



*Uma abordagem para a aprendizagem de  
Programação baseada em Chats Inteligentes*

**Cláudio Roberto Ribeiro**

Setembro / 2024

Dissertação de Mestrado em Ciência da  
Computação

# **Uma abordagem para a aprendizagem de Programação baseada em Chats Inteligentes**

Esse documento corresponde à dissertação apresentada à Banca Examinadora no curso de Mestrado em Ciência da Computação do Centro Universitário Campo Limpo Paulista.

Campo Limpo Paulista, 10 de setembro de 2024.

Cláudio Roberto Ribeiro

Rodrigo Bonacin (Orientador)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## **Resumo**

*Aprender programação apresenta desafios para muitos alunos, incluindo dificuldades na resolução de problemas, no desenvolvimento do raciocínio lógico e no pensamento abstrato. Chats inteligentes podem ser utilizados para aumentar a eficiência e eficácia deste processo de aprendizagem. Os chats inteligentes são sistemas avançados que simulam a conversa humana por meio de texto ou voz, utilizando para isso, técnicas de processamento de linguagem natural e aprendizado de máquina. No contexto da educação, os chats inteligentes podem atuar como tutores virtuais, auxiliando os alunos na aquisição de conhecimentos e habilidades. Os chats inteligentes podem ser uma alternativa, oferecendo uma experiência de aprendizado mais interativa, personalizada e envolvente. No entanto, é importante ressaltar que o aprendizado de programação baseado em chats inteligentes não deve ser visto como um substituto para os professores, mas sim, como uma ferramenta complementar. O principal objetivo desta dissertação é propor uma abordagem de ensino de programação, utilizando chats inteligentes como ferramenta de apoio ao ensino integrada a um Sistema de Gestão da Aprendizagem. Assim, esta dissertação apresenta primeiramente uma revisão da literatura, seguida da definição de uma abordagem de aprendizado de programação, o desenvolvimento de um sistema e um estudo de caso com alunos do ensino técnico em Informática. O estudo de caso utilizou o sistema proposto e investigou a aceitação da tecnologia, bem como avaliou os resultados dos alunos em testes. Os resultados destacaram uma boa aceitação da tecnologia, bem como benefícios e desafios associados. Espera-se contribuir com o avanço do conhecimento na área de educação em computação e abrir caminhos para esforços futuros, envolvendo chats inteligentes como ferramentas de apoio educacional.*

**Palavras-chave:** Aprendizado de Programação, Chats Inteligentes, Informática na Educação, Inteligência Artificial na Educação.

## ***Abstract***

*Programming learning presents challenges for many students, including difficulties in solving problems, developing logical reasoning, and abstract thinking. Intelligent chat systems can be used to increase the efficiency and effectiveness of this learning process. These chats are advanced systems that simulate human conversation through text or voice, using natural language processing and machine learning techniques. In education, intelligent chat systems can act as virtual tutors, helping students acquire knowledge and skills. They can be an alternative, by offering a more interactive, personalized, and engaging learning experience. However, it is important to emphasize that programming learning based on intelligent chats should not be seen as a replacement for teachers, but rather as a complementary tool. The main objective of this master dissertation is to propose an approach to teaching programming, using intelligent chats as a teaching support tool integrated into a Learning Management System. Therefore, this dissertation first presents a literature review, followed by the definition of an approach to programming learning, the development of a system and a case study with students of technical education in Information Technology. The case study used the proposed system and investigated the acceptance of the technology, as well as evaluating student results in tests. The results highlighted good acceptance of the technology, as well as associated benefits and challenges. It is expected to contribute to the advancement of knowledge in the field of computing education and open paths for future efforts, involving intelligent chats as educational support tools.*

**Keywords:** Programming Learning, Intelligent Chat Systems, Computers in Education, Artificial Intelligence in Education.

## Sumário

1. Introdução.....	11
1.1. Contexto e motivação.....	12
1.2. Problemática e justificativa.....	14
1.3. Objetivos, contribuições e métodos.....	17
1.4. Estrutura da dissertação.....	19
2. Referencial teórico e metodológico.....	21
2.1. Ensino de programação.....	21
2.2. Chats inteligentes.....	23
2.2.1 GPT 3.0, GPT 3.5 e ChatGPT.....	25
2.2.2 Outras soluções de chats inteligentes.....	27
2.3. Sistemas de gestão da aprendizagem.....	28
2.4. Design instrucional.....	30
3. Revisão bibliográfica e trabalhos relacionados.....	33
3.1. Método de revisão.....	33
3.2. Execução da revisão da literatura.....	35
3.2.1 Resultados quantitativos.....	35
3.2.2 Análise e discussão dos resultados.....	36
3.3. Trabalhos relacionados.....	42
4. Metodologia de pesquisa e análise comparativa.....	48

4.1.	Metodologia de pesquisa.....	48
4.2.	Comparação e seleção de chats inteligentes.....	52
4.3.	Comparação e seleção de abordagens pedagógicas.....	54
4.4.	Comparação e seleção de tecnologias para o protótipo.....	55
5.	Design, desenvolvimento e funcionamento do protótipo.....	57
5.1.	Design e desenvolvimento da ferramenta.....	57
5.2.	Funcionamento da ferramenta.....	60
6	Avaliação da abordagem e protótipo.....	66
6.1.	Local, participantes e metodologia de avaliação .....	66
6.2.	Resultados da avaliação.....	68
6.3.	Análise e discussão dos resultados.....	74
7	Conclusão da abordagem e protótipo.....	76
7.1.	Contribuições e considerações finais.....	76
7.2.	Limitações e trabalhos futuros.....	77
	Referências.....	79
	Apêndice I: Instruções para instalação e configuração do sistema.....	85
	Apêndice II: Instruções para elaboração de um curso no sistema.....	91
	Apêndice III: Avaliação da abordagem e do protótipo do sistema.....	96
	Apêndice IV: Questões da abordagem e do protótipo do sistema.....	99

## Glossário

- ADDIE - Análise, Design, Desenvolvimento e Implementação
- LMS - *Learning Management System* - Sistema de gestão de aprendizagem
- 4C/ID - Quatro Componentes Essenciais de Instrução
- MIT - Instituto de Tecnologia de Massachusetts
- PLN - Processamento de Linguagem Natural
- GPT - *Generative Pre-trained Transformer* - Transformador Generativo Pré-treinado
- BARD - *Bidirectional Encoder Representations from Transformers for Adversarial Retrosynthesis of Diverse* - Representações de Codificadores Bidirecionais de Transformadores para Retrossíntese Adversarial Diversificada
- PaLM - Modelo de Processamento de Linguagem
- NLP - Processamento de Linguagem Natural
- AWS - *Amazon Web Services* - Serviços Web da Amazon
- SAM - Modelo de Design instrucional iterativo
- IEEE - Instituto de Engenheiros Eletricistas e Eletrônicos
- ACM - *Association for Computing Machinery* - Associação para Máquinas Computacionais
- IA - Inteligência Artificial
- LSTM - *Long Short-Term Memory* - Memória de Longo Prazo e Curto Prazo

RNN - Rede Neural Recorrente

GRU - Unidades de Rede Neural Recorrente

3D - Três Dimensões

CMS - Sistema de Gerenciamento de Conteúdo

UML - Linguagem de Modelagem Unificada

TAM - Modelo de Aceitação da Tecnologia

AVA - Ambiente Virtual de Aprendizagem

## Lista de Tabelas

Tabela 1 - Critérios de busca.....	34
Tabela 2 - Critérios de inclusão.....	35
Tabela 3 - Artigos selecionados.....	36
Tabela 4 - Trabalhos relacionados.....	46
Tabela 5 - Análise comparativa de chats inteligentes.....	53
Tabela 6 - Análise comparativa de modelos de design instrucionais.....	54
Tabela 7 - Análise comparativa de tecnologias de LMS.....	56
Tabela 8 - Sequência de passos para utilização do sistema.....	58

## Lista de Figuras

Figura 1	Revisão exploratória sobre o tema.....	48
Figura 2	Revisão sistemática da literatura.....	49
Figura 3	Análise comparativa dos chats inteligentes.....	50
Figura 4	Desenvolvimento do protótipo.....	51
Figura 5	Avaliação com alunos do ensino técnico.....	52
Figura 6	Diagrama de caso de uso para o sistema.....	57
Figura 7	Diagrama de componentes do sistema.....	59
Figura 8	Página principal do sistema.....	62
Figura 9	Página de curso no sistema.....	63
Figura 10	Página com informações sobre um curso.....	63
Figura 11	Página com informações detalhadas sobre um curso.....	64
Figura 12	Visão geral do dashboard.....	64
Figura 13	Exemplo do uso do ChatGPT na realização de exercícios.....	65
Figura 14	Grupos GC e GE realizando a avaliação.....	69
Figura 15	Nível de conhecimento prévio em Lógica da Programação.....	70
Figura 16	Respostas do grupo de controle às questões 2 a 4.....	71
Figura 17	Respostas do grupo experimental às questões 2 a 5.....	72
Figura 18	Boxplots com pontuação do GC e GE no pré-teste e pós-teste.....	73
Figura 19	Respostas dos grupos GC e GE	74

## 1. Introdução

Aprender programação de computadores pode ser uma tarefa desafiadora para muitas pessoas de diferentes idades, e até mesmo para aqueles com bons níveis de conhecimento em áreas relacionadas, tais como matemática e lógica. Aprender programação envolve o domínio de conceitos teóricos, como lógica de programação e algoritmos, bem como habilidades práticas de escrita de código (Carreira et al., 2022).

Aprender programação pode apresentar diferentes desafios, especialmente para aqueles que estão iniciando na área, tais como: *lógica*, pois os estudantes precisam desenvolver habilidades de raciocínio lógico; *sintaxe*, pois a linguagem pode ser complexa e a necessidade de aprender uma nova gramática; *complexidade*, pois à medida que os estudantes avançam são apresentados conceitos cada vez mais complexos (e.g., estruturas de dados avançada); *falta de prática*, pois a programação é uma habilidade prática que requer prática consistente; *pensamento abstrato por envolver* muita abstração e raciocínio dedutivo; e, *falta de recursos*, como ambiente de aprendizagem limitado e ausência de orientação e suporte adequado, pois nem sempre estes estão disponíveis (Lin et al., 2019).

Existem diversas metodologias que podem ser utilizadas para aprender programação de computadores (e.g., a aprendizagem baseada em projetos e problemas), e a escolha da mais adequada dependerá do perfil e da necessidade do estudante. Por exemplo, o aluno pode ser desafiado a desenvolver um aplicativo móvel para resolver um problema do mundo real. Essa abordagem não só oferece uma experiência prática, mas também permite que o aluno aplique os conceitos de programação em contextos reais, tornando o aprendizado mais significativo. De acordo com Farias (2022), a aprendizagem significativa segundo Ausubel (1982) desempenha um papel essencial no desenvolvimento de competências, como a competência em informação, ao relacionar novos conhecimentos com os já existentes. Algumas diretrizes, como praticar bastante e utilizar diferentes formas de aprendizagem, podem deixar o aluno mais confortável com os erros e fundamentos na programação de computadores. É importante ressaltar que essas metodologias e diretrizes podem ser combinadas e adaptadas conforme a necessidade de cada estudante, visando proporcionar uma abordagem de aprendizado mais personalizada e eficaz (Banić et al., 2023).

Assim, as novas tecnologias de chats inteligentes podem apoiar de maneira significativa a aprendizagem de programação. São tecnologias emergentes que podem ser usadas para criar uma abordagem mais interativa e personalizada para o ensino de programação (Henno et al., 2023).

O principal objetivo desta dissertação é propor uma abordagem de ensino de programação, utilizando chats inteligentes como ferramenta de apoio ao ensino integrada a um Sistema de Gestão da Aprendizagem (LMS - Learning Management System). A aceitação da tecnologia e o desempenho dos alunos com esta abordagem são avaliados em uma turma do ensino técnico em Informática. No entanto, é importante ressaltar que, embora o foco inicial seja no ensino técnico, as descobertas e metodologias desenvolvidas neste estudo podem ser aplicadas e adaptadas a outros cenários educacionais.

A Seção 1.1 detalha o contexto e motivação para esta proposta de dissertação, incluindo as dificuldades do ensino de programação e as motivações para o estudo da abordagem de aprendizagem de programação baseada em chats inteligentes; a Seção 1.2 apresenta a problemática envolvida no desenvolvimento da abordagem proposta, bem como a justificativa para propô-la; a Seção 1.3 detalha a questão de pesquisa, objetivos específicos e as contribuições esperadas desta pesquisa; já a Seção 1.4 apresenta o método de pesquisa empregado e a estrutura dessa proposta de dissertação.

### **1.1. Contexto e Motivação**

A disciplina de programação é um desafio recorrente em diversos cursos que incluem essa disciplina, levando a altos índices de evasão. Muitos estudantes enfrentam dificuldades ao tentar compreender os conceitos fundamentais da programação, como lógica de programação, algoritmos e estrutura de dados. Além disso, os termos técnicos utilizados na linguagem de programação geralmente estão no idioma inglês, o que pode representar uma barreira para aqueles que têm dificuldades com o idioma (Carreira et al., 2022).

Existem vários motivos que podem levar à evasão das disciplinas de programação, como complexidade da matéria, baixo rendimento, desinteresse, motivação, ambiente de aprendizagem limitado, entre outros fatores (Gonda & Chu,

2019). Além disso, a falta de apoio adequado por parte dos professores ou colegas pode ser um fator importante na evasão da disciplina de programação. O aluno pode se sentir desmotivado ou desencorajado se não receber o apoio necessário para superar suas dificuldades, se sentir isolado ou desestimulado no ambiente de aprendizagem.

Para enfrentar esses desafios, é necessário buscar abordagens e recursos pedagógicos inovadores que possam motivar e engajar os estudantes, tornando a aprendizagem de programação mais efetiva e acessível. Nesse sentido, a utilização de chats inteligentes pode ser uma solução promissora, por permitir que os alunos tenham acesso a um ambiente de aprendizagem mais interativo e personalizado. Os chats inteligentes podem auxiliá-los na compreensão dos conceitos e na resolução de problemas de programação a qualquer momento, sem limitações de tempo ou disponibilidade (Ji & Yuan, 2022).

Nesse contexto, a nova geração de chats inteligentes, tais como o ChatGPT, podem proporcionar uma experiência de aprendizagem mais interativa e personalizada para os alunos (Luo & Gonda, 2019; Silva Junior et al., 2023). O uso de chats inteligentes no ensino de programação é motivado por diversos aspectos e seu potencial na melhoria do aprendizado, tais como:

- Acesso rápido a informação: o chat oferece um meio rápido e conveniente de obter informações e respostas para perguntas específicas relacionadas à programação.
- Suporte personalizado: o chat permite interações individuais e personalizadas, o que significa que o aluno pode obter suporte adaptado às suas necessidades específicas.
- Aprendizado interativo: o chat pode fornecer uma experiência de aprendizado interativa, onde você pode fazer perguntas, receber respostas imediatas e até mesmo ter discussões sobre tópicos de programação.
- Feedback e correções: ao compartilhar seu código com um chat, o aluno pode obter feedback e sugestões para melhorar sua solução. Os chats podem ajudar a identificar erros, fornecer alternativas mais eficientes e oferecer orientações para aprimorar sua lógica de programação.

- Compartilhamento de recursos: chats podem ser um ótimo meio para compartilhar recursos de aprendizado, como tutoriais, documentações e artigos relevantes.
- Ajuda para resolução de problemas: chats podem auxiliar os alunos a identificar e resolver erros de sintaxe e semântica em seus códigos.

Assim, espera-se que o uso de chats inteligentes ajude a tornar o processo de aprendizado de programação mais personalizado, acessível e envolvente, o que pode aumentar a motivação dos alunos para aprender e melhorar sua compreensão da programação de computadores.

## **1.2. Problemática e Justificativa**

Embora a utilização de chats inteligentes para a aprendizagem de programação possa apresentar diversas vantagens potenciais, há também desafios e dificuldades a serem enfrentados na construção dessa abordagem. Tais dificuldades podem ser técnicas, ou seja, relacionadas à construção dos chats inteligentes, ou do seu uso correto por alunos e professores em atividades de aprendizagem (Luo & Gonda, 2019).

Alguns exemplos de problemas/dificuldades técnicas no contexto de ensino de programação incluem: (1) compreender a linguagem natural dos alunos, especialmente se estiverem usando gírias ou termos técnicos específicos; (2) identificar erros de programação, que podem ser muito sutis e difíceis de detectar; (3) fornecer feedbacks precisos e úteis, que auxiliem os alunos a corrigir seus erros e melhorar suas habilidades de programação; e (4) ambiguidade e falta de contexto, pois muitas vezes, as perguntas feitas pelos alunos podem ser ambíguas ou carecer de contexto, tornando difícil para o chatbot de programação fornecer resposta precisa e adequada (Sophia & Jacob, 2021).

Alguns exemplos de problemas/dificuldades relacionadas ao uso e à didática no contexto de ensino de programação incluem:

- Como ensinar os alunos a tirarem maior proveito dos chats inteligentes de modo a obter boas respostas às suas dúvidas com maior precisão e menor esforço.
- Como utilizar o chat de modo a não substituir atividades práticas que deveriam ser realizadas pelos alunos, mas sim utilizá-los como apoio a estas.

- Como formar alunos para serem profissionais que façam uso correto de chats inteligentes posteriormente em suas atividades profissionais que envolvam programação.
- Como incluir e articular o uso de chats inteligentes com outros recursos de aprendizagem.

Essas são algumas das dificuldades que podem ser encontradas na construção de uma abordagem baseada em chats inteligentes para a aprendizagem de programação. No entanto, se esses desafios forem superados e se o chat for projetado adequadamente, espera-se que ele seja uma ferramenta útil para tornar o processo de aprendizagem de programação mais eficiente e eficaz (Mallikarjuna et al., 2021).

Do ponto de vista da educação, a elaboração de uma abordagem baseada em chats inteligentes para a aprendizagem de programação requer considerar critérios pedagógicos adequados. É importante que o chat e seu uso sejam planejados para motivar e engajar os alunos, além de fornecer um ambiente de aprendizagem seguro e colaborativo. Em resumo, a construção de uma abordagem baseada em chats inteligentes para a aprendizagem de programação pode ser um desafio tanto do ponto de vista da computação quanto da educação.

É importante lembrar que mesmo os chats inteligentes com diversas limitações podem ser ferramentas úteis. Assim, como em qualquer abordagem de ensino, é importante avaliar cuidadosamente os benefícios e limitações dos recursos de aprendizagem, tais como chats inteligentes, e adaptar sua utilização às necessidades específicas dos alunos. Isso passa por um entendimento do ponto de vista computacional da construção e funcionamento dos chats inteligentes.

Embora o uso de chats inteligentes na aprendizagem seja uma área de pesquisa em crescimento, ainda há lacunas na literatura sobre o assunto. Nos parágrafos seguintes, são destacados exemplos de algumas das áreas que ainda carecem de pesquisas mais aprofundadas (Sophia & Jacob, 2021).

A primeira área é a avaliação da eficácia da abordagem, na prática, uma vez que embora haja muitos estudos que mostram que a abordagem de aprendizagem baseada em chats inteligentes é promissora, ainda há poucos estudos que efetivamente avaliam sua eficácia em comparação com outras abordagens de ensino de programação.

A adaptação dos chats inteligentes é outra área que precisa ser explorada. Como os alunos têm necessidades diferentes em diferentes estágios de aprendizagem, o chat inteligente precisa ser capaz de adaptar sua abordagem para atender às necessidades dos alunos em diversos momentos.

Além disso, embora muitos estudos mostrem que a abordagem de aprendizagem baseada em chats inteligentes pode ser mais envolvente para os alunos, ainda há poucos estudos que avaliem o impacto dessa abordagem no desempenho dos alunos no aprendizado de programação (Ji & Yuan, 2022).

Por fim, destaca-se que estas lacunas estão presentes no ensino de programação, bem como a necessidade de criação de itinerários, métodos e ferramentas que explorem chats inteligentes em atividades em conjunto com outros recursos de aprendizagem.

Embora existam limitações e necessidade de estudos, como os supracitados, o uso de chats inteligentes na aprendizagem de programação pode ajudar a superar algumas das barreiras enfrentadas pelos alunos (Henno et al., 2023; Carreira et al., 2022). Por exemplo, muitos alunos acham a programação intimidante e difícil de entender. A abordagem baseada em chats inteligentes pode ajudar a tornar a programação mais acessível, permitindo que os alunos façam perguntas em linguagem natural e recebam respostas imediatas e claras.

Além disso, a abordagem de aprendizagem com auxílio de chats inteligentes pode ser personalizada para atender às necessidades individuais de cada aluno. Isso significa que os alunos podem aprender no seu próprio ritmo, receber feedback instantâneo e ter acesso a exemplos de código relevantes para suas necessidades específicas.

Algumas das maneiras pelas quais os chats inteligentes podem ser usados para melhorar a aprendizagem de programação incluem (Wang et al., 2023): *dar repostas rápidas*, ou seja, quando o aluno está enfrentando um problema específico ou tem uma dúvida sobre um conceito de programação, um chat inteligente pode fornecer respostas rápidas e precisas; *criar exemplos de códigos*, uma vez que chats inteligentes podem fornecer exemplos de código prontos para ilustrar conceitos e técnicas de programação; *explicar conceitos complexos*, visto que alguns conceitos de programação podem ser complexos de entender inicialmente; e *dar sugestões de recursos de aprendizagem* tais

como recomendar livros, tutoriais, cursos online e outros recursos de aprendizagem relevantes para auxiliar no seu progresso na programação.

Embora existam desafios computacionais para que esses benefícios sejam atingidos (*e.g.*, compreensão de linguagem natural, feedback preciso, melhoria no design de interface, melhoria na identificação de erros de programação) (Lin et al., 2022), o avanço contínuo em tecnologia de inteligência artificial e aprendizado de máquina já tornou factível seu uso na aprendizagem de programação de uma maneira altamente eficaz e envolvente (Mallikarjuna et al., 2021). Sendo assim, chats inteligentes são recursos de aprendizado com grande potencial, mas que ainda carecem de abordagens e métodos eficazes para potencializar seus benefícios em sala de aula.

### **1.3. Objetivos e Contribuições**

A seguinte questão de pesquisa principal norteia este trabalho: *Como os sistemas de chats inteligentes podem ser utilizados no ensino de programação e qual a sua aceitação e influência no desempenho dos estudantes de programação?* Para responder a essa questão de pesquisa, este trabalho responderá às seguintes questões específicas:

- *Como integrar chats inteligentes a LMS e métodos instrucionais para fomentar o aprendizado de programação?*
- *Qual a aceitação desta tecnologia e sua influência no desempenho dos estudantes?*
- *Quais são os desafios e limitações associados ao uso de chats inteligentes na programação de computadores?*

Como base nas questões supracitadas, os seguintes objetivos gerais específicos foram definidos:

- *Objetivo geral:* propor e avaliar uma abordagem de aprendizagem de programação baseada em chats inteligentes para apoiar docentes e discentes no ensino técnico a superar as barreiras enfrentadas ao aprender programação.

A partir do objetivo geral, estabelecemos o seguinte objetivo específico:

- Medir a aceitação da tecnologia e desempenho dos estudantes em avaliações sobre conceitos de programação após interagirem com o chat inteligente, em comparação com o método tradicional de ensino, que normalmente envolve aulas presenciais, pesquisa online, material didático sem o uso de tecnologia interativa.

A hipótese deste trabalho é que: “ao utilizar um chat inteligente integrado a um LMS para apoiar o ensino de programação, com base em um método de design instrucional, os alunos terão uma experiência de aprendizagem mais interativa e personalizada às suas necessidades individuais, levando a uma boa aceitação da tecnologia e elevando seu desempenho.”

Do ponto de vista metodológico, será realizado o estudo das tecnologias sobre os sistemas de chats inteligentes atuais, os problemas no ensino de programação e, na sequência, identificar possíveis problemas que possam ser abordados, possibilidades de aprimoramento e demanda por profissionais que façam uso dessa tecnologia para tornarem-se mais produtivos.

Uma vez identificados tais problemas e possibilidades, será proposta uma abordagem composta por diretrizes, métodos e ferramentas computacionais para apoio ao aluno e ao professor. Ao elaborar esta abordagem, serão considerados tanto aspectos tecnológicos como modelos e métodos de design instrucional, tais como o ADDIE (análise, design, desenvolvimento e implementação) (Martenstyaró et al., 2015). Sobre as ferramentas, será construído um LMS (Kita et al., 2020) integrando o chat inteligente às atividades das disciplinas, segundo a abordagem proposta.

Na sequência, será conduzida avaliação controlada com grupos de estudantes que utilizaram a abordagem com chat inteligente e grupos de controle que utilizam métodos tradicionais de ensino. Pesquisas, questionários e observações podem ser usados para coletar dados sobre a aceitação da tecnologia e desempenho dos estudantes. O foco principal desta avaliação serão alunos de cursos técnicos voltados à área de computação e correlatas.

Entre as principais contribuições esperadas para esta pesquisa se destacam: melhor compreensão dos benefícios da integração de chat inteligente no ensino de programação, fornecimento de diretrizes práticas aos professores em utilizar chat

inteligente na aprendizagem com foco principal em programação, abordagem de ensino de programação que faz uso dessa tecnologia e ferramentas de apoio a esta abordagem.

