



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal Catarinense
Campus Rio do Sul

MARCOS MOMM

**AVALIAÇÃO DE USABILIDADE EM PLATAFORMAS DE ENSINO DE
ALGORITMOS: AUXILIO AO PROCESSO DE ENSINO-APRENDIZAGEM**

Rio do Sul
2018

MARCOS MOMM

**AVALIAÇÃO DE USABILIDADE EM PLATAFORMAS DE ENSINO DE
ALGORITMOS: AUXÍLIO NO PROCESSO DE ENSINO APRENDIZAGEM**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Instituto Federal Catarinense – campus Rio
do Sul, para obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Wesley dos Reis Bezerra, M. Eng.

Co-orientador: Daniel Baumann, Esp.

Rio do Sul

2018

MARCOS MOMM

**AVALIAÇÃO DE USABILIDADE EM PLATAFORMAS DE ENSINO DE
ALGORITMOS: AUXÍLIO NO PROCESSO DE ENSINO APRENDIZAGEM**

Este Trabalho de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Rio do Sul.

Rio do Sul (SC), 07 de dezembro de 2018

Prof. e Orientador Wesley dos Reis Bezerra, M. Eng.

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Rio do Sul

Prof. e Co-Orientador Daniel Baumann, Esp.

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Rio do Sul

Prof. Indianara Gonçalves Camilo Curvêllo, Esp.

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Rio do Sul

Prof. Fábio Alexandrini, Dr. Eng.

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Rio do Sul

AGRADECIMENTOS

A Deus que me concede a vida, saúde, inteligência e a capacidade.

A meus pais, Mario Momm e Lucia Momm, e minha irmã Cristiani Momm pelo incentivo e motivação a mim dedicados.

Agradeço a minha esposa Silvana Floriano da Silva Momm, pela paciência, compreensão, incentivo e motivação a mim dedicados.

Ao meu orientador, Wesley dos Reis Bezerra, e co-orientador, Daniel Baumann, pelo auxílio, apoio, conselhos e orientações prestados com muita dedicação durante o desenvolvimento deste trabalho.

E por fim, a todos os professores e colegas do curso, sem os quais muitos dos conhecimentos deixariam de ser adquiridos durante esta caminhada.

RESUMO

A educação tem apresentado um grande interesse e uma necessidade crescente de abordagens pedagógicas que contemplem experiências cognitivas dos alunos. As novas Tecnologia de Informação e Comunicação, sobretudo o advento do computador, mudaram a provocaram novas maneiras pedagógicas entre o professor e o aluno. Este trabalho apresenta como objetivo, identificar a plataforma de ensino mais adequada quanto a usabilidade para lógica de programação, ensino de algoritmos, para os alunos iniciantes em matérias de programação. Para a avaliação desses ambientes de aprendizagem, como ferramenta de avaliação, foi definido o uso de um *checklist*, que é ferramenta desenvolvida e validade por profissionais da área de usabilidade e ergonomia. O *checklist* escolhido para a aplicação do teste foi o TICESE, que tem um enfoque na avaliação da usabilidade em *interfaces*, seus critérios são desenvolvidos, buscando avaliar e selecionar o *software* educacional que possibilita uma melhor experiência de uso para o aluno. Os resultados mostraram que a plataforma Khan Academy, obteve a nota mais alta em relação à Codecademy.

Palavras chaves: Plataforma de ensino; Algoritmos; *Checklist*.

ABSTRACT

Education has shown a great interest and a growing need for pedagogical approaches that contemplate students' cognitive experiences. The new Information and Communication Technology, especially the advent of the computer, have changed and provoked new pedagogical ways between the teacher and the student. This work aims to identify the most appropriate teaching platform for usability for programming logic, algorithm teaching, for beginner students in programming. For the evaluation of these learning environments, as an evaluation tool, we defined the use of a checklist, that is a tool developed and validated by professionals in the area of usability and ergonomics. The checklist chosen for the test application was TICESE, which focuses on the evaluation of interface usability, its criteria are developed, seeking to evaluate and select the educational software that enables a better use experience for the student. The results showed that the platform Khan Academy, obtained the highest mark in relation to the Codecademy.

Keywords: Educational platform; Algorithms; Check list.

LISTA DE GRÁFICOS

Gráfico 1 – Notas geral das plataformas de ensino	78
---	----

LISTA DE FIGURAS

Figura 1 – Procedimento do trabalho	19
Figura 2 – Pseudocódigo maior número	22
Figura 3 – Processo de interação com a <i>interface</i>	28
Figura 4 – Classificação dos instrumentos avaliativos	35
Figura 5 – Diagrama de preparação do teste	50
Figura 6 – Diagrama de aplicação do teste	51
Figura 7 – Diagrama de sequência do processo de avaliação	52
Figura 8 – Tela da estrutura do curso Khan Academy	56
Figura 9 – Tela de programação	57
Figura 10 – Tela da estrutura do curso Codecademy	59
Figura 11 – Diagrama de procedimento da avaliação	61
Figura 12 – Comparação de mensagens de erros das plataformas	68

LISTA DE QUADROS

Quadro 1 – Identificação dos pesos	40
Quadro 2 – Identificação das notas.....	40
Quadro 3 – Trabalhos correlatos	46
Quadro 4 – Arranjo dos critério.....	61
Quadro 5 – Critérios de avaliação do <i>checklist</i>	62
Quadro 6 – Planilha para apresentação dos resultados.....	63
Quadro 7 – Notas dos critérios avaliação da documentação	65
Quadro 8 – Notas critério condução	67
Quadro 9 – Notas do critério carga de trabalho.....	69
Quadro 10 – Notas dos critérios recursos de apoio e adaptabilidade	71
Quadro 11 – Notas dos critérios controle explícito e gestão de erros	73
Quadro 12 – Notas avaliação aprendizagem, homogeneidade e compatibilidade	75
Quadro 13 – Notas dos critérios significado de códigos e adequabilidade	77
Quadro 14 – Notas das avaliações das plataformas.....	87
Quadro 15 – Notas gerais das plataformas e dos critérios.....	89

LISTA DE ABREVIATURAS E SIGLAS

AVA – Ambiente Virtual de Aprendizagem

GQM – Goal Question Metric

IDE – *Integrated Development Environment* - Ambiente de Desenvolvimento Integrado

IFC – Instituto Federal de Educação, Ciência e Tecnologia Catarinense

IHC – Interação Humano-Computador

ISO - International Organization for Standardization

OA – Objeto de Aprendizagem

SBC – Sociedade Brasileira de Computação

SE – *Software* Educativo

TIC – Tecnologia de Informação e Comunicação

TICESE – Técnica de Inspeção de Conformidade Ergonômica de *Software* Educacional

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 PROBLEMATIZAÇÃO	15
1.1.1 Solução proposta	16
1.1.2 Delimitação do escopo.....	16
1.1.3 Justificativa.....	17
1.2 OBJETIVOS	18
1.2.1 Objetivo geral	18
1.2.2 Objetivos específicos	18
1.3 METODOLOGIA	19
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 PROCESSO DE ENSINO-APRENDIZAGEM.....	21
2.1.1 Processo de ensino-aprendizagem de algoritmos	21
2.2 AMBIENTES VIRTUAIS DE APRENDIZAGEM	23
2.2.1 <i>Softwares</i> educativos	24
2.2.2 Classificação dos <i>softwares</i> educativos	25
2.3 ERGONOMIA EM INTERAÇÃO HUMANO-COMPUTADOR.....	26
2.3.1 Interação Humano-Computador	26
2.3.2 Interface e interação	27
2.4 USABILIDADE.....	28
2.4.1 Heurísticas de usabilidade.....	29
2.4.2 Sistema e o grau de usabilidade	31
2.5 TESTE DE USABILIDADE	31
2.5.1 Processo de avaliação.....	32
2.5.1.1 Design do questionário para coleta de dados	33
2.6 MÉTODOS DE AVALIAÇÃO	34
2.6.1 Avaliação com <i>checklist</i>	36

2.6.2 Checklist TICESE	38
2.6.2.1 Instruções para aplicação do <i>checklist</i> TICESE.....	39
3 TRABALHOS CORRELATOS.....	42
3.1 AVALIAÇÃO DA USABILIDADE DA <i>INTERFACE</i> DO AMBIENTE VIRTUAL MOODLE	42
3.2 AVALIAÇÃO DE OBJETOS DE APRENDIZAGEM.....	43
3.3 USABILIDADE E OS CRITÉRIOS DE AVALIAÇÃO PARA AMBIENTE VIRTUAL DE ENSINO-APRENDIZAGEM DO CURSO LETRASLIBRAS	44
3.4 AVALIAÇÃO DE <i>SOFTWARES</i> EDUCATIVOS PARA O ENSINO DE PROGRAMAÇÃO	45
3.5 COMPARAÇÃO ENTRE TRABALHOS CORRELATOS	46
4 ESPECIFICAÇÕES FORMAIS.....	48
4.1 CRITÉRIOS PARA A ESCOLHA DAS PLATAFORMAS.....	48
4.2 DIAGRAMA DE ATIVIDADE	49
4.3 DIAGRAMA DE SEQUÊNCIA.....	52
5 DESENVOLVIMENTO	54
5.1 FUNCIONAMENTO DAS PLATAFORMAS DE PROGRAMAÇÃO	54
5.1.1 Plataforma Khan Academy	55
5.1.2 Plataforma Codecademy	58
5.2 PREPARAÇÃO PARA APLICAÇÃO DO <i>CHECKLIST</i> TICESE	59
5.3 APLICAÇÃO DO <i>CHECKLIST</i> TICESE	60
6 ANÁLISE DOS RESULTADOS.....	64
6.1 AVALIAÇÃO DOS RESULTADO POR CRITÉRIO	64
6.1.1 Dados de identificação e qualidade da informação impressa	65
6.1.2 Condução	66
6.1.3 Carga de trabalho	69
6.1.4 Recursos de apoio à compreensão dos conteúdos e adaptabilidade	70
6.1.5 Controle explícito e gestão de erros.....	72

6.1.6 Avaliação de aprendizagem, homogeneidade e compatibilidade.....	74
6.1.7 Significado de códigos e combinações e adequabilidade	76
6.1.8 Resultado final.....	77
7 CONSIDERAÇÕES FINAIS	80
REFERÊNCIAS	83
APÊNDICE A – NOTAS DAS AVALIAÇÕES DAS PLATAFORMAS	87
APÊNDICE B – NOTAS GERAIS DAS PLATAFORMA E DOS CRITÉRIOS	89
ANEXO A – I MÓDULO DE CLASSIFICAÇÃO	90
ANEXO B – II MÓDULO DE AVALIAÇÃO	91
ANEXO C – III MÓDULO DE AVALIAÇÃO CONTEXTUAL	101

1 INTRODUÇÃO

As novas Tecnologia de Informação e Comunicação (TICs), sobretudo o advento do computador, segundo Kenski (2012), movimentaram a educação e provocaram diferentes metodologias entre a abordagem do professor, a compreensão do aluno e o conteúdo em que está sendo ensinado. A imagem, o som, torna o processo de ensino-aprendizagem mais interativo para o aluno e conseqüentemente aumenta a absorção do conteúdo que está sendo ministrado.

Alguns alunos têm uma baixo índice de entendimento e compreensão das disciplinas que exigem programação, e isso tem sido um problema enfrentado por muitas instituições de ensino. Uma das abordagens para enfrentar esse problema, de acordo com Mota, Pereira e Favero (2008), é a utilização de ferramentas para auxiliar o aluno no processo de ensino aprendizagem. Essas ferramentas, tem como objetivo ajudar o aluno a compreender e entender determinados conteúdos, dando suporte a atividades de ensino, auxiliando o aluno demonstrando explicações e inserindo-o na resolução de exercícios, diversificando assim a maneira como é ministrado o conteúdo.

Os sistemas destinados ao suporte de atividades de ensino-aprendizagem são conhecidos como Ambientes Virtuais de Aprendizado (AVA), que são ferramentas disponíveis na *internet*. Buscam integrar múltiplas mídias, recursos, gerenciamento de matérias pedagógicos e do desenvolvimento dos alunos, e desenvolver uma ambiente de interação entre pessoas e colaboração entre os usuários, (KEMCZINSKI, 2005).

Para um AVA, a usabilidade é um atributo desejável. A usabilidade está relacionada com os aspectos que garantem a facilidade e o aprendizado no uso, além da satisfação do usuário com o sistema. Um fator importante, é a interação do usuário com a *interface* devido a fins educacionais. Atender a critérios de usabilidade se torna fundamental, pois, caso o aluno tenha dificuldades na utilização do sistema, pode acarretar na desmotivação por parte do aluno e assim o aluno deixa de utilizar a ferramenta, (WINCLER, 2001).

Para realizar avaliação do AVA, nos critérios de usabilidade, é utilizado a aplicação de um teste de usabilidade. Segundo Barbosa e Silva (2010), é um método de avaliação de Interface Humano-Computador (IHC), criado para encontrar problemas de usabilidade durante um processor interativo. Essa técnica orienta os avaliadores a inspecionar sistematicamente a *interface* em busca de problemas que prejudiquem a usabilidade do sistema.

Para a realização do teste de usabilidade em AVA, buscou-se determinar a escolha de *Softwares* Educativos (SE), tendo em vista a utilização de plataformas de programação *online*. Segundo Moraes (2003) e Silva et al. (2016), os SE são desenvolvidos para auxiliar o processo de ensino-aprendizagem, desenvolvendo atividades interativas para uma melhor imersão do aluno na construção do conhecimento.

As plataformas de ensino de programa *online*, são SE que buscam auxiliar os alunos na construção do conhecimento. Elas inserem o aluno, em um contexto de explicação e prática, utilizando de recursos de som e vídeo e de uma *interface* que o aluno possa desenvolver as suas atividades na própria plataforma. Essas ferramentas visam ajudar o aluno na construção do conhecimento, e não ser a única forma de o aluno aprender a programar, sendo o professor papel importante e fundamental para o desenvolvimento do aluno.

Para a avaliação das plataformas de programação *online*, propõem-se a aplicação de um *checklist*, como método de avaliação de usabilidade em *interface*. A ferramenta de avaliação permite a verificação de critérios, que procuram analisar e avaliar problemas de ergonomia e usabilidade. Os critérios e os critérios desenvolvidos para avaliação, são validados e definidos por profissionais nas áreas de usabilidade e ergonomia, o conhecimento dessas duas áreas está aplicado no desenvolvimento do ferramenta.

Este trabalho, tem como objetivo analisar a usabilidade da *interface* de plataformas de ensino de programação *online*, com base nos resultados da avaliação pode ser indicada aos alunos como ferramenta de apoio no ensino e aprendizagem de programação. Tendo como base os critérios de escolha definidos pelo avaliador, as plataformas a serem avaliadas é a Khan Academy e a Codecademy. Como ferramenta de avaliação, buscou-se a aplicação de um *checklist* para determinar qual a ferramenta de ensino está em conformidade com os critérios do método de avaliação.

1.1 PROBLEMATIZAÇÃO

Segundo Sirotheau et al. (2011), o esforço empreendido pelo aluno na construção de algoritmos e programas tende a se tornar um obstáculo para si mesmo. Essas questões resultam em situações, tais como: alto índice de repetência, acúmulo de dificuldades que influenciam nos níveis de desistência dos cursos e além de dificuldades demonstradas ao longo das disciplinas dependentes das habilidades de programar. Utilizar a lógica e resolver problemas, de acordo com Sirotheau et al. (2011) sendo que o aluno já passa dificuldades em aprender algoritmos na própria disciplina e não consegue um bom rendimento nas demais

disciplinas que precisam de programação. Essas dificuldades que o aluno passa acarreta o descontentamento e consequentemente a sua desistência do curso.

As plataformas de ensino de programação, vieram à ser uma ferramenta de apoio ao aluno e ao professor no ensino de algoritmos. São ferramentas *online* em que o aluno tem uma *interface* para interagir e resolver os problemas proposto pela plataforma, trazendo elementos lúdicos para complementar o processo de ensino-aprendizagem. O trabalho em questão visa selecionar uma ferramenta, tendo como base os critérios de usabilidade e ergonomia, para auxiliar o aluno no processo de ensino e aprendizagem de algoritmos. Essa ferramenta não visa substituir as aulas e as metodologias aplicadas pelo professor no ensino da matéria. Se as plataformas de ensino são uma boa ferramenta de aprendizagem, então, qual seria a mais adequada para os alunos iniciantes em matérias de algoritmos e programação?

1.1.1 Solução proposta

A Tecnologia da Informação (TI) está auxiliando os alunos no processo de ensino-aprendizagem, em que existe diversos *sites* em que o aluno pode aprender as mais variadas matérias dos mais variados assuntos. Existem tantas ferramentas que auxiliam, é difícil determinar qual seria a melhor ferramenta de apoio no processo de ensino-aprendizagem.

Visando essas plataformas de ensino *online*, o trabalho se propõe a analisar as plataformas de ensino a programação, como ferramenta de apoio para as aulas de algoritmos e programação. Alguns conceitos se tornam abstratos para os alunos, e a utilização de ferramentas interativas no apoio do ensino pode ajudar no entendimento do assunto pelo aluno e na absorção de conceitos referentes as linguagens e estruturas de algoritmos.

Para determinar qual é a plataforma mais adequada, será utilizado um teste de usabilidade, desenvolvido e validado por profissionais da área de ergonomia e usabilidade e aplicado nas ferramentas de apoio ao ensino escolhidas para a aplicação do teste. Com os resultados da avaliação será possível determinar qual é a plataforma que mais indicada para os alunos e iniciantes nas matérias de programação.

1.1.2 Delimitação do escopo

O escopo deste documento é limitado à aplicação e discussão de um teste de usabilidade em plataformas de ensino de programação *online*. Porém, para alcançar os objetivos propostos, visando assegurar os resultados dos testes, foi definido critérios para a

escolha da plataforma e optou-se na utilização de um *checklist* para avaliação, por ser uma ferramenta validada e desenvolvida por profissionais nas áreas de IHC. O teste de usabilidade é aplicado na *interface* das plataformas, visando coletar e compilar os dados para assim obter qual a plataforma que está de acordo com critérios da ferramenta de avaliação. Para aplicação da avaliação nas plataformas, o trabalho avaliou a versão *web*, não avaliando como as plataformas se comportam em versões *mobile*. O trabalho ainda limita-se em aplicar avaliações heurísticas, questionários de satisfação e utilização dos alunos para aplicação do *checklist* e utilização das plataformas para analisar questões pedagógicas.

1.1.3 Justificativa

A disciplina de algoritmos, segundo Frantz e Pontes (2014), é uma das matérias mais importantes na formação dos profissionais na área da Computação. O foco da disciplina é estimular o uso do raciocínio lógico na descoberta, organização das ideias e implementação de soluções para os mais diversos problemas. A resolução de problemas através da criação de programas possibilita o amadurecimento das habilidades de interpretar, organizar o pensamento, produzindo assim ideias factíveis como solução para os problemas analisados, (FRANTZ; PONTES, 2014).

Segundo Raabe e Silva (2005), na perspectiva do aluno a disciplina exige o desenvolvimento de estratégias de solução de problemas com base no uso da lógica e na matemática, sendo que para muitos alunos são altamente abstratas e distantes do cotidiano. Para o professor a disciplina exige uma grande demanda a fim de atender, acompanhar, mediar e avaliar o desempenho do aluno. Essa demanda é dificilmente atendida por diversos fatores em sala de aula, esses fatores podem ser: a grande quantidade de alunos na sala de aula, a curva de aprendizagem de cada aluno, tentar explicar conceitos abstratos para que os alunos consigam entender, sendo que muitas das vezes é a primeira vez que o aluno tem o contato com a disciplina, (RAABE; SILVA, 2005).

De acordo com Amaral (2013), uma pesquisa realizada demonstra que a mente humana retém apenas 10% da informação de material apenas escrito. Com recursos auditivos a retenção fica em 20% daquilo que se ouve e com a aplicação de recursos visuais, a retenção aumenta para 30% daquilo que é visto. Associando recursos multimídia, a mente humana retém 50% do que se vê e escuta. Utilizando a introdução de mecanismos de participação em sala de aula eleva o índice de apreensão da informação por volta de 70% daquilo que se diz e se discute, ou seja, a introdução de elementos lúdicos no processo de ensino-aprendizagem

tem uma grande diferencial que faz com que o aluno absorva mais conhecimento, retém um pouco mais a informação.

Muitos alunos sentem dificuldades em aprender uma matéria que não faz parte do cotidiano deles, pois, na matéria de algoritmos tem que abstrair a solução do problema e transformá-lo em código. Com a dificuldade de tentar assimilar conceitos abstratos da matéria de algoritmos e desenvolver um pensamento lógico, muitos alunos desanimam e acabam desistindo ou reprovando na matéria. Uma maneira de tentar contornar a situação é oferecendo aos alunos alguma ferramenta que os ajudem a entender a lógica de programação por meio de um sistema interativo que auxilie no processo de ensino-aprendizagem, tais como as plataformas de ensino de programação *online*.

As plataformas de aprendizagem *online* são grandes ferramentas no auxílio do processo de ensino-aprendizagem do aluno, pois, maneiras interativas de aprender fazem com que o aluno consiga absorver melhor o conhecimento. Com uma grande oferta de plataformas de aprendizagem, fica difícil apontar uma delas para a utilização dos alunos, e uma das maneiras de estar escolhendo uma das plataformas é a aplicação de um teste de usabilidade. Tendo como base os resultados da aplicação do teste, pode-se indicar a utilização de uma ferramenta de auxílio no processo de ensino-aprendizagem para o aluno, que atenda aos requisitos do teste de usabilidade e com isso garantir que o aluno tenha uma boa experiência de uso.

1.2 OBJETIVOS

Esta seção descreve o objetivo geral, que apresenta a ideia central do trabalho e os específicos, que descrevem as etapas da pesquisa na sequência de execução.

1.2.1 Objetivo geral

Identificar a plataforma de ensino mais adequada quanto a usabilidade, para lógica de programação, para os alunos iniciantes em matérias de programação de computadores.

1.2.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- i. Escolher plataforma de ensino de programação para avaliação;

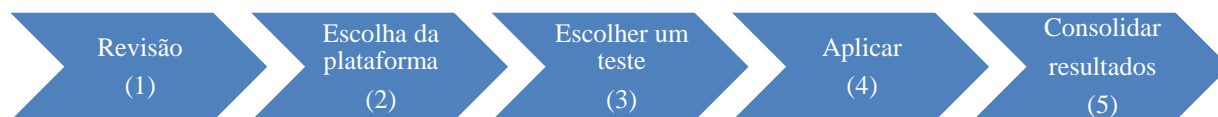
- ii. Escolher e aplicar um teste de usabilidade;
- iii. Coletar dados através do teste de usabilidade;
- iv. Analisar e relatar resultados.

1.3 METODOLOGIA

O presente trabalho será desenvolvido observando as seguintes etapas, conforme Figura 1, onde o mesmo foi baseado após análise das literaturas. Com base no procedimento proposto, segue as etapas específicas do trabalho são:

- 1) Revisão bibliográfica;
- 2) Escolha da plataforma;
- 3) Escolher um teste de usabilidade;
- 4) Aplicar teste de usabilidade;
- 5) Consolidação dos resultados.

Figura 1 – Procedimento do trabalho



Fonte: Desenvolvido pelo autor, 2018.

A revisão bibliográfica, item (1), foi realizada com a pesquisa em livros e em trabalhos acadêmicos, coletando informações que fornecerão suporte para a fundamentação do trabalho e para a escolha do teste de usabilidade para determinar a plataforma de ensino a programação mais adequada quanto a usabilidade aos alunos, tendo como base os critérios avaliativos da ferramenta.

Para a escolha da plataforma, item (2), para a realização do teste foi determinado alguns critérios como: plataformas que trabalham somente *online* e que tenha uma *interface* em que o aluno possa programar diretamente no *site* e que não necessite que o aluno faça qualquer instalação e configuração, a única preocupação do aluno é com a programação, e caso possível, plataformas que tenham o idioma português.

Escolher o teste de usabilidade, item (3), tendo como base a revisão bibliográfica, trabalhos correlatos, será selecionado um teste de usabilidade para testar as plataformas de

ensino *online* de programação, tendo um foco na escolha de *checklist* em que são ferramentas desenvolvidas e validadas por profissionais das áreas de IHC.

Aplicar teste de usabilidade, item (4), após a escolha das plataformas e do teste de usabilidade que vai ser utilizado, o avaliador vai aplicar o teste em forma de *checklist* nas respectivas plataformas. A coleta dos dados será feita através de um formulário, conforme o avaliador vai utilizando a plataforma, vai respondendo o formulário.

Consolidação dos resultados, item (5), com base no teste aplicado nas plataformas e os formulários respondidos pelo avaliador, serão coletados os dados e apresentado qual a plataforma mais adequada quanto a usabilidade para os alunos, tendo como base os critérios propostos na avaliação.

A pesquisa aplicada neste trabalho, é uma pesquisa do tipo quantitativa e descritiva, segundo Fonseca (2002), os resultados da pesquisa pode ser quantificados e seus resultados apresentados em forma de números, podendo gerar gráficos, planilhas para uma melhor análise e discorrer sobre os resultados obtidos. A pesquisa descritiva, exige do avaliador uma série de informações sobre o que se deseja pesquisar, determinando técnicas e ferramentas para obter os resultados. A obtenção desses resultados permite ao avaliador uma descrição dos fenômenos e fatos, gerar relatórios que consigam descrever os dados obtidos e elencar os fatos que levaram a avaliação obter esses resultado, (TRIVIÑOS, 1987).

A seção metodologia tem como objetivo descrever o tipo de pesquisa apresentada neste trabalho, e, elencar os processos para o seu desenvolvimento. A definição dos processos, auxilia e organiza a pesquisa, mantendo uma organização e objetivando alcançar os objetivos geral e específicos definidos neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão abordados conceitos fundamentais para a realização e entendimento deste trabalho. Conceitos estes, referentes às áreas de plataformas de ensino *online*, Interação Humano-Computador e Teste de Usabilidade.

2.1 PROCESSO DE ENSINO-APRENDIZAGEM

O processo de ensino, na visão de Libâneo (2013), é a transmissão da matéria aos alunos e desenvolvimento de atividades, portanto, o ensino tem como função principal assegurar o processo de transmissão e assimilação dos conteúdos do saber escolar, desenvolvendo o pensar e o saber do aluno. Para Freire (2011), não existe ensino sem a aprendizagem, o ensino é o processo de construção de conhecimento de ambas as partes, professor e o aluno, em que ambos, um aprender com o outro, e o professor é o interlocutor deste processo da construção do conhecimento do aluno.

A aprendizagem é um processo de assimilação do conhecimento por meio de atividades, segundo Libâneo (2013), essas atividades são o estudo dos conteúdos das matérias e dos modos de como resolver os problemas propostos pelas atividades. O aluno aprende a resolução de problemas através da aprendizagem. Todo o conteúdo passado para o aluno serve como base de conhecimento para a resolução de problemas ao desenvolvimento do pensamento crítico.

Os conteúdos de ensino, são um conjunto de conhecimentos e habilidades, organizados de forma pedagógica e didática, tendo em vista a assimilação do conhecimento por parte dos alunos. Os conteúdos retratam a experiência social da humanidade, no que se refere a conhecimento e modo de ação, a explicação e resolução dos problemas, os conteúdos são a base de conhecimento, afim de ser assimilada pelos alunos, compreendendo e enfrentando as exigências teóricas e práticas tanto na vida escolar como na social, (LIBÂNEO, 2013).

2.1.1 Processo de ensino-aprendizagem de algoritmos

Segundo Lopes e Garcia (2012), a lógica de programação é a maneira de encadear os pensamentos para atingir determinado objetivo. É a estruturação dos pensamentos em forma de passos, para a resolução de problemas. O aprendizado de lógica de programação, de

acordo com Lopes e Garcia (2012), é necessário para quem deseja trabalhar com desenvolvimento de sistemas e programas.

