

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

VIVYANN FERNANDES CEDRAZ

Orientador: Prof. Dr. Reinaldo Silva Fortes

Coorientador: Prof. Dr. Saul Emanuel Delabrida Silva

**UMA AVALIAÇÃO DE USABILIDADE DO CORRETOR DE
EXERCÍCIOS DE INTRODUÇÃO À PROGRAMAÇÃO OPCODERS
JUDGE.**

Ouro Preto, MG
2023

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

VIVYANN FERNANDES CEDRAZ

**UMA AVALIAÇÃO DE USABILIDADE DO CORRETOR DE EXERCÍCIOS DE
INTRODUÇÃO À PROGRAMAÇÃO OPCODERS JUDGE.**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Reinaldo Silva Fortes

Coorientador: Prof. Dr. Saul Emanuel Delabrida Silva

Ouro Preto, MG
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C389a Cedraz, Vivyann Fernandes.
Uma avaliação de usabilidade do corretor de exercícios de introdução
à programação opCoders Judge. [manuscrito] / Vivyann Fernandes
Cedraz. - 2023.
63 f.: il.: color., tab..

Orientador: Prof. Dr. Reinaldo Silva Fortes.
Coorientador: Prof. Dr. Saul Emanuel Delabrida Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da
Computação .

1. Usabilidade de software. 2. Programação (Computadores). 3.
Interação humano-máquina. I. Fortes, Reinaldo Silva. II. Silva, Saul
Emanuel Delabrida. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 004

Bibliotecário(a) Responsável: Elton Ferreira de Mattos - CRB6/2824



FOLHA DE APROVAÇÃO

Vivyann Fernandes Cedraz

Uma avaliação de usabilidade do corretor de exercícios de introdução à programação opCoders Judge

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Março de 2023.

Membros da banca

Reinaldo Silva Fortes (Orientador) - Doutor - Universidade Federal de Ouro Preto
Saul Emanuel Delabrida Silva (Coorientador) - Doutor - Universidade Federal de Ouro Preto
Valéria de Carvalho Santos (Examinadora) - Doutora - Universidade Federal de Ouro Preto
Marcelo Luiz Silva (Examinador) - Mestre - Universidade Federal de Ouro Preto

Reinaldo Silva Fortes, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/03/2023.



Documento assinado eletronicamente por **Reinaldo Silva Fortes, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/03/2023, às 10:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0493052** e o código CRC **3D467246**.

Dedico esta monografia à minha mãe, principal responsável por eu ter chegado até aqui.

Agradecimentos

Agradeço, primeiramente, a Deus por me dar sabedoria para finalizar esse ciclo tão importante. Aos meus pais Vinicius e Eliene por todo amor e dedicação. Em especial à minha mãe, por ter acreditado em mim e me dado forças para continuar até o fim. Ao meu avô Joel (*in memoriam*) por ser o meu exemplo de ser humano. Às minhas avós Dilcinha e Neilte pelas orações e cuidado durante toda a minha vida. Ao meu tio Joel pelas orientações e incentivo. Aos meus irmãos Vinicius e Lunna pelo carinho. E à toda minha família por ser o meu porto seguro.

Aos meus amigos, em especial Débora e Jean por todo apoio desde o primeiro dia que ingressei na Universidade. À República Aconchego, por ser o meu lar em Ouro Preto e por todos os momentos inesquecíveis vividos ao longo desses anos ao lado das minhas irmãs. À Fernanda por me ajudar na aplicação dos questionários. Ao meu namorado Guilherme, pelo incentivo e companheirismo no decorrer dessa caminhada.

Aos professores Reinaldo e Saul por terem apoiado e colaborado para construção desse trabalho. À Universidade Federal de Ouro Preto e à todos os professores do DECOM pelo ensino de qualidade.

“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz.”

(Bill Gates, s.d.)

Resumo

Com o crescente avanço tecnológico, ferramentas online com foco educativo se tornaram aliadas no processo de ensino-aprendizagem de diversas disciplinas ao redor do mundo todo. Atualmente, para suporte ao ensino de programação introdutória na UFOP é utilizado o *opCoders Judge*, um sistema de correção automática de código-fonte. No entanto, é de fundamental importância entender se tal ferramenta oferece aos usuários uma interface fácil de usar e que possibilite que os mesmos concluam as suas tarefas de forma satisfatória e eficiente. Para mensurar a qualidade da interface do sistema é preciso avaliá-la a partir dos critérios de usabilidade, que tem como objetivo garantir a facilidade de uso, eficiência, efetividade e satisfação dos usuários em relação a um determinado produto. Para esse tipo de avaliação existem diversos métodos e regras para qualificar uma interface. Neste contexto, o presente trabalho tem como objetivo avaliar a usabilidade da interface do *opCoders Judge*. Para esta análise foram utilizados dois métodos conhecidos na literatura de Interação Humano-Computador, a avaliação por inspeção, por meio das 10 Heurísticas de Nielsen (NIELSEN, 1994c), e a avaliação por investigação, a partir da aplicação dos questionários *System Usability Scale* (SUS) e *Self-Assessment Manikin* (SAM). Por se tratar de uma pesquisa com seres humanos, o projeto referente a este trabalho foi enviado ao Comitê de Ética em Pesquisa da UFOP (CEP-UFOP) e foi aprovado. Com base na avaliação por inspeção, foi possível identificar problemas graves de usabilidade na interface da ferramenta. Dessa forma, optou-se por realizar as correções dos problemas encontrados através da construção de um protótipo de alta fidelidade e, a partir dele, realizar os testes com os usuários. Com o objetivo de avaliar se a interface proposta permitiria uma boa experiência de uso. Os resultados mostraram que o protótipo proposto apresenta uma boa usabilidade, obtendo 86,02 pontos no SUS. Resultado que, segundo BROOKE et al. (1996) classifica uma interface como excelente. Além disso, o resultado do questionário SAM também apresentou uma avaliação positiva nas três dimensões avaliadas (satisfação, motivação e sentimento de controle). Assim, com base nessas avaliações foram propostas alterações a serem feitas na interface do *opCoders Judge*, com o objetivo de garantir a boa usabilidade da interface e contribuir ainda mais no processo de ensino-aprendizagem de programação.

Palavras-chave: Usabilidade. Corretor de código-fonte. Programação. Interação Humano-Computador.

Abstract

With the growing technological advance, online tools with an educational focus have become allies in the teaching-learning process of various disciplines around the world. Currently, to support introductory computer programming teaching at UFOP an automatic source code correction system is used, the *opCoders Judge*. However, it is of fundamental importance to understand whether such a tool offers users an easy-to-use interface that allows them to complete their tasks satisfactorily and efficiently. To measure the quality of the system's interface, it is necessary to evaluate it based on usability criteria, which aim to ensure ease of use, efficiency, effectiveness and user satisfaction in relation to a given product. For this type of evaluation, there are several methods and rules to qualify an interface. In this context, the present work aims to evaluate the usability of the *opCoders Judge* interface. For this analysis, two methods known in the Human-Computer Interaction literature were used: evaluation by inspection, using Nielsen's 10 Heuristics (NIELSEN, 1994c), and evaluation by investigation, based on the application of questionnaires *System Usability Scale* (SUS) and *Self-Assessment Manikin* (SAM). As it is a research with human beings, the project referring to this work was sent to the Research Ethics Committee of UFOP (CEP-UFOP) and was approved. Based on the evaluation by inspection, it was possible to identify serious usability problems in the tool's interface. In this way, it was decided to correct the problems found by building a high-fidelity prototype and, based on it, carrying out tests with users. In order to assess whether the proposed interface would allow a good user experience. The results showed that the proposed prototype has good usability, obtaining 86.02 points in the SUS. Result that, according to BROOKE et al. (1996) classifies an interface as excellent. In addition, the result of the SAM questionnaire also showed a positive evaluation in the three evaluated dimensions (satisfaction, motivation and feeling of control). Thus, based on these evaluations, changes were proposed to be made in the *opCoders Judge* interface, with the objective of guaranteeing good usability of the interface and contributing even more to the programming teaching-learning process.

Keywords: Usability. Source code corrector. Programming. Human-Computer Interaction.

Lista de Ilustrações

Figura 2.1 – Interface opCoders Judge.	9
Figura 2.2 – Estrutura de Usabilidade.	10
Figura 2.3 – Escala SAM - Satisfação, motivação e sentimento de domínio.	14
Figura 3.1 – Interface da tela inicial do <i>opCoders Judge</i>	17
Figura 3.2 – Interface com dicas e informações sobre o conteúdo a ser abordado nas questões.	18
Figura 3.3 – Interface para acesso às questões.	18
Figura 3.4 – Interface com enunciado das questões.	19
Figura 3.5 – Interface para submissão e correção das questões.	19
Figura 3.6 – Interface da tela de correção para uma tarefa com erros.	20
Figura 3.7 – Interface da tela de correção para uma tarefa correta.	20
Figura 3.8 – Fluxo de desenvolvimento do trabalho.	23
Figura 4.1 – Tela de login.	26
Figura 4.2 – Tela de primeiro uso.	27
Figura 4.3 – Tela de tarefas abertas.	28
Figura 4.4 – Tela de tarefas concluídas.	28
Figura 4.5 – Tela de tarefas perdidas.	29
Figura 4.6 – Tela para envio da tarefa.	30
Figura 4.7 – Tela para seleção da linguagem de programação.	30
Figura 4.8 – Tela para seleção do arquivo.	31
Figura 4.9 – Tela de aviso que o arquivo foi enviado para avaliação.	31
Figura 4.10–Tela de erro para aviso de extensão diferente da linguagem.	32
Figura 4.11–Tela de erro para selecionar o arquivo desejado.	32
Figura 4.12–Tela da tarefa após envio.	33
Figura 4.13–Tela com mensagem de confirmação do download da tarefa.	34
Figura 4.14–Tela para visualização da correção da tarefa.	34
Figura 4.15–Tela para visualização da correção da tarefa por teste.	35
Figura 4.16–Tela para visualização do detalhamento da correção da tarefa.	35
Figura 4.17–Tela de perfil.	36
Figura 4.18–Tela de aviso que a senha foi alterada.	36
Figura 4.19–Tela de dicas.	37
Figura 4.20–Tela com detalhamento das dicas.	37
Figura 4.21–Tela de dicas dentro das tarefas.	38
Figura 4.22–Tela de ajuda.	38
Figura 4.23–Repetência na disciplina.	39
Figura 4.24–Período dos participantes.	39
Figura 4.25–% de alunos que já utilizaram algum corretor automático.	40

Figura 4.26–% de alunos que já programavam antes da disciplina.	40
Figura 4.27–Resultado da avaliação de usabilidade a partir do SUS.	42
Figura 4.28–Média das respostas para as afirmações positivas.	43
Figura 4.29–Média das respostas para as afirmações negativas.	43
Figura 4.30–Variação de respostas na dimensão satisfação.	44
Figura 4.31–Variação de respostas na dimensão motivação.	45
Figura 4.32–Variação de respostas na dimensão sentimento de controle.	45
Figura 4.33–Tempo de execução da tarefa por participante.	46

Lista de Tabelas

Tabela 2.1 – Tabela comparativa dos trabalhos relacionados e este trabalho quanto aos métodos de avaliação adotados.	8
Tabela 2.2 – Afirmções questionário SUS.	13
Tabela 3.1 – Métodos que respondem as hipóteses de pesquisa apresentadas na Subseção 1.3.2.	21
Tabela 4.1 – Conjunto de ações executadas no <i>opCoders Judge</i>	25
Tabela 4.2 – Total de problemas encontrado nas ferramentas por escala de classificação. .	26
Tabela 4.3 – Opinião dos participantes sobre a interface do <i>opCoders Judge</i>	41

Lista de Abreviaturas e Siglas

AVA	Ambiente Virtual de Aprendizagem
BTI	Bacharelado em Tecnologia da Informação
CEP	Comitê de Ética em Pesquisa
CNS	Conselho Nacional de Saúde
CONEP	Comissão Nacional de Ética em Pesquisa
DECOM	Departamento de Computação
IHC	Interação Humano-Computador
MOODLE	<i>Modular Object-Oriented Dynamic Learning Environment</i>
SAM	<i>Self-Assessment Manikin</i>
SUMI	Software Usability Measurement Inventory
SUS	System Usability Scale
TCLE	Termo de Consentimento Livre e Esclarecido
UnB	Universidade de Brasília
UFERSA	Universidade Federal Rural do Semi-Árido
UFOP	Universidade Federal de Ouro Preto
UFSC	Universidade Federal de Santa Catarina
USP	Universidade de São Paulo
VPL	Laboratório Virtual de Programação

Sumário

1	Introdução	1
1.1	Justificativa	3
1.2	Objetivo Geral	3
1.2.1	Objetivos Específicos	3
1.3	Problema e Hipóteses de Pesquisa	3
1.3.1	Problema	4
1.3.2	Hipóteses de Pesquisa	4
1.3.3	Delimitação do Estudo	4
1.4	Organização do Trabalho	4
2	Revisão Bibliográfica	6
2.1	Trabalhos Relacionados	6
2.2	Fundamentação Teórica	8
2.2.1	Corretor automático de código-fonte: opCoders Judge	8
2.2.2	Usabilidade	9
2.2.2.1	Avaliação por Inspeção	10
2.2.2.2	Heurísticas de Nielsen	11
2.2.2.3	Avaliação por Investigação	12
2.2.2.3.1	System Usability Scale (SUS)	13
2.2.2.3.2	Self-Assessment Manikin (SAM)	14
2.2.3	Prototipação	15
3	Desenvolvimento	17
3.1	opCoders Judge	17
3.2	Definição dos Métodos de Avaliação	20
3.3	Descrição do Método	22
3.3.1	Organização dos Testes	22
3.4	Submissão ao Comitê de Ética	24
4	Resultados	25
4.1	Avaliação por Inspeção (Heurísticas de Nielsen)	25
4.2	Protótipo Desenvolvido	26
4.3	Questionário de perfil	39
4.4	Avaliação por Investigação (SUS)	42
4.5	Avaliação por Investigação (SAM)	44
4.6	Tempo de execução da tarefa	46
4.7	Discussão	46
5	Considerações Finais	48
5.1	Conclusão	48

5.2	Trabalhos Futuros	48
	Referências	50
	Apêndices	53
APÊNDICE A	Termo de Consentimento Livre e Esclarecido (TCLE)	54
APÊNDICE B	Problemas encontrados na interface a partir da avaliação por inspeção	58
APÊNDICE C	Questionário de Perfil	62

1 Introdução

Diante de tantos avanços tecnológicos nas últimas décadas, é inevitável a presença da tecnologia nas mais diversas organizações e segmentos. E na educação não tem sido diferente, tanto com o aumento do número de matrículas em cursos superiores de EaD, quanto o aumento expressivo da utilização de ferramentas e sistemas para suporte ao ensino presencial. Em decorrência da pandemia causada pela Covid-19, tais ferramentas online passaram a ser amplamente utilizadas por alunos de diversos níveis escolares e de conhecimentos tecnológicos, que persistem mesmo após o retorno das aulas presenciais. Dessa forma, uma ferramenta educacional online deve oferecer aos seus usuários uma interface eficiente e que seja capaz de conduzi-los ao seu objetivo final. Segundo [PRESSMAN \(1995\)](#), se um programa não for *user-friendly*, ou seja, não tenha uma usabilidade amigável, frequentemente estará destinado ao fracasso, por mais que o programa execute ótimas funções.

São inúmeras as ferramentas disponíveis na *web* com essa finalidade. No cenário dos cursos superiores de Tecnologia da Informação (TI) a utilização de ferramentas educacionais online são ainda mais comuns. Exemplo disso são os sistemas de correção automática de código-fonte, também conhecidos como juízes online. Muito utilizados para apoio ao ensino de programação de computadores. Um exemplo desse modelo de ferramenta é o *Beecrowd*¹, um portal de acesso gratuito que disponibiliza desafios de programação e corrige as respostas submetidas instantaneamente. Os algoritmos enviados para submissão podem estar escritos em diversas linguagens de programação, como por exemplo: C, C++, Java, php, Python, Ruby, entre outras.

Nesse contexto também temos o *Run.codes*², uma ferramenta de correção automática de código-fonte desenvolvida por alunos da Universidade de São Paulo (USP) capaz de corrigir exercícios de programação em segundos. Atualmente o sistema é disponibilizado para universidades públicas de forma gratuita e nela é possível acompanhar o desempenho de cada aluno por meio de gráficos.

Estudos realizados em diversas universidades apontam a disciplina de programação como uma das que os alunos mais reprovam. De acordo com [BOSSE; GEROSA \(2015\)](#), na USP entre 2010 e 2014 as disciplinas de introdução à programação registraram um índice 30% de reprovação, chegando até 50% em alguns casos. Segundo [SANTOS; COSTA \(2006\)](#), os alunos sentem muita dificuldade em entender os conceitos passados em sala de aula, o que acarreta em reprovações, trancamento da disciplina e até desistência do curso.

Em seu trabalho, [QUEIROZ \(2018\)](#) ressalta a importância do uso da tecnologia na sala de aula integrada ao currículo como ferramenta pedagógica para inclusão de pessoas com

¹ <https://www.beecrowd.com.br/>

² <https://run.codes/>

dificuldade de aprendizagem. Dentro desse contexto, por se tratar de uma matéria que os estudantes apresentam alto grau de dificuldade, um professor da disciplina de Programação de Computadores I na UFOP coordenou o desenvolvimento do *opCoders Judge*, um corretor automático de código-fonte que é utilizado para o ensino de programação introdutória na UFOP para submissão dos exercícios práticos, com o objetivo de facilitar o processo de aprendizagem dos alunos.

De acordo com BRITO; FORTES (2019), a introdução de corretores automáticos para o ensino de programação colaboram para o aprendizado dos alunos, ajudam no desenvolvimento lógico e os mantêm motivados para realização das atividades. Diferentemente dos juízes online disponíveis na *web* para correção de código-fonte, o *opCoders Judge* avalia o percentual de semelhança entre a resposta submetida pelo aluno e a resposta correta para o exercício. Possibilitando ao aluno a consolidação do conhecimento através de um *feedback* apresentando os seus erros e acertos, evidenciando o potencial pedagógico da ferramenta. Além disso, o sistema também avalia a saída obtida não apenas em relação à formatação, como ocorre nos demais corretores. No *opCoders Judge* cada arquivo de saída é submetido a vários procedimentos que avaliam se a lógica da solução está correta (BRITO; FORTES, 2019).

No entanto, o *opCoders Judge* é utilizado há algum tempo para suporte ao ensino da disciplina na UFOP, mas nunca foi realizado um estudo sistemático para avaliar a sua usabilidade. Nesse contexto, o presente trabalho tem como objetivo avaliar a usabilidade da interface do *opCoders Judge*. Para esta avaliação foram utilizadas as 10 Heurísticas de Nielsen e aplicados os questionários *System Usability Scale* (SUS) e *Self-Assessment Manikin* (SAM).

A partir da avaliação por inspeção através das heurísticas de Nielsen, foi possível identificar alguns problemas graves de usabilidade na interface do corretor. Diante desse cenário, optou-se pela elaboração de um protótipo com as propostas de solução para os problemas encontrados e, posteriormente, o mesmo foi utilizado para avaliar se as correções de fato possibilitariam que a ferramenta tivesse uma boa usabilidade. Para que as avaliações pudessem ser aplicadas com os participantes da pesquisa, foi elaborado e submetido ao Comitê de Ética em Pesquisa da UFOP (CEP - UFOP) o projeto referente a este trabalho para ser avaliado. O projeto foi aprovado e as avaliações contaram com a participação de 30 alunos matriculados em uma disciplina introdutória de programação. Os resultados das avaliações apontaram uma usabilidade excelente da interface prototipada.

