



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

EDERSON JOSÉ ANDRADE MEDEIROS

**DESENVOLVIMENTO DE UM SISTEMA *WEB* PARA AUXÍLIO DO PROCESSO DE  
ENSINO-APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO APLICADO AO NOVO  
ENSINO MÉDIO**

EDERSON JOSÉ ANDRADE MEDEIROS

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA AUXÍLIO DO PROCESSO DE  
ENSINO-APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO APLICADO AO NOVO  
ENSINO MÉDIO**

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

Orientador: Prof. Dr. Elton Rafael Alves.

Belém – Pará – Brasil  
2024

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD Sistema de Bibliotecas da Universidade Federal do Pará  
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

---

M488d Medeiros, Ederson José Andrade.

DESENVOLVIMENTO DE UM SISTEMA WEB PARA  
AUXÍLIO DO PROCESSO DE ENSINO-  
APRENDIZAGEM LÓGICA DE PROGRAMAÇÃO  
APLICADO AO NOVO

ENSINO MÉDIO / Ederson José Andrade Medeiros. — 2024.

130 f. : il. color.

Orientador(a): Prof. Dr. Elton Rafael Alves

Dissertação (Mestrado) - Universidade Federal do  
Pará, Núcleo de Desenvolvimento Amazônico em Engenharia,  
Mestrado Profissional em Computação Aplicada, Tucuruí, 2024.

1. Sistema Web. 2. Lógica de Programação. 3.  
Novo Ensino Médio. 4. Educação. 5. Tecnologia  
Educativa.

I. Título.

CDD 001.64404

---

EDERSON JOSÉ ANDRADE MEDEIROS

## DESENVOLVIMENTO DE UM SISTEMA WEB PARA AUXÍLIO DO PROCESSO DE ENSINO-APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO APLICADO AO NOVO ENSINO MÉDIO

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

Orientador: Prof. Dr. Elton Rafael Alves.

Aprovada em \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

BANCA EXAMINADORA:



Documento assinado digitalmente

ELTON RAFAEL ALVES

Data: 19/09/2024 09:55:35-0300

Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Elton Rafael Alves  
PPCA/NDAE/UFPA  
Orientador



Documento assinado digitalmente

OTAVIO NOURA TEIXEIRA

Data: 18/09/2024 20:09:53-0300

Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Otávio Noura Teixeira / UFPA  
PPCA/NDAE/UFPA  
Membro Interno



Documento assinado digitalmente

FABRICIO DE SOUZA FARIAS

Data: 04/09/2024 13:43:23-0300

Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Fabrício de Souza Farias  
PPCA/NDAE/UFPA  
Membro Interno



Documento assinado digitalmente

WARLEY MURICY VALENTE JUNIOR

Data: 04/09/2024 15:50:45-0300

Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Warley Muricy Valente Junior  
PPGCF/IGE/UNIFESSPA  
Membro Externo

## Dedicatória

Aos meus pais, **Francisco da Costa Medeiros** e **Maria de Jesus Andrade Medeiros**, pelo amor incondicional, apoio e encorajamento em todos os momentos da minha vida. Vocês são a base de todas as minhas conquistas.

Ao meu tio-padrinho **Areosnaldo Nogueira de Andrade**, *in memoriam*, professor e exemplo de educador, que foi um dos maiores responsáveis por eu ter chegado até aqui.

## **AGRADECIMENTOS**

A toda minha família: meu pai Francisco, minha mãe Maria de Jesus, minha irmã Elizabeth e minha sobrinha e afilhada Maysa, por serem minha motivação diária e por me ensinarem o verdadeiro significado de dedicação e perseverança.

A todos os membros da direção da instituição SENAI, especialmente ao Diretor e Coordenador do CEDAM, que sempre me apoiaram e acreditaram no meu trabalho.

Aos meus colegas de trabalho, que sempre estiveram ao meu lado, oferecendo palavras de incentivo, ajudando e compartilhando momentos de alegria e desafios.

Aos meus alunos (*Padawans*) e futuros colegas de profissão, que se dispuseram a participar dos testes e responder questionários, muitas vezes considerados chatos por eles próprios.

E, finalmente, ao meu orientador e professores, que com sua sabedoria, orientação e dedicação, contribuíram imensamente para a realização deste trabalho.

## Epígrafe

“Educação não transforma o mundo.  
Educação muda as pessoas. Pessoas  
transformam o mundo”.

(Paulo Freire)

“Passe adiante o que aprendeu”

(Mestre Yoda – O Retorno de *Jedi*-1983)

## RESUMO

As tecnologias educacionais têm desempenhado um papel fundamental no aprimoramento do processo de ensino-aprendizagem, proporcionando soluções inovadoras que contribuem para uma educação mais dinâmica e eficaz. Apesar do desenvolvimento de várias ferramentas voltadas para diferentes áreas do conhecimento, ainda existem lacunas em determinadas disciplinas, como a lógica de programação no contexto do Novo Ensino Médio. Este trabalho propõe a criação de um sistema web para o ensino e aprendizado de lógica de programação, voltado especificamente para alunos do Novo Ensino Médio. O sistema foi projetado para ser uma ferramenta eficiente e acessível, capaz de facilitar a compreensão de conceitos complexos de programação. Com base em um levantamento de requisitos educacionais e em testes de usabilidade, a plataforma foi desenvolvida para atender às necessidades dos alunos, proporcionando uma experiência de aprendizado intuitiva e interativa. Este estudo foi realizado na cidade de Belém do Pará. Os resultados obtidos em relação ao uso da aplicação pelos participantes, evidenciam nível considerável de satisfação (mais de 60%), possibilitando ainda que a mesma possa ser utilizada por qualquer aluno que queira aprender lógica de programação dentro e fora do ambiente escolar.

Palavras-chave: Educação. Lógica de Programação. Novo Ensino Médio. Sistema *Web*. Tecnologia Educacional. BNCC.

## **ABSTRACT**

*Educational technologies have played a crucial role in enhancing the teaching-learning process, offering innovative solutions that contribute to a more dynamic and effective education. Despite the development of various tools aimed at different areas of knowledge, gaps still exist in certain subjects, such as programming logic within the context of the New High School. This study proposes the creation of a web-based system for teaching and learning programming logic, specifically designed for New High School students. The system is intended to be an efficient and accessible tool, capable of facilitating the understanding of complex programming concepts. Based on an educational requirements survey and usability tests, the platform was developed to meet the needs of students, providing an intuitive and interactive learning experience. This study was conducted in the city of Belém do Pará. The results obtained regarding the use of the application by participants show a considerable level of satisfaction (over 60%), demonstrating that the application can be used by any student wishing to learn programming logic both inside and outside the school environment.*

**Keywords:** Education. Programming Logic. New High School. Web System. Educational Technology. BNCC.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Metodologia de desenvolvimento ICONIX .....	31
Figura 2 - Fases da Metodologia ICONIX .....	33
Figura 3 - Representação visual do Modelo MTV no framework Django, destacando as camadas.....	36
Figura 4 - Fluxo ORM: Interação entre Models, SQLite3 (banco de dados), Django Views, Templates e o Browser na arquitetura do Django .....	39
Figura 5 - Etapas do Mapeamento Sistemático.....	59
Figura 6 - Demonstrativo do processo de acontecimentos dos Testes. ....	65
Figura 7 - Diagrama de Caso de Uso - Visão Geral do Sistema. ....	71
Figura 8 - Modelo de domínio .....	72
Figura 9 - Diagrama de classe .....	75
Figura 10 - Estrutura das tabelas e os relacionamentos entre elas do Banco de Dados. ....	76
Figura 11 - Tela inicial do Sistema Web de Ensino de Lógica de Programação.....	80
Figura 12 - Tela inicial do sistema com informações para o aluno .....	81
Figura 13 - Tela de Cadastro: preencha os dados para começar.....	82
Figura 14 - Tela de Login para uma experiência personalizada. ....	83
Figura 15 - Visão geral (Dashboards) dos testes disponíveis e total de questões.....	84
Figura 16 - Janela que exibe a lista de testes: O aluno decide qual teste realizar.....	87
Figura 17 - Regras e Informações sobre o teste escolhido. ....	88
Figura 18 - Janela mostrando as questões dos testes .....	89
Figura 19 - Janela mostrando as questões dos testes e botão de envio de resposta.....	89
Figura 20 - Histórico de Testes: Revise e Compare Suas Realizações.....	90
Figura 21 - Janela de Estatísticas mostra os números e resultados dos testes.....	91
Figura 22 - Relatório do teste: Questões e respostas com o botão “Ver recomendações” ..	92
Figura 23 - Relatório do teste: Questões e respostas com o botão “Ver sugestões” .....	92
Figura 24 - Janela de sugestões com a capacidade de expressar sua opinião. Dê 'Like' para Aprovar e 'Dislike' para desaprovar.....	96
Figura 25 - Tela inicial do sistema com informações para o professor .....	98
Figura 26 - Tela de Cadastro: preencha os dados para começar.....	99
Figura 27 - Tela de Login para uma experiência personalizada. ....	100
Figura 28 - Dashboard do professor .....	101
Figura 29 - Gerenciamento de Cursos .....	102
Figura 30 - Janela de cadastro do curso .....	102
Figura 31 - Relatório com os cursos cadastrados pelo professor. ....	103
Figura 32 - Janela que mostra a gestão das questões.....	104
Figura 33 - Tela para inserção das questões. ....	105
Figura 34 - Tela com as perguntas disponíveis.....	106
Figura 35 - Tela para Adicionar Sugestão ou Ver Sugestão cadastrada. ....	107
Figura 36 - Tela de Cadastro das sugestões personalizadas.....	108
Figura 37 - Tela com as sugestões cadastradas pelo professor. ....	109
Figura 38 - Resultado em gráfico da primeira questão do questionário diagnostico.....	114
Figura 39 - Exemplo de questão utilizada no teste diagnóstico.....	117
Figura 40 - Número de Acertos por Questão para Cada Itinerário Formativo.....	118
Figura 41 - Distribuição das Médias de Acertos por Itinerário Formativo.....	118
Figura 42 - Fluxo de formação do Novo Ensino Médio.....	120
Figura 43 - Avaliação dos alunos em relação as sugestões personalizadas. ....	122

## LISTA DE TABELAS

Tabela 1 - Comparação das Características dos Sistemas de Ensino de Lógica de Programação. .....	18
Tabela 2 - Atributos associados ao conjunto de heurísticas de Nielsen. ....	41
Tabela 3 - Requisitos funcionais do Sistema Web de Ensino de Lógica de Programação. ....	69
Tabela 4 - Detalhamento do Caso de Uso “Realização de Testes (Quiz)” .....	84
Tabela 5 - Detalhamento das Sugestão de Vídeos Educacionais .....	93
Tabela 6 - Distribuição de alunos conforme itinerário formativo. ....	112
Tabela 7 - Matriz de Referência Curricular – Habilitação Técnica em Redes de Computadores. .....	116

## SUMÁRIO

1.	<b>INTRODUÇÃO</b> .....	14
1.1.	<b>Contextualização do Problema</b> .....	18
1.2.	<b>Justificativa</b> .....	20
1.3.	<b>Objetivos</b> .....	21
1.4.	<b>Estrutura do Trabalho</b> .....	21
2.	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	23
2.1.	<b>Interação Humano-Computador (IHC)</b> .....	23
2.2.	<b>Engenharia de Requisitos</b> .....	25
2.3.	<b>Modelagem UML</b> .....	28
2.4.	<b>Metodologia Ágil ICONIX</b> .....	30
2.5.	<b>Linguagem de Programação Python</b> .....	34
2.9.	<b>Ambiente de Desenvolvimento Integrado (IDE)</b> .....	39
2.10.	<b>Heurísticas de usabilidade para sistemas web</b> .....	40
2.11.	<b>Usabilidade</b> .....	42
3.	<b>O NOVO ENSINO MÉDIO</b> .....	44
3.1.	<b>O Ensino Médio</b> .....	44
3.2.	<b>Base Nacional Comum Curricular</b> .....	47
3.3.	<b>Microlearning e Nanolearning</b> .....	49
4.	<b>TRABALHOS RELACIONADOS</b> .....	52
5.	<b>MÉTODOS E MATERIAIS</b> .....	58
5.1.	<b>Descrição da Pesquisa</b> .....	58
5.2.	<b>Procedimentos Metodológicos</b> .....	60
5.2.1.	<b>Análise de requisitos</b> .....	60
5.2.2.	<b>Análise de viabilidade</b> .....	60
5.2.3.	<b>Modelagem do Sistema</b> .....	61
5.2.4.	<b>Escolha de Tecnologia</b> .....	62
5.2.5.	<b>Desenvolvimento da Ferramenta</b> .....	63
5.2.6.	<b>Teste</b> .....	65
6.	<b>APRESENTAÇÃO DO SOFTWARE</b> .....	68
6.1.	<b>Resultados da Modelagem utilizando metodologia ICONIX</b> .....	68
6.2.	<b>A Plataforma Web</b> .....	79
6.2.1.	<b>A Visão do Aluno</b> .....	80
6.2.2.	<b>A Visão do Professor</b> .....	97
7.	<b>RESULTADOS</b> .....	110
7.1.	<b>Caso de Estudo</b> .....	110
7.2.	<b>Avaliação dos Resultados</b> .....	113
7.2.1.	<b>Teste Diagnóstico</b> .....	115
7.2.2.	<b>Avaliação das sugestões personalizadas</b> .....	121
8.	<b>CONCLUSÃO</b> .....	124
9.	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	126

## 1. INTRODUÇÃO

O cenário atual do mercado de tecnologia no Brasil, segundo a Associação das Empresas de Tecnologia da Informação e Comunicação e de Tecnologias Digitais (Brasscom, 2021), aponta para uma demanda crescente por profissionais nas áreas de *software*, serviços de TIC <sup>1</sup>e *TI In-House*<sup>2</sup>. Esse crescimento é impulsionado por investimentos significativos em tecnologias de transformação digital, como *Cloud Computing*, Internet das Coisas, *Big Data/Analytics* e desenvolvimento de *software*, estimado em R\$ 413,5 bilhões nos próximos cinco anos. O relatório *The Future of Jobs do World Economic Forum* (2020) destaca a necessidade crescente de humanização nas empresas e processos, e identifica a Lógica de Programação como uma habilidade essencial para o profissional do futuro, alinhada com as demandas globais.

No entanto, a aprendizagem de Lógica de Programação representa um desafio significativo para muitos estudantes do Ensino Médio, o que pode levar ao desinteresse e, em alguns casos, ao abandono da disciplina. De acordo com De e Lima (2018), as dificuldades estão ligadas à falta de experiência com Lógica de Programação, à carência em interpretação de textos, matemática básica e inglês, além de outros problemas, podendo causar reprovação na disciplina cursada e até a evasão; e conforme discutido no trabalho de Mercedes Gómez-Albarrán (2005) sobre ferramentas de suporte ao ensino e aprendizagem de programação alguns desafios comuns enfrentados no ensino de programação incluem a complexidade inerente dos conceitos e estruturas de programação, dificuldades relacionadas à sintaxe e semântica das linguagens de programação e a complexidade dos ambientes de programação. Além disso, os alunos iniciantes podem ter dificuldades em selecionar exemplos relevantes ao usar ambientes baseados em exemplos. Diante dessa realidade, torna-se crucial desenvolver soluções inovadoras e personalizadas para auxiliar os alunos na compreensão da Lógica de Programação e, assim, aprimorar suas habilidades nessa área.

Uma solução promissora para esse desafio é o desenvolvimento de um

---

<sup>1</sup> Tecnologia da Informação e Comunicação

<sup>2</sup> Refere-se à prática em que as atividades de Tecnologia da Informação são conduzidas internamente pelos funcionários da empresa, em vez de serem terceirizadas para fornecedores externos.

Sistema *Web* para Ensino e Aprendizado de Lógica de Programação Aplicado ao Novo Ensino Médio. O propósito desse sistema é oferecer sugestões personalizadas e adaptativas para cada aluno, e contribuir para uma compreensão mais efetiva da Lógica de Programação e aumentar a motivação e o interesse dos alunos pela disciplina. A quantificação ou qualificação do nível de conhecimento dos alunos pode ser realizada por meio de avaliações objetivas, testes práticos ou exercícios específicos relacionados ao conteúdo de programação, e proporcionar resultados mensuráveis que contribuem para o desenvolvimento do aluno e possibilitam ao professor um planejamento mais democrático, que considera o nível de aprendizagem individual dos alunos.

Diversas soluções baseadas em software têm sido desenvolvidas para facilitar o ensino de programação. No entanto, muitos desses sistemas não consideram a personalização e adaptação às necessidades individuais dos alunos. A seguir são citadas algumas dessas ferramentas, seus pontos fortes e fracos.

BlueJ (Kölling et al., 2003) é um ambiente de desenvolvimento integrado (IDE) especificamente projetado para ensino de programação orientada a objetos em Java que inclui recursos como diagramas de classes interativos, detecção de erros de programação e testes de unidade. Além disso, o BlueJ é destacado por sua abordagem pedagógica e suas contribuições para o ensino de programação orientada a objetos, contudo apresenta uma limitação, em relação à eliminação de dois aspectos complicados da linguagem Java: o método *main* e a entrada/saída de console. Essa abordagem visa simplificar o início da programação para estudantes iniciantes, removendo a necessidade de lidar com conceitos mais avançados logo no início do aprendizado. No entanto, essa simplificação pode limitar a exposição dos alunos a certos conceitos importantes da linguagem Java. Essa discussão destaca a importância de equilibrar a simplificação do ambiente de programação com a necessidade de apresentar conceitos fundamentais aos alunos.

DrJava (Allen et al., 2002) é um ambiente de desenvolvimento Java criado na *Rice University*. Ele apresenta características como suporte à indentação automática, destaque de palavras-chave e correspondência de chaves, integração com compiladores Java, incluindo o *Generic Java* e o *framework* de testes *JUnit*, e suporte à geração de documentação *Javadoc*. Além disso, o DrJava utiliza o *DynamicJava*

para interpretar o código, detectando erros semelhantes aos que o Java *Virtual Machine* (JVM) encontraria ao executar o código correspondente. Essa menção destaca a relevância do DrJava como uma ferramenta de suporte ao ensino e aprendizagem de programação em Java.

As limitações do programa DrJava, incluem uma simplificação excessiva e a falta de preparação para conceitos avançados. A remoção de aspectos complicados da linguagem Java, como a necessidade do método *main* e a entrada/saída de console, pode limitar a exposição dos alunos a conceitos importantes, resultando em uma compreensão superficial da estrutura e funcionamento do Java. Além disso, ao eliminar a necessidade de lidar com certos conceitos desde o início, os alunos podem não estar adequadamente preparados para enfrentar tópicos mais complexos posteriormente, dificultando a transição para ambientes de programação mais avançados. Esses pontos fracos ressaltam a necessidade de um equilíbrio entre a simplificação do ambiente de programação e a apresentação de conceitos fundamentais essenciais para o aprendizado completo da linguagem.

Alice (Dann et al., 2005) é um Ambiente de programação 3D que permite aos alunos criar animações para contar histórias, jogar jogos interativos ou compartilhar vídeos, além de permitir aos alunos escrever programas usando uma linguagem semelhante ao Java, tornando o aprendizado mais envolvente e visual. Com um editor de arrastar e soltar, os alunos podem selecionar habitantes do mundo e programar seu comportamento, facilitando a criação de programas sem lidar com erros de sintaxe. As ações integradas permitem movimentos e alterações físicas nos objetos, ilustrando conceitos de programação de forma prática. Além disso, os alunos podem criar simulações interativas e não interativas, proporcionando uma ampla gama de aplicações e experimentações. No entanto, o Alice também apresenta limitações, como a proteção contra erros de sintaxe e a complexidade potencial do ambiente 3D. A proteção contra erros de sintaxe, embora vantajosa, pode limitar a compreensão dos alunos sobre a sintaxe real da linguagem Java, pois eles não são expostos a esses erros. Para alguns alunos, a interface e funcionalidades do ambiente 3D podem ser complexas e causar confusão ao invés de facilitar o aprendizado.

JHAVÉ (Naps et al., 2000): é um ambiente de visualização algorítmica projetado para ajudar os alunos a entenderem algoritmos através de animações interativas, ele

oferece um ambiente visual que permite aos alunos verem a execução de seus programas em tempo real, ajudando na compreensão dos conceitos de programação. O suporte a várias linguagens de programação proporciona flexibilidade para aprender diferentes paradigmas. A interatividade aumenta o engajamento e a motivação, enquanto o feedback imediato ajuda os alunos a entenderem melhor a relação entre o código escrito e sua execução. Por outro lado, as limitações do Jhavé incluem a complexidade do ambiente e a falta de suporte a conceitos avançados. A interface e as funcionalidades podem ser complexas para iniciantes, resultando em uma curva de aprendizado acentuada. Além disso, Jhavé pode não oferecer suporte suficiente para conceitos mais avançados de programação, limitando o progresso dos alunos à medida que se tornam mais experientes.

Apesar da eficácia dessas ferramentas em introduzir conceitos de programação, elas não abordam plenamente a personalização e a adaptação ao estilo de aprendizado individual dos alunos. Uma solução promissora para esse desafio é o desenvolvimento de um Sistema *Web* para Ensino e Aprendizado de Lógica de Programação Aplicado ao Novo Ensino Médio. O propósito desse sistema é oferecer sugestões personalizadas e adaptativas para cada aluno, e contribuir para uma compreensão mais efetiva da Lógica de Programação, além de aumentar a motivação e o interesse dos alunos pela disciplina. Diferente das outras ferramentas que levam rapidamente para a prática, o sistema proposto dá ênfase em conceitos fundamentais de programação, garantindo que os alunos compreendam a base teórica antes de avançarem para a aplicação prática.

A quantificação ou qualificação do nível de conhecimento dos alunos pode ser realizada por meio de avaliações objetivas, testes práticos ou exercícios específicos relacionados ao conteúdo de programação, proporcionando resultados mensuráveis que contribuem para o desenvolvimento do aluno e possibilitam ao professor um planejamento mais democrático, que considera o nível de aprendizagem individual dos alunos.

Além disso, a consideração dos estilos de aprendizado individuais é crucial, uma vez que as preferências variam. Ferramentas de avaliação, como a teoria de VARK (Visual, Auditivo, Leitura/Escrita, Cinestésico), podem ser aplicadas para definir essas preferências. Contudo, a diversificação de métodos de ensino é igualmente

essencial para atender às diversas preferências e garantir uma abordagem inclusiva. Essa perspectiva é reforçada por Lum, Bradley e Rasheed (2011), que destacam a diversidade de tipos de aprendizagens em sala de aula e a importância de identificar os estilos de aprendizado e adotar métodos de ensino diversos para atender a essa diversidade.

Quanto aos critérios para definir o progresso do estudante, no contexto da programação, isso pode ser medido por marcos alcançados em projetos, níveis de complexidade de problemas de programação resolvidos com sucesso, pontuações em avaliações formativas e somativas, participação em atividades práticas, entre outros. A avaliação personalizada desses critérios permitirá uma compreensão mais holística do progresso individual de cada aluno.

A seguir apresenta-se uma tabela comparativa destacando os diferenciais do sistema em relação aos existentes:

Tabela 1 - Comparação das Características dos Sistemas de Ensino de Lógica de Programação.

<b>Características</b>	<b>BlueJ</b>	<b>DrJava</b>	<b>Alice</b>	<b>JHAVÉ</b>	<b>Sistema Proposto</b>
Foco em Lógica de Programação	Parcial	Parcial	Não	Sim	Sim
Personalização do Ensino	Não	Não	Não	Não	Sim
Feedback Imediato	Não	Não	Não	Não	Sim
Foco nos Conceitos	Não	Não	Não	Não	Sim

Fonte: Autor

Diante das limitações das soluções atuais, o desenvolvimento de um sistema *web* que não apenas ensina Lógica de Programação, mas também personaliza a experiência de aprendizado de acordo com as necessidades individuais dos alunos, se apresenta como uma solução promissora. Este sistema visa aumentar a motivação e o interesse dos alunos pela disciplina, contribuindo para um aprendizado mais efetivo e significativo.

### **1.1. Contextualização do Problema**

A crescente diversidade de perfis e ritmos de aprendizado entre os alunos do ensino médio demanda estratégias educacionais inovadoras e personalizadas. No

entanto, a carência de abordagens eficazes para lidar com essa diversidade específica no contexto do ensino de Lógica de Programação cria um desafio significativo. A falta de personalização na oferta de materiais de apoio pode resultar em menor engajamento, satisfação e desempenho dos alunos.

As metodologias tradicionais de ensino de Lógica de Programação, chamadas de método magistral, frequentemente adotam abordagens centradas no professor, onde a transmissão de conhecimento ocorre predominantemente por meio de aulas expositivas, exercícios repetitivos e com uma falta de contextualização em relação ao mundo real. Essa abordagem tem sido criticada por seu potencial falta de envolvimento e motivação dos alunos, o que pode resultar em um desinteresse pela disciplina Lógica de Programação e um desempenho insatisfatório.

Segundo Borges (2000), o modo tradicional não consegue facilmente motivar os alunos a se interessar pela disciplina. Entre outras razões, isso ocorre, pois a importância de determinado conteúdo para sua formação não é clara.

As aulas expositivas, ao colocarem os alunos em uma posição passiva de receptores de informação, podem tornar o aprendizado desinteressante e pouco motivador. A dificuldade de acompanhamento, especialmente para alunos com diferentes estilos de aprendizagem, contribui para os desafios enfrentados nessa modalidade de ensino. Da mesma forma, os exercícios repetitivos, por sua natureza maçante e frustrante, podem falhar em promover uma compreensão profunda dos conceitos de Lógica de Programação, especialmente para aqueles que carecem de uma base teórica sólida.

A falta de contextualização, por sua vez, emerge como outro obstáculo, pois a apresentação abstrata dos conceitos de lógica de programação sem uma conexão clara com o mundo real dificulta a compreensão significativa. Os alunos necessitam compreender como esses conceitos podem ser aplicados em situações práticas para desenvolverem um entendimento profundo e valioso.

Em face desses desafios, torna-se evidente a necessidade de repensar e renovar as metodologias de ensino de Lógica de Programação. A transição de abordagens centradas no professor para modelos mais centrados no aluno é crucial. Métodos que promovam a participação ativa, a aplicação prática dos conceitos e a

contextualização no mundo real podem superar as limitações das metodologias tradicionais. A busca por estratégias pedagógicas mais eficazes, alinhadas com os princípios da aprendizagem significativa, é essencial para estimular o interesse, aumentar o desempenho dos alunos e fornecer uma base sólida para o entendimento e aplicação da lógica de programação.

O desenvolvimento de um sistema *web* destina-se a servir como um elo para a sugestão de materiais de apoio, como vídeos com base no desempenho dos alunos em testes de avaliação. Esta abordagem inovadora, alinhada a metodologias ágeis como *nanolearning*<sup>3</sup> e *microlearning*<sup>4</sup>, busca personalizar e aprimorar a experiência de aprendizado.

## 1.2. Justificativa

A habilidade mais crucial para um programador, segundo Sedgewick (2011), é a Lógica de Programação, essencial para desenvolver algoritmos eficientes e programas de alta qualidade. No entanto, a assimilação dessa habilidade requer prática contínua, especialmente para iniciantes na área. A introdução de um Sistema Web para Ensino e Aprendizado de Lógica de Programação visa aprimorar a experiência educacional dos alunos, especialmente no contexto do Novo Ensino Médio.

A Lei nº 13.415/2017, que alterou a Lei de Diretrizes e Bases da Educação Nacional, trouxe mudanças significativas, como a ampliação do tempo mínimo do estudante na escola e a organização curricular mais flexível (MÉDIO, [s.d.]). O Novo Ensino Médio, conforme definido pela lei, objetiva garantir educação de qualidade para todos os jovens brasileiros e aproximar as escolas da realidade dos estudantes contemporâneos. A estrutura curricular inclui a Base Nacional Comum Curricular (BNCC) e a implementação de itinerários formativos, oferecendo aos estudantes a oportunidade de escolher entre diferentes áreas de conhecimento e se aprofundar em formação técnica e profissional.

Diante dessas mudanças, a implementação de um sistema *web* que ofereça

---

<sup>3</sup> Refere-se a pequenas unidades de conhecimento extremamente curtas e específicas, geralmente apenas alguns segundos.

<sup>4</sup> É uma abordagem de aprendizado que oferece conteúdo educacional em pequenas e gerenciáveis porções, tipicamente variando de alguns minutos a até 15 minutos.

sugestões personalizadas e relevantes para o ensino de Lógica de Programação se torna uma necessidade premente. Cada aluno possui interesses e ritmos de aprendizagem únicos, e um sistema que se adapta a essas diferenças e fornece material educacional alinhado às preferências individuais promove uma compreensão mais profunda dos conceitos, aumentando o engajamento e a satisfação dos alunos.

Além disso, ao proporcionar um ambiente educacional interativo e estimulante, espera-se fortalecer o interesse dos alunos nas atividades educacionais. A análise do comportamento dos alunos permite insights valiosos sobre padrões de ensino e aprendizagem, preferências individuais e áreas de dificuldade, informando decisões educacionais mais fundamentadas e possibilitando melhorias contínuas no sistema.

Por meio desta dissertação, busca-se contribuir para o estado da arte em tecnologias educacionais, propondo uma abordagem eficaz para a personalização da experiência de aprendizado no Ensino Médio. A resolução de desafios específicos do domínio educacional, como a diversidade de interesses e ritmos de aprendizado, justifica a implementação deste sistema *web* inovador e centrado no aluno.

### **1.3. Objetivos**

#### **Geral**

Projetar e desenvolver uma ferramenta para testes e indicação de materiais didáticos para estudo de lógica de programação.

#### **Específicos**

- Avaliar a ferramenta em relação a usabilidade
- Aplicar de forma prática em ambiente educacional e coletar o desempenho de 30 alunos participantes.

### **1.4. Estrutura do Trabalho**

Com finalidade de atender os objetivos desta pesquisa, e expor o caminho que será percorrido no desenrolar deste documento, este trabalho é composto por sete capítulos, mais as referências bibliográficas. Sendo estruturados da seguinte forma:

1 – Introdução: Neste capítulo, apresentamos a pesquisa, destacando os objetivos, a relevância e as questões que serão abordadas;

2 – Fundamentação Teórica: Neste capítulo exploramos os principais conceitos, teorias e estudos relacionados ao tema da pesquisa. Destacam-se as contribuições existentes, lacunas identificadas e fundamentos teóricos;

3 – O Novo Ensino Médio: Este capítulo aborda as principais características e transformações do Novo Ensino Médio, destacando as inovações pedagógicas e os impactos no sistema educacional, com ênfase nas adaptações curriculares e metodológicas relevantes para a pesquisa;

4 – Trabalhos Relacionados: Este capítulo revisa pesquisas e estudos anteriores que abordam temas similares ao da pesquisa, destacando abordagens, metodologias e resultados relevantes, e identificando como o trabalho atual se posiciona em relação a esses estudos;

5 – Métodos e Materiais: Neste capítulo descrevemos os métodos e as ferramentas empregadas no desenvolvimento da pesquisa. Serão apresentadas as escolhas metodológicas e tecnológicas, justificando suas seleções;

6 – Apresentação do Software: Este capítulo detalha a implementação do sistema proposto, destacando seu funcionamento. Inclui também considerações sobre o uso do sistema;

7 – Resultados: Apresentamos os resultados obtidos durante a implementação e testes de usabilidade do sistema. Essa seção oferece *insights* sobre o desempenho e eficácia do sistema até o momento;

8 – Conclusão: O capítulo de conclusões sintetiza os principais achados, discute-se implicações práticas e teóricas, além de apontar direções para pesquisas futuras.

## 2. FUNDAMENTAÇÃO TEÓRICA

O presente capítulo aborda o referencial teórico deste trabalho com a finalidade de delimitar a área de abrangência do estudo, bem como conhecer os últimos estudos realizados na temática. Aborda ainda conceituação de Interação Humano-Computador, Engenharia de Requisitos, dentre outros aspectos relevantes ao tema.

### 2.1. Interação Humano-Computador (IHC)

A Interação Humano-Computador (IHC) é uma área interdisciplinar de estudo que se concentra na interação entre os usuários e os sistemas computacionais. A IHC busca compreender, projetar e avaliar interfaces que facilitem essa interação, tornando-a mais eficiente, efetiva e satisfatória. No contexto do desenvolvimento de sistemas web educacionais, a IHC desempenha um papel crucial na criação de interfaces que promovam um aprendizado intuitivo e envolvente para os alunos.

Os princípios de IHC são fundamentais para o desenvolvimento de interfaces de usuário que sejam fáceis de usar e acessíveis. Entre os princípios mais relevantes estão a usabilidade, a acessibilidade e o *feedback*. Usabilidade refere-se à facilidade com que os usuários podem aprender a usar uma interface, sua eficiência durante o uso e a satisfação que obtêm da interação. Jakob Nielsen (1994) define usabilidade com cinco atributos: facilidade de aprendizado, eficiência, facilidade de memorização, taxa de erros e satisfação do usuário. A acessibilidade é a capacidade do sistema de ser usado por pessoas com uma ampla gama de habilidades e deficiências, essencial para garantir que todos os alunos possam interagir com o sistema de maneira eficaz. O *feedback* é importante para fornecer aos usuários respostas claras e imediatas sobre suas ações, vital em um ambiente educacional onde os alunos precisam entender rapidamente o impacto de suas interações para aprender de forma eficiente.

A avaliação da interface é um componente essencial do processo de design de IHC. Existem várias metodologias para avaliar a eficácia de uma interface, incluindo a avaliação heurística, os testes de usabilidade e a análise de tarefas. A avaliação heurística é uma técnica de inspeção onde especialistas em IHC usam um conjunto de heurísticas para identificar problemas de usabilidade na interface (Nielsen, 1994). Os testes de usabilidade envolvem a observação de usuários reais enquanto eles interagem com o sistema, fornecendo *insights* valiosos sobre problemas de

usabilidade que podem não ser aparentes durante a fase de design. A análise de tarefas é um método que envolve a decomposição das tarefas que os usuários precisam realizar no sistema para identificar possíveis obstáculos e melhorar a eficiência da interação.

No contexto educacional, especialmente no ensino de lógica de programação, a IHC tem um impacto direto na eficácia do aprendizado. Interfaces bem projetadas podem facilitar a compreensão de conceitos complexos, aumentar o engajamento dos alunos e melhorar a retenção de conhecimento. Algumas práticas recomendadas incluem a criação de interfaces intuitivas, que são fáceis de navegar e compreender, ajudando os alunos a focar no conteúdo educacional, em vez de se perderem em funcionalidades complicadas. O suporte multimodal, utilizando diferentes formas de mídia (texto, vídeo, animações), pode atender a diferentes estilos de aprendizagem, tornando o processo de ensino mais inclusivo. O *feedback* imediato e personalizado em tempo real sobre o desempenho dos alunos e sugestões personalizadas, pode ajudar a corrigir erros rapidamente e direcionar o aprendizado de forma mais eficaz.

O sistema *web* proposto para o auxílio do ensino-aprendizagem de lógica de programação será desenvolvido com base nos princípios e metodologias de IHC mencionados. A interface será projetada para ser intuitiva, acessível e fornecer *feedback* imediato e relevante aos alunos. As funcionalidades incluirão um painel de controle personalizado, onde cada aluno terá acesso a um painel que adapta o conteúdo educacional de acordo com seu progresso e necessidades específicas. Os módulos interativos incluirão animações e vídeos que ilustram conceitos de lógica de programação de forma clara e envolvente. Ferramentas de avaliação contínua serão implementadas para que os alunos recebam avaliações regulares, e o sistema fornecerá *feedback* detalhado para ajudá-los a entender seus erros e melhorar continuamente.