O termo algoritmos, segundo Lopes e Garcia (2012) e Medina e Fertig (2006), é a sequência de passos finitos com o objetivo de solucionar problemas, é o detalhamento de ações a serem executadas para realizar alguma tarefa. É a organização do pensamento, das ideias, definindo os passos a serem executados para a resolução de problemas ou execução de tarefas.

De acordo com Medina e Ferting (2006), o algoritmo pode ser representado diretamente em linguagem de programação (Java, Python, C e outras), ou a utilização de um pseudocódigo. O pseudocódigo utilizado no Brasil é o Portugol, ou, português estruturado, que utiliza termos do idioma português para representar as funções do algoritmo. Essa forma de desenvolvimento de algoritmo, visa facilitar a criação de programas, sem preocupar-se com o rigor de uma linguagem de programação, pois, utiliza-se de termos no idioma do estudante de algoritmos, (MANAZO; OLIVEIRA, 2012).

O desenvolvimento de algoritmos utilizando o Portugol, faz o uso de termos em português, os termos específicos da linguagem Portugol são chamados de palavras-chaves¹, que representam funções, tipos de dados e procedimentos. Conforma a Figura 2, apresenta um algoritmo para encontrar o maior número de dois números informados pelo usuário.

Figura 2 – Pseudocódigo maior número

```
Algoritmo Maior;
var num1, num2, maior: inteiro;
inicio
    leia(num1, num2);
    se (num1 > num2) então
        maior ← num1;
    senão
        maior ← num2;
    fim se;
    escreva(maior);
fim;
```

Fonte: (MEDINA; FERTING, 2006).

O algoritmo da Figura 2, inicia com o nome do programa e em seguida é feita a definição das variáveis a serem utilizadas. As palavras **var** e **inteiro** estão em negrito por serem palavras exclusivas da linguagem, palavras-chaves, seu uso é específico. A palavra ‘var’

¹ Na informática, palavra-chave é utilizada numa programação para desencadear um processo num computador, são palavras específicas de determinada linguagem de programação, (SIGNIFICADOS, 2018).

é para definir as variáveis e a palavra ‘inteiro’, é utilizado para dizer que as variáveis (num1, num2 e maior) só podem receber número do tipo inteiro. Com a definição e tipo das variáveis é iniciado o programa. O usuário informa dois número, especificado na função ‘leia’ e passa por uma estrutura de decisão ‘se’, para identificar qual é o maior número de dois números informados pelo usuário. Ao final é apresentado a resposta e encerrado o programa com a palavra fim, também em negrito.

O pseudocódigo da Figura 2, apresenta o processo passo a passo para a resolução do problema. Todo esse procedimento é realizado no caderno e não tem a possibilidade do aluno executar o programa e verificar se está correto. Alguns alunos segundo Medina e Ferting (2006), tem uma dificuldade de organizar suas ideias e desenvolver uma pensamento lógico e transcrever para o caderno em forma de pseudocódigo.

2.2 AMBIENTES VIRTUAIS DE APRENDIZAGEM

O Ambiente Virtual de Aprendizagem (AVA), são sistemas computacionais destinados ao suporte de atividades de ensino-aprendizagem, disponibilizados na *internet*. Os AVAs integram múltiplas mídias; linguagens e recursos; apresentam informação de maneira organizada; desenvolvem interações entre pessoas e objetos de conhecimento, tendo em vista atingir determinados objetivos de aprendizagem. As atividades desenvolvidas no AVA, podem ter a mediação de um professor, indicando trabalhos e tarefas, ou sistemas que apresentam as lições para o aluno, (KEMCZINSKI, 2005).

O sistema AVA, proporciona o desenvolvimento e distribuição de conteúdo para diversos para cursos *online*. É desenvolvido para ajudar professores e tutores no gerenciamento de conteúdo e materiais complementares e na gestão de curso *online*. Por meio da ferramenta, é possível acompanhar todo o processo de aprendizagem do aluno, além de gerar relatórios sobre seu desempenho. Outro ponto que os AVAs são utilizados como ferramentas para o gerenciamento de cursos a distância, sendo utilizados para disponibilização de materiais pedagógicos, aplicação de atividades e acompanhamento tanto da disciplina como dos alunos.

A diversos AVA das mais diversas aplicações. Como por exemplo os ambientes Canvas² e Moodle³, que são ferramentas utilizadas para o gerenciamento das aulas, em que o professor disponibiliza os materiais para estudo e mantém uma agenda e uma linha de

² Disponível em: <https://canvas.instructure.com/login/canvas>

³ Disponível em: https://moodle.org/?lang=pt_br

comunicação com os alunos. Outra aplicação dos AVA é para o ensino de idiomas, um exemplo é o *software* Duolingo⁴, que oferece o ensino de idiomas por meio de áudio e atividades diversas. Na questão de AVA de programação *online*, as plataformas Khan Academy e Codecademy, oferecem para o aluno cursos de programação, abordando os principais fundamentos das linguagens e o aluno realiza os exercícios na própria *interface* do portal de aprendizagem.

2.2.1 *Softwares* educativos

O *software* educativo (SE), tem como objetivo de facilitar o processo de ensino aprendizagem, fazendo com que o aluno construa o conhecimento a um dado conteúdo didático. O computador se torna o mediador do processo de ensino do aluno, os SE são vistos como uma ferramenta de apoio ao processo de aprendizagem do aluno, e não como uma única e definida forma de aprendizagem, (MORAIS, 2003).

Segundo Silva et al. (2016), o SE precisa funcionar corretamente e seus elementos pedagógicos precisam estar de acordo com os propósitos dos conteúdos abordados e a deve possuir uma *interface* intuitiva para que o usuário/aluno tenha a melhor experiência possível com a utilização. O uso dos SE, possuem uma boa aceitação em sala de aula, pois, acrescentam elementos lúdicos ao processo de ensino-aprendizagem.

O SE permite ao aluno um ambiente interativo, capaz de estimular o raciocínio e a criatividade, facilitando a compreensão dos conteúdos estudados. São desenvolvidos buscando utilizar recursos que chamem a atenção do aluno, recursos visuais, imagens, sons e vídeos, ao mesmo tempo em que levam uma aprendizagem criativa. A utilização do SE é a capacidade de interação e do tempo de resposta que um *software* pode dar a uma intervenção do aluno, pois, mantém o aluno atento e estimulando a construir o conhecimento.

As plataformas de programação *online*, em sua concepção, são consideradas SE, pois tem como objetivos ensinar o aluno, são ferramentas que oferecem para o aluno a introdução de conceitos de linguagens de programação, introduzindo o aluno a uma metodologia de ensino. É apresenta os conceitos, explicação e inserido os exercícios para o desenvolvimento do raciocínio. O aluno realiza todas as atividades no próprio *site* da empresa, em que avalia se a resposta informada pelo aluno está correta, caso tenha dificuldades na resolução do exercício, auxilio na ajuda da resolução do mesmo. E mantém todo o progresso

⁴ Disponível em: <https://pt.duolingo.com/>

do aluno armazenado, afim de manter uma organização e o aluno ter um controle no seu processo de ensino, podendo escolher as linguagens que deseja aprender e os cursos que já foram concluídos.

2.2.2 Classificação dos *softwares* educativos

No trabalho de Kemczinski (2005) e Gamez (1998), é descrito que os SE são classificados de acordo com seus objetivos pedagógicos, cada qual com suas características e métodos de ensino que é a forma como é passado o conhecimento para o usuário, entre eles temos:

- Exercício e prática;
- Tutorial;
- Sistemas tutores inteligentes;
- Simulação e modelagem;
- Jogos educativos;
- Informativos;
- Hipertexto/hipermídia.

Os SE classificados como exercício e prática, visa a exercitação de um conteúdo ou habilidade já conhecido, ou não, pelo aluno, possui exercícios que envolvem a repetição e memorização. Os tutoriais, constituem numa versão computacional de instrução programada, servem como apoio ou reforço para as aulas.

Os sistema tutores inteligentes, ajustam a estratégia de ensino aprendizagem ao conteúdo e a forma do que se aprende. Os SE simulação e modelagem, simulam a realidade envolvendo situações de risco para a avaliação de tomadas de decisões, como por exemplo simuladores de voos, em que o piloto aprende a tomar decisões nas mais diversas e possíveis situações em um voo.

Os jogos educativos, são fonte de recreação com o objetivo de aquisição de um determinado tipo de aprendizagem, geralmente envolvem desafios ou competições e os informativos, são *softwares* em que é apresentado o material didático para o aluno, são apresentados em forma de texto, gráficos ou tabelas. Na questão de SE hipertexto/hipermídia, é definido como, a forma não linear de armazenamento e recuperação de informações, na qual, a informação pode ser organizada em qualquer ordem, através da seleção de tópicos de interesse.

Segundo Kemczinski (2005), existe uma razão para a classificação dos SE, pois, possibilita uma melhor análise dos aspectos pedagógicos, auxiliar os projetistas de *software* na determinação de características básicas para o desenvolvimento. O SE é um produto orientado a diversas abordagens pedagógicas, aplicado em diferentes estratégias, e a classificação ajuda na escolha de uma ferramenta para auxiliar no processo de ensino.

2.3 ERGONOMIA EM INTERAÇÃO HUMANO-COMPUTADOR

A usabilidade é uma qualidade, de acordo com Cybsi, Betiol e Faust (2010), que caracteriza o uso do sistemas e suas aplicações, sendo estes, fáceis de utilizar, e possuir uma *interface* intuitiva para o usuário. E a ergonomia tem como objetivo, garantir que sistemas estejam adaptados à maneira como o usuário reage com o sistema, proporcionado assim usabilidade para a aplicação ou sistema.

A ergonomia e a usabilidade são dois parâmetros que sempre andam juntos, em que um sistema que tem um alto grau de usabilidade ao mesmo tempo tem um alto grau de ergonomia. Para desenvolver uma *interface* ergonômica e que proporcionem usabilidade, os profissionais que trabalham no desenvolvimento da *interface* do sistema, devem conhecer as pessoas que utilizaram o sistema, e com isso focar o desenvolvimento do sistema com base no público alvo e no trabalho em que o usuário vai realizar, (CYBIS; BETIOL; FAUST, 2010).

“Pode-se dizer que a ergonomia está na origem da usabilidade, pois, ela visa proporcionar a eficácia e eficiência, além do bem-estar e saúde do usuário, por meio da adaptação do trabalho ao homem. Isto significa que seu objetivo é garantir que sistemas e dispositivos estejam adaptados à maneira como o usuário pensa, comporta-se e trabalha e, assim, proporcionar usabilidade. (p. 16 – 17)”

Quando a *interface* e o sistema são desenvolvidos sempre focando no público alvo, aquele que realmente vai utilizar o sistema e como vai ser utilizado no trabalho, a ergonomia visa proporcionar a melhor interação do usuário com o sistema, garantindo assim uma melhor aplicabilidade e ao mesmo tempo sendo um sistema intuitivo e fácil de ser utilizado.

2.3.1 Interação Humano-Computador

A área de IHC é um campo de estudo em que se toma como base diversas matérias da área da computação. Que tem como objetivo entender como e por que as pessoas utilizam, ou não, as tecnologias da informação. A área de IHC se preocupa o projeto e

desenvolvimento de sistemas com propósito de melhorar a eficácia e proporcionar o desenvolvimento de *interfaces* focadas no usuário, visando a usabilidade. É uma área que se preocupa como projetar, implementar e utilizar sistemas computacionais interativos, (SANTA ROSA; MORAES, 2010).

Segundo Barbosa e Silva (2010), além da IHC utilizar disciplinas dentro da área da computação, a IHC utiliza conhecimentos e métodos fora da área da computação para conhecer melhor os fenômenos envolvidos no uso de sistemas computacionais interativos.

De acordo Sociedade Brasileira de Computação, a área de IHC se dedica a estudar os fenômenos de comunicação entre pessoas e sistemas computacionais, envolve áreas como a Ciência da Computação, informação, ciências sociais e comportamentais e envolve todos os aspectos, variáveis e tudo o que está relacionado com a interação entre usuário e sistema, (SBC, 2018).

A área de IHC tem uma forte preocupação com o desenvolvimento, tanto de sistemas como de *interface*. O estudo da interação do usuário com o sistema, implica no desenvolvimento de métodos de avaliação, e recomendações de desenvolvimento de *interfaces* que visam no usuário final, objetivando a experiência de uso em relação com aspectos de ergonomia e usabilidade.

2.3.2 Interface e interação

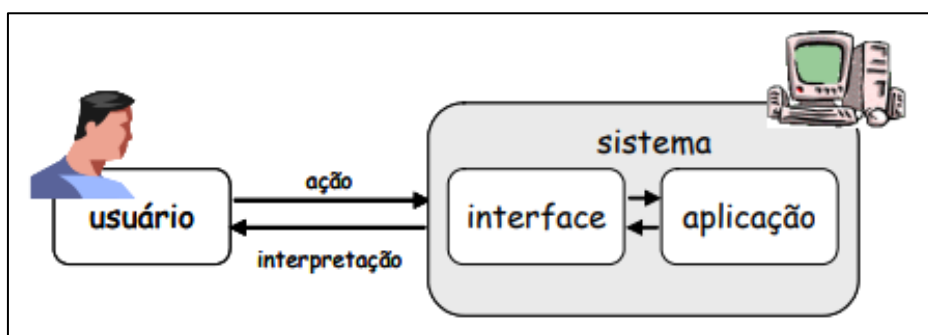
A interação é definida como a maneira como o usuário interage com um produto ou uma aplicação, é o processo de comunicação entre pessoas e sistemas interativos. Neste processo o usuário e o sistema trocam de mensagens, em que um fala e o outro recebe, interpreta e realiza a tarefa. A área de IHC estuda os processos de interação, do ponto de vista do usuário, as ações que realiza usando uma *interface* e as respostas que o usuário recebe do sistema, (PRATES; BARBOSA, 2003); (ROGERS; SHARP; PREECE, 2013).

Na definição de interação, sendo a maneira como uma pessoa interage com o sistema, Rogers, Sharp e Preece (2013), propõem quatro tipos principais de interação que o usuário realiza com o sistema: (i) instrução, (ii) conversação, (iii) manipulação e (iv) exploração. A instrução (i) é definida pelo modo que o usuário emite instruções para um sistema. Já a conversação, define o estilo e forma em que o usuário tem um diálogo com o sistema. A manipulação (iii) que define a maneira em que os usuários interagem com os objetos em um espaço virtual ou físico e por último, a exploração (iv), o modo que os

usuários se movem por um ambiente virtual (sistemas de realidade aumentada) ou um espaço físico.

A *interface* é definido como a porção do sistema com a qual o usuário mantém contato ao utilizá-lo. Ela engloba tanto *software* como *hardware* e considerando a interação do usuário como um processo de comunicação, a *interface* pode ser vista na Figura 3, com o sistema de comunicação, ou, a ponte em que faz a ligação do usuário com a aplicação. Os elementos envolvidos no processo de interação estão relacionados, eles influenciam a forma como os usuários percebem e interagem com a *interface* e seus objetivos, (PRATES; BARBOSA, 2003); (BARBOSA; SILVA, 2010).

Figura 3 – Processo de interação com a *interface*



Fonte: (PRATES; BARBOSA, 2003).

Segundo Prates e Barbosa (2003), as *interfaces* mais comuns envolvem elementos visuais e sonoros, com entradas de dados via teclado e mouse. Outros tipos de entrada de dados são por voz, toque na tela e através de canetas, que estão se tornando frequentes, devido à grande utilização pelos usuários de dispositivos móveis.

A diversidade de *interface* com qual o usuário tem acesso muda a interação do usuário com o sistema, possibilitando diversas abordagens para realizar esta ação. A interação e a *interface* são elementos que são levados em conta no desenvolvimento de uma aplicação, elementos estes, em que seu desenvolvimento focam em questões de usabilidade e ergonomia.

2.4 USABILIDADE

A usabilidade é definida, segundo Nielsen e Loranger (2007), como o atributo de qualidade relacionado a facilidade do uso, ou seja, é a rapidez com que o usuário aprende a usar alguma coisa, a eficiência ao usá-lo e o quanto se lembra de como usar. A palavra usabilidade pode ser compreendida em diversos contextos e produtos, desde o foco em

desenvolvimento de *softwares*, bem o desenvolvimento de produtos, em qualquer âmbito que a palavra usabilidade é utilizada, refere-se com a facilidade de uso.

A usabilidade, tem por definição que os sistemas devem ser fáceis de utilizar, aprender, flexíveis e em termos funcionais-humanos, de ser utilizado com facilidade e eficácia por um segmento de usuários, é isso que a IHC sempre vem buscando, através de estudos e técnicas e a utilização de outras disciplinas para ter como base o projeto, desenvolvimento a aplicação de *interfaces* e sistemas mais amigáveis com o usuário, (BENYON, 2011).

De acordo com Cybis, Betiol e Faust (2010), usabilidade é a qualidade que caracteriza o uso de programas e aplicações de maneira fácil e intuitivo para o usuário. A usabilidade não se caracteriza como uma qualidade intrínseca de um sistema, mas sim, depende de diversos fatores, tendo como principais as características de sua *interface* e as características do usuário ao buscarem determinados objetos ao executarem determinadas tarefas. A usabilidade é um acordo com *interface*, usuário, tarefa e ambiente.

Conforme a norma International Organization for Standardization, (ISO), 9241:11 (1998), a usabilidade é definida quando os usuários, na utilização do sistema, conseguem alcançar seu objetivo, sendo caracterizada pelo nível de eficácia, em que qualquer tipo de usuário consegue desenvolver suas tarefas no sistema sem muitos problemas. A eficiência, a quantidade de recursos que o sistema solicita de seus usuários para a realização das tarefas, recursos esses que seriam tempo, esforço físico e cognitivo. E por último a satisfação, que é quando o usuário consegue realizar suas tarefas de maneira rápida, segura e os resultados são os esperados. Outra característica do sistema é a ergonomia, em que é a qualidade de adaptação de um sistema e seu usuário para a realização de uma determinada tarefa.

No desenvolvimento do sistema focado no usuário, as questões de usabilidade são levadas em conta no desenvolvimento da aplicação. Prover uma experiência de uso significativa para o usuário, desenvolvendo sistemas e *interfaces* intuitivas e fáceis de manipular, é um desafio para os desenvolvedores, sendo que, o sistema deve atender a diferentes tipos de usuários e diferentes tipos de necessidades.

2.4.1 Heurísticas de usabilidade

As heurísticas de usabilidade foram criadas para encontrar problemas de usabilidade em *softwares*, sistemas e *sites* web. São heurísticas utilizadas no desenvolvimento de testes de usabilidade e em avaliação heurística. Segundo Nielsen (1995), as heurísticas foram desenvolvidas com a colaboração de Rolf Molich em 1990, tendo como base a análise

de 249 problemas de usabilidade. Analisando todos esses problemas, foi possível o desenvolvimento das heurísticas, que são:

- a) Visibilidade do estado do sistema: o sistema deve sempre manter os usuários informados sobre o que está acontecendo através do *feedback*;
- b) Correspondência entre o sistema e o mundo real: o sistema deve utilizar palavras, expressões e conceitos que são familiares aos usuários;
- c) Controle e liberdade do usuário: caso o usuário escolha alguma função do sistema por engano, o sistema tem que permitir que o usuário refaça ou desfaça alguma ação dentro do sistema;
- d) Consistência e padronização: os usuários não devem se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. Siga as convenções da plataforma.
- e) Reconhecimento em vez de memorização: minimize a carga de memória do usuário, tornando os objetos, ações e opções visíveis;
- f) Flexibilidade e eficiência de uso: utilizados para tornar a interação do usuário mais rápida e eficiente, permitindo que o sistema sirva igualmente tanto os usuários experientes e inexperientes;
- g) Projeto estético e minimalista: a *interface* não deve conter informações que seja irrelevante ou raramente necessária;
- h) Prevenção de erros: melhor que uma mensagem de erro é um sistema cuidadosa que evita que um problema ocorra, caso possível;
- i) Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros: as mensagens de erro devem ser expressas em linguagens simples (sem códigos do número referente ao erro), indicar precisamente o problema e sugerir uma possível solução;
- j) Ajuda na documentação: o melhor é que o sistema seja utilizado sem documentação, é necessário fornecer a ajuda e documentação de alta qualidade, caso o usuário venha a ter alguma dúvida no decorrer da utilização do sistema.

Todas essas heurísticas desenvolvidas tendo como base diversos problemas de usabilidade, ainda são utilizadas como referencial para o desenvolvimento de testes, questionários e demais formas ou métodos para a análise de *interfaces* e para a realização de testes de usabilidade.

2.4.2 Sistema e o grau de usabilidade

No desenvolvimento da *interface* do sistema, tem que ter em vista os objetivos da aplicação e as necessidades do usuário. As avaliações de usabilidade, são adotadas procurando encontrar problemas de usabilidade no desenvolvimento dos sistemas, (SANTA ROSA; MORAES, 2010). Para atender aos requisitos, e desenvolver aplicações, os sistemas devem atender a o grau de usabilidade, que são tópicos que permeiam questões de qualidade, focadas no usuário.

Segundo Cybis, Betiol e Faust (2010), um sistema com um alto grau de usabilidade terá as seguintes características:

- a) Eficiência no uso do sistema mediante uma quantidade adequada de esforço;
- b) Contém as informações e funções de maneira adequada e apropriada;
- c) Facilidade no processo de aprendizado de utilização e de memorização de como fazer as tarefas no sistema;
- d) Seguro de operar na variedade de contexto em que em que será usado;
- e) Alto grau de utilidade em que fará as coisas que o usuário quer que sejam feitas.

Para alcançar esse grau de usabilidade e conseguir respeitar essas características fundamentais, o *design* do sistema deve ser centrado no humano, no usuário e adotar uma abordagem de *design* na qual a avaliação seja fundamental, (CYBIS; BETIOL; FAUST, 2010).

O desenvolvimento de uma aplicação ou sistema em que foi realizado tendo em vista os graus de usabilidade, torna a experiência do usuário suficiente agradável. O usuário compreende como o sistema funciona interagindo com ele, e a posição dos comandos, o modo de como foi desenvolvida a *interface*, torna o sistema mais intuitivo ao usuário, entendendo assim o fluxo de processo da aplicação.

2.5 TESTE DE USABILIDADE

Os métodos de avaliação de IHC possuem as seguintes atividades básicas, para aplicação e desenvolvimento da avaliação, tais como: preparação, coleta de dados, interpretação, e consolidação dos resultados, (BARBOSA; SILVA, 2010). Independentemente da avaliação a ser utilizada para a análise de um sistema ou *interface*, os passos iniciais são os mesmo de cada teste, criando assim uma metodologia inicial para todos os testes, mudando apenas o teste em si.

O teste de usabilidade tem como objetivo avaliar a qualidade das interações, e verificar os problemas e analisar os impactos negativos sobre as interações do usuário com o sistema. Um teste de usabilidade envolve os usuários reais do sistema ou uma quantidade amostral da população-alvo⁵ que vai utilizar o sistema para realizar tarefas específicas em um contexto de operação real, (CYBIS; BETIOL; FAUST, 2010).

No teste de usabilidade é definido quais critérios devem ser medidos. Os critérios são explorados por perguntas específicas e são associados a algum dado mensurável, que ao final do teste é compilado todas as respostas e com isso chegar a conclusões com base nos resultados. Para realizar as medições é convidado um grupo de usuários para a realização da avaliação, esse grupo segue tarefas determinadas pelo avaliador e é disponibilizado um formulário para o usuário responder as questões pertinentes a tarefa, pois, conforme o progresso do usuário no teste, pode ir respondendo as questões e tornando assim o teste mais dinâmico, isso garante uma maior fidelidade das respostas, (BARBOSA; SILVA, 2010).

2.5.1 Processo de avaliação

Uma das formas bastante utilizadas no processo de avaliação, para a coleta de dados, é a aplicação de questionários, tais questionários podem ser tanto de dados qualitativos e quantitativos. O uso de questionários é uma técnica bem estabelecida para a coleta de dados demográficos e opiniões do usuário. Para o desenvolvimento das perguntas do questionário é necessário um grande esforço e habilidade para assegurar que as perguntas sejam claras e objetivas e que os dados coletados possam ser analisados de forma eficiente, (ROGERS; SHARP; PREECE, 2013).

No desenvolvimento do questionários, segundo Benyon (2011), aplicar o emprego de alguma métrica para avaliar e quantificar as respostas dos usuários em caso de perguntas fechadas. Usar os termos corretos e escolher as afirmações adequadas para extrair informações relevantes a pesquisa é difícil, por isso que a pergunta deve ser bem formulada, sendo objetivas, cuidar com uso de palavras ambíguas e ser mais diretas possíveis.

⁵ Quantidade amostral da população-alvo, é a população de onde provém a amostra. Geralmente é a população que possui as mesmas característica como a amostra, (SIGNIFICADOS, 2018).

2.5.1.1 Design do questionário para coleta de dados

Na hora do desenvolvimento do questionário, Benyon (2011), aponta alguns conselhos para um design eficiente do questionário:

- a) Perguntas específicas são melhores que perguntas genéricas;
- b) Perguntas fechadas são geralmente preferíveis a perguntas abertas;
- c) Considera uma opção sem opinião;
- d) Varie a orientação das escalas de classificação ou intercale com outras perguntas;
- e) Aparência, ordem e instruções são fundamentais;
- f) Acrescente notas de introdução.

As perguntas genéricas tendem a produzir diversas interpretações pelo usuário que está participando do teste. O desenvolvimento de perguntas fechadas ajudam na interpretação da resposta, pois, perguntas abertas são mais difíceis de analisar, só que podem ser úteis quando o avaliador quer saber a opinião do usuário com relação ao sistema em que está sendo testado.

Questões sem opinião, se não houver essa opção a pessoa que está sendo avaliado pode fabricar uma opinião. Uma estratégia é omitir a escolha intermediária e usar uma escala em que o usuário possa responder sobre determinada questão. Variar a orientação das escalas de avaliação ou intercalar com outras perguntas, se um questionário seguir o padrão, isso pode induzir o usuário a responder todas as questões do mesmo lado, inverter algumas escalas ou colocar outro tipo de pergunta entre elas, faz com que o usuário leia as questões e garanta a validade de sua resposta.

No desenvolvimento do questionário, aparência, ordem e instruções são fundamentais, deve parecer fácil de preencher, bastante espaço entre as perguntas e respostas, iniciar com questões fáceis e as questões intermediárias devem cobrir as áreas mais difíceis.

Acrescentar uma introdução no teste, deve explicar o objetivo, garantir a confidencialidade e estimular a resposta. Na parte da conclusão, agradeça as pessoas que participaram do teste e se possível pergunte a opinião deles sobre o teste

Um questionário bem formulado, com perguntas bem claras e objetivas que não devem induzir uma dada resposta e um *layout* do arranjo das perguntas, torna o questionário mais agradável tanto para leitura quanto para resposta. São fatores que influenciam nas respostas, pois, seguindo os conselhos de desenvolvimento do questionário, Benyon (2011), torna-o mais agradável para a pessoas que está fazendo a avaliação e garante uma fidelidade com as respostas.

2.6 MÉTODOS DE AVALIAÇÃO

A avaliação tem por objetivo revisar, experimentar, testar uma ideia de *design*, testar um *software* ou um produto, extrair informações qualitativas e quantitativas, por meio de diversos testes, como usabilidade ou ergonomia e descobrir se o produto ou serviço atende determinados critérios que fazem parte da avaliação, (BENYON, 2011); (SILVA, 1998).

Segundo Barbosa e Silva (2010, p. 290), “a questão fundamental de uma avaliação de IHC é definir quais são os objetivos da avaliação, a quem eles interessam e por quê”. Definindo e os objetivos da avaliação, fica mais fácil decidir, qual a avaliação, o método para a obtenção dos resultados no contexto em que será aplicado.

O processo de avaliação pode ser utilizado durante o desenvolvimento de um *software* ou *interface* para a verificação de possíveis erros e problemas, corrigi-los antes do término do projeto, diminuindo assim os custo do projeto. Em outras situações a avaliação é utilizada para analisar e verificar, qual *software*, produto ou ferramenta se adequa a determinado público alvo.

