



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

JANIZE MONTEIRO DE CASTILHO

ALGORITMOS PARA SELEÇÃO DE METODOLOGIAS DE
AValiação DE SOFTWARES EDUCACIONAIS

Tucuruí
2023



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

JANIZE MONTEIRO DE CASTILHO

ALGORITMOS PARA SELEÇÃO DE METODOLOGIAS DE
AVALIAÇÃO DE SOFTWARES EDUCACIONAIS

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada – PPCA do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para obtenção do título de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Fabricio de Souza Farias

Tucuruí
2023

Castilho, Janize

ALGORITMOS PARA SELEÇÃO DE METODOLOGIAS DE AVALIAÇÃO DE SOFTWARES EDUCACIONAIS/ JANIZE MONTEIRO DE CASTILHO. – Tucuruí, 2023.

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

Orientador: Prof. Dr. Fabricio de Souza Farias

Dissertação – UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
, 2023.

1. Software Educacional. 2. Algoritmo. 3. Inteligência Artificial. 4. Metodologia. 5. Avaliação. 6. Random Forest. 7. k-Nearest Neighbor. 8. Redes Neurais Artificiais. 9. Processamento de Linguagem Natural. I. Título.

JANIZE MONTEIRO DE CASTILHO

**ALGORITMOS PARA SELEÇÃO DE
METODOLOGIAS DE AVALIAÇÃO DE SOFTWARES
EDUCACIONAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada – PPCA do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para obtenção do título de Mestre em Computação Aplicada.

Data da Defesa: 26 de Setembro de 2023.

Conceito: APROVADO.

Banca Examinadora

Prof. Dr. Fabricio de Souza Farias

Universidade Federal do Pará

Orientador

Prof. Dr. Carlos dos Santos Portela

Universidade Federal do Pará

Membro da Banca



Prof. Dr. Daniel da Conceição Pinheiro

Universidade Federal do Pará

Membro da Banca

Prof. Dr. Adalberto Portilho Costa

Universidade Federal do Pará

Membro da Banca

Tucuruí

2023

Este trabalho é dedicado à minha família.

AGRADECIMENTOS

Agradeço, primeiramente, ao meu Deus por ser o meu refúgio e a minha fonte inesgotável de paz.

Aos meus pais, que acreditam em mim e me dão forças para seguir em frente.

Aos meus irmãos com quem compartilho a vida. Em especial a minha irmã Jéssica, que é minha parceira diária de reflexões e na busca de uma vida virtuosa.

Aos meus tios, primos e cunhada pela confiança e solicitude todos os dias.

Aos meus sobrinhos, que são a força que me motiva a sempre continuar lutando.

Aos meus amigos, com quem divido momentos da vida e que sempre me incentivam a ser melhor.

Aos integrantes do LABEX (Laboratório de Programação Extrema), pelo acolhimento e incentivo, especialmente a Laciene que me ajudou muito no desenvolvimento deste trabalho.

Ao meu orientador, professor Fabricio Farias, pela experiência e ensinamentos a mim repassados e pelo suporte ao longo de toda a realização deste trabalho.

A todos os professores que contribuíram para a minha evolução no mundo do conhecimento.

Ao Programa de Pós-Graduação em Computação Aplicada – PPCA, pela oportunidade de realização de trabalhos na minha área de pesquisa.

*"Feliz o homem que acha sabedoria, e o homem que adquire conhecimento."
(Provérbios: 3,13)*

RESUMO

Com o propósito de auxiliar nos processos de ensino-aprendizagem, muitos professores têm decidido utilizar *Software* Educacional (SE) em suas aulas. No entanto, para escolher um SE como recurso didático é fundamental avaliar a metodologia empregada pelo professor, uma vez que esta precisa ser pedagogicamente e funcionalmente apropriada para suprir as necessidades e objetivos presentes em sala de aula. Além disso, é necessário fazer uso de mecanismos que avaliem o SE para verificar sua adequação aos objetivos do professor. Atualmente, verifica-se que existem diversas técnicas e metodologias disponíveis na literatura para avaliação de SE, no entanto ainda não há uma solução para tomada de decisão e escolha de um SE que atenda plenamente os perfis de usuários e suas diferentes necessidades a serem atendidas por determinada aplicação metodológica, o que gera a demanda oriunda do desenvolvimento de soluções feitas sob demanda e com baixa capacidade de generalização em termos de aplicação prática. Desta forma, são disponibilizadas soluções sem padronização e que diversas vezes não levam em consideração critérios relativos à qualidade, escalas de mensuração e procedimentos de verificação do SE. Essa heterogeneidade dificulta muito a avaliação de um SE, uma vez que a subjetividade na escolha da metodologia de avaliação de SE pode produzir resultados inconclusivos. Diante deste contexto, esse trabalho criou um modelo de qualidade que considera 24 metodologias de avaliação de SE disponíveis na literatura e objetiva automatizar a escolha da metodologia de avaliação de SE a partir da aplicação de algoritmos de inteligência artificial (IA), reduzindo a possibilidade de subjetividade no processo de escolha. Durante a investigação foram utilizados processamento de linguagem natural (PLN), *Random Forest*, *k-Nearest Neighbors* e Redes Neurais Artificiais. Em todos os cenários de pesquisa, o processamento de linguagem natural foi combinado com os demais algoritmos, oferecendo uma solução baseada na aplicação de algoritmos de IA híbridos e fracamente acoplados, com vistas na melhoria dos resultados. Deste modo, sendo realizadas simulações considerando PLN+*Random Forest*, PLN+*k-Nearest Neighbors* e PLN+Redes Neurais Artificiais. Após as simulações, os resultados indicam que é possível determinar a melhor metodologia de avaliação de SE utilizando algoritmos de IA, sendo obtido os melhores resultados com a combinação de PLN+*Random Forest*.

Palavras-chave: software educacional, algoritmo, inteligência artificial, metodologia, avaliação, random forest, k-nearest neighbors, redes neurais artificiais, processamento de linguagem natural.

ABSTRACT

In order to assist the teaching-learning processes, many teachers have decided to use Educational Software (ES) in their classrooms. However, to choose a ES as a teaching resource it is essential to endorse the methodology used by the teacher, once it needs to be pedagogically and functionally appropriate to meet the needs and objectives present in the classroom. Also, it is necessary to use mechanisms that the ES endorses to verify its adequacy to the professor's objectives. Currently, it is verified that there are various techniques and methodologies available in the literature for ES assessment, but there is still no solution for decision making and selection of a ES that fully addresses the profiles of users and their different needs to be met by certain methodological application, or that arises from demand originating from the development of solutions based on demand and with a low capacity for generalization in terms of practical application. In this way, solutions are available without standardization and that several times do not take into consideration criteria relating to quality, measurement scales and verification procedures of the ES. This heterogeneity makes the evaluation of an ES very difficult, since the subjectivity in the selection of ES evaluation methodology can produce inconclusive results. Given this context, this work created a quality model that considers 24 ES assessment methodologies available in the literature and aims to automate the selection of ES assessment methodology based on the application of artificial intelligence (AI) algorithms, reducing the possibility of subjectivity in the screening process. During the investigation we used Natural Language Processing (NLP), Random Forest, k-Nearest Neighbors and Artificial Neural Networks. In all research scenarios, the natural language algorithm was combined with other algorithms, offering a solution based on the application of hybrid and loosely coupled AI algorithms, with excellent results. In this way, simulations were carried out considering NLP+Random Forest, NLP+k-Nearest Neighbors and NLP+Artificial Neural Networks. After the simulations, the results indicate that it is possible to determine the best ES assessment methodology using AI algorithms, with the best results obtained with the combination of NLP+Random Forest.

Keywords: educational software, algorithm, artificial intelligence, methodology, validation, random forest, k-nearest neighbors, artificial neural networks, natural language processing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma dos processos realizados para automatização da escolha da metodologia de avaliação de SE mais adequada.	22
Figura 2 – Processo de desenvolvimento do Modelo de Qualidade Castilho-Farias . . .	47
Figura 3 – Nuvem de palavras da variável de saída desejada com stopwords	55
Figura 4 – Nuvem de Palavras após remoção de stopwords	55
Figura 5 – Exemplo de classificação do KNN com três rótulos de classe e $k=5$	61
Figura 6 – a) Árvore de decisão e b) as regiões de decisão no espaço de atributos x_1 e x_2 .	63
Figura 7 – Passos do algoritmo <i>Random Forest</i>	64
Figura 8 – a) Modelo de neurônio biológico b) Modelo de neurônio artificial.	66
Figura 9 – a) Camada única b) Multicamadas.	67
Figura 10 – Perceptron multi-saídas.	68
Figura 11 – Rede Neural <i>Perceptron</i> Multi-camada.	68
Figura 12 – Matriz de Confusão.	69
Figura 13 – Código para remoção de <i>stopwords</i>	71
Figura 14 – Código para realizar o pré-processamento.	71
Figura 15 – Vetor de palavras com a base de dados.	72
Figura 16 – Matriz da frequência que as palavras aparecem nos registros.	72
Figura 17 – Matriz dos valores dos pesos de cada palavra.	73
Figura 18 – Código definição da variável de saída e divisão da base de dados.	74
Figura 19 – Relação da Acurácia X Valor de k	74
Figura 20 – Matriz de Confusão KNN	75
Figura 21 – Matriz de Confusão RNA	77
Figura 22 – Taxa de erro OOB de acordo com o número de árvores na <i>Random Forest</i> . .	78
Figura 23 – Matriz de Confusão <i>Random Forest</i>	79

LISTA DE TABELAS

Tabela 1 – Sumarização dos itens considerados nos estudos	31
Tabela 3 – Normas de qualidade de produtos e processos de desenvolvimento de <i>softwares</i>	43
Tabela 4 – Critérios Socioculturais	48
Tabela 5 – Critérios Socioeconômicos	49
Tabela 6 – Critérios Pedagógicos	50
Tabela 7 – Critérios Técnicos	51
Tabela 8 – Amostra de registros da tabela criada com os dados	53
Tabela 9 – Vantagens e desvantagens do <i>K-Nearest Neighbor</i>	62
Tabela 10 – Vantagens e desvantagens do <i>Random Forest</i>	65
Tabela 11 – Medidas de desempenho do algoritmo <i>k-Nearest Neighbors</i>	75
Tabela 12 – Medidas de desempenho do algoritmo Redes Neurais Artificiais	76
Tabela 13 – Medidas de desempenho do algoritmo <i>Randon Forest</i>	79

LISTA DE ABREVIATURAS E SIGLAS

ABES	Associação Brasileira das Empresas de <i>Software</i>
ABNT	Associação Brasileira de Normas Técnicas
ACC	<i>Accuracy</i>
ACM	<i>Association for Computing Machinery</i>
APP	Aplicativo
ASE	Avaliação de <i>Software</i> Educativo
CASE	<i>Checklists for the Evaluation of Educational Software</i>
CE	Critérios de Exclusão
CI	Critérios de Inclusão
CLATES	Centro Latino-Americano de Tecnologia Educacional
CNE	Conselho Nacional de Educação
ENEM	Exame Nacional do Ensino Médio
FASE	Ferramenta Especialista para Avaliação de <i>Software</i> Educacional
FN	<i>False Negative</i>
FNS	<i>False Negative Rate</i>
FP	<i>False Positive</i>
FPR	<i>False Positive Rate</i>
FRASE	FRAmework de Suporte ao Ensino
GP	Jogabilidade
GU	Usabilidade do Jogo
IA	Inteligência Artificial
IAQSEM	Instrumento de Avaliação da Qualidade para <i>Software</i> Educacional de Matemática
IDEB	Índice de Desenvolvimento da Educação Básica
IDF	<i>Inverse Document Frequency</i>

IEC	<i>International Electrotechnical Commission</i>
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
ISO	<i>International Organization for Standardization</i>
K-NN	<i>K-Nearest Neighbor</i>
LC	Conteúdo de Aprendizagem
LDB	Lei de Diretrizes e Bases da Educação
LORI	<i>Learning Object Review Instrument</i>
MAP	<i>Mean Absoluto Error</i>
MAEP	Método Ergopedagógico Interativo de Avaliação para Produtos Educacionais Informatizados
MAJE	Modelo de Avaliação de Jogos Educacionais
MAQSE	Metodologia para Avaliação da Qualidade de <i>Software</i> Educacional
MEC	Ministério da Educação
MEEGA	<i>Model for the Evaluation of Educational Games</i>
MLP	<i>Multi Layer Perceptron</i>
MO	Mobilidade
MOLEVA	<i>Mobile Learning Evaluation</i>
MSL	Mapeamento Sistemático da Literatura
NBR	Norma Brasileira
NUTES	Núcleo de Tecnologia Educacional para a Saúde
OOB	<i>Out-of-bag</i>
PBL	<i>Problem-based Learning</i>
PETESE	<i>Pedagogical Ergonomic Tool for Educational Software Evaluation</i>
PLN	Processamento de Linguagem Natural
PRO	Precisão
PROINFO	Programa Nacional de Tecnologia Educacional

PRONINFE	Programa Nacional de Informática Educativa
RNA	Redes Neurais Artificiais
SAEB	Sistema de Avaliação da Educação Básica
SE	<i>Software</i> Educacional
SEN	<i>Sensitivity</i>
SEP	<i>Specificity</i>
TACASE	Taxonomia de critérios para avaliação de <i>Software</i> Educativo
TF	<i>Term Frequency</i>
TIC	Tecnologia da Informação e Comunicação
TICESE	Técnica de Inspeção Ergonômica de <i>Software</i> Educacional
UFRJ	Universidade Federal do Rio de Janeiro
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Contexto e Motivação	17
1.2	Justificativa	18
1.3	Objetivos	21
1.3.1	Objetivo Geral	21
1.3.2	Objetivos Específicos	21
1.4	Metodologia	22
1.5	Organização	23
2	MAPEAMENTO SISTEMÁTICO	24
2.1	Definição de Questões de Pesquisa	24
2.2	Definição da Estratégia de Busca	24
2.3	Construção da <i>String</i> de Busca	24
2.4	Definição de Questões de Pesquisa	25
2.5	Seleção dos Estudos	25
2.6	Resultado do Levantamento Sistemático	25
2.7	Considerações Finais	32
3	TRABALHOS RELACIONADOS	33
3.1	Métodos para Avaliação de SE	33
3.1.1	Método de Reeves	33
3.1.2	Técnica de TICESE	33
3.1.3	Técnica de Mucchielli	34
3.1.4	Método de LORI - <i>Learning Object Review Instrument</i>	34
3.1.5	MAQSEI - Metodologia para Avaliação da Qualidade de <i>Software</i> Educa- cional Infantil	34
3.1.6	FASE - Ferramenta Especialista para Avaliação de <i>Software</i> Educacional	35
3.1.7	Método de Savi - Modelo de Avaliação de Jogos Educacionais	35
3.1.8	PETESE - <i>Pedagogical Ergonomic Tool for Educational Software Evaluation</i>	35
3.1.9	MAQSE - Metodologia para Avaliação da Qualidade de <i>Software</i> Educacional	35
3.1.10	Método Rocha	36
3.1.11	IAQSEM - Instrumento de Avaliação da Qualidade para <i>Software</i> Educacional de Matemática	36
3.1.12	MAEP - Método Ergopedagógico Interativo de Avaliação para Produtos Educação Informatizados	36
3.1.13	Método Rodrigues - Instrumento para Avaliação de Jogos Eletrônicos Educa- tivos do Ensino Fundamental I	37
3.1.14	TUP Model - <i>Evaluation Model Technology, Usability and Pedagogy</i>	37
3.1.15	Método ASE - Método de Avaliação de <i>Software</i> Educativo	37

3.1.16	CASE - <i>Checklists for the Evaluation of Educational Software</i>	38
3.1.17	MEEGA+ - <i>Model for the Evaluation of Educational Games</i>	38
3.1.18	MoLeVa - <i>Mobile Learning Evaluation</i>	38
3.1.19	Questionário de Silva e Batista	39
3.1.20	<i>Checklist</i> de Porto	39
3.1.21	<i>Checklist</i> de Soad	39
3.1.22	Avaliação Ergonômica-Pedagógica	40
3.1.23	<i>Heuristic Evaluation Strategy</i>	40
3.1.24	Método de Lulia e Simões	40
4	REFERENCIAL TEÓRICO	41
4.1	Software Educacionais	41
4.2	Qualidade de Software	42
4.3	Modelo de Qualidade	44
4.3.1	Modelo de Qualidade do MoLEva	45
4.3.2	Modelo de Qualidade do TACASE	46
5	PROPOSIÇÃO DO MODELO DE QUALIDADE (CASTILHO-FARIAS)	47
6	ETAPAS PARA DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS	54
6.1	Aquisição de Dados	54
6.2	Análise Exploratória de Dados	54
6.3	Pré-processamento de Dados	54
6.3.1	Limpeza dos dados	55
6.3.2	Integração dos dados	56
6.3.3	Transformação dos dados	56
6.3.3.1	Redução dos dados	56
6.4	Processamento de Linguagem Natural	57
6.4.1	Algoritmos de Vetorização <i>Embedding</i>	57
6.4.1.1	TF-IDF	58
6.5	Inteligência Artificial Aplicada	59
6.5.1	Modelo <i>K-Nearest Neighbor</i>	59
6.5.2	Modelo <i>Random Forest</i>	62
6.5.3	Modelo Redes Neurais Artificiais	65
6.6	Métricas de Desempenho	68
6.6.1	Matriz de Confusão	69
7	ANÁLISE DOS RESULTADOS	71
7.1	Desempenho dos Algoritmos	74
7.1.1	Algoritmo <i>k-Nearest Neighbors</i>	74
7.1.2	Algoritmo Redes Neurais Artificiais	76
7.1.3	Algoritmo <i>Random Forest</i>	77

8	CONCLUSÃO	80
8.1	Dificuldades Encontradas	81
	REFERÊNCIAS	82
	APÊNDICE A – LISTA COM A DEFINIÇÃO DOS CRITÉRIOS . . .	92
	APÊNDICE B – <i>SCRIPTS DA APLICAÇÃO DOS ALGORITMOS</i> . . .	110

1 INTRODUÇÃO

1.1 Contexto e Motivação

O crescimento da utilização de aplicativos móveis (*app*) em nossa sociedade é um caminho sem precedentes e sem volta. Atualmente, os usuários de *apps* se acostumaram a resolver diversas tarefas em seus *smartphones* e *tablets*, por exemplo, chamar um carro para transporte de passageiros (PAYPAL-CORP, 2022), fazer compras ou efetuar transações bancárias (PAYPAL-CORP, 2021), pesquisar sobre uma notícia (MATTOS, 2021), participar de reuniões *online* (CAMARGO; CHAVES, 2023; RIBEIRO; WELLS, 2021), e até mesmo aprender sobre determinado assunto a partir de vídeos ou tutoriais disponíveis em incontáveis sites distribuídos por todo o mundo (NIC.BR, 2022; STATISTA, 2023).

No Brasil, existe uma massiva tendência no uso de *apps*. De acordo com os dados do relatório *State of Mobile* de 2022 (DATA.AI, 2022), o Brasil ocupa a 1ª colocação no *ranking* dos países em relação ao tempo de uso diário de *apps*, obtendo uma média de 5h e 24 minutos por dia. Também segundo o relatório, em 2021 os brasileiros realizaram mais de 10.326 bilhões de *downloads* de *apps*. Deste modo, tornando o país em um dos maiores e mais aquecidos mercados para se trabalhar na indústria de *software* e em atividades de vendas de produtos de Tecnologia da Informação e Comunicação (TIC).

De acordo com o relatório da Associação Brasileira das Empresas de *Software* (ABES), em 2021 a produção mundial de tecnologia da informação (TI) teve um crescimento de 11%, enquanto no Brasil o crescimento alcançou a marca de 17,4 % (ABES, 2022). Tal momento favorável resulta na produção de diversos *apps* com intuito de atender o crescente número de usuários de TICs, o que gera o desafio de se entregar soluções com qualidade e confiabilidade para seus usuários finais, sendo esta necessidade intensificada quando se diz respeito ao desenvolvimento de *softwares* que serão adotados como ferramentas de apoio ao processo de ensino e aprendizagem, gerando desafios e questionamentos sobre qualidade, eficácia no ensino, e também em relação a metodologia de ensino que mais se adequa para determinada ferramenta proposta pelos desenvolvedores.

Para atender as demandas técnicas e pedagógicas presentes no exercício de ensinar via uma ferramenta baseada em *software*, se faz necessário levar em consideração o fato de que a produção de soluções voltadas ao propósito de ensinar devem ter caráter multidisciplinar, sendo observadas por vários olhares, isto é, desde o olhar do aluno, do pedagogo e dos professores, até o olhar dos desenvolvedores e *designs* da solução (PEREIRA et al., 2016). Neste contexto desafiador, e no melhor do conhecimento da autora, não há uma solução para tomada de decisão e escolha de um *software* educacional (SE) que atenda plenamente todos os perfis de usuários e suas diferentes necessidades a serem atendidas por determinada aplicação metodológica, o que gera a demanda oriunda do atendimento precário a partir do desenvolvimento de soluções

feitas sob demanda e com baixa capacidade de generalização em termos de aplicação prática (ECONOMIDES, 2008; KEARNEY et al., 2012; ABDURRAHMAN; BEER; CROWTHER, 2015; SOAD; BARBOSA, 2017).

1.2 Justificativa

No Brasil o direito a educação é estabelecido na Constituição Federal e na Lei de Diretrizes e Bases da Educação (LDB) (BRASIL, 2010; BRASIL, 1996). Além do direito à educação tais Leis determinam a garantia de padrão de qualidade do ensino. O Conselho Nacional de Educação (CNE), orienta sobre a qualidade do ensino, por meio do Parecer nº 5/2020. Esse parecer relata que o ensino deve ser ministrado com garantia de padrão de qualidade, visando o pleno desenvolvimento do educando e seu preparo para o exercício da cidadania (BRASIL, 2020).

Uma forma de verificar se há o atendimento dos padrões de qualidade esperados é por meio de avaliação. Para isso, o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) avalia por meio de exames nacionais os estudantes brasileiros, desse modo, construindo uma base de dados capaz de mapear os problemas e sucessos dos estudantes, professores e escolas. De modo complementar, desde 2007, o INEP passou a socializar livremente os resultados das pesquisas estatísticas e indicadores educacionais oriundos do Índice de Desenvolvimento da Educação Básica (Ideb) ¹ oportunizando o acesso a resultados relevantes que tratem sobre o fluxo escolar e médias de desempenho estudantil nas avaliações.

Em virtude da divulgação anual do Ideb, observou-se a intensificação do debate entre pesquisadores da área e responsáveis pela criação de políticas educacionais sobre quais características seriam determinantes ou contribuíam para explicar o diferencial de qualidade entre as escolas (DIAS, 2021; LIMA, 2023).

Em Dias (2021) o autor investigou no estado do Ceará as escolas que mais melhoraram seus índices no Ideb, entre os anos de 2007 e 2019, e mapeou quais foram os métodos educativos que predominavam as salas de aula dessas instituições de ensino, assim identificando os fatores que mais colaboraram para a melhoria nos índices educacionais. Identificou-se que o secretário municipal de educação, o diretor e o coordenador pedagógico como atores fundamentais na implementação das iniciativas em educação nos municípios e nas escolas públicas dos anos iniciais. Durante a pesquisa, os autores entrevistaram os gestores das escolas cearenses e descobriram quais métodos de monitoramento e gestão estavam sendo adotados de forma exitosa. Com a descoberta das estratégias de sucesso, foi considerado como trabalho futuro a implantação dos métodos em outros ambientes escolares que ainda não são contemplados.

¹ O Ideb é um indicador sintético que relaciona as taxas de aprovação escolar, obtidas no Censo Escolar, com as médias de desempenho em língua portuguesa e matemática dos estudantes no Sistema de Avaliação da Educação Básica (Saeb). À vista disso, apresentam melhores resultados no Ideb os sistemas que alcançam, de forma concomitante, maiores taxas de aprovação e proficiência nas avaliações (INEP, 2022).

Embora o Ideb seja uma ferramenta validada para medição do desempenho dos alunos, tal meio de análise não leva em consideração a inserção das TICs como caminhos para o aperfeiçoamento da aprendizagem. Nesse contexto, sabemos que as TICs fazem parte do dia a dia da juventude, fato que não permite ser negligenciada como meio de acesso a informação e conhecimento. Por essa razão, Sobrinho e Rivera (2021) destacam a importância do uso das TICs como ferramentas de suporte ao ensino, sendo indissociáveis do ambiente escolar.

Diversos estudos tem investigado a adoção de TICs em ambientes escolares. Sanches et al. (2019) avaliaram 26 (vinte e seis) alunos voluntários com a aplicação de um pré-teste sobre conhecimentos em matemática e um pós-teste em 13 (treze) alunos após a utilização da metodologia PBL+*M-learning* e nos outros 13 (treze) alunos após a utilização do método de ensino tradicional. Os autores demonstraram que a metodologia PBL+*M-learning* tem um melhor resultado em relação ao método de ensino tradicional. Os resultados encontrados na investigação reforçam a importância do contato dos estudantes com as TICs, assim como, indicam que alunos em vulnerabilidade social e econômica precisam ser aparados pelas escolas para que seja evitada a defesagem em sua aprendizagem por conta de pouco ou nenhum acesso as TICs.

Embora seja verdadeira e legítima a necessidade de garantia de infraestrutura escolar universal na forma de acesso à *Internet* e TICs, Cunha et al. (2020) destacam em seu estudo que ainda existem diversas deficiências presentes na infraestrutura de escolas, as quais são enfrentadas por muitos alunos quando necessitam utilizar equipamentos de TICs para terem acesso à *Internet*. Deste modo, superar tais problemas é importante para que os atores envolvidos no processo de ensino e aprendizagem possam alcançar índices satisfatórios de aprendizagem, o que refletirá em seus desempenhos durante as aplicações de exames de avaliação, tais como Ideb ou exame nacional do ensino médio (ENEM).

Além da infraestrutura, diversos autores propõem metodologias que buscam analisar características de *softwares* voltados ao ensino, no entanto se limitam a análise de atributos limitados, por exemplo, Gurgel et al.(2013) averiguaram em que grau a interface do programa facilita a aprendizagem do estudante. Para isso, avaliaram a simplicidade de uso, o conteúdo e a organização, a fim de verificar se o SE contribuía significativamente no processo de ensino e aprendizagem (GURGEL; AGUIAR; SILVA, 2013). No entanto, avaliar somente estes atributos não contribuem para a escolha do SE ideal para aplicação em sala de aula pelo professor, uma vez que vários outros atributos necessitam ser considerados; Mocbel et al. (2020) propuseram o Framework de Suporte ao Ensino (FRASE), que objetiva auxiliar o desenvolvimento de aplicações *M-learning* considerando requisitos de interface de usuário, aprendizagem, organização e satisfação, baseando-se em critérios aderentes às características propostas na norma ISO/IEC 25000 (2014), entre outras (MOCBEL et al., 2020). Desta forma, o autor propõe mais uma solução que avaliará atributos limitados, disponibilizando mais soluções que não conseguem selecionar um SE considerando todos os atributos essenciais para sua utilização em sala de aula. Com tamanha diversidade de opções, selecionar a que mais se adequa a tarefa de ensino e aprendizagem se

torna um exercício difícil e que deve ser levado em consideração durante o desenvolvimento de uma nova solução. Por essa razão, a adoção de um SE requer cautela, em virtude da existência de distintos fatores que podem atrapalhar o processo de ensino e aprendizagem, dentre os quais destacam-se:

- (i) a implantação de um SE no contexto incorreto;
- (ii) incoerência entre o conteúdo tratado no SE e aquele planejado pelo professor;
- (iii) problemas de usabilidade na interface que possam afetar no processo interativo tornando a utilização do SE desagradável ou complexo;
- (iv) conteúdo didático mal estruturado ou mal representado;
- (v) fundamentação didática e pedagógica insuficiente ou ausente no SE; e
- (vi) problemas funcionais do SE que possam acarretar ocorrências não desejadas (SILVA, 2012; FRANÇA; SILVA, 2014; SILVA; GOMES, 2015).

Devido à existência de fatores que podem atrapalhar a utilização do SE, existe a necessidade de utilizar formas de análise e avaliação do SE, buscando as características e objetivos desejáveis. Dada a importância do tema e considerando a diversidade de SE disponíveis e que buscam apoiar alunos e professores no ensino e aprendizagem, é importante que a escolha desse SE leve em consideração critérios que permitam ao aluno e professor o desenvolvimento eficiente de suas habilidades, isto é na forma de meio facilitador do processo de ensino e aprendizagem, assim promovendo o acesso a conhecimentos que ofereçam a possibilidade da melhoria do desempenho dos alunos (ANDRADE; JR; SILVEIRA, 2017).

Um dos fatores que dificulta a criação de soluções que respeitem os critérios necessários para a oferta de ensino e aprendizagem de qualidade via *software* diz respeito aos atores envolvidos na criação das soluções, por exemplo, Da Silva et al. (2016) constataram que o processo avaliação para criação de um SE é majoritariamente realizado por *designers*; professores especialistas de uma área específica; e por alunos usuários das tecnologias. Tendo os diferentes olhares técnicos, características distintas em relação aos critérios considerados como fundamentais aos instrumentos de mensuração da qualidade dos produtos produzidos, levando a entrega de soluções que não atendem de forma plena o propósito de ensinar (SILVA; NETTO; SOUZA, 2016).

Deste modo, a falta de padronização resulta em diversos problemas como: indefinição dos perfis das pessoas envolvidas; necessidade de excessivas habilidades para operação do *software* avaliado; alta variedade de critérios, escalas de avaliação e tipos de avaliação; conhecimentos preexistentes distintos; inexistência de familiaridade com os critérios que serão examinados (GLADCHEFF; ZUFFI; SILVA, 2001); e pouco conhecimento sobre as características da avaliação (Educação, Engenharia de *Software* e Interação Humano-Computador) (WEBBER; BOFF; BONO, 2009).

Neste contexto, antes de se desenvolver ou aplicar qualquer solução baseada em *software*, se faz necessário compreender as reais necessidades metodológicas que devem ser supridas pela adoção da ferramenta, e para alcançar o desejado bom desempenho dos alunos em sala de aula se faz necessário planejar e aplicar uma metodologia de ensino e aprendizagem adequada. Deste modo, quando definida a adoção de SE no processo de ensino e aprendizagem se faz necessário a aplicação de uma etapa apropriada para seleção dos aplicativos que deverão ser adotados nas atividades de ensino (ANDRADE; JR; SILVEIRA, 2017). Com isso, é necessário conhecer quais objetivos que se pretende atingir durante o ensino, a fim de realmente se garantir a escolha de uma solução de SE que cause um impacto significativo na aprendizagem de determinado grupo de alunos. Muito se fala sobre como criar e como testar SE. Mas quase nada se diz sobre (1) qual é a melhor metodologia para avaliar e escolher o melhor SE para determinado cenário. (2) Qual é a melhor ferramenta de *software* para selecionar o SE mais adequado para o meu contexto? Esses são os desafios que esta pesquisa busca responder eliminando a subjetividade do processo.

1.3 Objetivos

1.3.1 Objetivo Geral

Automatizar o processo de seleção da metodologia de avaliação de *software* educacional utilizando inteligência artificial.