Ao aplicar os princípios de IHC no desenvolvimento do sistema *web*, espera-se que a interface resultante não apenas facilite o aprendizado de lógica de programação, mas também aumente o engajamento e a satisfação dos alunos, alinhando-se aos objetivos do Novo Ensino Médio.

## 2.2. Engenharia de Requisitos

A engenharia de requisitos é uma disciplina essencial no desenvolvimento de sistemas de *software*, responsável por identificar, documentar e gerenciar os requisitos do sistema. Ela garante que o sistema final atenda às necessidades e expectativas dos stakeholders, proporcionando uma base sólida para todo o ciclo de vida do desenvolvimento do *software*. No contexto de um sistema web para o auxílio do ensino-aprendizagem de lógica de programação no Novo Ensino Médio, a engenharia de requisitos desempenha um papel crucial ao assegurar que o sistema seja eficaz, útil e alinhado com os objetivos educacionais (LAMSWEERDE, 2009).

A elicitação de requisitos é a fase inicial do processo, em que os requisitos são identificados através de técnicas como entrevistas, questionários, workshops e análise de documentos. Esta fase é crucial, pois erros ou omissões aqui podem levar a problemas significativos no desenvolvimento subsequente (SOMMERVILLE, 2015). No contexto educacional, a elicitação deve envolver uma ampla gama de stakeholders, incluindo professores, alunos, administradores escolares e especialistas em pedagogia. Este envolvimento garante que todas as perspectivas e necessidades sejam consideradas, resultando em um conjunto de requisitos mais robusto e representativo.

Após a elicitação, os requisitos coletados precisam ser analisados para garantir que sejam claros, completos, consistentes e viáveis. A análise de requisitos envolve a identificação de conflitos, a priorização de requisitos e a verificação de sua viabilidade técnica e econômica (PRESSMAN, 2014). Ferramentas como diagramas de casos de uso, modelagem de processos e protótipos são frequentemente utilizadas para ajudar na análise e na comunicação dos requisitos. A análise criteriosa dos requisitos permite identificar possíveis ambiguidades e resolver conflitos antes de avançar para a próxima fase.

A especificação é o processo de documentar os requisitos de maneira detalhada e clara, geralmente em um documento de especificação de requisitos de *software* (SRS). Este documento serve como uma referência oficial para desenvolvedores, testadores e outros stakeholders ao longo do ciclo de vida do projeto (PFAHL; HEINDL; WERNER, 2009). No desenvolvimento de um sistema web

educacional, a especificação deve incluir requisitos funcionais (o que o sistema deve fazer) e requisitos não funcionais (como o sistema deve se comportar), tais como usabilidade, desempenho, segurança e acessibilidade. Um SRS bem elaborado é fundamental para assegurar que todos os envolvidos tenham uma compreensão comum do que será desenvolvido.

A validação é o processo de garantir que os requisitos especificados atendam às necessidades dos stakeholders e estejam livres de erros e omissões. Técnicas como revisões de requisitos, prototipagem e testes de aceitação podem ser utilizadas para validar os requisitos (WEINBERG, 1989). A validação contínua com os stakeholders é crucial para assegurar que o sistema final seja útil e satisfatório. A validação eficaz evita retrabalho e garante que os requisitos realmente reflitam as necessidades do usuário final.

A gestão de requisitos envolve o controle e a rastreabilidade dos requisitos ao longo do ciclo de vida do projeto. Isso inclui a gestão de mudanças nos requisitos, a manutenção da consistência entre os requisitos e a documentação, e a comunicação contínua com os stakeholders (SOMMERVILLE, 2015). Ferramentas de gestão de requisitos, como softwares de rastreamento de requisitos, podem ajudar a manter um controle eficiente e atualizado dos requisitos. A gestão eficaz de requisitos é essencial para lidar com mudanças inevitáveis durante o desenvolvimento do sistema.

No desenvolvimento de um sistema *web* para o ensino de lógica de programação, a engenharia de requisitos deve considerar tanto os aspectos técnicos quanto os educacionais. Requisitos relacionados à interface do usuário, por exemplo, devem assegurar que o sistema seja intuitivo e fácil de usar para alunos com diferentes níveis de habilidade técnica (PRESSMAN, 2014). Requisitos pedagógicos devem garantir que o sistema suporte metodologias de ensino eficazes e ofereça recursos que promovam a compreensão e o engajamento dos alunos.

A personalização e a adaptabilidade do sistema são aspectos críticos a serem considerados. Requisitos que permitam a adaptação do conteúdo e das atividades às necessidades individuais dos alunos podem aumentar significativamente a eficácia do ensino e a satisfação dos alunos (LAMSWEERDE, 2009). Um sistema adaptativo pode, por exemplo, oferecer diferentes níveis de dificuldade nas atividades de

programação, fornecer *feedback* personalizado e recomendar materiais de estudo adicionais com base no desempenho individual.

As considerações pedagógicas são essenciais no desenvolvimento de um sistema educacional. O sistema deve ser capaz de integrar-se com as metodologias de ensino existentes e apoiar os professores na implementação de práticas pedagógicas inovadoras. Requisitos que garantam a integração com outros sistemas educacionais, como plataformas de gestão de aprendizagem (LMS), podem facilitar a adoção do sistema pelos professores e aumentar a sua eficácia (SOMMERVILLE, 2015).

A usabilidade e a acessibilidade são requisitos não funcionais importantes que devem ser considerados. O sistema deve ser fácil de usar e acessível a todos os alunos, incluindo aqueles com deficiências (PRESSMAN, 2014). Isso pode incluir a implementação de recursos como leitores de tela, navegação por teclado e conformidade com padrões de acessibilidade web. Uma boa usabilidade e acessibilidade garantem que todos os alunos possam beneficiar-se igualmente do sistema.

A segurança e a privacidade dos dados dos alunos são preocupações críticas em qualquer sistema educacional. Os requisitos de segurança devem garantir que os dados dos alunos sejam protegidos contra acessos não autorizados e violações de dados (WEINBERG, 1989). Políticas de privacidade claras e conformidade com regulamentações como a Lei Geral de Proteção de Dados (LGPD) são essenciais para proteger a privacidade dos alunos e manter a confiança dos usuários.

A engenharia de requisitos é, portanto, uma fase fundamental no desenvolvimento de um sistema web educacional eficaz. Uma abordagem sistemática e colaborativa para a elicitación, análise, especificação, validação e gestão de requisitos podem assegurar que o sistema final atenda às necessidades dos stakeholders e apoie os objetivos educacionais do Novo Ensino Médio. A compreensão profunda dos requisitos e a sua gestão eficiente ao longo do ciclo de vida do projeto são essenciais para o sucesso do sistema e para a satisfação dos usuários finais.

### 2.3. Modelagem UML

A Modelagem UML (*Unified Modeling Language*) é uma ferramenta essencial no desenvolvimento de sistemas de *software*, proporcionando uma linguagem padronizada para especificar, visualizar, construir e documentar os artefatos de um sistema. No contexto do desenvolvimento de um sistema *web* para auxílio do processo de ensino-aprendizagem de lógica de programação aplicado ao Novo Ensino Médio, a UML desempenha um papel crucial na garantia de que o sistema seja bem estruturado e atenda às necessidades dos stakeholders (BOOCH; RUMBAUGH; JACOBSON, 2005).

A UML oferece diversos tipos de diagramas que ajudam a capturar e comunicar diferentes aspectos do sistema. Os diagramas de casos de uso são frequentemente o ponto de partida na modelagem UML, descrevendo as interações entre os usuários (atores) e o sistema. Eles ajudam a identificar as funcionalidades principais do sistema e a entender como os usuários irão interagir com ele. No contexto educacional, isso pode incluir casos de uso como "Realizar Teste", "Realizar *Feedback*", "Acessar Sugestão de Estudo" e "Administrar Curso" (ARMERDORFER et al., 2012).

Os diagramas de classes são utilizados para modelar a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas. No desenvolvimento de um sistema *web* educacional, o diagrama de classes pode incluir entidades como "Aluno", "Professor", "Exercício", "Sugestão de Estudo" e "*Feedback*". Estes diagramas são essenciais para definir a estrutura de dados do sistema e as interações entre diferentes componentes (RUMBALL; POOLE, 2017).

Os diagramas de sequência são úteis para modelar a interação dinâmica entre os objetos ao longo do tempo. Eles mostram como os objetos se comunicam entre si para realizar uma funcionalidade específica. Por exemplo, um diagrama de sequência pode ilustrar o processo de um aluno submetendo um exercício de programação e recebendo *feedback* imediato do sistema. Estes diagramas ajudam a garantir que a lógica de negócios do sistema seja corretamente implementada (FOWLER, 2003).

Os diagramas de atividades são usados para modelar o fluxo de trabalho ou processos no sistema. Eles são particularmente úteis para descrever processos complexos que envolvem múltiplas etapas e decisões. No sistema educacional, um

diagrama de atividades pode representar o fluxo desde a inscrição do aluno em um curso até a conclusão de todas as atividades e avaliação final. Estes diagramas são valiosos para entender e otimizar os processos do sistema (BOOCH; RUMBAUGH; JACOBSON, 2005).

Os diagramas de estado descrevem os estados pelos quais um objeto passa durante seu ciclo de vida e os eventos que causam a transição entre esses estados. No contexto de um sistema educacional, um diagrama de estado pode ser usado para modelar o ciclo de vida de um exercício de programação, desde a criação e submissão até a avaliação e *feedback*. Estes diagramas são importantes para garantir que o comportamento do sistema seja previsível e gerenciável (ARMERDORFER et al., 2012).

Os diagramas de componentes e de implantação são utilizados para modelar a arquitetura física do sistema. Os diagramas de componentes mostram a organização e dependências entre os componentes de *software*, enquanto os diagramas de implantação mostram a distribuição física desses componentes em servidores e outros dispositivos de *hardware*. Para um sistema *web* educacional, esses diagramas são cruciais para planejar a infraestrutura necessária para suportar o sistema, garantindo escalabilidade, segurança e desempenho adequados (RUMBALL; POOLE, 2017).

A UML não só facilita a comunicação entre os membros da equipe de desenvolvimento, mas também com os stakeholders não técnicos. Os diagramas UML ajudam a criar uma visão compartilhada do sistema, minimizando mal-entendidos e erros de comunicação. A utilização adequada da UML ao longo do ciclo de vida do projeto pode melhorar significativamente a qualidade do *software* e a satisfação dos usuários finais (FOWLER, 2003).

A modelagem UML também suporta a documentação do sistema, o que é fundamental para a manutenção e evolução futura do *software*. A documentação detalhada baseada em UML proporciona uma referência clara para desenvolvedores, testadores e novos membros da equipe. Isso assegura que as mudanças e melhorias possam ser realizadas de maneira eficiente e consistente com o design original do sistema (BOOCH; RUMBAUGH; JACOBSON, 2005).

Em resumo, a modelagem UML é uma prática essencial no desenvolvimento de um sistema web educacional. Ela fornece as ferramentas necessárias para capturar de forma precisa e detalhada os requisitos do sistema, definir sua estrutura e comportamento, e garantir que o sistema final atenda às expectativas dos usuários e stakeholders. A aplicação sistemática da UML ao longo do ciclo de vida do desenvolvimento do sistema contribui para a criação de um software robusto, eficiente e alinhado com os objetivos educacionais do Novo Ensino Médio.

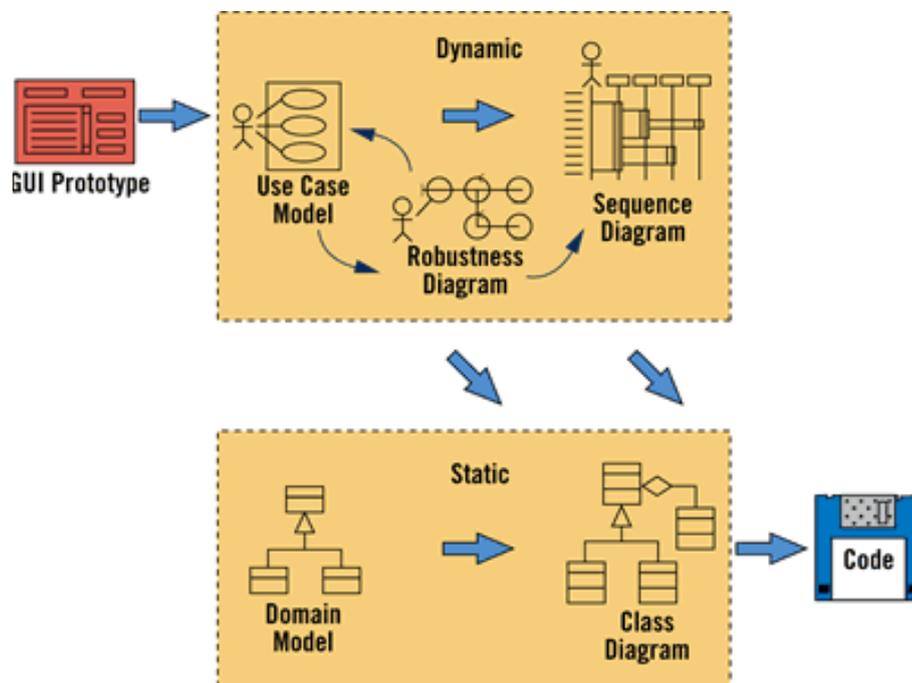
#### **2.4. Metodologia Ágil ICONIX**

A metodologia ICONIX é reconhecida como uma abordagem ágil no desenvolvimento de *software* que se diferencia das metodologias convencionais, como ISO (*International Organization for Standardization*) - 9000 e CMM (*Capability Maturity Model*), por sua inovação e ampla aceitação na comunidade de desenvolvimento de *software*. Ela se destaca pela simplicidade de suas regras e práticas, sendo especialmente orientada para equipes de desenvolvimento de pequeno porte. Sua flexibilidade diante de mudanças e a ênfase na colaboração são aspectos fundamentais que a tornam atrativa para os desenvolvedores. Em resumo, o ICONIX oferece uma estrutura organizacional eficiente que promove uma abordagem ágil e adaptável para o desenvolvimento de *software*.

Com o objetivo de melhorar o controle de tempo, realizar cada atividade de maneira mais eficiente e otimizar os processos de criação do sistema de recomendação, foi conduzida uma análise das metodologias de desenvolvimento de *software* que pudessem atender aos requisitos de tempo e funcionalidade para a implementação e publicação de um sistema amigável, alinhado com as mais recentes tecnologias de informação e comunicação.

A Figura 1 representa visualmente a Metodologia de Desenvolvimento ICONIX, conforme delineada por Doug Rosenberg e Kendall Scott (1999). Essa metodologia é adaptada aos padrões, oferece suporte a UML, é orientada por casos de uso e segue uma abordagem iterativa e incremental. As etapas do ICONIX, como análise de requisitos, análise e *design* preliminar, *design* e implementação, estão alinhadas com as necessidades específicas do projeto.

Figura 1 - Metodologia de desenvolvimento ICONIX



Fonte: de San Martín.

A imagem apresentada descreve um diagrama que ilustra um fluxo típico da metodologia ICONIX utilizada para desenvolvimento de software, especialmente em projetos que envolvem a criação de interfaces gráficas (GUIs). Este diagrama demonstra como diferentes modelos e diagramas se conectam para transformar uma ideia inicial em um produto codificado.

O diagrama começa com o protótipo da interface gráfica (*GUI Prototype*), que representa a primeira concepção visual da interface do usuário. Este protótipo mostra como a aplicação se parecerá para o usuário final, servindo como uma visão inicial do sistema.

Em seguida, o desenvolvimento avança para os modelos dinâmicos. O modelo de casos de uso (*Use Case Model*) descreve as funcionalidades do sistema do ponto de vista do usuário, onde cada caso de uso representa uma interação específica entre o usuário e o sistema. O diagrama de sequência (*Sequence Diagram*) mostra a interação entre os objetos do sistema ao longo do tempo em resposta a um evento específico. O diagrama de robustez (*Robustness Diagram*) é uma variação do diagrama de casos de uso, enfatizando a relação entre os atores (usuários), os casos de uso e os objetos do sistema.

Depois de desenvolver os modelos dinâmicos, o processo transita para os modelos estáticos. O modelo de domínio (*Domain Model*) representa os conceitos do mundo real que o sistema irá modelar, oferecendo uma visão abstrata dos dados e das relações entre eles. O diagrama de classes (*Class Diagram*) mostra as classes, atributos e métodos que compõem o sistema, fornecendo uma representação mais detalhada da estrutura do *software*.

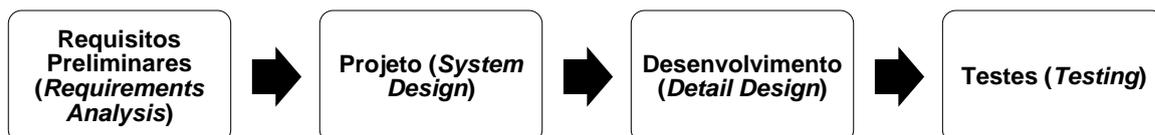
O fluxo do processo de desenvolvimento segue uma sequência bem definida. O processo inicia-se com a criação do protótipo da interface gráfica, que serve como uma visão inicial do sistema e da interação do usuário. A partir do protótipo, são definidos os casos de uso, que detalham as funcionalidades necessárias. Em seguida, diagramas de sequência e robustez são utilizados para detalhar como os objetos do sistema interagem para realizar cada caso de uso. Com base na análise dos casos de uso e na compreensão do domínio do problema, é criado um modelo de domínio que representa os conceitos do mundo real. Este modelo é então refinado em um diagrama de classes, que define as classes, atributos e métodos a serem implementados no código. Finalmente, o diagrama de classes serve como base para a geração do código fonte do sistema, onde ferramentas de desenvolvimento podem automatizar parte desse processo, transformando os elementos visuais do diagrama em código executável.

O significado do diagrama reside na sua capacidade de ilustrar o processo de desenvolvimento de software orientado a objetos, enfatizando a importância de modelar tanto o comportamento dinâmico do sistema (casos de uso, sequências) quanto sua estrutura estática (classes, objetos). Seguindo este fluxo, os desenvolvedores podem garantir que o software atenda às necessidades dos usuários e seja bem estruturado.

É importante notar que o processo de desenvolvimento de software raramente é linear. É comum retornar a etapas anteriores para fazer ajustes e refinamentos, assegurando a qualidade e a adequação do software às necessidades do usuário. Diversas ferramentas de modelagem de software estão disponíveis para criar os diagramas apresentados, com a UML sendo a linguagem de modelagem mais utilizada para esse fim.

A Figura 2, portanto, representa visualmente a estrutura e as fases dessa metodologia no contexto do desenvolvimento do Sistema *Web* de Ensino de Lógica de Programação.

Figura 2 - Fases da Metodologia ICONIX



Fonte: Adaptado de Doug Rosenberg e Matt Stephens, "Use Case Driven Object Modeling with UML: Theory and Practice"

- **Requisitos Preliminares (*Requirements Analysis*):** Nesta fase, os requisitos iniciais do sistema são levantados e analisados em colaboração com os *stakeholders*. O objetivo é entender completamente o escopo do projeto e identificar os principais requisitos funcionais e não funcionais;
- **Projeto (*System Design*):** Durante esta fase, o sistema é projetado em nível arquitetural e de alto nível. Os componentes principais e suas interações são identificados, e uma arquitetura de sistema geral é definida. Isso inclui a elaboração de diagramas de classes, diagramas de sequência e outros modelos de design;
- **Desenvolvimento (*Detail Design*):** Aqui, o *design* de sistema de alto nível é refinado em um nível mais detalhado. Os detalhes de implementação são especificados, o que inclui o *design* de classes, métodos e interfaces. Esta fase também pode incluir o *design* de banco de dados e a elaboração de diagramas de atividades;
- **Testes (*Testing*):** A última fase envolve a execução de testes para verificar se o sistema atende aos requisitos estabelecidos. Isso inclui testes de unidade, integração e sistema, além de testes de aceitação pelo usuário. Os resultados dos testes são usados para validar e verificar a qualidade do sistema finalizado.

Em resumo, a metodologia ágil ICONIX se revelou uma abordagem robusta e

adaptável para o desenvolvimento de sistemas, independentemente de sua natureza ou escopo. Sua ênfase na colaboração, flexibilidade e excelência técnica proporciona uma estrutura sólida para o progresso eficiente do projeto. Ao seguir as fases delineadas pela ICONIX, desde a análise inicial dos requisitos até os testes finais, os desenvolvedores podem contar com uma orientação clara e organizada para alcançar os objetivos do sistema. A aplicação dos princípios do ICONIX promove a superação de desafios, a adaptação às mudanças e a entrega de um produto de alta qualidade, alinhado às expectativas e necessidades dos usuários.

## **2.5. Linguagem de Programação Python**

*Python*, uma linguagem de programação de alto nível, interpretada e marcada por tipagem forte e dinâmica, foi concebida por *Guido van Rossum* nos anos 80, com a intenção de simplificar a programação e aliviar as preocupações dos programadores com *hardware*. Ao longo do tempo, evoluiu para se tornar uma linguagem poderosa, abrangendo áreas como *Data Science*, Inteligência Artificial e Desenvolvimento *Web*. Seu ecossistema vasto, composto por mais de 246.000 bibliotecas (*Python*, 2023), promove o reuso eficiente de funcionalidades no desenvolvimento de *software*.

A notável ascensão de *Python* no cenário de programação, rivalizando com linguagens tradicionais como C++ e Java, é evidenciada pelo *RedMonk*, colocando *Python* na segunda posição entre as linguagens mais populares (Stephens, 2020). Sua natureza multiplataforma, interoperabilidade e código aberto contribuem para sua ampla adoção, possibilitando que desenvolvedores integrem ferramentas de outras linguagens.

A filosofia do *Python* destaca-se pela ênfase na conveniência na programação, expressa em códigos concisos, organizados e de fácil leitura. A indentação como parte integral da sintaxe impulsiona a criação de códigos limpos, reduzindo erros. Projetada para oferecer facilidade tanto a iniciantes quanto a programadores experientes, a linguagem destaca-se por sua flexibilidade e simplicidade.

Devido a essas características distintivas, várias empresas líderes, como *Netflix*, *Google*, *Instagram* e *Spotify*, escolhem implementar partes ou a totalidade de seus produtos em *Python*. Essa preferência é notável em diversos setores, desde a segurança de dados até o desenvolvimento em nuvem (Demchenko, 2019)

## 2.6. Framework Django

O *Django* é um *framework* para o desenvolvimento de aplicações *web* escrito em *Python*, originado em 2005 por uma equipe de programadores do *Lawrence Journal-World* com o objetivo de agilizar o processo de criação de aplicações *web* (*Django* 2023). Este *framework* ganhou destaque por oferecer soluções abrangentes para muitos dos desafios tradicionais encontrados no desenvolvimento *web*. Ele disponibiliza dezenas de tarefas comuns já implementadas e prontas para reutilização, abrangendo áreas como autenticação de usuário, administração de conteúdo, mapas de site, entre outras.

O *Django* não apenas simplifica o desenvolvimento, mas também contribui para a segurança das aplicações, ajudando a evitar erros comuns de segurança. Além disso, proporciona escalabilidade aos sistemas, o que permite a expansão de um sistema sem comprometer seu desempenho, como observado em casos de uso notáveis, incluindo *Mozilla Firefox*, *Pinterest* e *Instagram*. Muitas empresas optam pelo *Django* devido à sua extrema versatilidade para criar uma variedade de sistemas, desde plataformas de gerenciamento de conteúdo até soluções científicas.

## 2.7. Padrão arquitetural *Model-Template-View* (MTV)

O padrão MTV é uma arquitetura de desenvolvimento de *software* amplamente utilizado em *frameworks web*, especialmente em *Python*, onde é implementado pelo *framework Django* para construção de aplicativos robustos e escaláveis (*Django* 2023).

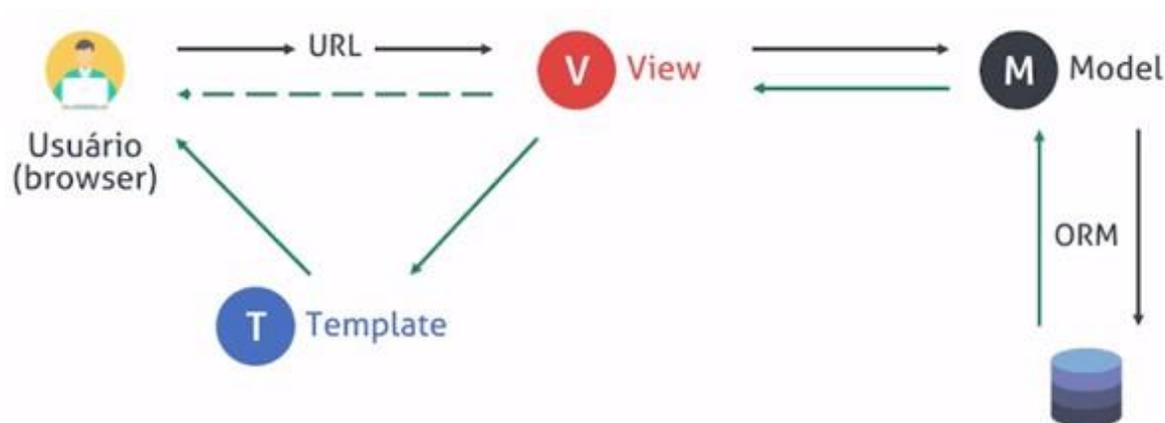
A estrutura MTV, de acordo com a Figura 3, característica do *framework Django*, compreende as camadas *Model*, *Template* e *View*, proporcionando um fluxo eficaz entre elas, a seguir uma breve descrição sobre cada um dos componentes:

- **Model (Modelo):** O *Django* inclui uma solução para mapeamento objeto-relacional (do inglês, ORM - *Object-Relational Mapping*), onde o esquema do banco de dados é definido em código *Python* (*Python* 2023). Abstrações *Python* são empregadas para criar consultas complexas sem interações diretas no banco;
- **Template (Modelo de Apresentação):** Responsável pela apresentação, o

*template* define como as informações são exibidas aos usuários. Combina partes estáticas em HTML (*HyperText Markup Language*) com sintaxe especial para apresentar o conteúdo dinâmico. O *Django* utiliza um caminho de pesquisa para minimizar redundâncias entre *templates*. As *views* recuperam dados, carregam um *template* e o renderizam (*Django 2023*);

- **View (Visualização):** As *views* processam informações e o tipo de requisição do cliente, formatam dados para armazenamento no banco via *models* da camada *Model* e se comunicam com o banco para recuperar dados transferidos posteriormente aos *templates*. Cada *view* retorna um objeto *HTTPResponse* contendo o conteúdo para a página requisitada ou levanta uma exceção como *Http404* (*Django 2023*).

Figura 3 - Representação visual do Modelo MTV no framework Django, destacando as camadas



Fonte: <https://docs.djangoproject.com/en/4.2/>

A imagem apresentada descreve um diagrama simplificado, mas eficaz, de uma arquitetura MTV com a adição de um componente ORM. Essa arquitetura é amplamente utilizada no desenvolvimento de aplicações *web* para separar as diferentes responsabilidades do software, promovendo maior organização, manutenibilidade e escalabilidade.

Os elementos principais da arquitetura e suas funcionalidades são: o Usuário (*browser*), que é o ponto de partida da interação. Ao digitar um URL (*Uniform Resource Locator*) no navegador, o usuário inicia uma requisição, que é identificada e direcionada para o servidor. A *View* é a camada responsável pela interface do usuário, recebendo dados do modelo e os apresentando de forma formatada para o usuário, geralmente através de *templates*. Esses *templates* são arquivos que definem

a estrutura e o *layout* da página, combinando dados do modelo com elementos HTML para gerar a interface final. A camada Model (Modelo) contém a lógica de negócio da aplicação, gerenciando os dados e as regras de negócio, representando os dados da aplicação. Finalmente, a ORM é uma camada de abstração que facilita a interação entre a aplicação e o banco de dados, mapeando objetos do modelo para tabelas no banco de dados e simplificando as operações de leitura e escrita.

No contexto do desenvolvimento do Sistema *Web* de Ensino de Lógica de Programação proposto, a adoção da arquitetura MTV se mostra fundamental. A camada *Model* permite a integração eficiente com o banco de dados, onde são armazenados os dados pertinentes às sugestões de aprendizado. A *Template*, por sua vez, é responsável por definir como essas sugestões serão apresentadas ao usuário, que garante uma interface amigável e personalizada. A *View*, age como elo entre o modelo e a apresentação, processa as informações e coordena a lógica por trás dos testes e sugestões de conteúdo, o que garante uma experiência coesa e eficiente para os usuários do sistema.

Em um site convencional, as aplicações Django aguardam requisições HTTP do navegador, calculam as ações necessárias com base na URL e dados enviados via POST ou GET, realizam operações no banco e geram dinamicamente uma página HTML para exibição no navegador, que utiliza os arquivos *urls.py*, *views.py* e *models.py* para organizar o código. O arquivo *urls.py* mapeia URLs e redireciona requisições para a *view* apropriada. As *views* manipulam requisições HTTP, e os *models* definem a estrutura dos dados, independentemente do banco escolhido. *Migrations* são utilizadas para propagar alterações nos *models* para o banco (Django 2023).

Os objetos *models* também oferecem um mecanismo simples para recuperação de dados no banco por meio de consultas. Existem vários tipos de consultas disponíveis, como *filtragens* e *ordenações*. A *view*, após tratar a requisição, retorna o resultado para a camada *Template*, onde é renderizado um arquivo de saída, geralmente um HTML, que contém trechos de código Python para manipulação dos objetos retornados. O Django oferece recursos adicionais, como formulários, autenticação segura, armazenamento em *cache* e interface administrativa padrão (Django 2023).

## 2.8. Banco de Dados *SQLite*

*SQLite* é uma biblioteca de linguagem C<sup>5</sup> que implementa um mecanismo de banco de dados SQL pequeno, rápido, autônomo, de alta confiabilidade e completo. *SQLite* é o mecanismo de banco de dados mais usado no mundo. *SQLite* é construído em todos os celulares e na maioria dos computadores e vem empacotado dentro de inúmeros outros aplicativos que as pessoas usam todos os dias.

O formato de arquivo *SQLite* é estável, multiplataforma e compatível com versões anteriores, e os desenvolvedores prometem mantê-lo assim até o ano de 2050. Os arquivos de banco de dados *SQLite* são comumente usados como contêineres para transferir conteúdo rico entre sistemas e como um formato de arquivamento de longo prazo para dados. Existem mais de 1 trilhão de bancos de dados *SQLite* em uso ativo. O código-fonte do *SQLite* é de domínio público e pode ser usado por qualquer pessoa para qualquer finalidade. (*SQLite*, 2024)

O banco de dados *SQLite* armazena de maneira eficiente as informações dos usuários, que consistem em suas respostas aos testes e os conteúdos sugeridos pelos professores. Essa escolha visa possibilitar o acesso rápido e preciso a essas informações, o que permite ao sistema gerar sugestões personalizadas para cada usuário e promove uma experiência mais adaptativa.

O *SQLite* é representado como uma entidade independente, conectada de forma bidirecional ao componente *Models*. Este último representa as classes de objetos do *Django*, que por sua vez mapeiam as tabelas do banco de dados. A interação com o *SQLite* é realizada por meio do ORM do *Django*, proporcionando a capacidade de executar operações no banco de dados usando objetos *Python*, sem a necessidade de manipulação direta de SQL.

As *Views* do *Django*, responsáveis por atender às solicitações HTTP dos usuários, utilizam os dados armazenados no *SQLite*. Elas podem recuperar, manipular e renderizar essas informações em um *Template*, que, por sua vez, é retornado como

---

<sup>5</sup> Linguagem de programação de propósito geral desenvolvida no início dos anos 1970 por Dennis Ritchie, na Bell Labs. A linguagem C é conhecida por sua eficiência e controle de baixo nível sobre os recursos do sistema, sendo amplamente utilizada para o desenvolvimento de sistemas operacionais, compiladores, e outros softwares que requerem alta performance e interação direta com o hardware.

uma resposta HTTP para o navegador do usuário.

A Figura 4 representa o Fluxo ORM na Arquitetura do *Django* e destaca a interação entre *Models*, *SQLite* (banco de dados), *Django Views*, *Templates* e o *Browser*. Essa visualização ilustra a dinâmica do sistema, onde os *Models* definem a estrutura dos dados, o *SQLite* atua como o banco de dados, as *Views* processam informações e se comunicam com o banco, os *Templates* apresentam os dados visualmente, e o *Browser* exibe a interface final.

Figura 4 - Fluxo ORM: Interação entre *Models*, *SQLite3* (banco de dados), *Django Views*, *Templates* e o *Browser* na arquitetura do *Django*



Fonte: Autor

## 2.9. Ambiente de Desenvolvimento Integrado (IDE)

Um Ambiente de Desenvolvimento Integrado (IDE) é um software essencial para a criação de aplicações, que combina ferramentas comuns de desenvolvimento em uma única interface gráfica de usuário (GUI). Um dos componentes centrais de um IDE é o editor de código-fonte, que funciona como um editor de texto especializado, projetado para facilitar a escrita de código de software. Esse editor oferece funcionalidades como destaque de sintaxe, com indicadores visuais que ajudam na legibilidade do código, além de recursos de preenchimento automático específicos da linguagem e verificação de erros durante o desenvolvimento, tornando o processo de codificação mais eficiente e menos propenso a erros.

Outro elemento importante dentro de um IDE é a automação de compilação local. Essa automação envolve utilitários que simplificam e agilizam tarefas repetitivas durante a criação de uma compilação local do software, que é usada pelo desenvolvedor. Entre essas tarefas estão a compilação do código-fonte em código binário, a criação de pacotes de código binário, e a execução de testes automatizados. Esses processos automatizados ajudam a manter a consistência e a qualidade do software durante o desenvolvimento, além de economizar tempo e esforço dos desenvolvedores.

Por fim, o *debugger* é uma ferramenta vital oferecida pelos IDEs. Ele é usado para testar outros programas, permitindo ao desenvolvedor identificar e corrigir erros de forma eficiente. O *debugger* mostra graficamente a localização dos bugs no código original, facilitando a análise e resolução de problemas. Com o uso do debugger, o desenvolvedor pode rastrear a execução do código, verificar valores de variáveis em tempo real, e entender melhor o comportamento do programa, o que é crucial para a depuração e otimização do *software*.

O processo de desenvolvimento contou com o auxílio do *Visual Studio Code* (VSCode), um editor de texto de código aberto amplamente adotado. Sua escolha se baseou em sua expressiva popularidade, evidenciada pelo fato de que, em 2019, o VSCode representava mais de 50% do uso de Ambientes de Desenvolvimento Integrado (IDE) (OVERFLOW, 2019). No que diz respeito ao versionamento, o *GitHub* foi a escolha, sendo amplamente reconhecido como o sistema de controle de versão mais utilizado globalmente (Alura, 2023).

## **2.10. Heurísticas de usabilidade para sistemas web**

Conceitualmente, as heurísticas são entendidas como diretrizes gerais que descrevem a usabilidade de sistemas. Inseridas no campo da engenharia de *software*, essas regras são amplamente adotadas como critérios para avaliar a usabilidade de softwares. De acordo com a NBR 9241-11, que aborda a usabilidade, esta é definida como a capacidade de um produto de ser utilizado por um usuário para atingir seus objetivos, considerando aspectos como eficácia, eficiência e satisfação (PREECE, 2005; ABNT, 2002).

