



UNIVERSIDADE FEDERAL DO MARANHÃO

Centro de Ciências Exatas e Tecnologia

Curso de Ciência da Computação

Elydillse Maria Botelho Lindoso

**Avaliação da Ferramenta de Ensino de  
Programação Caça-Bugs**

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Diretoria Integrada de Bibliotecas/UFMA

Lindoso, Elydillse Maria Botelho.

Avaliação da ferramenta de ensino de programação Caça-Bugs / Elydillse Maria Botelho Lindoso. - 2021.  
77 f.

Orientador(a): Carlos de Salles Soares Neto.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2021.

1. Algoritmos. 2. Depuração de código. 3. Informática na educação. 4. Objeto de aprendizagem. I. Soares Neto, Carlos de Salles. II. Título.

Elydillse Maria Botelho Lindoso

# **Avaliação da Ferramenta de Ensino de Programação Caça-Bugs**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Carlos de Salles Soares Neto

São Luís - MA

2021

Elydillse Maria Botelho Lindoso

## **Avaliação da Ferramenta de Ensino de Programação Caça-Bugs**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação.

Trabalho aprovado. São Luís - MA, 29 de abril de 2021:

---

**Prof. Dr. Carlos de Salles Soares Neto**  
Orientador  
Universidade Federal do Maranhão

---

Prof. MSc. Alana de Araújo Oliveira  
Meireles Teixeira  
Examinador  
Universidade Federal do Maranhão

---

Prof. MSc. Carlos Eduardo Portela Serra de  
Castro  
Examinador  
Universidade Federal do Maranhão

São Luís - MA  
2021

*Dedico este trabalho à todos que contribuíram para o final desta jornada.*

# Agradecimentos

O antagonismo dos agradecimentos é quase utópico de ser completo e aceitavelmente breve. Primeiramente, agradeço a Deus por minha vida, saúde, família e amigos. Por conceder discernimento em minhas escolhas, superação nas adversidades, por nunca ter me desamparado e permitir a concretização deste sonho.

Aos meus pais, por não medirem esforços, por sempre estarem ao meu lado, me apoiando em todas as minhas decisões e por sempre priorizarem o caminho da educação. Agradeço a eles todo o amor, carinho, princípios e compreensão conferidos ao longo minha vida. À minha irmã, por ser compreensiva em momentos em que eu não podia estar presente. À minha tia Maria dos Santos, pelo inestimável cuidado desde o meu nascimento. Ao meu tio Luís, Edmilson e Edson, por serem presentes, prestativos e atenciosos sempre recorro à eles. À todos os meus familiares pelo incentivo e zelo. E, óbvio que não menos importante, ao meu cachorrinho Harry: meu fiel companheiro que assistia às aulas, resolvia provas e passava a noite em claro junto comigo e que sabe a hora certa de um lambeijo (toda hora) e de me acordar.

Ao meu orientador, o Prof. Dr. Carlos de Salles Soares Neto, que desde o início do curso, sempre acreditou em mim e no meu potencial. Por guiar-me com paciência, maestria e ter proporcionado momentos únicos através do seu excepcional talento na docência e amor ao que faz. Não encontro palavras que possam exprimir a minha admiração e gratidão em tê-lo como professor, orientador e amigo. Sem sua imprescindível contribuição, não chegaria até aqui e, tampouco, teria me desenvolvido ao longo da minha trajetória acadêmica. Agradeço imensamente o reconhecimento, incentivo, paciência e dedicação. Espero continuar trabalhando e aprendendo ainda mais contigo.

Aos professores José Ribamar Braga Jr. e Francisco Silva, que além de excelentes profissionais, me acompanharam em um momento muito delicado, oferecendo um ombro amigo e mostrando que o ato de ensinar não cabe só à sala de aula. Fica aqui o meu mais sincero agradecimento. Também gostaria de deixar meu agradecimento ao professor Alex Barradas e Márcio Cerqueira, do curso de Engenharia da Computação, pelos sinceros encorajamentos em momentos difíceis.

Aos amigos Erickson e Asan, que tive a honra de conhecer, a oportunidade de aprender e partilhar muitas alegrias, dores e conquistas desde o início do curso, sempre com incentivo e companheirismo mútuo. Vocês são os irmãos que a UFMA me deu. Serei sempre grata pela amizade de vocês. Gostaria de pontuar, em especial, todos os momentos em que o Erickson esteve presente e paciente com as atribulações da vida.

Aos amigos e UFmores Hannah, Paulinho, Eduardo, Stanley, Adriane, Wesley, Erick,

Gabriel, Ramon, Daniel, Júlio, Jéssica, Douglas e Bruno dos Anjos que proporcionaram inúmeros momentos de descontração, incentivos, troca de conhecimentos, diversão, companhia e lugar na fila do RU. Ao Jorge, Elisa e Alessa por todas as palavras de carinho e compreensão, apesar de não entenderem nada sobre o curso. Ao Daniel Geovane, Magno Lopes e Alan Galesco, por serem sempre excepcionais e me aturarem há mais de uma década. Ao Danielzinho, Antônio Busson e Djefferson por serem os companheiros de laboratório mais legais e por me ajudarem com muito entusiasmo.

Aos profissionais da saúde Érica, Vitor, João Paulo, Sâmia, Lays e tantos outros, que me acompanham desde 2018, mostrando o lado florido da montanha. Obrigada por não desistirem de mim, por todo o cuidado, compreensão e doação. Toda a minha gratidão a vocês. Vocês são parte primordial disso.

Aos meus adoráveis monitores da disciplina de algoritmos, André Filipe e Paulo Victor. Obrigada por aceitarem o desafio de elaboração das questões em pouquíssimo tempo. Vocês foram importantes neste trabalho e espero ter contribuído e incentivado-lhes. Desejo muito sucesso na jornada de vocês. Contem sempre comigo.

Ademais, à todos que contribuíram direta ou indiretamente para o fim desta etapa.

À você, que dispôs de um pouco do seu tempo para prestigiar este trabalho.

*"Não importa o que aconteça, continue a nadar"*  
(Walters, Graham; *Procurando Nemo*, 2003)

# Resumo

Os altos índices de evasão e de reprovação representam um dos principais gargalos dos cursos de Computação e são problemas comumente encontrados em disciplinas voltadas ao ensino de algoritmos. Isso é decorrente da dificuldade de adaptação dos alunos, pois a abordagem de ensino normalmente requer o desenvolvimento do raciocínio lógico e da capacidade de resolução de problemas, e não apenas a memorização de conteúdo. Este trabalho propõe a aplicação de mudanças na metodologia tradicional de ensino de programação por meio da avaliação e implementação de um ambiente virtual de aprendizado: o Caça Bugs. Ele é focado na depuração de códigos, voltado para atividades da disciplina de algoritmos como apoio ao ensino, visando promover engajamento e motivação, contribuindo também para o desenvolvimento dos alunos através da análise de códigos e resolução de problemas pré-definidos. No decorrer desse processo, um grupo de voluntários participou de um experimento controlado de utilização do ambiente, que resultou na aplicação de um formulário de avaliação. Os dados coletados mostram que o sistema alcançou resultados promissores quanto ao envolvimento e incentivo no aprendizado dos participantes, possuindo grande aceitação, e sendo reconhecida como uma forma não tradicional de transmissão de informações em uma abordagem criativa e estimulante.

**Palavras-chaves:** Algoritmos. Objeto de aprendizagem. Depuração de código. Informática na educação.

# Abstract

The high dropout and failure rates represent one of the main bottlenecks in Computer courses and are problems commonly found in classes aimed at teaching algorithms. This is due to the difficulty of adaptation of the students, since the teaching approach generally requires the development of logical reasoning and problem-solving capacity, and not just the memorization of contents. This work proposes the application of changes in the traditional methodology of teaching programming, through the evaluation and implementation of a learning management system: the Caça Bugs. It is focused on debugging codes, focusing on activities in the course of algorithms to support teaching, aiming to promote engagement and motivation, also contributing to the development of students through the analysis of codes and the resolution of predefined problems. During this process, a group of volunteers participated in a controlled experiment using the environment, which resulted in the application of an evaluation form. The data collected showed that the system achieved promising results regarding the involvement and encouragement of the participants learning, having considerable acceptance, and being recognized as a non-traditional way of transmitting knowledge in a creative and stimulating approach.

**Keywords:** Algorithms. Learning object. Code debugging. Informatics in education.

# Lista de ilustrações

Figura 1 – Dashboard do Cosmo . . . . .	19
Figura 2 – Área de Resposta do Cosmo . . . . .	20
Figura 3 – Dashboard do Professor . . . . .	21
Figura 4 – Dashboard do Aluno . . . . .	22
Figura 5 – Teste de personalidade . . . . .	23
Figura 6 – Painel de estudo . . . . .	23
Figura 7 – Painel de projeto prático . . . . .	24
Figura 8 – Tela de login . . . . .	25
Figura 9 – Tela de questões . . . . .	26
Figura 10 – Tela próximo nível . . . . .	26
Figura 11 – Tela entrada inválida . . . . .	27
Figura 12 – Tela você venceu o caça bugs . . . . .	27
Figura 13 – Arquitetura do Cliente-Servidor . . . . .	28
Figura 14 – Arquitetura do Caça Bugs - Front-End . . . . .	29
Figura 15 – Arquitetura do Caça Bugs - Back-End . . . . .	29
Figura 16 – Diagrama de Caso de Uso do Caça Bugs . . . . .	30
Figura 17 – Diagrama de Casos de Uso do Janela de Feedback . . . . .	31
Figura 18 – Diagrama de Sequência - Responder questão . . . . .	32
Figura 19 – Diagrama de Entidade e Relacionamento . . . . .	33
Figura 20 – Tela com as melhorias na interface . . . . .	34
Figura 21 – Nova ordem das questões . . . . .	35
Figura 22 – Relatório de log . . . . .	35
Figura 23 – Tela de <i>feedback</i> . . . . .	36
Figura 24 – Gráfico de distribuição de usuários por sexo e distribuição de usuários por idade . . . . .	40
Figura 25 – Gráfico de distribuição de usuários com experiência em programação e distribuição de usuários por nível de experiência em ferramentas de ensino	40
Figura 26 – Distribuição de como consideram o Caça Bugs . . . . .	41
Figura 27 – Distribuição da facilidade na utilização . . . . .	42
Figura 28 – Distribuição quanto à apresentação do sistema . . . . .	43
Figura 29 – Gráfico de distribuição baseada na compreensão dos assuntos abordados e na maneira utilizada para a compreensão do conteúdo . . . . .	44
Figura 30 – Gráfico de distribuição da avaliação geral do Caça Bugs e baseada na intuitividade do sistema . . . . .	44
Figura 31 – Distribuição do grau de satisfação com o Caça Bugs . . . . .	45
Figura 32 – Distribuição do grau de dificuldade no uso do Caça Bugs . . . . .	46

Figura 33 – Gráfico de distribuição da disposição das questões e baseada na complexidade	46
Figura 34 – Gráfico de distribuição da repetitividade das questões e referente ao números de questões . . . . .	47
Figura 35 – Distribuição da quantidade sugerida de questões . . . . .	47
Figura A1 – Termo de Consentimento Livre e Esclarecido Adaptado - Página 1 . . .	54
Figura A2 – Resultado do Termo de Consentimento Livre e Esclarecido Adaptado - Página 2 . . . . .	55
Figura B1 – Perfil de Análise Demográfica . . . . .	56
Figura B2 – Avaliação do Caça Bugs - Página 1 . . . . .	57
Figura B3 – Avaliação do Caça Bugs - Página 2 . . . . .	58
Figura B4 – Avaliação do Caça Bugs - Página 3 . . . . .	59
Figura B5 – Avaliação das Questões do Caça Bugs - Página 4 . . . . .	60
Figura B6 – Avaliação das Questões do Caça Bugs - Página 5 . . . . .	61
Figura C1 – Resultado Perfil de Análise Demográfica - Página 1 . . . . .	62
Figura C2 – Resultado Perfil de Análise Demográfica - Página 2 . . . . .	63
Figura C3 – Resultado da Avaliação do Caça Bugs - Página 1 . . . . .	64
Figura C4 – Resultado da Avaliação do Caça Bugs - Página 2 . . . . .	65
Figura C5 – Resultado da Avaliação do Caça Bugs - Página 4 . . . . .	66
Figura C6 – Resultado da Avaliação do Caça Bugs - Página 5 . . . . .	67
Figura C7 – Resultado da Avaliação do Caça Bugs - Página 6 . . . . .	68
Figura C8 – Resultado da Avaliação do Caça Bugs - Página 7 . . . . .	69
Figura C9 – Resultado da Avaliação do Caça Bugs - Página 8 . . . . .	70
Figura C10–Resultado da Avaliação do Caça Bugs - Página 9 . . . . .	71
Figura C11–Resultado da Avaliação do Caça Bugs - Página 10 . . . . .	72
Figura C12–Resultado da Avaliação do Caça Bugs - Página 11 . . . . .	73
Figura C13–Resultado da Avaliação do Caça Bugs - Página 12 . . . . .	74

# Lista de tabelas

Tabela 1 – Objetivos do Caça Bugs . . . . .	41
Tabela 2 – Principais dificuldades encontradas . . . . .	43
Tabela 3 – Principais utilidades encontradas . . . . .	45
Tabela 4 – Tabela de Alunos . . . . .	49
Tabela 5 – Tabela de Questões . . . . .	50

# Lista de abreviaturas e siglas

AJAX	Asynchronous Javascript and XML
CSS	Cascading Style Sheets
ER	Entidade e Relacionamento
HTML	Hypertext Markup Language
MYSQL	My Structured Query Language
OA	Objetos de Aprendizagem
PHP	Hypertext Preprocessor
SGDB	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
TIC	Tecnologias da Informação e Comunicação
UFMA	Universidade Federal do Maranhão
UML	Unified Modeling Language
URI	Universidade Regional Integrada
W3C	World Wide Web Consortium
XML	Extensible Markup Language

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Objetivos</b>	<b>17</b>
1.1.1	Objetivo geral	17
1.1.2	Objetivos específicos	17
<b>1.2</b>	<b>Organização do trabalho</b>	<b>17</b>
<b>2</b>	<b>TRABALHOS RELACIONADOS</b>	<b>18</b>
<b>2.1</b>	<b>Objetos de Aprendizagem</b>	<b>18</b>
<b>2.2</b>	<b>Ferramentas Educacionais para apoio ao Ensino de Algoritmos</b>	<b>19</b>
2.2.1	Cosmo	19
2.2.2	URI Online Judge Academic	20
2.2.3	Codecademy	22
<b>3</b>	<b>OBJETO DE APRENDIZAGEM CAÇA BUGS</b>	<b>25</b>
<b>3.1</b>	<b>Primeira Versão do Caça Bugs</b>	<b>25</b>
<b>3.2</b>	<b>Arquitetura do Caça Bugs</b>	<b>28</b>
3.2.1	Modelagem de Sistema	28
3.2.1.1	Diagrama de Casos de Uso	29
3.2.1.2	Diagrama de Sequência	31
3.2.2	Modelagem do banco de dados	32
<b>3.3</b>	<b>Contribuições deste trabalho para a nova versão do Caça Bugs</b>	<b>33</b>
<b>4</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>37</b>
<b>4.1</b>	<b>Cenário</b>	<b>37</b>
<b>4.2</b>	<b>Coleta de dados</b>	<b>38</b>
<b>4.3</b>	<b>Análise dos resultados</b>	<b>39</b>
4.3.1	Primeira Seção: Perfil Demográfico	39
4.3.2	Segunda Seção: Avaliação do Caça Bugs	40
4.3.3	Terceira Seção: Avaliação sobre as Questões do Caça Bugs	45
4.3.4	Dados referentes ao desempenho dos alunos	48
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>51</b>

<b>APÊNDICES</b>	<b>53</b>
<b>APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO ADAPTADO . . . . .</b>	<b>54</b>
<b>APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO CAÇA BUGS</b>	<b>56</b>
<b>APÊNDICE C – RESULTADOS DO QUESTIONÁRIO DE AVALIAÇÃO DO CAÇA BUGS . . . . .</b>	<b>62</b>
<b>REFERÊNCIAS . . . . .</b>	<b>75</b>

# 1 Introdução

Um problema comum às disciplinas voltadas ao ensino de algoritmos é o alto índice de evasão e de reprovação, sendo um dos gargalos nos cursos de graduação, particularmente de computação. Isso se deve à provável dificuldade de adaptação dos alunos por ser uma abordagem que exige o desenvolvimento do raciocínio lógico e a capacidade de resolução de problemas, e não apenas a memorização de conteúdo (HINTERHOLZ, 2009).

Nesse contexto, o presente trabalho propõe o desenvolvimento de um ambiente virtual de ensino capaz de promover uma alteração significativa na metodologia tradicional de ensino de algoritmos, que não dispõe das ferramentas necessárias para a conduzir a instrução de conteúdos que envolvem um alto nível de abstração.

Essa ferramenta, denominada de Caça Bugs, é voltada para atividades da disciplina de algoritmos e focada na depuração de códigos, visando promover o engajamento e a motivação dos alunos. Dessa forma, também contribui para desenvolvê-los no que diz respeito à análise de códigos e resolução de problemas pré-definidos.

Tendo isso em vista, a ferramenta visa a busca por alternativas que estimulem o interesse dos estudantes, mantendo-os engajados durante todo o processo e auxiliando-os a contornar os desafios da falta de interesse, que podem ser causados pela não compreensão dos componentes curriculares; pela falta de estímulo externo e por parte do docente; pela didática do professor; pela ausência de autoconfiança, dentre outras motivações.

Assim, o Caça Bugs se diferencia das demais ferramentas educacionais existentes por ser voltado ao desenvolvimento da capacidade analítica e de resolução de problemas dos estudantes. Nessa ferramenta os alunos são estimulados a descobrir erros propositalmente inseridos em códigos, corrigi-los e apresentar a resposta do problema desconsiderando a existência do erro.

Este trabalho é desenvolvido por meio de uma série de etapas, que consistem: na construção de um sistema de depuração de códigos, a ser disponibilizado como uma plataforma web; na criação de um banco de questões a serem solucionadas pelos estudantes; e na condução de um experimento controlado de utilização do ambiente com um grupo de voluntários.

Por fim, durante o experimento, são coletados *feedbacks* dos voluntários, que informam se gostaram ou não das questões e seus graus de dificuldade. Além disso, é aplicado um questionário relativo ao sistema, buscando aferir seu grau de intuitividade e usabilidade, e sobre a avaliação das questões disponibilizadas, como a quantidade e complexidade.

## 1.1 Objetivos

Esta seção visa ao detalhamento dos objetivos gerais e específicos deste trabalho.

### 1.1.1 Objetivo geral

O objetivo principal deste trabalho é avaliar o impacto da utilização de uma abordagem baseada em resolução de problemas e em análise e depuração de códigos, no contexto da disciplina de algoritmos.

### 1.1.2 Objetivos específicos

- Compreender o funcionamento e mapear as principais deficiências da versão atual do sistema Caça Bugs;
- Analisar e projetar novas funcionalidades para o sistema Caça Bugs, tais como o armazenamento de *logs* e a coleta de *feedbacks*;
- Desenvolver uma nova versão do sistema do Caça Bugs, contemplando as funcionalidades suprarreferidas;
- Realizar um experimento para validação das novas funcionalidades do sistema com um grupo de voluntários;

## 1.2 Organização do trabalho

Este trabalho estrutura-se da seguinte forma: além do [Capítulo 1](#), dispõe-se de mais 4 capítulos, sendo o [Capítulo 2](#) referente aos trabalhos relacionados a este.