1.3.2 Objetivos Específicos

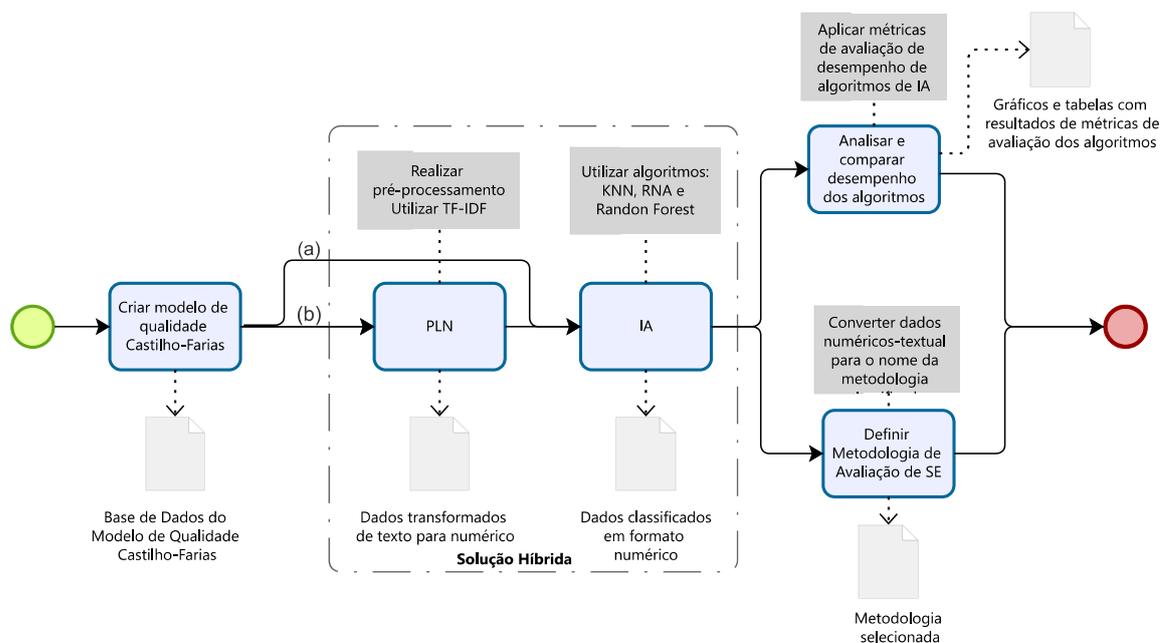
- Avaliar as lacunas presentes nos trabalhos da literatura em relação as metodologias para seleção de *software* voltados ao ensino e aprendizagem.
- Levantar métodos de avaliação para identificar técnicas utilizadas em sistemas de *software* educacional;
- Mapear os critérios adotados pelos métodos para avaliação de SE com os dos modelos de Qualidade de *Software*, de Uso e Pedagógica; e
- Propor um modelo denominado Castilho-Farias para criação de base de dados composta por 1037 (um mil e trinta e sete) registros que representam a relação entre 24 (vinte e quatro) metodologias, seu critérios e os perfis dos usuários avaliadores.
- Utilizar a base de dados oriunda do modelo Castilho-Farias como dado de entrada nos algoritmos de IA para determinação da melhor metodologia de seleção de SE.
- Investigar o desempenho de diferentes algoritmos híbridos fracamente acoplados para a tarefa de classificação, selecionando a melhor metodologia de avaliação de SE, confirme critérios dos usuários.

1.4 Metodologia

Esta seção descreve as etapas que foram realizadas para o alcance dos objetivos desta pesquisa. A Figura 1, ilustra o processo de desenvolvimento realizado para automatização do processo de seleção de metodologia de avaliação de SE utilizando IA. O processo é composto pelas seguintes etapas: (i) Criar um modelo de qualidade Castilho-Farias; (ii) Aplicar técnicas de processamento de linguagem natural (PLN); (iii) Aplicar técnicas de IA, (iv-1) Analisar e comparar o desempenho dos algoritmos e (iv-2) Definir metodologia de avaliação de SE.

A primeira etapa considera criar um modelo de qualidade que será entrada para a fase de aplicação de algoritmos de IA para automatização do processo de seleção da metodologia de avaliação de SE. Esta fase esta descrita no Capítulo 5.

Figura 1 – Fluxograma dos processos realizados para automatização da escolha da metodologia de avaliação de SE mais adequada.



Fonte: elaborada pela autora.

Conforme apresentado na Figura 1, são possíveis dois caminhos, (a) e (b) para realizar a segunda etapa do processo. No entanto, serão explanados os procedimentos realizados em cada etapa do fluxograma para o caminho (b), que foi o procedimento realizado que alcançou melhores resultados. O caminho (a) foi testado utilizando a técnica de *Label Encoding*, que consiste em um método que atribui cada categoria a um número inteiro único, no entanto os resultados não foram satisfatórios e não serão explanados na dissertação.

A segunda etapa, consiste na utilização de uma solução híbrida, onde os dados do modelo de qualidade Castilho-Farias são aplicados como entrada do bloco PLN, onde são aplicadas técnicas de pré-processamento na base de dados, incluindo a utilização da técnica de PLN TF-IDF. Após a obtenção dos dados no formato e adequados para utilização como entrada dos

algoritmos de IA, foram aplicados os algoritmos de classificação KNN, RNA e *Random Forest*, gerando o resultado da classificação.

Os dados de saída dos algoritmos de IA então foram utilizados como entrada para a etapa de análise e comparação do desempenho dos algoritmos, onde foram utilizadas métricas de avaliação para a comparação e obtidos gráficos e tabelas com as métricas de comparação. Estes dados também foram utilizados como entrada da etapa de definição da metodologia de avaliação de SE, onde foram aplicadas técnicas de transformação numérico-textual nos dados e então apresentado ao usuário o nome da metodologia selecionada.

Todas estas etapas, são explicadas em detalhes nos Capítulos 5, 6 e 7.

1.5 Organização

Este trabalho está estruturado em oito capítulos: Introdução, Mapeamento Sistemático, Trabalhos Relacionados, Referencial Teórico, Proposição do Modelo de Qualidade, Procedimentos Metodológicos, Análise dos Resultados e Conclusão.

O Capítulo 1: Introdução - destina-se a contextualizar o trabalho sucintamente, descreve o problema encontrado e qual a solução proposta, expõe, ainda, os objetivos gerais e específicos a serem atingidos com esta pesquisa.

O Capítulo 2: Mapeamento Sistemático – destina-se a apresentar o Mapeamento Sistemático da Literatura, descrevendo todas as etapas;

O Capítulo 3: Trabalhos Relacionados – destina-se a apresentar as metodologias de avaliação de SE que foram selecionadas e serão utilizadas no trabalho;

O Capítulo 4 Referencial Teórico – Apresenta conceitos e definições utilizadas no trabalho;

O Capítulo 5: Proposição do Modelo de Qualidade(Castilho-Farias) - apresenta como o modelo de qualidade proposto foi concebido;

O Capítulo 6:Procedimento Metodológico - apresenta como foi executada a metodologia para a coleta de dados e para a extração do conhecimento utilizado na pesquisa, explicitando como os algoritmos proposto foram implementados e como foram avaliados.

O Capítulo 7: Análise dos Resultados – apresenta os resultados alcançados na pesquisa; e

Por fim, o Capítulo 8: Conclusão – destina-se a destacar as considerações obtidas durante o desenvolvimento do trabalho.

2 MAPEAMENTO SISTEMÁTICO

O Mapeamento Sistemático da Literatura (MSL) é um método que realiza o levantamento dos estudos catalogados, possibilitando a compreensão do estado da arte em uma determinada área do conhecimento, a fim de direcionar e fornecer uma quantidade de fatos identificados sobre mesmo (JONES et al., 2016). O MSL realizado nesta pesquisa segue seis etapas fundamentais a serem seguidas: 1. Definição de questões de pesquisa; 2. Definição da estratégia de busca; 3. Construção da *string* de busca; 4. Definição de critérios de inclusão e exclusão; 5. Seleção dos estudos; e 6. Resultado do levantamento sistemático.

2.1 Definição de Questões de Pesquisa

Considerando a pluralidade de perspectivas de avaliadores de SE (SILVA; NETTO; SOUZA, 2016); a grande variedade de critérios e escalas de mensuração na avaliação dos SE (GLADCHEFF; ZUFFI; SILVA, 2001); e as necessidades de conhecimentos prévios e aprofundados sobre as características de avaliação (WEBBER; BOFF; BONO, 2009), buscou-se entender como os SE são selecionados para apoiar no ensino e aprendizagem de alunos e professores, desconsiderando a área de conhecimento e nível escolar, isto é série, dos usuários. Para isso, foram definidas as seguintes Questões de Pesquisa (QP):

QP1: Como é feita a seleção de *apps* educacionais para utilização no processo de ensino e aprendizagem?

QP2: A forma de seleção de *apps* atualmente disponível é eficaz para escolha de um app de suporte ao ensino e aprendizagem?

QP3: Como é feita a avaliação dos *apps* educacionais?

2.2 Definição da Estratégia de Busca

O objetivo do MSL é obter o estado da arte das abordagens usadas para selecionar e avaliar aplicativos educacionais, entre 2016 e 2021. O levantamento aconteceu entre os meses de fevereiro e abril de 2021, sendo aplicado na biblioteca de busca do *Google Scholar* e na *Association for Computing Machinery* (ACM).

2.3 Construção da *String* de Busca

Para definir a *string* de busca do mapeamento, foram utilizadas as palavras-chaves e sinônimos relacionados ao tema da pesquisa. Desta forma, obtiveram-se as seguintes palavras-chave: Avaliação, Qualidade, Jogos educacionais, Aplicativos educacionais, *Software* Educacionais,

Aprendizagem móvel. A *string* de busca final resultou em: (“*software* educacionais” OR “aplicativo educacional” OR “jogos educacionais” OR “aprendizagem móvel” OR “*educational software*” OR “*mobile learning*” OR “*m-learning*” OR “*educational apps*”) AND (“qualidade” OR “*quality*”) AND (“avaliação” OR “*evaluation*”).

2.4 Definição de Questões de Pesquisa

A fim de filtrar os estudos relacionados pela *string* de busca e selecionar os estudos que atendiam os objetivos da pesquisa, foram adaptados de Soad e Barbosa (2017) e Alencar e Marques (2021) os Critérios de Inclusão (CI) e os Critérios de Exclusão (CE):

CI1: Os artigos devem estar disponíveis na *web*;

CI2: Os artigos devem apresentar textos completos dos estudos em formato eletrônico;

CI3: Os artigos devem estar descritos em português;

CI4: Os artigos devem ter ano de publicação a partir de 2016;

CE1: Artigos repetidos;

CE2: Artigos que não estão relacionados a proposta de seleção de *softwares* educacionais.

2.5 Seleção dos Estudos

Foram selecionadas as bases de dados *Google Scholar* (<https://scholar.google.com.br/>) e a plataforma ACM (<https://dl.acm.org/>) por já serem usadas em artigos como o de Alencar e Marques (2021) e o de D’Carlo et al.(2016). Após o uso da *string* de busca no *Google Scholar* foram retornados 8.740 (oito mil, setecentos e quarenta) estudos e na plataforma ACM foram retornados 431 (quatrocentos e trinta e um) estudos. Para a seleção dos artigos, foram feitas as leituras dos títulos, dos resumos e palavras-chaves dos 300 (trezentos) artigos mais citados, retornados nas buscas, dos quais foram selecionados 86 (oitenta e seis). Após a aplicação dos CE e CI, foram selecionados 12 (doze) trabalhos para leitura do texto completo, os quais são apresentados na seção 2.1.6.

2.6 Resultado do Levantamento Sistemático

O primeiro trabalho foi proposto por Alencar e Marques (2021), tendo como título “Criação de um guia para selecionar métodos de avaliação para tecnologias de *software M-learning*”. Neste trabalho, as autoras propuseram a criação de um guia para avaliar SE para *M-learning*. Para isso, o guia sugeria métodos de avaliação existentes na literatura, onde o usuário definia os níveis de qualidade desejados (ALENCAR; MARQUES, 2021).

Para o desenvolvimento do guia as autoras realizaram as seguintes etapas: (i) busca e seleção dos métodos de avaliação para tecnologias de *software M-learning*; (ii) criação do guia; e (iii) avaliação do guia. Os resultados da pesquisa demonstraram a viabilidade da criação de um guia para auxiliar diferentes perfis interessados em avaliar a qualidade de tecnologias *M-learning*. Verificando que o guia facilita o processo de seleção de métodos adequados, pois sugere um ou mais métodos de avaliação com base em critérios de avaliação selecionados pelo usuário avaliador. De acordo com as conclusões obtidas, as autoras indicam que o guia permite que avaliadores de diferentes perfis e objetivos tenham um direcionamento sobre qual método de avaliação de qualidade devem adotar em uma determinada situação. Embora o trabalho apresente o guia, este não foi avaliado considerando um número significativo de métodos, o que limita o trabalho. Também não utiliza soluções de IA para resolver o problema, podendo o processo sofrer por subjetividade.

Petri (2020), no estudo “Avaliação de *Softwares* Educacionais com Ênfase em Jogos: Um Panorama da Literatura”, buscou identificar e analisar o rigor científico das abordagens de avaliação de SE, com ênfase em jogos. Para isso, realizou um MSL em estudos publicados entre janeiro de 1995 e dezembro de 2019. Como resultado, Petri constatou que existem poucas abordagens que fornecem apoio para avaliação de jogos educacionais. Identificou que a maioria dos estudos são em relação a *frameworks*, não sendo métodos abrangentes e indicando a pouca disponibilidade de suporte para conduzir uma avaliação adequada. Ademais, o autor também concluiu que a maior parte das abordagens parece ser desenvolvida de maneira *ad-hoc*, não estabelecendo uma definição explícita do objetivo, medidas ou instrumentos de coleta de dados confiáveis e válidos. Desta forma, se confirma a necessidade de propor uma solução que possibilite definir o perfil de usuário e a perspectiva de avaliação que resulte na indicação do método de avaliação do SE mais adequado (PETRI, 2020).

No trabalho proposto por Leite (2020), com título “Aplicativos para Aprendizagem Móvel no Ensino de Química”, o autor buscou realizar um levantamento de aplicativos para dispositivos móveis disponíveis, que poderiam ser usados no ensino de Química e estavam disponíveis na loja *Google Play*. Para isso, pesquisou os aplicativos que poderiam ser utilizados por discentes e docentes residentes no Brasil, selecionando os aplicativos em língua portuguesa para a discussão. A busca na loja de aplicativos foi realizada utilizando as palavras de busca “Química” e “*Chemistry*”. Percebeu-se na verificação inicial que os aplicativos disponíveis com o objeto de apoiar no ensino de Química são substanciais. A cada busca eram retornados 250 (duzentos e cinquenta) aplicativos, os quais eram limitados pelo número de exibição possível de aplicativos. Para a seleção dos aplicativos que se buscava apresentar, foi usado o critério de mais baixados pelos usuários na loja. Por fim, foram apresentados os aplicativos selecionados e mais cinco que foram produzidos, pelo grupo de pesquisa do autor, no intuito de oferecer aos discentes temáticas não contempladas diretamente nos aplicativos selecionados. As conclusões obtidas no trabalho, foram que muitos aplicativos não disponibilizam somente o conteúdo, mas também disponibiliza jogos, *quiz*, reações etc. O que os autores consideraram interessante, pois

possibilita ao aluno o uso de diversos recursos para a construção de seu conhecimento. Em contraponto, também concluíram que quanto mais recursos diferentes o aplicativo dispõe, o risco de não aprofundar no conteúdo é maior. Ainda assim, a seleção dos aplicativos considerando somente o critério de mais baixados na loja de aplicativos, pode apresentar fragilidades, uma vez que não garante que o aplicativo selecionado atenda aos requisitos importantes para o usuário, sendo necessário aplicar alguma metodologia de avaliação que considere os critérios definidos pelo usuário como importantes (LEITE, 2020).

No trabalho proposto por Dutra et al. (2020), com título “*Mobile Learning* e o uso de apps como proposta para o ensino de Ciências”, foram analisados aplicativos em língua portuguesa e gratuitos que poderiam ser usados nas aulas de Ciências e Biologia na Educação Básica. Para isso, foi realizado um levantamento na loja de aplicativos *Google Play*, com palavras-chaves para aplicativos potenciais para o Ensino de Ciências em contexto escolar. Os aplicativos foram categorizados de acordo com a tipologia (vídeo ou animação, simulador, conjunto de exercícios, material de consulta de informações, livro ou outro); nível de ensino (Ensino Fundamental anos iniciais ou anos finais e Ensino Médio); temática e conteúdo. Como resultado foram selecionados 42 (quarenta e dois) aplicativos os quais foram avaliados com potencial para serem utilizados pelos professores de Ciências e Biologia numa proposta *M-learnig* em sala de aula, todavia também se concluiu ser necessário o planejamento adequado da aula com o uso de aplicativos, com o professor selecionando e avaliando previamente. No entanto, o trabalho não propõe uma metodologia para o professor avaliar os aplicativos, baseado em critérios importantes para necessidade do usuário, assim como, não utilizou técnicas de IA para efetuar a seleção (DUTRA; BERVIAN; GÜLLICH, 2020).

Junior et al. (2020), em “Taxonomia de critérios para avaliação de *Software* educativo – TaCASE”, propuseram uma Taxinomia de Critérios para Avaliação de *Software* Educacionais – TaCASE, que contemplou três classes: qualidade de *software*, qualidade de uso e pedagógica. Para o desenvolvimento, fizeram uma análise da literatura e na elaboração de TaCASE consideram a análise de 413 (quatrocentos e treze) critérios de avaliação de *software* educativo, extraídos de 14 (quatorze) abordagens disponíveis na literatura, os quais contemplaram duas classes que agrupam 21 (vinte e um) critérios de qualidade de *software* e 12 (doze) critérios de qualidade de uso (usabilidade). Para avaliar a taxinomia, foram aplicados questionários a fim de detectar a concordância (ou não) de especialistas de Engenharia de *Software* e de Usabilidade, diante dos critérios e definições propostos. Como resultados, foram obtidas 48 (quarenta e oito) respostas válidas, 27 (vinte e sete) para o primeiro perfil e 21 (vinte e um) para o segundo, sendo a maior concordância dos especialistas para os critérios de qualidade de uso. Para os critérios de qualidade de *software*, as discordâncias foram significativas, chegando a apresentar 52,3% no requisito Instabilidade. Embora o TaCASE englobe 14 (quatorze) abordagens, com 33 (trinta e três) critérios de qualidade de *software* e uso, ainda se apresenta limitado para avaliação mais generalista de SE, visto que não considera características importantes para aplicativos educacionais, como as características pedagógicas (JUNIOR; AGUIAR; MOURA, 2020).

Já o trabalho com título “*Mobile Learning: Avaliação e Seleção de um Aplicativo para o Ensino de Elementos Químicos*”, Da Silva et al. (2018) propuseram avaliar e selecionar um aplicativo educacional para o ensino de Química, com foco no estudo dos Elementos Químicos. A pesquisa foi realizada analisando o número de downloads, avaliando o grau de satisfação dos usuários na loja de aplicativo e uma avaliação de acordo com as concepções de aprendizagem. Nos resultados foi observado que o aplicativo com maior número de *downloads* foi o “Tabela Periódica *Quiz*” (500.000 – 1000.000), porém o que ofereceu maior grau de satisfação aos usuários foi o aplicativo “Elementos Químicos e Tabela Periódica – Nomes e Testes” (73%). Em relação às avaliações utilizando-se das *checklists* das concepções de aprendizagem, pode-se observar que aplicativos mostraram mais características da concepção empirista, outros mostraram características da concepção racionalista e além dos que apresentaram 50% de características interacionista. Observando os resultados, os autores concluíram que os aplicativos Objeto Educacional Elementos Químicos e Tabela Periódica: Nomes e Testes pode ser utilizado pelos usuários, devido a maior aceitabilidade, como complemento a aulas sobre elementos químicos. Em contraponto, é observado que os *checklists* não contemplam requisitos importantes para diferentes tipos de usuários e a avaliação não é feita de forma automatizada (CRISÓSTOMO et al., 2018).

Nunes e Santos (2018) no trabalho “A importância da avaliação ergonômico-pedagógica de aplicativos educacionais e os desafios encontrados no campo da aprendizagem móvel”, propuseram uma avaliação ergonômico-pedagógica relacionados às disciplinas abordadas no Exame Nacional do Ensino Médio (ENEM), na perspectiva de um profissional de pedagogia. A proposta considerou a revisão das principais abordagens de avaliação de *softwares* educativos realizada pelas autoras. No desenvolvimento, inicialmente o pedagogo fez uma exploração individual em cada aplicativo para identificar aspectos básicos como o tipo de abordagem e proposta educativa. Em seguida foram anotados os quesitos observados para avaliação, tomando por base critérios adaptados da literatura. Como resultado, a avaliação do profissional de pedagogia concluiu que os aplicativos analisados trabalham com pouca estrutura e exploram superficialmente os conceitos referentes aos temas abordados, além de comumente adotarem metodologias baseadas em memorização e fragmentação do conhecimento. Embora o trabalho tenha buscado analisar a perspectiva ergonômica-pedagógica, pouco observada na literatura, é necessário que a avaliação seja feita considerando requisitos mais amplos, como técnicos, e também seja feita de forma automatizada (NUNES; SANTOS, 2018).

Já no trabalho “*Checklists Específicas para Softwares Educacionais: uma Proposta e um Mapeamento Sistemático*”, Silva et al. (2018) propuseram a criação de *checklists* específicas para diferentes modalidades de SE. Para isso, realizaram uma MSL sobre a utilização de *checklists* específicas para diferentes tipos de SE, com temas relacionados à utilização de métricas para as avaliações. Para a criação das *checklists*, alguns critérios encontrados com o resultado do mapeamento sistemático foram utilizados, e se integrou certas métricas encontradas na normatização ISO/IEC 25010. Ao final foram selecionadas as modalidades avaliativas: Características

pedagógicas, Facilidade de uso, Características de interface, Adaptabilidade, Documentação, Portabilidade. Para avaliação dos *checklists* foi apresentado um formulário para os avaliadores, que continha a seguinte pergunta “Qual *checklist* se enquadra mais para cada categoria de *software* avaliado?”, a qual era respondida ao final da avaliação de ambas as *checklists*. Como resultado, foi observado que respostas em sua maioria deu preferência pela *checklist* específica com um resultado geral de 80%, contra 20%, que escolheram a *checklist* do PROINFO (Programa Nacional de Tecnologia Educacional). Os autores concluíram que existem poucos trabalhos que abordem de *checklists* para avaliação de SE, destacando-se as metodologias de avaliação divididas em critérios, o que colaborou na criação de *checklists* específicas. Embora criado o *checklist*, a proposta não contempla critérios relacionados como importantes para avaliação em perspectivas variadas, como pedagógica, ergonômica, de engenharia de *software*, tampouco é feita de forma automatizada, com a utilização de IA, por exemplo (SILVA et al., 2018).

Andrade et al. (2017) no trabalho “Estabelecimento de Critérios de Qualidade para Aplicativos Educacionais no Contexto dos Dispositivos Móveis (*M-Learning*)”, propuseram categorizar os critérios de qualidade de um aplicativo bem como estabelece um instrumento para avaliação de SE. Em sua pesquisa os autores observaram que cada categoria apresenta suas próprias características e especificidades, as quais podem ser explicitadas e discutidas de modo que os diferentes tipos de aplicativos possam ser adotados nas situações de ensino-aprendizagem adequadas. Além disso, as categorias e os critérios permitem um maior número de análises desta forma, atenderão um maior número de usuários e suas necessidades de avaliação de SE. Os autores destacam também que as categorias e os critérios constituíram um instrumento com pontuação – escala Likert, que compõe um instrumento para avaliação de aplicativos para SE. Concluiu-se então que assim como é importante os critérios terem qualidade, os objetivos que se desejam alcançar devem estar claros para a escolha de um aplicativo e quais conceitos do conhecimento específico ao aplicativo oferece. Desta forma, uma solução com a utilização de IA para automatizar a seleção da metodologia para avaliação do aplicativo, que considerasse os critérios e os objetivos de conhecimento, auxiliaria na seleção do SE que melhor atendesse as necessidades do usuário (ANDRADE; JR; SILVEIRA, 2017).

Já no trabalho “Avaliação de *Software* Educativo: a complexidade de escolher uma abordagem adequada”, Silva et al. (2016) buscaram identificar a influência da escolha da abordagem de avaliação de SE no diagnóstico resultante da sua aplicação. No desenvolvimento o trabalho apresenta a aplicação de quatro abordagens de avaliação para um SE de aprendizado de idiomas. Nos resultados os autores destacaram que as abordagens possuem enfoques diferentes para avaliar aspectos específicos do SE, afirmando que a escolha da abordagem pode variar de acordo com o objetivo do avaliador. Desta forma, concluíram que a escolha por uma abordagem pode resultar na ausência da observação de critérios definidos como importantes pelo avaliador, uma vez que deve haver uma comunhão entre o que se pretende avaliar e os critérios presentes na abordagem. Desta forma, uma solução com a indicação da melhor metodologia dentre as disponíveis na literatura, através de IA, considerando o perfil do usuário, características e requisitos

poderá sanar nesta ausência de observação de critérios importantes (SILVA; NETTO; SOUZA, 2016).

Nascimento e Werneck (2016) no trabalho “Aprendizagem Móvel no Ensino de Matemática Básica: Proposta de um Instrumento de Avaliação de Aplicativos”, propuseram fornecer e avaliar um instrumento para avaliação de SE. Para isso elaboraram um MSL sobre instrumento de avaliação de aplicativos para o ensino de matemática básica. No desenvolvimento, o instrumento proposto fez a avaliação de aplicativos com base em 4 (quatro) dimensões: pedagógica, ensino da matemática, experiência do usuário e interface. Este trabalho envolveu as seguintes etapas: identificação dos aplicativos, definição dos critérios de avaliação dos artigos, escolha dos aplicativos a serem avaliados, e realização da avaliação dos aplicativos. Os autores selecionaram 2 (dois) aplicativos que eram gratuitos e em português, após buscas realizadas na *internet*, na literatura e em lojas virtuais de aplicativos. Os resultados apontaram que o instrumento proposto apontou o melhor aplicativo como o Rei da Matemática Júnior que ficou 98,8% dos pontos possíveis (Classificação Muito Bom), e o aplicativo *Kids Numbers and Math* (Classificação Muito Bom), com 83,3% dos pontos totais possíveis, recomendando tais aplicativos para uso pelas crianças para o aprendizado móvel. Concluiu-se que o professor pode utilizar o instrumento proposto como um mecanismo de decisão formal como ferramenta de suporte a decisão de qual aplicativo seria mais adequado para os alunos. Embora contemplando dimensões importantes para a avaliação de SE, o instrumento não avalia de forma automatizada, como com a utilização de IA (NASCIMENTO; WERNECK, 2016).

D’Carlo et al. (2016) no trabalho “Usabilidade em Aplicativos Móveis Educacionais: Um Conjunto de Heurísticas para Avaliação”, propuseram um conjunto de heurísticas específicas para avaliar e caracterizar a usabilidade de SE. Para isso, realizaram um MSL que identificou as heurísticas de usabilidade propostas para aplicativos móveis em geral, e apreciou-se o grau de cobertura das mesmas em relação à usabilidade de aplicativos educacionais. No desenvolvimento, realizou-se uma análise comparativa para comprovar a aplicabilidade do conjunto de heurísticas específicas para usabilidade de aplicativos educacionais. Nessa análise, os aplicativos Nota10 e o Passei!ENEM foram avaliados por dois grupos de especialistas em Interação Homem-Computador, seguindo os passos da Avaliação Heurística. Um grupo de especialistas foi guiado pelo conjunto de heurísticas genéricas para aplicativos móveis, e o outro grupo pelas heurísticas propostas neste trabalho. Embora a Avaliação Heurística seja importante, é necessário analisar os SE por outras perspectivas e de forma automatizada (D’CARLO; BARBOSA; OLIVEIRA, 2016).

Em linhas gerais, é possível observar a presença das lacunas presentes no trabalho, as quais são preenchidas pelo cumprimento dos objetivos geral e específicos dessa dissertação. A Tabela 1 ilustra a sumarização dos itens que merecem a devida atenção.

Conforme é possível observar, o trabalho proposto cobre os pontos em aberto presentes na literatura.

Tabela 1 – Sumarização dos itens considerados nos estudos

Título do artigo	Minimização subjetividade	Automa-tização	Pedagógica	Técnica	Socio-cultural	Socio-econômica	Download
Criação de um guia para selecionar métodos de avaliação para tecnologias de <i>software M-learning</i>			X	X	X	X	
Avaliação de <i>Softwares</i> Educacionais com Ênfase em Jogos: Um Panorama da Literatura				X			
Aplicativos para Aprendizagem Móvel no Ensino de Química			X	X			X
<i>Mobile Learning</i> e o uso de <i>apps</i> como proposta para o Ensino de Ciências			X				X
Taxonomia de critérios para avaliação de <i>Software</i> educativo – TaCASE			X	X			
<i>Mobile Learning</i> : Avaliação e Seleção de um Aplicativo para o Ensino de Elementos Químicos							X
A importância da avaliação ergonômico-pedagógica de aplicativos educacionais e os desafios encontrados no campo da aprendizagem móvel			X				
<i>Checklists</i> Específicas para <i>Softwares</i> Educacionais: uma Proposta e um Mapeamento Sistemático			X	X			
Estabelecimento de Critérios de Qualidade para Aplicativos Educacionais no Contexto dos Dispositivos Móveis (<i>M- Learning</i>)				X			
Avaliação de <i>Software</i> Educativo: a complexidade de escolher uma abordagem adequada			X	X			
Aprendizagem Móvel no Ensino de Matemática Básica: Proposta de Um Instrumento de Avaliação de Aplicativos			X	X			
Usabilidade em Aplicativos Móveis Educacionais: Um Conjunto de Heurísticas para Avaliação				X			
Algoritmos para Seleção de Metodologias de Avaliação de <i>Softwares</i> Educativos	X	X	X	X	X	X	X

Fonte: criada pela autora.

2.7 Considerações Finais

Embora os trabalhos tenham apresentado suas vantagens. É possível observar as lacunas em relação ao uso de SE. A falta de padronização nos critérios de qualidade, mesmo quando propõem fazer análises semelhantes, nas escalas de mensuração e nos procedimentos de verificação. Essa falta de uniformização deixa complexo a avaliação dos SE, e torna os resultados inconclusivos.

3 TRABALHOS RELACIONADOS

3.1 Métodos para Avaliação de SE

Na revisão da literatura realizada neste trabalho, encontrou-se muitas propostas de modelos, abordagens e metodologias para avaliação da qualidade de *software* educacional, conforme detalhado no item 2.6. A aplicação desses modelos e abordagens para a avaliação de *software*, quando realizados criteriosamente, permitem a redução dos riscos de aplicar-se um produto mal concebido, muitas vezes sem os recursos motivadores e reforçadores que ajudem o usuário a atingir o seu objetivo. Este capítulo apresenta os 24 (vinte e quatro) métodos de avaliação de *software* educacional que irão compor o sistema automatizado baseado em IA.

A seguir é realizada uma breve descrição de cada um dos métodos, assim como, serão apresentadas suas devidas referências bibliográficas.

3.1.1 Método de Reeves

O método Reeves possibilita a avaliação do SE tanto do ponto de vista pedagógico quanto do ponto de vista de interface. O qual utiliza um total de 24 (vinte e quatro) critérios. Suas aplicações incluem a avaliação (objetiva), através de diagramas com setas bidirecionais, nos quais são feitos diagnósticos pelos educadores por meio da análise da configuração dos pontos marcados nas setas (REEVES; HARMON, 1994).