As seções a seguir estão organizadas da seguinte forma. Na Seção 1.1 é apresentada a justificativa para realização deste trabalho. Na Seção 1.2 são esclarecidos os objetivos gerais e específicos. Na Seção 1.3 é apresentado o problema e as hipóteses de pesquisa. Por último, na Seção 1.4 é apresentada a organização deste trabalho.

1.1 Justificativa

Um fator importante que apontou a necessidade desse estudo de usabilidade é que, embora o corretor automático de código-fonte, *opCoders Judge* já esteja sendo utilizado há algum tempo, ainda não existe nenhuma avaliação de usabilidade detalhada sobre a interface da respectiva ferramenta. Apesar da utilização da mesma ser positiva, nem sempre produz o resultado esperado e pode acabar atrapalhando o aluno, caso ele não entenda perfeitamente a ferramenta e não saiba utilizá-la corretamente.

Para que os alunos tenham uma experiência satisfatória ao utilizar o sistema, é necessário que o mesmo tenha uma interface amigável e esteja de acordo com os critérios de usabilidade de uma interface web. De acordo com [NIELSEN \(1994b\)](#), usabilidade é um atributo que avalia o quão fácil a interface é de usar. Nesse sentido, como nunca foi feita uma avaliação de usabilidade da ferramenta utilizada, torna-se imprescindível a realização deste trabalho para avaliar a qualidade da interface da ferramenta.

1.2 Objetivo Geral

O objetivo geral deste trabalho é avaliar a usabilidade da interface da ferramenta de correção automática de código-fonte, *opCoders Judge*, utilizada como suporte no processo de ensino-aprendizado de programação introdutória na Universidade Federal de Ouro Preto. Bem como, propor melhorias na interface de forma que seja possível obter a boa experiência do usuário.

1.2.1 Objetivos Específicos

Para possibilitar o alcance do objetivo geral, os objetivos específicos estabelecidos para este trabalho são:

- Avaliar a interface do Corretor automático a partir do método de inspeção, analisando se há falhas de usabilidade;
- Realizar testes de usabilidade com os participantes da pesquisa e identificar as principais dificuldades dos usuários;
- Analisar os resultados obtidos e propor melhorias na usabilidade da ferramenta de acordo com os resultados dos testes.

1.3 Problema e Hipóteses de Pesquisa

Para a construção deste trabalho foi definido um problema de pesquisa, as hipóteses para esse problema e realizada a escolha dos métodos de avaliação que possibilitassem o alcance

do objetivo proposto. Na Subseção 1.3.1 é apresentado o problema o qual este trabalho estuda. Na Subseção 1.3.2 são apresentadas as hipóteses de pesquisa. E por fim, na Subseção 1.3.3 é apresentada a delimitação de escopo para realização desse trabalho.

1.3.1 Problema

O uso das ferramentas online para apoio ao aprendizado é importante e necessário. No entanto, podem resultar em uma experiência ruim ao usuário caso não tenham uma interface que atenda aos critérios de usabilidade. A interface do corretor automático de código-fonte, *opCoders Judge*, é de fácil aprendizagem, mas pode apresentar falhas de usabilidade que impedem que os usuários concluam suas tarefas de forma satisfatória.

1.3.2 Hipóteses de Pesquisa

Para o problema descrito acima as hipóteses de pesquisa são:

- **Hipótese I:** a interface da ferramenta é de fácil aprendizagem;
- **Hipótese II:** os usuários conseguem concluir as tarefas de forma satisfatória;
- **Hipótese III:** a interface da ferramenta apresenta falhas de usabilidade.

1.3.3 Delimitação do Estudo

Este trabalho está inserido na Linha de Pesquisa em Ergonomia e Usabilidade, que faz parte da área de Interação Humano-Computador, um campo de estudo dentro da Ciência da Computação.

A partir da identificação do problema, foram levantadas hipóteses de pesquisa que poderiam apontar um caminho para solucioná-lo. Desse modo, foi realizado um estudo para identificar quais metodologias de avaliação de usabilidade existentes na literatura poderiam responder as hipóteses desta pesquisa e possibilitar identificar a solução para o problema tratado.

A investigação do problema foi realizada entre alunos matriculados em uma disciplina introdutória de programação de computadores na UFOP, através da aplicação de técnicas de avaliação de usabilidade, visando avaliar se a ferramenta utilizada para o ensino de programação de computadores possui problemas de usabilidade.

1.4 Organização do Trabalho

O presente trabalho está organizado da seguinte forma:

Capítulo 1: introduz a respeito do problema, os objetivos do trabalho e as hipóteses de pesquisa;

Capítulo 2: traz um resumo dos trabalhos relacionados existentes na literatura e o referencial teórico;

Capítulo 3: apresenta a metodologia proposta para o trabalho e os métodos utilizados para avaliação;

Capítulo 4: apresenta as conclusões, propostas de melhorias e trabalhos futuros.

2 Revisão Bibliográfica

Esta seção é dedicada à apresentação dos conceitos fundamentais para a construção deste trabalho, estando organizado da seguinte forma: na Seção 2.1 são apresentados os trabalhos relacionados. Na Seção 2.2 são apresentados os conceitos referentes a usabilidade, corretor automático e os métodos de avaliação utilizados.

2.1 Trabalhos Relacionados

Na literatura pouco se encontra sobre a avaliação de corretores online ou de ferramentas utilizadas no ensino de programação. De forma pontual, ainda não existe nenhum trabalho que avalie a usabilidade da interface do *opCoders Judge*.

FREITAS et al. (2016) analisou a usabilidade do módulo Laboratório Virtual de Programação do Moodle utilizado na Universidade Federal de Santa Catarina. A análise foi feita com 37 alunos de duas turmas das disciplinas de Programação (Programação de Computadores I e Programação de Computadores II) do curso de Bacharelado em Tecnologias de Informação e Comunicação da UFSC, Campus Araranguá. Para a análise, a autora realizou a avaliação heurística seguindo os critérios ergonômicos de BASTIEN; SCAPIN (1993). A usabilidade foi verificada por meio da observação e aplicação do questionário de satisfação. A observação dos usuários considerou três pontos: experiência do usuário por meio do Cubo do Utilizador (NIELSEN, 1994b); perspectiva sobre eficiência, eficácia e satisfação; facilidade de aprender e usar a interface. O questionário de satisfação SUMI foi aplicado, havendo avaliação do grau de confiabilidade das respostas através do coeficiente alfa de CRONBACH (1951). Os resultados mostraram que a interface do Módulo VPL possui problemas de usabilidade. A partir da avaliação heurística foi possível identificar problemas na condução, carga de trabalho, controle explícito, adaptabilidade e gestão de erros. Durante o teste de usabilidade todos os usuários manifestaram insatisfação com a interface. Alguns deles sentiram dificuldade para usar a interface, sendo necessário repetir a mesma tarefa várias vezes e fazer perguntas sobre o funcionamento de algumas funcionalidades.

SCHNEIDER (2017) realizou uma avaliação de usabilidade no Moodle Mobile entre 16 alunos de um Curso Técnico em Informática na cidade de Novo Hamburgo/RS. Para a avaliação foi aplicado o SUS por ser, segundo ele, um questionário simples e de rápida aplicação, que consiste em avaliar a satisfação do usuário em relação a determinado sistema. Para o critério “Facilidade de Aprendizagem” o Moodle Mobile recebeu nota 92,07. Já o critério “Eficiência” a nota foi 79,69 e por fim, o critério “Satisfação do usuário” obteve nota 84,38. Na escala global do SUS a aplicação alcançou 84,69 pontos, comprovando que o Moodle Mobile responde aos critérios de usabilidade avaliados pelo SUS.

ALMEIDA (2021) analisou de usabilidade da ferramenta Aprender3, Moodle utilizado na Universidade de Brasília. Para a avaliação de usabilidade a autora realizou uma análise com base no Conjunto de Heurísticas de Usabilidade de Nielsen (NIELSEN, 1994c) e aplicou os questionários ISONORM 9241 e AttrakDiff para mensurar a satisfação do usuário com a interface. 35 alunos dos cursos de graduação da UnB responderam a pesquisa via ferramenta *Google Forms*. A autora concluiu que a ferramenta Aprender3 tem uma estética boa, mas precisa melhorar em relação a performance, segurança e disponibilidade.

SOUZA (2019) desenvolveu o protótipo de um software educacional *Loop Academic* para suporte no ensino-aprendizagem de programação introdutória no curso de Bacharelado em Tecnologia da Informação - BTI da Universidade Federal Rural do Semi-árido, Campus Pau dos Ferros. O protótipo foi concebido a partir das heurísticas de usabilidade propostas por NIELSEN (1994c) e das heurísticas reformuladas por BENYON (2011). O teste de usabilidade foi aplicado a 7 alunos da disciplina de Algoritmos e Laboratório de Algoritmos, de acordo com NIELSEN; LANDAUER (1993) este é um número aceitável de usuários. O autor avaliou o protótipo de alta fidelidade de acordo com: o nível de acertos na realização das atividades e o percentual de defeitos (VALENTIM et al., 2017); tempo de execução das tarefas; qualidade efetiva da interface com base na estratégia proposta por VALENTIM et al. (2017), com aplicação da Escala SAM, proposta por LANG (1980). Como resultado, foi possível concluir que o protótipo se adequa à necessidade dos alunos, alcançando quase todas as máximas nas avaliações de Satisfação, Controle e Motivação, a partir da Escala SAM.

Conforme apresentado nos trabalhos relacionados, é possível perceber a importância da realização da análise de usabilidade em ferramentas de suporte ao aprendizado. Visto que um sistema com falhas na interface pode trazer uma experiência ruim ao usuário, principalmente quando a sua utilização é frequente, como é o caso das ferramentas em questão.

Assim como a maioria dos trabalhos apresentados nessa seção, o presente trabalho também irá avaliar o sistema a partir de duas metodologias. Uma avaliação por meio de um método de inspeção e duas avaliações a partir de métodos de investigação. Conforme apresentado na Tabela 2.1. Esse trabalho se destaca, pois não há na literatura uma avaliação específica sobre o *opCoders Judge*. Logo, o foco deste trabalho será avaliar a interface da ferramenta mencionada e a partir do resultado da avaliação propor melhorias.

Para o Método de Inspeção será utilizada a Avaliação Heurística proposta por Nielsen (NIELSEN, 1994c). Complementar a ela, a avaliação por investigação será feita por meio do *System Usability Scale (SUS)* e *Self-Assessment Manikin (SAM)*.

Tabela 2.1 – Tabela comparativa dos trabalhos relacionados e este trabalho quanto aos métodos de avaliação adotados.

Trabalho	Método de Inspeção	Método de Investigação
(FREITAS et al., 2016)	(BASTIEN; SCAPIN, 1993)	SUMI
(SCHNEIDER, 2017)		(BROOKE et al., 1996)
(ALMEIDA, 2021)	(NIELSEN, 1994c)	ISNORM9241 e AttrakDiff
(SOUZA, 2019)	(NIELSEN, 1994c) e (BENYON, 2011)	Escala SAM
Este trabalho	(NIELSEN, 1994c)	(BROOKE et al., 1996) e Escala SAM

2.2 Fundamentação Teórica

Esta seção é dedicada à apresentação dos conceitos fundamentais para a compreensão deste trabalho. Na Subseção 2.2.1 é apresentado o conceito de corretor automático de código-fonte, bem como a apresentação do *opCoders Judge*. Na Subseção 2.2.2 é apresentado o conceito de usabilidade.

2.2.1 Corretor automático de código-fonte: *opCoders Judge*

Um corretor automático de código-fonte, também conhecido como juiz online, é uma ferramenta computacional desenvolvida com o objetivo de auxiliar no processo de aprendizagem de programação. Nesse ambiente é possível encontrar uma série de desafios separados por níveis a serem resolvidos e submetidos para correção. A validação das respostas é feita de forma instantânea dentro da plataforma, verificando a consistência do código por meio de vários casos de teste. Essa se caracteriza como uma das principais vantagens para utilização desse tipo de sistema, pois o usuário consegue receber na mesma hora um *feedback* a respeito da solução (BELTRAME; LOPES, 2018) (2018).

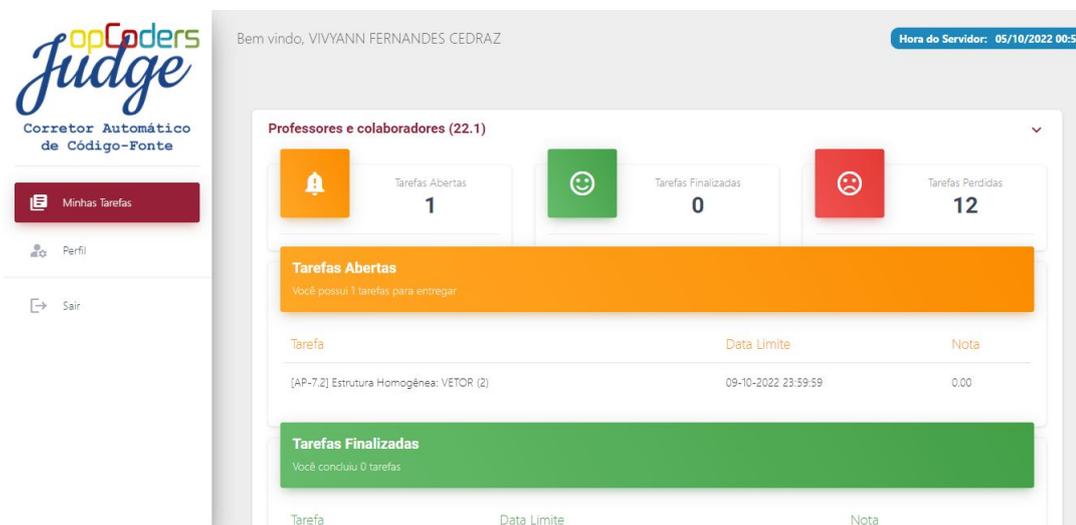
A utilização desse tipo de ferramenta para o ensino de programação tem se tornado cada vez mais comum, pois também auxilia o professor na correção de exercícios e no entendimento das dificuldades e particularidades de cada aluno. Em seu estudo, SILVA et al. (2015) realizou uma busca entre artigos acadêmicos e observou que 45% deles propõem que sejam utilizadas ferramentas de software para o ensino-aprendizagem de programação.

O corretor automático de código-fonte, *opCoders Judge*, foi desenvolvido por alunos e um professor do Departamento de Computação (DECOM)¹⁰ da UFOP e teve a sua primeira versão desenvolvida em 2019 (BRITO, 2019). Atualmente é utilizado para apoio ao ensino de programação introdutória na universidade.

¹⁰ Prof. Reinaldo Silva Fortes, orientador deste trabalho.

Desde a sua primeira versão, a plataforma vem passando por atualizações, a Figura 2.1 apresenta a interface da última versão da ferramenta.

Figura 2.1 – Interface opCoders Judge.



Fonte: (DECOM, 2022)

A metodologia proposta para o desenvolvimento do sistema visa considerar o raciocínio lógico do aluno, com o objetivo de flexibilizar o aprendizado e torná-lo mais didático. Para isso o corretor realiza dois tipos de análises para correção dos exercícios, a análise dinâmica e a análise estática. A dinâmica analisa a solução a partir das saídas geradas pelos códigos e por critérios pré-definidos. A estática avalia a solução submetida pelo aluno a partir da complexidade e do encadeamento adequado dos comandos do código (BRITO, 2019).

2.2.2 Usabilidade

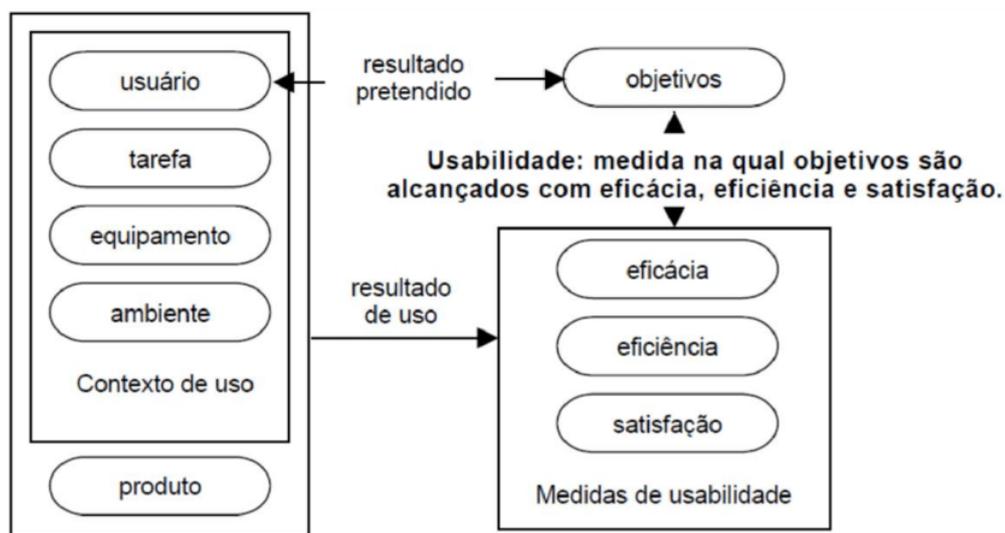
Usabilidade é um conceito dentro da Interação Humano-Computador que está relacionada à qualidade de uma interface (WINCKLER; PIMENTA, 2002). Há diversos métodos na literatura capazes de avaliar a usabilidade de uma interface. De acordo com a norma ISO 9241-11 (1998), usabilidade pode ser definida como “a medida na qual um produto pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”. A Figura 2.2 representa a estrutura que explica os componentes de usabilidade e suas relações.

Para NIELSEN (1994b) é possível avaliar a usabilidade de uma interface a partir de cinco critérios, são eles:

- **Facilidade de aprendizado:** está relacionado ao tempo e esforço que o usuário irá dedicar para aprender utilizar o sistema com competência;

- **Facilidade de memorização:** refere-se aos mecanismos oferecidos para ajudar o usuário a lembrar como utiliza o sistema;
- **Eficiência:** diz respeito ao tempo necessário para conclusão de uma atividade no sistema;
- **Segurança no uso:** avalia o grau de proteção de um sistema contra ações perigosas para o usuário;
- **Satisfação:** está relacionado a emoção e satisfação do usuário ao utilizar o sistema.

Figura 2.2 – Estrutura de Usabilidade.



Fonte: (ABNT et al., 2002)

2.2.2.1 Avaliação por Inspeção

Os métodos de avaliação por inspeção constituem um modelo de avaliação que não envolvem a participação dos usuários. A análise da interface é feita por um especialista e tem o objetivo de identificar problemas de usabilidade. Dessa forma, esse tipo de método permite que o especialista pesquise uma solução em IHC para tentar prevenir problemas que os usuários poderiam enfrentar ao interagir com o sistema (BARBOSA; SILVA, 2010).

De acordo com MATERA; RIZZO; CARUGHI (2006), a avaliação heurística é um dos principais métodos de avaliação por inspeção. Esse tipo de avaliação segue as heurísticas definidas por alguns autores e são descritas como regras gerais que revelam princípios de usabilidade que devem ser empregados na construção de uma interface (COSTA, 2019). O processo de avaliação se dá a partir da utilização do sistema pelo avaliador. Este por sua vez, irá explorar a interface em busca de problemas de usabilidade. Caso seja encontrado, é preciso indicar qual heurística foi violada, onde o problema se encontra e classificá-lo de acordo com a sua gravidade.

Para [NIELSEN \(2007\)](#) existem três fatores que tornam um problema grave para os usuários, são eles:

- **Frequência:** se o problema ocorre muitas vezes ou não, caso seja encontrado por poucos usuários ele será considerado de gravidade baixa;
- **Impacto:** quantas dificuldades são geradas aos usuários a partir da identificação do problema;
- **Persistência:** a constância com que o problema causa dificuldades aos usuários. Interfaces confusas que geram repetidas dificuldades e recebem o grau de gravidade mais alto.