O [Capítulo 3](#) descreve a proposta deste trabalho em sua primeira versão, as contribuições feitas e o processo de implementação do objeto de aprendizagem.

Já o [Capítulo 4](#) detalha a metodologia proposta diante de avaliação, apresentação e discussão dos resultados obtidos;

E, por último, o [Capítulo 5](#) contendo as conclusões, as considerações finais e a proposta de evolução para trabalhos futuros.

## 2 Trabalhos Relacionados

Este capítulo apresenta trabalhos relacionados ao tema pesquisado, isto é, ferramentas educacionais voltadas a auxiliar o ensino da disciplina de algoritmos. Com isso, as seções subsequentes estruturam-se em duas partes que esclarecem seu modelo de uso e suas aplicações. A primeira categoria centra-se na definição de objetos de aprendizagem, observada na [seção 2.1](#), e a segunda destina-se às plataformas de ensino, como especificada na [seção 2.2](#).

### 2.1 Objetos de Aprendizagem

Atualmente, o modo tradicional de ensino, nos mais variados níveis, enfrenta o desafio de inserções de tecnologias capazes de estabelecer novas formas de interação entre alunos e professores. O incremento do uso de Tecnologias da Informação e Comunicação (TIC) no eixo educacional demanda que o próprio processo de ensino e aprendizagem seja revisto. Com isso, o contexto educacional foi refatorado com o espaço que as TICs ganharam a partir da popularização da internet, como retrata ([LUZ; SANTOS; GUIMARAES, 2015](#)).

Diante disso, o conceito de objetos de aprendizagem (OA) molda esse processo no cenário de ensino aprendizagem. Em que ([LUZ; SANTOS; GUIMARAES, 2015](#)) ainda destaca que ([WILEY, 2000](#)) trata os (OAs) como qualquer entidade digital ou não-digital que pode ser utilizada, reutilizada e referenciada durante o processo de ensino.

Sendo assim, pode ser definido como um componente digital criado para apoiar alunos no processo de aprendizagem, por meio do uso de páginas web, textos, imagens, animações etc, tendo como objetivo central o reúso de seus recursos, podendo dispor, ainda, de conteúdo completo ou modular.

Ademais, pode-se perceber que a utilização de jogos para apoiar o ensino e a aprendizagem de algoritmos está em voga. No processo de pesquisa, pesquisadores estão empenhados em desenvolver métodos e sistemas educacionais para apoiar a aprendizagem no ensino de algoritmos. A seguir, são listados alguns desses mecanismos de maneira simplificada.

## 2.2 Ferramentas Educacionais para apoio ao Ensino de Algoritmos

### 2.2.1 Cosmo

O Cosmo é um ambiente virtual de ensino, multitarefa e com foco no estudo de lógica de programação. Foi criado e desenvolvido pelo laboratório de pesquisa Telemídia-MA, da Universidade Federal do Maranhão.

Possui estrutura extensível a *plugins*, concebível desde que possua suporte a HTML5, vinculando ao ambiente diferentes tipos de atividades como: 1) solucionar problemas; 2) apresentação de funções com erros de código; 3) permitir ao aluno assistir a um vídeo e ao final responder questões sobre o conteúdo, como relata (JúNIOR et al., 2018).

O Cosmo dispõe de um *dashboard* formado por: 1) atividades: contém o *card* da atividade e é exibido aleatoriamente; 2) pular: ação de ir para próxima atividade; 3) perfil do usuário: configurações do usuário e sair do sistema; 4) responder: leva à área de resposta; 5) menu do usuário: visualização das atividades respondidas e conecta links ao *dashboard*. A Figura 1 ilustra esse arranjo.

Figura 1 – Dashboard do Cosmo



Fonte: Adaptado de Júnior et al. (2018)

Na Figura 2, tem-se a área de resposta com a descrição do problema e o campo de resposta das atividades, em que o item 1) corresponde ao enunciado da questão e o

item 2) ao conteúdo que deriva do *plugin* instanciado, que, no caso, foi usado o "resolver problema".

Figura 2 – Área de Resposta do Cosmo

The screenshot shows the 'Resolva o problema' (Solve the problem) interface in CosmoTelemidia. The header includes 'CosmoTelemidia', 'Dashboard', 'Histórico de atividades', and a user profile for 'Dilson José'. The main content area is titled 'Resolva o problema' and contains a problem description, input/output examples, and a code editor. The problem is 'Maior que a soma de dois inteiros' (Greater than the sum of two integers), published by Carlos Salles on 02/11/2016. The problem statement asks for a program that takes three integers and prints true if the sum of the first two is greater than the third. The input/output examples are as follows:

Entrada	Respectiva Saída
100 300 20	true
10 10 100	false

The code editor is set to C/C++ and is currently empty. The interface also includes a 'Voltar' (Back) button and an 'Enviar' (Submit) button. The footer contains the copyright information: 'Copyright © 2014-2016 Almsaeed Studio e Telemidia. All rights reserved.' and the version number 'Version 0.1.1'.

Fonte: Adaptado de Júnior et al. (2018)

Dessa forma, professores podem criar conteúdos para o ensino de programação mediante um conjunto de atividades, vídeos e jogos, diferenciando o modo tradicional de ensino.

## 2.2.2 URI Online Judge Academic

O URI Online Judge Academic é um sistema de avaliação automática de código desenvolvido pela URI (Universidade Regional Integrada)<sup>1</sup>. Possui integração com o módulo URI Online Judge e foi desenvolvido com o papel de facilitar o aprendizado do aluno,

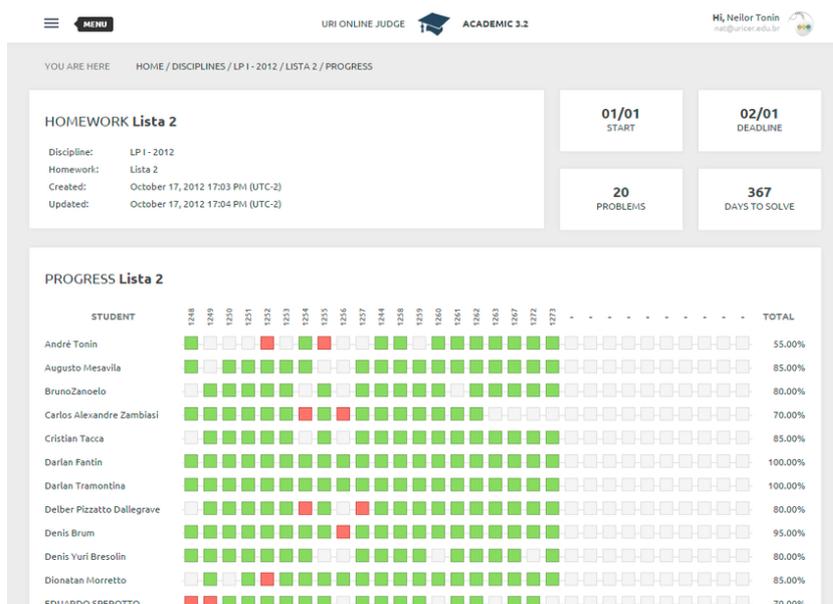
<sup>1</sup> Para mais informações, acesse: <http://www.uri.br/>

conduzindo-o a uma alternativa do método tradicional de ensino, com possibilidade do acompanhamento, rendimento e evolução dos alunos.

A ferramenta tem como características centrais: 1) respostas nas linguagens (C, C++, Java e Python); 2) correção automática; 3) organização das atividades por categorias e graus de dificuldade; 4) ranking por problema e linguagem de programação; 5) utilização de juízes especiais; e 6) fórum.

O ambiente *academic* possui dois *dashboards*, um centrado no professor e outro no estudante. No *dashboard* do professor encontram-se as funcionalidades de: 1) cadastrar turmas por meio de convite enviado para o aluno, cadastrar disciplinas e listas de tarefas que irão compor a disciplina; 2) escolha de linguagem utilizada; 3) definir prazo para solução e submissão dos problemas; 4) registrar instruções e verificar o rendimento dos alunos pelo percentual exibido; e 5) status do problema que quando solucionado, assume marcação na cor verde ou vermelho, quando não. A [Figura 3](#) apresenta o painel descrito.

Figura 3 – Dashboard do Professor

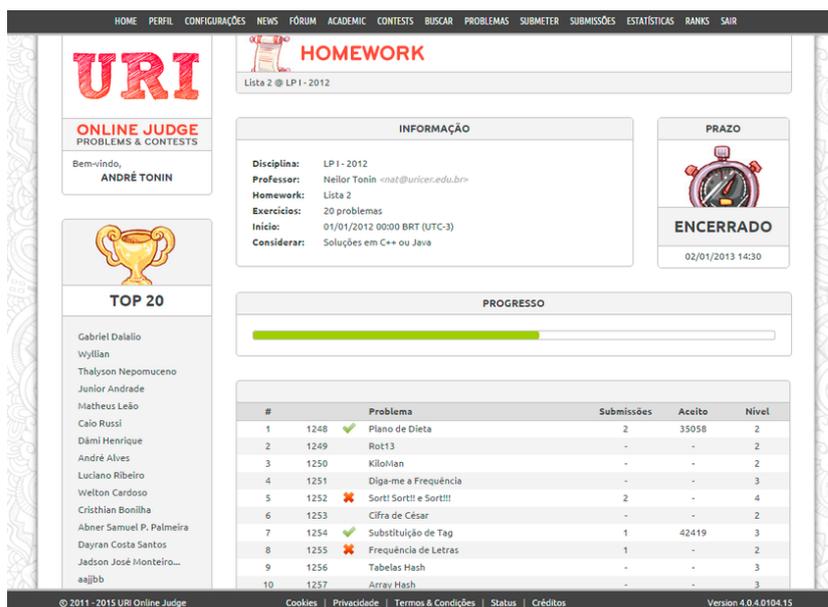


Fonte: (SELIVON; BEZERRA; TONIN, 2015)

Já no *dashboard* do aluno, encontra-se: 1) disponibilidade de acesso da turma pertencente; 2) listas disponibilizadas; 3) verificação do status das questões pela barra de progresso; 4) visualização das informações relativas às tarefas - id da questão, quantidade de submissões, se foi aceito ou não e o grau de dificuldade; 6) acesso às informações sobre a disciplina; e o 7) prazo de conclusão, como retratado na [Figura 4](#).

Tendo em vista os aspectos mencionados, diversos são os sistemas que possuem a vertente de avaliação automática de código. Destaca-se, ainda, o perfil competitivo presente na maioria deles, que resulta em maior estímulo e participação por parte dos estudantes na construção e estudo de algoritmos.

Figura 4 – Dashboard do Aluno



Fonte: (SELIVON; BEZERRA; TONIN, 2015)

### 2.2.3 Codecademy

A *Codecademy* é uma plataforma interativa para aprender a programar. Fundada por Zach Sims e Ryan Bubinski em 2011, oferece conteúdos gratuitos de programação de forma direta e intuitiva, dentre os quais estão cursos a respeito das linguagens Python, Java, PHP, Rubi, JavaScript, HTML e CSS, C++, etc.

Para utilizar a plataforma, é necessário que o usuário se cadastre nela e realize o processo de autenticação. Após autenticado, o usuário recebe um direcionamento quanto aos conteúdos, mediante a realização de um teste de aptidões, conforme [Figura 5](#). No mais, o usuário tem a independência de escolher o que aprender sem passar por esse teste, indo do nível mais básico ao avançado.

O objetivo do aluno é concluir os desafios para conseguir avançar de nível e, conseqüentemente, ir para tópico seguinte. Para isso, não há necessidade de instalação de nenhum componente para realização, já que as tarefas são feitas no próprio site, por meio de uma plataforma de alta disponibilidade.

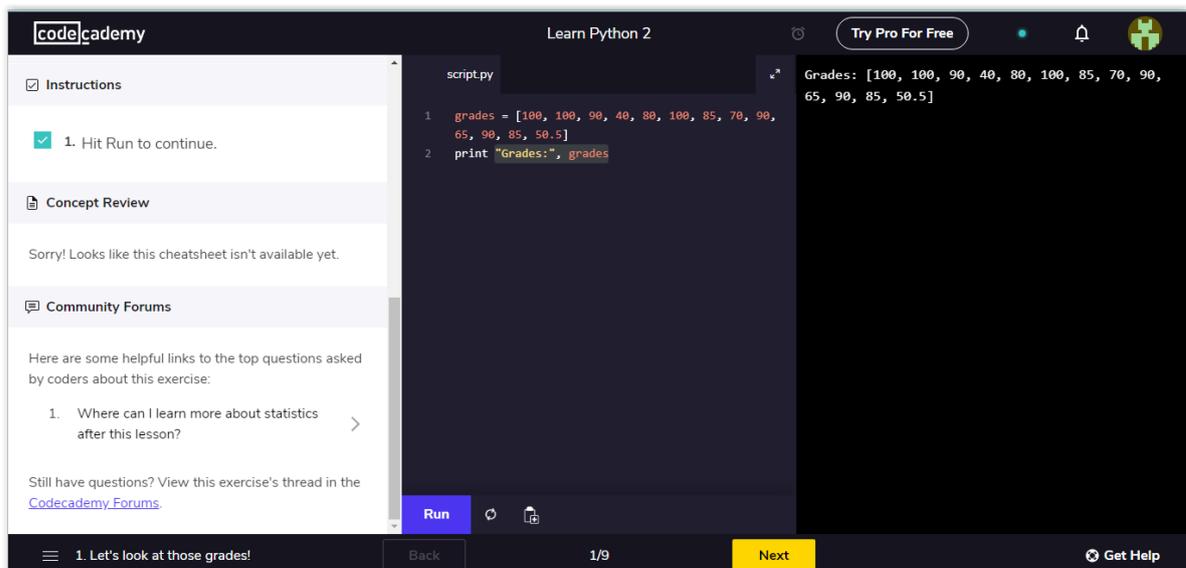
A plataforma dispõe de um conjunto de mecanismos de apoio ao ensino de programação, tais como teste de código automatizado; feedback instantâneo; aplicação dos conteúdos abordados por meio de projetos práticos e questionários personalizados. Além disso, a interface gráfica fornece em uma única tela o material didático; um editor de código-fonte; e um local em que o código pode ser testado, conforme apresentado nas [Figuras 6 e 7](#).

Figura 5 – Teste de personalidade



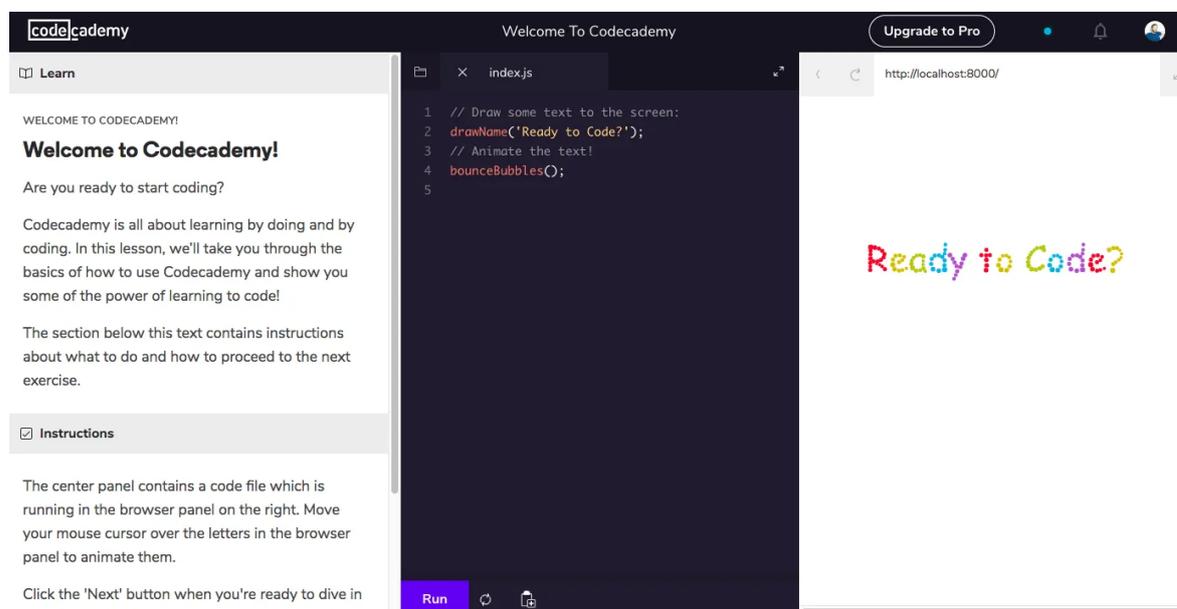
Fonte: (CODEACADEMY, 2021)

Figura 6 – Painel de estudo



Fonte: (CODEACADEMY, 2021)

Figura 7 – Painel de projeto prático



Fonte: (CODEACADEMY, 2021)

## 3 Objeto de Aprendizagem Caça Bugs

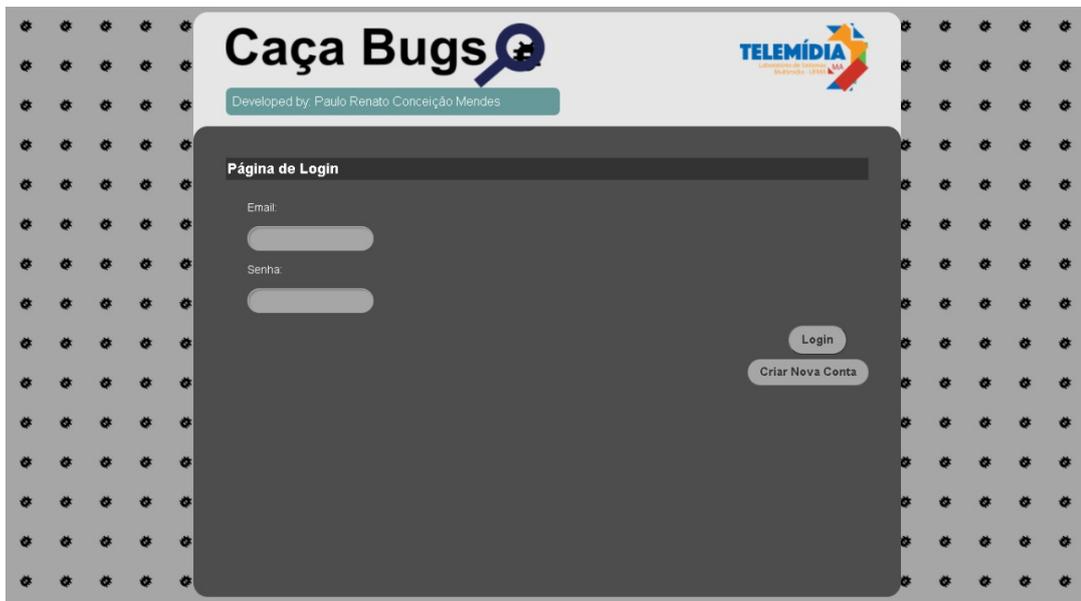
O Caça Bugs é um jogo educacional cujo objetivo é auxiliar no ensino da programação, facilitando a identificação de erros em códigos por alunos dos primeiros períodos de computação. Sua estrutura se apresenta em forma de *quiz* (perguntas e respostas) e possui relevância no que tange à interpretação de problemas computacionais e qualidade de código.

Está distribuído de acordo com os conteúdos abordados nas aulas das disciplinas iniciais de programação, tais como: estruturas condicionais, estruturas de repetição, cadeias de caracteres e cadeias de vetores. Em cada questão há um enunciado e um código referente ao problema proposto, sendo necessário que o aluno, para avançar de nível, explore um erro existente no código e forneça uma entrada válida para a questão.