Os critérios de usabilidade para sistemas *web* são fundamentais, uma vez que os usuários interagem com essas plataformas em diversas atividades do cotidiano e esperam não apenas um alto índice de eficácia, mas também de satisfação. As práticas tradicionais de usabilidade e interação têm sido transformadas pelo avanço da tecnologia *web*, que ampliou as possibilidades e elevou as expectativas dos usuários (FEIJOÁ, GONÇALVEZ e GOMEZ, 2013).

Os estudos iniciais sobre heurísticas de usabilidade datam da década de 1990, com autores como Nielsen (1993), Molich (1990) e Lewis (1990) desenvolvendo métodos que serviram como diretrizes para a construção e avaliação de aplicações

web. Esses métodos incluem técnicas de avaliação de software baseadas em questionários, que se tornaram essenciais para a análise e melhoria da usabilidade de sistemas *web*.

As heurísticas propostas por Nielsen (1993) totalizam dez, e a partir delas é possível destacarmos seis atributos principais a se considerar quanto a forma como os usuários irão interagir com o software, estas são apresentadas na tabela a seguir:

Tabela 2 - Atributos associados ao conjunto de heurísticas de Nielsen.

H1	Eficácia
H2	Eficiência
H3	Segurança
H4	Facilidade de memorização
H5	Satisfação do usuário
H6	Baixa taxa de erros

Fonte: (OLIVEIRA, 2020)

A crescente complexidade das interfaces *web*, impulsionada pela diversidade de dispositivos e navegadores, torna a avaliação da usabilidade um desafio cada vez maior. As heurísticas propostas por Nielsen, embora sejam um ponto de partida fundamental, podem não ser suficientes para capturar as nuances específicas das interfaces *web* modernas. É necessário considerar outros fatores, como a compatibilidade com diferentes resoluções de tela, a otimização para mecanismos de busca e a experiência do usuário em geral. Nesse sentido, diversos pesquisadores têm proposto novas heurísticas e métricas para avaliar a usabilidade de sistemas *web*, buscando atender às demandas de um mercado em constante evolução.

Ao analisar interfaces *web*, é fundamental garantir uma experiência intuitiva e eficiente para os usuários. A navegação deve ser clara e objetiva, a informação precisa ser apresentada de forma organizada e a interação com os elementos da interface deve ser natural. A experiência do usuário, que engloba aspectos como satisfação, facilidade de uso e eficiência, deve ser o foco principal durante o processo de desenvolvimento e avaliação.

Diversos estudos têm identificado a necessidade de adaptar as heurísticas de

Nielsen para o contexto das interfaces *web* modernas. Por exemplo, a importância de fornecer *feedback* claro para as ações do usuário, evitar tempos de espera excessivos e organizar a informação de forma visualmente agradável. Além disso, existem heurísticas específicas para aplicativos educacionais que podem ser adaptadas para cursos *online*, como a necessidade de explicar claramente os objetivos de aprendizagem e de promover a interação entre os usuários.

Para avaliar a usabilidade de um sistema *web*, é possível utilizar tantos testes de usabilidade, que envolvem a participação de usuários reais, quanto inspeções de usabilidade, que são realizadas por especialistas. Independentemente do método escolhido, é fundamental realizar avaliações em diferentes etapas do desenvolvimento, a fim de identificar e corrigir problemas de usabilidade o mais cedo possível.

### **2.11. Usabilidade**

Os testes de usabilidade são essenciais para avaliar o grau de satisfação do usuário ao interagir com um sistema ou aplicação. A área de Interação Humano-Computador (IHC) integra os conceitos de usabilidade no ciclo de desenvolvimento de software, visando melhorar a eficácia e a satisfação do usuário. Isso envolve a criação de sistemas que permitam a execução de tarefas de forma segura, eficiente e satisfatória (PREECE; ROGERS; SHARP, 2007).

Para obter informações sobre a usabilidade de um sistema ou aplicação, diversas técnicas de análise podem ser empregadas, entre as quais se destacam os Questionários de Satisfação de Usuários. Esses questionários são aplicáveis a qualquer área e são fundamentais para todas as aplicações. Exemplos incluem o *Questionnaire for User Interface Satisfaction* (QUIS), o *Software Usability Measurement Inventory* (SUMI), a *System Usability Scale* (SUS) e o *Web Analysis and Measurement Inventory* (WAMMI) (SAURO; LEWIS, 2012).

O QUIS avalia componentes gerais da interface, como tela, terminologia, retorno do sistema, capacidade do sistema, nível de aprendizado, manuais técnicos, tutoriais online e multimídia. Pode ser aplicado de maneira extensa, com setenta e uma questões, ou de forma reduzida, com vinte e seis questões. Suas vantagens incluem alta confiabilidade, baixa variabilidade e uma ampla gama de informações

úteis para os projetistas (CHIN, 1988; KULKARNI et al., 2013).

De forma semelhante ao QUIS, o SUMI mede a qualidade de software considerando fatores como eficiência, estabilidade, aprendizado, controle e afeto. O SUMI consiste em cinquenta declarações, onde o usuário responde com "Concordo", "Indeciso" ou "Discordo". Para obter dados suficientes, recomenda-se a participação de pelo menos dez usuários (KULKARNI et al., 2013).

John Brooke propôs a Escala de Usabilidade de *Software* (SUS) em 1996. Esta é uma ferramenta robusta para avaliar a usabilidade de qualquer produto ou serviço, composta por dez afirmações com uma escala de concordância que varia de "Concordo plenamente" a "Discordo totalmente". A pontuação resultante, variando de zero a cem pontos, reflete o nível de usabilidade do produto. Devido à sua facilidade de administração e pontuação, o SUS é amplamente utilizado por profissionais de IHC (FINSTAD, 2006; BANGOR; KORTUM; MILLER, 2008).

Para avaliar a usabilidade de sites, é utilizado o WAMMI. Este questionário inclui vinte questões, com respostas em uma escala *Likert* de cinco pontos, de "concordo totalmente" a "discordo fortemente". As subescalas de análise são semelhantes às do SUMI, incluindo eficiência, controle, aprendizagem, entre outras. Uma vantagem significativa do WAMMI é a possibilidade de comparar resultados com avaliações de outros sites similares, permitindo a organização de uma média geral de usabilidade (HAYAT; LOCK; MURRAY, 2015).

### **3. O NOVO ENSINO MÉDIO**

Neste capítulo, é discutida a evolução do Ensino Médio no Brasil, com ênfase nas características e mudanças introduzidas pelo Novo Ensino Médio, essenciais para compreender o contexto e a proposta deste trabalho.

#### **3.1. O Ensino Médio**

O Ensino Médio brasileiro, ao longo de sua trajetória histórica, tem sido moldado por contextos políticos, sociais e econômicos que refletem as desigualdades estruturais da sociedade. Antes da implantação do Novo Ensino Médio, o sistema educacional apresentava uma dualidade marcante, onde a educação secundária era segmentada de acordo com as classes sociais. Para a elite, o ensino médio era um espaço de preparação para o ingresso no ensino superior, enquanto para a classe popular, a oferta educacional se restringia a cursos profissionalizantes que não visavam a continuidade dos estudos, mas sim a inserção imediata no mercado de trabalho (ALVES; SILVA; JUCÁ, 2022, p. 149).

Durante as décadas de 1960 e 1970, o Brasil vivenciou um processo de industrialização acelerada, que influenciou diretamente as diretrizes educacionais. O governo militar, em resposta às demandas do desenvolvimento econômico, priorizou a formação de uma força de trabalho qualificada, promovendo a profissionalização compulsória no ensino médio. Essa estratégia visava não apenas atender às necessidades do mercado, mas também conter a demanda por vagas no ensino superior, que já apresentava sinais de saturação (ALVES; SILVA; JUCÁ, 2022, p. 151). O currículo do Ensino Médio, portanto, foi orientado para a formação técnica e utilitária, despolitizando o ambiente escolar e limitando a formação crítica dos estudantes.

Adicionalmente, a obrigatoriedade do ensino de língua estrangeira, especialmente a Língua Inglesa, foi implementada em 1976, refletindo a influência dos Estados Unidos e a necessidade de preparar os alunos para um mercado de trabalho cada vez mais globalizado (ALVES; SILVA; JUCÁ, 2022, p. 149). Essa mudança curricular, embora ampliasse o leque de conhecimentos oferecidos, também evidenciava a continuidade da exclusão social, uma vez que a educação de qualidade permanecia restrita a uma minoria privilegiada.

Assim, o Ensino Médio antes do Novo Ensino Médio consolidou-se como um espaço de reprodução das desigualdades sociais, onde a formação educacional era desigualmente distribuída. A transição para o Novo Ensino Médio, portanto, não pode ser compreendida sem uma análise crítica das estruturas que sustentaram o modelo anterior, que perpetuou a desqualificação da educação para as classes populares e a elitização do conhecimento.

A transição para o Novo Ensino Médio no Brasil representa uma tentativa de reverter as desigualdades históricas que marcaram a educação secundária, buscando uma formação mais flexível e adaptada às necessidades contemporâneas dos estudantes. Essa reforma curricular, que introduz itinerários formativos e permite uma maior personalização do aprendizado, visa não apenas atender às demandas do mercado de trabalho, mas também promover uma educação mais inclusiva e diversificada. Ao reconhecer a necessidade de uma formação que dialogue com as realidades sociais e econômicas dos alunos, o Novo Ensino Médio busca superar a lógica de desqualificação que predominou no modelo anterior, onde a educação era frequentemente vista como um privilégio das elites. Assim, a proposta do Novo Ensino Médio se insere em um contexto de busca por equidade educacional, refletindo uma nova visão sobre o papel da escola na formação integral dos jovens e na construção de uma sociedade mais justa e igualitária.

Nesse contexto o Novo Ensino Médio propõe uma reforma na matriz de referência curricular dos alunos do 1º, 2º e 3º ano dessa etapa escolar. A Lei nº 13.415/2017<sup>6</sup>, que institui as alterações, estabelece maior integração e flexibilidade curricular na direção de substituir o modelo único de currículo do Ensino Médio por um modelo diversificado e flexível, a Lei supracitada alterou a Lei de Diretrizes e Bases da Educação Nacional (LDB), estabelecendo que o currículo do ensino médio será

---

<sup>6</sup> Lei nº 13.415, de 16 de fevereiro de 2017. Altera as Leis nº 9.394, de 20 de dezembro de 1996, que estabelece as diretrizes e bases da educação nacional, e nº 11.494, de 20 de junho de 2007, que regulamenta o Fundo de Manutenção e Desenvolvimento da Educação Básica e de Valorização dos Profissionais da Educação, a Consolidação das Leis do Trabalho – CLT, aprovada pelo Decreto-Lei nº 5.452, de 1º de maio de 1943, e pelo Decreto-Lei nº 236, de 28 de fevereiro de 1967; revoga a Lei nº 11.161, de 5 de agosto de 2005; e institui a Política de Fomento à Implementação de Escolas de Ensino Médio em Tempo Integral. Diário Oficial da União, Brasília, 17 de fevereiro de 2017. Disponível em: < [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2017/lei/113415.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/113415.htm) >. Acesso em: 16 jul. 2024.

composto pela Base Nacional Comum Curricular (BNCC) e por itinerários formativos<sup>7</sup>, que deverão ser organizados por meio da oferta de diferentes arranjos curriculares, conforme a relevância para o contexto local e a possibilidade dos sistemas de ensino, a saber:

- I. linguagens e suas tecnologias;
- II. matemática e suas tecnologias;
- III. ciências da natureza e suas tecnologias;
- IV. ciências humanas e sociais aplicadas;
- V. formação técnica e profissional (LDB, Art. 36; ênfases adicionadas).

Além de ser um novo modelo de aprendizagem, dado agora, por áreas de conhecimento que permitirá ao jovem optar por uma formação técnica e profissionalizante. Ao final do ensino médio, o aluno receberá além do certificado do ensino médio regular e o certificado do curso técnico ou profissionalizante que cursou.

Como mencionado anteriormente são cinco itinerários que a escola pode ofertar – entre eles, o de formação técnica e profissional – e os alunos escolherão qual cursar de acordo com as áreas de seu interesse e projetos de vida e de carreira. O novo currículo do Ensino Médio é organizado por áreas de conhecimento e não por matérias e será composta por 4 áreas de conhecimento mais uma de formação Técnica e Profissional. Na nova estrutura, até 1.800 horas da carga horária contemplam habilidades e competências relacionadas às 04 áreas do conhecimento. São eles: Matemáticas e suas Tecnologias; Linguagens e suas Tecnologias; Ciências da Natureza e suas Tecnologias; Ciências Humanas e Sociais Aplicadas; e, no mínimo, 1.200 horas são flexíveis e ficarão reservados para a Formação Técnica e Profissional.

As principais mudanças do Novo Ensino Médio são o aumento da carga horária dos estudantes, a adoção de uma base comum curricular e a escolha dos itinerários formativos por parte do aluno. O novo modelo começou ser implementado de forma

---

<sup>7</sup> No Brasil, a expressão “itinerário formativo” tem sido tradicionalmente utilizada no âmbito da educação profissional, em referência à maneira como se organizam os sistemas de formação profissional ou, ainda, às formas de acesso às profissões. No entanto, na Lei nº 13.415/17, a expressão foi utilizada em referência a itinerários formativos acadêmicos, o que supõe o aprofundamento em uma ou mais áreas curriculares, e, a itinerários da formação técnica profissional.

gradual a partir de 2022 é um modelo de aprendizagem focada na formação de cidadãos e no desenvolvimento de competências e habilidades, com disciplinas integradas em quatro áreas do conhecimento que possibilita que os alunos escolham itinerários formativos de acordo com áreas de seu interesse e projetos de vida e de carreira.

Escolas de ensino médio de todo Brasil enfrentarão o desafio de implementar as novas diretrizes curriculares nacionais. A adaptação das escolas começou em 2018 e será realizada de forma gradativa. Mais de 78 escolas do Serviço Social da Indústria (SESI), em 22 estados, já estão integradas às novas regras. Para a oferta do itinerário V, de formação técnica profissional, o SESI conta com a parceria do Serviço Nacional de Aprendizagem Industrial (SENAI), como forma de conectar educação com mercado de trabalho.

Os estudos mencionados, tentam de alguma forma reduzir as dificuldades atribuídas ao ensino da Lógica de Programação, seja através da mudança na condução das atividades pedagógicas, mediação ou estratégias metodológicas ou através do uso de ferramentas específicas, contudo é sabido que essas tarefas não são fáceis ou um tanto dificultosas de serem superadas, porém podem ser amenizadas em alta escala com metodologias pedagógicas adequadas e as ferramentas inovadoras aplicadas ao contexto do Novo Ensino Médio.

### **3.2. Base Nacional Comum Curricular**

A Base Nacional Comum Curricular (BNCC) é um documento de caráter normativo que define o conjunto orgânico e progressivo de aprendizagens essenciais que todos os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica, de modo a que tenham assegurados seus direitos de aprendizagem e desenvolvimento, em conformidade com o que preceitua o Plano Nacional de Educação (PNE). Este documento normativo aplica-se exclusivamente à educação escolar, tal como a define o § 1º do Artigo 1º da Lei de Diretrizes e Bases da Educação Nacional (LDB, Lei nº 9.394/1996), e está orientado pelos princípios éticos, políticos e estéticos que visam à formação humana integral e à construção de uma sociedade justa, democrática e inclusiva, como fundamentado nas Diretrizes Curriculares Nacionais da Educação Básica (DCN).

Referência nacional para a formulação dos currículos dos sistemas e das redes

escolares dos Estados, do Distrito Federal e dos Municípios e das propostas pedagógicas das instituições escolares, a BNCC integra a política nacional da Educação Básica e vai contribuir para o alinhamento de outras políticas e ações, em âmbito federal, estadual e municipal, referentes à formação de professores, à avaliação, à elaboração de conteúdos educacionais e aos critérios para a oferta de infraestrutura adequada para o pleno desenvolvimento da educação.

Nesse sentido, espera-se que a BNCC ajude a superar a fragmentação das políticas educacionais, enseje o fortalecimento do regime de colaboração entre as três esferas de governo e seja balizadora da qualidade da educação. Assim, para além da garantia de acesso e permanência na escola, é necessário que sistemas, redes e escolas garantam um patamar comum de aprendizagens a todos os estudantes, tarefa para a qual a BNCC é instrumento fundamental.

Ao longo da Educação Básica, as aprendizagens essenciais definidas na BNCC devem concorrer para assegurar aos estudantes o desenvolvimento de dez competências gerais, que consubstanciam, no âmbito pedagógico, os direitos de aprendizagem e desenvolvimento.

Na BNCC, competência é definida como a mobilização de conhecimentos (conceitos e procedimentos), habilidades (práticas, cognitivas e socioemocionais), atitudes e valores para resolver demandas complexas da vida cotidiana, do pleno exercício da cidadania e do mundo do trabalho.

Ao definir essas competências, a BNCC reconhece que a “educação deve afirmar valores e estimular ações que contribuam para a transformação da sociedade, tornando-a mais humana, socialmente justa e, também, voltada para a preservação da natureza” (BRASIL, 2013), mostrando-se também alinhada à Agenda 2030 da Organização das Nações Unidas (ONU).

É imprescindível destacar que as competências gerais da Educação Básica, apresentadas a seguir, inter-relacionam-se e desdobram-se no tratamento didático proposto para as três etapas da Educação Básica (Educação Infantil, Ensino Fundamental e Ensino Médio), articulando-se na construção de conhecimentos, no desenvolvimento de habilidades e na formação de atitudes e valores, nos termos da LDB.

### 3.3. *Microlearning* e *Nanolearning*

Com o advento das mídias sociais, duas abordagens educativas distintas evoluíram a partir dos métodos de *e-learning*<sup>8</sup> existentes, conhecidas como *Microlearning* e *Nanolearning*. Em síntese, *nanolearning* é inspirada nos princípios da nanotecnologia, apresenta pequenas peças independentes e unificadas, enquanto a abordagem *microlearning* consiste em atividades de aprendizagem de curto prazo divididas em unidades menores para alcançar objetivos específicos.

O *Microlearning* é uma modalidade de aprendizado que abrange aspectos de ensino e educação, com foco direcionado para o nível micro, em particular, micro conteúdo ou micro mídia (recursos midiáticos em tamanho micro). Ele lida com unidades de aprendizado relativamente pequenas e atividades educacionais de curto prazo.

As atividades de *Microlearning*, por definição, dependem do acesso a recursos e conteúdo de aprendizado, o que pode ocorrer em momentos de pausa ou intervalo nas atividades cotidianas dos estudantes que trabalham. Uma vez que esses intervalos podem ocorrer em lugares e momentos diferentes, o *Microlearning* é, sem dúvida, a maneira típica de aprender a qualquer hora e em qualquer lugar (Gabrielli, Kimani, & Catarci, 2006). O *Microlearning* é considerado especialmente adequado para a aprendizagem informal em atividades específicas em que os alunos estão interessados em informações curtas e específicas, em vez de acessar um corpo sólido de conhecimento sobre um assunto específico. Portanto, o *Microlearning* significa a experiência de microensino como um método de aprendizado eficaz para os alunos (Hug, 2006), o que requer conteúdo e mídia apropriados.

A abordagem *Nanolearning* é definida como um programa de aprendizagem projetado para permitir que um participante aprenda qualquer assunto em um período de dez minutos. Isso ocorre por meio de mídia eletrônica e sem interação em tempo real com um instrutor. Os programas de *nanolearning* tendem a concentrar-se num

---

<sup>8</sup> O *e-learning* é uma abreviação de “*electronic learning*”, que em português pode ser traduzido como “aprendizado eletrônico”. Trata-se de um tipo de educação remota, que tem como base a utilização de recursos audiovisuais de computadores para promover o aprendizado. Todo o conteúdo, bem como materiais de apoio como apostilas e fóruns são realizados on-line. A comunicação, entre os alunos ou entre os discentes e os professores, se dá por meio de mensagens instantâneas. Tudo isso é realizado por meio de um ambiente virtual de aprendizagem, sem nada físico.

objetivo de aprendizagem específico ou num assunto específico. Não mais do que uma disciplina pode ser trocada em um curso presencial, e o foco do aprendizado geralmente não muda de um objetivo. É importante lembrar que a nano educação não é um programa de grupo. O objetivo é dar a cada aluno a cápsula exata de conhecimento de que ele precisa. Esta abordagem elimina a possibilidade de treinamento em grupo, mesmo nos menores grupos. Outro ponto crítico é que *nanolearning* não substitui programas abrangentes que abordam questões complexas. Em vez disso, o formato é adequado para abordar áreas específicas do assunto que não são claras ou que explicam tópicos restritos. Para uma compreensão qualitativa de um assunto mais amplo, o aluno precisa de tempo suficiente para uma explicação extensa, o que o *nanolearning* não pode fornecer (Pritchard, 2017). Isso ocorre porque uma nano lição geralmente dura entre dois e dez minutos. Este é o tempo suficiente para explicar um tópico restrito ou esclarecer certos aspectos de um tópico amplo.

Tanto o *nanolearning* quanto o *microlearning* partilham muitas semelhanças; por exemplo, ambos procuram apresentar formas de conceber, distribuir e utilizar pequenos elementos de aprendizagem.

Em comparação com o *microlearning*, a *nanolearning* envolve uma miniaturização ainda mais acentuada do processo de aprendizagem. Ao contrastar as duas abordagens, as unidades de *microlearning* são desmembradas em subunidades, concentrando-nos em unidades mais extensas para alcançar uma atenção sustentada e uma compreensão abrangente por meio da descoberta. Essa abordagem visa resolver problemas e fomentar a expressão criativa nos objetos de aprendizagem.

Outra distinção entre *microlearning* e *nanolearning* reside no alcance de sua aplicação. Enquanto o *microlearning* pode ser utilizado tanto em ambientes de aprendizagem formais quanto informais (Buchem, I., Heilmann, H., 2010) e (Nikou, S. A., & Economides, A. A., 2018), a *nanolearning* é mais restrita, sendo adequada principalmente para ambientes de aprendizagem informais (Gramming, A. C.; Ejemyr, E.; Thunell, E., 2019).

O conceito de *microlearning* é altamente relevante quando se trata de ensinar Lógica de Programação, pois permite a entrega de conteúdo educacional de forma concisa e focada em conceitos específicos. Um Sistema *Web* para Ensino de Lógica

de Programação, que utiliza o *microlearning* como abordagem principal, oferece uma solução eficaz para ajudar os alunos a dominar a Lógica de Programação. Esse sistema pode analisar o desempenho do aluno em uma prova de conhecimento sobre determinado conteúdo e, com base nessa análise, sugerir vídeos colocados por especialistas que abordam os conceitos com os quais o aluno enfrenta dificuldades. Esses vídeos seriam curtos e diretos ao ponto, e forneceria explicações claras e exemplos práticos que ajudam os alunos a compreender e aplicar os conceitos de forma eficaz. Ao utilizar o *microlearning*, o sistema permite que os alunos aprendam em pequenas doses, adapte-se ao seu ritmo de aprendizado e ofereça uma experiência de aprendizado personalizada e eficiente. Isso ajuda a maximizar a retenção de informações e a promover um progresso contínuo na compreensão da Lógica de Programação.

#### 4. TRABALHOS RELACIONADOS

O presente capítulo apresenta alguns trabalhos relacionados ao desenvolvimento de soluções para o Ensino e Aprendizagem de Lógica de Programação, assim como trabalhos que trazem ferramentas que proporcionem uma experiência nessa prática, e que serviram como referência para construção desta proposta.

Para otimizar o ensino de Lógica de Programação no ensino médio, é crucial investigar os desafios particulares enfrentados por esses alunos. Esses estudos aprofundados permitem a formulação de estratégias pedagógicas mais eficazes. Da mesma forma, pesquisas que identificam padrões de dificuldades entre os novatos em programação contribuem para a personalização do ensino, que permite abordagens mais adaptativas. Essas descobertas informam intervenções educacionais específicas para superar obstáculos comuns e promover uma experiência de aprendizado mais eficaz e envolvente.

No artigo “Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação”. analisado destaca a hipótese fundamental de que a identificação prévia das dificuldades no processo de ensino-aprendizagem de Lógica de Programação, seja no contexto acadêmico ou técnico, é essencial. Posteriormente, busca-se desenvolver a estratégia metodológica mais eficaz para potencializar esse aprendizado. Nesse sentido, Arimoto e Weldrey (2019) propuseram a aplicação de um método de pesquisa (*survey*) para coletar e analisar dados relacionados à aprendizagem de Lógica de Programação. O principal resultado obtido destacou a relação entre o conhecimento prévio e o desempenho em disciplinas que envolvem programação. Essas investigações servirão como referência para o subsequente levantamento e análise de dados do Sistema *Web* para Apoio ao Processo de Ensino e Aprendizado aplicado aos alunos do Novo Ensino Médio.

Bigolin et al. (2020) trazem em seu trabalho, “Metodologias Ativas de Aprendizagem: um relato de experiência nas disciplinas de programação e estrutura de dados”, uma experimentação do uso de metodologias ativas para a melhoria do processo de ensino-aprendizagem de Lógica de Programação e Estrutura de Dados

nas séries iniciais do curso de Bacharelado em Sistemas de Informação. O compartilhamento de tais experiências servirá como parâmetro comparativo entre o uso de metodologias tradicionais de ensino e o uso de metodologias em que o aluno passa a ter o papel de protagonista, além de permitir abordagens de conteúdo personalizadas e automatizadas. Com a implantação desse sistema, houve uma melhora significativa no índice de aprovações na disciplina de Lógica de Programação.

Em seu artigo “O Ensino de Programação na Educação Básica: Uma Revisão da Literatura”, Souza et al. (2021) aborda as diversas dificuldades enfrentadas no ensino de programação, destaca-se aspectos como os métodos tradicionais de ensino, a complexidade da Lógica de Programação e os desafios relacionados à infraestrutura educacional. O trabalho ressalta a falta de motivação dos estudantes e a urgência de adotar novas metodologias e estratégias de ensino, alinhadas às transformações nos paradigmas educacionais.

A motivação central do trabalho concentra-se na identificação dos objetivos do ensino de programação na educação básica, assim como nas técnicas e ferramentas frequentemente utilizadas para atingir esses objetivos. A estrutura delineada do artigo inclui seções que abordam trabalhos relacionados, ferramentas e estratégias de ensino, método empregado na revisão da literatura, resultados obtidos e considerações finais.

A crescente integração da tecnologia na educação tem proporcionado avanços significativos, especialmente no desenvolvimento de sistemas de recomendação voltados para ambientes educacionais. Esta vertente de pesquisa busca explorar e compreender a aplicação desses sistemas no contexto educacional que visa aprimorar a experiência de aprendizagem dos alunos. No âmbito desse avanço, destaca-se a personalização do ensino, onde os sistemas de recomendação desempenham um papel crucial. Estes buscam não apenas oferecer conteúdo relevante, mas também adaptá-lo de maneira específica ao perfil de cada aluno, considerando suas necessidades individuais. Neste cenário, investigações que exploram a eficácia e a implementação desses sistemas ganham relevância que contribui para a evolução das práticas pedagógicas e para a promoção de um aprendizado mais personalizado e eficiente.

No trabalho “Desenvolvimento de sistemas *Web* orientado a reuso com *Python*,

*Django e Bootstrap*” proposto por (Santiago, et al, 2020) apresenta uma contribuição significativa. Por meio de um passo-a-passo abrangente, o trabalho explora a utilização dos *frameworks Django e Bootstrap* em conjunto com a linguagem de programação *Python* para criar uma solução *web* completa.

Ao proporcionar extenso reuso de arquiteturas e bibliotecas, os *frameworks Django e Bootstrap* emergem como facilitadores essenciais no desenvolvimento ágil e eficiente de sistemas *web*. Essas ferramentas permitem não apenas reduzir o tempo de desenvolvimento, mas também simplificar a complexidade envolvida na criação de aplicações *web* robustas e eficazes.

O trabalho apresentado demonstra como os *frameworks Django e Bootstrap* podem ser integrados de forma harmoniosa para desenvolver um sistema *web* orientado a reuso. Desde a instalação dos *softwares* necessários até a publicação em produção, o passo-a-passo detalhado oferece *insights* valiosos para desenvolvedores interessados em criar soluções *web* de alta qualidade.

Embora o trabalho de Santiago et al. (2020) apresente uma valiosa contribuição ao demonstrar a efetividade dos *frameworks Django e Bootstrap* no desenvolvimento *web*, o sistema proposto neste estudo se distingue por seu foco específico no ensino de Lógica de Programação e na adequação às necessidades do Novo Ensino Médio Brasileiro.

O sistema vai além da aplicação dos *frameworks*, adaptando-os para atender às demandas específicas do ensino de lógica de programação o que inclui a criação de interfaces interativas, intuitivas e a personalização do conteúdo para diferentes níveis de conhecimento.

O sistema segue uma metodologia pedagógica alinhada às diretrizes da BNCC, que garante que o conteúdo seja relevante, engajador e eficaz na formação dos alunos, preparando-os para os desafios do mercado de trabalho atual.

Além de oferecer recursos complementares que enriquecem o aprendizado, como tutoriais em vídeo, *links* de sites de apoio, fóruns de discussão etc. Isso garante que os alunos tenham acesso a diferentes ferramentas de apoio para aprofundar seus conhecimentos e sanar suas dúvidas. O sistema foi construído com base em tecnologias abertas e livres, o que garante sua sustentabilidade a longo prazo e facilita

a sua manutenção e atualização. Isso significa que o sistema poderá ser aprimorado e adaptado às novas demandas.

No contexto do presente estudo fornece uma base sólida e relevante para o desenvolvimento do sistema proposto para o ensino de Lógica de Programação. Ao adotar abordagens e técnicas semelhantes, é possível obter benefícios similares em termos de eficiência, escalabilidade e qualidade da experiência do usuário.

No trabalho “*Web Based Algorithm Exercise and Assessment Management System for Computer Programming Students: AlgoBug*” proposto por (Murat Karabulut; Islam Mayda, 2020) propõe a construção de um sistema *web* denominado *AlgoBug*, projetado para fornecer uma plataforma completa de exercícios e avaliações em Lógica de Programação. O *AlgoBug* permitirá que os usuários pratiquem questões de programação em diversos níveis de dificuldade, organizadas de forma progressiva para facilitar o aprendizado. Além disso, as soluções submetidas pelos usuários serão avaliadas por outros membros da comunidade, proporcionando feedback detalhado e auxiliando no processo de aprendizagem.

Uma das características distintivas do *AlgoBug* é sua abordagem de aprendizagem colaborativa, onde usuários mais experientes terão a oportunidade de atuar como mentores para orientar e apoiar aqueles que iniciam o estudo da Lógica de Programação. Essa interação entre os membros da comunidade não apenas promoverá a prática intensiva, mas também estimulará a troca de conhecimentos e o desenvolvimento de habilidades entre os participantes.

Acredita-se que o *AlgoBug* possa desempenhar um papel significativo na superação das dificuldades encontradas no ensino de Lógica de Programação, que fornece aos alunos uma plataforma interativa e acessível para a prática e aprimoramento de suas habilidades nesse campo. Além disso, ao disponibilizar recursos em ambiente *web*, o *AlgoBug* visa tornar o aprendizado de programação mais acessível e inclusivo para um público mais amplo.

Embora o trabalho de Karabulut e Mayda (2020) apresente uma proposta interessante para o ensino de Lógica de Programação, o sistema proposto neste estudo se distingue por diversos aspectos, por exemplo, o sistema fornece ferramentas para monitorar o progresso individual dos alunos, o que permite que

professores acompanhem o desempenho de cada estudante e identifiquem pontos que precisam de mais atenção. Isso facilita a personalização do aprendizado e a intervenção oportuna quando necessário.

Essa abordagem garante que cada aluno receba o suporte adequado de acordo com suas necessidades específicas. Além disso, ao direcionar os testes e atividades com base no conhecimento prévio de cada aluno, o sistema evita que os alunos revisem conceitos já dominados, maximizando o tempo de aprendizado e proporcione uma experiência mais eficaz e direcionada.

No trabalho “Sistema de Recomendação Baseado em Estilos de Aprendizagem em Sala de Aula” proposto por (Cazella et al., 2009) visa desenvolver um sistema de recomendação para personalizar a sugestão de objetos de aprendizagem, leva-se em consideração as preferências individuais de cada aluno em relação a determinados itens de aprendizagem. A recomendação, nesse contexto, está focada nas competências a serem desenvolvidas nas disciplinas, alinha-se com as preferências manifestadas pelos alunos.

Para a implementação dessa proposta, os autores desenvolveram uma solução abrangente que envolve uma página *web*, um simulador e um programa de perguntas e respostas pré-programadas. Esses elementos compõem o sistema, que permite uma experiência integrada aos usuários. A estratégia não se limita apenas à recomendação de conteúdos, mas também incorpora ferramentas interativas para melhorar a interação e a personalização do processo de aprendizagem.

Na abordagem contemporânea da educação, a inserção de tecnologias inovadoras desempenha um papel crucial na promoção do aprendizado. No contexto do ensino de Lógica de Programação no novo ensino médio, a proposta deste trabalho se distingue por sua ênfase na personalização da experiência educacional e simplicidade do ambiente. Ao contrário de abordagens convencionais, que muitas vezes oferecem soluções genéricas e complexas, o sistema de recomendação baseado em itens é meticulosamente projetado para se alinhar às necessidades dinâmicas e diversificadas dos estudantes do novo ensino médio.

Enquanto o trabalho anterior aborda recomendações de maneira mais abrangente, a proposta deste trabalho destaca-se por sua especificidade e foco na

sugestão de vídeos e material de apoio relevantes, alinhados aos diferentes níveis de conhecimento do usuário, essa metodologia baseia-se na realização de testes pré-definidos pelo professor e na categorização dos alunos em diferentes níveis de habilidade em programação, como iniciante, intermediário e avançado, inspirado nas abordagens de autores, como Matthes (2016), autor do livro "*Python Crash Course*". Essa segmentação permite uma personalização mais precisa do conteúdo sugerido, tornando a experiência educacional mais envolvente e adaptada às nuances do Novo Ensino médio atual. Essa diferenciação, centrada na personalização e contextualização, representa uma contribuição significativa para o avanço da tecnologia educacional e o aprimoramento do processo de ensino da Lógica de Programação no cenário do Novo Ensino Médio.

## 5. MÉTODOS E MATERIAIS

Este capítulo descreve a Metodologia e as Ferramentas Tecnológicas usadas no desenvolvimento do projeto, destacando a importância da metodologia na orientação do processo e na garantia de práticas eficientes. A integração dessas ferramentas com a metodologia é essencial para criar soluções inovadoras, robustas e alinhadas aos objetivos do projeto, resultando em um sistema escalável e de alta qualidade.

### 5.1. Descrição da Pesquisa

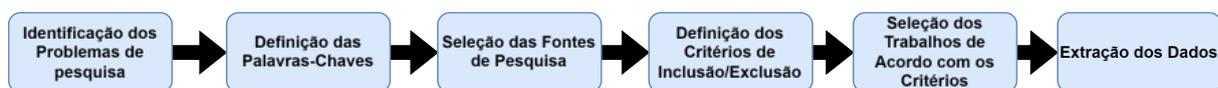
De acordo com Prodanov e Freitas (2013), as pesquisas podem ser classificadas em diferentes tipos, de acordo com suas características específicas. Com base nessa premissa, este trabalho caracteriza-se como uma pesquisa de natureza aplicada, cujo objetivo é oferecer uma solução prática e imediata para um problema específico. No que diz respeito aos procedimentos adotados, trata-se de uma pesquisa experimental, que utiliza o estudo de caso como método para aplicar o conhecimento adquirido na solução de problemas sociais (YIN, 2001).

