UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

LUCAS DO NASCIMENTO DINIZ

PADRÕES DE LINGUAGEM PARA CLASSIFICAR AUTOMATICAMENTE AS REVISÕES DO USUÁRIO COM BASE NAS HEURÍSTICAS DE USABILIDADE DE NIELSEN

QUIXADÁ

2022

LUCAS DO NASCIMENTO DINIZ

PADRÕES DE LINGUAGEM PARA CLASSIFICAR AUTOMATICAMENTE AS REVISÕES DO USUÁRIO COM BASE NAS HEURÍSTICAS DE USABILIDADE DE NIELSEN

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Rainara Maia Carvalho

Coorientador: Bel. José Cezar Junior de Souza Filho

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D611p Diniz, Lucas do Nascimento.
Padrões de linguagem para classificar automaticamente as revisões do usuário com base nas heurísticas de usabilidade de Nielsen / Lucas do Nascimento Diniz. – 2022.
66 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2022.
Orientação: Profa. Dra. Rainara Maia Carvalho.
Coorientação: Prof. José Cezar Junior de Souza Filho.

1. Heurísticas. 2. Usabilidade. 3. Loja virtual. 4. Usuários da Internet. I. Título.

CDD 005.1

LUCAS DO NASCIMENTO DINIZ

PADRÕES DE LINGUAGEM PARA CLASSIFICAR AUTOMATICAMENTE AS REVISÕES DO USUÁRIO COM BASE NAS HEURÍSTICAS DE USABILIDADE DE NIELSEN

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: ____/ ____/ ____.

BANCA EXAMINADORA

Profa. Dra. Rainara Maia Carvalho (Orientadora)
Universidade Federal do Ceará (UFC)

Bel. José Cezar Junior de Souza Filho (Coorientador)
Universidade Federal do Amazonas (UFAM)

Profa. Dra. Marília Soares Mendes
Universidade Federal do Ceará (UFC)

Profa. Ma. Lívia Almada Cruz
Universidade Federal do Ceará (UFC)

À minha família. Tudo é por vocês.

AGRADECIMENTOS

Agradeço a Deus pelo dom da vida.

À minha mãe, Ines Cristina do Nascimento Diniz, por ser meu refúgio e proteção durante toda a minha vida.

Ao meu pai, Antoniel Ferreira Diniz, por ter lutado e abdicado de tanto para me permitir viver um futuro digno.

À minha irmã, Daniele do Nascimento Diniz, por me fazer sentir querido da forma mais sincera e pura possível, espero me tornar um bom exemplo para a sua vida.

Ao PTF, meus grandes amigos, por me incentivarem e vibrarem com cada conquista minha, vocês são minha âncora.

À Rainara Maia Carvalho, minha orientadora, por ter visto potêncial em mim e por acreditar no meu trabalho.

Ao José Cezar Junior de Souza Filho, meu coorientador, por ter me aconselhado e ajudado durante todo o tempo em que nos conhecemos.

À Lívia Almada Cruz e Marília Soares Mendes, minha banca avaliadora, pela disponibilidade em participar da banca desse trabalho, e pelas suas colaborações e sugestões.

Este trabalho foi financiado em parte pelo Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação – PIBITI. Portanto, minha gratidão ao PIBITI por me apoiar com uma bolsa de estudos durante a minha graduação.

Aos meus amigos de Pedra Branca que ressignificaram a minha vida.

Por fim, agradeço a mim mesmo, pelo meu esforço e por não desistir durante essa caminhada.

Muito obrigado!

“Privado dos ventos, pegue os remos.”

(Provérbio Latino)

RESUMO

As revisões do usuário de aplicativos móveis em lojas online têm sido utilizadas como fonte para inúmeros estudos. Muitos deles focam em indicar novos requisitos funcionais e extrair problemas referentes às características de qualidade, como usabilidade. Uma das estratégias mais conhecidas para avaliar a usabilidade é através das heurísticas de Nielsen. As revisões dos usuários em lojas online podem indicar violações de heurísticas, apoiando assim avaliações mais focadas no *feedback* do usuário. Além disso, essas violações podem ter potencial para automatização. Portanto, o objetivo desse trabalho é investigar o potencial de automatização da identificação das violações às heurísticas de usabilidade de Nielsen. Para isso, primeiro, investigou-se com que extensão as revisões do usuário apresentam violações. Foram coletadas e analisadas 200 revisões do usuário, divididas em 528 sentenças, de 10 aplicativos gratuitos e pagos para Android e iOS. Os resultados mostram que as revisões do usuário têm certo potencial para indicar as violações. Com base na análise realizada, 23 padrões de linguagem para classificar as revisões do usuário foram definidos. Tais padrões revelam um potencial para direcionar os esforços de análise de revisões do usuário em busca de problemas de usabilidade.

Palavras-chave: Heurísticas. Usabilidade. Loja virtual. Usuários da Internet.

ABSTRACT

User reviews of mobile applications in online stores have been used as a source for numerous studies. Many of them focus on indicating new functional requirements and extracting problems regarding quality characteristics, such as usability. One of the most known strategies to evaluate usability is through Nielsen's heuristics. User reviews in online stores can indicate heuristics issues, thus supporting evaluations more focused on the user feedback. In addition, these violations may have the potential for automation. Therefore, the objective of this work is to investigate the potential for automating the identification of violations of Nielsen's usability heuristics. To do this, we first investigate the extent to which user reviews have violations. We collected and analyzed 200 user reviews, divided into 528 sentences, of 10 free and paid apps for Android and iOS. The results showed that user reviews have some potential to indicate violations. Based on the analysis performed, 23 language patterns for classifying user reviews were defined. Such patterns reveal the potential to drive user review analysis efforts for usability issues.

Keywords: Heuristics. Usability. Virtual store. Internet users.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE CÓDIGOS-FONTE

SUMÁRIO

1 INTRODUÇÃO

O *feedback* dos usuários sobre um sistema em execução é essencial para apoiar a evolução dele (SCHNEIDER, 2011). Existem diversas técnicas que coletam dados do usuário para apoiar a evolução do *software* e favorecem a elicitação contínua de requisitos, tais como questionários, entrevistas, grupos de foco e *brainstorming* (NUSEIBEH; EASTERBROOK, 2000). Os usuários também podem fornecer informações sobre o uso de aplicativos de *software* através de revisões do usuário publicadas em sites onde os aplicativos estão disponíveis para *download* (CARREÑO; WINBLADH, 2013).

As lojas de aplicativos, que reúnem as revisões do usuário, fornecem uma fonte considerável de dados. De acordo com Pagano e Maalej (2013), são 22.09 revisões por dia para cada aplicativo, que incluem a data de criação da revisão, uma classificação em estrelas e uma mensagem de texto. A comunidade de Engenharia de Requisitos (ER) está cada vez mais interessada em visualizar as revisões do usuário em lojas de aplicativos como uma fonte potencial dos requisitos do usuário (GROEN *et al.*, 2017). De acordo com Pohl (2010), esses requisitos abrangem as seguintes categorias: requisitos funcionais, restrições e requisitos de qualidade.

Os requisitos de qualidade descrevem as características do produto em várias dimensões que são importantes para os usuários ou para desenvolvedores e mantenedores (WIEGERS; BEATTY, 2013). A importância dos requisitos de qualidade é enfatizada por Wiegers e Beatty (2013), afirmando que eles podem distinguir um produto que apenas faz o que deveria de outro que encanta seus usuários.

Nos últimos anos, diversas pesquisas têm surgido para apoiar a extração de requisitos de qualidade através das revisões do usuário em lojas de aplicativos *online*. Estudos como o de Groen *et al.* (2017) buscam extrair indicativos de qualidade baseados na ISO/IEC 25010 (2011) através da classificação manual de revisões do usuário e definição de padrões de linguagem que permitam automatizar essa classificação. Os resultados deste estudo afirmam que o *feedback* do usuário através das suas revisões podem fornecer informações relevantes sobre a qualidade do aplicativo, mas apenas nos atributos pelos quais os usuários são afetados diretamente, tais como: usabilidade e confiabilidade (GROEN *et al.*, 2017).

Nielsen (1994b) define a usabilidade como um conjunto de fatores que qualificam quão bem uma pessoa pode interagir com um sistema interativo. Esses fatores estão relacionados com a facilidade de aprendizado e recordação, eficiência, segurança e a satisfação do usuário ao utilizar um sistema. Desse modo, a usabilidade endereça, principalmente, a capacidade cognitiva,

perceptiva e motora dos usuários empregada durante a interação (BARBOSA; DA SILVA, 2010).

Existem várias formas de avaliar a usabilidade de um sistema, uma delas é através de um conjunto de princípios de usabilidade chamados de heurísticas. Esses princípios de usabilidade descrevem um conjunto de características de interação e interface desejáveis e apoiam a realização da Avaliação Heurística, um método de inspeção (NIELSEN; MOLICH, 1990). Um dos conjuntos de princípios mais conhecidos é o de Nielsen (1994a), composto, originalmente, por dez heurísticas.

Essas heurísticas são capazes de descrever problemas de interação e interface específicos de usabilidade (NIELSEN; MOLICH, 1990). Mesmo que sejam utilizadas, tradicionalmente, por especialistas durante as inspeções, uma hipótese é que elas podem ser mencionadas nas revisões do usuário e indicar a existência de problemas de usabilidade. Sendo assim, o presente trabalho busca aprofundar as análises do atributo de usabilidade presente nas revisões do usuário, explorando com que extensão é possível identificar violações às heurísticas e, por fim, propor uma abordagem de automatização do processo de classificação.

Dessa forma, pretende-se definir padrões de linguagem como solução para automatizar esse processo de classificação. Essa proposta surge para apoiar a realização de outras atividades de avaliação da interação humano-computador ao invés de substituí-las. O propósito de identificar violações de heurísticas através das revisões do usuário é contribuir na priorização de esforços para a correção dos problemas de usabilidade pelos profissionais que atuam no contexto de aplicações móveis. Além disso, é possível usar as heurísticas de Nielsen para categorizar os problemas em diversos grupos, o que representa um maior nível de granularidade e pode ajudar os profissionais de forma mais precisa. Sendo assim, não existe a intenção de realizar inspeções de software, visto que existem diferenças fundamentais entre as duas atividades. A principal diferença é que o presente trabalho não lida com a interface do sistema, enquanto a inspeção, por exemplo, lida diretamente com a interface.

O objetivo principal deste trabalho é propor padrões de linguagem para automatizar a classificação de revisões do usuário com base nas heurísticas de usabilidade de Nielsen. Para isto, foi necessário classificar manualmente um conjunto de revisões do usuário e, posteriormente, analisar a classificação realizada buscando identificar a recorrência de palavras, radicais ou estruturas linguísticas e a semântica geral observada nas revisões classificadas. A partir dessa análise, tornou-se viável definir um conjunto de padrões de linguagem em alto nível para representar a classificação realizada e, por fim, os padrões foram aplicados atráves de expressões

regulares para classificar automaticamente uma grande quantidade de revisões do usuário e visualizar o comportamento dessa classificação automática.

Este trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta os conceitos teóricos importantes no contexto deste trabalho. No Capítulo 3, são apresentados os trabalhos relacionados, bem como suas semelhanças e diferenças com o presente trabalho. No Capítulo 4, são apresentados os procedimentos metodológicos necessários para a condução deste trabalho. No Capítulo 5, são apresentados os resultados obtidos. No Capítulo 6, são apresentadas as discussões propostas neste trabalho. Por fim, o Capítulo 7 apresenta as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo reúne todos os conceitos relevantes para o pleno entendimento do presente trabalho. A Seção 2.1 conceitua e apresenta características relacionadas com as revisões do usuário. A Seção 2.2 apresenta dois conceitos sobre usabilidade. A Seção 2.3 conceitua as heurísticas de usabilidade de Nielsen e detalha cada uma delas. A Seção 2.4 conceitua e apresenta exemplos de expressões regulares. A Seção 2.5 define e apresenta exemplos dos padrões de linguagem.

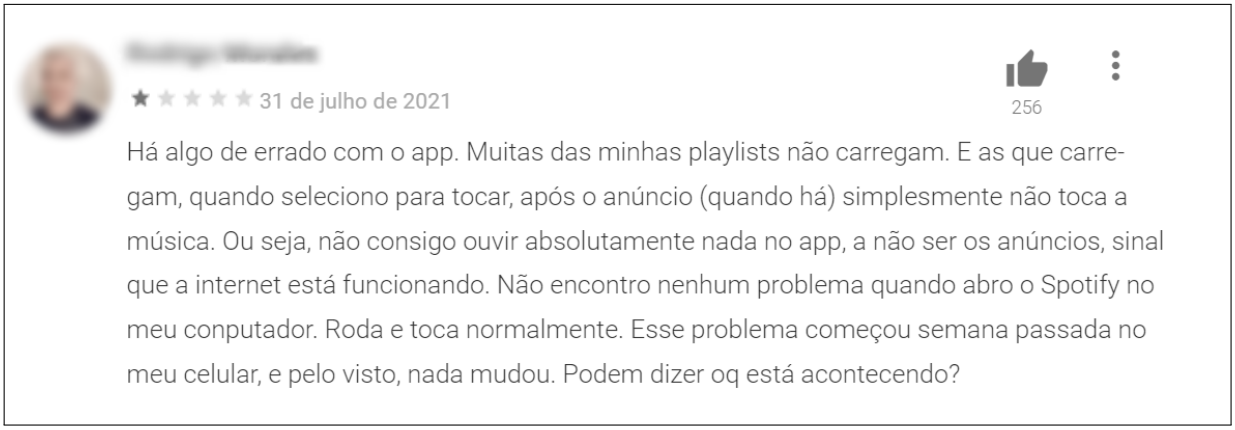
2.1 Revisões do usuário

Uma revisão do usuário pode ser definida como:

> Um pedaço de texto que detalha os prós e contras de um produto e, possivelmente, uma avaliação dele, além de recomendações para compradores em potencial, escritas por um usuário do produto que já possui o mesmo e o utiliza há algum tempo (HEDEGAARD; SIMONSEN, 2013, p. 2090, tradução nossa).

Esse texto pode estar acompanhado de dados adicionais que podem representar uma classificação em estrelas ou até mesmo a data da submissão da revisão (MAALEJ *et al.*, 2016). Elas podem ser escritas por um revisor profissional ou um usuário final comum (HEDEGAARD; SIMONSEN, 2013). Um exemplo de revisão do usuário pode ser visto na Figura 1.

Figura 1 – Revisão do usuário do aplicativo *Spotify* disponível na plataforma *Google Play Store*



Fonte: retirado da *Google Play Store* em 16/08/2021 às 10:17 AM.

O texto das revisões do usuário possuem um conteúdo não estruturado (DABROWSKI *et al.*, 2020). No estudo realizado por Gebauer *et al.* (2008), descobriu-se que as revisões do usuário estão cheias de abreviações, expressões coloquiais e ortografia fora do padrão (intencional e não intencional). De acordo com Hedegaard e Simonsen (2013), essas revisões podem ser

vistas como uma comunicação *eletronic world of mouth* (eWoM). Hennig-Thurau *et al.* (2004) definem a comunicação eWoM como qualquer declaração positiva ou negativa feita por clientes potenciais, reais ou antigos sobre um produto ou empresa, que é disponibilizada a uma infinidade de pessoas e instituições através da internet.

As revisões do usuário nas lojas de aplicativos contêm uma grande diversidade de informações, incluindo reclamações, elogios, cenários de uso, *feedback* sobre recursos e relatórios de *bugs*. Essas classificações são baseadas em uma taxonomia desenvolvida por Guzman *et al.* (2015). O estudo apresentado em Groen *et al.* (2017) afirma em seus resultados que as revisões do usuário podem fornecer informações importantes sobre a qualidade do aplicativo, mas apenas nos atributos pelos quais os usuários são diretamente afetados. Essas revisões têm um impacto direto na popularidade do aplicativo em questão, visto que um aplicativo com uma alta avaliação também tem uma alta classificação nas listas de principais aplicativos, que são disponibilizadas pelas lojas de aplicativos (PAGANO; MAALEJ, 2013).

As revisões do usuário representam um conceito importante para o presente trabalho, visto que tais revisões são a principal fonte de informação utilizada durante esta pesquisa.