### 3.1 Primeira Versão do Caça Bugs

A ferramenta desenvolvida por (MENDES et al., 2018) dispõe de uma interface eficiente com uma página de login, cadastro de novos usuários como mostra a Figura 8:

Figura 8 – Tela de login



Fonte: (MENDES et al., 2018)

No que tange às questões, há um *dashboard* com 20 questões fixas (Figura 9) nas linguagens Lua e C, as quais vão sendo liberadas à medida que cada questão é respondida corretamente. Sendo assim, a próxima questão só é destravada quando a anterior a ela

obtiver uma resposta válida e, com isso, possui uma janela indicando “Você Passou de Nível”, com um botão “Menu” para retornar ao menu principal e um “Next Question” para ir para a próxima questão, como na Figura 10.

Figura 9 – Tela de questões



Fonte: (MENDES et al., 2018)

Figura 10 – Tela próximo nível

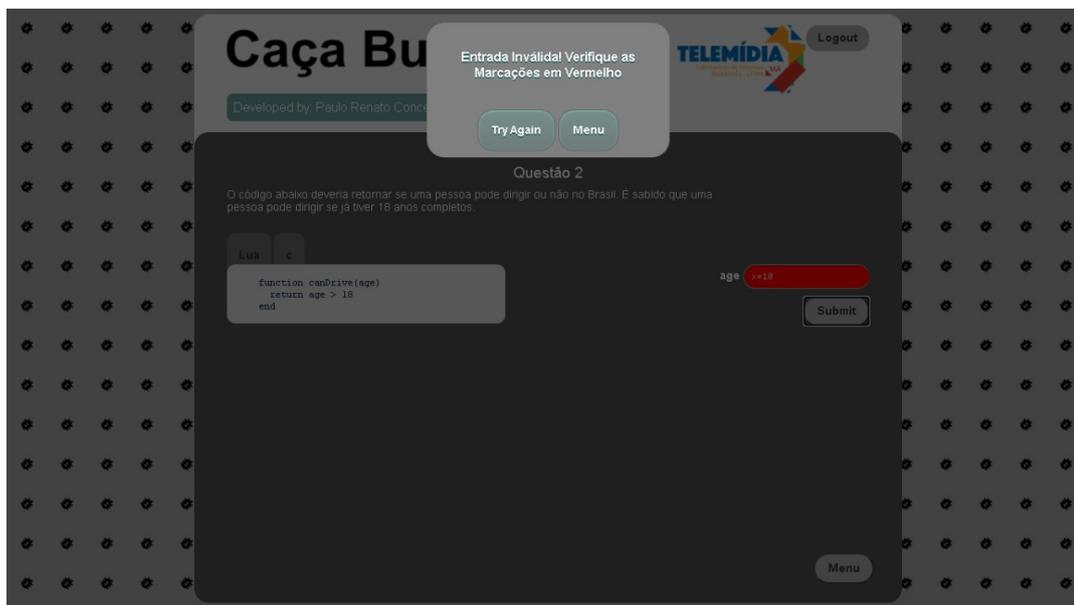


Fonte: (MENDES et al., 2018)

Já quando a entrada é inválida, é exibida uma marcação em vermelho com a opção “Try Again”, para uma nova tentativa; e “Menu”, para retornar ao menu principal (Figura 11). Ao final de todas as questões, uma tela indica que o jogador obteve êxito com

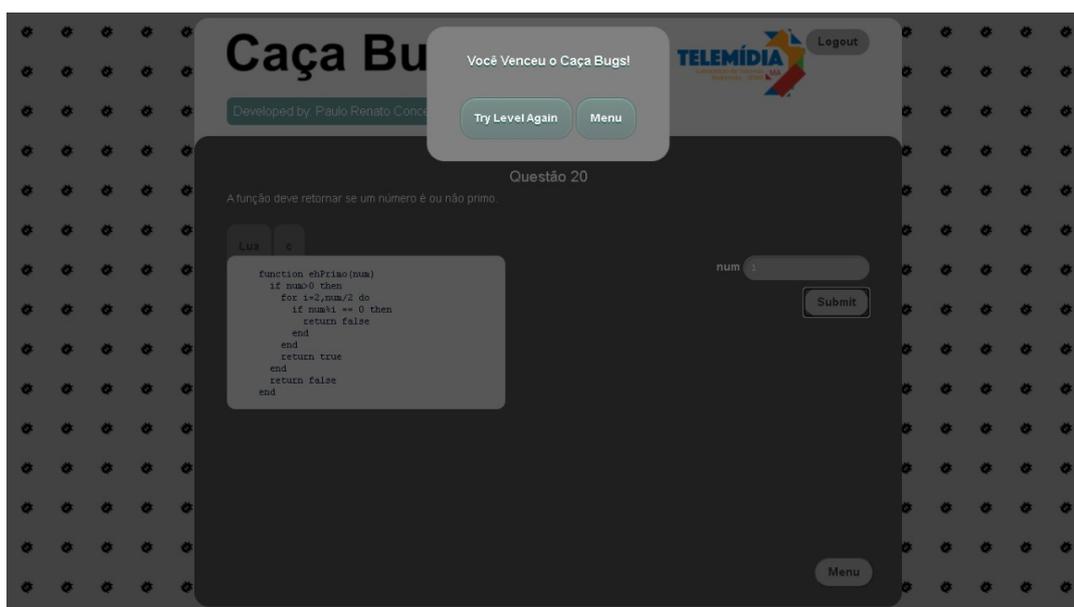
a seguinte informação: “Você venceu o Caça Bugs!” (Figura 12) e, conseqüentemente, que o jogador passou por todas as questões; simultaneamente, são exibidas as opções “*Try Level Again*”, para tentar o nível novamente; e “Menu”, para retornar ao menu principal, são exibidas com todas as questões disponíveis (Figura 9).

Figura 11 – Tela entrada inválida



Fonte: (MENDES et al., 2018)

Figura 12 – Tela você venceu o caça bugs

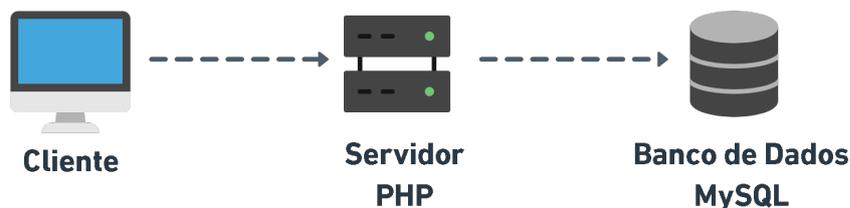


Fonte: (MENDES et al., 2018)

## 3.2 Arquitetura do Caça Bugs

A arquitetura proposta segue o modelo Cliente-Servidor, como ilustra a figura a seguir:

Figura 13 – Arquitetura do Cliente-Servidor



Fonte: Elaborada pela autora

Neste módulo, a arquitetura possui um papel claro e bem distribuído, sendo o Cliente responsável por requisitar dados durante a interação do usuário com a interface gráfica. O Servidor PHP<sup>1</sup> se encarrega de gerenciar e processar os dados do Cliente, além de enviar e receber dados do banco de dados MySQL<sup>2</sup>. Já este, por fim, é quem executa as funções de gerenciar e persistir todos os dados da aplicação.

Em razão disso, na parte que compõe o *Front-End* foram utilizadas as tecnologias HTML5 (W3C, 2014), JavaScript (ECMAScript E.; Association, 2020) e CSS3 (W3C, 2018), que consiste em um monolito web (PHP, HTML, CSS e JavaScript) que faz uma requisição AJAX<sup>3</sup> para o PHP, mantendo uma conexão assíncrona e assim os dados são enviados para o Servidor sem que a página seja recarregada. Diante disso, a página é carregada apenas uma única vez, ou seja, não muda durante a interação do usuário. Na Figura 14, pode-se observar essa relação.

Já a parte do *Back-End* foi escrita com o subsídio da linguagem de programação PHP na versão 8.0.2 e do SGBD (Sistema de Banco de Dados) MySQL. Com base nisso, as questões ficam localizadas em um arquivo XML<sup>4</sup> (Cliente) e o usuário, por sua vez, dispõe de informações que estão armazenadas no banco de dados (Servidor) e juntamente com ele, as tabelas referentes às credenciais e progresso dos usuários, registro de *logs* e *feedbacks*. Na Figura 15, podemos observar a ilustração.

### 3.2.1 Modelagem de Sistema

Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva diferente

<sup>1</sup> Disponível em: <https://www.php.net/>. Acesso em 27 mar. 2021

<sup>2</sup> Para informações acesse: <https://www.mysql.org>

<sup>3</sup> Para mais informações, acesse: [https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp)

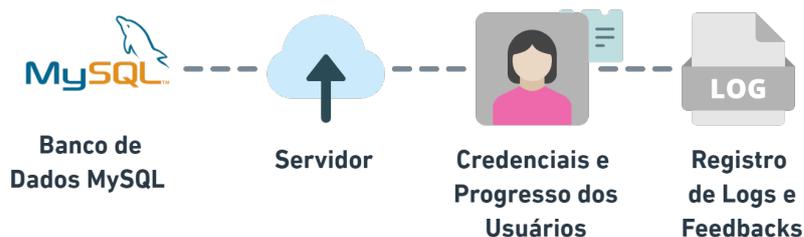
<sup>4</sup> <https://www.w3.org/TR/REC-xml/>

Figura 14 – Arquitetura do Caça Bugs - Front-End



Fonte: Elaborada pela autora

Figura 15 – Arquitetura do Caça Bugs - Back-End



Fonte: Elaborada pela autora

do sistema (SOMMERVILLE, 2018). O sucesso de um projeto de *software* está ligado diretamente à sua capacidade de atendimento às necessidades do usuário (DEVMEDIA, 2015). Posto isso, a modelagem torna-se imprescindível para uma melhor compreensão do sistema.

Portanto, neste projeto tem-se a representação gráfica da linguagem de modelagem unificada baseada na UML (Unified Modeling Language)<sup>5</sup>, a fim de garantir uma padronização e adequada estruturação do *software*. A seguir, são apresentados os diagramas de Casos de Uso e Sequência na notação da UML, que visam explicar as principais funcionalidades do sistema Caça Bugs.

### 3.2.1.1 Diagrama de Casos de Uso

Segundo (GUEDES, 2018), o diagrama de casos de uso utiliza uma linguagem simples e fácil de entender e expressa o conceito geral de como o sistema irá operar e é

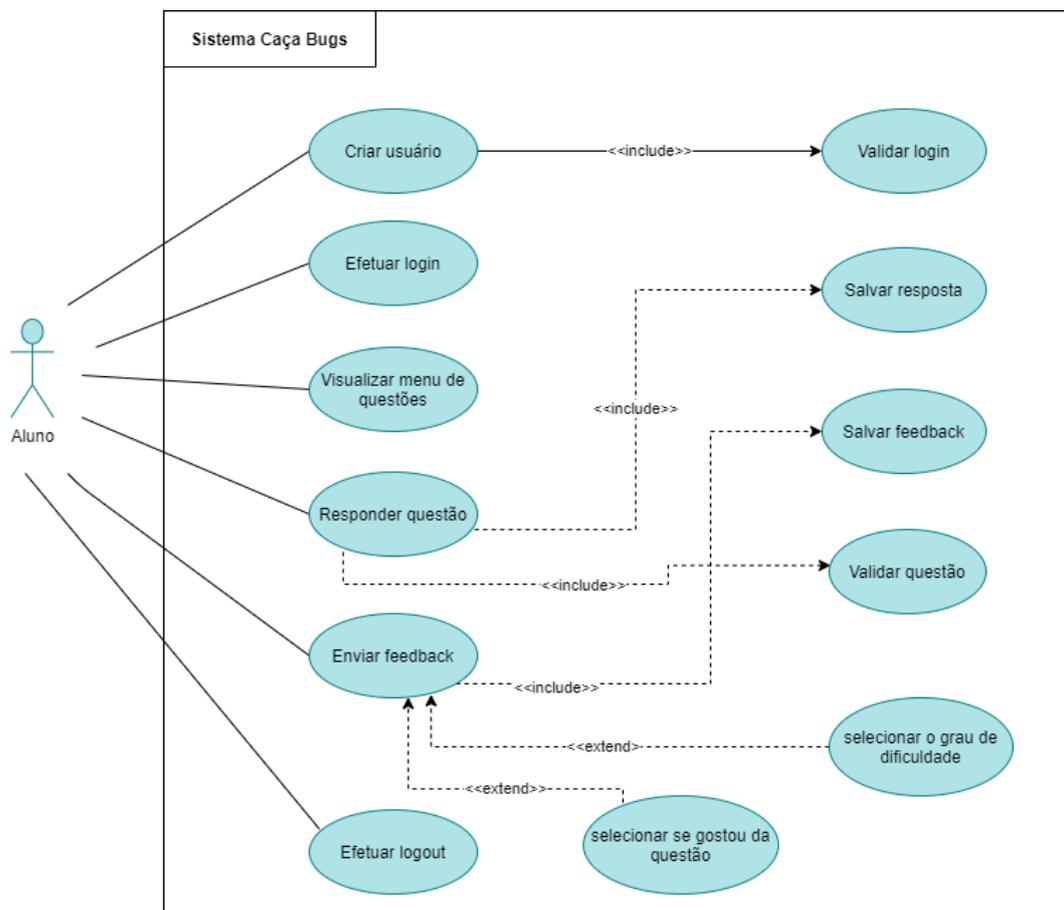
<sup>5</sup> Para informações acesse: <https://www.uml.org>

usado como base de consulta para todo o processo de modelagem e construção de outros diagramas. Os diagramas de casos de usos identificam os atores. O Ator é algo ou alguém que interage de alguma maneira com o sistema, buscando atingir alguma meta, como por exemplo, usuários, outros sistemas ou até mesmo um *hardware*.

Nesta subseção, temos o diagrama de Casos de Uso do Caça Bugs (Figura 16), que tem o intuito de elucidar suas principais funcionalidades e como os atores interagem com o sistema. Nele notamos dois atores que interagem com o sistema a fim de atingir uma meta: o usuário e o sistema. É notório destacar que a descrição do diagrama a seguir é feita de forma sucinta, ficando o detalhamento das funcionalidades para o diagrama de sequência, na próxima subseção.

Conforme a Figura 16, o ator Aluno age sobre duas principais funcionalidades que satisfazem esse ator, dentre elas ressaltam-se: 1) responder questão e 2) enviar *feedback*, além de "Criar usuário", "Efetuar login", "Visualizar menu de questões" e "Efetuar logout". Em contrapartida, o ator Sistema atua sobre o panorama de quatro funcionalidades, sendo: 1) validar *login*; 2) salvar resposta; 3) salvar *feedback*; e 4) validar questão.

Figura 16 – Diagrama de Caso de Uso do Caça Bugs

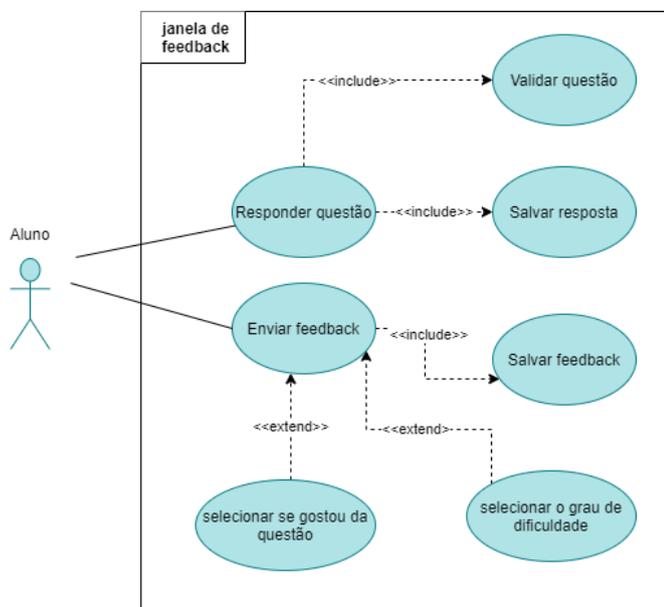


Fonte: Elaborada pela autora

Em decorrência disso, é fundamental detalhar a funcionalidade "Enviar feedback" do

ator Aluno, que possui relacionamento de dependência em "Salvar feedback" e comportamento extensível aos casos de uso: 1) selecionar se gostou da questão e 2) selecionar grau de dificuldade, como ilustrado na [Figura 17](#).

Figura 17 – Diagrama de Casos de Uso do Janela de Feedback



Fonte: Elaborada pela autora

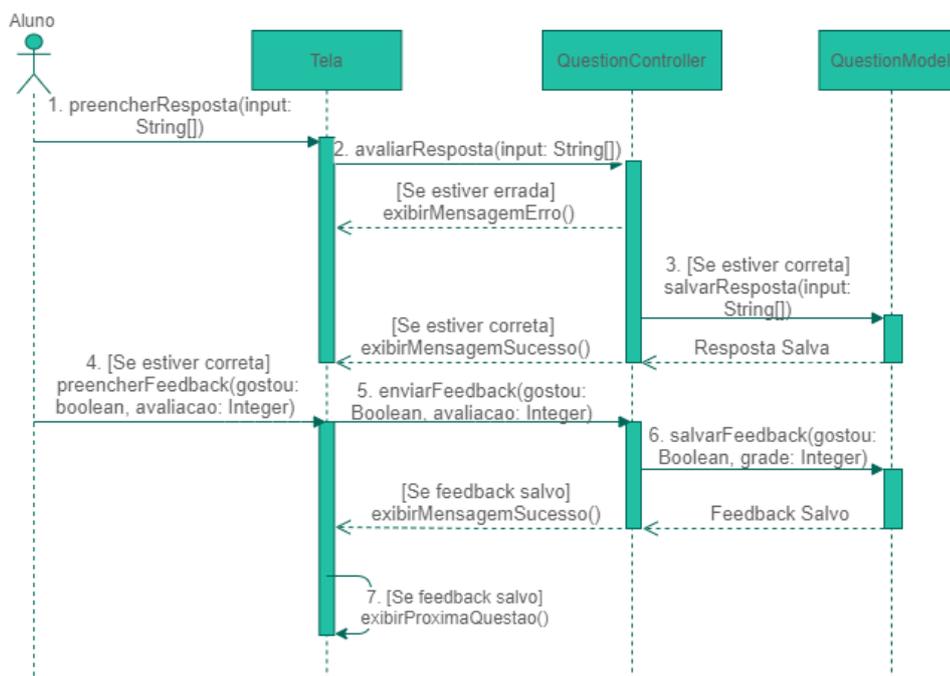
### 3.2.1.2 Diagrama de Sequência

Um diagrama de sequência mostra a sequência de interações que ocorrem durante um caso de uso em particular ou em uma instância de caso de uso. São usados para modelar interações entre os componentes do sistema, embora os agentes externos também possam ser incluídos ([SOMMERVILLE, 2018](#)).

Tomando-se por base os casos de uso do Sistema Caça Bugs ([Figura 16](#)) apresentado na subseção anterior, o caso de uso 1) do ator Aluno foi utilizado com o intuito de detalhar o seu funcionamento por intermédio do diagrama de sequência da [Figura 18](#).

No diagrama em questão, temos a sequência em que os eventos ocorrem no sistema, a identificação dos eventos disparados e a ordem em que ocorrem, assim como a interação desses objetos dentro desse processo. Com base nisso, esse fluxo é representado por um Ator (aluno) e três objetos (sistema, banco de dados e o servidor). Essa sucessão desencadeia uma série de ocorrências que se inicia com o aluno fazendo uma requisição ao Sistema para "responder questão". Após, a validação dela é feita no Banco de dados e, em seguida, no Servidor que carrega a questão e seus atributos e valida a operação. Daí, é realizado o teste da entrada inserida mediante as funções corretas e incorretas, que fazem a comparação entre as duas e o resultado é registrado no sistema.