O método foi aplicado para avaliação nas áreas de matemática, informática e saúde, em SE de exercícios e práticas, simulações, multimídia, jogos digitais e internet (GOÉS et al., 2015; SILVA et al., 2016). Também foi utilizado na avaliação o da Rede Internacional Virtual de Aprendizagem e Banco Internacional de Objetos Educacionais (MEDEIROS; SCHIMIGUEL, 2012).

3.1.2 Técnica de TICESE

A Técnica de Inspeção Ergonômica de *Software* Educacional (TICESE) avalia 17 (dezesete) critérios para descobrir o uso apropriado de um *software* como um instrumento de ensino para assessorar o educador (GAMEZ, 1998).

A aplicação da técnica é uma avaliação objetiva, que utiliza uma escala numérica de 3 (três) pontos: 0 (zero), 1 (um) e 1,5 (um e meio). Como resultado se encontra o relacionamento dos valores com os critérios, através de um gráfico de barras com os respectivos critérios e valores concedidos.

A técnica é considerada ampla, por abordar os critérios por meio de uma vasta estrutura de checklists. A TICESE foi utilizada para examinar SE nas disciplinas de português e matemática, em exercício e prática, simulação e jogos digitais (BRAGA, 2006; SILVA et al., 2013).

3.1.3 Técnica de Mucchielli

A Técnica de Mucchielli tem como objetivo possibilitar a análise global de SE considerando seu público-alvo. A técnica utiliza o *checklist* como instrumento de avaliação (objetiva) e propõe 10 (dez) critérios de qualidade de *software* (MUCCHIELLI, 1987).

Na utilização da técnica, realiza-se uma avaliação em uma escala numérica que varia de 1 (um) até 5 (cinco), de modo que se um critério for apontado como positivo ele será sinalado com a nota 5 (cinco) e, se for julgado negativo, será sinalado com a nota 1 (um). Como resultado se encontram as associações dos valores aos critérios, apresentados em um gráfico de barras com critérios e os valores atribuídos.

A técnica foi utilizada para avaliar distintos SE, das disciplinas de engenharias de *software*, português e matemática, do tipo exercício, prática e simulação (ANDRES; CYBIS, 2000; OLIVEIRA et al., 2015).

3.1.4 Método de LORI -*Learning Object Review Instrument*

Desenvolvido pela *e-Learning Research and Assessment Network*, o *Learning Object Review Instrument* (LORI) é usado em alguns países para a análise de objetos de aprendizagem acessíveis pela *internet*. São analisados 3 (três) critérios de qualidade de *software*, 3 (três) critérios de qualidade pedagógica e 3 (três) critérios classificados pelo autor como de qualidade, não definindo qual tipo (NESBIT; BELFER; LEACOCK, 2003).

A utilização do método corresponde a uma avaliação objetiva representada por uma escala numérica que varia de 1 (um) até 5 (cinco). A pontuação é dada considerando a análise do critério do objeto de aprendizagem. O resultado é a observação da relação dos valores aos critérios, representado por um gráfico de barras com critérios e valores atribuídos.

O instrumento foi utilizado para avaliar jogos digitais das disciplinas matemática, programação e psicologia (SANTOS et al., 2012; MARKOVIĆ; KLIČEK; VUKOVAC, 2014; OLIVEIRA et al., 2015).

3.1.5 MAQSEI - Metodologia para Avaliação da Qualidade de *Software* Educacional Infantil

O MAQSEI é uma proposta de metodologia para avaliar a qualidade de SE a partir de uma relação de critérios de verificação com 14 (quatorze) critérios associados a heurísticas pedagógicas e de usabilidade. A avaliação é subjetiva, visto que utiliza uma escala não numérica e é verificado se o critério é contemplado, isto é, sim ou não. O resultado é um relatório que contém uma relação de possíveis falhas identificadas no SE (ATAYDE; TEIXEIRA; PÁDUA, 2003).

A metodologia MAQSEI já foi usada em pesquisas para avaliar jogos digitais na área de computação (ZORZAL et al., 2008; SOUZA, 2014).

3.1.6 FASE - Ferramenta Especialista para Avaliação de *Software* Educacional

A Ferramenta Especialista para Avaliação de *Software* Educacional (FASE), desenvolvida por Webber, Boff e Bono (2009), analisa 48 (quarenta e oito) critérios para identificar o uso ideal de um SE. Sua avaliação é objetiva e o resultado da ferramenta é apresentado através do cálculo de inferência que determina a razão o peso obtido pela avaliação do SE.

A ferramenta foi utilizada para analisar vários SE do tipo jogo digital e programação nas disciplinas de português e computação (WEBBER; BOFF; BONO, 2009; SILVA; BATISTA, 2015a; OLIVEIRA et al., 2015).

3.1.7 Método de Savi - Modelo de Avaliação de Jogos Educacionais

Savi propôs o Modelo de Avaliação de Jogos Educacionais (MAJE), que analisa 43 (quarenta e três) critérios observando as interações disponíveis em um SE. Sua avaliação é mista (utiliza escala não numérica) e o diagnóstico é dado por meio da pontuação sobre os aspectos avaliados pelo questionário (SAVI et al., 2010).

O instrumento avaliou vários SE dos tipos exercício e prática, jogos digitais, simulação e programação, nas disciplinas português, matemática, pedagogia, computação, biologia, física e história (MOITA; HENRIQUE; DANIELE, 2015; OLIVEIRA et al., 2013; OLIVEIRA et al., 2015; SIQUEIRA et al., 2011; TELLES; ALVES, 2015).

3.1.8 PETESE - *Pedagogical Ergonomic Tool for Educational Software Evaluation*

A ferramenta *Pedagogical Ergonomic Tool for Educational Software Evaluation* (PETESE), utiliza 68 (sessenta e oito) critérios que avaliam os SE durante o processo de desenvolvimento. Sua avaliação é mista e o diagnóstico dos resultados não é informado pelos autores, que recomendam que a equipe de avaliação seja composta por um profissional específico para cada atributo presente na ferramenta (COOMANS; LACERDA, 2015).

A ferramenta foi usada para avaliar jogos digitais e tutoriais das disciplinas de português, matemática e computação (NUNES; SANTOS, 2018; GOMES et al., 2019).

3.1.9 MAQSE - Metodologia para Avaliação da Qualidade de *Software* Educacional

Campos (1994) (CAMPOS, 1994) propôs a Metodologia para Avaliação da Qualidade de *Software* Educacional, que avalia 9 (nove) critérios por meio de um *checklist* técnico que

contém fatores e subfatores que determinam os objetivos avaliados e o propósito de cada um. A avaliação é objetiva, utilizando uma escala numérica de 5 (cinco) pontos (0; 0,25; 0,50; 0,75 e 1). O resultado é a pontuação sobre os aspectos analisados pelo *checklist*; sendo a nota do critério próxima do 0 (zero) considerada negativa e próxima do 1 (um) positiva.

O MAQSE foi utilizado para avaliar exercício e prática, simulação, multimídia e *internet* nas disciplinas de português, matemática e química (FARIAS, 2007; MACHADO et al., 2007; ROQUE et al., 2009).

3.1.10 Método Rocha

O método Rocha utiliza 23 (vinte e três) critérios para analisar a qualidade de SE. O método fundamenta-se em alguns conceitos que se distribuem em duas categorias: objetos de qualidade e fatores de qualidade do produto. A avaliação é objetiva e utiliza uma escala numérica de 4 (quatro) pontos (0.00 a 0.59; 0.60 a 0.89; 0.90 a 0.94; 0.95 a 1). O resultado possibilita a análise da relação dos valores com critérios apresentados em um gráfico de barras (ROCHA; CAMPOS, 1993).

O método Rocha avaliou SE de exercício e prática nas disciplinas de matemática e português (FRECKI, 2008; OLIVEIRA et al., 2013).

3.1.11 IAQSEM - Instrumento de Avaliação da Qualidade para *Software* Educacional de Matemática

Gladcheff et al. propuseram o Instrumento de Avaliação da Qualidade para *Software* Educacional de Matemática (IAQSEM), que analisam 55 (cinquenta e cinco) critérios que contribuem para identificar o melhor uso de um *software* de matemática. Sua avaliação é subjetiva (utiliza uma escala não numérica) e o resultado do instrumento é um relatório (GLADCHEFF; ZUFFI; SILVA, 2001).

O instrumento foi utilizado para avaliar SE de exercício e prática, simulação e jogos digitais, das áreas de química, português e matemática (CAVALCANTI; FERREIRA, 2011; ABREU et al., 2012; NOGUEIRA et al., 2013; PIVATTO, 2014; TENÓRIO; SOUZA; TENÓRIO, 2015).

3.1.12 MAEP - Método Ergopedagógico Interativo de Avaliação para Produtos Educacionais Informatizados

O Método Ergopedagógico Interativo de Avaliação para Produtos Educacionais Informatizados (MAEP), desenvolvido por Silva (2002), analisa 36 (trinta e seis) critérios para verificar a usabilidade pedagógica de *softwares* de ambientes de aprendizagem virtual. A avaliação é

objetiva e considera uma escala numérica de 3 (três) pontos (1 (um), 2 (dois) e 3 (três)). O resultado é a pontuação sobre os diferentes aspectos analisados no *checklist*.

O MAEP foi utilizado para avaliar SE dos tipos de multimídia e internet, jogos digitais e programação, das áreas português, computação e engenharia civil (MELO et al., 2009; RIBEIRO; CABRAL, 2014).

3.1.13 Método Rodrigues - Instrumento para Avaliação de Jogos Eletrônicos Educativos do Ensino Fundamental I

Rodrigues propôs um instrumento para avaliação SE, do tipo jogo digital, sob três perspectivas: técnica, pedagógica e opinião dos alunos. Analisa 48 (quarenta e oito) critérios e sua avaliação é mista. O resultado é a pontuação relativa aos diferentes atributos avaliados, que calcula o peso da avaliação para cada perspectiva do SE (RODRIGUES, 2014).

O instrumento avaliou exercício e prática, simulação e jogos digitais das disciplinas de computação e biologia (FONSECA et al., 2017; REIS; JÚNIOR, 2019).

3.1.14 TUP Model - *Evaluation Model Technology, Usability and Pedagogy*

Bednarik et al., apresenta um modelo para a avaliação de SE, com 16 (dezesseis) critérios que consideram três perspectivas: técnica, usabilidade e pedagógica. Sua avaliação é subjetiva (utiliza uma escala não numérica, que verifica se é contemplado sim ou não) e o resultado é um relatório, que contém uma lista das possíveis falhas encontradas no SE (BEDNARIK et al., 2004).

O modelo avaliou SE dos tipos exercício e prática, programação e tutorial das disciplinas de computação e física (NAUMANEN MINNAMARI; TUKIAINEN, 2010; PERSICO; MANCA; POZZI, 2014; MOMANI, 2021).

3.1.15 Método ASE - Método de Avaliação de *Software* Educativo

A proposta de Vieira (1999), tem como principal objetivo analisar 34 (trinta e quatro) critérios para examinar as interações disponíveis em um SE, fundamentado em um ponto de vista construtivista de aprendizagem. Sua avaliação é mista e o resultado é a pontuação sobre as diferentes perspectivas avaliadas pelo questionário e um relatório.

O método avaliou SE dos tipos exercício e prática, simulação e jogos digitais contemplando as disciplinas de ciência, matemática e gestão ambiental (RAABE et al., 2007; FIALHO; MATOS, 2010; GUEDES; DALMORO, 2010; MACHADO; OLIVEIRA; ALMEIDA, 2011; LOPES et al., 2011).

3.1.16 CASE - *Checklists for the Evaluation of Educational Software*

Tergan desenvolveu o *Checklists for the Evaluation of Educational Software* (CASE) para avaliar 15 (quinze) critérios de qualidade de *software*, pedagógica e híbrida, que cooperaram para definir o melhor uso de um SE em um determinado contexto educacional. A avaliação é subjetiva (utiliza uma escala não numérica) e o resultado é um relatório com os comentários dos avaliadores sobre as dimensões avaliadas de acordo com cada critério apresentado (TERGAN, 1998).

O CASE foi utilizado para avaliar SE do tipo exercício e prática, simulação, multimídia e internet nas disciplinas de medicina, odontologia e pedagogia (NATTESTAD et al., 2002; SILIUS; KAILANTO; TERVAKARI, 2011).

3.1.17 MEEGA+ - *Model for the Evaluation of Educational Games*

O MEEGA+ baseia-se em um modelo para avaliar 33 (trinta e três) critérios de jogos educacionais voltados para o ensino de computação. O modelo busca analisar a experiência e percepção de aprendizagem do na perspectiva de alunos e instrutores no contexto de cursos superiores da área de computação (PETRI et al., 2018).

Sua avaliação é objetiva, realizada com uma escala numérica que apresenta variação de números em uma sequência de -2 (menos dois) até 2 (dois). O resultado é pontuação de cada critério diante do SE, onde -2 (menos dois) corresponde a discordo totalmente e 2 (dois) a concordo totalmente.

O MEEGA+ avaliou SE dos tipos de simulação, jogos digitais e programação contemplando as áreas de engenharia de *software*, fundamentos de desenvolvimento de *software*, algoritmos e complexidade, interação humano-computador e gerenciamento de projetos (PETRI et al., 2018; RIPASY et al., 2018; WANGENHEIM et al., 2019).

3.1.18 MoLeVa - *Mobile Learning Evaluation*

Soad e Barbosa desenvolveram o *Mobile Learning Evaluation*, que propõe avaliar a qualidade de aplicativos educacionais móveis. Utilizam 85 (oitenta e cinco) critérios que avaliam a qualidade de aplicativos móveis não somente sob os aspectos técnicos, mas também as questões intrínsecas relacionadas com atividades diárias dos usuários das aplicações, sendo elas: educacionais, socioculturais e socioeconômicos (SOAD; BARBOSA, 2017).

A avaliação é objetiva, onde pontuações são atribuídas a cada pergunta do *checklist*. O cálculo final do nível de qualidade é definido pela média aritmética da pontuação de cada critério.

3.1.19 Questionário de Silva e Batista

Silva e Batista, propuseram uma Metodologia de Avaliação da Qualidade de Aplicativos Educacionais para Matemática do Ensino Médio, que analisa 30 (trinta) critérios para avaliar a qualidade de *software* e pedagógica dos SE de matérias do ensino médio na disciplina matemática. A avaliação é subjetiva (utiliza uma escala não numérica) e o resultado é o relatório com as respostas (SILVA; BATISTA, 2015b).

As autoras realizaram 3 (três) testes com grupos diferentes de usuários. O primeiro teste foi realizado por dois alunos do Bacharelado em Sistemas de Informação e uma aluna da Pós-Graduação Lato Sensu de Análise e Gestão de Sistemas de Informação, todos com experiência de uso de aplicativos para dispositivos móveis. O segundo teste foi realizado por 3 (três) alunos do Ensino Técnico de Informática, que trabalhavam na seleção e análise de SE destinados ao estudo de áreas curriculares diversas. E o terceiro grupo foram 6 (seis) estudantes de licenciatura em Matemática, todos com experiência na utilização de recursos tecnológicos no processo de ensino e aprendizagem de matemática.

3.1.20 Checklist de Porto

A proposta de Porto e Barbosa foi um *Checklist* de Avaliação de Usabilidade de Aplicativos *Android* no Contexto Educacional. O *checklist* utiliza 19 (dezenove) critérios, definidos a partir do *Material Design* (Google, 2017), para fazer a avaliação de *design* de interface de aplicativos *Android* criados no *App Inventor* no contexto do Ensino Básico. Sua avaliação é subjetiva (utiliza uma escala não numérica). O *checklist* foi considerado adequado pelos 12 (doze) pesquisadores e profissionais de computação que realizaram a avaliação no trabalho (PORTO; BARBOSA; WANGENHEIM, 2018).

3.1.21 Checklist de Soad

Soad et al. desenvolveram um checklist, que compreende 80 (oitenta) questões, classificadas em 8 (oito) diferentes grupos que contemplam 30 (trinta e seis) critérios de qualidade. Cada resposta do checklist definida é atribuída uma pontuação, permitindo que a questão receba um valor que caracterize seu nível de qualidade. O resultado é o somatório das questões de cada grupo (SOAD; FILHO; BARBOSA, 2016).

Os autores testaram a proposta avaliando 3 (três) SE da disciplina de inglês, e os resultados obtidos sugeriram que o método é adequado para avaliar a qualidade de aplicativos de aprendizagem móvel.

3.1.22 Avaliação Ergonômica-Pedagógica

Nunes e Santos apresentam uma avaliação ergonômico-pedagógica de aplicativos relacionados às disciplinas abordadas no Exame Nacional do Ensino Médio (ENEM), na perspectiva de um profissional de pedagogia. Utilizam 14 (quatorze) critérios para analisar 3 (três) SE em uma avaliação subjetiva (utiliza escala não numérica) onde o resultado é uma tabela com o grau de satisfação para cada critério, que pode ser “E”- excelente, “B”- bom, “R”- regular e “P”- péssimo (NUNES; SANTOS, 2018).

3.1.23 *Heuristic Evaluation Strategy*

A *Heuristics Evaluation Strategy for Mobile Game-Based Learning* apresentada por Zaibon e Shiratuddin, avalia os jogos baseados em aprendizagem considerando componentes de cada uma das características: Usabilidade do Jogo (GU), Mobilidade (MO), Jogabilidade (GP) e Conteúdo de Aprendizagem (LC) (ZAIBON; SHIRATUDDIN, 2010).

Os 10 (dez) componentes GU descrevem a interface e os controles do jogo com os quais o jogador interage com o jogo. A MO é composta por 3 (três) componentes que tratam das questões que afetam a mobilidade do jogo. Os 10 (dez) componentes do GP descrevem especificamente como o jogo é jogável, executado de maneira suave e consistente, significativo e não entediante para o jogador. E os componentes LC estão especificamente concentrados no conteúdo de aprendizagem. O resultado da avaliação são as respostas dos usuários do SE que são baseadas escalas Likert de 5 (cinco) pontos.

3.1.24 Método de Lulia e Simões

Vieira et al., apresentam uma proposta de Avaliação de SE, analisando os aspectos pedagógicos e técnicos para o ensino fundamental I. O qual utiliza um total de 24 (vinte e quatro) critérios (VIEIRA; SIMÕES; BARRETO, 2012).

O método de avaliação foi à observação, mediante o procedimento do manuseio prático do funcionamento e da dinâmica do *software*. Em seguida foram anotados os quesitos observados para análise e discussão, resultando em uma tabela com o grau de satisfação para cada critério, que pode ser “E”- excelente, “B”- bom, “R”- regular e “P”- péssimo.

4 REFERENCIAL TEÓRICO

4.1 *Software* Educacionais

Sobre *softwares* educacionais, Almeida e Almeida (2015, cap. 1, p. 10) afirmam que:

Softwares educacionais são programas que visam atender necessidades vinculadas à aprendizagem. Devem ter objetivos pedagógicos e sua utilização deve estar inserida em um contexto e em uma situação de ensino baseados em uma metodologia que oriente o processo, através da interação, da motivação e da descoberta, facilitando a aprendizagem de um conteúdo.

Já para Moraes (2016), “os SE podem auxiliar no processo de ensino-aprendizagem a partir da contextualização de determinados conteúdos, diante da realidade de cada aluno e fazendo com que este construa determinado conhecimento relativo ao conteúdo didático abordado.” Jogos, tutoriais, simulação, ebooks, aplicativos exercícios, dentre outros, são exemplos de SE.

A utilização de *softwares* no ensino tem acompanhado a própria história e evolução dos computadores (VALENTE et al., 1999). No mundo, esse uso começou em 1978, quando o físico americano Alfred Bork disse, na conferência “Aprendizagem Interativa” da Associação Americana de Professores de Física (*American Association of Physics Teachers*), que o computador seria um meio de aprendizagem nas escolas e que, até o ano 2000, a principal forma de aprender, em todos os níveis e em quase todas as áreas, seria usando os computadores de forma interativa (FIOLHAIS; TRINDADE, 2003). Segundo Moraes, os computadores foram usados pela primeira vez na educação no Brasil em 1971, em um seminário da Universidade de São Carlos, com a ajuda de um especialista da Universidade de Dartmouth, Estados Unidos, que discutiu o uso de computadores no ensino de Física (MORAES, 1997). Na UFRJ (Universidade Federal do Rio de Janeiro), em 1973, o Núcleo de Tecnologia Educacional para a Saúde e o Centro Latino-Americano de Tecnologia Educacional (NUTES/CLATES) usaram um *software* de simulação no ensino de Química (ANDRADE, 1993; VALENTE, 1997). Na UFRGS (Universidade Federal do Rio Grande do Sul), nesse mesmo ano, foram feitas algumas experiências usando simulação de fenômenos de Física com alunos da graduação. Entretanto, no Brasil, o programa de informática na educação começou com o primeiro e o segundo Seminário Nacional de Informática em Educação, realizados na Universidade de Brasília, em 1981, e na Universidade Federal da Bahia, em 1982. Desses seminários, surgiu o projeto Educom, que foi o primeiro projeto público nacional sobre informática educacional (ANDRADE, 1993). Em 1989, foi criado o Programa Nacional de Informática Educativa (PRONINFE), que almejava desenvolver a informática educativa no Brasil. Em 1999, o Programa Nacional de Informática na Educação (PROINFO), do Ministério da Educação (MEC), finalizou a instalação de 100 mil computadores em 16 mil escolas públicas de ensino básico (EICHLER; PINO, 2000). O

principal objetivo era estimular o uso pedagógico das tecnologias de informação e comunicação na educação básica pública. Com a evolução dos computadores e dos *softwares*, esse movimento se atualizou, e usar os computadores como recurso didático é um caminho sem volta, por causa da variedade dos *softwares* educacionais que existem hoje (JUCÁ, 2006).

Valente (1999) defende que o professor precisa ter uma formação mais ampla e sólida para usar os *softwares* educacionais, tanto no aspecto computacional quanto no curricular (VALENTE et al., 1999). Os *softwares* educacionais podem ajudar na construção do conhecimento, dependendo da criatividade de como o educador planeja e aplica os objetivos e as atividades. Há um interesse crescente dos educadores em usar ferramentas interativas e lúdicas, que abrangem diversas disciplinas escolares e melhoram a relação ensino-aprendizagem. Pela quantidade de *softwares* educacionais produzidos e procurados, os recursos computacionais estão cada vez mais valorizado como instrumento didático (BERTOLETTI; MORAES; COSTA, 2001).

Um exemplo da utilização de *softwares* educacionais são os usados para o ensino de Ciências, como apresentados por Paula et al. (2014), que utilizam diversos conceitos didáticos e pedagógicos na sua construção e operação. Esses conceitos incluem ilustrações e simulações interativas de fenômenos e leis, que facilitam o aprendizado de teorias e conceitos, e experimentos virtuais que possibilitam o controle de situações e variáveis, que podem substituir, em alguns casos, a prática real de laboratório sem prejudicar a aprendizagem. Assim, eles podem ajudar a superar o problema de manter laboratórios de ciências nas escolas, principalmente nas públicas, que têm alto custo de instalação, manutenção e operação. Laboratórios virtuais são mais simples de instalar e manter, estão alinhados com as políticas públicas brasileiras e podem ser usados em várias disciplinas, não apenas em ciências. (PAULA et al., 2014).

4.2 Qualidade de Software

A NBR (Norma Brasileira) ISO (*International Organization for Standardization*) 8402 (ABNT, 2000) conceituou qualidade como "A totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas". Sendo entidade, o produto ou serviço ao qual estamos nos referimos. As necessidades explícitas são as próprias condições e objetivos propostos pelo produtor. As necessidades implícitas incluem as diferenças entre os usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas (BOENTE; OLIVEIRA; ALVES, 2008). Na NBR ISO 9000:2015 (ABNT, 2015) qualidade foi definido como "grau em que um conjunto de características inerentes de um objeto satisfaz requisitos". Ou seja, se algum produto ou serviço atende aos requisitos especificados, este mesmo produto ou serviço possui a qualidade desejada.

O conceito de qualidade de software é abordado a partir de diversas definições existentes. Pressman (2011) afirma que "Qualidade de software pode ser definida como: uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para

aqueles que o produzem e para aqueles que o utilizam. (PRESSMAN, 2011)”

Para Bourque e Fairley (2014), “Outro ponto de vista da qualidade do software começa com enumerar as características desejadas de um produto de software e selecionando medidas ou outras avaliações para determinar o nível desejado de características foi alcançada” (BOURQUE; FAIRLEY, 2014). A qualidade de software é multidimensional, sendo assim, os requisitos de qualidade entram em conflito e os benefícios em atingi-los não são fáceis de medir (ROCHA; MALDONADO; WEBER, 2001).

De acordo com a NBR ISO/IEC 12207:1998, atributo é uma propriedade mensurável, física ou abstrata, de uma entidade (atributos podem ser internos ou externos), são exemplos de atributos de qualidade: adequação, conformidade, maturidade, tolerância a falhas, inteligibilidade, operacionalidade, utilização de recursos, testabilidade, modificabilidade, estabilidade, adaptabilidade, capacidade para substituir, entre outros. Para esse trabalho a definição de critério e subcritério é a mesma que a de atributo, pois durante a leitura de outras obras, percebeu-se essa variação no termo atributo (ABNT, 1998).

Rocha e Campos (1993) determinam uma estrutura para avaliar um software visando a qualidade, onde definiram que os objetivos de qualidade, que são atingidos através dos fatores de qualidade, podem ser compostos por outros fatores, que por sua vez são avaliados através de critérios. Os critérios definem atributos de qualidade para os fatores. Medidas são os valores resultantes da avaliação de um produto segundo um critério específico (ROCHA; CAMPOS, 1993).

Existem algumas normas que possibilitam avaliar a qualidade dos produtos e dos processos de desenvolvimento de *softwares*. A Tabela 3 apresenta as principais normas nacionais e internacionais nesta área.

Tabela 3 – Normas de qualidade de produtos e processos de desenvolvimento de *softwares*

Normas	Comentários
NBR ISO/IEC 25001	Engenharia de software - Requisitos e avaliação da qualidade de produto de software (SQuaRE) - Planejamento e gestão
NBR ISO/IEC 25020	Engenharia de software - Requisitos e Avaliação da Qualidade de Produto de Software (SQuaRE) - Guia e modelo de referência para medição
NBR ISO/IEC 25030	Engenharia de software - Requisitos e Avaliação da Qualidade de Produto de Software (SQuaRE) - Requisitos de qualidade
NBR ISO/IEC 25062	Engenharia de software - Requisitos e avaliação da qualidade de produto de software (SQuaRE) - Formato comum da indústria (FCI) para relatórios de teste de usabilidade
NBR ISO 9000	Sistemas de qualidade - Modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo)

NBR ISO 19011	Diretrizes para auditoria de sistemas de gestão
NBR ISO/IEC- IEEE 12207	Engenharia de Sistemas e Software - Processos de ciclo de vida de software
ISO/IEC TR 33015	Tecnologia da informação - Avaliação de processos — Orientação para a determinação de riscos de processos
IEEE P1061	Standard for Software Quality Metrics Methodology. Norma que trata das metodologias para padronização da qualidade de <i>software</i> , incluindo algumas abordagens de medição.
CMMI	Capability Maturity Model Integration, Modelo da SEI (Instituto de Engenharia de <i>Software</i> do Departamento de Defesa dos USA) para avaliação da qualidade do processo de desenvolvimento de <i>software</i> . Não é uma norma ISO, mas é bem-aceita no mercado.

Tendo em vista os diversos modelos para avaliação de *software*, percebe-se o quanto se torna difícil a tarefa de estabelecer-se parâmetros gerais para a avaliação de *software* com qualidade pedagógica, frente a gama de tipologias desses *software* existentes no mercado. Somente através de uma avaliação eficiente, pode-se concluir que determinado produto, satisfaz ou não a sua tarefa como ferramenta de apoio ao professor e aluno, no processo ensino-aprendizagem. As normas da ABNT (Associação Brasileira de Normas Técnicas), apresentam pontos importantes para uma avaliação, porém não discute uma forma de dar valores aos itens avaliados, o que dificulta o usuário na escolha do *software* a ser definido como instrumento de apoio no seu trabalho, tendo em vista a falta de uma tabela que permita atribuir valores às qualidades de um determinado produto.

4.3 Modelo de Qualidade

Na revisão da literatura realizada neste trabalho, encontrou-se muitas propostas de modelos, abordagens e metodologias para avaliação da qualidade de *software* educacional, conforme detalhado no item 2.6. A aplicação desses modelos e abordagens para a avaliação de *software*, quando realizados criteriosamente, permitem a redução dos riscos de aplicar-se um produto mal concebido, muitas vezes sem os recursos motivadores e reforçadores que ajudem o usuário a atingir o seu objetivo.

4.3.1 Modelo de Qualidade do MoLEva

O modelo de qualidade do método MoLEVA é responsável por determinar o conjunto de características que fornece a base para a avaliação de qualidade de aplicativos educacionais móveis. O modelo de qualidade do MoLEva é dividido em três categorias, fundamentadas na divisão dos requisitos de qualidade para aplicativos *M-learning* (SOAD; BARBOSA, 2017; SOAD; FILHO; BARBOSA, 2015; FILHO; BARBOSA, 2013). As categorias são:

- Pedagógica: são os aspectos educacionais presentes no aplicativo, ou seja, são as características que têm como foco o aprendizado do aluno;
- Social: referem-se aos aspectos sociais presentes no aplicativo, abordando características econômicas e culturais;
- Técnica: são os aspectos técnicos do aplicativo, ou seja, referem-se às características do aplicativo que contribuem para seu uso e funcionamento.

O modelo de qualidade possui uma estrutura composta por categorias, critérios e características de qualidade, dispostos de maneira hierárquica. Essa divisão foi fundamentada na norma ISO/IEC 25010 (ISO/IEC, 2011). Com o objetivo de desenvolver o método de avaliação de qualidade para aplicativos educacionais móveis, foi necessário realizar estudos sobre as características de qualidade presentes nesse tipo de aplicativo. Para isso, foi conduzido um mapeamento sistemático a fim de obter um conjunto de características de qualidade para aplicativos educacionais móveis.

A partir dos resultados obtidos no mapeamento sistemático, foi definido um conjunto de características de qualidade para o contexto de aplicativos educacionais móveis, estabelecendo assim a base para a criação do modelo de qualidade para o método MoLEva. Além disso, o método também teve como base as melhores práticas e conhecimentos estabelecidos na norma ISO 14598 (ISO/IEC 14598, 1998). Outras pesquisas como Duarte Filho e Barbosa (2013a) e Acharya e Sinha (2013) também foram utilizadas para auxiliar na definição das etapas do processo de avaliação utilizado pelo método MoLEva.

Em linhas gerais, o processo de desenvolvimento do método MoLEva é composto por seis etapas, em conformidade com a ISO/IEC 14598 (ISO/IEC 14598, 1998), as quais foram simplificadas e adaptadas para o contexto de aplicativos educacionais móveis. Para definir os principais componentes do método MoLEva, utilizou-se como base a norma ISO/IEC 14598 (ISO/IEC 14598, 1998), estabelecendo assim um método formado por três componentes fundamentais:

- Modelo de qualidade: define um conjunto de características de qualidade para aplicativos educacionais móveis, fornecendo assim a base para a avaliação de qualidade;

- Métricas: definem a forma de coleta de dados para possibilitar a medição da qualidade dos aplicativos;
- Critérios de julgamento: definem como o aplicativo será julgado a partir dos dados coletados durante a avaliação dos critérios de julgamento; (ii) execução da avaliação, que é a etapa na qual o checklist é aplicado; e (iii) análise dos resultados da avaliação, que apresenta os resultados obtidos com a execução da avaliação.