([NIELSEN, 1994c](#)) classifica os problemas de usabilidade em uma escala de cinco níveis para priorização de implementação das soluções, sendo:

- **Problema cosmético:** tipo de problema cuja correção é viável e não interfere na usabilidade do sistema;
- **Problema menor:** problema no qual a correção é de baixa prioridade e o usuário consegue evitá-lo;
- **Problema médio:** a correção para esse problema é de média prioridade e pode confundir o usuário ao utilizar o sistema. No entanto, o usuário pode adaptar-se;
- **Problema grave:** a correção desse tipo de problema deve ocorrer antes do sistema ser disponibilizado aos usuários. Caso não seja possível, a interface deve oferecer aos usuários ajuda para contornar o problema. Pois ele dificulta a utilização do sistema;
- **Problema catastrófico:** a correção desse problema é fundamental e urgente, pois ele inviabiliza a conclusão da tarefa por parte do usuário.

Para este trabalho serão aplicadas as 10 heurísticas propostas por Nielsen ([NIELSEN, 1994c](#)), descritas na Subseção a seguir.

2.2.2.2 Heurísticas de Nielsen

A avaliação heurística foi proposta por Nielsen juntamente com Rolf Molich ([NIELSEN; MOLICH, 1990](#)) e dispõe de um conjunto de regras que devem ser observadas pelos avaliadores ao analisar uma interface. Essas heurísticas foram refinadas por Nielsen e um novo conjunto de heurísticas foi proposto a partir de 249 problemas de usabilidade ([NIELSEN, 1994a](#)). A partir delas, foram derivadas o conjunto das 10 heurísticas ainda mais detalhadas ([NIELSEN, 1994c](#)), descritas a seguir:

- **H1 - Visibilidade do estado do sistema:** oferecer aos usuários feedback sobre o que está acontecendo dentro de um intervalo aceitável;
- **H2 - Correspondência entre o sistema e o mundo real:** utilização de palavras e conceitos que são conhecidos pelo usuário, trazendo padrões do mundo real e apresentando as informações por meio de uma estrutura lógica e natural;
- **H3 - Controle e liberdade do usuário:** ofereça ao usuário uma opção capaz de desfazer uma ação indesejada;
- **H4 - Consistência e padrões:** mantenha padrões para palavras, situações e ações, de modo que o sistema tenha as mesmas características aos demais desse setor;
- **H5 - Prevenção de erros:** mais do que boas mensagens de erro, evite que problemas aconteça. Remova as possibilidades de erro e forneça os usuários opções de confirmação antes de executarem ações;
- **H6 - Reconhecimento em vez de memorização:** todas as informações que o usuário precisa para utilizar a interface devem estar visíveis, evitando que seja necessário memorizar informações;
- **H7 - Flexibilidade e eficiência de uso:** a utilização de atalhos ocultos podem facilitar a interação de usuários experientes, de forma que a aplicação satisfaça usuários experientes e inexperientes;
- **H8 - Projeto estético minimalista:** a interface não pode ter informações irrelevantes para não diminuir a visibilidade das informações relevantes;
- **H9 - Ajuda ao usuário a reconhecer, diagnosticar e recuperar erros:** apresente mensagens de erros fáceis de entender, indicando exatamente onde está o problema e oferecendo uma solução;
- **H10 - Ajuda e documentação:** o ideal é que o sistema não precise oferecer nenhuma explicação complementar. Mas caso seja necessário, forneça documentos que auxiliem o usuário a finalizar a sua tarefa.

Segundo Nielsen, as dez regras citadas são chamadas de heurísticas, pois são regras gerais e não normas individuais de usabilidade (NIELSEN, 2020). Caso o avaliador veja necessidade, podem ser adicionadas mais heurísticas à avaliação.

2.2.2.3 Avaliação por Investigação

Na Interação Humano-Computador existem várias técnicas para aplicar uma avaliação por investigação. Algumas delas são: questionários, estudos de campo, observação, teste com usuário,

entre outros. Nesse tipo de avaliação temos a participação dos usuários, e a partir dos resultados obtidos é possível examinar as opiniões, comportamentos, pontos de vista e expectativas do usuário em relação ao sistema.

Nas Subseções a seguir são apresentados dois modelos de avaliação por investigação, *System Usability Scale* (Subseção 2.2.2.3.1) e *Self-Assessment Manikin* (Subseção 2.2.2.3.1).

2.2.2.3.1 System Usability Scale (SUS)

System Usability Scale é um modelo de questionário proposto por BROOKE et al. (1996), que visa avaliar a usabilidade de uma interface através de uma escala numérica, avaliando três critérios: efetividade, eficiência e satisfação. O questionário contém 10 afirmações, conforme apresentado na Tabela 2.2. A resposta para cada uma das afirmações é obtida através da escala Likert¹¹, que varia entre 1 e 5. No qual 1 significa “Discordo totalmente” e 5 significa “Concordo totalmente”. Dessa forma, o usuário deve selecionar a opção que mais representa a sua opinião após utilizar o sistema.

Tabela 2.2 – Afirmações questionário SUS.

N	Afirmações
1	Eu acho que gostaria de usar esse sistema com frequência.
2	Eu acho o sistema desnecessariamente complexo.
3	Eu achei o sistema fácil de usar.
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5	Eu acho que várias funções do sistema estão muito bem integradas.
6	Eu acho que o sistema apresenta muita inconsistência.
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8	Eu achei o sistema atrapalhado de usar.
9	Eu me senti confiante ao usar o sistema.
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Para conhecer a pontuação geral da interface a partir das respostas do usuário é necessário realizar o seguinte cálculo: nas afirmações de números ímpares (1, 3, 5, 7 e 9) deve subtrair 1 da resposta obtida. Nas afirmações de números pares (2, 4, 6, 8 e 10) deve subtrair 5 da resposta obtida. Por fim, some o valor de todas as afirmações e multiplique por 2,5. Dessa forma, a pontuação final irá variar entre 0 e 100 pontos. Para SAURO (2011), a pontuação média considerada aceitável para uma interface é 68 pontos. Interfaces com resultado abaixo de 50 pontos apresentam uma má usabilidade. Segundo BROOKE et al. (1996), a pontuação obtida através do SUS corresponde à usabilidade geral do sistema. Dessa forma, as pontuações obtidas em cada afirmação não podem

¹¹ É um tipo de escala utilizada em questionários onde os perguntados respondem as afirmações de acordo com o seu nível de concordância à ela (LIKERT, 1932).

ser avaliadas de forma individual, sendo recomendado que se tenha a avaliação de mais de 20 usuários.

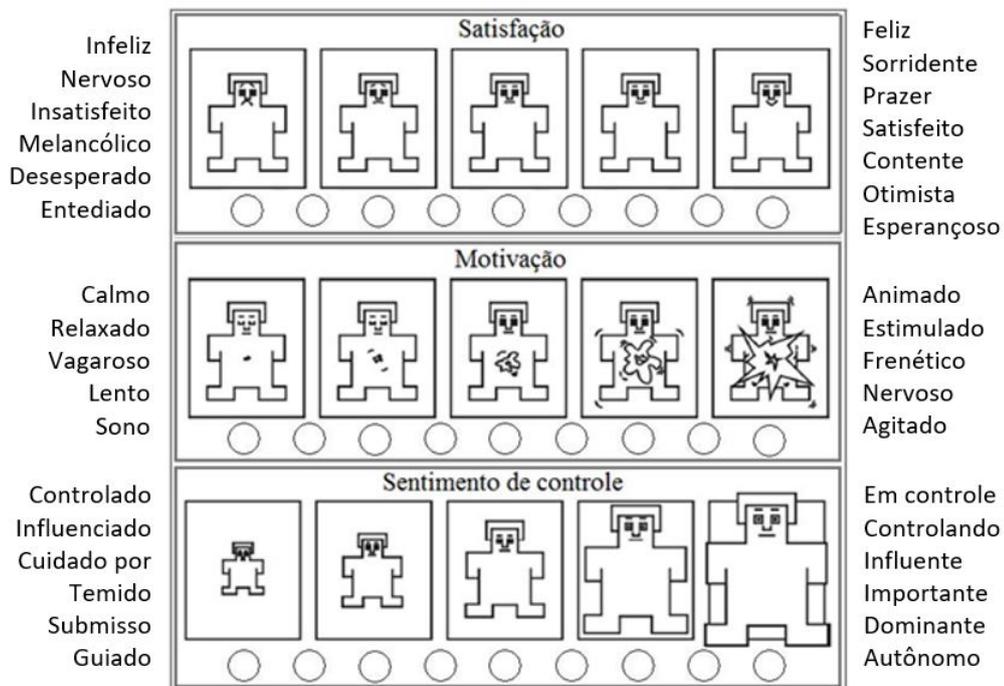
O SUS apresenta-se como uma ferramenta eficiente e eficaz para avaliar o grau de satisfação do usuário em relação a uma interface, bem como para análise do nível de usabilidade de sistemas computacionais (LEITE; PAZ, 2017).

2.2.2.3.2 Self-Assessment Manikin (SAM)

Proposto por LANG (1980), o *Self-Assessment Manikin* é um modelo de questionário que utiliza pictogramas para avaliar a qualidade de um sistema a partir de três dimensões: satisfação, motivação e sentimento de domínio do usuário ao utilizar o sistema. A Figura 2.3 apresenta o formato do questionário.

O questionário possui uma escala com valores de 1 a 9 e utiliza 5 pictogramas para cada uma das dimensões, representando as alterações afetivas em cada uma delas. O usuário deve escolher o pictograma que corresponde ao seu sentimento após utilizar o sistema. A pontuação mínima é um e a pontuação máxima é nove, para cada uma das dimensões. Por utilizar ícones, a escala SAM é facilmente compreendida e se torna fácil aplicá-la em diversos cenários (AGUIRRE et al., 2019). Além disso, os ícones são capazes de ativar estruturas cerebrais, de modo que a resposta do usuário seja muito similar aos estímulos verdadeiros (LANG; BRADLEY, 2007).

Figura 2.3 – Escala SAM - Satisfação, motivação e sentimento de domínio.



Fonte: Adaptado de (LANG, 1980)

2.2.3 Prototipação

A prototipação é uma técnica utilizada para verificar um produto ou ideia antes da solução ser construída. Nesse momento o produto é produzido em uma escala reduzida, apresentando o conteúdo final da solução para ser testado, com o objetivo de avaliar como será a experiência do usuário e estabelecer as melhores soluções para cada problema identificado. De modo que, ao final dos testes, as soluções prototipadas sejam revisadas e melhoradas a partir do resultado da experiência dos usuários (OLIVEIRA, 2022). Para construção de um protótipo há diversas ferramentas disponíveis na *web* que podem ser utilizadas, como por exemplo: Adobe XD, Figma, Framer, Canva, Sketch, entre outras.

De acordo com PIAZZA (2021), há 6 tipos de prototipação que poderão ser utilizadas a depender do objetivo do protótipo, podendo classificá-las em:

- **Prototipação de baixa fidelidade:** apresenta um baixo grau de detalhes e fidelidade ao produto final. É uma opção rápida, barata e pode ser desenhada em papel;
- **Prototipação de alta fidelidade:** apresenta um alto grau de detalhes e fidelidade ao produto final, visando representar os aspectos visuais da solução. Além de permitir que o usuário interaja com as telas, simulando o uso do produto e possibilitando avaliar a experiência do usuário através de testes;
- **Prototipação horizontal:** apresenta as funcionalidades do produto através de ramificações, sem aprofundar em cada uma delas. Ideal para etapas iniciais do projeto;
- **Prototipação vertical:** apresenta as funcionalidades do produto através de ramificações de forma mais profunda, com mais detalhes. Ideal para etapas mais avançadas do projeto;
- **Prototipação evolutiva:** utilizado para apresentar as evoluções e transformações do produto até a versão final, sendo reaproveitado no decorrer do desenvolvimento;
- **Prototipação descartável:** utilizado apenas para apresentar o produto, podendo ser feito por meio de ferramentas ou papel e posteriormente o protótipo é descartado.

Para esse trabalho o tipo de prototipação escolhida para o aprimoramento da interface é a prototipação de alta fidelidade, pois com ela conseguimos simular como as funcionalidades do sistema estão se relacionando e testar as soluções propostas com o público-alvo do produto. A ferramenta escolhida para o desenvolvimento do protótipo foi o Figma, por se tratar de uma ferramenta com boas funções e de fácil manuseio.

O Figma é uma ferramenta para prototipagem de projetos de design e teve seu lançamento ao público em 2016. O software é direcionado a criação de interfaces gráficas e aprimoramento da experiência do usuário com a interface. A ferramenta permite que designers criem projetos em grupo e desenhem juntos na tela em tempo real, oferecendo também o recurso para controle

de versão. Outra grande vantagem do Figma é o funcionamento em navegador, permitindo que a partir de qualquer máquina o usuário consiga acessar a ferramenta.

3 Desenvolvimento

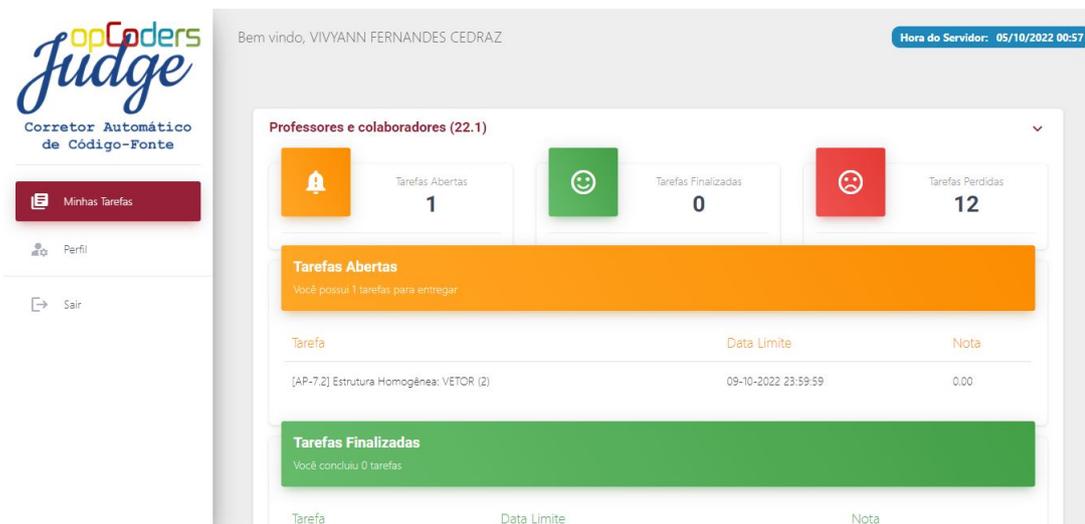
A Subseção 3.1 será dedicada à apresentação da interface do *opCoders Judge*. A Subseção 3.2 apresenta a justificativa para escolha dos métodos de avaliação. Na Subseção 3.3 são descritos os métodos para execução das avaliações e como os testes de usabilidade estão organizados. Por último, a Subseção 3.4 trata a respeito da submissão deste projeto ao Comitê de Ética.

3.1 opCoders Judge

Atualmente, o *opCoders Judge* está sendo utilizado na disciplina de Programação de Computadores I [BCC701] na UFOP como ferramenta de apoio ao ensino de programação. Essa ferramenta foi pensada por um dos professores dessa disciplina, e contou com o apoio de alguns alunos do curso de Ciência da Computação da UFOP para o seu desenvolvimento.

Ao acessar a página do *opCoders Judge*, os alunos têm acesso a tela inicial da ferramenta. Nela é exibido um mini *dashboard*, com o total de tarefas abertas, finalizadas e perdidas, conforme apresentado na Figura 3.1.

Figura 3.1 – Interface da tela inicial do *opCoders Judge*.



Fonte: (DECOM, 2022).

Ao selecionar uma tarefa, o aluno tem acesso à uma série de dicas e informações referente ao conteúdo do exercício, que são oferecidas com o objetivo de auxiliá-lo ainda mais no processo de entendimento do problema. Além de fornecer exemplos de códigos. Essas são características importantes que fortalecem o aspecto educativo da plataforma, como mostra a Figura 3.2.

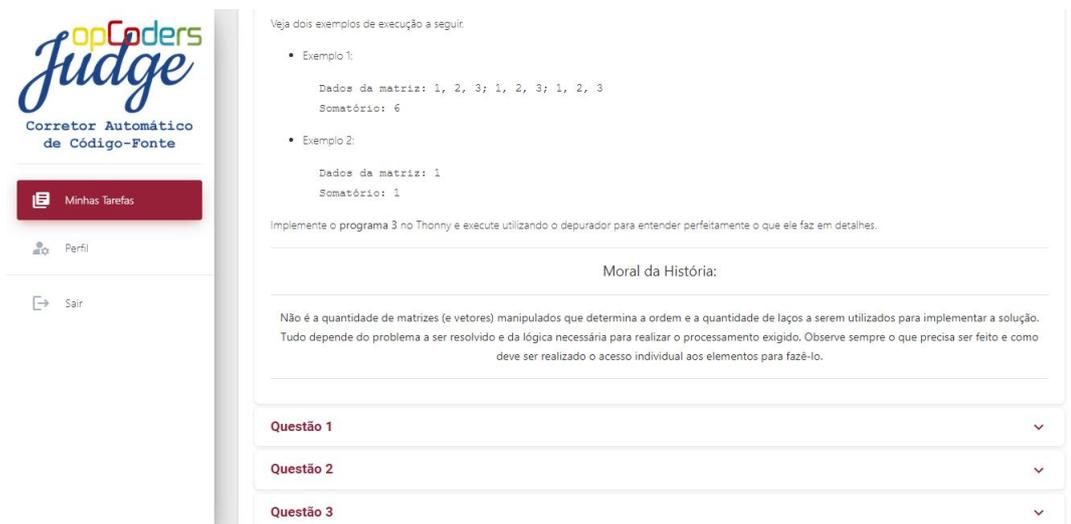
Figura 3.2 – Interface com dicas e informações sobre o conteúdo a ser abordado nas questões.



Fonte: (DECOM, 2022).

Após as informações sobre o conteúdo, na mesma página o aluno encontra as questões a serem respondidas relativas ao tópico abordado, conforme ilustra a Figura 3.3.

Figura 3.3 – Interface para acesso às questões.



Fonte: (DECOM, 2022).

Ao selecionar uma questão, o aluno tem acesso ao seu enunciado e exemplos de execuções corretas, como apresentado na Figura 3.4.

Por fim, ao final da página encontra-se o campo para seleção do arquivo para envio. Como a correção da atividade é automática e a ferramenta não limita a quantidade de vezes que o aluno pode enviar o seu código, a cada arquivo submetido é informado ao aluno a data de envio, a nota, o código-fonte e a correção, apresentado na Figura 3.5.

Figura 3.4 – Interface com enunciado das questões.

[AP-7.4] Estrutura Homogênea: MATRIZ (2)

Descrição da Tarefa

Questão 1

Implemente um programa para apurar as notas dos alunos de BCCT01 em uma prova de múltipla escolha. O programa faz a leitura de um vetor contendo o gabarito das questões, ou seja, cada elemento deste vetor contém a resposta correta de uma questão da prova (valores textuais). Em seguida, o programa faz a leitura das respostas dos alunos em uma matriz. Nesta matriz, as colunas representam as questões e as linhas representam os alunos. Para as entradas do vetor e da matriz são usadas as funções `inputVector` e `inputMatriz`, respectivamente. Se a quantidade de alunos for 0 (string vazia na entrada), o programa imprime a mensagem de erro "Nenhum aluno a avaliar". Caso contrário, se a quantidade de questões representada na matriz for diferente da quantidade de questões no gabarito, o programa imprime a mensagem de erro "Quantidade de questões incompatível". Em ambos os casos, o programa imprime a mensagem de erro e é encerrado. Por outro lado, caso as entradas sejam válidas, seu programa chama a função `notaAluno`, implementada por você, que recebe o vetor de gabarito e a matriz de respostas e preenche um novo vetor contendo as notas dos alunos. Para cada aluno, a nota é obtida nesta função fazendo a comparação entre o gabarito e as suas respostas, contabilizando o número de acertos, a nota final é o número de acertos multiplicado por 10 e dividido pela quantidade de questões. Arredonde os valores das notas para uma casa decimal, usando a função `round(nota, 1)`. Por fim, o programa principal imprimirá as notas dos alunos usando a função `print`. Siga os exemplos de execução apresentados a seguir.