Cada método de avaliação de *interfaces* possui uma série de propriedades que podem fornecer tanto vantagens quanto desvantagens, por exemplo: o tempo, o esforço e o nível de habilidade e conhecimento para a utilização dos métodos, facilidades e equipamentos necessários para a condução dos mesmos, (JORDAN, 1998).

Na avaliação de *software* ou *interfaces*, existe uma gama de técnicas para a avaliação como é demonstrado na Figura 4, é demonstrado diversas técnicas e qual o meio utilizado para realizar o teste. Na Figura 4, na parte de cima da imagem é apresentado o nome de diversas ferramentas para avaliação e na coluna do lado esquerdo é apresentado o método de avaliação: *checklist*, escalas de avaliação, questionários, modelo conceitual e dentre outros.

Figura 4 – Classificação dos instrumentos avaliativos

	CSEI	MAEP	PROINFO	TICESE	Cronje [1998]	Hanna et al. [1997]	Squires & Preece [1999]	Reeves & Harmon [1996]	ASE	ESEF	PCEM	SEF	SK	MAQSE	MAQSEI	Modelo JIGSAW	IAQSEM	PEDACTICE	CASE	MEMI	MEDA	SASE	SOFTMAT	Total
checklist	•	•	•	•																				4
diretrizes					•	•	•																	3
escala de avaliação								•																1
formulários									•	•	•	•	•	•										5
híbridos														•	•									2
modelo conceitual																•								1
questionários																	•	•						2
sistemas																			•	•	•	•	•	5

Fonte: (GODOI; PADOVI, 2011).

De acordo com Godoi e Padovi (2011), no que se refere ao formato e a técnica dos instrumentos avaliativos de materiais didáticos digitais podem ser classificados em:

- *Checklist*: considerado uma lista de itens que podem apresentar na forma de questões ou ações a serem realizadas;
- Diretrizes: são conjuntos de instruções ou indicações para se levar a termo um plano ou uma ação;
- Escalas de avaliação: um instrumento científico de observação e mensuração dos fenômenos sociais;
- Formulários: uma lista informal, catálogo ou inventário, destinado à coleta de dados resultantes de observações ou interrogações, cujo preenchimento é feito pelo próprio investigador;
- Híbridos: o agrupamento de diferentes técnicas para a identificação de problemas de usabilidade;
- Modelo conceitual: estrutura conceitual geral através da qual a funcionalidade (de um sistema) é apresentada;
- Questionários: instrumento de coleta de dados constituído de uma relação de perguntas que o entrevistado responde, escrevendo ou assinalando respostas;
- Sistemas: programas destinados a auxiliar o usuário no processo de avaliação.

Uma das formas de avaliação proposta neste trabalho é a utilização de *checklist* que facilitam a análise das recomendações ergonômicas durante o processo de avaliação de

usabilidade. Trata-se de um tipo de inspeção de custo relativamente baixo e que poder ser adaptado às diversas situações de avaliação e que pode ser aplicado sem a necessidade de um especialista de usabilidade.

2.6.1 Avaliação com *checklist*

O *checklist* é uma ferramenta de avaliação ergonômica e usabilidade de *software*, em que é verificado se o sistema está de acordo com as especificações e recomendações dos atributos da ferramenta. A técnica de avaliação é desenvolvida com os conhecimentos ergonômicos provenientes de pesquisas, de profissionais da área. Depois a técnica é validada por profissionais com conhecimento na área, Gamez (1998).

A inspeção de ergonomia e usabilidade por meio de listas de verificação (*checklist*) permite que seja identificado problemas das *interfaces*. Um *checklist* bem construindo, conteúdo organizado e pertinente para as avaliações podem levar a produção de resultados uniformes e abrangentes, (CYBIS; BETIOL; FAUST, 2010).

Segundo Cybis, Betiol e Faust (2010), a avaliação realizada por meio de *checklist* pode apresentar as seguintes vantagens:

- Fornecer conhecimento ergonômico sobre os aspectos a avaliar;
- Sistematizar as avaliações em se tratando de qualidades a inspecionar;
- Sistematiza as avaliações em se tratando de abrangência de componentes a inspecionar;
- Reduzir a subjetividade normalmente associada a processo de avaliação;
- Reduzir os custos da avaliação, pois, é um método que não demanda pessoal especializado.

Segundo Jordan (1998), a ferramenta de inspeção *checklist* apresenta uma série de fundamentos baseados nas teorias do *design*, da ergonomia e que asseguram uma *interface* fácil de utilizar. Tendo como base esses fundamentos, a ferramenta indica questões de usabilidade que uma *interface* deve ter, como consistência, compatibilidade, padrões, bom retorno para os usuários. Questões de nível menos elevado, como o tamanho do caracteres apresentado na tela, a utilização de cores e nomenclaturas dos botões. A ideia do *checklist* é que o avaliador analise o sistema e descubra se o mesmo se encontra de acordo com os atributos da técnica de avaliação.

Para Cybis, Betiol e Faust (2010), a qualidade do *checklist* utilizado implica em muito nos resultados obtidos da avaliação. Listas mal elaboradas que apresentam questões subjetivas e em número insuficiente, podem levar a produção de dados duvidosos. Na utilização dos *checklist* o avaliador tem a possibilidade de configurar a ferramenta conforma o *software*, possibilitando assim que questões que não são aplicáveis ao contexto influenciem no resultado final do teste avaliativo da plataforma.

Existe uma variedade de *checklist* disponíveis para avaliação de SE, segundo Silva et al. (2016), podem ser citados:

- Método de Reeves: consistem em 24 critérios, sendo 10 relacionados a *interface* e 14 em aspectos pedagógicos. Cada critério é associado a uma escala bidirecional não numérica, que se utiliza de conceitos antagônicos⁶;
- Técnica de Mucchielli: propõe 10 critérios associados à escala numérica com pontuação variando entre 1 (um) e 5 (cinco);
- Avaliação de Lori: *checklist* com 9 critérios (1 pedagógico, 4 de interface e 4 de qualidade de *software*). Os critérios estão associados a escalas numéricas de 5 pontos e a obtenção dos resultados se assemelha a técnica de Mucchielli;
- Técnica TICESE: possui 3 módulos, classificação, de avaliação do SE e contextual. A técnica pode ser configurado atribuindo pesos aos critérios e se adequando ao SE em questão.

Para Silva et al. (2016), a variedade de abordagens disponíveis para avaliar SEs reflete em diferenças significativas em relação ao número de critérios considerados. A natureza e construção dos critérios (pedagógicos, qualidade de *software* e de uso, quanto a escala e o método de obtenção e análise dos dados e a qualidade dos resultados alcançados. São fatores determinantes para a escolha do método de avaliação.

A facilidade do uso da ferramenta de avaliação, depende de como foi formulado os atributos, e para a qualidade dos resultados depende mais da qualidade dos critérios e subcritérios que compõem a ferramenta de avaliação do que do próprio avaliador em si. O uso de uma *checklist* cuidadosamente elaborado, testado e validado por profissionais, pode facilitar o processo de avaliação e se tornando uma forma mais econômica de avaliação, pois, não necessariamente necessita de um especialista na área de usabilidade.

⁶ Antagônico é uma expressão que se refere a tudo aquilo que apresenta antagonismo, ou seja, tudo que se apresenta de forma contrária, que faz oposição entre duas noções, dois conceitos, duas ideias, (SIGNIFICADOS, 2018).

2.6.2 Checklist TICESE

O *checklist* TICESE⁷ (Técnica de Inspeção de Conformidade Ergonômica de *Software* Educacional) é uma técnica desenvolvida por Gamez (1998) no Laboratório de Utilizabilidade (LabiUtil⁸) em Florianópolis e tem um enfoque em apoiar os processo de avaliação de SE. Segundo Gamez (1998), a ferramenta propõe a integração entre aspectos pedagógicos e de usabilidade no processo de avaliação ergonômica.

De acordo com Andres e Cybis (2000), a técnica é formado por um conjunto específico de critérios de análise e tem seu suporte teórico nas ciências cognitivas, ergonomia de *software*, psicologia de aprendizagem e pedagogia. Os critérios estão associados a um conjunto de atributos que visa orientar o avaliador na tarefa de inspecionar um SE. Os objetivos da TICESE são:

- Fornece diretrizes para avaliadores na tarefa de avaliar qualidade em SE;
- Orientar na realização de inspeção de conformidade ergonômica em *software*;
- Enfoque sobre a ergonomia aplicado em produtos SE;
- Considerados tanto os aspectos pedagógicos como os aspectos referentes à *interface* e usabilidade.

Segundo Gamez (1998), a ferramenta oferece ao avaliador identificar produtos educacionais informatizados com qualidade, fornecer parâmetros de orientação para auxiliar na tomada de decisão sobre a aquisição de um SE e identificar e propor soluções para os possíveis problema que o SE apresentar no decorrer da avaliação. A ferramenta desenvolvida é dividida em três módulos, de classificação, módulo de avaliação e módulo de avaliação contextual.

O módulo de classificação, Anexo A, não tem a intenção de avaliar, e sim, de classificar o *software* a partir de seus atributos e propostas pedagógicas. O processo de avaliação deve considerar as diferenças entre vários tipos de programas e que uma taxonomia⁹ pode ajudar a identificar algumas categorias de produtos, tais como: exercício e prática, simuladores, tutorial, tutorial inteligente, jogos educativos, hipertexto/hipermídia e informativos. Cada um desses programa possuem características próprias e devem ser avaliados de acordo com a sua classificação.

⁷ Disponível em: <http://www.labiutil.inf.ufsc.br/estilo/Ticese.htm>

⁸ Disponível em: <http://www.labiutil.inf.ufsc.br/>

⁹ Taxonomia é o estudo científico responsável por determinar a classificação sistemática de diferentes em coisas em diferentes categorias, (SIGNIFICADOS, 2018).

No módulo de avaliação, Anexo B, é o principal módulo da técnica. É o módulo relacionado com a avaliação da conformidade com os padrões ergonômicos de qualidade do SE. O objetivo é avaliar até que ponto o *software* auxilia ou não no processo de aprendizagem. A avaliação captura o fato, caso o *software* não for intuitivo, o aluno terá que realizar um esforço desnecessário para aprender a manipular o *software*, atrasando ou dificultando a aprendizagem de seu conteúdo.

O módulo de avaliação contextual, Anexo C, refere-se a verificação da adequabilidade do *software* em um dado contexto pedagógico ou situação específica. Mesmo o *software* corresponda às exigências ergonômicas, a sua utilização dependerá do contexto pedagógico na qual a instituição está inserida.

O *checklist* TICESE foi elaborado coletando uma série de informações por meio de consultas bibliográficas (heurísticas, ergonomia e pedagogias), opinião de especialistas de usabilidade, pedagogia e psicologia. A TICESE foi validada por pessoas com conhecimentos e práticas de avaliação de *software* que aplicaram e deram as suas opiniões. Com os resultados deste processo, a ferramenta foi melhorada e adaptada para proporcionar melhores facilidades na sua aplicação, (GAMEZ, 1998).

2.6.2.1 Instruções para aplicação do *checklist* TICESE

Para a aplicação do *checklist* TICESE, Gamez (1998), sugere que o avaliador tenha um primeiro contato com o *software* assim de compreender o seu funcionamento. E conseqüentemente, ao avaliador, faça uma leitura da técnica para conhecer os critérios e subcritérios que compõem a ferramenta de avaliação.

Após a leitura da técnica, o avaliador deve atribuir pesos aos critérios avaliativos do *checklist*. Implica em ler cada questão e verificar a existência do atributo no *software*. Conforme Quadro 1, o avaliador deve informar o peso do atributo. Para a questão com resposta não se aplica, atribuir peso 0 (zero), caso o atributo seja muito relevante, ou, muito importante aplica-se o peso 1,5 (um e meio) e para questões importantes 1 (um). Esse procedimento é para determinar quais os atributos são aplicáveis ao contexto da plataforma. Os atributos que receberam peso 0 (zero), não vão influenciar no resultado final do *checklist*.

Quadro 1 – Identificação dos pesos

Questão	Peso
Não se aplica	0
Muito importante	1,5
Importante	1

Fonte: Adaptado de Gamez (1998).

Com a preparação do processo de atribuição de pesos concluídos, o avaliador irá responder as questões, atribuindo valores as mesmas. Conforme Quadro 2, na avaliação do atributo, se a plataforma responder satisfatoriamente com o requisito, o avaliador atribui a uma nota ‘sim’ que representa valor 1 (um). Se a plataforma responder parcialmente ao atributo é atribuída a nota 0,5 (meio), e se a plataforma não atende ao critério é atribuído a nota 0 (zero). Critérios que tem um peso de avaliação e recebem como nota o valor 0 (zero), tem uma grande influência no resultado final do *checklist*.

Quadro 2 – Identificação das notas

Resposta	Valor
Sim	1
Parcialmente	0,5
Não	0

Fonte: Adaptado de Gamez (1998).

Para encontrar o valor de cada critério e subcritério é aplicado uma equação (1), em que é realizado o somatório da multiplicação do peso pela valor da nota do critério, dividido pelo somatório de todos os pesos das critérios e multiplicado por 100 (cem), para obter o valor em porcentagem. Onde, o j é o critério, q_j é o número da questão por critérios, o $a(i)$ é o valor da questão e o $p(i)$ é o peso atribuído à questão. O resultado indica o percentual de conformidade ergonômica do *software* ao critério de avaliação. A conformidade ergonômica indica o quanto a plataforma está de acordo com os critérios de avaliação do *checklist* TICESE.

$$X_{(j)} = \frac{\sum_{(i=1)}^{q(j)} \frac{a(i) * p(i)}{\sum p(i) q(j)}} \quad (1)$$

Com os resultados, o avaliador pode identificar problemas de ergonomia e usabilidade em SE, desenvolver relatórios e apresentar os problemas identificados para possíveis correções. A possibilidade também de utilizar a técnica como método comparativo

entra demais plataformas e assim analisar e indicar a plataforma que mais se adequa aos critérios do *checklist* para questões de escolha de SE, (GAMEZ; 1998).

3 TRABALHOS CORRELATOS

Nesta seção são abordados trabalhos correlacionados ao objeto de estudo desta monografia. Foi realizado uma pesquisa com o intuito de identificar trabalhos correlatos para a avaliação de ferramentas que auxiliem os alunos no processo de ensino-aprendizagem na matéria de algoritmos. Além disto, a pesquisa buscou também os métodos que os autores dos trabalhos utilizaram para a avaliação das ferramentas, quais plataformas foram avaliadas e o resultado final da avaliação.

Desta forma, nas seções subsequentes são apresentados trabalhos em que os autores utilizaram técnicas da área de IHC, para realizar a avaliação quanto a usabilidade nas plataformas.

Nesta etapa do trabalho, foram selecionados quatro trabalhos correlatos, que serão elucidados a seguir.

3.1 AVALIAÇÃO DA USABILIDADE DA *INTERFACE* DO AMBIENTE VIRTUAL MOODLE

No trabalho de Antonino e Freire (2018), tem como objetivo analisar a usabilidade da *interface* do Moodle que é um Ambiente Virtual de Aprendizagem (AVA), sob a perspectiva dos alunos do curso de Licenciatura em Computação da Universidade Estadual do Ceará. A usabilidade é um atributo desejável em um AVA, pois seu foco é a interação com o usuário devido aos fins educacionais. Consequentemente, um AVA deve atender as necessidades da usabilidade, pois, se for um sistema muito difícil e complicado de utilizar, pode fazer com que o aluno se sinta desmotivado e consequentemente desistindo de utilizá-lo.

Para a realização da pesquisa foi aplicado um questionário da ISONORM para uma amostra de alunos do curso para medir o nível de satisfação dos usuários em relação a aspectos de usabilidade do sistema. Foi realizado uma busca na bibliografia os questionários que estivessem entre os mais utilizados. Optou-se pelo ISONORM 9241/10, pois, é de domínio público, pode ser usado gratuitamente e por qualquer pesquisador, encontra-se traduzido para a língua portuguesa, é sucinto e de fácil entendimento, sendo bastante utilizado ou referenciado nos estudos sobre usabilidade.

A estrutura do questionário foi dividida em sete seções seguindo os princípios do diálogo da parte 10 da norma ISO 9241:10, para cada seção foi definido um limite de 5 perguntas totalizando ao todo 35 questões. O questionário foi constituído de perguntas abertas

a respeito do usuário e sua experiência com computadores e o *software* utilizado e nas demais perguntas foi utilizado uma escala com dois extremos, positivo e negativo. Os valores para as respostas variam entre ‘---’, ‘--’, ‘-’, ‘+/-’, ‘++’, ‘+’ e ‘+++’, em um intervalo entre 1 e 7 cabendo ao avaliador marcar um ‘X’ na opção desejada.

Como resultado final, analisando os dados obtidos com o questionário, o Moodle teve uma avaliação positiva e que os usuários estão satisfeitos com a qualidade de interação por ela proporcionada e que a utilização do questionário tendo como base a norma ISO 9241:10 se demonstrou muito adequado ao contexto em que ela foi aplicada, trazendo resultados sucintos e ao mesmo tempo mostrando os principais pontos em que a plataforma pode melhorar, (ANTONINO; FREIRE, 2018).

3.2 AVALIAÇÃO DE OBJETOS DE APRENDIZAGEM

O trabalho proposto por Mussoi, Flores e Behar (2010) é avaliar um Objeto de Aprendizagem (OA). Para realizar uma análise de forma sistemática, devem-se identificar critérios padrão, construir formulários de validação dos critérios e criar escalas de avaliação. Dentro de uma concepção epistemológica os OAs reforçam, principalmente, o comportamentalismo ou o construtivismo.

Para avaliar um OA, compreendendo como material pedagógico deve-se ter bem claro a sua concepção epistemológica, as características e os objetivos. Cada concepção pedagógica engloba diferentes paradigmas de aprendizagem, que permitem identificar sua concepção e uso, podendo ser de modo comportamentalista ou construtivista, (MUSSOI; FLORES; BEHAR, 2010).

A avaliação de um OA é muito importante, pois, são matérias pedagógicos que auxiliam no processo de construção do conhecimento. Para fazer a avaliação, é considerado alguns parâmetros como: concepção epistemológica; qualidade do conteúdo; adequação do conteúdo ao público alvo/faixa etária; definição de objetivos a serem alcançados; forma de feedback ao usuário; motivação; forma de apresentação (*layout*, usabilidade); re-usabilidade, que mostra o nível em que o OA pode ser reutilizado em diferentes contextos de aplicação.

No trabalho de Mussoi, Flores e Behar (2010), foi analisado uma ficha de avaliação baseada em um estudo dos trabalhos de Center for Curriculum Transfer and Technology, dividido em três categoria, características gerais, ferramentas de apoio ao professor e de apoio ao aluno. Outro questionário analisado foi o Banco Internacional de Objetos Educacionais (BIOE). A partir das análises do questionário foi montado uma ficha

para avaliação de um OA, a qual pode ser usada somente com os aspectos pedagógicos e técnicos, ou se o OA for maior ela pode ser usada para avaliar além desses aspectos, as ferramentas de apoio ao professor e ao aluno.

Dentro das características gerais foram abordados tópicos sobre o trabalho educacional e a usabilidade. O questionário recebeu pontos para avaliação de 1 a 5, constando também a opção sem opinião. O questionário desenvolvido tendo como base outro estudo é utilizado para avaliar qualquer OA, sendo que essas plataformas compartilham do mesmo objetivo, construir e auxiliar no desenvolvimento do processo de ensino-aprendizagem do aluno como ferramenta de apoio, tanto para o aluno como para o professor.

3.3 USABILIDADE E OS CRITÉRIOS DE AVALIAÇÃO PARA AMBIENTE VIRTUAL DE ENSINO-APRENDIZAGEM DO CURSO LETRASLIBRAS

No trabalho de Ribas e Pereira (2011), foca na avaliação da *interface* do curso de Letras LIBRAS da Universidade Federal de Santa Catarina implementado na modalidade de curso a distância. O curso tem o enfoque em formar profissionais qualificados para atuar no ensino da língua de sinais como primeira e segunda língua. Esse curso é pioneiro no Brasil na modalidade de ensino a distância e tem como público-alvo: surdos, instrutores de LIBRAS e ouvintes fluentes na língua de sinais que tenham concluído o ensino médio.

Para o curso foi desenvolvido um AVA para atender a demanda de seu público-alvo e seus objetivos. Uma análise da usabilidade do ambiente *online* torna-se necessário para um controle da qualidade do sistema e sempre visando a melhoria continua da plataforma.

A avaliação de usabilidade foi dividida em dois métodos, o primeiro método foi a realização de um questionário exploratório que foi aplicado aos alunos que utilizam o *site*. O questionário foi criado com base na cultura surda, nas teorias de Bastien & Scappin e Schneiderman. O questionário foi formado com quarenta questões de múltipla escolha, cada questão possui uma única resposta. O segundo método foi a avaliação heurística que consistiu na análise de usabilidade utilizando um *checklist* originalmente pela equipe do Labutil da Universidade Federal de Santa Catarina, (RIBAS; PEREIRA, 2011).

Com as respostas dos questionários e dos *checklist* foi possível a elaboração de um relatório pontuando pontos falhos e sugestões. Na análise das respostas, verificou-se que a usabilidade do *site* apresentou falhas que podem ser corrigidas sem muita dificuldade. Este teste teve como principal objetivo apontar os problemas de usabilidade e a satisfação com o uso da plataforma com relação aos usuários.

3.4 AVALIAÇÃO DE *SOFTWARES* EDUCATIVOS PARA O ENSINO DE PROGRAMAÇÃO

O trabalho proposto por Silva, de Pontes e Padilha (2016), apresenta a avaliação de quatro plataformas para o ensino de programação: Code.org¹⁰, Scratch¹¹, RoboMind¹² e LightBot¹³. Por serem plataformas livres que oferecem fácil manuseio e atividades relacionadas a jogos, o que incentiva o aprendizado da programação. Utilizando o método de Reeves, com o intuito de subsidiar a escolha desses recursos pelos professores. Para cada *software* avaliado, é mostrado o quanto cada critério (pedagógico e de *interface*) se aplica.

O método de Reeves consiste em 24 critérios, sendo 10 relacionados à *interface* e 14 pedagógicos. Para cada critério, há uma escala representada por uma seta dupla em que deve ser realizada uma marcação do quanto este critério se aplica ao *software*. Na extremidade esquerda da seta fica situado o conceito mais negativo e à direita o mais positivo. Os conceitos variam de acordo com cada critério avaliado. Como o método de Reeves contempla os aspectos pedagógicos e também os de usabilidade do *software* e se apresenta como um *checklist* simplificado, foi o método escolhido para avaliação das plataformas.

Na análise dos resultados, que a maioria dos critérios relacionados com *interface*, obteve conceitos mais positivos. A plataforma LightBot apresentou melhor avaliação em termos de *interface* se comparados com as demais plataformas. Considerando os critérios pedagógicos, as avaliações dos quatro *softwares* tiveram um comportamento similar na determinação dos conceitos em que o aluno vai aprender ao utilizar as plataformas.

O uso destas plataformas de ensino de programação tende a levar um desenvolvimento de habilidades para resolução de problemas devido à necessidade de estruturar uma sequência de ações dado um problema. Como resultado final, das quatro plataformas avaliadas utilizando o método de Reeves, LightBot e Scratch foram considerados com maior potencial para o aprendizado de programação devido à facilidade de uso e diversidade de conteúdo, (SILVA; DE PONTES; PADILHA, 2016).

¹⁰ Disponível em: <https://code.org/>

¹¹ Disponível em: <https://scratch.mit.edu/>

¹² Disponível em: <https://www.robomind.com.br/>

¹³ Disponível em: <http://lightbot.com/hour-of-code.html>

3.5 COMPARAÇÃO ENTRE TRABALHOS CORRELATOS

Os trabalhos correlatos apresentados foram dispostos num quadro para mostrar os autores dos trabalhos, as plataformas avaliadas e qual foi o método de avaliação utilizado, conforme Quadro 3. Cada trabalho teve um enfoque em alguma plataforma de ensino, sendo que no trabalho de Mussoi, Flores e Behar (2010), não foi avaliado uma plataforma de ensino em específico, mas sim, foi feita a avaliação de objetos de aprendizagem que seriam os módulos que compõem uma plataforma de ensino.

Quadro 3 – Trabalhos correlatos

Referências	Plataforma Avaliada	Checklist Utilizado
Antonino e Freire (2018)	Moodle	ISO 9241:10
Mussoi, Flores e Behar (2010)	Objetos de aprendizagem Não especifica plataforma	Center for Curriculum Transfer and Technology Banco Internacional de Objetos Educacionais
Ribas e Pereira (2011)	LetrasLIBRAS	Método exploratório Avaliação heurística
Silva, de Pontes e Padilha (2016)	Code.org Scratch RoboMind LightBot	Método de Reeves

Fonte: Desenvolvido pelo autor, 2018.

No trabalho de Antonino e Freire (2018), o foco foi criar uma avaliação da plataforma Moodle, que é uma plataforma para o gerenciamento de ensino, na qual, o professor disponibiliza os matérias, exercícios e o aluno tem o acesso a esses conteúdo diretamente na plataforma. A teste utilizado para fazer a avaliação da plataforma foi a utilização da norma ISO 9241:10 que é uma norma que se preocupa mais com a satisfação do usuário e pouco com relação a critérios de usabilidade.

Na avaliação da plataforma LetrasLIBRAS, pelas autores do trabalho Ribas e Pereira (2011), o teste teve como foco tentar encontrar problemas de usabilidade por dois métodos (exploratório e avaliação heurística). Como já era uma plataforma já utilizada pelos alunos o teste apontou os problemas de usabilidade e foi criado um relatório para as possíveis melhorias do *site* e com isso a possibilidade de manter a qualidade do portal em que os alunos utilizam para estudar.

O teste apresentado por Silva, de Pontes e Padilha (2016), teve como principal objetivo definir qual a plataforma de ensino de programação o professor possa indicar para os

alunos dentre as quatro plataforma escolhidas para o estudo: Code.org, Scratch, RoboMind, LightBot. O teste foi aplicado utilizando o método de Reeves em que a avaliação é feita com base na *interface* do sistema e na parte pedagógica, as resposta são dadas através de pontos, que na ligação desses pontos gera um gráfico e nessa análise é decidido qual seria a plataforma, podendo ser mais de uma, mais adequada para estar indicando aos alunos.

Os trabalhos correlatos apresentados nesta seção, apresentaram a aplicação de avaliação em diversas plataformas, tanto plataformas de ensino de programação, como em Ambientes Virtuais de Aprendizado, mostrando assim a importância dessas plataformas de ensino e como podem estar auxiliando os alunos no processo de ensino-aprendizagem. A utilização dessas avaliações servem para selecionar, identificar ou indicar qual a melhor plataforma de ensino para os usuários.

O trabalho proposto nesta monografia apresenta a aplicação de um teste de usabilidade em formato de *checklist*, são ferramentas desenvolvidas, testadas e validadas por profissionais das área de usabilidade e ergonomia e não precisam ser aplicadas por profissionais de IHC, tendo como mesmo objetivo dos trabalhos apresentados, a análise e verificação da plataforma que mais se adequa quanto a usabilidade para alunos iniciantes em matérias de algoritmos e programação.

4 ESPECIFICAÇÕES FORMAIS

Nesta seção será apresentado os requisitos para a escolha do SE, requisitos esses, definidos pelo avaliador, foram considerados utilização dos alunos, tanto dentro do ambiente acadêmico e em sua residência. Os diagramas para a aplicação do teste de usabilidade, evidenciando os passos que o avaliador deve executar para a avaliação das plataformas em ensino de programação *online*.

Para a apresentação dos diagramas foi utilizado a UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), que é uma linguagem para especificação, construção, visualização e documentação de artefatos. De acordo com Lima (2011), também é empregada para demonstrar os passos e os procedimentos de uma determinada tarefa. A UML é extensível e independente de processos ou linguagens de programação, o que garante a vantagem para o desenvolvedor adotar qualquer processo, metodologia e conseguir expressá-las claramente o fluxo do processo e seus objetivos, (LIMA, 2011).