Figura 18 – Diagrama de Sequência - Responder questão



Fonte: Elaborada pela autora

### 3.2.2 Modelagem do banco de dados

De acordo com (HEUSER, 2008), um modelo de (banco de) dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. É a descrição formal da estrutura de um banco de dados.

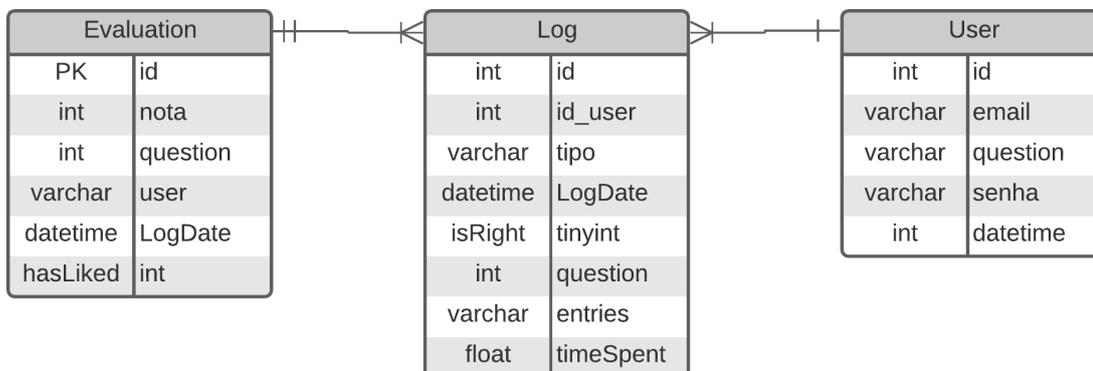
Para construir tal modelo, a linguagem de modelagem de dados escolhida foi a que contempla a abordagem entidade-relacionamento (ER), visando a organização dos dados no formato de tabelas. É um nível de abstração padrão e independe do tipo de SGDB relacional utilizado.

Em razão disso, a Figura 19 apresenta uma ilustração desse modelo lógico que definirá quais tabelas o banco contém e, para cada tabela, o nome de suas respectivas entidade, atributos e chaves.

Como indica a Figura 19 dispõe-se de três tabelas englobando os seguintes campos:

- 1) tabela de avaliação (*Evaluation*): *id(PK)*, *id(nota)*, *question(int)*, *user(varchar)*, *LogDate(datetime)*, *hasLiked(int)*, e é responsável pelo gerenciamento da janela de avaliação e intrinsecamente relativa à tela de *feedback* Figura 17.
- 2) tabela de logs (*Log*): *id(int)*, *id user(int)*, *tipo(varchar)*, *LogDate(datetime)*, *isRight(tinyint)*, *question(int)*, *entries(varchar)*, *timeSpent(float)* e encarrega-se de armazenar os dados especificamente das questões Figura 22.

Figura 19 – Diagrama de Entidade e Relacionamento



Fonte: Elaborada pela autora

- 3) tabela de usuários (*User*): *id* (*int*), *email* (*varchar*), *question* (*varchar*) *senha*(*varchar*), *datetime* (*int*) os dados relativos aos usuário.

Ademais, a cardinalidade apresentada entre a tabela *Evaluation* e *Log* propõe que o usuário pode avaliar uma, e somente uma questão, para um ou vários *logs*. E no relacionamento entre a tabela *Log* e *User* é necessário um usuário para cada *log* e cada *log* para um ou muitos usuários.

### 3.3 Contribuições deste trabalho para a nova versão do Caça Bugs

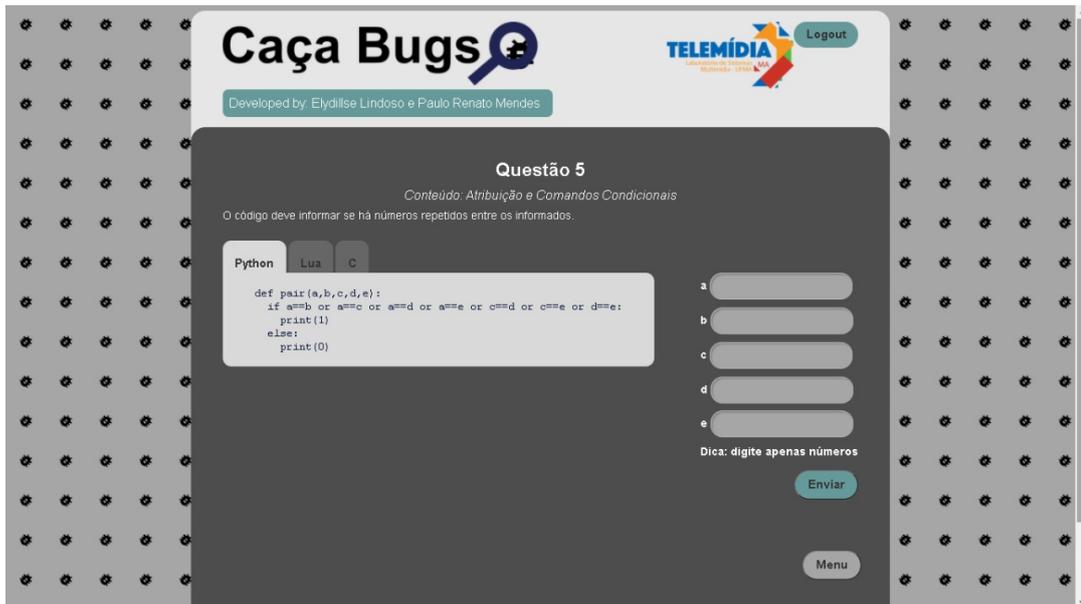
As principais contribuições deste trabalho dizem respeito à realização de melhorias na interface do usuário; à reorganização/categorização de questões; e à inclusão de um feedback visual para o usuário.

Particularmente, no que concerne à melhoria da interface do usuário (Figura 20), tem-se as seguintes contribuições:

- inclusão do assunto da questão;
- incorporação de dicas de respostas;
- padronização dos botões em um único idioma;
- botões "login", "enviar" e "logout" destacados.

Especificamente, no que diz respeito à alteração da ordem das questões, foram proporcionadas as seguintes contribuições:

Figura 20 – Tela com as melhorias na interface



Fonte: Elaborada pela autora

- total de 20 questões que seguem o plano de aula da disciplina (comando de atribuição, incremento e estrutura condicional) categorizadas de acordo com a complexidade estabelecida pelos monitores (Figura 21);
- relatório de informações sobre as questões - informações coletadas pelo sistema - (Figura 22):
  - quanto tempo demora para os alunos responderem cada questão;
  - quantas tentativas foram feitas para cada questão.

Com relação à inclusão de um *feedback* visual para o usuário, isto é, informações coletadas (Figura 23) explicitamente quando o aluno resolve uma questão, tem-se as contribuições listadas a seguir:

- os alunos respondem se gostaram ou não da questão: SIM ou NÃO;
- utilizam a escala Likert de dificuldade para categorizar o esforço na resolução da questão.

Figura 21 – Nova ordem das questões

### Questão 1

*Conteúdo: Atribuição e Comandos Condicionais*

O código abaixo deveria retornar se uma pessoa pode dirigir ou não no Brasil. É sabido que uma pessoa pode dirigir se já tiver 18 anos completos.

Python
Lua
C

```
def CanDrive(age):
    if age > 18:
        age = True
    else:
        age = False
    return age
```

Dica: digite apenas números

Enviar

---

### Questão 2

*Conteúdo: Comandos Condicionais*

O código abaixo deveria retornar a soma de dois números.

Python
Lua
C

```
def soma(a,b):
    if a==b:
        print(a)
    else:
        print(a+b)
```

Dica: digite apenas números

Enviar

---

### Questão 3

*Conteúdo: Comandos Condicionais*

O código deve retornar qual dos números informados é o maior.

Python
Lua
C

```
def ehmaior(a,b):
    if a > b:
        return a
    if b > a:
        return b
```

Dica: digite apenas números

Enviar

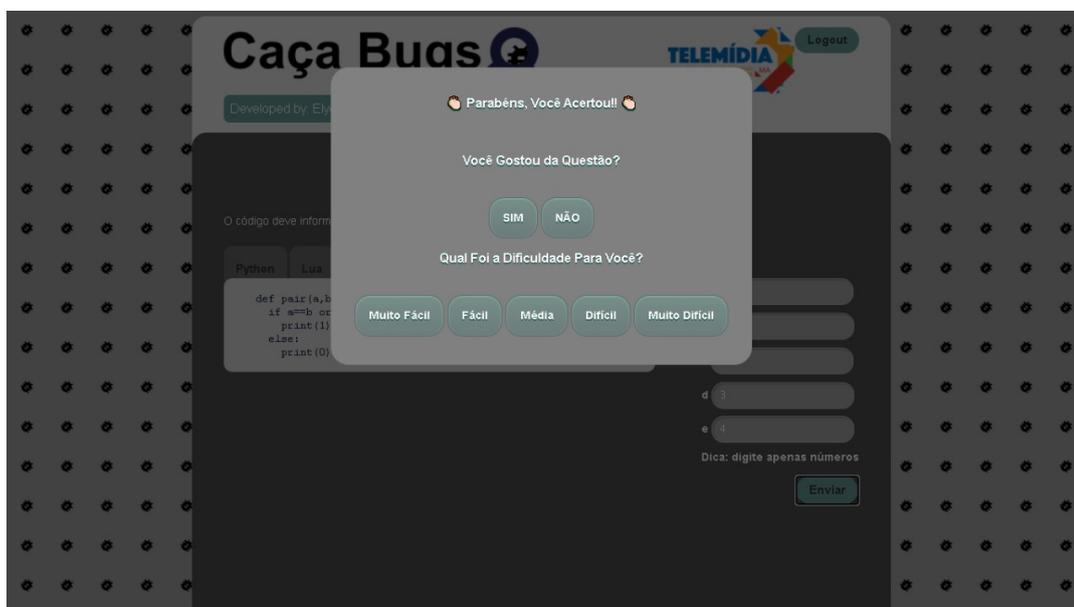
Fonte: Elaborada pela autora

Figura 22 – Relatório de log

	id	id_user	tipo	LogDate	isRight	question	entries	timeSpent
4	312	6	submit	2021-02-17 10:03:12	1	2	a: 10; b: 10	92.076
5	311	6	open	2021-02-17 10:01:40	0	2		0
6	310	6	submit	2021-02-17 10:01:07	1	1	age: 18	7.519
7	309	6	open	2021-02-17 10:00:59	0	1		0

Fonte: Elaborada pela autora

Figura 23 – Tela de *feedback*



Fonte: Elaborada pela autora

## 4 Experimentos e resultados

A avaliação da aplicação desenvolvida ocorreu por meio de experimento realizado com a turma da disciplina de Algoritmos I, do curso de bacharelado em Ciência da Computação da Universidade Federal do Maranhão.

O experimento empírico possuiu caráter qualitativo, propondo avaliar a implicação do Caça Bugs como objeto de aprendizagem através do engajamento dos discentes na resolução das questões, coletando informações e opiniões destes com relação ao uso do ambiente, sua relevância na disciplina e o impacto geral no processo de ensino-aprendizagem, visando melhorias que serão realizadas posteriormente.

### 4.1 Cenário

O experimento foi realizado em uma etapa na disciplina de Algoritmos I, no segundo semestre letivo de 2020. A escolha desse cenário se deu pelo fato de a experiência da turma em relação à análise e depuração de código ser inicial, uma vez que é a primeira disciplina de programação do curso. Outro fator relevante para tal circunstância, foi a de que o orientador deste trabalho é o professor da disciplina, o que garantiu melhor comunicação com os voluntários.

A etapa experimental ocorreu durante o horário da disciplina e consistiu na aplicação e elucidação do experimento, na apresentação e utilização do ambiente Caça Bugs e na aplicação do questionário de perfil demográfico e avaliativo sobre o objeto de aprendizagem e referente às questões.

Em sessão com o professor, foi explicado do que se tratava o experimento, enfatizando que a participação dos alunos era voluntária. Com a concordância, o termo de consentimento (Apêndice A) foi validado por um total de 29 alunos, por meio de formulário criado e disponibilizado na ferramenta Formulário do pacote Google Docs<sup>1</sup>.

Posteriormente, deu-se início à utilização do sistema Caça Bugs pelos discentes. A aplicação constituiu-se em solucionar os problemas e resolver as questões que abrangiam os tópicos de operações aritméticas básicas, comandos de atribuição e comandos condicionais.

No total, foram distribuídas 20 questões, criadas pelos monitores da disciplina e pelo professor. Dos alunos, todos os 29 responderam ao questionário aplicado e participaram da resolução dos problemas.

A seguir, são apresentados os resultados obtidos com a realização do experimento.

---

<sup>1</sup> Para mais informações, acesse: <https://docs.google.com/forms>

## 4.2 Coleta de dados

Nesta seção, aborda-se a coleta dos dados extraídos a partir da utilização do ambiente Caça Bugs na sessão experimental, bem como as perguntas e respostas do questionário aplicado.

O questionário elaborado constituiu-se de 22 questões estruturadas por 4 (quatro) questões a respeito do perfil demográfico do voluntário, 11 (onze) questões sobre a experiência de uso e avaliação da aplicação e 7 (sete) relativas às questões contidas no Caça Bugs. Nessa fase do experimento, obteve-se a participação dos respondentes que testaram o sistema e concordaram em serem voluntários na pesquisa.

As questões que compuseram o questionário destacaram-se em 3 seções: perfil demográfico; avaliação do Caça Bugs; e avaliação sobre as questões do Caça Bugs. No que concerne ao perfil demográfico, foram feitos os seguintes questionamentos:

- 1) qual sua idade?;
- 2) qual seu sexo?;
- 3) você já programava antes de entrar no curso?;
- 4) qual seu nível de conhecimento em ferramentas de ensino de programação?

Com relação à avaliação geral do Caça Bugs, foram elaboradas as seguintes perguntas:

- 1) em sua opinião, qual o objetivo do Caça Bugs?;
- 2) foi fácil utilizar?;
- 3) como você considera o Caça Bugs?;
- 4) destaque as principais dificuldades encontradas (caso exista)?;
- 5) quanto à apresentação (interface, botões, interação etc) do sistema?, fez com que lhe instigasse a explorar suas possibilidades de uso?;
- 6) o Caça Bugs auxiliou na compreensão dos assuntos abordados?;
- 7) o Caça Bugs lhe permitiu uma maneira diferente para compreensão do conteúdo?;
- 8) se você não gostou, porque não gostou?;
- 9) qual sua avaliação sobre o Caça Bugs de maneira geral?;
- 10) para você, qual foi a maior utilidade da aplicação?;

- 11) você considera o sistema intuitivo?

Finalmente, no que diz respeito à avaliação das questões do Caça Bugs, tem-se as seguintes questões:

- 1) com base na sua experiência até agora, você está satisfeito com a aplicação?;
- 2) você sentiu dificuldade em resolver as questões?;
- 3) sobre a disposição das questões, você se sente satisfeito com a ordem em que as questões foram apresentadas?;
- 4) sobre o nível na abordagem das questões, do simples ao mais complexo, como você se sente em relação a isso?;
- 5) você achou as questões repetitivas?;
- 6) você gostou da quantidade de questões apresentadas na tela inicial?;
- 7) quantas questões você sugere que tenha?

Dito isso, na próxima seção são discutidos os resultados da pesquisa de opinião, dispondo-se, sempre que possível, a representação gráfica dos dados coletados.

## 4.3 Análise dos resultados

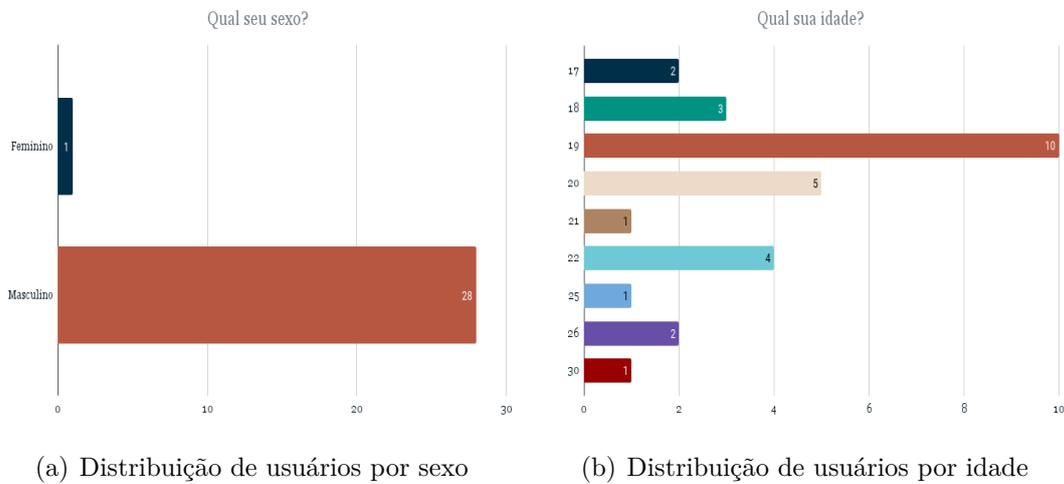
Nesta seção, tem-se como propósito fornecer uma análise prática das informações extraídas do experimento, frisando os dados relacionados ao objeto de aprendizagem e seu uso.

### 4.3.1 Primeira Seção: Perfil Demográfico

O perfil demográfico da amostra é composto por apenas uma pessoa do sexo feminino e 28 pessoas do sexo masculino. Além disso, a faixa etária se concentra no intervalo de 17 a 30 anos, conforme ilustrado na [24](#)

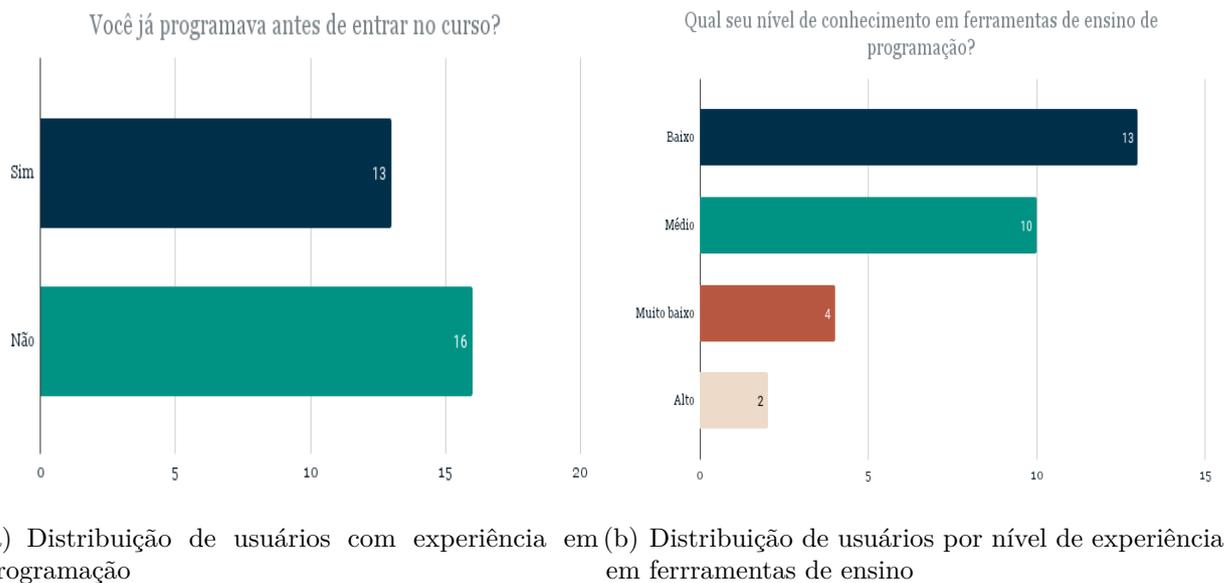
Percebe-se, também, que a maioria (16 alunos) não tinha experiência com programação antes do curso, contrastando com 13 alunos que já possuíam. Além disso, o gráfico mensura o nível de conhecimento em ferramentas de ensino de programação, mostrando que a maioria dos alunos (17) classificaram o quesito como baixo e muito baixo, e consideraram seu nível nessas ferramentas como baixo. Já um terço dos alunos (10) avaliou como médio e apenas 2 alunos consideraram seu nível como alto, caracterizando, assim, como um baixo nível de conhecimento nas ferramentas de ensino de programação.

