UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

LARISSA PEREIRA DE SOUSA

CATÁLOGO DO REQUISITO NÃO-FUNCIONAL JUSTIÇA PARA APLICAÇÕES DE APRENDIZADO DE MÁQUINA

QUIXADÁ

2023

LARISSA PEREIRA DE SOUSA

CATÁLOGO DO REQUISITO NÃO-FUNCIONAL JUSTIÇA PARA APLICAÇÕES DE APRENDIZADO DE MÁQUINA

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

QUIXADÁ

2023

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

S697c Sousa, Larissa Pereira de.
Catálogo do Requisito Não-Funcional Justiça para Aplicações de Aprendizado de Máquina / Larissa Pereira de Sousa. – 2023.
48 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2023.
Orientação: Profa. Dra. Rainara Maia Carvalho.

1. aprendizado de máquina. 2. requisito não-funcional. 3. catálogo. I. Título.

CDD 005.1

LARISSA PEREIRA DE SOUSA

CATÁLOGO DO REQUISITO NÃO-FUNCIONAL JUSTIÇA PARA APLICAÇÕES DE APRENDIZADO DE MÁQUINA

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: 11 de Dezembro de 2023

BANCA EXAMINADORA

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

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

Prof. Dr. Jefferson de Carvalho Silva
Universidade Federal do Ceará (UFC)

A Deus, por ser meu alicerce e nunca me desamparar. Aos meus pais, pela dedicação e paciência em me educar com muito amor e empatia os valores da vida, e sempre acreditarem em mim.

AGRADECIMENTOS

A minha orientadora, Profª. Dra. Rainara Maia Carvalho, pela paciência, apoio e competência durante as aulas e pelo direcionamento no desenvolvimento do trabalho.

Aos professores participantes da banca examinadora Profª. Dra. Lívia Almada Cruz e Prof. Dr. Jefferson de Carvalho Silva pelo tempo, pelas valiosas sugestões e colaborações.

Aos meus pais, pelo incentivo, apoio, orientação, dedicação e amor sem medidas.

A minha avó paterna, minha mamãe Lúcia, pelos conselhos, zelo e por sempre estar ao meu lado.

Ao meu irmão Arthur, por me tornar uma pessoa melhor, mais responsável e protetora.

A minha tia Maria Madalena e meu tio Werton, por cuidarem de mim e me apoiarem em tudo.

Ao meu namorado Bruno por me encorajar a seguir meus sonhos.

As minhas amigas do apartamento 306, Vitória e Taynan, por me acolherem durante minha estadia e partilharem dessa jornada comigo. A minha companheira do apartamento 102, Kilvia, pelo carinho e cuidado.

Aos amigos que fiz durante a graduação, Klyvia, Elenilson, Maria Eduarda, Júnior, Eandro e Caio, pelas risadas, suporte e por dividirem suas histórias comigo. A todas as pessoas que não foram citadas, mas que contribuíram imensamente para que este trabalho e a graduação fossem concluídos.

"Comece fazendo o que é necessário, depois o que é possível, e de repente você estará fazendo o impossível."

(São Francisco de Assis)

RESUMO

Aprendizado de máquina (AM) é um ramo da inteligência artificial, onde um computador tem a habilidade de aprender com a experiência, utilizando dados para o seu aprendizado. É uma área que tem chamado bastante atenção, contudo garantir sistemas habilitados para AM seguros, justos, transparentes e precisos é um desafio. Essas são algumas características de qualidade importantes para AM. Características de qualidade são um tipo de requisito não-funcional (RNF), que são importantes para o comportamento esperado e a qualidade dos sistemas. Justiça é um desses RNFs, pois quando Justiça não é considerada uma prioridade, resulta em sistemas com comportamento inesperado e tendencioso. Considerando o Gráfico de Interdependência de *Softgoal* (SIG), que é um formato conhecido para catalogar RNFs, este trabalho tem como objetivo capturar subcaracterísticas para Justiça e catalogá-las em um SIG. Foi utilizada uma abordagem já existente para catalogar o RNF, denominada ARRANGE, usando método de pesquisa bem definido: *grounded theory*. Como resultado final, foi obtido um SIG de Justiça composto por duas subcategorias principais e nove subcaracterísticas. Dessa forma, os engenheiros de software, sempre que precisarem, podem e devem consultar esse catálogo para os ajudar a tomar decisões em relação ao uso e aplicação desse RNF, para sistemas de AM que consideram Justiça como um RNF.

Palavras-chave: aprendizado de máquina; requisito não-funcional; catálogo.

ABSTRACT

Machine learning (ML) is a branch of artificial intelligence, where a computer has the ability to learn from experience, using data for its learning. It is an area that has attracted a lot of attention, however ensuring safe, fair, transparent and accurate ML-enabled systems is a challenge. These are some important quality characteristics for ML. Quality characteristics are a type of non-functional requirement (NFR), that are important for the expected behavior and quality of systems. Fairness is one of these NFRs, because when Fairness is not considered a priority, it results in systems with unexpected and biased behavior. Considering the Softgoal Interdependence Graph (SIG), which is a known format for cataloging NFRs, this work aims to capture sub-characteristics for Fairness and catalog them in a SIG. An existing approach was used to catalog the NFR, called ARRANGE, using a well-defined research method: grounded theory. As a final result, a Justice SIG was obtained, consisting of two main subcategories and nine subcharacteristics. Therefore, software engineers, whenever they need to, can and should consult this catalog to help them make decisions regarding the use and application of this NFR, for ML systems that consider Fairness as an NFR.

Keywords: machine learning; non-functional requirement; catalog.

LISTA DE FIGURAS

Figura 1 – Fluxo de desenvolvimento de aprendizado de máquina. 19
Figura 2 – Exemplo de catálogo de subcaracterísticas. 22
Figura 3 – Exemplo de um SIG. 23
Figura 4 – Visão geral do processo CORRELATE. 23
Figura 5 – Passo 2 do processo CORRELATE. 24
Figura 6 – Visão geral do processo ARRANGE. 26
Figura 7 – Visão geral da atividade de codificação aberta. 27
Figura 8 – Processo de avaliação da codificação aberta. 28
Figura 9 – Visão geral da codificação axial e seletiva. 29
Figura 10 – Exemplo de codificação de dados na ferramenta MAXQDA. 35
Figura 11 – Fases e resultados da seleção de artigos. 36
Figura 12 – Exemplo do formulário de extração utilizado. 37
Figura 13 – Justiça Individual - Códigos e sua quantidade de segmentos de texto codificado. 39
Figura 14 – Justiça de Grupo - Códigos e sua quantidade de segmentos de texto codificado. 39
Figura 15 – Codificação axial - Justiça Individual 40
Figura 16 – Codificação axial - Justiça de Grupo 40
Figura 17 – SIG final de Justiça 42

LISTA DE QUADROS

Quadro 1 – Comparativo entre os trabalhos relacionados e o trabalho proposto. 32
Quadro 2 – Critérios de exclusão. 34
Quadro 3 – Questões de Pesquisa. 34
Quadro 4 – Formulário de Extração de Dados. 34
Quadro 5 – Quantidade de segmentos de texto por trabalho. 38
Quadro 6 – Exemplos de segmentos de textos 38
Quadro 7 – *Softgoals*, suas definições e trabalhos em que foram codificados. 43

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
ARRANGE	*An Approach based on the Grounded Theory Method to Refine a Quality Characteristic*
CORRELATE	*Process to Capture, Analysis and CatalOg CoRRElations between QuaLity ChAracTEristics*
ER	Engenharia de Requisitos
ES	Engenharia de *Software*
GT	*Grounded Theory*
IA	Inteligência Artificial
RNF	Requisito Não Funcional
SIG	Gráfico de Interdependência da *Softgoal*

SUMÁRIO

1 INTRODUÇÃO 13
2 OBJETIVOS 16
2.1 Objetivo geral 16
2.2 Objetivo específico 16
3 REFERENCIAL TEÓRICO 17
3.1 Aprendizado de Máquina 17
3.1.1 Categorias de Aprendizado de Máquina 17
3.1.2 Desenvolvimento de Sistemas de Aprendizado de Máquina 19
3.2 Requisitos Não-Funcionais 20
3.2.1 Tipos e Exemplos de Requisitos Não-Funcionais 20
3.2.2 Catálogos de Requisitos Não-Funcionais 21
3.2.3 Processo para Construção de Catálogos 22
3.2.3.1 Passo 1: Selecionando a característica de qualidade *23*
3.2.3.2 Passo 2: Refinando a característica em subcaracterísticas *24*
3.2.3.3 ARRANGE *25*
3.3 Requisitos Não-Funcionais para Aprendizado de Máquina 29
3.3.1 Requisito Não-Funcional Justiça 30
4 TRABALHOS RELACIONADOS 31
5 METODOLOGIA 33
5.1 Selecionando a característica Justiça 33
5.2 Execução do processo ARRANGE 33
5.2.1 Planejamento e coleta 33
5.2.2 Análise 34
5.2.3 Relatório dos resultados 35
6 RESULTADOS 36
6.1 Resultados Parciais 36
6.2 Resultado Final 39
6.2.1 Validação do SIG 41
7 CONCLUSÕES E TRABALHOS FUTUROS 44
REFERÊNCIAS 45

1 INTRODUÇÃO

Samuel (1959) propôs um novo paradigma, o Aprendizado de Máquina (AM), no qual afirmou que um computador tem a habilidade de aprender com a experiência. Jordan e Mitchell (2015) descrevem AM como uma ciência que procura construir tecnologias e sistemas que evoluem e/ou aprendem espontaneamente a partir da experiência, utilizando dados para o seu aprendizado.

Cada vez mais o AM tem despertado atenção, sendo utilizado em aplicativos de tomada de decisão e de recomendação (por exemplo, a Netflix[1] usa AM para criar experiências de usuário mais personalizadas obtendo muito sucesso atualmente), incluindo reconhecimento de imagem, processamento de linguagem e sistemas autônomos. AM contém algoritmos que usam grandes quantidades de dados para aprender e executar automaticamente tarefas que para softwares tradicionais são desafiadoras (SMOLA; VISHWANATHAN, 2008).

Contudo, sistemas baseados em AM estão se tornando mais complexos, garantir sistemas habilitados para AM seguros, justos, transparentes e precisos é um desafio (HABIBULLAH; HORKOFF, 2021). Essas são algumas características de qualidade importantes para AM. Características de qualidade são um tipo de Requisito Não Funcional (RNF), que descrevem atributos de serviço ou característica de desempenho de um produto (WIEGERS; BEATTY, 2013).

Logo, todo sistema de software é composto em parte por sua funcionalidade e a outra parte por características de qualidade como, desempenho, confiabilidade, manutenção e portabilidade. Esses RNFs desempenham um papel crítico durante o desenvolvimento do sistema. Os erros na definição de tais requisitos para o sistema costumam ser os mais caros e difíceis de corrigir, caso o sistema de software já tenha sido implementado (CHUNG *et al.*, 2000). Dessa forma, RNFs são de importância absoluta para o comportamento esperado e a qualidade dos sistemas.

Na Engenharia de Requisitos (ER), o conhecimento sobre RNFs para o software tradicional é relativamente bem estabelecido e compreendido. Porém, quando se trata de AM, parte do conhecimento adquirido sobre atributos de qualidade não se aplica, pois a maneira como são projetados, executados e mantidos difere dos tradicionais (HORKOFF, 2019). O conhecimento e experiência em relação aos RNFs para AM é escasso. Portanto, esta é relativamente uma nova área de exploração para a comunidade da ER (HABIBULLAH; HORKOFF, 2021).

[1] https://www.netflix.com/br/

Muitos pesquisadores têm se dedicado bastante para definir RNFs para AM. O trabalho realizado por Brun e Meliou (2018) representa uma chamada à ação para pesquisadores de Engenharia de *Software* (ES) para encarar o desafio que é desenvolver sistemas justos. Mostra também que, quando Justiça não é considerada uma prioridade, resulta em sistemas com comportamento inesperado e tendencioso.

Um modo de ajudar a comunidade é especificar RNFs através de catálogos. Um catálogo é um conjunto de conhecimento sobre um RNF que os engenheiros acumulam com experiências anteriores (CHUNG *et al.*, 2000).