Do ponto de vista prático, o objetivo é investigar a aceitação da tecnologia e melhorar o desempenho no processo de aprendizagem. Planeja-se alcançar isso, utilizando uma ferramenta interativa e conversacional, como os chats inteligentes, para tornar o processo de aprendizagem mais agradável, diminuindo a percepção de complexidade e dificuldade associada à programação.

Outra contribuição indireta esperada é a melhoria na qualidade dos desenvolvedores formados. Isso ocorre porque a abordagem proposta incentiva a interação e o diálogo constante entre aluno e ferramenta, permitindo a resolução de dúvidas e o esclarecimento de conceitos e promovendo o desenvolvimento da autonomia na procura de soluções para os problemas encontrados ao longo do processo de aprendizagem.

Além disso, a utilização de chats inteligentes pode contribuir para o entendimento sobre o uso de modelos de linguagem em aplicações educacionais e fornecer insights sobre como outras ferramentas de inteligência artificial podem ser utilizadas para melhorar o aprendizado de programação. Ainda assim, é importante ressaltar que o chat inteligente, neste trabalho, é visto como uma ferramenta complementar ao ensino tradicional de programação, e não como uma substituição total (Wang et al., 2023).

#### **1.4. Estrutura da Dissertação**

Esta dissertação está estruturada da seguinte maneira:

- O Capítulo 2 apresenta o referencial teórico e metodológico, incluindo o ensino de programação, chats inteligentes, LMS e design instrucional.
- O Capítulo 3 detalha uma revisão bibliográfica sobre o tema, bem como apresenta os trabalhos relacionados e destaca o diferencial desta dissertação perante os demais trabalhos.
- O Capítulo 4 descreve a metodologia de pesquisa utilizada e uma análise comparativa dos sistemas de chats inteligentes, LMS e métodos de design instrucional.

- O Capítulo 5 detalha o design, desenvolvimento e funcionamento de protótipo de sistema construído com base na abordagem proposta.
- O Capítulo 6 apresenta a avaliação da abordagem e ferramenta com alunos de curso técnico em Informática.
- O Capítulo 7 apresenta as contribuições, limitações, considerações finais e trabalhos futuros.
- O Apêndice I detalha as instruções para instalação e configuração do sistema proposto.
- O Apêndice II apresenta instruções para elaboração de um curso com a abordagem e sistema proposto.
- O Apêndice III inclui os formulários de avaliação da aceitação da abordagem e do protótipo do sistema.
- O Apêndice IV inclui os questionários de pré-teste e pós-teste.

## **2. Referencial Teórico e Metodológico**

Este capítulo apresenta o referencial teórico e metodológico utilizado neste trabalho. Ele é resultado de pesquisa exploratória na literatura e inclui os principais conceitos, fundamentos e métodos empregados nesta dissertação. A Seção 2.1, sobre Ensino de Programação, apresenta a importância da disciplina de programação abordando seus benefícios, desafios enfrentados pelos alunos, aspectos relacionados à evolução tecnológica e ensino, bem como enfatizar a pluralidade de abordagens no aprendizado de programação e seus benefícios para o desenvolvimento de habilidades críticas e oportunidades profissionais. A Seção 2.2, sobre chats inteligentes, apresenta a evolução histórica dos chats inteligentes, destacando suas aplicações atuais em várias áreas, discute os avanços tecnológicos, apresenta soluções populares e foca especificamente no ChatGPT, do qual aborda suas características e potenciais benefícios, enquanto reconhece desafios e limitações associados a essa tecnologia. A Seção 2.3, sobre sistemas de Gestão de Aprendizagem, explica o que é um LMS, destacando suas funcionalidades principais, exemplos e, em seguida, explora a capacidade do WordPress de ser transformado em um LMS, mencionando algumas extensões para essa finalidade. A Seção 2.4, sobre design instrucional, apresenta vários modelos e abordagens de design instrucional, como ADDIE, 4C/ID, Gagne's Nine, explorando as características de cada um.

### **2.1. Ensino de Programação**

A disciplina de programação é atualmente essencial não só para cursos de programação e tecnologia, mas também para formação em diversas áreas de atuação, ao preparar o aluno para um mundo cada vez mais digital. Entre os benefícios, destaca-se que a programação promove a capacidade de analisar problemas, dividindo-os em partes menores e criando soluções lógicas (Carreira et al., 2022). Além disso, os alunos aprendem a criar soluções únicas para diversos problemas, incentivando a criatividade e a inovação, bem como proporcionam habilidades práticas valiosas em diversas áreas, tais como tecnologia, engenharia e negócios (Banić et al., 2023).

Já quanto aos desafios de aprendizagem, a complexidade da programação pode parecer intimidante para iniciantes. Sendo assim, manter o aluno motivado e confiante é

um desafio. Aprender programação pode ser um processo desafiador e demorado. Alguns alunos podem perder a motivação ou a persistência diante de dificuldades e erros frequentes. É fundamental incentivar a persistência, dar feedback construtivo e mostrar aos alunos como seus esforços levam a resultados tangíveis (Henno et al., 2023).

Esta complexidade está muitas vezes ligada ao desenvolvimento do pensamento lógico e à resolução de problemas. Alguns alunos podem enfrentar dificuldades para desenvolver essas habilidades ou podem ter dificuldade em aplicá-las, na prática. Assim, é importante desenvolver atividades que estimulem o pensamento crítico e a resolução de problemas (Banić et al., 2023).

A complexidade também está relacionada ao desenvolvimento da capacidade de abstração de conceitos. A programação envolve conceitos abstratos, como algoritmos, estrutura de dados e lógica de programação. Especialmente no início, alguns alunos podem ter dificuldade em compreender esses conceitos. Assim, é necessário encontrar maneiras de explicar esses conceitos claramente.

Outro aspecto a ser considerado é o estilo de aprendizagem, uma vez que os alunos têm diferentes estilos e é crucial adaptar abordagens e recursos para atender às necessidades individuais. Alguns alunos podem aprender melhor por meio de exemplos práticos, enquanto outros podem preferir explicações teóricas. Isso leva a refletir que não existe um único método, sendo importante adotar uma abordagem flexível (Wang et al., 2023).

Além disso, a evolução tecnológica das linguagens de programação e as tecnologias é desafiante. Isso leva à necessidade de atualizar frequentemente os currículos para refletir essas mudanças. O ensino de programação requer acesso a recursos adequados, como computadores novos, ambientes atualizados e acesso rápido à internet. Nem todos os alunos têm acesso a esses recursos em casa ou na instituição de ensino, o que pode limitar sua capacidade de praticar e se envolver plenamente com o aprendizado, principalmente em disciplinas avançadas que demandam por mais recursos (Henno et al., 2023).

Do ponto de vista tecnológico, a demanda por aprender diferentes linguagens no tempo tem se tornado um desafio. Não são raras as discussões, que permanecem por

décadas, sobre qual deve ser a primeira linguagem ou a principal linguagem no ensino de programação.

Por fim, é importante ressaltar que o ensino de programação, conforme supracitado, envolve uma pluralidade de abordagens e métodos para transmitir conhecimento sobre linguagem de programação. Apesar dos desafios, os benefícios de aprender programação são inegáveis, pois desenvolvem habilidades críticas e abrem portas para oportunidades em diversas áreas (Ji & Yuan, 2022).

## **2.2. Chats Inteligentes**

A história dos chats inteligentes remonta ao final dos anos 1960, quando o MIT (Instituto de Tecnologia de Massachusetts) desenvolveu o ELIZA, um programa que imitava um psicoterapeuta. O ELIZA usava uma técnica de Processamento de Linguagem Natural (PLN) para simular uma conversa humana e responder a perguntas inteligentemente. O programa se tornou popular rapidamente, sendo reconhecido como uma das primeiras aplicações bem-sucedidas de Inteligência Artificial (Santhana et al., 2020).

Durante as décadas seguintes, os chats inteligentes continuaram a evoluir, sendo desenvolvidos programas que atendiam a uma variedade de funções, desde assistentes virtuais em sites de comércio eletrônico até chatbots em jogos de computador. Nos últimos anos, os avanços na tecnologia e na inteligência artificial permitiram que os chats inteligentes se tornassem mais sofisticados e personalizados.

Atualmente, os chats inteligentes são amplamente utilizados em diversas áreas, incluindo suporte ao cliente, assistência médica, comércio eletrônico e educação. A capacidade dos chats inteligentes de fornecer respostas rápidas e precisas a perguntas comuns e rotineiras, bem como de aprender com a interação com usuários, tornou-os uma ferramenta valiosa em muitos setores.

Chats inteligentes têm se beneficiado muito dos avanços na área de inteligência artificial, em particular no campo do PLN (Ganesan et al., 2020). Algumas das principais inovações em PLN que têm permitido melhorias significativas em chats inteligentes são os modelos de linguagem pré-treinados e o aprendizado por reforço (Ashfaq et al., 2020).

Os modelos de linguagem pré-treinados são inicialmente treinados em grandes quantidades de dados textuais e podem ser usados para tarefas de PLN, como classificação de sentimentos, sumarização de texto e geração de texto. Esses modelos, como o GPT-3 (*Generative Pre-trained Transformer*) da OpenAI<sup>1</sup>, podem ser usados para melhorar a qualidade das respostas de um chatbot. Já o aprendizado por reforço permite que um chatbot aprenda a partir da interação com os usuários, visando maximizar uma recompensa. Por exemplo, um chatbot que ajuda as pessoas a agendar compromissos pode aprender a partir da interação com usuários para agendar mais compromissos com sucesso (Finnie-Ansley et al., 2022).

Existem diversas tecnologias disponíveis no mercado que permitem o desenvolvimento de chats inteligentes. Algumas das principais alternativas incluem: Dialogflow<sup>2</sup>: uma plataforma de desenvolvimento de chatbots do Google; IBM Watson Assistant<sup>3</sup>: uma plataforma de desenvolvimento de chatbots da IBM; Amazon Lex<sup>4</sup>: um serviço de conversação automatizada da Amazon que permite criar chatbots; e o Microsoft Bot Framework<sup>5</sup>: uma plataforma de desenvolvimento de chatbots da Microsoft. Essas são apenas algumas das alternativas de chats inteligentes disponíveis atualmente. A escolha da melhor solução depende das necessidades específicas do projeto, recursos disponíveis e habilidades técnicas da equipe responsável pelo desenvolvimento do chatbot (Santhana et al., 2020).

Essas tecnologias têm um grande potencial de impacto em diversas áreas, desde assistentes virtuais, chatbots, análises de dados e assistência em processos de tomada de decisão. No entanto, é importante notar que elas ainda apresentam limitações e desafios, como a possibilidade de gerar respostas enviesadas ou inadequadas em algumas situações, além de questões de privacidade e segurança dos dados. As próximas subseções detalham alguns sistemas para o desenvolvimento de chats inteligentes, que farão parte desta proposta de dissertação.

---

<sup>1</sup> <https://openai.com/product>

<sup>2</sup> <https://cloud.google.com/dialogflow>

<sup>3</sup> <https://www.ibm.com/br-pt/products/watson-assistant>

<sup>4</sup> <https://aws.amazon.com/pt/pm/lex>

<sup>5</sup> <https://azure.microsoft.com/en-us/products/bot-services>

### 2.2.1 GPT 3.0, GPT 3.5 e ChatGPT

O ChatGPT<sup>6</sup> é uma implementação específica do GPT desenvolvida pela OpenAI para fins de conversação. Os GPT-3 e GPT-3.5 são modelos de linguagem de grande escala treinados com bilhões de parâmetros para gerar texto de alta qualidade em uma ampla variedade de tarefas, incluindo geração de texto, tradução, respostas a perguntas, etc. O ChatGPT, por sua vez, é um chatbot baseado no GPT-3.5 e GPT-4.0 que utilizam as capacidades de linguagem natural do modelo para interagir com os usuários de forma fluida e coerente, como se fosse uma conversa humana. Ele pode responder a perguntas, dar sugestões, fornecer informações e realizar outras tarefas similares (OpenAI, 2023).

GPT-3.0 é um modelo de linguagem baseado em redes neurais de grande escala, especificamente o Transformer, sendo um tipo de arquitetura de rede neural projetada para processar sequências de texto. O GPT-3 é um dos maiores modelos de linguagem já criados, possuindo 175 bilhões de parâmetros. Ele foi treinado em uma quantidade massiva de dados da Internet para aprender a prever palavras e entender a estrutura da linguagem natural. O GPT-3.0 possui uma ampla gama de aplicações, como geração automática de texto, tradução automática, respostas a perguntas e assistentes de chat. Ele consegue gerar respostas coerentes e fluentes, e pode até mesmo realizar tarefas complexas, como escrever código de programação simples.

O GPT-3.5 é uma versão aprimorada do GPT-3.0, com algumas melhorias e refinamentos adicionais. Embora a OpenAI não tenha divulgado especificamente as alterações entre o GPT-3.0 e o GPT-3.5, é provável terem sido efetuadas atualizações na arquitetura do modelo, nos algoritmos de treinamento e na qualidade geral da geração de texto (Wang et al., 2023).

A versão mais recente no momento da escrita desta proposta é o GPT-4.0, que trabalha com processamento de imagens, entrada de textos e produção de respostas. As principais limitações das versões anteriores, como “alucinações” (respostas incorretas ou inventadas), uma janela de contexto limitada e não aprender (em tempo real) pela experiência, continuam presentes do GPT-4.0. Entretanto, esta versão reduz significativamente as “alucinações” e obteve por volta de 19 por cento a mais em

---

<sup>6</sup> <https://openai.com/blog/chatgpt>

avaliações internas quando comparada ao GPT-3.5. No teste HumanEval para tarefas de programação em Python (Chen et al., 2021), o GPT-4 obteve uma pontuação de 67,0%, enquanto o GPT-3.5 alcançou 48,1% (OpenAI, 2023; Ahmed et al., 2023).

O ChatGPT é uma implementação específica do modelo GPT-3.5 (e mais recentemente GPT-4.0) que foi desenvolvida para interações de conversação em formato de chat. Ele foi treinado com uma abordagem de aprendizado por reforço de modelagem por imitação, em que especialistas humanos forneceram exemplos de conversas para guiar o treinamento do modelo. O objetivo era tornar o ChatGPT mais adequado para diálogos e fornecer respostas mais relevantes e úteis para perguntas e solicitações dos usuários.

O ChatGPT é projetado para responder a perguntas, fornecer informações, realizar tarefas simples e até mesmo manter uma conversa casual. Embora seja um chatbot sofisticado, ele também possui limitações e pode gerar respostas vagas ou incorretas em certas situações.

O ChatGPT é um modelo de linguagem natural treinado em uma grande quantidade de dados textuais, capaz de gerar respostas coerentes e relevantes para perguntas e declarações em diferentes áreas de conhecimento. Uma das principais características do ChatGPT é a sua capacidade de inferir informações implícitas nas perguntas e respostas, permitindo uma comunicação mais natural e fluída.

Além disso, o ChatGPT consegue realizar tarefas de linguagem natural como sumarização de textos, tradução de idiomas e geração de texto baseado em um prompt. Ele pode ser utilizado em diferentes aplicações, como assistentes virtuais, sistemas de atendimento ao cliente e chatbots. (Koundinya et al., 2020).

Outra característica importante do ChatGPT é a sua capacidade de aprendizado contínuo. Como ele é treinado em uma grande quantidade de dados textuais, ele pode ser adaptado para diferentes domínios de conhecimento, permitindo a personalização de suas respostas e aprimoramento de sua capacidade de entender e responder aos usuários.

## 2.2.2 Outras soluções de Chats Inteligentes

Existem diversas soluções de Chats Inteligentes disponíveis no mercado, cada uma com suas características e funcionalidades específicas.

O Google BARD<sup>7</sup> é uma ferramenta experimental do Google que usa inteligência artificial para gerar poesia e outras formas de escrita criativa. O nome BARD significa “*Bidirectional Encoder Representations from Transformers for Adversarial Retrosynthesis of Diverse*” sendo baseado, atualmente, na arquitetura de rede neural PaLM 2<sup>8</sup>. Este o novo sistema de IA generativa do Google (Ahmed et al., 2023).

O Google BARD possui capacidades e limitações distintas do ChatGPT (Ahmed et al., 2023). Embora o Google BARD possa superar o ChatGPT em algumas situações, não há estatísticas consistentes da comparação da sua performance atual no que diz respeito à programação de computadores. Estudos anteriores apontam uma performance da arquitetura PaLM de 26.2% no teste HumanEval (Chen et al., 2021), muito inferior ao 67,0% obtido pelo GPT 4.0 (OpenAI, 2023). Entretanto, o principal fato que levou à decisão de não considerar Google BARD como solução para esta proposta de dissertação, é que ele não estava disponível para contas do google no Brasil no momento da elaboração da proposta (Rahaman et al., 2023).

Outras alternativas populares de chats inteligentes incluem:

- **Dialogflow**<sup>9</sup>: é uma plataforma de desenvolvimento de chatbots do Google que usa o processamento de linguagem natural para criar conversas personalizadas e engajadoras com os usuários. A plataforma usa tecnologias de processamento de linguagem natural (NLP) e machine learning para entender a intenção do usuário e responder de maneira inteligente e possui integrações com diversas plataformas, como Facebook Messenger, Slack, Telegram e Skype.

---

<sup>7</sup> <https://bard.google.com/>

<sup>8</sup> <https://ai.google/discover/palm2/>

<sup>9</sup> <https://dialogflow.cloud.google.com/>

- **IBM Watson Assistant<sup>10</sup>**: é uma plataforma de chatbot da IBM que usa inteligência artificial e machine learning para permitir que as empresas criem e personalizem chatbots para suas necessidades específicas. A plataforma é altamente flexível e pode ser usada em uma ampla variedade de setores, incluindo saúde, finanças, varejo e muito mais.
- **Amazon Lex<sup>11</sup>**: é uma solução de chatbot da Amazon Web Services (AWS) que usa tecnologia de reconhecimento de voz e processamento de linguagem natural para permitir a criação de chatbots interativos.
- **Microsoft Bot Framework<sup>12</sup>**: é uma plataforma de desenvolvimento de chatbots que permite criar bots altamente interativos e personalizados para uma ampla gama de canais, como Facebook Messenger, Skype, Slack, entre outros.

### 2.3. Sistemas de Gestão da Aprendizagem

Um LMS é um sistema utilizado para administrar, distribuir e gerenciar atividades de aprendizado online. Ele fornece uma plataforma centralizada para criar, entregar e avaliar cursos e conteúdos educacionais. Os LMSs são amplamente utilizados em ambientes educacionais, corporativos e em outras organizações que buscam fornecer treinamento e desenvolvimento online eficientemente (Wijastuti et al., 2020).

Entre os principais recursos de um LMS estão:

- **Gestão de conteúdo:** permite o desenvolvimento, armazenamento e organização de materiais educacionais, como aulas, apresentações, documentos, vídeos e questionários.
- **Distribuição de cursos:** oferece um meio de disponibilizar cursos e conteúdo para os alunos ou participantes, muitas vezes com opções para agendar ou controlar o acesso.

---

<sup>10</sup> <https://www.ibm.com/cloud/watson-assistant/>

<sup>11</sup> <https://aws.amazon.com/lex/>

<sup>12</sup> <https://dev.botframework.com/>

- **Acompanhamento e avaliação:** permite aos instrutores rastrear o progresso dos alunos, avaliar seu desempenho por meio de testes e questionários e fornecer feedback.
- **Comunicação:** facilita a comunicação entre instrutores e alunos por meio de mensagens, fóruns de discussão e chat.
- **Gestão de usuários:** permite a criação e gestão de perfis de usuários, como alunos, instrutores e administradores, com diferentes níveis de acesso e permissões.

Existem diversos LMSs que são largamente utilizados por instituições de ensino e em diversos tipos de cursos, tais como:

- **Moodle**<sup>13</sup>: amplamente utilizado em instituições educacionais e corporativas. Oferece uma variedade de recursos para criar cursos, interagir com alunos e avaliar o progresso.
- **Canvas**<sup>14</sup>: utilizado em instituições educacionais, oferecendo recursos como a criação de cursos personalizados, integração de multimídia e colaboração em grupo.
- **Blackboard**<sup>15</sup>: oferece ferramentas para criar e entregar cursos online, incluindo recursos de comunicação, avaliação e colaboração.
- **Google classroom**<sup>16</sup>: embora seja mais simples em comparação com outros LMSs, é amplamente utilizado para gerenciar atividades educacionais.

Esses são apenas alguns exemplos e a escolha de um LMS depende das necessidades específicas da instituição, das características desejadas e da facilidade de uso para instrutores e alunos (Trilaksono & Santoso, 2018).

O WordPress é uma plataforma de gerenciamento de conteúdo extremamente popular que permite criar e gerenciar sites de forma flexível e intuitiva. Ele é amplamente utilizado, desde blogs pessoais até sites de comércio eletrônico e portfólios

---

<sup>13</sup> <https://moodle.org/>

<sup>14</sup> <https://www.instructure.com/canvas>

<sup>15</sup> <https://www.blackboard.com/>

<sup>16</sup> <https://classroom.google.com/>

de empresas. A seguir, apresentamos alguns exemplos de como utilizar o WordPress: criação de blogs, sites de negócios, sites de notícias, lojas online, educação e aprendizado online.

É possível implementar funcionalidades de um LMS no WordPress. Isso é útil para instituições educacionais, treinamento corporativo e até mesmo instrutores independentes que desejam oferecer cursos online. Existem várias extensões e temas disponíveis que transformam o WordPress em um ambiente de aprendizado online, alguns dos plugins mais populares são:

- **LearnDash**<sup>17</sup>: uma das extensões LMS mais conhecidas para WordPress. Ele oferece recursos avançados para criar cursos, módulos, lições, quizzes e acompanhar o progresso dos alunos.
- **Tutor LMS**<sup>18</sup>: uma extensão LMS que oferece recursos para criar cursos, quizzes, aulas ao vivo e gerenciar inscrições.
- **Namaste LMS**<sup>19</sup>: uma extensão simples que permite criar cursos, aulas e quizzes.

Ao usar o WordPress para elaboração do LMS, é possível aproveitar a flexibilidade da plataforma e personalizar a experiência de aprendizado de acordo com suas necessidades (Murad et al., 2019).

## 2.4. Design Instrucional

Design Instrucional refere-se ao processo sistemático de planejar, desenvolver e organizar experiências de aprendizagem de forma a maximizar a eficácia e a eficiência do aprendizado. O objetivo do design instrucional é criar ambientes e materiais de aprendizagem que sejam pedagogicamente sólidos e alinhados com as necessidades dos alunos e os objetivos educacionais.

---

<sup>17</sup> <https://www.learndash.com/>

<sup>18</sup> <https://www.themeum.com/product/tutor-lms/>

<sup>19</sup> <https://namaste.education/>

Existem várias abordagens e modelos de design instrucional que orientam o processo de criação de experiências de aprendizado eficazes. Segue abaixo alguns dos modelos e abordagens mais amplamente utilizados (Martenstyaro et al., 2015):

- **Modelo ADDIE:**
  - Abordagem: é um processo sequencial que envolve análise, design, desenvolvimento, implementação e avaliação de materiais e programas de ensino.
  - Explicação: cada fase do ADDIE tem um propósito específico no ciclo de design instrucional. Começa com a análise das necessidades de aprendizado e termina com a avaliação da eficácia do programa, permitindo ajustes contínuos.
- **Modelo SAM (*Successive Approximation Model*):**
  - Abordagem: é um modelo iterativo que enfatiza o desenvolvimento rápido de protótipos e iterações com base no feedback.
  - Explicação: divide o design instrucional em três fases principais: análise, design e desenvolvimento iterativo. Ele promove a colaboração e a adaptação contínua.
- **Modelo *Gagne's Nine Events of Instruction*:**
  - Abordagem: este modelo se baseia em nove eventos de ensino sequenciais, desde a captação da atenção até a avaliação do desempenho.
  - Explicação: cada evento tem um propósito específico, como a apresentação do conteúdo, a orientação da aprendizagem e a prática ativa. Foca na criação de experiências de aprendizado envolventes.
- **Modelo Dick e Carey:**
  - Abordagem: este é um modelo sistemático que enfatiza a análise detalhada das necessidades de aprendizado e a criação de um objetivo de instrução claro.
  - Explicação: o modelo inclui etapas como análise de metas, design de instrução, desenvolvimento de materiais, implementação e avaliação. É altamente estruturado.
- **Modelo 4C/ID (*Four-Component Instructional Design*):**

- Abordagem: esse modelo destaca quatro componentes essenciais para o aprendizado: tarefas autênticas, aprendizado just-in-time, aprendizado baseado em casos e aprendizado orientado pelo aluno.
- Explicação: ele é frequentemente usado em ambientes de aprendizado baseados em problemas e promove a aplicação prática do conhecimento.

A escolha do melhor modelo de design instrucional depende das necessidades específicas do público-alvo, dos objetivos de aprendizado e do contexto em que a aprendizagem ocorrerá (Trilaksono & Santoso, 2018).

