comentários por estudantes da disciplina Qualidade de *Software* do curso de Engenharia de Software.

Além de possuir o contexto para a categorização dos comentários, as discentes também haviam estudado sobre as Heuristicas de Nielsen. Essa decisão da escolha foi inspirada pelo principio de desviesamento (LACEY, 2011).

- **Segunda versão**

O valor mínimo de acurácia continuou de 75%.

Para a nova rodada de classificação, foram distribuídos 50 comentários, por grupo, para 9 grupos compostos por entre 4 e 6 estudantes, nos quais cada estudante deveria, individualmente, fazer a classificação dos comentários.

Os comentários que receberam uma classificação convergente com a maioria das integrantes da equipe (50% + 1) entravam na lista de comentários classificados por heurísticas, os que receberam classificações divergentes foram classificados como *OTHER* e não foram incluídas no *input* de treinamento do modelo.

O *output* de cada grupo, portanto, foi um pacote de dados classificados, que foi adicionado a nossa base de treinamento inicial e inserido no modelo.

Após o processo de treinamento, percebeu-se que a acurácia tinha sido modificada, mas, mesmo assim, ainda não havia evidências concretas de uma melhora pois, como pode ser observado na Figura 12, a maior parte dos comentários foi categorizado como *OTHER*.

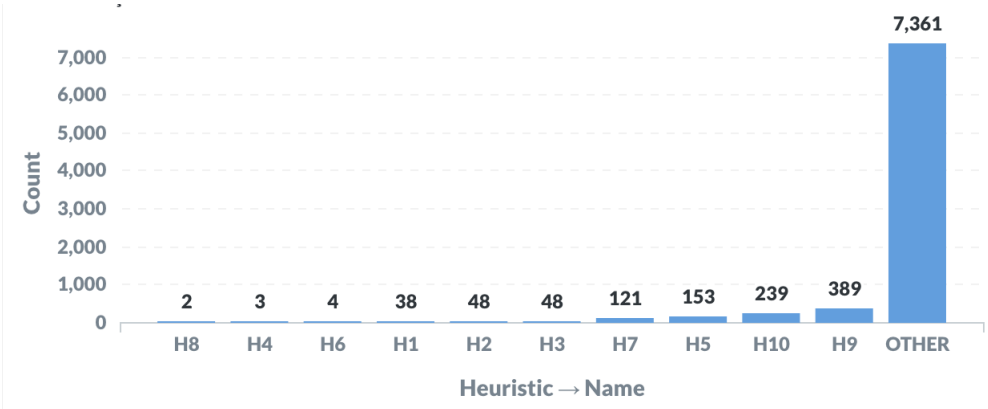


Figura 12 – Distribuição de classificações por heurística na segunda versão do modelo.

*Fonte: Elaborada pelos autores*

Essa versão obteve um aumento de 10% no *score* em relação à versão anterior, chegando a 78%, porém, acertou menos vezes embora de forma mais acurada.

Isso pode ser notado observando tanto os valores da Tabela 12 quanto a distribuição menos destoante entre as heurísticas na Figura 12, demonstrando que a classificação pelos voluntários pode ter ajudado a diminuir o enviesamento.

Porém, as heurísticas H8, H4 e H6 continuaram com poucos resultados, indicando que pode haver poucos comentários disponíveis para treinamento e classificação. Uma das hipóteses é que de fato essa heurística pode ter sido pouco notada dado o contexto da aplicação e dos *reviews*, e consequentemente, ela afetou o resultado.

| **Etiqueta** | ***Precision*** | ***Recall*** | ***F-score*** |
|---|---|---|---|
| H1 | 0.68 | 0.26 | 0.38 |
| H2 | 0.31 | 0.2 | 0.24 |
| H3 | 0.69 | 0.63 | 0.67 |
| H4 | 0.0 | 0.0 | 0.0 |
| H5 | 0.54 | 0.39 | 0.46 |
| H6 | 0.75 | 0.12 | 0.20 |
| H7 | 0.73 | 0.67 | 0.70 |
| H8 | 0.25 | 0.08 | 0.12 |
| H9 | 0.79 | 0.82 | 0.80 |
| H10 | 0.80 | 0.84 | 0.82 |

Tabela 10 – Métricas de acurácia para a segunda versão do modelo.