Uma forma de representar catálogos proposta por Chung *et al.* (2000) é conhecida como Gráfico de Interdependência da *Softgoal* (SIG). Os SIGs são usados para apresentar todo o conhecimento sobre RNFs (característica de qualidade, subcaracterísticas, estratégias e correlações), é composto de *softgoals* e da ligação entre eles. O *softgoal* representa uma meta que não tem uma definição ou critérios claros sobre sua satisfação. Por isso, *softgoals* são usados para representar RNFs, porque é muito difícil ou mesmo impossível satisfazer totalmente as características de qualidade (CARVALHO, 2019).

No trabalho realizado por Carvalho (2019), um processo para catalogar RNFs do tipo característica de qualidade foi proposto. O processo é intitulado como *Process to Capture, Analysis and CatalOg CoRRElations between QuaLity ChAracTEristics* (CORRELATE). Ele é composto por quatro etapas, sendo elas: i) selecionar a característica de qualidade, ii) refinar a característica em subcaracterísticas, iii) identificar estratégias de desenvolvimento para apoiar a implementação das subcaracterísticas e iv) definir correlações com outras características de qualidade. Cada passo desse processo é apoiado por estratégias e métodos. Por exemplo, o passo 2 é apoiado pela estratégia *An Approach based on the Grounded Theory Method to Refine a Quality Characteristic* (ARRANGE), que é baseada no método de *Grounded Theory* (GT), ela é composta por quatro etapas: i) planejamento, ii) coleta, iii) análise e iv) relatar os resultados.

Este trabalho visa executar o processo ARRANGE para catalogar e refinar um RNF específico de sistemas AM a fim de contribuir com a comunidade de pesquisa nessa área. Sabendo que Justiça é um RNF crítico para tais sistemas (HORKOFF, 2019) e que não foi encontrado um catálogo para Justiça, este trabalho visa catalogar e refinar o RNF Justiça em formato de SIG. Tal catálogo deverá auxiliar desenvolvedores e pesquisadores quando for necessário tomar decisões acerca do RNF durante o desenvolvimento de sistemas AM.

O trabalho está organizado da seguinte maneira: no Capítulo 2, são definidos os

objetivos da pesquisa, geral e específicos. No Capítulo 3, são apresentados os termos e conceitos que embasam este trabalho. No Capítulo 4, são retratados os trabalhos relacionados com o presente. O Capítulo 5, contém a metodologia. No Capítulo 6, são apresentados os resultados obtidos. Por fim, o Capítulo 7 apresenta as considerações finais.

2 OBJETIVOS

Neste Capítulo, os objetivos deste trabalho são apresentados.

2.1 Objetivo geral

Este trabalho tem como objetivo geral catalogar o RNF Justiça no formato de SIG. Dessa forma, os engenheiros de software, sempre que precisarem, podem e devem consultar esse catálogo para os ajudar a tomar decisões em relação ao uso e aplicação desse RNF.

2.2 Objetivo específico

- Fazer levantamento bibliográfico sobre o RNF Justiça
- Definir e/ou identificar subcaracterísticas do RNF Justiça
- Executar a metodologia ARRANGE
- Validar o resultado do trabalho com especialistas

3 REFERENCIAL TEÓRICO

Nas Seções a seguir são apresentados os principais conceitos para o desenvolvimento deste trabalho.

3.1 Aprendizado de Máquina

Atualmente, com os avanços na tecnologia da computação, temos a capacidade de armazenar e processar grandes quantidades de dados. Para Alpaydin (2009) os dados armazenados só se tornam úteis quando analisados e transformados em informações que podemos utilizar, pois existem certos padrões neles. O aprendizado de máquina (AM) está diretamente ligado à mineração desses dados e a estatística (CORCOVIA; ALVES, 2019) e é considerado um ramo da Inteligência Artificial (IA).

Jordan e Mitchell (2015) descrevem AM como uma ciência onde o computador tem a habilidade de aprender com a experiência, utilizando dados para o seu aprendizado. Sendo uma área que estuda métodos computacionais para ganhar novas habilidades, novas formas de organizar o conhecimento existente e novos conhecimentos (MITCHELL *et al.*, 1990).

Alpaydin (2009) define aprendizado de máquina como:

> O aprendizado de máquina é a programação de computadores para otimizar um critério de desempenho usando dados de exemplo ou experiências anteriores. Temos um modelo definido até alguns parâmetros, e o aprendizado é a execução de um programa de computador para otimizar os parâmetros do modelo usando os dados de treinamento ou experiência anterior. O modelo pode ser preditivo para fazer previsões no futuro, ou descritivo para obter conhecimento dos dados, ou ambos.

Como exemplo de suas vastas áreas de aplicação temos desde recomendações automáticas de quais filmes assistir como a Netflix faz, quais alimentos pedir ou quais produtos comprar como acontece em sites e aplicativos *e-commerce*, até reconhecimento de seus amigos em suas fotos como acontece no Facebook[2]. Atualmente muitos sites e dispositivos têm algoritmos de aprendizado de máquina em sua essência (MÜLLER; GUIDO, 2016).

3.1.1 Categorias de Aprendizado de Máquina

De acordo com Raschka e Mirjalili (2019), há três tipos de aprendizagem de máquina, sendo eles: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por

[2] https://www.facebook.com/

reforço.

O aprendizado supervisionado tem como objetivo principal fazer previsões precisas para dados novos e nunca antes vistos. É utilizado quando deseja-se prever um determinado resultado de uma determinada entrada, com exemplos de pares de entrada/saída (MÜLLER; GUIDO, 2016). Como exemplo, o filtro de *spam* de *e-mail*, onde são usados *e-mails* classificados como *spam* ou não para ensinar o algoritmo. Logo, quando um novo *e-mail* chegar, o algoritmo irá prever se pertence a uma das duas categorias (RASCHKA; MIRJALILI, 2019).

No aprendizado supervisionado há duas subcategorias, sendo elas: classificação e regressão. Na classificação, o objetivo é prever os rótulos de classes, que são valores discretos e não ordenados, com base em observações anteriores, o exemplo dado anteriormente da filtragem de *spam* de *e-mail* se encaixa na subcategoria de classificação (RASCHKA; MIRJALILI, 2019). Outro exemplo de uso de classificação é quando uma instituição financeira vai analisar um cliente para saber se lhe empresta ou não uma quantia em dinheiro que será paga com juros, logo é muito importante que o banco seja capaz de prever o risco associado a esse empréstimo, neste cenário é identificado duas classes: clientes de alto risco e clientes de baixo risco. A tarefa do classificados é atribuir a entrada a uma dessas duas classes (ALPAYDIN, 2009).

Já na regressão é recebido um número de variáveis descritoras (explicativas) e uma variável de resposta contínua (resultado), e tem como objetivo encontrar uma relação entre essas variáveis, assim prevendo um resultado (RASCHKA; MIRJALILI, 2019). Um bom exemplo de sua aplicação é de um sistema que prevê o preço de um carro usado. Como entrada tem-se os atributos do carro (marca, ano, cilindrada, quilometragem e entre outras informações), onde a saída é o preço do carro (ALPAYDIN, 2009).

O aprendizado por reforço caracteriza-se como um sistema que aperfeiçoa o seu desempenho com base nas interações com o ambiente, por meio de recompensas negativas ou positivas. As recompensas são medidas a partir das interações do sistema, é analisado o quão bem o sistema atuou (RASCHKA; MIRJALILI, 2019). Um exemplo interessante é de um robô navegando em um ambiente em busca de uma localização alvo. O robô pode se mover em uma das várias direções. Depois de diversas tentativas, ele deve aprender a sequência correta de ações para alcançar o alvo sem bater em nenhum obstáculo (ALPAYDIN, 2009).

Por último o aprendizado não supervisionado, ele lida com dados não rotulados ou dados de estrutura desconhecida a priori. Diferentemente dos aprendizados supracitados, que precisam da orientação de uma variável de resultado conhecida ou função de recompensa, o não

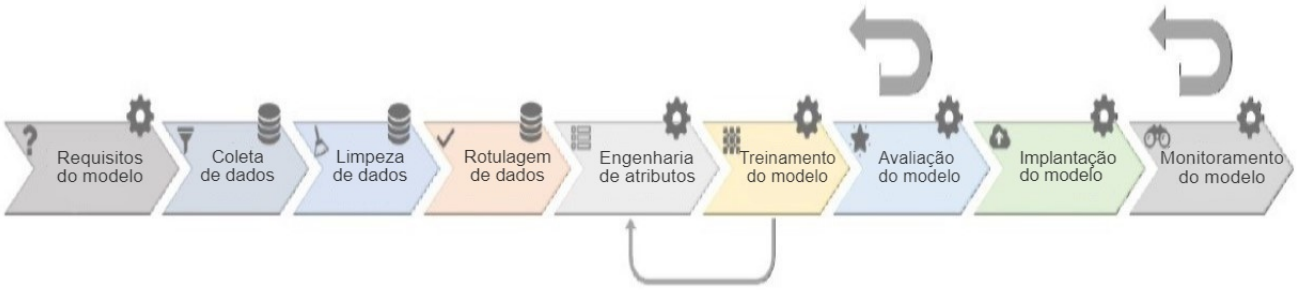
supervisionado explora a estrutura dos dados para extrair informações significativas (RASCHKA; MIRJALILI, 2019).

Um exemplo de uso do aprendizado não supervisionado é quando uma empresa decide ver que tipo de clientes que aparecem com frequência. Utilizando os dados de clientes anteriores, que contêm as informações demográficas e transações anteriores com a empresa. Diante disso, um modelo de agrupamento aloca clientes semelhantes de acordo com seus atributos para o mesmo grupo, proporcionando à empresa agrupamentos naturais de seus clientes. Quando são identificados tais grupos, a empresa pode decidir estratégias, seja de *marketing*, serviços e/ou produtos, específicos para os diferentes grupos (ALPAYDIN, 2009).

3.1.2 Desenvolvimento de Sistemas de Aprendizado de Máquina

Amershi *et al.* (2019) apresentam um fluxo de desenvolvimento de sistemas AM, o qual possui nove estágios (ver Figura 1), sendo eles: requisitos do modelo, coleta de dados, limpeza de dados, rotulagem de dados, engenharia de atributos, treinamento do modelo, avaliação do modelo, implantação do modelo e monitoramento do modelo. Os fluxos de trabalho de AM não são sequenciais, pois contêm vários ciclos de *feedback*, devido à quantidade de experimentação necessária para definir um bom modelo para o problema.

Figura 1 – Fluxo de desenvolvimento de aprendizado de máquina.



Fonte: Adaptada de (AMERSHI *et al.*, 2019).

No estágio de requisitos do modelo, os *designers* decidem quais tipos de modelos são mais apropriados para o problema em questão, quais atributos são viáveis e quais podem ser úteis de implementar com AM. Para o estágio de coleta de dados, as equipes procuram e integram conjuntos de dados disponíveis, seja internos ou de código aberto, ou coletam seus próprios. O estágio de limpeza de dados envolve a remoção de registros imprecisos ou ruidosos do conjunto de dados coletados. Durante a rotulagem de dados, são atribuidos rótulos reais para cada registro, os rótulos podem ser fornecidos pelos próprios engenheiros.

A engenharia de atributos trata de todas as atividades realizadas para extrair e selecionar atributos informativos para modelos de AM. É através do treinamento do modelo, que os modelos escolhidos, utilizando os atributos selecionados durante o estágio de requisitos do modelo, são treinados e ajustados nos dados limpos e coletados, com seus respectivos rótulos. Em seguida, na avaliação do modelo, os engenheiros avaliam o modelo de saída nos conjuntos de dados testados utilizando métricas predefinidas. Só então, o código do modelo é implantado no(s) dispositivo(s) de destino e monitorado constantemente quanto a possíveis erros durante a execução no mundo real.

3.2 Requisitos Não-Funcionais

Um sistema de software é composto em parte por sua funcionalidade e a outra parte por características de qualidade como desempenho, confiabilidade, manutenção e portabilidade. Esses requisitos não-funcionais (RNFs) desempenham um papel crítico durante o desenvolvimento do sistema (CHUNG *et al.*, 2000). Para garantir que os sistemas funcionem da melhor maneira possível é muito importante que os engenheiros de software forneçam as especificações corretas, pois as características de qualidade dos sistemas têm um papel crucial e precisam ser pensadas desde o início do desenvolvimento (CARVALHO, 2019).

