

Perspectiva dos Estudantes sobre um Agente Pedagógico Baseado em Exemplos para a Aprendizagem de Programação: uma análise qualitativa

João Emilio Antonio Villa¹, Renato Garcia^{1,2}, Andre L. M. Miranda¹, Ana Oran³,
Gilleanes T. A. Guedes^{1,2}, Brenda S. Santana⁴, Davi G. Silva⁵, Pedro Valle⁶,
Williamson Silva^{1,2}

¹ Universidade Federal do Pampa - UNIPAMPA (Alegrete), Alegrete, RS, Brasil

²PPGES (UNIPAMPA - Alegrete), Alegrete, RS, Brasil

³Universidade Federal do Amazonas – UFAM, Manaus, AM, Brasil

⁴Universidade Federal de Pelotas – UFPEL, Pelotas, RS, Brasil

⁵Instituto Federal do Pará - IFPA, Itaituba, PA, Brasil

⁶Universidade de São Paulo – USP (IME), São Paulo, SP, Brasil

{¹joaovilla.aluno,¹andremiranda.aluno,^{1,2}renatogarcia.aluno}@unipampa.edu.br

³ ana.oran@icomp.ufam.edu.br,^{1,2}gilleanesguedes@unipampa.edu.br

⁴bssalenave@inf.ufpel.edu.br,⁵davi.guimaraes@ifpa.edu.br

⁶pedrohenriquevalle@usp.br,^{1,2}williamson.silva@gmail.com

Abstract. *Programming is becoming increasingly important in our society. However, learning presents significant challenges, especially for beginners. Incorporating chatbots is an effective educational strategy, offering continuous support and personalized feedback. We present CoderBot, an educational pedagogical agent based on Example-Based Learning, integrating correct and erroneous code examples. We also present the results of a qualitative study performed with 103 undergraduates from introductory programming courses. The results showed that CoderBot helped students understand the exercises, and the available examples were considered clear and detailed. However, CoderBot needs to provide more examples and improve the clarity of answers to maximize its educational potential.*

Resumo. *A programação está se tornando cada vez mais importante em nossa sociedade, mas seu aprendizado apresenta desafios significativos, especialmente para iniciantes. A incorporação de chatbots surge como uma estratégia educacional eficaz, oferecendo suporte contínuo e feedback personalizado. Este artigo apresenta o CoderBot, um agente pedagógico educacional baseado na Aprendizagem por Exemplos, que integra exemplos práticos corretos e errôneos de código. Um estudo exploratório qualitativo com 103 estudantes de disciplinas introdutórias evidenciou que o CoderBot ajudou na compreensão dos exercícios. Contudo, identificou-se a necessidade de mais exemplos e melhorias na clareza das respostas para maximizar seu potencial educacional.*

1. Introdução

A cada ano, o aprendizado de programação ganha mais importância, tanto profissionais da área de tecnologia quanto para indivíduos interessados em adquirir novas habilidades técnicas [Morais 2022]. Esse aumento é refletido no número crescente de estudantes ingressando em cursos de Computação [Papakostas et al. 2024]. Entretanto, o processo de aprendizagem da programação frequentemente apresenta desafios e complexidades significativas, especialmente nos estágios iniciais dos cursos [Morais 2022]. Esses desafios podem estar relacionados à falta de exposição prévia dos estudantes aos temas, à ausência de contato com conceitos e práticas de programação, à falta de motivação e, principalmente, à carência de um acompanhamento individualizado por parte dos docentes [Dantas 2020].