4.3.2 Modelo de Qualidade do TACASE

Em Ciência da Informação, taxonomia é um sistema utilizado para classificar e descomplicar o acesso à informação, sendo seu objetivo apresentar conceitos através de termos, melhorar a diversificação e oferecer um mapa do processo de conhecimento (BRASIL et al., 2013). Junior et al.(2020) propuseram uma taxonomia de critérios para avaliação de SE a partir de 14 (quatorze) abordagens descritas na literatura, contemplando aspectos de qualidade de *software*, de uso e pedagógica (JUNIOR; AGUIAR; MOURA, 2020).

À vista das características consideradas para SE serem apropriadas ao seu propósito, à literatura disponibiliza modelos para mensurar à qualidade de *software* (DUPUIS; BOURQUE; ABRAN, 2003), à qualidade de uso (NIELSEN, 1994; SHNEIDERMAN; PLAISANT, 2004; BASTIEN; SCAPIN, 1993) e a qualidade pedagógica (OLIVEIRA; COSTA; MOREIRA, 2001; SILVA, 2012).

A norma ISO 9126-1 (2003) e o Swebok Guide (2004) são modelos de qualidade de *software* considerados na área de engenharia de *software* e foram utilizados para definição do TACASE. As características relativas ao uso são orientadas a capacidade que uma solução tem de ser usada por usuários singulares para alcançar objetivos específicos com satisfação, eficácia e eficiência em uma condição específica de uso (NORMA ISO 9241-1, 2003).

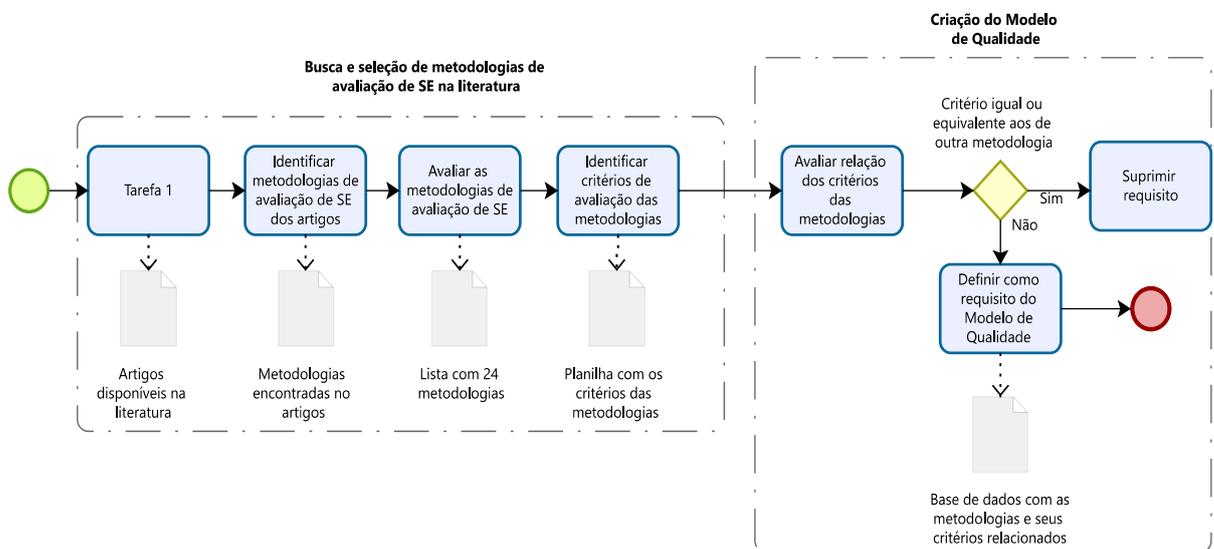
Para o modelo de qualidade de uso foram empregadas a norma ISO 9241-1 (2003), as 10 (dez) heurísticas de usabilidade de Nielsen (1994), as 8 (oito) sugeridas por Shneiderman e Plaisant (2004) e as 8 (oito) propostas por Bastien e Scapin (1993). Em relação ao SE, é essencial que exista cuidado no que se refere à qualidade pedagógica que está associada com a capacidade que o mesmo tem de satisfazer as exigências e necessidades do usuário relativas ao ensino e aprendizagem simultaneamente (SILVA; GOMES, 2015).

Logo, para analisar a qualidade pedagógica, adotou-se o modelo de qualidade de Oliveira et al. (2001) e Silva (2012). Esse modelo requer conhecimentos sobre as teorias de aprendizagem, bem como clareza das concepções educacionais e reflexão sobre as práticas pedagógicas. Além disso, é preciso saber sobre as técnicas computacionais e o papel do computador, do aluno e do professor no ambiente educacional.

5 PROPOSIÇÃO DO MODELO DE QUALIDADE (CASTILHO-FARIAS)

Este capítulo apresenta as etapas realizadas para criação do Modelo de Qualidade Castilho-Farias, que fornecerá a base de dados utilizada como entrada para a fase de aplicação dos algoritmos de IA. A Figura 2, ilustra as etapas percorridas para a criação do modelo e as atividades realizadas em casa etapa. A seguir será explanada cada uma das atividades realizadas nas etapas: (i) Busca e seleção das metodologias de avaliação de SE e (ii) Criação do Modelo de Qualidade.

Figura 2 – Processo de desenvolvimento do Modelo de Qualidade Castilho-Farias



Fonte: criada pela autora.

Busca e seleção de metodologias de avaliação de SE disponíveis na literatura: nesta atividade foi realizada a pesquisa, identificação e análise de metodologias de avaliação de SE disponíveis na literatura.

- **Pesquisa sobre metodologias de avaliação de SE:** esta etapa tem como objetivo encontrar artigos que descrevam metodologias de avaliação de SE.
- **Identificar metodologias de avaliação de SE dos artigos:** nesta etapa foram analisados os artigos a fim de identificar as metodologias de avaliação de SE que utilizavam.
- **Avaliar as metodologias de avaliação de SE:** nesta etapa foram avaliados os artigos e definidas as 24 metodologias que seriam utilizadas no Modelo de Qualidade proposto.
- **Identificar critérios de avaliação das metodologias:** nesta etapa foi criada uma planilha que continha as metodologias e todos os requisitos que eram utilizados por ela.

Criação do Modelo de Qualidade: O modelo de qualidade proposto neste trabalho, foi a interseção entre os modelos de qualidade MoLEva e TACASE. Do modelo de qualidade do método MoLEva foram utilizadas as categorias, renomeadas para características que são: Pedagógicas, Técnicas, Socioculturais e Socioeconômicas, conforme estabelecidos por Alencar e Marques, (2021). Do método TACASE as características consideradas foram as de qualidade Pedagógicas, de Uso e de Software (JUNIOR; AGUIAR; MOURA, 2020). A seguir são listadas e explicadas as etapas realizadas nesta fase.

- **Analisar relação dos critérios das metodologias:** nesta etapa foram analisados os requisitos de cada uma das metodologias e identificados os critérios que eram iguais ou equivalentes para suprimir e os que não atendiam tal condição eram considerados para o Modelo de Qualidade Castilho-Farias.
- **Suprimir requisito:** os requisitos que eram iguais ou equivalentes eram suprimidos para que não tivessem requisitos repetidos no Modelo de Qualidade proposto.
- **Definir como requisito do Modelo de Qualidade Castilho-Farias:** Após a análise de todos os requisitos das metodologias e após a etapa de suprimir requisitos, foi criada a base de dados com as 24 metodologias e os critérios relacionados a elas.

Para ilustrar os critérios que foram considerados na metodologia proposta e suas definições e referências, são apresentados na a Tabela 4 os critérios Socioculturais e na Tabela 5 os critérios Socioeconômicos.

Tabela 4 – Critérios Socioculturais

Critério	Definição
Atividades individuais e em grupos (Alencar e Marques, 2021)	Os aplicativos educacionais devem fornecer atividades individuais e em grupos, motivando o desenvolvimento de tarefas em conjunto.
Colaboração (Alencar e Marques, 2021)	Os aplicativos de educacionais devem prover ferramentas e mecanismos específicos de colaboração, como wiki, jogos, microblogs e grupos.
Compartilhamento (Alencar e Marques, 2021)	A colaboração deve estar presente nas aplicações educacionais, permitindo que os aprendizes e tutores se relacionem. Ela pode ser proporcionada por meio de wiki, jogos, microblogs, fóruns e grupos.
Curva de aprendizado (Soad e Barbosa, 2017)	O tempo em que o usuário consegue aprender a manusear a tecnologia móvel.

Harmonia entre as Atividades (Alencar e Marques, 2021)	Quando o material pedagógico inserido nas atividades está condizente com os objetivos e adequado à faixa etária proposta.
Idade do usuário (Alencar e Marques, 2021)	O aplicativo deve ser adaptável a fim de atender as diferentes idades dos usuários.
Perfil do usuário (Alencar e Marques, 2021)	Configurabilidade do aplicativo para tratar diferentes possíveis limitações de acordo com a idade, sexo, religião, raça, entre outras características dos usuários.
Satisfação do usuário (Alencar e Marques, 2021)	O aplicativo deve atingir as expectativas dos usuários, a fim de lhes proporcionar satisfação com o aplicativo.

Tabela 5 – Critérios Socioeconômicos

Critério	Definição
Custo-benefício (Soad e Barbosa, 2017)	Os custos de desenvolvimento, aquisição e manutenção devem ser baixos. Além disso, o aplicativo deve ser economicamente viável.
Custo para desenvolver o material de aprendizagem (Alencar e Marques, 2021)	Os gastos com o desenvolvimento do material de aprendizagem influenciam diretamente os gastos com a aplicação e se o seu custo for repassado para os usuários, implica em um aumento no investimento por parte do mesmo.
Nível de serviço (Alencar e Marques, 2021)	O estabelecimento de níveis de serviço possibilita maior segurança na utilização das aplicações e permite existir continuidade no seu uso.
Período de teste (Alencar e Marques, 2021)	Disponibilização de um período de teste do aplicativo ou permissão de acesso às funcionalidades básicas para permitir que o usuário experimente o aplicativo antes da aquisição.
Personalização (Soad e Barbosa, 2017)	Qualquer mudança ou ação no ambiente de aprendizagem deve ser adaptada para o usuário final de acordo com seu perfil, preferências e restrições. Além disso, as respostas a qualquer mudança (física ou ambiental) devem ser tratadas de forma clara.
Plano de continuidade (Soad e Barbosa, 2017)	Possuir estratégias e procedimentos de recuperação dos ambientes de aprendizagem móvel, evitando assim a interrupção das práticas de aprendizagem.

Suporte (Alencar e Marques, 2021)	Os serviços oferecidos pela aplicação educacionais devem possuir um suporte eficiente aos usuários. Relacionados a isso, dois pontos são considerados importantes: (i) qualquer mudança ou ação no ambiente educacionais deve ser adaptada para o usuário final de acordo com seu perfil, preferências e restrições; e (ii) as respostas a qualquer mudança (física ou ambiental) devem ser tratadas de forma clara.
Tempo para desenvolver o material de aprendizagem (Alencar e Marques, 2021)	O Adoção de mecanismos para facilitar o desenvolvimento do material de aprendizagem, diminuindo o seu tempo de criação e, conseqüentemente, seu custo.

Para definição dos critérios Pedagógicos do modelo de qualidade proposto, foram comparados com os critérios Pedagógicos estabelecidos por Alencar e Marques (2021), baseados no MoLEva, com critérios pedagógicos do modelo TACASE (JUNIOR; AGUIAR; MOURA, 2020), sendo feita a interseção dos critérios dos modelos. Foram excluídos os critérios que se repetiam nos modelos e os que tinham denominações diferentes, no entanto, eram similares durante o processo de avaliação de SE. Alencar e Marques, definiram 61 critérios e do modelo TACASE foram listados 42 critérios pedagógicos. Após a interseção e exclusão de critérios, foram empregados neste trabalho 85 critérios Pedagógicos. A Tabela 6 exemplifica alguns critérios Pedagógicos e no Apêndice A são listados todos os critérios Pedagógicos.

Tabela 6 – Critérios Pedagógicos

Critério	Definição
Acompanhamento (Alencar e Marques, 2021)	Uso de um conjunto de informações do cotidiano de ensino e aprendizado e informações de avaliações sumativas, para prover uma visão geral do progresso do aprendiz.
Adaptabilidade (Alencar e Marques, 2021)	Capacidade do aplicativo em se adaptar a diferentes tipos de dispositivos móveis.
Adaptação ao contexto (Alencar e Marques, 2021)	A aplicação deve ser capaz de se adaptar ao contexto do usuário, por exemplo, adaptar-se a um usuário do meio rural, com idade adulta e com escolaridade baixa.
Adequação funcional (Alencar e Marques, 2021)	Capacidade de disponibilizar funções que atendam às necessidades implícitas e explícitas em relação ao aplicativo.

Ambiente de aprendizagem (Alencar e Marques, 2021)	Capacidade de prover um ambiente que favoreça aprendizagem ao aluno, provendo maneiras para manter o interesse do aluno.
Aplicabilidade (Alencar e Marques, 2021)	Deve prover um sistema onde se possa empregar, colocar em prática ou usar para o objetivo definido a fim de promover ganho de conhecimento para o usuário.
Aplicação (Alencar e Marques, 2021)	Refere-se aos aspectos técnicos do aplicativo relacionados às questões pedagógicas.
Aprendizado Cooperativo (Alencar e Marques, 2021)	Possibilidade de realizar tarefas e atividades em grupo.
Aprendizagem (Alencar e Marques, 2021)	Capacidade do aplicativo em disponibilizar funcionalidades que contribuam para o aprendizado do aluno.
Aprendizagem de curto prazo (Alencar e Marques, 2021)	Oferece ambiente com capacidade do aluno armazenar e manipular uma quantidade limitada de informações por um período breve de tempo.
Aprendizagem e Recordação no processo de Interação (Alencar e Marques, 2021)	Relacionam-se com a condução as heurísticas de reconhecimento no lugar de memorização, qualidade das opções de ajuda, legibilidade, feedback e agrupamento e distinção de itens. Quando o usuário é orientado passo a passo nos procedimentos a serem seguidos.

Já os critérios Técnicos foram definidos utilizando-se os critérios Técnicos estabelecidos por Alencar e Marques(2021), baseados no MoLEva, e os critérios de qualidade de Uso e de *Software* estabelecidos pelo método TACASE (JUNIOR; AGUIAR; MOURA, 2020). Alencar e Marques, definiram 114 critérios Técnicos, já no método TACASE foram estabelecidos 111 critérios. Após a interseção e exclusão de critérios repetidos ou equivalentes resultou em 136 critérios Técnicos que são utilizados neste trabalho. A Tabela 7 exemplifica alguns critérios Técnicos e a totalidade dos critérios Técnicos estão listados na tabela do Apêndice A.

Tabela 7 – Critérios Técnicos

Critério	Definição
Acessibilidade (Alencar e Marques, 2021)	Mecanismos de acessibilidade, permitindo a utilização por pessoas com necessidades especiais.
Acessível (Alencar e Marques, 2021)	O aplicativo deve estar disponível e acessível a qualquer hora e em qualquer lugar.

Acesso (Alencar e Marques, 2021)	Deve-se prover acesso seguro e privado, ou seja, o usuário e seus dados devem ter a privacidade assegurada.
Acurácia (Alencar e Marques, 2021)	Fornecimento de resultados exatos e livre de erros.
Adaptabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo <i>software</i> considerado.
Adaptação às condições físicas (Alencar e Marques, 2021)	O grau em que o aplicativo se adapta às condições físicas, por exemplo, ajuste de luz, controle de ruído, condições climáticas, entre outros.
Adaptação de dispositivos (Alencar e Marques, 2021)	Grau em que o aplicativo consegue se adaptar a diferentes tipos de dispositivos móveis. Com a grande quantidade de dispositivos móveis e suas variadas características, o aplicativo deve conseguir se adaptar, permitindo assim o acesso ao aplicativo por diversos dispositivos móveis.
Adequação de mensagem à funcionalidade e ao usuário (Alencar e Marques, 2021)	Mensagens operacionais explicitam de forma compreensiva e objetiva informações da aplicação, sendo de fácil compreensão ao usuário.
Adequação funcional (Alencar e Marques, 2021)	Refere-se à disponibilidade de funções que atendam as necessidades implícitas e explícitas em relação à aplicação.

Desta forma, foram agrupados 8 (oito) critérios Socioculturais, 8 (oito) Socioeconômicos, 85 (oitenta e cinco) critérios Pedagógicos, 136 (cento e trinta e seis) critérios Técnicos. Após agrupados as características e critérios, foram relacionados ao perfil do usuário que avaliava o método nos estudos, que poderiam ser o Aluno, o Professor, o Instrutor e o Especialista.

Sendo assim, foi possível criar uma tabela relacional entre os métodos elencados pelos autores e os critérios que cada método avalia, renomeando os critérios nomeados de maneira diferente nos estudos, mas com a mesma definição. Sendo obtida uma tabela com 24 (vinte e quatro) métodos que poderiam ser utilizados por 4 (quatro) perfis de usuários e 4 (quatro) características com 85 (oitenta e cinco) critérios na característica Pedagógica, 136 (cento e trinta e seis) critérios na característica Técnica, 8 (oito) critérios na Socioculturais e 8 (oito) critérios na característica Socioeconômica. A Tabela 8 ilustra uma amostra dos dados da tabela criada com os dados do perfil do usuário, características, critérios/subcritérios e os métodos.

Tabela 8 – Amostra de registros da tabela criada com os dados

Usuário avaliador	Característica	Critério/subcritério	Método que avalia
Professor	Técnicas	Eficiência do Desempenho	TUP Model
Professor	Técnicas	Equipamentos de software e hardware	TUP Model
Professor	Técnicas	Utilização de recursos computacionais	TUP Model
Especialista	Técnicas	Compatibilidade	PETESE
Professor	Técnicas	Design de interface	Checklist de Porto
Professor	Técnicas	Dependências	TUP Model
Especialista	Técnicas	Manutenção	PETESE
Professor	Pedagógicas	Conteúdos matemáticos	Questionário de Silva e Batista
Professor	Pedagógicas	Aplicação	Questionário de Silva e Batista
Professor	Pedagógicas	Aplicação	Método de Lúlia e Simões
Especialista	Pedagógicas	Papel do professor	PETESE
Especialista	Socioeconômicas	Custo para desenvolver o material de aprendizagem	MoLEva
Especialista	Socioeconômicas	Período de teste	MoLEva
Especialista	Socioeconômicas	Tempo para desenvolver o material de aprendizagem	MoLEva
Especialista	Socioeconômicas	Nível de serviço	MoLEva
Especialista	Socioculturais	Harmonia entre as atividades	MoLEva
Especialista	Socioculturais	Perfil do usuário	MoLEva
Especialista	Socioculturais	Satisfação do usuário	MoLEva
Especialista	Socioculturais	Colaboração	Checklist de Soad
Professor	Socioculturais	Colaboração	TUP Model
Professor	Socioculturais	Colaboração	Método de Barbosa
Aluno	Socioculturais	Colaboração	Método de Barbosa

Das relações criadas resultou uma base de dados com 1046 registros, onde cada registro contém os atributos perfil do usuário avaliador, característica que seria avaliada, critério e método. Esta base de dados é a entrada para os algoritmos de IA que são utilizados para selecionar a metodologia de SE mais adequada ao contexto do usuário, desconsiderando a subjetividade na escolha.

Os métodos que utilizados no modelo de qualidade proposto para avaliação do SE após o relacionamento das características, critérios e perfil do usuário avaliador, estão entre os 24 (vinte e quatro), utilizados por Alencar e Marques (2021) e Junior et al. (2020), os quais são explanados nos Capítulo de Trabalhos Relacionados.

6 ETAPAS PARA DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

Neste capítulo serão apresentadas as etapas para descoberta de conhecimento em Base de Dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) utilizadas para automatizar a escolha do método de avaliação de *softwares* educacionais mais adequado, considerando o perfil do usuário avaliador, as características e critérios que o avaliador considera importante analisar, utilizando a base de dados criada no modelo de qualidade proposto no Capítulo 5 juntamente com a solução híbrida de IA (técnica de processamento de linguagem natural e algoritmos de classificação).

6.1 Aquisição de Dados

Os dados utilizados para no processo para automatizar a escolha do método de avaliação de *softwares* educacionais, foram os dados propostos no Modelo de Qualidade Castilho-Farias, conforme explanado no Capítulo 5.

6.2 Análise Exploratória de Dados

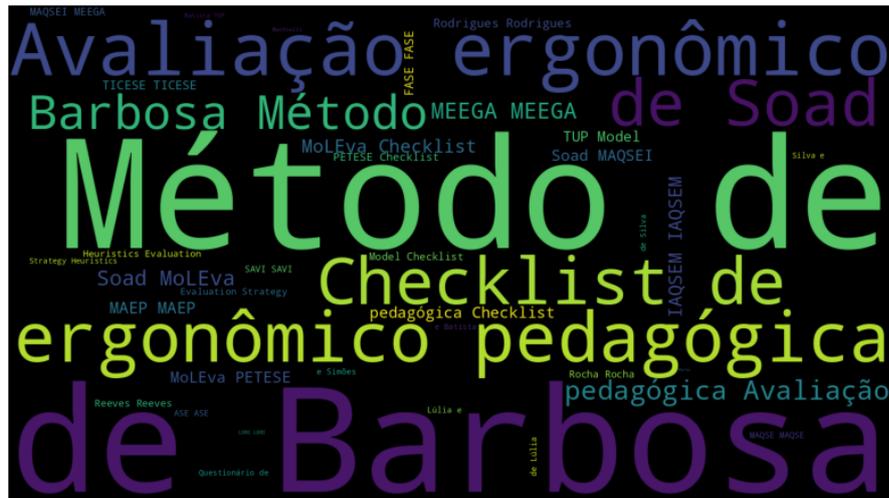
Uma ferramenta intuitiva que permite rápida análise de ocorrências de palavras em um documento é a chamada Nuvem de Palavras. Uma nuvem de palavras é uma forma de representar visualmente a frequência e o valor das palavras em um texto ou conjunto de dados. Ela mostra as palavras em diferentes tamanhos e cores, de acordo com a sua relevância ou importância (ATENSTAEDT, 2012). Na Figura 3 é exibido a Nuvem de Palavras para a variável de saída desejada, que é a variável Método.

A Nuvem de Palavras evidencia uma importante característica do banco de dados em questão, sendo ela a presença expressiva de preposições. Além disso, palavras com pouca relevância e que não contribuem para distinção do texto podem ser removidas, sendo esse procedimento chamado de remoção de *stopwords* (FRANCISCO, 2018). A Figura 4 ilustra a nuvem de palavras após a remoção das *stopwords*.

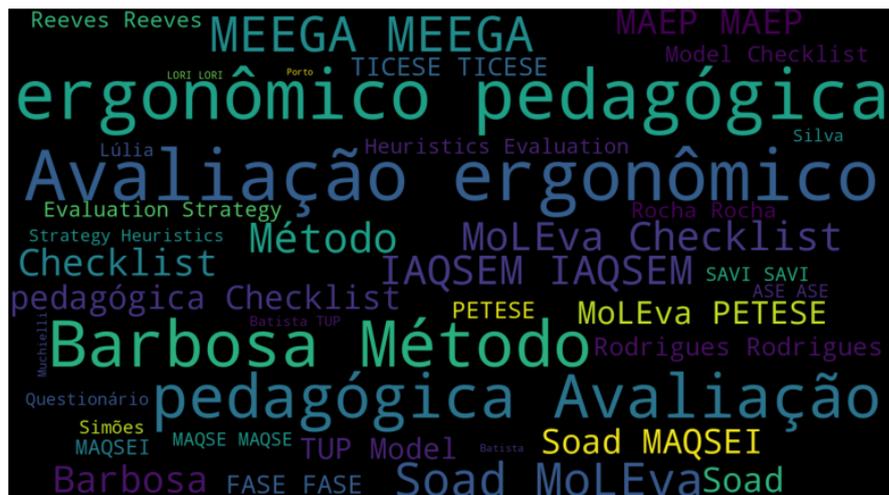
6.3 Pré-processamento de Dados

A etapa de pré-processamento de dados é uma etapa essencial para utilização de algoritmos de IA, uma vez que visa assegurar que os dados disponibilizados aos algoritmos sejam completos e consistentes, buscando encontrar o melhor desempenho e os melhores resultados.

Para Sivakumar e Gunasundari (2017), a etapa de pré-processamento envolve as seguintes fases:

Figura 3 – Nuvem de palavras da variável de saída desejada com stopwords

fonte: criada pela autora.

Figura 4 – Nuvem de Palavras após remoção de stopwords

fonte: criada pela autora.

- Limpeza dos dados;
- Integração dos dados;
- Transformação dos dados;
- Redução dos dados.

6.3.1 Limpeza dos dados

Consiste em identificar e resolver dados faltantes, errados, irrelevantes ou repetidos. Sivakumar e Gunasundari (2017) descrevem alternativas para corrigir estes problemas, tais como:

- Eliminar as linhas do banco de dados com valores em falta;
- Estimar o valor em falta com uma média ou mediana.

6.3.2 Integração dos dados

Consiste em unir dados de diferentes origens em uma única base de dados de informação. Os metadados das diversas bases de dados são usados para prevenir erros de validação, inconsistências e redundâncias nos atributos do banco de dados final (SIVAKUMAR; GUNASUNDARI, 2017).

6.3.3 Transformação dos dados

Esta etapa consiste em alterar os dados para que fiquem em formatos adequados para os modelos. Essa alteração inclui as seguintes etapas (SIVAKUMAR; GUNASUNDARI, 2017):

- Normalização dos dados;
- Técnicas de *Smoothing* como *clustering* e regressão;
- Agregação dos dados;
- Substituição do valor de atributos para o formato categórico.

6.3.3.1 Redução dos dados

Consiste em diminuir o tamanho ou a dimensão do banco de dados, sem comprometer a sua qualidade e integridade. Os bancos de dados podem ser muito grandes e complexos, o que pode dificultar o processo de análise e predição com os algoritmos. Por isso, frequentemente são aplicadas técnicas de redução do banco de dados, para obter um banco de dados menor e mais simples. Nessa fase, podem-se eliminar atributos desnecessários ou reduzir o número de exemplos fornecidos ao algoritmo. Algumas estratégias para realizar a redução de um conjunto de dados são as seguintes (SIVAKUMAR; GUNASUNDARI, 2017):

- Operações de agregação dos dados;
- Redução da dimensão do banco de dados pela eliminação de valores irrelevantes, redundantes, fracos, etc.;
- Compressão do banco de dados através de mecanismos de *encoding*;
- Redução da numerosidade dos dados: os dados são substituídos conjuntos menores como modelos paramétricos (regressão ou modelos Log-Linear) ou não paramétricos (*clustering*, *sampling* ou o uso de histogramas);

- *Concept Hierarchy Generation*: dados em formato bruto são substituídos por valores conceituais de maior nível, o que possibilita a mineração dos dados a múltiplos níveis de abstração.

6.4 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma área relevante para a IA e a Ciência da Computação. Esse campo de estudo ganha importância à medida que aumenta a quantidade de informações textuais disponíveis em formato digital na história. O PLN é um campo científico que abrange conceitos de ciência da computação, linguística e matemática com a finalidade principal de converter a linguagem humana em instruções que podem ser executadas por computadores. Apesar de existirem várias técnicas consolidadas de PLN, a linguagem natural ainda é um desafio para o processamento. Por exemplo, é possível que a omissão de palavras e a má gramática não prejudiquem a comunicação entre duas pessoas. Para os computadores, contudo, isso é um grande empecilho (KANG et al., 2020). O PLN pode ser dividido em quatro fases: pré-processamento, vetorização, treinamento do modelo e avaliação do modelo (KANG et al., 2020). O pré-processamento de texto tem como objetivo obter um texto “limpo”, eliminando símbolos e outros elementos que possam causar ruído na análise posterior. Depois do pré-processamento do texto, é escolhido um algoritmo que transforme palavras em uma representação numérica, como vetores. Com base nesses vetores de palavras, é possível aplicar algoritmos para treinar um modelo que possa resolver o problema real. Por último, após o treinamento do modelo, é necessário sua avaliação para verificar sua qualidade e sua adaptabilidade em outros cenários.

6.4.1 Algoritmos de Vetorização *Embedding*

Os computadores, mesmo sendo capazes de realizar uma grande variedade de tarefas envolvendo multimídia, ainda são máquinas operando instruções sobre números no sistema binário. Assim, para possibilitar a classificação de textos com recursos computacionais, é necessário a escolha de alguma estratégia de tradução que possa representar os conjuntos de palavras do documento em forma numérica, sem que haja perda de características importantes (SINGH; SHASHI, 2019).

Existem duas categorias principais de estratégias de conversão de textos em valores numéricos: a vetorização e o *embedding*. A vetorização utiliza técnicas estatísticas para associar cada palavra de um texto a um número que represente sua importância no texto, utilizando-se de uma ou mais métricas para seu cálculo (RAJARAMAN; ULLMAN, 2011). Já no *embedding*, cada palavra é associada a um vetor em um espaço multidimensional, de tal forma que seu valor semântico é mantido, permitindo que relações semânticas sejam definidas a partir de operações vetoriais nos vetores associados (LE; MIKOLOV, 2014).

Entre os algoritmos de vetorização e *embedding* mais utilizados, podemos citar *Term Frequency times Inverse Document Frequency* (TF-IDF), Word2Vec, Doc2Vec e GloVe. O TF-IDF que foi utilizado neste trabalho será detalhado na próxima seção.

6.4.1.1 TF-IDF

O algoritmo TF-IDF (*Term Frequency times Inverse Document Frequency*) é uma técnica de vetorização que calcula a relevância de uma palavra em um documento, considerando um conjunto de documentos sobre um determinado assunto. Essa técnica é muito usada em tarefas de processamento de linguagem natural, como busca de informação e análise de sentimentos (RAJARAMAN; ULLMAN, 2011).

O algoritmo usa duas medidas: a frequência do termo (TF) e a frequência inversa do documento (IDF). A frequência do termo (TF) é o número de vezes que um termo aparece em um documento, dividido pelo número total de palavras no documento, conforme a Eq. 6.1:

$$TF_{ij} = \frac{f_{ij}}{\max_k (f_{kj})} \quad (6.1)$$

onde:

f_{ij} : Número de ocorrências do termo i no texto j .

$\max_k (f_{kj})$: Maior índice k das palavras do texto j , ou seja, a quantidade total de palavras no texto j .

Já a frequência inversa do documento (IDF) é a medida de quanto um termo é raro em relação a um conjunto de documentos, definido pela Eq. 6.2:

$$IDF_i = \log_2 \frac{N}{n_i} \quad (6.2)$$

onde:

N : Número total de documentos;

n_i : Número de documentos que contém o termo i .

A relevância de um termo em um documento é obtida pela multiplicação das medidas TF e IDF (RAJARAMAN; ULLMAN, 2011). Assim, o valor $TF - IDF$ para cada palavra i do documento j é dado pela Eq. 6.3:

$$TF - IDF_{ij} = TF_{ij} \cdot IDF_i \quad (6.3)$$

Por exemplo, imagine que temos um conjunto de documentos sobre tecnologia da informação, e queremos saber a relevância de uma palavra específica em cada documento. Se

a palavra “algoritmo” aparecer 10 vezes no documento A, que tem um total de 1000 palavras, temos a Eq. 6.4:

$$TF = \frac{10}{1000} = 0,01 \quad (6.4)$$

Se a palavra “algoritmo” aparecer em 125 dos 1000 documentos que temos, então Eq. 6.5:

$$IDF = \log_2 \frac{1000}{125} = \log_2 8 = 3 \quad (6.5)$$

O $TF - IDF$ para a palavra “algoritmo” no documento A será Eq. 6.6:

$$TF - IDF = 0,01 \cdot 3 = 0,03 \quad (6.6)$$

Com isso, podemos calcular o $TF - IDF$ para cada palavra em cada documento e, depois, usar esses valores para identificar as palavras mais importantes em cada documento e também comparar a importância das palavras em diferentes documentos.