2.2 Usabilidade

Barbosa e DA SILVA (2010) afirmam que a usabilidade é o critério de qualidade de uso mais conhecido e, por conseguinte, o mais frequentemente considerado entre outros três critérios de qualidade de uso: experiência do usuário, acessibilidade e comunicabilidade. Nielsen (1994b) define a usabilidade como um critério que mede o quão bem o usuário pode utilizar as funcionalidades do sistema para atingir o objetivo necessário, bem como a satisfação do usuário em decorrência desse uso e está associada com cinco outros atributos que são listados a seguir:

1. Facilidade de aprendizado: o sistema deve ser fácil de aprender para que o usuário possa rapidamente começar a fazer algum trabalho com o sistema;
2. Eficiência: o sistema deve ser eficiente para uso, de modo que uma vez que o usuário aprendeu a usá-lo, um alto nível de produtividade seja possível;
3. Facilidade de recordação: o sistema deve ser fácil de lembrar, para que o usuário casual seja capaz de retornar ao sistema após algum período não tendo usado, sem ter que aprender tudo novamente;
4. Prevenção de erros: o sistema deve ter uma baixa taxa de erros, para que os usuários cometam poucos erros durante o uso do sistema, e para que possam

se recuperar facilmente dos erros. Além disso, erros catastróficos não devem ocorrer;

5. Satisfação do usuário: o sistema deve ser agradável de usar, para que os usuários fiquem, subjetivamente, satisfeitos ao usá-lo.

A usabilidade é aplicada a todos os aspectos do sistema com o qual o ser humano interage, incluindo procedimentos de instalação e manutenção (NIELSEN, 1994b). Segundo a ISO/IEC 25010 (2011), usabilidade é uma característica de qualidade do software composta de seis subcaracterísticas. Ela é definida como sendo "o grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado" (ISO 9241-210, 2019, tradução nossa). Suas subcaracterísticas são listadas a seguir:

1. Reconhecimento de adequação: grau no qual os usuários podem reconhecer se um produto ou sistema é apropriado para suas necessidades;
2. Aprendizagem: grau no qual um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos de aprender a usar o produto ou sistema com eficácia, eficiência, isenção de riscos e satisfação em um contexto de uso especificado;
3. Operabilidade: grau em que um produto ou sistema possui atributos que o tornam fácil de operar e controlar;
4. Proteção de erros do usuário: grau em que um sistema protege os usuários contra cometer erros;
5. Estética da interface do usuário: grau em que uma interface do usuário permite uma interação agradável e satisfatória para o usuário;
6. Acessibilidade: grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades para atingir um objetivo específico em um contexto de uso especificado.

Estudos como o de Groen *et al.* (2017) identificaram a usabilidade como sendo o principal atributo de qualidade presente em revisões do usuário disponíveis em lojas de aplicativos. Portanto, o foco do presente trabalho é indicar violações de heurísticas de usabilidade nas revisões do usuário. No decorrer deste trabalho, será utilizada a definição de usabilidade apresentada por Nielsen (1994b).

2.3 Heurísticas de usabilidade de Nielsen

As heurísticas de Nielsen são um conjunto de diretrizes de usabilidade, que descrevem características desejáveis da interação e da interface (BARBOSA; DA SILVA, 2010). Um conjunto inicial de dez heurísticas de usabilidade, que podem ser utilizadas para identificar problemas encontrados em interfaces do usuário, foi proposto por Nielsen (1994a). Essas heurísticas de usabilidade são utilizadas por especialistas durante a inspeção para realizar a avaliação da interação humano-computador em sistemas interativos (BARBOSA; DA SILVA, 2010). A seguir é listado o conjunto de dez heurísticas de usabilidade de acordo com Nielsen (1994a):

- **H1**. Visibilidade do estado do sistema: o sistema deve sempre manter os usuários informados sobre o que está acontecendo através de *feedback* (resposta às ações do usuário) adequado e no tempo certo;
- **H2**. Correspondência entre o sistema e o mundo real: o sistema deve utilizar palavras, expressões e conceitos que são familiares aos usuários, em vez de utilizar termos orientados ao sistema ou jargão dos desenvolvedores. O *designer* deve seguir as convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica, conforme esperado pelos usuários;
- **H3**. Controle e liberdade do usuário: os usuários frequentemente realizam ações equivocadas no sistema e precisam de uma "saída de emergência" claramente marcada para sair do estado indesejado sem ter de percorrer um diálogo extenso. A interface deve permitir que o usuário desfaça e refaça suas ações;
- **H4**. Consistência e padronização: os usuários não devem ter de se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. O *designer* deve seguir as convenções da plataforma ou do ambiente computacional;
- **H5**. Prevenção de erros: melhor do que uma boa mensagem de erro é um projeto cuidadoso que evite que um problema ocorra, caso isso seja possível;
- **H6**. Reconhecimento em vez de memorização: o *designer* deve tornar os objetos, as ações e opções visíveis. O usuário não deve ter de se lembrar para que serve um elemento de interface cujo símbolo não é reconhecido diretamente, nem deve ter de se lembrar de informações de uma parte da aplicação quando tiver passado para uma outra parte dela. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário;
- **H7**. Flexibilidade e eficiência de uso: aceleradores, imperceptíveis aos usuários novatos,

podem tornar a interação do usuário mais rápida e eficiente, permitindo que o sistema consiga servir igualmente bem os usuários experientes e inexperientes. Exemplos de aceleradores são botões de comando em barras de ferramentas ou teclas de atalho para acionar itens de menu ou botões de comando. Além disso, o *designer* pode oferecer mecanismos para os usuários customizarem ações frequentes;

- **H8**. Projeto estético e minimalista: a interface não deve conter informação que seja irrelevante ou raramente necessária. Cada unidade extra de informação em uma interface reduz sua visibilidade relativa, pois compete com as demais unidades de informação pela atenção do usuário;
- **H9**. Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros: as mensagens de erro devem ser expressas em linguagem simples (sem códigos indecifráveis), indicar precisamente o problema e sugerir uma solução de forma construtiva;
- **H10**. Ajuda e documentação: embora seja melhor que um sistema possa ser utilizado sem documentação, é necessário oferecer ajuda e documentação de alta qualidade. Tais informações devem ser facilmente encontradas, focadas na tarefa do usuário, enumerar passos concretos a serem realizados e não ser muito extensas.

No presente trabalho, as heuristícas de usabilidade de Nielsen são o principal conjunto de diretrizes de usabilidade utilizado para classificar as revisões do usuário. Portanto, é necessário possuir entendimento sobre suas definições.

2.4 Expressões regulares

Uma expressão regular é "uma forma de descrever um conjunto de *strings* sem ter que listar todas as *strings* em seu conjunto" (CHRISTIANSEN *et al.*, 2012, p. 39-40, tradução nossa). Essas *strings* são usadas, fundamentalmente, em tarefas de busca e substituição de texto, como busca de palavras, edição de texto, análise de arquivo, validação de entrada do usuário e controle de acesso (CHAPMAN *et al.*, 2017). Friedl (2006) afirma que expressões regulares são compostas por dois tipos de caracteres listados a seguir:

1. Caracteres especiais chamados de metacaracteres que podem representar intervalos, posições e repetições;
2. Caracteres normais chamados de caracteres literais que representam a si mesmos.

Outra definição é a de Menezes (1998), em que o autor define expressões regulares como "um formalismo denotacional capaz de representar e gerar palavras de uma linguagem".

Além disso, o autor lista exemplos de expressões regulares sobre o alfabeto {a, b}, apresentadas no Quadro 1.

Quadro 1 – Exemplos de expressões regulares e suas respectivas linguagens

Expressão Regular	Linguagem Representada
aa	Somente a palavra “aa”.
ba*	Todas as palavras que iniciam por “b”.
(a + b)*	Todas as palavras sobre {a, b}.
(a + b)*aa(a + b)*	Todas as palavras contendo “aa” como subpalavra.
a*ba*ba*	Todas as palavras contendo exatamente dois “b”.
(a + b)*(aa + bb)	Todas as palavras que terminam com “aa” ou “bb”.

Fonte: (MENEZES, 1998).

Após a construção de uma expressão regular, ela pode ser utilizada para identificar padrões no texto. Existem ferramentas capazes de receber expressões regulares e encontrar uma seção do texto que corresponde à linguagem definida. Um exemplo de ferramenta é o utilitário *egrep* disponível em muitos sistemas operacionais, incluindo *DOS*, *MacOS*, *Windows*, *Unix* e assim por diante (FRIEDL, 2006).

Sendo assim, as expressões regulares são fundamentais para a realização do presente trabalho, visto que serão utilizadas para representar os padrões identificados na classificação das revisões do usuário, além disso, viabilizam a utilização desses padrões para identificar outras revisões do usuário parecidas. No decorrer deste trabalho, será utilizada a definição de expressões regulares apresentada por Christiansen *et al.* (2012).

2.5 Padrões de linguagem

Os padrões de linguagem são definidos utilizando como base o estudo de Groen *et al.* (2017), em que são citados dois aspectos determinantes para identificar um padrão, esses aspectos são: recorrência de palavras, radicais ou estruturas linguísticas e a semântica geral identificada em determinado conjunto de conteúdos textuais. O Quadro 2 exibe dois exemplos de padrões de linguagem definidos por Groen *et al.* (2017) que são utilizados para capturar sentenças que possuem em seu conteúdo alguma menção aos problemas de usabilidade.

A notação desses padrões de linguagem é uma extensão da notação utilizada para representar expressões regulares. Para um conjunto de palavras que possuem significados

Quadro 2 – Exemplos de padrões de linguagem

Padrões
(?<!EN_Negations)(that |)(easy)(to |)(navigate |customize)
(?i)(?<!EN_Negation)(EN_Emphasis)(intuitive)

Fonte: (GROEN *et al.*, 2017).

semelhantes foi definido uma notação especial, por exemplo, *(EN_Persons)* que pode representar qualquer uma das palavras *"I", "you", "we", "us"* etc.

Os padrões apresentados são essenciais para a realização do presente trabalho, visto que é através deles que será possível documentar estruturas recorrentes que serão utilizadas para classificar automaticamente as revisões do usuário. Além disso, a possibilidade de utilizar padrões de linguagem de alto nível permite que esses padrões possam ser representados através de diversos métodos e, consequentemente, aumenta as possibilidades de aplicação desses padrões.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados, que abordam a classificação de revisões do usuário e a automatização dessa classificação. A proposta dos trabalhos, suas semelhanças e diferenças comparadas com o presente trabalho, são detalhadas a seguir.

3.1 *Users – The Hidden Software Product Quality Experts? A Study on How App Users Report Quality Aspects in Online Reviews*

O estudo de Groen *et al.* (2017) é dividido em duas etapas. A primeira etapa verifica a possibilidade de utilizar as revisões dos usuários em lojas de aplicativos para a elicitação e priorização dos requisitos de qualidade. A segunda etapa busca descobrir padrões de linguagem e verificar se esses padrões podem ser identificados por meios automatizados. Para a realização deste estudo, os autores realizaram uma classificação manual das revisões do usuário com base nas características e subcaracterísticas definidas na ISO/IEC 25010 (2011). Posteriormente, compararam as classificações identificando estruturas linguísticas comuns para verificar se essas estruturas podem ser utilizadas na automatização do processo de classificação.

Os resultados de Groen *et al.* (2017) afirmam que as revisões do usuário podem fornecer informações sobre os requisitos de qualidade que os usuários tem contato. Nesse caso, o principal requisito de qualidade em contato com os usuários é a usabilidade. Além disso, Groen *et al.* (2017) definiram padrões de linguagem que correspondem à metade das sentenças classificadas como sendo de usabilidade.

Assim como a pesquisa realizada por Groen *et al.* (2017), o presente trabalho busca verificar a possibilidade de realizar uma classificação das revisões do usuário em lojas de aplicativos e definir uma forma de automatização desse processo. Entretanto, o trabalho citado anteriormente busca classificar as revisões do usuário utilizando categorias definidas com base nas características e subcaracterísticas apresentadas na ISO/IEC 25010 (2011), enquanto o presente trabalho busca classificar as revisões do usuário utilizando diretrizes de usabilidade nomeadas como heurísticas de usabilidade de Nielsen, apresentadas na Seção 2.3.

3.2 *Extracting Usability and User Experience Information from Online User Reviews*

O estudo de Hedegaard e Simonsen (2013) busca mapear a distribuição de informações em revisões do usuário referentes às diferentes dimensões de usabilidade e experiência

do usuário, além de extrair um vocabulário associado para cada dimensão usando técnicas de processamento de linguagem natural e aprendizado de máquina. Exemplos de dimensões utilizadas no estudo são a facilidade de recordação, simpatia, antecipação, afeto e emoção. Para a realização deste estudo, os autores classificaram, manualmente, um conjunto de revisões do usuário baseado em categorias que foram denominadas como dimensões. Para extrair o vocabulário, foi construído um classificador de aprendizado de máquina que discrimina as dimensões com base em palavras ou outros recursos da revisão, que são calculados automaticamente durante o treinamento do classificador.

Hedegaard e Simonsen (2013) encontraram em seus resultados uma ênfase nas menções sobre as dimensões associadas com a usabilidade. Quando se trata das revisões do usuário de jogos, foi encontrada uma ênfase nas menções referentes às dimensões associadas com a experiência do usuário. Os resultados também sugerem que mapear as revisões do usuário para dimensões específicas da experiência do usuário é algo, parcialmente, possível, visto que existem dimensões complexas que dificultam esse processo. Além disso, os resultados afirmam que o classificador funciona bem e tem um potencial para atuar na remoção de sentenças irrelevantes.

Assim como a pesquisa realizada em Hedegaard e Simonsen (2013), o presente trabalho classifica manualmente as revisões do usuário e busca definir uma forma de automatização através dessa classificação. Entretanto, o trabalho citado anteriormente identifica o vocabulário através de técnicas de processamento de linguagem natural e aprendizado de máquina, enquanto o presente trabalho busca definir padrões de linguagem atráves da análise e comparação das classificações realizadas.

3.3 *Automatic Classification of Non-Functional Requirements from Augmented App User Reviews*

O estudo de Lu e Liang (2017) busca classificar as revisões do usuário de forma automática utilizando quatro técnicas de classificação com três algoritmos de aprendizado de máquina. Para a realização do estudo, os autores realizaram uma classificação manual das revisões do usuário, em que as revisões foram rotuladas com base em algumas categorias: usabilidade, confiabilidade, portabilidade, performance, requisitos funcionais e outras. Posteriormente, os autores aplicaram o processo de treinamento dos classificadores escolhidos para a realização do estudo. Por fim, esses classificadores foram testados e avaliados.

Os resultados de Lu e Liang (2017) foram satisfatórios. Os autores chegaram em

uma combinação de técnica de classificação e algoritmo de aprendizado de máquina que possui uma precisão considerável (0.720). Analisando os resultados, a combinação com melhor precisão identifica usabilidade com a segunda maior precisão (0.757) entre a lista de rótulos disponíveis para as revisões do usuário. Esse estudo evidência a viabilidade de automatizar o processo de classificação das revisões do usuário. O processo de classificação pode ser usado para guiar esforços na análise das revisões do usuário.

Assim como o estudo realizado em Lu e Liang (2017), o presente trabalho classifica as revisões do usuário manualmente e propõe uma forma de automatização desse processo de classificação. Entretanto, o trabalho citado anteriormente classifica as revisões do usuário com base em seis rótulos citados nos paragráfos anteriores, enquanto o presente trabalho classifica as revisões do usuário com base nas heurísticas de usabilidade de Nielsen. Além disso, Lu e Liang (2017) propõe uma automatização utilizando algoritmos de aprendizado de máquina, enquanto o presente trabalho propõe a automatização através da definição de padrões de linguagem, a serem identificados através da classificação manual.

3.4 MALTU – Um modelo para avaliação da interação em sistemas sociais a partir da linguagem textual do usuário

O estudo de Mendes (2015) busca construir um modelo para avaliação da interação em sistemas sociais a partir da linguagem textual do usuário (MALTU), que considera textos espontâneos postados pelos usuários em sistemas sociais. Esses textos são as Postagens Relacionadas ao Uso (PRU). Esse modelo busca avaliar a interação com base em duas características: usabilidade e experiência do usuário.