Diante desse cenário, conclui-se ser necessário que os docentes utilizem recursos e ferramentas educacionais adicionais para proporcionar um suporte eficaz aos estudantes, garantindo uma aprendizagem mais eficiente e engajadora [de Sousa Alves et al. 2021]. Uma estratégia eficaz que vem sendo adotada por diversos docentes e os auxiliando a enfrentar os desafios no aprendizado de programação é a incorporação de agentes conversacionais baseados em textos (*chatbots*) no ensino [Kernezha 2023]. Esses sistemas oferecerem suporte contínuo aos estudantes, proporcionando respostas imediatas, esclarecendo dúvidas e reforçando conceitos ensinados em sala de aula [Bii 2013, Okonkwo and Ade-Ibijola 2021]. Além disso, os *chatbots* fornecem *feedback* personalizado e motivacional, auxiliam na prática de programação por meio de exercícios interativos e permitem que os estudantes aprendam no seu próprio ritmo [Roca et al. 2024].

Contudo, muitos *chatbots* utilizados no ensino apresentam limitações devido ao uso de conjuntos de dados, gerais ou desatualizados, comprometendo a precisão das informações fornecidas [Correia et al. 2024]. Outro problema é a formalidade excessiva na linguagem e falta de didática nas respostas, dificultando a compreensão dos iniciantes [Chaves 2023]. Além disso, essas ferramentas, muitas vezes, priorizam a produtividade profissional ao invés de facilitar a aprendizagem, fornecendo soluções prontas em vez de orientações didáticas [Penney et al. 2023]. Focar apenas na implementação técnica, sem considerar o desenvolvimento do pensamento computacional, desmotiva os estudantes devido às restrições da sintaxe das linguagens de programação [Edwards et al. 2020]. Logo, há uma necessidade de *chatbots* educacionais mais adaptáveis e intuitivos, que tornem o aprendizado mais acessível e eficaz para iniciantes [Penney et al. 2023].

Neste sentido, este trabalho apresenta o CoderBot, um agente pedagógico educacional para apoiar a aprendizagem de programação. O CoderBot é fundamentado na Aprendizagem Baseada em Exemplos (ABE), integrando exemplos práticos, corretos e incorretos, para potencializar a experiência de ensino e aprendizagem. O CoderBot emerge como uma ferramenta tecnológica educacional que facilita novas abordagens de ensino e aprendizagem para docentes e estudantes iniciantes, além de impulsionar na qualidade e eficiência do aprendizado. Além disso, este artigo apresenta um estudo experimental exploratório de caráter qualitativo realizado com 103 estudantes de graduação em disciplinas introdutórias de programação, de diferentes regiões do Brasil. Após o uso do CoderBot em sala de aula, foram analisadas as percepções dos estudantes utilizando procedimentos de codificação para interpretar os dados. Essa análise permitiu entender melhor as dificuldades e os benefícios percebidos pelos estudantes, destacando como o CoderBot pode apoiar a aprendizagem de conteúdos de programação de maneira eficaz.

2. Fundamentação Teórica e Trabalhos Relacionados

Com o avanço das tecnologias e a demanda por novas opções de aprendizagem, surgiu a necessidade de ferramentas educacionais mais inovadoras, como os *chatbots* [Ilieva et al. 2023]. Os *chatbots*, ou assistentes virtuais, podem ser consideradas ferramentas educacionais importantes no processo de ensino e/ou aprendizagem, pois possibilitam que os estudantes gerenciem seu ritmo de estudo e tenham acesso contínuo a recursos educacionais [Roca et al. 2024]. Pesquisas recentes indicam que essas tecnologias não só apoiam o processo de aprendizagem, mas também aumentam sua eficácia, incentivando a participação ativa dos estudantes [Sousa and Fecchio 2021, Roca et al. 2024]. Eles proporcionam *feedback* construtivo, promovem a autoavaliação e aprimoram competências, tornando o aprendizado mais intuitivo e colaborativo [Roca et al. 2024, Pane et al. 2017]. Apesar de suas vantagens, é importante ressaltar que os *chatbots* são complementares aos docentes, que continuam desempenhando o papel principal na educação [Roca et al. 2024].