O sucesso do software não está apenas relacionado a fornecer a funcionalidade certa, mas também com a expectativa do usuário sobre o desempenho do produto. Esse desempenho corresponde aos atributos de qualidade, que são um tipo de RNF que retrata um serviço ou uma característica de desempenho de um produto (WIEGERS; BEATTY, 2013), tal definição será considerada para este trabalho.

3.2.1 Tipos e Exemplos de Requisitos Não-Funcionais

Wiegers e Beatty (2013) afirmam que os atributos de qualidade são divididos em dois tipos: de qualidade externa e de qualidade interna. Os externos influenciam profundamente a compreensão do usuário sobre a qualidade do software e sua experiência, sendo observados durante a execução do sistema. Logo abaixo encontram-se exemplos:

- Desempenho: corresponde a capacidade de resposta do sistema a várias consultas e ações do usuário.
- Usabilidade: aborda os inúmeros fatores que constituem a facilidade de uso e engenharia

humana.

- Segurança: lida com o bloqueio de acesso não autorizado a funções ou dados do sistema.

Diferentemente dos atributos de qualidade externos, Wiegers e Beatty (2013) definem que os internos não são observados durante a execução do sistema, são propriedades notadas ao verificar o *design* ou código, que podem afetar indiretamente a percepção do cliente sobre a qualidade do software. A seguir exemplos:

- Modificabilidade: trata da facilidade com que os projetos e o código do sistema podem ser alterados, estendidos e compreendidos.
- Escalabilidade: é a capacidade do aplicativo de crescer sem comprometer o desempenho ou a exatidão.
- Reusabilidade: revela o esforço relativo necessário para converter um componente do sistema para uso em outras aplicações.

3.2.2 Catálogos de Requisitos Não-Funcionais

Para facilitar a consideração dos RNFs, foram criados catálogos de qualidades de software (DAMYANOVA, 2020). Para identificarem as interações entre as características, os desenvolvedores devem fazer uso de catálogos (WIEGERS; BEATTY, 2013). Tais catálogos são gerados a partir de um conjunto de conhecimento de *design* e técnicas de desenvolvimento que foi acumulado de experiências anteriores (CHUNG *et al.*, 2000).

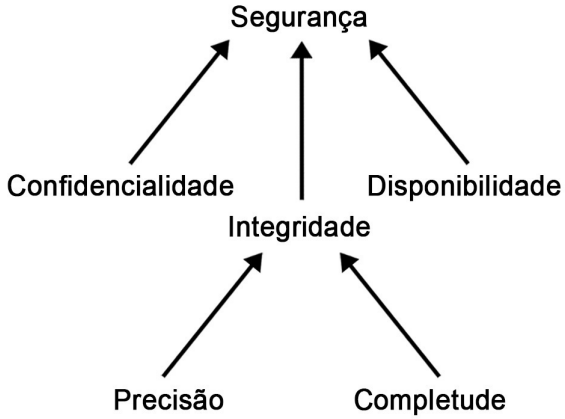
Carvalho (2019) mostra que existem três tipos de informações a catalogar quanto às características de qualidade, sendo elas: subcaracterísticas, soluções (estratégias de desenvolvimento que apoiam a característica de qualidade em um sistema) e correlações.

Todo o conhecimento sobre um dado RNF é representado por subcaracterísticas, informações como conceitos e terminologia, normalmente são documentadas como subcaracterísticas. Como exemplo, a Figura 2 representa um catálogo do RNF Segurança, onde tal RNF é refinado em três subcaracterísticas: Confidencialidade, Integridade e Disponibilidade. A partir da subcaracterística Integridade são geradas mais duas sub subcaracterísticas: Precisão e Completude.

Uma maneira de representar catálogos, proposta por Chung *et al.* (2000), é chamada Gráfico de Interdependência da Softgoal (SIG), ela tem sido usada para representar todo o conhecimento sobre RNFs (característica de qualidade, subcaracterística, estratégias e correlações).

Um SIG é composto de *softgoals*, que é o seu elemento chave, e *links* entre eles. Um

Figura 2 – Exemplo de catálogo de subcaracterísticas.



Fonte: Adaptada de (CHUNG *et al.*, 2000).

softgoal representa uma meta que não possui definição ou critérios claros sobre sua satisfação. Os *softgoals* podem ser divididos em *softgoals* RNF (ou seja, características de qualidade) ou *softgoals* de operacionalização.

Os *softgoals* RNF são representados como uma nuvem no SIG. Eles podem ser refinados em *softgoals* mais específicos. Como no exemplo dado na Figura 2, que apresenta um SIG em que o principal *softgoal* RNF é Segurança, composto por três *softgoals* RNF específicos: Integridade, Confidencialidade e Disponibilidade.

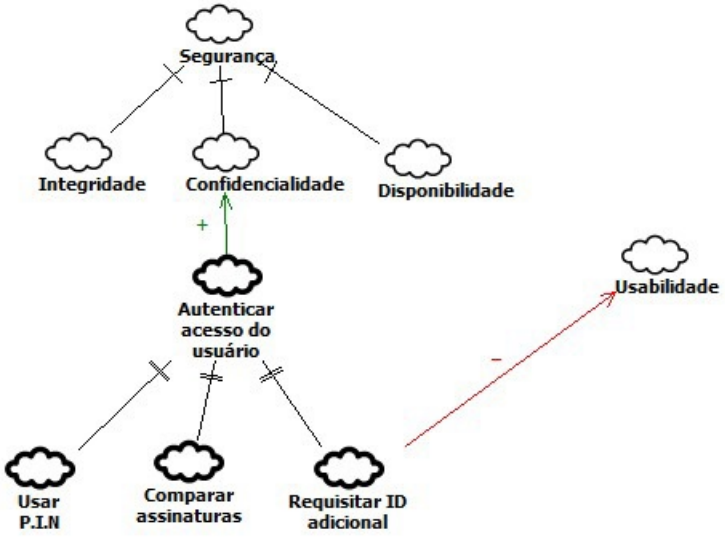
Os *softgoals* RNF são satisfeitos pelos *softgoals* operacionalizantes, desenhados como nuvens escuras no gráfico. Eles representam estratégias de *design* ou implementação. Na Figura 3, tem um *softgoal* operacionalizador chamado “Autenticar acesso do usuário”, que é usado para ajudar a Confidencialidade, e que foi refinado outra vez e resultou em mais três opções: Usar P.I.N, Comparar Assinaturas e Requisitar ID adicional. É possível observar no gráfico, que "Requisitar ID adicional", tem um conflito com a característica Usabilidade.

3.2.3 Processo para Construção de Catálogos

A abordagem desenvolvida por Carvalho (2019) é descrita como um processo sistemático e reutilizável para capturar, analisar e catalogar as correlações entre os RNFs. O processo foi nomeado CORRELATE (*Process to Capture, Analyze and Catalog Correlations between Quality Characteristics*), ele também possibilita a criação de uma estrutura hierárquica de um RNF, contendo subcaracterísticas e métodos. A sua execução é dividida em etapas relacionadas ao refinamento de um RNF até alcançar as estratégias de desenvolvimento de software.

Esse processo é composto por quatro passos gerais: (1) selecionar a característica de

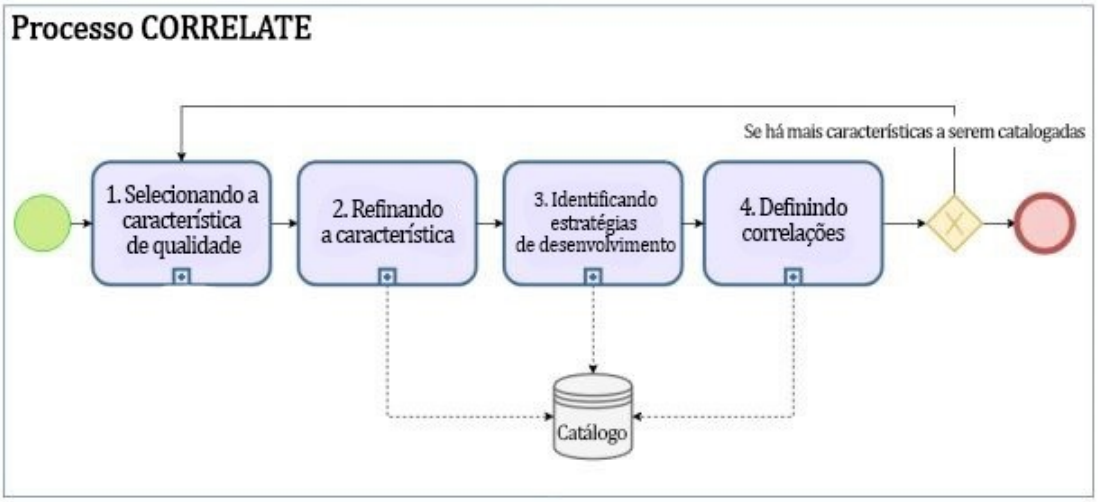
Figura 3 – Exemplo de um SIG.



Fonte: Adaptada de (CHUNG *et al.*, 2000).

qualidade, (2) refinar a característica, (3) identificar estratégias de desenvolvimento e (4) definir correlações. Na Figura 4 é possível ter uma visão geral do processo.

Figura 4 – Visão geral do processo CORRELATE.



Fonte: Adaptada de (CARVALHO, 2019).

3.2.3.1 Passo 1: Selecionando a característica de qualidade

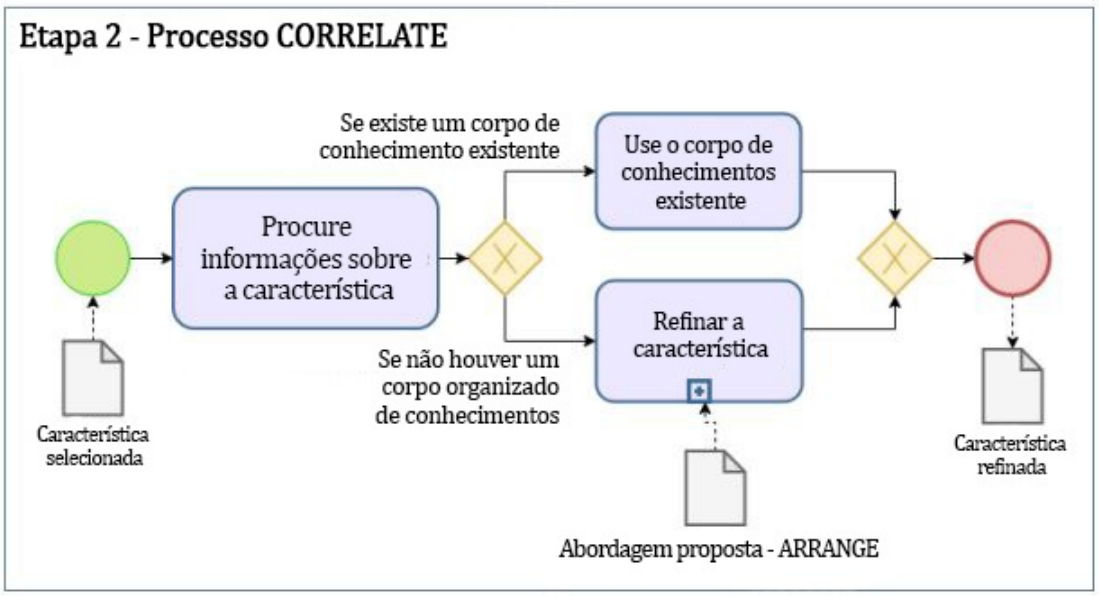
Neste trabalho vamos tratar RNF como característica de qualidade. Para selecionar uma característica o usuário do processo CORRELATE tem duas formas de fazê-lo, a primeira é priorizando RNF através de alguns critérios, seja aplicando técnicas como revisão de literatura ou

questionários, ou por entrevistas com especialistas. A segunda forma de selecionar a característica é feita de maneira arbitrária, ou seja, o RNF que o usuário desejar catalogar.

3.2.3.2 Passo 2: Refinando a característica em subcaracterísticas

Após selecionar a característica na etapa anterior, o próximo passo é pesquisar informações sobre ela, logo utilizando este conhecimento existente para refinar a característica. Caso o conhecimento sobre a característica seja fragmentado ou disperso, uma abordagem denominada ARRANGE (*An Approach based on the Grounded Theory Method to Refine a Quality Characteristic*) é utilizada. A Figura 5 representa a visão completa do passo 2.