*Fonte: Elaborada pelos autores.*

- **Terceira e quarta versão**

Dessa vez, convencidos de que o modelo precisaria de mais treinamento manual, foram planejadas mais duas versões. A acurácia mínima do modelo foi mantida em 75% para a terceira e diminuída em 70% para e quarta iteração.

Nesses dois modelos, também, foi realizada uma tentativa de balanceamento entre os comentários que serviriam de treinamento, de forma que a técnica de separar em 80/20 não fosse feita em relação à base inteira, mas para cada classificação individualmente.

Outra pequena experimentação, no terceiro modelo, foi adicionar também a classificação de *OTHER* como uma nova etiqueta, modificando o limite de acurácia para aceitar como *OTHER* aqueles comentários que obtiverem a acurácia maior que 75% e ignorar quaisquer outras classificações. Isso não aparentou ser tão expressivo, e não foi mantido para a versão seguinte.

Os scores da terceira e quarta versão foram 78% e 72%, respectivamente. É interessante notar a perda obtida entre cada versão para as etiquetas H8, H4 e H6. Na terceira versão, além desses resultados, houve uma diminuição das métricas do H2, que na versão dois, havia tido uma leve melhora.

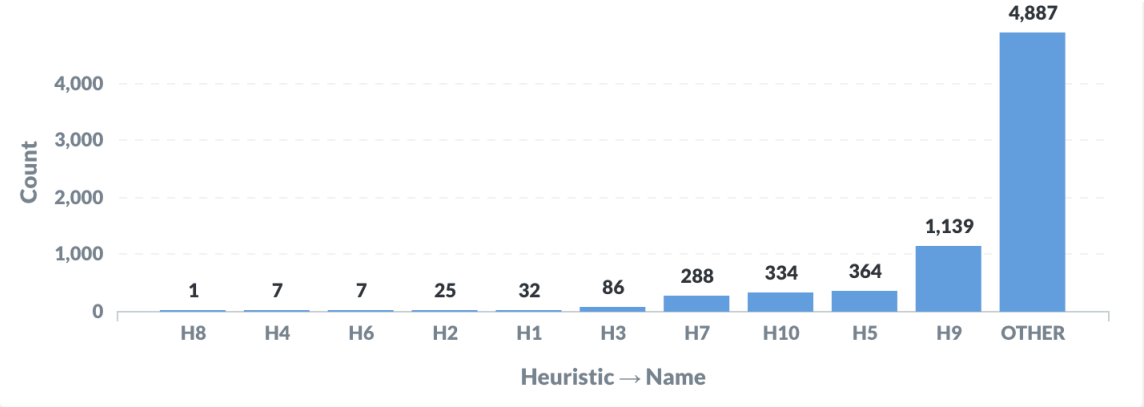


Figura 13 – Distribuição de classificações por heurística na terceira versão do modelo.

*Fonte: Elaborada pelos autores*

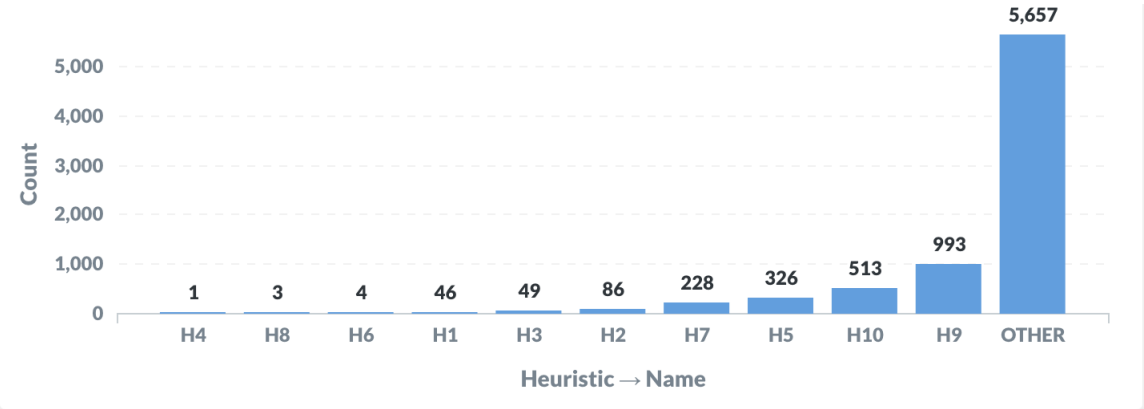


Figura 14 – Distribuição de classificações por heurística na quarta versão do modelo.