Diante desse cenário, é possível encontrar na literatura diversos trabalhos sobre o uso de *chatbots* na educação. Hobert (2019), em seu trabalho, relatou o desenvolvimento e avaliação de um *chatbot* denominado TutorCoder. O TutorCoder oferece suporte individualizado, incluindo explicações, dicas e *feedback* sobre o código-fonte dos alunos. O autor conduziu um estudo de avaliação com 40 alunos de sistemas de informação para avaliar a usabilidade do TutorCoder. Os resultados indicaram que os estudantes consideraram o *chatbot* útil para complementar o ensino de programação, especialmente quando o docente não estava disponível. Iqbal Malik *et al.* (2022), por sua vez, desenvolveram um *chatbot* para auxiliar estudantes de programação no desenvolvimento de tarefas e aprendizado. O estudo avaliou a influência do *chatbot* no desempenho dos estudantes por meio de análise das notas finais da disciplina. Os resultados indicaram que o *chatbot* foi eficaz em ajudar os estudantes a compreender conceitos fundamentais de programação e a identificar erros comuns de semântica e sintaxe.

Carreira *et al.* (2022) desenvolveram o *chatbot* Pyo para apoiar iniciantes em programação, focando na linguagem Python. O Pyo oferece suporte personalizado ajudando os estudantes no aprendizado de programação. Os autores ainda conduziram uma avaliação que mostrou que os estudantes acharam o Pyo útil para complementar seu aprendizado, destacando sua eficácia em fornecer suporte personalizado, ajudar na compreensão de conceitos e na identificação de erros. A interação positiva com o Pyo ajudou na aprendizagem dos alunos, superando dificuldades comuns entre programadores novatos. Por fim, Kasinathan *et al.* (2018) desenvolveram o TicTad, um *chatbot* para auxiliar alunos com dificuldades de aprendizagem ou interessados em aprender C#. O TicTad foi projetado para oferecer uma forma interativa e eficaz de memorizar conceitos, reduzindo tempo e esforço em comparação com métodos tradicionais. Testado por 30 estudantes iniciantes, o TicTad foi bem avaliado, sendo considerado divertido e educativo. Os resultados indicaram que o TicTad atendeu às necessidades dos usuários-alvo, proporcionando um ambiente agradável para aprender C#.

Com base nos estudos mencionados, observa-se que os trabalhos exploraram o uso de *chatbots* para apoiar os estudantes na aprendizagem, alcançando resultados positivos. Contudo, há uma lacuna na literatura no que diz respeito ao desenvolvimento de *chatbots* fundamentados em abordagens pedagógicas, como a Aprendizagem Base-

ada em Exemplos. *Chatbots* educacionais que integram exemplos corretos e incorretos podem proporcionar orientações didáticas aos estudantes, promovendo uma compreensão mais profunda dos conceitos de programação e uma aprendizagem mais significativa [Große and Renkl 2007, Adams et al. 2014, McLaren et al. 2016]. Diante disso, evidencia-se a necessidade do desenvolvimento de *chatbots* educacionais baseados em exemplos para auxiliar tanto os discentes no aprendizado quanto os docentes no ensino, além de tornar a jornada de aprendizado mais flexível e acessível.

3. CoderBot

O CoderBot é um agente educacional desenvolvido para auxiliar estudantes na aprendizagem de programação. Ele foi projetado para tornar o ensino deste tema mais eficiente, sendo importante tanto para professores quanto para estudantes, facilitando a compreensão e o aprendizado dos conteúdos de uma maneira prática e interativa. Para atingir este objetivo, o CoderBot está fundamentado na Aprendizagem Baseada em Exemplos (ABE) [Huang 2017, Van Gog and Rummel 2018, Sweller et al. 1998], que defende que o ensino é aprimorado quando não há uma sobrecarga de informações, permitindo que o foco seja mantido nos pontos-chave do problema [Sweller et al. 1998]. A ABE, alinhada com os princípios da Teoria da Carga Cognitiva, utiliza a demonstração de como realizar determinada tarefa ou assimilar uma habilidade [Van Gog and Rummel 2018]. Quando um estudante observa um exemplo de uma tarefa sendo executada com sucesso, sua crença em ser capaz de realizar o mesmo feito aumenta [Huang 2017], auxiliando na redução da carga cognitiva e fornecendo uma estrutura para alcançar o resultado esperado, o que beneficia a compreensão e o entendimento do conteúdo em questão.