Inicialmente, os autores definiram uma metodologia para a coleta de postagens espontâneas feitas pelos usuários em sistemas sociais. Essa metodologia consiste em definir a base de dados, selecionar os dados, pré-processar os dados, minerar os dados e interpretar os dados. No fim dessa etapa, foi possível observar os padrões nas postagens extraídas, padrões para cada tipo de postagem e padrões para cada meta de usabilidade presente nas postagens. Já a metodologia definida para a avaliação da interação consiste em cinco passos: definição do contexto de avaliação, extração das postagens, classificação das postagens, interpretação dos resultados e o relato dos resultados. A classificação citada como um dos passos da metodologia classifica as postagens com base em seis critérios: tipo de postagem, intenção do usuário, análise de sentimento, funcionalidade, critérios de qualidade de uso e artefato.

Os passos da metodologia definida por Mendes (2015) podem ser aplicados de forma manual ou de forma automática. Para a aplicação de forma automática, os autores construíram uma ferramenta chamada UUX-Posts. Esse estudo enfatiza a possibilidade de definir padrões com base nos conteúdos postados pelos usuários e construir uma ferramenta que utiliza esses padrões para apoiar as avaliações de usabilidade e experiência do usuário.

Assim como o estudo realizado em Mendes (2015), o principal foco do presente trabalho é utilizar um grande conjunto de informações criadas por usuários de aplicativos para extrair informações relacionadas ao uso. Mendes (2015) realiza a classificação por intenção do usuário, por análise de sentimento e por critérios de qualidade de uso, entre outros. Já o presente trabalho busca classificar os dados através das heurísticas de usabilidade de Nielsen. Além disso, Mendes (2015) busca definir um modelo, enquanto o presente trabalho busca definir um conjunto de padrões representados através de expressões regulares para classificar automaticamente as revisões do usuário.

3.5 Comparação dos trabalhos

O Quadro 3 apresenta uma comparação entre os trabalhos relacionados e o presente trabalho. É possível visualizar três critérios de comparação: a classificação inicial, os rótulos e a proposta de automatização da classificação. A classificação inicial se refere à forma que os autores classificaram as revisões do usuário. Essa classificação gera insumos para as demais etapas necessárias para automatizar esse processo. Os rótulos são referentes às categorias utilizadas durante a classificação e representam as informações que devem ser extraídas das revisões do usuário. Por fim, o último critério se relaciona com a estratégia escolhida para construir uma forma de automatizar o processo de classificação.

Quadro 3 – Comparação dos trabalhos relacionados com o presente trabalho

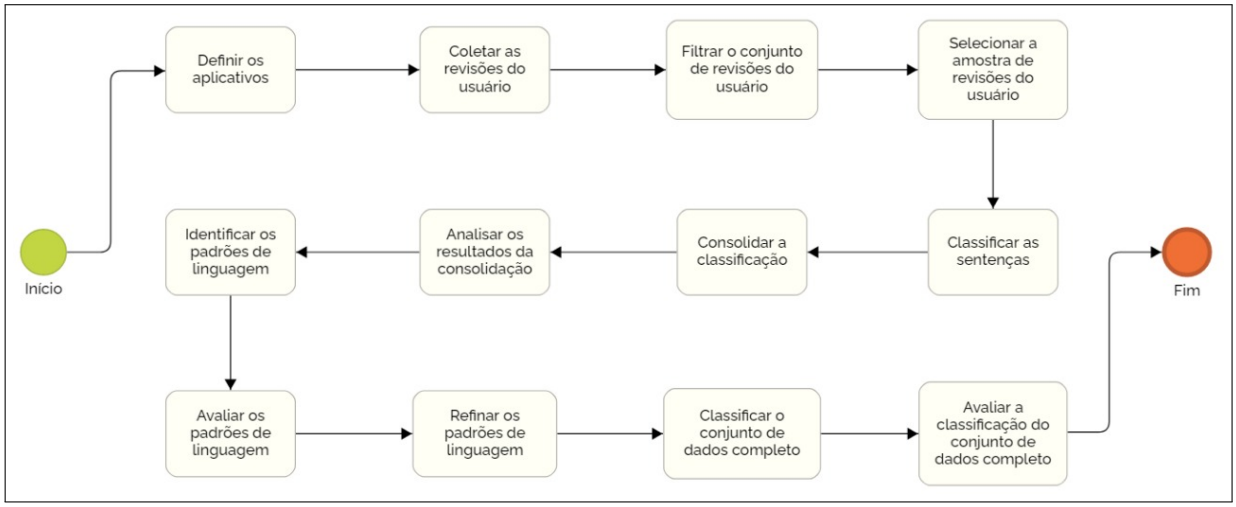
Referência	Classificação inicial	Rótulos	Proposta de automatização da classificação
(GROEN *et al.,* 2017)	Manual	Características e subcaracterísticas definidas na ISO/IEC 25010.	Baseada em padrões de linguagem representados através de expressões regulares.
(HEDEGAARD; SIMONSEN, 2013)	Manual	Dimensões de usabilidade e experiência do usuário.	Baseada em um algoritmo de aprendizado de máquina.
(LU; LIANG, 2017)	Manual	Seis categorias, entre elas, usabilidade confiabilidade, portabilidade, performance, requisitos funcionais e outros.	Baseada em algoritmos de aprendizado de máquina.
(MENDES, 2017)	Manual	Seis categorias, entre elas, tipo, intenção do usuário, análise de sentimento, funcionalidade, critérios de qualidade de uso e artefato.	Baseada em padrões extraídos das postagens.
Presente trabalho	Manual	Heurísticas de usabilidade de Nielsen.	Baseada em padrões de linguagem representados através de expressões regulares.

Fonte: elaborado pelo autor.

4 PROCEDIMENTOS METODOLÓGICOS

Este capítulo apresenta os procedimentos metodológicos que são baseados nas atividades realizadas por Groen *et al.* (2017) A Figura 2 sumariza o fluxo dessas atividades.

Figura 2 – Fluxo de execução dos procedimentos metodológicos



Fonte: elaborada pelo autor.

4.1 Definir os aplicativos

Nesse passo, foi realizada uma pesquisa para selecionar os aplicativos que terão as revisões do usuário coletadas. Essa seleção seguiu quatro critérios para garantir um conjunto diversificado e representativo de fontes de coleta das revisões do usuário. Esses critérios são listados a seguir.

1. Pertencer a uma das cinco categorias identificadas por Pagano e Maalej (2013) que acumulam mais revisões do usuário, listadas a seguir: Entretenimento, Redes sociais, Comunicação, Produtividade e Jogos.
2. Deve possuir mais de duas revisões em cada classificação no *ranking* de cinco estrelas, para que seja possível extrair uma amostra que contenha duas revisões classificadas com cada estrela.
3. Em cada categoria, deve ser selecionado um aplicativo pago e um gratuito.
4. Os aplicativos selecionados devem estar disponíveis em duas lojas de aplicativos: *Google Play Store*[1] e *Apple Store*[2].

[1] https://play.google.com/store
[2] https://www.apple.com/br/app-store/

As lojas de aplicativos fornecem um agrupamento dos aplicativos por categoria e por popularidade, o que facilitou o processo de pesquisa e definição dos aplicativos. As revisões do usuário utilizadas nesse trabalho devem estar escritas em Inglês, sendo assim, as características dos aplicativos foram avaliadas de acordo com a versão disponibilizada nas lojas de aplicativos dos EUA durante os meses de janeiro e fevereiro de 2021. Após realizar a pesquisa para seleção dos aplicativos, foi definido um conjunto de dez aplicativos, listados na Tabela 1. Isso significa que as revisões do usuário foram coletadas de vinte fontes distintas: cinco categorias x dois aplicativos por categoria (um pago e um gratuito) x duas lojas de aplicativos.

Tabela 1 – Aplicativos selecionados em ambas lojas de aplicativos

Aplicativos	Categoria	Monetização	Revisões coletadas
Netflix	Entretenimento	Gratuito	10.000
Akinator	Entretenimento	Pago	8.845
Google Docs	Produtividade	Gratuito	10.000
eDrawings	Produtividade	Pago	591
Facebook	Redes Sociais	Gratuito	7.600
TweetCaster Pro	Redes Sociais	Pago	10.000
Telegram	Comunicação	Gratuito	10.000
TeamSpeak	Comunicação	Pago	3.497
Brawl Stars	Jogos	Gratuito	10.000
NBA2K20	Jogos	Pago	7.132

Fonte: elaborada pelo autor.

4.2 Coletar as revisões do usuário

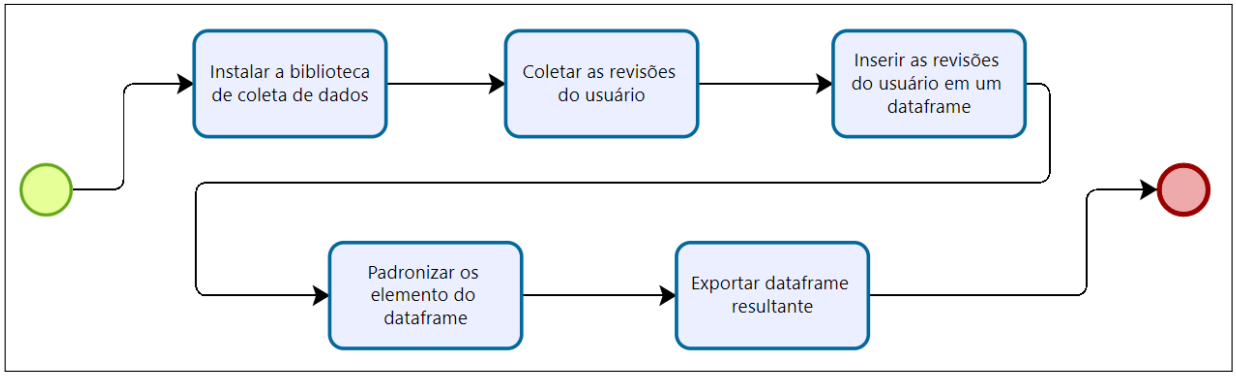
Esse passo consistiu em coletar as revisões do usuário e construir um conjunto de dados com essas revisões. Foram coletados as revisões associadas aos aplicativos definidos no passo anterior. A coleta das revisões do usuário foi feita através da execução de dois *scripts* desenvolvidos utilizando a linguagem de programação *Python* e duas bibliotecas chamadas *Google Play Scraper*[3] e *Apple Store Scraper*[4]. Os dois *scripts* foram construídos com base no fluxograma apresentado na Figura 3. Após recuperar as revisões do usuário, elas foram introduzidas em uma estrutura de dados chamada *dataframe*, presente na biblioteca *Pandas*. As informações coletadas para cada revisão do usuário são apresentadas a seguir:

1. Conteúdo textual da revisão.
2. Classificação em estrelas da revisão.

[3] https://pypi.org/project/google-play-scraper/
[4] https://pypi.org/project/app-store-scraper/

3. Nome do aplicativo da revisão.
4. Loja de aplicativos da revisão.

Figura 3 – Fluxograma de construção dos *scripts* de coleta das revisões



Fonte: elaborada pelo autor.

Os *scripts* utilizados para coletar as revisões do usuário são apresentados no Apêndice B. Ambos *scripts* realizam a sequência de passos descritos na Figura 3. Os algoritmos citados anteriormente foram executados dez vezes cada, uma vez para cada aplicativo. Eles foram utilizados para coletar as revisões do usuário de cada aplicativo definido na etapa anterior. Em cada execução, o *"appName"*, *"appId"*, *"store"* e o *"datasetName"* foram alterados de acordo com o aplicativo em questão. Sendo assim, foram construídos vinte conjunto de dados distintos. A quantidade de revisões do usuário máxima que poderia ser coletado foi limitada em 5.000 revisões por loja de aplicativos, para evitar esforços computacionais desnecessários.

Sendo assim, após a aplicação dos *scripts* citados anteriormente, foi possível construir um conjunto de dados composto de revisões do usuário coletadas de vinte fontes distintas. Além disso, os dados foram todos unificados em um único conjunto de dados que reúne todas as revisões do usuário coletadas. Esse conjunto de dados é composto de 77.665 revisões do usuário distribuídas de acordo com a Tabela 1. Por fim, essa estrutura de dados foi exportada em formato *comma-separated values* (csv) para facilitar a integração com outras ferramentas que foram utilizadas nas demais etapas dos procedimentos metodológicos.

4.3 Filtrar o conjunto de revisões do usuário

Como explicado na Seção 2.1, as revisões do usuário possuem um formato desestruturado. Por esse motivo, foi necessário realizar uma filtragem no conjunto de dados para retirar dados irrelevantes e prejudiciais para a análise. A filtragem do conjunto de dados lidou com três situações distintas. Essas situações são representadas pelos três critérios listados a seguir:

1. Revisões do usuário que possuem conteúdo textual em branco.
2. Revisões do usuário que o conteúdo textual possui apenas *emojis*.
3. Revisões do usuário que possuem conteúdo textual menor do que duas palavras.

Durante a aplicação do primeiro filtro foram encontradas 37 revisões do usuário que não possuíam conteúdo textual. Portanto, foi necessário eliminar essas 37 revisões do conjunto de dados coletado como descreve a Seção 4.2, resultando em um novo conjunto de dados composto de 77.628 revisões do usuário. Na aplicação do segundo filtro, foi possível encontrar 517 revisões do usuário compostas apenas de *emojis*. Após a aplicação desse filtro, o conjunto de dados resultante foi composto de 77.111 revisões do usuário. A aplicação do terceiro filtro revelou 8.024 revisões do usuário compostas de apenas uma palavra. Retirar essas revisões do usuário resultou em um conjunto de dados que possui 69.087 revisões do usuário. Esses dados são apresentados na Tabela 2.

Tabela 2 – Relação entre aplicação dos filtros e o tamanho do conjunto de dados resultante

Filtro	Revisões do usuário descartadas	Tamanho do conjunto de dados resultante
Primeiro filtro	37 revisões do usuário	77.628 revisões do usuário
Segundo filtro	527 revisões do usuário	77.111 revisões do usuário
Terceiro filtro	8.024 revisões do usuário	69.087 revisões do usuário

Fonte: elaborada pelo autor.

4.4 Selecionar a amostra de revisões do usuário

Esse passo consistiu em selecionar, aleatoriamente, uma amostra de dados de revisões do usuário após a filtragem descrita na Seção 4.3. Para formar a amostra de revisões do usuário, foram selecionadas, aleatoriamente, vinte revisões do usuário de cada aplicativo, sendo dez revisões em cada loja de aplicativo (*Google Play Store* e *Apple Store*) e duas revisões para cada uma das cinco possibilidades na classificação por estrelas. Sendo assim, a amostra coletada foi composta de 200 revisões do usuário.

Com o objetivo de simplificar o processo de classificação manual, foi definido que a etapa de classificação iria classificar sentenças ao invés de revisões do usuário por completo,

a fim de simplificar e atomizar a classificação. Nesse ponto, o Código-fonte 3, presente no Apêndice C, foi construído para separar as revisões do usuário de acordo com os seguintes critérios: (1) separar as revisões por "." e (2) separar as revisões por ".(quebra de linha)".

Essas sentenças foram exportadas no formato csv para facilitar o armazenamento dos dados e a integração com outras ferramentas. Após realizar a separação das revisões do usuário, foi criado um conjunto de dados composto de 530 sentenças para serem classificadas de acordo com as heurísticas de usabilidade de Nielsen.

4.5 Classificar as sentenças

Esse passo consistiu em realizar a classificação das 530 sentenças originadas pelas 200 revisões do usuário. A classificação foi realizada por três pesquisadores que possuem conhecimentos relacionados à Interação Humano-Computador. Um dos pesquisadores possui o título de Doutor, outro possui o título de Bacharel e o último pesquisador é um estudante de graduação. Cada pesquisador leu todo o conjunto de sentenças e atribuiu uma ou mais heurísticas que representem a sentença. Além da classificação por heurísticas, cada pesquisador informou seu nível de confiança para aquela classificação e o quanto ele/ela acredita que aquela sentença corresponde à heurística classificada.