Figura 5 – Passo 2 do processo CORRELATE.



Fonte: Adaptada de (CARVALHO, 2019).

A literatura é utilizada no processo como fonte de conhecimento visto que é a fonte de informação mais utilizada para se obter conhecimento sobre subcaracterísticas. Além disso, este trabalho utiliza uma técnica de análise de dados qualitativos para analisar dados da literatura, a *Grounded Theory* (GT), que é um método desenvolvido para construir teoria a partir de dados (CORBIN; STRAUSS, 2014).

Tal método liga conceitos derivados de dados até chegar à categoria principal, seu uso como uma abordagem para extrair e relacionar conceitos dos dados da literatura se encaixa na estrutura de um catálogo para RNF no formato SIG. Dessa forma, o método GT embasa a abordagem ARRANGE, que será descrita a seguir.

3.2.3.3 ARRANGE

O ARRANGE é utilizado para apoiar a atividade em que o usuário do processo CORRELATE deve refinar a característica com base em outras fontes, ele é constituído por quatro fases, sendo elas: planejamento, coleta, análise e relatório dos resultados, a Figura 6 resume a execução da abordagem. Suas etapas são detalhadas abaixo.

Na fase de planejamento é definida a área de pesquisa, diante disso temos a questão geral da pesquisa: "Como o <RNF> em <tipo de sistema> pode ser definido e refinado em subcaracterísticas e soluções?".

Logo após, na fase de coleta de dados visa coletar os dados precisos para responder a questão que embasa a pesquisa, por meio de uma revisão sistemática da literatura ou também pelo procedimento de *snowballing* (KITCHENHAM *et al.*, 2007) (PETERSEN *et al.*, 2015) para estruturar essa tarefa. Tem duas formas de utilizar a técnica *snowballing*, a primeira se refere ao uso de uma lista de referência de um artigo (*backward snowballing*) e a segunda maneira são citações a esse artigo (*forward snowballing*) para identificar artigos adicionais.

Ademais é preciso extrair informações dos papéis para construir um conjunto de dados de um RNF. As questões abaixo foram propostas por Carvalho (2019) para orientar a extração de dados:

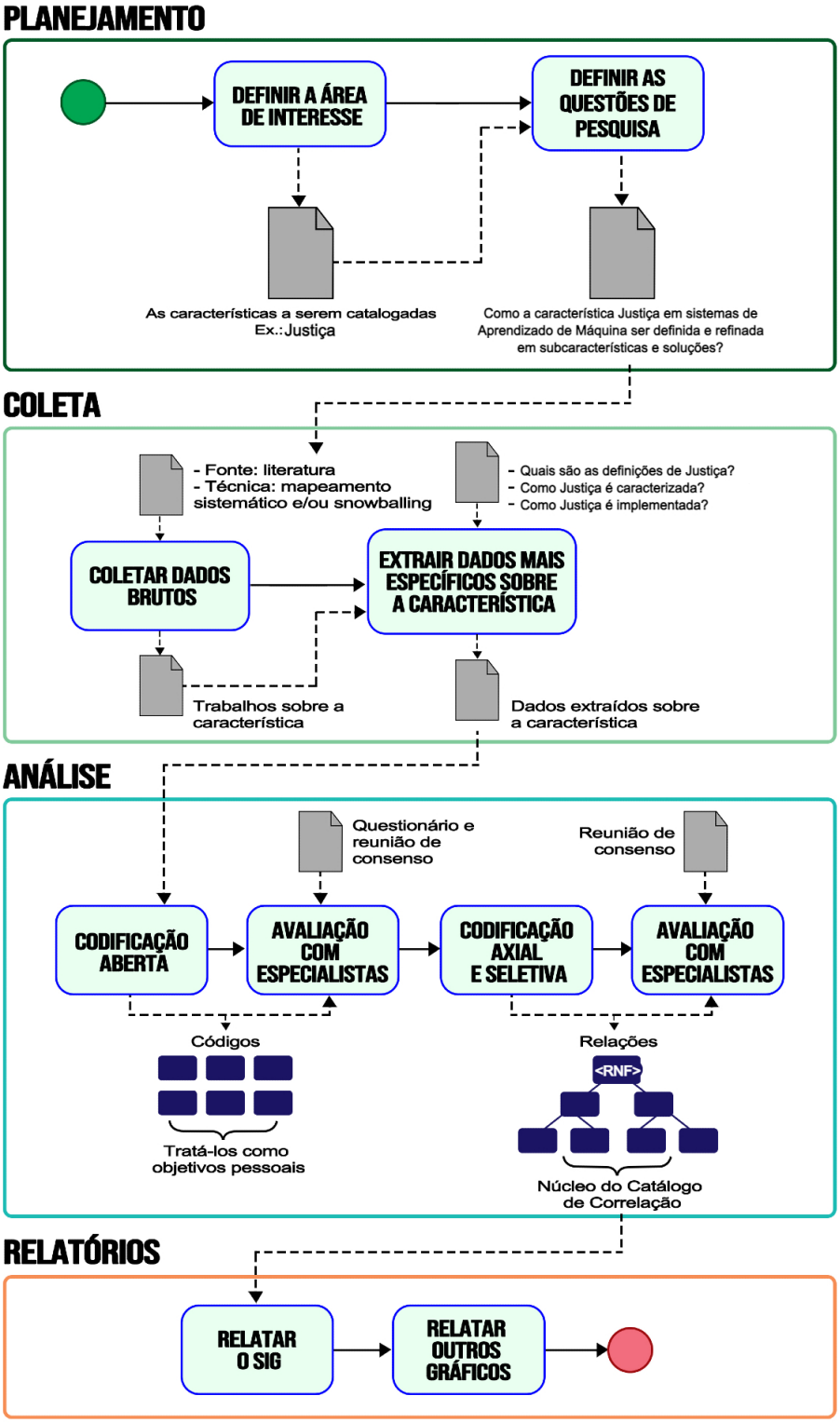
- Quais são as definições do <RNF>?
- Como o <RNF> é caracterizado?
- Como o <RNF> é implementado?

A fase de análise tem como objetivo obter um conjunto bem definido de subcaracterísticas inter-relacionadas dos dados extraídos na fase anterior através de codificação. Para realizar a codificação existem ferramentas que auxiliam a execução da atividade. A ferramenta MAXQDA24 [3], que após o período de teste é paga e a QDA Miner Lite [4], que é gratuita. O GT é utilizado nessa etapa, nele a codificação é feita em três atividades: codificação aberta, axial e seletiva.

Durante a codificação aberta os dados extraídos dos papéis anteriormente são inspecionados, gerando as respostas para o formulário de extração (ver a Figura 7). Dessa maneira, seguindo os campos do formulário de extração, a codificação deve iniciar com a leitura de todos os textos relacionados à definição do RNF, que neste trabalho é Justiça, para responder todos os

[3] https://www.maxqda.com/pt
[4] https://provalisresearch.com/products/qualitative-data-analysis-software/

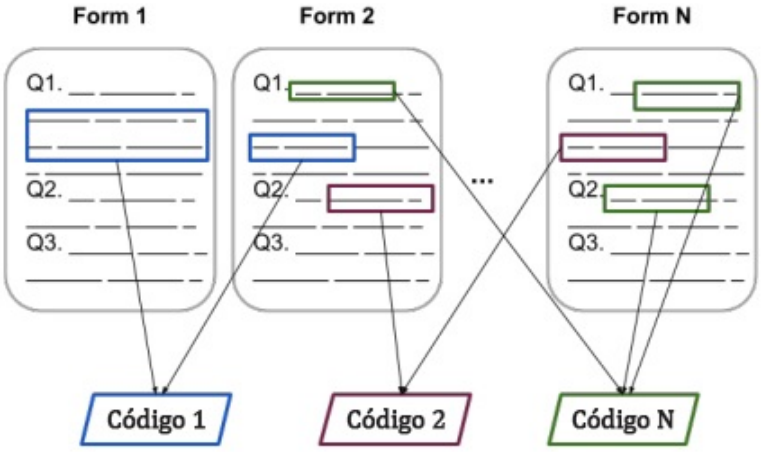
Figura 6 – Visão geral do processo ARRANGE.



Fonte: Adaptada de (CARVALHO, 2019).

dados do formulário. Em seguida, o usuário do processo deve criar um nome conceitual (código) para representar sua compreensão dos dados. Os códigos podem se caracterizar como uma única palavra, uma frase ou um parágrafo inteiro.

Figura 7 – Visão geral da atividade de codificação aberta.



Fonte: Adaptada de (CARVALHO, 2019).

Cada informação que está sendo lida precisa ser comparada com outras informações, esse conjunto de dados serve para checar as semelhanças e diferenças. A cada momento que encontrar informações semelhantes, o código existente deve ser reutilizado. O nome dessa estratégia é “comparação constante”, geralmente utilizada para análises qualitativas (CORBIN; STRAUSS, 2014).

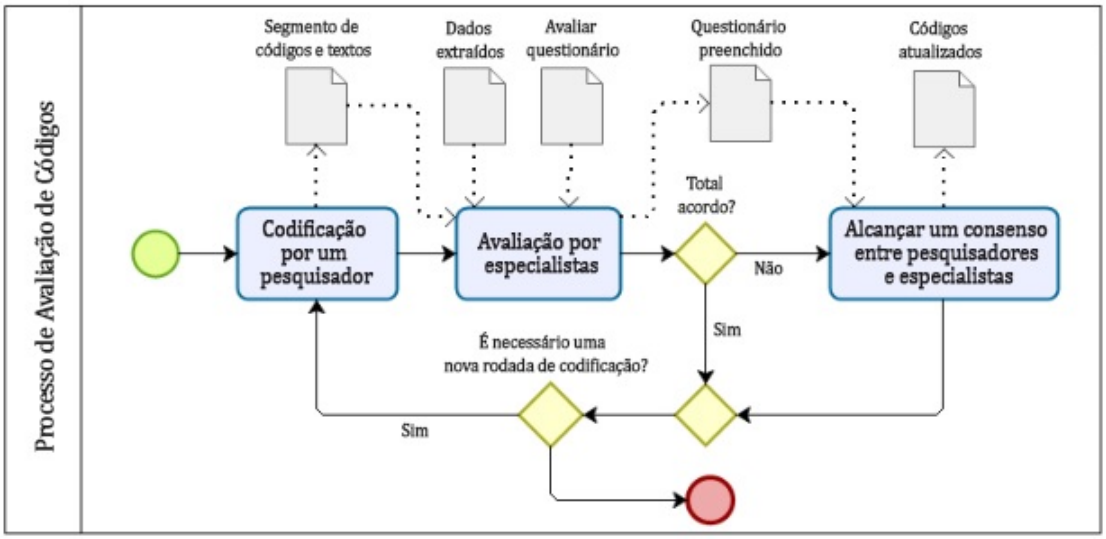
Ao final da codificação aberta, um conjunto de códigos que reflete a compreensão do usuário do processo das subcaracterísticas do RNF. Para reduzir possíveis interpretações incorretas e colaborar com conceitos especializados relacionados ao RNF, é de suma importância avaliar essas codificações com a ajuda de especialistas na área.

Para a avaliação ser feita, primeiramente devem ser enviados aos especialistas os códigos e segmentos de texto, depois, por meio de um questionário, onde cada informação enviada deve ser avaliada e classificada em: (i) concordo; (ii) concordo parcialmente; ou (iii) discordo. Depois é preciso analisar os resultados, caso não tenha sido obtido um acordo total, se faz necessário realizar uma reunião para discussão e consenso. A Figura 8 mostra a visão geral do processo de avaliação.

O conjunto de códigos resultante significa objetivos programáticos do SIG, seja como objetivo programático do RNF (ou seja, características ou subcaracterísticas) ou como um objetivo programático operacional (isto significa, estratégia de desenvolvimento). O próximo passo é relacioná-los entre si, no método GT isto é feito por meio da codificação axial.

A codificação axial é o processo de relacionar conceitos ou agrupá-los criando categorias (um conceito de alto nível que representa um grupo de códigos) (CORBIN; STRAUSS, 2014). Neste trabalho as relações são criadas através da análise do significado implícito entre os

Figura 8 – Processo de avaliação da codificação aberta.



Fonte: Adaptada de (CARVALHO, 2019).

códigos e da utilização dos tipos de contribuições da notação SIG (*AND*, *OR*, *MAKE*, *HELP*, *BREAK* e *HURT*). O usuário do processo também pode agrupar categorias em novas categorias, criando uma cadeia de categorias.