### **3. Revisão bibliográfica e trabalhos relacionados**

Este capítulo apresenta a revisão bibliográfica e trabalhos relacionados. Ele é resultado de pesquisa da literatura e inclui os principais conceitos, fundamentos e métodos empregados nesta proposta de dissertação. A Seção 3.1, sobre Método de revisão, descreve o método utilizado para a revisão de estudos sobre o tema proposto, seguindo o guia apresentado por Kitchenham (Babar & Zhang, 2009). Além disso, destaca-se que a dissertação foca no desenvolvimento de uma abordagem para a aprendizagem de programação baseada em chats inteligentes, realizando uma pesquisa exploratória inicial para indicar a literatura relevante e respondendo a perguntas genéricas relacionadas ao tema. A Seção 3.2, sobre Revisão da literatura, apresenta dados quantitativos relacionados às buscas em bases científicas específicas para a dissertação. Além disso, destaca a seleção de oito trabalhos para análise qualitativa detalhada, visando responder às questões principais de pesquisa. A Seção 3.3, sobre trabalhos relacionados, descreve a seleção de trabalhos mais alinhados ao objetivo da dissertação, analisando aqueles que abordam problemas semelhantes ou têm contribuições próximas aos propósitos da pesquisa.

#### **3.1. Método de Revisão**

O método utilizado para a revisão de estudos sobre o tema proposto foi baseado no guia apresentado por Kitchenham (Babar & Zhang, 2009). Esse método identifica, avalia e interpreta toda a pesquisa existente para uma questão de pesquisa específica, área temática ou fenômeno de interesse. A revisão é composta de três fases principais: (1) planejamento e revisão, (2) realização da revisão e (3) reporte dos resultados da revisão. O planejamento é dividido em duas fases: na Fase 1, são identificadas as necessidades da revisão, enquanto na Fase 2 desenvolve-se um protocolo de revisão.

Como o foco desta dissertação é desenvolver uma abordagem para a aprendizagem de programação baseada em chats inteligentes, e para tanto, se faz necessário fazer uma pesquisa exploratória inicial para indicação da literatura sobre esse tema. Após a busca exploratória, optou-se por estabelecer perguntas mais genéricas relacionadas ao tema, o que possibilitou a obtenção de resultados mais abrangentes em

temas relacionados. Portanto, essa revisão tem por objetivo responder às seguintes questões:

*Q1: Como os chats inteligentes podem influenciar no desempenho dos estudantes de programação e, conseqüentemente, superar as barreiras dos métodos tradicionais de ensino?*

*Q2: Quais são os desafios e limitações associados ao uso de chats inteligentes na programação de computadores?*

A Tabela 1 apresenta os parâmetros da pesquisa realizada, incluindo estratégia, fontes de pesquisa e palavras-chave. Para responder às questões de pesquisa, foi elaborada uma *string* de busca para seleção preliminar dos trabalhos conforme a Tabela 1. Essa *string* de busca, foi construída para direcionar a pesquisa para recursos que abordam a interseção entre sistemas de chat inteligente (chatbot<sup>20</sup> ou gpt) e o aprendizado de programação, fundamentais para investigar como esses sistemas podem influenciar no desempenho dos estudantes de programação.

**Tabela 1** - Critérios de Busca

Estratégia	Artigos disponíveis de 2019 a 2023 (novembro) publicados em revistas e congressos em inglês e português
Fontes de pesquisa	IEEE Explore Digital Library, ACM Digital Library, Springer Link e Google Scholar
Palavras-chave	chatbot, gpt, learning, programming.
String de busca	(chatbot OR gpt) AND (learning) AND (programming)

Após a pesquisa às bases de dados online, os esforços foram direcionados para a busca e seleção de estudos primários, estabelecendo critérios de inclusão e exclusão (Tabela 2). Nessa etapa da pesquisa, foram considerados os seguintes campos durante a busca: título, resumo e palavras-chave.

---

<sup>20</sup> Foi utilizado o termo chatbot no lugar de apenas bot, pois algumas bases retornava um número excessivamente grande de resultados.

**Tabela 2** - Critérios de Inclusão

<b>Critério</b>	<b>Descrição</b>
Inclusão	Estudos que envolvam chatbot na aprendizagem e programação
Exclusão	Artigos com idiomas diferentes do inglês e português (Brasil)
	Artigos do mesmo tema e autor em diferentes bases
	Artigos que não estejam relacionados com o tema da pesquisa
	Não serão considerados dissertações de mestrado e teses doutorado

Neste trabalho, os dados foram registrados com auxílio de uma planilha desenvolvida no Microsoft Excel®. O objetivo da planilha foi facilitar o processo de classificação dos artigos e posterior extração de dados considerados relevantes para responder às questões da pesquisa. Ao mesmo tempo, foram registradas todas as informações para revisão, como: autor, fonte, dados bibliográficos, data de publicação, resumo, entre outros.

### **3.2. Execução da Revisão da Literatura**

Nesta Seção são apresentados os dados quantitativos (Subseção 3.2.1), bem como uma análise e discussão sobre estes resultados.

#### **3.2.1 Resultados Quantitativos**

A Tabela 3 apresenta resultados quantitativos sobre as buscas nas bases científicas IEEE Xplore Digital Library, ACM Digital Library, Springer Link e Google Scholar, que foram escolhidas devido à relevância para o tema dessa dissertação. Foram analisados os elementos: Título (Title), Resumo (Abstract) e Palavras-chave, aplicando-se os critérios de inclusão e exclusão. Os artigos que atendiam aos requisitos definidos foram classificados para uma leitura integral.

**Tabela 3** – Artigos selecionados

Base de dados digital	Resultado da Busca	Artigos selecionados	Artigos selecionados - Questão 1	Artigos selecionados - Questão 2
IEEE Explore Digital Library	126	30	3	3
ACM Digital Library	83	3		1
Springer Link	89	1	1	
Google Scholar	33	1		

Visando responder às questões principais de pesquisa (Q1 e Q2), foram revisados todos os trabalhos potencialmente relevantes (Tabela 3), além de ser realizada análise comparativa a partir das questões mencionadas. Desse conjunto, destacam-se 08 trabalhos, descritos e analisados nos próximos parágrafos. Em relação às questões, temos os seguintes trabalhos, apresentados na próxima subseção.

### 3.2.2 Análise e Discussão dos Resultados

Malinka et al. (2023) avaliam no artigo *On the educational impact of ChatGPT, is artificial intelligence ready to obtain a university degree*, o impacto do ChatGPT na educação universitária. Os autores investigam o uso do ChatGPT para completar exames e artigos acadêmicos. Eles discutem os benefícios no aprendizado, mas também os riscos de uso indevido, como plágio. A principal contribuição do artigo está em fornecer uma avaliação do impacto no uso do ChatGPT na educação universitária. Os autores também propõem medidas para lidar com os desafios apresentados na adoção.

Ross et al. (2023) exploram no artigo *The programmer's assistant, conversational interaction with a large language model for software development*, a interação com desenvolvimento de software. Eles destacam a importância de projetar sistemas que maximizem a sinergia entre humanos e IA. Em resumo, o artigo contribui fornecendo insights sobre como a interação com IA, pode ser eficaz e valiosa no contexto do desenvolvimento de software.

Beganovic et al. (2023) apresentam no artigo *Methods and applications of ChatGPT in software development, a literature review*, uma revisão sistemática que

explora o uso do ChatGPT no desenvolvimento de *software*. Ele destaca o potencial do ChatGPT em tarefas como reparo automático de programas e geração de código. O artigo aborda preocupações no uso do ChatGPT como dependência excessiva. Sua contribuição está em oferecer uma compreensão abrangente e direcionamento futuro sobre o uso do ChatGPT no domínio do desenvolvimento de software.

Santhana et al. (2020) fornecem no artigo *A study on machine learning based conversational agents and designing techniques*, uma visão sobre o desenvolvimento de chatbots, abordando diferentes técnicas de design. Além disso, discute *frameworks* e metodologias, oferecendo *insights* sobre a escolha de abordagens de desenvolvimento com base na aplicação. A contribuição do artigo está em fornecer um guia abrangente e prático para profissionais e interessados em compreender e aplicar chatbot em diferentes contextos.

Ganesan et al. (2020) fornecem no artigo *A survey on chatbots using artificial intelligence*, um levantamento sobre o uso de chatbot em diversas áreas, destacando sua importância e benefícios. Além disso, discute técnicas de design e interação humano - computador, oferecendo uma análise comparativa entre chatbot e aplicativos existentes.

Rakesh et al. (2022) propõem no artigo *A proposed academic chatbot system using NLP techniques*, um sistema de chatbot acadêmico, para fornecer respostas imediatas para consultas acadêmicas e não acadêmicas. A contribuição do artigo está na apresentação de um sistema, que visa facilitar o acesso a informações relevantes no site da universidade, melhorando a usabilidade e reduzindo o tempo de busca do usuário.

Ashfaque et al. (2020) fornecem no artigo *A review on techniques characteristics and approaches of na inteligente tutoring chatbot system*, uma revisão das técnicas, características e aplicações dos chatbots em diversos setores. Além disso, compara diferentes sistemas de tutoria inteligente e discute o papel crucial da IA no desenvolvimento futuro do chatbot. A contribuição do artigo está em fornecer uma visão geral do estado atual e futuro desses sistemas.

Chempavathy et al. (2022) exploram no artigo *AI based chatbots using deep neural networks in education*, o uso de chatbot baseado em inteligência artificial e redes neurais na educação, destacando a aplicação prática de métodos de aprendizado profundo, como redes neurais recorrentes de longa memória (LSTM - *Long Short-Term*

*Memory*), para treinar chatbot a fornecer respostas precisas. A contribuição do artigo reside na aplicação dessas tecnologias para melhorar a comunicação e fornecer informações acadêmicas eficientemente.

Kumar et al. (2021) analisam e comparam no artigo *Analysis of automated text generation using deep learning*, o desempenho de diferentes modelos de chatbots, como os baseados em RNN (*Recurrent Neural Networks*), GRU (*Gated Recurrent Unit*) e LSTM, para geração de texto. A contribuição do artigo está na avaliação da eficácia desses modelos em diferentes conjuntos de dados, destacando o GRU como a melhor opção para o conjunto de dados, mas ressaltando que o LSTM pode ser mais adequado para conjuntos de dados complexos.

Santos et al. (2021) apresentam no artigo *Benchmark application for scenario analysis in the educational chatbots development*, um estudo que analisou plataformas para a construção de chatbots educacionais. Após a análise, o Rasa NLU foi identificado como o mais adequado, atendendo a todas as características desejadas. A contribuição do estudo é oferecer uma base sólida para a tomada de decisões na escolha de plataformas para a construção de chatbots educacionais.

Alaoui et al. (2023) fornecem no artigo *Building inteligente chatbots tools technologies and approaches*, uma visão das abordagens utilizadas na construção de chatbots. Os autores discutem métodos de *pipeline*, modelos gerativos neurais e métodos baseados em recuperação. A contribuição do artigo está em oferecer uma comparação das forças e limitações dessas abordagens.

Prenner et al. (2022) avaliam no artigo *Can openais codex fix bugs na evaluation on quixbugs*, a capacidade do Codex da OpenAI em localizar e corrigir bugs em softwares. Os resultados mostram que o Codex é eficaz, especialmente na correção de bugs em Python. A contribuição do artigo está em demonstrar o potencial do Codex como ferramenta útil no reparo automático, o que pode economizar tempo e recursos para desenvolvedores de software.

Gonda & Chu. (2019) apresentam no artigo *Chatbot as a learning resource creating conversational bots as a supplement for teaching assistant training course*, um estudo de caso sobre a aplicação de chatbot na educação, focando no treinamento de assistente educacional. A contribuição do artigo está em fornecer orientações práticas

para a implementação de chatbots em ambientes educacionais, demonstrando como essa tecnologia pode abordar desafios enfrentados nesse cenário.

Ji & Yuan. (2022) investigam no artigo *Conversational intelligent tutoring systems for online learning what do students and tutors say*, as percepções de estudantes e tutores sobre o uso de um Sistema de tutoria inteligente para o aprendizado online. Os resultados indicam uma visão positiva sobre o potencial do chatbot em aumentar o engajamento dos estudantes. A contribuição do artigo está em fornecer *insights* valiosos sobre o potencial impacto do ITS no aprendizado online.

Karri & Kumar. (2020) abordam no artigo *Deep learning techniques for implementation of chatbots*, a implementação de chatbot utilizando técnicas de aprendizado profundo. O artigo apresenta comparações entre diferentes chatbots. A contribuição do artigo está em oferecer *insights* práticos e avançar o conhecimento sobre as técnicas de aprendizado profundo no contexto da implementação de chatbots.

Kita et al. (2020) descrevem no artigo *Development of a moodle ui using line chat for casual learning as a part of a learner assistive LMS*, o desenvolvimento de um bot com a aplicação LINE, que permite ao aluno interagir com o sistema de gestão da aprendizagem Moodle. A contribuição é a integração entre o LINE e o Moodle, visando melhorar a interação e a motivação dos alunos no processo de aprendizagem.

Mallikarjuna et al. (2021) abordam no artigo *Development of information technology telecom chatbot, an artificial intelligence and machine learning approach*, o desenvolvimento de um chatbot com foco em telecomunicações voltado para a educação, visando superar desafios e dar suporte constante aos alunos. A contribuição do artigo está na sugestão do chatbot para incluir informações sobre circuitos de eletrônica e telecomunicações, visando aprimorar no processo de ensino e aprendizagem.

Sophia & Jacob. (2021) apresentam no artigo *Edubot-a chatbot for education in covid19 pandemic and vqabot comparison*, o desenvolvimento de um chatbot educacional para apoiar estudantes durante a pandemia de Covid-19. A contribuição do artigo está na inovação tecnológica para fornecer suporte acadêmico personalizado e na análise comparativa que oferece *insights* valiosos sobre a eficácia dessas tecnologias. A integração com a plataforma Dialogflow e as sugestões de melhorias futuras também contribuem para o avanço e aprimoramento de chatbots educacionais.

Vazhayil et al. (2019) abordam no artigo *Focusing on teacher education to introduce AI in schools, perspectives and illustrative findings*, a introdução da IA nas escolas, com foco na formação de professores na Índia. O artigo destaca desafios enfrentados pelos professores, como lacunas na comunicação e infraestrutura, e discute a perspectiva dos professores em relação ao potencial da IA. Os resultados indicam que a formação dos professores em IA pode ser eficaz ao adotar abordagens pedagógicas criativas, como o aprendizado exploratório e o ensino entre pares. A contribuição do artigo está em fornecer *insights* relevantes para orientar futuros esforços de implementação de IA na educação.

Banić et al. (2023) apresentam no artigo *Pair programming education aided by ChatGPT*, resultados de uma pesquisa sobre o uso da programação em pares com o uso do ChatGPT. A pesquisa revela uma mudança na motivação dos alunos em relação à programação tradicional. O estudo sugere que essa ferramenta pode ter um papel significativo no engajamento e na aprendizagem dos alunos em programação. A contribuição do artigo está em destacar o potencial dessa abordagem para melhorar a educação em programação.

Alam (2021) analisa no artigo *Should Robots Replace Teachers? Mobilisation of AI and Learning Analytics in Education*, o impacto da IA na educação, abordando a administração e o ensino personalizado. A pesquisa destaca a ampla adoção da IA por instituições educacionais, permitindo a personalização do currículo e materiais para atender às necessidades individuais dos alunos. Além disso, o estudo ressalta a importância de políticas e iniciativas para aproveitar os benefícios da IA e mitigar possíveis efeitos adversos na educação.

Koundinya et al. (2020) apresentam no artigo *Smart college chatbot using ML and python*, um chatbot utilizando inteligência artificial e processamento de linguagem natural. A principal contribuição é a melhoria da interação dos usuários com o site da faculdade, permitindo que façam perguntas sobre atividades acadêmicas. Essa contribuição é significativa para aprimorar a experiência do usuário e fornecer um canal eficaz de comunicação para a comunidade acadêmica.

Kumarasiri et al. (2022) propõem no artigo *Student teaching and learning system for academic institutions*, um sistema que combina técnicas de Processamento de

Linguagem Natural, Aprendizado de Máquina e Lógica Fuzzy para coletar e analisar o feedback dos alunos em instituições acadêmicas. A contribuição do artigo está na apresentação de um método eficaz para avaliar o desempenho dos alunos.

Lajko et al. (2022) apresentam no artigo *Towards javascript program repair with generative pre-trained transformer GPT-2*, a aplicação do modelo GPT-2, para reparar programas JavaScript, demonstrando sua capacidade de correção do código. Além disso, os autores disponibilizam o conjunto de dados e os modelos treinados, contribuindo para pesquisas futuras na área de reparo automático de programas.

Henno et al. (2023) discutem no artigo *What do we know about learning conversations with ChatGPT*, a capacidade do chatbot em produzir conversas humanas, baseadas em redes neurais e aprendizado profundo. A contribuição do artigo está em fornecer insights sobre o impacto desses chatbots na educação, incentivando a reflexão sobre os benefícios e desvantagens de sua implementação.

Martenstyaro et al. (2015) propõem no artigo *A Framework for Designing Survey Training based on 3D Virtual Learning Environment Using SLOODLE*, um framework para desenvolver treinamentos virtuais baseados em um ambiente de aprendizagem 3D. A contribuição, está na integração do modelo ADDIE com um framework de simulação, oferecendo uma solução virtual mais acessível para retomar treinamentos cancelados devido a restrições de custo.

Jalil et al., (2023) analisam no artigo *ChatGPT and Software Testing Education: Promises & Perils*, o desempenho do ChatGPT, em responder a perguntas relacionadas a testes de software. Eles descobrem que o ChatGPT, fornece respostas corretas ou parcialmente corretas em 55,6% das vezes, com uma precisão ligeiramente maior em contextos compartilhados. A contribuição do artigo, reside em fornecer insights valiosos sobre o potencial e as limitações no uso em ambientes educacionais, bem como em destacar a importância de considerar a confiabilidade e o uso ético dessa tecnologia.

### 3.3. Trabalhos Relacionados

Com base na análise dos trabalhos pré-selecionados, foi realizada a seleção de trabalhos com objetivos mais próximos a esta dissertação, por abordarem problemas semelhantes ou possuem contribuições próximas às pretendidas. Tais trabalhos são analisados com maior profundidade nesta Seção.

O artigo *The robots are coming, exploring the implications of openai CODEX on introductory programming* (Finnie-Ansley et al., 2022) explora as implicações do *CODEX* da OpenAI na programação inicial, comparando seu desempenho com o de estudantes. O *CODEX* é um modelo de inteligência artificial desenvolvido pela OpenAI, especializado em entender e gerar código de programação, oferecendo suporte na escrita e otimização de código. Ele destaca a capacidade do *CODEX* superar a maioria dos estudantes e discute questões de integridade acadêmica relacionadas à prevenção do plágio. A contribuição do artigo está em fornecer insights sobre os desafios e oportunidades apresentados por tecnologias como o *CODEX* na educação, destacando a necessidade de adaptação a essa mudança. O artigo apresenta um estudo comparativo do desempenho do *CODEX* com o de estudantes em exames de programação inicial, fornecendo uma visão clara da capacidade do *CODEX* de gerar soluções de código de alta qualidade. A discussão sobre integridade acadêmica destaca a importância de considerar como o *CODEX* pode afetar a prevenção de plágio, levantando preocupações relevantes para a comunidade acadêmica. A análise sobre a avaliação automática de códigos de estudantes em relação ao uso do *CODEX* oferece *insights* valiosos sobre como a tecnologia pode impactar os métodos de avaliação tradicionais.

O artigo *Parallel intelligent education with ChatGPT* (Wang et al., 2023) apresenta um modelo de educação paralela, destacando o uso do ChatGPT para complementar o aprendizado em sala de aula. Portanto, a contribuição do artigo reside na exploração de um novo paradigma educacional e na identificação de oportunidades e desafios associados à implementação bem-sucedida da educação paralela com o ChatGPT. O artigo fornece uma visão abrangente das implicações do uso do ChatGPT na educação, abordando tanto os benefícios quanto as preocupações éticas e práticas. Ele também destaca a importância de integrar o ChatGPT em um sistema de educação

paralela, reconhecendo os desafios associados à implementação bem-sucedida desse modelo.

O artigo *A conversational assistant on mobile devices for primitive learners of computer programming* (Lin et al., 2019) propõe e desenvolve um chatbot baseado no IBM Watson Assistant e no Facebook Messenger, para auxiliar estudantes iniciantes na aprendizagem de programação de computadores. A contribuição do artigo está na aplicação da tecnologia para melhorar o engajamento dos alunos e seu interesse na programação. A aplicação do IBM Watson Assistant e do Facebook Messenger para desenvolver o chatbot é uma abordagem inovadora e prática visando melhorar a experiência de aprendizagem dos alunos. A aplicação da teoria da carga cognitiva para o design instrucional destaca o cuidado com a eficácia do chatbot em facilitar a compreensão e retenção do conhecimento por parte dos alunos. A realização de experimentos controlados para avaliar o impacto do chatbot no desempenho dos alunos evidencia a preocupação com a validação e eficácia do chatbot como ferramenta educacional.

O artigo *Chatbot script design for programming language learning* (Lin, 2022) aborda a criação de chatbots para auxiliar no ensino de linguagens de programação para estudantes iniciantes. Em resumo, o artigo oferece uma visão abrangente sobre a criação de chatbots para o ensino de programação, destacando a importância de considerar o perfil do usuário, a interação do chatbot e o contexto de suporte, contribuindo para o avanço do uso de chatbots no ensino de programação. O artigo destaca a importância da aplicação no ensino de programação para estudantes iniciantes, fornecendo uma contribuição significativa para a compreensão do papel do chatbot nesse contexto educacional. A discussão sobre a escolha de plataformas comerciais, como IBM Watson, fornece informações práticas sobre as considerações envolvidas na implementação de chatbot educacional, enriquecendo o entendimento sobre as opções disponíveis para os desenvolvedores.

O artigo *Code free bot, an easy way to jumpstart your chatbot* (Luo & Gonda, 2019) aborda a integração de chatbot no ensino superior, na criação de experiências de aprendizagem online. Os autores propõem o uso de plataformas comerciais para simplificar o desenvolvimento de chatbot, permitindo que os professores se concentrem

no diálogo em vez dos aspectos técnicos de IA. A contribuição principal está em oferecer orientações práticas, bem como recursos para capacitar os educadores a adotar e implementar chatbots de forma eficaz em seus ambientes de ensino. O artigo fornece uma visão prática e acessível para professores e profissionais da educação, interessados em adotar chatbot e outras tecnologias inovadoras no ensino superior. Destaca-se a importância das plataformas comerciais para simplificar o desenvolvimento de chatbots, permitindo que os educadores se concentrem na utilização prática dessas ferramentas.

O artigo *Learning support system using chatbot in kejar c package homeschooling program* (Murad et al., 2019) apresenta um sistema para aprendizagem, que integra o chatbot com o aplicativo de mensagens LINE e um sistema de gerenciamento de conteúdo (CMS - *Content Management System*) para estudantes e professores. A pesquisa aponta que o uso do chatbot em smartphones pode maximizar a aprendizagem do aluno, aumentando o interesse e o desempenho dele. A contribuição do artigo está em apresentar uma solução que visa melhorar o processo de ensino e aprendizagem. Os autores destacam a importância da inovação tecnológica na educação, fornecendo uma ferramenta que atende às necessidades de estudantes e professores. Além disso, o estudo ressalta a relevância do chatbot e do CMS na promoção da independência do aluno, na facilitação do acesso a materiais de estudo e na criação de um ambiente de aprendizagem interativo.

O artigo *Programming an educational chatbot to support virtual learning* (Ai et al., 2020) descreve a criação e implementação de um chatbot educacional chamado Ed.dy, projetado para apoiar o ensino virtual e híbrido. O chatbot oferece recursos como lembretes, cronômetros e enquetes visando melhorar a experiência de aprendizado virtual. A contribuição do artigo está na criação de uma ferramenta prática e eficaz para melhorar a comunicação e a experiência de aprendizado do aluno. O chatbot Ed.dy representa uma contribuição significativa para a educação, fornecendo uma ferramenta prática e eficaz para melhorar a comunicação e a experiência de aprendizado dos alunos em ambientes virtuais e híbridos. A variedade de recursos oferecidos pelo chatbot, como lembretes, cronômetros, enquetes e salas de discussão, demonstra uma abordagem abrangente para atender às necessidades dos alunos e professores no contexto do ensino virtual.

O artigo *Pyo a chatbot assistant for introductory programming students* (Carreira et al., 2022) apresenta a avaliação do chatbot Pyo, desenvolvido para auxiliar estudantes iniciantes em programação. A pesquisa exploratória revelou que o Pyo foi útil, mas também apontou a necessidade de explicações mais diretas sobre suas funcionalidades. A contribuição do artigo está na apresentação de um chatbot projetado para oferecer suporte personalizado aos estudantes e na discussão sobre a importância e os desafios da implementação de chatbots na educação. A implementação do chatbot Pyo oferece uma solução inovadora para auxiliar estudantes iniciantes em programação, abordando a necessidade de suporte personalizado em um ambiente de aprendizado online. A avaliação com 22 estudantes fornece insights sobre a eficácia do chatbot, destacando a importância de considerar as necessidades e perfil do usuário.

A Tabela 4 apresenta uma síntese dos trabalhos relacionados, incluindo o foco de cada trabalho, a metodologia utilizada e a contribuição de cada um deles, bem como compara com esta proposta de dissertação.