*Fonte: Elaborada pelos autores*

Pela falta de tempo hábil, decidiu-se não realizar mais treinamentos pois os resultados já satisfaziam as questões de pesquisa. Serão discutidas possíveis de melhorias que podem ser alcançadas com mais treinamento posteriormente na seção sobre trabalhos futuros.

Todas as verões de análise dos modelos são exibidas no *dashboard* do Metabase para melhor visualização da efetividade de cada modelo. A partir disso, a pessoa encarregada pela análise dentro da equipe pode visualizar e elaborar os outros filtros baseada em cada modelo.

| **Etiqueta** | ***Precision*** | ***Recall*** | ***F-score*** |
|---|---|---|---|
| H1 | 0.27 | 0.15 | 0.19 |
| H2 | 0.0 | 0.0 | 0.0 |
| H3 | 0.55 | 0.72 | 0.62 |
| H4 | 0.0 | 0.0 | 0.0 |
| H5 | 0.42 | 0.3 | 0.35 |
| H6 | 0.0 | 0.0 | 0.0 |
| H7 | 0.70 | 0.78 | 0.74 |
| H8 | 0.0 | 0.0 | 0.0 |
| H9 | 0.82 | 0.86 | 0.84 |
| H10 | 0.89 | 0.91 | 0.90 |
| OTHER | 0.72 | 0.53 | 0.61 |

Tabela 11 – Métricas de acurácia para a terceira versão do modelo.

*Fonte: Elaborada pelos autores*

| **Etiqueta** | ***Precision*** | ***Recall*** | ***F-score*** |
|---|---|---|---|
| H1 | 0.08 | 0.05 | 0.06 |
| H2 | 0.5 | 0.1 | 0.18 |
| H3 | 0.6 | 0.68 | 0.63 |
| H4 | 0.0 | 0.0 | 0.0 |
| H5 | 0.4 | 0.29 | 0.33 |
| H6 | 0.0 | 0.0 | 0.0 |
| H7 | 0.70 | 0.74 | 0.72 |
| H8 | 0.0 | 0.0 | 0.0 |
| H9 | 0.74 | 0.81 | 0.77 |
| H10 | 0.77 | 0.91 | 0.84 |

Tabela 12 – Métricas de acurácia para a quarta versão do modelo.

*Fonte: Elaborada pelos autores*

## 5.3 Etapa 3 - Carregar

Após a classificação e atualização no banco de dados de cada modelo, já é possível relacionar cada comentário e suas respectivas classificações com outros metadados, já coletados anteriormente. Essa visualização é feita através do Metabase. Nas fases anteriores, os gráficos apresentados foram todos elaborados na mesma plataforma.

Podem ser extraídas várias métricas dentre os comentários. Desde contagens, como o total de classificações realizadas para cada modelo, como os total de comentários obtidos ou a evolução do *score* do aplicativo ao longo do tempo, tanto em média quanto por heurística classificada. Esses resultados são consolidados em uma *dashboard*, como na Figura 15.