Considerando a abordagem do problema, esta pesquisa adota um enfoque qualitativo, por meio de um estudo de caso envolvendo alunos do Novo Ensino Médio, utilizando a aplicação proposta em um ambiente controlado. Dados quantitativos também são apresentados na seção de resultados, com o intuito de complementar a análise e discussão sobre a utilização da aplicação.

Foi realizada uma revisão bibliográfica sobre pesquisas relacionadas ao tema proposto, utilizou-se um protocolo de mapeamento sistemático, conforme mostra a Figura 5. A coleta de dados ocorreu de 18 de janeiro a 22 de março de 2022 e continuou durante todo o desenvolvimento da pesquisa, abrangendo diversas bases de dados como Portal Capes, Google Acadêmico, Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), Comissão Especial de Informática na Educação(CEIE), Sociedade Brasileira de Computação (SBC), *ScienceDirect*, *Association For Computing Machinery (ACM) Digital Library* e *Institute of Electrical and Electronic Engineers (IEEE Xplore)*. A plataforma *IEEE Xplore* foi destacada pela sua relevância e qualidade no contexto do ensino-aprendizagem de Lógica de Programação. A inclusão de diversas fontes assegurou uma revisão bibliográfica abrangente e

aprofundada.

Figura 5 - Etapas do Mapeamento Sistemático



Fonte: Autor

Em seguida, a etapa de levantamento de requisitos que teve a finalidade de obter as informações que identificam as características essenciais para o desenvolvimento da solução proposta, uma vez que é importante conhecer o ponto de vista do usuário da aplicação.

Na etapa de modelagem do sistema são propostas representações gráficas, baseadas nos conceitos de Engenharia de Softwares e UML, possibilitando a elaboração de diagramas distintos e casos de uso para exibir funcionalidades e interações da aplicação.

Com a conclusão da modelagem, a próxima etapa é o desenvolvimento da aplicação, levando em conta os resultados obtidos através do levantamento de requisitos. Nesta etapa fez-se uso de recursos de *software*, através da plataforma *Visual Studio Code* e recursos de hardware, por meio de computadores *desktops* e um servidor local utilizado para implementação.

Em seguida, foram realizados testes preliminares, com alguns alunos do Novo Ensino Médio. A principal finalidade desta etapa é verificar variáveis específicos relativos a funcionalidades e usabilidade da ferramenta, para então diagnosticar possíveis erros ou melhorias a serem implementadas.

Na etapa de validação da ferramenta, um estudo de caso é proposto com a versão final da aplicação, bem como serão descritos os procedimentos adotados e pré-estabelecidos para realização dos testes finais, identificando cada passo a ser seguido.

Por fim, a etapa de análise dos resultados reuniu todos os dados e informações obtidas na fase de testes preliminares e finais. Aqui serão evidenciados de que maneira estes resultados serão organizados, visando facilitar o entendimento e

discussões sobre eles.

## **5.2. Procedimentos Metodológicos**

Nesta seção são detalhadas as técnicas utilizadas para realização das etapas definidas.

### **5.2.1. Análise de requisitos**

A análise de requisitos é uma etapa fundamental que visa identificar, documentar e validar as necessidades dos usuários e os objetivos do sistema, garantindo assim que o produto atenda às expectativas e exigências estabelecidas.

Inicialmente, conforme Pressman (2021), realizou-se o levantamento de requisitos que envolveu diversas fontes de informação, como professores da área de programação que fazem parte do corpo docente do SENAI/CEDAM/Belém, alunos do Novo Ensino Médio – Habilitação Técnica em Redes de Computadores e especialistas em tecnologia (analista desenvolvedores de *software*) e educação (gerente de educação do SENAI/PARÁ). Utilizou-se técnicas como entrevistas e questionários para capturar os requisitos funcionais e não funcionais.

Após o levantamento inicial, seguiu-se com a análise e priorização dos requisitos identificados, conforme sugerido por *Sommerville* (2020). Nesta etapa, classificou-se os requisitos de acordo com sua importância e viabilidade, também se levou em consideração os recursos disponíveis, as restrições técnicas e as necessidades específicas dos usuários finais. Essa análise permitiu definir uma lista de requisitos essenciais para a fase inicial do desenvolvimento do sistema.

Por fim, conforme orientado por *Sommerville* (2020), conduziu-se um processo de validação de requisitos, que envolveu revisões, prototipagem e *feedback* dos usuários. Essa etapa permitiu garantir que os requisitos especificados estivessem corretos, completos e consistentes com as expectativas dos usuários finais. Qualquer ambiguidade identificada foi abordada e corrigida antes do início efetivo do desenvolvimento do sistema.

### **5.2.2. Análise de viabilidade**

Inicialmente, realizou-se uma análise dos requisitos técnicos necessários para a construção e operação do sistema *web*. Isso incluiu a avaliação da infraestrutura de

*hardware* e *software* disponível na instituição SENAI/CEDAM, a compatibilidade com os padrões e tecnologias existentes, bem como a disponibilidade de recursos técnicos para o desenvolvimento e manutenção contínua do sistema. Verificou-se que a infraestrutura existente era adequada tanto para o desenvolvimento da ferramenta quanto para a implantação nos laboratórios e os testes posteriores com os alunos.

No que diz respeito à viabilidade econômica, analisou-se os custos associados ao desenvolvimento, implementação e manutenção do sistema. Isso inclui não apenas os custos iniciais de desenvolvimento, mas também os custos operacionais contínuos e o retorno esperado sobre o investimento em termos de melhoria da qualidade educacional e eficiência no processo de ensino-aprendizagem.

Por fim, avaliou-se a viabilidade cronológica do projeto, o tempo necessário para desenvolver, testar e implementar o sistema dentro dos prazos estabelecidos. Identificou-se os principais marcos de tempo e as fases do projeto, bem como possíveis áreas de risco e estratégias para mitigar atrasos potenciais.

### **5.2.3. Modelagem do Sistema**

A escolha da metodologia de desenvolvimento ICONIX foi embasada em trabalhos de autores como *Doug Rosenberg* e *Matt Stephens*, autores de *Use Case Driven Object Modeling with UML: Theory and Practice* (ROSENBERG; STEPHENS, 2007), que introduzem e detalham a abordagem do ICONIX para o desenvolvimento de *software* baseado em casos de uso. Esta metodologia é reconhecida por sua eficácia na captura e representação de requisitos de sistema de forma clara e concisa, o que a torna adequada para projetos de desenvolvimento de *software* educacional.

Além disso, autores como *Grady Booch*, autor de *Object-Oriented Analysis and Design with Applications* (BOOCH et al., 2007), destacam a importância da modelagem visual na engenharia de *software*, argumento que ela facilita a compreensão e comunicação dos requisitos e arquitetura do sistema entre os membros da equipe. Nesse sentido, a metodologia ICONIX, que enfatiza a criação de modelos visuais como parte integrante do processo de desenvolvimento, foi escolhida como a abordagem principal para a modelagem do sistema.

Esses autores e obras contribuíram significativamente para a escolha da

metodologia de desenvolvimento ICONIX para o projeto do sistema *web* de ensino de lógica de programação. Sua ênfase na clareza, concisão e comunicação eficaz dos requisitos e arquitetura do sistema alinha-se perfeitamente com os objetivos do projeto.

Para criação dos diagramas propostos foi utilizado o software *Draw.io* (DRAW.IO, 2019).

#### 5.2.4. Escolha de Tecnologia

Para alcançar os objetivos propostos neste projeto, foi essencial realizar escolhas satisfatórias em relação às tecnologias a serem empregadas. A seleção adequada das ferramentas de desenvolvimento é crucial para garantir a eficiência, escalabilidade e manutenibilidade do sistema.

A primeira etapa desse processo foi a escolha das tecnologias fundamentais para o desenvolvimento do sistema, que incluem *Python*, *Django* e *Bootstrap*. Optou-se pelo *Python* como linguagem de programação principal devido à sua versatilidade, simplicidade e robustez, torna-se uma escolha ideal para o desenvolvimento de aplicações *web* (Van Rossum, 2020).

Em seguida, selecionou-se o *framework Django* devido à sua escalabilidade, segurança e eficiência no desenvolvimento de aplicações *web* complexas (Holovaty, A. Kaplan-Moss, J. 2016). A estrutura MTV (*Model-Template-View*) do *Django* proporciona uma organização clara e modular do código que facilita a manutenção e extensão do sistema.

Quanto à escolha do *Bootstrap*, decidiu-se utilizá-lo devido à sua vasta coleção de componentes e estilos “pré-projetados”, que permitem uma rápida prototipagem e uma experiência de usuário consistente em diferentes dispositivos e navegadores.

No processo de justificar as escolhas tecnológicas, analisou-se alternativas disponíveis no mercado. Linguagens de programação como PHP foram descartadas devido à preferência pela sintaxe simples e expressiva do *Python* (LUTZ, 2013). *Frameworks* como *Ruby on Rails* também foram considerados, porém, optou-se pelo *Django* devido à sua integração nativa com o *Python* e sua robustez comprovada em projetos de grande escala (HOLLER et al., 2016). Embora *JavaScript* seja

amplamente utilizado no desenvolvimento web, especialmente no *frontend*, optou-se por não o utilizar como linguagem principal no *backend*. Isso se deve a algumas limitações do *JavaScript* em termos de processamento intensivo de dados e operações de *back-end* complexas (FLANAGAN, 2011), além da maturidade e estabilidade que o *Python* oferece para essas tarefas (BEAZLEY; JONES, 2013). *Python*, com sua versatilidade e ampla gama de bibliotecas, permite uma implementação mais direta e eficiente das funcionalidades necessárias para o sistema (VANDERPLAS, 2016). O Django, em particular, proporciona uma estrutura robusta e segura, facilitando o desenvolvimento de aplicações web de grande porte, enquanto o uso de *JavaScript* foi restrito ao *frontend*, onde sua capacidade de criar interfaces dinâmicas e interativas é mais bem aproveitada (RESIG; BIBEULT, 2013).

Além disso, abordou-se as vantagens do *Bootstrap* em relação a outros *frameworks* de *front-end*, como *Foundation* e *Materialize*. Ele foi escolhido devido à sua ampla adoção pela comunidade de desenvolvimento web, o que facilita a disponibilidade de recursos e suporte, além de sua integração perfeita com o *Django*.

Finalmente, se descreve como as tecnologias escolhidas foram implementadas e integradas no processo de desenvolvimento do sistema. Explora-se os desafios encontrados durante a implementação e como foram superados, bem como os benefícios obtidos com a escolha dessas tecnologias para o sucesso do projeto.

### **5.2.5. Desenvolvimento da Ferramenta**

O processo de desenvolvimento foi conduzido segundo o que estabelece Pressman (2011), iniciou-se com uma análise detalhada dos requisitos necessários, como mencionado anteriormente, resultando em um plano de desenvolvimento que apresentava as funcionalidades e o escopo do sistema.

A arquitetura do sistema foi cuidadosamente planejada para ser modular e escalável, seguindo as diretrizes de Sommerville (2016). Essa abordagem permitiu definir a estrutura e organização dos principais componentes, como o banco de dados, a camada de lógica de negócios e a interface do usuário.

Durante o desenvolvimento, foram criadas as *views* (visões), que são responsáveis por processar as requisições dos usuários e retornar as respostas adequadas. As *views* contém toda a lógica de negócios do sistema web, determinando

o que será exibido aos usuários com base nas informações fornecidas.

Para definir a aparência visual das páginas da *web*, foram utilizados *templates*. Os *templates* em *Django* são arquivos HTML que contêm marcações especiais para inclusão dinâmica de conteúdo. Eles ajudam a separar o código HTML da lógica de apresentação o que torna o desenvolvimento mais organizado e flexível. Os *templates* podem incluir variáveis, *tags* de controle de fluxo e filtros para personalizar a renderização das páginas.

As URLs desempenham um papel fundamental no processo de desenvolvimento. As URLs em *Django* são mapeamentos entre as requisições dos usuários e as *views* correspondentes que devem processá-las. Elas são definidas no arquivo de configuração de URLs (*urls.py*) e especificam padrões de URL que serão correspondidos às *views* apropriadas. Esse sistema de roteamento de URLs ajuda a organizar e manter a estrutura da aplicação *web* de forma clara e concisa que facilita a navegação dos usuários e a manutenção do código.

Além disso, o desenvolvimento foi facilitado pelo uso do *framework Django*, que oferece uma variedade de funcionalidades e ferramentas para desenvolvimento rápido e seguro de aplicações *web*. Com *Django*, foi possível criar a estrutura básica do projeto, gerenciar a autenticação de usuários, lidar com formulários e automatizar tarefas comuns, entre outras coisas, um exemplo de aplicação no sistema foi o gerenciamento de formulários (*forms*) que são uma parte essencial do desenvolvimento de aplicações *web*, pois facilitou a coleta e validação de dados submetidos pelos usuários através de páginas da *web*. Esses formulários em *Django* foram construídos usando classes *Python* que herdaram funcionalidades de uma classe base chamada ***django.forms.Form***.

Em suma, a utilização de *views*, *templates*, *forms* e URLs no desenvolvimento do sistema contribuiu significativamente para a robustez, eficiência e usabilidade da plataforma de ensino de Lógica de Programação no novo currículo do Novo Ensino Médio. Durante o desenvolvimento, uma abordagem iterativa foi adotada, com entregas incrementais de funcionalidades. Testes foram realizados em cada etapa do processo para garantir a qualidade e confiabilidade do sistema, conforme proposto por Pressman (2011). Após a conclusão do desenvolvimento, o sistema foi implantado em um ambiente de produção, onde o *feedback* dos usuários foi coletado e analisado. Isso

permitiu ajustes e melhorias contínuas no sistema, visando sempre aprimorar a experiência de aprendizado dos usuários.

### 5.2.6. Teste

O processo metodológico dos testes utilizado pode ser visto na Figura 6, que é fundamental para garantir que o sistema seja intuitivo, eficiente e satisfatório para os usuários. Também será explorado os critérios de sucesso estabelecidos para determinar se o sistema atende às necessidades e expectativas dos usuários. Ao final, espera-se fornecer uma compreensão abrangente da abordagem adotada para avaliar a usabilidade do sistema, destacando sua importância na entrega de uma experiência de usuário positiva e satisfatória.

Figura 6 - Demonstrativo do processo de acontecimentos dos Testes.



Fonte: Autor

A preparação do laboratório é uma etapa crucial que assegura a efetividade e precisão dos resultados. Inicialmente, o ambiente deve ser configurado com máquinas suficientes para todos os participantes, garantindo que cada uma esteja equipada com as especificações de *hardware* e *software* necessárias para o funcionamento adequado do sistema. Os computadores devem ter navegadores atualizados e quaisquer *plug-ins* ou extensões específicas requeridas pelo sistema *web*. A conexão à internet deve ser estável e de alta velocidade para evitar interrupções durante os testes. Além disso, é fundamental que o ambiente seja propício ao foco e à concentração, minimizando ruídos e distrações. Serão disponibilizadas contas de acesso pré-configuradas para os participantes, bem como instruções claras e detalhadas sobre o procedimento dos testes.

Para a realização dos testes foram selecionados, alunos (as) que estejam cursando o Novo Ensino Médio, dos itinerários formativos que a escola disponibiliza, com conhecimento básico do assunto Lógica de Programação. Os alunos selecionados devem estar matriculados e frequentando a escola no período

correspondente aos testes.

Preferencialmente, mas não obrigatoriamente os alunos deverão possuir habilidades quanto ao manuseio de computadores, mas caso contrário estes também poderão participar dos testes, recebendo anteriormente instruções de uso do equipamento. A seleção dos participantes foi feita após consulta à coordenação da instituição, para verificar a disponibilidade dos alunos.

Dentro da metodologia de testes desenvolvida para avaliar o Sistema *Web* voltado ao ensino-aprendizado de lógica de programação, um passo crucial é a elaboração de questões de Lógica de Programação, por professores da área de tecnologia da informação, com base em um documento padronizado chamado de Matriz Curricular, desenvolvido pela instituição. Este procedimento garante que as questões formuladas sejam consistentes com os padrões educacionais estabelecidos, permitindo uma avaliação precisa e relevante das habilidades dos alunos. Esse passo não apenas assegura a qualidade das questões utilizadas no teste, mas também alinha o processo de avaliação às diretrizes pedagógicas vigentes, proporcionando um ambiente de aprendizagem mais estruturado e direcionado.

Como parte da metodologia de testes criada para avaliar o sistema, a realização dos testes com os alunos é um passo fundamental. Este processo envolve a aplicação prática das questões de Lógica de Programação previamente elaboradas, em um ambiente controlado, onde os alunos interagem diretamente com o sistema. A participação dos alunos permite a validação do conteúdo pedagógico e a avaliação da usabilidade e eficácia do sistema em termos de facilitar o aprendizado. Durante esses testes, dados são coletados para analisar o desempenho dos alunos e a adequação das funcionalidades do sistema às necessidades educacionais, fornecendo *insights* valiosos para ajustes e melhorias contínuas na plataforma. Essa abordagem garante que o sistema seja rigorosamente avaliado e otimizado antes de sua implantação definitiva.

Para a coleta de dados foi criado um questionário de usabilidade aplicado aos alunos participantes da pesquisa, que se enquadravam no perfil definido. O questionário, composto por dez questões, foi respondido pelos alunos após utilizarem a aplicação. Cada participante recebeu orientações detalhadas sobre como preencher

o questionário, além de ser informado sobre a importância de sua contribuição para a pesquisa.

A criação do questionário foi baseada no modelo SUS (*System Usability Scale*), utilizando a escala *Likert* para avaliar tanto a direção (positiva ou negativa) quanto a intensidade da satisfação dos usuários em relação ao sistema (LIKERT, 1932). Conforme mencionado por Costa (2011), uma das vantagens dessa técnica é sua facilidade de aplicação e a possibilidade de adaptar as métricas conforme necessário, sendo amplamente utilizada em diversos estudos.

As questões do questionário foram desenvolvidas a partir de adaptações das Heurísticas de Avaliação para projetos de desenvolvimento de aplicações *web* educacionais. As perguntas foram estruturadas para avaliar os aspectos de funcionalidade e interatividade proporcionados pela utilização da aplicação (NIELSEN, 1993; BERTINI, 2009; SALAZAR, 2012).

A etapa de análise dos resultados coletados seguiu os procedimentos descritos anteriormente, com foco na avaliação dos dados obtidos por meio do questionário de usabilidade SUS (*System Usability Scale*). Esse questionário, composto por dez questões baseadas em uma escala *Likert*, permitiu uma visão abrangente sobre os critérios de usabilidade da aplicação proposta. Os dados coletados serão apresentados na forma de gráficos e tabelas, que estão detalhados no capítulo que trata sobre os resultados.

## **6. APRESENTAÇÃO DO SOFTWARE**

O presente capítulo descreve o processo de construção da aplicação, desde o levantamento dos requisitos até a definição das interfaces do usuário. São apresentados os mecanismos que permitem a interação entre o sistema e o usuário, garantindo a usabilidade e a eficiência da aplicação.

### **6.1. Resultados da Modelagem utilizando metodologia ICONIX**

O levantamento de requisitos foi realizado em janeiro de 2023, utilizando técnicas como entrevistas semiestruturadas e observação. Essa etapa ocorreu no SENAI/CEDAM-Belém, com a participação de professores de Tecnologia da Informação e alunos do Novo Ensino Médio. Durante o processo, os professores descreveram os métodos que utilizavam para ensinar Lógica de Programação, identificaram suas principais dificuldades e explicaram como as avaliações eram conduzidas, entre outras informações relevantes.

Nas aulas, os professores inicialmente expõem o conteúdo para a turma utilizando slides. Em seguida, definem uma atividade prática relacionada ao conteúdo ministrado, indicando uma plataforma específica para a construção dos algoritmos. Durante a execução da atividade, os professores oferecem apoio individual aos alunos, sempre que possível.

A avaliação do conhecimento teórico dos alunos era realizada por meio de métodos tradicionais. Os estudantes recebiam uma prova impressa contendo questões da unidade curricular em questão. Para as perguntas de múltipla escolha, as respostas eram marcadas com caneta esferográfica de cor azul ou preta. Nas questões que exigiam a construção de algoritmos, era apresentada uma problemática específica, sendo solicitado que os alunos desenvolvessem a solução em uma folha separada, evitando, assim, o uso de ferramentas de inteligência artificial para resolver as tarefas.

A metodologia ICONIX possibilitou o desenvolvimento ágil, a curto prazo, do sistema *web*, ao empregar a técnica de entrevista para coletar os requisitos, como descrito inicialmente. Isso contribuiu para a rápida implementação das fases do processo, identificou-se requisitos prioritários (ver Tabela 3), atores e casos de uso.

Essa abordagem também envolveu a descrição detalhada desses

elementos noplanejamento, conforme as diretrizes da metodologia.

Tabela 3 - Requisitos funcionais do Sistema Web de Ensino de Lógica de Programação.

ID	Requisito Funcional	Descrição
RF01	Autenticação de Usuário	Permitir que alunos e professores acessem a plataforma usando nome de usuário e senha.
RF02	Gestão de Perfis	Permitir a criação e gerenciamento de perfis de usuário (aluno e professor), com funcionalidades administrativas para professores.
RF03	Criação e Edição de Testes (Quiz)	Professores devem poder criar, editar e deletar testes em formato de quiz, configurando perguntas e alternativas de resposta.
RF04	Realização de Testes (Quiz)	Alunos devem poder acessar e responder aos testes disponíveis, com registro de respostas e fornecimento de <i>feedback</i> imediato sobre o desempenho.
RF05	Sugestão de Vídeos Educacionais	Após a conclusão do teste, exibir uma sugestão de vídeo relacionado ao conteúdo abordado, postado previamente por um professor, alinhado ao desempenho do aluno.
RF06	<i>Feedback</i> e Avaliação	Gerar relatórios de desempenho para o aluno após a conclusão do quiz, e permitir que professores monitorem o progresso dos alunos.
RF07	Banco de Questões	Professores devem poder criar, organizar e reutilizar um banco de questões para diferentes <i>quizzes</i> .
RF08	Controle de Acesso ao Conteúdo	Restringir o acesso a determinados conteúdos e vídeos apenas para alunos que completaram testes específicos.
RF09	Interface de Usuário Intuitiva	Apresentar uma interface de usuário clara e intuitiva, facilitando a navegação e uso tanto para alunos quanto para professores.

Fonte: Autor

Com base nos requisitos levantados, foi realizado a modelagem do sistema, que incluiu a elaboração de protótipos da interface gráfica do sistema *web* e a definição do banco de dados com seus modelos correspondentes. No *design* foram desenvolvidos os seguintes diagramas: de casos de uso (ver Figura 7), modelo de domínio (ver Figura 8) e diagrama de classe (ver Figura 9), para representar a modelagem e a estrutura do sistema.

O diagrama de casos de uso, ilustrado na Figura 7, é uma representação gráfica dos perfis que o sistema *web* possui e suas funcionalidades. Destaca-se a interação entre os atores principais - Administrador, Professor e Aluno – cada um desempenhando funções específicas no contexto do sistema.

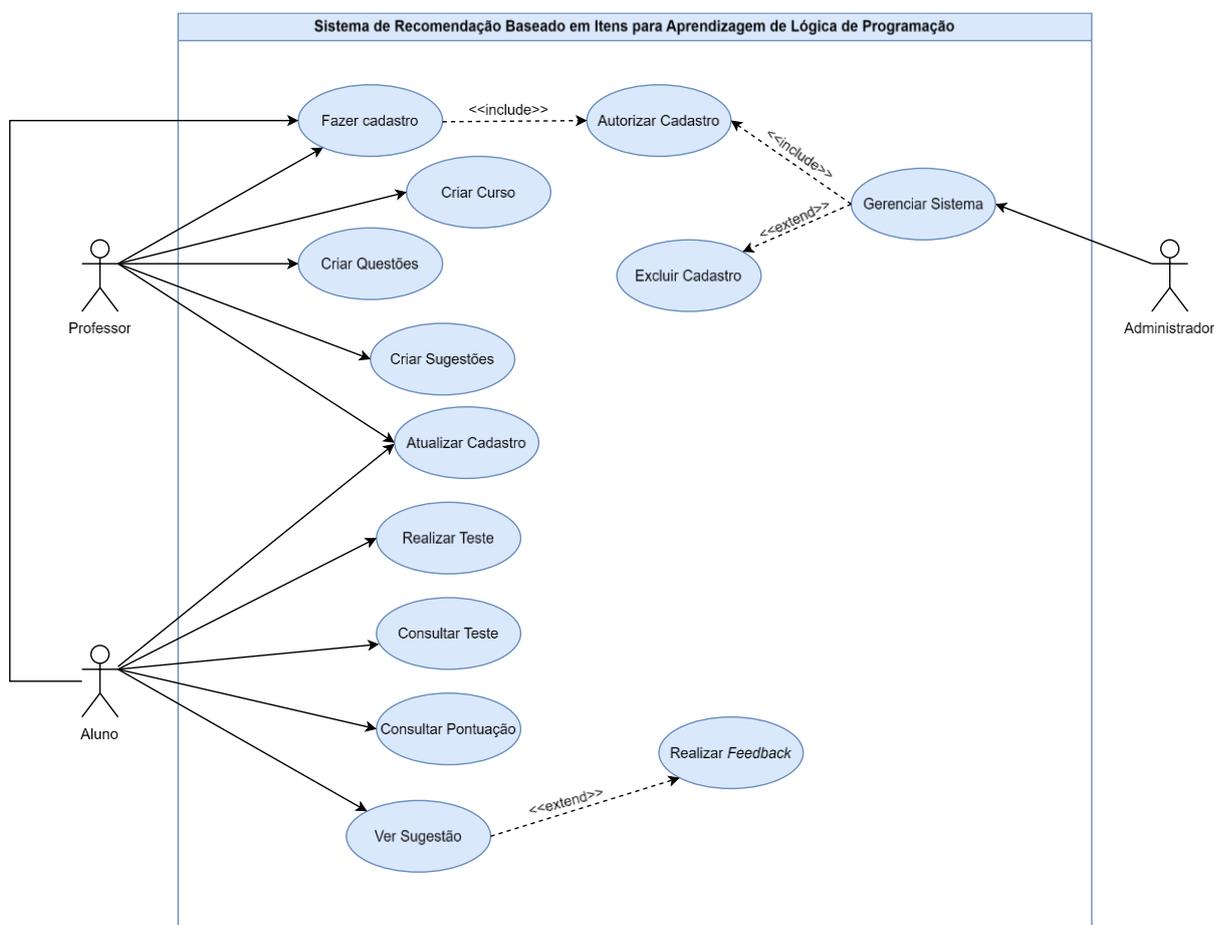
O administrador, identificado como o ator "**Administrador**", possui uma função de gerenciamento abrangente. Através dos casos de uso "Gerenciar Sistema", "**Autorizar Cadastro**" e "**Excluir Cadastro**", o administrador pode controlar o acesso, realizar a gestão de usuários, e obter *insights* detalhados sobre o uso e desempenho global do sistema.

O ator "**Professor**" desempenha um papel ativo, sendo capaz de contribuir significativamente para o sistema. Por meio dos casos de uso "**Criar Cursos**", "**Criar Questões**" e "**Criar Sugestões**" e "**Visualizar Relatórios de Desempenho**", o professor pode adicionar novo conteúdo, como questões e sugestão de conteúdo, e monitorar o desempenho dos alunos por meio de relatórios.

Os alunos, representados pelo ator "**Aluno**", são centrais para a dinâmica do sistema. Os casos de uso "**Realizar Teste**" e "**Ver Sugestões**" capacitam os alunos a interagirem ativamente com o sistema. Eles podem responder às questões propostas e, com base em seu desempenho, receber sugestões de conteúdo personalizadas que se alinham ao seu progresso e conhecimento.

Este diagrama de casos de uso proporciona uma visão clara e estruturada das interações entre os atores e as principais funcionalidades do sistema. Ele serve como um guia fundamental para o desenvolvimento e compreensão do Sistema *Web* de Ensino de Lógica de Programação, que fornece uma representação visual da arquitetura e das operações essenciais.

Figura 7 - Diagrama de Caso de Uso - Visão Geral do Sistema.



Fonte: Autor

O modelo de domínio representado pela Figura 8 para o Sistema *Web* de Ensino de lógica de programação é uma representação das entidades fundamentais que dão vida a essa plataforma educacional. No epicentro desse ecossistema, destacam-se entidades: Administrador, Professor, Aluno, Questão, Teste, Pontuação, Sugestão, Curso e *Feedback*.

O Administrador desempenha um papel central, assegurando a integridade global do sistema ao supervisionar atividades, administrar recursos e gerenciar usuários. Os Professores, por sua vez, são os arquitetos da experiência educacional, podendo cadastrar questões desafiadoras, elaborar testes abrangentes e atribuir sugestões de conteúdos personalizadas com base no desempenho dos alunos.

Os Alunos constituem o cerne do aprendizado ativo na plataforma, interagindo com questões, realizando testes e recebendo pontuações. As Questões representam desafios cuidadosamente elaborados para avaliar o conhecimento em Lógica de

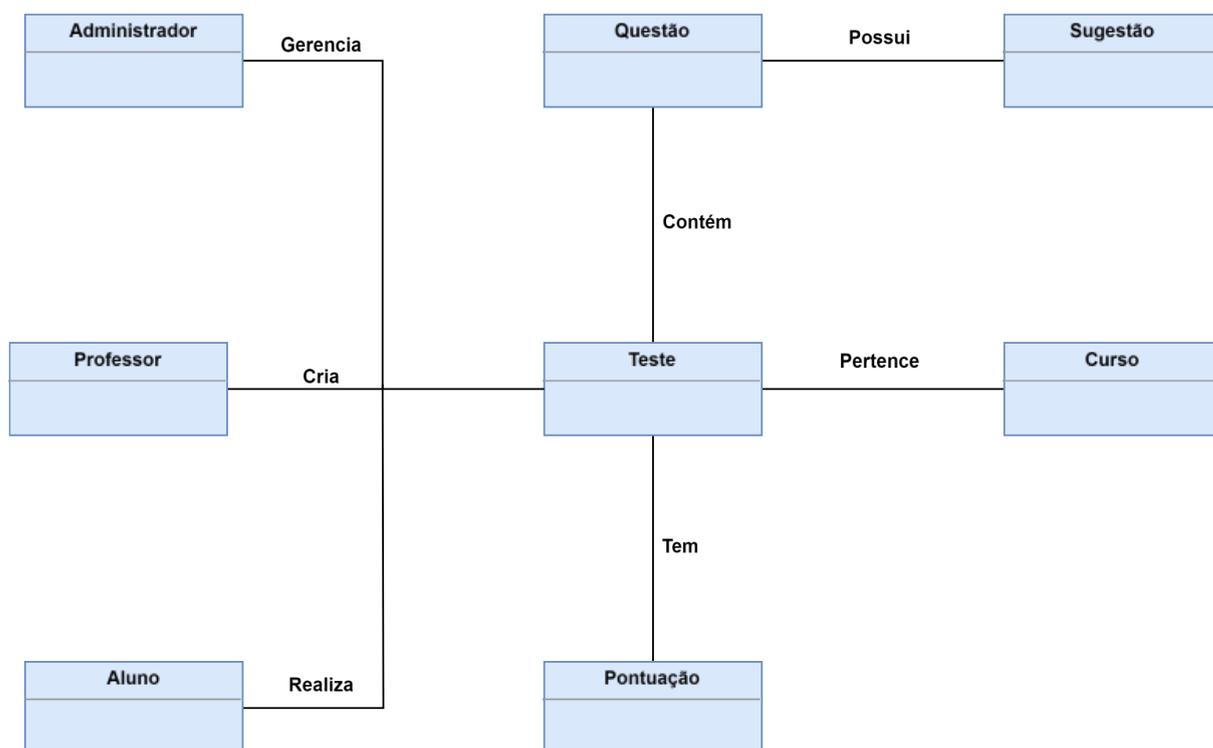
Programação. Os Testes representam uma forma de organização do sistema com pontuações individuais que proporciona uma visão global do desempenho do aluno.

A Pontuação registra meticulosamente o desempenho dos alunos, o que permite análises cruciais e direciona sugestões de conteúdos personalizadas, como de vídeos de apoio e *links*, o que enriquece o aprendizado com base no desempenho do aluno.

Os Cursos fornecem uma estrutura organizada para o aprendizado, alinhada aos objetivos pedagógicos. A entidade *Feedback* permite que os alunos forneçam valiosas opiniões sobre as sugestões de conteúdo personalizado e a experiência de aprendizado, contribuindo para a melhoria contínua da plataforma.

Ao entrelaçar essas entidades de maneira sinérgica, o modelo de domínio visa criar uma experiência educacional personalizada e rica, capacitando professores e alunos a atingirem seus objetivos acadêmicos de maneira eficaz e envolvente. Este sistema não apenas avalia o desempenho, mas também inspira uma jornada de aprendizado enriquecedora e adaptativa.

Figura 8 - Modelo de domínio



Fonte: Autor

O diagrama de classe apresentado na Figura 9 fornece uma visão da estrutura do sistema e destaca suas classes, atributos, métodos e as interações entre os objetos.

A classe Administrador assume um papel crucial no gerenciamento global do sistema. Com métodos como **autorizar\_cad**, **rejeitar\_cad**, **deletar\_cad** e **atualizar\_cad**, o Administrador tem o controle sobre a gestão de usuários, especialmente vinculado à classe Teste.

A Classe Professor é essencial para a personalização da experiência educacional. Com atributos como **user\_id**, **foto**, **email**, **telefone** e **status**, e métodos como **criar\_teste** e **atribuir\_sugestao**, os professores podem contribuir significativamente para o conteúdo do sistema, estando diretamente conectados à classe Teste.

A Classe Aluno representa a base do sistema, com atributos como **user\_id**, **foto**, **email** e **telefone**. Seus métodos, como **realizar\_teste**, **visualizar\_sugestao** e **visualizar\_resultado** permitem uma interação ativa com o conteúdo educacional, estando, assim, associados à classe Teste.

A Classe Teste é o núcleo do sistema que possui o atributo id para identificação única que consolidam informações cruciais relacionadas a questões, sugestões e resultados dos estudantes.

A Classe Curso, conectada à Classe Teste, é responsável por estruturar o conteúdo educacional. Atributos como **nome**, **numero\_questao** e **pontuacao**, aliados a métodos como **criar\_curso**, **adicionar\_questao** e **calcular\_pontuacao**, contribuem para a organização e eficácia do aprendizado.

A Classe Sugestão desempenha um papel vital no enriquecimento do aprendizado. Com atributos como **id**, **nome**, **categoria**, **sugestao1**, **sugestao2**, **sugestao3** e **feedback**, e métodos como **dar\_feedback**, ela proporciona sugestões dos estudantes, influenciando diretamente a classe Teste.

A Classe Questão, vinculada à Classe Curso, estrutura os desafios de aprendizado. Seus atributos incluem **curso**, **pontuacao**, **questao**, **opcao1**, **opcao2**, **opcao3**, **opcao4** e **resposta**, com métodos como **criar\_questao**, **atribuir\_resposta**

e **atribuir\_pontuacao** e fornece elementos fundamentais para a avaliação do conhecimento.