Antes de realizar, efetivamente, a classificação, cada pesquisador recebeu uma planilha individual no *Google Planilhas*[5], a fim de evitar o viés durante a classificação, anexada com instruções para realizar a classificação. Os pesquisadores foram instruídos a definir valores de correspondência e confiança para suas classificações de acordo com duas escalas Likert de sete pontos. A escala de nível de confiança possui os seguintes valores nas extremidades da escala: (1) Nenhum pouco confiante e (7) Completamente confiante. A escala de correspondência possui os seguintes valores nas extremidades da escala: (1) Não corresponde nenhum pouco e (7) Corresponde totalmente. As instruções disponibilizadas para os pesquisadores classificarem as sentenças são apresentadas a seguir:

1. A sentença a ser classificada com nenhuma heurística deverá ser anotada com o rótulo NaN.
2. A sentença a ser classificada com determinada heurística deverá ser anotada com o rótulo H*, onde o * será substituído pelo número da heurística de acordo com a Seção 2.3.

[5] https://www.google.com/sheets/about/

Durante a realização da classificação foram encontradas duas sentenças em um idioma diferente do Inglês, sendo assim, tais sentenças foram descartadas. O Quadro 4 contém alguns exemplos da classificação realizada nesta etapa.

Quadro 4 – Exemplos da classificação das sentenças

Sentenças	Heurística
"*I'm happy to help improve it, but i don't understand why it's so easy to destroy your format, and so hard to make it look presentable.*"	H1
"*And this game won't allow me to get past creating my character.*"	H3
"*Moving the icon that allows you to go forward or backwards in the video doesn't fix the issue, as a matter of fact if you try to move the icon before the buffering starts the video freezes at the exact same spot.*"	H9

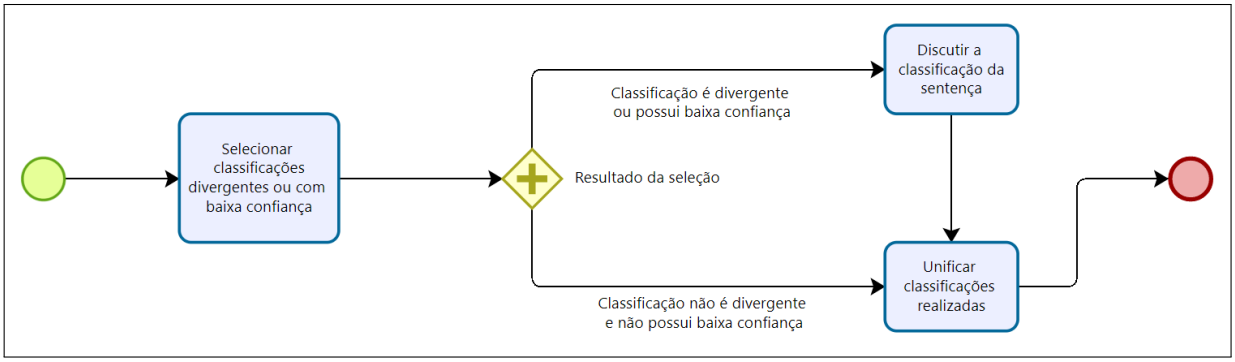
Fonte: elaborada pelo autor.

4.6 Consolidar a classificação

Esse passo consistiu em promover uma discussão entre todos os pesquisadores para chegar em uma classificação comum das sentenças que divergiram no final da etapa de classificação. Para atingir o objetivo desse passo, foi necessário selecionar as sentenças classificadas de forma diferente por um ou mais pesquisadores. Além disso, outro critério para selecionar as classificações foi o nível de confiança menor do que cinco.

Após selecionar as sentenças que divergiram ou que possuem uma classificação de confiança baixa, foi necessário promover uma discussão entre os pesquisadores para analisar a classificação, favorecendo o diálogo e a troca de ideias, e chegar em um acordo sobre a classificação. O fluxo de execução dessa etapa é apresentado na Figura 4.

Figura 4 – Fluxo de execução do passo de consolidação



Fonte: elaborada pelo autor.

A aplicação desses dois filtros revelaram 251 sentenças que seriam pauta das reuniões

para buscar uma classificação em comum entre os pesquisadores. As reuniões realizadas tinham um duração máxima de duas horas onde todos os pesquisadores discutiam suas classificações. Foi estabelecido uma meta de dez sentenças discutidas por reunião, entretanto, os pesquisadores tiveram a autonomia de decidir se iriam discutir mais do que dez sentenças. Baseado na meta estabelecida, 25 reuniões seriam necessárias para consolidar todos os dados. Porém, existiram reuniões em que os pesquisadores discutiram mais do que dez sentenças, consequentemente, diminuindo a quantidade de reuniões. Existiram situações em que a classificação final da sentença foi diferente da classificação inicial de todos os três pesquisadores. Por fim, os dados que passaram pela consolidação foram anexados aos dados que não precisaram seguir por essa etapa e convertidos para o formato csv.

4.7 Analisar os resultados da consolidação

Esse passo consistiu em realizar uma análise quantitativa do conjunto de sentenças classificadas e consolidadas como descrito, respectivamente, nas Seções 4.5 e 4.6. Essas análises foram realizadas utilizando a linguagem de programação *Python*. As principais análises que foram realizadas nesse passo são listadas a seguir:

1. Proporção da ocorrência das heurísticas.
2. Relação entre tamanho das sentenças e ocorrência de alguma heurística.
3. Relação entre a forma de monetização do aplicativo e a ocorrência de alguma heurística.
4. Relação entre a categoria dos aplicativos e a ocorrência de alguma heurística.
5. Relação entre a loja de aplicativos e a ocorrência de alguma heurística.
6. Co-ocorrência de heurísticas.

Além disso, foram realizadas análises para verificar o grau de concordância entre os pesquisadores de acordo com a classificação inicial. Das 528 sentenças, os três pesquisadores concordam totalmente em 344 afirmações (grau de concordância de 65%).

4.8 Identificar os padrões de linguagem

Esse passo consistiu em fazer uma análise do conteúdo buscando estruturas linguísticas comuns nas sentenças classificadas, a fim de definir padrões de linguagem que foram representados através de expressões regulares. A intenção é que tais padrões possam ser utiliza-

dos para classificar qualquer conjunto de dados composto de revisões do usuário postadas em lojas de aplicativos. Para realizar essa análise foi necessário agrupar as sentenças com base em suas respectivas classificações. Nessa etapa, foi extremamente importante considerar o maior nível de detalhes possível, por exemplo: palavras que são comuns, combinações de palavras que são comuns e suas relações com as heurísticas associadas.

Os padrões foram definidos com base nos aspectos determinantes para identificar um padrão (Seção 2.5). Esses aspectos são: recorrência de palavras, radicais ou estrutruras linguísticas e a semântica geral observada em determinado conjunto de sentenças.

Sendo assim, foram gerados dez conjuntos de sentenças agrupadas de acordo com sua classificação. Posteriormente, os agrupamentos foram explorados de forma individual, elencando estruturas linguísticas como: verbos, adjetivos, pronomes, substantivos, entre outros. Essas estruturas linguísticas foram analisadas de acordo com a recorrência e a semântica observada de modo geral no conjunto de sentenças agrupadas. Assim, foi possível identificar e definir um conjunto de 21 padrões de linguagem.

Para aumentar o poder de representatividade dos padrões, foi necessário definir conjuntos de palavras que são agrupadas e podem ser inseridas dentro de um determinado padrão. Por exemplo, o conjunto (EN_Pronome) agrupa as palavras que representam alguns dos pronomes ingleses (*I, you, it*). Um exemplo de padrão em alto nível que foi definido para a heurística H10 (Ajuda e documentação) é apresentado a seguir:

(EN_Pronome) (EN_Negação) know ... about

Esse padrão em alto nível busca representar a dificuldade de encontrar, buscar ou procurar algo. Entretanto, esse padrão ainda precisa ser representado utilizando uma expressão regular para que seja útil no processo de testagem automatizada das sentenças. A seguir é apresentado o padrão descrito anteriormente no formato de uma expressão regular:

(i|you|it) . (not|don't|no) .* know .* about*

Uma distinção entre os padrões de linguagem definidos no presente trabalho e as expressões regulares apresentadas na Seção 2.5 é o elemento "..." que representa qualquer ocorrência de texto entre dois outros elementos. A primeira versão dos padrões de linguagem e suas representações através de expressões regulares é apresentada no Apêndice A. Cada padrão de linguagem foi representado utilizando uma expressão regular. Houve um conjunto de sentenças que não permitiu a extração de algum padrão de recorrência ou semântico. Sendo assim, não foi possível definir padrões para a heurística H5 (Prevenção de erros).

4.9 Avaliar os padrões de linguagem

Esse passo consistiu em avaliar o quão bem os padrões de linguagem escritos através de expressões regulares representam o conjunto de dados consolidados na Seção 4.6. Para fazer essa avaliação foi necessário verificar a correspondência dos padrões através de uma classificação baseada nas expressões regulares definidas. Essa classsificação foi realizada no conjunto de dados previamente classificado. As expressões regulares foram utilizadas em conjunto com a linguagem de programação chamada *Python* e a biblioteca *re*[6]. Essa ferramenta utiliza as expressões regulares para capturar a sua ocorrência em textos que estão de acordo com o padrão representado na expressão. Por fim, a classificação realizada na Seção 4.5 foi comparada com a classificação realizada nesse passo. Baseado nessa comparação, foi possível visualizar a precisão, acurácia, abrangência absoluta de cada padrão e abrangência do conjunto de padrões de determinada heurística que se relacionam com a porcentagem de sentenças capturadas por cada padrão e por cada conjunto de padrões. A acurácia foi calculada de acordo com a Fórmula 4.1 e a precisão foi calculada de acordo com a Fórmula 4.2. As métricas coletadas foram baseadas em dados de verdadeiros positivos(VP), falsos positivos(FP), verdadeiros negativos(VN) e falsos negativos(FN). Todas essas informações podem ser observadas no Apêndice A.

$$Acurácia = \frac{(VP+VN)}{total} \tag{4.1}$$

Fonte: (DOWNEY, 2014)

$$Precisão = \frac{VP}{VP+FP} \tag{4.2}$$

Fonte: (DOWNEY, 2014)

4.10 Refinar os padrões de linguagem

Esse passo consistiu em refinar as expressões regulares definidas de acordo com a Seção 4.8. As métricas coletadas na Seção 4.9 foram utilizadas para guiar os esforços de evolução dos padrões de linguagem previamente definidos. Além disso, os pesquisadores analisaram os padrões e suas expressões regulares em busca de oportunidades de melhoria. Algumas

[6] https://docs.python.org/3/library/re.html

modificações foram necessárias para melhorar as métricas associadas a cada expressão regular. Essas modificações foram determinantes para o aprimoramento de todo o conjunto de padrões de linguagem definidos e são guiadas através de duas ações:

1. Inserir novos termos nas expressões regulares existentes;
2. Criar novas expressões regulares.

4.11 Classificar o conjunto de dados completo

Esse passo utilizou as expressões regulares definidas na Seção 4.10 para classificar o conjunto de dados coletado como descrito na Seção 4.2. Para isso, foi necessário manipular o conjunto de dados geral coletado nos passos iniciais desta metodologia e separá-los em sentenças com base nos critérios definidos anteriormente. Sendo assim, o conjunto de dados composto de 69.087 revisões do usuário originaram 178.371 sentenças. Essas sentenças foram inseridas em uma estrutura de dados chamada *dataframe* junto com as outras informações relacionadas a cada sentença (Seção 4.2. Por fim, esses dados foram extraídos em formato csv.

A partir do arquivo csv, através da linguagem de programação *Python* e uma biblioteca chamada *re*, as sentenças originadas foram classificadas utilizando os padrões de linguagem definidos nas etapas anteriores. Sendo assim, o conjunto de dados recebeu uma nova coluna que possuia a classificação automática realizada com base no conjunto de padrões definidos.

4.12 Avaliar a classificação do conjunto de dados completo

Esse passo realizou uma análise quantitativa das revisões do usuário classificadas na Seção 4.11. Assim como descrito na Seção 4.7, as principais análises realizadas foram:

1. Proporção da ocorrência das heurísticas.
2. Relação entre tamanho das sentenças e ocorrência de alguma heurística.
3. Relação entre a forma de monetização do aplicativo e a ocorrência de alguma heurística.
4. Relação entre a categoria dos aplicativos e a ocorrência de alguma heurística.
5. Relação entre a loja de aplicativos e a ocorrência de alguma heurística.
6. Co-ocorrência de heurísticas.

Com base nessa análise, foi possível identificar como as expressões regulares definidas se comportam ao classificar revisões do usuário fora do conjunto de dados referente à

amostra utilizada durante o trabalho.

5 RESULTADOS

Neste capítulo, são apresentados os resultados do presente trabalho. A Seção 5.1 apresenta uma análise dos resultados da classificação manual. A Seção 5.2 apresenta os padrões de linguagem em seu estágio final após a etapa de refinamento. Por fim, a Seção 5.3 apresenta os resultados da análise realizada com o conjunto geral de sentenças classificadas automaticamente.

5.1 Análise dos resultados da consolidação

A análise das classificações consolidadas revelou pontos interessantes que são discutidos a seguir. Foi possível observar que das 528 sentenças, 132 foram classificadas com alguma das dez heurísticas de usabilidade de Nielsen. Sendo assim, as demais 396 sentenças foram classificadas sem nenhuma heurística. Isso significa que 25% das sentenças foram classificadas com alguma heurística, enquanto as outras 75% foram classificadas sem nenhuma heurística. Nas próximas subseções, são apresentados os resultados das análises definidas na Seção 4.7.

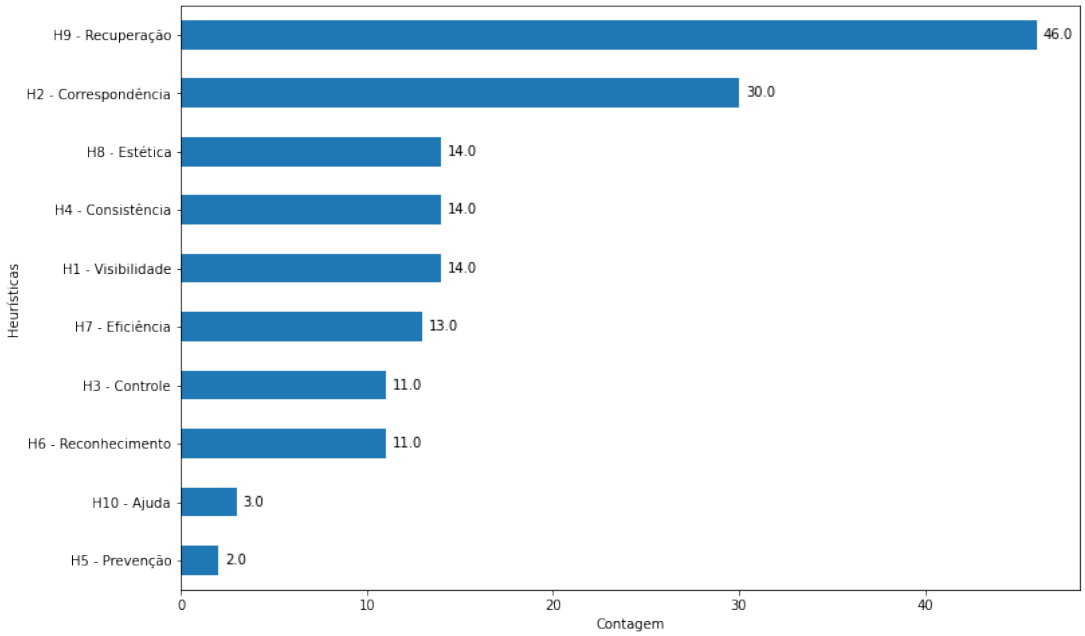
5.1.1 Proporção da ocorrência das heurísticas

Quando se trata da presença individual de cada heurística atrelada às sentenças, pode-se observar a proporção apresentada na Figura 5. Por um lado, percebe-se que a heurística que mais ocorre (46 vezes) nas sentenças é a H9 (ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros). Por outro lado, a heurística que menos ocorre (2 vezes) é a H5 (prevenção de erros).