4.1 CRITÉRIOS PARA A ESCOLHA DAS PLATAFORMAS

O primeiro passo para a aplicação do teste de usabilidade é a escolha das plataformas de programação. Para realizar esse procedimento, foram definidos alguns critérios para a escolha, critérios esses definidos pelo avaliador e pensando no fato do aluno poder utilizar em casa e não somente no ambiente acadêmico. Os critérios de escolha são:

- 1) Ser totalmente *online*;
- 2) Ser uma plataforma gratuita para a utilização;
- 3) Possuir *interface* para programar;
- 4) Ter conteúdo em português (caso possível);
- 5) Apresentar conteúdo explicativo da linguagem;
- 6) *Interface* que apresente todos módulos da plataforma;
- 7) Apresenta o progresso do aluno.

Um dos pontos em ser uma plataforma totalmente *online*, item 1, é o fato de o aluno não precisar realizar nenhuma instalação de nenhuma IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) para programação ou precisar realizar qualquer outra configuração para começar a programar. Com a plataforma sendo totalmente *online*, a preocupação do aluno é aprender o conteúdo e a sintaxe e a semântica da

linguagem em que está sendo aprendida, esses pontos facilitam o aluno e ao mesmo tempo que a plataforma possui uma *interface* para programar, item 3.

A plataforma deve ser gratuita, item 2, todo o conteúdo e atividades realizados pelo aluno devem ser de graça e caso possível possuir ter conteúdo em português, item 4, sendo que alguns alunos não tem conhecimento da língua inglesa. Se a plataforma possuir somente a língua inglesa, o aluno tem o auxílio de diversas ferramentas para realizar a tradução e o entendimento do conteúdo.

Apresentar o conteúdo explicativo da linguagem, item 5, em que o aluno está aprendendo, bem como os comandos e dando exemplos de como digitar os códigos da linguagem. O aluno tem que ter uma visão de um todo da plataforma, item 6, mostrando todas as tarefas e os exercícios vai aprender e apresentar o progresso, item 7, mostrando os módulos que já foram concluídos e em qual parte da atividade o aluno parou, podendo assim dar continuidade no ponto em que tinha parado para a conclusão da tarefa.

Para encontrar as plataformas foi realizado algumas pesquisas em portais de buscas e verificado quais as ferramentas de ensino de programação que mais apareciam nas pesquisas. Para a escolha do *software* educativo foi levado em consideração que as mesmas trabalham de forma igual, pois, na pesquisa foi encontrado plataformas que trabalham com jogos, em que o aluno faz parte de uma história e cria seu personagem e vai informando as linhas de código para o seu personagem realizar as ações. Outras plataformas encontradas, são sistemas em que possuem vídeos e os matérias e o aluno precisa instalar e configurar o ambiente de programação e realizar as tarefas.

As plataformas encontradas foram a Khan Academy e a Codecademy, que mais atendem os critérios de escolha. Ambas são gratuitas e são totalmente *online*, e a diferença encontrada entre elas é que a Codecademy, tem o seu conteúdo em inglês, na explicação dos exercícios.

4.2 DIAGRAMA DE ATIVIDADE

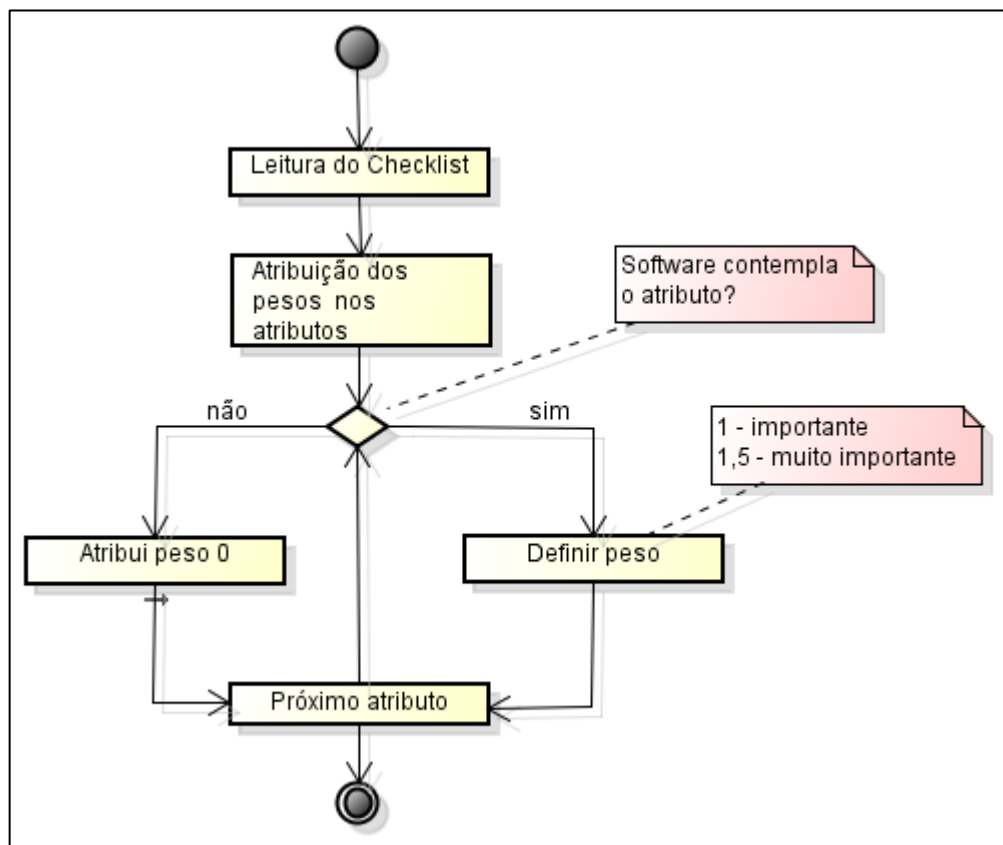
O diagrama de atividade permite modelar o fluxo de atividade, denotando os caminhos que um processo deve seguir. É um modelo gráfico que analisar a estrutura e os caminhos de um determinado processo e os eventos que fazem parte do processo. Cada evento possui precondições para sua ocorrência e pós-condições dela decorrentes, (LIMA, 2011).

O diagrama da de preparação teste, conforme Figura 5, demonstra o procedimento realizado para a preparação do teste. Os passos realizados para a preparação do *checklist*

TICESE. O primeiro passo é a leitura, para o avaliador ter conhecimento da ferramenta avaliativa, depois da leitura, o avaliador precisa realizar a atribuição dos pesos. O avaliador precisa ler e analisar cada atributo para a definição do peso.

Ao ler um atributo, o avaliador é apresentado a uma pergunta: “*Software* contempla o atributo, caso não o avaliador atribui peso 0 (zero), e no cálculo da nota, esse atributo que recebeu peso 0 não afetará na compilação dos resultados. Se o atributo se aplica, é definido por uma ordem de importância, sendo 1 (um) para “importante” e 1,5 (um e meio) para muito importante, depois da definição do peso o avaliador passa para o próximo atributo. Demonstrado na Figura 5, esse processo é cíclico e o término é quando o avaliador termina toda a análise dos atributos da ferramenta avaliativa.

Figura 5 – Diagrama de preparação do teste

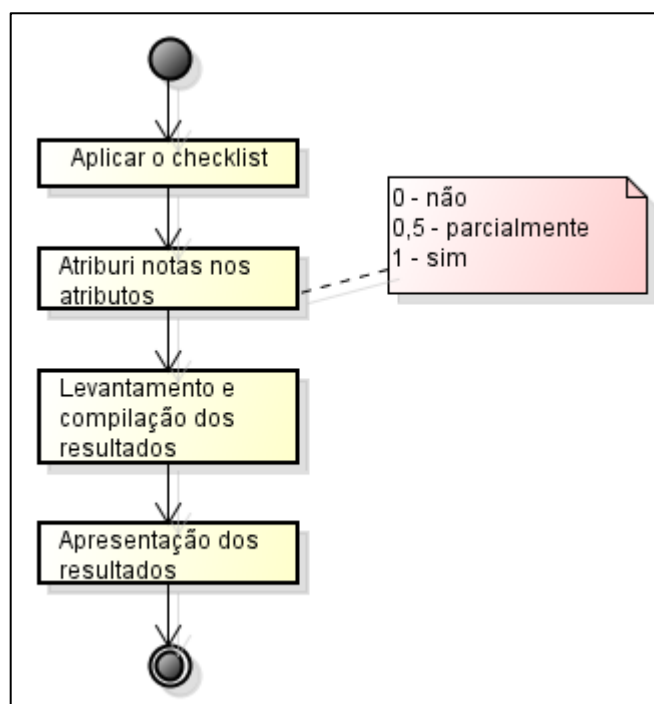


Fonte: Desenvolvido pelo autor, 2018.

Esse processo de leitura de cada questão e definição dos pesos é realizada apenas no segundo e terceiro módulo. O primeiro módulo não necessita deste processo, pois, não interfere na análise dos resultados, sendo apenas um módulo para classificação das plataformas de programação *online*.

A aplicação do teste nas plataformas de ensino de programação é demonstrado pela Figura 6, em que é mostrado os passos para a realização do teste. Como primeiro passo para a aplicação do *checklist* o avaliador retoma o uso de um dos SE e inicia o teste. O avaliado lê um atributo, verifica o comportamento do sistema conforme o atributo e atribui um determinado peso. Caso atende satisfatoriamente, é dado nota 1 (um), se a plataforma não atende totalmente o atributo e o resultado é médio, é dado uma nota 0,5 (meio). E caso o SE não atende o atributo, é dado uma nota 0 (zero). Cada critério do *checklist* e de seus subcritérios, são avaliados individualmente, na busca de uma melhor avaliação e na lisura dos resultados obtidos através do teste.

Figura 6 – Diagrama de aplicação do teste



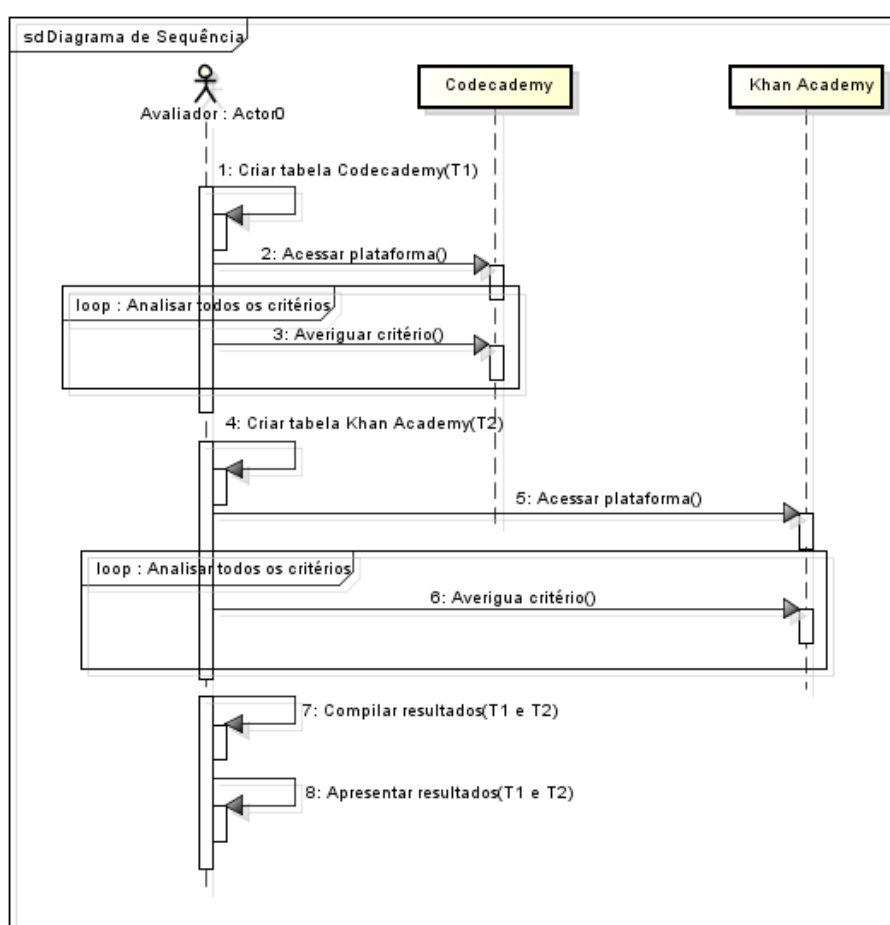
Fonte: Desenvolvido pelo autor, 2018.

Com a realização do procedimento de atribuição das notas, o avaliador realiza o levantamento e compilação dos resultados, conforme Figura 6, utilizando o auxílio de planilhas para a realização dos cálculos e do gerenciamento de teste. Na compilação dos resultados o avaliador busca saber a porcentagem de conformidade ergonômica que a plataforma atingiu. Com os resultados previamente compilados, é realizada a apresentação dos resultados. Cada critério é explicado e suas notas obtidas e ao final o desenvolvimento de um gráfico à ser apresentado no resultado final do processo de avaliação.

4.3 DIAGRAMA DE SEQUÊNCIA

Os diagramas de sequência mostra os fluxos de uma determinada tarefa. O ator que inicia a ação, os objetos envolvidos e as mensagens ou ações que trocadas e a ordem em que isso acontece. Isso possibilita uma análise completa da realização das tarefas e todos os elementos envolvidos nela, facilitando a compreensão dos procedimentos, (LIMA, 2011).

Figura 7 – Diagrama de sequência do processo de avaliação



Fonte: Desenvolvido pelo autor, 2018.

A Figura 7 ajuda a compreender todos os eventos para a realização da avaliação nas plataformas de ensino. O avaliador, cria a tabela (passo 1) Codecademy, que faz referência ao *checklist* já preparado para o teste, acessa a plataforma Codecademy (passo 2) e averigua os critérios (passo 3). Nesse passo está dentro de um loop, em que o avaliador terá que passar por todos os critérios do *checklist*. Os mesmos passos são executados para a

plataforma Khan Academy, cria tabela (passo 4), acessa o sistema (passo 5) e averigua os critérios (passo 6).

Com a realização desses passos mencionados anteriormente, o avaliador compila os resultados de ambas as plataformas (passo 7), para uma análise e descrição dos resultados de cada plataforma, apontando as diferenças entre elas e informando suas respectivas notas. Com esses procedimentos realizados o avaliador apresenta os resultados (passo 8), utilizando planilhas e gráficos e discernindo os resultados e apresentado a melhor plataforma para estar auxiliando no processo de ensino-aprendizagem em programação.

5 DESENVOLVIMENTO

Neste capítulo serão abordados os passos utilizados para a aplicação do *checklist* TICESE nas plataformas de ensino de programação *online* Khan Academy e Codecademy. A aplicação do *checklist* teve como intuito avaliar e analisar as ferramentas para dessa forma eleger a melhor ferramenta de ensino de programação para os alunos, tendo como base o resultado final do *checklist* TICESE executado nas referidas plataformas.

Para a aplicação do teste, foram realizadas 4 etapas, tempo de ambientação na plataforma, especificação dos pesos dos atributos do *checklist*, aplicação do teste e análise dos resultados. Essas etapas serviram para a preparação do teste, tais como: entender o funcionamento das ferramentas, conhecer os atributos do *checklist* TICESE, definir os pesos dos atributos. Todos esses passos serão abordados no decorrer deste capítulo.

5.1 FUNCIONAMENTO DAS PLATAFORMAS DE PROGRAMAÇÃO

As plataformas definidas para a aplicação do *checklist* foram a Khan Academy e a Codecademy, ambas as tem o objetivo de oferecer cursos gratuitos de programação *online*. O fato de ambos os sistemas serem *online*, não necessita que o aluno instale no seu computador algum tipo de *software*, sendo que apenas necessita de um computador conectado à rede e um navegador de *internet*. A vantagem dessa abordagem é que aluno não precisa se preocupar em instalar IDE, *plugins*¹⁴, ou realizar algumas configurações para fazer funcionar, o aluno apenas se preocupa com o processo de ensino-aprendizagem proposto pela plataforma.

Para a utilização das plataformas, é necessário criar um usuário e senha, podendo utilizar as credenciais do Google ou do Facebook, caso o aluno desejar. Isso se faz necessário para a armazenar as informações do progresso do aluno, os cursos que já foram concluídos e os que ainda falta concluir. Ao utilizar as contas tanto para fazer o cadastro tanto do Google como do Facebook, facilita no processo de acesso aos *sites* das ferramentas, poupando o usuário do preenchimento de um cadastro.

Ambas as plataformas dispõe de uma área de ajuda, para o usuário mandar qualquer dúvida a respeito do *site* ou resolução de exercícios ou entendimentos de comandos. Para a explicação dos comandos, ou palavras chaves referentes a linguagem em que o aluno

¹⁴ Todo programa ferramenta ou extensão que se encaixa em outro programa principal para adicionar mais recursos e funções a ele. Disponível em: <https://www.tecmundo.com.br/hardware/210-o-que-e-plugin-.htm>. Acesso em: 26 out. 2018.

está aprendendo, os alunos tem acesso a um glossário para os termos utilizados nas linguagens, ou para entender como funciona alguns métodos da linguagem que o aluno está aprendendo.

As ferramentas de ensino Khan Academy e Codecademy, possibilitam ao aluno parar, ou mudar no decorrer do curso. Caso o aluno opte por parar o curso em que está fazendo, os sistemas armazenam o progresso do aluno e quando o aluno voltar para terminar o curso, vai continuar de onde tinha parado pela última vez. O aluno não precisa concluir um módulo para trocar de curso, ou linguagem. As plataformas oferecem essa liberdade para o aluno.

5.1.1 Plataforma Khan Academy

A plataforma Khan Academy¹⁵ foi criado por Sal Khan em 2007 que começou a fazer vídeos para ensinar a sua prima que tinha dificuldades em matemática. A plataforma cresceu e se tornou uma organização com mais de 150 funcionários, cuja missão é oferecer uma educação de qualidade para qualquer um em qualquer lugar, por se tratar de uma plataforma totalmente *online*, as únicas coisas que o aluno precisa para acessar o portal Khan Academy é um computador, um navegador de *internet* e uma conta no próprio portal da instituição.

A Khan Academy oferece exercícios, vídeos de instruções e um painel de aprendizado personalizado que habilita os estudantes a aprender no seu próprio ritmo dentro e fora da sala de aula. Os cursos que a plataforma oferece vão desde fundamentos básicos de matemática até o cálculo integral, computação, com tecnologias Web, HTML, CSS e Javascript, linguagem SQL, disciplinas de economia e diversas outras matérias, todas seguindo a mesma forma de ensino, vídeo introdutório, explicação e exercícios para o aluno praticar.

No processo de ensino-aprendizagem do aluno, conforme o aluno vai assistindo os vídeos e resolvendo os exercícios vai ganhando ponto de experiências e ao completar alguma atividade ou módulo o aluno ganha um emblema como recompensa pela sua jornada no ensino, é a gamificação sendo utilizada no processo de ensino-aprendizagem. Essas recompensas motivam o aluno a cada vez ir mais em frente, e conseqüentemente, para o aluno

¹⁵ Disponível em: <https://pt.khanacademy.org/>

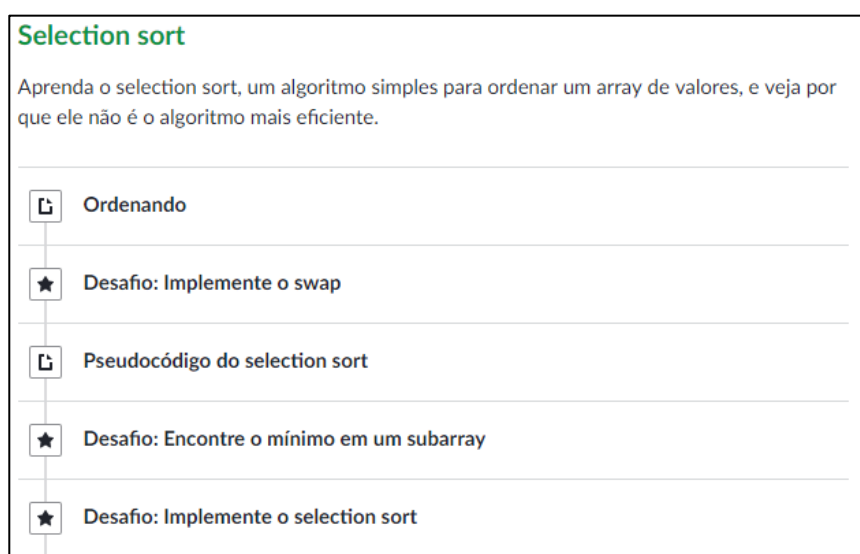
ir mais em frente o aluno precisa estudar mais e garantir mais conhecimento para conseguir resolver as atividades da plataforma, (KHAN ACADEMY, 2018).

O Stanford Consulting Group realizou uma pesquisa em 2015 com a seguinte pergunta. Você acha que a Khan Academy é significativa para sua educação? Dos 504 entrevistados da faculdade de Stanfor 65% disseram que sim, dos alunos das principais faculdades como Harvard, Princeton, Yale entre outras, dando um total de 159 entrevistador 57% afirmaram que a Khan Academy é significativa para a sua própria educação. Esse estudo pode demonstrar a importância da plataforma Khan Academy e como ela vem ganhando espaço como uma ferramenta no processo de auxílio-aprendizagem dos alunos ao redor do mundo, (KHAN ACADEMY, 2018).

A metodologia aplicada na plataforma Khan Academy faz o uso de vídeos e textos explicativos. Ao iniciar um curso, a primeira etapa é o aluno visualizar um vídeo totalmente em português explicando o assunto e demonstrando alguns comandos da linguagem. A parte do texto explica as atividades e alguns comandos da linguagem.

Ao escolher um curso, o aluno é direcionado a uma página, conforme Figura 8, em que apresenta a uma breve explicação e a estrutura do curso escolhida, com os capítulos que aluno vai realizar. Caso desejar avançar algum capítulo e ir direto a parte prática, o aluno tem liberdade em avançar nas atividades e pode pular as partes explicativas. Fazendo esse tipo de caminho o aluno deixa de ganhar pontos de energia, essa metodologia faz com que o aluno realize todos os módulos do curso e sem pular nenhuma etapa.

Figura 8 – Tela da estrutura do curso Khan Academy

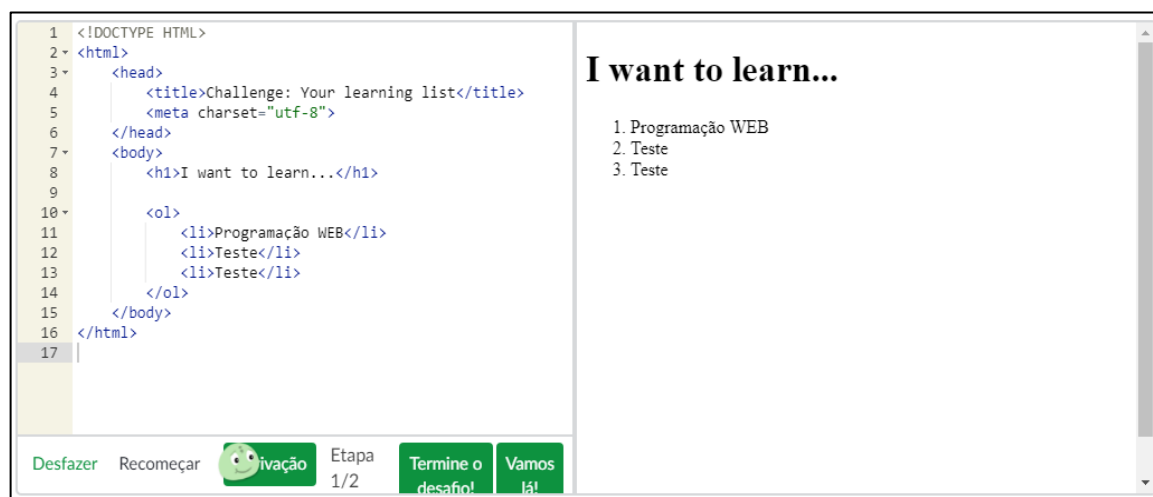


Fonte: <https://pt.khanacademy.org>, 2018.

No decorrer dos cursos, a plataforma oferece uma atividade em que o aluno tem que fazer o uso dos conhecimentos aprendidos e usar a imaginação. Nesses desafios não é o sistema que vai corrigir a atividade, quem vai corrigir ou dar sugestões são os outros alunos que fazem parte da comunidade Khan Academy. Isso possibilita um ambiente de troca de ideias, conhecimentos e informações, fazendo com que vários alunos se ajudem.

Ao iniciar o curso o aluno é redirecionado à página para a realização das atividades, conforme a Figura 9, nesta tela o aluno insere as linhas de código no lado esquerdo e no lado direito vai sendo executado cada linha que o aluno cria. Nesta mesma tela encontram-se os botões de desfazer, que é desfeito a última ação e o botão de recomendar, que apaga tudo o que o aluno fez e volta a atividade do zero.

Figura 9 – Tela de programação



Fonte: <https://pt.khanacademy.org>, 2018.

O *site* oferece um espaço para o professor e também para os pais. O espaço do professor ajuda a visualizar como está o andamento dos alunos em relação as atividades e os pontos de energia ganhos de cada. Essa função ajuda o professor a ter um controle dos alunos, propor atividades e cursos para os seus alunos, ao mesmo tempo em que possibilita realizar dinâmicas, premiando os alunos que atingirem determinada quantidade de pontos, ou, conseguir algumas medalhas que a plataforma oferece.

No andamento do curso, é oferecida ao aluno a leitura de outros documentos, apresenta *sites*, tudo para aumentar o conhecimento do aluno e propor outras fontes de estudo para estar incrementando ainda mais o conhecimento. Outra metodologia que a plataforma

oferece é a realização de perguntas e resposta, para verificar como estão os conhecimentos adquiridos ao longo do curso.

5.1.2 Plataforma Codecademy

A plataforma Codecademy¹⁶ foi fundada em agosto de 2011 por Zach Sims e Ryan Bubinski. É um *site* que possui uma plataforma interativa *online* e oferece aulas gratuitas de programação em linguagens como jQuery, Javascript, Python e linguagens de marcação HTML e CSS. O serviço funciona de maneira que os usuários progredam nas lições, começando com exercícios básicos para fundamentar os comandos da linguagem em que está sendo aprendida e conforme o aluno vai avançando pelos exercícios a dificuldade vai aumentando e se tornando cada vez mais desafiador, (GOMES, 2012).

O sistema propõe a construir um caminho para qualquer pessoa aprender a programar. Seu objetivo é transformar um mundo de consumidores de tecnologia em mundo de construtores de tecnologia por meio da educação. O sistema ensina conceitos de programação através de cursos *online*, a qualidade dos cursos é garantida pela própria equipe do sistema. O curso é direcionado a um usuário, como também não existe suporte para um professor acompanhar o desempenho de uma turma, (PAES et al., 2013).

Cada usuário tem o seu próprio perfil e para motivar os usuários a participar dos exercícios e das aulas, o *site* oferece feedback e emblemas ao completar os exercícios, é a aplicação da gamificação no processo de ensino-aprendizagem. Segundo a Codecademy (2018), somos uma instituição de ensino que tenta desenvolver a melhor experiência de aprendizado, pois, a *Internet* vez com que repensássemos em tudo e o grande objetivo é construir a educação que o mundo precisa, o primeiro sistema de educação nativo da *Internet*.

A plataforma Codecademy, oferece uma gama de cursos para o aluno e diversas linguagens de programação, como: Java, Python, Ruby e dentre outras. Muitos dos seus conteúdos ainda estão em inglês, que são a parte explicativa da linguagem e dos exercícios. A plataforma também oferece uma *interface* que é bastante similar a uma IDE dando a impressão do aluno estar utilizando realmente um ambiente de programação.