**Tabela 4** – Trabalhos Relacionados

Artigo	Objetivo	Contribuição	Questão Q1	Questão Q2
(Finnie-Ansley et al., 2022)	Codex OpenAI na programação	insights sobre o impacto do Codex no desempenho, integridade acadêmica e avanço tecnológico.	Não há comparações entre o uso de chats inteligentes e métodos tradicionais de aprendizagem.	Integridade acadêmica, preocupação com dependência da ferramenta.
(Wang et al., 2023)	Educação paralela com ChatGPT	Oportunidades e desafios associados à implementação do ChatGPT na educação paralela.	Uso do ChatGPT para fornecer feedback, suporte, promovendo a aprendizagem colaborativa, criatividade e inovação.	Preocupação com a saúde mental e bem-estar dos alunos no uso excessivo da tecnologia.
(Lin et al., 2019)	IBM Watson na aprendizagem de programação	Insights sobre o impacto do IBM Watson com Facebook Messenger relacionado as questões de desempenho e validação da ferramenta educacional	Melhoria no engajamento, acesso instantâneo a informações e suporte em tempo real	limitações na compreensão da linguagem natural e dependência de infraestrutura.
(Lin, 2022)	Chatbots para aprendizagem de programação	insights sobre o uso de chatbots no ensino de programação, destacando a importância do perfil do usuário.	O artigo destaca a disponibilidade 24/7, engajamento adicional para os alunos, ajudado a superar desafios.	Preocupações em relação ao perfil do usuário, interação do chatbot e o contexto de informações de suporte.
(Luo & Gonda, 2019)	Adoção de Chatbots pelos educadores	insights sobre como os Chatbots, podem ser utilizados contribuir com a gestão dos cursos.	Ele destaca o suporte 24/7, gestão de cursos e apoio aos alunos de forma assíncrona.	Ele destaca a criação de chatbots tradicionalmente, exigindo conhecimento avançado de programação
(Murad et al., 2019)	Integração do aplicativo Line com CMS	Integração do aplicativo LINE com (CMS), facilitando o trabalho dos professores no processo de ensino e aprendizagem.	Pode maximizar as funções de aprendizagem dos alunos, aumentando o interesse e desempenho.	controle de acesso e confiabilidade dos dispositivos móveis.
(Ai et al., 2020)	Chatbot (Ed.dy) no Ensino virtual e Híbrido.	Chatbot educacional Ed.dy, oferecendo recursos de lembrete, cronômetro, enquetes e salas de diálogo.	Ele enfatiza as ferramentas oferecida pelo chatbot, melhorando a produtividade do usuário.	O artigo não aborda os desafios e limitações associados ao uso de chats inteligentes
(Carreira et al., 2022)	Chatbot (Pyo) na fase inicial de desenvolvimento.	abordagem inovadora para auxiliar estudantes iniciantes em programação, destacando tanto a eficácia quanto os desafios associados ao uso de chatbots.	Suporte aos estudantes introvertidos.	Preferência da interação com colegas e professores em vez do chatbot.
<i>Este Trabalho</i>	<i>Integração ChatGPT com LMS</i>	<b>Proposta de um LMS Integrado com ChatGPT: investigação sobre a Influência de Chats Inteligentes no Desempenho dos Alunos</b>		

Com base na análise dos trabalhos selecionados (Tabela 4), destacam-se:

- No contexto da aprendizagem de programação baseada em chats inteligentes, a palavra "ferramenta" se refere a tecnologias, como os chats inteligentes, utilizados para auxiliar os estudantes durante o processo de aprendizado e prática da programação. Esses chats inteligentes são concebidos para fornecer suporte personalizado, orientação e esclarecimento de dúvidas, visando auxiliar os alunos a compreenderem os princípios da lógica de programação e aprimorarem suas habilidades.
- Personalização da experiência de aprendizagem envolve o uso de chats inteligentes não apenas como uma ferramenta para fornecer respostas prontas, mas também como uma forma de oferecer uma experiência de aprendizagem adaptada às necessidades individuais de cada estudante. Isso significa que o chat inteligente é explorado de maneira a responder de forma personalizada, considerando o nível de conhecimento e necessidades específicas de cada usuário. Essa abordagem beneficia tanto iniciantes quanto estudantes avançados, garantindo que recebam suporte e orientação adequados às suas necessidades.
- A metodologia adotada visa desenvolver um aplicativo integrado, facilitando a transição entre o uso do LMS e o chat inteligente, proporcionando uma experiência mais contínua. O chat estará incorporado aos conteúdos e tarefas disponibilizados ao estudante, eliminando a necessidade de sair do ambiente.
- Ainda referente à metodologia, destaca-se o uso de design instrucional como processo de criação de conteúdo integrado às atividades com o chat inteligente (Kita et al., 2020).

Embora os artigos supracitados foquem elementos importantes para o avanço do ensino de programação com o uso de chats inteligentes, tais como estudos empíricos e integração com outras tecnologias, nenhum dos estudos abordou a integração de um LMS e chat inteligente por meio de um modelo de design instrucional (no desenvolvimento do sistema e conteúdo) para o ensino de programação. No melhor do nosso conhecimento, o presente estudo é original em endereçar essa linha de investigação.

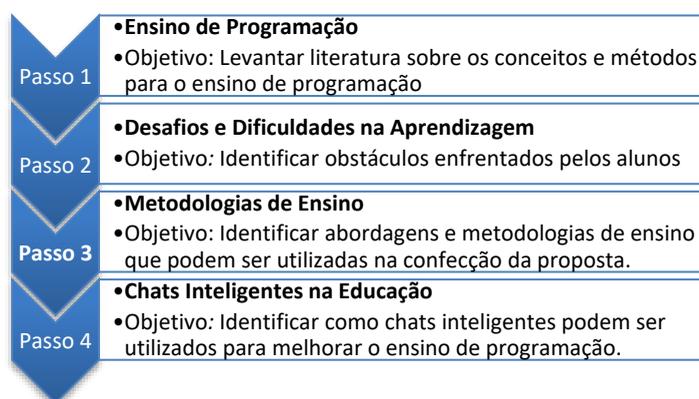
## 4. Metodologia de Pesquisa e Análise Comparativa

Este capítulo apresenta primeiramente a metodologia de pesquisa utilizada nesta dissertação (Seção 4.1). Em sequência, é realizada uma análise comparativa de elementos-chave que guiam o design e desenvolvimento do protótipo proposto, incluindo: chats inteligentes (Seção 4.2), abordagens pedagógicas com foco no design instrucional (Seção 4.3) e LMS como base tecnológica para a implementação do protótipo (Seção 4.4).

### 4.1. Metodologia de Pesquisa

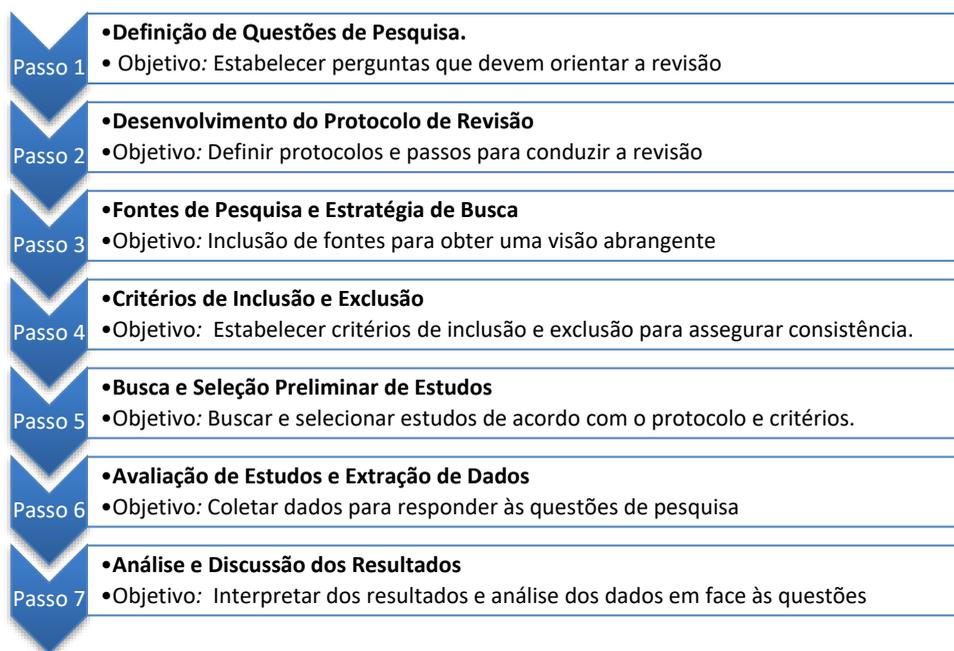
A metodologia de pesquisa empregada na dissertação está estruturada em 6 etapas, que visam, a partir da literatura da área, propor e avaliar um método que use chats inteligentes no ensino de programação. Estão descritas cada uma dessas etapas (algumas etapas podem ter atividades em paralelo):

- 1) *Etapa 1*: revisão exploratória sobre o tema, definição de lacunas e dificuldades no ensino de programação. A Figura 1 apresenta os passos da primeira etapa e respectivos objetivos, que incluem: revisões exploratórias sobre o ensino de programação, desafios e dificuldades na aprendizagem de programação, metodologias para o ensino de programação e o uso de chats inteligentes no ensino de programação. Os resultados desta etapa foram apresentados no Capítulo 2 desta dissertação.



**Figura 1** – Passos da revisão exploratória inicial

2) *Etapa 2*: revisão sobre uso de chats inteligentes no ensino de computação. A Figura 2 apresenta os passos da revisão sistemática com os respectivos objetivos. A execução da revisão, resultados e trabalhos relacionados foram apresentados no Capítulo 3 desta dissertação.



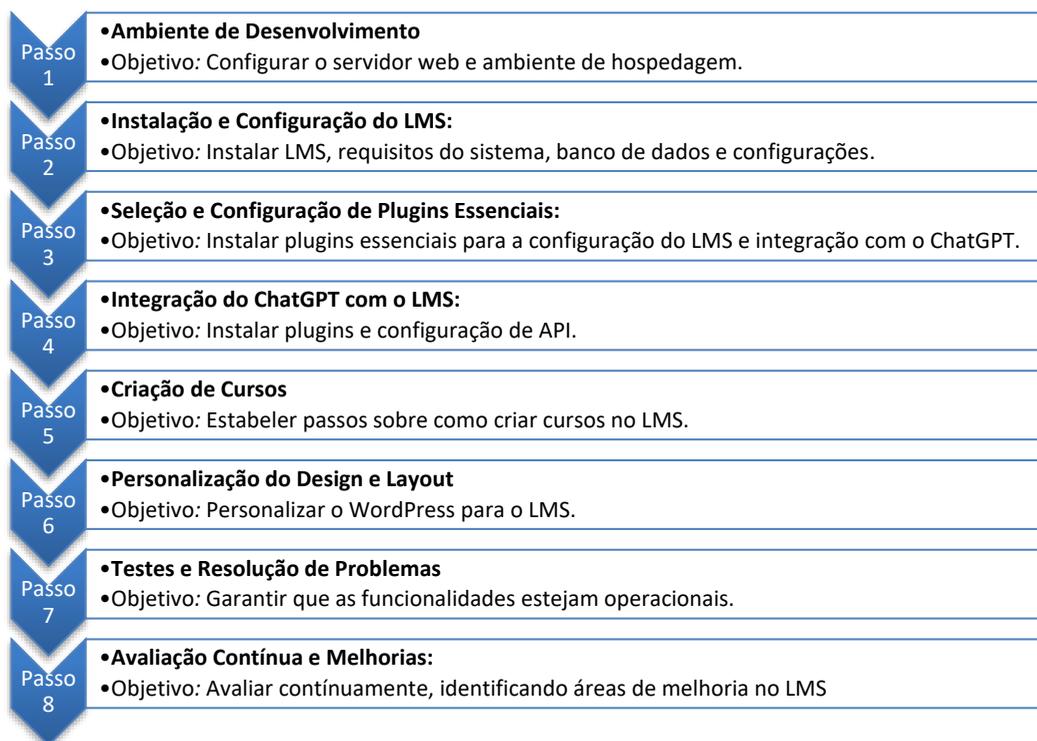
**Figura 2** – Passos da revisão sistemática

3) *Etapa 3*: análise comparativa de chats inteligentes, tecnologias e modelos de design instrucional para o ensino de programação, visando definir quais serão considerados no design do protótipo proposto. A Figura 3 apresenta os passos desta análise, bem como os respectivos objetivos. A análise e os resultados obtidos são apresentados nas próximas seções deste capítulo.



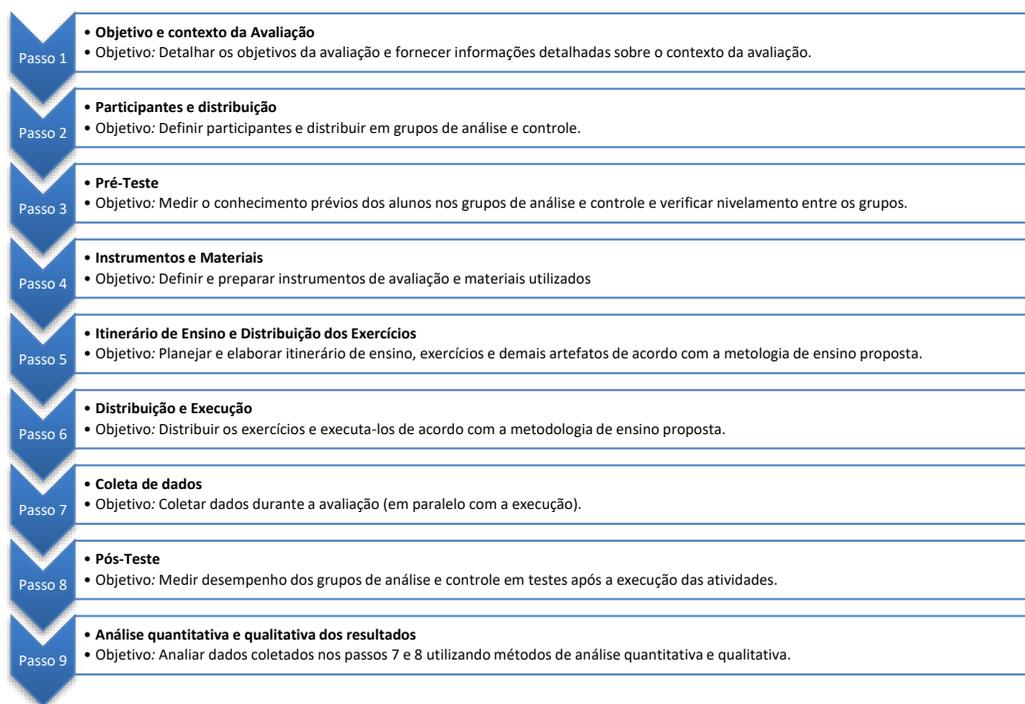
**Figura 3** – Passos da análise comparativa

4) *Etapa 4*: design e desenvolvimento de um protótipo funcional que materializa a abordagem, bem como permite avaliação com estudantes. A Figura 4 apresenta os passos de desenvolvimento do protótipo proposto. O Capítulo 5 apresenta o design, desenvolvimento, características e funcionalidades de protótipo.



**Figura 4** – Passos do desenvolvimento do protótipo

5) *Etapa 5*: avaliação com alunos do ensino técnico, incluindo com o objetivo geral avaliar o impacto da abordagem e protótipo no ensino de programação. A Figura 5 apresenta os passos e objetivos de cada passo da avaliação. O Capítulo 6 apresenta o plano de avaliação completo, as características dos participantes e a metodologia de análise. Este plano foi executado no Senac Mogi Guaçu durante o primeiro semestre de 2024.



**Figura 5** – Passos da avaliação com alunos do ensino técnico

## 4.2. Comparação e Seleção de Chats Inteligentes para o Ensino de Programação

A Tabela 5 foi desenvolvida por meio de uma análise comparativa considerando características-chave de cada chat inteligente considerado para este trabalho. Em cada entrada são destacados os pontos positivos e negativos de cada chat inteligente. O objetivo é oferecer uma visão clara e concisa, facilitando a comparação e contribuindo para a escolha adequada em diferentes cenários.

Na Tabela 5 é apresentada uma comparação entre diversos chats inteligentes existentes na data de início da escrita desta proposta de dissertação, incluindo ChatGPT, Google Bard, Dialogflow, IBM Watson e Amazon Lex. Em cada linha, são destacados os pontos positivos (vantagens percebidas) e negativos (possíveis desafios ou limitações) de cada chat inteligente. O propósito dessa análise é oferecer uma referência ágil para orientar a seleção.

**Tabela 5** – Análise Comparativa de Chats Inteligentes para o Ensino de Programação

Chatbot	Pontos Positivos	Pontos Negativos	Cenário recomendado para uso
ChatGPT (GPT-3)	Compreensão avançada de linguagem natural.	Pode gerar respostas incorretas ou inapropriadas.	Assistência Virtual, Conversação Natural
Google Bard	Integração eficaz com pesquisa na web	Limitado ao ecossistema Google	Busca de informações, assistência personalizada.
Dialogflow	Poderoso processamento de linguagem natural	Limitações na compreensão de linguagem natural	Desenvolvimento de Chatbots, Interações Complexas
IBM Watson	Recursos avançados de análise de texto.	Exige habilidades técnicas para configuração.	Análise de Texto Avançada e Pesquisa de Dados.
Amazon Lex	Integração simples com serviços da AWS	Pode não ser tão avançado em compreensão de linguagem natural.	Aplicações Empresariais e Automatização de Tarefas

A escolha de integrar o ChatGPT ao nosso LMS é respaldada por sua avançada compreensão da linguagem natural e sua capacidade de manter conversas naturais e fluidas com os usuários. Esta integração não apenas oferece assistência virtual aos alunos, mas também promove uma experiência de aprendizado envolvente e interativa. Outro fator a ser considerado é que o ChatGPT é uma ferramenta familiar para muitos usuários, contribuindo para sua aceitação e uso efetivo dentro de nosso ambiente educacional.

Durante a escrita desta proposta, foram lançados novos chats inteligentes e versões, no entanto, optou-se por realizar uma atualização desta avaliação apenas antes da execução das atividades. Com isso, evita-se retrabalho, bem como é importante salientar que uma versão mais avançada tende a trazer melhorias nos resultados. Assim, independente da escolha, este trabalho trará uma linha de base (*baseline*) para trabalhos futuros, até então inexistentes.

### 4.3. Comparação e Seleção de Abordagens Pedagógicas para o Ensino de Programação

A fim de selecionar uma abordagem pedagógica para este trabalho e guiar o desenvolvimento do protótipo, foi elaborada uma tabela comparativa com diferentes modelos de Design Instrucional. Para cada modelo foram identificados seus pontos positivos e negativos. A análise considerou características como abordagem, estrutura, flexibilidade e complexidade (Falcade, 2015).

A Tabela 6 compara modelos de Design Instrucional destacando pontos positivos e negativos de cada um na visão deste trabalho, incluindo: ADDIE, SAM, Gagné's Nine, Dick e Carey, e 4C/ID. A análise visa oferecer uma visão equilibrada para orientar a escolha do modelo mais adequado às necessidades do projeto, considerando diferentes abordagens e características.

**Tabela 6** – Análise Comparativa de Modelos de Design Instrucionais para o Ensino de Programação

Modelo de Design Instrucional	Pontos Positivos	Pontos Negativos	Exemplos de Aplicações
ADDIE	Abordagem sistemática e bem estruturada.	Pode ser visto como muito rígido e linear.	Cursos online, treinamentos corporativos
SAM	Ênfase na colaboração e iteração	Requer mais tempo para a fase de design inicial.	Projetos educacionais interativos e simulações com etapas: análise, design e desenvolvimento.
Gagné's Nine	Estrutura clara e sequencial.	Pode ser menos flexível para projetos não lineares.	Módulos de treinamento sequenciais e aulas tradicionais com nove etapas.
Dick e Carey	Abordagem baseada em pesquisa e teoria.	Pode ser visto como muito estruturado.	Desenvolvimento de cursos acadêmicos e treinamento corporativo com nove etapas.
4C/ID	Foco na construção do conhecimento.	Pode ser complexo de implementar e requer planejamento cuidadoso.	Treinamento profissional e simulações complexas com quatro componentes.

A escolha do modelo ADDIE para elaborar nosso LMS com o ChatGPT como guia para avaliar a ferramenta com os alunos foi baseada em sua abordagem sistemática e estruturada. O modelo ADDIE oferece uma metodologia clara e sequencial, que nos

permite organizar o processo de desenvolvimento do LMS. Começando pela análise das necessidades educacionais, seguida pelo design do sistema, desenvolvimento da plataforma, implementação do ambiente e, finalmente, a avaliação do impacto e eficácia do LMS. Essas etapas bem definidas foram um fator-chave na escolha do ADDIE, pois muitos outros modelos não fornecem essa clareza e sequencialidade em seus processos de desenvolvimento.

#### **4.4. Comparação e Seleção de Tecnologias para Serem Utilizadas no Desenvolvimento do Protótipo**

Para comparação e seleção de tecnologias a serem empregadas, foi elaborada uma tabela que destaca diferentes LMSs, apresentando seus pontos positivos e negativos na visão deste trabalho. Cada LMS foi avaliado, considerando características como: código aberto, recursos, custos e flexibilidade. Essa análise proporciona uma visão abrangente para orientar a escolha do LMS mais adequado às necessidades do desenvolvimento do protótipo (Maquiné, 2020).

A Tabela 7 apresenta uma síntese da análise comparativa de LMSs, destacando os pontos positivos e negativos de cada um. Os seguintes sistemas foram considerados por serem amplamente utilizados e de conhecimento do pesquisador: Moodle, Canvas, Blackboard, Google Classroom, Edmodo e WordPress. Embora o WordPress seja originalmente um CMS, ele foi incluído na tabela devido à sua flexibilidade e capacidade de ser transformado em um LMS por meio de extensões que fazem parte de sua plataforma.

**Tabela 7** – Análise Comparativa de tecnologias de LMS para construção do Protótipo

LMS	Pontos Positivos	Pontos Negativos	Integrações com outras Plataformas
Moodle	Código aberto e gratuito.	Pode exigir configuração técnica inicial.	Diversas integrações possíveis
Canvas	Interface amigável e intuitiva.	Custo associado, não é gratuito.	Integrações com outras ferramentas Google, Microsoft
Blackboard	Recursos avançados para ensino online.	Custo geralmente elevado.	Integrações com sistemas acadêmicos e de terceiros
Google Classroom	Integração perfeita com o Google Workspace.	Limitado em funcionalidades comparado a outros LMSs	Integrado nativamente ao Google Workspace.
Edmodo	Plataforma gratuita com opção de upgrade.	Recursos limitados em comparação com outros LMSs	Integrações limitadas
WordPress	Flexibilidade para criar cursos e conteúdo.	Requer a configuração de plugins e temas específicos para fins de ensino.	Integração com plugins de aprendizado

O WordPress foi escolhido principalmente devido à sua flexibilidade, existência de várias extensões e ao fato de ser uma plataforma gratuita, possibilitando uma melhor customização para o desenvolvimento deste trabalho. A plataforma possibilita a personalização da experiência de aprendizado por meio de plugins específicos, como LearnDash, Tutor LMS, MasterStudy LMS e Namaste LMS, que oferecem recursos avançados para a criação e administração de cursos online. Essa abordagem proporciona uma solução adaptável e robusta para a construção do protótipo.

## 5. Design, Desenvolvimento e Funcionamento do Sistema CodeChatPro

Este capítulo apresenta o protótipo do sistema CodeChatPro, um sistema para auxílio ao ensino de programação utilizando chats inteligentes e design instrucional. A Seção 5.1 apresenta o design e desenvolvimento do sistema, enquanto a Seção 5.2 descreve as funcionalidades e o uso do sistema.

### 5.1. Design e Desenvolvimento do Sistema CodeChatPro

O desenvolvimento de um LMS é um processo abrangente, desde a concepção inicial até a implementação, configuração e uso na prática. Neste trabalho, são utilizados componentes e extensões prontos para criar um sistema integrado de LMS e Chats Inteligentes. A Figura 4 (ver Capítulo 4) apresenta uma visão geral dos passos executados no design e desenvolvimento do protótipo, cujos pontos-chave são descritos nesta Seção.

A Figura 6 apresenta o diagrama de caso de uso mais geral para o protótipo proposto, incluindo os dois autores principais, o professor e o aluno. Os seguintes atores foram definidos: (1) o Professor é o responsável por ministrar o curso, podendo interagir com o ChatGPT; e (2) o Aluno é o envolvido no processo de aprendizagem, o qual inclui a interação com o ChatGPT.