Exemplo de Execução 1:

```

Notas de BCCT01
Digite o gabarito:
>>> B, A, B, E, D

```

Fonte: (DECOM, 2022).

Figura 3.5 – Interface para submissão e correção das questões.

[AP-12] Questão 2 - Exemplo de código 3 hosted with GitHub view raw

Experimente outras soluções trocando as mensagens de texto e/ou alterando os cálculos.

Enviar nova solução (código fonte):

Escolher arquivo: Nenhum arquivo escolhido ENVIAR ARQUIVO ↗

Entregas

Data do Envio	Nota	Código Fonte	Correção
20-10-2022 20:36:34	10	BAIXAR CÓDIGO ↓	VER CORREÇÃO ↕
20-10-2022 20:33:28	0	BAIXAR CÓDIGO ↓	VER CORREÇÃO ↕

Fonte: (DECOM, 2022).

Desse modo, através da correção é possível identificar qual foi o erro cometido na atividade por meio da exibição dos testes de execução do programa. Permitindo que o aluno compreenda onde errou, identificando a diferença entre a saída obtida e a saída esperada e assim corrija a sua tarefa. Podendo enviá-la novamente para correção. A Figura 3.6 apresenta um exemplo onde a execução do programa falhou.

Por outro lado, a Figura 3.7 apresenta um exemplo de correção de um código correto. Exibindo a saída esperada e a saída obtida para cada um dos testes.

A primeira etapa deste trabalho consistiu no estudo e levantamento das características e funcionalidades da ferramenta citada. Além do entendimento do ambiente ao qual está sendo utilizada, permitindo identificar as hipóteses pesquisa e, conseqüentemente, guiar a escolha dos

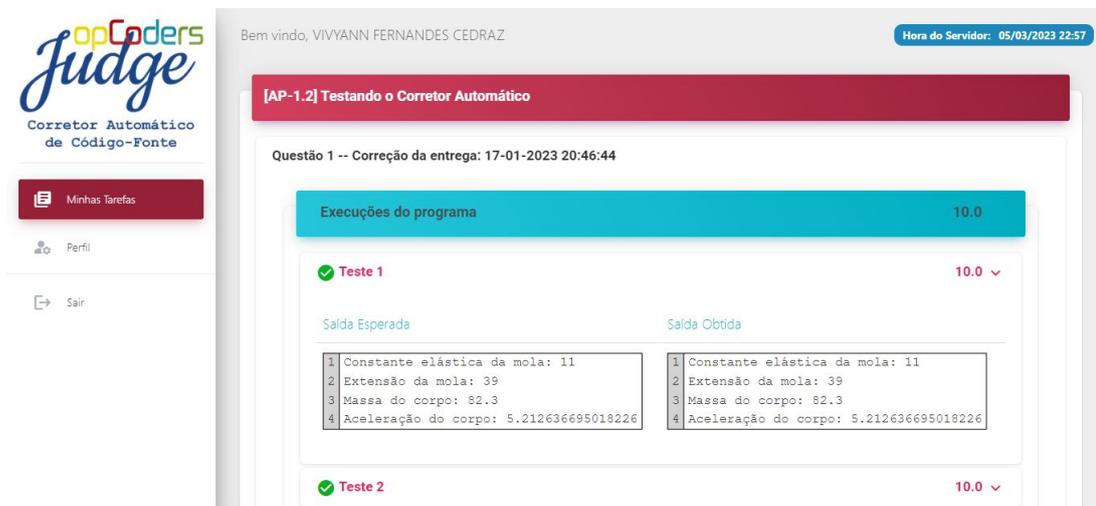
métodos de avaliação de usabilidade mais adequados dado o contexto e problema desta pesquisa.

Figura 3.6 – Interface da tela de correção para uma tarefa com erros.



Fonte: (DECOM, 2022).

Figura 3.7 – Interface da tela de correção para uma tarefa correta.



Fonte: (DECOM, 2022).

3.2 Definição dos Métodos de Avaliação

A partir das hipóteses levantadas e da investigação dos recursos utilizados no *opCoders Judge*, foi realizado um estudo para identificar quais métodos de avaliação de usabilidade existentes na literatura responderiam às hipóteses desta pesquisa. A Tabela 3.1 apresenta os métodos definidos.

A hipótese I desta pesquisa coloca em foco a facilidade de aprendizagem dos usuários para utilizar a ferramenta. Diante disso, um dos métodos disponíveis na literatura capaz de avaliar

a facilidade de uso de um produto é o *System Usability Scale* (SUS), mediante às afirmações 2, 3, 4, 7 e 10 do questionário. Esse método foi escolhido para responder a hipótese I, pois além de ser um método fácil e rápido de ser aplicado, diversos estudos comprovam a sua confiabilidade e efetividade, apresentando-se como uma ferramenta eficaz para mensurar a facilidade de uso, aprendizado e usabilidade de um sistema (SAURO, 2011). Além disso, outra métrica também foi escolhida para comprovar a hipótese I, o tempo de execução da tarefa. Segundo FILHO (2010), quando um produto é fácil de usar, o usuário consegue concluir a sua tarefa com rapidez, pois ele consegue perceber com facilidade a resposta da interface.

A hipótese II desta pesquisa visa comprovar a satisfação dos usuários com a interface, aspecto que também é possível de ser avaliado através do SUS, por meio das afirmações 1, 4, 5, 6, 8 e 9 do questionário. Ademais, além de avaliar a satisfação do usuário, tornou-se necessário também avaliar como a utilização desse sistema interfere nas emoções e sentimentos dos usuários, visto que a utilização de sistemas interativos passou a ser bastante frequente em muitas atividades humanas (BARBOSA; SILVA, 2010). O que justifica também a escolha da utilização de um segundo método de investigação para responder a hipótese II. O *Self-Assessment Manikin* (SAM), uma escala capaz de identificar a satisfação, motivação e sentimento de domínio do usuário ao utilizar o sistema.

Já a hipótese III, direciona para identificação de falhas de usabilidade na interface. Para responder essa hipótese o método escolhido foi a avaliação de heurística, proposta por Jakob Nielsen, e muito utilizada para avaliar sistemas *web*. A escolha dessa técnica se justifica pelo fato de que esse método de avaliação costuma ser mais rápido e tem um custo mais baixo em relação aos métodos por investigação, pois não demanda tempo para seleção e aplicação de testes com usuários, além de possibilitar a identificação de uma série de problemas. No entanto, o resultado da análise representa apenas a experiência do avaliador (BARBOSA; SILVA, 2010). Dessa forma, quando o foco da avaliação também está voltado para a experiência do usuário é interessante que também seja aplicada uma avaliação por investigação, com o objetivo de analisar a opinião do usuário em relação à interface.

Tabela 3.1 – Métodos que respondem as hipóteses de pesquisa apresentadas na Subseção 1.3.2.

Hipótese	Método
I	SUS (afirmações 2, 3, 4, 7 e 10) e tempo de execução da tarefa
II	SUS (afirmações 1, 4, 5, 6, 8 e 9) e SAM
III	Heurísticas de Nielsen

Segundo TENÓRIO et al. (2010), é possível encontrar nas afirmações do SUS as Heurísticas de Nielsen, tornando as avaliações complementares uma a outra. já que as hipóteses anteriores serão respondidas por meio dos resultados do SUS. Assim, optou-se por utilizar o conjunto das 10 Heurísticas de Nielsen para identificação dos problemas de usabilidade na

interface da ferramenta.

3.3 Descrição do Método

A primeira etapa deste trabalho envolve a realização de estudos e pesquisa bibliográfica em artigos, dissertações, livros e teses a respeito dos métodos disponíveis na literatura para análise da ferramenta e comprovações sobre a importância deste trabalho.

A partir dos métodos de avaliação de usabilidade definidos na seção anterior, a segunda etapa do trabalho consistiu na realização da avaliação por inspeção, a partir das heurísticas propostas por Nielsen. A avaliação consistiu na própria avaliadora analisar se as heurísticas estão implementadas na ferramenta avaliada.

Por se tratar de uma ferramenta que já estava sendo utilizada, para evitar gerar ruído entre os usuários, optou-se por construir um protótipo do corretor automático inserindo as correções propostas a partir da avaliação por inspeção e testá-la. A ferramenta utilizada para prototipação da interface foi o Figma⁴.

Posteriormente, foi realizado um teste piloto do protótipo com o professor co-orientador desse projeto e mais alterações foram realizadas após a execução do teste piloto. Foi realizada também uma reunião com o coordenador do desenvolvimento do *opCoders Judge*, e professor da disciplina de Programação de Computadores, para entender o contexto ao qual o sistema está sendo utilizado e discutir as soluções propostas pela avaliadora visando garantir a boa usabilidade do corretor automático.

Por último foi realizada a avaliação por investigação com 30 alunos matriculados em uma disciplina introdutória de programação de computadores. A Figura 3.8 a seguir, apresenta o esquema referente ao fluxo de desenvolvimento deste trabalho.

Os problemas de usabilidade encontrados na ferramenta foram descritos e para cada um deles foi informado onde está localizado, qual a heurística foi violada, a gravidade do problema e a proposta de solução.

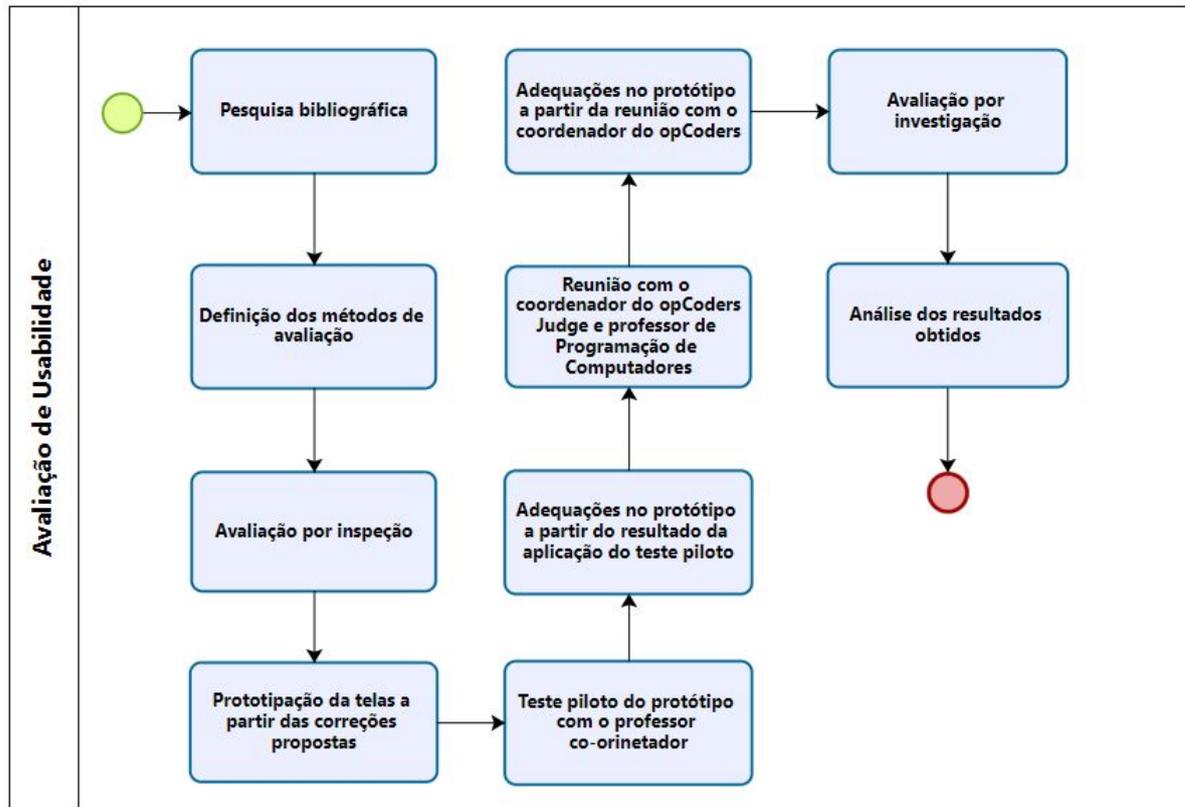
As avaliações por investigação foram realizadas através da aplicação dos questionários SUS e SAM. A Subseção 3.3.1 a seguir apresenta a organização para aplicação dos questionários mencionados.

3.3.1 Organização dos Testes

NIELSEN (2006) recomenda que em estudos qualitativos os testes de usabilidade devam testar 20 usuários. No entanto, não há concordância com esse número entre os pesquisadores do ramo. Segundo SIX; MACEFIELD (2016), em uma amostra com 30 participantes é possível

⁴ <https://www.figma.com/login>

Figura 3.8 – Fluxo de desenvolvimento do trabalho.



Fonte: Elaborado pela autora.

encontrar em média 99% dos problemas da interface, podendo alcançar 100% se aplicado os testes com 50 participantes. Dessa forma, para esta pesquisa foram selecionados de forma aleatória 30 participantes matriculados em uma disciplina introdutória de programação de computadores, para a realização dos testes de usabilidade no protótipo desenvolvido para o corretor automático de código-fonte *opCoders Judge*.

A definição da amostra de participantes da pesquisa levou em consideração o público-alvo do *opCoders Judge*, visto que a ferramenta foi desenvolvida com foco nos alunos que cursam disciplinas introdutórias de programação de computadores. Dessa forma, para garantir a similaridade entre os públicos que utilizam a ferramenta e os que realizariam os testes, foram selecionados alunos matriculados em uma disciplina introdutória de programação de computadores do curso de Ciência da Computação da UFOP. A escolha desse público se deu pelo fato de que esses alunos não possuem conhecimento prévio a respeito do *opCoders Judge*, já que a ferramenta ainda não é utilizada para apoio ao ensino aprendido com os alunos desse curso, evitando que os resultados da pesquisa fossem enviesados, caso fossem aplicados com alunos que já possuíam conhecimento da ferramenta.

Antes de iniciar os testes, todos os participantes da pesquisa receberam os esclarecimentos sobre o Termo de Consentimento Livre e Esclarecido e assinaram, conforme apresentado no

(Apêndice A). A tarefa avaliada durante o teste foi a submissão de uma atividade dentro do *opCoders Judge* para correção.

Após a execução da tarefa os participantes da pesquisa responderam um questionário via *Google Forms* com as afirmações referentes ao SUS, de acordo como o modelo apresentado na Tabela 2.2, e um questionário impresso referente ao SAM, conforme modelo apresentado na Figura 2.3, de acordo com a experiência ao utilizar a interface. Ao final do teste todos os participantes também preencheram um questionário de perfil, via *Google Forms*, com o objetivo de conhecer as características da população inserida no estudo, de acordo com o (Apêndice C).

3.4 Submissão ao Comitê de Ética

A partir da Resolução 466/2012 do Conselho Nacional de Saúde (CNS), todas as pesquisas realizadas no Brasil, independente da área de conhecimento, que envolvam seres humanos devem seguir os procedimentos éticos (BRASIL, 2012). Em 1996, por meio da Resolução 196/1996 do CNS, foi criada a Comissão Nacional de Ética em Pesquisa (CONEP), um sistema responsável por coordenar a rede de Comitês de Ética em Pesquisa que existe em cada instituição, formando o Sistema CEP-CONEP. Tendo também o papel de avaliar os projetos de pesquisa a serem desenvolvidos, verificando se os mesmos respeitam a dignidade humana, sua vontade, faz a correta ciência do consentimento e que não coloque o participante da pesquisa diante de riscos indevidos. O Comitê de Ética em Pesquisa (CEP) tem o dever de proteger os participantes envolvidos na pesquisa ao aprovar eticamente os projetos. Garantindo o respeito aos direitos dos participantes, juntamente com o pesquisador, a instituição e o patrocinador (MARTINS; JUNQUEIRA; ARAUJO, 2021).

Dessa forma, tais pesquisas devem ser submetidas ao CEP para avaliação através da Plataforma Brasil¹³. Para isso o pesquisador deve fazer o seu cadastro e da instituição e enviar os seguintes documentos: material do projeto detalhado, modelo do Termo de Consentimento Livre e Esclarecido (TCLE), instrumentos de coleta de dados, cronograma, orçamento e folha de rosto assinada pela instituição. É também através da Plataforma que o pesquisador consegue acompanhar em qual etapa de avaliação o seu projeto está.

Por se tratar de um trabalho que envolve testes com seres humanos, foi encaminhado ao CEP-UFOP todos os documentos referentes a esta pesquisa para avaliação. O presente trabalho foi aprovado pelo Comitê de Ética da UFOP e pode ser identificado através do Certificado de Apresentação de Apreciação Ética (CAAE) de número 63182722.9.0000.5150.

¹³ <https://plataformabrasil.saude.gov.br>

4 Resultados

Nesta seção serão apresentados os resultados obtidos a partir da aplicação dos testes com os participantes da pesquisa e as discussões a respeito da mesma. A Subseção 4.1 será dedicada à apresentação dos resultados a partir da avaliação por inspeção da interface do *opCoders Judge*. A Subseção 4.2 será dedicada à apresentação das telas propostas e as diferenças entre elas e a versão avaliadas. Na Subseção 4.3 são apresentados os principais resultados do questionário de perfil. Na Subseção 4.4 são apresentados os resultados da avaliação por investigação com o SUS. Já na Subseção 4.5 são apresentados os resultados da avaliação por investigação a partir do questionário SAM. A Subseção 4.6 traz os resultados do tempo de execução dos testes e por fim a Subseção 4.7 discorre sobre os resultados alcançados.

4.1 Avaliação por Inspeção (Heurísticas de Nielsen)

Para realizar a avaliação por inspeção a partir das 10 Heurísticas de Nielsen, foram definidas um conjunto de ações a serem executadas no *opCoders Judge*, conforme apresentado na Tabela 4.1. Com o objetivo de direcionar o foco da avaliação, permitindo uma análise de qualidade sob pontos essenciais da aplicação a partir dos critérios de usabilidade descritos através das heurísticas de Nielsen. A inspeção foi realizada pela autora deste trabalho a partir da execução das ações mencionadas na ferramenta.

Tabela 4.1 – Conjunto de ações executadas no *opCoders Judge*.

N	Ações
1	Após acessar o corretor envie a atividade “Testando o corretor automático”
2	Após enviar a atividade consulte a correção
3	Altere a sua senha de acesso à ferramenta

Todas as ações executadas na ferramenta foram avaliadas com base em cada Heurística de Nielsen. A Tabela 4.2 apresenta a quantidade de problemas encontrados para cada escala de classificação de um problema. A lista com todos os problemas identificados estão apresentados no (Apêndice B).

A partir dos problemas encontrados na interface através da avaliação por inspeção, foi construído um protótipo de alta fidelidade baseado nas soluções propostas para cada um dos problemas identificados. Posteriormente esse protótipo foi testado, com o objetivo de verificar se as soluções apresentadas trariam uma boa usabilidade à ferramenta. O protótipo construído é apresentado na Subseção 4.2 a seguir.

Tabela 4.2 – Total de problemas encontrado nas ferramentas por escala de classificação.