A estrutura do curso na Codecademy, inicia o aluno com uma parte explicativa, mostrando os comandos básicos da linguagem e uma breve explicação das atividades, conforme o aluno vai avançando nas atividades, o nível de dificuldades. Para não ficar só na

¹⁶ Disponível em: <https://www.codecademy.com/pt-BR>

programação, o sistema muda os exercícios e coloca o aluno para responder a questões referentes a linguagem e analisar os conhecimentos aprendido pelo aluno.

Ao aluno escolher um curso, é direcionado a uma página que mostra toda a grade do curso, conforme Figura 10, em que apresenta o nome do curso e todos os capítulos que compõem a grade do curso. Tendo sempre a mesma estrutura, parte introdutória, com uma breve explicação da linguagem e logo em seguida os conteúdos propostos pelo módulo.

Figura 10 – Tela da estrutura do curso Codecademy



Fonte: <https://www.codecademy.com>, 2018.

Ao iniciar o curso, o aluno é direcionado a tela para a realização das atividades, em que se assemelha a uma IDE, do lado esquerdo da tela o aluno insere as linhas de código e do lado direito aparece a execução do código.

O *site* Codecademy sempre mantém o aluno informado através de *e-mails* quando um novo curso for criado, ou um novo evento de programação está acontecendo, tudo isso para manter o interesse do aluno na programação e atualizado sobre o que acontece tanto no *site* quanto nos eventos de programação que ocorrem.

5.2 PREPARAÇÃO PARA APLICAÇÃO DO *CHECKLIST* TICESE

Para a aplicação do teste, o avaliador teve um tempo para a utilização do Khan Academy e Codecademy, analisar, conhecer e aprender como elas funcionam. Essa etapa tem

como objetivo a ambientação do avaliador nas plataformas, isso favorece um melhor conhecimento e garante que os resultados da avaliação sejam mais concisos.

O tempo de ambientação é importante para o avaliador, para com isso, ter o conhecimento do sistema em que vai ser analisado, pois, se fosse aplicado o *checklist* sem esse tempo de experimentar as plataformas, o teste poderia gerar dados inconclusivos e a avaliação poderia não alcançar seus objetivos.

O *checklist* TICESE é dividido em três módulos, sendo eles: o (i) módulo de classificação, (ii) módulo de avaliação e o (iii) módulo de avaliação contextual. O primeiro módulo (i) tem a função de classificar o *software* a partir de seus atributos e propostas pedagógicas. O segundo (ii), tem como função avaliar o *software* com base em critérios de ergonomia e o módulo (iii), verifica a adequabilidade do *software* em uma dado contexto pedagógico.

Para a aplicação do teste, o avaliador precisa ler cada atributo do *checklist* e verificar a existência ou não do atributo no *software*. Esse processo é importante para configurar e adaptar a ferramenta TICESE ao contexto em que vai ser utilizado. A cada leitura da questão é verificado se o *software* contempla ou não o atributo e assim é definido os pesos do atributo. Esse processo se faz necessário para que os atributos que as plataformas de programação não contempla, não irá interferir no resultado final do *checklist*.

5.3 APLICAÇÃO DO *CHECKLIST* TICESE

Com a realização das primeiras etapas, ambientação das plataformas, leitura e conhecimento do *checklist*, atribuição dos pesos dos atributos, é aplicado o *checklist* TICESE. Para a realização do teste foi necessário criar uma planilha para o armazenar as notas e fazer os cálculos das notas ao final do teste.

A planilha desenvolvida, conforme Quadro 4, apresenta o título do critério e faz referência ao seu subcritério, que neste caso é o critério: documentação; subcritério: densidade informacional. Apresenta as colunas referentes a cada *software*, a coluna referente as notas e outra coluna com o resultado da multiplicação da nota pelo peso.

O objetivo da planilha é ajudar o avaliador a manter organizado o teste, contendo o número da questão, o peso, as notas de ambas as plataformas e por fim o cálculo feito multiplicando o peso pela nota. Para cada critério e subcritério foi desenvolvido uma tabela, afim de manter a organização e poder discorrer sobre os resultados obtidos no teste.

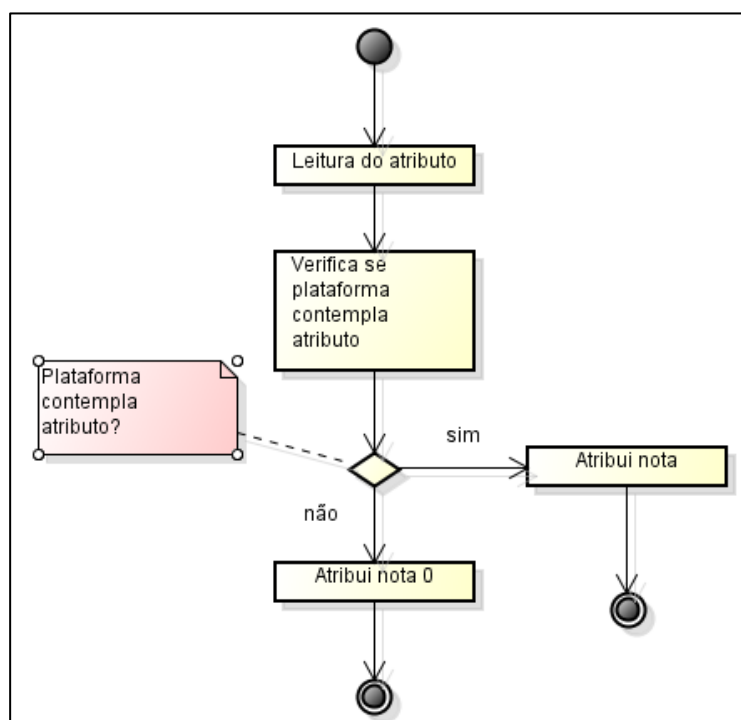
Quadro 4 – Arranjo dos critério

Documentação: Densidade Informacional					
Questão	Peso	Nota Khan Academy	Nota Codecademy	Peso * Nota Khan Academy	Peso * Nota Codecademy
1	1,5	1	1	1,5	1,5
2	1,5	1	1	1,5	1,5
3	1,5	1	1	1,5	1,5
4	1,5	1	1	1,5	1,5

Fonte: Desenvolvido pelo autor, 2018.

Para a aplicação do teste, foi retomado o uso das plataformas. Mesmo com o tempo de ambientação que o avaliador teve, foi retomado a utilização juntamente com o *checklist*. Isso foi para garantir ainda mais a veracidade das resposta, pois, o avaliador pode esquecer alguma informação ou ação da sistema de ensino e não informar uma nota coerente com a realidade das ferramenta. A Figura 11 exemplifica o processo de avaliação dos sistema Khan Academy e Codecademy.

Figura 11 – Diagrama de procedimento da avaliação



Fonte: Desenvolvido pelo autor, 2018.

Conforme a Figura 11, depois de retomar a utilização da plataforma, o avaliador faz a leitura do atributo do *checklist* e analisa se a ferramenta contempla ou não aquele

determinado critério. Caso sim, o avaliador informa uma nota dentro dos parâmetros da técnica e caso não, é atribuído nota 0 (zero), e o peso deste atributo continua o mesmo. Tendo uma nota 0 (zero) em algum critério que tenha um peso, implica no resultado final da avaliação, por isso, a importância de retomar o uso das plataformas e ter uma sequência bem definida dos procedimentos para a realização do teste. Conforme Quadro 5, demonstra uma parte da avaliação do *checklist*, o avaliador lê a questão, e informa a nota fazendo uma marcação em um dos quadrados relacionado com S, P e N. Fazendo a marcação, facilita a avaliação, na qual, o avaliador não precisa inserir o valor de uma nota, mais sim, atender a critérios como: Sim, Parcialmente e Não.

Quadro 5 – Critérios de avaliação do *checklist*

Documentação: Consistência	S	P	N	PESO
1) Há consistência na numeração das páginas apresentadas no sumário?				
2) Há consistência entre as informações descritas na documentação e a operação do produto?				
3) Todas as páginas estão presentes e postas na sequência correta?				

Fonte: Adaptado de Gamez (1998).

Para a aplicação do teste, foi retomado o uso das plataformas, e o teste procedeu da seguinte forma, o avaliador lê um atributo, verifica e faz os testes para ver se a plataforma contempla aquele atributo e a maneira como a plataforma realiza o atributo é atribuído uma nota. Esse procedimento foi realizado com todos os critérios do *checklist*.

A avaliação aconteceu primeiro na plataforma Khan Academy e depois para a plataforma Codecademy. As avaliações foram realizadas em dias diferentes para manter a lisura do processo de avaliação, pois, acredita-se que a fadiga do avaliador possa interferir na avaliação. De acordo com Iale (2015), caso o avaliador responder muitas avaliações, começa a sentir fadiga, não avaliando com qualidade e dando respostas sem veracidade. Como o processo de avaliação demanda tempo e uma análise de cada critério, optou-se por realizar em dias diferentes para garantir o resultado do teste.

Ao final do teste, é feito o levantamento de todas as notas e pesos dos atributos, conforme Quadro 6, e realizado a multiplicação dos pesos pelas notas dos atributos. As informações apresentadas, é a nota de cada subcritério e a nota geral tendo como base os subcritérios, cada coluna da planilha representado as plataformas avaliadas.

Quadro 6 – Planilha para apresentação dos resultados

Crítérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Gestão de Erros	Correção de Erros	-	-	-	-
	Qualidade das Mensagens de Erro	-	-		
	Proteção contra Erros	-	-		

Fonte: Desenvolvido pelo autor, 2018.

A utilização das planilhas ajudaram no desenvolvimento e organização do teste. No desenvolvimento, foi preparados as planilhas de cada tópico do *checklist* e adicionado os pesos de cada atributos atribuídos no processo de preparação do teste. Conforme o avaliador avalia a plataforma, é anotado a nota na planilha e a mesma realiza o cálculo da nota multiplicada pelo peso. A avaliação de cada critério, pode ser vista no Quadro 14, no Apêndice A, em que mostra as notas e os pesos que cada SE obteve na avaliação.

Na organização do teste, cada planilha está identificada com seu respectivo nome e seus atributos e número da questão, facilitando o processo de avaliação e cria uma espécie de roteiro para o avaliador. Nas mesmas planilhas foi gerada uma tabela contendo todos os resultados, conforme Quadro 15, Apêndice B, em que o avaliador tem uma visão geral dos resultados, tanto dos critérios, subcritérios e um valor final, tendo como base os resultados finais de cada critério.

Ao final, para uma melhor detalhamento do resultado final, foi gerado um gráfico, que apresenta as notas nos três módulos do *checklist* TICESE, módulo de avaliação, classificação e avaliação contextual, conforme Gráfico 1, e que por fim apresenta a nota geral que tem como base de cálculo as notas obtidas pelos três módulos de avaliação.

A nota é apresentada em forma de porcentagem (%), que é o valor de conformidade ergonômica em que determinado critério atingiu, consequentemente quanto maior o valor, mais em conformidade ergonômica o *software* está, tendo como base os pesos atribuídos pelo avaliador no *checklit*.

6 ANÁLISE DOS RESULTADOS

As notas de todos os critérios e subcritérios, bem como as notas gerais de cada plataforma, foram dispostos numa planilha para uma melhor análise. Conforme Quadro 15 Apêndice B, pode ser observado uma divisão dos critérios na avaliação, sendo estes a avaliação da documentação, que levou em conta informações referentes as plataformas em suas próprias páginas, não recorrendo a páginas de terceiros para verificar a funcionalidade das plataformas. A avaliação da plataforma, depois que foi definido o peso dos critérios do *checklist* e do tempo de ambientação das plataformas, foi retomado o uso das plataforma e feito a aplicação do teste e responde aos critérios e subcritérios do *checklist* e a avaliação contextual que analise a aplicabilidade da plataforma no ambiente de ensino.

O módulo de classificação, Anexo A, não tem a função de avaliar as plataformas, mais sim de orientar o avaliador quanto a sua classificação. Ambos os SEs se classificam como exercício e pratica, pois, o aluno é apresentado a explicação do assunto e inserido na resolução de exercícios. Exercitando assim os conhecimentos adquiridos.

Os valores percentuais obtidos são calculados a partir de subcritérios e dos seus respectivos critérios. O detalhamento dos critérios em subcritérios guia o avaliador e minimiza o nível de abstração na aplicação da técnica de avaliação. A apresentação dos resultados é dada na forma de números percentuais, pois, facilitam na análise e interpretação dos resultados e uma forma mais fácil de comparar os resultados com outras plataformas.

Os resultados em ambas as plataformas se mostraram muito parecidos, em alguns critérios tem uma diferença bastante significativa. A análise de cada critério e dos seus respectivos subcritérios, feita no decorrer deste capítulo, faz uma análise das diferenças de cada plataforma, transformando os valores obtidos nas avaliações em informações para tirar as conclusões de qual é a melhor plataforma para os alunos iniciantes em matérias de lógica de programação.

6.1 AVALIAÇÃO DOS RESULTADO POR CRITÉRIO

Nesta seção do trabalho, será apresentado as notas das plataformas obtidas de cada critério e subcritério do *checklist* TICESE. Compilando os resultados e analisando as diferenças entre as notas, pontuando características que difere umas das outras.

6.1.1 Dados de identificação e qualidade da informação impressa

Segundo Gamez (1998), o critério dados de identificação descrevem os pré-requisitos técnicos necessário para orientar o usuário na utilização do produto. O *software* deve fornecer informações sobre os objetivos educacionais e dos requisitos pedagógicos, para guiar tanto alunos como professores em relação ao uso da plataforma. Com relação a qualidade da informações impressa, refere-se a documentação referente ao *software* que pode orientar na tarefa da compreensão do conteúdo e auxiliar na redução de desapontamentos e experiências frustrantes, geralmente causadas pela dificuldade no manuseio dos sistemas.

Os critérios dados de identificação e qualidade da informação impressa fazem parte da avaliação da documentação, conforme Quadro 7, foram dispostos os resultados da avaliação de ambas as plataformas. As notas foram divididas em duas seções, tendo as notas individuais de cada subcritério e a nota geral de cada critério, que é gerada com base nas notas dos seus subcritérios. O critério dados de identificação não é composto por nenhum subcritério como mostrado no Quadro 7.

Quadro 7 – Notas dos critérios avaliação da documentação

Crítérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Dados de identificação	-	-	-	100%	87,50%
Qualidade da informação impressa	Presteza	100%	75%	88,90%	80,50%
	Legibilidade	92,85%	82,14%		
	Agrupamento de Itens	100%	100%		
	Densidade Informacional	100%	100%		
	Consistência	100%	100%		
	Significado dos Códigos e Denominações	81,81%	72,72%		

Fonte: Desenvolvido pelo autor, 2018.

No requisitos dados de identificação, a plataforma forma Khan Academy obteve uma nota maior (100%) em relação a nota da Codecademy (87,50%), a diferença foi devido ao fato da plataforma ter mais informações sobre seu funcionamento, demonstrar dados das instituições que a utilizam e ter uma grande comunidade ativa e um blog de perguntas e respostas dentro da mesma plataforma, onde, sempre tem um profissional para responder.

Além de ter um quadro de seus funcionários e relatos de usuários, isso ajuda e motiva o aluno ao ler e visualizar essas histórias. A plataforma Codecademy tem poucas informações do funcionamento da plataforma e quando o usuário clica na aba sobre, existe poucas informações, informações essas que fazem menção ao objetivo da plataforma e existe uma seção em que mostra as pessoas que trabalham na plataforma, sem muitas informações.

No critério qualidade da informação impressa, muitos atributos tiveram o valor do peso atribuído com o valor 0 (zero), pois, os atributos faziam referência como se a plataforma fosse instalada no computador do usuário. Os atributos faziam relação se a plataforma especificava o sistema operacional, requisitos mínimos de configuração do computador e dentre outros atributos, como as plataformas são *online* essas questões não foram respondidas e receberam o peso 0 (zero).

A nota de ambas as plataformas foram bastantes aproximada 88,90% e 80,50%, para Khan Academy e Codecademy, respectivamente. Nos subcritérios que houve uma maior diferença foi, *presteza* (100% e 75%), o fato dessa diferença é a acessibilidade do glossário que na plataforma Khan Academy fica de fácil acesso e o usuário não precisa sair da tela em que está realizando as atividades, e o significado dos códigos e denominações (81,81%, 72,72%), a diferença entre os valores é baixo, e o fator que levou a essa diferença foi que a plataforma Codecademy tem uma mistura de conteúdos em inglês e português, isso com o tempo pode a plataforma pode ir se ajustando a cada grupo de usuários ao redor do mundo.

6.1.2 Condução

Para Gamez (1998), a condução faz referência aos meios disponíveis para orientar, informar e conduzir o utilizador na interação do sistema. Quatro subcritérios fazem parte do critério condução, que são:

- *Presteza*: engloba os meios utilizados para levar o utilizador a realizar determinadas ações, como a entrada de dados;
- *Qualidade das opções de ajuda*: avalia a conformidade da opção de ajuda oferecida pelo *software* e sua qualidade na busca de informações na resolução de problemas;
- *Legibilidade*: como é a apresentação de mensagens, tanto de erros, como mensagens de ajuda para a resolução de problemas;
- *Feedback imediato*: respostas do sistema as ações do utilizador;

- Agrupamento e distinção de itens: quanto a organização visual e localização dos itens de informação.

O critério condução faz parte da avaliação da plataforma, conforme Quadro 8, foi dispostos os resultados da avaliação de ambas as plataformas. O critério condução é composto por vários subcritérios e as notas foram divididas em duas seções, tendo as notas individuais de cada subcritério de cada plataforma e a nota geral que representa a nota do critério condução. A nota geral de cada plataforma, no critério condução, é gerado através das notas dos subcritérios.

Quadro 8 – Notas critério condução

Crítérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Condução	Presteza	93,18%	82,92%	86,30%	81%
	Qualidade das opções de ajuda	83,33%	77,14%		
	Legibilidade	87,09%	90,74%		
	Agrupamento e distinção por formato	92,85%	100%		
	Agrupamento e distinção por localização	94,44%	76,47%		
	Feedback imediato	85,71%	86,11%		

Fonte: Desenvolvido pelo autor, 2018.

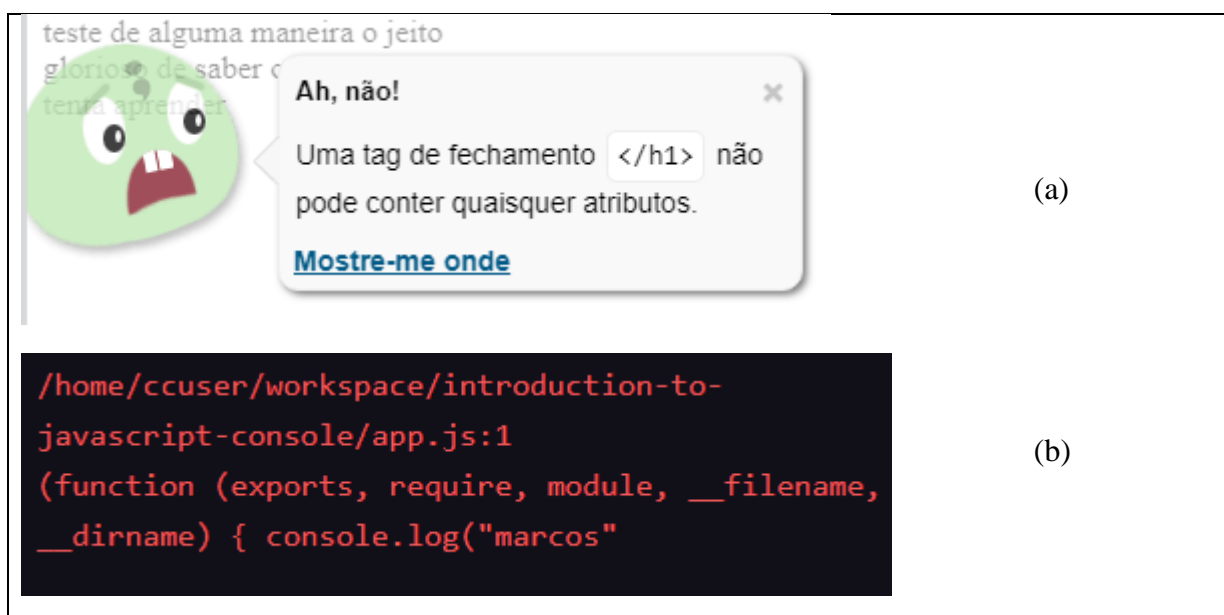
Na avaliação do critério condução, no resultado final tendo como base os seus subcritérios, a nota da plataforma Khan Academy foi de 86,30% enquanto a Codecademy foi de 81%, sendo uma pouca diferença entre as plataformas, no nota final. O subcritérios presteza, ambas as plataformas dispõem de um glossário no auxílio da compreensão dos termos técnicos, na plataforma Khan Academy o glossário e a documentação ficam logo abaixo da área onde o aluno executa as atividades, enquanto na plataforma Codecademy, em muitos o aluno precisa ser direcionado para outra página para verificar o glossário.

No requisito qualidade de opções de ajuda, ambas as plataformas apresentam um bom sistema para ajudar o usuário, caso tenha dificuldades na realização de alguma tarefa. A plataforma Codecademy, se o usuário realiza muitas tentativas de uma inserção errada do código, a plataforma apresenta uma mensagem se o usuário aceita receber a resposta do exercício, em tese isso pode ajudar, pois o usuário consegue analisar a resposta dado, fazer

alguma edição e passar no exercício. A plataforma Khan Academy apresenta uma série de dicas para o usuário, para conseguir realizar a tarefa, a plataforma não apresenta a resposta pronto e faz com que o aluno continue até conseguir acertar, por um lado isso motiva o aluno e ao mesmo tempo caso o aluno não consiga resolver o exercício o desmotiva a continuar.

No critério legibilidade, que faz referência como são apresentados as mensagens de erros e as mensagens de ajuda, cada plataforma tem uma jeito de apresentar as mensagens. Conforme a Figura 12, apresenta a mensagem de erros de ambas as plataformas, a Figura 12 (a) apresenta a mensagem de erro da plataforma Khan Academy, que faz o uso de um avatar com uma feição de assustado, como se estivesse falando para o erro que aconteceu. Na plataforma Codecademy, na Figura 12 (b), a mensagem de erro apresentada para o usuário faz uma alusão de mensagem apresentado por uma IDE. Isso aproxima o usuário da realidade de uma mensagem de erro, fazendo com que a plataforma Codecademy tenha nota mais alta.

Figura 12 – Comparação de mensagens de erros das plataformas



Fonte: Desenvolvido pelo autor, 2018.

Nos demais subcritérios, não houve grande diferenças das notas. Com relação aos critérios agrupamento e distinção por formato e agrupamento e distinção por localização, ambas as plataformas sempre mantém um mesmo padrão de uma *interface* para outra, sempre tentando manter a experiência do usuário a mais melhor possível, tentando sempre manter o foco usuário na realização dos exercícios e na conclusão das atividades.

6.1.3 Carga de trabalho

O critério carga de trabalho é relacionado a todos os elementos da *interface* que realizam um papel importante na redução da carga cognitiva e perceptiva do utilizador. Quanto maior for os passos para que o usuário tenha que realizar para executar alguma tarefa, maior será a probabilidade de cometer erros e mais longo será o processo de aprendizagem. O usuário tem que se distrair o mínimo possível com informações desnecessárias, mais ele será capaz de desempenhar as suas tarefas eficientemente e atingir os objetivos educacionais propostos, (GAMEZ, 1998). O critério carga de trabalho é subdividido em três subcritérios:

- Carga informacional: a objetividade e a clareza com que a informação pedagógica é apresentada;
- Brevidade: corresponde ao objetivo delimitar a carga de trabalho de leitura e entradas e o número de passos;
- Densidade informacional: faz relação com a carga de trabalho do utilizador de um ponto de vista perceptivo e cognitivo, a carga de memorização deve ser minimizada, para o usuário se concentrar somente na aprendizagem.

O critério carga de trabalho faz parte da avaliação da plataforma, os resultados foram dispostos, conforme Quadro 9, para uma melhor análise dos resultados desse critério. A nota do critério carga de trabalho é feita tendo como base a nota de seus subcritérios, carga informacional, concisão, ações mínimas e densidade informacional. No Quadro 9 as notas estão divididas em duas seções, uma que é a nota de cada subcritério, e a segunda é a nota do critério carga de trabalho, todas as notas divididas por sua respectiva plataforma.

Quadro 9 – Notas do critério carga de trabalho

Critérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Carga de trabalho	Carga Informacional	83,33%	86,11%	64,10%	72,10%
	Concisão	100%	100%		
	Ações mínimas	50%	64,28%		
	Densidade informacional	80%	75%		

Fonte: Desenvolvido pelo autor, 2018.

Na avaliação do critério carga de trabalho a plataforma Codecademy obteve a maior nota sendo 72,10% contra 64,10% da plataforma Khan Academy. No subcritério carga

informacional, a plataforma Codecademy recebe a nota mais alta, por apresentar de melhor maneira o exercício proposto, mesmo a explicação do exercício estando em inglês, é apresentado a proposta do exercício e um exemplo, isso facilita ao aluno a compreensão do exercício, enquanto na plataforma Khan Academy o aluno tem algumas linhas de código informadas na área de trabalho e poucas dicas para ajuda na execução da atividade.

No subcritério concisão a nota foi a mesma nas plataformas, esse subcritério avalia os nomes dos botões e menus da *interface*, se são concisos com o nome e a função que executam e se a cada mudança de tela, os botões e comando continuam com o mesmo nome e as mesmas funções, tudo isso diz a padronização da plataforma, com relação a botões, menus, abas e demais componentes. Ambas as plataformas atenderam ao subcritério e com isso tirando a mesma nota.

A avaliação do subcritério ações mínimas, as notas das plataformas Khan Academy e Codecademy foi 50% e 64,28%, respectivamente. A plataforma Codecademy obteve a nota mais alta por manter somente as informações necessárias na tela em que o aluno realiza os exercícios. A tela que o aluno os exercícios, na plataforma Codecademy, ocupa toda a tela do computador, mostrando somente a explicação do exercício, área para realização da atividade (digitação do código) e outra parte para a execução do código, caso o aluno precise de outras informações, no canto esquerdo da tela o aluno tem acesso a um sistema de abas, na qual, fornece informações complementares para a execução do exercício.

Na plataforma Khan Academy, uma pequena parte da tela é reservada para a execução do exercício, e caso o aluno queira acessar o glossário, ou verificar outra informação deve ‘rolar’ a página e isso tira a atenção do aluno. No critério densidade informacional, a plataforma Khan Academy tem a nota mais alta, pelo fato de boa parte da documentação da linguagem e o glossário com informações dos comandos se encontram na mesma tela e isso facilita o acesso a essas informações, e a plataforma Codecademy, o aluno tem que mudar de página para acessar outras funcionalidades da plataforma ou acessar informações como o glossário ou outras informações para a execução do exercício.

6.1.4 Recursos de apoio à compreensão dos conteúdos e adaptabilidade

Os meios que a plataforma faz uso de técnicas para auxiliar na compreensão dos conteúdos pedagógicos. A utilização de recursos multimídia, recursos motivacionais, que auxiliam na explicação do conteúdo para o aluno. O *software* deve possuir recursos que auxiliem na aquisição de conhecimentos por parte do aluno, fazendo uso de diferentes

estratégias, como o uso de vídeos, recompensar pelo fato de conseguir completar as missões e diversas outras abordagens para prender o interesse do aluno, (GAMEZ, 1998).

A adaptabilidade tem relação com a capacidade de reagir conforme o contexto e conforme as necessidades do usuário. Segundo Gamez (1998), a adaptabilidade é composta por dois subcritérios:

- Flexibilidade: possibilidade de personalizar a *interface*;
- Considerações da experiência do utilizador: meios implementados que permitem que o *software* respeite os níveis de experiências de cada um.

Os critérios recursos de apoio a compreensão dos conteúdos e adaptabilidade fazem parte da avaliação da plataforma. As notas de ambos os critérios foram dispostas conforme a Quadro 10. O critério recurso de apoio a compreensão dos conteúdos, não é composto por nenhum subcritério, esse fato de algumas colunas não informações de nenhuma nota. O critério adaptabilidade são compostos por vários subcritérios, flexibilidade e considerações da experiência do utilizador, cada qual com a sua nota e juntos forma a nota do critério adaptabilidade.