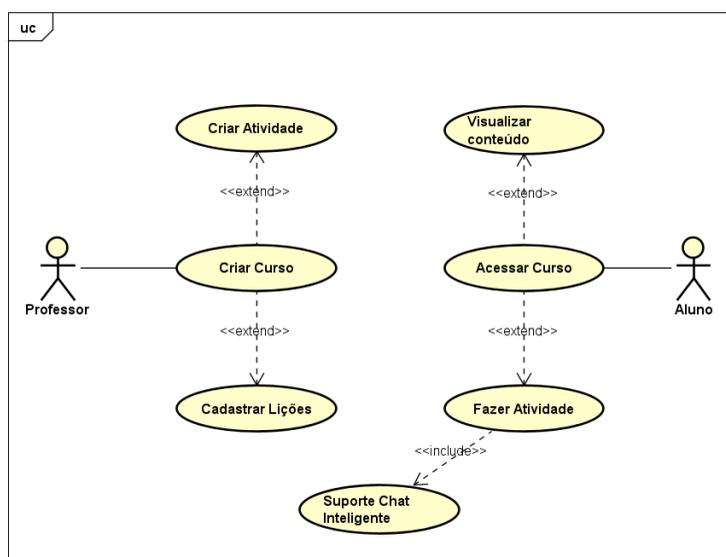


Figura 6 – Diagrama de Caso de Uso para o Sistema CodeChatPro

Os seguintes cenários estão associados aos casos de uso previstos na visão do ator professor:

- *Criar curso*: como professor, quero criar um novo curso no sistema, para disponibilizar materiais educacionais aos alunos.
- *Cadastrar atividade*: como professor, quero poder cadastrar atividades para os cursos, para avaliar o desempenho dos alunos.
- *Cadastrar lição*: como professor, desejo cadastrar lições no curso para fornecer conteúdo didático aos alunos.

Os seguintes cenários estão associados aos casos de uso previstos na visão do ator aluno:

- *Acessar curso*: como aluno, quero acessar os cursos disponíveis no sistema, para começar a aprender no meu próprio ritmo.
- *Realizar atividades*: como aluno, desejo fazer atividades atribuídas pelos professores para praticar e aplicar o conhecimento adquirido.
- *Suporte chat inteligente*: como aluno, durante a execução da atividade designada pelo professor, tenho a necessidade de utilizar o suporte do Chat Inteligente. Este recurso visa proporcionar uma assistência interativa para a aplicação eficaz do conhecimento adquirido.
- *Visualizar conteúdo*: como aluno, quero visualizar o conteúdo do curso, incluindo lições e materiais, para compreender os conceitos apresentados.

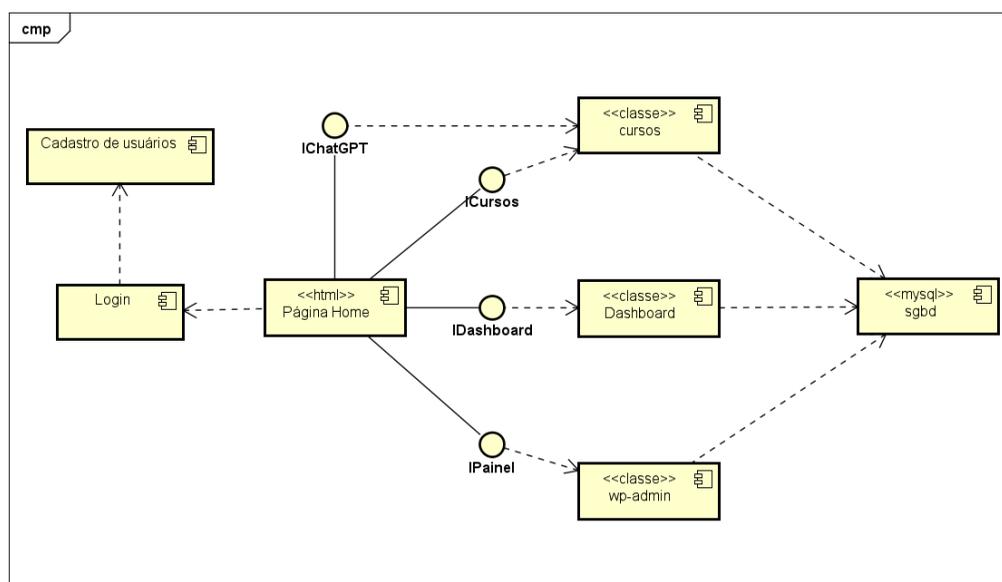
A Tabela 8 apresenta a sequência de passos para utilização do sistema. Os passos de 1 a 3 ocorrem durante a elaboração do curso, enquanto os passos de 4 a 7 ocorrem repetidas vezes por diversos alunos durante o processo de aprendizagem.

**Tabela 8** – Sequência de passos previstas como fluxo normal de utilização do sistema

	<b>Passo</b>	<b>Descrição</b>
1	Configuração do Curso	Antes de criar o curso, o professor realiza uma análise para identificar objetivos de aprendizagem. Em seguida, acessa o LMS para criar o curso, definindo objetivos, módulos e atividades.
2	Integração do ChatGPT	Durante a fase de design, o professor configura a integração do ChatGPT, ativando a ferramenta para oferecer suporte adicional aos alunos.
3	Design de Atividades	Utilizando o ADDIE, o professor projeta atividades considerando a interação do ChatGPT. Identifica pontos em que os alunos

		podem ter dúvidas e planeja o uso do ChatGPT para fornecer suporte.
4	Acesso dos Alunos	Na fase de implementação, os alunos acessam o curso no LMS, exploram módulos e iniciam as atividades propostas.
5	Consulta ao ChatGPT	Durante a implementação, os alunos têm a opção de utilizar o ChatGPT para obter esclarecimentos sobre conceitos, dúvidas ou desafios específicos.
6	Interpretação e Resposta do ChatGPT	Na fase de avaliação, o ChatGPT interpreta as perguntas dos alunos, fornecendo respostas contextualizadas, explicações detalhadas ou exemplos práticos.
7	Interagir com ChatGPT	Também na fase de avaliação, tanto o professor quanto o aluno podem interagir com o ChatGPT para obter suporte, esclarecimentos ou ajuda com o conteúdo do curso

**Figura 7** apresenta o diagrama de componentes do sistema CodeChatPro, incluindo, além dos componentes, as principais interfaces entre eles.



**Figura 7** – Diagrama de componentes do Sistema CodeChatPro

A seguir, apresenta-se uma descrição detalhada do diagrama de componentes:

- *Login*: este componente permite que aluno ou professor acesse o sistema. Se não houver um cadastro, o componente também facilita o processo de registro. Principais funcionalidades: autenticação de usuários (alunos e professores) e criação de novas contas, se necessário.
- *Home*: após o login, a página Home é apresentada, contendo quatro interfaces distintas para os componentes destacados a seguir:

- a. *Cursos*: permite ao aluno acessar o conteúdo programático e realizar atividade relacionada ao curso.
  - b. *ChatGPT*: fornece acesso ao suporte inteligente do ChatGPT para o aluno, auxiliando na resolução de dúvidas e auxiliando durante a atividade do curso.
  - c. *Dashboard*: permite ao aluno monitorar desempenho no sistema, incluindo estatísticas e progresso.
  - d. *wp-admin*: acesso exclusivo para professores, permitindo a criação e gestão de cursos.
- *Armazenamento de Dados*: as bases de dados das interfaces (Cursos, ChatGPT, Dashboard, wp-admin) são centralizadas em um Sistema de Gerenciamento de Banco de Dados (SGBD) específico do LMS.

O Apêndice I apresenta um manual com os passos para instalação, configuração, customização, bem como para criação de cursos. Na próxima sessão é apresentada a ferramenta em funcionamento, incluindo as principais telas e funcionalidades.

## 5.2. Funcionamento da Ferramenta CodeChatPro

A tela inicial desse LMS desempenha um papel crucial na experiência do usuário, oferecendo acesso às seções essenciais. Conforme apresenta a Figura 8, a página inicial do CodeChatPro foi projetada com um layout minimalista e focando em ser amigável ao usuário, com avatar ao fundo e rápido acesso às opções (parte superior da Figura 8) descritas a seguir:

- A guia "**Home**" serve como ponto de partida, oferecendo uma visão geral e personalizada para cada aluno ao destacar informações relevantes como notificações, progresso recente e recomendações de cursos.
- A guia "**Cursos**" possibilita aos alunos explorar e se inscrever em uma variedade de cursos. A interface é projetada para ser visualmente atraente, exibindo informações essenciais sobre cada curso, visando facilitar a escolha informada por parte do aluno.

- A guia "**Dashboard**" fornece uma visão consolidada do progresso do aluno, estatísticas de desempenho e atividades recentes. Essa abordagem centrada no aluno visa melhorar a autoconsciência e motivar o engajamento contínuo.
- A guia "**Sobre o Projeto**" oferece informações sobre o propósito, objetivos e valores do CodeChatPro. Este recurso visa estabelecer uma conexão entre os usuários e o projeto, bem como promover transparência e confiança.

O LMS foi desenvolvido no WordPress 6.6.2, integrado ao ChatGPT versão GPT-3.5, com o propósito de auxiliar os alunos no desenvolvimento da competência lógica de programação para o curso técnico em informática. Este processo foi guiado pelo modelo ADDIE, abordando as fases específicas de Análise, Design, Desenvolvimento, Implementação e Avaliação para garantir uma abordagem sistemática e eficaz, conforme destacado a seguir:

- *Análise*: foi realizada uma análise das necessidades dos alunos nas lições de lógica de programação e revelou lacunas no entendimento dos conceitos fundamentais. Esse procedimento orientou a integração do ChatGPT, visando fornecer suporte e esclarecimento de dúvidas durante o curso relacionadas ao tema em questão. Além disso, identificou-se a necessidade de uma plataforma intuitiva e de fácil navegação. Essa análise serviu como base para guiar as próximas etapas do ADDIE.
- *Design*: com base nas conclusões da análise, foi planejada uma interface, enfatizando a colaboração do ChatGPT nas lições para otimizar a experiência educacional. A estrutura da página inicial foi planejada para integrar as interfaces de Cursos, ChatGPT e Dashboard, proporcionando uma experiência colaborativa.
- *Desenvolvimento*: na fase de desenvolvimento, especialmente no componente de Cursos, buscou-se incorporar recursos do ChatGPT visando fomentar uma colaboração ativa. O foco está em esclarecer dúvidas e orientar os alunos de maneira eficaz, garantindo uma integração intuitiva do ChatGPT. Essa abordagem visa proporcionar suporte contínuo durante as atividades de programação, conforme delineado no design.

- **Implementação:** na etapa de implementação, buscou-se garantir a disponibilidade do LMS, assegurando uma colaboração contínua do ChatGPT nas lições de lógica de programação. O ChatGPT foi incorporado como uma ferramenta colaborativa essencial, proporcionando suporte inteligente aos alunos, com o intuito de oferecer uma experiência envolvente e significativa.
- **Avaliação:** a integração do ChatGPT para fornecer *feedback* durante a realização dos exercícios complementa nossa abordagem de avaliação, evidenciando uma atenção contínua à análise e ao desempenho dos alunos. Além disso, um canal de diálogo com o professor para esclarecimento de dúvidas. Adicionalmente, os alunos podem monitorar seu próprio desempenho através do *Dashboard* com estatísticas do curso, oferecendo autonomia na gestão do aprendizado.

Espera-se que a abordagem inspirada no ADDIE resulte em um LMS eficiente, integrando de maneira colaborativa ao ChatGPT nas lições de lógica de programação, proporcionando uma experiência educacional mais personalizada e eficaz para os alunos.



**Figura 8** – Página principal do sistema CodeChatPro

O CodeChatPro possibilita aos alunos a inscrição em cursos disponíveis na plataforma. A Figura 9, apresenta a parte superior de uma página de curso. Esta interface foi projetada para ser visualmente atraente, exibindo informações essenciais sobre cada curso. Todas as imagens presentes no sistema, foram geradas por IA.

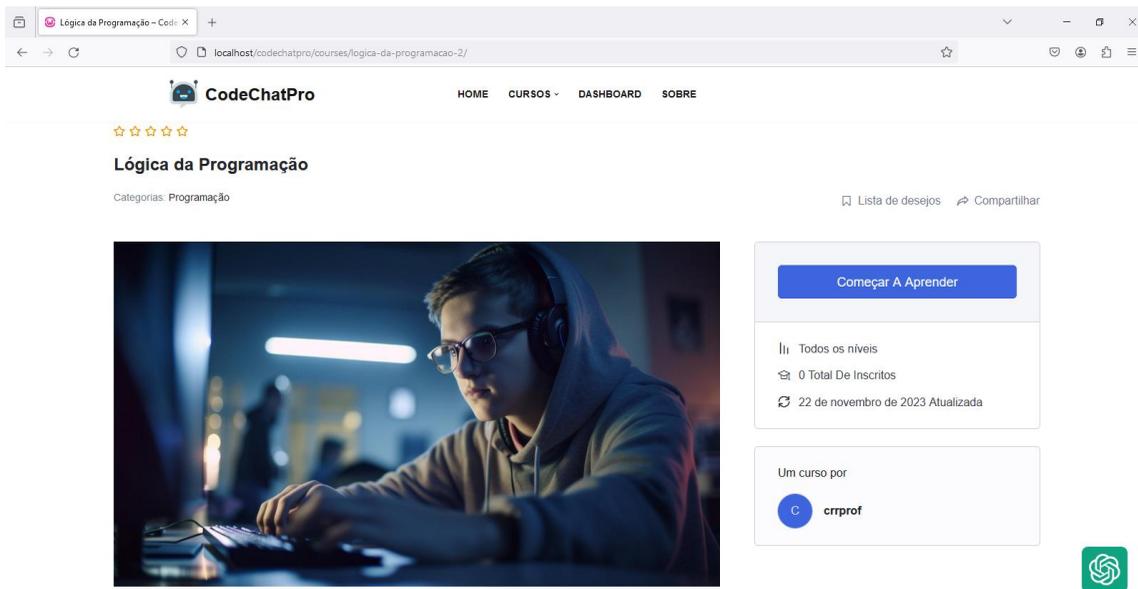


Figura 9 – Exemplo de página de curso no sistema CodeChatPro

A Figura 10 apresenta a parte inferior da página, onde estão disponíveis as abas “informações sobre o curso”, “comentários” e “anúncios”. A aba “informações sobre o curso”, conforme exibida na Figura 10, apresenta o conteúdo do curso, bem como as lições associadas a ele.

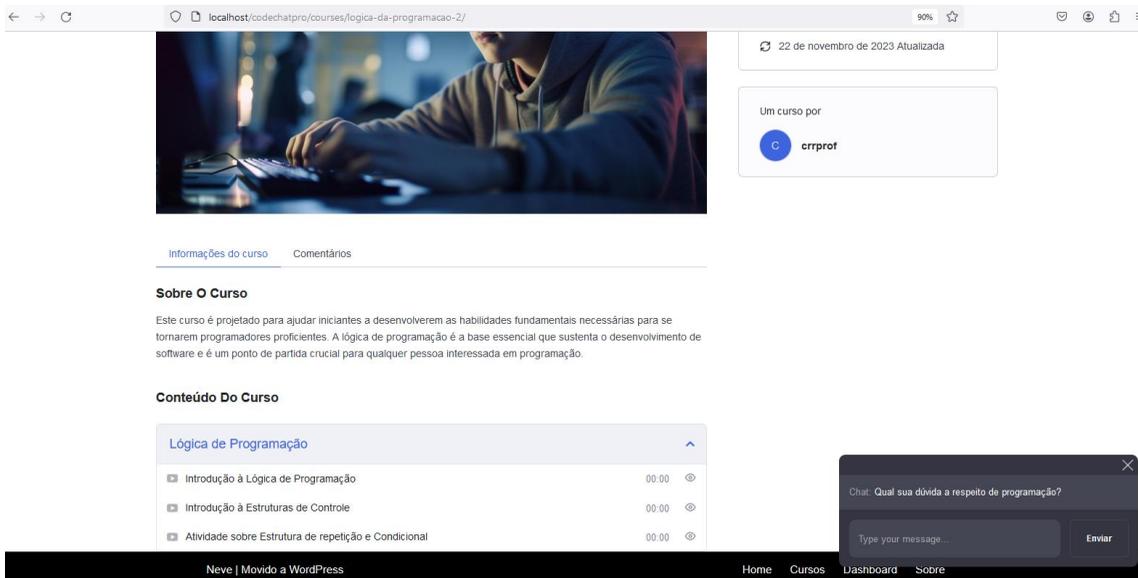


Figura 10 – Exemplo de página com informações sobre um curso no sistema CodeChatPro

A Figura 11 apresenta um exemplo de uma lição sobre “introdução à lógica de programação”, incluindo abas com a visão geral da lição e comentários.

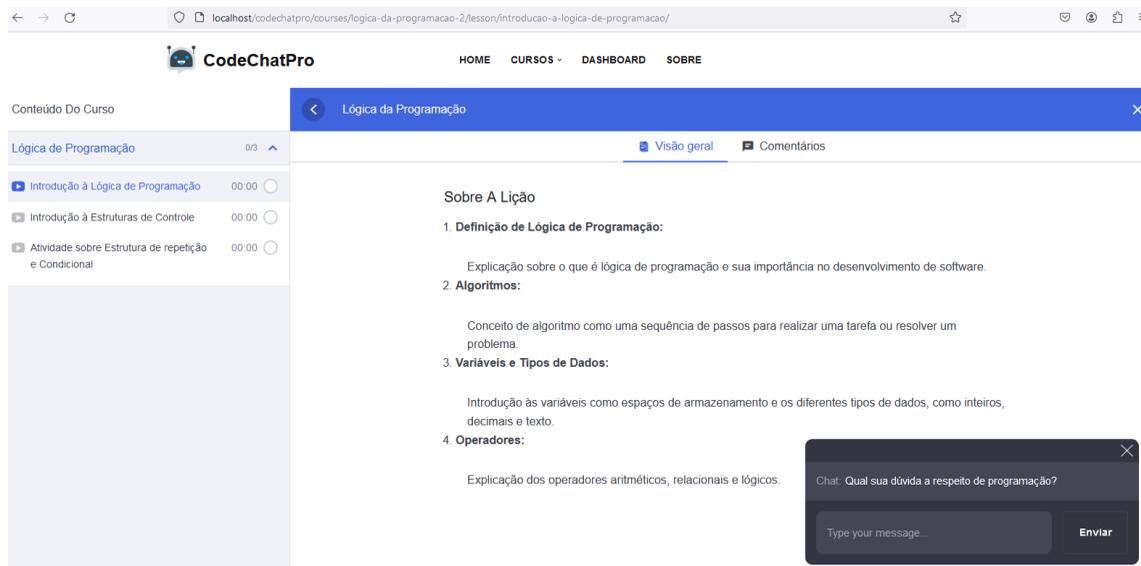


Figura 11 – Exemplo de página com informações sobre um curso no sistema CodeChatPro

A Figura 12 apresenta uma visão geral do dashboard, o qual fornece uma visão consolidada do progresso do aluno, estatísticas de desempenho e atividades recentes. Essa abordagem centrada no aluno visa melhorar a autoconsciência e motivar o engajamento contínuo.

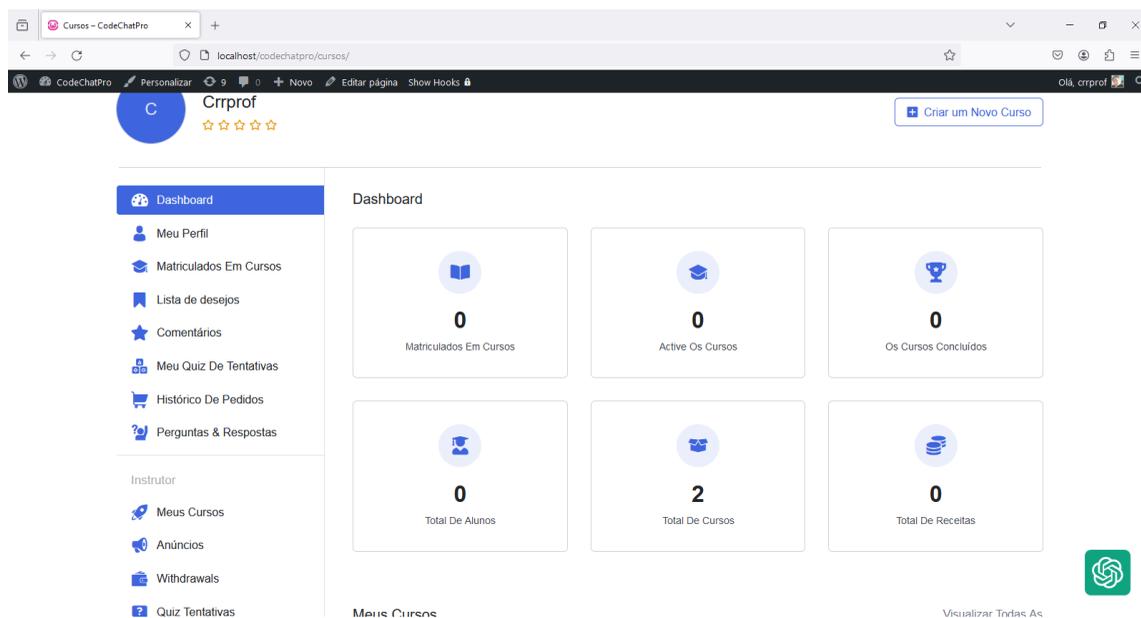
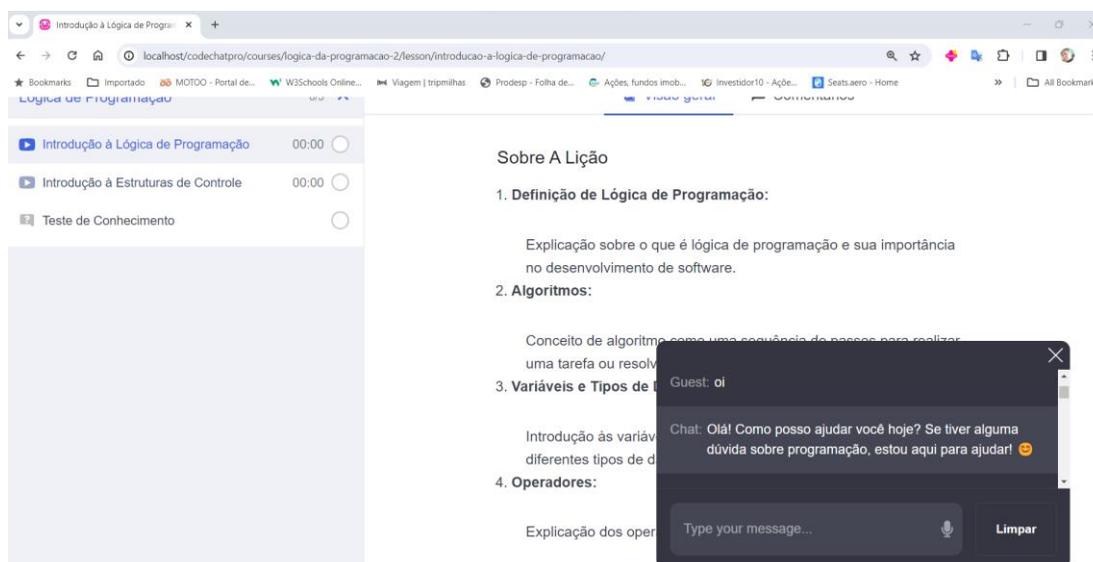


Figura 12 – Visão geral do dashboard do sistema CodeChatPro

A Figura 13 apresenta a lição de “introdução à Lógica de Programação”, onde um aluno utiliza o ChatGPT para superar desafios conceituais, recebendo explicações e exemplos práticos, dando suporte ao aprendizado durante a realização da lição.



**Figura 13** – Exemplo do uso do ChatGPT na realização de exercícios.

O Apêndice II apresenta as instruções destinadas ao professor na elaboração de cursos no sistema, incluindo a adição de módulos, lições, configuração de avaliações e atividades.

## 6. Avaliação da Abordagem e do Protótipo do Sistema CodeChatPro

Este capítulo detalha como a abordagem e o protótipo proposto foram avaliados em atividades práticas com alunos no ensino técnico em Informática. A Seção 6.1 apresenta o local, o perfil dos participantes e a metodologia de avaliação; a Seção 6.2 apresenta os resultados da avaliação; e a Seção 6.3 realiza uma análise e discute os resultados obtidos.

### 6.1. Local, Participantes e Metodologia de Avaliação

Esta pesquisa planeja investigar e analisar a aceitação do LMS desenvolvido no WordPress, incorporando o ChatGPT, como ferramenta de apoio no processo de ensino/aprendizagem de programação no âmbito de um curso técnico em Informática. O foco avaliou como essa tecnologia foi aceita pelos estudantes, o apoio fornecido e as possíveis melhorias no desempenho em testes.

A pesquisa foi realizada no Senac Mogi Guaçu, com termos de consentimento assinados pelos participantes (todos maiores) e aprovados pelo comitê da instituição. Participaram do estudo 20 alunos do curso técnico em Informática, matriculados na unidade curricular de desenvolvimento de algoritmos, com idades entre 18 e 50 anos. Os alunos foram divididos aleatoriamente em dois grupos: 10 participantes formaram o Grupo de Controle (GC) e 10 participantes compuseram o Grupo Experimental (GE).

O seguinte procedimento foi adotado:

- **Pré-teste:** foi conduzida uma avaliação preliminar sobre a linguagem C#, com 10 perguntas básicas para determinar a habilidade prévia dos participantes na realização de exercícios sobre programação. O formulário de avaliação preliminar é apresentado no Apêndice IV. O objetivo é conhecer os grupos, bem como detectar possíveis diferenças entre eles.
- **Instrumentos e materiais:** incluem conteúdo planejado conforme metodologia ADDIE, sistema CodeChatPro e acesso à Internet. O conteúdo (que inclui exercícios) se refere à lógica da programação básica, tais como estrutura sequencial, decisão e repetição.