Um termo é mais importante para um documento se ele ocorre frequentemente no documento e raramente ocorre nos outros documentos. Isso faz com que o algoritmo $TF - IDF$ consiga ressaltar palavras-chave relevantes em um documento, o que é útil em tarefas de busca de informação e indexação de documentos (RAJARAMAN; ULLMAN, 2011).

6.5 Inteligência Artificial Aplicada

6.5.1 Modelo *K-Nearest Neighbor*

O modelo do *K-Nearest Neighbor* (K-NN) é um modelo não paramétrico e que se baseia na ideia de distância, isto é, na proximidade entre os dados. É um método simples, mas muito usado e que costuma ter bons resultados. Segundo Faceli et al. (2011), a hipótese fundamental é que dados parecidos tendem a estar próximos no espaço de entrada e, por outro lado, dados diferentes tendem a estar afastados. O algoritmo atribui uma classe a um dado elemento conforme as classes dos k vizinhos mais próximos que pertencem ao conjunto de dados de treinamento.

É fácil notar que não há trabalho computacional na fase de treinamento, pois o algoritmo não aprende um modelo para os dados, apenas guarda os objetos de treinamento para depois calcular a distância de cada um deles até os objetos do conjunto de teste. Após calcular, as distâncias até cada novo objeto são ordenadas de forma ascendente e escolhe-se somente os k primeiros elementos de treinamento para fazer a classificação.

Dois aspectos importantes no k -vizinhos mais próximos são a escolha (1) da função que mede a distância entre os objetos e (2) da regra de classificação, ou seja, do valor de k . As mudanças que o algoritmo pode sofrer, então, estão relacionadas aos diferentes valores que o parâmetro k pode ter e às diferentes medidas de distância que podem ser aplicadas.

Existem vários métodos para calcular essa distância, dos quais os métodos mais conhecidos são - distância Euclidiana, Manhattan (para contínuo) e distância de Hamming (para categórico).

Distância Euclidiana: A distância Euclidiana é calculada como a raiz quadrada da soma das diferenças quadráticas entre um novo ponto x e um ponto existente y , conforme mostra a Eq. 6.7.

$$D_E = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (6.7)$$

onde:

D_{x_i, y_i} é a distância entre os objetos x_i e y_i dos conjuntos de treinamento e teste, respectivamente, representados por vetores cujos elementos são os valores de cada um dos d atributos. No 1-NN, como também é chamado, a classe associada ao objeto de treinamento mais perto é usada para marcar o novo objeto.

Distância Manhattan: Esta é a distância entre vetores reais usando a soma de sua diferença absoluta, dada pela Eq. 6.8.

$$D_M = \sum_{i=1}^k |x_i - y_i| \quad (6.8)$$

Distância Hamming: Ele é usado para variáveis categóricas, conforme Eq. 6.9. Se o valor (x) e o valor (y) forem iguais, a distância D_H será igual a 0. Caso contrário, $D_H = 1$.

$$D_H = \sum_{i=1}^k |x_i - y_i| \quad (6.9)$$

$$D_H = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}$$

Faceli et al. (2011) afirmam que a forma mais simples do algoritmo KNN é o 1-vizinho mais próximo e a métrica de distância mais usada é a distância Euclidiana.

Em vez de considerar 1 vizinho mais próximo, pode-se considerar k vizinhos mais próximos, como foi dito. Nesse caso, Faceli et al. (2011) dizem que as previsões dos vários vizinhos são agregadas para classificar o ponto de teste. Em problemas de classificação, cada vizinho escolhe uma classe e o objeto de teste é classificado conforme a classe mais escolhida - em problemas de regressão, outras estratégias podem ser usadas.

Para James et al. (2013), o algoritmo KNN é um método que, dado um número inteiro positivo k e uma observação x_0 do conjunto de teste, estima a probabilidade condicional da

classe c como a fração dos k pontos do conjunto de treinamento mais próximos à x_0 cuja variável resposta é igual a c e associa à observação de teste a classe cuja probabilidade é maior, conforme a Eq. 6.10.

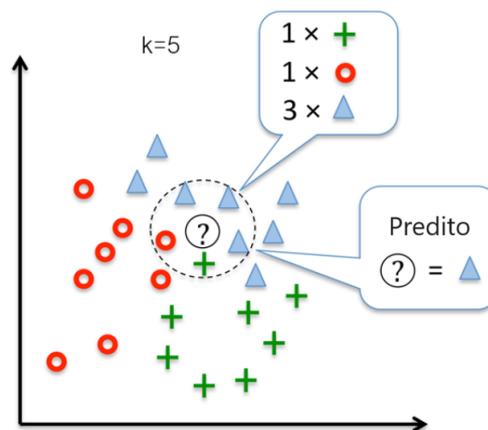
$$Pr(y = c|x = x_0) = \frac{1}{k} \sqrt{\sum_{i \in v_0}^d I(y_i = c)} \quad (6.10)$$

onde: v_0 o conjunto formado pelos k vizinhos mais próximos.

Segundo Faceli et al. (2011), o KNN é um algoritmo preguiçoso, porque toda a computação é deixada para a fase de classificação. A previsão, porém, pode ser cara, pois é preciso medir a distância de um novo objeto até cada objeto de treinamento. Por isso, para um conjunto grande de dados esse processo pode ser lento. Por ser um método baseado em distância, o desempenho pode ser prejudicado pela presença de atributos desnecessários e sem importância. Apesar disso, o algoritmo é simples, pode ser usado mesmo em problemas complicados e é incremental, ou seja, quando novos exemplos de treinamento estão disponíveis, basta guardá-los na memória.

A Figura 5 mostra as disposições dos pontos em um gráfico de dispersão com três classes para k -vizinhos mais próximos igual a 5 com o objetivo de classificar se o novo dado pertence a classe círculo vermelho, triângulo azul ou símbolo verde.

Figura 5 – Exemplo de classificação do KNN com três rótulos de classe e $k=5$.



Fonte: (HOSSSEN, 2022), adaptado.

Desta forma, o funcionamento do KNN pode ser explicado com base no algoritmo abaixo:

Passo 1: Selecione o número K dos vizinhos.

Passo 2: Calcular a distância Euclidiana do número K de vizinhos.

Passo 3: Pegue os vizinhos K mais próximos de acordo com a distância Euclidiana calculada.

Passo 4: Entre esses k vizinhos, conte o número de pontos de dados em cada categoria.

Passo 5: Atribua os novos pontos de dados à categoria para a qual o número do vizinho é máximo.

Passo 6: O modelo está pronto.

A Tabela 9 sintetiza algumas vantagens e desvantagens da utilização do algoritmo KNN para treinar modelos.

Tabela 9 – Vantagens e desvantagens do *K-Nearest Neighbor*

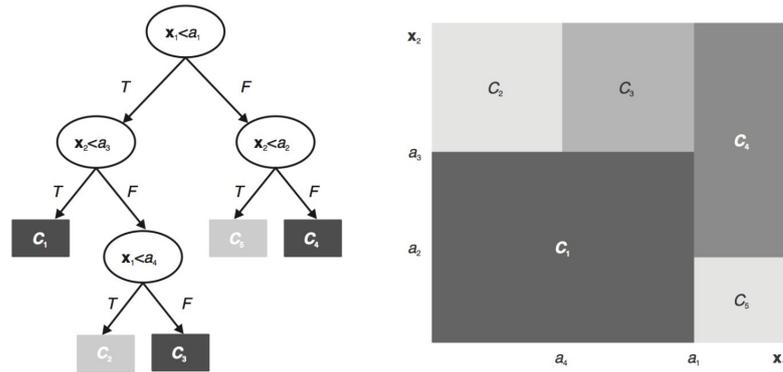
Vantagens	Desvantagens
O KNN não faz suposições sobre a distribuição dos dados, o que é uma vantagem sobre outros métodos paramétricos, como regressão logística e análise discriminante linear	O KNN requer uma grande quantidade de memória para armazenar todos os pontos de dados de treinamento, uma vez que é preciso calcular a distância entre um novo ponto de dados e todos os pontos de dados de treinamento. Isso pode ser problemático ao trabalhar com grandes conjuntos de dados.
KNN pode ser usado para problemas de classificação e regressão	Outra desvantagem do KNN é que pode ser computacionalmente caro calcular a distância entre um novo ponto de dados e todos os pontos de dados de treinamento.
KNN funciona bem com pequenos conjuntos de dados, bem como grandes conjuntos de dados	KNN pode não funcionar bem com dados dimensionais altos, uma vez que a métrica de distância pode se tornar distorcida em dimensões mais altas.
KNN é um algoritmo muito simples de entender e implementar	
O KNN também não é sensível a <i>outliers</i> nos dados	

6.5.2 Modelo *Random Forest*

Random Forest são um tipo de algoritmo de IA usado para tarefas de classificação e regressão. O modelo classificador recebe a entrada de dados e a atribui a uma das várias categorias. Um algoritmo *Random Forest* funciona criando várias árvores de decisão, cada uma das quais é baseada em um subconjunto aleatório dos dados. As árvores de decisão são um tipo de algoritmo que faz previsões observando as entradas de dados e determinando a qual categoria elas pertencem. As *Random Forests* levam isso um passo adiante, criando várias árvores de decisão e, em seguida, calculando a média de seus resultados. Isso ajuda a reduzir a chance de *overfitting*, que é quando o algoritmo só funciona bem nos dados de treinamento e não em novos dados.

Formalmente, uma árvore de decisão é um grafo acíclico direcionado em que cada nó é um nó de divisão, com dois ou mais sucessores, ou um nó folha (FACELI et al., 2011). Segundo os mesmos autores, um nó folha é marcado com uma função e normalmente contém apenas os valores da variável resposta. Um nó de divisão, por outro lado, contém um teste condicional baseado nos valores das variáveis independentes. A Figura 6 mostra uma árvore de decisão e o respectivo espaço particionado pelos atributos x_1 e x_2 .

Figura 6 – a) Árvore de decisão e b) as regiões de decisão no espaço de atributos x_1 e x_2 .



Fonte: (FACELI et al., 2011).

Como enfatizam James et al. (2013), computacionalmente não é possível considerar toda partição possível do espaço das variáveis preditoras e, por isso, uma abordagem chamada de divisão recursiva binária, ou *recursive binary splitting*, é usada. Essa é uma abordagem em que a divisão começa no topo da árvore e a melhor partição é escolhida a cada passo do processo. Várias medidas podem ser usadas para determinar a melhor partição, e elas são definidas em termos da distribuição das classes antes e depois da divisão.

Em um problema binário, a distribuição das classes em qualquer nó, ou partição, pode ser escrita como (p_0, p_1) , onde $p_1 = 1 - p_0$ e $p(c|t)$, ou simplesmente $p(c)$, indica a fração de observações que pertencem à classe c em uma dada partição t . A distribuição das classes determina o grau de impureza dos nós, de modo que quanto menor o grau de impureza, mais enviesada é a distribuição. Um nó com distribuição $(0, 1)$, por exemplo, tem impureza zero, enquanto que um nó com classes igualmente distribuídas tem a maior impureza possível. Entre as medidas de impureza, estão a Entropia, dada pela Eq. 6.11, a Gini, dada pela Eq. 6.12 e o Erro de Classificação, dado pela Eq. 6.13:

$$Entropia(t) = - \sum_{c=1}^l p(c) \log_2 p(c) \tag{6.11}$$

$$Gini(t) = 1 - \sum_{c=1}^l [p(c)]^2 \tag{6.12}$$

$$\text{Erro de Classificação } (t) = 1 - \sum_{c=1}^l [p(c)]^2 \tag{6.13}$$

onde l é o número de classes e $0.\log_2 0 = 0$ no cálculo da *Entropia*. O índice de *Gini* é uma medida de variância total dentre as classes e assume valores pequenos se as proporções $p(c)$ são próximas de 0 ou 1, indicando que um nó contém predominantemente observações de uma única classe (JAMES; WITTEN, 2013). A *Entropia*, por sua vez, toma valores próximos de 0 se as proporções $p(c)$ são próximas de 0 ou 1, indicando que, assim como o índice de Gini, esta medida assumirá valores pequenos se um nó é puro.

As principais etapas para execução do algoritmo de *Random Forest* são:

Passo 1: Selecione uma amostra aleatória de tamanho n (*Bootstrap*).

Passo 2: Crie árvores de decisão a partir do Passo 1, com base no seguinte:

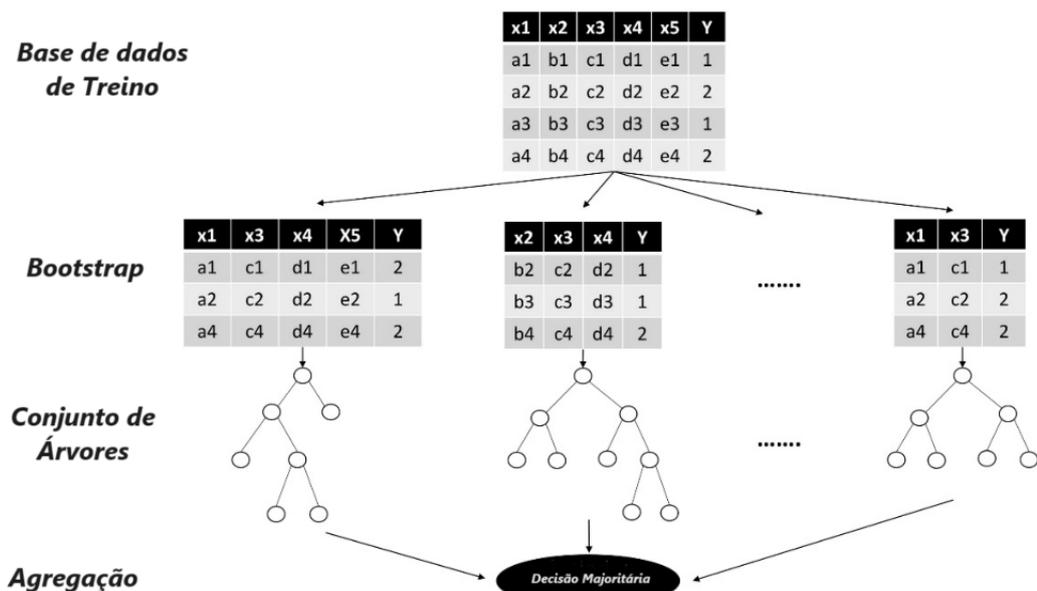
- Dentre as amostras n , selecione m recursos de forma aleatória;
- Crie a árvore dividindo os dados usando recursos m com base na função objetivo (maximizando o ganho de informações).

Passo 3: Repita a etapa acima para o número de árvores especificadas.

Passo 4: Agregue o resultado da previsão de diferentes árvores e chegue a uma previsão final com base na votação da maioria ou na média.

A Figura 7 exemplifica as etapas para execução do algoritmo *Random Forest*:

Figura 7 – Passos do algoritmo *Random Forest*.



Fonte: (MISRA; LI; HE, 2020), adaptado.

É possível observar que amostras aleatórias de dados (usando amostragem de *Bootstrap*)

com diferentes conjuntos de recursos são escolhidas e usadas para criar árvores de decisão de diferentes tamanhos. É por isso que esse conjunto de árvores é chamado de *Random Forest*. A previsão é uma agregação da saída de classificação de cada uma das árvores de decisão.

A Tabela 10 sintetiza algumas vantagens e desvantagens da utilização do algoritmo *Random Forest* para treinar modelos.

Tabela 10 – Vantagens e desvantagens do *Random Forest*

Vantagens	Desvantagens
<i>Random Forest</i> são mais precisas do que as árvores de decisão porque reduzem a variância do modelo e têm menor probabilidade de sobreajuste. Isso é feito pela média das previsões das árvores individuais.	Os classificadores de <i>Random Forest</i> podem demorar a treinar. No entanto, a precisão e a flexibilidade dos modelos <i>Random Forest</i> fazem com que valha a pena o investimento de tempo extra.
<i>Random Forest</i> também são mais fáceis de ajustar do que árvores de decisão.	Os classificadores <i>Random Forest</i> podem ser difíceis de interpretar.
<i>Random Forest</i> também podem lidar com valores ausentes e <i>outliers</i> melhor do que árvores de decisão.	
Os classificadores <i>Random Forest</i> são fáceis de ajustar com a ajuda do ajuste de hiperparâmetros.	

6.5.3 Modelo Redes Neurais Artificiais

Redes Neurais Artificiais (RNAs) são sistemas computacionais que imitam o funcionamento do cérebro humano, usando unidades de processamento simples que podem aprender com dados, armazenar conhecimento e aplicá-lo em diversas tarefas como reconhecimento de padrões, tomada de decisões e previsão em ambientes dinâmicos. O algoritmo de aprendizado é o procedimento que ajusta os pesos das conexões entre as unidades de processamento, de forma a alcançar um objetivo desejado.

McCulloch e Pitts (1943) elaboraram o primeiro modelo matemático para representar o neurônio biológico (MCCULLOCH; PITTS, 1943). O modelo proposto extraía apenas o essencial para representar o neurônio biológico com precisão, removendo todos os detalhes desnecessários (MARSLAND, 2011). Esse modelo artificial proposto, chamado McCulloch Pitts (MCP), pode ser visto na Figura 8 b).

Rosenblatt (1958) introduziu o conceito de aprendizado em RNA. Seu modelo, conhecido por *Perceptron*, era basicamente um neurônio MCP com algoritmo de treinamento, capaz de

ajustar os pesos de entrada. O neurônio recebe um conjunto de sinais de entrada, transforma esses sinais em uma combinação linear e gera um sinal de saída (ROSENBLATT, 1958).

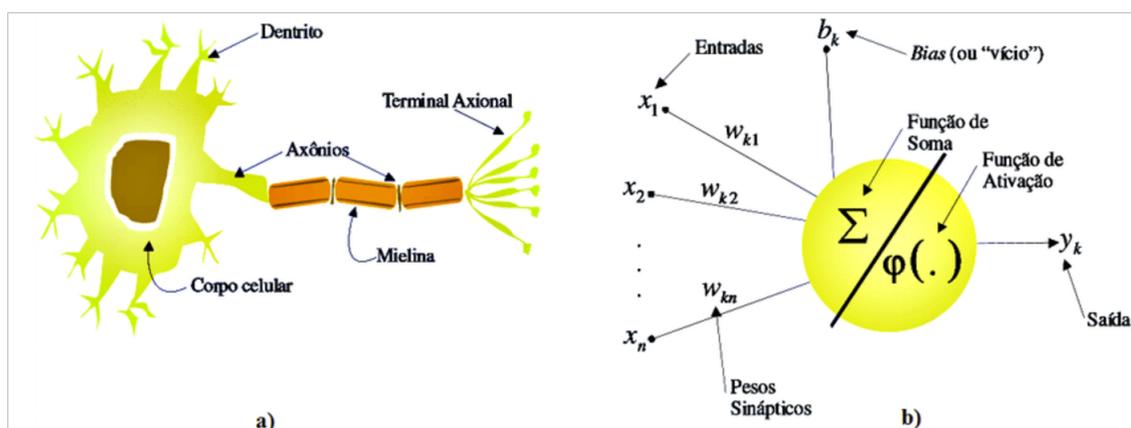
Um *Perceptron* é a unidade mais fundamental que é usada para construir uma rede neural e se assemelha a um neurônio no cérebro (ROSENBLATT, 1958). No caso de um neurônio biológico, vários sinais de entrada são alimentados em um neurônio via dendrito, e um sinal de saída é disparado apropriadamente com base na força do sinal de entrada e em algum outro mecanismo. A Figura 8 a) representa como os sinais de entrada passam através do neurônio biológico.

Computacionalmente, o neurônio na RNA é uma unidade que recebe como entrada x_1 , x_2 , x_n e um termo de interceptação. A saída y pode ser obtido pela Eq. 6.14:

$$y_k = f \left(\sum_{j=1}^n W_{kj} x_j + b_k \right) \quad (6.14)$$

onde f é chamada de função de ativação, muitas vezes escolhida para ser a função sigmóide; W são os parâmetros do modelo (ou pesos); b é um escalar. Uma RNA interliga muitos neurônios e a saída de um neurônio pode ser a entrada de outro. Os pesos W são obtidos por um procedimento de treinamento iterativo, baseado em padrões de entrada-saída conhecidos. As RNAs implementam algoritmos que tentam alcançar um desempenho neurológico relacionado, como aprender com a experiência, fazer generalizações a partir de situações semelhantes e julgar estados onde os resultados ruins foram alcançados no passado (LIU et al., 2018). A Figura 8 b) representa o modelo de um neurônio artificial.

Figura 8 – a) Modelo de neurônio biológico b) Modelo de neurônio artificial.



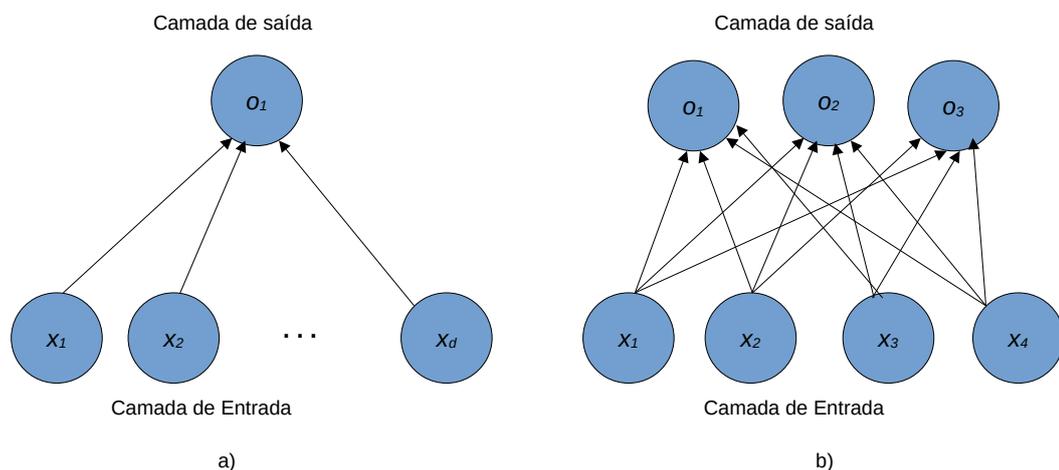
Fonte: (MOTA, 2007)

Um *Perceptron* é uma representação de como as redes neurais funcionam, ele também pode ser chamado de rede neural de camada única. Em um modelo de neurônio artificial três elementos são essenciais: o conjunto de sinapses, o somador e a função de ativação. Os sinais entram e saem de cada neurônio pelas sinapses, que são as conexões. Cada sinapse tem um peso

associado que corresponde à informação armazenada no neurônio e na rede. O somador soma os sinais de entrada, ponderando os pesos sinápticos do neurônio. A função de ativação ou função de esmagamento limita a amplitude do sinal de saída do neurônio para um valor finito (HAYKIN, 2001).

Os *Perceptrons* dispostos na forma de redes neurais de camada única ou multicamadas podem ser usados para executar tarefas de regressão e classificação. A Figura 9 a) ilustra uma rede neural de camada única que representa a regressão linear e a b) regressão *Softmax*. A regressão *Softmax* é uma forma de regressão logística que normaliza um valor de entrada em um vetor de valores que segue uma distribuição de probabilidade cujo total soma até 1. Os valores de saída estão no intervalo $[0,1]$. Isso permite que a rede neural preveja quantas classes ou dimensões forem necessárias. É por isso que a regressão *Softmax* é por vezes referida como uma regressão logística multinomial.

Figura 9 – a) Camada única b) Multicamadas.

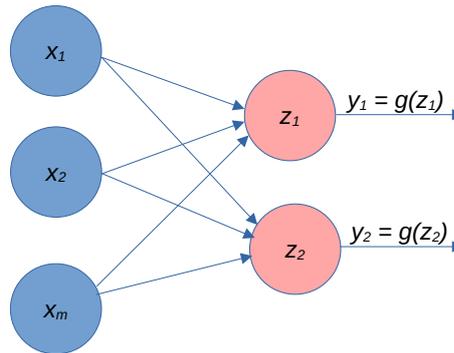


Fonte: (KUMAR, 2023), adaptado.

A Figura 10 representa um *Perceptron* multi-saída. Neste exemplo, o *Perceptron* é empilhado e há duas saídas. Além disso, os perceptrons estão totalmente conectados, por exemplo, cada saída é conectada com cada entrada. As camadas que consistem em *perceptrons* que estão conectados a todas as entradas são chamadas de camadas densas ou camadas totalmente conectadas. A camada densa também é chamada de pilha de perceptrons. Uma configuração como a da Figura 10 com vários neurônios de saída pode ser usada para resolver problemas de classificação multiclasse.

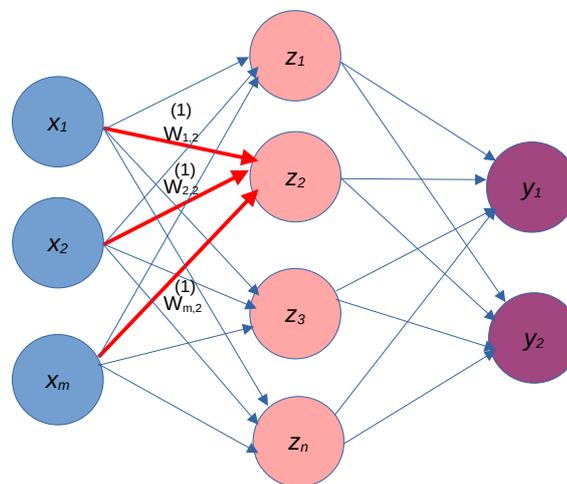
Segundo Liu et al. (2018), a RNA é o algoritmo mais usado. Para sua forma mais popular, o *Multi Layer Perceptron* (MLP) ou *perceptron* de múltiplas camadas, existem três componentes: camada de entrada, camada oculta e camada de saída. As unidades na camada oculta são chamadas de unidades ocultas, porque seus valores não são observados. A Figura 11 é um modelo simples de RNA, um *Multi Layer Perceptron* (MLP) e a entrada em z_2 é calcula conforme a Eq. 6.15 e a Eq. 6.16.

Figura 10 – Perceptron multi-saídas.



Fonte: (KUMAR, 2023), adaptado.

Figura 11 – Rede Neural Perceptron Multi-camada.



Fonte: (KUMAR, 2023), adaptado.

$$z_2 = w_{0,2}^{(1)} + \sum_{j=1}^m x_j w_{j,2}^{(1)} \quad (6.15)$$

$$z_2 = w_{0,2}^{(1)} + x_1 w_{1,2}^{(1)} + x_2 w_{2,2}^{(1)} + x_m w_{m,2}^{(1)} \quad (6.16)$$

As RNAs têm propriedades e capacidades únicas como a não linearidade, o mapeamento de entradas em saídas, a adaptabilidade, a tolerância ao erro e a uniformidade da análise e design, sendo uma ferramenta poderosa para resolver problemas (HAYKIN, 2001).

6.6 Métricas de Desempenho

Classificadores são treinados para reduzir erros. Para avaliar o erro uma ou mais métricas são utilizadas, essa escolha tem sido motivo de debate contínuo na pesquisa e na indústria muitos anos (YACOUBY; AXMAN, 2020).

6.6.1 Matriz de Confusão

Para medir a qualidade dos resultados, é preciso comparar as respostas que os modelos geraram com as respostas reais, ou seja, os rótulos originais que se tem. Uma forma simples e muito usada para avaliar os resultados de problemas de classificação binária é a matriz de confusão - e as medidas de desempenho que dela se derivam. A matriz de confusão, mostrada na Figura 12, é um conceito simples que apresenta as quantidades previstas e observadas em cada classe da variável resposta. Na Figura 12, VP são os verdadeiros positivos, ou seja, número de exemplos da classe positiva que foram classificados corretamente; VN são os verdadeiros negativos, ou seja, número de exemplos da classe negativa que foram classificados corretamente; FP são os falsos positivos, ou seja, número de exemplos da classe negativa que foram classificados erroneamente pelo modelo; e FN são os falsos negativos, ou seja, número de exemplos da classe positiva que foram classificados erroneamente.

Figura 12 – Matriz de Confusão.

		Classe Predita	
		+	-
Classe Verdadeira	+	VP	FN
	-	FP	VN

Fonte: (FACELI et al., 2011)

Na matriz de confusão o número de acertos, para cada classe, se localiza na diagonal principal $M(C_i, C_i)$ da matriz. Os demais elementos $M(C_i, C_j)$, para $i \neq j$, representam erros na classificação. A matriz de confusão de um classificador ideal possui todos os elementos fora da diagonal iguais a zero, uma vez que a classificação não comete erros.

As seguintes medidas de desempenho, baseadas na matriz de confusão e usadas neste trabalho, são descritas abaixo:

- Erro Médio Absoluto (*Mean Absoluto Error* - MAE): é a média da diferença entre o valores reais com os preditos. No entanto, um módulo é adicionado por existirem valores positivos e negativos. Além disso, esta métrica não é afetada por *outliers*. Representada pela Eq. 6.17:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6.17)$$

- Acurácia (*Accuracy*- ACC): é a proporção da soma de todos os casos classificados corretamente, sem distinção de classe, pelo número total de objetos no conjunto de teste. Ou seja, é o complemento da taxa de erro e, portanto, calculada pela Eq. 6.18:

$$ACC = 1 - ER = \frac{VP + VN}{VP + VN + FN + FP} \quad (6.18)$$

• Sensibilidade/*Recall* (*Sensitivity* - SEN): altas taxas de sensibilidade, conforme Viana (2015), indicam modelos bons em detectar o evento de interesse, ou seja, em detectar o evento quando ele realmente ocorreu. Ela corresponde, portanto, à taxa de acerto na classe positiva verdadeira e é obtida pela Eq. 6.19:

$$SEN = \frac{VP}{VP + FN} \quad (6.19)$$

• Especificidade (*Specificity* - SPE): a especificidade é complementar à sensibilidade e, portanto, corresponde à taxa de acerto na classe negativa, dada pela Eq. 6.20. Segundo Viana (2015), ela mostra o quão bom o modelo é para não detectar evento quando ele realmente não ocorreu:

$$SPE = 1 - SEN = \frac{VN}{VN + FP} \quad (6.20)$$

• Taxa de Falso Negativo *False Negative Rate* - FNR): também chamada de erro do tipo I, reflete a proporção dos casos positivos classificados incorretamente e fornece, como destaca Viana (2015), a probabilidade do modelo não detectar o evento quando ele ocorreu. A expressão é a Eq. 6.21:

$$FNR = 1 - SEN = \frac{FN}{VP + FN} \quad (6.21)$$

• Taxa de Falsos Positivos (*False Positive Rate* - FPR): conhecida como erro do tipo II, reflete a proporção de casos negativos classificados incorretamente. Estima, portanto, a probabilidade do modelo detectar o evento quando ele não ocorreu. É equacionada como Eq. 6.22:

$$FPR = 1 - SPE = \frac{FP}{VN + FP} \quad (6.22)$$

• Precisão (PRE): é a porcentagem de todos os exemplos rotulados como positivos que eram verdadeiramente positivos. Representada pela Eq. 6.23

$$PRE = \frac{VP}{VP + FP} \quad (6.23)$$

• O F1-score: é a média harmônica entre precisão e *recall* (DALIANIS, 2018). É representada pela Eq. 6.24.

$$F1 = \frac{2 * PRE * SEN}{PRE + SEN} \quad (6.24)$$

7 ANÁLISE DOS RESULTADOS

Inicialmente, foi realizada a aquisição dos dados, conforme explicado no item 6.1. Na Tabela 8 é exibido um extrato dos registros que compõem a planilha de dados criada, a partir da qual foi feita a transformação dos dados para *.csv* e a importação no ambiente de desenvolvimento *Colab* utilizando a linguagem de programação *Python*.