A Classe Resultado, também atrelada à Classe Teste, registra informações cruciais sobre o desempenho dos estudantes. Com atributos como **estudante**, **teste**, **pontuacao**, **data** e **porcentagem**, e métodos como **apresentar\_resultado**, oferece uma visão abrangente do progresso acadêmico.

O diagrama de classes proposto se revela como um alicerce robusto e interconectado, delineando não apenas as entidades-chave, mas também as interações fundamentais que fundamentarão o sistema. Essa representação visual, integral à metodologia de desenvolvimento ICONIX, não apenas esboça uma estrutura coerente, mas também se apresenta como uma ferramenta indispensável no processo de programação.

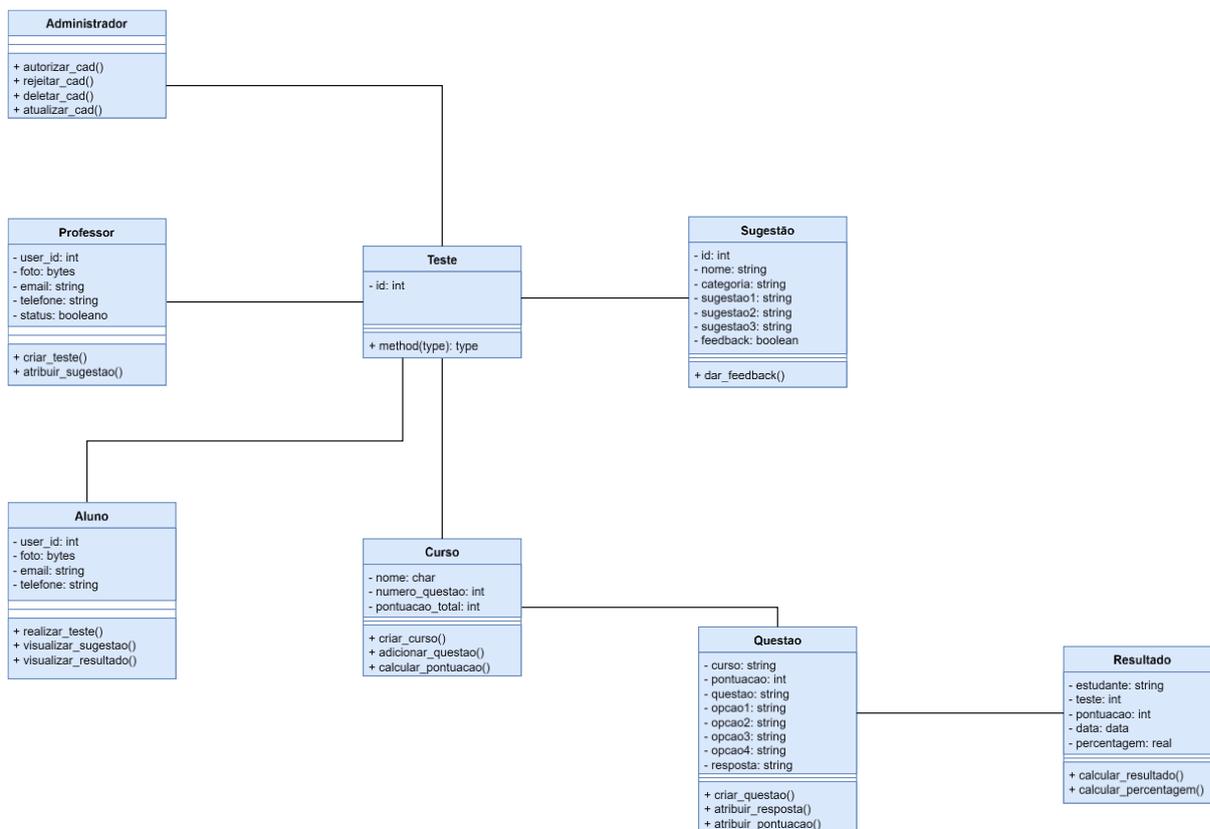
Na conformidade com as diretrizes da Lei Geral de Proteção de Dados (LGPD), o atributo foto na Classe Professor e Classe Aluno será tratado com extrema cautela e em estrita observância às normas de privacidade estabelecidas. A inclusão da foto visa personalizar a experiência do usuário, mas é essencial respeitar integralmente os princípios de proteção de dados.

O sistema garantirá que a coleta e armazenamento da foto estejam em conformidade com a LGPD que assegura a obtenção do consentimento explícito dos usuários. Além disso, serão implementadas medidas de segurança para proteger a integridade e confidencialidade desses dados sensíveis.

Os usuários terão controle total sobre suas fotos, podendo visualizar, modificar ou excluir essas informações a qualquer momento. O sistema não utilizará as fotos para finalidades além daquelas explicitamente autorizadas pelos usuários, e não serão compartilhadas com terceiros sem consentimento expresso.

O tratamento transparente e ético desses dados respeitará a privacidade dos usuários, desta forma, contribui para a construção de um ambiente virtual seguro e confiável, em total conformidade com as regulamentações da LGPD.

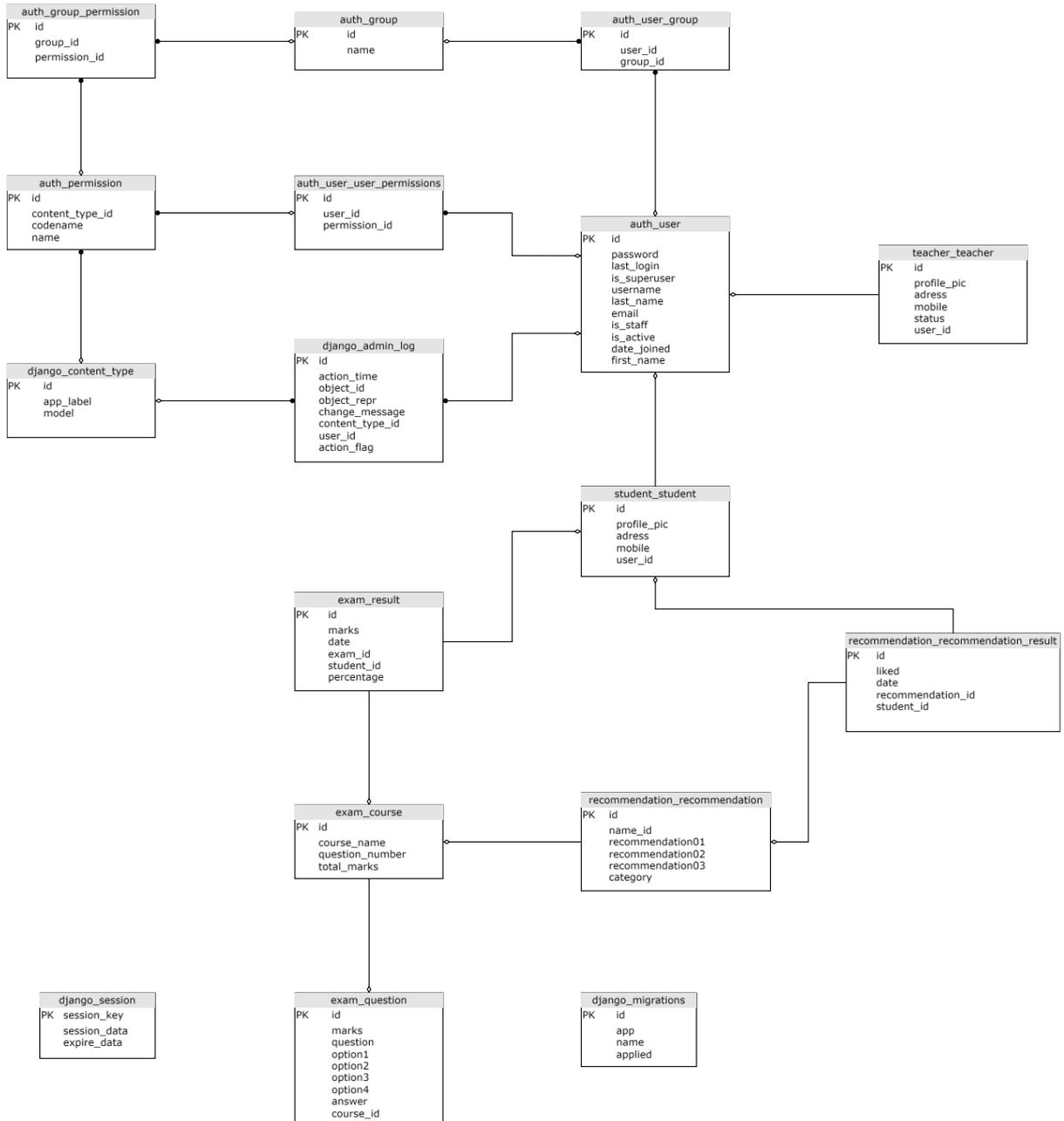
Figura 9 - Diagrama de classe



Fonte: Autor

O Banco de Dados desempenha um papel crucial e armazena de maneira eficiente informações essenciais, como dados do usuário, vídeos recomendados e material de apoio, como sugestões personalizadas. Para garantir a integridade e a eficiência dos dados, seguimos as práticas recomendadas pelas metodologias de desenvolvimento de *software*, especificamente a ICONIX. Assim, o Diagrama de Entidade-Relacionamento (DER), ver Figura 10, foi projetado em conformidade com os requisitos levantados durante a análise inicial, as entidades, atributos e relacionamentos necessários para o funcionamento adequado do sistema.

Figura 10 - Estrutura das tabelas e os relacionamentos entre elas do Banco de Dados.



Fonte: Autor

A seguir, uma explicação sobre cada uma das tabelas que compõem o diagrama: As tabelas do Django desempenham papéis fundamentais no funcionamento interno do sistema e são criadas de forma automática. A tabela **auth\_group\_permission** armazena as permissões associadas a cada grupo de usuários que facilita a gestão de acesso e administração de usuários em grande escala. Já a tabela **auth\_group** organiza os usuários em grupos e permite atribuir permissões coletivas. A tabela **auth\_user\_group** estabelece a relação entre usuários

e grupos, enquanto a ***auth\_permission*** registra as permissões disponíveis no sistema, controlando o acesso dos usuários.

A tabela ***auth\_user\_user\_permissions*** determina as permissões individuais concedidas a cada usuário e permite diferentes conjuntos de permissões com base em suas funções no sistema. A tabela ***auth\_user*** armazena informações dos usuários registrados, como nome de usuário e senha criptografada, essenciais para a autenticação e identificação de usuários.

A tabela ***django\_content\_type*** registra todos os modelos definidos no sistema Django, o que facilita consultas genéricas e dinâmicas. A ***django\_admin\_log*** registra ações administrativas no painel de administração, enquanto a ***django\_session*** armazena informações de sessão dos usuários, crucial para o gerenciamento de sessões.

A tabela ***django\_migrations*** mantém o registro de todas as migrações de banco de dados aplicadas ao projeto e garante a consistência do banco de dados ao longo do tempo. Essas tabelas são essenciais para a implementação de recursos como autenticação de usuários, controle de acesso e rastreamento de alterações administrativas.

A tabela ***student\_student*** armazena informações sobre os alunos, como seu identificador único (***id***), foto de perfil (***profile\_pic***), endereço (***adress***), número de telefone (***mobile***) e o ID associado ao usuário do Django (***user\_id***). Esses atributos são essenciais para manter registros detalhados dos alunos matriculados no sistema.

Da mesma forma, a tabela ***teacher\_teacher*** registra detalhes sobre os professores, incluindo seu identificador único (***id***), foto de perfil (***profile\_pic***), endereço (***adress***), número de telefone (***mobile***), status (ativo/inativo) e o ID do usuário do Django (***user\_id***). Esses atributos permitem o gerenciamento eficiente dos professores na plataforma.

A tabela ***exam\_result*** é responsável por armazenar os resultados dos exames realizados pelos alunos. Ela registra informações como o identificador único (***id***) do resultado, as notas obtidas (***marks***), a data da avaliação (***date***), o ID do exame (***exam\_id***), o ID do aluno (***student\_id***) e a porcentagem de acertos (***percentage***).

Esses dados são cruciais para avaliar o desempenho dos alunos ao longo do tempo.

A tabela **exam\_course** contém detalhes sobre os cursos associados aos exames. Ela inclui o identificador único (**id**) do curso, o nome do curso (**course\_name**), o número de questões no exame (**question\_number**) e o total de pontos possíveis (**total\_marks**). Esses atributos ajudam a organizar e estruturar os exames por curso.

A tabela **exam\_question** registra as perguntas dos exames, juntamente com suas opções e respostas. Ela armazena informações como o identificador único (**id**) da pergunta, as notas atribuídas (**marks**), o texto da pergunta (**question**), as opções de resposta (**option1**, **option2**, **option3** e **option4**), a resposta correta (**answer**) e o ID do curso associado (**course\_id**). Esses dados são cruciais para a criação e administração de exames online.

A tabela **recommendation\_recommendation** mantém registros das recomendações fornecidas aos alunos. Ela inclui o identificador único (**id**) da recomendação, o ID do aluno associado (**student\_id**), as recomendações específicas (**recommendation01**, **recommendation02** e **recommendation03**) e a categoria da recomendação (**category**). Esses atributos são úteis para oferecer sugestões personalizadas aos alunos com base em seus interesses e necessidades.

Por fim, a tabela **recommendation\_recommendation\_result** registra os resultados das recomendações dadas pelos alunos. Ela inclui o identificador único (**id**) do resultado, um indicador de preferência (**liked**), a data da recomendação (**date**), o ID da recomendação associada (**recommendation\_id**) e o ID do aluno (**student\_id**). Esses dados são valiosos para avaliar a eficácia das recomendações e personalizar ainda mais o processo de recomendação para cada aluno.

Ao concluir esta etapa dedicada à modelagem empregada no desenvolvimento do sistema *web*, é inegável o impacto positivo que a escolha da metodologia adequada teve no decorrer do projeto. A utilização de uma abordagem estruturada e baseada em metodologias reconhecidas, como a ICONIX, proporcionou uma direção clara e organizada para o desenvolvimento do sistema e garante a eficiência, a qualidade e alinhamento com os requisitos estabelecidos. A etapa seguinte da modelagem, aborda a construção dos protótipos de tela e visualização da interface do usuário. Os

protótipos desempenharam um papel crucial ao fornecer uma representação visual das funcionalidades e fluxos de interação do sistema, permitindo uma avaliação preliminar da usabilidade e navegabilidade. Através da criação e iteração dos protótipos, foi possível validar conceitos, identificar requisitos adicionais. Esses protótipos serviram como guias visuais durante o processo de desenvolvimento e orientou a implementação da interface de usuário de forma coerente e consistente com as necessidades e expectativas dos usuários finais. Assim, os protótipos de tela desempenharam um papel essencial ao traduzir os requisitos e especificações do sistema em uma experiência de usuário tangível e intuitiva, preparando o terreno para a fase de implementação e refinamento do sistema final.

Após a fase de modelagem, o sistema entrou na fase de escolhas das tecnologias que permitiu o desenvolvimento e codificação, encontra-se atualmente em uma etapa de testes de aplicabilidade com acesso local ao ambiente de testes da instituição. Durante esta fase, testes de usabilidade foram realizados para identificar e corrigir falhas ou aplicar melhorias, os testes escolhidos foram o teste moderado, o teste presencial e o teste avaliativo. O objetivo é progredir gradualmente até alcançar um estágio em que o sistema seja considerado completamente utilizável. Após a conclusão dos testes e correções necessárias, os resultados de usabilidade serão gerados, preparando o sistema para a implementação no ambiente de produção permitindo que os usuários o acessem em caráter experimental.

## **6.2. A Plataforma Web**

O produto resultante desse conjunto de tecnologias é um Sistema *Web* para Ensino de Lógica de Programação que vai além da implementação funcional. Destaca-se por sua eficiência, modularidade e usabilidade. Ao longo deste capítulo, exploraremos detalhadamente a utilização desse sistema, proporcionando uma compreensão abrangente de como ele pode ser empregado para impulsionar o ensino.

Na Figura 11, temos a página inicial do sistema, onde o visitante pode se inscrever como Aluno, Professor ou acessar o Sistema como o Administrador, através dos botões “**Aluno**”, “**Professor**” e “**Administrador**” disposto na barra de menu na parte superior da tela, nesse local ainda temos a possibilidade de visualizar as informações de contato com os desenvolvedores do sistema. Por ser a tela inicial do

sistema, ela é comum a todos os usuários.

Figura 11 - Tela inicial do Sistema Web de Ensino de Lógica de Programação.



Fonte: Autor

### 6.2.1. A Visão do Aluno

A tela de apresentação destinada aos alunos, conforme pode ser visto na Figura 12, acessada através da opção "**Aluno**" no menu principal, representa o ponto de partida crucial na experiência do usuário com o sistema de ensino de lógica de programação. Essa interface, projetada de forma intuitiva e convidativa, desempenha um papel na recepção e engajamento dos alunos, estabelecendo as bases para uma jornada de aprendizado eficaz e personalizada.

Além de servir como ponto de verificação para identificar se o usuário já possui cadastro no sistema. Através dos botões "**Inscreva-se**" e "**Login**", o sistema direciona o usuário para as respectivas funcionalidades, garantindo um fluxo de navegação claro e eficiente, com isso indica claramente os próximos passos que o aluno deve seguir, seja realizando o cadastro ou acessando sua área pessoal. Essa clareza evita que o usuário se sinta perdido ou confuso.

Figura 12 - Tela inicial do sistema com informações para o aluno

Home Professor Aluno Administrador Contato

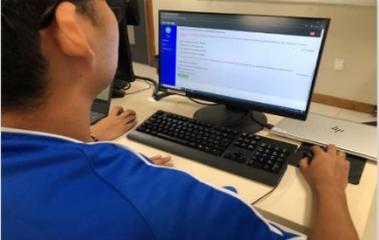
## Prezado Aluno

Bem-vindo ao Sistema Web para Apoio ao Processo de Ensino e Aprendizagem em Lógica de Programação, desenvolvido para promover uma experiência de aprendizagem personalizada para você.



**Feedback**

Após cada sugestão de conteúdo personalizado, é possível dá seu *feedback*.



**Interface Intuitiva**

Navegue facilmente pelo sistema com uma interface amigável e intuitiva, projetada para facilitar seu aprendizado e maximizar sua produtividade.



**Podemos Sempre Melhorar**

Entre em contato conosco para sugestões de melhorias.

O sistema foi desenvolvido para oferecer a você, aluno, uma ferramenta fácil e intuitiva, que visa tornar o processo de aprendizado mais eficiente e eficaz. Aproveite todas as funcionalidades disponíveis.

Para começar clique em uma das opções abaixo

[Inscreva-se](#)
[Login](#)

Copyright © Sistema de Teste Online - 2024

Fonte: Autor.

A Figura 13 representa a tela de cadastro e é visualizada dentro das páginas dos usuários “**Aluno**” e “**Professor**”, ao clicar no botão “**Inscreva-se**”. Na página “**Inscrição**”, são apresentadas informações necessárias a serem preenchidas para realizar o cadastro, conforme Requisito Funcional Dois (RF02).

Ao preencher todos os campos corretamente e clicar no botão “**Criar Conta**”, os dados serão inseridos no banco de dados do sistema conforme o perfil inicial escolhido (professor ou aluno). Com isso, será possível fazer login para utilizar o sistema

Figura 13 - Tela de Cadastro: preencha os dados para começar

Home Professor Aluno Administrador Contato

### Inscrição

Nome

Primeiro Nome Sobrenome

Contato

Telefone Endereço

Nome de Usuário

Usuário Senha

Foto

Escolher arquivo Nenhum arquivo escolhido

Criar Conta Limpar

Copyright © Sistema de Teste Online - 2024

Fonte: Autor.

A Figura 14 representa a interface de *login* do Sistema *Web* de Ensino de Lógica de Programação, proporcionamos aos usuários, em particular aos alunos, uma entrada segura e personalizada para explorar aos recursos educacionais oferecidos. Através da digitação do nome de usuário e senha, os alunos podem acessar de maneira rápida e intuitiva um ambiente dedicado, adaptado às suas necessidades individuais de aprendizado.

Após a inserção dos dados corretos na página de *login*, o sistema realizará uma busca na base de dados para verificar a existência desse cadastro. Se os dados forem encontrados, o acesso será concedido à área do aluno. Caso contrário, uma mensagem será exibida informando que os dados não foram encontrados.

Essa abordagem visa fornecer uma experiência de usuário fluida, onde a autenticação é o ponto de partida para uma jornada educacional personalizada. Ao inserir suas credenciais, os alunos desbloqueiam não apenas o acesso aos testes, mas também à sua própria recomendação personalizada.

O cuidado com a segurança e a facilidade de acesso são prioridades nesse design, garantindo que os alunos possam imergir no ambiente educacional de forma

conveniente, mantendo a integridade de suas informações pessoais. O *login* representa a porta de entrada para um mundo de aprendizado adaptativo e interativo, contribuindo assim para uma experiência educacional mais eficaz e centrada no aluno.

Figura 14 - Tela de Login para uma experiência personalizada.

A imagem mostra a interface de login de um sistema. No topo, há uma barra de navegação azul com botões para 'Home', 'Professor', 'Aluno', 'Administrador' e 'Contato'. O formulário de login centralizado tem um fundo escuro e contém o título 'Login', um campo de texto para 'Nome de Usuário', um campo de texto para 'Senha', um botão azul 'Login' e um botão cinza 'Limpar'. Na base, uma barra azul contém o texto 'Copyright © Sistema de Teste Online - 2024'.

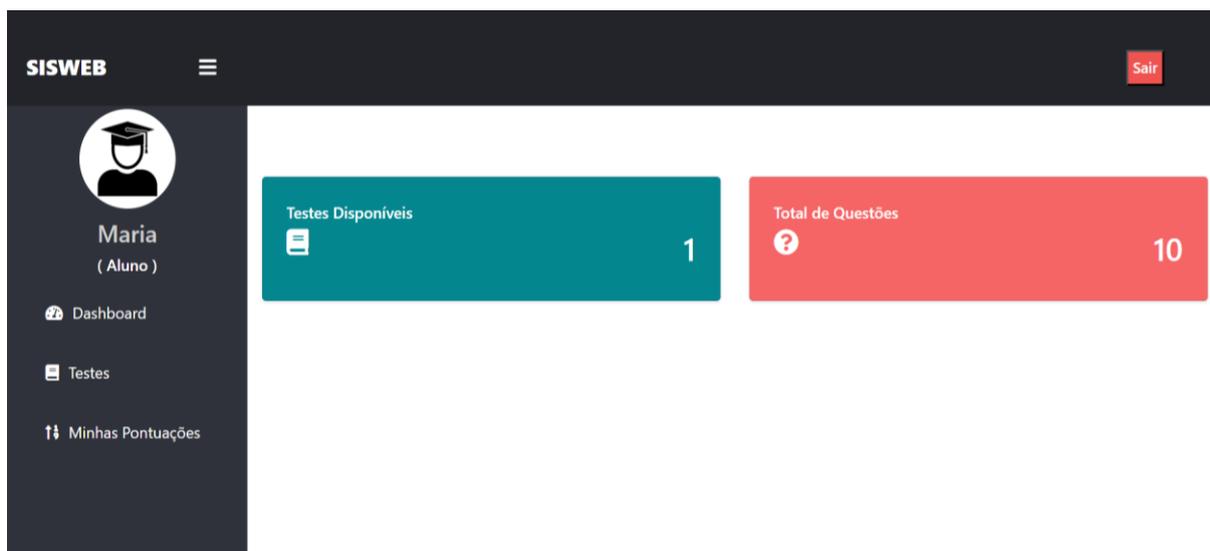
Fonte: Autor.

No painel destinado aos alunos dentro do sistema, é apresentado um dashboard que oferece uma visão instantânea e prática do progresso e das atividades disponíveis. Duas informações fundamentais são destacadas para fornecer clareza e guiar a interação do aluno:

- **Testes Disponíveis:** Nesta seção, os alunos podem visualizar diretamente o número de testes disponíveis para realização. Essa abordagem facilita o planejamento e a gestão eficiente do tempo de estudo, permitindo que os alunos escolham estrategicamente as avaliações a serem realizadas.
- **Total de Questões:** Na mesma interface, é apresentado o total de questões cobertas nos testes disponíveis. Essa informação é crucial para que os alunos compreendam a amplitude do conteúdo a ser abordado nas avaliações, permitindo uma preparação mais abrangente e direcionada.

Esse *design* centrado no aluno, ao fornecer informações transparentes sobre avaliações disponíveis e o escopo total de questões, visa simplificar a navegação e promover um engajamento mais eficaz, contribuindo para uma experiência personalizada e focada no aprendizado.

Figura 15 - Visão geral (Dashboards) dos testes disponíveis e total de questões.



Fonte: Autor.

Ao explorar o painel lateral e selecionar a opção "**Testes**", os alunos são conduzidos a uma visão abrangente dos testes disponíveis, proporcionando-lhes uma visão panorâmica das oportunidades de avaliação à sua disposição. Nesse ambiente dedicado, os alunos podem visualizar uma lista completa de testes disponíveis, cada um representando uma oportunidade única de avaliar e aprimorar seus conhecimentos em lógica de programação, de acordo com o RF04, detalhado na Tabela 4.

Tabela 4 - Detalhamento do Caso de Uso "Realização de Testes (Quiz)"

<b>ID:</b>	RF04
<b>Requisito Funcional:</b>	Realização de Testes (Quiz)
<b>Descrição:</b>	Alunos devem poder acessar e responder aos testes disponíveis, com registro de respostas e fornecimento de feedback imediato sobre o desempenho.
<b>Caso de Uso:</b>	Realização de Testes (Quiz)
<b>Ator Principal:</b>	Aluno
<b>Objetivo:</b>	Permitir que os alunos acessem, respondam a testes ( <i>quizzes</i> ) disponíveis na plataforma, e recebam <i>feedback</i> imediato sobre seu desempenho.

<b>Pré-condições</b>	
<ul style="list-style-type: none"> <li>O aluno deve estar autenticado no sistema.</li> </ul>	
<ul style="list-style-type: none"> <li>O professor deve ter criado e disponibilizado um teste (quiz) para os alunos na plataforma.</li> </ul>	
<b>Fluxo Principal:</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1. O aluno faz login no sistema e navega até a seção de testes (quiz).	2. O sistema exibe uma lista de testes disponíveis que o aluno pode realizar.
3. O aluno seleciona um teste da lista.	4. O sistema apresenta as instruções do teste, incluindo tempo limite e número de questões.
5. O aluno inicia o teste clicando no botão "Iniciar Teste".	6. O sistema exibe as questões do teste uma a uma ou todas de uma vez, dependendo da configuração.
7. O aluno responde às questões	
8. Após responder todas as questões, o aluno clica em "Enviar Teste".	9. O sistema registra as respostas do aluno e processa o desempenho.
	10. O sistema fornece <i>feedback</i> imediato, exibindo: <ul style="list-style-type: none"> <li>A pontuação total do aluno.</li> <li>A lista de respostas corretas e incorretas.</li> <li>Sugestões de vídeos ou materiais adicionais, se configurados pelo professor.</li> </ul>
	11. O sistema armazena o resultado do teste no perfil do aluno para futuras consultas.
<b>Fluxo Alternativo</b>	
Se o aluno sair do teste antes de finalizá-lo, o sistema poderá salvar as respostas parciais, permitindo que o aluno continue de onde parou, dependendo da configuração do teste.	
<b>Pós-condições</b>	
<ul style="list-style-type: none"> <li>As respostas do aluno são registradas e armazenadas no sistema.</li> </ul>	

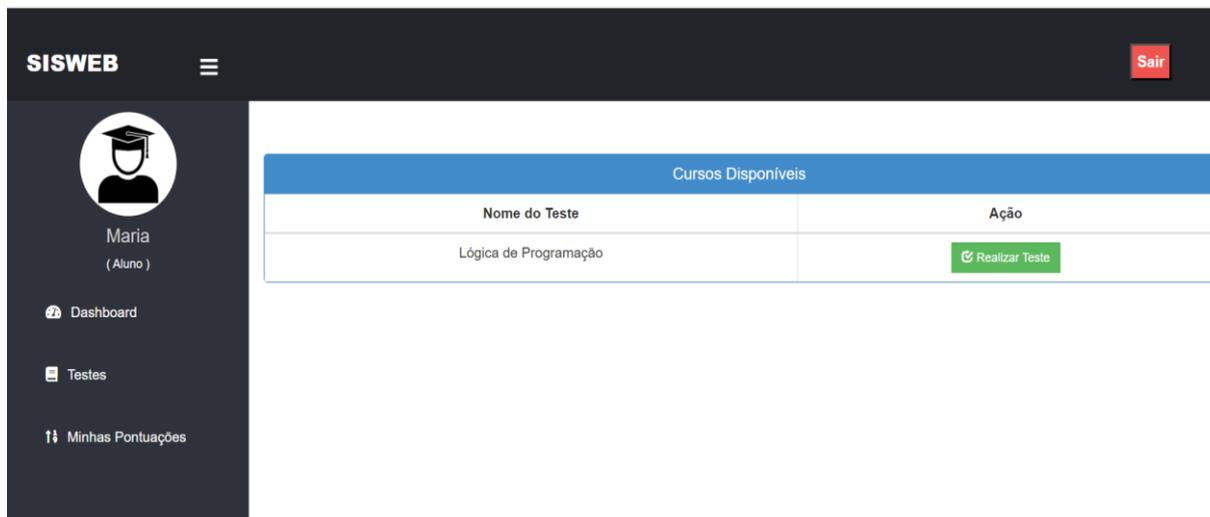
<ul style="list-style-type: none"> <li>• O <i>feedback</i> é fornecido ao aluno imediatamente após a finalização do teste.</li> </ul>
<ul style="list-style-type: none"> <li>• Os resultados são acessíveis tanto pelo aluno quanto pelo professor para análise posterior.</li> </ul>
<b>Regras de Negócio</b>
<ul style="list-style-type: none"> <li>• O sistema deve garantir que cada aluno possa realizar cada teste apenas uma vez, a menos que o professor permita múltiplas tentativas.</li> </ul>
<ul style="list-style-type: none"> <li>• Os testes devem ser bloqueados para alunos que não atenderem a pré-requisitos, se aplicável.</li> </ul>
<b>Exceções</b>
<ul style="list-style-type: none"> <li>• Se o sistema detectar problemas técnicos (como perda de conexão), o teste pode ser pausado e retomado posteriormente, conforme a configuração.</li> </ul>

Fonte: Autor

A interface oferece informações sobre cada teste como mostrado na Figura 16, incluindo a temática abordada. Isso permite que os alunos façam escolhas sobre quais testes desejam realizar, alinhando-se aos seus objetivos de aprendizado e áreas que desejam aprimorar. Além disso, a clareza na apresentação dos testes disponíveis contribui para uma experiência de usuário transparente e eficaz.

Essa funcionalidade não apenas oferece um panorama imediato das oportunidades de avaliação, mas também se integra à abordagem mais ampla do sistema. Com base no desempenho passado e preferências do aluno, o sistema, futuramente, pode sugerir testes específicos que se alinhem ao seu nível de habilidade e áreas de interesse, personalizando ainda mais a jornada educacional de cada aluno. Assim, a seção de testes representa não apenas uma lista estática, mas uma porta de entrada dinâmica para o desenvolvimento contínuo do conhecimento em programação.

Figura 16 - Janela que exibe a lista de testes: O aluno decide qual teste realizar.

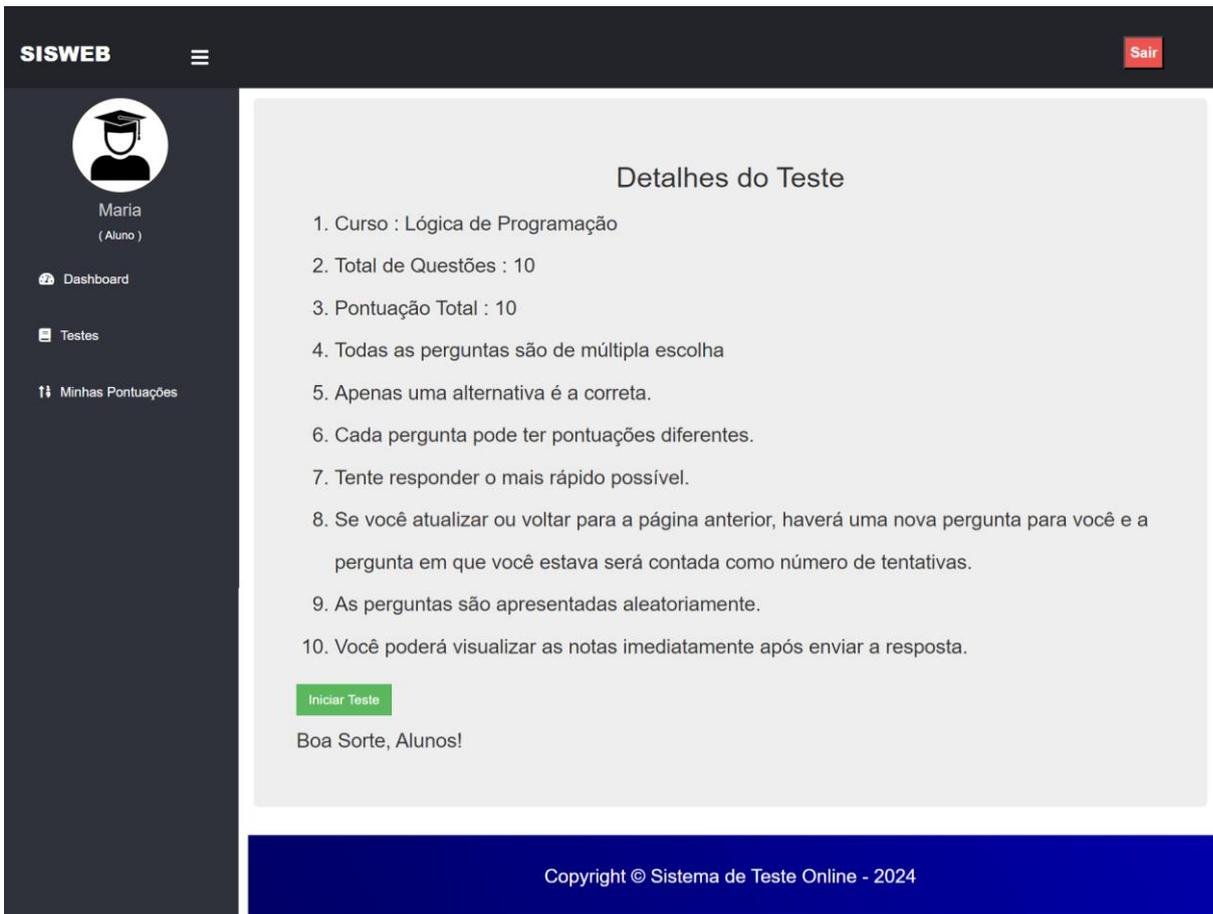


Fonte: Autor.

Ao decidir realizar um teste específico, os alunos são encaminhados para uma tela dedicada que oferece informações detalhadas sobre o teste selecionado representado através da Figura 17. Nesse espaço, são apresentados detalhes do teste, como o nome do curso, o total de questões e a pontuação total, assim como as regras do teste. Essa prévia detalhada proporciona uma visão abrangente do que o teste envolve, permitindo que os alunos se preparem adequadamente.