- **Distribuição dos exercícios:** durante esta fase, os grupos GC e GE foram criados. Para o GC os exercícios foram recebidos via e-mail, já para o GE os exercícios foram previamente cadastrados e acessados via o CodeChatPro.
- **Execução:** os participantes realizaram as atividades em dois dias com 4 aulas cada (total de 8 horas). Após o pré-teste, o conteúdo foi distribuído. O GC podia pedir apoio aos colegas, utilizar material didático e consultar a Web. O GE poderia pedir apoio ao ChatGPT integrado ao sistema em caso de dúvidas. Após realizar a atividade, foi executado o pós-teste.
- **Pós-teste:** foi conduzida uma nova avaliação sobre a linguagem C#, com 10 perguntas para determinar o desempenho dos dois grupos no tópico abordado, conforme apresentado no Apêndice IV.
- **Métodos de coleta de dados:** os métodos de coleta de dados incluem formulários de avaliação quantitativa e qualitativa, conforme apresentado no Apêndice III.
- **Local:** a atividade foi realizada presencialmente em um laboratório com 24 PCs, onde 10 alunos do GC utilizaram o lado A e os alunos do GE utilizaram o lado B.

Para medir a aceitação da tecnologia pelos alunos, foi aplicado um formulário baseado no TAM (*Technology Acceptance Model*). Esse modelo planeja avaliar a aceitação no uso de novas tecnologias. O modelo é baseado na ideia de que a atitude do usuário em relação a uma tecnologia influencia significativamente na sua intenção de usá-la (Silva et al., 2012). O TAM possui os seguintes componentes: *utilidade percebida*, que se refere ao grau em que uma pessoa acredita que o uso de uma determinada tecnologia aprimoraria seu desempenho; *facilidade de uso percebida* que se refere à extensão em que, uma pessoa acredita que usar uma tecnologia, seria livre de esforço ou fácil de usar; *intenção comportamental de uso* que, segundo o TAM, os indivíduos são mais propensos a adotar uma tecnologia se a perceberem como útil e fácil de usar; e, *uso real do sistema* que se refere à extensão em que, o indivíduo realmente usa uma tecnologia em ambientes reais.

Para integrar o TAM no contexto da investigação e análise da efetividade do LMS, incorporando o ChatGPT como ferramenta de apoio no processo de ensino-aprendizagem de programação, os seguintes passos serão seguidos:

1. *Identificação das variáveis*: foram determinadas as variáveis-chave sobre o LMS integrado com ChatGPT, incluindo a facilidade de uso percebida (facilidade de uso), a intenção comportamental do aluno em usar essas ferramentas (intenção de uso) e o uso real delas (uso real).
2. *Desenvolvimento de instrumentos de coleta de dados*: foi elaborado um formulário (Apêndice III) para coletar dados sobre as percepções dos alunos em relação à utilidade percebida e à facilidade de uso tanto do LMS quanto do ChatGPT, bem como sua intenção de usar essa ferramenta.
3. *Coleta de dados*: foi realizada a coleta de dados com os alunos participantes do curso técnico em informática, solicitando que eles avaliassem o LMS integrado com o ChatGPT com base nas variáveis identificadas anteriormente.
4. *Análise dos dados*: por meio de uma análise estatística, foi verificado se a utilidade percebida e a facilidade de uso tanto do LMS quanto do ChatGPT foram positivamente relacionadas à intenção de uso e ao uso real dessas tecnologias.
5. *Interpretação dos resultados e recomendações*: os resultados obtidos visam verificar a percepção dos alunos em relação à realização de exercícios práticos e ao seu aprendizado de programação. Com base nessa análise, será possível fazer recomendações para aprimorar o design da ferramenta.

## **6.2. Resultados da avaliação**

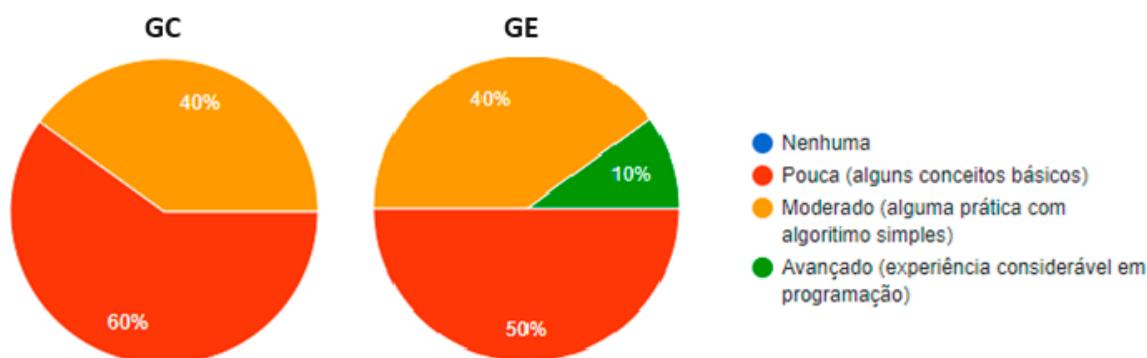
Os resultados foram analisados por métodos quantitativos e qualitativos, buscando investigar o impacto da integração do ChatGPT com o LMS, incluindo a sua aceitação e mudanças no desempenho dos alunos nos testes. Primeiramente, a Figura 14

apresenta um conjunto de fotos que documentam a avaliação realizada, bem como apresenta detalhes do ambiente no qual ela foi realizada.



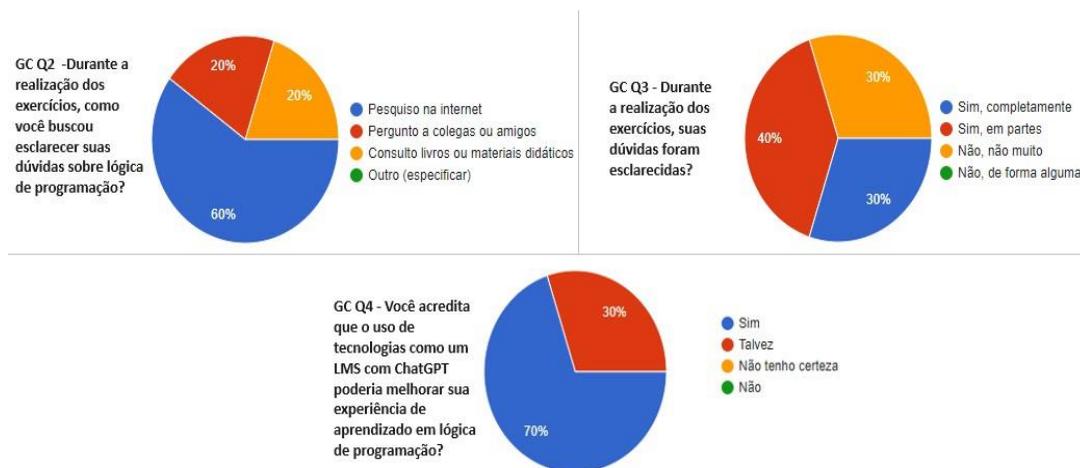
**Figura 14** – GC e GE realizando a avaliação

A Figura 15 apresenta que 60% dos alunos do GC tinham conhecimento prévio limitado e 40% tinham nível moderado de conhecimento. Enquanto para o GE temos que 50% indicaram conhecimento prévio limitado, 40% indicaram um nível moderado e 10% indicaram conhecimento avançado. Isso sugere que muitos alunos poderiam se beneficiar de suporte adicional durante os exercícios.



**Figura 15** - Nível de conhecimento prévio em Lógica da Programação

A Figura 16 apresenta as respostas para questões 2 a 4 endereçadas ao GC. A questão 2 diz respeito à onde eles buscaram ajuda em caso de dúvidas, sendo que 60% buscaram informações na web; 20% com colegas e 20% em livros. A questão 3 diz respeito à efetividade da ajuda obtida, sendo que 40% disseram que as dúvidas foram esclarecidas parcialmente, 30% disseram que não foram muito bem esclarecidas e apenas 30% tiveram as dúvidas completamente esclarecidas. Já a questão 4 considerou a expectativa do GC (que não usou o sistema) perante o uso do ChatGPT para auxílio às atividades, sendo que 70% achavam que ele poderia ajudar, enquanto 30% acham que existe a possibilidade de ajuda (talvez). Os resultados destacam uma clara dependência no uso da internet para resolver problemas e esclarecer dúvidas. Isso também evidencia um certo grau de autonomia no aprendizado. Os dados mostram que a estratégia de utilizar diretamente a internet é limitada para resolver completamente as dúvidas dos alunos devido à falta de objetividade nas pesquisas. Isso indica a necessidade de um apoio mais estruturado e acessível para aprimorar a experiência de aprendizado dos alunos.



**Figura 16** - Respostas do grupo de controle às questões 2 a 4

A Figura 17 apresenta as respostas para as questões 2 a 5 endereçadas ao GE, sendo estas baseadas no TAM. A questão 2 diz respeito à efetividade da ajuda obtida por meio do sistema, sendo que 60% foram completamente resolvidas, 30% em partes e apenas 10% disseram que não foram muito bem esclarecidas.

A questão 3 diz respeito à percepção dos participantes quanto à capacidade do sistema em melhorar o desempenho em atividades de lógica de programação, sendo que 60% disseram que houve melhoras e 30% disseram que essas melhoras foram parciais. A questão 4 considerou o quão intuitivo foi o uso do ChatGPT no sistema, sendo que (dos 9 que responderam) 77,8% consideraram intuitivo e 22,2% consideraram intuitivo “em partes”. A questão 5 diz respeito ao quão propensos estão os participantes a usar o ChatGPT durante as aulas de programação (após a experimentação), dos quais 90% responderam que usariam e 10% responderam que não.

Assim, a maioria dos alunos vê benefícios no uso de tecnologias como um LMS com ChatGPT para melhorar sua aprendizagem em lógica de programação, indicando uma aceitação positiva de avanços tecnológicos na educação. Considerando o experimento realizado, os resultados sugerem que o ChatGPT proporcionou um apoio mais eficiente e eficaz, oferecendo respostas mais satisfatórias e imediatas, e reduzindo a quantidade de questões não completamente respondidas.



**Figura 17** - Respostas do grupo experimental às questões 2 a 5

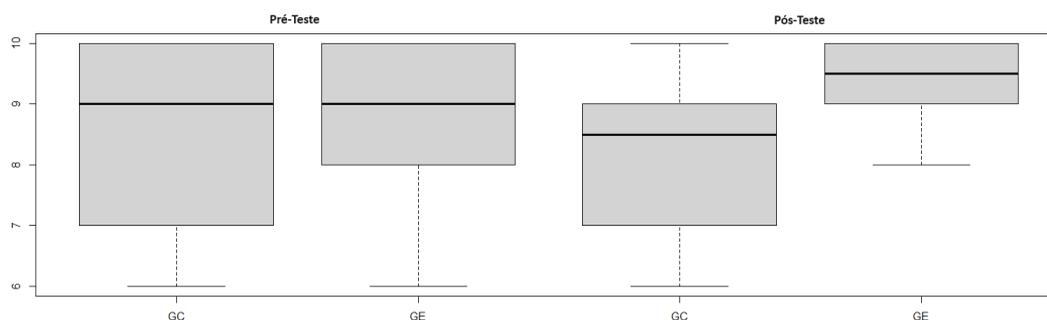
O tamanho da amostra de 20 alunos foi determinado com base na disponibilidade dos participantes do curso técnico em informática. Embora uma amostra maior pudesse fornecer mais robustez às conclusões em populações maiores, a escolha de 20 alunos é adequada para a análise e permite obter *insights* sobre o impacto com base na avaliação na população restrita aos alunos do curso técnico em Informática do Senac Mogi Guaçu.

O objetivo foi testar a significância em relação aos resultados numéricos e contínuos variando de 0 a 10 (Figura 18) obtidos pelos alunos no pré-teste e pós-teste (Apêndice IV). Para tanto, foi utilizado o Teste de Wilcoxon Rank Sum, também conhecido como Teste de Mann-Whitney U ou Teste de Wilcoxon–Mann–Whitney. Este é um teste não paramétrico que serve para analisar hipóteses sobre duas amostras independentes. O Teste de Wilcoxon Rank-Sum é particularmente útil em situações com amostras menores.

O nível de significância de 0,05 (ou seja, 95% de confiança) foi adotado para as seguintes hipóteses: *Pré-teste*:  $H_{0pre}$  não há diferença de pontuação entre os participantes dos GC e GE no pré-teste;  $H_{apre}$ : há diferença de pontuação entre os participantes dos GC e GE no pré- teste. *Pós-Teste*:  $H_{0pos}$  não há diferença de pontuação entre os participantes dos GC e GE no pós-teste;  $H_{apos}$ : há diferença de pontuação entre os participantes dos GC e GE no pós-teste.

Ao considerar os resultados do teste de Wilcoxon Rank Sum para o pré-teste, temos  $p$ -valor = 0.5536, ou seja,  $p > 0, 05$ . Portanto, aceitamos a hipótese nula  $H_{0pre}$  e

concluimos que não há diferença de pontuação entre os GC e GE para o pré-teste. Já ao considerar o pós-teste, temos  $p - \text{valor} = 0.01919$ , ou seja,  $p < 0, 05$ . Portanto, rejeitamos a hipótese nula  $H0_{\text{pos}}$  e concluimos que há diferença de pontuação entre os GC e GE para o pós-teste. A Figura 18 apresenta diagramas Box Plot para pontuação do GC e GE no pré-teste, à esquerda, e para pontuação do GC e GE no pós-teste, à direita.



**Figure 18.** Boxplots com pontuação do GC e GE no pré-teste e pós-teste.

Os dois grupos responderam questões dissertativas sobre suas experiências nas atividades. As respostas são apresentadas na Figura 7.

Nas respostas às questões dissertativas, os participantes do GC destacaram a Web como fonte de acesso a uma vasta gama de recursos, mas também que essa quantidade pode ser uma fonte de distração; bem como o sentimento de estar sobrecarregado e a frustração com problemas de conexão. Os participantes do GE destacaram positivamente as respostas instantâneas, deixando a aprendizagem mais dinâmica e interativa, enquanto outros alunos expressaram frustração perante as expectativas e a dificuldade em obter respostas para as dúvidas específicas.

GC Você poderia compartilhar sua experiência com as ferramentas que usou.	GE Você poderia compartilhar sua experiência sobre o uso do LMS com ChatGPT?
5 respostas	8 respostas
A conexão instável ou lenta com a internet era uma fonte constante de frustração.	A experiência com o LMS com ChatGPT foi muito positiva para mim. Eu me senti mais apoiado e confiante.
A internet me proporcionou acesso a uma variedade de recursos de aprendizado	Às vezes eu sentia que suas respostas eram genéricas demais e não abordavam minhas dúvidas específicas.
Embora a internet ofereça acesso a uma vasta gama de tutoriais e recursos de aprendizado, muitas vezes eu me sentia perdido	Sempre que eu estava presa em um exercício, bastava digitar minha pergunta e obter uma resposta útil em segundos
A dependência da internet como minha principal fonte de aprendizado me deixava vulnerável a distrações online.	A experiência com o LMS com ChatGPT não foi tão boa quanto eu esperava
me senti sobrecarregado pela quantidade de conteúdo disponível,	A ferramenta tornou a aprendizagem mais interativa e envolvente
	Pude receber respostas instantâneas para minhas perguntas, o que acelerou meu aprendizado.*
	tornou as atividades de lógica de programação mais dinâmicas
	Minha experiência com o LMS com ChatGPT foi incrível! A ferramenta tornou a aprendizagem mais interativa

**Figura 19** - Respostas dos grupos GC e GE sobre a experiência com o uso da ferramenta de apoio

### 6.3. Análise e discussão dos Resultados

Os alunos do GC possuíam um conhecimento prévio básico em lógica de programação, com pouca prática, além de conceitos fundamentais. Em relação ao esclarecimento de dúvidas, a maioria não teve suas questões completamente respondidas durante os exercícios. Apesar dos desafios relacionados à dependência da Internet, excesso de informações e problemas de conexão, muitos demonstraram interesse em testar novas tecnologias, como um LMS com ChatGPT, para melhorar sua experiência de aprendizado.

Por outro lado, os alunos do GE, que utilizaram o LMS integrado ao ChatGPT, avaliaram positivamente a ferramenta, destacando o suporte eficaz para esclarecer dúvidas durante o aprendizado. A personalização das respostas do ChatGPT foi identificada como uma área importante para melhorias, visando atender de forma mais precisa às necessidades individuais dos alunos.

Destaca-se que, ao analisar a experiência no desenvolvimento do protótipo do sistema CodeChatPro, a tecnologia de chats inteligentes já está suficientemente madura e passível de integração com LMS, possibilitando apoio instantâneo aos alunos. Foi possível integrá-lo via ícone permanente no canto inferior esquerdo que aciona uma janela flutuante com resposta suficientemente rápida ao prompt inicial personalizado sobre o conteúdo apresentado ao aluno. É possível explorar outras maneiras de utilizar

chats inteligentes no apoio tanto ao aluno quanto ao professor, tais como geração automática de exercícios, correção de códigos e geração de questões.

Este estudo corrobora com o estudo publicado no artigo em 2023 por (Silva Jr. et al., 2023) sobre ChatGPT no auxílio da aprendizagem de programação: "Os resultados obtidos mostram que o ChatGPT é uma tecnologia promissora para auxiliar os alunos na aquisição de conhecimentos e habilidades de programação, fornecendo explicação de conceitos, exemplos de código, suporte e feedback personalizado aos estudantes durante seu aprendizado." A análise atual reforça essa afirmação ao mostrar uma evolução no desempenho em comparação ao estudo de 2023. No cenário anterior, foi feita uma avaliação do uso do ChatGPT isoladamente; neste estudo, propôs-se um LMS integrado com ChatGPT, no qual a familiaridade e o aprimoramento do ChatGPT de 2023 para 2024 são mais evidentes, resultando em uma experiência de aprendizagem mais eficaz e eficiente.

Em suma, o uso de tecnologias como o LMS com ChatGPT demonstra um grande potencial para enriquecer o ensino de lógica de programação, proporcionando um aprendizado mais dinâmico, interativo e eficiente, capaz de fornecer respostas rápidas e precisas às questões dos alunos. Os resultados desta pesquisa são promissores e indicam que a abordagem baseada em chats inteligentes conseguiu produzir um sistema com potencial no ensino de programação.

## 7. Conclusão

Este capítulo apresenta as considerações finais (Seção 7.1), bem como as limitações e os trabalhos futuros relacionados à abordagem e ao protótipo proposto (Seção 7.2).

### 7.1. Contribuições e Considerações Finais

O ensino de programação continua a ser um desafio tanto para alunos quanto para professores. Embora chats inteligentes sejam ferramentas poderosas para apoiar atividades de programação, eles são frequentemente vistos como uma ameaça ao ensino, por poderem fornecer respostas prontas para problemas complexos que antes exigiam raciocínio lógico e habilidades de programação. Contrariando a tendência inicial de proibir seu uso no desenvolvimento de atividades de ensino de programação, esta dissertação explora como eles podem se tornar uma ferramenta útil, não apenas para o aprendizado de conceitos de programação, mas também para futuras atividades profissionais a serem desenvolvidas por alunos do ensino técnico em informática.

Para responder à seguinte questão de pesquisa: *Como os sistemas de chats inteligentes podem ser utilizados no ensino de programação e qual a sua aceitação e influência no desempenho dos estudantes de programação?* Esta dissertação investigou o uso de Chats Inteligentes em uma abordagem que integra esta tecnologia com um LMS construído com base em um método de design instrucional. Em termos concretos, o sistema CodeChatPro foi construído explorando o uso do ChatGPT 3.5, integrado a um LMS desenvolvido utilizando uma extensão do WordPress, em um processo de design e inclusão de cursos baseado no modelo ADDIE. Nossa solução foi avaliada experimentalmente com 20 alunos do curso técnico, organizados em um grupo de controle e outro de experimentação. Os resultados indicaram uma boa aceitação e uso dos chats inteligentes pelo GE em comparação aos participantes do GC, que utilizaram outras alternativas para buscar respostas às suas dúvidas (ex: Web, colegas e material didático). Destacamos que os participantes do GE obtiveram melhores notas no pós-teste do que os participantes do GC.

Os resultados desta pesquisa indicam que a integração do ChatGPT ao LMS propiciou a construção de ambientes com potencial de aprimorar a experiência de

aprendizado dos alunos, oferecendo um suporte adaptativo e instantâneo que pode ajudá-los a superar desafios conceituais e a resolver problemas de programação de forma mais eficiente. Embora existam algumas limitações e desafios a serem superados, a perspectiva geral é positiva, e espera-se que esta pesquisa contribua significativamente para o avanço da computação na educação.

Em suma, o uso de tecnologias como o LMS com ChatGPT apresenta um grande potencial para enriquecer o ensino de lógica de programação, proporcionando um aprendizado mais dinâmico, interativo e eficiente, capaz de fornecer respostas rápidas e precisas às questões dos alunos.

## **7.2. Limitações e Trabalhos futuros**

Embora tenha apresentado resultados promissores, vale ressaltar que o estudo apresentado é limitado em termos de quantidade de participantes (20), tempo de estudo (8H) e quantidade de questões (20). Portanto, não é possível ter conclusões definitivas e com validade estatística para uma grande população. Contudo, este fato não invalida as lições aprendidas no estudo. Salienta-se que os resultados apresentam indícios muito positivos da abordagem, que devem ser mais bem estudados em trabalhos futuros ao longo do tempo e em larga escala.

Os próximos passos desta pesquisa incluem o aprimoramento da integração do Chat Inteligente com o sistema CodeChatPro, visando identificar as dificuldades dos alunos na execução dos exercícios e, assim, prover um auxílio mais direcionado quando solicitado e personalizado sempre que possível. É importante considerar a adaptação contínua do sistema às necessidades dos alunos, utilizando técnicas de aprendizado de máquina para melhorar a precisão das respostas e a relevância do suporte fornecido. Além disso, explorar a incorporação de novas funcionalidades, como a análise de progresso individual e a geração de relatórios personalizados, pode contribuir para um acompanhamento mais detalhado e eficaz do desenvolvimento dos alunos. Pretende-se também explorar novas versões do ChatGPT, e outras tecnologias que não estavam

disponíveis no momento da execução deste trabalho como o MS Copilot<sup>21</sup> e o LLaMA<sup>22</sup>, e modelos adaptados à língua portuguesa como o Maritaca AI<sup>23</sup> e o Amazônia IA<sup>24</sup>.

Por fim, a colaboração com outras instituições de ensino para validar e aprimorar o sistema em diferentes contextos educacionais será essencial para garantir a robustez e a aplicabilidade da abordagem em um cenário mais amplo. A pesquisa futura deve também considerar a inclusão de métricas de satisfação e engajamento dos alunos, além de avaliar o impacto da ferramenta na motivação e no interesse dos estudantes pela lógica de programação.