Enquanto os estudantes estão aprendendo programação, podem utilizar o Coderbot para ajudar na resolução dos exercícios. O CoderBot disponibiliza exemplos (corretos e errôneos) de códigos. Dado esse contexto, o CoderBot foi integrado a um portal Web, possibilitando a apresentação de exemplos de código corretos, com seus passos detalhados, e incorretos, desafiando o estudante a identificar problemas no código e fornecendo *feedback* imediato ao estudante. As funcionalidades estão divididas em dois grupos: uma para os estudantes e outra para os professores. Para os estudantes, o CoderBot permite a escolha de uma gama de temas e conteúdos a serem estudados, apresentando exemplos corretos e incorretos de código relacionados ao conteúdo selecionado. Para os professores, o CoderBot oferece uma interface que permite personalizar os exemplos, adequando os conteúdos às necessidades específicas das turmas. O foco deste trabalho concentra-se nos estudantes. A interface do CoderBot foi desenvolvida para oferecer uma experiência de aprendizado intuitiva e fácil de entender.

Na tela inicial (Figura 1), o estudante encontra uma breve apresentação do CoderBot e uma visão inicial do seu conteúdo. Ao acessar a tela de chat do CoderBot (item (b) da Figura 1), o estudante é recebido com uma mensagem de boas-vindas e dois botões: “Funções” e “Finalizar a sessão”. A opção “Funções” apresenta um conjunto de exemplos de código (por exemplo, “Soma de dois números”, “Cálculo fatorial”), permitindo ao estudantes selecionar o item desejado. Após a seleção do tema, o conteúdo é apresentado através do enunciado de um exercício. O estudante pode optar por visualizar um exemplo correto ou errôneo do exercício proposto. Se optar por um exemplo correto, é apresentado um código correto com um passo a passo para alcançar sua resolução (Figura 2). Ao visualizar os exemplos corretos, espera-se que os estudantes desenvolvam o pensamento

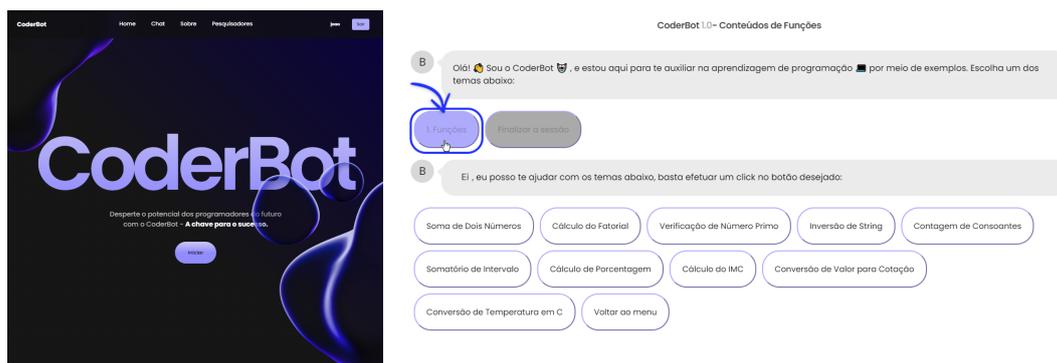


Figura 1. (a) Tela inicial e (b) Interface do Chat do CoderBot. Fonte: Autor

computacional, compreendendo a lógica por trás das questões e as etapas necessárias para resolver os problemas [Jury et al. 2024, Chen et al. 2023]. Caso opte por um exemplo errôneo, é apresentado um código com erro, e o estudante deve identificar a linha do erro entre as opções. Selecionando a linha correta, é exibida uma mensagem de felicitação, uma explicação do erro e a solução correta (Figura 3).