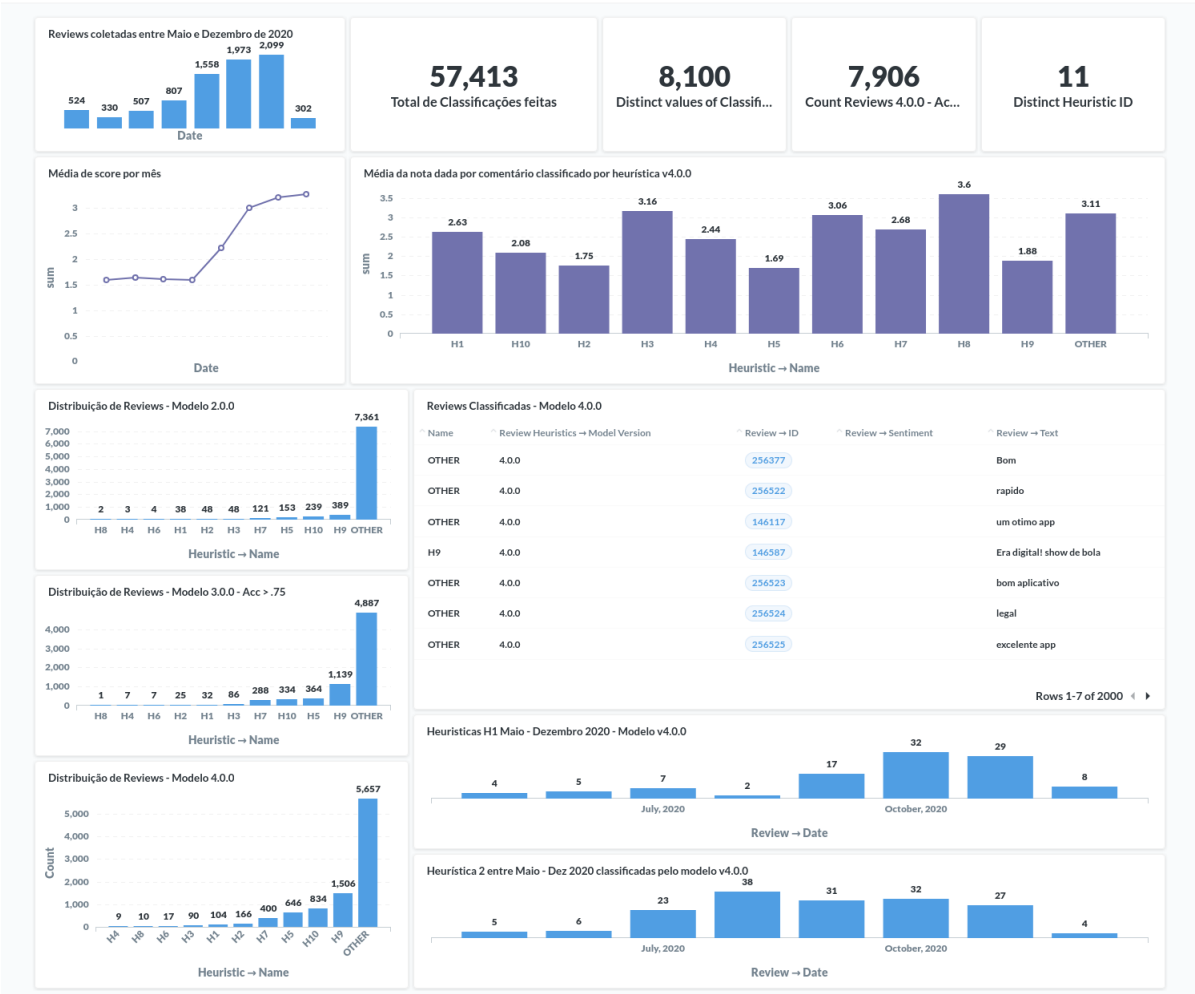


Figura 15 – Representação da *dashboard* criada no Metabase.

*Fonte: Elaborada pelos autores*

Outra métrica que pode ser extraída é a média do *score* dado pelo autor de cada comentário classificado por determinada heurística, como na Figura 16. Esse tipo de dado pode ser útil para auxiliar na priorização da resolução de uma heurística e planos de ação gerais. Por exemplo, se o aplicativo apresenta uma média baixa em relação à

H9, poderiam ser implementadas funcionalidades que ajudem quem está utilizando a diagnosticar, reconhecer e se recuperar de erros.

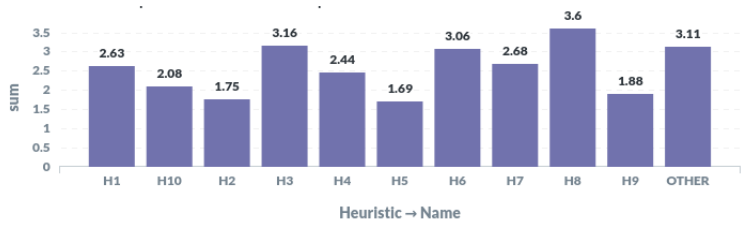


Figura 16 – Média de likes por heurística.