Escala de classificação dos problemas	<i>opCoders Judge</i>
Cosmético	1
Menor	1
Médio	3
Grave	3
Catastrófico	5

4.2 Protótipo Desenvolvido

A construção do protótipo de alta fidelidade do *opCoders Judge* se deu a partir da correção dos problemas descritos no Apêndice B. A seguir são apresentadas as telas desenvolvidas e o detalhamento de cada uma delas.

Tela de login: como os alunos são cadastrados automaticamente no *opCoders Judge* ao serem matriculados em uma disciplina de Programação de Computadores I (BCC701), a ferramenta não oferece ao aluno a funcionalidade que o permita realizar cadastros no sistema. Dessa forma, visando prevenir erros, a grande diferença entre a tela avaliada e a tela proposta é a presença de um texto que informa ao usuário que as suas credenciais de acesso foram enviadas ao seu e-mail institucional, com o objetivo de evitar que o usuário fique perdido e não consiga realizar o login ou seja necessário definir uma nova senha. A tela corresponde a Figura 4.1.

Figura 4.1 – Tela de login.

Fonte: Elaborado pela autora.

A primeira diferença entre as interfaces é o posicionamento do menu. Na versão avaliada o menu estava localizado no canto esquerdo da tela na vertical, ocupando uma parte considerável. Com o objetivo tornar o projeto estético minimalista, o menu foi reduzido e transferido para a parte superior da tela. Para garantir a flexibilidade e eficiência de uso, foi inserido em todas as telas um atalho para voltar à tela anterior. Tal recurso não existia na interface avaliada.

Tela de primeiro uso: visando garantir que os usuários conseguirão concluir as suas tarefas, foi identificada a necessidade de inserir uma tela de primeiro uso que irá apresentar ao usuário informações básicas que ajude-o a utilizar o sistema. Essa tela aparecerá apenas na primeira vez que o aluno fizer login na ferramenta. Na versão avaliada não havia essa tela. A tela corresponde à Figura 4.2.

Figura 4.2 – Tela de primeiro uso.

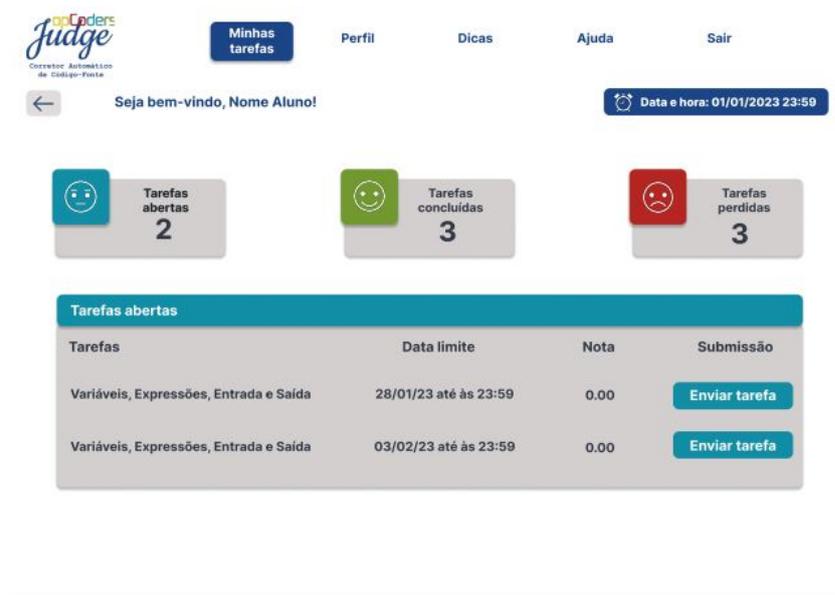


Fonte: Elaborado pela autora.

Tela de tarefas abertas: ao realizar o login na ferramenta, o usuário já é direcionado a página das tarefas abertas e um mini *dashboard* com o *status* de todas as tarefas. A diferença da tela proposta é que na anterior, ao realizar o login era exibida a lista com todas as tarefas abertas, perdidas e concluídas, gerando um grande volume de informações. Outra diferença importante é a inserção de um botão para enviar a tarefa. A tela corresponde a Figura 4.3.

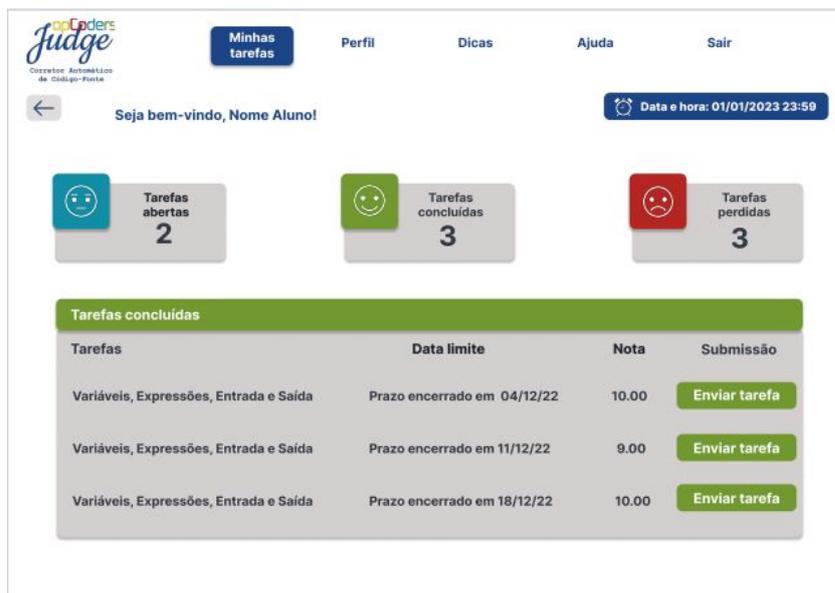
Tela de tarefas concluídas: a tela de tarefas concluídas segue o mesmo padrão e características da tela de tarefas abertas. A tela é aberta ao clicar no *status* do *dashboard* das tarefas concluídas e corresponde a Figura 4.3.

Figura 4.3 – Tela de tarefas abertas.



Fonte: Elaborado pela autora.

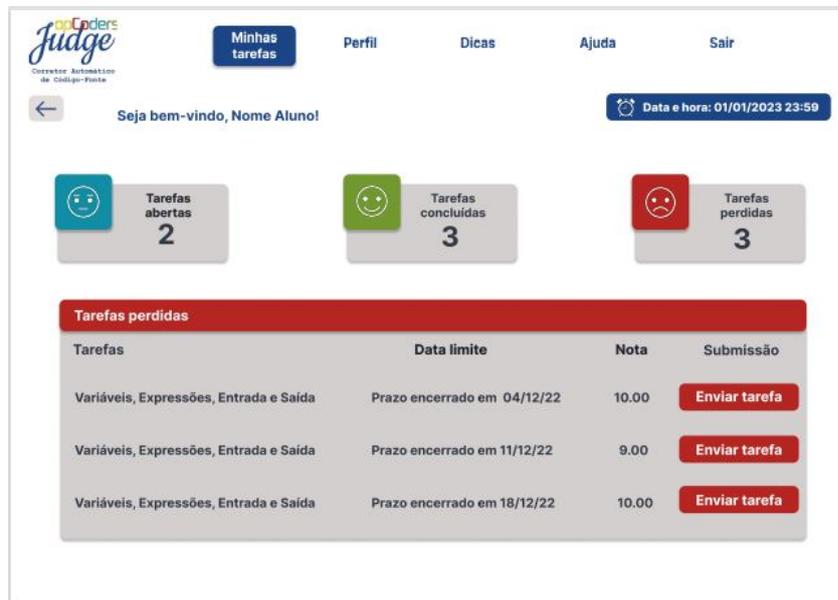
Figura 4.4 – Tela de tarefas concluídas.



Fonte: Elaborado pela autora.

Tela de tarefas perdidas: a tela de tarefas perdidas segue o mesmo padrão e características da tela de tarefas abertas. A tela é aberta ao clicar no *status* do *dashboard* das tarefas perdidas e corresponde a Figura 4.5.

Figura 4.5 – Tela de tarefas perdidas.



Fonte: Elaborado pela autora.

Tela para envio de tarefas: a tela para envio das tarefas traz várias mudanças na interface. Além de manter a nova proposta de menu, temos:

- Substituição da informação do horário do sistema para tempo restante para que a tarefa em questão fosse enviada;
- Inserção do botão “Dicas” com as informações referentes ao conteúdo da atividade para dar suporte na construção da solução, com o objetivo de reduzir a quantidade de informações na tela e dar visibilidade à descrição da tarefa;
- Para evitar erros, foi inserido um campo para seleção da linguagem de programação que será utilizada pelo aluno para resolver o exercício. Na versão anterior essa caixa de seleção não existia, nem mesmo uma mensagem para informar ao usuário qual a linguagem permitida para envio;
- Para manter os padrões da interface o botão “Escolher arquivo” teve a sua cor alterada e tamanho aumentado.

A Figura 4.6 apresenta a tela descrita.

Figura 4.6 – Tela para envio da tarefa.



Fonte: Elaborado pela autora.

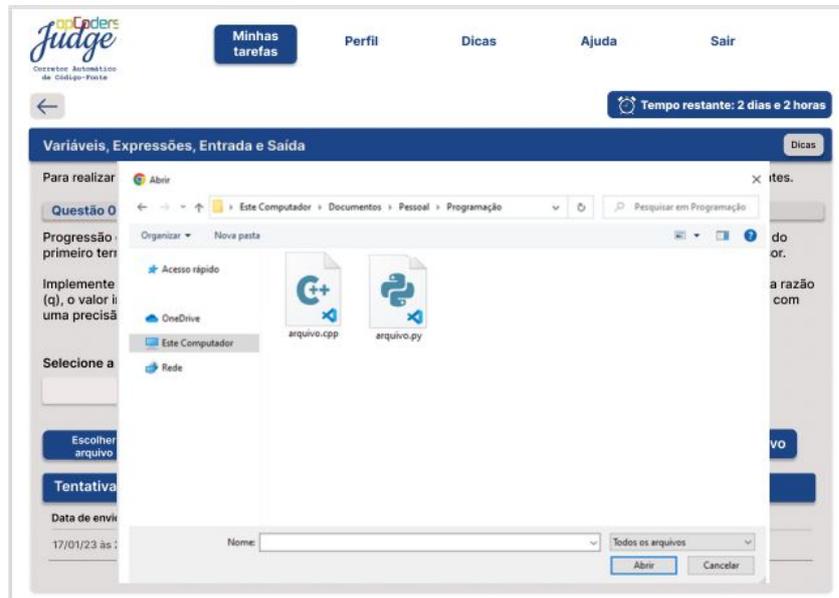
A Figura 4.7 exibe as linguagens a serem escolhidas a partir da inserção do novo campo para seleção da linguagem de programação. Já a Figura 4.8 apresenta a tela para seleção do arquivo referente ao exercício de programação a ser enviado para correção. As duas telas descritas seguem o mesmo padrão da tela referente ao envio da tarefa, representada pela Figura 4.6.

Figura 4.7 – Tela para seleção da linguagem de programação.



Fonte: Elaborado pela autora.

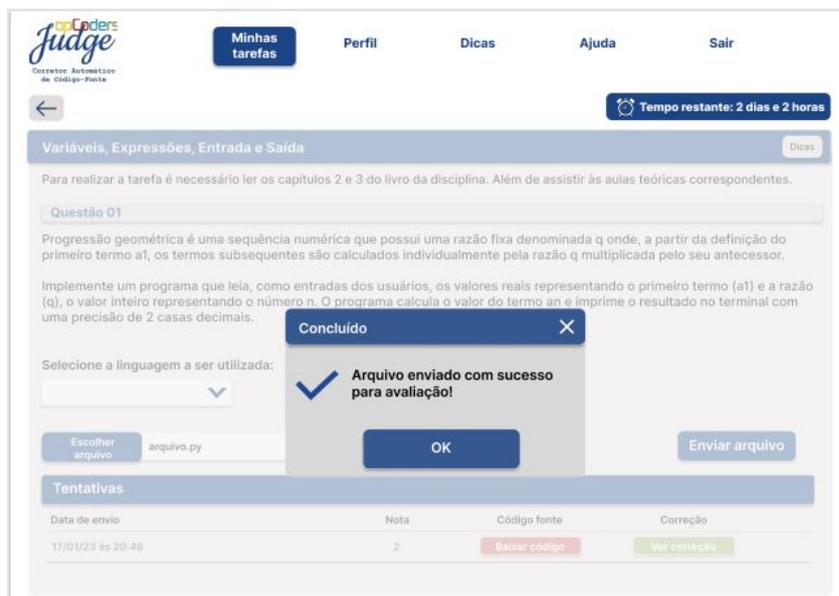
Figura 4.8 – Tela para seleção do arquivo.



Fonte: Elaborado pela autora.

Tela de aviso que o arquivo foi enviado para avaliação: essa tela foi alterada para dar mais destaque à mensagem, garantindo o feedback ao usuário caso o arquivo enviado corresponda a extensão selecionada, confirmando a visibilidade do estado do sistema. Na versão avaliada essa mensagem aparecia em um tamanho reduzido. A Figura 4.9 apresenta a tela descrita.

Figura 4.9 – Tela de aviso que o arquivo foi enviado para avaliação.

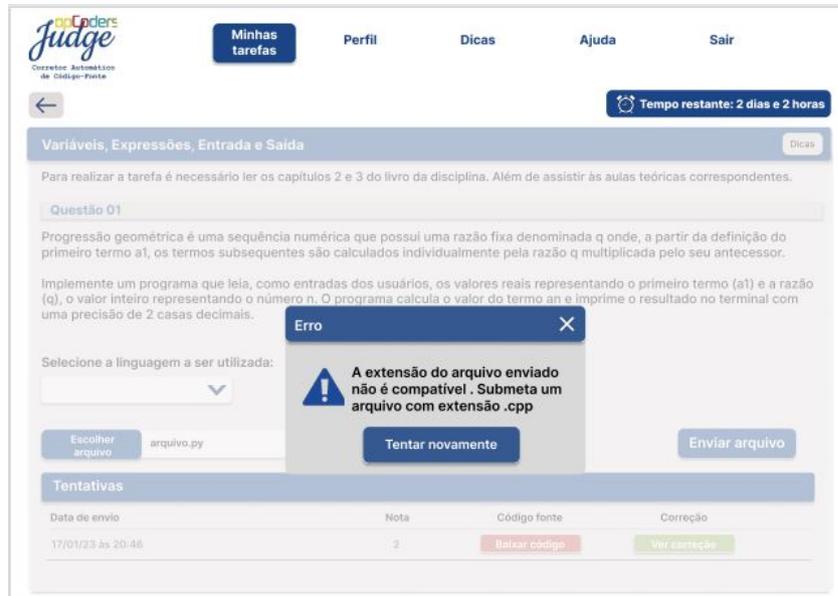


Fonte: Elaborado pela autora.

Tela de erro para aviso de que a extensão é diferente da linguagem: essa tela foi criada com o objetivo de oferecer feedback ao usuário caso o arquivo enviado não corresponda a

extensão selecionada, garantindo a visibilidade do estado do sistema. Na versão avaliada essa tela não existia. A Figura 4.10 apresenta a tela descrita.

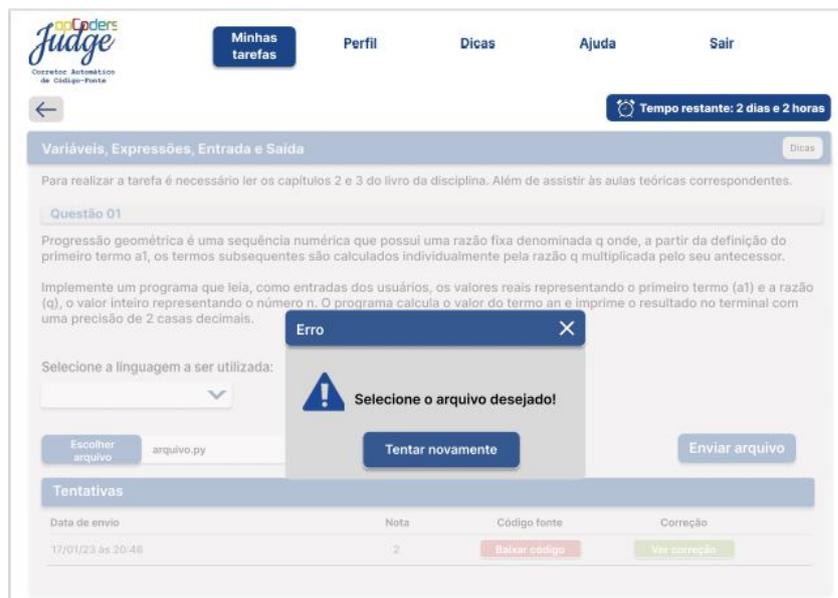
Figura 4.10 – Tela de erro para aviso de extensão diferente da linguagem.



Fonte: Elaborado pela autora.

Tela de erro para avisar ao usuário para selecionar o arquivo: essa tela foi criada com o objetivo de oferecer feedback ao usuário caso ele tente enviar a tarefa antes de selecionar o arquivo correspondente, garantindo a visibilidade do estado do sistema. Na versão avaliada essa tela não existia. A Figura 4.11 apresenta a tela descrita.

Figura 4.11 – Tela de erro para selecionar o arquivo desejado.



Fonte: Elaborado pela autora.

Tela da tarefa após o envio: essa tela também apresenta todas as características descritas na interface para envio da tarefa (Figura 4.6). A diferença entre a versão proposta e a versão avaliada é que, após enviar o arquivo, enquanto o sistema realiza a correção, aparecia para o usuário um botão escrito “Pendente”, podendo gerar dúvidas no aluno de que a sua tarefa não foi enviada. Para garantir a correspondência entre o sistema e o mundo real, foi realizada a troca do nome do botão para “Aguarde a correção”. Além disso, foi inserida uma caixa com a mensagem “Último envio” para indicar qual foi a última tentativa submetida. A Figura 4.12 apresenta a tela descrita.

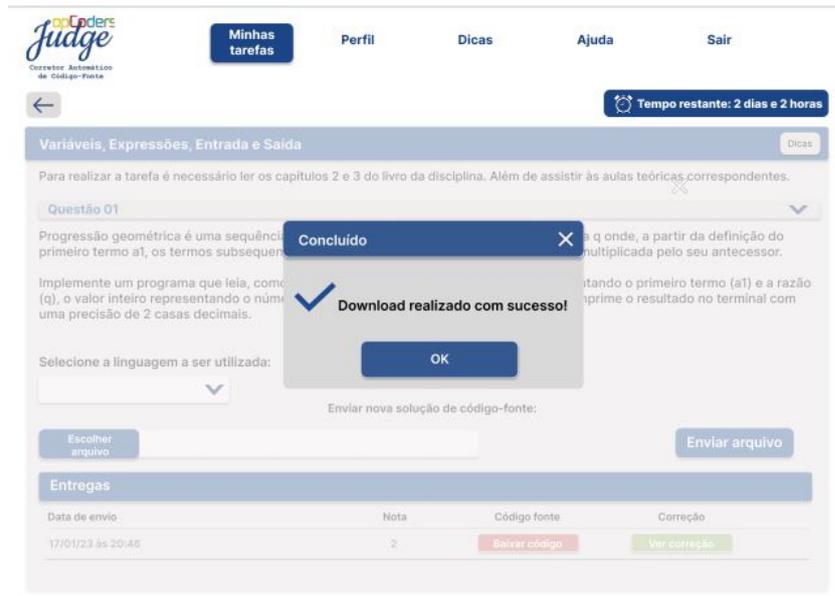
Figura 4.12 – Tela da tarefa após envio.

Data de envio	Nota	Código fonte	Correção
17/01/23 às 20:30	2	Baixar código	Ver correção
17/01/23 às 20:40	10	Baixar código	Aguarde correção

Fonte: Elaborado pela autora.