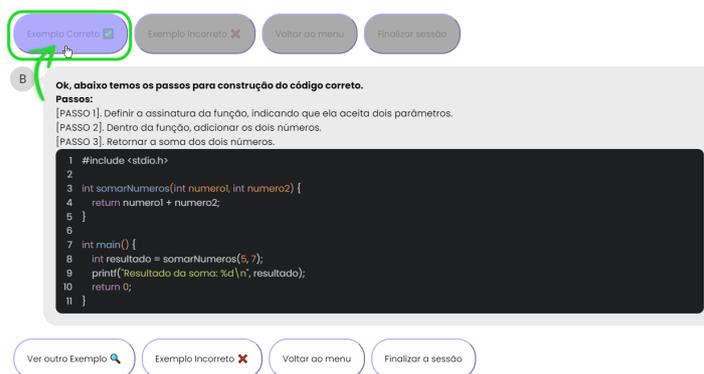


Figura 2. Exemplo Correto. Fonte: Autor

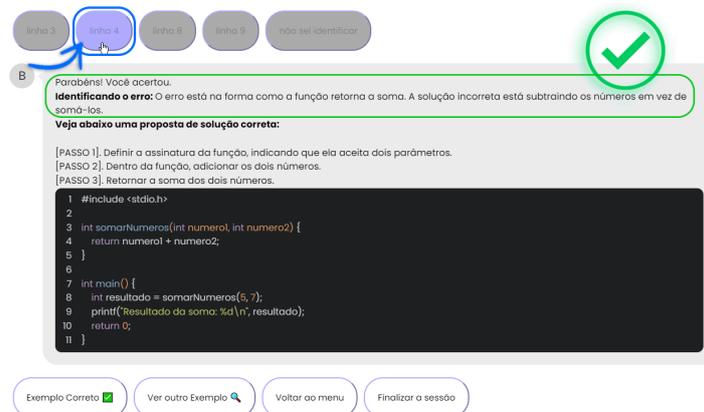


Figura 3. Selecionando a opção Correta em um Exemplo Errôneo. Fonte: Autor

Se a linha incorreta for selecionada, é exibida ao estudante uma mensagem de identificação do problema explicando o erro, uma proposta da solução correta com o

código corrigido e um passo a passo para a resolução final (Figura 4). Os exemplos errôneos visam encorajar um pensamento mais profundo sobre o conteúdo, incentivando os estudantes a se questionarem sobre o motivo do erro, a localizá-lo, compreendê-lo, explicá-lo e corrigi-lo de maneira adequada [Große and Renkl 2007]. Isso potencializa o aprendizado dos estudantes e proporciona um ambiente mais estimulante e eficaz para o ensino. Dessa forma, o CoderBot promove uma abordagem reflexiva e participativa no processo de aprendizagem. Além disso, o CoderBot oferece flexibilidade tanto para os professores quanto para os estudantes. Os professores podem disponibilizar materiais de acordo com suas necessidades, enquanto os estudantes podem consumir esses materiais em seu próprio ritmo.