Em seguida, foi realizada uma breve análise exploratória dos dados, buscando direcionar a etapa posterior, que era o pré-processamento dos dados. Para realização desta análise foi utilizada a técnica de nuvem de palavras, além de alguns contadores que foram aplicados ao banco de dados. Na análise da nuvem de palavras foram removidos os chamados *stopwords*, isto é, conjunto composto por preposições e artigos. A Figura 13 exibe o código que foi utilizado para a exclusão das *stopwords*.

Figura 13 – Código para remoção de *stopwords*.

```
1 # lista de stopword
2 stopwords = set(STOPWORDS)
3 stopwords.update(["da", "e", "que", "a", "de", "ao", "os", "na", "do", "dos"])
```

fonte: criada pela autora.

No pré-processamento dos dados foram também removidos os atributos nulos e linhas duplicadas. Na Figura 14 é mostrado o código utilizado para o pré-processamento. Tais edições resultaram na redução da base de dados de 1046 registros para 1037 registros. Em virtude da base de dados ter sido oriunda do modelo de qualidade proposto por Castilho-Farias, a base não continha muitos dados nulos, brancos e linhas duplicadas. Sendo necessária a aplicação de poucas técnicas de pré-processamento na base de dados.

Figura 14 – Código para realizar o pré-processamento.

```
1 dataset_sem_duplicadas = df.drop_duplicates()
2 print("Dataset sem duplicadas:")
3 print(dataset_sem_duplicadas, "\n")
4
5 # verificando a presença de atributos nulos na base de dados
6 dataset.isnull().sum()
7
8 # eliminar as colunas com valores ausentes
9 metodos = dataset.dropna(subset=['Metodo que avalia'], axis=0)[Metodo que avalia']
```

fonte: criada pela autora.

Em seguida, foi feita a conversão do conteúdo da base de dados em correspondentes numéricos. Primeiramente cada texto foi convertido em um vetor de palavras. A Figura 15

exemplifica a base de dados representada por um vetor de palavras, em vermelho estão marcadas as palavras nas cinco primeiras linhas da matriz de dados que serão mostradas nas representações numéricas posteriores.

Figura 15 – Vetor de palavras com a base de dados.

```
['Aluno Pedagógicas Acompanhamento Avaliação ergonômico-pedagógica',
'Especialista Pedagógicas Acompanhamento Avaliação ergonômico-pedagógica',
'Professor Pedagógicas Acompanhamento Avaliação ergonômico-pedagógica',
'Monitor Pedagógicas Acompanhamento Avaliação ergonômico-pedagógica',
'Especialista Pedagógicas Acompanhamento Checklist Soad',
...
'Professor Técnicas Controle usuário TICESE',
'Professor Técnicas Recuperabilidade TICESE',
'Especialista Socioeconômicas Custo-benefício Checklist Soad',
'Especialista Socioeconômicas Personalização Checklist Soad',
'Especialista Socioeconômicas Plano de continuidade Checklist Soad']
```

fonte: criada pela autora.

Em seguida o vetor de palavras foi convertido em correspondentes numéricos. A Figura 16 ilustra uma parte do conjunto gerado, onde a coluna 1 representa o atributo "abordagem", a coluna 2 represente o atributo "acessibilidade", etc, até a coluna 367 que representa a totalidade de palavras que podem ocorrer nos vetores de palavras, enquanto cada linha representa um dos 1037 registros da base de dados resultante.

Figura 16 – Matriz da frequência que as palavras aparecem nos registros.

Contagem Vetorizador

	abordagem	acessibilidade	acesso	acessível	acompanhamento	acurácia	\
0	0	0	0	0	1	0	
1	0	0	0	0	1	0	
2	0	0	0	0	1	0	
3	0	0	0	0	1	0	
4	0	0	0	0	1	0	
...
1032	0	0	0	0	0	0	
1033	0	0	0	0	0	0	
1034	0	0	0	0	0	0	
1035	0	0	0	0	0	0	
1036	0	0	0	0	0	0	

	adaptabilidade	adaptação	adequação	ajuda	...	usabilidade	uso	\
0	0	0	0	0	...	0	0	
1	0	0	0	0	...	0	0	
2	0	0	0	0	...	0	0	
3	0	0	0	0	...	0	0	
4	0	0	0	0	...	0	0	
...	
1032	0	0	0	0	...	0	0	
1033	0	0	0	0	...	0	0	
1034	0	0	0	0	...	0	0	
1035	0	0	0	0	...	0	0	
1036	0	0	0	0	...	0	0	

[1037 rows x 367 columns]

fonte: criada pela autora.

Ainda na Figura 16, em vermelho foram marcados os números que representam a

frequência com que a palavra acompanhamento aparecem nas nas cinco primeiras linhas da matriz.

Logo após transformar o vetor de palavras em uma matriz numérica com a representação da frequência que as palavras apareciam no vetor, o algoritmo *TF – IDF* foi aplicado a base de dados, gerando um conjunto de dados numéricos que representa o documento. A Figura 17 ilustra os pesos atribuídos a palavra "acompanhamento" considerando a frequência com que ela ocorre no documento e no conjunto de documentos.

Figura 17 – Matriz dos valores dos pesos de cada palavra.

	abordagem	acessibilidade	acesso	acessível	acompanhamento	acurácia	\
0	0.0	0.0	0.0	0.0	0.694426	0.0	
1	0.0	0.0	0.0	0.0	0.722494	0.0	
2	0.0	0.0	0.0	0.0	0.708030	0.0	
3	0.0	0.0	0.0	0.0	0.686324	0.0	
4	0.0	0.0	0.0	0.0	0.758347	0.0	
...	
1032	0.0	0.0	0.0	0.0	0.000000	0.0	
1033	0.0	0.0	0.0	0.0	0.000000	0.0	
1034	0.0	0.0	0.0	0.0	0.000000	0.0	
1035	0.0	0.0	0.0	0.0	0.000000	0.0	
1036	0.0	0.0	0.0	0.0	0.000000	0.0	

	adaptabilidade	adaptação	adequação	ajuda	...	usabilidade	uso	\
0	0.0	0.0	0.0	0.0	...	0.0	0.0	
1	0.0	0.0	0.0	0.0	...	0.0	0.0	
2	0.0	0.0	0.0	0.0	...	0.0	0.0	
3	0.0	0.0	0.0	0.0	...	0.0	0.0	
4	0.0	0.0	0.0	0.0	...	0.0	0.0	
...	
1032	0.0	0.0	0.0	0.0	...	0.0	0.0	
1033	0.0	0.0	0.0	0.0	...	0.0	0.0	
1034	0.0	0.0	0.0	0.0	...	0.0	0.0	
1035	0.0	0.0	0.0	0.0	...	0.0	0.0	
1036	0.0	0.0	0.0	0.0	...	0.0	0.0	

[1037 rows x 367 columns]

fonte: criada pela autora.

É possível observar que mesmo a palavra aparecendo somente 1 (uma) vez no documento (uma linha da matriz) o valor do peso atribuído é diferente. Isto ocorre porque a relevância da palavra considera o documento em que ela está e o conjunto de documentos do qual ela faz parte também.

Posteriormente, foi definida a variável de saída e os dados divididos em grupos de teste e treino, sendo definidos 70% de dados para treino e 30% para teste. A Figura 18 contém o código define a variável de saída e divide a base de dados em dados de treino e teste.

Após a preparação da base de dados a partir da aplicação da PLN e das etapas de pré-processamento. A etapa seguinte consistiu nos testes e comparações dos algoritmos de IA. Os testes foram realizados utilizando 3 (três) algoritmos: *K-Nearest Neighbor*, *Random Forest* e Redes Neurais Artificiais. Para isso, foram aplicadas algumas métricas de desempenho, sintetizadas em gráficos e tabelas para melhor visualização dos resultados. O Apêndice B apresenta os *scripts* utilizados para o KNN, RNA e *Random Forest*.

Figura 18 – Código definição da variável de saída e divisão da base de dados.

```
16 # Definindo a variavel de saida
17 y = dataset['Metodo que avalia']
18 # Codifique as classes de saida
19 label_encoder = LabelEncoder()
20 y = label_encoder.fit_transform(y)
21
22 # Dividindo os dados em conjuntos de treinamento e teste
23 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
    random_state=42, stratify=y)
```

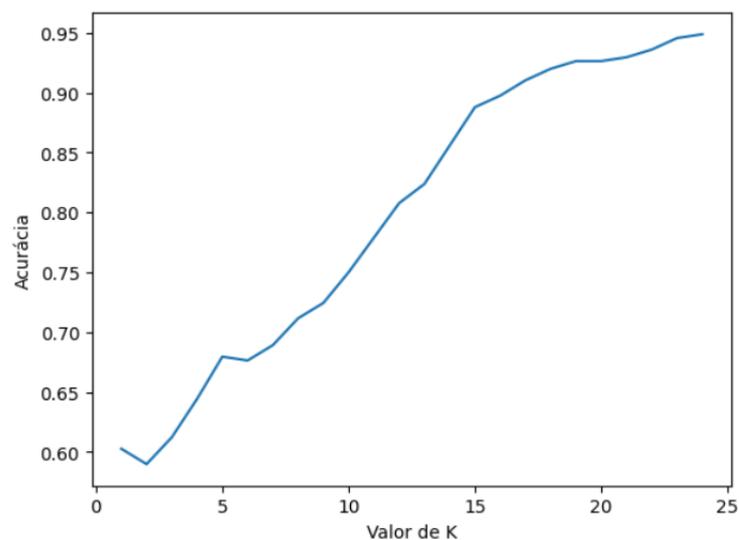
fonte: criada pela autora.

7.1 Desempenho dos Algoritmos

7.1.1 Algoritmo *k-Nearest Neighbors*

Na aplicação do algoritmo *k-Nearest Neighbors* para classificar os dados é utilizada a função *KNeighborsClassifier* da biblioteca *SKlearn* em *Python* e seguidos os seguintes passos:

I. Em primeiro lugar, o valor de K e a métrica de distância são selecionados. No *SKlearn* *KNeighborsClassifier*, a métrica de distância é especificada usando o parâmetro *metric*. O valor padrão de *metric* é Minkowski. Neste trabalho foi usada a distância Euclidiana e $k = 24$, encontrado a partir da relação do valor de k com a acurácia, encontrada com a simulação com o valor de k variando de 1 a 25. Esta relação pode ser visualizada na Figura 19.

Figura 19 – Relação da Acurácia X Valor de k .

fonte: criada pela autora.

Uma vez que K e a métrica de distância são selecionados, o algoritmo KNN passa pelas seguintes etapas:

Conforme observado na Tabela 11, o valor alcançado de MAE foi de 0,257, o que representa que a média do erro absoluto entre o valor predito e o valor real teve diferença de aproximadamente 25,7%. No valor de ACC, que representa quanto o modelo acertou das previsões possíveis, o modelo alcançou 0,95, o que significa que das 1037 possibilidades de classificação, 95% foram classificadas de forma correta, isto é como verdadeiros positivos e verdadeiros negativos. Já a PRE, que significa que dentre todas as classificações de classe positivo que o modelo fez, quantas estão corretas, o algoritmo KNN alcançou 0,94. A SEN alcançou 0,95, o que significa à taxa de acerto na classe positiva verdadeira. O valor atingido pelo F1 foi de 0,94, que significa a média harmônica entre PRE e SEN.

7.1.2 Algoritmo Redes Neurais Artificiais

Para implementação foi utilizado o modelo *Sequencial* a biblioteca *Keras* do *Python*. Tal modelo permite inserir camadas em série, onde a saída da primeira camada serve como entrada da segunda, e assim sucessivamente.

Foi adicionada a camada de entrada da rede do tipo *Dense*. Esta camada tem como objetivo calcular uma função de ativação em conjunto com dados de entrada e pesos.

A camada de entrada recebeu o conjunto de dados *X_train*. A camada terá 3 neurônios *hidden*. O número de neurônios esta relacionado com a forma que a rede neural interpreta os dados. Também foi definido que a função de ativação que seria utilizada era a função "*relu*", *Rectified Linear Unit*. O papel da função de ativação é decidir se um neurônio da camada é acionado ou não.

Foi adicionada uma segunda camada com 100 neurônios e também com a função de ativação "*relu*" também.

Para finalizar foi criada a camada de saída com 24 neurônios e a função de ativação "*softmax*", que retorna a probabilidade de cada classe de saída.

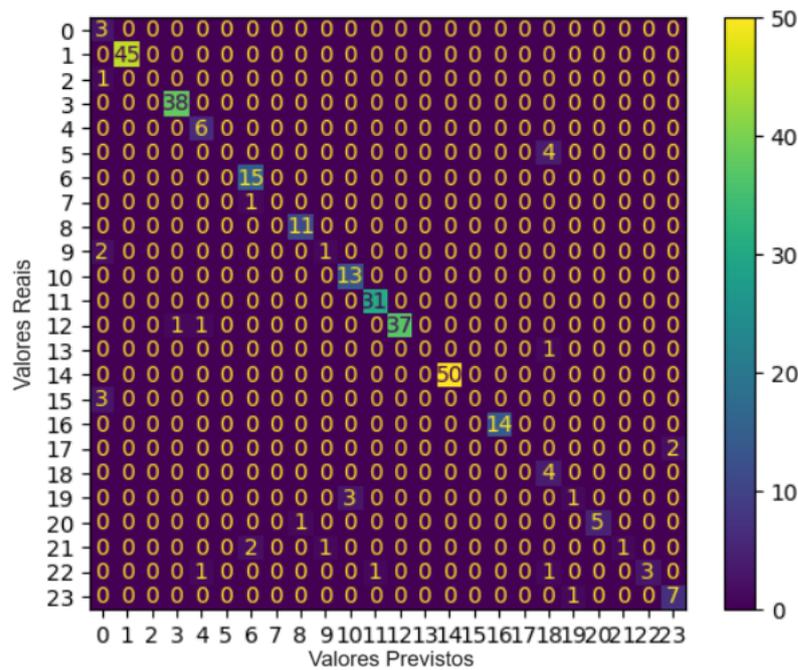
A Figura 21 representa a matriz de confusão da classificação e a Tabela 12 as medidas de desempenho alcançadas.

Como pode ser observado na Figura 21 na matriz de confusão do algoritmo RNA também existem elementos fora da diagonal da matriz, como o elemento $M(C_5, C_{18})$, o que significa que o elemento foi classificado como falso positivo ou falso negativo, desta forma, houve erro na classificação.

Tabela 12 – Medidas de desempenho do algoritmo Redes Neurais Artificiais

MAE	ACC	PRE	SEN	F1
0.173	0.97	0.97	0.98	0.98

Figura 21 – Matriz de Confusão RNA



fonte: criada pela autora.

A Tabela 12 mostra que o valor do MAE foi de 0,173, desta forma a média do erro absoluto entre o valor predito e o valor real teve diferença de aproximadamente 17,3%. O valor de ACC alcançou 0,97, o que representa que o modelo acertou 97% das previsões de verdadeiros positivos e verdadeiros negativos. Para a métrica PRE, o modelo alcançou 0,97, que significa que dentre todas as classificações de classe positivo que o modelo fez, 97% estão corretas. Já a SEN alcançou 0,98, o que significa 98% à taxa de acerto na classe positiva verdadeira. Na média harmônica entre PRE e SEN, o valor atingido por F1 foi de 0,98.

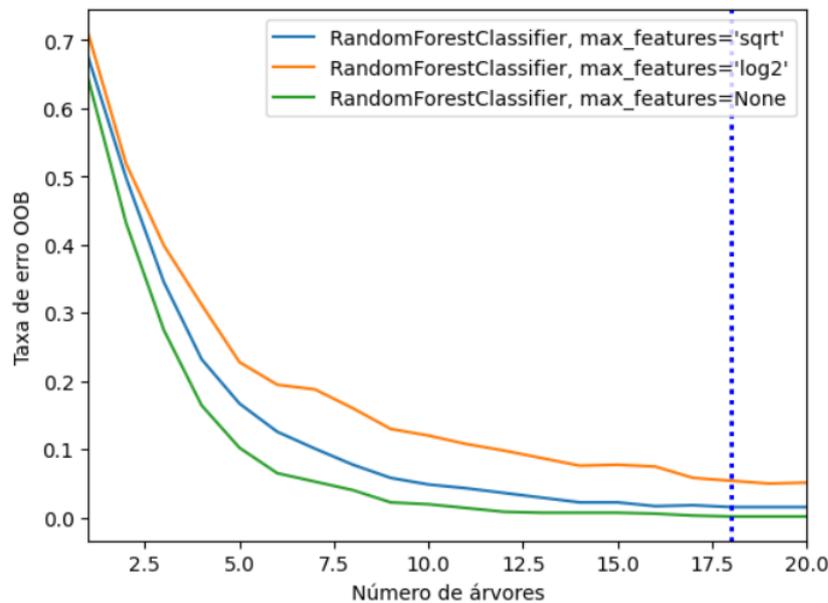
7.1.3 Algoritmo *Random Forest*

Na utilização do algoritmo *Random Forests*, o número de árvores, um dos hiperparâmetros do modelo, foi definido a partir do erro OOB (*out-of-bag*). Sendo escolhido o número de árvores 18, visto que erro OOB associado é apenas ligeiramente superior ao produzido por florestas com mais árvores. A Figura 22 exibe o erro OOB em relação ao número de árvores utilizando-se 3 (três) possibilidades para o parâmetro *max_features*, sendo uma linha traçada no número de árvores escolhido.

Alguns dos hiperparâmetros que podem ser ajustados para classificadores de *Random Forest* na biblioteca *tensorflow.keras* do *Python*:

critério: A função de medir a qualidade de uma divisão. Os critérios suportados são "gini" para a impureza de Gini e "entropia" para o ganho de informação. A impureza de Gini é definida como a soma das probabilidades quadráticas de cada classe, enquanto o ganho

Figura 22 – Taxa de erro OOB de acordo com o número de árvores na *Random Forest*



fonte: criada pela autora.

de informação é definido como a diminuição da entropia. No caso da floresta aleatória, uma diminuição na entropia pode ser entendida como o aumento da pureza do nó. Em outras palavras, a floresta aleatória tenta maximizar o ganho de informações em cada nó.

max_depth: A profundidade máxima da árvore. Se Nenhum, então os nós são expandidos até que todas as folhas estejam puras ou até que todas as folhas contenham menos de `min_samples_split` amostras.

min_samples_split: O número mínimo de amostras necessárias para dividir um nó interno.

random_state: A semente aleatória usada para gerar os subconjuntos aleatórios de recursos e dados.

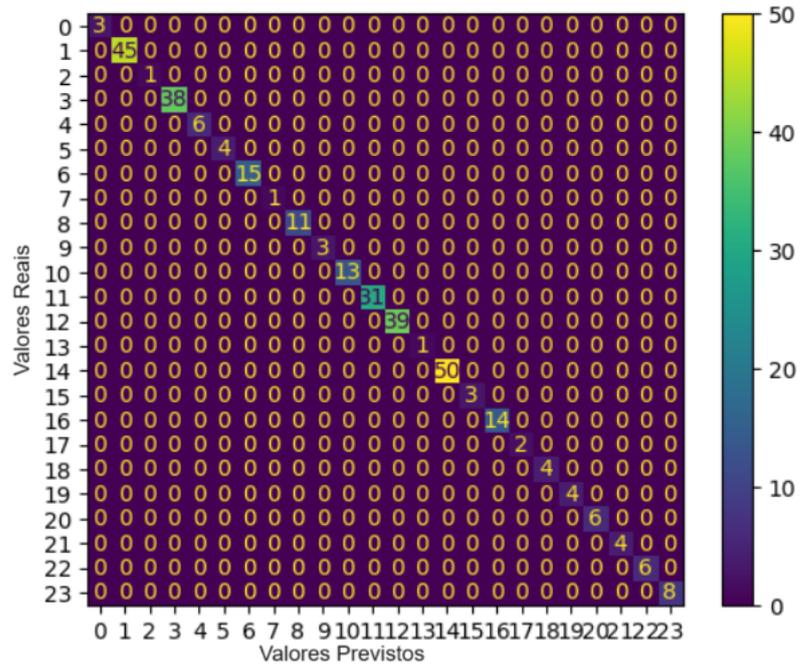
n_estimators: O número de árvores de decisão na floresta aleatória.

max_features: O número máximo de recursos a serem considerados ao procurar a melhor divisão.

A Figura 23 ilustra a matriz de confusão da classificação com *Random Forest* e a Tabela 13 as medidas de desempenho do algoritmo.

Como pode ser observado na matriz de confusão, todos os elementos $M(C_i, C_j)$, para $i \neq j$, que estão fora da diagonal da matriz, possuem valores iguais a 0, o que significa que a classificação com o algoritmo *Random Forest* conseguiu acertar todas as previsões, obtendo o valor 0 de MAE, conforme mostrado na Tabela 13.

Figura 23 – Matriz de Confusão *Random Forest*



fonte: criada pela autora.

Tabela 13 – Medidas de desempenho do algoritmo *Random Forest*

MAE	ACC	PRE	SEN	F1
0	1	1	1	1

Também como reflexo da taxa de acerto de 100% que o algoritmo *Random Forest* alcançou, a Tabela 13 mostra que a ACC do modelo alcançou valor 1, assim como os valores de SE e F1.

Desta forma, os resultados da classificação utilizando o algoritmo *Random Forest* alcançaram resultados ideais para a classificação da melhor metodologia de avaliação de SE.

8 CONCLUSÃO

Ao analisar a literatura, através de um MSL, foi constatado que existem diversas técnicas e metodologias para avaliação de SE, no entanto, no melhor do nosso conhecimento ainda é inexistente a proposição de um tipo de padronização que leve em consideração a critérios relativos a qualidade, escalas de mensuração e procedimentos de verificação do SE. Essa heterogeneidade dificulta muito a avaliação de um SE, e pode produzir resultados inconclusivos.

Por esta razão foi proposto um modelo de qualidade (Castilho-Farias) que considerou 24 metodologias de avaliação de SE, atentando as características Pedagógicas, Técnicas, Socioculturais e Socioeconômicas disponíveis em cada uma das metodologias. Desta forma, no modelo de qualidade foram agrupados 8 (oito) critérios Socioculturais, 8 (oito) Socioeconômicos, 85 (oitenta e cinco) critérios Pedagógicos, 136 (cento e trinta e seis) critérios Técnicos. Após agrupados as características e critérios, foram relacionados ao perfil do usuário que avaliava o método nos estudos, que poderiam ser o Aluno, o Professor, o Instrutor e o Especialista.

Após a criação da base de dados com os dados das 24 (vinte e quatro) metodologias de avaliação de SE, foram utilizados algoritmos híbridos fracamente acoplados, a fim de eliminar a subjetividade na escolha da metodologia de avaliação. Para a tarefa de pré processamento dos dados, além da limpeza de registros nulos e brancos e eliminação e registros repetidos, foram aplicadas técnicas de processamento de linguagem natural, dentre elas a TF-IDF. Esta técnica calcula um peso para cada palavra em cada documento, com base na frequência da palavra no documento e em todo o *corpus*. Isso ajuda a destacar palavras-chave que são mais discriminantes.

Após preparada a base de dados, foi definido como os dados seriam distribuídos para aplicação nos algoritmos de seleção que seriam escolhidos para serem utilizados. Nesta etapa, foi definido que 70% de dados da base de dados seriam usados para o treinamento dos algoritmos e os 30% restantes para a realização dos testes.

Em seguida foram aplicados os algoritmos de IA para classificação *k-Nearest Neighbors*, Redes Neurais Artificiais e *Random Forest*. A utilização dos algoritmos buscou automatizar e eliminar a subjetividade no processo de selecionar a metodologia mais adequada para avaliar um SE dentre as 24 (vinte e quatro) consideradas no estudo. Esta seleção busca oferecer ao professor a metodologia mais adequada para que o mesmo possa avaliar os SE que pretende usar e encontrar o que melhor se adequa aos seus objetivos. Dentre os itens que podem ser considerados para esta seleção estão o perfil do usuário avaliador e a característica e os critérios julgados como importantes de se considerar na avaliação do SE, estes itens são definidos pelo usuário no início da utilização da solução proposta.

Finalizando o processo, foram feitas análises nas métricas derivadas da Matriz de Confusão, as métricas utilizadas foram o Erro Médio Absoluto, a Acurácia, a Sensibilidade ou *Recall*, Taxa de Falso Negativo, Taxa de Falsos Positivos, Precisão e *FI-Score*. Considerando os valores

das métricas utilizadas para avaliar o desempenho dos algoritmos de classificação é possível observar que o método *Random Forest* produziu o melhor resultado, apresentando um resultado ideal, com 100% de acertos, seguido do algoritmo Redes Neurais e com o pior resultado ficou o algoritmo *k-Nearest Neighbors*.

8.1 Dificuldades Encontradas

Na realização deste trabalho foram encontradas dificuldades na seleção das metodologias, uma vez que foi encontrada um padrão de avaliação, de métricas e objetivos. Foram encontradas metodologias que avaliavam com perspectivas iguais, como por exemplo a avaliação de características pedagógicas, no entanto as metodologias consideravam requisitos diferentes e avaliação com métricas diferentes. Também houve dificuldades na escolha da técnica de transformação dos dados textuais para numéricos. Foram tentadas técnicas como a *Label encoder* que ofereciam resultados muito ruins. Desta forma, foi importante o estudo das técnicas de transformação de dados textuais para numéricos para definir o que alcançava bons resultados.

REFERÊNCIAS

- ABDURRAHMAN, J.; BEER, M.; CROWTHER, P. Pedagogical requirements for mobile learning: A review on mobile learn task model. **Journal of Interactive Media in Education**, Ubiquity Press, v. 2015, n. 1, p. 1–17, 2015.
- ABES. **Mercado brasileiro de software: Panorama e tendências**. [S.l.]: ABES-Associação Brasileira das Empresas de Software São Paulo, SP, 2022.
- ABREU, F. et al. Métodos, técnicas e ferramentas para o desenvolvimento de software educacional: um mapeamento sistemático. **Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012)**, ISSN, p. 2316–6533, 2012.
- ALENCAR, A. I. D.; MARQUES, A. B. Criação de um guia para selecionar métodos de avaliação para tecnologias de software m-learning. **Revista Brasileira de Informática na Educação**, v. 29, p. 73–96, 02 2021.
- ANDRADE, M.; JR, C.; SILVEIRA, I. Estabelecimento de critérios de qualidade para aplicativos educacionais no contexto dos dispositivos móveis (m-learning). **EaD em FOCO**, v. 7, 09 2017.
- ANDRADE, P. Ferreira de. Projeto educom: Realizações e produtos. **Brasília: Ministério da Educação e Organização dos Estados Americanos**, 1993.
- ANDRES, D. P.; CYBIS, W. d. A. Um estudo teórico sobre as técnicas de avaliação de software educacional. In: **VI Congreso Argentino de Ciencias de la Computación**. [S.l.: s.n.], 2000.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207 - Tecnologia da Informação - Processos de ciclo de vida de software**. Rio de Janeiro, 1998. 35 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 8402 - Gestão da qualidade e garantia da qualidade - Terminologia**. Rio de Janeiro, 2000. 14 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9000 - Sistemas de gestão da qualidade - Fundamentos e vocabulário**. Rio de Janeiro, 2015. 59 p.
- ATAYDE, A. P. R.; TEIXEIRA, A. B. M.; PÁDUA, C. Maqsei-uma metodologia de avaliação de qualidade de software educacional infantil. In: **Brazilian Symposium on Computers in Education-Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2003. v. 1, n. 1, p. 356–365.
- ATENSTAEDT, R. Word cloud analysis of the bjgp. **British Journal of General Practice**, Royal College of General Practitioners, v. 62, n. 596, p. 148–148, 2012. ISSN 0960-1643. Disponível em: <<https://bjgp.org/content/62/596/148>>.
- BASTIEN, C.; SCAPIN, D. **Ergonomic criteria for the evaluation of human-computer interfaces**. Tese (Doutorado) — Inria, 1993.
- BEDNARIK, R. et al. Development of the tup model-evaluating educational software. In: **IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings**. [S.l.], 2004. p. 699–701.

- BERTOLETTI, A. C.; MORAES, M. C.; COSTA, A. C. da R. Avaliação do módulo de aprendizagem do museu virtual sagres quanto a usabilidade de um software educacional. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2001. v. 1, n. 1, p. 19–27.
- BOENTE, A.; OLIVEIRA, F.; ALVES, J. Rup como metodologia de desenvolvimento de software para obtenção da qualidade de software. **Anais: V Simpósio de Excelência em Gestão e Tecnologia**, 2008.
- BOURQUE, P.; FAIRLEY, R. E. **Guide to the Software Engineering Body of Knowledge**. [S.l.]: IEEE Computer Society, 2014.
- BRAGA, M. da M. **Design de software educacional baseado na teoria dos campos conceituais**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2006. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/2616>>. Acesso em: 05 set. 2023.
- BRASIL. Lei nº 9.394, de 20 de dezembro de 1996 - estabelece as diretrizes e bases da educação nacional. **Diário Oficial da União**, Senado Federal Brasília, DF, 1996.
- BRASIL. Constituição da república federativa do brasil de 1988. **Recuperado de <http://www.ritmodeestudos.com.br>**, 2010.
- BRASIL. **Parecer CNE/CP nº 5/2020. Reorganização do Calendário Escolar e da possibilidade de cômputo de atividades não presenciais para fins de cumprimento da carga horária mínima anual, em razão da Pandemia da COVID-19**. [S.l.]: MEC Brasília, DF, 2020.
- CAMARGO, S. M. S.; CHAVES, F. A. B. A contribuição das audiências por videoconferências durante a pandemia para a efetivação princípios gerais do direito processual brasileiro. **REVISTA FOCO**, v. 16, n. 5, p. e1867–e1867, 2023.
- CAMPOS, G. H. B. d. Metodologia para avaliação da qualidade de software educacional. **Diretrizes para desenvolvedores e usuários**, v. 232, 1994.
- CAVALCANTI, P. d. L.; FERREIRA, J. C. Análise descritiva do software educacional gcompris. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2011. v. 1, n. 1.
- COOMANS, S.; LACERDA, G. S. Petese, a pedagogical ergonomic tool for educational software evaluation. **Procedia Manufacturing**, Elsevier, v. 3, p. 5881–5888, 2015.
- CRISÓSTOMO, L. C. da S. et al. Mobile learning: Avaliação e seleção de um aplicativo para o ensino de elementos químicos. **Redin-Revista Educacional Interdisciplinar**, v. 7, n. 1, 2018.
- DALIANIS, H. Evaluation metrics and evaluation. **Clinical Text Mining: secondary use of electronic patient records**, Springer, p. 45–53, 2018.
- DATA.AI. **State of Mobile 2022**. 2022. Disponível em: <<https://www.data.ai/en/go/state-of-mobile-2022/>>. Acesso em: 02 abr. 2022.
- D’CARLO, D.; BARBOSA, G.; OLIVEIRA, E. Usabilidade em aplicativos móveis educacionais: Um conjunto de heurísticas para avaliação. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 777.