5.1.2 Relação entre tamanho das sentenças e a ocorrência de alguma heurística

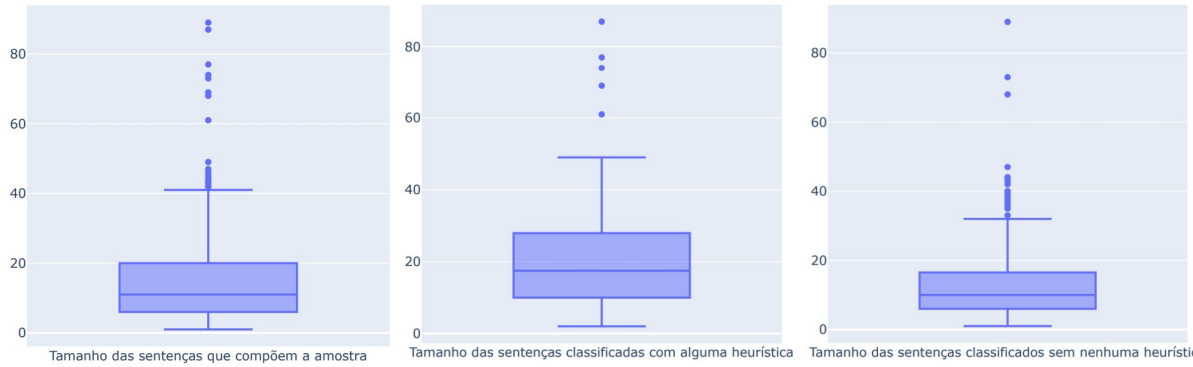
Observando o conjunto de dados classificados, foi possível identificar que o tamanho médio das sentenças classificadas é de 15 palavras, enquanto o desvio padrão é de 12 (Figura 6). Analisando as características das sentenças que foram classificadas com alguma heurística ou não, foi possível identificar que o tamanho médio das sentenças classificadas com alguma heurística é de 21 palavras e desvio padrão 15. Quando se trata das sentenças que não foram classificadas com alguma heurística, a média de tamanho dessas sentenças é de 13 palavras e desvio padrão 11.

Figura 5 – Proporção da ocorrência de heurísticas únicas nas sentenças da amostra



Fonte: elaborada pelo autor.

Figura 6 – Métricas relacionadas com o tamanho das sentenças que compõem a amostra



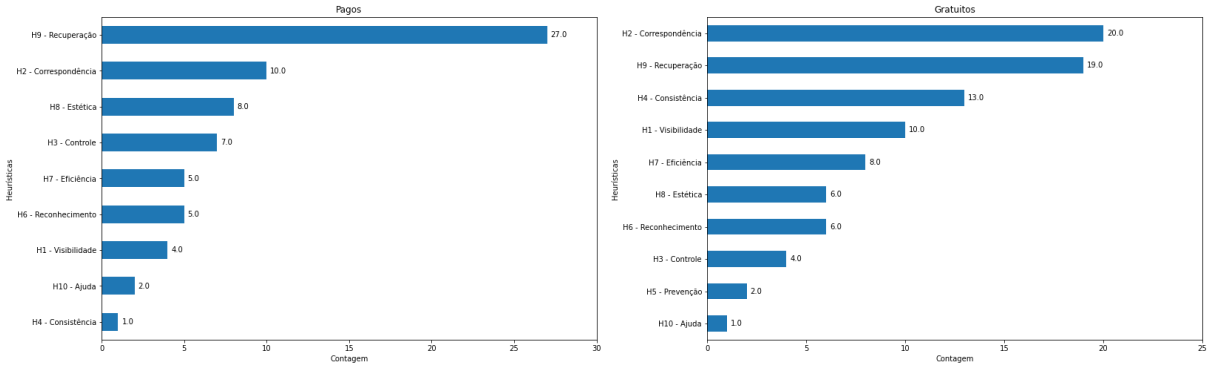
Fonte: elaborada pelo autor.

5.1.3 Relação entre a monetização dos aplicativos e a ocorrência de alguma heurística

Quando se trata da relação entre a forma de monetização do aplicativo, foi possível observar a proporção apresentada na Figura 7. Essa proporção representa apenas o conjunto de sentenças relacionadas com os (a) aplicativos pagos e (b) aplicativos gratuitos.

O conjunto de sentenças de aplicativos pagos é composto por 246 sentenças. Essas sentenças foram classificadas e revelaram uma proporção de 61 (25%) sentenças classificadas com alguma heurística e 185 (75%) sentenças classificadas sem nenhuma heurística. A heurística H9 (ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros) é a mais recorrente nos aplicativos pagos, aparecendo 27 vezes, enquanto a heurística H4 (consistência e padrões) é a que menos ocorre, aparecendo apenas uma vez. A heurística H5 (prevenção de

Figura 7 – Proporção da ocorrência de heurísticas únicas em (a) aplicativos pagos e (b) aplicativos gratuitos na amostra



Fonte: elaborada pelo autor.

erros) não aparece nenhuma vez no conjunto de sentenças dos aplicativos pagos. Também é possível observar que a média de tamanho das sentenças dos aplicativos pagos é de 18 palavras, com um desvio padrão de 13 palavras.

O conjunto de sentenças dos aplicativos gratuitos é composto por 282 sentenças. Essas sentenças revelaram uma proporção de 71(25%) sentenças classificadas com alguma heurística e 211(75%) sentenças classificadas sem nenhuma heurística. A heurística H2 (correspondência entre o sistema e o mundo real) é a mais recorrente (20 vezes) nos aplicativos gratuitos, enquanto a heurística H10 (ajuda e documentação) é a que menos ocorre (uma vez). Também é possível observar que a média de tamanho das sentenças dos aplicativos gratuitos é de 24 palavras, com um desvio padrão de 16 palavras.

5.1.4 Relação entre a categoria dos aplicativos e a ocorrência de alguma heurística

A Tabela 3 apresenta a distribuição das sentenças classificadas de acordo com as categoria dos aplicativos. As categorias e seus aplicativos podem ser vistos na Tabela 1.

Tabela 3 – Distribuição das sentenças pela categoria dos aplicativos na amostra

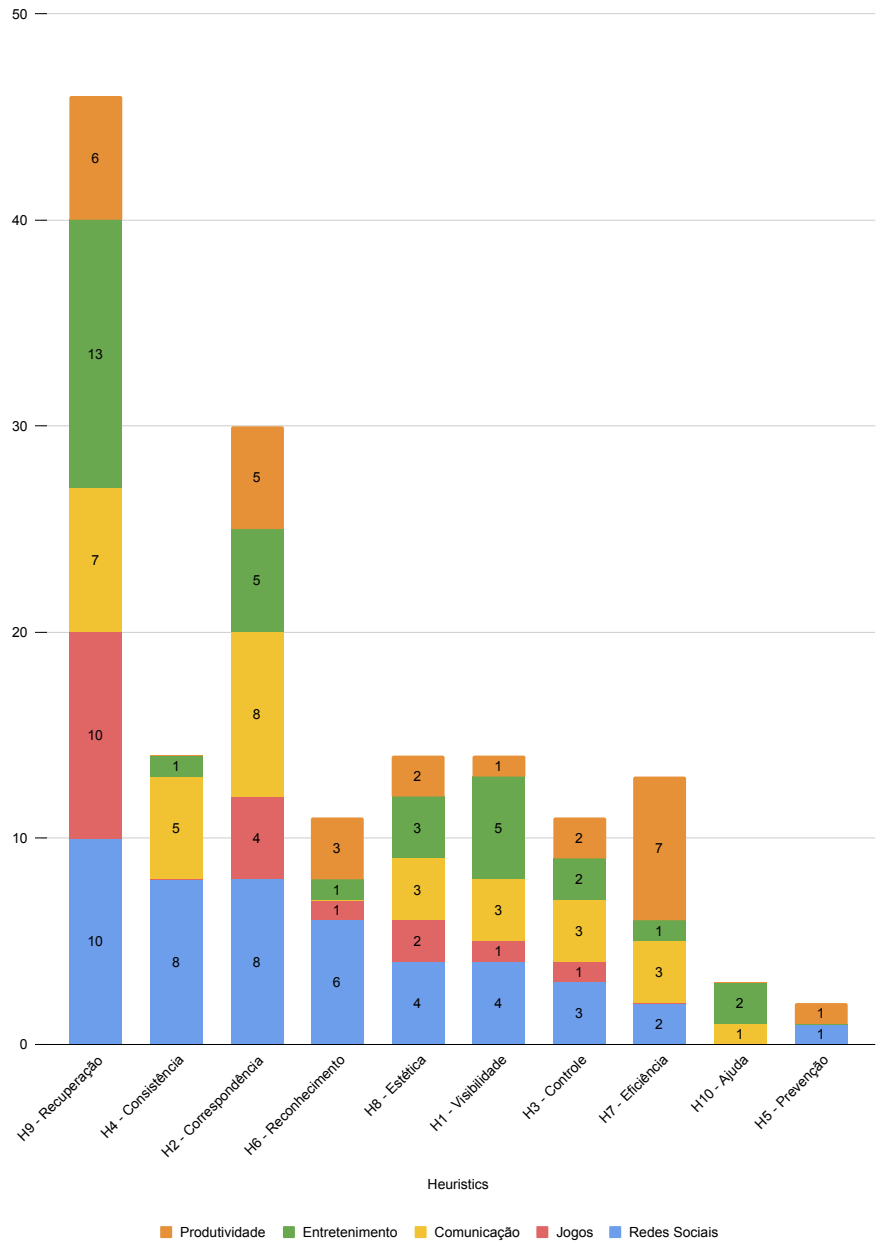
Categorias	Sentenças classificadas
Jogos	125
Produtividade	106
Redes Sociais	105
Entretenimento	102
Comunicação	90

Fonte: elaborada pelo autor.

Considerando as categorias de aplicativos, Jogos é a categoria com mais sentenças

(125), seguida por Produtividade (106), Redes Sociais (105), Entretenimento (102) e Comunicação (90). A Figura 8 apresenta uma visão geral da ocorrência de heurísticas para cada categoria de aplicativos. Analisando especificamente as heurísticas presentes nas sentenças de cada categoria, observa-se que as sentenças em Jogos contêm seis heurísticas, o menor número de heurísticas dentre as categorias de aplicativos.

Figura 8 – Proporção da ocorrência de heurísticas pela categoria dos aplicativos na amostra



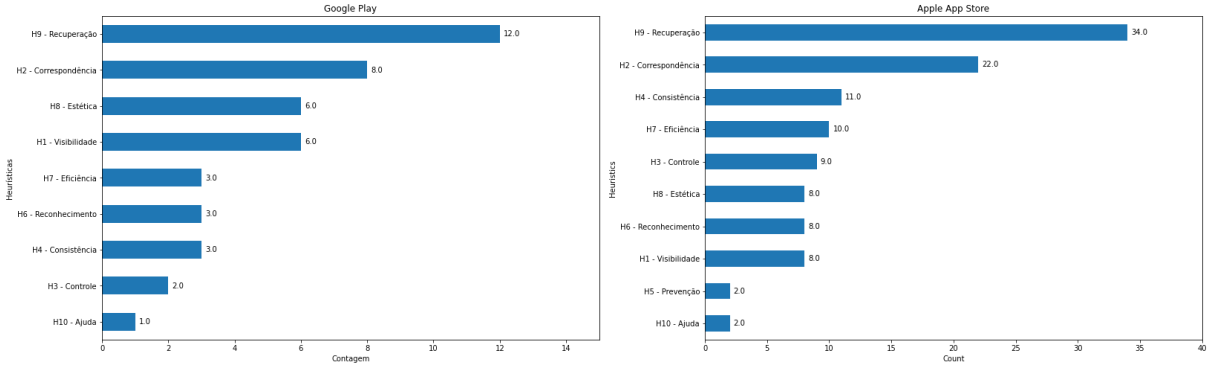
Fonte: elaborada pelo autor.

5.1.5 Relação entre a loja de aplicativos e a ocorrência de alguma heurística

Levando em consideração as lojas de aplicativos para verificar a ocorrência de heurísticas, pode-se dividir o conjunto de dados em duas partes, uma referente às sentenças presentes na *Google Play Store* e outra na *Apple App Store*. No conjunto de dados referente à *Google Play Store* existem 155 sentenças, em que 36 foram classificadas com alguma heurística. Essas sentenças possuem uma média de 18 palavras e um desvio padrão de 16 palavras.

A Figura 9 apresenta a proporção da ocorrência de heurísticas com base nas duas lojas de aplicativos. Pode-se observar que a heurística H9 (ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros) é a que mais aparece (12 vezes) no conjunto de dados da *Google Play Store*, enquanto a heurística H10 (ajuda e documentação) é a que menos aparece (uma vez). Já a heurística H5 (prevenção de erros) não aparece nenhuma vez.

Figura 9 – Proporção da ocorrência de heurísticas nas sentenças da (a) *Google Play Store* e (b) *Apple App Store* na amostra



Fonte: elaborada pelo autor.

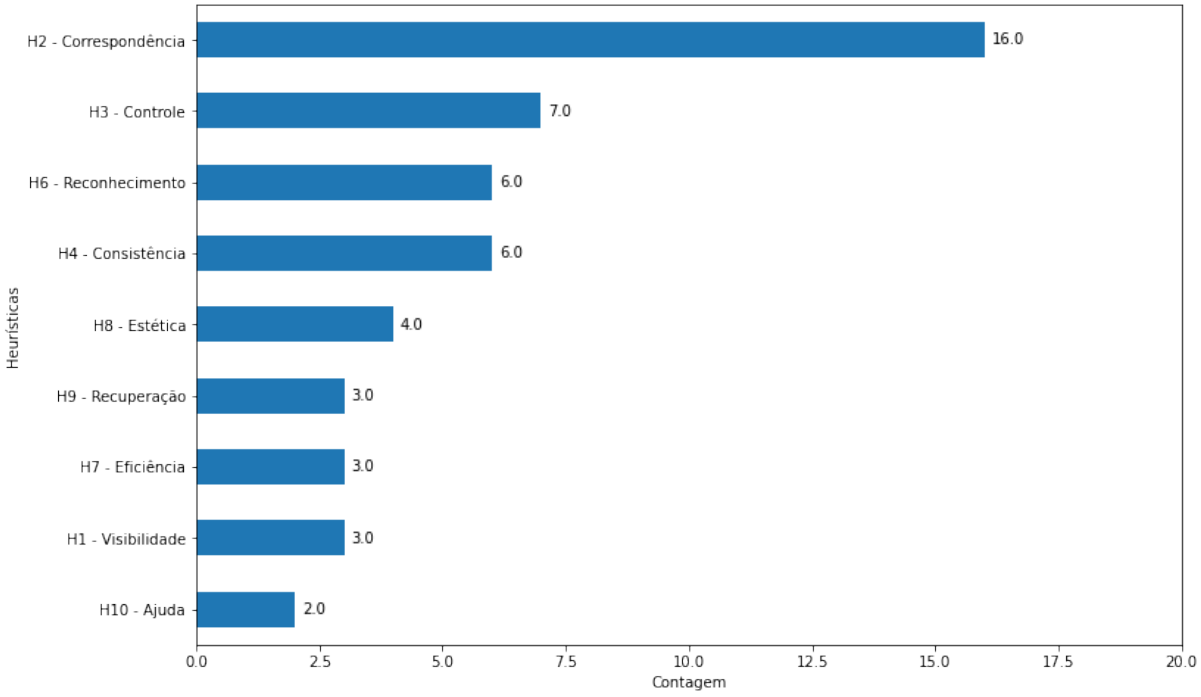
Por outro lado, no conjunto de dados referente à *Apple App Store* existem 373 sentenças, em que 96 foram classificadas com alguma heurística. Nesse conjunto de dados, a heurística H9 é a mais frequente, aparecendo 34 vezes, enquanto a heurística H10 é a que menos aparece (uma vez). Diferente do conjunto de dados discutido no paragráfo anterior, todas as heurísticas ocorrem nesse conjunto de dados.