Quadro 10 – Notas dos critérios recursos de apoio e adaptabilidade

Crítérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Recursos de apoio a compreensão dos conteúdos	-	-	-	87,20%	50%
Adaptabilidade	Flexibilidade	28,57%	14,28%	40,50%	29,80%
	Consideração da experiência do utilizador	76,47%	76,47%		

Fonte: Desenvolvido pelo autor, 2018.

O critério recursos de apoio a compreensão dos conteúdos, não é composto por nenhum subcritério. Esse critério avalia a metodologia aplicada por cada plataforma para ensinar os conteúdos ao aluno, a plataforma Khan Academy obteve uma nota mais alta, 87,20% enquanto a Codecademy obteve a nota de 50%. A Khan Academy se sobressai neste requisito por apresentar de uma maneira diferente dos conteúdos para os alunos. Sempre no início de cada atividade, é apresentado um vídeo explicando os comandos das linguagens e onde essa linguagem é aplicada, e conforme o vídeo vai avançando, na área em o aluno digita

as linhas de códigos, vai surgindo as linhas de programação que o vídeo vai explicando e ao final o aluno pode estar analisando as linhas de códigos ou realizando alterações.

Outro fator na metodologia de ensino em que a plataforma Khan Academy ganha pontos é na questão de leitura complementar para o aluno, no decorrer dos cursos, existe etapas em apresenta diversos conteúdos para o aluno aumentar o seu conhecimento e a apresentação de outros *sites* para estar complementando o ensino. A plataforma Codecademy se restringe a pequenos textos e explicações, não que isso prejudique a experiência do aluno, sendo que cada aluno aprende de forma diferente, mas o fato de explorar outras formas de passar o conteúdo faz com que a plataforma Khan Academy obtenha uma nota maior.

No critério adaptabilidade, ambas as plataformas tiveram notas bem baixas, sendo 40,50% para Khan Academy e 29,80% para Codecademy. As notas bem abaixo é que em muitos requisitos desse critério foi definido com peso 0 (zero), ambas as plataformas não tem configurações para ajustar conforme a deficiência ou necessidades de seus usuários. O subcritério flexibilidade, diz respeito a essas alterações na plataforma para adaptar conforme as necessidades do usuário. No subcritério considerações da experiência do utilizador, a nota foi a mesma para ambas as plataformas, 76,47%, pois, os exercícios vão aumentando de dificuldade conforme o aluno vai avançando e sempre apresentado novos desafios e caso o aluno ainda sinta um pouco de dificuldades pode repetir os níveis ou as atividades para assimilar os conteúdos.

6.1.5 Controle explícito e gestão de erros

De acordo com Gamez (1998), o controle explícito faz relação com o processamento explícito pelo sistema das ações do utilizador, quando os usuários, que neste caso são os alunos, definem as suas entradas e quando estas entradas estão sob seu controle. O critério controle explícito é subdividido em dois subcritérios:

- Ações explícitas do utilizador: computador deve processar somente as ações solicitadas pelo usuário;
- Controle do utilizador: ter o controle sobre o processamento de informações do sistema e com isso diminui a probabilidade de erros.

Quais os mecanismos utilizados que permitem evitar ou reduzir a ocorrência de erros e, quando eles ocorrem que favoreçam sua correção. Erros esses considerados, como entrada de dados incorreta ou comando com sintaxes incorretas, etc. Para Gamez (1998), o critério gestão de erros é dividido em três subcritérios:

- Correção de erros: meio colocado ao aluno para que possa corrigir os seus erros;
- Qualidade das mensagens de erros: qualidade das mensagens de erros, pontuando a legibilidade e exatidão da informação e sobre a ação que o aluno tem que executar para corrigir o erro;
- Proteção contra erros: mecanismos que possam prevenir e detectar os erros de entrada de dados, comandos, etc.

Os critérios controle explícito e gestão de erros fazem parte da avaliação da plataforma. As notas foram dispostas na Quadro 11, mostrando assim a nota em que cada subcritério ganhou na avaliação de sua respectiva plataforma, bem como a nota geral de cada plataforma que é composta tendo como base a nota de cada subcritério. Neste pode-se observar a diferença entre alguns subcritérios e isso impacta na nota final, fazendo que no critério gestão de erros a plataforma Khan Academy tenha uma melhor nota.

Quadro 11 – Notas dos critérios controle explícito e gestão de erros

Crítérios	Subcritérios	Nota Khan Subcritério	Nota Code Subcritério	Nota Geral Khan Academy	Nota Geral Codecademy
Controle explícito	Ações explícitas do utilizador	72,22%	83,33%	73,60%	84,72%
	Controle de utilizador	77,77%	86,88%		
Gestão de erros	Correção de erros	93,47%	67,39%	92,10%	66,50%
	Qualidade das mensagens de erros	95,83%	77,08%		
	Proteção contra erros	90,90%	63,63%		

Fonte: Desenvolvido pelo autor, 2018.

No critério controle explícito, a plataforma Codecademy teve uma nota maior com 84,72% enquanto a plataforma Khan Academy, obteve uma nota de 73,60%. Em ambos os subcritérios, ações explícitas do utilizador e controle do utilizador a plataforma Codecademy obteve as maiores notas, essa diferença na nota é que plataforma Codecademy, o usuário ao inserir as linhas de códigos e se achar conveniente, o mesmo clica no botão para executar aqueles comandos, consequentemente pode apresentar dois resultados, que ocorreu tudo certo ou apresentar uma mensagem de erro. Na plataforma Khan Academy, conforme o aluno vai desenvolvendo os códigos, já vai sendo executado e caso o aluno digita alguma informações,

já aparece o avatar informando uma mensagem de erro e isso pode prejudicar um pouco o desenvolvimento do aluno na resolução das atividades.

No critério gestão de erros, a plataforma Khan Academy obteve uma maior nota, 92,10% contra 66,50% da Codecademy. Ao mesmo tempo em que todos os outros subcritérios a plataforma Khan Academy obteve uma maior nota. No requisito qualidade das mensagens de erro, conforme apresentado na Figura 12, a plataforma apresenta um mensagem de erro mais simples, mesmo a plataforma Codecademy fazendo uma alusão a uma IDE, para alguns alunos uma mensagem de erro cheio de códigos fica difícil a compreensão, onde, a plataforma Khan Academy apresenta uma mensagem simples e intuitiva.

No requisito proteção contra erros e correção de erros, cada plataforma atua de uma maneira, enquanto na Khan Academy, conforme o aluno vai inserindo as linhas de código, vai sendo executada no lado direito da tela e caso o aluno informa ou esquece de encerrar a digitação da linha de código automaticamente a plataforma apresenta um mensagem e na sequência o aluno vai corrigindo, em partes pode ser ruim para o aluno essa metodologia ao mesmo tempo em que o aluno começa a se preocupar desde cedo com o mínimo de erros possíveis na digitação dos comandos. Ao contrário da plataforma Codecademy, o aluno primeiro digita os comandos e depois executa e caso tenha algum erro é apresentado as mensagens, em algum ponto pode desanimar o aluno, dependendo da complexidade do exercício e do número de linhas do programa digitado, dificulta a correção do erro por parte do aluno. Com isso, a plataforma Khan Academy tem as melhores notas neste critério.

6.1.6 Avaliação de aprendizagem, homogeneidade e compatibilidade

De acordo com Gamez (1998), são os meios disponíveis no sistema para verificar a aprendizagem do conteúdo por parte do aluno. Independentemente do tipo de plataforma, é importante que o sistema apresente alguma forma de analisar se os conceitos estão sendo aprendidos pelo aluno. Um exemplo desse tipo de avaliação, é a aplicação de questionários ou de situações problemas para a verificação e a análise dos conhecimentos adquiridos pelo aluno.

No critério homogeneidade, de acordo com Gamez (1998), refere-se ao modo que como as escolhas das posições dos comandos na *interface* são conservadas de uma tela para outra. É necessário escolher opções similares de códigos, procedimentos, denominações para contextos idênticos e utilizar os mesmos meios para obter os mesmos resultados. A falta de

padronização do sistema dificulta o uso intuitivo do sistema, sendo que cada tela funciona de maneira diferente, e a padronização é importante para que casos como esse não ocorram e não dificultam a experiência do usuário no uso da plataforma.

De acordo com Gamez (1998), a compatibilidade faz referência às relações que possam existir entre as características do utilizador na realização de suas tarefas e a organização das entradas e do diálogo de uma dada aplicação. A eficiência aumenta quando os procedimentos necessários e a informação é apresentada de uma forma diretamente utilizável para o cumprimento das tarefas são compatíveis com as características psicológicas do usuário.

Os critérios avaliação de aprendizagem, homogeneidade e compatibilidade fazem parte da avaliação da plataforma. As notas dos critérios foram dispostos na Quadro 12 para uma melhor análise das notas. Os três critérios não possuem subcritérios, são apenas um único atributo para realizar a avaliação nas plataformas. Em dois critérios as plataformas tem um diferença de notas, na avaliação de aprendizagem e no critério de homogeneidade e no critério compatibilidade ambas as plataformas conseguiram obter a mesma nota.

Quadro 12 – Notas avaliação aprendizagem, homogeneidade e compatibilidade

Crítérios	Nota Khan Academy	Nota Codecademy
Avaliação de aprendizagem	100%	72,22%
Homogeneidade	83,33%	91,66%
Compatibilidade	100%	100%

Fonte: Desenvolvido pelo autor, 2018.

No critério avaliação de aprendizagem, a plataforma Khan Academy teve uma nota melhor que a Codecademy, 100% e 72,22%, respectivamente. Uma fator que vez a plataforma Codecademy tirar uma nota foi pelo fato de que muitos de seus conteúdos não estão traduzidos na língua portuguesa, o que prejudica o processo de aprendizagem, fazendo que o aluno tenha que recorrer a *sites* para traduzir o que está escrito no enunciado dos exercícios. Ambas as plataformas dispõem de mecanismos para avaliar o grau de compreensão dos alunos, como a inserção de outras atividades, como perguntas e resposta. A plataforma Khan Academy se destaca por trazer o espaço do professor e o espaço dos pais, e isso possibilita um acompanhamento de pessoas externas no processo de aprendizagem do aluno.

Em relação ao critério homogeneidade, a plataforma Codecademy obteve uma nota maior em relação a Khan Academy, 91,66% a 83,33% respectivamente. A plataforma

Khan Academy tirou uma nota mais baixa, pelo fato de que em alguns momentos, o sistema reproduz um vídeo utilizando o mesmo *layout* da tela em que o aluno realiza as atividades. A *interface* é a mesma e só tem um botão de *play*, com relação a Codecademy, a plataforma mantém a mesma *interface* do início até o fim das atividades desenvolvidas pelos alunos.

No requisito compatibilidade, ambas as plataformas tirar a mesma nota, 100%, como as plataformas tratam de ensinar programação para os seus usuários, a entrada de dados é padrão em ambas as plataformas, o aluno está desenvolvendo o raciocínio em cima de uma determina linguagem de programação e com isso aprende todos os tipos de dados e as formas de inserir esses dados que a linguagem aceita. Por esse motivo que existe uma padronização na inserção de dados.

6.1.7 Significado de códigos e combinações e adequabilidade

O critério significados dos códigos e denominações refere-se a informação apresentada ou pedida, possuem uma relação semântica com seu referente. O nome dado para os botões e abas no *software* e entre outros elementos da tela, tem uma forte relação com o seu nome e a sua função. Códigos e denominações não significativos podem sugerir operações inadequadas ao contexto, conduzindo os usuários da plataforma a cometer erros, (GAMEZ, 1998).

O critério adequabilidade, segundo Gamez (1998), tem como objetivo auxiliar na decisão de adotar ou não a plataforma como ferramenta de ensino ou auxílio no processo de ensino-aprendizagem, mediante a confrontação dos resultados obtidos no módulo de avaliação, tendo como base as características gerais da disciplina como da instituição de ensino.

O critério significados de códigos e combinações faz parte da avaliação da plataforma, enquanto o critério adequabilidade faz parte da avaliação contextual. Ambos os critérios não possuem subcritérios, e a Quadro 13 apresenta os resultados de cada critério na avaliação de cada plataforma. Nos critérios apresentados pelo Quadro 13, ambas as plataformas obtiveram notas bem parecidas uma das outras, com uma pouca diferença de uma nota para outra. Em ambos os critérios a plataforma Khan Academy, consegue notas mais altas em relação a plataforma Codecademy, que tem bem pouca diferença percentual.

Quadro 13 – Notas dos critérios significado de códigos e adequabilidade

Crítérios	Nota Khan Academy	Nota Codecademy
Significado de códigos e combinações	96%	92%
Adequabilidade	86,20%	82,75%

Fonte: Desenvolvido pelo autor, 2018.

Na avaliação do critério significados de códigos e combinações, não houve uma muita diferença das notas entre as plataformas, 96% para Khan Academy e 92% para Codecamy. Essa diferença das notas é pelo fato da plataforma Codecademy, em muitas das explicações dos exercícios ainda estarem na língua inglesa, isso em parte dificulta a aprendizagem, sendo que nem todo mundo tem um certo nível de domínio da língua inglesa. Em ambas as plataforma, na hora de realizar o desenvolvimento dos exercícios, as plataformas mostram que quando o aluno utiliza alguma palavra reservada de determinada linguagem, a mesma muda de cor e fica diferente das demais, mostrando assim para o aluno que aquela faz parte da sintaxe da linguagem.

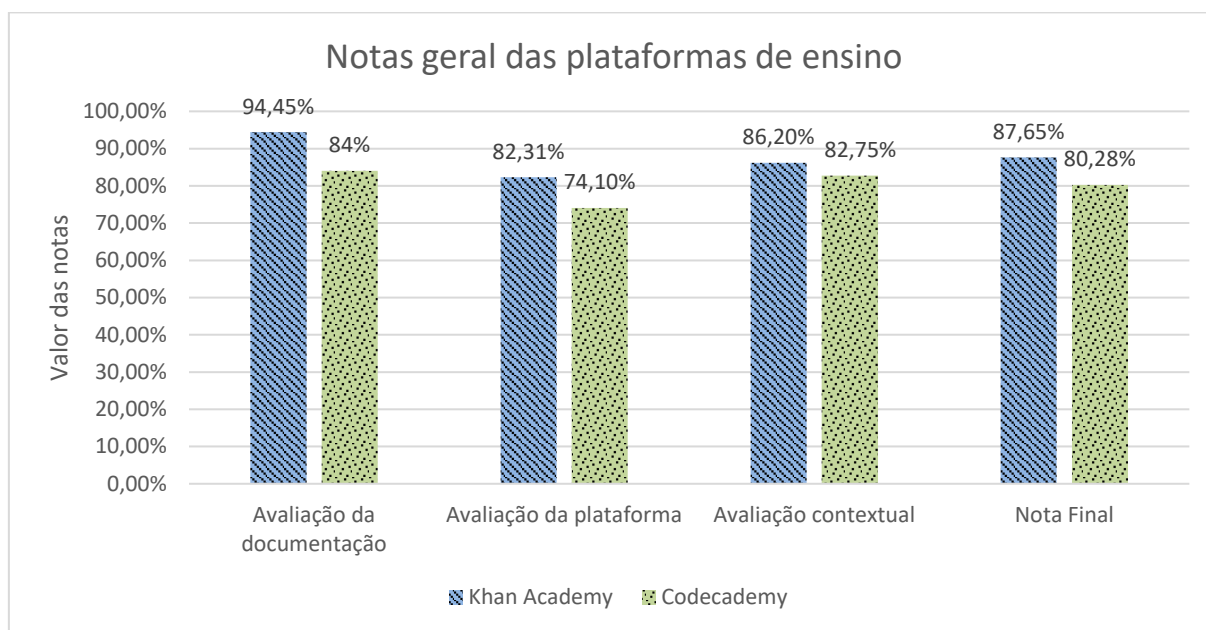
No critério adequabilidade, faz referência ao processo de avaliação contextual, ambas as plataformas conseguiram notas bastante parecidas. As plataforma apresenta uma metodologia de ensino para o aluno, com explicações e a aplicação de exercícios. Conforme o aluno vai avançando nos exercícios, conseqüentemente a dificuldade dos mesmos vai aumentando. No decorrer do ensino a plataforma muda o tipo de exercício e o aluno faz uma atividade de perguntas e resposta para verificar a compreensão de conceitos referentes a linguagem em que se está aprendendo. O que faz essa pequena diferença entre uma plataforma e outra, foi pelo fato da plataforma Khan Academy oferecer o espaço do professor e possibilitar um acompanhamento dos alunos e ao mesmo tempo oferecer diversos exercícios aos seus alunos. Outro fator foi a Khan Academy, oferecer um tipo de atividade em que o aluno tem que usar a criatividade e os seus conhecimentos adquiridos, sendo que quem vai avaliar essa atividade são os outros usuários da plataforma, favorecendo a troca de conhecimento entre os demais aluno.

6.1.8 Resultado final

Os resultados finais de ambas as plataformas foram plotados no Gráfico 1, mostrando os tópicos principais, avaliação da documentação, da plataforma e avaliação contextual e a nota geral que é gerada tendo como base esses três critérios. O gráfico ajuda a

analisar os resultados finais e apresentam uma noção do quão diferente em matéria de nota ficou uma plataforma longe da outra. Em alguns contextos não houve uma diferença tão significativa como na avaliação contextual já em outros tem um diferença realmente grande como a avaliação da plataforma e da documentação. No decorrer da seção será melhor avaliada os resultados obtidos através da avaliação.

Gráfico 1 – Notas geral das plataformas de ensino



Fonte: Desenvolvido pelo autor, 2018.

A Khan Academy, como pontos positivos que podem ser citados, são pelo fato da plataforma sempre iniciar qualquer lição com um pequeno vídeo introdutório, explicando para o aluno os comandos, o que acontece com tal comando e ao mesmo tempo da uma explicação em que lugar é aplicado a linguagem que o aluno está aprendendo. Na questão dos vídeos, conforme o vídeo vai avançando, vai aparecendo os códigos que são sendo explicados e ao final da execução do vídeo o aluno pode analisar e alterara o código que apareceu.

Um dos aspectos interessantes da Khan Academy é o fato de oferecer outras leituras e indicar *sites* com conteúdo para o aluno estar incorporando e assimilando no seu conhecimento. As atividades em que o aluno tem que usar a imaginação e os conhecimentos adquiridos para desenvolver os projetos, e quem corrige os projetos são os próprios alunos, dando dicas e sugestões, compartilhando o conhecimento e uns ajudando os outros. Outro ponto relevante é o espaço do professor ou dos pais, em que possibilita verificar o desenvolvimentos dos alunos no decorrer das atividades ao mesmo tempo em que possibilita a

indicação de exercícios. Essa verificação do professor abre possibilidade de criar dinâmicas com seus alunos e outras formas de fomentar o uso da plataforma.

A plataforma Codecademy tem uma metodologia aplicada ao aluno em que tem uma breve explicação da linguagem e já tem o primeiro contato com exercícios e assim começa a digitar as primeiras linhas de código. Toda a sua *interface* se assemelha a uma IDE, dando o aluno a entender que está programando em uma IDE da respectiva linguagem que o aluno está aprendendo. Tanto que as mensagens de erros que a plataforma apresenta são os erros que a linguagem sendo utilizado numa IDE apresentaria, aumentando a experiência do usuário.

Os cursos da Codecademy tem seus conteúdos ainda na língua inglesa, explicação das atividades e linguagem, e o resto das informações do *site* estão traduzidos para o português. O aluno precisa de um esforço para entender os enunciados dos exercícios e em alguns casos utilizar de *sites* para compreender o conteúdo. O Codecademy, oferece uma gama de cursos com diversos conteúdos e diversas linguagens de programação para o aluno estar aprendendo. A plataforma sempre mantém contato com seus usuários pelo envio de *e-mails*, apresentado para o aluno novos eventos que estão acontecendo ou novas linguagens e cursos que está sendo incorporado à plataforma.

Ambas as plataformas apresentam uma metodologia parecida, com uma explicação da linguagem e a introdução de atividades para o aluno realizar e ambas apresentam outras atividades, como perguntas e respostas para verificar se o aluno estão assimilando os conteúdos e os termos utilizados nas linguagens, isso tudo para diversificar o método de ensino e criar outras abordagens de aprendizado. Os SEs tentam recompensar seu alunos com algum tipo de prêmio por conseguir realizar algum módulo ou alguma atividade, na Khan Academy o aluno ganha pontos de energia e algumas medalhas para mostrar as conquistas do aluno, bem como na Codecademy trabalha com a conquista de medalhas.

Mesmo apresentando metodologias parecidas em sua essência, sempre vai haver uma diferença entre ambas, e a na nota geral tendo como base os critérios do *checklist* TICESE e nos três tipos de avaliação, documentação 94,45% na avaliação da plataforma 82,31% na avaliação contextual 86,20% e na nota geral tendo como base esses três critérios com 87,65%, com base no Gráfico 1, foi a Khan Academy. Com diversos pontos positivos, uma *interface* limpa, sistema de ajuda, metodologia de aprendizado e sempre buscando informar o aluno com leituras complementares e o fato de permitir o acompanhamento dos professores no desenvolvimento, faz com que a Khan Academy atende melhor os requisitos do *checklist* TICESE.

7 CONSIDERAÇÕES FINAIS

O presente trabalho, buscou aplicar um processo de avaliação de usabilidade em plataformas de programação *online*. Buscando definir a plataforma que mais se encontra de acordo com os critérios da ferramenta avaliativa. O método de avaliação escolhido foi a aplicação de um *checklist*, que é uma ferramenta desenvolvida e validada por profissionais da área de usabilidade e ergonomia. A ferramenta facilita o processo de avaliação, onde, o avaliador deve analisar se o SE atende aos critérios estabelecidos pelo método de avaliação.

A ferramenta escolhida para a realização da avaliação, foi o *checklist* TICESE, desenvolvido por Gamez (1998). A ferramenta conta com questões relacionadas a ergonomia e usabilidade de *interfaces* para a aplicação em SE. Ela une conceitos de usabilidade e ergonomia, juntamente com conceitos pedagógicos, buscando assim desenvolver critérios de avaliação que buscam identificar os principais requisitos para o aluno ter uma experiência de uso e aprendizado com a utilização de uma plataforma de ensino. Para a utilização da ferramenta, o avaliador deve ler cada critério e atribuir pesos, definindo uma ordem de importância, na qual, quanto mais importante for o critério de avaliação, maior será seu peso. Isso possibilita a configuração do *checklist* para a avaliação de diferentes tipos de SE, fazendo com que atributos que não tenham relevância na avaliação não interfiram no resultado final.

Para a aplicação do *checklist* TICESE, foram selecionadas duas plataformas de ensino de programação *online* tendo como base os critérios de escolha, na subseção 4.1 Critérios para a escolha das plataformas. Os SEs escolhidos para a aplicação do teste de usabilidade foram a Khan Academy e a Codecademy. Ambas as plataformas apresentam metodologias parecidas, em que é explicado como funciona a linguagem e insere o aluno na resolução de exercícios.

O SE Codecademy, apresenta para o aluno uma explicação da linguagem, como comandos básicos para ir se ambientando e assim, as próximas lições são as resoluções de exercícios. A *interface* de ensino que o aluno utiliza para a resolução dos exercícios se assemelha a uma IDE, passando a sensação para o aluno de estar realmente utilizando uma ferramenta de programação. Sendo que, as mensagens de erros informadas pela plataforma, caso o algoritmo tenha algum erro, é apresentado da mesma forma que se estivesse utilizando uma *interface* de programação. No desenvolvimento do curso, o aluno passa por atividades de perguntas e respostas para avaliar a aprendizagem com relação a termos usuais das linguagens. Um ponto a ser comentado, é o fato da explicação dos exercícios estar na língua

inglesa, podendo alguns ter dificuldade, que podem ser contornados com a utilização de ferramentas de tradução.

A plataforma Khan Academy apresenta uma metodologia parecida com a Codecademy, apresentando o conteúdo ao aluno, expondo a comando básicos da linguagem e inserindo na resolução de exercícios. Todos os cursos se inicia com um vídeo explicativo, que inseri o aluno no contexto do curso. A *interface* que o aluno desenvolve as atividades é mais simples, e procura ser agradável, ao contrário da Codecademy que se assemelha a uma IDE a Khan Academy mantém um formato mais minimalista. As mensagens de erros apresentadas para os usuários, faz o uso de personagens, mostrando mensagens simples que informa o erro. Um diferencial da plataforma, é o fato do SE oferecer o espaço do professor, que não foi avaliado pelo *checklist*, em que pode acompanhar o desenvolvimento dos alunos e oferecer tarefas ou exercícios para complementar no aprendizado.

Na aplicação do teste e levantamento dos resultados, o SE Khan Academy obteve a maior porcentagem de conformidade ergonômica, tendo como base os módulos de avaliação da documentação, da plataforma e avaliação contextual com uma nota de 87,65% contra os 80,28% da Codecademy. A Khan Academy conseguiu a nota mais alta, pelos fatores, mensagens de erros mais simples aos alunos, oferecer material didático ou outras fontes de informações para estar complementando o ensino e atividades em que o aluno tem que usar a imaginação e desenvolver pequenos projetos que posteriormente serão avaliados por outros alunos, criando assim um ambiente de colaboração de troca de conhecimento entre os alunos.

As plataformas de ensino *online*, não devem ser utilizadas para a substituição do professor, essas tecnologias não provocam ainda alterações radicais na estrutura do curso e não mudam a maneira didática de como o professor passa a matéria. Essas plataformas devem ser encaradas como recursos didáticos no processo de ensino-aprendizagem do aluno, pois, ainda estão muito longe de substituírem um professor e de serem usadas em todas as suas possibilidades para uma melhor educação, (KENSKI, 2012).

Para o professor adotar uma plataforma de ensino, buscando servir como ferramenta para auxiliar e complementar o processo de ensino aprendizagem dos alunos, deve estar familiarizado com ferramenta e ter bem claro os objetivos de aprendizagem. São ferramentas de auxílio ao professor em sua prática pedagógica.

O *checklist* procurou avaliar as plataformas com base nos critérios e subcritérios da ferramenta de avaliação, ergonomia e usabilidade, não focando em um grupo de usuários. Pelos resultados, e as metodologias de cada SE, uma indicação de uso desses dois ambientes de ensino, seria iniciar o aluno com a Khan Academy, para ter uma base e fundamentos, por

se tratar de um ambiente de ensino em apresenta uma *interface* mais simples para o usuário, bem como a apresentação das mensagens de erros e a possibilidade do acompanhamento do professor na construção do conhecimento. Tendo como base e fundamentos aprendidos no primeiro SE, o aluno pode ser inserido na utilização do Codecademy, que apresenta uma *interface* e as mensagens de erros, como se o aluno estivesse realmente programando numa IDE, aumentando assim a experiência e a construção do conhecimento por parte do aluno.

Para trabalhos futuros, sugere-se a aplicação de outras técnicas de avaliação de usabilidade e ergonomia, ou, a utilização de dois métodos em conjunto, um para definir qual o SE a ser utilizado pelos alunos e a outra pode ser aplicado um questionário de satisfação. Definindo a plataforma a ser utilizada com uma técnica, deixando os alunos utilizarem o SE proposto pelos critérios da ferramenta de avaliação. Com isso, aplicar um questionário de satisfação, colhendo opiniões, experiências com a utilização das plataformas de ensino *online*, desenvolvendo relatórios de usabilidade e ergonomia, e identificando pontos de melhoria no SE.

REFERÊNCIAS

AMARAL, Sergio Ferreira do. **Projetando a Educação 3.0**. Campinas, 2013. 9 slides, color. Disponível em: <<http://www.lantec.fe.unicamp.br/inova2013/images/trabalhos/palestras/1.pdf>>. Acesso em: 23 abr. 2018.