DIAS, M. G. G. **Boas práticas em educação básica: uma análise qualitativa das políticas educacionais cearenses em escolas com maior variação no IDEB entre 2007 e 2019.**

Dissertação (Dissertação de Mestrado) — Instituto Brasileiro de Ensino, Desenvolvimento e Pesquisa, 2021. Disponível em: <<https://repositorio.idp.edu.br/handle/123456789/4213>>. Acesso em: 21 ago. 2023.

DUPUIS, R.; BOURQUE, P.; ABRAN, A. Swebok guide an overview of trial usages in the field of education. In: **33rd Annual Frontiers in Education, 2003. FIE 2003.** [S.l.: s.n.], 2003. v. 3, p. S3C–19.

DUTRA, P.; BERVIAN, P. V.; GÜLLICH, R. I. da C. Mobile learning e o uso de apps como proposta para o ensino de ciências. **Revista Polyphonia**, v. 31, n. 2, p. 121–136, 2020.

ECONOMIDES, A. A. Requirements of mobile learning applications. **International Journal of Innovation and Learning**, Inderscience Publishers, v. 5, n. 5, p. 457–479, 2008.

EICHLER, M.; PINO, J. C. D. Computadores em educação química: estrutura atômica e tabela periódica. **Química Nova**, SciELO Brasil, v. 23, p. 835–840, 2000.

FACELI, K. et al. Inteligência artificial—uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, v. 2, p. 45, 2011.

FARIAS, M. M. d. R. **As representações matemáticas mediadas por softwares educativos em uma perspectiva semiótica: uma contribuição para o conhecimento do futuro professor de matemática.** Dissertação (Dissertação de Mestrado) — Universidade Estadual Paulista, 2007. Disponível em: <<https://repositorio.unesp.br/handle/11449/91110>>. Acesso em: 05 set. 2023.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37–37, 1996.

FIALHO, N. N.; MATOS, E. L. M. A arte de envolver o aluno na aprendizagem de ciências utilizando softwares educacionais. **Educar em Revista**, SciELO Brasil, p. 121–136, 2010.

FILHO, N. F. D.; BARBOSA, E. F. A requirements catalog for mobile learning environments. In: **Proceedings of the 28th annual ACM symposium on applied computing.** [S.l.: s.n.], 2013. p. 1266–1271.

FIOLHAIS, C.; TRINDADE, J. Física no computador: o computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 25, p. 259–272, 2003.

FONSECA, M. O. et al. Zika gamification: Mobile application for endemic disease control agents training. In: SBC. **Anais do XVII Workshop de Informática Médica.** [S.l.], 2017.

FRANÇA, R. S. de; SILVA, A. C. B. da. Avaliação de softwares educativos para o ensino de língua portuguesa. **Revista Brasileira de Informática na Educação**, v. 22, n. 03, p. 23, 2014.

FRANCISCO, O. J. **Recuperação de informação em atas de reunião utilizando segmentação textual e extração de tópicos.** Dissertação (Dissertação de Mestrado) — Universidade Federal de São Carlos, 2018. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/10456>>. Acesso em: 05 set. 2023.

- FRESCKI, F. **Avaliação da qualidade de softwares educacionais para o ensino de Álgebra**. Tese (Doutorado) — Dissertação (Licenciatura em Matemática)-Universidade Estadual do Oeste do ..., 2008.
- GAMEZ, L. **TICESE: Técnica de Inspeção de conformidade ergonômica de software educacional. Ergonomia escolar e as novas tecnologias no ensino: enfoque na avaliação de software educacional**. Tese (Doutorado) — Dissertação—Mestrado em Engenharia Humana. Braga: Universidade do Minho, 1998.
- GLADCHEFF, A. P.; ZUFFI, E. M.; SILVA, D. M. d. Um instrumento para avaliação da qualidade de softwares educacionais de matemática para o ensino fundamental. 2001.
- GOMES, A. P. L. et al. Geometriar: aplicativo educacional com realidade aumentada para auxiliar o ensino de sólidos geométricos. **RENOTE**, v. 17, n. 1, p. 405–414, 2019.
- GUEDES, A. L.; DALMORO, C. G. Recicla: software educativo para o ensino ambiental. **Unoesc & Ciência-ACET**, v. 1, n. 1, p. 77–84, 2010.
- GURGEL, C. R.; AGUIAR, G. E. d.; SILVA, N. d. N. Avaliação como espaço de aprendizagem em softwares educativos. **Ensaio: avaliação e políticas públicas em Educação**, SciELO Brasil, v. 21, p. 371–388, 2013.
- HAYKIN, S. **Redes neurais: princípios e prática**. [S.l.]: Bookman Editora, 2001.
- HOSSEN, M. K. Heart disease prediction using machine learning techniques. **American Journal of Computer Science and Technology**, v. 5, n. 3, p. 146–154, 2022.
- INEP. **Resultados do Índice de Desenvolvimento da Educação Básica (Ideb)**. 2022. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Disponível em: <<https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/ideb/resultados/>>. Acesso em: 10 nov. 2022.
- JAMES, G.; WITTEN, D. **An introduction to statistical learning with applications in R**. [S.l.]: springer publication, 2013.
- JONES, S. et al. The impact of application system messages on the usability of healthcare software applications. In: IEEE. **2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.], 2016. p. 306–311.
- JUCÁ, S. C. S. A relevância dos softwares educativos na educação profissional. **Ciências & Cognição**, v. 8, 2006.
- JUNIOR, O. d. O. B.; AGUIAR, Y. P. C.; MOURA, H. P. de. Taxonomia de critérios para avaliação de software educativo—tacase/taxonomy of criteria for educational software evaluation-tacase. **Brazilian Journal of Development**, v. 6, n. 3, p. 15082–15095, 2020.
- KANG, Y. et al. Natural language processing (nlp) in management research: A literature review. **Journal of Management Analytics**, Taylor & Francis, v. 7, n. 2, p. 139–172, 2020.
- KEARNEY, M. et al. Viewing mobile learning from a pedagogical perspective. **Alt-J-Research In Learning Technology**, Taylor & Francis Open Access, v. 20, n. 1, 2012.
- KUMAR, A. **Neural Network and Multi-layer Perceptron Examples**. 2023. Disponível em: <<https://vitalflux.com/how-do-we-build-deep-neural-network-using-perceptron/>>. Acesso em: 21 mar. 2023.

- LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: PMLR. **International conference on machine learning**. [S.l.], 2014. p. 1188–1196.
- LEITE, B. S. Aplicativos para aprendizagem móvel no ensino de química. **Ciências em Foco**, v. 13, p. e020013–e020013, 2020.
- LIMA, V. A. d. **Cotas na Educação Básica: uma análise da atuação de políticas em uma escola pública federal**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2023. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/50081>>. Acesso em: 21 ago. 2023.
- LOPES, A. C. C. et al. Construção e avaliação de software educacional sobre cateterismo urinário de demora. **Revista da Escola de Enfermagem da USP**, SciELO Brasil, v. 45, p. 215–222, 2011.
- MACHADO, C.; OLIVEIRA, M.; ALMEIDA, J. O software educativo multimídia “os miúdos e a história de portugal”: análise e avaliação da usabilidade. In: **VII Conferência Internacional de TIC na Educação**. [S.l.: s.n.], 2011. p. 1315–1325.
- MACHADO, R. C. et al. **Um software educativo de exercício-e-prática como ferramenta no processo de alfabetização infantil**. Dissertação (Dissertação de Mestrado) — Universidade Federal de Uberlândia, 2007. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14656>>. Acesso em: 05 set. 2023.
- MARKOVIĆ, M. G.; KLIČEK, B.; VUKOVAC, D. P. The effects of multimedia learning materials quality on knowledge acquisition. **Information Systems Development: Transforming Organisations and Society through Information Systems**, 2014.
- MARSLAND, S. **Machine learning: an algorithmic perspective**. [S.l.]: Chapman and Hall/CRC, 2011.
- MATTOS, F. S. de. Plataformização da notícias e consumo de informação: tendências do jornalismo em um novo ambiente informacional. **Anais do 19º Encontro Nacional de Pesquisadores em Jornalismo**, Galoá, 2021.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, p. 115–133, 1943.
- MEDEIROS, M. de O.; SCHIMIGUEL, J. Uma abordagem para avaliação de jogos educativos: ênfase no ensino fundamental. **RENOTE**, v. 10, n. 3, 2012.
- MELO, J. A. P. et al. **Avaliação de objetos de aprendizagem: cruzando caminhos e produzindo novos olhares**. Dissertação (Dissertação de Mestrado) — Universidade Federal de Uberlândia, 2009. Disponível em: <<https://repositorio.ufu.br/handle/123456789/13792>>. Acesso em: 05 set. 2023.
- MISRA, S.; LI, H.; HE, J. Noninvasive fracture characterization based on the classification of sonic wave travel times. **Machine learning for subsurface characterization**, Gulf Professional Publishing Houston, TX, USA, v. 4, p. 243–287, 2020.
- MOCBEL, M. Â. R. et al. Um framework para desenvolvimento de softwares de suporte ao ensino para plataformas móveis. **RENOTE**, v. 18, n. 1, 2020.

MOITA, F.; HENRIQUE, L.; DANIELE, D. Recursos educacionais inovadores: o spore no ensino da evolução dos seres vivos. **Anais temporários do LACLO 2015**, v. 10, n. 1, p. 178, 2015.

MOMANI, A. M. Using multi-attribute decision-making approach to evaluate learning management systems. **International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)**, IGI Global, v. 16, n. 4, p. 117–131, 2021.

MORAES, M. C. Informática educativa no brasil: uma história vivida, algumas lições aprendidas. **Revista brasileira de Informática na educação**, v. 1, n. 1, p. 19–44, 1997.

MOTA, J. Um estudo de caso para a determinação do preço de venda de imóveis urbanos via redes neurais artificiais e métodos estatísticos multivariados. Setor de Ciências Exatas e de Tecnologia, Universidade Federal do Paraná . . . , 2007.

MUCCHIELLI, A. L'enseignement par ordinateur. **Prensas universitaires de France**, 1987.

NASCIMENTO, A. L.; WERNECK, M. Aprendizagem móvel no ensino de matemática básica: Proposta de um instrumento de avaliação de aplicativos. **II Congresso Internacional de Educação Superior a Distância**, 2016.

NATTESTAD, A. et al. 4.1 web-based interactive learning programmes. **European journal of dental education**, Wiley Online Library, v. 6, p. 127–137, 2002.

NAUMANEN MINNAMARI; TUKIAINEN, M. Practices in old age ict education: Three contexts considered: Clubs, courses and home teaching. **Learning and instruction in the digital age**, Springer, p. 273–288, 2010.

NESBIT, J. C.; BELFER, K.; LEACOCK, T. Learning object review instrument (lori). **E-learning research and assessment network**, p. 33–68, 2003.

NIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras: Pesquisa TIC Educação (Edição COVID-19 - Metodologia adaptada)**. Núcleo de Informação e Coordenação do Ponto BR, 2022. Disponível em: <<http://cetic.br/pt/arquivos/educacao/2021/professores/>>. Acesso em: 28 jun. 2023.

NIELSEN, J. **Usability engineering**. [S.l.]: Morgan Kaufmann, 1994.

NOGUEIRA, T. C. A. et al. Software educativos gratuitos para o ensino de matemática. In: **Anais do Congresso Internacional de Informática Educativa-TISE, Porto Alegre**. [S.l.: s.n.], 2013. p. 483–486.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION AND INTERNATIONAL ELECTROTECHNICAL COMMISSION. **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE**. [S.l.], 2011. 34 p.

NUNES, R.; SANTOS, I. A importância da avaliação ergonômico-pedagógica de aplicativos educacionais e os desafios encontrados no campo da aprendizagem móvel. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2018. v. 29, n. 1, p. 41.

OLIVEIRA, C. C. de; COSTA, J. W. da; MOREIRA, M. **Ambientes informatizados de aprendizagem: produção e avaliação de software educativo**. [S.l.]: Papyrus, 2001.

- OLIVEIRA, S. F. et al. Softwares de simulação no ensino de atomística: Experiências computacionais para evidenciar micromundos. **Revista Química nova na escola**, v. 35, n. 3, p. 147–151, 2013.
- OLIVEIRA, W. et al. Avaliação de jogos educativos: Uma abordagem no ensino de matemática. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2015. v. 26, n. 1, p. 657.
- PAULA, A. C. de et al. Softwares educacionais para o ensino de física, química e biologia. **Revista Ciências & Ideias**, 2014.
- PAYPAL-CORP. **1 em cada 4 brasileiros pretende continuar fazendo compras online diariamente após a pandemia**. 2021. Disponível em: <<https://newsroom.br.paypal-corp.com/consumo-online-no-brasil>>. Acesso em: 28 jun. 2023.
- PAYPAL-CORP. **Pagamentos online por serviços de mobilidade urbana já fazem parte da vida de 8 em cada 10 brasileiros e brasileiras**. 2022. Disponível em: <<https://newsroom.br.paypal-corp.com/everyday-spending-mobilidade-urbana>>. Acesso em: 28 jun. 2023.
- PEREIRA, W. S. et al. Avaliação de software educativo: análise de abordagens para definição de diretrizes. **Nuevas ideas en informática educativa**, v. 12, p. 557–62, 2016.
- PERSICO, D.; MANCA, S.; POZZI, F. Adapting the technology acceptance model to evaluate the innovative potential of e-learning systems. **Computers in Human Behavior**, p. 614–622, 2014.
- PETRI, G. Avaliação de softwares educacionais com ênfase em jogos: um panorama da literatura. **Revista Tecnologias Educacionais em Rede (ReTER)**, p. e2–01, 2020.
- PETRI, G. et al. Games for teaching software project management: an analysis of the benefits of digital and non-digital games. **J. Univers. Comput. Sci.**, v. 24, n. 10, p. 1424–1451, 2018.
- PIVATTO, W. A utilização do jogo educativo geonandia 3d no ensino de matemática: uma investigação com estudantes de séries iniciais do ensino fundamental sobre o tema geometria. **Revista Profissão Docente**, v. 14, n. 30, p. 109–113, 2014.
- PORTO, J. V.; BARBOSA, H.; WANGENHEIM, C. G. von. Proposta de um checklist de avaliação de usabilidade de aplicativos android no contexto educacional. **Anais do Computer on the Beach**, p. 870–879, 2018.
- PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. [S.l.]: Porto Alegre:[sn], 2011.
- RAABE, A. L. et al. Oficinas de utilização de software educacional: Um relato de experiência. In: **Anais do Workshop de Informática na Escola**. [S.l.: s.n.], 2007. v. 1, n. 1, p. 264–271.
- RAJARAMAN, A.; ULLMAN, J. D. **Mining of massive datasets**. [S.l.]: Cambridge University Press, 2011.
- REEVES, T.; HARMON, S. Systematic evaluation procedures for interactive multimedia for education and training", multimedia computing-preparing for the 21st century. **Idea Group**, v. 33, 1994.

- REIS, D. A. dos; JÚNIOR, N. V. Games como estratégia de ensino de ciências para abordar o saneamento básico. **Research, Society and Development**, v. 8, n. 12, p. e428121846–e428121846, 2019.
- RIBEIRO, D.; WELLS, A. **Com pandemia, demanda por videoconferências dispara em empresas brasileiras**. 2021. CNN BRASIL. Disponível em: <<https://www.cnnbrasil.com.br/economia/com-pandemia-demanda-por-videoconferencias-dispara-em-empresas-brasileiras/>>. Acesso em: 28 jun. 2023.
- RIBEIRO, I. G.; CABRAL, M. d. F. N. O hipertexto didático virtual e o software educacional hot potatoes: ferramentas pedagógicas no ensino a distância de física. In: **Congresso Brasileiro de Ensino Superior a Distância**. [S.l.: s.n.], 2014. v. 11.
- RIPASY, R. et al. Assistant meega+: Uma ferramenta de apoio para avaliação de jogos educacionais usando modelo meega+. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2018. v. 29, n. 1, p. 615.
- ROCHA, A.; CAMPOS, G. Avaliação da qualidade de software educacional. **Em aberto**, v. 12, n. 57, 1993.
- RODRIGUES, G. Instrumento para avaliação de jogos eletrônicos educativos do ensino fundamental. **Universidade Federal da Paraíba, João Pessoa, PB, Brasil**, 2014.
- ROQUE, G. O. B. et al. Conteúdos digitais multimídias: construindo novas práticas docentes. **Rio de Janeiro, Artigo**, 2009.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- SANTOS, C. et al. Use of shannon information to relate function and structure in the brain using diffusion spectrum imaging mri. In: IEEE. **2012 IEEE 2nd Portuguese Meeting in Bioengineering (ENBENG)**. [S.l.], 2012. p. 1–4.
- SAVI, R. et al. Proposta de um modelo de avaliação de jogos educacionais. **Renote**, v. 8, n. 3, 2010.
- SHNEIDERMAN, B.; PLAISANT, C. Designing the user interface: Strategies for effective human-computer interaction. hardcover, published march 1st. 2004.
- SILIUS, K.; KAILANTO, M.; TERVAKARI, A.-M. Evaluating the quality of social media in an educational context. In: IEEE. **2011 IEEE global engineering education conference (EDUCON)**. [S.l.], 2011. p. 505–510.
- SILVA, A. et al. Proposta de um modelo preditivo para avaliação da acessibilidade em softwares educativos: um estudo de caso sobre o “menino curioso”. **Nuevas Ideas en Informática Educativa**, 2013.
- SILVA, A. et al. Checklists específicas para softwares educacionais: uma proposta e um mapeamento sistemático. **Santiago de Chile**, v. 14, p. 412–417, 2018.
- SILVA, A. C. B. d. **Softwares educativos: critérios de avaliação a partir dos discursos da interface, da esfera comunicativa e do objeto de ensino**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2012. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/12987>>. Acesso em: 21 ago. 2023.

- SILVA, A. C. B. d.; GOMES, A. S. Conheça e utilize software educativo: avaliação e planejamento para a educação básica. **Recife: Pipa Comunicação**, v. 83, p. 3322–3222, 2015.
- SILVA, G.; NETTO, J. F.; SOUZA, R. A abordagem didática da simulação virtual no ensino da química: Um olhar para os novos paradigmas da educação. In: SBC. **Anais do XXII Workshop de Informática na Escola**. [S.l.], 2016. p. 339–348.
- SILVA, M. G. da; BATISTA, S. C. F. Metodologia de avaliação: análise da qualidade de aplicativos educacionais para matemática do ensino médio. **RENOTE**, v. 13, n. 1, 2015.
- SILVA, M. G. da; BATISTA, S. C. F. Metodologia de avaliação: análise da qualidade de aplicativos educacionais para matemática do ensino médio. **RENOTE**, v. 13, n. 1, 2015.
- SINGH, A. K.; SHASHI, M. Vectorization of text documents for identifying unifiable news articles. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 10, n. 7, 2019.
- SIQUEIRA, E. S. et al. Um jogo com reconhecedor de voz para o ensino de crianças com dificuldade de aprendizagem em leitura e escrita. In: **XXXI Congresso da Sociedade Brasileira de Computação, Anais CSBC**. [S.l.: s.n.], 2011. v. 1, p. 1279–1292.
- SIVAKUMAR, A.; GUNASUNDARI, R. A survey on data preprocessing techniques for bioinformatics and web usage mining. **International Journal of Pure and Applied Mathematics**, v. 117, n. 20, p. 785–794, 2017.
- SOAD, G. W.; BARBOSA, E. F. Moleva: um método de avaliação de qualidade para aplicativos educacionais móveis. In: SBC. **Anais do XVI Simpósio Brasileiro de Qualidade de Software**. [S.l.], 2017. p. 74–88.
- SOAD, G. W.; FILHO, N. F. D.; BARBOSA, E. F. Uma contribuição ao estabelecimento de características de qualidade para aplicações educacionais móveis. In: SBC. **Anais do XIV Simpósio Brasileiro de Qualidade de Software**. [S.l.], 2015. p. 165–179.
- SOAD, G. W.; FILHO, N. F. D.; BARBOSA, E. F. Quality evaluation of mobile learning applications. In: IEEE. **2016 IEEE Frontiers in Education Conference (FIE)**. [S.l.], 2016. p. 1–8.
- SOUZA, A. B. d. Avaliando a usabilidade do software educacional gcompris: estudo de caso com os educandos do ensino fundamental. 2014.
- STATISTA. **Leading learning and education apps worldwide in 2022**. 2023. Disponível em: <<https://www.statista.com/statistics/1284623/top-education-apps-global-by-downloads/>>. Acesso em: 28 jun. 2023.
- TELLES, H.; ALVES, L. Ensino de história e videogame: problematizando a avaliação de jogos baseados em representações do passado. **Anais do Seminário de Jogos Eletrônicos, Educação e Comunicação**, 2015.
- TENÓRIO, A.; SOUZA, S. M. R. de; TENÓRIO, T. O uso do software educativo geogebra no estudo de geometria analítica. **Revista do Instituto GeoGebra Internacional de São Paulo**, v. 4, n. 2, p. 103–121, 2015.

- TERGAN, S.-O. Checklists for the evaluation of educational software: Critical review and prospects. **Innovations in education and training international**, Taylor & Francis, v. 35, n. 1, p. 9–20, 1998.
- VALENTE, J. A. O uso inteligente do computador na educação. **Revista Pátio**, v. 1, n. 1, p. 19–21, 1997.
- VALENTE, J. A. et al. O computador na sociedade do conhecimento. **Campinas: Unicamp/NIED**, v. 6, 1999.
- VIEIRA, M. M. S.; SIMÕES, L. L. F.; BARRETO, A. L. de O. Avaliação de software educativo: aspectos pedagógicos e técnicos. **Faculdade Cearense em Revista**, v. 5, n. 1, 2012.
- WANGENHEIM, C. G. V. et al. Splashcode-a board game for learning an understanding of algorithms in middle school. **Informatics in Education**, Institute of Mathematics and Informatics, v. 18, n. 2, p. 259, 2019.
- WEBBER, C.; BOFF, E.; BONO, F. Ferramenta especialista para avaliação de software educacional. **XX Simpósio Brasileiro de Informática na Educação**, 2009.
- YACOUBY, R.; AXMAN, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In: **Proceedings of the first workshop on evaluation and comparison of NLP systems**. [S.l.: s.n.], 2020. p. 79–91.
- ZAIKON, S. B.; SHIRATUDDIN, N. Heuristics evaluation strategy for mobile game-based learning. In: IEEE. **2010 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education**. [S.l.], 2010. p. 127–131.
- ZORZAL, E. R. et al. Aplicação de jogos educacionais com realidade aumentada. **RENOTE**, v. 6, n. 2, 2008.

APÊNDICE A – LISTA COM A DEFINIÇÃO DOS CRITÉRIOS

Critérios relacionados a Característica Pedagógica	
Critério	Definição
Acompanhamento (Alencar e Marques, 2021)	Uso de um conjunto de informações do cotidiano de ensino e aprendizado e informações de avaliações sumativas, para prover uma visão geral do progresso do aprendiz.
Adaptabilidade (Alencar e Marques, 2021)	Capacidade do aplicativo em se adaptar a diferentes tipos de dispositivos móveis.
Adaptação ao contexto (Alencar e Marques, 2021)	A aplicação deve ser capaz de se adaptar ao contexto do usuário, por exemplo, adaptar-se a um usuário do meio rural, com idade adulta e com escolaridade baixa.
Adequação funcional (Alencar e Marques, 2021)	Capacidade de disponibilizar funções que atendam às necessidades implícitas e explícitas em relação ao aplicativo.
Ambiente de aprendizagem (Alencar e Marques, 2021)	Capacidade de prover um ambiente que favoreça aprendizagem ao aluno, provendo maneiras para manter o interesse do aluno.
Aplicabilidade (Alencar e Marques, 2021)	Deve prover um sistema onde se possa empregar, colocar em prática ou usar para o objetivo definido a fim de promover ganho de conhecimento para o usuário.
Aplicação (Alencar e Marques, 2021)	Refere-se aos aspectos técnicos do aplicativo relacionados às questões pedagógicas.
Aprendizado Cooperativo (Alencar e Marques, 2021)	Possibilidade de realizar tarefas e atividades em grupo.
Aprendizagem (Alencar e Marques, 2021)	Capacidade do aplicativo em disponibilizar funcionalidades que contribuam para o aprendizado do aluno.
Aprendizagem de curto prazo (Alencar e Marques, 2021)	Oferece ambiente com capacidade do aluno armazenar e manipular uma quantidade limitada de informações por um período breve de tempo.
Aprendizagem e Recorção no processo de Interação (Alencar e Marques, 2021)	Relacionam-se com a condução as heurísticas de reconhecimento no lugar de memorização, qualidade das opções de ajuda, legibilidade, feedback e agrupamento e distinção de itens. Quando o usuário é orientado passo a passo nos procedimentos a serem seguidos.
Atividade do usuário (Junior et al. 2020)	Interação com o usuário que estimule a participação e auxilie na superação de conflitos cognitivos.

Atividades de instrução (Alencar e Marques, 2021)	Deve fornecer instruções para guiar o usuário durante a aprendizagem.
Atividades e tarefas (Alencar e Marques, 2021)	A aplicação deve oferecer diferentes tipos de atividades relacionadas à educação, que devem ser adequadas ao nível, aos interesses e às necessidades dos alunos, bem como aos objetivos e às metodologias do processo de ensino-aprendizagem.
Atividades educacionais (Alencar e Marques, 2021)	A aplicação deve oferecer diferentes tipos de atividades relacionadas à educação, provendo assim a aquisição de conhecimento e a possibilidade do aluno receber orientação de um tutor nas atividades.
Atividades Reflexivas (Alencar e Marques, 2021)	Oportunidade de reflexão e interpretação que a atividade oferece ao usuário.
Atração motivação (Junior el al., 2020)	Atratividade para despertar o interesse de uso.
Avaliação (Alencar e Marques, 2021)	Permita que o(a) docente ou os pais façam coleta, análise e interpretação de informações sobre a aprendizagem.
Complexidade de conteúdo (Alencar e Marques, 2021)	Os conteúdos não devem possuir excesso de informações nas lições, devendo ser oferecido gradativamente, evitando assim que um conteúdo complexo possa afetar a compreensão do usuário.
Complexidade multimídia (Alencar e Marques, 2021)	A aplicação não deve possuir muitos elementos multimídias, pois estes podem ocasionar em pontos de distração para o usuário.
Conhecimento na hora certa (Alencar e Marques, 2021)	O conteúdo deve ser oferecido na hora correta, evitando oferecer um conteúdo para ensinar algo que dependa de um conteúdo ainda não estudado pelo usuário.
Consistência pedagógica (Junior el al., 2020)	Apresenta objetivos específicos ou proposta educacional.
Consolidação de novos aprendizados (Alencar e Marques, 2021)	É importante que o <i>software</i> permita que o(a) docente ou os pais verifiquem se e quais os conceitos estão sendo aprendidos pelos alunos, ou seja, a efetividade da aprendizagem.
Conteúdo (Alencar e Marques, 2021)	Capacidade em fornecer conteúdo gerenciável e de qualidade.
Controle de aprendizado e Flexibilidade (Alencar e Marques, 2021)	Refere-se ao controle, gerenciamento e monitoramento dos conteúdos oferecidos. Avalia o grau em que o produto de <i>software</i> pode ser utilizado com eficácia, eficiência, sem riscos e satisfação em contextos diferentes dos especificados.

Controle do Usuário (Alencar e Marques, 2021)	O usuário deve possuir controle sobre seu aplicativo e decidir se deseja receber sua ajuda ou não. Além disso, o aplicativo deve prover ferramentas para concepção de conteúdo, criação e organização, acompanhamento do aluno, avaliação, entre outros.
Controle e autonomia do usuário (Alencar e Marques, 2021)	Avalia as possibilidades de o usuário controlar o encadeamento e a realização das ações.
Controle e personalização (Alencar e Marques, 2021)	O usuário controla o sistema. Ele pode, a qualquer momento, abortar uma tarefa, ou desfazer uma operação e retornar ao estado anterior.
Correção das simplificações do conteúdo (Alencar e Marques, 2021)	Revisão e aprimoramento de um texto que visa eliminar ou reduzir as simplificações excessivas ou inadequadas do conteúdo, que podem comprometer a clareza, a precisão, a coerência e a profundidade da mensagem.
Correção do conteúdo (Alencar e Marques, 2021)	Revisão e aprimoramento de um texto que visa verificar e corrigir possíveis erros, inconsistências, imprecisões ou incoerências no conteúdo, que podem prejudicar a qualidade, a credibilidade e a eficácia da comunicação.
Criatividade (Alencar e Marques, 2021)	Oferece ao usuário a oportunidade do uso da imaginação, da intuição, da curiosidade e da inventividade para criar ou transformar algo existente de maneira inovadora.
Desafios pedagógicos (Junior et al., 2020)	Exploração é a abordagem preferida e encoraja estudantes a buscarem conhecimento de modo independente.
Design estético e minimalista (Alencar e Marques, 2021)	Deve-se apresentar exatamente a informação que o usuário precisa no momento, nem mais nem menos.
Diálogos simples e naturais (Alencar e Marques, 2021)	Os “diálogos” do <i>software</i> precisam ser simples, diretos, naturais, e presentes nos momentos em que são necessários.
Efetividade da aprendizagem (Alencar e Marques, 2021)	Oferece ambiente para o aluno atingir os objetivos de aprendizagem em um determinado processo de ensino-aprendizagem.
Elementos multimídia (Alencar e Marques, 2021)	Os elementos multimídia em aplicativos educacionais podem facilitar a compreensão, a memorização e a motivação dos alunos, bem como estimular diferentes habilidades cognitivas, como atenção, percepção, raciocínio, etc.

Esforço cognitivo das atividades (Alencar e Marques, 2021)	O grau de esforço cognitivo que um usuário deve ter para utilizar o aplicativo e adquirir conhecimento.
Estilo de aprendizagem (Alencar e Marques, 2021)	A aplicação deve apresentar de diferentes formas de estilos de aprendizagem. Destas formas se destaca o método VAC (VISUAL, AUDITIVO e CINESTÉSICO) que é baseado nos sentidos e responde com eficiência as expectativas e exigências da escola.
Evolução do grau de complexidade (Alencar e Marques, 2021)	Deve ir incrementando o grau de dificuldade com o avançar do uso da aplicação.
Feedback (Alencar e Marques, 2021)	Oferecer retorno da avaliação do desempenho do usuário sobre ações ou resultados obtidos.
Filosofia pedagógica (Alencar e Marques, 2021)	Fundamentos pedagógicos que embasam o SE e a opção pedagógica utilizada no seu desenvolvimento.
Flexibilidade e eficiência de uso (Alencar e Marques, 2021)	O <i>software</i> precisa ser flexível, ou seja, ser familiar para usuários avançados e ao mesmo tempo ser fácil para usuários leigos.
Fonte de conteúdo (Alencar e Marques, 2021)	Todo material disponibilizado pela aplicação deve ser de fonte confiável.
Ganho de conhecimento (Alencar e Marques, 2021)	Deve prover um sistema de avaliação com o objetivo de verificar se a aplicação está provendo ganho de conhecimento para o usuário.
Gerenciamento de conteúdo (Alencar e Marques, 2021)	O conteúdo deve ser gerenciado, controlado e monitorado.
Guia de apoio pedagógico (Alencar e Marques, 2021)	Material com o objetivo de orientar e auxiliar o trabalho pedagógico, oferecendo sugestões de atividades, metodologias, recursos e avaliações relacionados aos conteúdos curriculares.
Identifica os objetivos pedagógicos (Alencar e Marques, 2021)	Expressam o que se espera que os alunos saibam, compreendam, façam ou sejam capazes de fazer ao final de uma unidade, curso ou programa.
Identificação de abordagem pedagógica (Junior et al., 2020)	Fundamentos pedagógicos que embasam o SE e a opção pedagógica utilizada no seu desenvolvimento.