Tela com mensagem de confirmação do download da tarefa: essa tela foi criada com o objetivo de oferecer feedback ao usuário após selecionar o botão para download da tarefa, garantindo a visibilidade do estado do sistema. Na versão avaliada essa tela não existia. A Figura 4.13 apresenta a tela descrita.

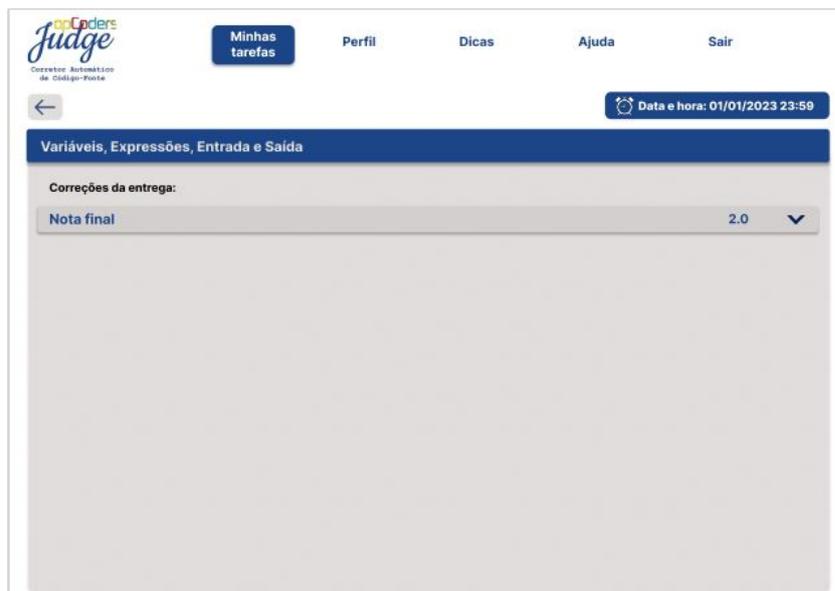
Figura 4.13 – Tela com mensagem de confirmação do download da tarefa.



Fonte: Elaborado pela autora.

Tela para visualização da correção da tarefa: nas telas de correção das tarefas não foram encontrados problemas de usabilidade. No entanto, é recomendado que ao selecionar o botão para visualização da correção não seja aberta outra página no navegador, como acontece na versão avaliada, e sim a atualização dentro da própria página. As Figuras 4.14, 4.15 e 4.16 apresentam as telas descritas.

Figura 4.14 – Tela para visualização da correção da tarefa.



Fonte: Elaborado pela autora.

Figura 4.15 – Tela para visualização da correção da tarefa por teste.

Correções da entrega:	
Nota final	2.0
Teste 1	2.0
Teste 2	2.0
Teste 3	2.0
Teste 4	2.0
Teste 5	2.0
Teste 6	2.0
Teste 7	2.0
Teste 8	2.0
Teste 9	2.0
Teste 10	2.0

Fonte: Elaborado pela autora.

Figura 4.16 – Tela para visualização do detalhamento da correção da tarefa.

Correção da entrega.	
Nota final	2.0
Teste	2.0

Saída esperada:	Saída obtida:
1 Constante elástica da mola: 11	1 Constante elástica da mola: 11
2 Extensão da mola: 39	2 Extensão da mola: 39
3 Massa do corpo: 82.3	3 Massa do corpo: 82.3
4 Aceleração do corpo: 5.212636695018226	4 Aceleração do corpo: 429.0

Fonte: Elaborado pela autora.

Tela de perfil: na tela de perfil não foram encontrados problemas de usabilidade, as mudanças são referentes apenas ao layout da ferramenta. As Figuras 4.17 e 4.18 apresentam as telas descritas.

Figura 4.17 – Tela de perfil.

The screenshot shows the 'Perfil' (Profile) page of the Judge system. At the top left is the logo for 'Judge' (Criador Automático de Código-Fonte) and the text 'Minhas tarefas'. A navigation menu includes 'Perfil', 'Dicas', 'Ajuda', and 'Sair'. A welcome message 'Seja bem-vindo, Nome Aluno!' is displayed, along with a clock icon and the date/time 'Data e hora: 01/01/2023 23:59'. The page is divided into two main sections: 'Dados pessoais:' (Personal Data) and 'Alterar senha:' (Change Password). The 'Dados pessoais:' section contains three input fields: 'Nome' (Nome e sobrenome), 'Email' (nome.sobrenome@aluno.ufop.edu.br), and 'CPF' (123.456.789-10). The 'Alterar senha:' section contains three input fields: 'Senha atual', 'Nova senha', and 'Repetir nova senha', followed by an 'Alterar senha' button.

Fonte: Elaborado pela autora.

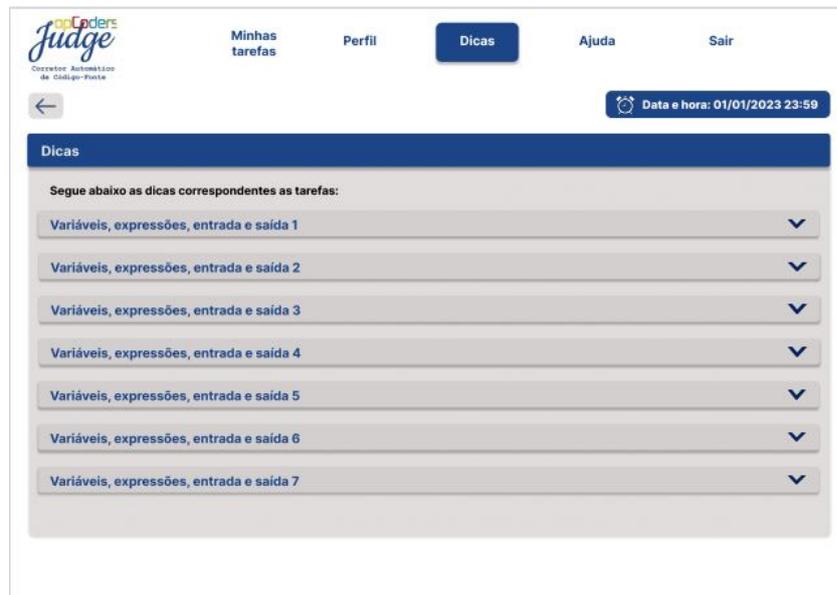
Figura 4.18 – Tela de aviso que a senha foi alterada.

This screenshot is identical to Figure 4.17, but with a modal dialog box overlaid in the center. The dialog box has a title bar 'Concluído' and a close button 'X'. It contains a blue checkmark icon and the text 'Senha alterada com sucesso!' (Password changed successfully!). Below the text is an 'OK' button. The background content of the profile page is dimmed.

Fonte: Elaborado pela autora.

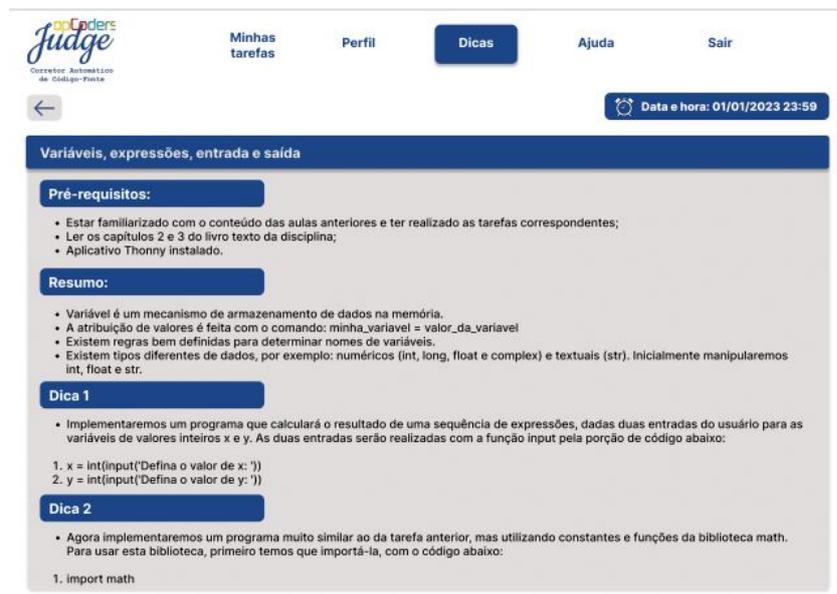
Tela de dicas: na versão avaliada da ferramenta, todas as dicas sobre um determinado conteúdo da matéria estavam junto com a descrição da tarefa, deixando a página com muita informação. Visando construir uma interface minimalista, foi criada uma nova opção no menu da ferramenta com todas as dicas sobre cada conteúdo da disciplina, organizando as informações dentro do sistema. As Figuras 4.19 e 4.20 apresentam as telas descritas.

Figura 4.19 – Tela de dicas.



Fonte: Elaborado pela autora.

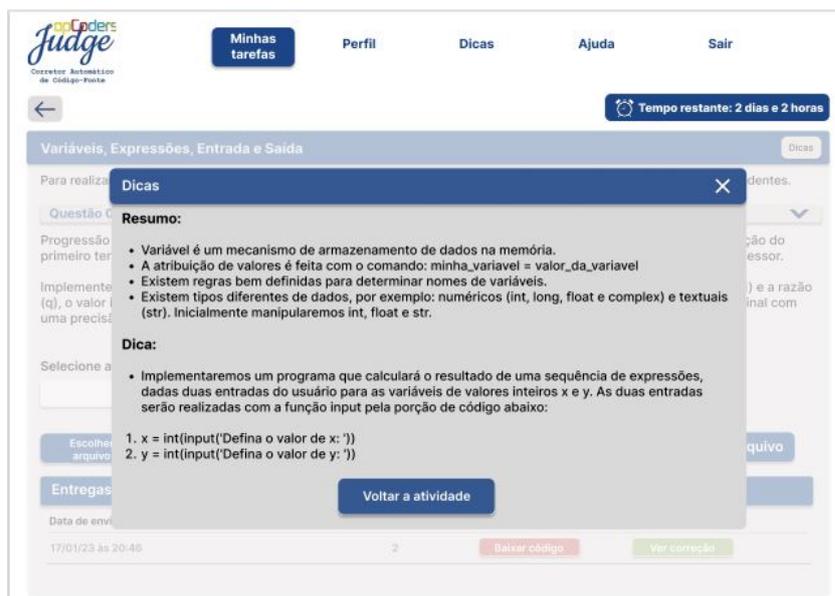
Figura 4.20 – Tela com detalhamento das dicas.



Fonte: Elaborado pela autora.

Tela de dicas dentro das tarefas: na versão avaliada da ferramenta essa tela não existia, a mesma foi criada com o objetivo de organizar as informações e oferecer ajuda de forma rápida ao usuário, sem ser necessário sair da tela das tarefas. A Figura 4.21 apresenta a tela descrita.

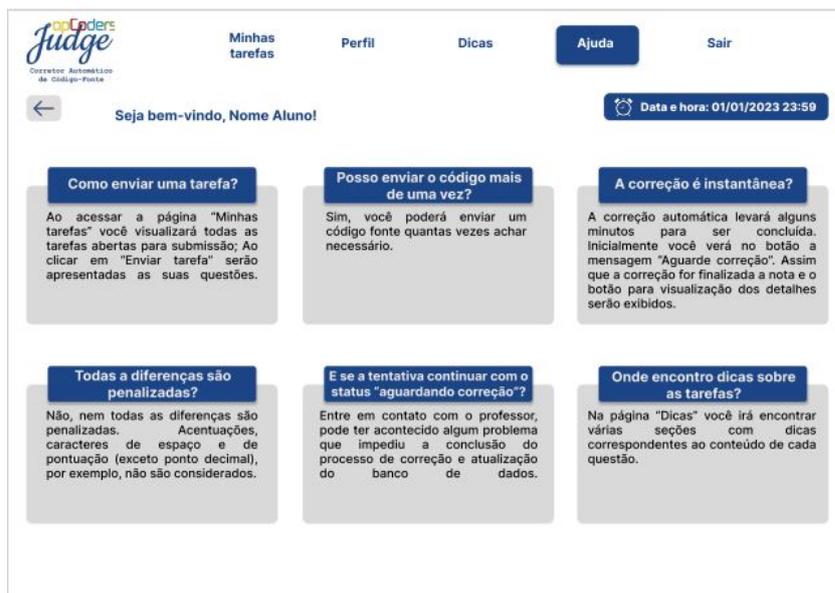
Figura 4.21 – Tela de dicas dentro das tarefas.



Fonte: Elaborado pela autora.

Tela de ajuda: na versão avaliada da ferramenta, todas as informações sobre a utilização do sistema estavam dentro da tarefa “Testando o corretor automático”, fazendo com que o usuário memorizasse as informações importantes. Na versão proposta, foi inserido no menu a opção “Ajuda” organizando e fornecendo todas as informações necessárias para auxiliar o usuário na realização da sua tarefa. A Figura 4.22 apresenta a tela descrita.

Figura 4.22 – Tela de ajuda.



Fonte: Elaborado pela autora.

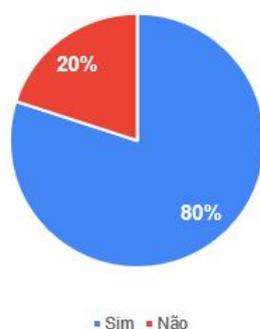
4.3 Questionário de perfil

Após a avaliação por investigação no protótipo desenvolvido, foi solicitado aos participantes o preenchimento de um questionário de perfil, com o objetivo de entender as características do público participante da pesquisa. O questionário foi disponibilizado via *Google Forms* e foi respondido por 30 alunos, o que corresponde ao total de alunos participantes. Deste total, 13 alunos (43%) declararam pertencer ao gênero feminino e 17 alunos (57%) ao gênero masculino. Todos os participantes são alunos do curso de Ciência da Computação da UFOP.

Dentre eles, 24 alunos (80%) afirmaram estar cursando a disciplina introdutória de programação pela primeira vez, e 6 alunos (20%) afirmaram estar cursando a disciplina pela segunda vez ou mais, conforme apresentado na Figura 4.23. Demonstrando um percentual significativo de reprovação dentro da amostra selecionada.

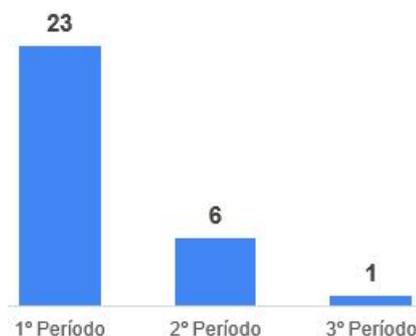
Questionados a respeito do período de graduação que estão cursando, 23 alunos (77%) informaram estar cursando o 1º período, 6 alunos (20%) informaram estar cursando o 2º período e 1 aluno (3%) informou estar cursando o 3º período, de acordo com a Figura 4.24.

Figura 4.23 – Repetência na disciplina.



Fonte: Elaborado pela autora.

Figura 4.24 – Período dos participantes.

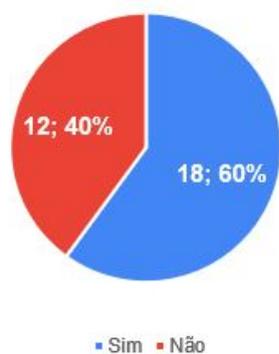


Fonte: Elaborado pela autora.

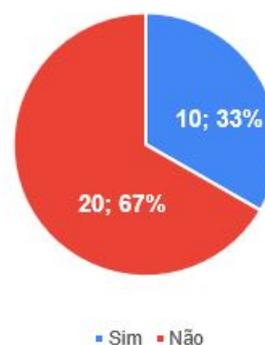
Para avaliar a nível de familiaridade dos alunos com o tipo de ferramenta avaliada e noções prévias de programação, foi questionado se já haviam utilizado algum corretor automático de código-fonte e se já programavam antes de iniciar a disciplina introdutória de programação. 18 alunos (60%) afirmaram já terem utilizado um corretor automático de código-fonte e 12 alunos (40%) ainda não tinham utilizado esse tipo de ferramenta, conforme apresentado na Figura 4.25. A respeito das noções de programação, apenas 10 alunos (33%) afirmaram que já programavam antes de iniciar a disciplina, enquanto 20 alunos (67%) ainda não programavam, como apresentado na Figura 4.26.

Diante desse cenário, podemos perceber que dado o resultado da amostra selecionada, um percentual considerável do público-alvo da ferramenta não possui conhecimentos anteriores sobre programação e ferramentas de correção automática, evidenciando a necessidade do sistema possuir uma interface intuitiva e que permita o usuário concluir a sua tarefa de forma satisfatória.

Figura 4.25 – % de alunos que já utilizaram algum corretor automático. Figura 4.26 – % de alunos que já programavam antes da disciplina.



Fonte: Elaborado pela autora.



Fonte: Elaborado pela autora.

Também foi solicitada a opinião dos participantes a respeito da interface do *opCoders Judge*. As respostas obtidas estão listadas na Tabela 4.3.

Tabela 4.3 – Opinião dos participantes sobre a interface do *opCoders Judge*.

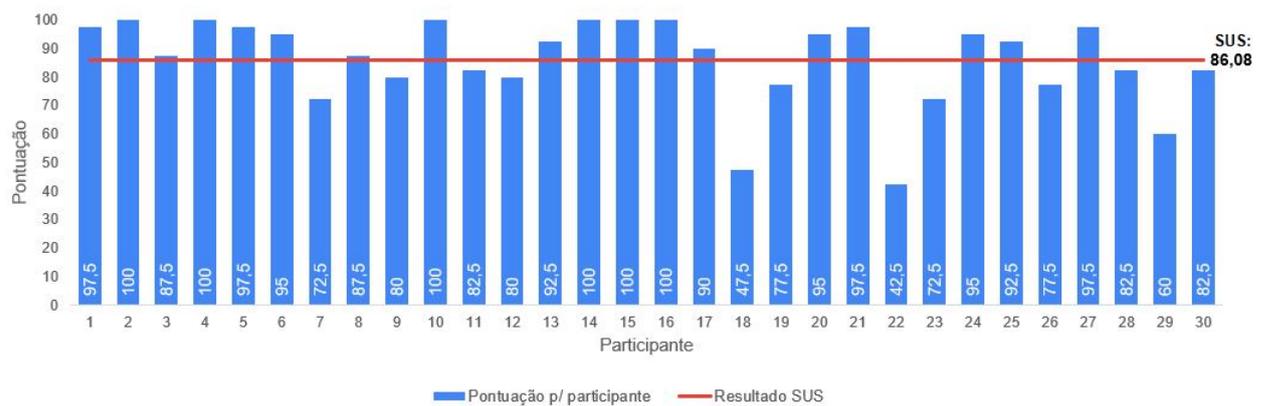
Participante	Opinião
01	Achei uma interface muito interessante e explicativa para os estudantes da computação, principalmente do primeiro período que está iniciando.
02	Achei a interface muito boa e intuitiva, não tive nenhuma dificuldade em me orientar pela ferramenta.
03	A interface é simples e bem fácil de compreender, talvez uma mudança do contraste de cores seria bom.
04	Muito bonita e prática.
05	Bastante intuitiva.
06	Achei bem simples e fácil de usar e que com poucos minutos de uma pessoa que não entende sobre computadores conseguiria usar tranquilamente.
07	Simple de usar e parece completa, porém, ao enviar o arquivo, gostaria de ter visto meu código para poder confirmar que enviei corretamente.
08	Gostei, achei bem dinâmica e simples para o entendimento. Além de dar dicas, que é muito útil para compreensão.
09	Muito prática.
10	Clara, sucinta e boa por ser específica para nosso curso.
11	Prática, limpa e fácil de usar e intuitiva.
12	Bem prática
13	Interface simples de entender.
14	Muito boa, fácil aplicabilidade e entendimento, layout confortável de manipular e bastante intuitiva.
15	Muito pratica, rápida e de fácil acesso.
16	Achei interessante, fácil de usar e prático.
17	Muito boa, fácil pra se usar.
18	Achei bem interativa e de fácil utilização. Muito bacana a ideia a forma para auxiliar as pessoas.
19	E boa a interface, estão bem dispostos os elementos.
20	Bastante prática e fácil de se usar.
21	Simple, completa, intuitiva e objetiva.
22	Interessante.
23	Achei boa, porém com pouca confirmação.
24	Simple e intuitiva.
25	Interface intuitiva e fácil de usar.
26	Simple.
27	Simple, fácil de usar.
28	Super útil, extremamente necessária.
29	Gostei da interface, achei bem intuitiva.
30	Simple e fácil de usar.