Figura 24 – Gráfico de distribuição de usuários por sexo e distribuição de usuários por idade



Fonte: Elaborada pela autora

Figura 25 – Gráfico de distribuição de usuários com experiência em programação e distribuição de usuários por nível de experiência em ferramentas de ensino



Fonte: Elaborada pela autora

### 4.3.2 Segunda Seção: Avaliação do Caça Bugs

Em relação ao objetivo do Caça Bugs, houve consenso por parte dos voluntários em julgá-lo como uma ferramenta capaz de auxiliar na depuração de código e contribuir para o aprendizado, como observa-se na [Tabela 1](#). Os dados completos da tabela encontram-se no [Apêndice C](#).

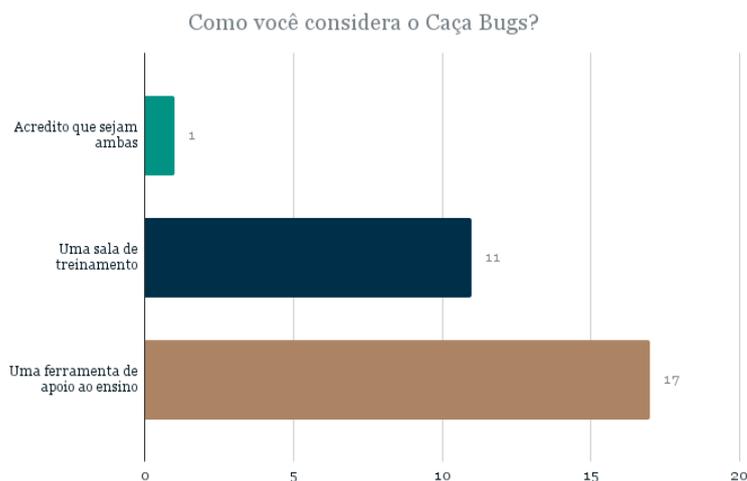
Em virtude disso, a [Figura 26](#) demonstra que 17 voluntários consideram o Caça Bugs como uma ferramenta de apoio ao ensino, 11 tratam como um sala de treinamento e 1 aluno acredita se enquadrar nas duas categorias. Sendo assim, a pesquisa evidencia que o sistema não é uma ferramenta de ensino, mas sim uma atividade de apoio ao ensino.

Tabela 1 – Objetivos do Caça Bugs

Identificar os erros na execução dos códigos
Aprimorar a habilidade de analisar os códigos e encontrar erros
Acredito que ajude a melhorar a percepção na leitura de códigos, e até ensina por que não usar alguma mecânica como (idade>18), que a teoria pode ser corrigida, mas a prática ajuda ainda mais a fixar a informação
Estimular a lógica na programação, através da procura de bugs nos códigos
Ajudar na percepção de bugs na sintaxe de um código
Auxiliar o usuário a entender e ler com mais facilidade bugs e erros
Maximizar o aprendizado do aluno
Ajudar no ensino das linguagens de programação através da identificação de erros
Ajudar a desenvolver habilidades que facilitam na visualização de bugs em seus códigos

Fonte: Elaborada pela autora

Figura 26 – Distribuição de como consideram o Caça Bugs

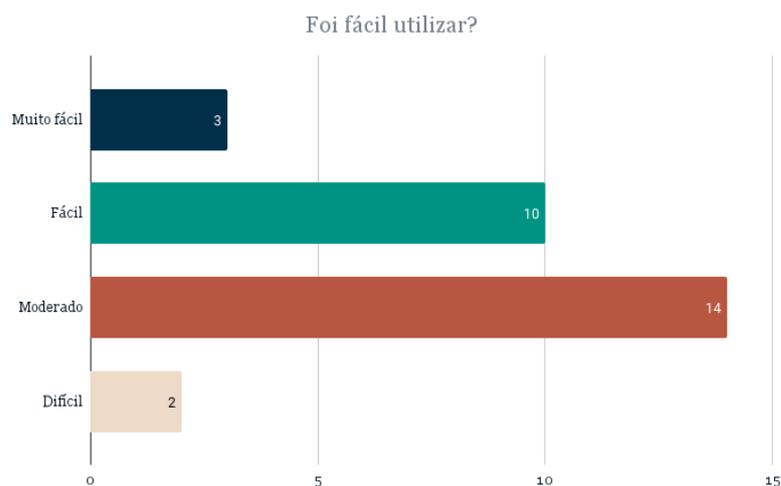


Fonte: Elaborada pela autora

No que tange à sua utilização, 14 alunos classificaram a facilidade de utilização como moderada; 10 consideraram como fácil; 3 como muito fácil; e 2 como difícil. Isso remete à sustentação de que a facilidade de utilização não foi nem muito fácil nem muito difícil, encontra-se, portanto, em um patamar intermediário. Tendo isso em vista, um estudo de usabilidade mais detalhado seria de grande valia, como revelam os dados do

gráfico abaixo. Além disso, dentre as principais dificuldades encontradas (Apêndice C), um dos voluntários menciona que "seria melhor que tivesse um tutorial" e outro relata "não ter um página inicial para explicar a finalidade do caçar *bugs*, mesmo o título sendo auto explicativo", sendo um fator determinante a superação de tal obstáculo.

Figura 27 – Distribuição da facilidade na utilização



Fonte: Elaborada pela autora

Sucedendo a isso, das 29 respostas, 11 delas são interpretadas como alunos que não tiveram nenhuma dificuldade, 3 delas são de alunos que estão muito confusos em relação à interface, como relatado: "não é muito intuitivo" e "achei a interface do site pouco explicativa para o propósito do site". Em relação a isso, mencionaram também dificuldade no que tange ao enunciado das questões: "entender entender o código" e "a compreensão dos problemas". A partir disso, conclui-se que os enunciados não são didáticos, faltando possuir teor mais explicativo. Na Tabela 2 algumas dessas dificuldades são listadas.

A avaliação quanto à apresentação do sistema e suas possibilidades de uso revela que 15 alunos sentiram-se instigados e a 11 não se sentiram estimulados a nada, fazendo-se considerar que há uma narrativa em que o público não julgou a interface intuitiva. Portanto, metade dos alunos se sentiram instigados e outra metade não, instituindo um paradoxo no qual, em uma face, as atividades estimulam os alunos e, em outra, a interface os desaponta. Na Figura 28, observa-se na essa distribuição.

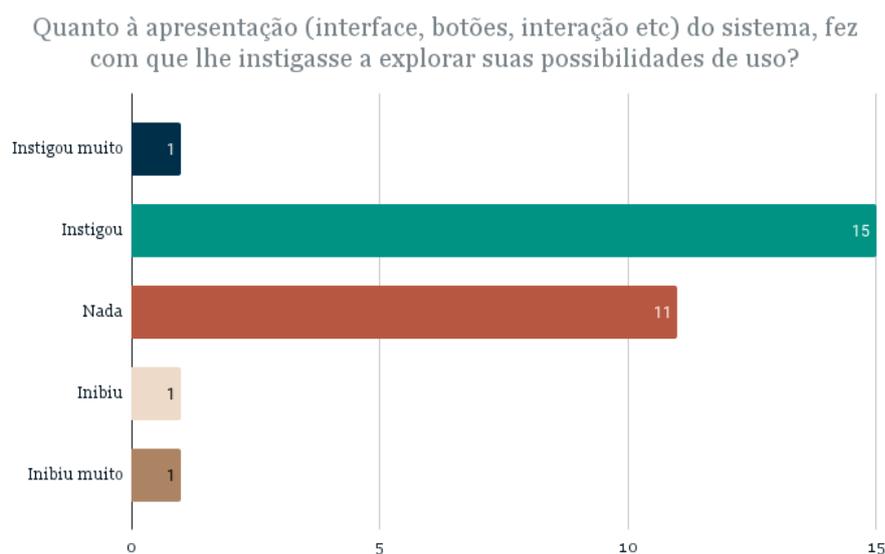
No que diz respeito à compreensão dos assuntos abordados, ninguém relatou ter se sentido prejudicado. Dos 29 voluntários, 14 afirmaram que o sistema colaborou na compreensão do assunto abordado; 4 disseram que o sistema não ajudou; e 11 responderam que ajudou um pouco. Entretanto, no tocante à maneira diferente para compreensão

Tabela 2 – Principais dificuldades encontradas

Não ter um pagina inicial para explicar a finalidade do caçar bugs, mesmo o título sendo auto explicativo acho que seria bom ter;
Às vezes as questões são meio confusas;
Achei a interface do site pouco explicativa para o propósito do site;
O site poderia ter uma explicação melhor, do objetivo da plataforma. Só entendi o propósito porque o professor falou antes, então ele não é muito intuitivo;
Não encontrei dificuldades no manuseio da ferramenta, porém pode evoluir nos enunciados;
As questões não são muito didáticas;
Não deixar claro o que temos que colocar no caixa de respostas, se é o código certo ou se é uma resposta que deixe o código bugado;
A compreensão dos problemas;
Explicar a finalidade e como operar o app;
A principal dificuldade foi entender o que os enunciados estavam pedindo. Falta informação.

Fonte: Elaborada pela autora

Figura 28 – Distribuição quanto à apresentação do sistema



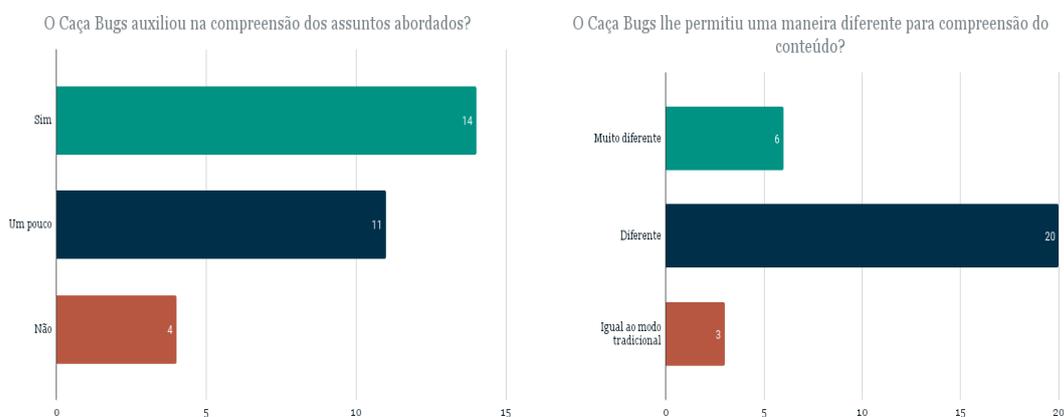
Fonte: Elaborada pela autora

do conteúdo, uma ampla maioria (20 alunos) classificou o Caça Bugs como uma forma diferente no contexto de aprender o conteúdo; 6 como muito diferente; e 3, igual ao modo tradicional.

Dentre os que não gostaram da aplicação, somente um voluntário destacou sua insatisfação em como observa-se na [Figura C8](#) do [Apêndice C](#).

De maneira geral, há um consenso na avaliação sobre o Caça Bugs, pois 22 voluntários disseram que gostaram e 7 que gostaram muito. Há, portanto, uma separação entre os eixos do quão positiva é a ferramenta e das dificuldades encontradas, como a

Figura 29 – Gráfico de distribuição baseada na compreensão dos assuntos abordados e na maneira utilizada para a compreensão do conteúdo

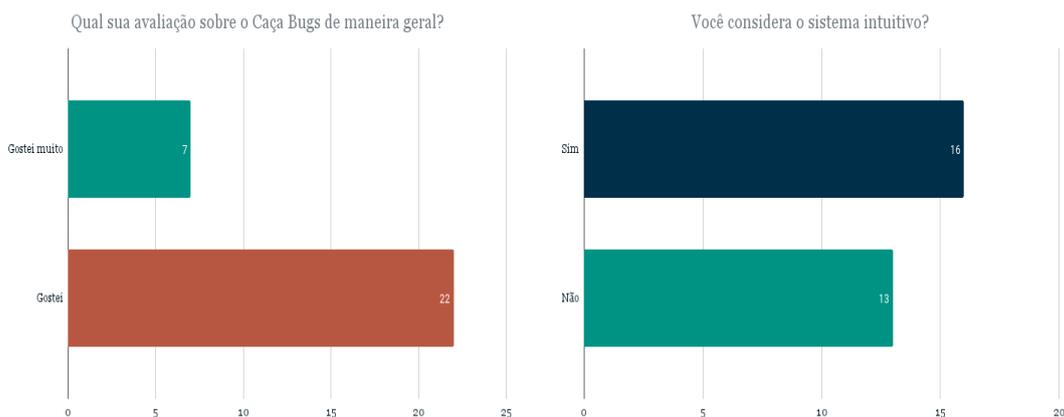


(a) Distribuição baseada na compreensão dos assuntos abordados (b) Distribuição da maneira utilizada para a compreensão do conteúdo

Fonte: Elaborada pela autora

carência de um tutorial introdutório sobre o sistema para torná-lo intuitivo. Particularmente, no que diz respeito à intuitividade do sistema, quase metade dos alunos (16) respondeu de forma positiva, e o restante dos alunos (13) posicionou-se de maneira negativa, como pode-se ver na 30.

Figura 30 – Gráfico de distribuição da avaliação geral do Caça Bugs e baseada na intuitividade do sistema



(a) Distribuição da avaliação geral do Caça Bugs (b) Distribuição baseada na intuitividade do sistema

Fonte: Elaborada pela autora

No que se refere à maior utilidade da aplicação, é perceptível o grau de aceitação por parte dos alunos, o que se reflete nas assertivas contidas na Tabela 3.

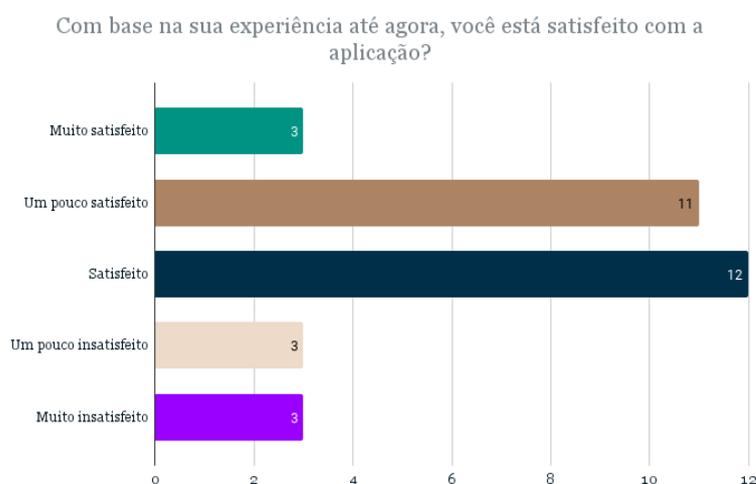
Tabela 3 – Principais utilidades encontradas

Aprendizado;
Melhorar o aprendizado do aluno;
Um abordagem diferente das aulas de programação e uma visão mais criativa sobre os códigos;
Estimulou o senso de detecção de bugs;
Testar meu conhecimento em reconhecer se estou compreendendo os códigos durante aprendizado no curso;
Treinar a mente, otimizando o tempo, para ver o bug de tal código;
Poder treinar algo em que tenho dificuldade;
Testar minha capacidade de raciocínio para ver em quais condições o código não seria executado.

Fonte: Elaborada pela autora

Diante disso, 12 voluntários encontram-se satisfeitos com a aplicação, 11 estão um pouco satisfeitos, 3 muito insatisfeitos e 3 um pouco insatisfeitos, o que remete ao fato de que o sistema tem espaço para melhorar, como aponta a [Figura 31](#).

Figura 31 – Distribuição do grau de satisfação com o Caça Bugs



Fonte: Elaborada pela autora

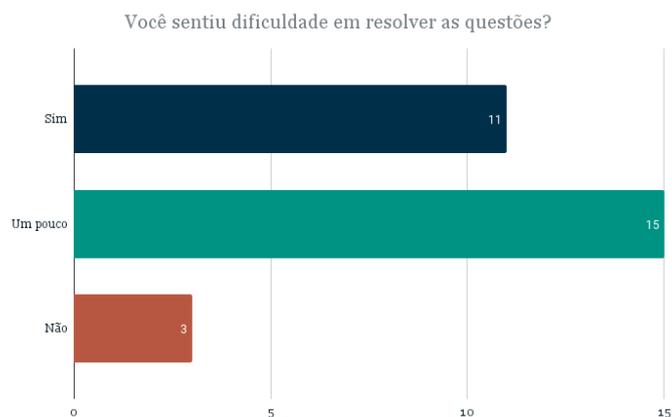
### 4.3.3 Terceira Seção: Avaliação sobre as Questões do Caça Bugs

Na indagação sobre a dificuldade em resolver as questões, 15 alunos sentiram-se com um pouco de dificuldade, 11 sentiram-se com dificuldade e 3 não, o que provoca uma reflexão quanto ao enunciado das questões, como apresentado na [Figura 32](#), retomando o que foi exposto na [Tabela 2](#).

Relativamente, na disposição das questões mencionadas, tem-se que: 17 voluntários enquadram-se como satisfeitos; 5 como um pouco satisfeitos; 2 como muito satisfeitos, 3 como um pouco insatisfeitos; 1 como insatisfeito; e 1 como muito insatisfeito. Sendo

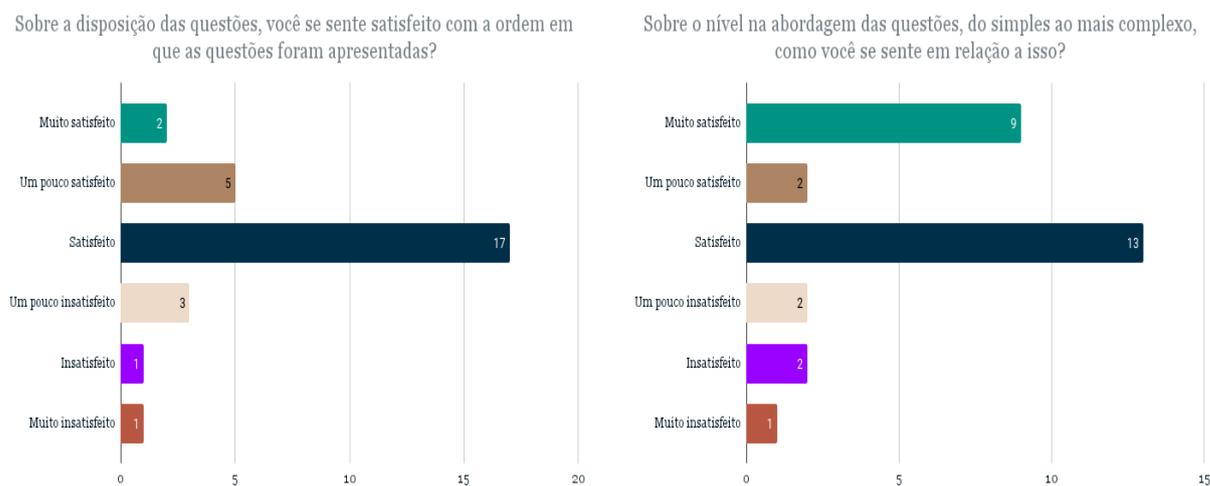
assim, 24 deles não se deparam com problemas na ordem das questões. Não obstante, a disposição das questões foi trabalhada manualmente, demandando um tempo considerável. Todavia, inquestionavelmente, obteve-se um resultado satisfatório como representou a amostra, tendo em vista a imprescindível organização das questões de forma automática. Na 33 isso é nítido.