---

<sup>21</sup> <https://copilot.microsoft.com/>

<sup>22</sup> <https://www.llama.com/>

<sup>23</sup> <https://www.maritaca.ai/>

<sup>24</sup> <https://amazoniaia.com.br/>

## Referências

- Ahmed, I., et al. (2023). *ChatGPT versus Bard: A comparative study*. *Engineering Reports*, e12890. <https://doi.org/10.1002/eng2.12890>
- Ai, R., et al. (2020). *Programming an educational chatbot to support virtual learning*. In *2020 IEEE MIT Undergraduate Research Technology Conference (URTC)* (pp. 1–5). <https://doi.org/10.1109/urtc51696.2020.9668855>
- Alam, A. (2021). *Should robots replace teachers? Mobilisation of AI and learning analytics in education*. In *Proceedings of the 2021 International Conference on Advances in Computing, Communication, and Control* (pp. 1–12). IEEE. <https://doi.org/10.1109/icac353642.2021.9697300>
- Alaoui, H. E., et al. (2023). *Building intelligent chatbots: Tools, technologies, and approaches*. In *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology* (pp. 1–12). <https://doi.org/10.1109/iraset57153.2023.10153005>
- Ashfaq, M. W., et al. (2020). *A review on techniques, characteristics, and approaches of an intelligent tutoring chatbot system*. In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing* (pp. 258–262). <https://doi.org/10.1109/icsidempc49020.2020.9299583>
- Ausubel, D. P. (1982) *A aprendizagem significativa: a teoria de David Ausubel*. São Paulo: Moraes, 1982.
- Babar, M. A., & Zhang, H. (2009). *Systematic literature reviews in software engineering: Preliminary results from interviews with researchers*. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. <https://doi.org/10.1109/esem.2009.5314235>
- Banić, B., et al. (2023). *Pair programming education aided by ChatGPT*. In *2023 46th MIPRO ICT and Electronics Convention* (pp. 911–915). <https://doi.org/10.23919/mipro57284.2023.10159727>

- Beganovic, A., et al. (2023). *Methods and Applications of ChatGPT in Software Development: A Literature Review*. *Southeast Europe Journal of Soft Computing*, 12(1), 08–12. <https://doi.org/10.21533/scjournal.v12i1.251>
- Carreira, G., et al. (2022). *Pyo, a chatbot assistant for introductory programming students*. In *2022 International Symposium on Computers in Education* (pp. 1–6). <https://doi.org/10.1109/siie56031.2022.9982349>
- Chempavathy, B., et al. (2022). *AI based Chatbots using Deep Neural Networks in Education*. In *2022 Second International Conference on Artificial Intelligence and Smart Energy* (pp. 124–130). <https://doi.org/10.1109/icaais53314.2022.9742771>
- Chen, M., et al. (2021). *Evaluating large language models trained on code*. <https://doi.org/10.48550/arxiv.2107.03374>
- Falcade, A. (2015). *Design instrucional aplicado ao mundo virtual TCN5*. Universidade Federal de Santa Maria. <http://repositorio.ufsm.br/handle/1/15406>
- Farias, G. B. de. (2022). *Contributos da aprendizagem significativa de David Ausubel para o desenvolvimento da Competência em Informação*. *Perspectivas Em Ciência Da Informação*, 27, 58–76. <https://doi.org/10.1590/1981-5344/39999>
- Finnie-Ansley, J., et al. (2022). *The robots are coming: Exploring the implications of OpenAI Codex on introductory programming*. In *Proceedings of the 24th Australasian Computing Education Conference* (pp. 10–19). ACM. <https://doi.org/10.1145/3511861.3511863>
- Ganesan, M., et al. (2020). *A survey on chatbots using artificial intelligence*. In *2020 International Conference on System, Computation, Automation and Networking* (pp. 1–5). ieee. <https://doi.org/10.1109/icscan49426.2020.9262366>
- Gonda, D. E., & Chu, B. (2019). *Chatbot as a learning resource? Creating conversational bots as a supplement for teaching assistant training course*. In *2019 IEEE International Conference on Engineering, Technology and Education* (pp. 1–5). <https://doi.org/10.1109/tale48000.2019.9225974>
- Henno, J., et al. (2023). *What do we know about learning – Conversations with ChatGPT*. In *2023 46th MIPRO ICT and Electronics Convention* (pp. 572–577). <https://doi.org/10.23919/mipro57284.2023.10159818>

- Jalil, S., et al. (2023). *ChatGPT and software testing education: Promises & perils*.  
<https://doi.org/10.48550/arxiv.2302.03287>
- Ji, S., & Yuan, T. (2022). *Conversational intelligent tutoring systems for online learning: What do students and tutors say?* In *2022 IEEE Global Engineering Education Conference* (pp. 292–298).  
<https://doi.org/10.1109/educon52537.2022.9766567>
- Karri Reddy, S. P., & Santhosh Kumar, B. (2020). *Deep learning techniques for implementation of chatbots*. In *Proceedings of the 2020 International Conference on Computer Communication and Informatics* (pp. 1–5). IEEE.  
<https://doi.org/10.1109/iccci48352.2020.9104143>
- Kita, T., et al. (2020). *Development of a Moodle UI using LINE Chat for casual learning as a part of a learner assistive LMS*. In *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering* (pp. 927–929).  
<https://doi.org/10.1109/tale48869.2020.9368321>
- Koundinya, H. K., et al. (2020). *Smart College Chatbot using ML and Python*. In *2020 International Conference on System, Computation, Automation and Networking* (pp. 1–5). <https://doi.org/10.1109/icscan49426.2020.9262426>
- Kumar, M., et al. (2021). *Analysis of automated text generation using deep learning*. In *2021 Fourth International Conference on Computational Intelligence and Communication Technologies* (pp. 14–18).  
<https://doi.org/10.1109/ccict53244.2021.00014>
- Kumarasiri, A. D. S. S., et al. (2022). *Student teaching and learning system for academic institutions*. In *Proceedings of the 2022 IEEE 7th International Conference for Convergence in Technology* (pp. 1–6).  
<https://doi.org/10.1109/i2ct54291.2022.9824437>
- Lajko, M., et al. (2022). *Towards JavaScript program repair with generative pre-trained transformer (GPT-2)*. In *Proceedings of the Third International Workshop on Automated Program Repair* (pp. 61–68). ACM.  
<https://doi.org/10.1145/3524459.3527350>

- Lin, Y.-H. (2022). *Chatbot script design for programming language learning*. In *Proceedings of the 2022 IEEE 5th Eurasian Conference on Educational Innovation* (pp. 123–125). <https://doi.org/10.1109/ecei53102.2022.9829460>
- Lin, Y.-H., & Tsai, T. (2019). *A conversational assistant on mobile devices for primitive learners of computer programming*. In *2019 IEEE International Conference on Engineering, Technology and Education* (pp. 1–4). <https://doi.org/10.1109/tale48000.2019.9226015>
- Luo, C. J., & Gonda, D. E. (2019). *Code free bot: An easy way to jumpstart your chatbot!* In *2019 IEEE International Conference on Engineering, Technology and Education* (pp. 1–3). <https://doi.org/10.1109/tale48000.2019.9226016>
- Malinka, K., et al. (2023). *On the educational impact of ChatGPT: Is artificial intelligence ready to obtain a university degree?* In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V* (Vol. 1, pp. 47–53). ACM. <https://doi.org/10.1145/3587102.3588827>
- Mallikarjuna Gowda, P. C., et al. (2021). *Development of Information Technology Telecom Chatbot: An Artificial Intelligence and Machine Learning Approach*. In *2021 2nd International Conference on Intelligent Engineering and Management* (pp. 216–221). <https://doi.org/10.1109/iciem51511.2021.9445354>
- Maquiné, G. O. (2020). *Recursos para avaliação da aprendizagem: Estudo comparativo entre ambientes virtuais de aprendizagem*. In *Anais do XXVI Workshop de Informática na Escola (WIE 2020)* (pp. 299–308). Sociedade Brasileira de Computação - SBC. <https://doi.org/10.5753/cbie.wie.2020.299>
- Martenstyaro, R., & Rosmansyah, Y. (2015). *A framework for designing survey training based on 3D Virtual Learning Environment using SLOODLE*. In *2015 International Conference on Information Technology Systems and Innovation* (pp. 1–6). <https://doi.org/10.1109/icitsi.2015.7437740>
- Murad, D. F., et al. (2019). *Learning support system using chatbot in 'Kejar C Package' homeschooling program*. In *2019 International Conference on Information and Communications Technology* (pp. 32–37). <https://doi.org/10.1109/icoiact46704.2019.8938479>

- OpenAI, et al. (2023). *GPT-4 technical report*.  
<https://doi.org/10.48550/arxiv.2303.08774>
- Prenner, J. A., et al. (2022). *Can OpenAI's Codex fix bugs?: An evaluation on QuixBugs*. In *Proceedings of the Third International Workshop on Automated Program Repair* (pp. 69–75). ACM. <https://doi.org/10.1145/3524459.3527351>
- Rahaman, M. S., et al. (2023). *The AI race is on! Google's Bard and OpenAI's ChatGPT head to head: An opinion article*. *SSRN Electronic Journal*.  
<https://doi.org/10.2139/ssrn.4351785>
- Rakesh, N., et al. (2022). *A Proposed Academic Chatbot System using NLP Techniques*. In *2022 6th International Conference on Trends in Electronics and Informatics* (pp. 1300–1304). <https://doi.org/10.1109/icoei53556.2022.9777231>
- Ross, S. I., et al. (2023). *The Programmer's Assistant: Conversational Interaction with a Large Language Model for Software Development*. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (pp. 491–514). ACM. <https://doi.org/10.1145/3581641.3584037>
- Santhana Lakshmi, V. (2020). *A study on machine learning based conversational agents and designing techniques*. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)* (pp. 965–968). <https://doi.org/10.1109/i-smac49090.2020.9243577>
- Santos, V. M. R., et al. (2021). *Benchmark application for scenario analysis in the educational chatbots development*. In *2021 XVI Latin American Conference on Learning Technologies* (pp. 302–309). <https://doi.org/10.1109/laclo54177.2021.00038>
- Silva Junior, S. M., Freitas, R. A. B., Morais, M. A. C., & Costa, D. L. V. (2023). *ChatGPT no auxílio da aprendizagem de programação: Um estudo de caso*. *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*. <https://sol.sbc.org.br/index.php/sbie/article/view/26762/26581>.
- Silva, P., Pimentel, V., & Soares, J. (2012). *A utilização do computador na educação: Aplicando o Technology Acceptance Model (TAM)*. *Biblioteca Virtual em Ciências*

<https://periodicos.ufpb.br/ojs/index.php/biblio/article/view/14208>

- Sophia, J. J., & Jacob, T. P. (2021). *EDUBOT-A chatbot for education in Covid-19 pandemic and VQAbot comparison*. In *2021 Second International Conference on Electronics and Sustainable Communication Systems* (pp. 1707–1714). <https://doi.org/10.1109/icesc51422.2021.9532611>
- Trilaksono, K., & Santoso, H. B. (2017). *Moodle based learning management system development for kinesthetic learning style*. In *2017 7th World Engineering Education Forum* (pp. 602–606). IEEE. <https://doi.org/10.1109/weef.2017.8467180>
- Vazhayil, A., et al. (2019). *Focusing on teacher education to introduce AI in schools: Perspectives and illustrative findings*. In *Proceedings of the 2019 IEEE Tenth International Conference on Technology for Education* (pp. 71–77). <https://doi.org/10.1109/t4e.2019.00021>
- Wang, J., et al. (2023). *Parallel intelligent education with ChatGPT*. *Frontiers of Information Technology & Electronic Engineering*, 25(1), 12–18. <https://doi.org/10.1631/fitee.2300166>
- Wijastuti, A., et al. (2019). *Design science education for students with special needs using learning management system platform Moodle*. In *2019 5th International Conference on Education and Technology* (pp. 94–97). <https://doi.org/10.1109/icet48172.2019.8987207>

## **Apêndice I: Detalhamento e instruções para instalação e configuração do sistema CodeChatPro**

**Configuração do Ambiente para o LMS.** Na fase inicial do processo de construção do LMS (Sistema de Gestão de Aprendizado), é crucial criar um ambiente propício para a instalação e operação do sistema.

1. Download da Versão Mais Recente do WordPress
  - 1.1. Acesse o site oficial do WordPress em [wordpress.org](http://wordpress.org).
  - 1.2. Procure pela opção "Baixar WordPress" e clique para obter o arquivo zip contendo a versão mais recente.
2. Instalação do WordPress em um Servidor Web Local.
  - 2.1. Extraia o conteúdo do arquivo zip baixado. Mova a pasta extraída para o diretório raiz do seu servidor local (por exemplo, htdocs para Apache).
  - 2.2. Configure o banco de dados, seguindo as instruções durante o processo de instalação.
  - 2.3. Preencha as informações solicitadas, como nome do banco de dados, usuário e senha.
3. Configuração do Ambiente para Provedor de Hospedagem (opcional). Ao escolher um provedor de hospedagem, verifique se atende aos requisitos específicos do sistema para o LMS. Considere optar por um servidor web pago para garantir recursos adequados e suporte técnico necessário.

Os seguintes plugins são essenciais para instalação, configuração e uso do sistema:

- *Elementor*: Este construtor visual de páginas simplifica a criação de layouts sofisticados, tornando o design visual do site acessível mesmo para quem não possui conhecimentos em programação.
  - Instalação e Configuração:  
Vá para "Plugins" > "Adicionar Novo".  
Procure por "Elementor" e instale-o.  
Ative o plugin após a conclusão da instalação.
- *Tutor LMS*: Plugin robusto de Gerenciamento de Aprendizado (LMS) para WordPress, permitindo a criação e administração de cursos online com

recursos abrangentes, incluindo criação de cursos, gestão de alunos e avaliações.

- Instalação e Configuração:

- Vá para "Plugins" > "Adicionar Novo".

- Busque por "Tutor LMS" e instale-o.

- Ative o plugin após a instalação.

- *Tutor LMS Elementor Addons*: Uma extensão que se integra perfeitamente ao Elementor e ao Tutor LMS, oferecendo elementos adicionais para tornar páginas e cursos mais interativos e visualmente atrativos.

- Instalação e Configuração:

- No painel, clique em "Plugins" > "Adicionar Novo".

- Pesquise por "Tutor LMS Elementor Addons" e instale.

- Ative o plugin após a instalação.

- *Loco Translate*: Facilitando a tradução e localização do site, este plugin permite traduzir temas e plugins diretamente no painel administrativo, simplificando o processo de internacionalização.

- Instalação e Configuração:

- No painel, clique em "Plugins" > "Adicionar Novo".

- Procure "Loco Translate" e clique em "Instalar Agora".

- Ative o plugin quando a instalação for concluída.

- *Automatic Translate Addon For Loco Translate*: Complementando o Loco Translate, este addon automatiza a tradução de plugins e temas, aprimorando a localização do site e garantindo a acessibilidade a usuários de diferentes idiomas.

- Instalação e Configuração:

- No painel administrativo, vá para "Plugins" > "Adicionar Novo".

- Pesquise por "Automatic Translate Addon For Loco Translate".

- Clique em "Instalar Agora" e, em seguida, ative o plugin.

- *AI Engine: Chatbots, Geradores, Assistentes, GPT-4 e muito mais!*: Oferece recursos avançados de inteligência artificial, como chatbots, geradores de texto e integração com modelos sofisticados, incluindo o ChatGPT. Aumenta a interatividade e a inteligência do sistema.

- Instalação e Configuração:
  - Acesse o painel administrativo do WordPress.
  - Vá para "Plugins" > "Adicionar Novo".
  - Pesquise por "AI Engine" e clique em "Instalar Agora".
  - Após a instalação, ative o plugin.

**Criação e Configuração de Páginas no WordPress para o LMS.** Os seguintes passos devem ser executados:

1. Acesso ao Painel Administrativo: Faça login no painel administrativo do WordPress.
2. Criação da Página "Home"
  - 2.1. No menu lateral, vá para "Páginas" e clique em "Adicionar Nova".
  - 2.2. Nomeie a página como "Home".
  - 2.3. Adicione o conteúdo desejado para a página inicial.
  - 2.4. Salve as alterações.
3. Criação da Página "Cursos"
  - 3.1. Novamente em "Páginas", clique em "Adicionar Nova".
  - 3.2. Nomeie a página como "Cursos".
  - 3.3. Salve a página.
4. Subpáginas para Cursos
  - 4.1. Crie subpáginas para diferentes cursos, como "Programação Orientada a Objetos".
  - 4.2. Adicione conteúdos específicos para cada curso.
  - 4.3. Salve as páginas.
5. Criação da Página "Dashboard"
  - 5.1. Adicione uma nova página com o nome "Dashboard".
  - 5.2. Esta página pode ser configurada para fornecer um painel central para o usuário, exibindo seus cursos, progresso, etc.
  - 5.3. Salve as alterações.
6. Criação da Página "Sobre"
  - 6.1. Adicione uma página chamada "Sobre".

- 6.2. Forneça informações sobre o LMS, seus objetivos e qualquer outra informação relevante.
- 6.3. Salve as alterações.
7. Configuração do Menu:
  - 7.1. Vá para "Aparência" e selecione "Menus".
  - 7.2. Adicione as páginas criadas ao menu principal na ordem desejada.
  - 7.3. Salve o menu.

**Criação e Configuração de Curso.** Os seguintes passos devem ser executados para criação e configuração do "Programação Orientada a Objetos" no WordPress LMS:

1. Acesso ao Painel Administrativo:
  - 1.1. Faça login no painel administrativo do WordPress.
2. Criação de um Novo Curso:
  - 2.1. No menu lateral, vá para "Tutor LMS" e clique em "Cursos".
  - 2.2. Selecione "Adicionar Novo Curso".
3. Configurações Básicas do Curso:
  - 3.1. Nomeie o curso como "Programação Orientada a Objetos".
  - 3.2. Adicione uma descrição que destaque os objetivos e conteúdo do curso.
  - 3.3. Defina a categoria, nível de dificuldade e outras configurações relevantes.
  - 3.4. Salve as alterações.
4. Configuração das Seções do Curso:
  - 4.1. No editor do curso, crie seções para organizar o conteúdo, como "Introdução", "Conceitos Básicos", "Exemplos Práticos", etc.
  - 4.2. Adicione lições específicas a cada Seção.
5. Adição de Lições:
  - 5.1. Em cada Seção, adicione lições que representem os módulos do curso.
  - 5.2. Configure as lições com títulos, descrições e conteúdos relevantes.
  - 5.3. Se desejar, inclua recursos multimídia, como vídeos e imagens.
6. Configuração de Avaliações e *Quizzes* (Opcional):
  - 6.1. Se o curso exigir avaliações, configure questionários ou quizzes para avaliar o progresso dos alunos.

- 6.2. Determine os critérios de avaliação e pontuação.
7. Definição de Pré-Requisitos (Opcional):
  - 7.1. Se houver pré-requisitos para o curso, especifique-os para garantir que os alunos tenham o conhecimento necessário.
8. Configurações Avançadas (Opcional):
  - 8.1. Explore outras configurações avançadas oferecidas pelo Tutor LMS, como opções de pagamento, certificados, etc.
9. Publicação do Curso:
  - 9.1. Quando todas as configurações estiverem concluídas, clique em "Publicar" para tornar o curso disponível.
10. Visualização e Teste:
  - 10.1. Visualize o curso como um aluno para garantir que tudo esteja configurado corretamente.

**Configuração do Plugin para o ChatGPT.** Para tanto, os seguintes passos devem ser executados:

1. Acesso ao Painel Administrativo:
  - 1.1. Faça login no painel administrativo do WordPress.
2. Obtenção das Credenciais do ChatGPT:
  - 2.1. Acesse o site do ChatGPT (ou do provedor do serviço que você está utilizando).
  - 2.2. Registre-se para obter as credenciais necessárias para integrar o ChatGPT com o seu site.
3. Configuração do Plugin:
  - 3.1. No menu lateral, vá para "AI Engine" e clique em "Configurações".
  - 3.2. Insira as credenciais fornecidas pelo ChatGPT.
  - 3.3. Configure outras opções disponíveis, como idioma, tema do chat, etc.
4. Integração com Elementor (Opcional):
  - 4.1. Se estiver utilizando o Elementor, verifique se o plugin é compatível e se há opções de integração.
  - 4.2. Configure a aparência do chat nas opções do Elementor, se aplicável.
5. Teste do ChatGPT:

- 5.1. Vá para uma página ou post do WordPress onde você deseja incluir o chat.
- 5.2. Certifique-se de que o elemento do chat esteja adicionado à página.
- 5.3. Faça um teste interagindo com o chat para garantir que a integração esteja funcionando corretamente.
6. Personalização Adicional (Opcional):
  - 6.1. Explore as opções de personalização do chat para se adequar ao design e às necessidades do seu site.
  - 6.2. Considere a inclusão de mensagens de boas-vindas, respostas automáticas, etc.
7. Personalização do Tema no LMS (Sistema de Gestão de Aprendizado) A personalização do tema no LMS envolve a adição de imagens e a mudança de cores para criar um ambiente visualmente atrativo. Aqui está um guia passo a passo para essa tarefa:
  - 7.1. Acesso ao Painel Administrativo: Faça login no painel administrativo do WordPress.
  - 7.2. Acesso às Configurações do LMS: vá para o menu lateral e encontre a opção "Tutor LMS".
  - 7.3. Personalização do Tema: Dentro das configurações do Tutor LMS, procure por opções relacionadas ao tema ou aparência. Muitos plugins LMS, incluindo o Tutor, têm seções específicas para personalização.
  - 7.4. Seleção do Tema ou Personalização Direta: Escolha um tema existente, se disponível, ou procure por opções de personalização direta. Se o Tutor LMS não oferecer opções específicas, verifique se há uma Seção "Aparência" no menu do WordPress.
  - 7.5. Adição de Imagens: Para adicionar imagens, procure por opções como "Cabeçalho" ou "Imagens de Fundo". Faça o upload de imagens relevantes para a página inicial, cursos, etc.
  - 7.6. Mudança de Cores: Em opções de personalização, geralmente há uma Seção dedicada a cores. Escolha esquemas de cores que complementem a identidade visual desejada.
  - 7.7. Visualização Prévia: Muitos temas e plugins LMS têm opções de visualização ao vivo. Utilize essa função para observar as mudanças em tempo real. Salve as Alterações.

## **Apêndice II: Instruções para elaboração de um curso no sistema CodeChatPro**

**Professor:** Para criar cursos em um LMS (Learning Management System) baseado no WordPress, o processo pode variar um pouco dependendo do plugin específico que está sendo utilizado, como LearnDash, Tutor LMS ou Namaste LMS. Abaixo, um guia geral passo a passo que pode ser aplicado a muitos desses plugins:

Certifique-se de que o WordPress esteja instalado e funcional.

Acesse o painel administrativo do WordPress.

**Criação de Cursos:** você encontrará uma nova Seção chamada "Tutor LMS" ou algo semelhante no painel administrativo. Crie um novo curso e preencha as informações básicas, como título, descrição, objetivos, etc.

Configure as configurações específicas do curso, como a duração, a dificuldade, os requisitos e as categorias.

**Adição de Módulos e Lições:** Em um curso, você tem geralmente a capacidade de adicionar módulos e lições. Organize seu conteúdo em módulos, representando seções maiores do curso. Adicione lições aos módulos, representando unidades de conteúdo específicas.

**Configuração de Avaliações e Atividades:** Configure avaliações, questionários ou atividades práticas associadas a cada lição. Defina requisitos para conclusão de módulos ou do curso.

Antes de lançar seu curso, faça testes para garantir que tudo funcione conforme o esperado.

Revise o conteúdo para garantir qualidade e precisão.