ANDRES, Daniele Pinto; CYBIS, Walter de Abreu. Um Estudo Teórico sobre as Técnicas de Avaliação de *Software*. In: CONGRESSO ARGENTINO DE CIÊNCIAS DE LA COMPUTACIÓN, 6., 2000, Buenos Aires. **Anais do VI Congresso Argentino de Ciências de la Computación**. Buenos Aires: Cacic, 2000. 11 p.

ANTONINO, Maria Ramone Lemos; FREIRE, Emmanuel Sávio Silva. Avaliação da Usabilidade da *Interface* do Ambiente Virtual Moodle: Uma Visão dos Alunos do Curso de Licenciatura em Computação. In: COMPUTER ON THE BEACH, 9, 2018, Florianópolis. **Anais do Computer on the beach**. Florianópolis: Computer On The Beach, 2018. p. 258 - 267. Disponível em: <<https://siaiap32.univali.br/seer/index.php/acotb/article/view/12758/7241>>. Acesso em: 02 jun. 2018.

BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010. 384 p. (Série SBC, Sociedade Brasileira de Computação). ISBN 9788535234183.

BENYON, David. **Interação humano-computador**. 2. ed. São Paulo: Prentice Hall, 2011. xx, 442 p. ISBN 9788579361098.

CODECADEMY (Org.). **Codecademy**. 2018. Disponível em: <<https://www.codecademy.com/>>. Acesso em: 03 jun. 2018.

CYBIS, Walter; BETIOL, Adriana Holtz.; FAUST, Richard. **Ergonomia e usabilidade**: conhecimentos, métodos e aplicações . 2. ed. rev. e ampl. São Paulo: Novatec, 2010. 422 p. ISBN 9788575222324.

FONSECA, J. J. S. **Metodologia da pesquisa científica**. Fortaleza: UEC, 2002. Apostila.

FRANTZ, Willian L. S.; PONTES, Herleson P.. Um Ambiente de Desenvolvimento Personalizável para o Ensino de Programação. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 34., 2014, Fortaleza. **Anais do Congresso da Sociedade Brasileira de Computação**. Fortaleza: Csbc, 2014. p. 1623 - 1630.

FREIRE, Paulo. **Pedagogia da autonomia**: saberes necessários à prática educativa. São Paulo: Paz e Terra, 2011. 141 p.

GAMEZ, Luciano. **TICESE**: Técnica de Inspeção de Conformidade Ergonômica de *Software* Educacional. 1998. 45 f. Dissertação (Mestrado) - Curso de Escola de Engenharia, Mestrado em Engenharia Humana, Universidade do Minho, Portugal, 1998. Disponível em: <<http://www.labiutil.inf.ufsc.br/estilo/Ticese.htm>>. Acesso em: 10 out. 2018

GODOI, Katia Alexandra de; PADOVANI, Stephania. Instrumentos avaliativos de *software* educativo: uma investigação de sua utilização por professores. In: ESTUDOS EM DESIGN, 19., 2011, Rio de Janeiro. **Anais do 19º estudos em design**. Rio de Janeiro: Estudos em Design, 2011. v. 1, p. 1 - 23.

GOMES, Patrícia. **Codecademy ensina programação de graça**. 2012. Disponível em: <<http://porvir.org/codecademy-ensina-programacao-de-graca-pelo-mundo/20120828/>>. Acesso em: 03 jun. 2018.

IALE, Gilza. **7 Erros mais frequentes nas avaliações de desempenho**. 2015. Disponível em: <<http://www.administradores.com.br/artigos/carreira/7-erros-mais-frequentes-nas-avaliacoes-de-desempenho/88296/>>. Acesso em: 24 nov. 2018.

ISO. ISO 9241-11: **Ergonomic requirements for office work with visual display terminals (VDTs)** - guidance on usability. Genebra, 1998.

JORDAN, Patrick W.. **An Introduction To Usability**. London: Taylor & Francis, 1998. 136 p.

KEMCZINSKI, Avanilde. **Método de Avaliação para ambientes e-learning**. 2005. 173 f. Tese (Doutorado) - Curso de Engenharia de Produção, Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2005. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/101724/224282.pdf?sequence=1>>. Acesso em: 19 set. 2018.

KENSKI, Vani Moreira. **Educação e tecnologias: o novo ritmo da informação**. 8. ed. Campinas: Papyrus, 2012. 141 p. (Papyrus educação.). ISBN 9788530808280.

KHAN ACADEMY (Org.). **Khan Academy**. 2018. Disponível em: <<https://pt.khanacademy.org/>>. Acesso em: 03 jun. 2018.

LIBÂNEO, José Carlos. **Didática**. 2. ed. São Paulo: Cortez, 2013. 288 p.

LIKERT, R. **A technique for the measurement of attitudes**. Archives in Psychology, 140, p. 1- 55, 1932. Disponível em: <https://legacy.voteview.com/pdf/Likert_1932.pdf>. Acesso em: 03 jun. 2018.

LIMA, Adilson da Silva. **UML 2.3: do requisito à solução**. São Paulo: Érica, 2011. 368 p.

LOPES, Anita; GARCIA, Guto. **Introdução à Programação: 500 Algoritmos Resolvidos**. Rio de Janeiro: Elsevier, 2002. 469 p.

MANAZO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 26. ed. São Paulo: Érica, 2012. 328 p.

MEDINA, Marcos; FERTIG, Cristina. **Algoritmos e Programação: Teoria e Prática**. 2. ed. São Paulo: Novatec Editora, 2006. 384 p.

MORAIS, Rommel Xenofonte Teles de. **Software Educacional: A importância de sua avaliação e do seu uso nas salas de aula.** 2003. 52 f. TCC (Graduação) - Curso de Bacharel em Ciência da Computação, Faculdade Lourenço Filho, Fortaleza, 2003.

MOTA, Marcelle Pereira; PEREIRA, Lis W. Kanashiro; FAVERO, Eloi Luiz. JAVATOOL: uma ferramenta para ensino de programação. In: Workshop sobre educação em computação, 28., 2008, Belém do Pará. **Anais do Workshop sobre Educação em Computação.** Belém do Pará: Sbc, 2008. p. 127 - 136.

MUSSOI, Eunice Maria; FLORES, Maria Lucia Pozzatti; BEHAR, Patricia Alejandra. Avaliação de Objetos de Aprendizagem. In: CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 1., 2010, Santiago de Chile. **Anais do Congresso Iberoamericano de Informática Educativa.** Santiago de Chile: Jaime Sánchez, 2010. v. 1, p. 122 - 126.

NIELSEN, Jakob. **10 Usability Heuristics for User Interface Design.** 1995. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 13 maio 2018.

_____, Jakob; LORANGER, Hoa. **Usabilidade na web: projetando websites com qualidade.** Rio de Janeiro: Elsevier, Campus, 2007. 406 p. ISBN 9788535221909.

PAES, Rodrigo de Barros et al. Ferramenta para Avaliação de Aprendizado de Alunos em Programação de Computadores. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2., 2018, Limeira. **Anais do II Congresso Brasileiro de Informática na Educação.** Limeira: Wcbie 2013, 2013. p. 203 - 212.

PRATES, Raquel Oliveira; BARBOSA, Simone Diniz Junqueira. Avaliação de interfaces de usuário—conceitos e métodos. In: **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Capítulo.** 2003. p. 28

RAABE, André Luís Alice; SILVA, Júlia Marques Carvalho da. Um Ambiente para Atendimento às Dificuldades de Aprendizagem de Algoritmos. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 25., 2005, São Leopoldo. **Anais do Congresso da Sociedade Brasileira de Computação.** São Leopoldo: Csbcc, 2005. p. 2326 - 2337.

RIBAS, Armando Cardoso; PEREIRA, Alice Theresinha Cybis. Usabilidade e os critérios de avaliação para ambiente virtual de ensino-aprendizagem do curso letraslibras enfocando a cultura surda. **Revista Ação Ergonômica**, v. 5, n. 1, 2011.

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. **Design de interação: além da interação homem-computador.** 3º. ed. Porto Alegre: Bookman, 2013. xvi, 585 p. ISBN 9788582600061.

SANTA ROSA, José Guilherme; MORAES, Anamaria de. **Avaliação e projeto no design de interfaces.** Rio de Janeiro: 2AB, 2010. 223 p. ISBN 9788586695469.

SBC (Brasil). Sociedade Brasileira de Computação (Comp.). **Interação Humano-Computador.** 2018. Disponível em: <<http://www.sbc.org.br/14-comissoes/390-interacao-humano-computador>>. Acesso em: 05 maio 2018.

SIGNIFICADOS. **Significados, Conceitos e Definições.** Disponível em: <<https://www.significados.com.br>>. Acesso em: 19 nov. 2018.

SILVA, Raphael Salviano da et al. Avaliação de *Software* Educativo: a complexidade de escolher uma abordagem adequada. In: CONGRESSO REGIONAL SOBRE TECNOLOGIAS NA EDUCAÇÃO, 1., 2016, Natal. **Anais do 1º Congresso Regional sobre Tecnologias na Educação.** Natal: Ctrl+e, 2016. p. 116 - 126. Disponível em: <http://ceur-ws.org/Vol-1667/CtrlE_2016_AC_paper_26.pdf>. Acesso em: 18 out. 2018.

SILVA, Williane Rodrigues Almeida; DE PONTES, Romario Ferreira; PADILHA, Thereza Patricia Pereira. Um Estudo Sobre Avaliação de *Softwares* Educativos para o Ensino de Programação. **SIED: EnPED-Simpósio Internacional de Educação a Distância e Encontro de Pesquisadores em Educação a Distância**, 2016. Disponível em: <<http://www.sied-enped2014.ead.ufscar.br/ojs/index.php/2016/article/view/1681/855>>. Acesso em: 23 jun. 2018.

SIROTHEAU, Silvério et al. Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de auto avaliação. In: SBIE, 22., 2011, Aracaju. **Anais da Sociedade Brasileira de Informática na Educação.** Aracaju: Sbie, 2011. v. 17, p. 750 - 759.

TRIVIÑOS, A. N. S. **Introdução à pesquisa em ciências sociais:** a pesquisa qualitativa em educação. São Paulo: Atlas, 1987.

APÊNDICE A – NOTAS DAS AVALIAÇÕES DAS PLATAFORMAS

No Quadro 14, contém os resultados de cada critério separado por sua respectiva plataforma. Na coluna critérios/subcritérios é identificado o critério de avaliação. Nas colunas Khan Academy e Codecademy, é apresentado as respostas obtidas na avaliação da plataforma. Para a leitura das resposta: 1-S/1, onde, o 1 (um) representa o número do requisito, o S é a nota obtida, e o /1, é o peso definido pelo avaliador. Quando aparece 1-0,significa que o requisito possui peso 0 (zero), na qual, o avaliador não precisa verificar e avaliar esse requisito. Vale lembrar que está forma de apresentação, Quadro 14, é apenas uma forma para demonstrar os valores que cada plataforma recebeu na avaliação.

Quadro 14 – Notas das avaliações das plataformas

Critérios/Subcritérios	Khan Academy	Codecademy
Identificação do produto	1-S/1; 2-0; 3-S/0; 4-S/0; 5-S/1; 6-S/1	1-S/1; 2-0; 3-S/0; 4-S/0; 5-S/1; 6-S/1
Identificação dos Pré-Requisitos Técnicos	1-0; 2-0; 3-0	1-0; 2-0; 3-0
Identificação dos Pré-Requisitos Pedagógicos	1-0; 2-0; 3-0	1-0; 2-0; 3-0
Identificação dos Objetivos Pedagógicos	1-S/1,5; 2-S/1,5; 3-0; 4-0; 5-P/1,5; 6-0	1-S/1,5; 2-S/1,5 3-0 4-0 5-P/1,5 6-0
Documentação: Presteza	1-0; 2-0; 3-0; 4-S/1,5; 5-S/1,5; 6-0; 7-0; 8-0; 9-0; 10-0; 11-0; 12-0	1-0; 2-0; 3-0; 4-S/1,5; 5-P/1,5; 6-0; 7-0; 8-0; 9-0; 10-0; 11-0; 12-0
Documentação: Legibilidade	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-S/1,5; 5-P/1,5	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-P/1,5; 5-P/1,5
Documentação: Agrupamento de Itens	1-S/1; 2-S/1; 3-S/1; 4-S/1	1-S/1; 2-S/1; 3-S/1; 4-S/1
Documentação: Densidade Informacional	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-S/1,5	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-S/1,5
Documentação: Consistência	1-0; 2-S/1; 3-0	1-0 2-S/1 3-0
Documentação: Significado Códigos e Deno.	1-S/1,5; 2-P/1; 3-S/1; 4-S/1; 5-P/1	1-S/1,5; 2-P/1; 3-S/1; 4-P/1; 5-P/1
Condução: Presteza	1-S/1,5; 2-0; 3-S/1; 4-P/1,5; 5-S/1,5; 6-S/1,5; 7-S/1,5; 8-S/1,5; 9S-/1; 10-0; 11-0	1-S/1,5; 2-0; 3-N/1; 4-N/1,5; 5-S/1,5; 6-S/1,5; 7-S/1,5; 8-S/1,5; 9-S/1; 10-0; 11-0
Condução: Qualidade da Opção de Ajuda	1-S/1,5; 2-0; 3-S/1; 4-S/1,5; 5-P/1; 6-S/1,5; 7-P/1; 8-P/1,5; 9-0; 10-S/1,5	1-S/1,5; 2-0; 3-S/1; 4-S/1,5; 5-N/1; 6-P/1,5; 7-P/1; 8-N/1,5; 9-0; 10-S/1,5
Condução: Legibilidade	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-P/1; 5-P/1; 6-S/1; 7-S/1; 8-S/1; 9-P/1; 10-P/1; 11-S/1; 12-S/1; 13-0; 14-S/1; 15-0; 16-S/1; 17-0; 18-0; 19-0	1-S/1,5; 2-S/1,5; 3-P/1,5; 4-P/1; 5-P/1; 6-S/1; 7-S/1; 8-S/1; 9-P/1; 10-S/1; 11-P/1; 12-P/1; 13-0; 14-S/1; 15-0; 16-S/1; 17-0; 18-0; 19-0
Condução: Agrupamento e Distinção por For.	1-S/1,5; 2-P/1; 3-S/1,5; 4-S/1; 5-0; 6-0; 7-S/1; 8-0; 9-S/1	1-S/1,5; 2-S/1; 3-S/1,5; 4-S/1; 5-0; 6-0; 7-S/1; 8-0; 9-P/1

[continua]

CrITÉRIOS/SubcrITÉRIOS	Khan Academy	Codecademy
Condução: Agrupamento e Distinção por Loc.	1-S/1; 2-S/1,5; 3S-/1,5; 4-S/1,5; 5-S/1,5; 6-S/1; 7-P/1; 8-0; 9-0	1-P/1; 2-N/1,5; 3-S/1,5; 4-S/1,5; 5-P/1,5; 6S-/1; 7-P/1; 8-0; 9-0
Condução: Feedback Imediato	1-S/1,5; 2-S/1,5; 3-S/1; 4-0; 5-P/1; 6-S/1; 7-S/1; 8-N/1; 9-S/1,5; 10-0; 11-S/1	1-S/1,5; 2-P/1,5; 3-S/1; 4-0; 5-S/1; 6-P/1; 7-P/1; 8-N/1; 9-S/1,5; 10-0; 11-S/1
Carga de trabalho: Carga Informacional	1-S/1,5; 2-P/1; 3-S/1; 4-S/1; 5-P/1; 6-P/1; 7-S/1; 8-S/1,5	1-S/1,5; 2-S/1; 3-S/1; 4-S/1; 5-P/1; 6-S/1; 7-S/1; 8-P/1,5
Carga de trabalho: Concisão	1-S/1; 2-S/1; 3-0	1-S/1; 2-S/1; 3-0
Carga de trabalho: Ações Mínimas	1-P/1,5; 2-0; 3-0; 4-P/1; 5-P/1; 6-0	1-P/1,5; 2-0; 3-0; 4S-/1; 5-P/1; 6-0
Carga de trabalho: Densidade Informacional	1-S/1,5; 2P-/1; 3-0; 4-0; 5-P/1; 6-S/1,5; 7-0; 8-0; 9-0; 10-0	1-P/1,5; 2-P/1; 3-0; 4-0; 5-S/1; 6-S/1,5; 7-0; 8-0; 9-0; 10-0
Recursos de apoio à compreensão de conteúdo	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-P/1,5; 5-P/1; 6-S/1,5; 7-S/1; 8-S/1,5; 9-P/1; 10-P/1; 11-0; 12-S/1,5; 13-S/1,5; 14-S/1; 15-0; 16-S/1,5; 17-S/1,5; 18-S/1,5;	1-P/1,5; 2-P/1,5; 3-P/1,5; 4-N/1,5; 5-N/1; 6-P/1,5; 7-N/1; 8-P/1,5; 9-S/1; 10-S/1; 11-0; 12-S/1,5; 13-P/1,5; 14-P/1; 15-0; 16-P/1,5; 17-P/1,5; 18-N/1,5;
Adaptabilidade: Flexibilidade	1-N/1,5; 2-0; 3-0; 4-P/1; 5-0; 6-0; 7-0; 8-0; 9-0; 10-P/1; 11-0	1-N/1,5; 2-0; 3-0; 4-N/1; 5-0; 6-0; 7-0; 8-0; 9-0; 10-N/1; 11-0
Adaptabilidade: Consideração da Experiência	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-N/1; 5-P/1; 6-P/1; 7-S/1; 8-0; 9-0; 10-0	1-S/1,5; 2-S/1,5; 3-S/1,5; 4-N/1; 5-P/1; 6-S/1; 7-P/1; 8-0; 9-0; 10-0
Controle Explícito: Ações Explícitas do Utili.	1-S/1; 2-P/1,5; 3-P/1; 4-0; 5-S/1	1-S/1; 2-P/1,5; 3-S/1; 4-0; 5-S/1
Controle Explícito: Controle do utilizador	1-S/1; 2-0; 3-0; 4-0; 5-P/1; 6-P/1; 7-S/1,5; 8-0	1-S/1; 2-0; 3-0; 4-0; 5-S/1; 6-P/1; 7-S/1,5; 8-0
Gestão de Erros: Correção de Erros	1-S/1; 2-P/1,5; 3-S/1,5; 4-S/1,5; 5-S/1; 6-0; 7-S/1,5; 8-0; 9-S/1,5; 10-S/1; 11-S/1;	1-S/1; 2-S/1,5; 3-P/1,5; 4-S/1,5; 5-N/1; 6-0; 7-N/1,5; 8-0; 9-S/1,5; 10-S/1; 11-P/1;
Gestão de Erros: Qualidade das Mensagem E.	1-S/1,5; 2-S/1,5; 3-S/1; 4-S/1; 5-S/1,5; 6-S/1; 7-S/1; 8-S/1,5; 9-P/1; 10-0; 11-S/1;	1-S/1,5; 2-P/1,5; 3-N/1; 4-S/1; 5-S/1,5; 6-S/1; 7-S/1; 8-S/1,5; 9-N/1; 10-0; 11-S/1;
Gestão de Erros: Proteção Contra Erros	1-S/1; 2-0; 3P-/1; 4-0; 5-0; 6-S/1; 7-0; 8-S/1,5; 9-S/1; 10-0;	1S-/1; 2-0; 3-N/1; 4-0; 5-0; 6-N/1; 7-0; 8-S/1,5; 9-S/1; 10-0;
Avaliação de Aprendizado	1-S/1,5; 2-S/1,5; 3-S/1; 4-S/1,5; 5-S/1,5; 6-S/1; 7-S/1	1-P/1,5; 2-P/1,5; 3-P/1; 4-S/1,5; 5-S/1,5; 6-P/1; 7-S/1
Homogeneidade	1-S/1; 2-S/1; 3-P/1; 4-S/1; 5-S/1; 6-P/1;	1-S/1; 2-S/1; 3-S/1; 4-P/1; 5-S/1; 6-S/1;
Significado de Códigos e Denominações	1-S/1,5; 2-S/1; 3-S/1,5; 4-S/1; 5-P/1; 6-S/1,5; 7-S/1; 8-S/1,5; 9-S/1; 10-S/1; 11-S/1;	1-S/1,5; 2-P/1; 3-S/1,5; 4-S/1; 5-S/1; 6-S/1,5; 7-S/1; 8-S/1,5; 9-S/1; 10-P/1; 11-S/1;
Compatibilidade	1-0; 2-S/1; 3-S/1;	1-0; 2-S/1; 3-S/1;
Adequabilidade	1-S/1; 2-P/1; 3-S/1; 4-S/1; 5-P/1; 6-S/1,5; 7-0; 8-S/1; 9P-/1; 10-P/1; 11-S/1,5; 12-S/1; 13-0; 14-0; 15-S/1,5; 16-S/1	1-S/1; 2-P/1; 3-P/1; 4-P/1; 5-P/1; 6-S/1,5; 7-0; 8-S/1; 9-S/1; 10-P/1; 11-S/1,5; 12-S/1; 13-0; 14-0; 15-S/1,5; 16-S/1

Fonte: Desenvolvido pelo autor, 2018.

APÊNDICE B – NOTAS GERAIS DAS PLATAFORMA E DOS CRITÉRIOS

Quadro 15 – Notas gerais das plataformas e dos critérios

Critérios	Nota Khan Academy	Nota Codecademy	Nota geral por categoria Khan Academy	Nota geral por categoria Codecademy	Nota geral Khan Academy	Nota geral Codecademy
Avaliação da Documentação					87,65%	80,28%
Dados de identificação	100%	87,50%	94,45%	84%		
Qualidade da informação impressa	88,90%	80,50%				
Avaliação da Plataforma						
Condução	86,30%	81%	82,31%	74,10%		
Carga de trabalho	64,10%	72,10%				
Recursos de apoio a compreensão dos conteúdos	87,20%	50%				
Adaptabilidade	40,50%	29,80%				
Controle explícito	73,60%	84,72%				
Gestão de erros	92,10%	66,50%				
Avaliação de aprendizagem	100%	72,22%				
Homogeneidade	83,33%	91,66%				
Significado de códigos e denominações	96%	92%				
Compatibilidade	100%	100%				
Avaliação Contextual						
Adequabilidade	-	-	86,20%	82,75%		

Fonte: Desenvolvido pelo autor, 2018.



ANEXO A – I MÓDULO DE CLASSIFICAÇÃO

1. Classificação da Modalidade: Assinale a modalidade do *software*:

- ☐ Simulador
- ☐ Tutorial
- ☐ Exercício e prática
- ☐ Apresentação
- ☐ Hipermissão
- ☐ Jogos Pedagógicos
- ☐ Ambiente de descoberta e aprendizagem
- ☐ Outros

2. Identificação da abordagem pedagógica

O *software* traz explicitamente a abordagem pedagógica na qual foi formulado e qual é esta abordagem?

3. Complexidade Cognitiva

1) O *software* evoca conhecimentos prévios e específicos, necessários para a compreensão de seu conteúdo?

2) O *software* enfatiza a aquisição do conhecimento, aumentando gradualmente a quantidade de informações apresentadas?

3) Assinale a (s) alternativa (s) que mais se assemelha (m) ao tipo de *software* em análise.

a) O *software* evita comportamentos do tipo tentativa e erro, fornecendo recursos que apoiem processos de raciocínio e que levem à decisões conceitualmente baseadas?

b) O *software* trabalha a valorização do erro possibilitando o aprendizado com a experiência?

4) Assinale a (s) alternativa (s) que mais se assemelha (m) ao tipo de *software* em análise.

a) O *software* solicita apenas a lembrança de ideias, materiais ou fenômenos memorizados.

b) O *software* não apenas solicita a lembrança de ideias, mas solicita que o utilizador compreenda os conteúdos transmitidos, e que faça algum uso dos materiais ou ideias nele abrangidos, evocando algum conhecimento já adquirido.

c) O *software* não apenas evoca a lembrança e a compreensão dos conteúdos, mas oferece-lhe a possibilidade de aplicar estes conhecimentos através do uso de abstrações em situações particulares e concretas que ajudam na memorização e fixação dos conceitos.

d) O *software*, para além de proporcionar a aplicação dos conhecimentos, oferece ao aluno a oportunidade de desenvolver um comportamento criador, através de exercícios que solicitem a união de elementos e partes, combinando-os para que constituam uma configuração ou estrutura.

e) Para além de desenvolver um comportamento criador, o *software* estimula o julgamento qualitativo e/ou quantitativo acerca do valor de determinadas ideias, trabalhos, soluções, métodos e materiais.



ANEXO B – II MÓDULO DE AVALIAÇÃO

Atribuição de peso as questões, implica em identificar as questões não aplicáveis ao *software*, e classificá-las segundo a sua ordem de importância:

- Para as questões *não se aplica*, atribuir peso 0 (zero);
- Para questões com resposta *muito importante*, atribuir peso 1,5 (um e meio);
- Para questões com respostas *importante*, atribuir peso 1 (um).

Responda as questões a seguir, conforme indica a legenda:

Legenda: S = sim P = parcialmente N = Não

Identificação do Produto	S	P	N	PESO
Na documentação e/ou na versão <i>online</i> do produto, estão identificados:				
1) O nome do produto?				
2) A versão do produto?				
3) O nome do produtor?				
4) A data de fabricação?				
5) Assistência técnica?				
6) A apresentação geral do produto?				

Identificação dos Pré-Requisitos Técnicos	S	P	N	PESO
Na documentação e/ou na versão <i>online</i> do produto, estão identificados:				
1) Os requisitos de <i>hardware</i> necessários para pôr o produto em funcionamento?				
2) Dados sobre a exigência de conhecimentos específicos de um dado sistema operacional?				
3) Dados sobre a necessidade em adquirir treinamento técnico específico para que possa operar, alterar ou personalizar o <i>software</i> ?				

Identificação dos Pré-Requisitos Pedagógicos	S	P	N	PESO
Na documentação e na versão <i>online</i> do produto, estão descritos:				
1) Os requisitos de <i>software</i> necessários para pôr o produto em funcionamento?				
2) As exigência de conhecimentos prévios específicos da área técnica em apresentação para o uso do <i>software</i> ?				
3) Informações sobre a necessidade em adquirir treinamento pedagógico específico para que possa operar, alterar ou personalizar o <i>software</i> ?				

Identificação dos Objetivos Pedagógicos	S	P	N	PESO
1) A documentação e a versão <i>online</i> do produto trazem a:				
2) Descrição dos objetivos gerais e específicos a que se destinam o <i>software</i> ?				
3) Identificação das principais atividades a serem realizadas com o uso do produto, de modo que se consiga obter maior rentabilidade do produto?				
4) Identificação da faixa etária a que se destina o produto?				
5) Identificação do nível escolar sugerido para a utilização do produto?				
6) Identificação das habilidades que o <i>software</i> pretende desenvolver?				
7) Alguma sugestão para trabalhos individuais ou em grupos correlacionados com as atividades desenvolvidas no <i>software</i> ?				

Documentação: Presteza	S	P	N	PESO
Os manuais que acompanham a documentação do <i>software</i> , possuem:				
1) Títulos coerentes nos capítulos?				
2) Sumário completo?				
3) Índice Remissivo?				
4) Glossário?				
5) Indicação de Ajuda <i>Online</i> ?				
6) Guia de instalação?				
7) Código de erros?				



8) Informações a respeito do uso de senhas de segurança para entrada no sistema?				
9) Informações sobre o estilo e funcionamento de <i>interface</i> com o utilizador?				
10) Informações sobre as teclas de atalho e as teclas de função disponíveis?				
11) Os exemplos [(textos, fotografias, desenhos, representações simbólicas (notação)] são coerentes e estão dentro do contexto?				
12) Há algum tipo de organização segundo uma sequência em que as informações se complementam na evolução entre os capítulos?				

Documentação: Legibilidade	S	P	N	PESO
1) O tamanho das letras é legível nas descrições textuais?				
2) As cores utilizadas são adequadas e facilitam a leitura das informações?				
3) A redação dos textos é isenta de erros ortográficos, gramaticais e pontuação?				
4) A redação e o estilo do texto estão bem escritos, clara, e de fácil compreensão?				
5) Evita-se o uso de palavras estrangeiras que possam dificultar a compreensão do texto?				