Figura 32 – Distribuição do grau de dificuldade no uso do Caça Bugs



Fonte: Elaborada pela autora

Figura 33 – Gráfico de distribuição da disposição das questões e baseada na complexidade



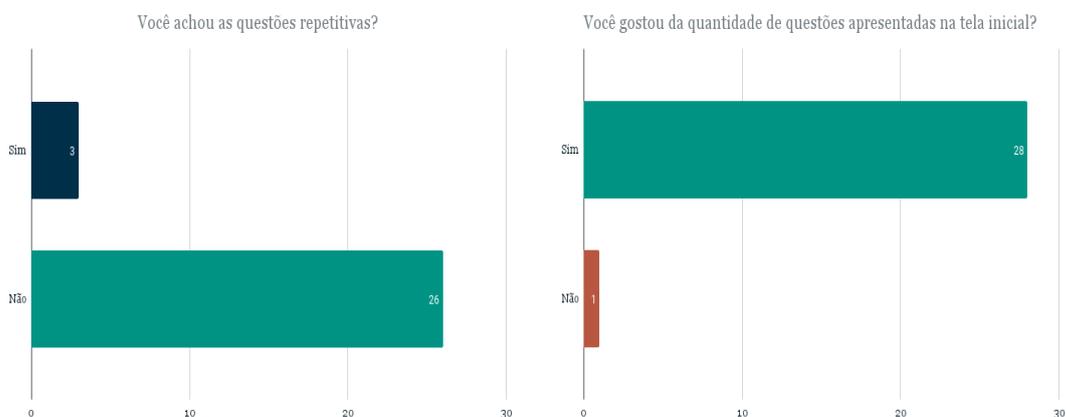
(a) Distribuição da disposição das questões

(b) Distribuição baseada na complexidade

Fonte: Elaborada pela autora

Sobre o grau de repetitividade das questões (34), houve unanimidade de 26 voluntários, declarando que não as consideraram repetitivas e 3 que consideraram. Paralelamente a isso, na avaliação sobre a quantidade de questões apresentadas, 28 voluntários gostaram do que foi disponibilizado e apenas 1 voluntário não gostou.

Figura 34 – Gráfico de distribuição da repetitividade das questões e referente ao números de questões

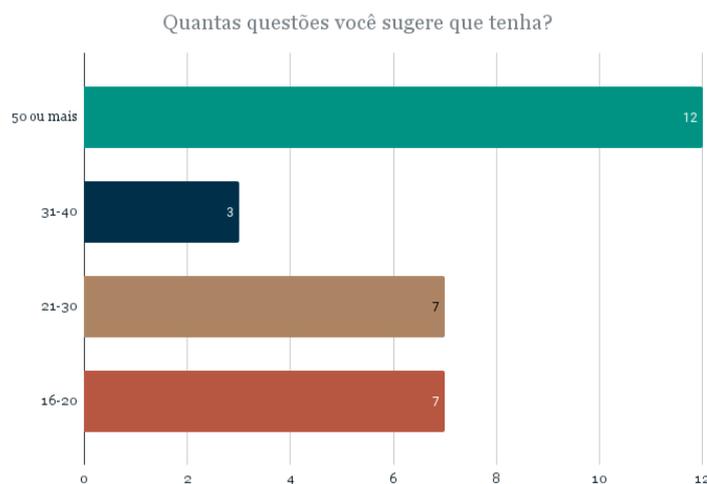


(a) Distribuição da repetitividade das questões (b) Distribuição referente ao números de questões

Fonte: Elaborada pela autora

Soma-se a isso, a proposta de acréscimo na quantidade de questões, em que: 7 voluntários sugeriram haver entre 16 e 20 questões; 7 voluntários entre 21 e 30 questões intercalando os níveis; 3 voluntários sugeriram haver de 31 a 40 questões; e 12 voluntários sugeriram a possibilidade de 50 questões ou mais, como representa a [Figura 35](#).

Figura 35 – Distribuição da quantidade sugerida de questões



Fonte: Elaborada pela autora

Diante disso, o resultado do formulário mostra que, dentre o grupo de voluntários, uma ampla maioria gostou da ferramenta. No que tange aos problemas, tem-se os eixos: o primeiro consiste em melhorar a interface da ferramenta e explicar o seu uso por meio de tutoriais; o outro eixo diz respeito à necessidade de melhorias nos enunciados, que devem

ser melhor explicados; e nas questões, que devem ser separadas e categorizadas, como sugerido.

#### 4.3.4 Dados referentes ao desempenho dos alunos

Neste tópico, tem-se como objetivo levantar os dados coletados relativos ao empenho dos voluntários na conclusão das atividades. Com relação ao nível de detalhamento, é importante mencionar que somente foram contabilizados os tempos de 19 das 20 questões, por motivo técnico identificado pela ausência de captura do evento de saída do sistema na última questão.

A extração dos dados dos usuários, a partir da tabela *log* e da tabela *evaluation*, visa complementar a análise dos resultados fornecendo estatísticas concernentes aos alunos e às questões, respectivamente. Portanto, apresentam-se duas tabelas, sendo a primeira designada como tabela de alunos (Tabela 4), que concentra dados ligados aos voluntários, assim denominados à título de anonimato; a quantidade de questões resolvidas; as questões que o aluno gostou; a média de avaliação das questões verificadas na Figura 23; o conceito (média de avaliação) referente à questão; seguido pelo tempo total estimado (em minutos).

Para a estimativa, pode-se calcular a quantidade de atividades concluídas por cada aluno, sendo que essa informação permite a análise do desempenho individual, não sendo requisito a obtenção de níveis máximos ou a realização de todas as questões, ficando à critério de cada um.

Considerando a tabela, percebe-se que a quantidade de questões feitas e a quantidade de questões apreciadas possuem valores correlativos, com média de avaliação que varia conforme a escala Likert (com valores de 1 a 5), tendo o conceito correspondente à média e tempos sortidos. Em uma análise minuciosa, o maior número de questões se aplica aos voluntários 4, 5, 14, 18 e 19, em que o conceito 2 (fácil) e 3 (médio) se repetem, sendo o tempo, um valor razoável para completar as tarefas.

Diante disso, na Tabela 5 calcula-se a correlação das taxas referente ao total de alunos por questões, que consiste na identificação individualizada de cada questão, contendo a identificação da mesma; a quantidade de alunos que a resolveram corretamente; a quantidade dos que gostaram; a média de avaliação dada; o conceito correspondente; e o tempo médio (em minutos) do total utilizado para resolução de cada uma.

Em geral, nota-se que há um declínio no número de questões à medida que a sequência das questões avança simultaneamente à quantidade de alunos que gostaram da questão. Para a média de avaliação das questões, situa-se entre fácil e médio, sendo o tempo mínimo de 1,23 minutos conceituado como fácil, para a questão 17, com 5 conclusões e 5 alunos que gostaram.

Tabela 4 – Tabela de Alunos

Voluntário	Questões feitas	Questões que gostou	Média de avaliação	Conceito	Tempo total (min)
1	8	8	2	Fácil	29,36
2	5	5	2	Fácil	23,77
3	6	6	5	Muito Difícil	56,04
4	19	19	3	Médio	56,54
5	19	19	2	Fácil	82,12
6	13	13	2	Fácil	25,89
7	4	4	2	Fácil	22,34
8	5	5	3	Médio	48,42
9	5	5	3	Médio	43,98
10	7	7	2	Fácil	27,37
11	12	12	3	Médio	36,63
12	2	2	3	Médio	249,25
13	15	15	2	Fácil	54,98
14	19	19	2	Fácil	1551,12
15	6	6	3	Médio	13,41
16	8	8	2	Fácil	35,28
17	9	9	3	Médio	54,82
18	18	18	2	Fácil	62,67
19	19	19	3	Médio	173,3
20	7	7	3	Médio	51,6
21	3	3	2	Fácil	27,39
22	2	2	1	Muito Fácil	0,56
23	7	7	2	Fácil	32,21
24	13	13	4	Difícil	1377,57
25	6	6	2	Fácil	51,83
26	11	11	3	Médio	54,26
27	8	8	3	Médio	28,11
28	9	9	2	Fácil	54,67
29	14	14	4	Difícil	70,41

Fonte: Elaborada pela autora

Isso se aplica ao contraste encontrado na questão 14, com o maior índice de tempo (225,32 minutos) caracterizada como difícil, de média 4, com 6 alunos que gostaram e 7 concluintes.

Tabela 5 – Tabela de Questões

Identificação da questão	Quantos resolveram	Alunos que gostaram	Média de avaliação	Conceito	Tempo médio (m)
1	28	28	2	Fácil	12,57
2	29	28	2	Fácil	2,44
3	27	27	2	Fácil	2,26
4	26	23	2	Fácil	5,63
5	29	26	3	Médio	6,72
6	26	25	3	Médio	5,75
7	24	21	3	Médio	3,6
8	17	17	3	Médio	3,72
9	12	12	3	Médio	6,2
10	12	12	3	Médio	11,34
11	11	11	3	Médio	115,63
12	10	10	2	Fácil	0,94
13	9	9	2	Fácil	21,26
14	7	6	4	Difícil	225,32
15	8	8	3	Médio	1,52
16	5	5	2	Fácil	2,52
17	5	5	2	Fácil	1,23
18	5	4	2	Fácil	6,77
19	5	5	3	Médio	0

Fonte: Elaborada pela autora

## 5 Considerações finais

Tendo em vista os aspectos mencionados sobre a elevada taxa de evasão e reprovação nos cursos de ensino superior brasileiros na área de computação, observa-se um grande fator centrado na dificuldade de tornar inteligíveis os conteúdos do plano curricular. Com base nisso, este trabalho teve o intuito de aplicar um ambiente virtual de ensino promovendo uma ruptura na metodologia tradicional de ensino de algoritmos, proporcionando um sentimento de motivação e percepção prática.

Nessa conjuntura, foi aplicado um sistema web voltado ao desenvolvimento da capacidade analítica e de resolução de problemas dos estudantes. Essa ferramenta foi denominada como Caça Bugs pelo fato de que os alunos são requisitados a desvendar erros propositais inseridos em códigos, corrigi-los e apresentar a resposta do problema desconsiderando a existência do erro.

A partir disso, foi possível ter um retorno do experimento empírico coletado por meio de questionário de caráter qualitativo, aplicado após 24 horas de uso do sistema, que indagou os voluntários acerca da experiência durante o uso; da influência e do grau de aceitação do sistema, e sobre as questões nele contidas. Também foi feita a análise das informações da tabela de *logs*, expresso pelo nível de satisfação do participante e se o mesmo gostou ou não da questão fornecida.

De acordo com o resultado obtido na pesquisa, verificou-se que uma ampla maioria gostou da ferramenta. No que compete aos problemas, foram encontradas duas vertentes: a primeira consiste em melhorar a interface da ferramenta e entregar uma explicação mais didática do seu uso por meio de tutoriais; a segunda diz respeito à necessidade de melhorias nos enunciados, que devem ser melhor elaborados; e nas questões, que devem ser categorizadas e separadas por conteúdo.

Desse modo, ratifica-se que o Caça Bugs cumpriu com o seu objetivo, vez que foi avaliado de forma positiva pelos voluntários que participaram do experimento. Todavia, há espaço para a inclusão de melhorias visando a evolução do sistema quanto à usabilidade, intuitividade, organização das questões por categorias, reformulação dos enunciados; tudo isso poderia intercorrer em uma maior influência na aprendizagem de algoritmos.

Posto isso, destacam-se as principais dificuldades encontradas durante o desenvolvimento deste trabalho:

1. a inexistência de documentação da versão anterior do sistema, que exigiu um tempo maior para compreensão do funcionamento do sistema;
2. a não utilização do email institucional por alguns alunos, que dificultou o cruzamento

de dados dos usuários;

3. o tempo exíguo para elaboração do trabalho monográfico (apenas 2 (dois) meses), que impossibilitou uma análise aprofundada dos dados coletados;
4. a constatação de uma falha na lógica de captura e contabilização do evento de saída do sistema, que, todavia, não comprometeu a análise realizada neste trabalho.

Por fim, as ideias que foram pensadas, e não incluídas na versão desenvolvida neste trabalho, mas que podem ser acrescentadas futuramente, são:

1. criar um manual (ou tutorial) de uso do sistema;
2. suportar outras linguagens de programação, tais como C++ e Java;
3. incorporar um mecanismo de suporte a múltiplos idiomas;
4. acrescentar um sistema de recomendação de questões baseado no *log* de informações e no *feedback* dos alunos;
5. incluir novas questões e revisar as cadastradas atualmente.

# Apêndices

# APÊNDICE A – Termo de Consentimento Livre e Esclarecido Adaptado

Figura A1 – Termo de Consentimento Livre e Esclarecido Adaptado - Página 1

## AVALIAÇÃO DO OBJETO DE APRENDIZAGEM CAÇA BUGS

Laboratório de Sistemas Multimídia (Telemídia-MA)

Seu endereço de e-mail será registrado quando você enviar este formulário.

Não é [elydillse.lindoso@discente.ufma.br](mailto:elydillse.lindoso@discente.ufma.br)? [Trocar de conta](#)

\*Obrigatório

---

**Termo de Consentimento Livre e Esclarecido**

Eu autorizo o uso, como fins estritamente acadêmicos, das informações fornecidas durante a entrevista sobre o uso de tecnologias na educação, realizada no dia 05/04/21, a partir das 14hs.

Estou ciente de que:

- (1) minha participação é voluntária;
- (2) a sessão será registrada com anotações e captura de texto;
- (3) esta sessão visa prover informações a uma pesquisa acadêmica relacionada ao uso de tecnologias educacionais, desenvolvido por Elydillse Maria Botelho Lindoso, sob orientação do professor Carlos Salles Soares Neto, na Universidade Federal do Maranhão;
- (4) será garantido o anonimato no uso de informações capturadas nesta sessão;
- (5) todos os dados brutos serão acessados exclusivamente pelos pesquisadores envolvidos nesta pesquisa.
- (6) a qualquer momento, até dois anos após o término da pesquisa, poderei solicitar mais informações sobre o estudo ou cópias dos materiais divulgados, entrando em contato com a pesquisadora através do email: [elydillse.lindoso@discente.ufma.br](mailto:elydillse.lindoso@discente.ufma.br)

---

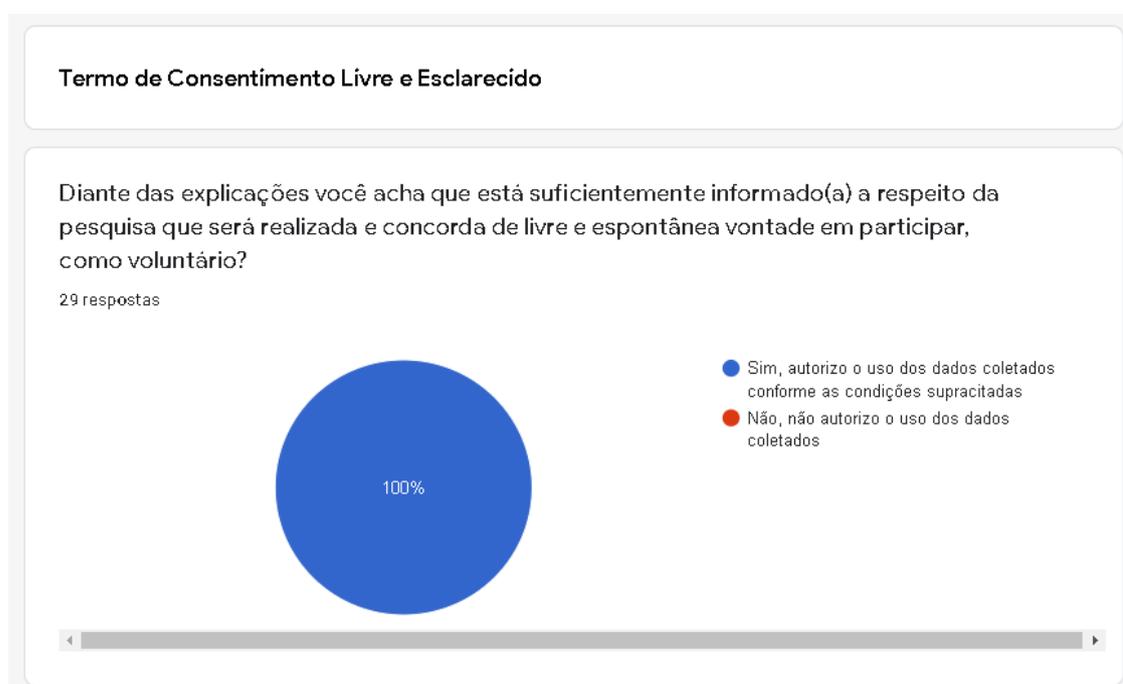
Diante das explicações você acha que está suficientemente informado(a) a respeito da pesquisa que será realizada e concorda de livre e espontânea vontade em participar, como voluntário? \*

Sim, autorizo o uso dos dados coletados conforme as condições supracitadas

Não, não autorizo o uso dos dados coletados

Fonte: Fonte: Elaborado pela autora

Figura A2 – Resultado do Termo de Consentimento Livre e Esclarecido Adaptado - Página 2



Fonte: Elaborado pela autora

# APÊNDICE B – Questionário de Avaliação do Caça Bugs

Figura B1 – Perfil de Análise Demográfica

**Perfil do Entrevistado**

Qual seu nome? \*

Sua resposta \_\_\_\_\_

Qual a sua idade? \*

Sua resposta \_\_\_\_\_

Qual seu sexo? \*

Feminino

Masculino

Você já programava antes de entrar no curso? \*

Sim

Não

Qual seu nível de conhecimento em ferramentas de ensino de programação? \*

Muito baixo

Baixo

Médio

Alto

Muito alto

Fonte: Elaborado pela autora

Figura B2 – Avaliação do Caça Bugs - Página 1

**Avaliação do Caça Bugs**

Este questionário tem o intuito de colher informações acerca o objeto de aprendizagem Caça Bugs, visando melhorias que possivelmente serão definidas por você.

Em sua opinião, qual o objetivo do Caça Bugs?

Sua resposta \_\_\_\_\_

Foi fácil utilizar? \*

Muito fácil

Fácil

Moderado

Difícil

Muito difícil

Como você considera o Caça Bugs \*

Uma ferramenta de apoio ao ensino

Uma sala de treinamento

Outro: \_\_\_\_\_

Destaque as principais dificuldades encontradas (caso exista) \*

Sua resposta \_\_\_\_\_

Fonte: Elaborado pela autora

Figura B3 – Avaliação do Caça Bugs - Página 2

Quanto à apresentação (interface, botões, interação etc) do sistema, fez com que lhe instigasse a explorar suas possibilidades de uso? \*

Instigou muito

Instigou

Nada

Inibiu

Inibiu muito

Caça Bugs auxiliou na compreensão dos assuntos abordados? \*

Atrapalhou a compreensão

Não

Um pouco

Sim

Caça Bugs lhe permitiu uma maneira diferente para compreensão do conteúdo? \*

Muito diferente

Diferente

Igual ao modo tradicional

Se você não gostou, porque não gostou?