Além disso, a tela exibe claramente um botão destacado para iniciar o teste, proporcionando uma experiência de usuário intuitiva e direta. Essa abordagem visa minimizar qualquer ambiguidade, permitindo que os alunos se concentrem na avaliação em si, sem distrações desnecessárias. A clareza na apresentação das informações e na navegação contribui para um ambiente propício à concentração e ao desempenho otimizado durante a realização do teste.

Figura 17 - Regras e Informações sobre o teste escolhido.



The screenshot displays the SISWEB interface. At the top left, the logo 'SISWEB' is visible next to a hamburger menu icon. In the top right corner, there is a red 'Sair' button. The left sidebar features a user profile for 'Maria (Aluno)' with a graduation cap icon, and navigation links for 'Dashboard', 'Testes', and 'Minhas Pontuações'. The main content area is titled 'Detalhes do Teste' and lists ten rules for the test. A green 'Iniciar Teste' button is located below the list, followed by the text 'Boa Sorte, Alunos!'. At the bottom of the page, a dark blue footer contains the text 'Copyright © Sistema de Teste Online - 2024'.

**SISWEB** Sair

**Detalhes do Teste**

1. Curso : Lógica de Programação
2. Total de Questões : 10
3. Pontuação Total : 10
4. Todas as perguntas são de múltipla escolha
5. Apenas uma alternativa é a correta.
6. Cada pergunta pode ter pontuações diferentes.
7. Tente responder o mais rápido possível.
8. Se você atualizar ou voltar para a página anterior, haverá uma nova pergunta para você e a pergunta em que você estava será contada como número de tentativas.
9. As perguntas são apresentadas aleatoriamente.
10. Você poderá visualizar as notas imediatamente após enviar a resposta.

[Iniciar Teste](#)

Boa Sorte, Alunos!

Copyright © Sistema de Teste Online - 2024

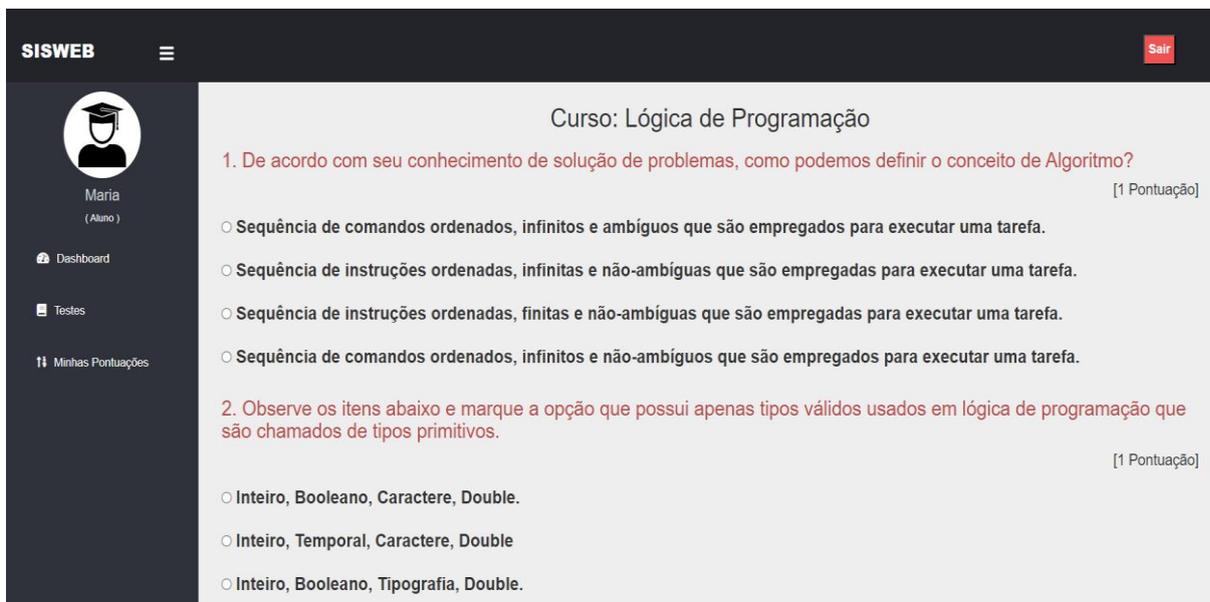
Fonte: Autor.

Ao optar por iniciar o teste, os alunos são conduzidos a uma sequência de perguntas relacionadas ao curso escolhido mostrado através da Figura 18. Cada questão é apresentada de maneira clara, exibindo o enunciado e fornecendo quatro alternativas de resposta em formato de múltipla escolha. A disposição organizada das informações visa facilitar a compreensão e resposta por parte dos alunos, oferecendo uma experiência de teste intuitiva.

Ao lado de cada questão, é indicada a pontuação associada, permitindo que os alunos tenham uma noção clara do peso de cada questão no total da avaliação. Após revisar as opções e fazer suas escolhas, os alunos encontram um botão “**Enviar Resposta**”, conforme pode ser visto na Figura 19. Esse botão marca o encerramento do teste, possibilitando que os alunos prossigam para a próxima área personalizada.

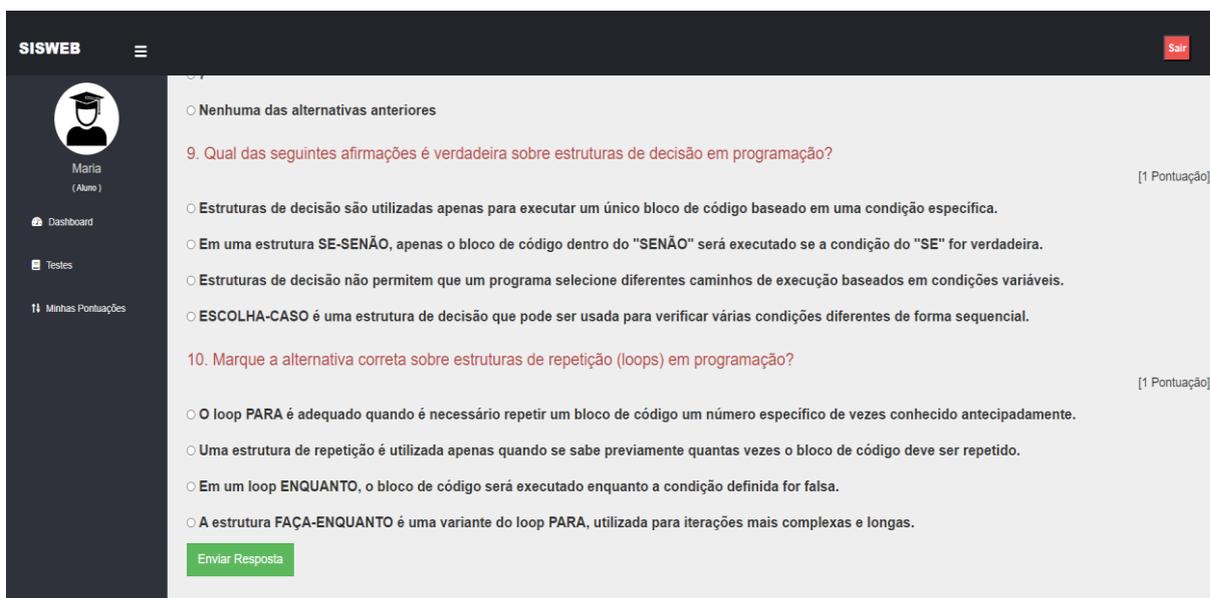
Essa abordagem interativa proporciona aos alunos uma experiência de teste dinâmica e controlada, promovendo o engajamento e permitindo que eles avancem pelo teste de maneira estruturada e eficiente.

Figura 18 - Janela mostrando as questões dos testes



Fonte: Autor.

Figura 19 - Janela mostrando as questões dos testes e botão de envio de resposta.



Fonte: Autor

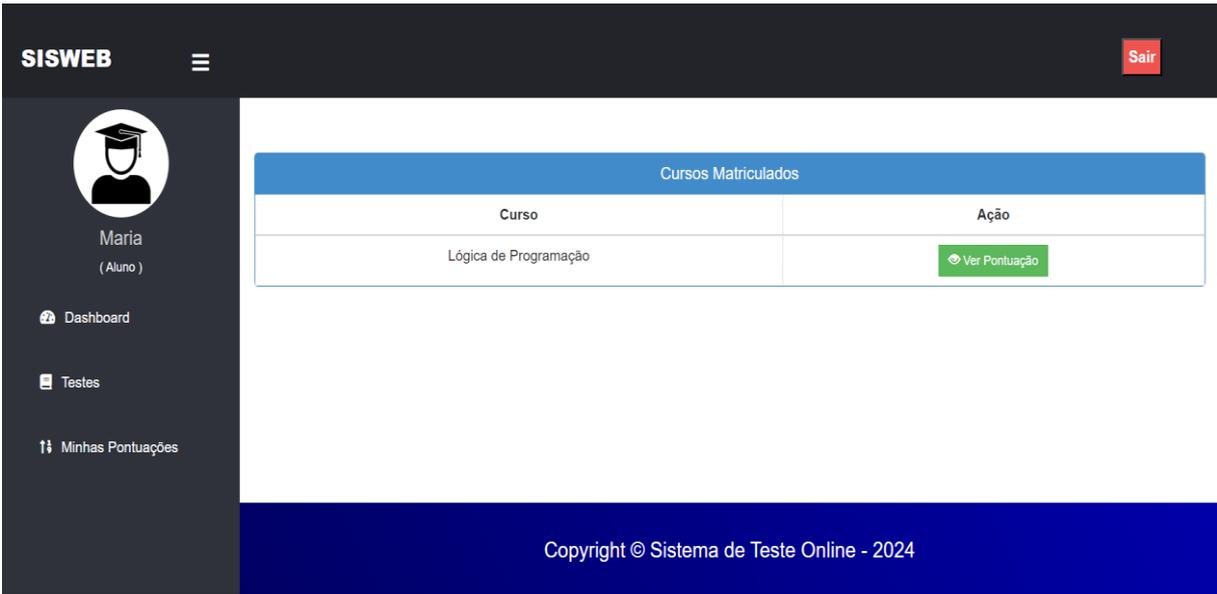
Ao concluir a série de perguntas do teste, os alunos são apresentados a uma janela informativa que exibe a lista dos testes realizados. Cada teste é identificado de maneira clara, possibilitando aos alunos visualizarem de forma organizada o histórico

de suas avaliações.

Para conferir as pontuações obtidas em cada teste, os alunos podem selecionar individualmente, clicando nos botões correspondentes “**Ver Pontuação**” como mostrado na Figura 20. Ao clicar em um desses botões, os alunos serão direcionados para uma nova janela detalhada que apresenta informações específicas sobre o teste escolhido, incluindo a pontuação alcançada.

Essa abordagem proporciona aos alunos uma visão abrangente de seu desempenho em diferentes avaliações, permitindo que acompanhem seu progresso ao longo do tempo. A interface intuitiva e acessível contribui para uma experiência positiva e facilita o entendimento das conquistas acadêmicas dos alunos.

Figura 20 - Histórico de Testes: Revise e Compare Suas Realizações.



The screenshot displays the SISWEB interface. On the left, a sidebar shows the user profile for Maria (Aluno) and navigation options: Dashboard, Testes, and Minhas Pontuações. The main content area features a table titled 'Cursos Matriculados' with the following data:

Curso	Ação
Lógica de Programação	<a href="#">Ver Pontuação</a>

At the bottom of the interface, a footer indicates 'Copyright © Sistema de Teste Online - 2024'.

Fonte: Autor.

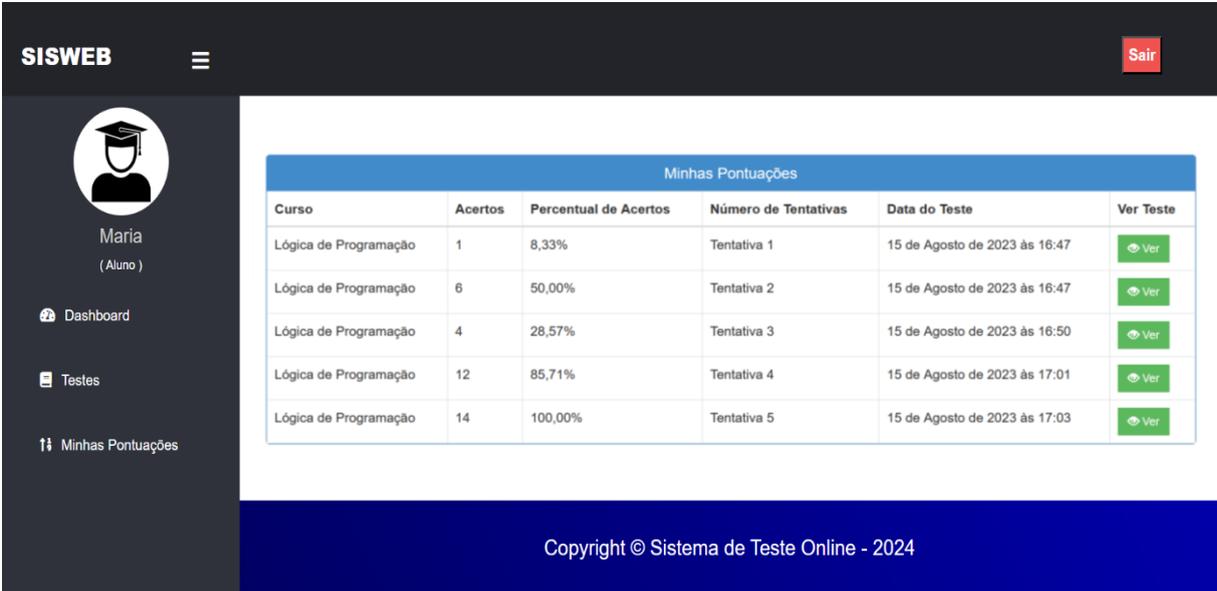
Ao explorar a seção de “**Minhas Pontuações**” como mostrado na Figura 21, os alunos visualizam uma janela informativa que oferece uma análise detalhada do desempenho naquele curso específico. Nessa janela, os alunos podem visualizar informações cruciais, incluindo o nome do curso, o número de acertos, a data da realização do teste e a opção de ver o teste.

O número de acertos e percentual de acertos reflete o desempenho acumulado ao longo de cada tentativa, proporcionando aos alunos uma visão abrangente de seu progresso no curso. O número de tentativas revela a dedicação do aluno em aprimorar

seu conhecimento, enquanto a data da última avaliação oferece um contexto temporal para a análise do desempenho.

Além disso, os alunos têm a oportunidade de revisar o teste, explorando as questões certas e erradas. Isso promove uma aprendizagem mais profunda, permitindo que os alunos identifiquem áreas de melhoria e reforcem os conceitos já dominados. Essa janela informativa é uma ferramenta valiosa para os alunos monitorarem e aprimorarem continuamente seu desempenho em cursos específicos.

Figura 21 - Janela de Estatísticas mostra os números e resultados dos testes.



The screenshot shows the SISWEB interface. On the left is a sidebar with the user's profile (Maria, Aluno) and navigation options: Dashboard, Testes, and Minhas Pontuações. The main content area displays a table titled 'Minhas Pontuações' with the following data:

Curso	Acertos	Percentual de Acertos	Número de Tentativas	Data do Teste	Ver Teste
Lógica de Programação	1	8,33%	Tentativa 1	15 de Agosto de 2023 às 16:47	Ver
Lógica de Programação	6	50,00%	Tentativa 2	15 de Agosto de 2023 às 16:47	Ver
Lógica de Programação	4	28,57%	Tentativa 3	15 de Agosto de 2023 às 16:50	Ver
Lógica de Programação	12	85,71%	Tentativa 4	15 de Agosto de 2023 às 17:01	Ver
Lógica de Programação	14	100,00%	Tentativa 5	15 de Agosto de 2023 às 17:03	Ver

At the bottom of the page, there is a copyright notice: Copyright © Sistema de Teste Online - 2024.

Fonte: Autor.

Ao clicar no botão **“Ver”** conforme mostrado na Figura 21, os alunos são apresentados a uma janela informativa abrangente que detalha cada questão do teste, incluindo suas alternativas e a alternativa correta, conforme mostra a Figura 22 e 23. Essa análise aprofundada permite que os alunos compreendam não apenas as respostas corretas, mas também os conceitos subjacentes a cada pergunta. Além disso, é disponibilizado um botão especial, **“Ver Sugestões”**, que direciona os alunos a uma seção fundamental deste sistema.

O foco primordial dessa funcionalidade é proporcionar sugestões personalizadas com base no desempenho do aluno. Ao clicar em **“Ver Sugestões”**, os alunos são guiados a uma seleção cuidadosamente ajustada de materiais, atividades e recursos adicionais que visam aprimorar as áreas identificadas como

desafiadoras durante o teste.

Essa abordagem proativa de oferecer sugestões personalizadas destaca a inovação e a eficácia do sistema, transformando a revisão do teste em uma oportunidade valiosa de aprendizado contínuo e aprimoramento da compreensão dos alunos.

Figura 22 - Relatório do teste: Questões e respostas com o botão “Ver recomendações”.



Teste		
Questão	Alternativas	Alternativa Correta
De acordo com seu conhecimento de solução de problemas, como podemos definir o conceito de Algoritmo??	(a) Sequência de comandos ordenados, infinitos e ambíguos que são empregados para executar uma tarefa. (b) Sequência de instruções ordenadas, infinitas e não-ambíguas que são empregadas para executar uma tarefa. (c) Sequência de instruções ordenadas, finitas e não-ambíguas que são empregadas para executar uma tarefa. (d) Sequência de comandos ordenados, infinitos e não-ambíguas que são empregados para executar uma tarefa.	C
Observe os itens abaixo e marque a opção que possui apenas tipos válidos usados em lógica de programação que são chamados de tipos primitivos.?	(a) Inteiro, Booleano, Caractere, Double. (b) Inteiro, Temporal, Caractere, Double (c) Inteiro, Booleano, Tipografia, Double. (d) Booleano, Real, Inteiro, Irracional	A

Fonte: Autor.

Figura 23 - Relatório do teste: Questões e respostas com o botão “Ver sugestões”



Marque a alternativa correta sobre estruturas de repetição (loops) em programação??	<p>programa seleccione diferentes caminhos de execução baseados em condições variáveis. (d) ESCOLHA-CASO é uma estrutura de decisão que pode ser usada para verificar várias condições diferentes de forma sequencial.</p> <p>(a) O loop PARA é adequado quando é necessário repetir um bloco de código um número específico de vezes conhecido antecipadamente. (b) Uma estrutura de repetição é utilizada apenas quando se sabe previamente quantas vezes o bloco de código deve ser repetido. (c) Em um loop ENQUANTO, o bloco de código será executado enquanto a condição definida for falsa. (d) A estrutura FAÇA-ENQUANTO é uma variante do loop PARA, utilizada para iterações mais complexas e longas.</p>	A
---	---	---

[Ver Sugestões](#)

Fonte: Autor

Na seção de “**Sugestões de Aprendizagem**” conforme apresentado na Figura 24 e de acordo com o RF05 detalhado na Tabela 4, os alunos se deparam com uma janela que oferece três sugestões cuidadosamente selecionadas, por professores para aprimorar seus estudos em lógica de programação. Cada sugestão é apresentada de forma clara, fornecendo informações concisas sobre o conteúdo proposto, seja um vídeo educacional, um tutorial em PDF ou uma plataforma de aprendizagem específica.

Tabela 5 - Detalhamento das Sugestão de Vídeos Educacionais

<b>ID:</b>	RF05
<b>Requisito Funcional:</b>	Sugestão de Vídeos Educacionais
<b>Descrição:</b>	Após a conclusão do teste, exibir uma sugestão de vídeo relacionado ao conteúdo abordado, postado previamente por um professor, alinhado ao desempenho do aluno.
<b>Caso de Uso:</b>	Realização de Testes (Quiz)
<b>Ator Principal:</b>	Aluno
<b>Objetivo:</b>	Fornecer ao aluno uma sugestão de vídeo educacional relacionado ao conteúdo do teste, com base em seu desempenho, após a conclusão do teste.
<b>Pré-condições</b>	
<ul style="list-style-type: none"> <li>• O aluno deve ter concluído um teste (quiz) na plataforma.</li> </ul>	
<ul style="list-style-type: none"> <li>• O professor deve ter previamente carregado e associado vídeos educacionais ao conteúdo do teste.</li> </ul>	
<ul style="list-style-type: none"> <li>• O sistema deve estar configurado para analisar o desempenho do aluno e sugerir vídeos de acordo.</li> </ul>	
<b>Fluxo Principal:</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>

1. Após o aluno finalizar o teste, o sistema processa as respostas e avalia o desempenho geral.	2. Com base no desempenho do aluno, o sistema determina o vídeo educacional mais relevante: <ul style="list-style-type: none"> <li>• Se o aluno teve aproveitamento baixo sistema prioriza vídeos que abordem os tópicos que ele teve mais dificuldade.</li> </ul>
	<ul style="list-style-type: none"> <li>• Se o aluno teve um desempenho excelente, o sistema pode sugerir vídeos que aprofundem o conhecimento ou explorem conteúdos avançados.</li> </ul>
	3. O sistema exibe a sugestão de vídeo na tela de resultados do teste: <ul style="list-style-type: none"> <li>• O aluno pode assistir ao vídeo diretamente na plataforma ou salvá-lo para visualização posterior.</li> </ul>
4. O aluno tem a opção de reproduzir o vídeo imediatamente ou navegar para outra seção da plataforma.	5. O sistema registra a visualização do vídeo no perfil do aluno para fins de acompanhamento e futuras recomendações.
<b>Pós-condições</b>	
<ul style="list-style-type: none"> <li>• A sugestão de vídeo é exibida ao aluno após a conclusão do teste.</li> </ul>	
<ul style="list-style-type: none"> <li>• O vídeo sugerido é registrado no perfil do aluno, indicando se foi assistido ou salvo para visualização posterior.</li> </ul>	
<b>Regras de Negócio</b>	
<ul style="list-style-type: none"> <li>• O sistema deve garantir que os vídeos sugeridos sejam sempre relevantes ao conteúdo do teste e ao desempenho do aluno.</li> </ul>	
<ul style="list-style-type: none"> <li>• O sistema pode priorizar vídeos que têm sido mais eficazes para outros alunos com desempenhos semelhantes, utilizando <i>feedback</i> histórico.</li> </ul>	
<b>Exceções</b>	
<ul style="list-style-type: none"> <li>• Se não houver vídeos educacionais disponíveis para o conteúdo específico, o sistema deverá notificar o aluno e o professor, e pode sugerir conteúdos externos ou tópicos alternativos.</li> </ul>	
<ul style="list-style-type: none"> <li>• Em caso de falhas na reprodução do vídeo (por exemplo, problemas técnicos), o sistema deve oferecer uma solução alternativa, como o download do vídeo ou a sugestão de outro material.</li> </ul>	

Para tornar o sistema ainda mais interativo, os alunos têm a oportunidade de avaliar cada sugestão com um simples clique que expressa sua preferência através de botões de "*Like*<sup>9</sup>" e "*Dislike*<sup>10</sup>". Essa funcionalidade visa adaptar continuamente as sugestões futuras, garantindo que o sistema refine suas recomendações com base no feedback direto dos alunos.

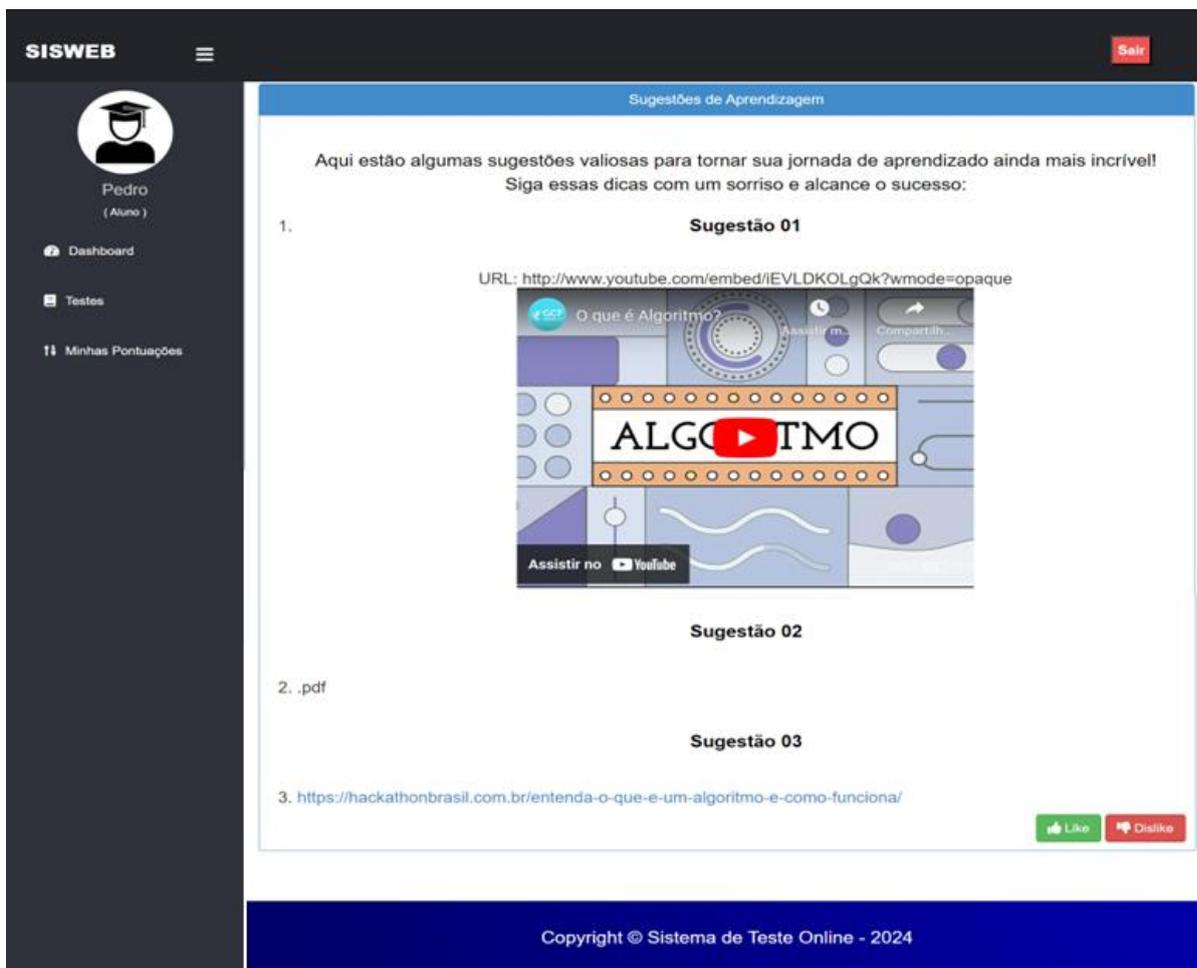
Essa abordagem inovadora não apenas oferece aos alunos acesso a recursos educacionais relevantes, mas também promove uma experiência de aprendizado mais personalizada. Ao capacitar os alunos a influenciar as sugestões com suas preferências, o sistema deixa de ser estático e se torna uma ferramenta dinâmica e adaptativa, alinhada aos interesses individuais e necessidades de cada aluno.

---

<sup>9</sup> *Like* ("Curtir"): Geralmente representado por um ícone de polegar para cima é usado para indicar que o usuário aprecia ou aprova o item avaliado. Em muitos casos, é associado a uma reação positiva ou satisfação.

<sup>10</sup> *Dislike* ("Não Curtir"): Representado por um ícone de polegar para baixo é utilizado para indicar que o usuário não aprecia ou não aprova o item avaliado. Pode ser associado a uma reação negativa ou insatisfação.

Figura 24 - Janela de sugestões com a capacidade de expressar sua opinião. Dê 'Like' para Aprovar e 'Dislike' para desaprovar.



Fonte: Autor

Na presente seção, exploramos de maneira abrangente as funcionalidades do Sistema *Web* que faz sugestões de forma não automática. Desde a navegação inicial até o ciclo completo de realização de testes, obtenção de pontuações e, por fim, as sugestões personalizadas, cada etapa foi delineada para proporcionar ao aluno uma jornada educacional mais eficiente e envolvente.

O aluno é guiado desde o acesso ao sistema, realização de testes específicos até a visualização detalhada das pontuações obtidas. Destaca-se a ênfase nas sugestões personalizadas como um diferencial significativo, onde o sistema não apenas avalia o desempenho, mas também adapta suas sugestões com base nas preferências e avaliações do aluno.

Ao integrar tecnologia e personalização, este sistema visa ir além do convencional, proporcionando aos alunos uma abordagem inovadora para o estudo

da Lógica de Programação. Ao concluir este capítulo, delineamos não apenas as características técnicas do sistema, mas também sua aplicabilidade prática na promoção de um ambiente de aprendizado mais eficaz e centrado no aluno. O próximo capítulo aprofundará ainda mais a análise dos resultados obtidos e as implicações pedagógicas dessa abordagem.

### **6.2.2. A Visão do Professor**

A tela de apresentação destinada aos professores, conforme pode ser vista na Figura 25, acessada através da opção "**Professor**" no menu principal, representa o ponto de partida para a gestão do ensino de lógica de programação no sistema. Projetada com o objetivo de oferecer uma experiência intuitiva e personalizada, essa interface reúne um conjunto de ferramentas que visam otimizar o processo de ensino e acompanhar o progresso dos alunos.

Além de servir como ponto de verificação para identificar se o usuário já possui cadastro no sistema. Através dos botões "**Inscreeva-se**" e "**Login**", o sistema direciona o usuário para as respectivas funcionalidades, garantindo um fluxo de navegação claro e eficiente, com isso indica claramente os próximos passos que o aluno deve seguir, seja realizando o cadastro ou acessando sua área pessoal. Essa clareza evita que o usuário se sinta perdido ou confuso.

Figura 25 - Tela inicial do sistema com informações para o professor

Home Professor Aluno Administrador Contato

## Prezado Professor

Bem-vindo ao Sistema Web para Apoio ao Processo de Ensino e Aprendizagem em Lógica de Programação, desenvolvido para facilitar o trabalho dos professores e promover uma experiência de aprendizagem personalizada para os alunos.



### Sugira materiais de estudo

Envie sugestões de leitura, exercícios adicionais e recursos multimídia para ajudar os alunos a melhorar seu desempenho.



### Acompanhe o progresso

Monitore o desenvolvimento dos alunos ao longo do tempo e ajuste suas sugestões conforme necessário.



### Podemos Sempre Melhorar

Entre em contato conosco para sugestões de melhorias.

O sistema foi desenvolvido para oferecer a você, professor, uma ferramenta fácil e intuitiva, que visa tornar o processo de ensino e avaliação mais eficiente e eficaz. Aproveite todas as funcionalidades disponíveis e proporcione uma educação mais adaptativa e personalizada para seus alunos.

Para começar clique em uma das opções abaixo

[Inscreva-se](#) [Login](#)

Copyright © Sistema de Teste Online - 2024

Fonte: Autor

A Figura 26 representa a tela de cadastro e é visualizada dentro das páginas dos usuários “**Aluno**” e “**Professor**”, ao clicar no botão “**Inscreva-se**”. Na página “**Inscrição**”, são apresentadas informações necessárias a serem preenchidas para realizar o cadastro.

Ao preencher todos os campos corretamente e clicar no botão “**Criar Conta**”, os dados serão inseridos no banco de dados do sistema conforme o perfil inicial escolhido (professor ou aluno). Com isso, será possível fazer login para utilizar o sistema

Figura 26 - Tela de Cadastro: preencha os dados para começar.

Home Professor Aluno Administrador Contato

### Inscrição

Nome

Primeiro Nome Sobrenome

Contato

Telefone Endereço

Nome de Usuário

Usuário Senha

Foto

Escolher arquivo Nenhum arquivo escolhido

Criar Conta Limpar

Copyright © Sistema de Teste Online - 2024

Fonte: Autor.

A Figura 27 representa a interface de *login* do Sistema *Web* de Ensino de Lógica de Programação, proporcionamos aos usuários, em particular aos alunos, uma entrada segura e personalizada para explorar aos recursos educacionais oferecidos. Através da digitação do nome de usuário e senha, os alunos podem acessar de maneira rápida e intuitiva um ambiente dedicado, adaptado às suas necessidades individuais de aprendizado.

Após a inserção dos dados corretos na página de *login*, o sistema realizará uma busca na base de dados para verificar a existência desse cadastro. Se os dados forem encontrados, o acesso será concedido à área do aluno. Caso contrário, uma mensagem será exibida informando que os dados não foram encontrados.

Essa abordagem visa fornecer uma experiência de usuário fluida, onde a autenticação é o ponto de partida para uma jornada educacional personalizada. Ao inserir suas credenciais, os alunos desbloqueiam não apenas o acesso aos testes, mas também à sua própria recomendação personalizada.

O cuidado com a segurança e a facilidade de acesso são prioridades nesse design, garantindo que os alunos possam imergir no ambiente educacional de forma

conveniente, mantendo a integridade de suas informações pessoais. O *login* representa a porta de entrada para um mundo de aprendizado adaptativo e interativo, contribuindo assim para uma experiência educacional mais eficaz e centrada no aluno.

Figura 27 - Tela de Login para uma experiência personalizada.

Home Professor Aluno Administrador Contato

Nome de Usuário

Senha

Login Limpar

Copyright © Sistema de Teste Online - 2024

Fonte: Autor.

No painel destinado aos professores dentro do sistema, é apresentado um *dashboard* que oferece uma visão instantânea das ações disponíveis. Três informações fundamentais são destacadas para fornecer clareza e guiar a interação do professor:

- **Gerenciar Cursos:** Através dessa área, o professor pode criar, editar e organizar seus cursos de forma personalizada, definindo as avaliações que o aluno pode realizar. Essa abordagem permite que o professor adapte o curso às necessidades específicas de seus alunos, promovendo um aprendizado mais eficaz e engajador. Além disso, a possibilidade de gerenciar diversos cursos a partir de uma única interface otimiza o trabalho do docente, reduzindo a carga administrativa e permitindo que ele se concentre nas atividades pedagógicas.
- **Gerenciar Questões:** A funcionalidade presente na interface do professor, desempenha um papel importante na criação e gestão de atividades avaliativas dentro do sistema. Esse recurso permite que o professor elabore questões objetivas, vinculando-as a um curso.
- **Gerenciar Sugestões:** A funcionalidade representa um mecanismo principal

do sistema que serve para propor sugestões aos alunos de conteúdo para ajudar no ensino e no aprendizado desses usuários.

Esse *design* centrado no professor, a Figura 28 mostra informações diretas sobre ações disponíveis e visa simplificar a navegação e promover um engajamento mais eficaz, contribuindo para uma experiência personalizada e otimizada.