Esses dados indicam que as revisões do usuário coletadas a partir da *Apple App Store* originam mais sentenças que as revisões do usuário coletadas da *Google Play Store*. Além disso, essa relação de quantidade se mantém, em proporções diferentes, na ocorrência de heurísticas.

5.1.6 Co-ocorrência de heurísticas

Existem 24 sentenças classificadas com mais de uma heurística. Elas possuem uma média de tamanho de 26 palavras e um desvio padrão de 20 palavras. A Figura 10 apresenta a proporção de heurísticas que co-ocorrem. A heurística que mais co-ocorre é a H2 (correspondência entre o sistema e o mundo real), co-ocorrendo 16 vezes, enquanto a que menos co-ocorre é a H10 (ajuda e documentação), co-ocorrendo duas vezes. Também é possível observar que a heurística H5 (prevenção de erros) não co-ocorre com nenhuma outra heurística.

Figura 10 – Proporção das heurísticas que coocorrem na amostra



Fonte: elaborada pelo autor.

A maior combinação na amostra de dados está entre as heurísticas H2 (correspondência entre o sistema e o mundo real) e H3 (controle e liberdade do usuário), ocorrendo 5 vezes. A segunda maior combinação está entre as heurísticas H2 e H4 (consistência e padronização), ocorrendo 4 vezes. Outra análise possível de observar é que a heurística H2 co-ocorre com outras sete heurísticas, menos com a heurística H10 (ajuda e documentação). Também é possível observar que a co-ocorrência mais comum está relacionada com apenas dois tipos de heurísticas, entretanto, apesar de ser raro, também pode existem co-ocorrências entre três tipos de heurísticas. Pode-se observar esse tipo de co-ocorrência duas vezes envolvendo as heurísticas H1 (visibilidade do estado do sistema), H2 (correspondência entre o sistema e o mundo real), H3

(controle e liberdade do usuário), H6 (reconhecimento em vez de memorização) e H8 (projeto estético e minimalista).

5.2 Refinamento dos padrões

Na etapa de refinamento, as métricas coletadas durante a avaliação embasaram esforços para corrigir e melhorar os padrões existentes. Nessa etapa os padrões identificados na etapa 4.8, foram modificados em busca de melhorar os números identificados na etapa de avaliação. O processo de evolução dos padrões foi baseado em uma nova análise de conjunto de dados classificados que foi realizado manualmente, adicionando assim aspectos que estavam faltando nos padrões definidos nas etapas anteriores. Durante essa etapa foi necessário realizar a testagem simultânea dos dados utilizando os padrões e, em paralelo, corrigindo os pontos relevantes dos padrões. Algumas das correções foram relacionadas com a restrição do padrão ou com o aumento da abrangência do padrão. Além disso, para realizar o refinamento dos padrões foi utilizado um conjunto de métricas coletadas na etapa anterior, essa métricas são baseadas em verdadeiros positivos, verdadeiros negativos, falsos positivos, falsos negativos e abrangência dos padrões. Os padrões refinados podem ser vistos no Quadro 5.

Quadro 5 – Listagem dos padrões de linguagem refinados

Heurística	ID	Padrões	Expressões Regulares
H1 - Visibilidade do status do sistema	P1.1	(EN_Questões) ,{0}...(EN_Verbo)	(who|where|why|what|which|when|how) (?!.*,.*).*(happen|find|wait|show|see|watch)
	P1.2	(EN_Oposição) ...(EN_Negação)...(EN_Verbo)	(however|but) .*(not|didn't|don't).*(happen|find|wait|show|see)
	P1.3	(EN_Oposição) ...((EN_Advérbio)|(EN_Preposição))...(EN_Verbo)	(however|but) .*((recently|there|when)|(about).*(happen|wait|find|show|see|watch)
H2 - Relação entre o sistema e o mundo real	P2.1	(EN_Pronome)...(EN_Verbo) ...((its|is)|(EN_Pronome))(|,)	(i|you|he|they).*(think|decides|know|see|seems|seen|want) .*((its|is)|(that|this|them))(|,)
	P2.2	((EN_Verbo_Can)|(was|went)) .*(EN_Verbo)	((can't|cannot|can)|(was|went)) .*(read|write|make|search|show)
H3 - Controle e liberdade do usuário	P3.1	(EN_Pronome) ...(EN_Negação)...want	(i|you|he) .*(not|dont|no).*want
	P3.2	(EN_Verbo_Can_Negado)...((EN_Verbo)|log)	(cant|cannot).*((read|play)|log)
H4 - Consistência e padrões	P4.1	(EN_Pronome) ...sometimes	(i|you|he) .*sometimes
	P4.2	(|^)some... ((EN_Pronome)|(EN_Substantivo))	(|^)some.* ((i|you|he)|(marketplace|iphone))
H6 - Reconhecimento em vez de memorização	P6.1	(EN_Advérbio) ... (EN_Verbo)	(never|clearly) .* (see|appear|dissapear|show|visible)
	P6.2	(EN_Verbo_Auxiliar) ...(EN_Verbo)	(will|was|is|can) .*(see|appear|disappear|visible|find)
H7 - Flexibilidade e eficiência de uso	P7.1	((EN_Advérbio) | is)...((EN_Adjetivo)|frustrating)	((very|extremely)|is).*((hard|difficult|impossible)|frustrating)
	P7.2	(EN_Verbo_Have)... (EN_Representação_Tempo)	(have|has|had).* (time|second)
H8 - Design estético e minimalista	P8.1	(EN_Substantivo)...(EN_Problema)	(layout|interface|ui|ux|app|clothe).*(lag|glitch|broke|garbage)
	P8.2	(EN_Substantivo)...(EN_Negação)?...(EN_Adjetivo)	(layout|interface|ui|ux|app|button).*(not|don't|no)? .*(perfectly|friendlier|cleaner|poor|unnecessary|fresh)
	P8.3	(EN_Adjetivo)...(EN_Substativo)	(perfectly|friendlier|cleaner|poor|unnecessary|garbage).*(layout|interface|ui|ux|app)
H9 - Ajude os usuários a reconhecer, diagnosticar e se recuperar dos erros	P9.1	(EN_Problema)... (EN_Advérbio)	(crash|erro|freeze).* (instantly|a lot|almost|as|completely|same)
	P9.2	crash... (EN_Preposição)	crash.* (after|at|in|on)
	P9.3	keep...crash	keep.*crash
	P9.4	(EN_Substantivo)...(EN_Problema)	(app|program|game|netflix).*(crash|glitch|freeze|bug)
	P9.5	(EN_Advérbio)...crash	(most|constanly).*crash
H10 - Ajuda e documentação	P10.1	(EN_Pronome) ((EN_Verbo) ... help)|((EN_Verbo) ... (EN_Adjetivo)))	(i|you|he) ((find|search) .* help)|((am|are|is) .* (hard|difficult)))
	P10.2	(EN_Pronome) ...(EN_Negação) ...know ...about	(i|you|he) .*(not|don't|no) .*know .*about

Fonte: elaborada pelo autor.

Por fim, ficou evidente que houve uma melhora substancial no conjunto de padrões definidos. As métricas de acurácia, precisão e abrangência em sua maioria melhorou. Essa melhora pode ser observada na Tabela 4.

5.3 Avaliação do conjunto de dados completo

Ao classificar todo o conjunto de dados, foi possível observar alguns detalhes que são discutidos a seguir. Foi possível observar que das 178.371 sentenças, 21.484 foram classificadas com alguma das dez heurísticas de usabilidade de Nielsen. Sendo assim, as demais 156.887 sentenças foram classificadas sem nenhuma heurística. Isso significa que 12% das sentenças foram classificadas com alguma heurística, enquanto as outras 88% foram classificadas sem nenhuma heurística. As próximas subseções apresentam os resultados das análises de acordo com os procedimentos da Seção 4.12.

5.3.1 Proporção da ocorrência das heurísticas

Quando se trata da presença individual de cada heurística atrelada às sentenças classificadas pelos padrões definidos respeitam a seguinte proporção apresentada na Figura 11. Por um lado, percebe-se que a heurística que mais ocorre (6.812 vezes) nas sentenças é a H2 (correspondência entre o sistema e o mundo real). Por outro lado, a heurística que menos ocorre (394 vezes) é a H10 (ajuda), já a heurística H5(prevenção) não aparece nenhuma vez visto que não foi possível identificar padrões para essa heurística.

5.3.2 Relação entre tamanho das sentenças e ocorrência de alguma heurística

Observando o conjunto de dados classificados, foi possível identificar que o tamanho médio das sentenças classificadas é de 15 palavras assim como o desvio padrão 15. Analisando as características das sentenças que foram classificadas com alguma heurística ou não, foi possível identificar que o tamanho médio das sentenças classificadas com alguma heurística é de 32 palavras e desvio padrão 25. Quando se trata das sentenças que não foram classificadas com alguma heurística, a média de tamanho dessas sentenças é de 12 palavras e desvio padrão 11. Essas informações podem ser observadas na Figura 12.

Tabela 4 – Métricas coletadas do conjunto geral de sentenças classificadas utilizando os padrões

ID	Precisão	Acurácia	Abrangência	Abrangência por heurística
P1.1	42%	96%	42%	
P1.2	66,6%	97%	14,2%	71%
P1.3	57%	97%	28%	
P2.1	52,2%	94%	30%	50%
P2.2	66,6%	94%	20%	
P3.1	50%	97%	27%	54%
P3.2	100%	98%	27%	
P4.1	100%	97%	14%	50%
P4.2	60%	97%	42%	
P6.1	33,3%	97%	9%	63,3%
P6.2	54%	98%	54,7%	
P7.1	50%	97%	30%	53%
P7.2	33,3%	96%	23	
P8.1	50%	97%	14%	
P8.2	75%	97%	21%	57%
P8.3	100%	98%	28%	
P9.1	100%	92%	17%	
P9.2	77,7%	92%	17%	
P9.3	80%	91%	8,4%	58%
P9.4	78%	93%	32%	
P9.5	100%	92%	10%	
P10.1	100%	99%	66,6%	100%
P10.2	33,3%	99%	33,3%	

Fonte: elaborada pelo autor.

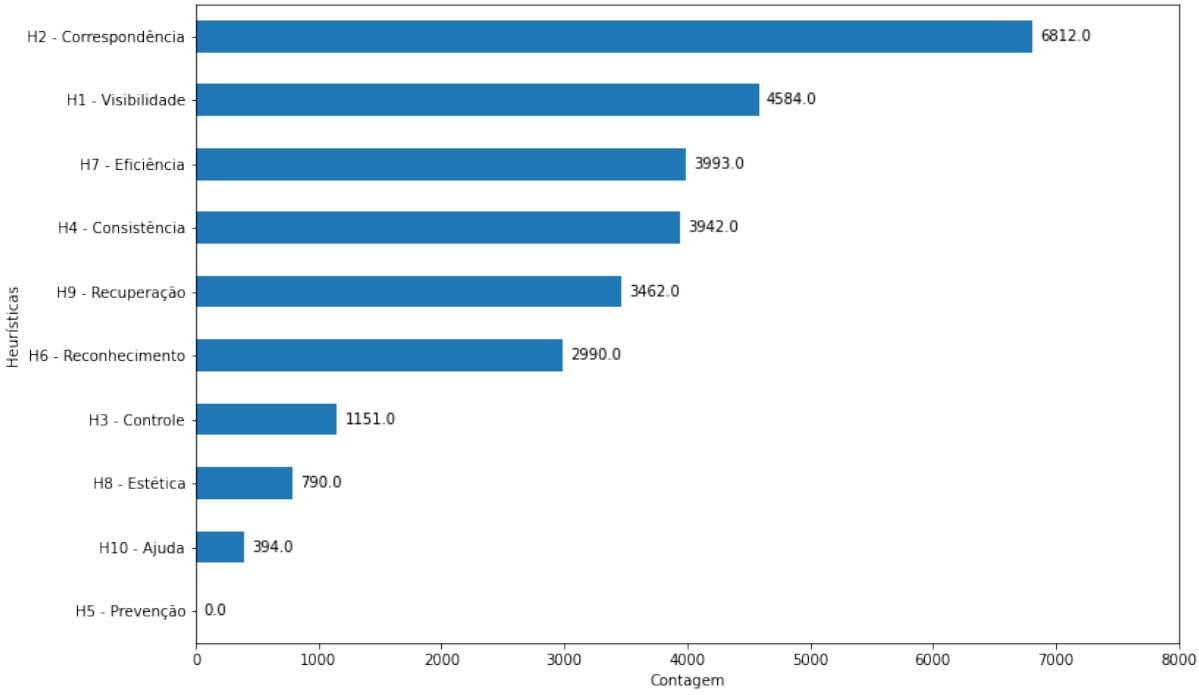
5.3.3 Relação entre a monetização dos aplicativos e a ocorrência de alguma heurística

Quando se trata da relação entre a forma de monetização do aplicativo, foi possível observar a proporção apresentada na Figura 13, essa proporção representa apenas o conjunto de sentenças relacionadas com os (a) aplicativos pagos e (b) aplicativos gratuitos.

Pode-se observar na Figura 13 que o conjunto de sentenças de aplicativos pagos é composto por 57.052 sentenças, em que 5.802 foram classificados com alguma heurística. A heurística Recuperação(H9) é a sentença mais recorrente nos aplicativos pagos, aparece 2.135 vezes, enquanto a sentença Ajuda(H10) é a que menos ocorre, aparecendo 266 vezes. A sentença Prevenção(H5), não aparece nenhuma vez no conjunto de sentenças dos aplicativos pagos. Também é possível observar que a média de tamanho das sentenças dos aplicativos pagos é de 23 palavras, com um desvio padrão de 17 palavras.

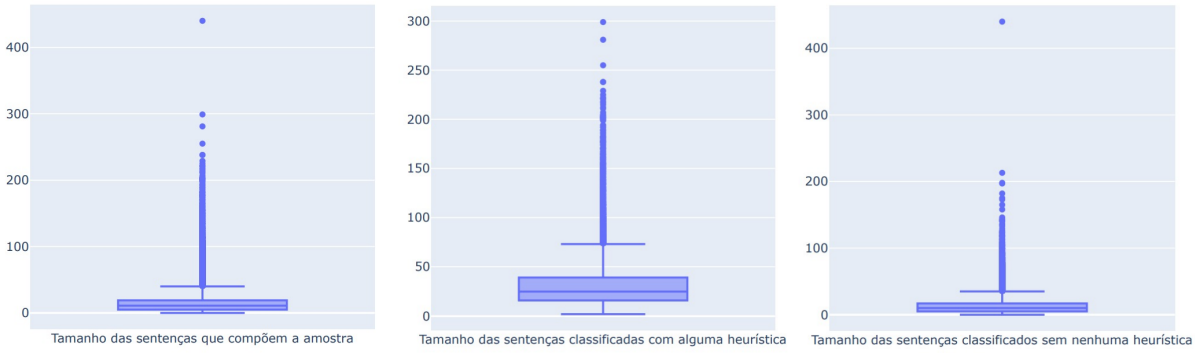
Pode-se observar na Figura 13 que o conjunto de sentenças de aplicativos gratuitos é composto por 121.319 sentenças, em que 15.682 sentenças foram classificadas com alguma

Figura 11 – Porporção de ocorrência das heurísticas no conjunto geral de sentenças classificadas



Fonte: elaborada pelo autor.