Sua resposta \_\_\_\_\_

Fonte: Elaborado pela autora

Figura B4 – Avaliação do Caça Bugs - Página 3

Qual sua avaliação sobre o Caça Bugs de maneira geral? \*

Desgostei muito

Desgostei

Irrelevante

Gostei

Gostei muito

Para você, qual foi a maior utilidade da aplicação? \*

Sua resposta \_\_\_\_\_

Você considera o sistema intuitivo? \*

Sim

Não

Fonte: Elaborado pela autora

Figura B5 – Avaliação das Questões do Caça Bugs - Página 4

**Avaliação Sobre as Questões do Caça Bugs**

Aqui, vamos discutir sobre as questões do sistema

Com base na sua experiência até agora, você está satisfeito com a aplicação? \*

Muito satisfeito

Um pouco satisfeito

Satisfeito

Um pouco insatisfeito

Insatisfeito

Muito insatisfeito

Você sentiu dificuldade em resolver as questões? \*

Sim

Um pouco

Não

Sobre a disposição das questões, você se sente satisfeito com a ordem em que as questões foram apresentadas? \*

Muito insatisfeito

Insatisfeito

Um pouco insatisfeito

Satisfeito

Um pouco satisfeito

Muito satisfeito

Fonte: Elaborado pela autora

Figura B6 – Avaliação das Questões do Caça Bugs - Página 5

Sobre o nível na abordagem das questões, do simples ao mais complexo, como você se sente em relação a isso? \*

Muito insatisfeito

Insatisfeito

Um pouco insatisfeito

Satisfeito

Um pouco satisfeito

Muito satisfeito

Você achou as questões repetitivas? \*

Sim

Não

Você gostou da quantidade de questões apresentadas na tela inicial? \*

Sim

Não

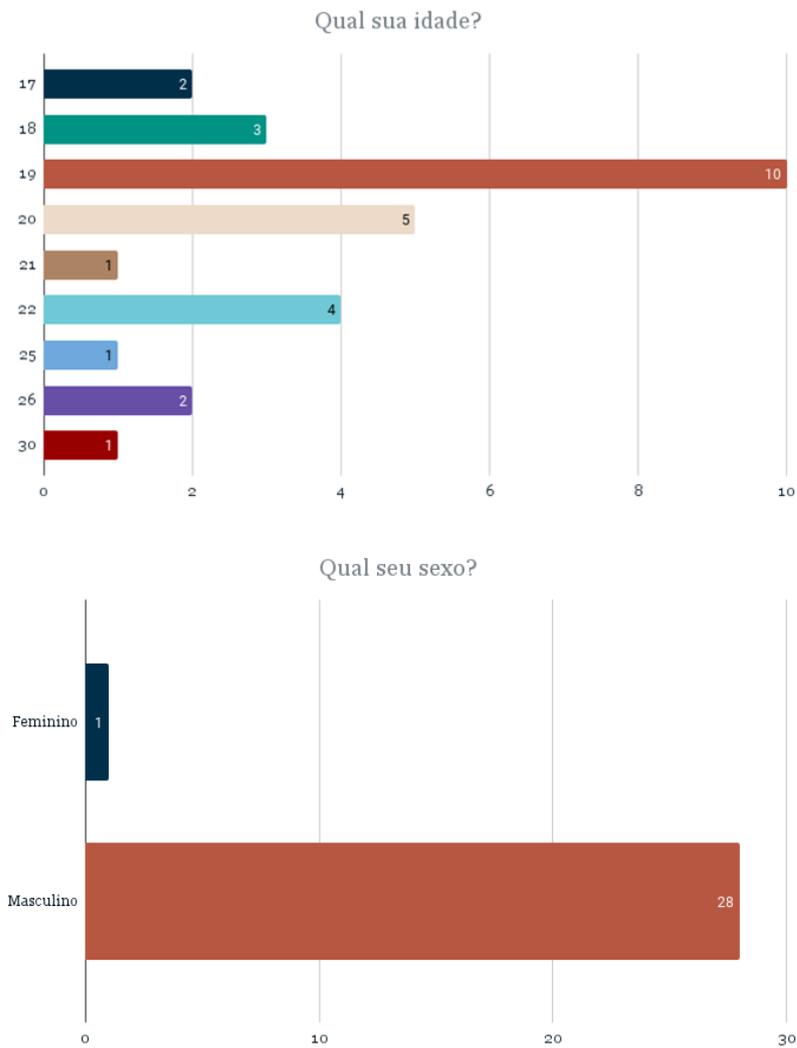
Quantas questões você sugere que tenha? \*

Sua resposta \_\_\_\_\_

Fonte: Elaborado pela autora

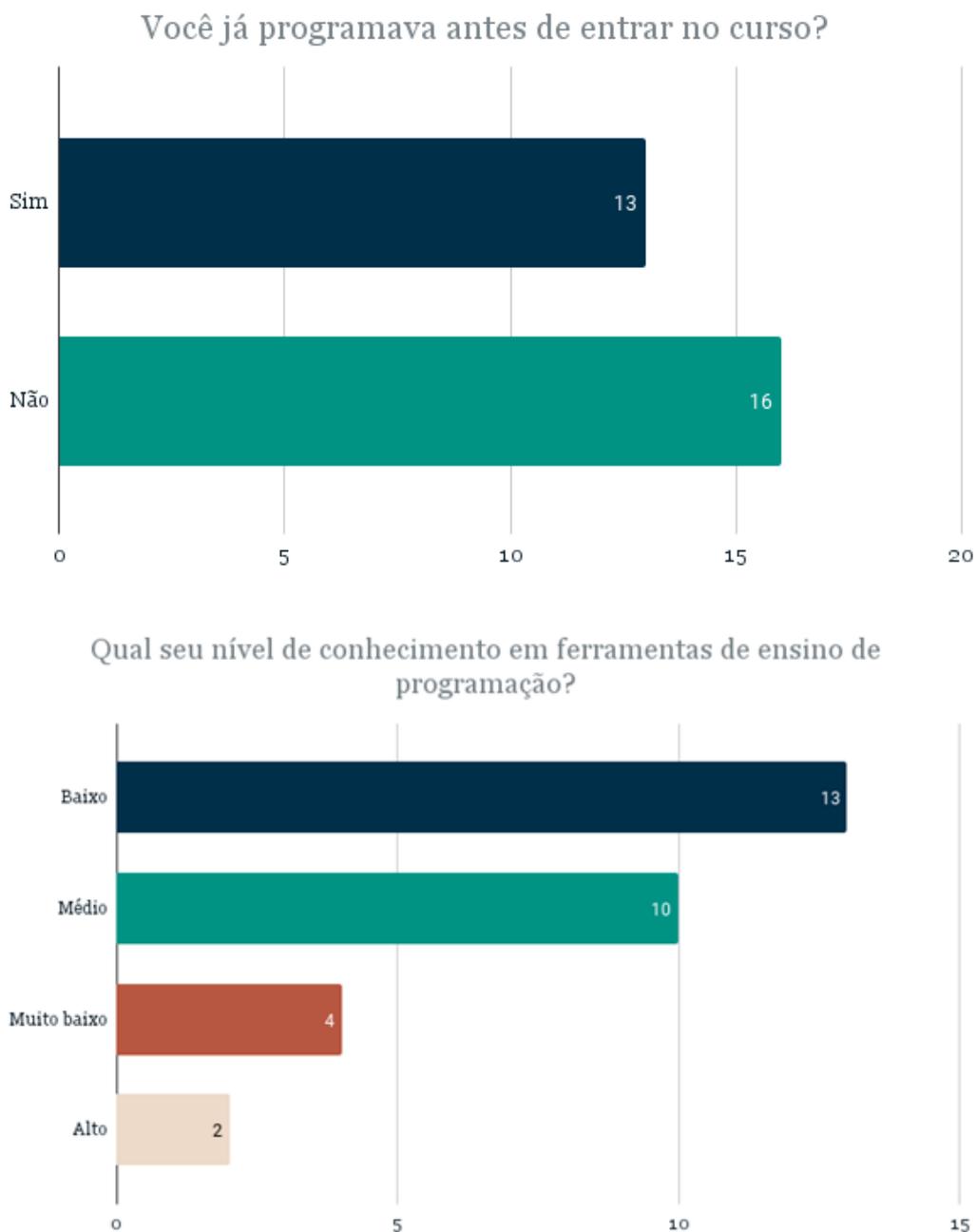
# APÊNDICE C – Resultados do Questionário de Avaliação do Caça Bugs

Figura C1 – Resultado Perfil de Análise Demográfica - Página 1



Fonte: Elaborado pela autora

Figura C2 – Resultado Perfil de Análise Demográfica - Página 2



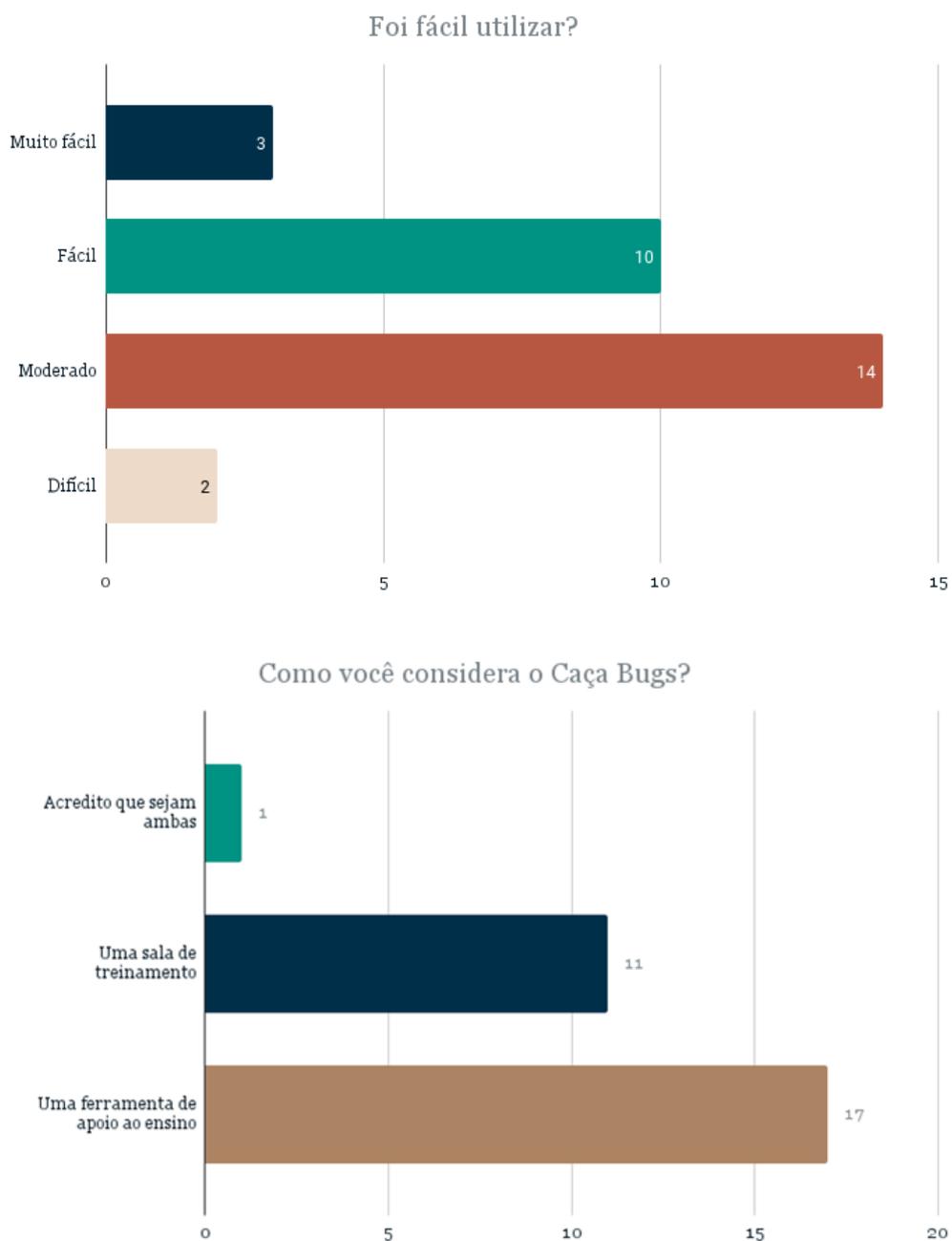
Fonte: Elaborado pela autora

Figura C3 – Resultado da Avaliação do Caça Bugs - Página 1



Fonte: Elaborado pela autora

Figura C4 – Resultado da Avaliação do Caça Bugs - Página 2



Fonte: Elaborado pela autora

## Figura C5 – Resultado da Avaliação do Caça Bugs - Página 4

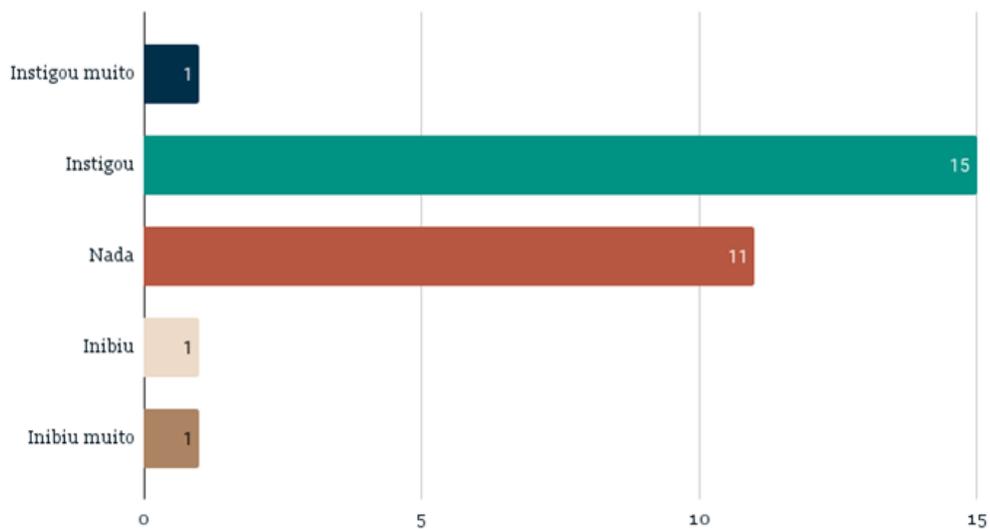
Destaque as principais dificuldades encontradas (caso exista)

29 respostas

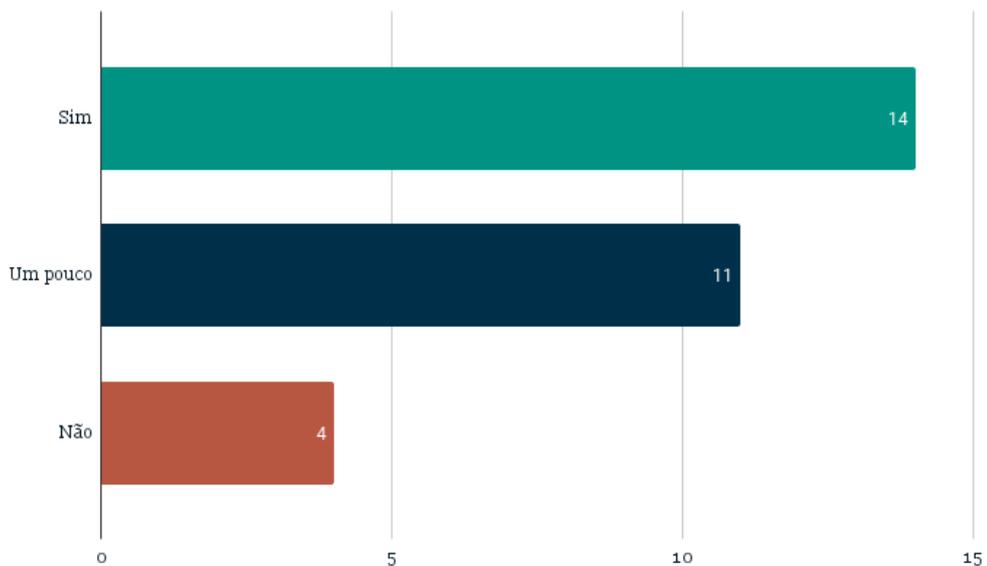
Em termos de explicações existe uma pendência nas questões , pq algumas falta algo a mais .
Não encontrei nada difícil.
não encontrei dificuldades no manuseio da ferramenta, porém pode evoluir nos enunciados
Não encontrei.
Entender entender o código
O site não se introduz, fica difícil de saber o seu uso.
Entender a como funciona a ferramenta.
Não existe
Alguns erros nas funções(como na 14, que a função em python necessita das variáveis a, b e c mas no exemplo só são passados a e b.)
Achei a interface do site pouco explicativa para o propósito do site
Algumas questões não estão tão claras
não tive muita dificuldade
Não é muito intuitivo.
Os enunciados de algumas questões que dificultam a interpretação da resposta
não
não deixar claro o que temos que colocar no caixa de respostas, se é o código certo ou se é uma resposta que deixe o código bugado
Dificuldade de entender como funciona
entender as questões
As questões não são muito didáticas
alguns enunciados são confusos
A compreensão dos problemas
Acredito que tirando a função citada pelo professor Carlos de Salles Soares ( Esqueci minha senha) não existe algo tão discrepante.
entender oque se pede as informações a ser posta.
explicar a finalidade e como operar o app
Não ter um pagina inicial para explicar a finalidade do caçar bugs, mesmo o titulo sendo alto explicativo acho que seria bom ter.
A principal dificuldade foi entender o que os enunciados estavam pedindo. Falta informação.
falta de mais explicações.

Figura C6 – Resultado da Avaliação do Caça Bugs - Página 5

Quanto à apresentação (interface, botões, interação etc) do sistema, fez com que lhe instigasse a explorar suas possibilidades de uso?



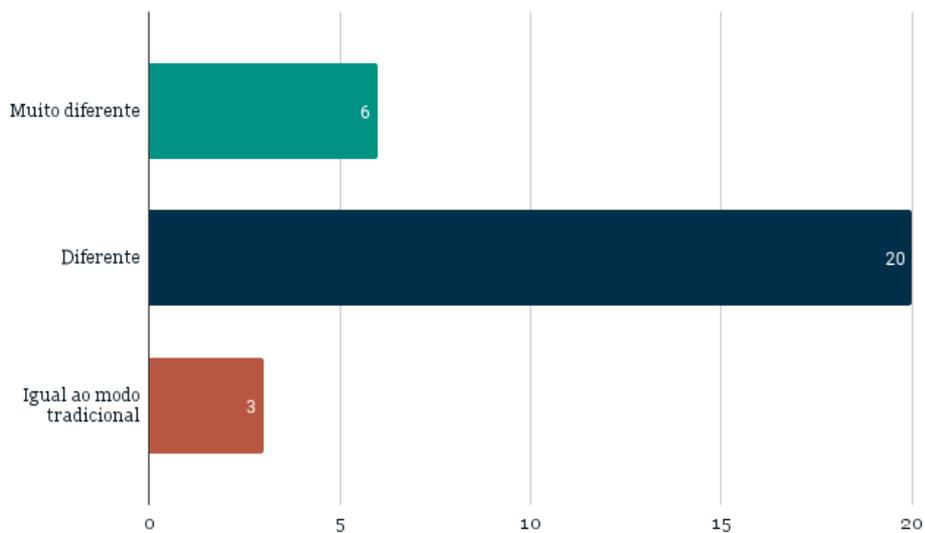
O Caça Bugs auxiliou na compreensão dos assuntos abordados?