4.4 Avaliação por Investigação (SUS)

Com a aplicação do questionário SUS, foi possível avaliar a efetividade, eficiência e satisfação dos participantes após o uso da interface. Ao final do teste de usabilidade, todos os participantes receberam um *link* em seus e-mails institucionais com o acesso ao questionário, conforme apresentado na Tabela 2.2, e as instruções para preenchimento da avaliação.

O resultado geral da avaliação de usabilidade do protótipo apresentado na Subseção 4.2, a partir da aplicação do questionário SUS, foi de **86,08** pontos, o que segundo BROOKE et al. (1996) é considerado uma usabilidade excelente. No entanto, o resultado final do SUS poderia ter sido maior, visto que o participante 22 não respondeu a afirmação de número 1. O desvio padrão das respostas é de 14,9, valor que corresponde 0,58% do total das respostas, evidenciando a similaridade entre as respostas. A Figura 4.27 apresenta o resultado da avaliação por participante e o resultado geral do SUS.

Figura 4.27 – Resultado da avaliação de usabilidade a partir do SUS.



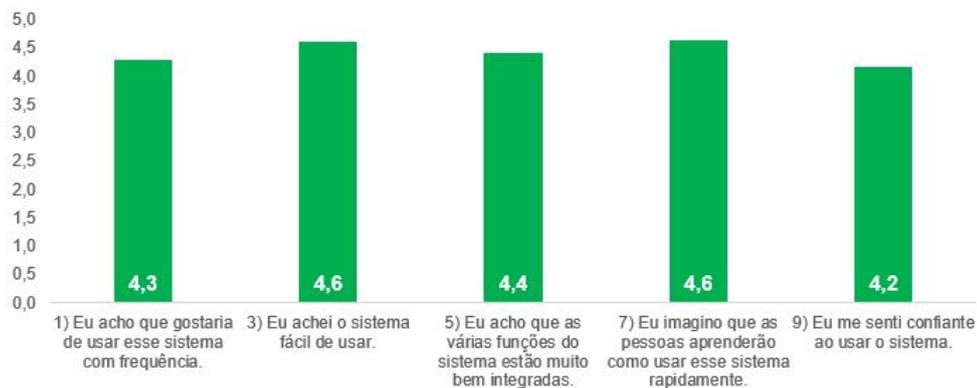
Fonte: Elaborado pela autora.

A partir do cálculo das notas individuais, identificou-se a pior avaliação com pontuação igual a 42,5 pontos. Tal resultado corresponde ao participante que deixou de responder uma afirmação, podendo esse resultado ter sido melhor. A segunda pior avaliação tem pontuação de 47,5 pontos. Porém, esse participante avaliou quase todas as afirmações com nota 5, apenas uma afirmação recebeu nota 4. Embora todos os participantes tenham recebido as orientações para preenchimento dos questionário, podemos interpretar que, possivelmente, o participante não tenha compreendido como responder o questionário. Uma evidência disto é que na avaliação SAM, esse mesmo participante atribuiu apenas notas positivas nas avaliações das dimensões. Além disso, ao responder a pergunta sobre o que achou da interface no questionário de perfil o usuário declarou: “Achei bem interativa e de fácil utilização. Muito bacana a ideia a forma para auxiliar as pessoas”. Já a melhor avaliação, apresenta pontuação igual a 100 pontos e foi obtida 6 vezes.

Dentre as afirmações positivas (1, 3, 5, 7 e 9), todas receberam pontuações altas, em média

as respostas variaram entre 4.2 e 4.6. Dessas afirmações, a que obteve o menor resultado foi a de número 9: “Eu me senti confiante ao usar o sistema”, impactando no resultado geral da avaliação. Dessa forma, podemos observar uma oportunidade de melhoria em relação à confiança dos usuários ao utilizar a ferramenta. Esse resultado pode ser confirmado a partir da observação dos participantes durante o teste. Já que uma dificuldade recorrente em grande parte dos participantes é que, ao enviar a tarefa para correção, eles aguardaram a interface retornar alguma mensagem com a sua nota, e não perceberam que a tarefa já havia sido concluída, gerando dúvida no usuário. Ainda nas afirmações positivas, a que obteve a maior pontuação foi a de número 7: “Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente”, evidenciando grande satisfação dos usuários quanto à interface. A Figura 4.28 a seguir apresenta a média de resposta das afirmações positivas.

Figura 4.28 – Média das respostas para as afirmações positivas.



Fonte: Elaborado pela autora.

Dentre as afirmações negativas (2, 4, 6, 8 e 10), todas receberam pontuações baixas, em média as respostas variaram entre 1.4 e 1.6. A Figura 4.29 a seguir apresenta a média de resposta das afirmações negativas.

Figura 4.29 – Média das respostas para as afirmações negativas.



Fonte: Elaborado pela autora.

4.5 Avaliação por Investigação (SAM)

A partir do questionário SAM, foi possível avaliar as três dimensões emocionais (satisfação, motivação e sentimento de domínio) dos participantes após o uso da interface. Ao final do teste de usabilidade, todos os participantes receberam uma folha com o questionário SAM impresso, conforme apresentado na Figura 2.3, e as instruções para preenchimento da avaliação da interface.

De acordo com as imagens apresentadas no questionário SAM, para análise dos resultados as respostas foram divididas em três grupos:

- **Respostas negativas:** respostas entre 1 e 4;
- **Respostas neutras:** respostas iguais a 5;
- **Respostas positivas:** respostas entre 6 e 9.

Dentre as dimensões avaliadas, a Satisfação obteve a maior pontuação entre as demais, com média de respostas em 8 pontos. Conforme apresentado na Figura 4.30, apenas o participante 7 atribuiu uma nota neutra nessa dimensão. As demais respostas se classificaram como positivas, sendo iguais ou superiores a 6, ocupando toda parte superior do gráfico.

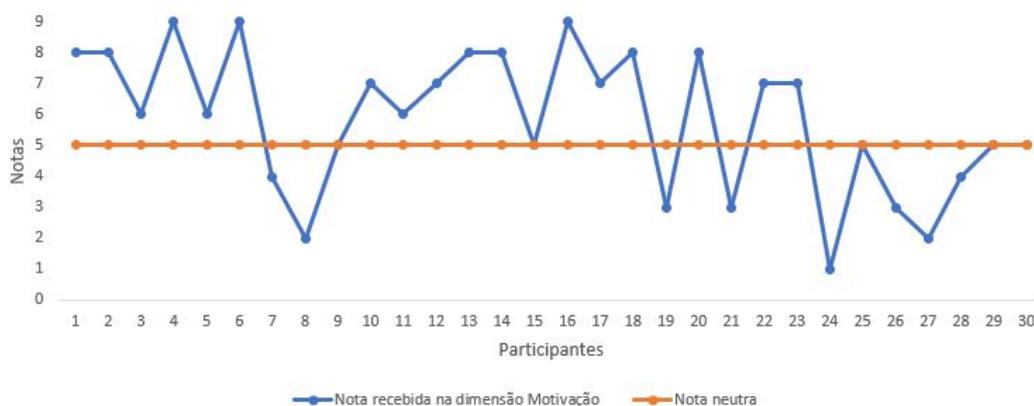
Figura 4.30 – Variação de respostas na dimensão satisfação.



Fonte: Elaborado pela autora.

A dimensão Motivação obteve a pontuação mais baixa entre as três, com média de 5.8 pontos. Conforme apresentado na Figura 4.31, há grande variação entre as respostas dos participantes. Dessa forma, podemos concluir que, embora essa dimensão tenha tido em média uma avaliação um pouco superior à neutra, a interface consegue reproduzir bons sentimentos nos usuários. No entanto, ainda há pontos de melhorias a serem trabalhados em relação à motivação dos alunos ao utilizar a ferramenta.

Figura 4.31 – Variação de respostas na dimensão motivação.



Fonte: Elaborado pela autora.

Já a dimensão Sentimento de controle também obteve uma pontuação positiva, com média de 7,8 pontos. Conforme apresentado na Figura 4.32, apenas o participante 26 atribuiu uma nota neutra nessa dimensão. As demais respostas se classificaram como positivas, sendo iguais ou superiores a 6, ocupando toda parte superior do gráfico.

Figura 4.32 – Variação de respostas na dimensão sentimento de controle.



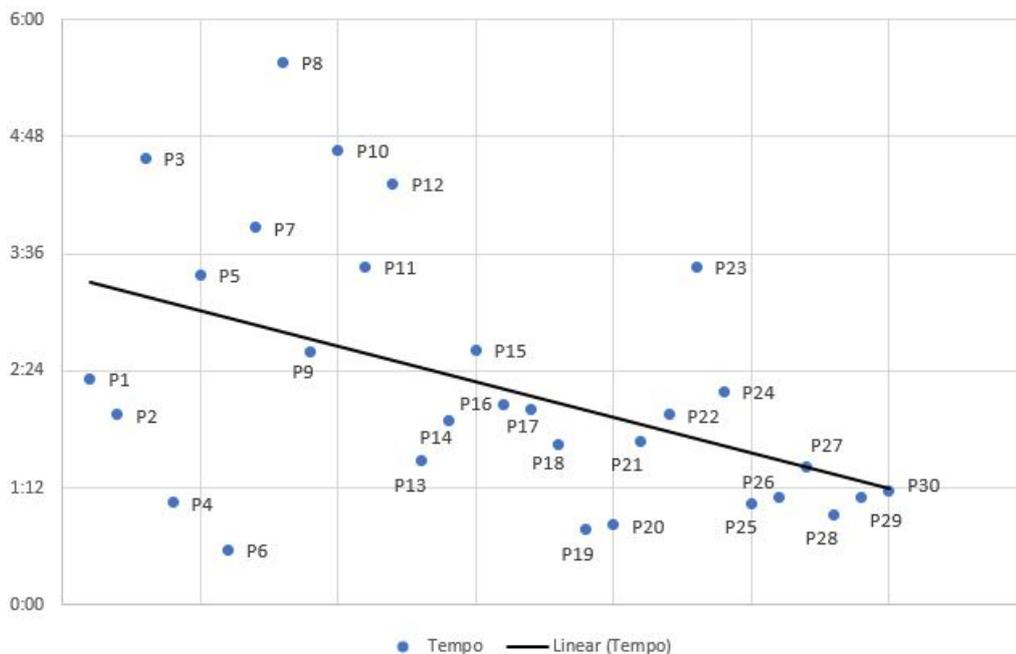
Fonte: Elaborado pela autora.

A partir da análise das três dimensões do questionário SAM, é possível concluir que a interface do protótipo proposto para o *opCoders Judge* alcançou uma avaliação positiva, com média de 7.2 pontos. No entanto, apesar da dimensão Motivação ter recebido uma avaliação superior a neutra, quando comparado com as outras duas dimensões, percebemos que ainda há pontos de melhoria a serem trabalhados na interface, evidenciando a necessidade da realização de ajustes que estimulem a motivação dos usuários durante o uso da ferramenta.

4.6 Tempo de execução da tarefa

Antes de iniciar os testes, todos os participantes da pesquisa foram orientados sobre a cronometragem do tempo de execução da tarefa. Essa métrica visa avaliar a facilidade de aprendizagem da ferramenta através do tempo que o usuário irá gastar para concluí-la. O tempo mais rápido para conclusão da tarefa foi 34 segundos (participante 6), e o maior tempo foi 5 minutos e 34 segundos (participante 8). Em média, os participantes levaram 2 minutos e 15 segundos para finalizar a tarefa. A Figura 4.33 apresenta o tempo de todos os participantes, de modo que os pontos mais distantes da linha correspondem aos tempos mais discrepantes. Apenas 8 participantes concluíram a tarefa em mais de 3 minutos. Assim, podemos concluir que o tempo de execução da tarefa foi consideravelmente baixo.

Figura 4.33 – Tempo de execução da tarefa por participante.



Fonte: Elaborado pela autora.

4.7 Discussão

A hipótese I deste trabalho buscou avaliar se a interface da ferramenta é de fácil aprendizagem, para isso foram utilizadas como referências os resultados obtidos através das afirmações 2, 3, 4, 7 e 10 do SUS, além do tempo gasto para conclusão da tarefa. Dentre as afirmações positivas (3 e 7), ambas receberam em média a melhor pontuação do formulário, como nota aproximada de 4.6 por participante. Para a afirmação 3 o desvio padrão das respostas é de 0.66. Já para a afirmação 7 o desvio padrão é ainda menor, com 0.60. Assim, como obtivemos desvios próximos a zero para essas afirmações, podemos afirmar que o conjunto de respostas é bastante homogêneo. Além disso, as duas afirmações possuem respostas com moda igual a 5, que representa a maior nota

possível a ser atribuída a uma afirmação. Para as afirmações negativas dessa hipótese (2, 4 e 10), em média, as três receberam notas baixas, o que significa que, em sua maioria, os participantes não concordam com as afirmações. As respostas para essas declarações, em média, variaram entre 1.4 e 1.6. A moda para as respostas dessas afirmações é igual a 1, que representa a menor nota possível a ser atribuída a uma afirmação.

Outra métrica utilizada para avaliar a facilidade de aprendizagem da ferramenta é o tempo gasto para conclusão da tarefa. Nessa questão o resultado alcançado é satisfatório, visto que 57% dos participantes gastaram até 2 minutos para concluir a tarefa. Assim, a partir dos resultados apresentados, podemos concluir que a hipótese I desta pesquisa está validada, o que significa que, de fato, a ferramenta é de fácil aprendizagem.

A hipótese II deste trabalho buscou avaliar se os usuários conseguem concluir as suas tarefas de forma satisfatória dentro da ferramenta. Com esse propósito, foram avaliadas as afirmações 1, 4, 5, 6, 8 e 9 do SUS, além do resultado do questionário SAM. Entre as afirmações positivas (1, 5 e 9), em média, as afirmações variaram entre 4.2 e 4.4, apontando que, em sua maioria os participantes concordam com as afirmações. Para essas afirmações os desvios padrões são, respectivamente, 0.77, 0.80 e 0.82. Ademais, as afirmações 1 e 5 possuem respostas com moda igual a 5. No entanto, a afirmação 9 que simboliza a confiança do usuário ao utilizar o sistema, apresentou duas modas, sendo elas 4 e 5, caracterizando-se como bimodal. Embora o resultado da afirmação 9 seja satisfatório, percebe-se há pontos a serem melhorados nesse aspecto. Para as afirmações negativas (4, 6 e 8), a média das respostas varia entre 1.4 e 1.6. Com moda igual a 1 nas três afirmações.

Com o questionário SAM, foi possível confirmar a satisfação dos participantes em relação à interface. Embora a dimensão “Motivação” apresente uma variação maior nas respostas, a média dos resultados ainda foi positiva. Já nas dimensões “Satisfação” e “Sentimento de controle” os resultados em sua maioria foram positivos, obtendo apenas uma avaliação neutra em cada uma delas. Dessa forma, com base nos resultados apresentados podemos validar a hipótese II desta pesquisa, ou seja, os usuários conseguem concluir as suas tarefas de forma satisfatória.

Por fim, a hipótese III deste trabalho buscou avaliar se a interface avaliada apresenta falhas de usabilidade. Conforme apresentado na Subseção 4.1, foram identificados 13 problemas de usabilidade na interface. Logo, também podemos validar a hipótese III deste trabalho.

5 Considerações Finais

Nesta seção serão apresentadas as considerações finais deste trabalho. A Subseção 5.1 discorre a respeito das conclusões obtidas a partir das avaliações realizadas na interface. Já a Subseção 5.2 apresenta os possíveis trabalhos a serem realizados com base nos resultados apresentados neste projeto.

5.1 Conclusão

Usabilidade tem sido um assunto cada vez mais recorrente entre as organizações, uma vez que seu objetivo é avaliar a qualidade de uma interface, a fim de proporcionar ao usuário uma boa experiência de uso. Para isso, a análise do produto é feita a partir dos métodos de avaliação de usabilidade existentes na literatura.

Como a aplicação da tecnologia no ambiente educacional tem se tornado cada vez mais comum, é notória a relevância da utilização de ferramentas online no processo de ensino aprendido. Dessa forma, é de fundamental importância avaliar se tais ferramentas apresentam uma boa usabilidade, possibilitando que seus usuários tenham uma experiência satisfatória ao utilizar suas interfaces, e assim contribuir para o processo de aquisição de conhecimento no contexto em que estiverem inseridas.

Nesse sentido, o presente trabalho avaliou a interface do *opCoders Judge* através de uma avaliação por inspeção e por meio da aplicação de testes com alunos matriculados em uma disciplina introdutória de programação de computadores.

Durante a avaliação por inspeção foram encontrados problemas na interface da ferramenta que, segundo [NIELSEN \(1994c\)](#) podem ser classificados como catastróficos. Com o objetivo de eliminar tais problemas, um protótipo da interface foi construído e testado, visando garantir que os problemas identificados foram solucionados. O resultado do SUS apresentou uma pontuação de 86.08 pontos, resultado que indica que a interface proposta possuiu uma usabilidade excelente. Além disso, o resultado do questionário SAM foi positivo na três dimensões avaliadas.

Diante de tais resultados, podemos concluir que a interface proposta oferece uma boa usabilidade aos seus usuários, de modo que será possível contribuir para o processo de ensino aprendido de programação introdutória.

5.2 Trabalhos Futuros

Como trabalho futuro, recomenda-se que os testes de usabilidade descritos neste trabalho sejam realizados com alunos de uma das turmas da disciplina de Programação de Computadores I

(BCC701) na primeira semana de aulas do próximo período letivo, visto que este é o público que utilizada a ferramenta aqui apresentada. Possibilitando alcançar resultados ainda mais precisos sobre a usabilidade da interface, principalmente no aspecto da facilidade de aprendizagem, visto que o teste aconteceria antes do primeiro contato do aluno com a ferramenta. Além disso, recomenda-se que a interface aqui proposta seja implementada e disponibilizada aos alunos.