Identificação de deficiências (Alencar e Marques, 2021)	Capacidade de auxiliar na identificação de lacunas de conhecimento.
Identificação de perfil (Alencar e Marques, 2021)	Deve ser possível a identificação dos diferentes perfis de aluno de acordo com suas características, como idade, nível de conhecimento, entre outros.
Indicação dos conteúdos prévios (Junior et al., 2020)	Apresenta os conteúdos que são pré-requisitos para o aprendizado.
Iniciativa de aquisição de conhecimento (Alencar e Marques, 2021)	A aplicação deve estimular o usuário a tomar a iniciativa em relação a aquisição de conhecimento, permitindo que o usuário decida sobre quais tarefas deseja realizar.
Integração de conteúdo (Alencar e Marques, 2021)	Forma de articular diferentes áreas do conhecimento para um propósito comum, como produzir novos conhecimentos ou resolver problemas reais.
Interação entre usuários (Alencar e Marques, 2021)	A aplicação deve permitir a interação entre seus usuários, podendo prover este recurso através de blogs, fóruns, grupos, entre outros.
Interação que auxilie na superação de conflitos cognitivos (Junior et al., 2020)	O aplicativo estimula o aprendizado e melhoram as habilidades gradualmente com a superação dos desafios.
Interação social (Alencar e Marques, 2021)	É possível interagir com outras pessoas durante o aplicativo. Promove momentos de cooperação e/ou competição entre os jogadores.
Interdisciplinaridade (Junior et al., 2020)	Faz a interação de conteúdos de duas ou mais disciplinas.
Interpretação dos erros (Junior et al., 2020)	Identificar, analisar e compreender os erros cometidos em uma determinada situação ou atividade, buscando aprender com eles e evitar repeti-los.
Limitações de aprendizagem (Alencar e Marques, 2021)	Capacidade de auxiliar na identificação dos obstáculos ou dificuldades que interferem no processo de aprendizagem dos alunos.
Linguagem (Soad e Barbosa, 2017)	A linguagem do SE possui nível de compreensão para alunos e conceitos embutidos corretos .
Objetivos de aprendizagem (Soad e Barbosa, 2017)	Resultados esperados do processo de ensino e aprendizagem, ou seja, o que os alunos devem saber, compreender ou fazer após uma determinada atividade ou unidade de ensino.

Objetivos educacionais (Soad e Barbosa, 2017)	Orientam a prática pedagógica e a elaboração dos currículos escolares, buscando atender às necessidades e demandas sociais, culturais, políticas e econômicas.
Operabilidade (Alencar e Marques, 2021)	Refere-se ao grau em que um produto de <i>software</i> é de fácil utilização e controle.
Organização (Alencar e Marques, 2021)	A organização dos conteúdos deve ser adaptável ao usuário.
Deve-se prover uma navegação fácil e intuitiva.	
Papéis dos participantes (Alencar e Marques, 2021)	O papel do usuário é definido buscando atingir o objetivo pedagógico do aplicativo.
Papel do professor (Alencar e Marques, 2021)	O <i>software</i> pode ser usado sem a intervenção de um instrutor. Os professores atuam no papel de guias, monitores, treinadores, tutores ou facilitadores.
Permite o registro de desempenho do estudante (Alencar e Marques, 2021)	Registra a evolução e o desempenho do usuário ao avançar nas tarefas.
Pertinência em relação ao programa curricular (Alencar e Marques, 2021)	O <i>software</i> educacional deve ser adequado e pertinente em relação a uma disciplina específica ou a um contexto educacional e permitir a identificação do modelo de aprendizagem que ele privilegia, deixando explícitos os seus objetivos pedagógicos.
Procedência do conteúdo (Alencar e Marques, 2021)	Deve ser oferecido conteúdo fornecido por materiais de procedência confiável.
Processos de ensino e aprendizagem (Alencar e Marques, 2021)	Processos inter-relacionados que envolvem as ações dos professores e dos alunos para promover a construção do conhecimento em diferentes contextos educacionais
Processos educativos e de aprendizagem definidos (Alencar e Marques, 2021)	Os processos educacionais e de aprendizagem utilizados na aplicação devem ser bem definidos.
Público Alvo (Alencar e Marques, 2021)	Segmento da sociedade com determinadas características em comum (idade, sexo, profissão, interesses, etc.) ao qual se dirige o aplicativo .
Qualidade audiovisual (Alencar e Marques, 2021)	A aplicação deve prover a aprendizagem utilizando mídias com qualidade.

Qualidade de conteúdo (Alencar e Marques, 2021)	A aplicação deve prover conteúdo com qualidade, além de ser relacionado com situações e problemas de interesse do aluno, o conteúdo não deve possuir erros ortográficos, mensagens invasivas, negativas e com conteúdo de discriminação racial ou religiosa.
Recursos didáticos (Alencar e Marques, 2021)	Recursos didáticos oferecidos para auxiliar o aprendizado.
Relevância (Alencar e Marques, 2021)	O conteúdo do aplicativo é relevante para os meus interesses. É claro para mim como o conteúdo do aplicativo está relacionado com a disciplina. O aplicativo é um método de ensino adequado para esta disciplina. Eu prefiro aprender com este aplicativo do que de outra forma (outro método de ensino).
Retrabalho com os conhecimentos prévios (Junior et al., 2020)	Consiste em revisar, reforçar ou ampliar os saberes que os alunos já possuem sobre um determinado assunto, antes de introduzir novos conteúdos.
Reutilização de conteúdo de aprendizagem (Alencar e Marques, 2021)	Capacidade da aplicação em prover o reúso de conteúdo.
Reutilização de materiais (Alencar e Marques, 2021)	Aproveita materiais educacionais já existentes para novos fins ou contextos de ensino-aprendizagem.
Saídas Claramente Demarcada (Alencar e Marques, 2021)	As saídas do aplicativo devem ser utilizadas de maneira eficiente, ajustando a qualidade das mídias de acordo com a velocidade da rede
Usuário alvo (Alencar e Marques, 2021)	Deve-se definir o tipo de usuário que se deseja atingir.
Utilização do SE como ferramenta didática (Soad e Barbosa, 2017)	Programas e aplicativos de tipo didático, criados com o propósito explícito de contribuir para o ensino e a aprendizagem, como recursos complementares ou integrados ao currículo escolar.
Visualizações separadas (Alencar e Marques, 2021)	Separar visão de aluno e tutor, ou seja, o aplicativo deve prover visões diferentes de acordo com o perfil do usuário que o está acessando.

Critérios relacionados a Característica Técnica	
Critério	Definição

Acessibilidade (Alencar e Marques, 2021)	Mecanismos de acessibilidade, permitindo a utilização por pessoas com necessidades especiais.
Acessível (Alencar e Marques, 2021)	O aplicativo deve estar disponível e acessível a qualquer hora e em qualquer lugar.
Acesso (Alencar e Marques, 2021)	Deve-se prover acesso seguro e privado, ou seja, o usuário e seus dados devem ter a privacidade assegurada.
Acurácia (Alencar e Marques, 2021)	Fornecimento de resultados exatos e livre de erros.
Adaptabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo <i>software</i> considerado.
Adaptação às condições físicas (Alencar e Marques, 2021)	O grau em que o aplicativo se adapta às condições físicas, por exemplo, ajuste de luz, controle de ruído, condições climáticas, entre outros.
Adaptação de dispositivos (Alencar e Marques, 2021)	Grau em que o aplicativo consegue se adaptar a diferentes tipos de dispositivos móveis. Com a grande quantidade de dispositivos móveis e suas variadas características, o aplicativo deve conseguir se adaptar, permitindo assim o acesso ao aplicativo por diversos dispositivos móveis.
Adequação de mensagem à funcionalidade e ao usuário (Alencar e Marques, 2021)	Mensagens operacionais explicitam de forma compreensiva e objetiva informações da aplicação, sendo de fácil compreensão ao usuário.
Adequação funcional (Alencar e Marques, 2021)	Refere-se à disponibilidade de funções que atendam as necessidades implícitas e explícitas em relação à aplicação.
Ajuda e/ou documentação (Junior et al. 2020)	Diz respeito a todos os mecanismos e documentos que permitam ajudar o usuário no momento em que o mesmo apresente alguma dificuldade
Analisabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de permitir o diagnóstico de deficiências ou causas de falhas no <i>software</i> , ou a identificação de partes a serem modificadas.
Apreensibilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de possibilitar ao usuário aprender sua aplicação.

Aprendizado orientado a um objetivo (Alencar e Marques, 2021)	O aplicativo deve explicar claramente seu objetivo de ensino, para que o usuário possa identificar a utilidade do mesmo para seu processo de aprendizado. Se aplicável ao contexto do ensino, conceder ao usuário a possibilidade de configurar objetivos e metas. As pessoas armazenam melhor suas experiências quando estas estão relacionadas as metas e objetivos.
Aprendizibilidade (Alencar e Marques, 2021)	Aprender a jogar este jogo foi fácil para mim. Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente
Apresentação de informações (Alencar e Marques, 2021)	Grau de limitação das informações de acordo com a área de exibição da tela. As informações devem ser adaptadas ao tamanho da tela ou divididas em páginas.
Aspectos Motivadores (Alencar e Marques, 2021)	O uso de recursos motivacionais (figuras, sons, animações) tem grande importância em <i>software</i> do tipo educacional infantil. Estes recursos podem estimular as crianças e proporcionar o aumento da vida útil do programa ao fazer com que ela deseje usá-lo por mais vezes.
Atenção focada (Alencar e Marques, 2021)	Houve algo interessante no início do jogo que capturou minha atenção. Eu estava tão envolvido no jogo que eu perdi a noção do tempo. Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.
Atração e/ou motivação (Alencar e Marques, 2021)	O aplicativo deve ser atrativo e motivar os usuários. Deve ser esteticamente atraente e agradável de usar.
Atratividade (Alencar e Marques, 2021)	Aplicativo atraente e motivador aos usuários que o utilizarão.
Atualização e configuração (Alencar e Marques, 2021)	A aplicação deve ser atualizada e configurada de maneira rápida e automática.
Auditoria (Alencar e Marques, 2021)	Registro de cada ação no aplicativo, auxiliando na não negação da ação caso o usuário negue-as.
Autenticidade (Alencar e Marques, 2021)	Refere-se à garantia de autenticidade das ações do usuário.
Capacidade de comunicação (Alencar e Marques, 2021)	Deve permitir a transmissão de dados entre múltiplos dispositivos, sem que os mesmos sejam danificados. Aborda, desta forma, mais uma característica que é a confiabilidade
Carga de trabalho (Alencar e Marques, 2021)	A carga de trabalho refere-se a toda informação contida nas interfaces do <i>software</i> educacional infantil que serão utilizadas para a realização de tarefas. As heurísticas carga informacional e brevidade referem-se à carga de trabalho.

Carregamento da tela (Alencar e Marques, 2021)	Quantidade de informações e elementos em uma tela.
Compatibilidade (Junior et al. 2020)	Se refere ao acordo que possa existir entre as características do usuário (memória, percepção, hábitos, competências, idade, expectativas, etc.) e as tarefas, de uma parte, e a organização das saídas, das entradas e do diálogo de uma dada aplicação, de outra
Compleitude da característica (Alencar e Marques, 2021)	As características propostas durante o projeto do aplicativo devem ser realmente implementadas. Algumas melhorias no projeto podem ser identificadas durante o desenvolvimento. Por este motivo, pode-se substituir ou incluir novas características, porém a definição final deverá apresentar todas as características estabelecidas.
Compleitude funcional (Alencar e Marques, 2021)	Refere-se ao grau em que o conjunto de funções é apropriado para que o usuário atinja os seus objetivos, ou seja, esse conjunto de funções existentes permite que o usuário realize suas tarefas.
Comportamento em relação ao tempo (Alencar e Marques, 2021)	Refere-se ao grau em que os tempos de resposta e processamento das funções do sistema são atendidos, de acordo com os requisitos estabelecidos.
Concisão das mensagens (Alencar e Marques, 2021)	Mede a concisão das mensagens de texto operacionais e de erro.
Confiabilidade (Alencar e Marques, 2021)	Refere-se à capacidade da aplicação em proporcionar um comportamento consistente com o esperado durante um longo período de tempo.
Confiança (Alencar e Marques, 2021)	Quando olhei pela primeira vez o jogo, eu tive a impressão de que seria fácil para mim. A organização do conteúdo me ajudou a estar confiante de que eu aprenderia com este jogo.
Consistência e bom aproveitamento do espaço da tela (Alencar e Marques, 2021)	Informações bem distribuídas na tela para melhor compreensão do usuário.
Consistência e padrões (Alencar e Marques, 2021)	Se refere à forma na qual as escolhas na concepção da interface (códigos, denominações, etc) são conservadas idênticas.
Continuidade (Alencar e Marques, 2021)	A aplicação deve ser fluída, não permitindo que o usuário chegue a um ponto da aplicação onde não consiga mais navegar por ela.
Controle do Usuário (Junior et al. 2020)	Se refere ao fato de que os usuários deveriam estar sempre no controle do processamento do sistema (por exemplo, interromper, cancelar, suspender e continuar)

Correspondência entre o <i>software</i> e o mundo real (Junior et al. 2020)	As informações devem ser organizadas conforme o modelo mental do usuário, usando terminologia baseada no usuário.
Criptografia (Alencar e Marques, 2021)	Deve-se prover criptografia para os dados sensíveis trafegados e armazenados pelo aplicativo
Desafio (Alencar e Marques, 2021)	Este jogo é adequadamente desafiador para mim. O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado. O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).
Design de interface (Alencar e Marques, 2021)	Visa maximizar a usabilidade e a experiência do usuário para tornar a interação do usuário efetiva, eficiente e satisfatória
Design estético e minimalista (Junior et al. 2020)	Deve-se apresentar exatamente a informação que o usuário precisa no momento, nem mais nem menos (seguir modelo mental e conceitual do usuário).
Disponibilidade (Alencar e Marques, 2021)	Refere-se à aplicação estar disponível qualquer hora e em qualquer lugar.
Disponibilizar diferentes experimentações de aprendizado, incluindo erros (Alencar e Marques, 2021)	Não há um único caminho que conduz ao sucesso. Permitir que o usuário interprete suas experiências durante e após suas ações a fim de antecipar em quais outros contextos e de que modo essas lições possam ser úteis novamente. Além disso, é essencial incluir os erros de aprendizado como parte do processo de aprendizagem.
Diversão (Alencar e Marques, 2021)	Eu me diverti com o jogo. Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir.
Diversidade de dispositivos (Alencar e Marques, 2021)	Os aplicativos devem ser construídos com base na diversidade de dispositivos móveis disponíveis no mercado.
Eficácia de execução de tarefas (Alencar e Marques, 2021)	Capacidade de facilitar a execução de tarefas, minimizando o tempo, gastos com conexão e quantidade de dados trafegados.
Eficiência do Desempenho (Alencar e Marques, 2021)	Verifica o desempenho em relação à quantidade de recursos utilizados sob condições estabelecidas, ou seja, refere-se à otimização do uso dos recursos disponíveis.
Eficiência na utilização de memória (Alencar e Marques, 2021)	Utilização de técnicas para otimizar o uso da memória do dispositivo.

Eficiência no consumo de energia (Alencar e Marques, 2021)	Utilização eficiente para não desperdiçar energia, dadas as limitações das baterias dos dispositivos.
Engajamento (Alencar e Marques, 2021)	Compromisso funcional com papéis e responsabilidades para que se atinja um objetivo aliado ao envolvimento emocional em desempenhar tais atividades.
Entrada de dados (Alencar e Marques, 2021)	Grau em que é facilitada a entrada de dados no aplicativo. Não se deve exigir a entrada de muitos dados em um aplicativo educacionais, já que existem dificuldades relacionadas ao tamanho da tela e do teclado, diminuindo a agilidade na entrada de dados para o aplicativo.
Equipamentos de <i>software</i> e hardware (Alencar e Marques, 2021)	Verifica o desempenho dos recursos utilizados sob condições estabelecida.
Escalabilidade (Alencar e Marques, 2021)	Capacidade do aplicativo de manipular uma porção crescente de trabalho de forma uniforme ou estar preparado para crescer
Estabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de evitar efeitos inesperados decorrentes de modificações no <i>software</i> .
Estética (Alencar e Marques, 2021)	Capacidade em oferecer ao usuário um <i>software</i> visivelmente agradável, colaborando assim para a satisfação do usuário.
Facilidade da saída de dados para o usuário (Alencar e Marques, 2021)	Mede a adequação do comprimento da saída e o tamanho da tela do dispositivo. Como os dispositivos possuem telas de diferentes tamanhos, é necessário sempre adequar a saída de dados de acordo com elas.
Facilidade de acesso às funcionalidades (Alencar e Marques, 2021)	A maneira como os processos estão apresentados, para tornar a aplicação intuitiva e evitar dúvidas, insatisfação ou frustração no usuário;
Facilidade de instalação (Alencar e Marques, 2021)	Tempo de instalação do aplicativo nos dispositivos. Deve ser facilmente instalado no dispositivo móvel, evitando problemas de versão, sistemas operacionais, distribuição, entre outros.
Facilidade de preenchimento (Alencar e Marques, 2021)	Mede o quão fácil é o preenchimento dos campos de entrada, como o uso de valores padrão, lista de valores e autocompletar. Deve-se sempre minimizar a quantidade de dados necessárias para a execução das tarefas do aplicativo.
Facilidade dos Procedimentos (Alencar e Marques, 2021)	A maneira como os processos estão apresentados, minimizando confusões, dúvidas, insatisfação ou frustração no usuário;

Flexibilidade e eficiência de uso (Junior et al. 2020)	Se refere aos meios colocados à disposição do usuário que lhe permitem personalizar a interface, a fim de levar em conta as exigências da tarefa, de suas estratégias ou seus hábitos de trabalho.
Gestão de dados (Alencar e Marques, 2021)	Controle de entrada e saída de dados considerando as limitações de tela e teclado.
Gestão de erros (Junior et al. 2020)	Diz respeito a todos os mecanismos que permitem evitar ou reduzir a ocorrência de erros e, quando eles ocorrem, que favoreçam sua correção.
Heurísticas Pedagógicas (Alencar e Marques, 2021)	Heurística é a arte de descobrir e inventar, uma característica típica dos seres humanos, principalmente quando estes estão em busca de respostas para questões complexas. No âmbito pedagógico, o método heurístico é aplicado para auxiliar os alunos a encontrarem as respostas sobre determinado assunto por conta própria, através do encaminhamento feito por meio de algumas perguntas, por exemplo.
Homogeneidade do layout (Alencar e Marques, 2021)	Os componentes de layout devem ser padronizados, ou seja, deve-se manter as suas características em todas as telas do aplicativo.
Homogeneidade dos componentes (Alencar e Marques, 2021)	Os componentes utilizados no aplicativo devem seguir um padrão, evitando o uso de uma variedade muito grande de componentes, o que poderia afetar no aprendizado da utilização do aplicativo.
Instabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> para ser instalado em um ambiente especificado.
Instabilidade (Alencar e Marques, 2021)	Capacidade do aplicativo ser instalado e desinstalado em diferentes dispositivos móveis sem implicar em erros.
Integridade (Alencar e Marques, 2021)	Utilização de diferentes perfis de usuário, restringindo assim o acesso a determinadas funcionalidades do aplicativo.
Inteligibilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de possibilitar ao usuário compreender se o <i>software</i> é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.
Interface amigável (Alencar e Marques, 2021)	O grau de superação das dificuldades para inserir dados por meio de telas e teclados pequenos. Uma interface amigável é necessária já que existem diversos tipos de dispositivos, alguns com teclados outros sem, além de diferentes tamanhos de telas.
Interoperabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de interagir com um ou mais sistemas especificados.
Layout (Alencar e Marques, 2021)	Os componentes utilizados no aplicativo devem ser consistentes e uniformes, além de utilizar um menu de opções por níveis adequados e também ser esteticamente atraente e agradável de ser utilizada.

Limpeza de memória (Alencar e Marques, 2021)	Não armazenamento de dados desnecessários no dispositivo, como arquivos que não serão mais utilizados pelo aplicativo.
Manutenção (Alencar e Marques, 2021)	Significa o mínimo de esforço para localizar e corrigir um erro ou inconsistência. Através deste requisito fica claro a importância que se tem em trabalhar e planejar testes e inspeções de código periodicamente a fim de evitar retrabalho e adaptações forçadas no <i>software</i> .
Mensagens concisas (Alencar e Marques, 2021)	Mede a concisão das mensagens de texto operacionais e de erro.
Mensagens de Erro (Alencar e Marques, 2021)	As mensagens de erro do aplicativo devem possuir uma redação simples e clara ao invés de intimidar o usuário.
Mensagens seguras e informações nos dispositivos (Alencar e Marques, 2021)	Mede o grau de utilização de mecanismos de segurança, como identificação, autorização e confidencialidade.
Mobilidade (Alencar e Marques, 2021)	Refere-se à aplicação estar disponível e acessível a qualquer hora e em qualquer lugar.
Modificabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de permitir que uma modificação especificada seja implementada.
Mudança de contexto (Alencar e Marques, 2021)	Comportamento do aplicativo ao mudar o contexto (por exemplo, quando um campo de entrada depende da localização do usuário, ele deverá ser alterado quando o usuário se mover). As funções que dependem da localização também devem se adequar às mudanças.
Navegação (Alencar e Marques, 2021)	A navegação deve ser simples, fácil e intuitiva, além de possuir alternativas e prover facilidades, tais como índice, botão de voltar, imprimir, entre outros.
Operacionalidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> se manter operacional após uma modificação implementada.
Oportunidade de retornar à atividade em caso de erro (Alencar e Marques, 2021)	Oferecer ao usuário a possibilidade de retornar à questão respondida de forma errada anteriormente.
Orientação e instruções (Alencar e Marques, 2021)	Apresentação das informações necessárias de acordo com a área de exibição da tela.

Permitir salvar pontos de backup em caso de bateria baixa (Alencar e Marques, 2021)	Deve salvar pontos de backup em caso de bateria baixa, podendo ser automática ou sugerindo a execução da ação para o usuário.
Pertinência em relação ao programa curricular (Alencar e Marques, 2021)	O <i>software</i> educacional infantil deve ser adequado e pertinente em relação a uma disciplina específica ou a um contexto educacional e permitir a identificação do modelo de aprendizagem que ele privilegia, deixando explícitos os seus objetivos pedagógicos.
Portabilidade (Alencar e Marques, 2021)	A importação e exportação para outros sistemas de dados e serviços devem ser fáceis, além de seus módulos poderem ser utilizados facilmente por outros aplicativos.
Precisão (Alencar e Marques, 2021)	Fornecer resultados precisos, ou seja, os resultados fornecidos devem ser exatos, não podendo conter erros. Se esta característica não for satisfeita, ela pode afetar a confiabilidade do aplicativo.
Prevenção de erros (Junior el al. 2020)	Diz respeito aos mecanismos empregados para detectar e prevenir os erros de entradas de dados, comandos, possíveis ações de consequências desastrosas e/ou não recuperáveis.
Privacidade (Alencar e Marques, 2021)	Deve preservar a privacidade do usuário, provendo confidencialidade, anonimato, privacidade e confiança.
Proteção de erro do usuário (Alencar e Marques, 2021)	Quando eu cometo um erro é fácil de me recuperar rapidamente.
Rastreabilidade (Alencar e Marques, 2021)	Deve-se manter os registros de eventos gravados. Desta maneira é possível rastrear os caminhos feitos pelo usuário no aplicativo.
Reconhecer em vez de lembrar (Junior el al. 2020)	Diz respeito a todos os elementos da interface que têm um papel importante na redução da carga cognitiva e perceptiva do usuário e no aumento da eficiência do diálogo.
Recuperabilidade (Junior el al. 2020)	Capacidade do produto de <i>software</i> de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Recursos (Junior el al. 2020)	Capacidade do produto de <i>software</i> de usar tipos e quantidades apropriados de recursos, quando o <i>software</i> executa suas funções sob condições estabelecidas.
Recursos motivacionais (Alencar e Marques, 2021)	O uso de recursos motivacionais (figuras, sons, animações) tem grande importância em <i>software</i> do tipo educacional infantil. Estes recursos podem estimular as crianças e proporcionar o aumento da vida útil do programa ao fazer com que ela deseje usá-lo por mais vezes.

Recursos multimídia (Alencar e Marques, 2021)	Capacidade de suportar diferentes tipos de mídias, como vídeo, texto, áudio, entre outros.
Requisitos mínimos (Alencar e Marques, 2021)	Cumprimento dos requisitos mínimos estabelecidos para uso do aplicativo, permitindo sua utilização sem perda de desempenho.
Resolução de erro (Alencar e Marques, 2021)	O <i>software</i> educacional infantil deve tratar os erros que ocorrerem de forma diferenciada, dependendo da sua classificação: erros de utilização ou erros conceituais. Duas heurísticas estão relacionadas à gestão de erros: prevenção de erros e auxílio no reconhecimento, no diagnóstico e na recuperação de erros. A primeira diz respeito a erros de utilização do <i>software</i> e a segunda, aos erros de utilização ou conceituais
Resposta Imediata e Visibilidade (Alencar e Marques, 2021)	Latência em que as respostas as solicitações são apresentadas na tela.
Segurança (Alencar e Marques, 2021)	Refere-se ao grau em que o produto de <i>software</i> protege as informações e dados, protegendo-os de acesso não autorizado e garantido seu acesso de acordo com os diferentes níveis de autorização.
Segurança de Acesso (Junior et al. 2020)	Capacidade do produto de <i>software</i> de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.
Substituibilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de ser usado em substituição a outro produto de <i>software</i> especificado, com o mesmo propósito e no mesmo ambiente.
Substituibilidade (Alencar e Marques, 2021)	Refere-se ao grau em que um produto de <i>software</i> pode ser substituído por outro em um mesmo ambiente e com os mesmos objetivos.
Sucesso de instalação (Alencar e Marques, 2021)	O grau em que um aplicativo educacionais pode ser instalado ou desinstalado com sucesso considerando vários tipos de redes e dispositivo.
Tamanho da aplicação (Alencar e Marques, 2021)	Quantidade de espaço necessário para que a aplicação possa ser utilizada no dispositivo.
Telas por tarefa (Alencar e Marques, 2021)	Mede o número de telas envolvidas em uma tarefa. Quanto menos telas, melhor a eficácia da utilização da funcionalidade, já que assim o usuário vai chegar mais rápido ao seu objetivo. Entretanto, deve-se evitar o excesso de informações dispostas em uma única tela.

Tempo de aprendizado (Alencar e Marques, 2021)	Tempo em que o usuário consegue aprender a utilizar a aplicação.
Tempo de carregamento (Alencar e Marques, 2021)	Mede o nível de qualidade em relação ao tempo de carregamento.
Tempo de recuperação após desconexão (Alencar e Marques, 2021)	O grau em que o aplicativo consegue restabelecer a conexão após uma desconexão. O aplicativo deve funcionar mesmo com interrupções inesperadas. Caso ocorra, deve ser capaz de se recuperar ou guiar o usuário para a recuperação. Desta forma, o usuário será capaz de reconhecer, diagnosticar e recuperar-se de erros.
Tempo de resposta (Junior et al. 2020)	Capacidade do produto de <i>software</i> de fornecer tempos de resposta e de processamento, além de taxas de transferência, apropriados, quando o <i>software</i> executa suas funções, sob condições estabelecidas.
Tempo de resposta para obter informação da memória (Alencar e Marques, 2021)	Mede o tempo de resposta para obter informações armazenadas na memória do dispositivo.
Tempo médio de resposta (Alencar e Marques, 2021)	O grau em que o aplicativo fornece resposta adequada às consultas do usuário, independentemente da localização e horário. Esta característica está relacionada a aplicativos do tipo cliente-servidor, já que o tempo medido nessa característica é o tempo em que o servidor retorna a resposta da requisição para o cliente (usuário) que a requisitou.
Testabilidade (Junior et al. 2020)	Capacidade do produto de <i>software</i> de permitir que o <i>software</i> , quando modificado, seja validado.
Tolerância a falhas (Alencar e Marques, 2021)	Reação diante de uma situação anômala, ou seja, o aplicativo continua funcionando mesmo que ocorram falhas.
Transferência de dados (Alencar e Marques, 2021)	Capacidade de minimização dos dados, para melhorar o tempo de transferência, gastos com conexão e quantidade de dados trafegados.
Usabilidade (Alencar e Marques, 2021)	Avaliar aspectos de usabilidade, verificando se o aplicativo educacionais pode ser entendido, aprendido, usado e atraente ao usuário.
Utilização de perfil de usuário (Alencar e Marques, 2021)	Mede o alcance em que o aplicativo se baseia no perfil do usuário, ajustando entradas e saídas para usuário e dispositivo.

Utilização de recursos computacionais (Soad e Barbosa, 2017)	Refere-se ao grau em que a quantidade e os tipos de recursos são utilizados pelo produto de <i>software</i> ao realizar suas funções.
Variação da rede (Alencar e Marques, 2021)	Comportamento do aplicativo ao variar sua conexão (por exemplo, pode-se optar por trabalhar apenas com texto, eliminando as imagens e permitindo o funcionamento mesmo com uma conexão ruim).
Visibilidade do estado do sistema (Junior et al. 2020)	Feedback imediato diz respeito às respostas do sistema às ações do usuário

APÊNDICE B – SCRIPTS DA APLICAÇÃO DOS ALGORITMOS

```

1     # Criando e treinando o modelo KNN
2 knn = KNeighborsClassifier(n_neighbors=24)
3 knn.fit(X_train, y_train)
4
5 # Fazendo as previsoes no conjunto de teste
6 y_pred_knn = knn.predict(X_test)

```

Listing B.1 – Definição do Algoritmo K-Nearest Neighbor

```

1 # importando bibliotecas
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.metrics import classification_report
4
5 # criando modelo random forest
6 random_forest = RandomForestClassifier(n_estimators=100, random_state
    =42)
7 random_forest.fit(X_train, y_train)
8
9 # fazendo a previsao dos conjuntos de teste
10 y_pred_RF = random_forest.predict(X_test)

```

Listing B.2 – Definição do Algoritmo Random Forest

```

1 # importando bibliotecas
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Dense
4 from tensorflow.keras.optimizers import Adam
5
6 # Crie a arquitetura da RNA
7 model = Sequential()
8 model.add(Dense(3, activation='relu', input_shape=(X_train.shape[1],)))
9 model.add(Dense(100, activation='relu'))
10 model.add(Dense(len(label_encoder.classes_), activation='softmax'))
11
12 # Compilar o modelo
13 model.compile(optimizer=Adam(learning_rate=0.001), loss='
    sparse_categorical_crossentropy', metrics=['accuracy'])
14
15 # treina o modelo
16 model.fit(X_train.toarray(), y_train, epochs=50, batch_size=32)
17
18 # Faça previsoes no conjunto de teste
19 y_pred = model.predict(X_test.toarray())
20 y_pred = label_encoder.inverse_transform(y_pred.argmax(axis=1))

```

Listing B.3 – Definição do Algoritmo Redes Neurais Artificiais