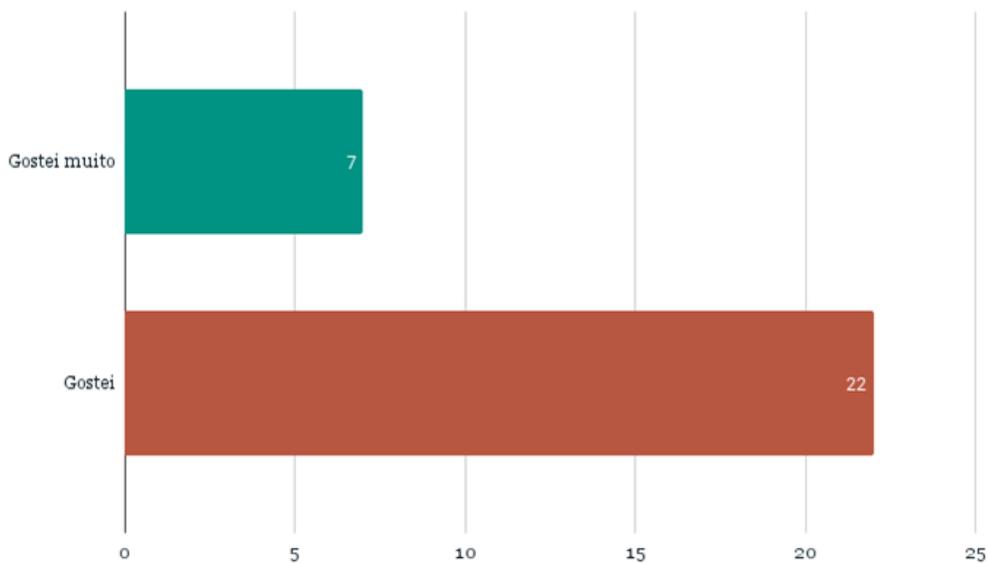
Fonte: Elaborado pela autora

Figura C7 – Resultado da Avaliação do Caça Bugs - Página 6

O Caça Bugs lhe permitiu uma maneira diferente para compreensão do conteúdo?



Qual sua avaliação sobre o Caça Bugs de maneira geral?



Fonte: Elaborado pela autora

Figura C8 – Resultado da Avaliação do Caça Bugs - Página 7

Se você não gostou, porque não gostou?

5 respostas

Bom a plataforma não intuitiva

Eu particularmente gostei, me fez querer melhorar detalhes q eu nunca procuro melhorar

gostei po rlx

Gostei sim

gostei bastante

Fonte: Elaborado pela autora

## Figura C9 – Resultado da Avaliação do Caça Bugs - Página 8

Para você, qual foi a maior utilidade da aplicação?

29 respostas

Bom foi o auxílio que obtive na hora de realizar as questões.

Um abordagem diferente das aulas de programação e uma visão mais criativa sobre os códigos.

exercício de atenção e capacidade de raciocínio

aprimorar a habilidade de leitura de código

Testar meu conhecimento em reconhecer se estou compreendendo os códigos durante aprendizado no curso.

Auxilia a conseguir entender erros lógicos matemáticos de um código não familiar.

Aprendizado

Observação de programas e detectar possíveis erros a partir de uma leitura sem uso de compiladores

Ela chama a atenção pra melhorarmos e não cairmos em erros bem comuns

Mostrar como corrigir código alheio é complicado

Melhorar o aprendizado do aluno

Ajudar a prestar maior atenção aos códigos escritos para identificar possíveis erros

atenção aos códigos

Entender alguns bug's.

Facilitar a capacidade de compreensão dos assuntos

estimulou o senso de detecção de bugs

treinar a mente, otimizando o tempo, para ver o bug de tal código

Ajuda a desenvolver raciocínio lógico

poder treinar algo em que tenho dificuldade

Ganhar experiência

Auxiliar no ensino da programação, utilizando a identificação de erros

Auxiliar na identificação de problemas nos códigos

achei muito interessante a questão de Strings, que ainda é um assunto que possuo dificuldade

Agilidade pra compreender a logica dos códigos pra identificar os erros mais rápidos!

refinar minhas habilidades analíticas

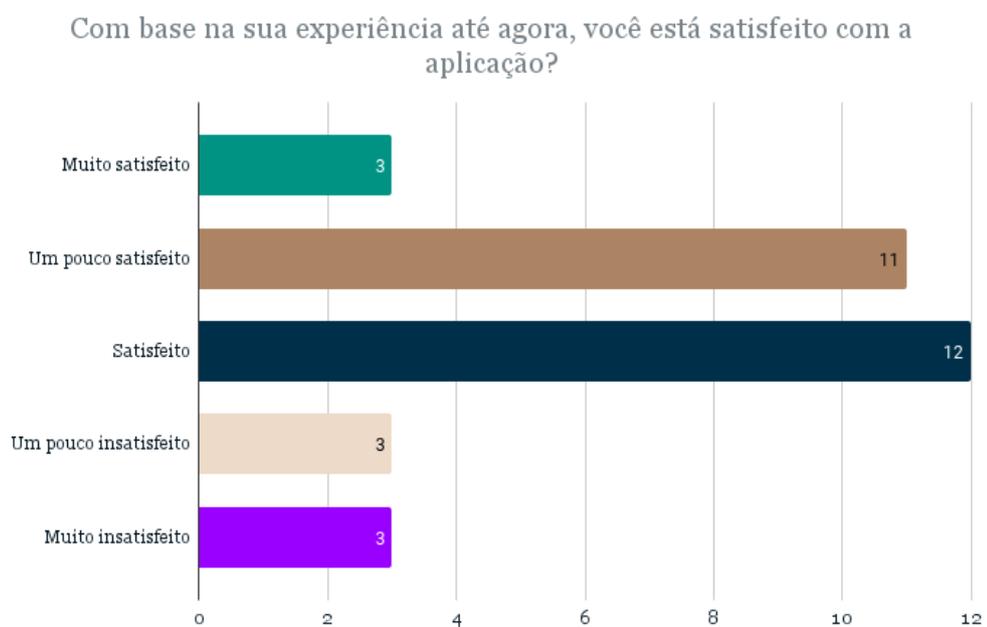
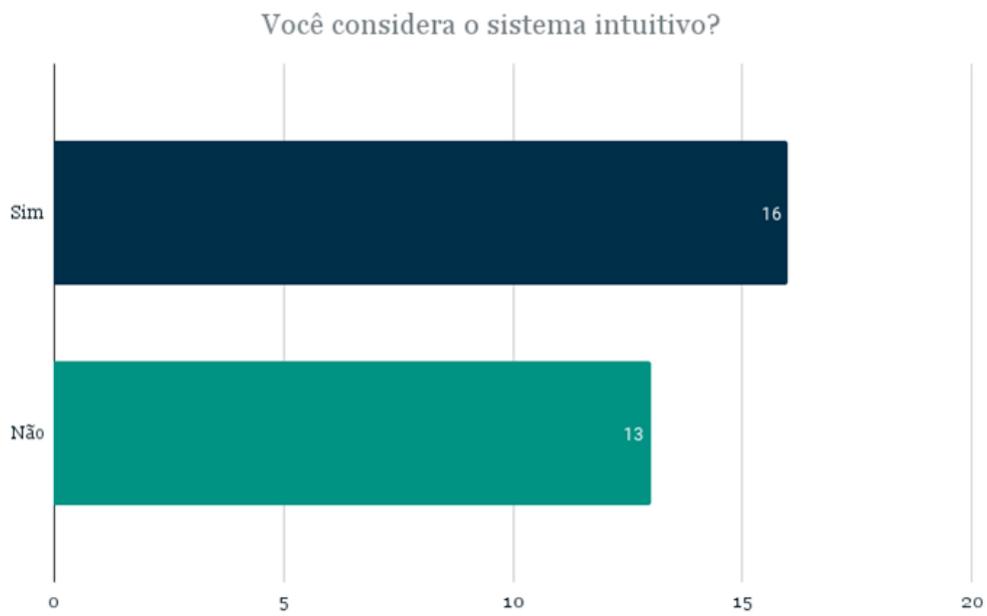
É muito boa para treinar a parte de ler e interpretar o código

A paciência e a calma para analisar o código e tentar achar o erro, pois no curso de ciência da computação saber analisar bugs é essencial.

Fazer a pessoa que está resolvendo os bugs, a manter a atenção e foco.

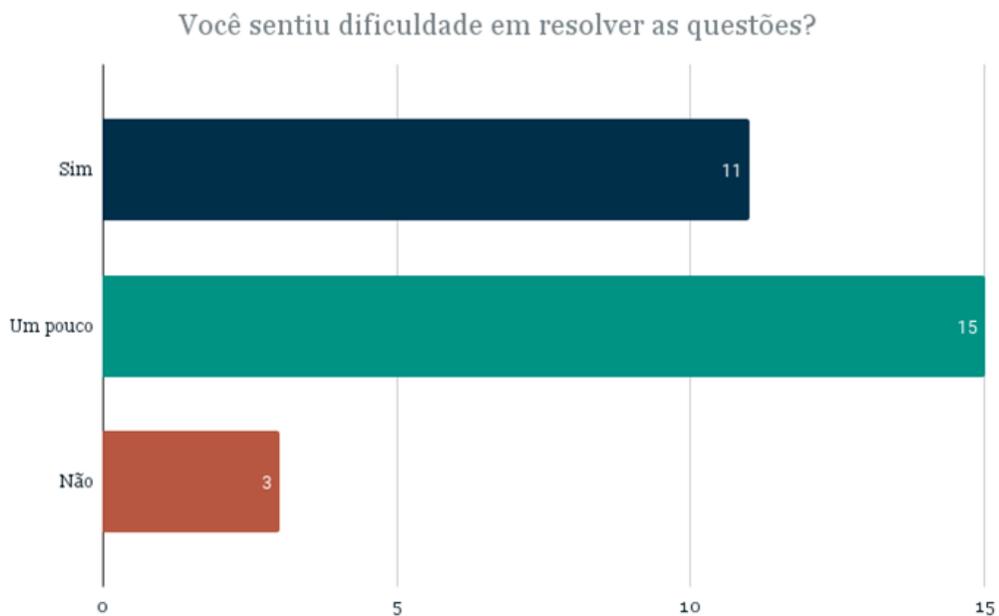
Testar minha capacidade de raciocínio para ver em quais condições o código não seria executado.

Figura C10 – Resultado da Avaliação do Caça Bugs - Página 9

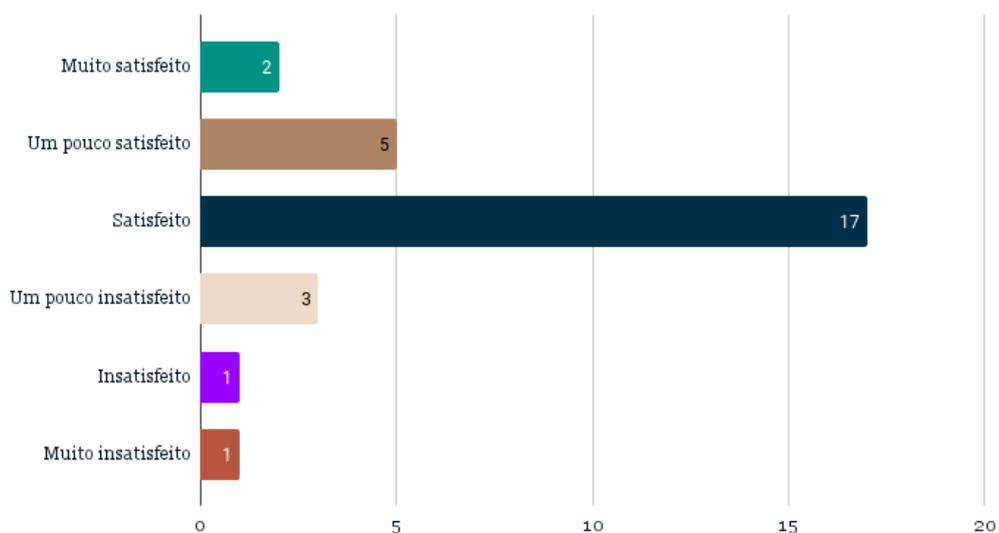


Fonte: Elaborado pela autora

Figura C11 – Resultado da Avaliação do Caça Bugs - Página 10



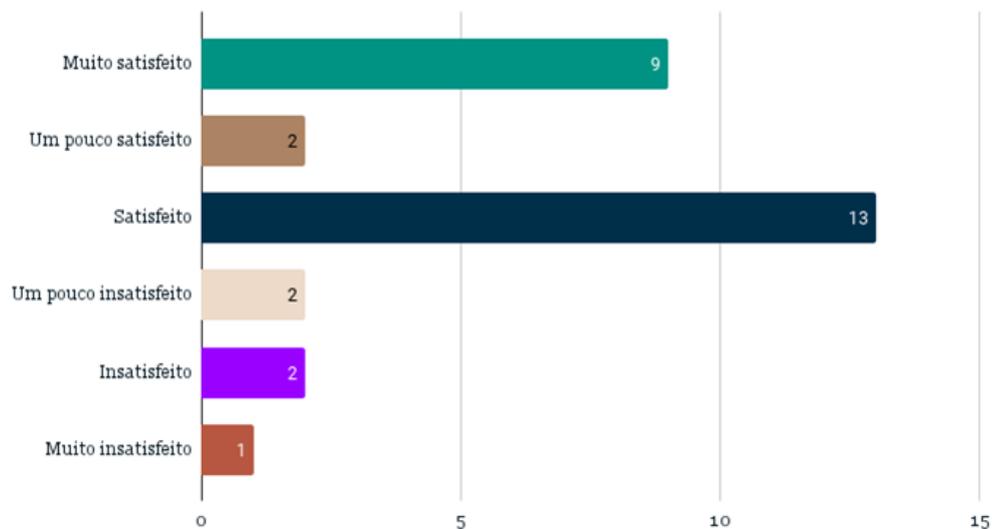
Sobre a disposição das questões, você se sente satisfeito com a ordem em que as questões foram apresentadas?



Fonte: Elaborado pela autora

Figura C12 – Resultado da Avaliação do Caça Bugs - Página 11

Sobre o nível na abordagem das questões, do simples ao mais complexo, como você se sente em relação a isso?

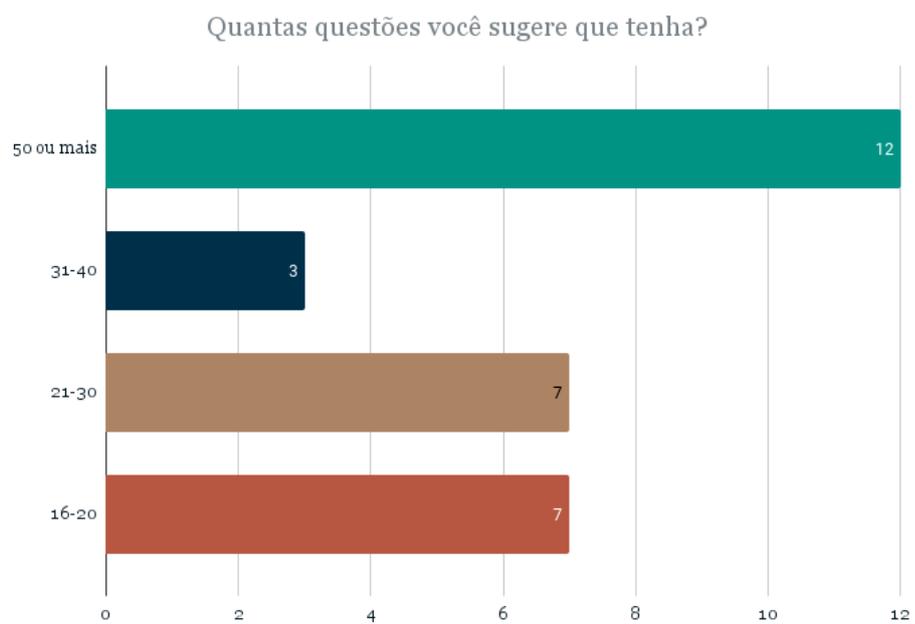


Você achou as questões repetitivas?



Fonte: Elaborado pela autora

Figura C13 – Resultado da Avaliação do Caça Bugs - Página 12



Fonte: Elaborado pela autora

# Referências

CODEACADEMY. *CodeAcademy Learn*. 2021. Disponível em: <<https://www.codecademy.com/learn>>. Acesso em: 20 de Abril de 2021. Citado 2 vezes nas páginas 23 e 24.

DEVMEDIA. *Modelagem de Software, porque, como e o que deve ser feito*. 2015. Disponível em: <<https://www.devmedia.com.br/modelagem-de-software-porque-como-e-o-que-deve-ser-feito/33794>>. Acesso em: 28 mar. 2021. Citado na página 29.

ECMAScript E.; ASSOCIATION, E. C. M. *Language Specification*. 2020. Disponível em: <<https://www.ecma-international.org/wp-content/uploads/ECMA-262.pdf>>. Acesso em: 27 de Março de 2021. Citado na página 28.

GUEDES, G. T. A. *UML 2: Uma abordagem prática*. [S.l.]: Novatec Editora. [S.l.], 2018. 488 p. Citado na página 29.

HEUSER, C. A. *Projeto de Banco de Dados. Edição 6*. [S.l.]: Grupo A. Selo: Bookman, 2008. 282 p. Citado na página 32.

HINTERHOLZ, O. J. Tepequém: uma nova ferramenta para o ensino de algoritmos nos cursos superiores em computação. p. 21, 07 2009. Citado na página 16.

JÚNIOR, D.; NETO, C.; RAPOSO, A.; NETO, L. Cosmo: Um ambiente virtual de aprendizado com foco no ensino de algoritmos. In: . [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 19 e 20.

LUZ, B.; SANTOS, R.; GUIMARAES, M. Objetos de aprendizagem interativos. *Sinergia*, v. 16, p. 131, 06 2015. Citado na página 18.

MENDES, P.; FREITAS, P. Almeida de; SOUSA, K.; NETO, C. S. Avaliação da aplicação de jogos no ensino de programação: Uma experiência em uma disciplina introdutória. p. 515, 10 2018. Citado 3 vezes nas páginas 25, 26 e 27.

SELIVON, M.; BEZERRA, J.; TONIN, N. Uri online judge academic: Integração e consolidação da ferramenta no processo de ensino/aprendizagem. In: *Anais do XXIII Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2015. p. 188–195. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/10235>>. Citado 2 vezes nas páginas 21 e 22.

SOMMERVILLE, I. *Engenharia de Software*. [S.l.]: Pearson Prentice Hall, 2018. 544 p. Citado 2 vezes nas páginas 29 e 31.

W3C. *HTML5 Specification*. 2014. Disponível em: <<http://www.w3.org/TR/2014/REC-html5-20141028/>>. Acesso em: 27 de Março de 2021. Citado na página 28.

W3C. *Introduction to CSS3*. 2018. Disponível em: <<https://www.w3.org/TR/selectors-3/>>. Acesso em: 27 de Março de 2021. Citado na página 28.

---

WILEY, D. A. “*Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*”. 2000. Disponível em: <<http://reusability.org/read/chapters/wiley.doc>>. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects: Online Version*. Retrieved MONTH DAY, YEAR, from the World Wide. Acesso em: 18 de Abril de 2021. Citado na página 18.