Figura 12 – Métricas relacionadas com o tamanho das sentenças que compõe o conjunto geral de sentenças classificadas



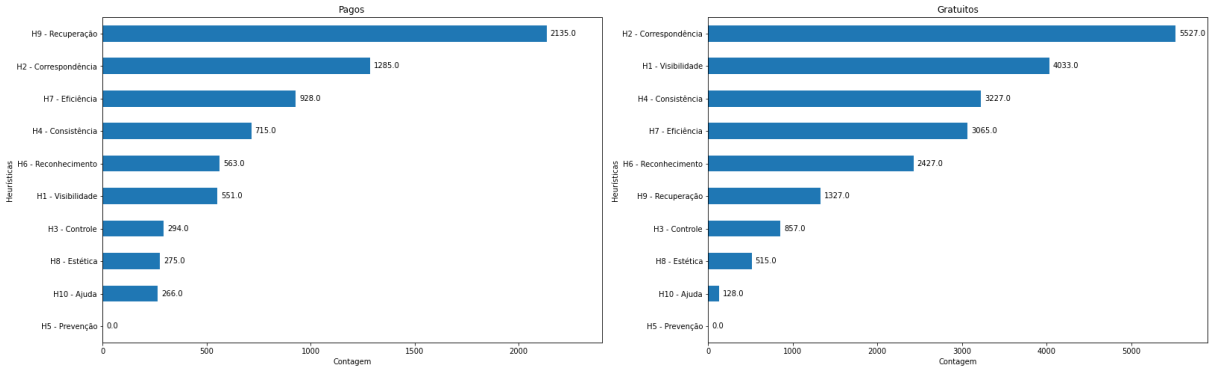
Fonte: elaborada pelo autor.

heurística. A heurística Correspondência(H2) é a mais recorrente nos aplicativos gratuitos, aparecendo 5.527 vezes, enquanto a heurística Ajuda(H10) é a que menos ocorre, aparecendo apenas 128 vezes. Também é possível observar que a média de tamanho das sentenças dos aplicativos pagos é de 35 palavras, com um desvio padrão de 27 palavras.

5.3.4 Relação entre a categoria dos aplicativos e a ocorrência de alguma heurística

Considerando as categorias dos aplicativos que tiveram suas sentenças classificadas, tem-se a seguinte distribuição de sentenças por categoria apresentada na Tabela 5. As categorias

Figura 13 – Proporção de ocorrência de heurísticas na sentenças presentes (a) Google Play Store e (b) Apple App Store no conjunto geral de sentenças classificadas



Fonte: elaborada pelo autor.

e seus respectivos aplicativos podem ser vistos na Tabela 1. Considerando a categoria das sentenças classificadas, a categoria Redes Sociais originou a maior quantidade de sentenças. Já a categoria Produtividade é a que gerou a menor quantidade. Além disso, pode-se observar a proporção de ocorrência das heurísticas na Figura 14.

Tabela 5 – Distribuição de sentenças por categoria no conjunto geral de sentenças classificadas

Categorias	Sentenças classificadas
Redes Sociais	45.279
Entretenimento	44.458
Jogos	42.870
Comunicação	24.025
Produtividade	21.739

Fonte: elaborada pelo autor.

5.3.5 Relação entre a loja de aplicativos e a ocorrência de alguma heurística

Quando se trata de levar em consideração as lojas de aplicativos para verificar a ocorrência de determinada heurística, pode-se dividir o conjunto de dados em duas partes, uma referente às sentenças presentes na Google Play e outra na Apple Store. No conjunto de dados referente à Google Play, tem-se 59.727 sentenças em que 3.753 foram classificadas com alguma heurística. Essas sentenças possuem uma média de 24 palavras e um desvio padrão de 18 palavras.

A Figura 15 apresenta a ocorrência de heurísticas com base nas duas lojas de aplicativos, pode-se observar que a heurística H9(Recuperação) é a que mais aparece (1.084 vezes) no conjunto de dados da *Google Play Store*, enquanto a heurística H10(Ajuda) é a que

menos aparece (129 vezes). Já a heurística H5 (Prevenção) não aparece nenhuma vez, pois não foram definidos padrões para esta heurística.

Por outro lado, no conjunto de dados referentes as sentenças presentes na Apple Store, pode-se observar a proporção apresentada na Figura 15. Aqui existe um conjunto de 118.644 sentenças, em que 17.731 foram classificadas com alguma heurística. Nesse conjunto de dados a heurística Correspondência(H2) é a mais popular, aparecendo 5940 vezes, enquanto a heurística Ajuda(H10) é a que menos aparece, no caso, 265 vezes. Assim como no conjunto discutido no parágrafo anterior, a sentença H5(Prevenção) não aparece nenhuma vez.

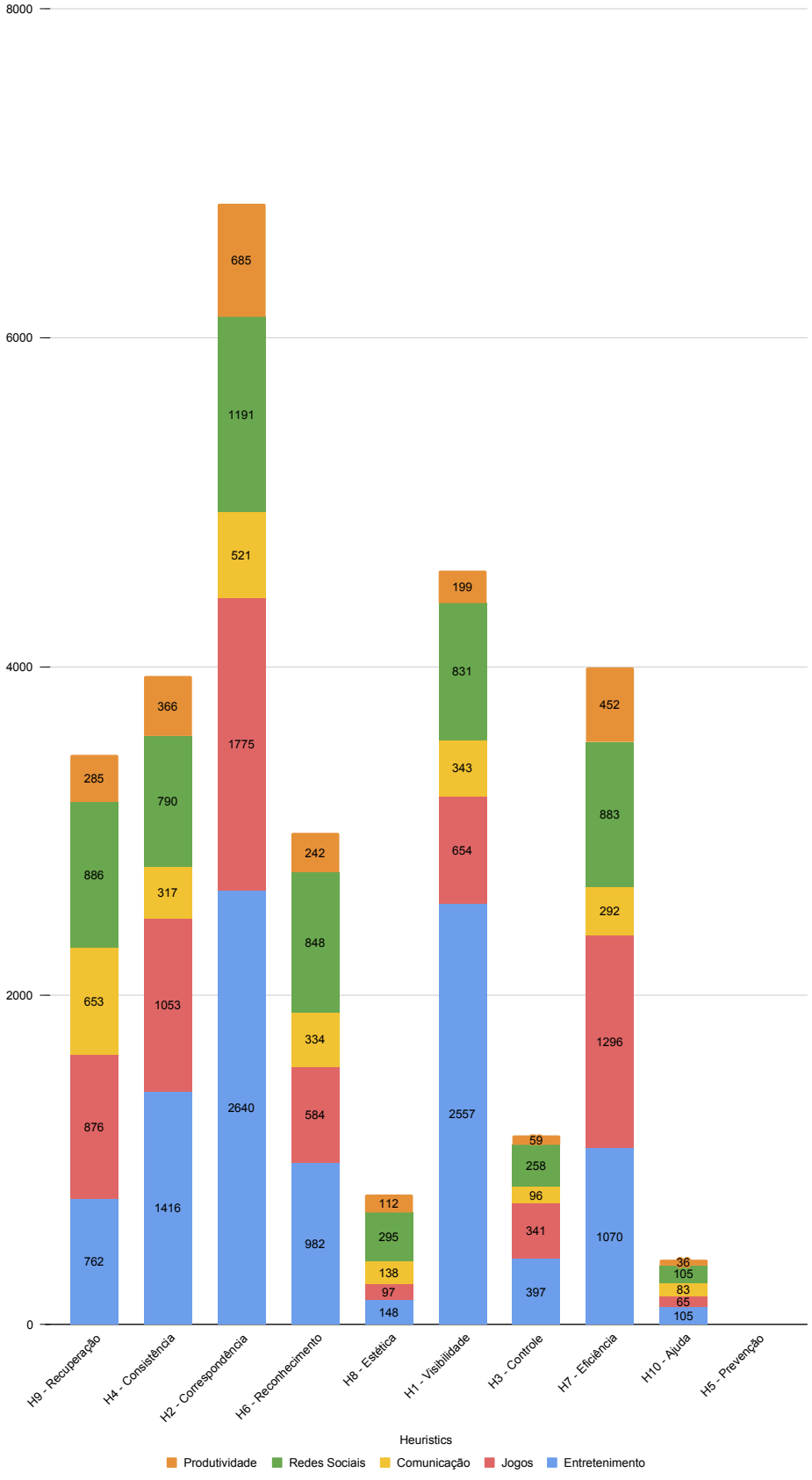
Com esses dados fica evidente que as revisões do usuário coletadas a partir da *Apple App Store* originam mais sentenças que as revisões do usuário coletadas da *Google Play Store*. Além disso, essa relação de quantidade se mantém, em proporções diferentes, durante a ocorrência de heurísticas.

5.3.6 Co-ocorrência de heurísticas

Existem 4.979 sentenças classificadas com mais de uma heurística. Eles têm um comprimento médio de 49 palavras e um desvio padrão de 35 palavras. A Figura 16 apresenta o número de coocorrências para cada heurística. H2 (correspondência entre sistema e mundo real) é a heurística que mais coocorre (2784 vezes), enquanto H8 (Estética) é a heurística que menos coocorre (322 vezes). A heurística H5 (prevenção de erros) não co-ocorreu com nenhuma outra heurística.

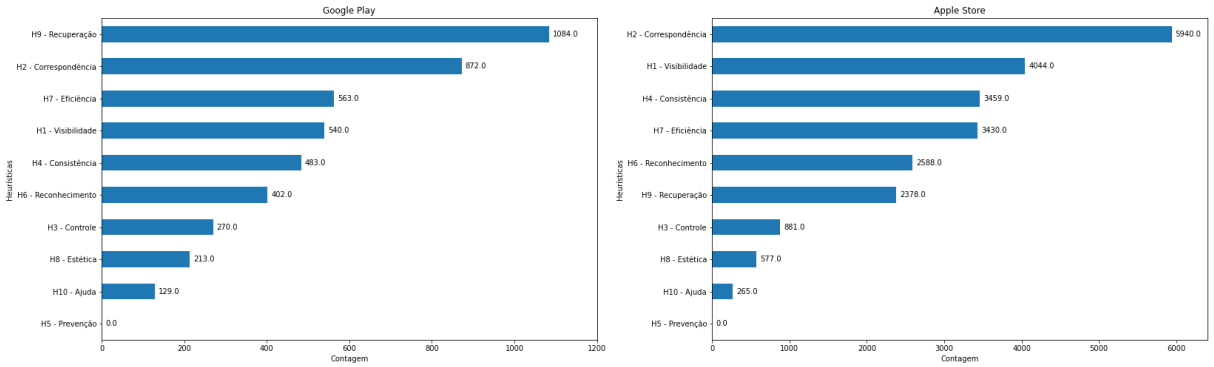
A maior correspondência em todo o conjunto de dados é H1 (visibilidade do status do sistema) e H1 (correspondência entre o sistema e o mundo real), ocorrendo 652 vezes. A segunda é H2 e H6 (reconhecimento ao invés de memorização), ocorrendo 496 vezes. Adicionalmente, foi possível observar que a coocorrência comum envolve apenas dois tipos de heurísticas. No entanto, também pode haver uma co-ocorrência entre três ou mais tipos de heurísticas.

Figura 14 – Proporção de ocorrência de heurísticas na sentenças presentes por categoria no conjunto de dados geral de sentenças classificadas



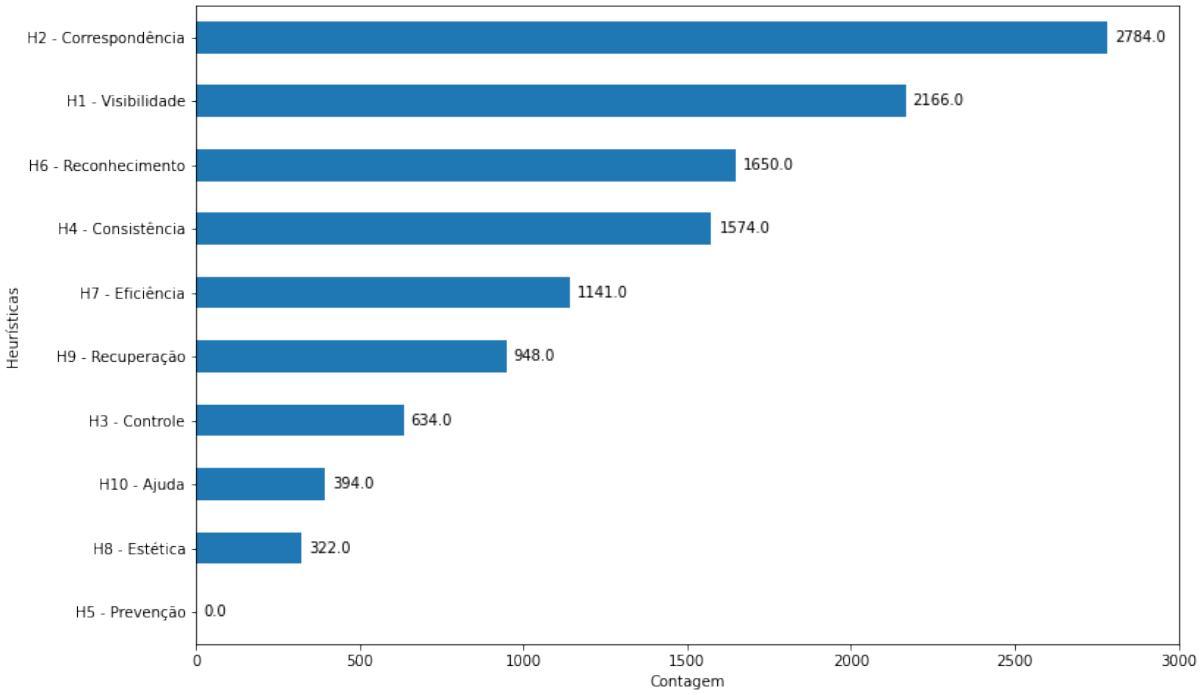
Fonte: elaborada pelo autor.

Figura 15 – Proporção de ocorrência de heurísticas na sentenças presentes (a) Google Play Store e (b) Apple App Store no conjunto geral de sentenças classificadas



Fonte: elaborada pelo autor.

Figura 16 – Proporção de coocorrência entre heurísticas no conjunto geral de sentenças classificadas



Fonte: elaborada pelo autor.

6 DISCUSSÃO

Neste capítulo, são apresentadas discussões oriundas dos resultados apresentados no capítulo anterior.

6.1 Classificação manual das revisões do usuário

Os resultados revelaram que apesar do pequeno número de sentenças classificadas (25%), as avaliações de usuários têm um certo potencial para indicar problemas relacionados às heurísticas de usabilidade da Nielsen. As heurísticas mais recorrentes foram a H9 (ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros) e a H2 (correspondência entre o sistema e o mundo real). Isso indica que os usuários relatam principalmente defeitos, falhas e dificuldades em entender como o aplicativo funciona de acordo com sua visão de mundo real. No entanto, a heurística de menor ocorrência foi a H5 (prevenção de erros), que trata de estratégias de design de interação para evitar que o usuário cometa um erro. Identificar essa heurística nas declarações foi um desafio, visto que tais problemas seriam simples de identificar ao analisar as interfaces do aplicativo, mas os usuários não elaboram esse tipo de informação em suas revisões. Esta é uma limitação em comparação ao uso de heurísticas em inspeções: não é possível determinar o local de ocorrência para todos os problemas relatados, já que isso depende da espontaneidade dos usuários ao escrever suas revisões.

Jogos foi a categoria de aplicativos com o menor número de heurísticas distintas identificadas. Isso pode estar relacionado ao fato de que na maioria dos depoimentos os usuários relataram insatisfação com as regras de negócio do jogo ao invés de descreverem problemas que impactaram seu uso, tais como: "*The first reason is that the chances of getting a new brawler is NEARLY impossible like the power points and coins are easy to get but somehow, the Big box and the mega box percentages are just the same*". Nesses casos, optou-se por classificar a sentença como sendo sem heurística.

No geral, essa classificação manual é desafiadora porque os usuários geralmente não elaboram o contexto de uso para que as informações implícitas possam ser claramente compreendidas. Assim, abordagens que utilizam análise semântica e/ou de sentimento (por exemplo, SentiStrength-SE (ISLAM; ZIBRAN, 2018)) podem trazer vantagens para a identificação de heurísticas em pesquisas futuras.

Foi possível identificar potencial, ainda que pequeno, para automização. Por exemplo,

para a heurística H1 (visibilidade do estado do sistema), trechos como "*I don't understand...*", "*I can't see...*", "*where is it..*" geralmente aparecem, expressando a ideia de falta de feedback ou compreensão. Essa possível automatização deve primeiro considerar o tamanho das sentenças. Conforme relatado nos resultados, as declarações com problemas relacionados às heurísticas de usabilidade costumam ter um número maior de palavras (21 em média), indicando que o usuário frequentemente detalha mais informações ao relatar, embora não saiba, um problema de heurística. Ferramentas automatizadas podem ser construídas para viabilizar essa tarefa de classificação a partir da identificação de padrões de linguagem em avaliações de usuários, como realizado no estudo de (GROEN *et al.*, 2017) ou também empregando técnicas de aprendizado de máquina.