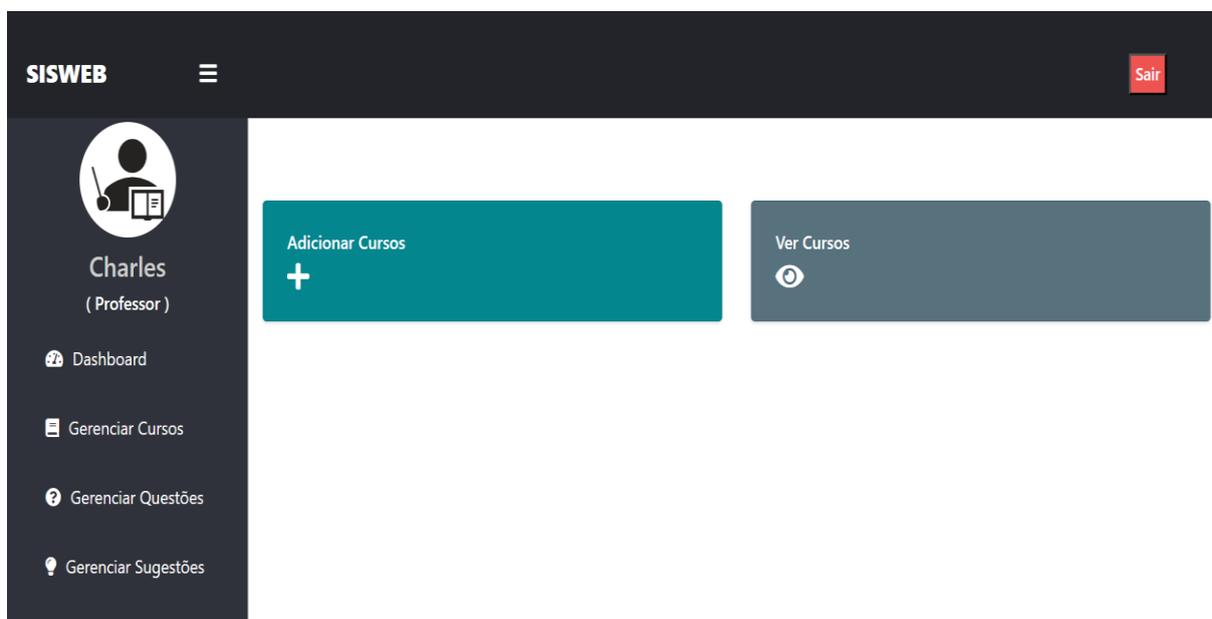
Figura 28 - Dashboard do professor



Fonte: Autor

Ao explorar o painel lateral e selecionar a opção "**Gerenciar cursos**", os professores são conduzidos a página de gerenciamento dos cursos, onde é possível "**Adicionar Cursos**" ou "**Ver Cursos**" previamente cadastrado pelo professor, conforme pode ser visto na Figura 29.

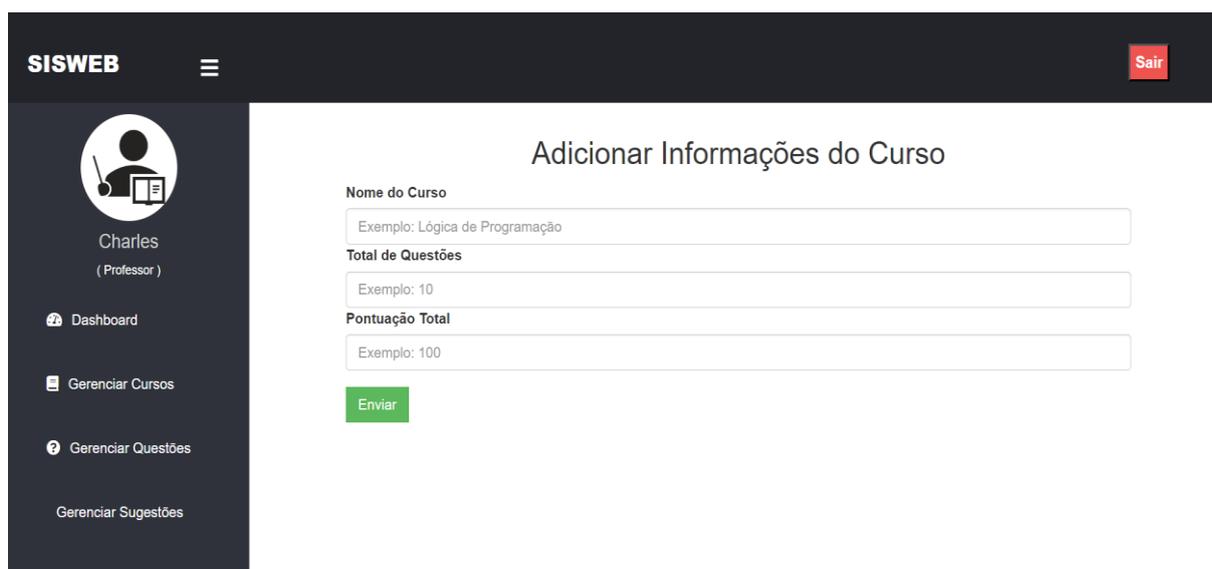
Figura 29 - Gerenciamento de Cursos



Fonte: Autor

Nesse ambiente dedicado, os professores possuem acesso as configurações dos cursos, incluindo a adição do nome do curso, a definição do número total de questões e a atribuição de pontuação a cada uma delas, como ilustrado na Figura 30.

Figura 30 - Janela de cadastro do curso

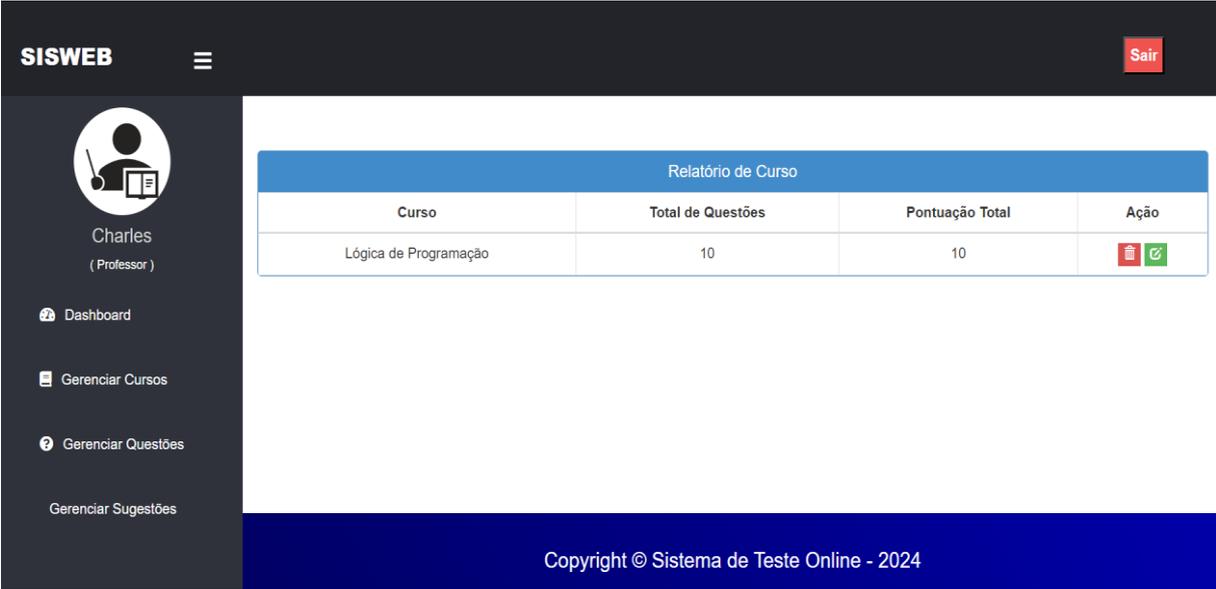


Fonte: Autor

Ao clicar em “**Enviar**”, as informações do novo curso são automaticamente

salvas em um banco de dados e o usuário é direcionado para a página de gerenciamento de cursos. Nessa página, é possível visualizar a lista completa de cursos cadastrados, com opções para editar, excluir ou visualizar detalhes de cada um, conforme pode ser visto na Figura 31.

Figura 31 - Relatório com os cursos cadastrados pelo professor.



The screenshot displays the SISWEB interface for a professor. The header shows the logo 'SISWEB', a menu icon, and a 'Sair' button. The user profile 'Charles (Professor)' is visible in the sidebar. The main content area features a 'Relatório de Curso' table with the following data:

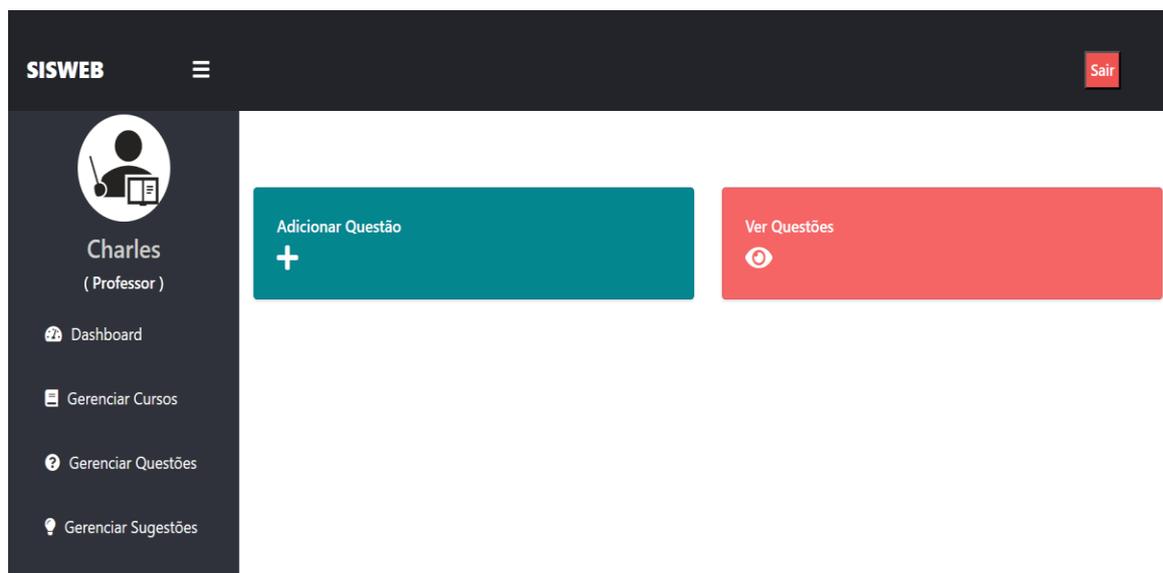
Curso	Total de Questões	Pontuação Total	Ação
Lógica de Programação	10	10	 

The footer of the interface contains the text 'Copyright © Sistema de Teste Online - 2024'.

Fonte: Autor

Ao acessar a opção “**Gerenciar Questões**”, o professor é direcionado para a área destinada à criação e gestão das questões do curso, conforme mostra a Figura 32. Nesta seção, é possível tanto adicionar novas perguntas quanto consultar o banco de questões já cadastradas.

Figura 32 - Janela que mostra a gestão das questões.



Fonte: Autor

Ao clicar em “**Adicionar Questão**”, o professor é direcionado para um formulário intuitivo onde pode criar perguntas para as avaliações. Nessa tela, é possível definir o enunciado da questão, as alternativas de resposta e indicar a resposta correta. Além disso, o professor pode atribuir uma pontuação específica para cada questão, personalizando assim as avaliações de acordo com os objetivos do curso, conforme pode ser visto na Figura 33.

Figura 33 - Tela para inserção das questões.

The screenshot shows the 'Adicionar Questão' (Add Question) interface in the SISWEB system. The interface is divided into a dark sidebar on the left and a main content area on the right. The sidebar contains the user profile 'Charles (Professor)' and navigation links for 'Dashboard', 'Gerenciar Cursos', 'Gerenciar Questões', and 'Gerenciar Sugestões'. The main content area features a form with the following fields:

- Curso:** A dropdown menu with the placeholder text 'Nome do Curso'.
- Questão:** A text input field with the example text 'Exemplo: Qual é a natureza da linguagem Python?'.
- Pontuação:** A text input field with the example text 'Exemplo: 10'.
- Opção 1:** A text input field with the example text 'Exemplo: É um tipo de linguagem procedural'.
- Opção 2:** A text input field with the example text 'Exemplo: É uma linguagem não procedural'.
- Opção 3:** A text input field with the example text 'Exemplo: É uma linguagem de alto nível'.
- Opção 4:** A text input field with the example text 'Exemplo: É uma linguagem de definição de dados'.
- Resposta Correta:** A dropdown menu with a placeholder text '.....'.
- Adicionar:** A green button to submit the question.

At the bottom of the page, there is a blue footer with the text 'Copyright © Sistema de Teste Online - 2024'.

Fonte: Autor

Após criar uma questão, o professor retorna à página principal de gerenciamento, onde pode optar por adicionar mais questões ou visualizar as já cadastradas. A opção **“Ver Questões”** oferece uma lista completa e organizada de todas as perguntas, como pode ser visto na Figura 34, facilitando a gestão do conteúdo das avaliações e otimizando o fluxo de trabalho do professor.

Figura 34 - Tela com as perguntas disponíveis

The screenshot shows the SISWEB interface. On the left is a dark sidebar with the user profile 'Charles (Professor)' and navigation links: 'Dashboard', 'Gerenciar Cursos', 'Gerenciar Questões', and 'Gerenciar Sugestões'. The top right has a 'Sair' button. The main area is titled 'Perguntas Disponíveis' and contains a table with the following data:

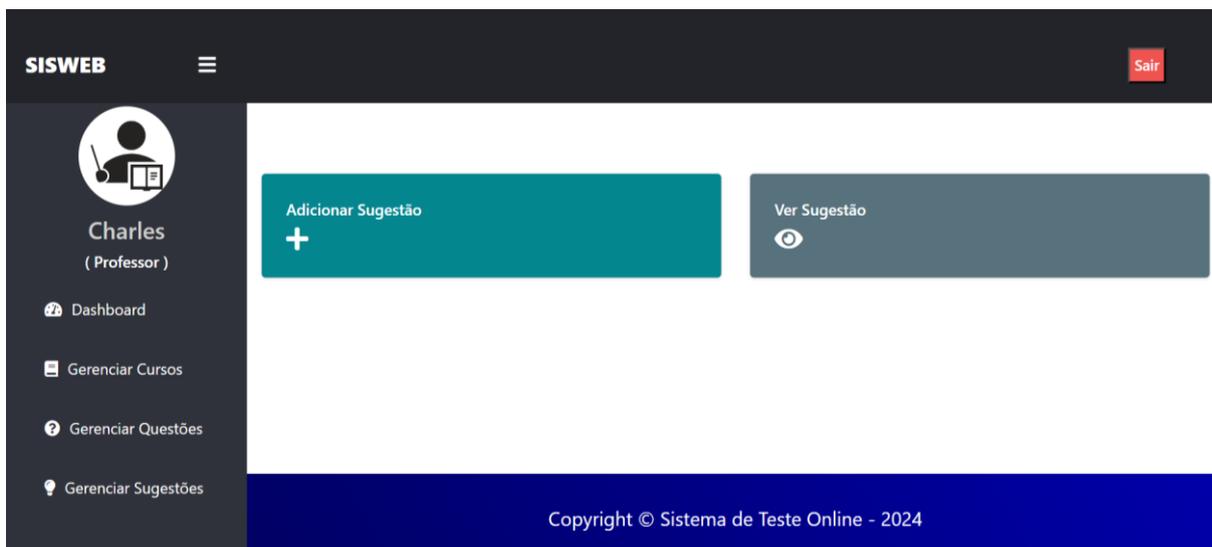
Questão	Pontuação	Ação
De acordo com seu conhecimento de solução de problemas, como podemos definir o conceito de Algoritmo?	1	[Red X] [Green Check]
Observe os itens abaixo e marque a opção que possui apenas tipos válidos usados em lógica de programação que são chamados de tipos primitivos.	1	[Red X] [Green Check]
Qual das seguintes opções é verdadeira sobre variáveis em um algoritmo?	1	[Red X] [Green Check]
Qual dos seguintes itens abaixo melhor representa um valor constante para ser usado em um algoritmo, qual deles você escolheria?	1	[Red X] [Green Check]
Qual das seguintes afirmações é verdadeira sobre operadores em um algoritmo?	1	[Red X] [Green Check]
Qual das seguintes afirmações é verdadeira sobre programas de computador?	1	[Red X] [Green Check]
Qual princípio fundamental da programação estruturada é verdadeiro?	1	[Red X] [Green Check]
Considere o seguinte algoritmo em pseudocódigo: 1. Ler um número inteiro do usuário. 2. Se o número for maior que 10, imprimir "Número maior que 10". 3. Caso contrário, imprimir "Número igual ou menor que 10". 4. Fim do algoritmo. Qual será a saída do algoritmo se o usuário inserir o número 7?	1	[Red X] [Green Check]
Qual das seguintes afirmações é verdadeira sobre estruturas de decisão em programação?	1	[Red X] [Green Check]
Marque a alternativa correta sobre estruturas de repetição (loops) em programação?	1	[Red X] [Green Check]

At the bottom of the interface, there is a blue footer with the text: 'Copyright © Sistema de Teste Online - 2024'.

Fonte: Autor

Ao acessar a seção “**Gerenciar Sugestões**” do painel lateral, o professor é direcionado a uma interface que oferece as funcionalidades de “**Adicionar Sugestão**” e “**Ver Sugestões**”, conforme ilustra a Figura 35. A opção “**Adicionar Sugestão**” conduz o docente a um formulário específico para o cadastro da sugestão personalizada, enquanto a opção “**Ver Sugestões**” permite a consulta e gestão das sugestões previamente registradas. Essa organização funcional da interface facilita a gestão do processo de cadastro e visualização, otimizando o tempo do professor e proporcionando uma experiência mais eficiente.

Figura 35 - Tela para Adicionar Sugestão ou Ver Sugestão cadastrada.



Fonte: Autor

Ao selecionar a opção "Adicionar Sugestão", o professor é direcionado a um formulário específico para o cadastro das sugestões de apoio ao aprendizado do aluno, conforme ilustra a Figura 36. Nesse formulário, o sistema permite a vinculação da sugestão a um curso pré-definido e a classificação do conteúdo de acordo com três níveis de proficiência: iniciante, intermediário e avançado. Essa classificação é determinada automaticamente com base no desempenho do aluno na avaliação, considerando o percentual de acertos obtidos. Alunos com menos de 40% de acertos recebem sugestões classificadas como "iniciante", aqueles com um percentual de acertos entre 41% e 90% recebem sugestões "intermediárias", e os que obtiveram mais de 90% de acertos recebem sugestões "avançadas". O sistema permite ao professor cadastrar até três tipos de materiais como sugestão: vídeos, imagens/PDF's<sup>11</sup> e *links* para plataformas de aprendizagem externas. Essa flexibilidade na criação de sugestões personalizadas contribui para a otimização do processo de ensino-aprendizagem, adaptando o conteúdo às necessidades específicas de cada aluno.

<sup>11</sup> PDF é a abreviação de *Portable Document Format* (formato portátil de documento). É um formato de arquivo versátil criado pela Adobe que proporciona uma maneira fácil e confiável de apresentar e compartilhar documentos em qualquer software, hardware ou sistema operacional usado pela pessoa que exibe o documento. O formato PDF agora é um padrão aberto mantido pela *International Organization of Standardization* (ISO). Documentos PDF podem conter links e botões, campos de formulário, áudio, vídeo e lógica de negócios. Disponível em: <https://www.adobe.com/br/acrobat/about-adobe-pdf.html>

Figura 36 - Tela de Cadastro das sugestões personalizadas.

The screenshot shows the 'Adicionar Sugestão' (Add Suggestion) form in the SISWEB system. The interface includes a dark sidebar on the left with the user profile 'Charles (Professor)' and navigation options: Dashboard, Gerenciar Cursos, Gerenciar Questões, and Gerenciar Sugestões. The main content area features a title 'Adicionar Sugestão' and a 'Sair' button in the top right. The form consists of three dropdown menus: 'Curso' (with 'Lógica de Programação' selected), 'Categoria' (with 'Iniciante' selected), and 'Sugestão' (with 'Sugestão 01' selected). Below these are three text input fields for suggestions, each with an example: 'Exemplo: Um curso em vídeo ou plataforma de aprendizado - www.alura.com.br', 'Exemplo: Um livro ou material em pdf - Lógica de Programação e Algoritmos com JavaScript', and 'Exemplo: Um especialista ou um canal especializado nas redes sociais - Canal PythonBrasil'. A green 'Salvar' button is located at the bottom left of the form. The footer contains the text 'Copyright © Sistema de Teste Online - 2024'.

Fonte: Autor

Após a conclusão do cadastro de uma nova sugestão e a confirmação do docente através do botão **“Salvar”**, o sistema procede ao armazenamento das informações em seu banco de dados. Em seguida, o professor é direcionado para uma tela de consulta e gestão das sugestões cadastradas, conforme ilustra a Figura 37. Essa interface permite ao professor visualizar de forma detalhada todas as sugestões criadas, bem como realizar edições e exclusões, garantindo a atualização constante do banco de dados e a otimização da experiência de aprendizagem dos alunos.

Figura 37 - Tela com as sugestões cadastradas pelo professor.

The screenshot shows the SISWEB interface for a professor named Charles. The main content area displays a table titled 'Sugestões' with the following data:

Curso	Categoria	Sugestão 01	Sugestão 02	Sugestão 03	👍	👎	Ação
Lógica de Programação	Iniciante		.pdf	<a href="https://hackathonbrasil.com.br/entenda-o-que-e-um-algoritmo-e-como-funciona/">https://hackathonbrasil.com.br/entenda-o-que-e-um-algoritmo-e-como-funciona/</a>	2	0	

Fonte: Autor

A implementação de sugestões personalizadas, aliada à utilização de tecnologias educacionais, representa um avanço significativo na personalização do ensino e do aprendizado. Ao oferecer materiais complementares e adaptados ao nível de conhecimento de cada aluno, a plataforma estimula a autonomia e a proficiência, ao mesmo tempo em que libera o professor para dedicar mais tempo a atividades que exigem maior interação humana. A possibilidade de os alunos fornecerem *feedback* sobre as sugestões cria um ciclo de aprimoramento contínuo, permitindo que os professores refinem o conteúdo e a forma como ele é apresentado, otimizando assim o processo de ensino-aprendizagem.

Ao longo deste capítulo, foram descritas as funcionalidades do sistema sob a perspectiva do professor e do aluno, os principais usuários da plataforma. O foco na experiência do usuário final permitiu evidenciar como o sistema oferece ferramentas intuitivas e eficazes para a criação de um ambiente de aprendizagem dinâmico e colaborativo. As funcionalidades destinadas ao administrador do sistema, embora essenciais para a manutenção e configuração da plataforma, não foram detalhadas neste trabalho, uma vez que estas se encontram além do escopo da pesquisa, que tem como objetivo principal analisar a experiência de ensino e aprendizagem dos professores e alunos.

## 7. RESULTADOS

Este capítulo apresenta os resultados da avaliação do Sistema *Web* de Ensino e Aprendizado de Lógica de Programação. O Sistema foi projetado para oferecer uma experiência interativa e eficaz aos estudantes que visa aprimorar suas habilidades fundamentais em programação. Serão detalhados os resultados dos testes de usabilidade realizados.

### 7.1. Caso de Estudo

Os testes foram realizados na escola SESI/SENAI que fica localizada na Travessa Mauriti, 3251 – Bairro: Marco, Belém-Pará, especificamente nos laboratórios de informática da instituição entre os meses de janeiro a maio de 2024. Os laboratórios de informática contam com 30 computadores *Desktops* com a seguinte configuração: Processadores Intel Core i5, 16GB de RAM e HD SSD de 512 GB, todos com o sistema operacional Windows® 11 versão 64 bits, monitores de 22 polegadas, mouse, teclado e acesso à internet.

A escolha da instituição de ensino SESI/SENAI foi motivada pelo fato de que ela já implementa a matriz do Novo Ensino Médio, conforme definido pela Base Nacional Comum Curricular (BNCC). Essa nova matriz permite a oferta de diferentes combinações curriculares, organizadas em itinerários formativos estratégicos, que proporciona uma flexibilização na organização curricular do Ensino Médio. Isso oferece aos estudantes novas opções de escolha, podendo ser agrupados em turmas com foco em áreas do conhecimento, formação técnica e profissional, ou ainda no desenvolvimento de competências e habilidades de diferentes áreas.

A escola SESI/SENAI oferece os seguintes itinerários formativos, organizados em turmas:

- Ciências da Natureza (CN),
- Ciências Humanas (CH),
- Matemática (MAT), e
- Formação Técnica e Profissional em Redes de Computadores, designada no SESI/SENAI como HT (Habilitação Técnica em Redes de Computadores).

Os alunos escolhem o itinerário formativo no início do 1º ano do ensino médio, baseando-se em suas preferências. Em relação as Unidades Curriculares (UC) elas são comuns, independente de este ser profissionalizante ou não. Por esse motivo, os alunos matriculados no 1º ano do ensino médio não foram selecionados para participar dos testes de usabilidade do sistema, uma vez que possuem um entendimento não mensurado em Lógica de Programação, pois não faz parte da sua formação.

Para os alunos do 2º ano do Novo Ensino Médio (NEM), a seleção foi feita com base em suas áreas de conhecimento distintas e no foco de seus itinerários, seja profissionalizante ou não. Isso permitirá uma avaliação do nível de conhecimento em Tecnologia da Informação, especialmente em relação à Lógica de Programação. Os alunos do itinerário HT já têm contato direto com as Unidades Curriculares (UC) de Lógica Computacional (equivalente a Algoritmos I) e Lógica de Programação (Python), o que não ocorre com os alunos de outros itinerários. Essa diferenciação nos perfis dos alunos selecionados possibilitará uma análise mais abrangente do sistema, considerando diferentes níveis de familiaridade com os conceitos de programação.

Para os alunos do 3º ano do Novo Ensino Médio (NEM), o processo de seleção foi semelhante ao dos alunos do segundo ano. No entanto, os estudantes desta série já têm um direcionamento mais definido, seja para aprimorar suas habilidades técnicas em áreas profissionalizantes ou para se preparar para o Exame Nacional do Ensino Médio (ENEM). Essa distinção reflete diferentes objetivos e focos de estudo entre os alunos do último ano que influencia sua participação nos testes de usabilidade do sistema.

O teste ocorrerá da seguinte forma: foi elaborado por professores da área de Tecnologia da Informação um conjunto de 10 questões de múltipla escolha, abordando os temas de Raciocínio Lógico, Lógica Computacional e Lógica de Programação, com diferentes níveis de dificuldade.

Para as turmas do 2º e 3º ano, o enfoque foi direcionado à Lógica Computacional e Lógica de Programação, com questões mais elaboradas que abordam conceitos de algoritmos, estrutura de dados, estruturas de decisão, laços de repetição, entre outros.

É importante ressaltar que as questões foram cuidadosamente selecionadas e validadas pelos professores, garantindo que abordem adequadamente os conceitos-

chave e os objetivos de aprendizagem estabelecidos no currículo escolar. Além disso, essa avaliação desempenha um papel importante na identificação de lacunas de conhecimento e áreas de melhoria tanto para os alunos quanto para melhorias no sistema.

A plataforma foi instalada localmente no servidor de arquivos da escola e acessada remotamente pelos alunos. Eles se inscreveram na plataforma e realizaram os testes, começando com os 30 alunos do 2º ano, que representavam todos os itinerários disponíveis. Os testes foram realizados nos dias de aulas que correspondiam a segunda-feira e terça-feira, no turno da manhã. Para esses testes, foram selecionados aleatoriamente 30 alunos do 3º ano de todos os itinerários que a instituição oferece, esse número de alunos selecionados totalizou 60 alunos.

Tabela 6 - Distribuição de alunos conforme itinerário formativo.

<b>Itinerário Formativo</b>	<b>2º Ano</b>	<b>3º Ano</b>
Formação Profissional - Redes de Computadores (HT)	10	10
Ciências da Natureza e suas Tecnologias (CN)	10	10
Matemática e suas Tecnologias (MAT)	5	5
Ciências Humanas e Sociais (CH)	5	5
<b>TOTAL DE ALUNOS</b>	<b>30</b>	<b>30</b>

Fonte: Autor

Após a conclusão dos testes, os alunos foram convidados a preencher um questionário abordando 5 questões relacionadas à usabilidade do sistema. Esse questionário foi utilizado para avaliar a eficácia do sistema. A diversidade dos alunos participantes, com diferentes níveis de conhecimento em tecnologia da informação, proporcionou uma avaliação abrangente do sistema. Essa abordagem permitiu uma análise quantitativa para determinar a utilidade do sistema como ferramenta de ensino de Lógica de Programação e sua facilidade de uso para os alunos. Os resultados desse questionário foram importantes para identificar pontos fortes e áreas de melhoria do sistema, contribuindo assim para aprimorar sua usabilidade e eficácia no ensino.

## 7.2. Avaliação dos Resultados

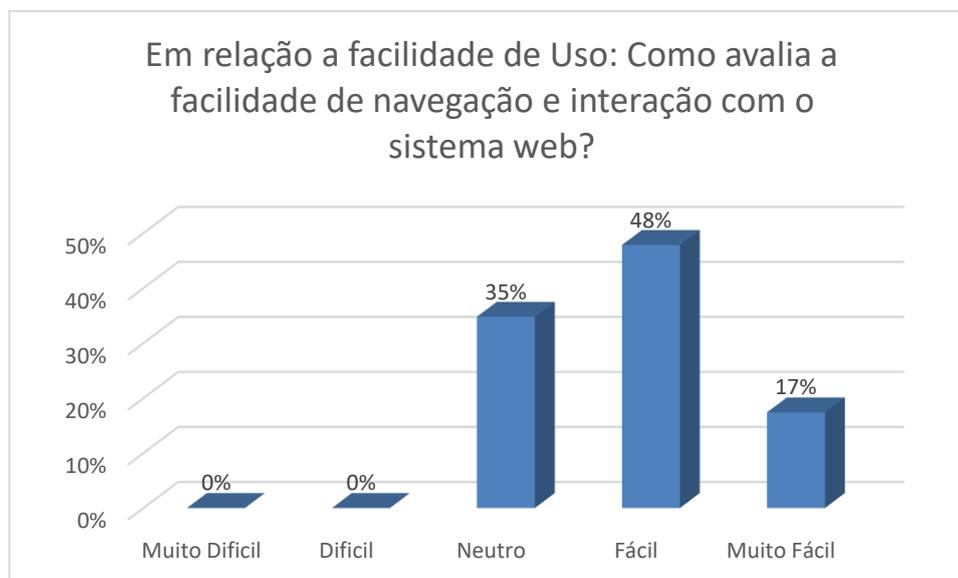
A eficácia de um Sistema Web de Ensino de Lógica de Programação é intrinsecamente ligada à experiência do usuário. Conforme destacado por Shneiderman (2010, p. 270) em seu livro *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, para avaliar de forma abrangente a utilidade e a adaptabilidade do sistema, é essencial ouvir diretamente aqueles que o utilizam.

Para entender melhor como os usuários se sentem ao usar o sistema, realizou-se uma pesquisa *online* com as turmas do Novo Ensino Médio dos seguintes itinerários formativos, CN, MAT, CH e HT da Escola Sesi-Belém/PA. As perguntas da pesquisa abordaram aspectos da usabilidade do sistema, incluindo a interface do usuário, a clareza das sugestões e a eficácia geral do sistema no apoio ao ensino e aprendizado dos alunos.

Essa abordagem de pesquisa visa coletar *feedback* valioso diretamente dos usuários-alvo. Essas informações podem ser usadas para melhorar a eficiência e a relevância do sistema para as necessidades específicas do contexto educacional. Ao analisar as respostas da pesquisa, identificamos áreas que precisam de melhorias imediatas e orientar futuras iterações do sistema. Isso garantirá que o sistema continue a evoluir para atender às expectativas e necessidades dos alunos.

A Figura 38 mostra a distribuição das respostas a questão “Como avalia a facilidade de navegação e interação com o sistema *web*?” indica que a maioria dos alunos (65%) avaliou a facilidade de uso do sistema como "muito fácil" ou "fácil". Apenas 35% dos alunos avaliaram a facilidade de uso como “neutro”, “difícil” ou “muitodifícil”, como mostra o gráfico.

Figura 38 - Resultado em gráfico da primeira questão do questionário diagnóstico.



Fonte: Autor

Essa distribuição de respostas pode ser explicada por uma série de fatores, como, os alunos que participaram da pesquisa são estudantes do Novo Ensino Médio itinerário profissionalizante Técnico em Redes de Computadores, bem como de outras áreas de formação. Esses alunos geralmente têm um alto nível de familiaridade com computadores e tecnologias digitais. Eles estão habituados a usar interfaces gráficas e menus intuitivos. Isso pode explicar por que a maioria dos alunos avaliou a facilidade de uso do sistema como "muito fácil" ou "fácil". Eles foram capazes de navegar pelo sistema com facilidade e encontrar as informações que precisavam.

O sistema foi usado em um contexto educacional, onde os alunos estavam motivados a aprender sobre o conteúdo. Eles estavam interessados em encontrar recursos que pudessem ajudá-los a aprender sobre Lógica Computacional e Lógica de Programação. Essa motivação pode ter ajudado os alunos a superar quaisquer desafios de usabilidade que o sistema possa ter apresentado.

O sistema foi projetado com foco na usabilidade, com menus e controles intuitivos. Os menus foram organizados de forma lógica e os controles foram fáceis de entender. Esse design intuitivo pode ter facilitado a navegação pelo sistema para os alunos. Eles foram capazes de encontrar as informações que precisavam sem ter que pensar muito. A combinação desses fatores pode explicar por que a maioria dos alunos avaliou a facilidade de uso do sistema como "muito fácil" ou "fácil".

Os resultados da pesquisa corroboram os resultados de outros estudos que investigaram a facilidade de uso de sistemas *web*. Por exemplo, um estudo realizado com alunos do ensino médio encontrou resultados semelhantes, com a maioria dos alunos avaliando a facilidade de uso do sistema como "muito fácil" ou "fácil".

É importante notar que esses resultados são baseados em uma amostra relativamente pequena de alunos. Para obter conclusões mais robustas, seria necessário realizar pesquisas com um número maior de alunos.

### **7.2.1. Teste Diagnóstico**

Para entender melhor como os alunos se sentem ao usar o sistema, realizou-se o teste diagnóstico com as turmas do Novo Ensino Médio dos seguintes itinerários formativos, CN, MAT, CH e HT da Escola Sesi-Belém/PA. As perguntas abordaram aspectos dos conhecimentos necessários para a unidade curricular Lógica Computacional, seguindo a Matriz de Referência Curricular do SENAI, conforme pode ser visto na Tabela 6.

De acordo com o SENAI/DN (2021, p. 3) define:

As matrizes de referência curricular do Novo Ensino Médio foram construídas na perspectiva de áreas de conhecimento e módulos do itinerário formativo, baseado no desenvolvimento de competências, pressupondo que os objetos do conhecimento apresentam conteúdo para a construção de habilidades, que é o saber fazer determinada ação.

Tabela 7 - Matriz de Referência Curricular – Habilitação Técnica em Redes de Computadores.