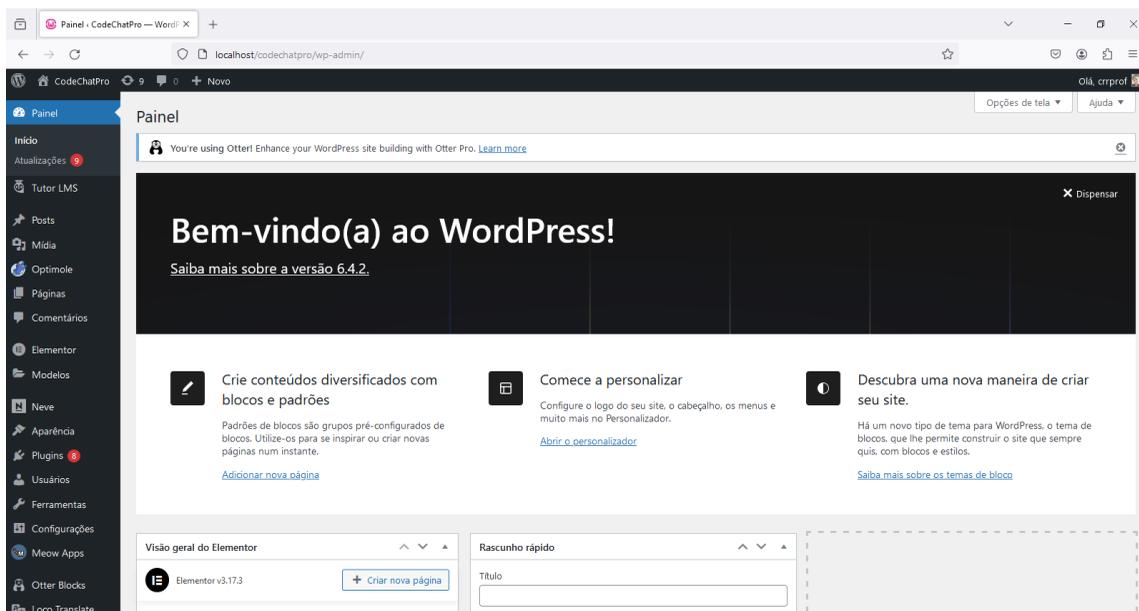


Figura II.1 – Interface sobre administração do LMS

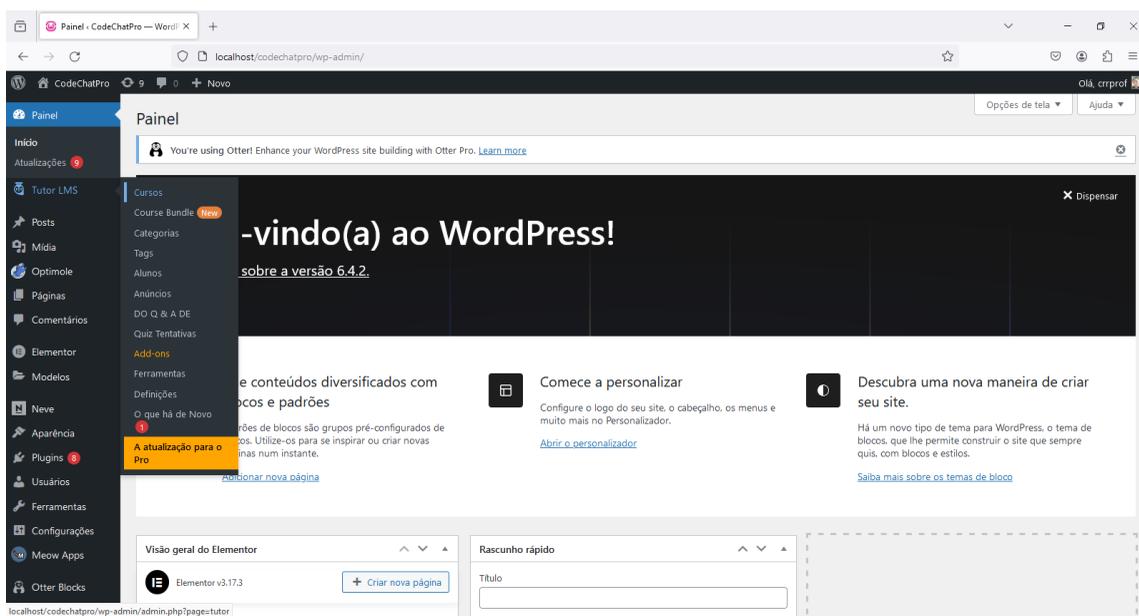
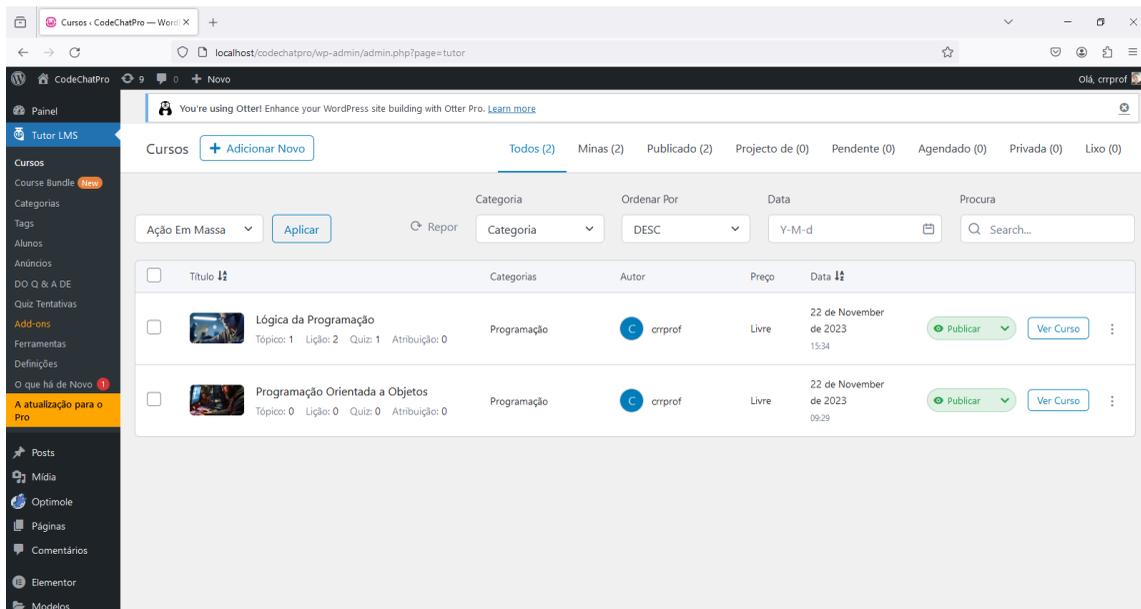
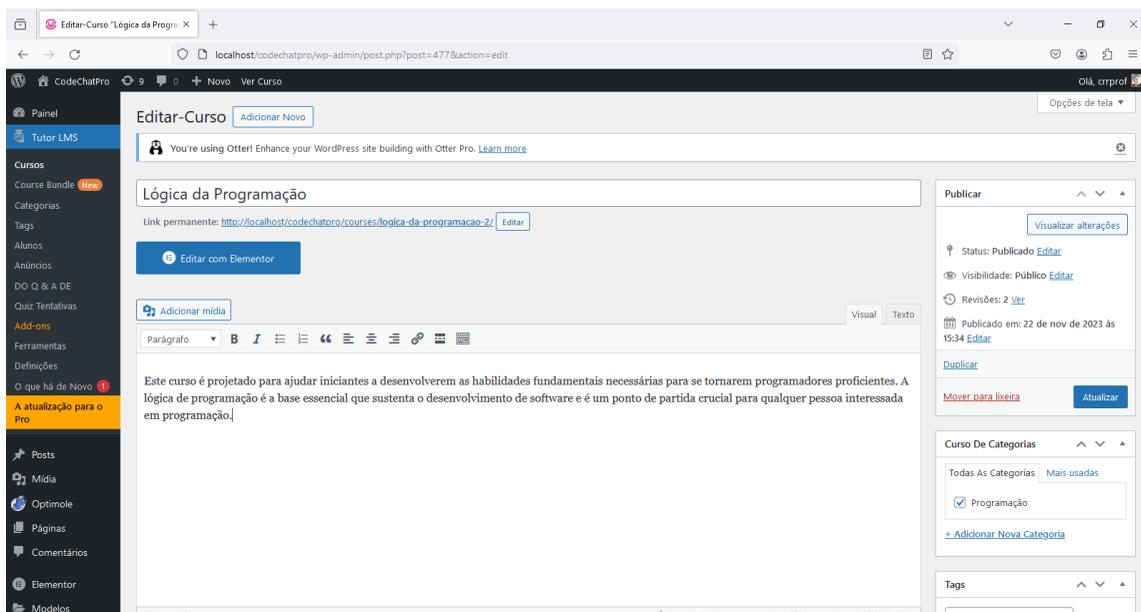


Figura II.2 – interface para criação de cursos.



**Figura II.3 – Exemplos dos cursos criados no LMS**



**Figura II.4 – Interface para edição ou criação curso.**

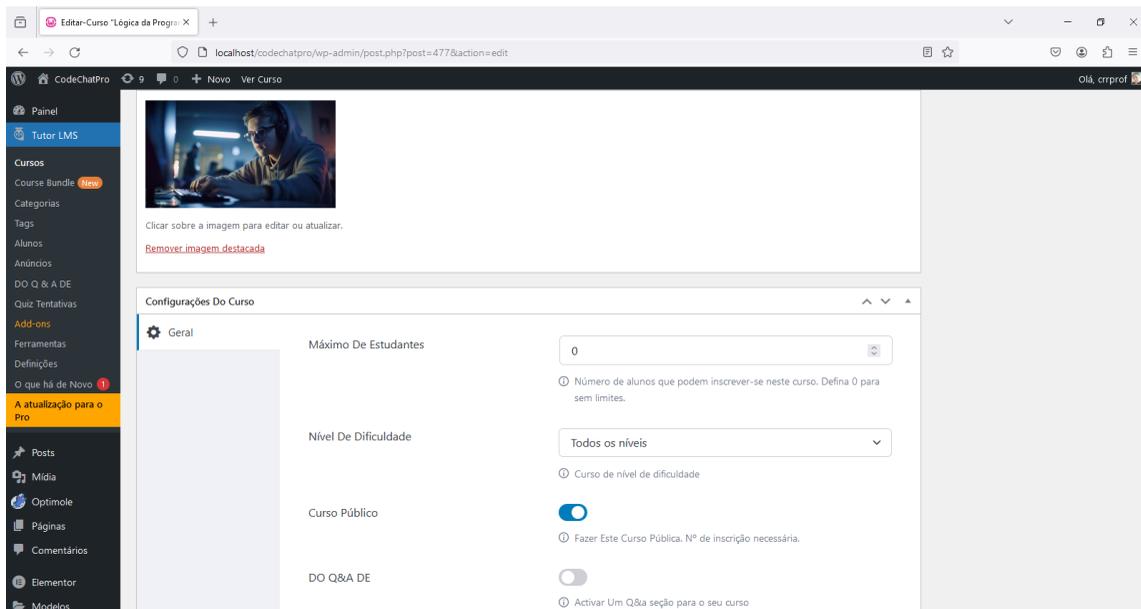


Figura II.5 – Interface para configuração de realização do curso.

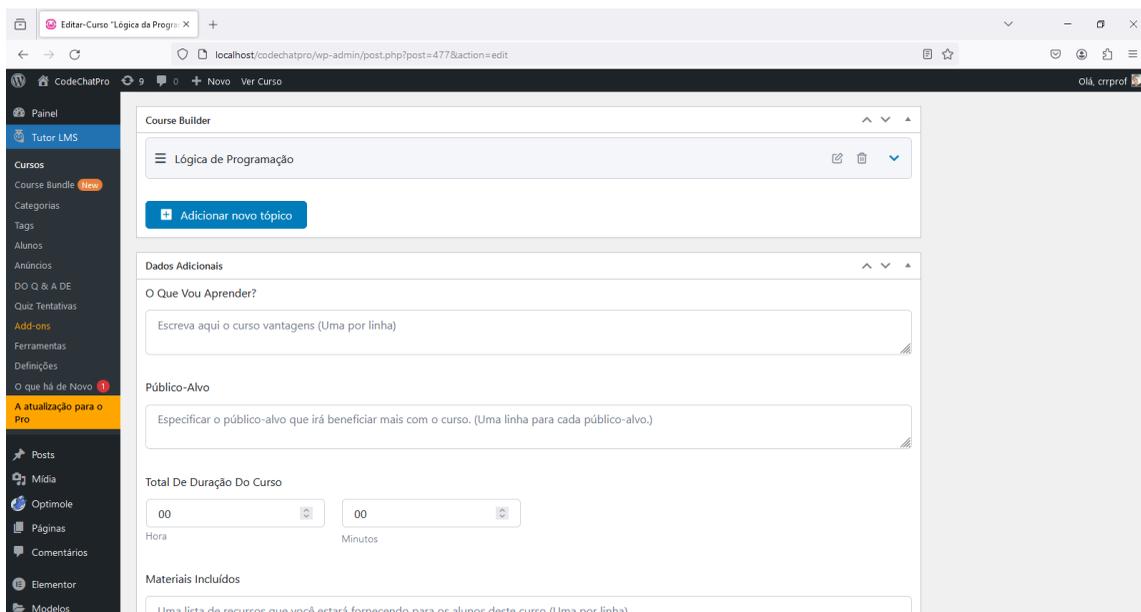
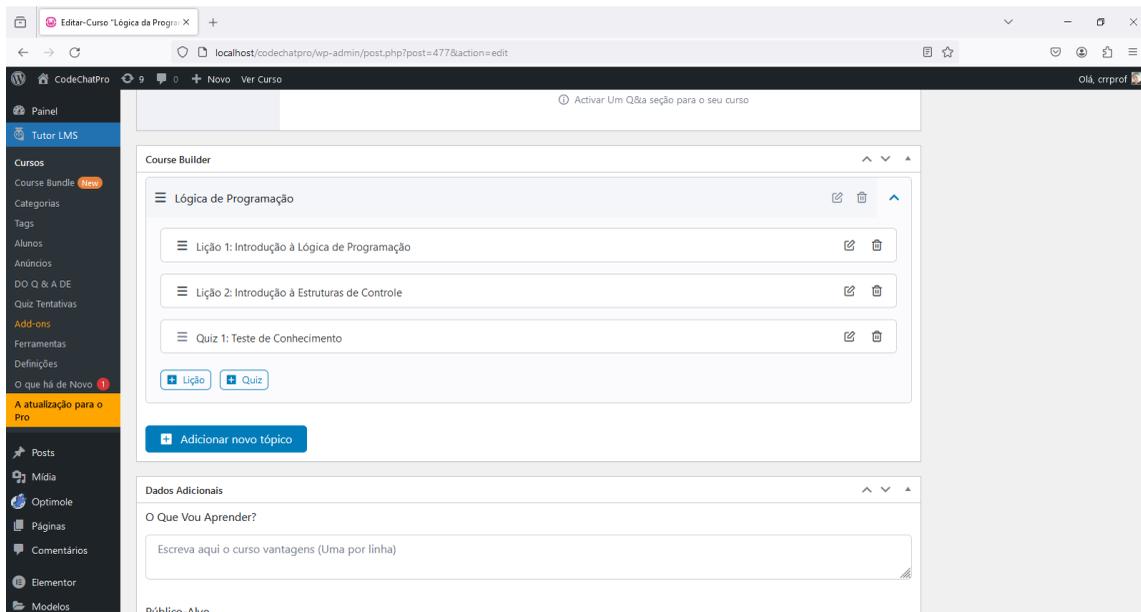
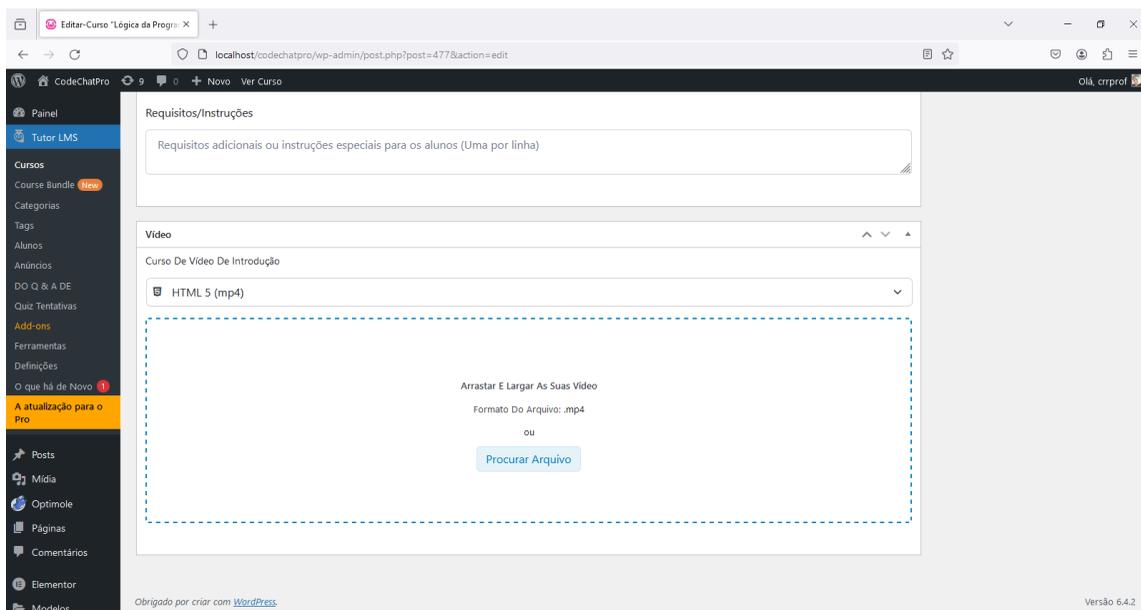


Figura II.6 – Interface para adicionar tópicos ao curso.



**Figura II.7** – Interface para adicionar lições e exercícios ao curso.



**Figura II.8** – Interface para adicionar vídeos ao curso.

## Apêndice III: Formulários de avaliação de aceitação da abordagem e do protótipo do sistema CodeChatPro

A seguir, apresentamos o conjunto de formulários utilizados para a coleta de dados após os testes, abrangendo tanto dados quantitativos quanto qualitativos.

**Qual é o seu nível de conhecimento prévio em lógica de programação?**

- Nenhuma
- Pouca (alguns conceitos básicos)
- Moderado (alguma prática com algoritmo simples)
- Avançado (experiência considerável em programação)

**Figura III.1** – Questão para verificar o nível de experiência em lógica da programação.

**Durante a realização dos exercícios, como você busca esclarecer suas dúvidas sobre lógica de programação?**

- Pesquisa na internet
- Pergunto a colegas ou amigos
- Consulto livros ou materiais didáticos
- Outro (especificar)

**Figura III.2** – Questão sobre meios para o esclarecimento de dúvidas

**Durante a realização dos exercícios, suas dúvidas foram esclarecidas?**

- Sim, completamente
- Sim, em partes
- Não, não muito
- Não, de forma alguma

**Figura III.3** – Questão sobre meios o esclarecimento de dúvidas

**Você acredita que o uso de tecnologias como um LMS com ChatGPT poderia melhorar sua experiência de aprendizado em lógica de programação?**

- Sim
- Talvez
- Não tenho certeza
- Não

**Figura III.4** – Questão sobre a experiência expectativa de uso LMS integrado com ChatGPT.

**Você poderia compartilhar sua experiência com as ferramentas que usou.**

Sua resposta

---

**Figura III.5** – Questão sobre realização dos exercícios com uso do LMS integrado ChatGPT

**Durante a realização dos exercícios com o uso do LMS integrado com ChatGPT, suas dúvidas foram esclarecidas?**

- Sim, completamente
- Sim, em partes
- Não, não muito
- Não, de forma alguma

**Figura III.6** – Questão sobre a contribuição na resolução de problemas com o uso da ferramenta

**Você acredita que o uso do LMS com ChatGPT melhorou seu desempenho nas atividades de lógica de programação?**

- Sim
- Em partes
- Não

**Figura III.7** – Questão sobre a usabilidade da ferramenta no processo de aprendizado.

**O LMS com ChatGPT foi intuitivo e fácil de usar durante a realização dos exercícios?**

- Sim
- Em partes
- Não

**Figura III.8** – Questão sobre o quão intuitivo foi o uso do LMS com o ChatGPT

**Tendo a possibilidade de usar o LMS com ChatGPT durante as aulas, você a usaria?**

- Sim
- Não

**Figura III.9** – Questão sobre a possibilidade de uso futuro nas aulas.

**Você poderia compartilhar sua experiência sobre o uso do LMS com ChatGPT?**

Sua resposta

---

**Figura III.10** – Questão sobre a experiência com uso do LMS integrado com ChatGPT.

## Apêndice IV: Questionário de Pré-teste e Pós-teste

A seguir, apresentamos os formulários com as questões utilizados para a avaliação de conhecimento (pré-teste), que foram elaborados para medir o aprendizado e compreensão dos participantes.

1: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int numero = 10;
if (numero > 5)
{
    Console.WriteLine("Número é maior que 5");
}
else
{
    Console.WriteLine("Número é menor ou igual a 5");
}
```

- O código irá compilar e imprimir "Número é maior que 5" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int numero = 10.
- O código não irá compilar porque a condição numero > 5 está incorreta.
- O código irá compilar e imprimir "Número é menor ou igual a 5" no console.

2: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int idade = 18;
if (idade >= 18)
{
    Console.WriteLine("Maior de idade");
}
else
{
    Console.WriteLine("Menor de idade");
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após int idade = 18.
- O código irá compilar e imprimir "Maior de idade" no console.
- O código não irá compilar porque a condição idade >= 18 está incorreta.
- O código irá compilar e imprimir "Menor de idade" no console.

3: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
for (int i = 0; i < 3; i++)  
{  
    Console.WriteLine(i);  
}
```

- O código irá compilar e imprimir os números de 0 a 2 no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int i = 0.
- O código não irá compilar porque a condição i < 3 está incorreta.
- O código irá compilar e imprimir os números de 0 a 3 no console.

4: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
for (int i = 1; i <= 5; i++)  
{  
    Console.WriteLine(i);  
}
```

- O código irá compilar e imprimir os números de 1 a 5 no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int i = 1.
- O código não irá compilar porque a condição i <= 5 está incorreta.
- O código irá compilar e imprimir os números de 1 a 4 no console.

5: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int contador = 0;  
while (contador < 3)  
{  
    Console.WriteLine(contador);  
    contador++;  
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após int contador = 0.
- O código não irá compilar porque a condição contador < 3 está incorreta.
- O código irá compilar e imprimir os números de 0 a 2 no console.
- O código irá compilar e imprimir os números de 0 a 3 no console.

6: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int numero = 5;
while (numero > 0)
{
    Console.WriteLine(numero);
    numero--;
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após `int numero = 5`.
- O código não irá compilar porque a condição `numero > 0` está incorreta.
- O código irá compilar e imprimir os números de 5 a 1 no console.
- O código irá compilar e imprimir os números de 1 a 5 no console.

7: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int mes = 1;
switch (mes)
{
    case 1:
        Console.WriteLine("Janeiro");
        break;
    case 2:
        Console.WriteLine("Fevereiro");
        break;
    default:
        Console.WriteLine("Outro mês");
        break;
}
```

- O código irá compilar e imprimir "Janeiro" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após `int mes = 1`.
- O código não irá compilar porque o caso `default` está incorreto.
- O código irá compilar e imprimir "Fevereiro" no console.

8: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int dia = 3;
switch (dia)
{
    case 1:
        Console.WriteLine("Domingo");
        break;
    case 2:
        Console.WriteLine("Segunda-feira");
        break;
    case 3:
        Console.WriteLine("Terça-feira");
        break;
    default:
        Console.WriteLine("Outro dia");
        break;
}
```

- O código irá compilar e imprimir "Terça-feira" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int dia = 3.
- O código não irá compilar porque o caso default está incorreto.
- O código irá compilar e imprimir "Segunda-feira" no console.

9: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int a = 10;
int b = 20
int resultado = a + b;
Console.WriteLine(resultado);
```

- O código irá compilar e imprimir "30" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int b = 20.
- O código irá compilar, mas lançará uma exceção em tempo de execução.
- O código não irá compilar, pois o faltou criar a função soma

10: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
string mensagem = "Olá, Mundo!";  
Console.WriteLine(mensagem)
```

- O código irá compilar e imprimir "Olá, Mundo!" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após Console.WriteLine(mensagem).
- O código irá compilar, mas lançará uma exceção em tempo de execução.
- O código irá compilar e imprimir "Olá e na próxima linha Mundo!" no console.

A seguir, apresentamos os formulários com as questões utilizados para a avaliação de conhecimento (pós-teste), que foram elaborados para medir o aprendizado e compreensão dos participantes.

1: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int temperatura = 25;  
if (temperatura > 30)  
{  
    Console.WriteLine("Está quente");  
}  
else  
{  
    Console.WriteLine("Está frio");  
}
```

- O código irá compilar e imprimir "Está quente" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int temperatura = 25.
- O código irá compilar e imprimir "Está frio" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int temperatura > 30.

2: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int numero = 15;
if (numero % 2 == 0)
{
    Console.WriteLine("Número é par");
}
else
{
    Console.WriteLine("Número é ímpar");
}
```

- O código irá compilar e imprimir "Número é par" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int numero = 15.
- O código irá compilar e imprimir "Número é ímpar" no console.
- O código irá compilar e imprimir "7.5" no console.

3: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
for (int i = 2; i <= 6; i++)
{
    Console.WriteLine(i);
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após int i = 2.
- O código irá compilar e imprimir os números de 2 a 6 no console.
- O código não irá compilar porque a condição i <= 6 está incorreta.
- O código irá compilar e imprimir os números de 2 a 5 no console.

4: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
for (int i = 0; i < 4; i++)
{
    Console.WriteLine("Valor de i: " + i);
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após int i = 0.
- O código irá compilar e imprimir os números de 0 a 3 no console.
- O código não irá compilar porque a condição i < 4 está incorreta.
- O código irá compilar e imprimir os números de 0 a 4 no console.

5: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int contador = 5;
while (contador > 0)
{
    Console.WriteLine(contador);
    contador--;
}
```

- O código não irá compilar devido à falta de um ponto e vírgula após int contador = 5.
- O código não irá compilar porque a condição contador > 0 está incorreta.
- O código irá compilar e imprimir os números de 5 a 1 no console.
- O código irá compilar e imprimir os números de 1 a 5 no console.

6: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int numero = 0;
while (numero < 5)
{
    Console.WriteLine("Número: " + numero);
    numero++;
}
```

- O código irá compilar e imprimir os números de 0 a 4 no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int numero = 0.
- O código não irá compilar porque a condição numero < 5 está incorreta.
- O código irá compilar e imprimir os números de 0 a 5 no console.

7: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int opcao = 2;
switch (opcao)
{
    case 1:
        Console.WriteLine("Opção 1 selecionada");
        break;
    case 2:
        Console.WriteLine("Opção 2 selecionada");
        break;
    case 3:
        Console.WriteLine("Opção 3 selecionada");
        break;
    default:
        Console.WriteLine("Opção inválida");
        break;
}
```

- O código irá compilar e imprimir "Opção 2 selecionada" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int opcao = 2.
- O código não irá compilar porque o caso default está incorreto.
- O código irá compilar e imprimir "Opção 1 selecionada" no console.

8: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
int diaDaSemana = 4;
switch (diaDaSemana)
{
    case 1:
        Console.WriteLine("Segunda-feira");
        break;
    case 2:
        Console.WriteLine("Terça-feira");
        break;
    case 3:
        Console.WriteLine("Quarta-feira");
        break;
    case 4:
        Console.WriteLine("Quinta-feira");
        break;
    case 5:
        Console.WriteLine("Sexta-feira");
        break;
    default:
        Console.WriteLine("Fim de semana");
        break;
}
```

- O código irá compilar e imprimir "Quinta-feira" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após int diaDaSemana = 4.
- O código não irá compilar porque o caso default está incorreto.
- O código irá compilar e imprimir "Número Quatro" no console.

9: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
double preco = 19.99;  
Console.WriteLine("O preço é: " + preco)
```

- O código irá compilar e imprimir "O preço é: 19.99" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após Console.WriteLine("O preço é: " + pr...
- O código irá compilar, mas lançará uma exceção em tempo de execução.
- O código irá compilar e imprimir "19.99" no console.

10: Qual mensagem será exibida no console quando o código abaixo for executado? \*

```
bool isAtivo = true  
if (isAtivo)  
{  
    Console.WriteLine("O sistema está ativo");  
}
```

- O código irá compilar e imprimir "O sistema está ativo" no console.
- O código não irá compilar devido à falta de um ponto e vírgula após bool isAtivo = true.
- O código irá compilar, mas não imprimirá nada no console.
- O código não irá compilar devido à falta de um ponto e vírgula após if (isAtivo).