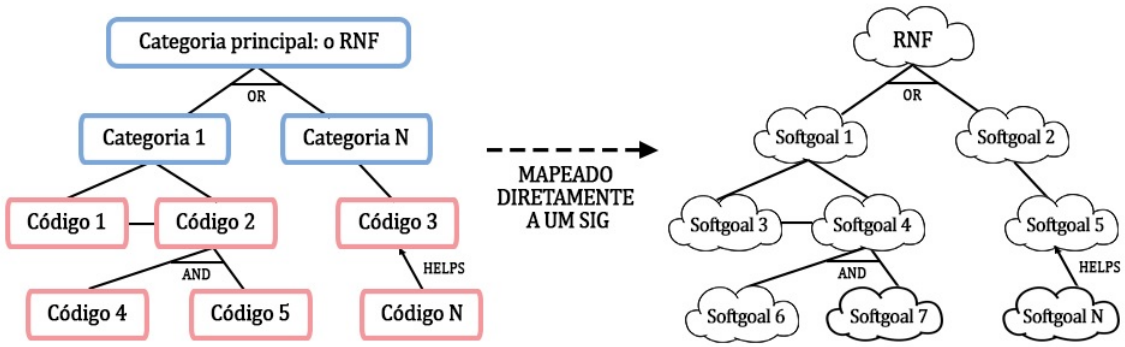
Ao final, quando todos os códigos e categorias conseguem ser relacionados a uma categoria principal, isso quer dizer que o usuário do processo está fazendo a Codificação Seletiva. No trabalho de Corbin e Strauss (2014), codificação seletiva é definida como um processo que vincula conceitos em torno de uma categoria central, refinando e promovendo o resultado da construção teórica do RNF investigado.

Depois da codificação axial e seletiva, os conceitos (isto é, os códigos e categorias) e suas relações devem ser mapeados diretamente para um SIG (Ver Figura 9). Cada conceito será um objetivo programático da RNF ou um objetivo programático operacional. Então, o usuário do processo deve classificar cada código de acordo com a definição do RNF e objetivos operacionalizados.

A codificação axial e seletiva podem ser realizadas em conjunto pelo usuário do processo. Os códigos devem ser analisados para agrupá-los em categorias, determinar as relações necessárias e relacioná-los com a categoria principal, sempre com o esforço de descrever como os conceitos se fundamentam nos dados.

Em seguida, reuniões e discussões podem ser realizadas com os mesmos especialistas que participaram da codificação aberta para chegar a um acordo sobre a organização dos conceitos. Finalmente, após a codificação seletiva, o usuário do processo poderá elaborar uma definição para o RNF que está sendo catalogado. Para tanto, as definições das subcaracterísticas devem ser

Figura 9 – Visão geral da codificação axial e seletiva.



Fonte: Adaptada de (CARVALHO, 2019).

utilizadas para compor a descrição do RNF.

A última etapa da abordagem ARRANGE, de relatório dos resultados, é nesse passo que o SIG pode ser reportado, bem como algumas estatísticas sobre os códigos. Cada conceito é fundamentado por duas medidas, ou seja, quantos segmentos de texto estão relacionados a ele e a densidade de cada conceito (ou seja, quantos artigos o suportam).

Conforme visto no passo anterior, é possível construir um SIG por meio do método GT. O SIG resultante já pode vir com estratégias que auxiliem o desenvolvedor na implantação da característica de qualidade, uma vez que uma das questões da extração é “Como é implementado o <RNF>?”.

3.3 Requisitos Não-Funcionais para Aprendizado de Máquina

Grandes investimentos em sistemas que visam explorar os poderes do aprendizado de máquina foram feitos na última década. Há muita preocupação e expectativa sobre a qualidade e comportamento esperado em sistemas de AM. A prática da engenharia de software para sistemas de AM é atualmente um tópico de pesquisa aberto (PONS; OZKAYA, 2019). Ishikawa e Yoshioka (2019) sugerem que a Engenharia de Requisitos é a atividade mais difícil para o desenvolvimento de sistemas baseados em AM.

Santhanam (2020) deixa claro que as aplicações de AM precisam de uma mentalidade diferente desde o início de um projeto, porque precisam ser projetados com características únicas (PONS; OZKAYA, 2019). Decisões como a definição das funções de adequação, seleção e preparação de dados e garantia de qualidade devem ser feitas baseadas na compreensão do domínio do negócio e nas necessidades das partes interessadas, isso se enquadra na profissão de engenheiro de requisitos (VOGELSANG; BORG, 2019).

Damyanova (2020) apresenta vários RNFs exclusivos para sistemas habilitados para AM, como explicabilidade, justiça, treinabilidade. Siebert *et al.* (2021) enfatizam que algumas das novas propriedades de qualidade são bastante genéricas, porque abrangem não apenas o aprendizado de máquina, mas também outras disciplinas de inteligência artificial (IA). Logo abaixo encontram-se as características de qualidade citadas no trabalho (SIEBERT *et al.*, 2021):

- Transparência e responsabilidade (por exemplo, interpretabilidade e explicabilidade).
- Diversidade, não discriminação, justiça, bem como bem-estar social e ambiental (por exemplo, evitar preconceitos injustos, acessibilidade e *design* universal)
- Segurança, proteção, proteção de dados (por exemplo, qualidade e integridade dos dados, capacidade de lidar com dados de entrada errôneos)
- Robustez e confiabilidade (por exemplo, exatidão da saída, robustez contra entradas prejudiciais, erros ou situações inesperadas)
- Agência humana e supervisão, aspectos legais e éticos (por exemplo, possibilidade de supervisão humana, respeito pelos direitos fundamentais)

3.3.1 Requisito Não-Funcional Justiça

Muito esforço tem sido feito pela comunidade de pesquisa sobre Justiça, por exemplo, o IEEE/ACM organizou um *workshop* internacional sobre justiça de software chamado *FairWare* 2018 [5]. Zhang *et al.* (2020) citam que a literatura tem proposto muitas definições de Justiça, mas nenhum consenso firme é alcançado neste momento.

Chouldechova (2017) define Justiça como a ausência de qualquer viés baseado nas características intrínsecas ou adquiridas de um indivíduo que são irrelevantes no contexto particular de tomada de decisão. Em muitos contextos, essas características intrínsecas (mencionadas como “atributos sensíveis” ou “atributos protegidos” na literatura) são gênero, religião, raça, cor da pele ou idade (SAXENA *et al.*, 2019).

Chakraborty *et al.* (2019) dividem Justiça em duas categorias: justiça do grupo e justiça individual. Justiça do grupo tem como objetivo que, com base no atributo protegido, grupos privilegiados e não privilegiados sejam tratados de forma semelhante e a justiça individual tem como objetivo que indivíduos semelhantes recebam resultados semelhantes.

[5] http://fairware.cs.umass.edu/

4 TRABALHOS RELACIONADOS

Para fornecer uma melhor base para este trabalho, foi realizada uma revisão da literatura que constituiu-se em buscas de trabalhos em máquinas de busca como Google Acadêmico[6]. Neste capítulo, são apresentados e discutidos os trabalhos resultantes desta revisão considerados relevantes e relacionados com a presente proposta a fim de identificar semelhanças e diferenças, assim como validar e tornar viável o trabalho proposto.

O trabalho realizado por Lourenço (2020) tem como objetivo capturar subcaracterísticas e soluções para *Calmness* e catalogá-las em um SIG. Foi utilizada uma abordagem já existente para catalogar o RNF, usando métodos de pesquisa bem definidos: *snowballing*, pesquisa em bases de dados e teoria fundamentada em questionários e *grounded theory*, pesquisa que une conceitos decorrentes de dados até alcançar a categoria principal.

Como resultado final, foi obtido um SIG de *Calmness* composto por duas subcaracterísticas principais e treze sub subcaracterísticas. O catálogo pode ser útil para apoiar engenheiros de software na especificação de requisitos e soluções práticas para aplicativos da Computação Ubíqua e *Internet* das coisas que consideram *Calmness* como um RNF.

Portanto, o trabalho de Lourenço (2020) se relaciona diretamente com a proposta deste trabalho, pois ambos têm como objetivos capturar subcaracterísticas e soluções para catalogar em um SIG, utilizam a abordagem CORRELATE desenvolvida por (CARVALHO, 2019). A grande diferença é o foco de pesquisa, enquanto o trabalho citado foca na característica *Calmness* usada em Computação Ubíqua e *Internet* das Coisas, o aqui proposto foca na característica Justiça utilizada em aplicações de Aprendizado de Máquina.

O trabalho realizado por Silva *et al.* (2020), visa construir um catálogo em formato SIG para RNFs usados em projetos de Sistemas Embarcados, assim ajudando desenvolvedores na implementação de soluções personalizadas. Foi utilizado o processo NFR4ES (*Non-Functional Requirements for Embedded Systems*) para a construção do catálogo. O catálogo foi avaliado através de uma prova de conceito e pela opinião de especialistas, a análise feita pelos especialistas resultou no aperfeiçoamento do catálogo através de sugestões de novos RNFs, ajustes nas definições e ajustes na hierarquia dos requisitos.

Tendo como resultado final um catálogo que contempla 44 RNFs, analisados por quatro especialistas do domínio, visando a verificação da consistência dos RNFs. Além do

[6] https://scholar.google.com

catálogo em formato SIG, os especialistas também identificaram correlações entre os RNFs apresentados.

Sendo assim, o trabalho de Silva *et al.* (2020) relaciona-se diretamente, porque manifesta objetivos similares, com uma metodologia bem estabelecida e também visa catalogar RNFs. A grande diferença é o foco de pesquisa, enquanto o trabalho citado foca em RNFs para Sistemas Embarcados, o aqui proposto foca na característica Justiça utilizada em aplicações de Aprendizado de Máquina.

Brun e Meliou (2018) descrevem uma visão de como a pesquisa de Engenharia de Software (ES) pode ajudar a reduzir os defeitos de imparcialidade e representa um apelo à comunidade de pesquisa de ES para dar forma material a essa visão.

Assim, Brun e Meliou (2018) apresenta os desafios de pesquisa significativos na criação de suporte ao desenvolvedor em cada uma das seguintes áreas: levantamento, design, teste e verificação de requisitos, ou seja, todas as partes críticas do ciclo de vida de ES e da garantia da qualidade do software, sendo todas elas importantes para garantir a justiça do software.

O trabalho de Brun e Meliou (2018) se relaciona com o aqui proposto, pois trata de Justiça e busca ajudar a comunidade de pesquisa e desenvolvedores a tomar decisões corretas em relação ao RNF no desenvolvimento de aplicações de Aprendizado de Máquina. A grande diferença é que ele não cataloga o RNF Justiça em nenhum formato.

O Quadro 1 apresenta um comparativo entre os trabalhos relacionados com o trabalho proposto, destacando quais RNFs são representados, o tipo do catálogo, qual o domínio e a metodologia utilizada.

Quadro 1 – Comparativo entre os trabalhos relacionados e o trabalho proposto.

	RNF representado	**Tipo de catálogo representado**	**Domínio**	**Metodologia**
(LOURENÇO, 2020)	*Calmness*	SIG	Computação Ubíqua e Internet das Coisas	CORRELATE
(SILVA *et al.*, 2020)	Requisitos Não-Funcionais	SIG	Sistemas Embarcados	NFR4ES
(BRUN; MELIOU, 2018)	Justiça	nenhum	Aprendizado de Máquina	Revisão da Literatura
Trabalho proposto	Justiça	SIG	Aprendizado de Máquina	ARRANGE

Fonte: Elaborado pelo autor.

5 METODOLOGIA

A metodologia utilizada neste trabalho foi desenvolvida em (CARVALHO, 2019). O presente trabalho refina o RNF Justiça por meio da abordagem ARRANGE, que faz parte do CORRELATE, ambos desenvolvidos no trabalho citado. A execução será detalhada a seguir.

5.1 Selecionando a característica Justiça

Inicialmente uma característica de qualidade é selecionada, Justiça foi selecionada por ser um RNF crítico para o Aprendizado de Máquina. Após uma pesquisa bibliográfica no Google Acadêmico [7] nenhum catálogo no formato SIG foi encontrado para o mesmo.