Com base nos resultados, percebe-se que a avaliação textual baseada em heurísticas de usabilidade poderia ser utilizada como uma técnica de apoio ao processo de avaliação de usabilidade, ao invés de um método independente. Além disso, a aplicação dessa estratégia pode ajudar a identificar as partes mais críticas do sistema da perspectiva de usuários reais. Essas partes podem ser avaliadas posteriormente com outros métodos, como métodos de inspeção (*e.g.*, Avaliação Heurística (NIELSEN; MOLICH, 1990)), e aqueles que envolvem diretamente a participação do usuário (*e.g.*, teste do usuário e diário do usuário).

6.2 Definição dos padrões de linguagem

Além da discussão relacionada com a classificação manual, é possível observar os resultados da classificação utilizando os padrões definidos durante a realização desse trabalho. É possível observar que as proporções de sentenças classificadas e não classificadas é consideravelmente diferente quando a classificação da amostra de dados é feita de forma manual e quando a classificação do conjunto de dados completo é feita utilizando os padrões. Isso acontece, principalmente, pelo fato de que não é possível definir padrões de linguagem para todas as sentenças classificadas de acordo com determinado grupo. O presente trabalho atingiu uma abrangência de 62% de sentenças classificadas, ou seja, os padrões definidos conseguem identificar 62% das sentenças que foram classificadas com alguma heurística. Dessa forma, algumas heurísticas não tiveram suas sentenças classificadas, afetando assim a proporção de ocorrência de heurísticas por sentença. Entretanto, nas duas classificações realizadas, é fato que existem mais sentenças classificadas sem nenhuma heurística em ambas.

A análise do conjunto de dados referente a heurística H5 (prevenção de erros) não

permitiu a identificação de nenhum padrão de linguagem visto que foi a heurística menos presente na classificação executada de forma manual na amostra selecionada.

No geral, a definição dos padrões de linguagem é uma tarefa árdua e exaustiva, visto que necessita que seja realizada uma análise qualitativa sobre os dados classificados identificando nuances relevates para aquela classificação. Dessa forma, utilizar ferramentas relacionadas com o processamento de linguagem natural pode atuar como um grande aliado para identificar termos em comum, termos que possuem um forte relacionamento e, com base nisso, é necessário relacionar essas informações com a ocorrência de alguma heurística.

As métricas coletadas após a utilização dos padrões de linguagem para classificar a amostra revelaram que apesar dos padrões terem uma precisão imprevisível, no pior dos casos é 33%, a acurácia tende a ser adequada. Isso significa que, no pior dos casos, pode existir uma proporção de dois terços das classificações de serem um falso positivo. Por outro lado, os padrões são adequados para classificar as sentenças, visto que eles acertam no mínimo 91% das vezes, tendo em vista a precisão dos padrões, é possível reduzir drasticamente a quantidade de sentenças para serem analisadas dentro de um conjunto de dados.

Quando se trata da utilização dos padrões de linguagem, a tarefa se torna mais simples, é fácilmente possível utilizar os padrões de linguagem em conjunto com alguma técnica de representação de padrões (no caso desse trabalho, as expressões regulares) e, posteriormente, utilizar esses padrões para classificar as sentenças utilizando alguma ferramenta (no caso desse trabalho, a linguagem de programação *Python*). Dessa forma, a construção de uma ferramenta que classifique as revisões do usuário e que seja amigável para o usuário é uma possibilidade.

Com base nisso, utilizar os padrões de linguagem é uma opção viável para aproveitar os feedbacks do usuário que estão inseridos no ambiente de produção do software que podem indicar problemas de usabilidade e apoiar o processo de Avaliação Heurística e de garantia de qualidade do produto de software.

7 CONSIDERAÇÕES FINAIS

Através deste trabalho, observou-se que as revisões de usuários têm um certo potencial para indicar problemas de usabilidade através das heurísticas de usabilidade de Nielsen. No entanto, classificar manualmente as avaliações de usuários requer conhecimento técnico e muito esforço por parte dos avaliadores. Dessa forma, um dos resultados gerais desse trabalho foi identificar a extensão com que as revisões do usuário indicam violações das heurísticas de Nielsen.

Além disso, foi possível identificar padrões de linguagem que capturam sentenças que possuem em seu conteúdo alguma menção às heurísticas de usabilidade de Nielsen. Esses padrões abrem caminho para automatizar o processo de classificação dessas sentenças, dessa forma, minimizando a problemática atrelada à complexidade de realizar a classificação manual. Por fim, os resultados afirmam que essa classificação automática é uma opção viável para contribuir no processo de pesquisa e de garantia de qualidade de *software*.

Como trabalhos futuros pretende-se construir uma ferramenta amigável para profissionais de software e que utilize os padrões de linguagem definidos no presente trabalho, o que é algo possível e benéfico para o processo de garantia de qualidade. Essa ferramenta tem o potencial de estar presente no dia a dia do desenvolvimento de *software* ágil. Além disso, aplicar essa ferramenta no processo de desenvolvimento ágil de um *software* é imprescindível para identificar os benefícios e malefícios do uso dessa ferramenta como fator determinante para a garantia de qualidade. A aplicação dessa ferramenta em um ambiente real de desenvolvimento de *software* é importante para que seja possível coletar métricas e, consequentemente, avaliar essas métricas.

REFERÊNCIAS

BARBOSA, S. D. J.; DA SILVA, B. S. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.

CARREÑO, L. V. G.; WINBLADH, K. Analysis of user comments: An approach for software requirements evolution. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 35., 2013, São Francisco. **Proceedings of the 35th International Conference on Software Engineering**. Piscataway: IEEE, 2013. p. 582–591.

CHAPMAN, C.; WANG, P.; STOLEE, K. T. Exploring regular expression comprehension. In: INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING, 32., 2017, Urbana. **Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering**. Piscataway: IEEE, 2017. p. 405–416.

CHRISTIANSEN, T.; FOY, B. D.; WALL, L.; ORWANT, J. **Programming Perl**: Unmatched power for text processing and scripting. 4. ed. Sebastopol: O'Reilly Media, Inc., 2012.

DABROWSKI, J.; LETIER, E.; PERINI, A.; SUSI, A. App review analysis for software engineering: A systematic literature review. **University College London, Tech. Rep**, [*S. l.*], 2020.

DOWNEY, A. B. **Think Stats**: Exploratory data analysis in Python. 2. ed. Sebastopol: O'Reilly Media, Inc., 2014.

FRIEDL, J. E. F. **Mastering regular expressions**: Understand your data and be more productive. 3. ed. Sebastopol: O'Reilly Media, Inc., 2006.

GEBAUER, J.; TANG, Y.; BAIMAI, C. User requirements of mobile technology: results from a content analysis of user reviews. **Information Systems and e-Business Management**, Springer, Heidelberg, v. 6, n. 4, p. 361–384, 2008.

GROEN, E. C.; KOPCZYŃSKA, S.; HAUER, M. P.; KRAFFT, T. D.; DOERR, J. Users — The hidden software product quality experts?: A study on how app users report quality aspects in online reviews. In: INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 25., 2017, Lisboa. **Proceedings of the 25th International Requirements Engineering Conference**. Piscataway: IEEE, 2017. p. 80–89.

GUZMAN, E.; EL-HALIBY, M.; BRUEGGE, B. Ensemble methods for app review classification: An approach for software evolution (n). In: INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING, 30., 2016, Lincoln. **Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering**. Piscataway: IEEE, 2015. p. 771–776.

HEDEGAARD, S.; SIMONSEN, J. G. Extracting usability and user experience information from online user reviews. In: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 31., 2013, Paris. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. Nova Iorque: ACM, 2013. p. 2089–2098.

HENNIG-THURAU, T.; GWINNER, K. P.; WALSH, G.; GREMLER, D. D. Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the internet? **Journal of interactive marketing**, Elsevier, Amsterdã, v. 18, n. 1, p. 38–52, 2004.

ISLAM, M. R.; ZIBRAN, M. F. SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text. **Journal of Systems and Software**, Elsevier, Amsterdã, v. 145, p. 125–146, 2018. Disponível em: https://www.sciencedirect.com/science/article/pii/S0164121218301675. Acesso em: 15 fev. 2022.

ISO 9241-210. **ISO 9241-210:** Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. ISO, Genebra, 2019.

ISO/IEC 25010. **ISO/IEC 25010:** Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. ISO/IEC, Genebra, 2011.

LU, M.; LIANG, P. Automatic classification of non-functional requirements from augmented app user reviews. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 21., 2017, Karlskrona. **Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering**. Nova Iorque: ACM, 2017. p. 344–353.

MAALEJ, W.; KURTANOVIĆ, Z.; NABIL, H.; STANIK, C. On the automatic classification of app reviews. **Requirements Engineering**, Springer, Londres, v. 21, n. 3, p. 311–331, 2016.

MENDES, M. S. **MALTU**: Um modelo para avaliação da interação em sistemas sociais a partir da linguagem textual do usuário. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Pós-Graduação em Ciência da Computação, Fortaleza, 2015.

MENEZES, P. F. B. **Linguagens formais e autômatos**. 2. ed. Porto Alegre: Sagra Luzzatto, 1998.

NIELSEN, J. **10 Usability Heuristics for User Interface Design**. Fremont: Nielsen Norman Group: UX Training, Consulting, & Research, 1994. Disponível em: https://www.nngroup.com/articles/ten-usability-heuristics/. Acesso em: 15 ago. 2021.

NIELSEN, J. **Usability engineering**. São Francisco: Morgan Kaufmann, 1994.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 8., 1990, Seattle. **Proceedings of the SIGCHI Conference on Human factors in computing systems**. Nova Iorque: ACM, 1990. p. 249–256.

NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: a roadmap. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 22., 2000, Limerick. **Proceedings of the Conference on the Future of Software Engineering**. Nova Iorque: ACM, 2000. p. 35–46.

PAGANO, D.; MAALEJ, W. User feedback in the appstore: An empirical study. In: INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, 21., 2013, Rio de Janeiro. **Proceedings of the 21st IEEE International Requirements Engineering Conference**. Piscataway: IEEE, 2013. p. 125–134.

POHL, K. **Requirements Engineering**: Fundamentals, principles, and techniques. 1. ed. Berlim: Springer, 2010.

SCHNEIDER, K. Focusing spontaneous feedback to support system evolution. In: INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, 19., 2011, Trento. **Proceedings of the IEEE 19th International Requirements Engineering Conference**. Piscataway: IEEE, 2011. p. 165–174.

WIEGERS, K. E.; BEATTY, J. **Software Requirements**. 3. ed. Redmond: Microsoft Press, 2013.

APÊNDICE A – PRIMEIRA VERSÃO DOS PADRÕES DE LINGUAGEM

Quadro 6 – Listagem dos padrões de linguagem definidos

Heurística	ID	Padrões	Expressões Regulares
H1 - Visibilidade	P1.1	(EN_Questões)...(EN_Verbo)	(who|where|why|what|which|when|how).* (happen|find|wait|show|see)
	P1.2	(EN_Oposição)...(EN_Negação)	(however|but).*(not|didn't|don't)
	P1.3	(EN_Oposição)((EN_Advérbios)/(EN_Preposição))	(however|but).*((recently|there|when)|(about))
H2 - Correspondência	P2.1	(EN_Pronome)...(EN_Negação)? (EN_Verbo)	(i|you|he).*(not|don't|no)? (think|surprise|decides|know|seems|want)
H3 - Controle	P3.1	(EN_Pronome)...(EN_Negação)want	(i|you|he).*(not|dont|no).*want
	P3.2	want(EN_Negação)	want.*(no|dont|nope)
H4 - Consistência	P4.1	(EN_Pronome)some...	(i|you|he).*some.*
	P4.2	some...(EN_Pronome)	some.*(i|you|he)
H6 - Reconhecimento	P6.1	(EN_Advérbio)...(EN_Verbo)	(never|clearly)...(see|appear|disappear|show|visible)
	P6.2	(EN_Verbo_Auxiliar)...(EN_Verbo)	(will|was|is|can).*(see|appear|disappear|show|visible)
H7 - Eficiência	P7.1	(EN_Advérbio)...(EN_Adjetivo)	(very|much|too|lot).* (hard|difficult|impossible|frustrating)
H8 - Estética	P8.1	(EN_Substantivo)...(EN_Problema)	(layout|interface|ui|ux|app).*(lag|glitch|broke)
	P8.2	(EN_Substantivo)...(EN_Negação)?(EN_Adjetivo)	(layout|interface|ui|ux|app).*(no|don't|no)?.* (perfectly|friendlier|cleaner|poor|unnecessary)
	P8.3	(EN_Adjetivo)...(EN_Substantivo)	(perfectly|friendlier|cleaner|poor|unnecessary).* (layout|interface|ui|ux|app)
H9 - Recuperação	P9.1	crash...(EN_Advérbio)	crash.*(instantly|a lot|almost|as)
	P9.2	crash...(EN_Preposição)	crash.*(after|at|in|on)
	P9.3	keep...crash	keep.*crash
	P9.4	crash...most	crash.*most
	P9.5	(EN_Advérbio)...crash	(most|constantly).*crash
H10 - Ajuda	P10.1	(EN_Pronome) ((EN_Verbo)...help)|((EN_Verbo)...(EN_Adjetivo)))	(i|you|it) ((find|search).*help)|((am|are|is).* (hard|difficult)))
	P10.2	(EN_Pronome)...(EN_Negação)...know...about	(i|you|he).*(not|don't|no).*know.*about

Fonte: elaborado pelo autor.

Tabela 6 – Métricas coletadas da amostra classificada utilizando os padrões

ID	Precisão	Acurácia	Abrangência	Abrangência por heurística
P1.1	27,7%	95%	35,7%	
P1.2	26,6%	96%	28,5%	64,2%
P1.3	22,2%	95%	28,5%	
P2.1	22%	89%	33,3%	33,3%
P3.1	37,5%	97%	27%	36%
P3.2	20%	97%	9%	
P4.1	13,3%	94%	21,4%	42,8%
P4.2	29,4%	96%	35,7%	
P6.1	33,3%	97%	9%	54,5%
P6.2	29,4%	96%	45,4%	
P7.1	50%	97%	7,6%	7,6%
P8.1	100%	97%	7,1%	
P8.2	100%	97%	7,1%	21,4%
P8.3	100%	97%	14,2%	
P9.1	100%	92%	10,8%	
P9.2	77,7%	92%	15,2%	
P9.3	100%	91%	6,5%	26%
P9.4	100%	91%	4,3%	
P9.5	100%	91%	4,3%	
P10.1	100%	99%	66,6%	100%
P10.2	25%	99%	33,3%	

Fonte: elaborado pelo autor.

APÊNDICE B – *SCRIPTS* PARA A EXTRAÇÃO DAS REVISÕES DO USUÁRIO

Código-fonte 1 – *Script* para coleta de dados da *Apple App Store*

```
from app_store_scraper import AppStore
import pandas as pd

result = AppStore(country="us", app_name="appName")
result.review(5000)

dataframe = pd.DataFrame(result.reviews)
dataframe = df[['review', 'rating']]
dataframe.columns = ['content', 'score']
dataframe['app'] = 'appName'
dataframe['store'] = 'store'
dataframe.to_csv('datasetName.csv', index=False)
```

Código-fonte 2 – *Script* para coleta de dados da *Google Play Store*

```
from google_play_scraper import reviews
import pandas as pd

result, continuation_token = reviews('appId', lang='us',
   country='us', count = 5000)

dataframe = pd.DataFrame(result)
dataframe = dataframe[['content', 'score']]
dataframe['app'] = 'appName'
dataframe['store'] = 'store'
dataframe.to_csv('datasetName.csv', index=False)
```

APÊNDICE C – *SCRIPT* PARA A SEPARAÇÃO DAS REVISÕES DO USUÁRIO

Código-fonte 3 – *Script* para separar as revisões do usuário em sentenças

```
1 import pandas as pd
2 import re
3
4 dataset = pd.read_csv("amostra.csv")
5
6 sentencas = [re.split(r"\. |\.\n", row) for row in dataset[
    'content']]
```