*Fonte: Elaborada pelos autores*

É possível, também, verificar a distribuição da classificação da heurística em um intervalo de tempo. Isso pode ser correlacionado, por exemplo, com uma nova versão do aplicativo que afetou a experiência do usuário. Esse exemplo pode ser visto na como na Figura 17, que mostra as classificações dadas como H1 pela quarta versão do modelo, entre Maio e Dezembro de 2020.

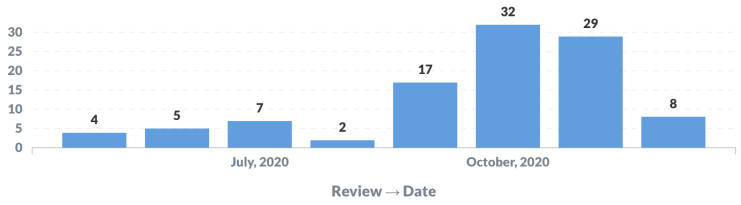


Figura 17 – Histograma das heurísticas classificadas como H1 ao longo do tempo.

*Fonte: Elaborada pelos autores*

A partir da leitura do texto de cada *review* e das suas classificações, é possível, entre outras coisas, priorizar quais são as as funcionalidades prioritárias a serem resolvidas.

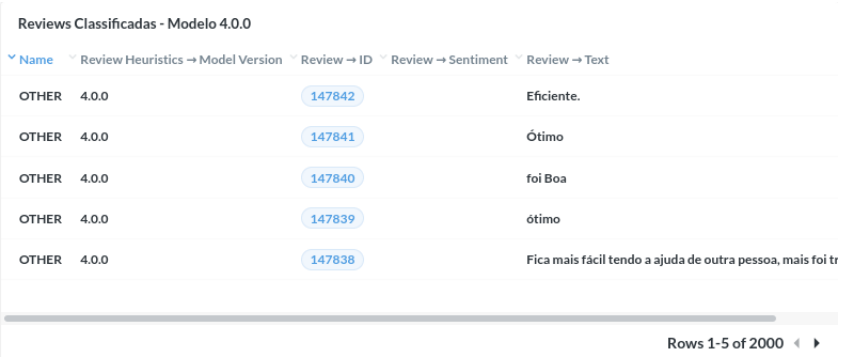


Figura 18 – Listagem dos textos das reviews classificadas por heurística.

*Fonte: Elaborada pelos autores*

Um caso de uso seria adicionar mais validações em um formulário para prevenir que as pessoas entrem em estados de erros não reversíveis em uma determinada funcionalidade

que aparece muito relacionada a heurísticas H5 e/ou H9. Uma evolução, discutida mais a fundo nos trabalhos futuros, seria automatizar também a classificação dos comentários por assuntos e correlacionar estes dados.

Para a geração de um relatório, por exemplo, a ferramenta oferece a possibilidade de que esses e novos gráficos sejam enviados periodicamente para partes interessadas ou sempre que houverem novas atualizações. Também é possível exportá-la para outros formatos como PDF ou JPG, para consolidação dos dados e utilização em análises pela equipe responsável.

## 5.4 Considerações finais do capítulo

Ao final dos resultados preliminares, algumas ideias e dúvidas foram elaboradas para guiar a implementação e interpretação dos resultados obtidos, além da evolução dos resultados preliminares. Nessa seção, elas são respondidas de acordo com os resultados finais apresentados neste capítulo.

- **Quais algoritmos mais sofisticados podem ser utilizados para classificar comentários que tratam de usabilidade ou não?**

  Essa abordagem não foi explorada nos resultados, uma vez que se optou por classificar utilizando heurísticas e não mais usabilidade ou não. Essa pergunta, entretanto, motivou a criação da etiqueta *OTHER*, que foi importante para selecionar os comentários que não se referem a nenhuma heurística.

- **Quais algoritmos mais sofisticados podem ser utilizados para classificar, a partir de um *set* de dimensões e/ou heurísticas de usabilidade, os comentários que tratam de usabilidade?**

  Essa pergunta guiou a escolha de um algoritmo de classificação *multilabel*, que pode classificar um texto dentre várias etiquetas pré-definidas e utilizadas para treinamento do modelo.