Portanto, este trabalho visa catalogar o RNF Justiça em formato SIG, como uma forma de ajudar desenvolvedores e pesquisadores quando for necessário tomar decisões acerca do RNF durante o desenvolvimento de sistemas de Aprendizado de Máquina.

5.2 Execução do processo ARRANGE

A execução de cada etapa do ARRANGE é descrita nas subseções abaixo, em conjunto com o resultado esperado de cada uma. A metodologia conta com 4 passos (planejamento, coleta, análise e relatório dos resultados), todos eles são apresentados a seguir.

5.2.1 Planejamento e coleta

O RNF Justiça foi definido como área de interesse, portanto a questão de pesquisa que irá direcionar o trabalho é: "Como a característica Justiça em sistemas de Aprendizado de Máquina pode ser definida e refinada em subcaracterísticas?". Logo após, na etapa de coleta, foi feito um levantamento bibliográfico em 3 bases de dados, sendo elas: ACM [8], Scopus [9] e IEEE [10].

Depois de feita a busca inicial, foi realizada a aplicação de critérios básicos de exclusão, que são mostrados no Quadro 2.

Para afunilar o conjunto de trabalhos resultante da busca nas bases de dados foi necessário fazer uma leitura do título e resumo. Em seguida, foi feita uma leitura mais aprofundada

[7] https://scholar.google.com
[8] https://dl.acm.org/, acesso em: 03 de Outubro de 2023
[9] https://www.scopus.com/home.uri, acesso em: 03 de Outubro de 2023
[10] https://ieeexplore.ieee.org/Xplore/home.jsp, acesso em: 03 de Outubro de 2023

Quadro 2 – Critérios de exclusão.

Critérios
Artigos não escritos em inglês
Trabalhos duplicados
Não ter uma definição de Justiça no contexto de Aprendizado de Máquina

Fonte: Elaborado pelo autor.

dos artigos, onde foram revisadas as partes mais relevantes. Ao final, foi realizada uma leitura completa dos artigos, com o objetivo de responder as questões de pesquisa, que são apresentadas no Quadro 3.

Quadro 3 – Questões de Pesquisa.

ID	Questão
QP1	Quais são as definições de Justiça?
QP2	Como a Justiça é caracterizada?
QP3	Como a Justiça é implementada?

Fonte: Elaborado pelo autor.

Durante a etapa de planejamento e coleta, os formulários de extração de dados foram preenchidos, coletando informações referente as questões de pesquisa definidas. Cada formulário é composto pelos dados mostrados no Quadro 4. O artigo não precisaria necessariamente responder as três questões de pesquisa.

Quadro 4 – Formulário de Extração de Dados.

I	II	III	IV	V
ID do Artigo	Título	QP1	QP2	QP3

Fonte: Elaborado pelo autor.

Deste modo, artigos que apresentam uma definição de Justiça já estariam inclusos na próxima etapa: a etapa de análise, que será detalhada logo abaixo.

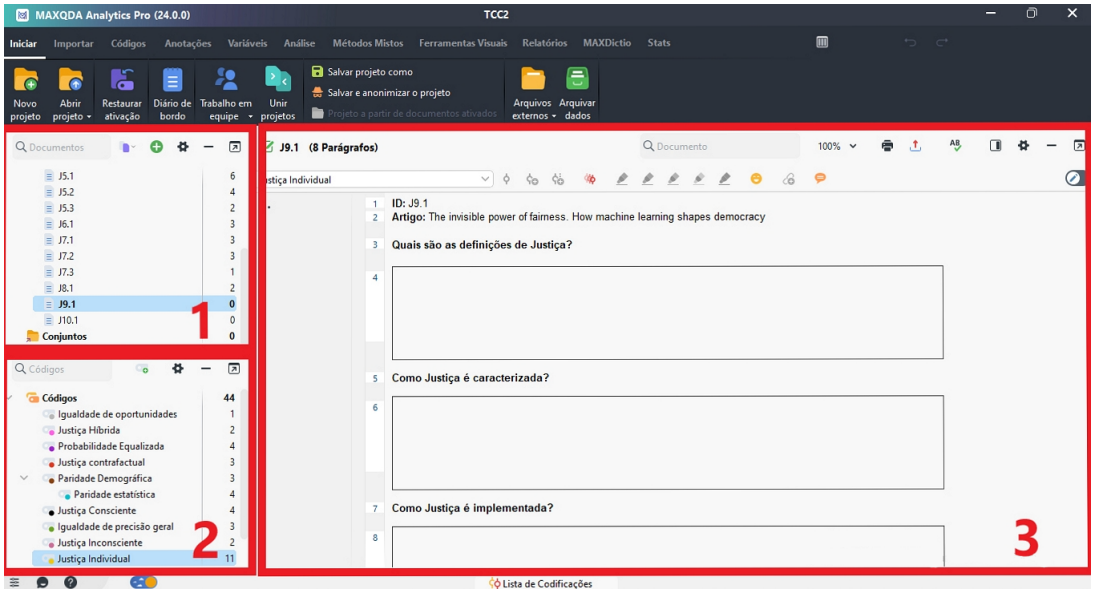
5.2.2 Análise

Durante a etapa de análise o método GT foi utilizado, com isso foram executadas três atividades: codificação aberta, axial e seletiva. Na codificação aberta os dados extraídos na etapa anterior foram inspecionados, gerando as respostas para o formulário de extração (ver

Quadro 4). A ferramenta MAXQDA foi escolhida para realizar a codificação dos dados.

A Figura 10 proporciona uma visão geral sobre a ferramenta utilizada. A caixa número 1 é o lugar no qual todos os formulários de extração estão listados, eles são tratados como documentos. A caixa número 2 apresenta os códigos, enquanto a caixa número 3 é usada para ver as informações nos formulários de extração e codificá-las.

Figura 10 – Exemplo de codificação de dados na ferramenta MAXQDA.



Fonte: Elaborado pelo autor.

Como foi explicado anteriormente, os códigos representam os dados em uma palavra ou frase. Os formulários resultantes na etapa passada foram todos importados para a ferramenta utilizada. Após realizada, a codificação aberta, a codificação axial e seletiva foi feita em conjunto, analisando os códigos e os agrupando em categorias, determinando suas relações e relacionando com a categoria principal. Com isso, todas as atividades do GT foram finalizadas. Os códigos foram avaliados por dois especialistas, de acordo com o que é proposto no ARRANGE.

5.2.3 Relatório dos resultados

No último passo do ARRANGE, foi possível criar um catálogo de requisitos para implementar Justiça, através da estrutura do código gerado pelo método *Grounded Theory* será representada e relatada em um SIG.

6 RESULTADOS

Com a característica selecionada e refinada pela abordagem ARRANGE. Todos os resultados deste trabalho são detalhados nesta Seção, onde estão divididos em resultados parciais, resultado final e uma subseção detalhando melhor a validação do resultado.

6.1 Resultados Parciais

Inicialmente foi feito o levantamento bibliográfico, a seguir as bases de dados e suas respectivas *strings* de busca. Na ACM, a *string* de busca utilizada foi: *[Title: fairness] AND [Keywords: machine learning] AND [Full Text: non-functional requirement] AND [Full Text: software requirement]*, como resultado foram encontrados 265 trabalhos. Na base de dados Scopus, utilizando a seguinte *string* de busca: *(TITLE-ABS-KEY("FAIRNESS") AND TITLE-ABS-KEY ("MACHINE LEARNING") AND SUBJAREA (comp) AND TITLE-ABS (requirement))*, foram obtidos 129 trabalhos. Na última base de dados consultada, a IEEE*: ("Document Title":FAIRNESS) AND ("Author Keywords":MACHINE LEARNING) AND ("Full Text Only":software requirement)*, foram obtidos 11 trabalhos como resultado. A Figura 11 mostra o resultado de cada passo do processo.

Figura 11 – Fases e resultados da seleção de artigos.

BASE DE DADOS	1ª FASE: CRITÉRIOS BÁSICOS DE EXCLUSÃO	2ª FASE: LER O TÍTULO E RESUMO	3ª FASE: LER PARTES RELEVANTES	4ª FASE: LER O TRABALHO COMPLETO
ACM	187 →	15 →	5 →	4
SCOPUS	84 →	15 →	4 →	3
IEEE	10 →	2 →	2 →	1

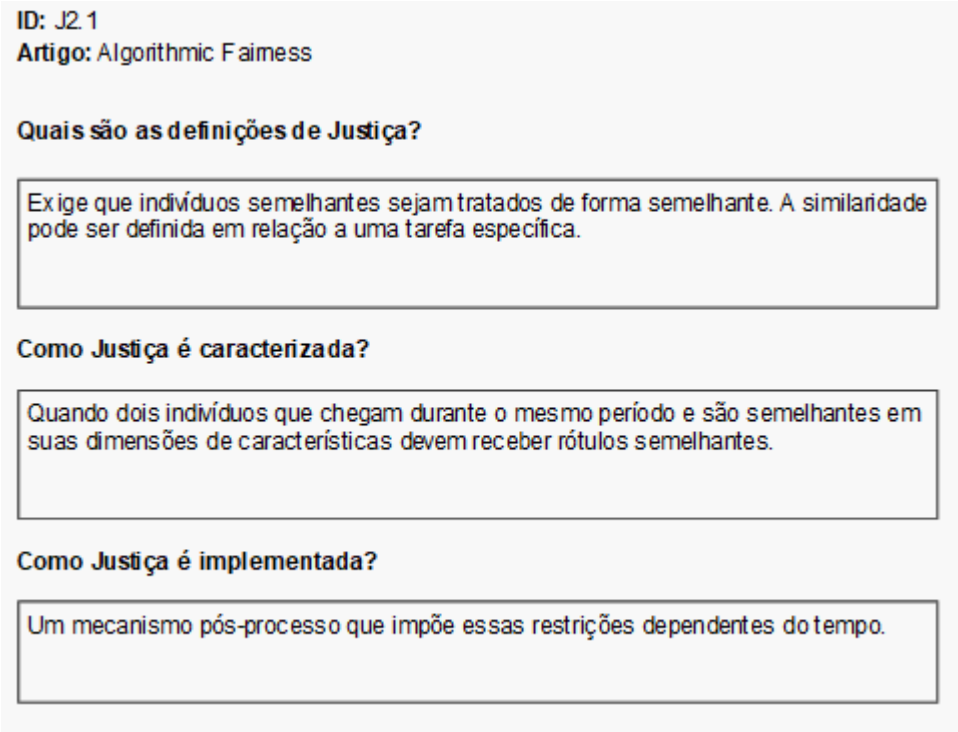
Fonte: Elaborado pelo autor.

Após o levantamento bibliográfico, foram aplicados os critérios básicos de exclusão (ver Quadro 2) nas três bases de dados, na ACM resultou em 187 trabalhos, na Scopus ficaram 84 trabalhos e na IEEE resultou em 10 trabalhos. Na segunda etapa, que consistiu em ler o título e resumo do trabalho, foram selecionados na ACM: 15 trabalhos, na Scopus: 15 trabalhos e na

IEEE: 2 trabalhos. O próximo foi selecionar por partes relevantes, estes foram os resultados na ACM: 5 trabalhos, na Scopus: 4 trabalhos e na IEEE: 1 trabalho. Por fim, na última fase, os resultados foram os seguintes com suas respectivas bases de dados: ACM 4, Scopus 3 e IEEE 1.

Na terceira fase do ARRANGE, a de análise, foi utilizado o conjunto final de 8 artigos, no próximo passo foram extraídos os dados por meio dos formulários de extração (Ver Figura 12), visando responder as questões de pesquisas definidas (Ver Quadro 3)

Figura 12 – Exemplo do formulário de extração utilizado.



Fonte: Elaborado pelo autor.

As informações foram analisadas para compreender como a característica Justiça pode ser subdividida em subcaracterísticas. Deste modo, na codificação aberta, que é a primeira etapa da fase de análise, foram extraídos 49 segmentos de texto, o Quadro 5 mostra a quantidade de segmentos por trabalho, com isso foram gerados 9 códigos, que são detalhados logo abaixo. O Quadro 6 mostra exemplos de segmentos de textos e seus respectivos códigos.