Documentação: Agrupamento de Itens	S	P	N	PESO
1) A informação é apresentada em tópicos organizados por funções e comandos?				
2) A informação é apresentada em tópicos organizados por objetivos?				
3) As informações possuem boa organização entre os itens, divididas em capítulos unidades ou secções?				
4) Existe alguma divisão na apresentação da informação entre módulos de cunho teórico e prático, e esta divisão está balanceada?				

Documentação: Densidade Informacional	S	P	N	PESO
1) A densidade de caráter técnico da informação é adequada?				
2) A densidade de leitura é adequada?				
3) As informações estão bem distribuídas na documentação e evitam a poluição visual?				
4) A utilização de cores é feita de maneira adequada que não provoca poluição visual?				

Documentação: Consistência	S	P	N	PESO
1) Há consistência na numeração das páginas apresentadas no sumário?				
2) Há consistência entre as informações descritas na documentação e a operação do produto?				
3) Todas as páginas estão presentes e postas na sequência correta?				

Documentação: Significado dos Códigos e Denominações	S	P	N	PESO
1) As denominações dos títulos correspondem aos assuntos apresentados?				
2) O vocabulário utilizado nos títulos é familiar ao utilizador e evita palavras difíceis?				
3) Os títulos das páginas são explicativos, refletindo a natureza da escolha a ser feita?				
4) Na ocorrência de abreviaturas, há uma descrição por extenso do seu significado?				
5) As abreviaturas são facilmente distinguíveis umas das outras, evitando confusões geradas por similaridade?				

Condução: Presteza	S	P	N	PESO
1) O <i>software</i> dispõe de um glossário para auxiliar o utilizador na compreensão de termos técnicos?				
2) O <i>software</i> permite a impressão das informações desejadas?				
3) O <i>software</i> utiliza recursos do tipo hipertexto providenciando links apropriados que facilitem a compreensão dos conteúdos?				
4) A apresentação do tipo hipertexto tem boa condução, de forma que o utilizador possa se localizar bem enquanto navega no programa?				
5) O <i>software</i> informa ao utilizador os resultados do estado da ação, de forma que ele possa acompanhar a evolução do processamento da informação, usando recursos como por exemplo, ampulhetas, relógio e/ou barra de progressão?				

Adaptado de Gamez (1998) pelo autor.



6) O utilizador encontra disponível no ecrã as informações necessárias para executar suas ações e efetuar as operações requeridas pelo <i>software</i> ?				
7) Na ocorrência de erros do sistema, o utilizador tem o acesso facilitado as informações necessárias ao diagnóstico e solução do problema?				
8) As mensagens que conduzem o utilizador para uma determinada ação são sempre afirmativas e na voz ativa?				
9) O <i>software</i> apresenta títulos nas caixas de diálogo, formulários, campos de entrada de dados, janelas, etc., e estes estão no alto, centrados ou justificados à esquerda?				
10) As opções que comandam a apresentação da abertura de outras opções de diálogo, apresentam em seus rótulos sinais indicadores da continuidade do diálogo? (tais como "... ou o sinal ">").				
11) No caso em que são apresentadas tabelas ao longo do <i>software</i> , estas possuem cabeçalho para linhas e colunas apresentadas de maneira distinguíveis dos restantes dados (quanto a cor, fonte ou tipo de letra)?				

Condução: Qualidade da Opções de Ajuda	S	P	N	PESO
1) O <i>software</i> disponibiliza ao utilizador a opção, <i>online</i> , de menu “Ajuda”?				
2) A ajuda <i>online</i> é consistente com a documentação em papel no que se refere ao conteúdo?				
3) A ajuda <i>online</i> é consistente com todas as outras dentro dela mesma?				
4) Os dispositivos de ajuda abrangem a totalidade do sistema?				
5) O acionamento da opção de ajuda, está estruturado no contexto da tarefa e da transação corrente?				
6) O sistema de ajuda funciona de forma exclusiva não interrompendo a continuação na execução do sistema?				
7) O sistema apresenta diferentes formas de acesso aos conteúdos de ajuda?				
8) O sistema de ajuda utiliza-se de princípios de hipertexto, permitindo ao utilizador expandir tópicos por palavras chave?				
9) O <i>software</i> disponibiliza ao utilizador bolhas de ajuda para informar sobre a função de um botão, menu ou caixa de diálogo?				
10) Nas caixas de mensagens de erro, o botão de comando “ajuda” está sempre presente?				

Condução: Legibilidade	S	P	N	PESO
1) Os conteúdos apresentados estão livres de equívocos conceituais?				
2) As redações das informações textuais estão corretas, livres de erros gramaticais e de pontuação?				
3) O estilo literário do texto favorece a compreensão dos conteúdos?				
4) O vocabulário utilizado é de fácil compreensão ao público alvo sugerido pelo fabricante?				
5) O vocabulário utilizado propõe uma interpretação específica no significado dos termos e dos conteúdos que se pretende transmitir, sem gerar problemas de interpretações errôneas?				
6) A apresentação do texto, nomeadamente o tipo e tamanho das letras, é de fácil legibilidade?				
7) É evitado o uso exclusivo de maiúsculas nos textos apresentados?				
8) O uso de recursos de estilo como sublinhado, negrito, itálico, é feito de maneira ponderada e não atrapalha a legibilidade do texto?				
9) Os parágrafos de texto são separados por uma linha em branco, pelo menos, e possuem margens bem definidas?				
10) É evitado o uso de abreviaturas nos menus, opções de menu, título das caixas de diálogo, e mostradores de dados?				
11) O uso de cores favorece a legibilidade do programa?				
12) A cor do fundo em relação à cor da letra permite uma boa leitura?				
13) O texto apresentado nas caixas de opções de menu, apresentam boa legibilidade, ou seja, estão adequadamente posicionados e separados das bordas neste tipo de caixa?				
14) Os ícones são legíveis e representativos de suas funções?				
15) Os títulos de caixas de diálogo evitam a utilização apenas de letras maiúsculas ?				



16) Os objetos de interação (botões, campos de edição, etc.) disponíveis nas caixas de diálogo encontram-se alinhados vertical e horizontalmente?				
17) As áreas livres são usadas para separar grupos lógicos em vez de tê-los todos de um só lado do ecrã, caixa ou janela?				
18) As informações codificadas através das cores apresentam uma codificação adicional redundante?				
19) Dados numéricos que se alterem rapidamente são apresentados analogicamente?				

Condução: Agrupamento e Distinção por Formato	S	P	N	PESO
1) O <i>software</i> apresenta uma distinção visual clara de áreas que possuem diferentes funções? (área de comandos, área de mensagens, etc.)				
2) Os dados críticos e que requeiram atenção imediata são diferenciados através do emprego de sinais sonoros ou diferenciados através do uso de cores conspícuas para alertar os utilizadores em relação às situações anormais?				
3) Em situações em que é exigida atenção especial do utilizador, as mensagens de alerta e de aviso são apresentadas de maneira distinta das demais?				
4) São empregues recursos de estilo, como itálico, negrito, sublinhado ou diferentes fontes para salientar palavras ou noções importantes na apresentação de textos?				
5) No caso de apresentação de tabelas, os cabeçalhos estão diferenciados através do emprego de cores diferentes, letras maiores ou sublinhadas?				
6) Quando as opções de menu apresentam-se indisponíveis no momento, o sistema mostra-as de forma diferenciada visualmente?				
7) Os rótulos dos mostradores de dados são visualmente diferentes dos dados em si aos quais estão associados?				
8) No caso em que o <i>software</i> apresenta ecrã de consulta, os diferentes elementos do ecrã de consulta (dados, comandos e instruções) são visualmente distintos uns dos outros?				
9) Quando várias opções ou ações são possíveis, a mais provável ou mais lógica é apresentada num formato que a distingue das outras, tal como uma borda circulando um botão (a opção <i>default</i>)?				

Condução: Agrupamento e Distinção por Localização	S	P	N	PESO
1) Existe alguma divisão na apresentação da informação entre módulos de cunho teórico e prático, e esta divisão está balanceada?				
2) No caso em que os conceitos teóricos são separados dos exercícios práticos, existe a facilidade do utilizador navegar facilmente entre a parte prática e a teórica do <i>software</i> ?				
3) As informações possuem boa organização entre os itens, divididas em capítulos unidades ou secções?				
4) A informação é apresentada em tópicos organizados por funções e comandos?				
5) A informação é apresentada em tópicos organizados por objetivos do utilizador?				
6) Os itens de menus estão organizados (agrupados) hierarquicamente segundo uma ordem lógica?				
7) As opções de menu estão ordenadas de forma lógica, agrupadas de forma a facilitar a interação com o utilizador?				
8) Nos agrupamentos de dados, os itens estão organizados espacialmente segundo um critério lógico e facilitador (como por exemplo, agrupamento por frequência de uso)?				
9) Em caso de presença de listas de seleção, as opções da lista estão organizadas segundo alguma ordem lógica e coerente?				

Condução: Feedback Imediato	S	P	N	PESO
1) O sistema emite algum <i>feedback</i> sonoro mediante respostas inadequadas do utilizador na resolução de exercícios?				
2) O sistema emite <i>feedback</i> encorajador, variado e isento de carga negativa mediante as respostas inadequadas do utilizador nas situações problemas??				
3) O sistema fornece informações sobre o progresso do processamento da informação?				
4) O sistema fornece informações sobre o tempo total requerido ao processamento da informação quando este é demorado?				
5) O sistema fornece " <i>feedback</i> " imediato de todas as entradas de dados dos				



utilizadores? (incluindo dados sigilosos, que neste caso devem produzir um <i>feedback</i> perceptível, como por exemplo o símbolo *)				
6) O tempo de resposta do produto é adequado à operação levando em consideração a complexidade, a abrangência e o volume dos dados manipulados?				
7) Nas operações interativas o tempo de resposta é adequado e homogêneo em todas as operações? (carregamento de ecrã, imagens, dados, etc.).				
8) Caso o utilizador interrompa um processamento de dados, o sistema mostrar uma mensagem garantindo-lhe que o sistema voltou ao seu estado prévio?				
9) Quando o processamento da informação é concluído, o sistema apresenta uma mensagem que informa sobre o sucesso ou fracasso da operação?				
10) Durante a tarefa de impressão, o sistema fornece informações sobre o estado destas?				
11) O sistema fornece " <i>feedback</i> " sobre as mudanças de atributos dos objetos de interação, ou seja, ao selecionar um botão o ícone correspondente a esta opção apresenta mudança de estado, entre acionado e não acionado?).				

Carga de Trabalho: Carga Informacional	S	P	N	PESO
1) A carga informacional apresentada é equilibrada e está bem distribuída em unidades de informação?				
1. Os conteúdos teóricos apresentados são objetivos?				
1) Os exercícios práticos propostos objetivos?				
2) É exigido do aluno um nível adequado de atenção e concentração?				
3) O <i>software</i> utiliza adequadamente estímulos para fixação de conceitos e habilidades cognitivas?				
4) A carga de informação apresentada está adequada aos utilizadores previamente definidos?				
5) A carga de informação apresentada está adequada à disciplina de ensino?				
6) O carga de informação apresentada possui bom equilíbrio entre a teoria e a prática?				

Carga de Trabalho: Concisão	S	P	N	PESO
1) A <i>interface</i> do <i>software</i> apresenta nomes concisos nas opções de menu, nas janelas, caixas de diálogo para serem lembrados facilmente??				
2) No caso em que é requerido do utilizador a utilização de senhas, que o mesmo deve memorizar, estas são sempre menores do que 4 ou 5 caracteres?				
3) O sistema oferece valores " <i>default</i> " para acelerar a entrada de dados?				

Carga de Trabalho: Ações Mínimas	S	P	N	PESO
1) Somente as informações necessárias e utilizáveis são apresentadas?				
2) O número de passos necessários para se fazer uma seleção em menu é minimizado?				
3) A <i>interface</i> possibilita repetir a entrada de dados quando estas podem ser reaproveitáveis?				
4) A <i>interface</i> possibilita que uma unidade de medida quando associada a um campo, a inclusão da unidade como parte do campo de dados.				
5) Para entrada de dados, valores <i>default</i> são exibidos nos campos apropriados?				
6) Quando várias páginas estiverem envolvidas, o sistema possibilita ir diretamente para uma página sem ter que passar pelas intermediárias?				

Carga de Trabalho: Densidade informacional	S	P	N	PESO
1) As informações estão bem distribuídas na ecrã e evitam a poluição visual?				
2) Todas as informações contidas na ecrã são imprescindíveis para guiar ou auxiliar o utilizador na compreensão dos conteúdos pedagógicos?				
3) A densidade global das janelas é reduzida?				
4) Em qualquer transação é apenas fornecido ao utilizador os dados necessários e diretamente usáveis?				
5) Tanto a Barra de Menu como as Opções de Menu apresentam apenas as opções necessárias para atingir os fins específicos?				
6) O sistema evita apresentar um grande número de janelas que possam desconcentrar ou sobrecarregar a memória do utilizador?				



7) O sistema minimiza a necessidade do utilizador lembrar dados exatos de um ecrã ao outro?				
8) No caso em que o <i>software</i> usa o emprego de listas de seleção e combinação, apresenta-as a uma altura correspondente a um máximo de nove linhas?				
9) O sistema promove computação automática de dados derivados, de forma que utilizador não tenha que calcular e entrar com elementos derivados de dados já acessíveis ao computador?				
10) As opções de codificação por cores são limitadas em número?				

Recursos de apoio à compreensão dos conteúdos	S	P	N	PESO
1) O <i>software</i> possui recursos motivacionais para despertar e manter a atenção do utilizador ao longo de sua interação?				
2) Os recursos motivacionais utilizados permanecem interessantes ao longo do tempo, sem tornarem-se aborrecidos através de repetições constantes?				
3) Recursos multimídia são utilizados de maneira moderada, sem provocar a distração do aluno no que se refere ao principal foco a ter em atenção na ecrã?				
4) Recursos sonoros são bem explorados, e utilizados pertinentemente?				
5) Os recursos sonoros empregues contribuem para a motivação e compreensão dos conteúdos?				
6) Imagens, desenhos, gráficos, etc. são utilizados pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
7) Animações são utilizadas pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
8) Exercícios de aplicação e resolução de problemas são utilizadas pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
9) Os exercícios de aplicação e soluções de problemas evitam aborrecimento, constrangimento ou desânimo por parte do aluno?				
10) Os exercícios de aplicação e soluções de problemas evitam a sensação de frustração que pode ser causada por dificuldade de manuseio do programa?				
11) Jogos são utilizados pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
12) Simulações são utilizadas pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
13) Exercícios de criatividade são utilizadas pertinentemente e contribuem para a motivação e compreensão dos conteúdos?				
14) Diálogos são utilizados ao longo do <i>software</i> de forma a apoiar e verificar a compreensão dos conteúdos?				
15) O utilizador tem controlo sobre a ordem de apresentação e sequência das informações?				
16) O <i>software</i> estimula a imaginação do utilizador através de um dado contexto ou situação que pode ser usada para auxiliar a aprendizagem?				
17) A apresentação das informações possui geração aleatória, variando estímulos textuais, visuais e ou sonoros?				
18) O <i>software</i> oferece a possibilidade de consulta à outras referências bibliográficas sobre o tema em estudo, tais como livros e outros materiais instrucionais?				

Adaptabilidade: Flexibilidade	S	P	N	PESO
1) O <i>Software</i> permite a introdução de novos elementos, personalizando-o de modo a acomodar diferenças individuais?				
2) O <i>software</i> contém a opção de seleção de entrada para níveis intermédios de dificuldade?				
3) O sistema fornece meios para que o utilizador tenha total controle sobre a sequência de apresentação das informações?				
4) O sistema propõe formas variadas de apresentação das mesmas informações à diferentes tipos de utilizador?				
5) O sistema fornecer a possibilidade de desativar temporariamente a apresentação de certas janelas?				
6) Os utilizadores têm a possibilidade de modificar ou eliminar itens irrelevantes das				



janelas?				
7) O sistema permite que se defina, mude ou suprima os valores definidos por default, alterando-os e personalizando-os?				
8) O utilizador tem a possibilidade de modificar a ordem e a sequência de entrada de dados, adaptando-a segundo sua ordem de preferência?				
9) Quando o formato de um texto não poder ser previsto com antecedência, o sistema proporciona meios para definir e salvar os formatos que ele venha a precisar?				
10) É permitido ao utilizador definir os nomes dos campos de dados que ele (a) venha criar?				
11) Ao utilizador é permitido personalizar o diálogo, através da definição de macros?				

Adaptabilidade: Consideração da Experiência do Utilizador	S	P	N	PESO
1) A sequência da apresentação dos conceitos evolui significativamente em grau de complexidade?				
2) Utilizadores mais experientes podem ter acesso direto aos módulos mais avançados?				
3) O <i>software</i> permite que o aluno possa retornar novamente no exato nível em que atingiu no seu último acesso?				
4) O sistema possibilita efetuar alterações em suas estruturas de modo a contemplar a experiência do utilizador?				
5) O <i>software</i> permite flexibilidade na resolução dos problemas propostos, não requerendo do aluno que o mesmo complete tarefas básicas antes que lhe seja permitido continuar no programa				
6) O sistema prevê a escolha de entradas simples ou múltiplas conforme a experiência do utilizador?				
7) O sistema fornece um tutorial passo a passo para os novatos e a entrada de comandos mais complexos por mais experientes?				
8) O sistema permite que utilizadores experientes contornem uma série de seleções por menu através da especificação de comandos ou de atalhos de teclado?				
9) É fornecida a possibilidade de escolher o nível de detalhe das mensagens de erro em função do nível de conhecimento?				
10) O sistema oferece a facilidade para que utilizadores de níveis de familiaridade diferentes possam facilmente adequarem-se ao sistema? (através da disponibilização de teclas de atalho/aceleração).				

Controle Explícito: Ações explícitas do utilizador	S	P	N	PESO
1) O processamento das ações é efetuado somente quando solicitadas pelo utilizador?				
2) O sistema sempre exige uma ação explícita de ENTER, para dar início ao processamento dos dados?				
3) O sistema posterga os processamentos até que as ações de entrada tenham sido encerradas?				
4) Durante a seleção de uma opção de menu o sistema permite a separação entre indicação e execução da opção?				
5) No caso de opções de preenchimento, é sempre o utilizador quem comanda a navegação entre os campos?				

Controle Explícito: Controle do Utilizador	S	P	N	PESO
1) O utilizador possui controle sobre os botões de comando?				
2) O utilizador pode controlar a sequência dos conteúdos?				
3) O utilizador pode controlar o ritmo da apresentação?				
4) O utilizador pode controlar a apresentação por meio de opções de escolha entre diferentes níveis de complexidade?				
5) O utilizador pode interromper, retomar e reiniciar um diálogo sequencial a qualquer instante?				
6) O sistema possibilita interromper ou cancelar a transação ou processo em andamento, sempre que se julgar necessário?				
7) O sistema fornece a opção CANCELAR a qual tem o efeito de apagar qualquer mudança efetuada pelo utilizador e trazer o ecrã para seu estado anterior?				

Adaptado de Gamez (1998) pelo autor.



8) Durante os períodos de bloqueio dos dispositivos de entrada, o sistema fornece uma opção para interromper o processo que causou o bloqueio?				
--	--	--	--	--

Gestão de Erros: Correção de Erros	S	P	N	PESO
1) A correção de erros durante a execução de exercícios é otimizada, ou seja, permite que o utilizador faça a correção sem ter que refazer vários passos anteriores?				
2) Na ocorrência de erros na resolução dos exercícios propostos, o <i>software</i> orienta e oferece ao aluno a possibilidade de tentar refazer o exercício?				
3) Persistindo no erro durante a resolução dos exercícios, o <i>software</i> conduz o utilizador fornecendo-lhe sequências explicativas para a correção das respostas inadequadas?				
4) O <i>software</i> fornece a resolução dos exercícios após longa persistência no mesmo erro?				
5) O <i>software</i> permite a mudança automática de exercício, se o aluno persiste no erro, conduzindo-o a outro tipo de exercício, com um menor grau de dificuldade?				
6) O <i>software</i> possui algum registro das dificuldades enfrentadas pelo aluno na resolução dos exercícios?				
7) Caso o utilizador tenha a necessidade de recorrer a teoria para a resolução dos exercícios, este acesso-lhe é facilitado por meio de um atalho?				
8) É disponibilizada a opção de menu “gravar”?				
9) Qualquer ação pode ser revertida através da opção DESFAZER?				
10) Através da opção REFAZER, a regressão do diálogo também pode ser desfeita?				
11) Os comandos para a opção DESFAZER e REFAZER o diálogo estão diferenciados?				

Gestão de Erros: Qualidade das Mensagens de Erros	S	P	N	PESO
1) Na ocorrência de erros durante a resolução dos exercícios, as mensagens de erro auxiliam e informam o utilizador na superação do erro?				
2) O feedback das respostas às dificuldades é encorajador e livre de conotação negativa?				
3) Perante uma dificuldade na resolução dos exercícios, o <i>software</i> evita a monotonia oferecendo mensagens de erro variadas?				
4) As frases das mensagens de erro adoptam um vocabulário neutro, não personalizado, não repreensivo e evitam o sentido de humor?				
5) As frases das mensagens de erro são curtas e construídas a partir de palavras curtas, significativas e de uso comum?				
6) As mensagens de erro são neutras e polidas?				
7) As mensagens de erro estão isentas de abreviaturas e/ou códigos gerados pelo sistema operacional?				
8) As mensagens de erro estão orientadas à tarefa?				
9) As mensagens de erro têm seu conteúdo modificado quando na repetição imediata do mesmo erro pelo mesmo utilizador?				
10) O utilizador pode escolher o nível de detalhe das mensagens de erro em função de seu nível de conhecimento?				
11) As mensagens de erro ajudam a resolver o problema do utilizador, fornecendo com precisão o local e a causa específica ou provável do erro, bem como as ações que o utilizador poderia realizar para corrigi-lo?				

Gestão de Erros: Proteção Contra Erros	S	P	N	PESO
1) O sistema permite que o utilizador interrompa uma operação contornando uma operação indesejada?				
2) O sistema permite que o utilizador possa voltar atrás, e informa-o em caso de comandos que induzam a erro?				
3) Quando o utilizador termina uma secção e existe o risco de perda de dados, o sistema emite uma mensagem que o avisa deste fato, pedindo-lhe confirmação do final da secção?				
4) As apresentações que acompanham as entradas de dados estão protegidas, de modo que não se possa modificar as informações contidas nestes campos?				
5) Os títulos dos campos estão protegidos e impedidos de serem alterados pelo utilizador?				



6) O sistema emite sinais sonoros quando ocorrem problemas na entrada de dados?				
7) Ao final de uma sessão de trabalho, antes de fechar o aplicativo, o sistema solicita a opção salvar e informa sobre o risco de perda dos dados?				
8) No caso de ocorrência de erros de digitação de um comando ou de dados, o sistema permite que o utilizador corrija somente a parte dos dados ou do comando que está errado?				
9) Em toda ação destrutiva, os botões selecionados por default realizam a anulação dessa ação?				
10) As teclas de funções perigosas e/ou rotineiras encontram-se agrupadas e/ou separadas das demais no teclado?				

Avaliação do Aprendizado	S	P	N	PESO
1) No caso em que questões são apresentadas para a verificação de um determinado conceito, estas são formuladas de maneira clara e objetiva, evitando que o utilizador faça uma interpretação errônea da questão?				
2) O <i>software</i> dispõe de algum recurso que permita avaliar o grau de compreensão dos alunos na resolução de problemas?				
3) Durante a sequência de apresentação o <i>software</i> propõe questões para verificar a compreensão dos conteúdos, simulando uma relação entre professor e aluno?				
4) O <i>software</i> possui bom grau de coerência no conteúdo das questões apresentadas em função dos objetivos a que se propôs?				
5) O <i>software</i> armazena informações relativas à interação dos alunos tais como pontuações, tempo de resposta, nível atingido?				
6) O <i>software</i> permite gravar automaticamente os registros do desempenho dos alunos mesmo que estes abandonem o programa?				
7) O programa permite o registro seguro dos resultados obtidos pelos alunos, sem que se corra o risco dos mesmos serem facilmente alterados por outrem?				

Homogeneidade	S	P	N	PESO
1) Os ícones são distintos uns dos outros e possuem sempre o mesmo significado de um ecrã ao outro?				
2) Os formatos de apresentação dos dados são mantidos homogêneos de um ecrã ao outro?				
3) A organização em termos da localização das várias características das janelas é mantida homogênea de um ecrã ao outro?				
4) Os significados dos códigos de cores são seguidos de maneira homogênea?				
5) A localização dos diferentes elementos funcionais são mantidos homogêneos de um ecrã ao outro?				
6) Os procedimentos de acesso às opções dos menus são homogêneos?				

Significado dos Códigos e Denominações	S	P	N	PESO
1) As denominações dos títulos estão de acordo com o que eles representam?				
2) O vocabulário técnico utilizado é familiar ao utilizador?				
3) Existe um glossário para os termos técnicos?				
4) O vocabulário utilizado nos títulos, convites e mensagens de orientação são familiares ao utilizador e evitam palavras difíceis?				
5) As denominações das opções de menu são familiares ao utilizador?				
6) Os títulos das páginas são explicativos, e refletem a natureza da escolha a ser feita?				
7) O sistema adopta códigos significativos ou familiares aos utilizadores em vez de códigos e denominações arbitrárias?				
8) As abreviaturas são significativas?				
9) As abreviaturas são facilmente distinguíveis umas das outras, evitando confusões geradas por similaridade?				
10) No caso de gráficos, as denominações das linhas e colunas são significativas e distintas?				
11) Os significados usuais das cores são respeitados nos códigos de cores definidos?				



Compatibilidade	S	P	N	PESO
1) Os procedimentos de diálogo são compatíveis com os definidos pelos padrões do ambiente em que roda o <i>software</i> ?				
2) O sistema segue as convenções dos utilizadores para dados padronizados?				
3) O sistema utiliza unidades de medida familiares ao utilizador?				



ANEXO C – III MÓDULO DE AVALIAÇÃO CONTEXTUAL

Adequabilidade	S	P	N	PESO
1) O resultado do módulo de avaliação indica conformidade aos padrões ergonômico/pedagógicos ?				
2) O <i>software</i> adapta-se ao programa curricular proposto?				
3) O <i>software</i> pode facilmente ser integrado no conteúdo curricular e outras partes do currículo escolar para auxiliar a aprendizagem desta disciplina?				
4) O <i>software</i> é pertinente aos objetivos educacionais propostos?				
5) Os objetivos do <i>software</i> são coerentes com as propostas pedagógicas do educador e/ou instituição escolar?				
6) O <i>software</i> realmente auxilia os alunos na aquisição das habilidades e conteúdos propostos?				
7) O mesmo conteúdo do <i>software</i> poderia ser facilmente ensinado sem o uso do recurso tecnológico do computador?				
8) É realmente indispensável o uso da tecnologia no ensino desta disciplina?				
9) O <i>software</i> é adequado ao público alvo da instituição, (idade, nível de ensino, bagagem cultural, nível socioeconômicos)				
10) As formas da apresentação das ideias estão coerentes com a fundamentação psicopedagógica adotada pela instituição?				
11) Os conhecimentos adquiridos pelo <i>software</i> possuem alguma aplicabilidade prática na vida pessoal e profissional dos utilizadores?				
12) Os recursos e estratégias dinâmicas propostos pelo <i>software</i> , podem contribuir para a melhoria do relacionamento professor aluno e para a relação entre os colegas em sala de aula?				
13) O <i>software</i> é neutro e não disponibiliza processos de julgamento acerca do valor de ideias, trabalhos, valores sociais, familiares e religiosos?				
14) O preço do produto é viável para a sua aquisição?				
15) A instituição possui os equipamentos necessários para rodar o produto (requisitos de hardware e <i>software</i>)?				
16) Os professores desta instituição teriam facilidade em adotar o <i>software</i> como parte das suas atividades pedagógicas?				