- **Como os critérios de classificação podem ser melhor definidos para categorizar os comentários deste *dataset*, que servirá para o treinamento do algoritmo?**

  Os critérios de classificação foram definidos utilizando as heurísticas de Nielsen. Cada comentário foi classificado de acordo com as respostas da pergunta elaborada na fase de classificação manual.

  Essa pergunta foi definida para responder de maneira simples ao problema do critério de classificação, para não bloquear o prosseguimento do trabalho, que era investigar

e propor uma forma de extrair as informações de forma automatizada e não apenas propor essa estratégia de classificação manual.

- **Como extrair a informação de qual funcionalidade se trata, evoluindo a partir da ideia obtida após a clusterização no dendograma?**

  Essa pergunta não foi respondida nesse trabalho, mas deixada para possíveis trabalhos futuros.

# 6 Conclusão

Para atingir o objetivo geral de propor uma forma de identificar problemas de usabilidade de aplicativos móveis a partir da análise de comentários dos usuários na loja de aplicativos, foi necessário dividir a implementação de um modelo de análise e apresentação de dados em quatro iterações.

Depois, ao analisar qual modelo era mais acurado, foi preciso verificar se existia enviesamento na primeira versão do classificador. Para isso, foram treinados novos modelos utilizando classificacões feitas por outras pessoas.

Esses objetivos e os desafios decorrentes dos objetivos específicos, derivados do objetivo geral, são discutidos individualmente nos próximos tópicos.

- **Compreender quais são as técnicas e métodos utilizados para classificar a qualidade de um produto de *software* em relação à usabilidade**

Características de usabilidade são um conjunto de atributos de qualidade, que avaliam o quão fácil é uma interface quanto a ser utilizada. Por isso, compreender e categorizar essas características é também uma parte muito importante do processo.

A partir da percepção a que pessoas desenvolvedoras têm ao identificar problemas de usabilidade em aplicativos baseadas nos comentários de texto das usuárias, foi pensada uma proposta de coleta, classificação e categorização desses, de forma semiautomatizada.

Para o desenvolvimento da solução proposta, foi relizada uma pesquisa exploratória, e, com uma base de referenciais teóricos, decidiu-se por resolver o problema identificado a partir de uma abordagem de processamento de linguagem natural para a classificação e categorização dos comentários feitos por pessoas usuárias na loja de aplicativos da Google. A abordagem de classificação por heurísticas de Nielsen se mostrou como a melhor opção para o contexto de usabilidade abordado no presente trabalho.

- **Interpretar os problemas relatados e catalogá-los segundo os métodos utilizados para classificação de características de usabilidade**

Além do ciclo de coleta e classificação dos comentários obtidos, um dos objetivos propostos pelo trabalho foi a disponibilização de uma ferramenta de análise de dados. Foi escolhida a *Metabase* como plataforma de visualização de dados, pois oferece a possibilidade de organizar as formas de visualização de acordo com a necessidade da pessoa usuária e de ter a alternativa de sicronizar os dados recebidos direto com o banco de dados quase em tempo real.

Com essas informaçoes, as usuárias podem realizar análises baseadas em dados relacionados a problemas de usabilidade.

Em linhas gerais, a implementação da solução proposta tem seu objetivo alcançado, o que pode ser evidenciado na extração de informações de heurísticas a partir do treinamento dos modelos apresentados.

## 6.1 Relevância do trabalho

Esse trabalho pode servir como inspiração para novas abordagens na extração de informações de comentários de usuários. Embora grandes equipes com muitos recursos hoje já consigam lidar com o aprimoramento da usabilidade de seus sistemas utilizando soluções como testes de usabilidade com usuários reais ou testes A/B, isso não é a realidade de muitas equipes que não possuem recursos (financeiros ou tencológicos) para investir na tomada de decisão de qualidade orientada à dados.

Soluções como esta podem beneficiar essas equipes a escalar a gerência de qualidade à medida que a quantidade de usuários aumenta. Esse benefício também se estende para equipes que já lidam com um número grande de informações, ajudando-as a ingerir esses dados e obter informações valiosas que seria muito difícieis de obter com soluções não automatizadas.