Em (MAKHLOUF *et al.*, 2021) (RYAN *et al.*, 2023) (CHIEN *et al.*, 2022) (BINNS, 2020) (CHOULDECHOVA; ROTH, 2020) Justiça é classificada em duas categorias: Justiça de Grupo e Justiça Individual. Segundo (MAKHLOUF *et al.*, 2021) (RYAN *et al.*, 2023) (CHIEN *et al.*, 2022) (BINNS, 2020) (CHOULDECHOVA; ROTH, 2020) Justiça de Grupo centra-se na equidade e igualdade dos grupos sobre os quais são tomadas decisões e a Justiça Individual que indivíduos semelhantes sejam tratados de forma semelhante.

Quadro 5 – Quantidade de segmentos de texto por trabalho.

Trabalho	**Quantidade de Segmentos de Texto**
(RYAN *et al.*, 2023)	7
(PESSACH; SHMUELI, 2023)	6
(MAKHLOUF *et al.*, 2021)	9
(LI *et al.*, 2023)	10
(SRIVASTAVA *et al.*, 2019)	3
(CHIEN *et al.*, 2022)	7
(BINNS, 2020)	3
(CHOULDECHOVA; ROTH, 2020)	4

Fonte: Elaborado pelo autor.

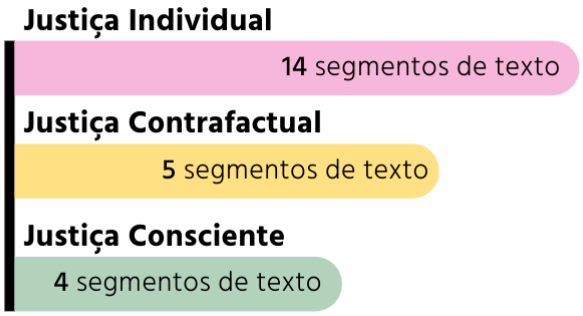
Quadro 6 – Exemplos de segmentos de textos

ID	**Trecho**	**Código**
1	"Uma decisão que relaxe a restrição rígida a qualquer descendente para limitações menos rigorosas."(PESSACH; SHMUELI, 2023)	Justiça Contrafactual
2	"No caso de decisões binárias pode-se pedir taxas iguais de resultados positivos incondicionalmente do resultado verdadeiro."(CHIEN *et al.*, 2022)	Paridade Estatística
3	"Verdadeiros positivos e verdadeiros negativos são igualmente considerados e desejados."(MAKHLOUF *et al.*, 2021)	Igualdade de Precisão Geral
4	"Para o exemplo de contratação, isto implica que a distância entre a distribuição dos resultados de dois candidatos deve ser, no máximo, a distância entre esses candidatos (ex.: nível de escolaridade ou experiência profissional)."(MAKHLOUF *et al.*, 2021)	Justiça Inconsciente
5	"Ambas as subpopulações tem as mesmas métricas, assim a predição é condicionalmente independente do atributo protegido, dado o resultado real."(MAKHLOUF *et al.*, 2021)	Probabilidades Equalizadas

Fonte: Elaborado pelo autor.

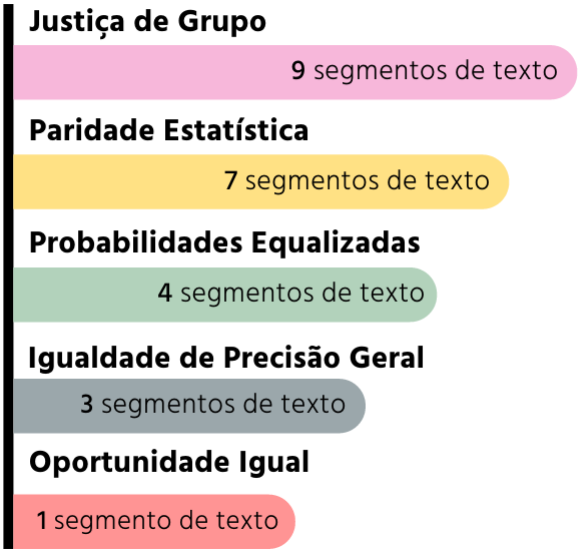
A Figura 13 apresenta os códigos e número de segmentos de textos da primeira subcategoria: Justiça Individual. A Figura 14 apresenta os códigos e o número de segmentos de textos gerados para a segunda subcategoria de Justiça: Justiça de Grupo. Além desses códigos tem mais um que não se enquadra em nenhuma dessas categorias, se chama Justiça Inconsciente que possui 2 segmentos de texto, esse código é independente do indivíduo.

Figura 13 – Justiça Individual - Códigos e sua quantidade de segmentos de texto codificado.



Fonte: Elaborado pelo autor.

Figura 14 – Justiça de Grupo - Códigos e sua quantidade de segmentos de texto codificado.



Fonte: Elaborado pelo autor.

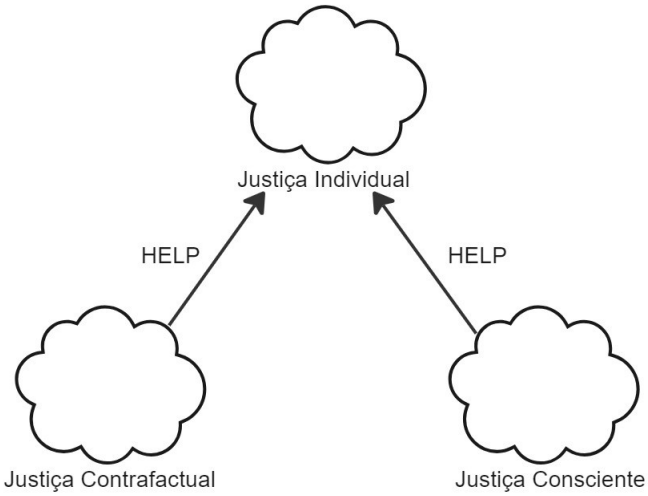
6.2 Resultado Final

A Figura 15 apresenta um exemplo de codificação axial. Os códigos “Justiça Contrafactual” e “Justiça Consciente” são subcaracterísticas para alcançar Justiça Individual. O que significa que a Justiça Individual pode ser alcançada usando esses dois objetivos.

A Figura 16 apresenta a subdivisão da Justiça de Grupo. Os códigos “Probabilidades Equalizadas”, “Paridade Estatística”, "Oportunidade Igual"e "Igualdade de Precisão Geral"são as subcaracterísticas para alcançar a Justiça de Grupo.

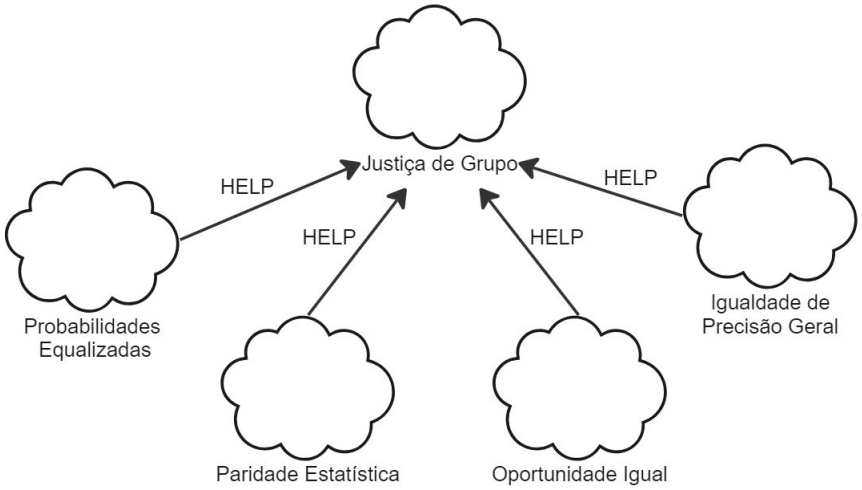
Assim, quando a categoria principal foi relacionada com todos os códigos e categorias, a codificação foi finalizada, onde a categoria principal deste trabalho é Justiça. Dessa maneira, todos os códigos e categorias de códigos estavam relacionados de alguma forma a

Figura 15 – Codificação axial - Justiça Individual



Fonte: Elaborado pelo autor.

Figura 16 – Codificação axial - Justiça de Grupo



Fonte: Elaborado pelo autor.

Justiça.

No presente trabalho, a codificação aberta, axial e seletiva foi realizada por apenas uma pesquisadora. Sendo feita a codificação, associando em categorias, para relacionar as informações, descrevendo os conceitos fundamentados nos dados obtidos sempre.

6.2.1 Validação do SIG

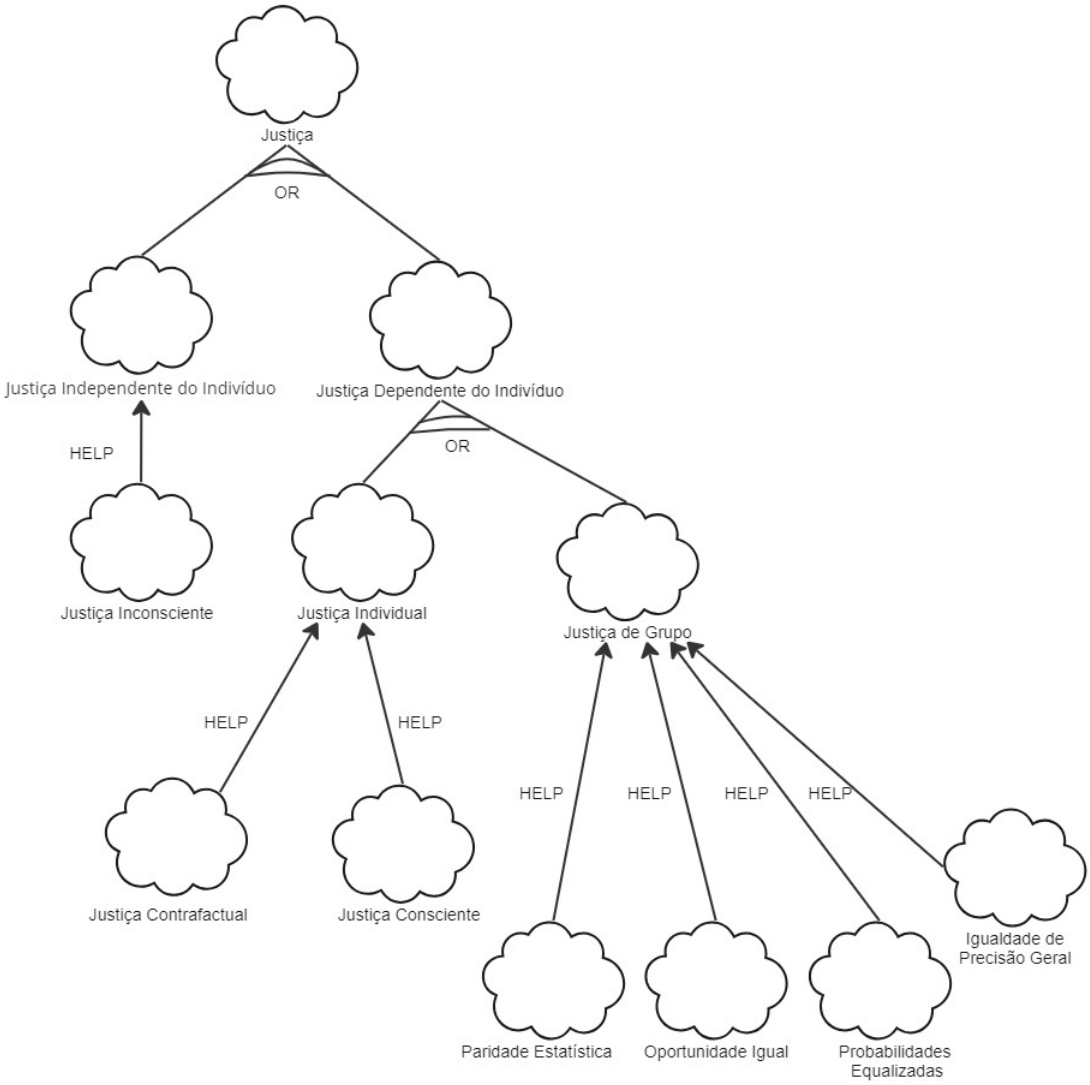
A validação da versão final foi feita por dois especialistas, ambos possuem doutorado em Ciência da Computação, suas áreas de interesse de estudo tem ênfase em Inteligência Artificial e Aprendizado de Máquina. A validação foi feita por meio de reuniões e discussões, além dessas formas, ela pode ser realizada por meio de formulários. Para alcançar o resultado final, que é apresentado logo abaixo, foi necessário criar duas versões até alcançá-lo.