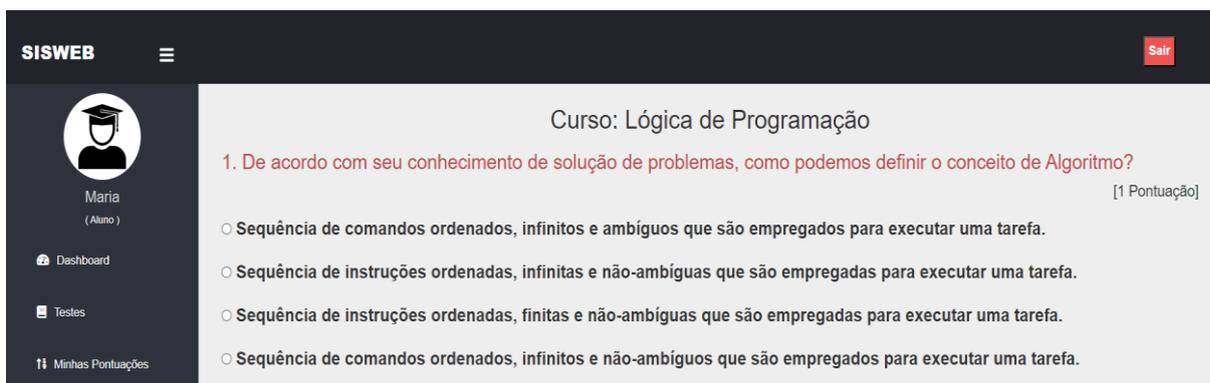
<b>FORMAÇÃO PROFISSIONAL – MÓDULO INTEGRADOR BÁSICO</b>		
<b>Unidade Curricular: Lógica Computacional (60h)</b>		
<b>Competências / Objetivo geral</b>	<b>Habilidades/Capacidades/Fundamentos Técnico-científicos</b>	<b>Ano</b>
		<b>2</b>
C1- Propiciar o desenvolvimento de fundamentos técnicos e científicos relativos à lógica computacional e suas aplicações, que subsidiarão o desenvolvimento das capacidades técnicas dos cursos técnicos de referência, bem como, as capacidades sociais, organizativas e metodológicas	H1- Interpretar situações problemas computacionais	
	H2- Utilizar técnicas de abstração para resolução de problemas	
	H3- Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo	
	H4- Aplicar expressões aritméticas, relacionais e lógicos para sistemas computacionais.	
	H5- Aplicar lógica de programação para resolução dos problemas	
	H6- Diferenciar os processos de compilação e interpretação	
	H7- Empregar as estruturas de controle e repetição adequadas à lógica dos algoritmos	
<b>Conhecimentos</b>		
Abstração lógica; Introdução ao Raciocínio Lógico; Tabela Verdade; Fundamentos da Programação; Programação estruturada.; Variáveis; Constantes; Operadores; Padrões de nomenclatura e convenções de linguagem; Ferramentas para elaboração de algoritmos; Instruções de entrada e saída de dados; Estrutura de repetição.		

Fonte: SENAI/DN (2021)

O Teste Diagnóstico teve como objetivo analisar a quantidade de respostas corretas e incorretas, respondidas diretamente pelos alunos. Essas respostas foram analisadas para verificar o nível de conhecimento sobre a unidade curricular escolhida.

A questão "**De acordo com seu conhecimento de solução de problemas, como podemos definir o conceito de Algoritmo?**", conforme pode ser visto na Figura 39, apresenta-se como uma forma de realizar o diagnóstico o conhecimento dos alunos do Novo Ensino Médio em relação à Lógica de Programação ou qualquer disciplina que envolva a necessidade de reconhecer os conceitos fundamentais da Ciência da Computação, especialmente no que tange à compreensão do conceito fundamental de algoritmo. Essa escolha se justifica, por permitir realizar a avaliação da compreensão conceitual do aluno. A questão exige que os alunos não apenas reproduzam definições decoradas, mas sim demonstrem uma compreensão profunda do que um algoritmo representa. Ao relacionar o algoritmo à solução de problemas, os alunos precisam articular o conceito com sua aplicação prática, evidenciando seu nível de apreensão.

Figura 39 - Exemplo de questão utilizada no teste diagnóstico.



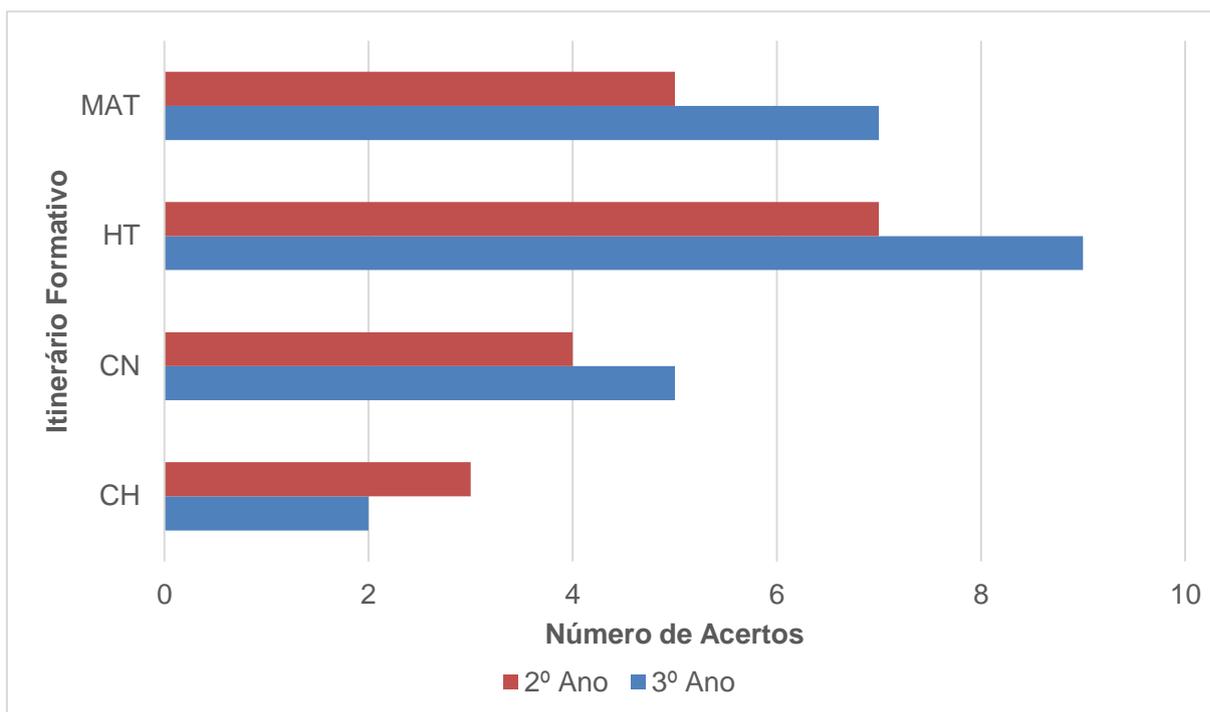
Fonte: Autor

A partir da finalização do teste foi possível analisar os resultados obtidos, observou-se 4 situações bem distintas:

- **Situação 1:** Os alunos do 2º e 3º anos do itinerário profissionalizante, denominado Habilitação Técnica (HT) em Redes de Computadores no SENAI, apresentaram o maior número de acertos, entre 7 e 9 questões respondidas corretamente, resultando em um aproveitamento de 80% das questões disponíveis.
- **Situação 2:** Os alunos do itinerário de Matemática e suas Tecnologias tiveram um aproveitamento mediano, com um número de acertos entre 5 e 7 questões, alcançando uma média de 60% de acertos.
- **Situação 3:** Os alunos do itinerário de Ciências da Natureza e suas Tecnologias obtiveram um aproveitamento abaixo da média, com um número de acertos entre 4 e 5 questões, correspondendo a 45% do total de questões.
- **Situação 4:** Os alunos do itinerário de Ciências Humanas e Sociais Aplicadas tiveram um aproveitamento baixo, com um número de acertos entre 2 e 3 questões respondidas corretamente, resultando em uma média de 25% de acertos do total de questões.

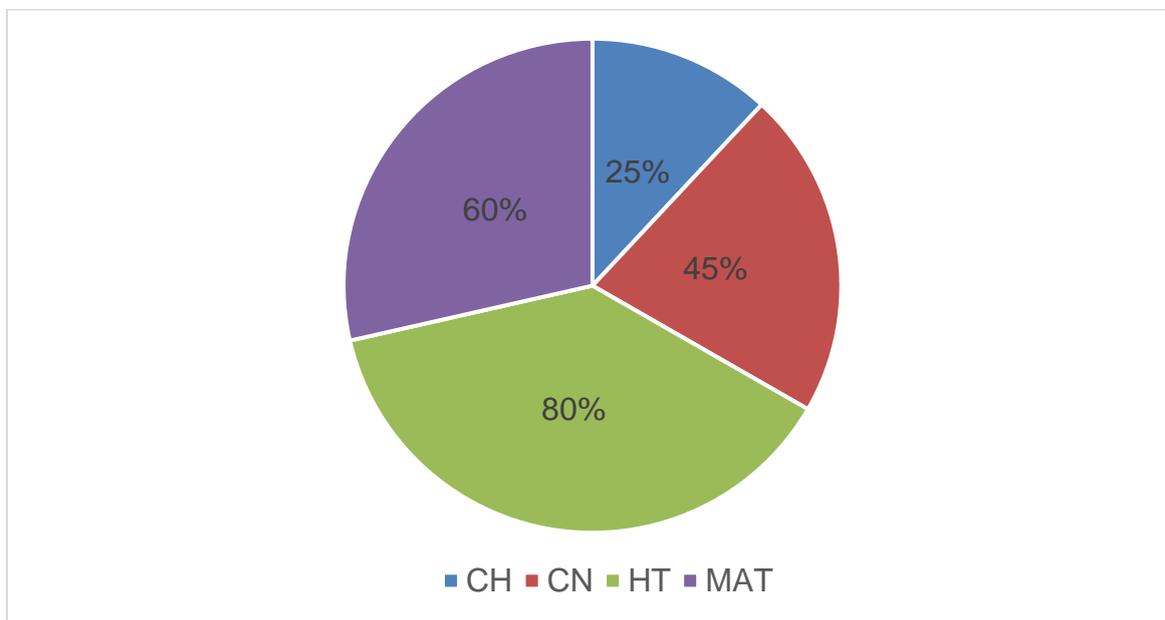
As situações acima descritas podem ser observadas nas Figura 40. e 41, respectivamente.

Figura 40 - Número de Acertos por Questão para Cada Itinerário Formativo.



Fonte: Autor

Figura 41 - Distribuição das Médias de Acertos por Itinerário Formativo.



Fonte: Autor

É importante ressaltar que as questões usadas no Sistema *Web* foram idênticas para alunos de diferentes itinerários formativos.

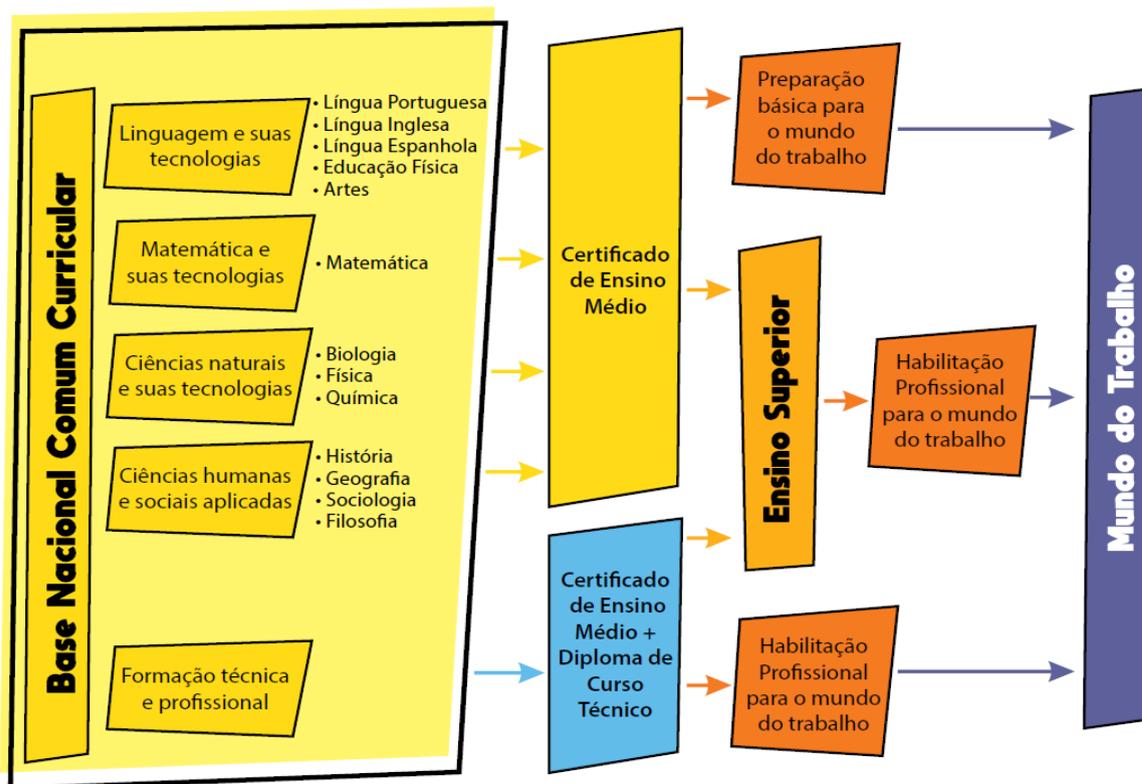
Ao fazer uma análise mais profunda, observou-se que os alunos do itinerário

HT em Redes de Computadores, Matemática e Suas Tecnologias e Ciências da Natureza e Tecnologias demonstraram maior facilidade em responder perguntas sobre algoritmos e lógica de programação, o que pode ser atribuído ao desenvolvimento de habilidades analíticas, lógicas e matemáticas, características predominantes nessa área de estudo.

Por outro lado, os alunos do itinerário de Ciências Humanas e Sociais Aplicadas tendem a se destacar em áreas que exigem compreensão contextual e interpretação de fenômenos sociais e históricos. Essa distinção reforça a importância de adaptar o ensino às competências específicas de cada área de conhecimento, permitindo um aprendizado mais direcionado e eficaz.

Como resultado de toda a análise realizada, este trabalho subsidia a comprovação da proposta apresentada para Novo Ensino Médio de alinhar as competências dos alunos com suas áreas de interesse e aptidão, conforme pode ser visto na Figura 42. As respostas dos alunos do SENAI-CEDAM, os quais já seguem a proposta do Novo Ensino Médio, comprova que os alunos que escolhem a área de Habilitação Técnica Profissional em Redes de Computadores, Matemática e suas Tecnologias e Ciências da Natureza e suas Tecnologias tem mais facilidade em responder perguntas sobre algoritmos e Lógica de Programação devido ao desenvolvimento de suas habilidades analíticas, lógicas e matemáticas, que é o foco da unidade curricular Lógica Computacional, por isso os alunos obtiveram maior índice de acerto. Enquanto alunos do itinerário de Ciências Humanas e Sociais Aplicadas não tem esta mesma facilidade ao tema proposto porque tendem a se destacar em áreas que exigem compreensão contextual e interpretação de fenômenos sociais e históricos, visto que não é o foco da unidade curricular Lógica Computacional e por isso os alunos deste itinerário obtiveram menor índice de acerto as questões.

Figura 42 - Fluxo de formação do Novo Ensino Médio.



Fonte: SENAI/DN (2018)

Os benefícios de implementar um teste de avaliação diagnóstica são inúmeros. Primeiramente, ele oferece um diagnóstico preciso das habilidades dos alunos, identificando áreas que necessitam de melhorias específicas. Em segundo lugar, os professores podem criar as sugestões personalizadas que ajudam os alunos a focar nos tópicos específicos que precisam de mais atenção, promovendo um aprendizado mais eficiente. Por fim, o *feedback* e as sugestões de estudo mantêm os alunos engajados e motivados a melhorar continuamente, incentivando um ambiente de aprendizado dinâmico e interativo.

Ao incorporar um "Teste Diagnóstico" no sistema *web* para Ensino e Aprendizado de Lógica de Programação, este trabalho destaca a importância de um diagnóstico e de sugestões personalizadas para o aprendizado. Essa abordagem não só otimiza a aprendizagem, mas também proporciona uma experiência educacional mais envolvente e adaptada às necessidades dos estudantes do Novo Ensino Médio. Dessa forma, a implementação de tais ferramentas representa uma contribuição significativa para o avanço da tecnologia educacional e o aprimoramento do processo

de ensino da lógica de programação no cenário educacional atual.

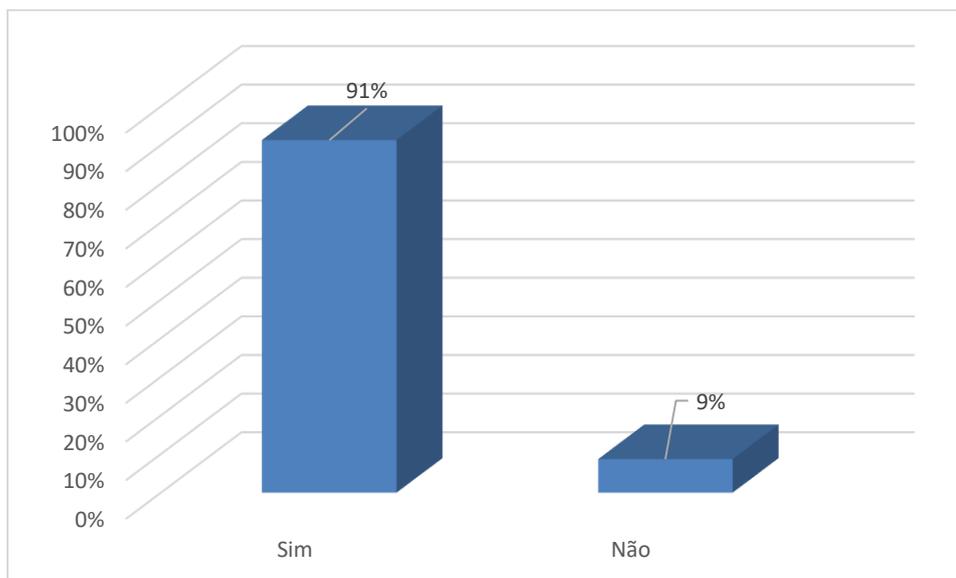
### 7.2.2. Avaliação das sugestões personalizadas

O presente estudo também avaliou a eficácia das sugestões personalizadas fornecidas pelo Sistema *Web* para Ensino e Aprendizado de Lógica de Programação. Após a realização do teste diagnóstico, os alunos receberam sugestões específicas de conteúdo, como vídeos e links de apoio, com o objetivo de aprimorar suas habilidades e conhecimentos na disciplina de lógica de programação.

Para avaliar a eficácia das sugestões personalizadas, foi realizado um acompanhamento dos alunos utilizando o mesmo questionário que colheu respostas sobre a usabilidade do sistema. Uma das perguntas feitas foi: **“Em relação às sugestões oferecidas pelo sistema, as sugestões foram relevantes?”**. A pesquisa envolveu alunos de diferentes itinerários formativos, incluindo Ciências da Natureza, Matemática, Ciências Humanas e Sociais Aplicadas, e Habilitação Técnica em Redes de Computadores. Cada grupo de alunos realizou o teste diagnóstico e o preenchimento do questionário de usabilidade do sistema.

Os resultados mostraram que a maioria dos alunos, 91% consideraram as sugestões personalizadas úteis e relevantes para o aprimoramento de seus conhecimentos em lógica de programação, enquanto 9% declararam que as sugestões não foram relevantes, como pode ser visto na Figura 43. Em particular, os alunos dos itinerários Habilitação Técnico, Ciências da Natureza e Matemática demonstraram um maior aproveitamento das sugestões, refletindo uma maior facilidade em assimilar conteúdos relacionados a algoritmos e lógica, conforme indicado pelos seus resultados positivos no teste diagnóstico.

Figura 43 - Avaliação dos alunos em relação as sugestões personalizadas.



Fonte: Autor

Por outro lado, alunos dos itinerários de Ciências Humanas e Sociais Aplicadas encontraram mais dificuldades, mas relataram que as sugestões de vídeos e links de apoio ajudaram a tornar os conceitos de lógica de programação mais acessíveis e compreensíveis. Esse *feedback* destaca a importância de adaptar os recursos educacionais às necessidades específicas de cada grupo de estudantes, considerando suas áreas de maior facilidade e dificuldade.

A avaliação das sugestões personalizadas revelou que tais sugestões podem desempenhar um papel importante no suporte ao aprendizado de lógica de programação, especialmente quando integradas de maneira sensível ao contexto educacional dos alunos. O uso de recursos diversificados e direcionados permite que os alunos recebam um apoio mais direcionado e eficaz, aumentando suas chances de sucesso na disciplina.

Além disso, a personalização das sugestões de conteúdo contribui para uma experiência de aprendizado mais envolvente e motivadora, o que é essencial para manter o interesse dos alunos e fomentar um ambiente educacional positivo. A eficácia dessa abordagem reforça a necessidade de desenvolvimento contínuo de sistemas educacionais que utilizem tecnologias avançadas para adaptar o ensino às necessidades individuais dos alunos.

A implementação de sugestões personalizadas no Sistema *Web* para Ensino e Aprendizado de Lógica de Programação mostrou-se eficaz em apoiar os alunos na

superação de suas dificuldades e no aprimoramento de suas habilidades. Este estudo confirma a relevância de abordagens educacionais personalizadas e adaptativas no ensino de disciplinas complexas, como a lógica de programação, e sugere que tais métodos podem ser amplamente benéficos em diversos contextos educacionais.

## 8. CONCLUSÃO

Os fundamentos da programação desempenham um papel crucial na formação dos estudantes do Novo Ensino Médio, especialmente aqueles que buscam carreiras na área de Tecnologia da Informação e Desenvolvimento de Sistemas. Apesar de os desafios iniciais em programação serem introduzidos nos estágios iniciais desses cursos de nível superior, observa-se que muitos estudantes enfrentam dificuldades significativas, impactando negativamente seu desempenho em disciplinas mais avançadas.

Com o objetivo de melhorar o rendimento dos estudantes nesse contexto específico, foi desenvolvida um sistema *web* para construção de testes e indicação de materiais didáticos para ensino de lógica de programação. É possível concluir que a ferramenta é útil por ter obtido um índice de usabilidade superior a 4 na escala de *likert*. Após aplicação em ambiente educacional, é possível concluir que a ferramenta atende os requisitos para ser considerada uma ferramenta de ensino.

Para avaliar a usabilidade do sistema, foram aplicados questionários aos alunos, focando em aspectos como facilidade de uso, eficiência das sugestões oferecidas pelo sistema e satisfação geral com a plataforma. Os resultados indicaram que 65% dos alunos consideraram o sistema intuitivo e fácil de usar, enquanto 91% afirmaram que o *feedback* recebido contribuiu significativamente para o seu aprendizado. As sugestões coletadas serão fundamentais para orientar as futuras melhorias.

Para aprimorar continuamente o sistema, planeja-se incluir uma métrica de avaliação por meio de um formulário *on-line*, permitindo que os usuários compartilhem suas experiências e sugestões. Essa métrica será essencial para direcionar futuras melhorias e ajustes, garantindo que o sistema atenda cada vez mais às necessidades específicas dos estudantes.

Além disso, pretende-se incorporar ao sistema um algoritmo de recomendação automatizado. Este algoritmo, a ser definido, buscará aprimorar a personalização das recomendações, considerando o desempenho passado do aluno, preferências individuais e padrões de aprendizado. A automação desse processo não apenas tornará as recomendações mais eficientes, mas também permitirá uma adaptação

contínua com base no *feedback* coletado.

Antevemos futuras melhorias como a avaliação da aprendizagem dos usuários e expansões no escopo do projeto, com foco especial em atividades colaborativas e cooperativas, alinhadas às diretrizes educacionais do Novo Ensino Médio, que foram o cerne deste trabalho. Agradece-se a todos os participantes e espera-se continuar aprimorando o sistema para proporcionar uma experiência de aprendizado cada vez mais eficaz e personalizada.

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

ABURIZAIZAH, Saeed Jameel, and Tahany Abdulaziz Albaiz. "Review of the Use and Impact of Nano-Learning in Education." 4th International Conference on Research in Education. 2021.

ALVES, Paula Trajano de Araújo; SILVA, Solonildo Almeida da; JUCÁ, Sandro César Silveira. O percurso histórico do ensino médio brasileiro (1837-2017). *Revista Contemporânea de Educação*, v. 17, n. 39, p. 137-155, mai/ago. 2022. Disponível em: <http://dx.doi.org/10.20500/rce.v17i39.45994>. Acesso em: 09/07/2024.

AMARO, C., Sarah, D., Valverde, R., & Jorge, C. (2007). *Metodologías Ágiles*. Trujillo, Perú.

ARIMOTO, M.; OLIVEIRA, W. Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação. *Anais do Workshop sobre Educação em Computação (WEI)*, 12 jul. 2019.

AVASTHI, A. Maximising Student Potential through Data — Aditi Avasthi of Embibe Tells Her Story! Disponível em: <<https://blog.embibe.com/maximising-student-potential-through-data-aditi-avasthi-of-embibe-tells-her-story-420046dadd00>>. Acesso em: 28 mar. 2023.

BACINGER, T. (2023). What is bootstrap? a short bootstrap tutorial on the what, why, and how. Disponível em: <<https://www.toptal.com/bootstrap/what-is-bootstrap-responsive-sites>>. Acesso 09/06/2023.

BIGOLIN, N. M. et al. Metodologias Ativas de Aprendizagem: um relato de experiência nas disciplinas de programação e estrutura de dados. *Research, Society and Development*, v. 9, n. 1, p. e74911648–e74911648, 1 jan. 2020.

BOOCH, G. et al. *Object-Oriented Analysis and Design with Applications*. [s.l.] Pearson Education, 2007.

*Bootstrap* (2023). Documentação *Bootstrap*. Disponível em: <https://getbootstrap.com/>. Acesso 08/07/2023).

BORGES, M. A. F. (2000). Avaliação de uma metodologia alternativa para a aprendizagem de programação. *Workshop de Educação em Computação Congresso anual da SBC*.

BUCHEM, I., Hamelmann, H. (2010). *Microlearning: a strategy for ongoing professional development*.

CAZELLA, S. C. et al., (2009) Recomendação de objetos de aprendizagem empregando filtragem colaborativa e competências. In: *Anais do Simpósio Brasileiro de Informática na Educação*.

Debrauwer, L & Heyde, F. (2009). *UML 2 Iniciación, ejemplos y ejercicios corregidos, segunda edición*. de San Martín, O. C. (2005). Disponível em: <<http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaCONIX.pdf>>

Acesso em: 20/08/2023.

DE CAMPOS, A.; CÉSAR CAZELLA, S. *Learning Analytics* em processos de personalização de aprendizagem: uma revisão sistemática de literatura. *RENOTE*, v. 16, n. 1, 21 ago. 2018.

DE, J.; LIMA, S. Uma análise das dificuldades de aprendizagem da Lógica de Programação no Curso Técnico Integrado em Informática do IFBA - Campus Jacobina. [s.l: s.n.]. Disponível em: <[http://www.repositorio.ifba.edu.br/jspui/bitstream/123456789/131/1/tcc\\_Jorge%20de%20Souza%20Lima%20Junior.pdf](http://www.repositorio.ifba.edu.br/jspui/bitstream/123456789/131/1/tcc_Jorge%20de%20Souza%20Lima%20Junior.pdf)>.

De Leone, L. (2017). *Bootstrap*: o que é, porque usar e como começar com o framework. Disponível em: <<https://becode.com.br/bootstrap-o-que-e-porque-usar-e-como-comecar>>. Acesso 09/07/2023.

De Oliveira, J. L. ENSINAR E APRENDER COM AS TECNOLOGIAS DIGITAIS EM REDE: POSSIBILIDADES, DESAFIOS E TENSÕES. *Revista Docência e Cibercultura*, v. 2, n. 2, 25 jun. 2018.

Demchenko, M. (2019). Six huge tech companies that use python: Does it fit your project? Disponível em: <<https://ncube.com/blog/6-huge-tech-companies-that-use-python-does-it-fit-your-project>>; Acesso em: 09/06/2023.

Doug Rosenberg and Kendall Scott, *Use Case Driven Object Modeling with UML: A Practical Approach* (New York: Addison-Wesley, 1999).

Dragos-Paul, P. and Altar, A. (2014). *Designing an MVC Model for Rapid Web Application Development*. *Procedia Engineering*, 69:1172–1179.

Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025. Disponível em: <<https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>>. Acesso em: 25/04/2022.

FORBELLONE, André Luiz Villar; EBERSPÄCHER; Henri Frederico. *Lógica de programação - A construção de Algoritmos e Estrutura de dados*. São Paulo: Makron Books, 1993.

Foundation, D. S. (2020a). Documentação do Django. Disponível em: <<https://docs.djangoproject.com/en/4.2/>>. Online, Acesso em: 28/06/2023.

Foundation, P. S. (2020b). The python package. Disponível em: <<https://docs.python.org/3>>; Acesso em: 09/06/2023.

Gabrielli, S., Kimani, S., & Catarci, T. (2006). The Design of Microlearning Experiences: A Research Agenda. In: T. Hug, M. Lindner, & P. A. Bruck, (Eds.), *Microlearning: Emerging Concepts, Practices and Technologies after E-Learning: Proceedings of Microlearning Conference 2005: Learning & Working in New Media* (pp. 45-53). Innsbruck, Áustria: Innsbruck University Press.

Git e Github: o que são, como configurar e primeiros passos. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-git-github>>. Acesso em: 12/08/2023.

Gramming, A. C.; Ejemyr, E.; Thunell, E. Implementing Nano-Learning in the Law Firm. \*Legal Information Management\*, v. 19, n. 4, p. 241-246, dez. 2019. DOI: 10.1017/s1472669619000562.

Hug, T. (2007). Didactics of Microlearning: Concepts, Discourses and Examples. Münster: Waxmann Verlag, 424 p. Hug, T. (2006). Microlearning: A New Pedagogical Challenge (Introductory Note). In T. Hug, M. Lindner, & P. A. Bruck, (Eds.), Microlearning: Emerging Concepts, Practices and Technologies After E-Learning: Proceedings of Microlearning Conference 2005: Learning & Working in New Media (pp. 8-11). Innsbruck, Áustria: Innsbruck University Press.

ISBRASIL (2017). O que é bootstrap?. Disponível em: <<https://www.isbrasil.info/blog/o-que-e-bootstrap.html#:~:text=Bootstrap%20%C3%A9%20um%20framework%20front,CSS%20para%20que%20tudo%20funcione>>. Acesso 21/07/2023.

jQuery, F. (2023). jquery write less, do more. Disponível em: <<https://jquery.com/>>; Acesso 22/06/2023.

Khlaif ZN, Salha S. Using TikTok in Education: A Form of Micro-learning or Nano-learning? Interdiscip J Virtual Learn Med Sci. 2021;12(3):213-218. doi: 10.30476/ijvlms.2021.90211.1087.

Kitchenham, B. Pfleeger, S. Personal Opinion Surveys. In: Guide to Advanced Empirical Software Engineering. Springer, 2008. Cap. 3, p. 63-92.

LUM, L.; BRADLEY, P.; RASHEED, N. Accommodating learning styles in international bridging education programs. Higher Education, Skills and Work-based Learning, v. 1, n. 2, p. 147-168, 2011.

MÉDIO, E. EDUCAÇÃO É A BASE. [s.l: s.n.]. Disponível em: <<http://portal.mec.gov.br/docman/abril-2018-pdf/85121-bncc-ensino-medio/file>>. Acesso: 08/07/2024.

MEYLIANA et al. The Enhancement of Learning Management System in Teaching Learning Process with the UTAUT2 and Trust Model. 2019 International Conference on Information Management and Technology (ICIMTech), ago. 2019.

MURAT KARABULUT; ISLAM MAYDA. Web Based Algorithm Exercise and Assessment Management System for Computer Programming Students: AlgoBug. 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 22 out. 2020.

Nikou, S. A., & Economides, A. A. (2018). Mobile-Based micro-Learning and Assessment: Impact on learning performance and motivation of high school students. *Journal of Computer Assisted Learning*, 34(3), 269-278. doi:10.1111/jcal.12240

Novo Ensino Médio 2022: entenda tudo que muda. Disponível em: <<https://www.portaldaindustria.com.br/industria-de-a-z/novo-ensino-medio/#bncc>>.

Acesso em: 2 mar. 2022.

Novo Ensino Médio - perguntas e respostas. Disponível em: <<http://portal.mec.gov.br/component/content/article?id=40361>>. Acesso em: 21/11/2023

O'Neil, E. J. (2008). Object/relational mapping 2008: hibernate and the entity data model (edm). In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, SIGMOD '08, pages 1351–1356, Vancouver, Canada. Association for Computing Machinery.

OVERFLOW, S. Developer Survey Results 2019. Disponível em: <<https://insights.stackoverflow.com/survey/2019>>. Acesso em: 03/08/2023

OVERFLOW, S. 2020 developer survey. 2020. Disponível em: <<https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-professional-developers>>. Acesso em: 03/08/2023.

PAPERT, S. *Mindstorms: children, computers and powerful ideas*. New York; London: Harvester Wheatsheaf, 1993.

PERLIN, R.; MACEDO, R. T.; SILVEIRA, S. R. Uma Abordagem Construtivista no Ensino de Algoritmos e Lógica de Programação com o Auxílio de uma Ferramenta Gamificada. *E-Xacta*, v. 12, n. 1, p. 29, 23 jun. 2019.

Pritchard, A. (2017). *Ways of learning: Learning theories for the classroom (4th Ed.)*. Routledge. <https://doi.org/10.4324/9781315460611>.

RESNICK, M.; ROBINSON, K. *Lifelong kindergarten: cultivating creativity through projects, passion, peers, and play*. Cambridge, Massachusetts: The Mit Press, 2018.

SANTIAGO, C. et al. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. p. 97–120, 10 set. 2020.

SANTOS, H. C. S. dos.; RODRIGUES JÚNIOR, M. S. The strategies used in the teaching-learning process before and after the advent of the COVID-19 pandemic: concepts of accounting students and teachers. *Research, Society and Development*, [S. l.], v. 11, n. 11, p. e238111129519, 2022. DOI: 10.33448/rsd-v11i11.29519. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/29519>. Acesso em: 1 abr. 2023.

SEDGEWICK, R. *Algorithms*. 4th ed. Upper Saddle River, NJ: Addison-Wesley Professional, 2011.

SILVA, T. R., et al. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, [s.l.], v. 23, n. 01, p.182-196, 30 abr. 2015. Comissão Especial de Informática na Educação. <http://dx.doi.org/10.5753/rbie.2015.23.01.182>.

Shneiderman, B.; Plaisant, C. (2010). *Designing the user interface: strategies for effective human-computer interaction*. 5. ed. Boston: Pearson.

SOULEIMAN, A. H. Orchestration and Adaptation of Learning Scenarios — Application to the Case of Programming Learning / Teaching. 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), out. 2017.

Stephens, R. (2020). Redmonk slackchat: January 2020 programminglanguage rankings. Procedia Engineering.

Valéria, Márcia & França, César. (2013). Ferramentas de Auxílio ao Aprendizado de Programação: Um Estudo Comparativo

WHITING, K. These are the top 10 job skills of tomorrow – and how long it takes to learn them. Disponível em: <<https://www.weforum.org/agenda/2020/10/top-10-work-skills-of-tomorrow-how-long-it-takes-to-learn-them/>>. Acesso em: 29/07/2022. Why alta. Disponível em: <<https://www.knewton.com/why-alta/>>. Acesso em: 28 abr. 2022.

Van Rossum, G. (2020). Python Programming Language. Python Software Foundation. Disponível em: <https://www.python.org/>

Holovaty, A., Kaplan-Moss, J. (2016). The Django Book. Disponível em: <https://djangobook.com/>

HOLOVATY, A.; KAPLAN-MOSS, J. The Django Administration Site. The Definitive Guide to Django, p. 83–94, 2008.

Otto, M., Thornton, J., Hazzard, D., & @mdo. (2020). Bootstrap. Disponível em: <https://getbootstrap.com/>