## 6.2 Trabalhos futuros

Embora o objetivo proposto tenha sido alcançado, a acurácia pode ser melhorada a partir da realização de mais ciclos de treinamento com uma quantidade maior de comentários sendo fornecida como insumo. Por isso, como trabalhos futuros, são sugeridas as seguintes abordagens:

- **Realização de um novos ciclos de treinamento com uma maior quantidade de comentários**

  Entende-se que o modelo pode entregar maior acurácia se outros tipos de aplicativos forem inseridos na classificação do modelo, diminuindo algum possível enviesamento entre as informações contidas nos textos específicas àquele contexto.

- **Aprimoramento dos critérios de classificação manual**

  Outro problema percebido durante o trabalho foram os critérios de classificação. Apenas responder uma pergunta que indicaria ou não se um comentário se refere a determinada heurística pode não ser suficiente ou trazer contexto necessário para

quem está classificando qual resposta é mais apropriada. Dessa forma, em trabalhos futuros, sugere-se investir em evoluir a forma que essa classificação manual é realizada.

- **Extração de novas informações dos comentários**

  Conforme mencionado brevemente na seção de resultados, automatizar a extração de qual assunto se trata ou identificar a funcionalidade mencionada nos textos pode trazer novas variáveis a serem correlacionadas durante a análise de resultados e auxiliar a equipe a escolhar e priorizar quais decisões tomar para resolver os problemas identificados.

  Assim, por exemplo, um novo algoritmo poderia detectar que um comentário poderia estar se tratando da funcionalidade "cadastro" e correlacionar "cadastro" com as heurísticas H5 e/ou H9. Dessa forma, as correlações com as percepções de uso ao longo do tempo de uma determinada funcionalidade também poderiam ser lidas de forma mais fácil.

- **Disponibilização da solução com uma licença *open source***

  A equipe acredita que as soluções desenvolvidas podem ser melhor aproveitadas se estiverem disponíveis ao público e que algum possível trabalho futuro possa se apoiar no que já foi desenvolvido. Abrir o trabalho para contribuição em um formato de código aberto é um caminho possível e que interessa os autores.

# Referências

BUIDU, R. *Quantitative vs. Qualitative Usability Testing*. 2017. Disponível em: <https://www.nngroup.com/articles/quant-vs-qual/>. Citado na página 27.

CLAUDE, W. G. S. Holdout evaluation. eds) Encyclopedia of Machine Learning. Springer, Boston, MA, 2011. Citado na página 51.

DOURADO, M. A. D.; CANEDO, E. D. Usability heuristics for mobile applications - a systematic review:. In: *Proceedings of the 20th International Conference on Enterprise Information Systems*. SCITEPRESS - Science and Technology Publications, 2018. p. 483–494. Disponível em: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006781404830494>. Citado na página 25.

FEHNERT, B.; KOSAGOWSKY, A. Measuring user experience: complementing qualitative and quantitative assessment. In: *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services - MobileHCI '08*. ACM Press, 2008. p. 383. ISBN 978-1-59593-952-4. Disponível em: <http://portal.acm.org/citation.cfm?doid=1409240.1409294>. Citado 2 vezes nas páginas 27 e 28.

GAO, C. et al. Online app review analysis for identifying emerging issues. In: *Proceedings of the 40th International Conference on Software Engineering*. Association for Computing Machinery, 2018. (ICSE '18), p. 48–58. Disponível em: <https://doi.org/10.1145/3180155.3180218>. Citado na página 30.

GÄRTNER, S.; SCHNEIDER, K. A method for prioritizing end-user feedback for requirements engineering. In: *2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)*. [S.l.: s.n.], 2012. p. 47–49. Citado na página 27.

GIGERENZER, G.; GAISSMAIER, W. Heuristic decision making. v. 62, n. 1, p. 451–482, 2010. Publisher: Annual Reviews. Disponível em: <https://www.annualreviews.org/doi/10.1146/annurev-psych-120709-145346>. Citado na página 25.