Os especialistas ajudaram na organização e na complementação de informações, com suas sugestões, após avaliarem a primeira versão do SIG, os conceitos e suas definições. Depois da avaliação com os especialistas, foram criadas 2 novas categorias, a fim de organizar melhor o SIG, sendo elas: Justiça Independente do Indivíduo e Justiça Dependente do Indivíduo, assim as subcaracterísticas encontradas após a pesquisa foram divididas entre essas duas categorias.

Justiça Independente do Indivíduo não precisa necessariamente definir o tipo de indivíduo, seja individual ou em grupo, já a Justiça Dependente do Indivíduo necessita dessa definição para selecionar as métricas e soluções específicas de cada categoria (Individual ou Grupo).

Ao final do GT, foi possível criar um SIG de Justiça, contendo 2 categorias, subdivididas em 9 subcaracterísticas. O conhecimento obtido se tornou o núcleo do catálogo de Justiça. A Figura 17 apresenta o SIG de Justiça, principal resultado deste trabalho. O Quadro 7 traz a definição de cada um dos *softgoals*.

Figura 17 – SIG final de Justiça



Fonte: Elaborado pelo autor.

Quadro 7 – *Softgoals*, suas definições e trabalhos em que foram codificados.

Softgoal	**Definição**	**Trabalhos que codificam**
Justiça Individual	Indivíduos semelhantes devem ser tratados de forma semelhante	(RYAN *et al.*, 2023) (PESSACH; SHMUELI, 2023) (MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023) (SRIVASTAVA *et al.*, 2019) (CHIEN *et al.*, 2022) (BINNS, 2020) (CHOULDECHOVA; ROTH, 2020)
Justiça Contrafactual	Baseada em causalidade em nível individual. Requer que, para qualquer indivíduo possível, o resultado previsto do sistema de aprendizagem seja o mesmo no mundo contrafactual e no mundo real	(PESSACH; SHMUELI, 2023) (LI *et al.*, 2023) (CHIEN *et al.*, 2022)
Justiça Consciente	Quaisquer dois indivíduos com características não sensíveis semelhantes devem receber resultados previstos semelhantes	(RYAN *et al.*, 2023) (MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023)
Justiça de Grupo	O seu objetivo comum é garantir que os grupos que diferem pelos seus atributos sensíveis sejam tratados igualmente	(RYAN *et al.*, 2023) (MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023) (CHIEN *et al.*, 2022) (BINNS, 2020) (CHOULDECHOVA; ROTH, 2020)
Probabilidades Equalizadas	Leva em conta a Taxa de Falsos Positivos (FPR) e exige que diferentes grupos tenham a mesma taxa de verdadeiros positivos e de falsos positivos	(MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023) (CHIEN *et al.*, 2022)
Paridade Estatística	Garante que o resultado positivo seja alcançado em proporções iguais por todos os grupos	(RYAN *et al.*, 2023) (SRIVASTAVA *et al.*, 2019) (CHIEN *et al.*, 2022) (MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023)
Oportunidade Igual	A Taxa Verdadeiramente Positiva (TPR) deve ser a mesma em diferentes grupos	(LI *et al.*, 2023)
Igualdade de Precisão Geral	Requer a mesma precisão entre os grupos, comparando todas as métricas básicas	(RYAN *et al.*, 2023) (MAKHLOUF *et al.*, 2021) (LI *et al.*, 2023)
Justiça Inconsciente	Um algoritmo é justo se não considerar atributos sensíveis durante seu processo de tomada de decisão	(RYAN *et al.*, 2023) (CHIEN *et al.*, 2022)

Fonte: Elaborado pelo autor.

7 CONCLUSÕES E TRABALHOS FUTUROS

Cada dia mais sistemas de Aprendizado de Máquina estão sendo adotados, seja para recomendação em *streamings*, assim como para contratar novos funcionários, empréstimos em bancos, entre outros. Com isso é necessário garantir que esses sistemas sejam justos. O conhecimento e experiência em relação ao requisito não-funcional Justiça é relativamente novo, sendo uma nova área de exploração.

Deste modo, o principal objetivo deste trabalho foi capturar subcaracterísticas de Justiça e catalogá-las em um SIG. Foi utilizada uma abordagem chamada ARRANGE, junto com outros métodos como: *grounded theory* (GT) e formulários de extração de dados. Como resultado final, foi possível criar o SIG de Justiça composto por 2 subcategorias, divididas em 9 subcaracterísticas.

O resultado obtido com o método GT é flexível, sendo possível sua evolução com a inclusão de novos conceitos, pois pode-se ter uma compreensão diferente dos dados de acordo com cada pesquisador, onde cada um pode ter sua própria interpretação dos conceitos.

O trabalho foi avaliado por 2 especialistas em Inteligência Artificial e Aprendizado de Máquina, mas seria interessante uma investigação em um contexto de uso real para validar, onde os desenvolvedores o utilizem e os *feedbacks* do uso sejam coletados. Além disso, verificar se o resultado obtido neste trabalho é útil para o desenvolvimento na área de Aprendizado de Máquina em experimentos controlados.

Para alcançar o maior número possível de interessados e expandir o conhecimento aqui adquirido, a disponibilização do catálogo *online* é uma ótima alternativa. Dessa forma, os engenheiros de software, sempre que precisarem, podem consultar o catálogo para os ajudar a tomar decisões em relação ao uso e aplicação desse RNF.

Como trabalhos futuros, o objetivo é dar continuidade na pesquisa e enriquecê-la ainda mais para ajudar a comunidade de IA e Aprendizado de Máquina, através de pesquisas com desenvolvedores e pesquisadores da área, para identificar estratégias de desenvolvimento, assim expandindo o conhecimento e o catálogo SIG, podendo assim avaliar a utilidade do mesmo.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2009.

AMERSHI, S.; BEGEL, A.; BIRD, C.; DELINE, R.; GALL, H.; KAMAR, E.; NAGAPPAN, N.; NUSHI, B.; ZIMMERMANN, T. Software engineering for machine learning: A case study. In: IEEE. **2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. [S.l.], 2019. p. 291–300.

BINNS, R. On the apparent conflict between individual and group fairness. In: **Proceedings of the 2020 conference on fairness, accountability, and transparency**. [S.l.: s.n.], 2020. p. 514–524.

BRUN, Y.; MELIOU, A. Software fairness. In: **Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering**. [S.l.: s.n.], 2018. p. 754–759.

CARVALHO, R. M. **Correlate & lead: process and catalog of non-functional requirements correlations in ubicomp and iot systems**. 2019.

CHAKRABORTY, J.; XIA, T.; FAHID, F. M.; MENZIES, T. Software engineering for fairness: A case study with hyperparameter optimization. **arXiv preprint arXiv:1905.05786**, 2019.

CHIEN, I.; DELIU, N.; TURNER, R.; WELLER, A.; VILLAR, S.; KILBERTUS, N. Multi-disciplinary fairness considerations in machine learning for clinical trials. In: **Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency**. [S.l.: s.n.], 2022. p. 906–924.

CHOULDECHOVA, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. **Big data**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 5, n. 2, p. 153–163, 2017.

CHOULDECHOVA, A.; ROTH, A. A snapshot of the frontiers of fairness in machine learning. **Communications of the ACM**, ACM New York, NY, USA, v. 63, n. 5, p. 82–89, 2020.

CHUNG, L.; NIXON, B. A.; YU, E.; MYLOPOULOS, J. **Non-functional requirements in software engineering**. [S.l.]: Springer Science & Business Media, 2000. v. 5.

CORBIN, J.; STRAUSS, A. **Basics of qualitative research: Techniques and procedures for developing grounded theory**. [S.l.]: Sage publications, 2014.

CORCOVIA, L. O.; ALVES, R. S. Aprendizagem de máquina e mineração de dados: avaliação de métodos de aprendizagem. **Revista Interface Tecnológica**, v. 16, n. 1, p. 90–101, 2019.

DAMYANOVA, B. **Quality attributes in AI-ML-based systems: differences and challenges**. Dissertação (B.S. thesis), 2020.

HABIBULLAH, K. M.; HORKOFF, J. Non-functional requirements for machine learning: Understanding current use and challenges in industry. **arXiv preprint arXiv:2109.00872**, 2021.

HORKOFF, J. Non-functional requirements for machine learning: Challenges and new directions. In: IEEE. **2019 IEEE 27th International Requirements Engineering Conference (RE)**. [S.l.], 2019. p. 386–391.

ISHIKAWA, F.; YOSHIOKA, N. How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey. In: IEEE. **2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)**. [S.l.], 2019. p. 2–9.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015.

KITCHENHAM, B.; CHARTERS, S. *et al.* **Guidelines for performing systematic literature reviews in software engineering**. [S.l.]: UK, 2007.

LI, Y.; CHEN, H.; XU, S.; GE, Y.; TAN, J.; LIU, S.; ZHANG, Y. Fairness in recommendation: Foundations, methods, and applications. **ACM Transactions on Intelligent Systems and Technology**, ACM New York, NY, v. 14, n. 5, p. 1–48, 2023.

LOURENÇO, L. M. **Catálogo de requisitos de calmness para aplicações ubíquas e de internet das coisas**. 2020.

MAKHLOUF, K.; ZHIOUA, S.; PALAMIDESSI, C. Machine learning fairness notions: Bridging the gap with real-world applications. **Information Processing & Management**, Elsevier, v. 58, n. 5, p. 102642, 2021.

MITCHELL, T.; BUCHANAN, B.; DEJONG, G.; DIETTERICH, T.; ROSENBLOOM, P.; WAIBEL, A. Machine learning. **Annual Review of Computer Science**, v. 4, n. 1, p. 417–433, 1990.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. [S.l.]: "O'Reilly Media, Inc.", 2016.

PESSACH, D.; SHMUELI, E. Algorithmic fairness. In: **Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2023. p. 867–886.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and software technology**, Elsevier, v. 64, p. 1–18, 2015.

PONS, L.; OZKAYA, I. Priority quality attributes for engineering ai-enabled systems. **arXiv preprint arXiv:1911.02912**, 2019.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning, 3rd Ed.** 3. ed. Birmingham, UK: Packt Publishing, 2019. ISBN 978-1789955750.

RYAN, S.; NADAL, C.; DOHERTY, G. Integrating fairness in the software design process: An interview study with hci and ml experts. **IEEE Access**, IEEE, v. 11, p. 29296–29313, 2023.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, IBM, v. 3, n. 3, p. 210–229, 1959.

SANTHANAM, P. Quality management of machine learning systems. In: SPRINGER. **International Workshop on Engineering Dependable and Secure Machine Learning Systems**. [S.l.], 2020. p. 1–13.

SAXENA, N. A.; HUANG, K.; DEFILIPPIS, E.; RADANOVIC, G.; PARKES, D. C.; LIU, Y. How do fairness definitions fare? examining public attitudes towards algorithmic definitions of fairness. In: **Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society**. [S.l.: s.n.], 2019. p. 99–106.

SIEBERT, J.; JOECKEL, L.; HEIDRICH, J.; TRENDOWICZ, A.; NAKAMICHI, K.; OHASHI, K.; NAMBA, I.; YAMAMOTO, R.; AOYAMA, M. Construction of a quality model for machine learning systems. **Software Quality Journal**, Springer, p. 1–29, 2021.

SILVA, R. A.; CASTRO, J.; PIMENTEL, J. Catalogando requisitos não-funcionais de sistemas embarcados. **Cadernos do IME-Série Informática**, v. 45, p. 61–79, 2020.

SMOLA, A.; VISHWANATHAN, S. Introduction to machine learning. **Cambridge University, UK**, v. 32, n. 34, p. 2008, 2008.

SRIVASTAVA, M.; HEIDARI, H.; KRAUSE, A. Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning. In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. [S.l.: s.n.], 2019. p. 2459–2468.

VOGELSANG, A.; BORG, M. Requirements engineering for machine learning: Perspectives from data scientists. In: IEEE. **2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)**. [S.l.], 2019. p. 245–251.

WIEGERS, K.; BEATTY, J. **Software requirements**. [S.l.]: Pearson Education, 2013.

ZHANG, J. M.; HARMAN, M.; MA, L.; LIU, Y. Machine learning testing: Survey, landscapes and horizons. **IEEE Transactions on Software Engineering**, IEEE, 2020.