Referências

- ABNT, A.; DEFESA, E.; GLOBAL, N.; DEFESA, E.; ZUCCARO, P.; DEFESA, E. Requisitos ergonômicos para trabalho de escritórios com computadores parte 11—orientações sobre usabilidade. *AAB d. N. Técnicas (Ed.)*. Brasil: ABNT—Associação Brasileira de Normas Técnicas, 2002.
- AGUIRRE, A. R.; CUNHA, S. M. da; DELUCHI, M.; GONÇALVES, R.; BIZARRO, L. Aplicação da escala sam na seleção de imagens de alimentos saudáveis e não saudáveis para utilização em tarefas experimentais. *Ciências & Cognição*, Ciências e Cognição, v. 24, n. 2, p. 245–264, 2019.
- ALMEIDA, A. A. d. Usabilidade da ferramenta moodle: uma análise da experiência do usuário no ambiente aprender. 2021.
- BARBOSA, S.; SILVA, B. *Interação humano-computador*. [S.l.]: Elsevier Brasil, 2010.
- BASTIEN, J. C.; SCAPIN, D. Ergonomics criteria for the evaluation of human-computer interfaces: Relatório de pesquisa nº 156. *INRIA-Institut National de Recherche en Informatique et en Automatique*, 1993.
- BELTRAME, F. S.; LOPES, G. A. W. Sistema de correção automática de código-fonte. 2018.
- BENYON, D. Interação humano-computador. *Tradução de Heloisa Coimbra de Souza*. 2a. ed. Sao Paulo: Person Prentice Hall, p. 464, 2011.
- BOSSE, Y.; GEROSA, M. A. Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: um estudo preliminar. In: SBC. *Anais do XXIII Workshop sobre Educação em Computação*. [S.l.], 2015. p. 426–435.
- BRASIL, C. N. d. S. *RESOLUÇÃO Nº 466, DE 12 DE DEZEMBRO DE 2012*. 2012. Disponível em: <<https://conselho.saude.gov.br/resolucoes/2012/Reso466.pdf>> Acesso em: 12 outubro 2022.
- BRITO, P.; FORTES, R. O uso de corretores automáticos para o ensino de programação de computadores para alunos de engenharia. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2019. v. 30, n. 1, p. 449.
- BRITO, P. S. S. O uso de ferramentas computacionais para o ensino de programação para alunos de engenharia. 2019.
- BROOKE, J. et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, London—, v. 189, n. 194, p. 4–7, 1996.
- COSTA, R. P. d. Conjunto de heurísticas de usabilidade para avaliação de aplicações móveis em smartphones. 2019.
- CRONBACH, L. J. Coefficient alpha and the internal structure of tests. *psychometrika*, Springer, v. 16, n. 3, p. 297–334, 1951.
- DECOM, U. *opCoders Judge*. 2022. Disponível em: <<https://bcc701.decom.ufop.br/tarefa>> Acesso em: 05 outubro 2022.

- FILHO, A. M. S. Avaliação de usabilidade: “separando o joio do trigo”. *Revista espaço acadêmico*, v. 10, n. 112, p. 10–14, 2010.
- FREITAS, L. M. d. et al. Análise de usabilidade do módulo laboratório virtual de programação do moodle. 2016.
- LANG, P. Behavioral treatment and bio-behavioral assessment: Computer applications. *Technology in mental health care delivery systems*, Ablex, p. 119–137, 1980.
- LANG, P.; BRADLEY, M. M. The international affective picture system (iaps) in the study of emotion and attention. *Handbook of emotion elicitation and assessment*, v. 29, p. 70–73, 2007.
- LEITE, M. C. D.; PAZ, F. J. A contribuição do system usability scale (sus) como ferramenta de avaliação da usabilidade. *ANAIS CONGREGA MIC-ISBN 978-65-86471-05-2*, p. 209, 2017.
- LIKERT, R. Escala de likert, método e evaluaciones sumarias. *Instituto de Investigación Social de Michigan. EE. UU*, 1932.
- MARTINS, V. F.; JUNQUEIRA, M. A.; ARAUJO, R. M. de. Ética da pesquisa em sistemas de informação: Por que e como submeter meu projeto ao comitê de ética? *Sociedade Brasileira de Computação*, 2021.
- MATERA, M.; RIZZO, F.; CARUGHI, G. T. Web usability: Principles and evaluation methods. In: *Web engineering*. [S.l.]: Springer, 2006. p. 143–180.
- NIELSEN, J. Enhancing the explanatory power of usability heuristics. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. [S.l.: s.n.], 1994. p. 152–158.
- NIELSEN, J. *Usability engineering*. [S.l.]: Morgan Kaufmann, 1994.
- NIELSEN, J. Usability inspection methods. In: *Conference companion on Human factors in computing systems*. [S.l.: s.n.], 1994. p. 413–414.
- NIELSEN, J. *Quantitative Studies: How Many Users to Test?* 2006. Disponível em: <<https://www.nngroup.com/articles/quantitative-studies-how-many-users/>> Acesso em: 08 outubro 2022.
- NIELSEN, J. *Usabilidade na web*. [S.l.]: Elsevier Brasil, 2007.
- NIELSEN, J. *10 Usability Heuristics for User Interface Design*. 2020. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/#poster>> Acesso em: 12 setembro 2022.
- NIELSEN, J.; LANDAUER, T. K. A mathematical model of the finding of usability problems. In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. [S.l.: s.n.], 1993. p. 206–213.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.: s.n.], 1990. p. 249–256.
- OLIVEIRA, P. S. F. d. *As 5 etapas do processo de design thinking*. 2022. Disponível em: <<https://www.sience.com.br/blog/design-thinking-paulo-oliveira/>> Acesso em: 01 março 2023.
- PIAZZA, L. *Prototipação: o que é, quais são os tipos e 10 FERRAMENTAS*. 2021. Disponível em: <<https://49educacao.com.br/mvp/prototipacao/>> Acesso em: 01 março 2023.

- PRESSMAN, R. S. *Engenharia de Software. Terceira edição*. [S.l.]: São Paulo: Pearson Makron Books, 1995.
- QUEIROZ, J. d. P. S. A importância do uso da tecnologia como ferramenta pedagógica na sala de aula. *CIET: EnPED*, 2018.
- SANTOS, R. P. dos; COSTA, H. A. X. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática. *INFOCOMP Journal of Computer Science*, v. 5, n. 1, p. 41–50, 2006.
- SAURO, J. *Measuring Usability with the System Usability Scale (SUS)*. 2011. Disponível em: <<https://measuringu.com/sus/>> Acesso em: 08 outubro 2022.
- SCHNEIDER, C. Moodle mobile (m-learning moodle) avaliação de usabilidade com o uso do sus-system usability scale. Universidade Federal de Santa Maria, 2017.
- SILVA, T. R. da; MEDEIROS, T.; MEDEIROS, H.; LOPES, R.; ARANHA, E. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, v. 23, n. 01, p. 182, 2015.
- SIX, J. M.; MACEFIELD, R. How to determine the right number of participants for usability studies. *San Francisco (CA): UXmatters*, 2016.
- SOUZA, D. M. O. Desenvolvimento e avaliação do protótipo do loop acadêmico: um software educacional para o auxílio no processo de ensino-aprendizagem de programação introdutória. Universidade Federal Rural do Semi-Árido, 2019.
- TENÓRIO, J. M.; COHRS, F. M.; SDEPANIAN, V. L.; PISA, I. T.; MARIN, H. de F. Desenvolvimento e avaliação de um protocolo eletrônico para atendimento e monitoramento do paciente com doença celíaca. *Revista de Informática teórica e aplicada*, v. 17, n. 2, p. 210–220, 2010.
- VALENTIM, N. M. C. et al. Antecipando a usabilidade nas fases iniciais do processo de desenvolvimento de software. Universidade Federal do Amazonas, 2017.
- WINCKLER, M.; PIMENTA, M. S. Avaliação de usabilidade de sites web. *Escola de Informática da SBC Sul (ERI 2002)*. Porto Alegre, v. 1, p. 85–137, 2002.

Apêndices

APÊNDICE A – Termo de Consentimento Livre e Esclarecido (TCLE)

Título da pesquisa: “Avaliação de usabilidade de ferramentas online no ensino de programação de computadores”.

Desenvolvido por: Prof. Dr. Saul Emanuel Delabrida Silva, Prof. Dr. Reinaldo Silva Fortes e Vivyann Fernandes Cedraz.

Prezado (a) participante,

Convidamos você a participar voluntariamente do estudo “**Avaliação de usabilidade de ferramentas online no ensino de programação de computadores**”, elaborado pelos professores Saul Emanuel Delabrida Silva e Reinaldo Silva Fortes, e pela aluna Vivyann Fernandes Cedraz.

O presente trabalho tem como objetivo geral avaliar a usabilidade da interface configurada do Moodle e da ferramenta de correção automática de código-fonte, OP Coders Judge.

O convite a sua participação se deve ao fato de que a avaliação das ferramentas devem ser feitas por alunos do ensino superior. Os resultados dos testes serão usados pelos pesquisadores para avaliar a usabilidade da interface dos sistemas.

Sua participação é voluntária, ou seja, não é obrigatória, e você tem plena autonomia para decidir se quer ou não participar, bem como suspender sua participação a qualquer momento, mesmo tendo iniciado as atividades. Você não será penalizado de nenhuma maneira caso decida não consentir sua participação, ou desistir da mesma. Contudo, ela é muito importante para conclusão da pesquisa.

Para participar você deverá fazer acesso à plataforma Moodle e ao OP Coders Judge por meio de computador com acesso à internet e realizar o conjunto de tarefas proposto. As tarefas são:

- Consultar material da aula 01;
- Abrir o Moodle da disciplina de Programação, responder a Atividade 01 e submeter;
- Consultar o resultado da correção da Atividade 01;
- Enviar uma dúvida ao professor da disciplina através do Moodle, de modo que todos os colegas de classe consigam acompanhar a discussão;
- Acessar o diário de classe e conferir notas;
- Acessar a bibliografia da disciplina;

- Identificar qual a data da primeira prova.

Ao final de realização das tarefas, devem ser preenchidos 2 formulários sobre a avaliação de usabilidade da interface configurada do Moodle e 2 formulários sobre a avaliação de usabilidade do sistema OP Coders Judge. Estes questionários fazem parte do objeto de pesquisa sobre a percepção dos participantes sobre as interfaces.

A pesquisa a ser desenvolvida tem aprovação do Comitê de Ética, segundo a Resolução CNS 510/2016, obedecendo às diretrizes e normas reguladoras de pesquisas que envolvem seres humanos. Essa Resolução abrange os cinco pilares básicos da bioética que são: autonomia, não maleficência, beneficência, justiça e equidade, visando assegurar os direitos e deveres sobre a comunidade científica, aos participantes da pesquisa e ao Estado.

Esta pesquisa apresenta riscos mínimos aos participantes já que se trata de avaliação de usabilidade. Para mitigar possíveis quebras de sigilo garante-se que apenas os pesquisadores terão acesso ao banco de dados. Se o(a) Sr(a) sentir constrangimento ao responder alguma pergunta, terá liberdade para não responder ou para interromper o preenchimento do questionário a qualquer momento. Em caso de desistência, os dados serão apagados do banco de dados. Manteremos apenas o registro da informação de desistência anonimizado para fins estatísticos como métrica da pesquisa.

Sua participação no estudo não deve acarretar despesas ou custos. Haverá garantia de ressarcimento se fizer necessário. Se houver algum dano, comprovadamente decorrente da presente pesquisa, você terá direito à indenização, por meio das vias judiciais, como dispõem o Código Civil, o Código de Processo Civil, e na Resolução nº 510/2016, do Conselho Nacional de Saúde.

Os benefícios e vantagens em participar é que esta pesquisa poderá implicar em melhorias na interface das ferramentas e possibilitar a boa experiência dos usuários.

Todos os documentos relativos à pesquisa serão guardados em local restrito pelo prazo de 5 (cinco) anos em custódia dos pesquisadores. Com o fim deste prazo todos os documentos da pesquisa serão descartados.

Como a identificação do participante da pesquisa é indispensável pela característica analítica do projeto, informamos que os dados coletados serão analisados de forma anônima para fins de avaliação e divulgação científica. Os dados poderão ser apagados se requisitado pelo participante da pesquisa, mesmo após o término das atividades.

Ressaltamos que a pesquisa está em conformidade com a Resolução 510/2012 do CNS e Ofício Circular nº 2/2021/CONEP/SECNS/MS.

Os resultados gerais poderão ser divulgados em palestras dirigidas ao público participante, artigos científicos e monografias/dissertações/teses. Os resultados de forma individual serão repassados aos participantes, a equipe de pesquisadores responsáveis fica à disposição para

eventuais esclarecimentos.

Caso tenha alguma dúvida sobre a pesquisa, durante a sua participação ou posteriormente, o(a) Sr(a) poderá entrar em contato conosco por meio dos contatos que estão explicitados neste Termo.

Em caso de dúvidas sobre o estudo, você poderá entrar em contato com os pesquisadores: Reinaldo Silva Fortes (reifortes@ufop.edu.br), Saul Emanuel Delabrida Silva (e-mail: saul.delabrida@ufop.edu.br) ou Vivyann Fernandes Cedraz (vivyann.cedraz@aluno.ufop.edu.br).

Em caso de denúncias ou reclamações sobre sua participação e sobre questões éticas do estudo, você pode entrar em contato com a secretaria do **Comitê de Ética em Pesquisa (CEP) da Universidade Federal Ouro Preto**. Endereço: Centro de Convergência, Campos Universitário, UFOP. Telefone: (31) 3559-1368. Email: cep.propp@ufop.edu.br.

O Comitê de Ética em Pesquisa envolvendo seres humanos (CEP) tem como finalidade defender os interesses dos participantes da pesquisa em sua integridade e dignidade e contribuir no desenvolvimento da pesquisa dentro de padrões éticos, sendo responsável pela avaliação e acompanhamento dos aspectos éticos de todas as pesquisas envolvendo seres humanos.

Este Termo de Consentimento Livre e Esclarecido deverá ser assinado pelo pesquisador responsável e pelo participante do estudo e rubricado em todas as folhas da via. É muito importante que você guarde em seus arquivos uma cópia deste documento

Consentimento Livre e Esclarecido

Após receber os esclarecimentos sobre os objetivos, importância e o modo como os dados serão coletados nessa pesquisa, além de conhecer os riscos, desconfortos e benefícios que ela trará para mim e ter ficado ciente de todos os meus direitos, concordo em participar da pesquisa **Avaliação de usabilidade de ferramentas online no ensio de programação de computadores.**

CONSENTIMENTO

Eu _____, após ter sido suficientemente esclarecido (a) pelo pesquisador sobre a realização desta pesquisa, como está escrito neste Termo, declaro que consinto em participar da pesquisa por livre e espontânea vontade.

Data: ___/___/___

Assinatura: _____

Pesquisador Responsável - Assinatura: _____

Prof. Dr. Reinaldo Silva Fortes

APÊNDICE B – Problemas encontrados na interface a partir da avaliação por inspeção

Resultado Avaliação Heurística					
Nº	Problema encontrado	Heurística(s) violada(s)	Classificação do problema	Proposta de solução	Local
1	Na descrição da tarefa não informa aos usuários quais são os tipos de arquivos permitidos para envio. Como por exemplo arquivos com extensão.py ou .cpp.	H5: Prevenção de erros	Problema grave	Detalhar no início das atividades os tipos de extensão de código que são aceitos ou colocar uma observação próximo a caixa de seleção do arquivo.	Tela de envio das tarefas
2	Ao tentar submeter um código que não tenha a extensão .py a ferramenta não exibe uma mensagem de erro que informe ao usuário o que está acontecendo e ele não consegue submeter o exercício.	H1: Visibilidade do estado do sistema. H9: Ajuda ao usuário a reconhecer, diagnosticar e recuperar erros.	Problema catastrófico	Exibir uma caixa de diálogo que informe ao usuário que ele não submeteu o arquivo com a extensão aceita pela ferramenta oferecendo uma solução para o problema.	Tela de envio das tarefas
3	A descrição de como utilizar a ferramenta se encontra apenas na tarefa "Testando o corretor automático", por serem informações importantes para a boa utilização da ferramenta deveriam estar de forma visível.	H6: Reconhecimento em vez de memorização.	Problema catastrófico	Disponibilizar na página inicial as principais informações para utilização da ferramenta. Criar uma opção no menu que forneça essas informações.	Tela de envio das tarefas

Resultado Avaliação Heurística					
Nº	Problema encontrado	Heurística(s) violada(s)	Classificação do problema	Proposta de solução	Local
4	A descrição da tarefa possui uma grande quantidade de informações e diminui a visibilidade das instruções mais relevantes.	H8: Projeto estético minimalista.	Problema grave	Deixar na janela das atividades apenas as informações fundamentais para realização daquela tarefa específica.	Tela de envio das tarefas
5	O botão "Escolher arquivo" na cor cinza não dá o destaque necessário a ele.	H5: Prevenção de erros.	Problema cosmético	Utilizar uma cor da paleta de cores da ferramenta para dar destaque ao botão.	Tela de envio das tarefas
6	Na página das tarefas, ao clicar em uma questão para ler o enunciado a descrição da mesma fica oculta, tirando do usuário o controle sobre o que ele deseja visualizar.	H7: Flexibilidade e eficiência de uso.	Problema médio	Inserir um botão para dar ao usuário a opção de manter as abas fixadas/ abertas na tela ou não.	Tela de envio das tarefas
7	As frases com cores diferentes na descrição das questões passam a impressão de que são clicáveis.	H4: Consistência e padrões.	Problema menor	Destacar os textos com negrito ou com uma fonte de tamanho maior.	Tela de envio das tarefas
8	Ao abrir a tela para visualizar a correção do código não tem um botão que permita o usuário retornar a tela anterior.	H7: Flexibilidade e eficiência de uso.	Problema médio	Inserir atalho de retorno a tela da tarefa para facilitar a interação do usuário.	Tela de envio das tarefas

Resultado Avaliação Heurística					
Nº	Problema encontrado	Heurística(s) violada(s)	Classificação do problema	Proposta de solução	Local
9	Ao enviar uma tarefa para correção o status dela fica "Pendente". O uso dessa expressão pode confundir o usuário, dando a entender que a tarefa não foi enviada.	H4: Consistência e padrões.	Problema grave	Trocar o termo "Pendente" por "Aguarde correção".	Tela de envio das tarefas
10	Para que o aluno consiga visualizar a tarefa corrigida é necessário atualizar a página. Ao clicar para visualizar a correção outra página no navegador é aberta, saindo da que o usuário estava.	H2: Correspondência entre o sistema e o mundo real.	Problema catastrófico	Implementar funcionalidade que mostre ao usuário assim que o código for corrigido, mantendo a visualização da correção na mesma página do navegador.	Tela de envio das tarefas
11	Ao realizar a alteração da senha a ferramenta não oculta os caracteres digitados e não oferece o botão para exibir ou não os caracteres na tela. Dessa forma, ao realizar essa ação, se o usuário voltar a tela anterior ele consegue visualizar a senha digitada, tornando a ferramenta pouco segura.	H4: Consistência e padrões.	Problema médio	Implementar uma funcionalidade que esconda os caracteres digitados e mude para um campo de senha.	Tela de perfil

Resultado Avaliação Heurística					
Nº	Problema encontrado	Heurística(s) violada(s)	Classificação do problema	Proposta de solução	Local
12	Como o cadastro na ferramenta é feito de forma automática, pode acontecer do usuário não lembrar que as credenciais de login foram enviadas por email e não conseguir acessar a ferramenta.	H4: Consistência e padrões.	Problema catastrófico	Colocar um aviso na tela inicial informando que as credenciais foram enviadas para o email institucional do aluno. Além disso, também pode ser implementada uma segunda opção de login a partir da conta institucional do Google.	Tela de perfil
13	Na ferramenta não há nenhum menu que forneça ao usuário informações básicas para que ele consiga utilizar o sistema e concluir sua tarefa de forma satisfatória.	H10: Ajuda e documentação	Problema catastrófico	Inserir no menu a opção "Ajuda" com informações básicas sobre como utilizar a ferramenta.	Tela inicial

APÊNDICE C – Questionário de Perfil

O presente formulário tem o objetivo de conhecer o perfil dos participantes da pesquisa "Avaliação de Usabilidade de Ferramentas Online no Ensino de Programação de Computadores".

1) Qual o seu gênero?

- a. Feminino
- b. Masculino
- c. Prefiro não dizer
- d. Outros

2) Qual o seu curso?

3) Em qual período da graduação você está? (Informe apenas o número. Ex. 1; 2; 3 etc.)

4) Já teve contato com o Moodle antes?

- a. Sim
- b. Não

5) Já usou algum corretor automático de código-fonte antes?

- a. Sim
- b. Não

6) Gosta de programação?

- a. Sim
- b. Não

7) Já programava antes de iniciar a disciplina de introdução a programação?

a. Sim

b. Não

8) Está cursando a disciplina de introdução a programação pela primeira vez?

a. Sim

b. Não

9) O que você achou da interface da ferramenta?