GIL, A. C.; VERGARA, S. C. Tipo de pesquisa. *Universidade Federal de Pelotas. Rio Grande do Sul*, 2015. Citado na página 31.

HASSENZAHL, M. User experience (ux) towards an experiential perspective on product quality. In: *Proceedings of the 20th Conference on l'Interaction Homme-Machine*. [S.l.: s.n.], 2008. p. 11–15. Citado na página 26.

HEDEGAARD, S.; SIMONSEN, J. G. Extracting usability and user experience information from online user reviews. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2013. (CHI '13), p. 2089–2098. Disponível em: <https://doi.org/10.1145/2470654.2481286>. Citado na página 29.

ISO/IEC. *ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. 2011. Citado 2 vezes nas páginas 23 e 24.

LACEY, H. A imparcialidade da ciência e as responsabilidades dos cientistas. 2011. Citado na página 55.

LANDIM, F. L. P. et al. Uma reflexão sobre as abordagens em pesquisa com ênfase na integração qualitativo-quantitativa. *Revista brasileira em promoção da saúde*, v. 19, n. 1, p. 53–58, 2006. Citado na página 31.

LIEBSCHER, P. Quantity with quality? teaching quantitative and qualitative methods in an lis master's program. Graduate School of Library and Information Science. University of Illinois, 1998. Citado na página 31.

MAALEJ, W. et al. Toward data-driven requirements engineering. v. 33, n. 1, p. 48–54, 2016. Conference Name: IEEE Software. Citado na página 27.

MacDonald, C. M.; ATWOOD, M. E. What does it mean for a system to be useful?: an exploratory study of usefulness. In: *Proceedings of the 2014 conference on Designing interactive systems - DIS '14*. ACM Press, 2014. p. 885–894. ISBN 978-1-4503-2902-6. Disponível em: <http://dl.acm.org/citation.cfm?doid=2598510.2598600>. Citado na página 26.

MARCONI, M. d. A.; LAKATOS, E. M. *Fundamentos de metodologia científica*. [S.l.]: 5. ed.-São Paulo: Atlas, 2003. Citado 2 vezes nas páginas 21 e 31.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1990. (CHI '90), p. 249–256. ISBN 978-0-201-50932-8. Disponível em: <https://doi.org/10.1145/97243.97281>. Citado 2 vezes nas páginas 25 e 26.

OHASHI, K. et al. Focusing requirements elicitation by using a UX measurement method. In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE, 2018. p. 347–357. ISBN 978-1-5386-7418-5. Disponível em: <https://ieeexplore.ieee.org/document/8491149/>. Citado na página 26.

PAGANO, D.; MAALEJ, W. User feedback in the appstore: An empirical study. In: *2013 21st IEEE International Requirements Engineering Conference (RE)*. [S.l.: s.n.], 2013. p. 125–134. ISSN: 2332-6441. Citado na página 29.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 29.

PETTERSSON, I. et al. A bermuda triangle?: A review of method application and triangulation in user experience evaluation. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, 2018. p. 1–16. ISBN 978-1-4503-5620-6. Disponível em: <http://dl.acm.org/citation.cfm?doid=3173574.3174035>. Citado na página 26.

PHONG, M. V. et al. Mining user opinions in mobile app reviews: A keyword-based approach (t). In: *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. [S.l.: s.n.], 2015. p. 749–759. Citado na página 29.

PRESSMAN, R. *Software Engineering: A Practitioner's Approach*. sixth. [S.l.]: McGraw-Hill, 2005. Citado na página 23.

SILVA, E. L. d.; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. 3. ed. rev. atual, 2001. Citado na página 32.

SOARES, M. C. F. Algumas considerações sobre coleta de dados para a pesquisa qualitativa. *Razão e Fé*, v. 9, n. 2, p. 67–76, 2007. Citado na página 31.

SURIKOV, A.; EGOROVA, E. Alternative method sentiment analysis using emojis and emoticons. *Procedia Computer Science*, v. 178, p. 182–193, 2020. 9th International Young Scientists Conference in Computational Science, YSC2020, 05-12 September 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050920323942>. Citado na página 49.

TAVAKOLI, M. a. et al. Extracting useful software development information from mobile application reviews: A survey of intelligent mining techniques and tools. 2018. Citado na página 28.