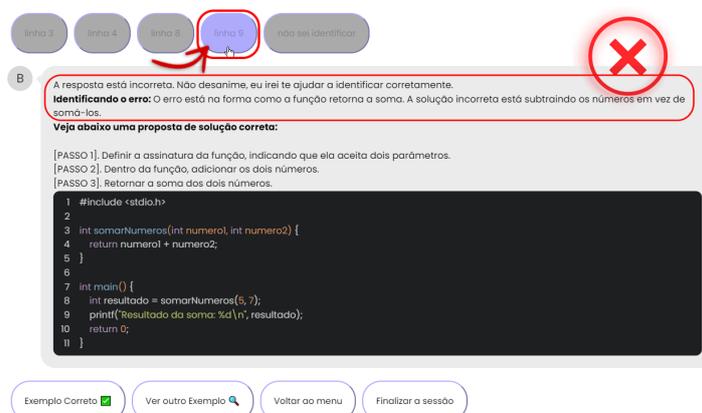


Figura 4. Selecionando a opção *Incorreta* em um Exemplo Errôneo. Fonte: Autor

4. Experimento

Para avaliar o CoderBot, foi realizado um estudo exploratório qualitativo visando compreender a percepção dos estudantes sobre CoderBot como uma ferramenta auxiliar no ensino de programação. Participaram deste estudo estudantes de diferentes Instituições de Ensino Superior (IES) que estavam cursando disciplinas iniciais de programação: O experimento ocorreu na Universidade Federal do Pampa (UNIPAMPA), Universidade Federal do Amazonas (UFAM), Instituto Federal do Pará (IFPA) e Instituto Federal de Mato Grosso do Sul (IFMS). No total, o experimento envolveu 103 estudantes, distribuídos da seguinte forma: 19 da UNIPAMPA, 15 do IFPA, 18 do IFMS e 51 da UFAM.

4.1. Planejamento do Experimento

Para facilitar a execução do estudo foram utilizadas as ferramentas disponibilizadas pelo *Google Workspace*. Os instrumentos elaborados para o experimento compreenderam: (i) termo de consentimento, que garantiu a confidencialidade dos dados fornecidos e o anonimato dos participantes; (ii) questionário de caracterização, para obter informações detalhadas sobre os conhecimentos e características dos estudantes; (iii) documentos contendo o roteiro do estudo, o link do CoderBot, a lista de exercícios a ser realizada, as instruções necessárias para a realização do experimento; e (iv) um questionário de avaliação pós-uso com perguntas abertas sobre a percepção dos estudantes sobre o Coderbot. Todos os artefatos do estudo foram revisados por outros dois pesquisadores.

4.2. Execução do Experimento

Inicialmente, foi decidido realizar um estudo piloto para verificar se o estudo alcançaria seus objetivos. Os resultados do piloto foram satisfatórios, não havendo necessidade de aprimorar o roteiro do estudo.

Os docentes das IES foram convidados via e-mail, que descrevia o objetivo do estudo e fornecia orientações detalhadas. Após a aceitação, um dos pesquisadores entrava em contato com os docentes e combinava uma data para a condução do estudo e, em seguida, enviava um e-mail com o roteiro de preparação do estudo. O e-mail enfatizava que o estudo deveria ocorrer presencialmente em laboratórios de informática e contaria com o apoio dos docentes, para que os estudantes pudessem tirar dúvidas durante a execução. Ressalta-se que esse contato com os docentes foi realizado no início do semestre letivo.

Na data agendada, o estudo foi realizado com os estudantes. O estudo foi realizado como uma atividade prática avaliativa, já prevista na ementa das disciplinas dos docentes. Durante o estudo, os docentes atuaram como moderadores, responsáveis por transmitir as informações sobre as atividades para os estudantes. Inicialmente, os moderadores solicitaram que os estudantes assinassem um termo de consentimento, concordando em participar do estudo e cedendo os dados de seus trabalhos para análise. Todos os estudantes concordaram e assinaram o termo de consentimento. Em seguida, os estudantes responderam a um formulário de caracterização, contendo perguntas sobre sua experiência em programação. A maioria dos estudantes relatou não possuir experiência prévia, uma vez que estavam nos semestres iniciais do curso. Após isso, os moderadores realizaram um treinamento sobre o CoderBot, suas funcionalidades e modo de uso, explicando como os estudantes poderiam utilizá-lo para auxiliar nas atividades de programação.

Posteriormente, os moderadores distribuíram as atividades de programação, que deveriam ser resolvidas utilizando o CoderBot como ferramenta educacional de suporte. O tempo médio para realização das atividades foi de 136 minutos (tempo mínimo 40 minutos e máximo de 150 minutos), indicando a viabilidade de uso do CoderBot durante as aulas. Após finalizar as atividades, os estudantes responderam a um questionário pós-uso, composto por perguntas abertas e específicas, projetadas para extrair a percepção de aprendizagem dos estudantes quanto ao uso do CoderBot.

4.3. Análise dos Resultados

Realizou-se uma análise específica sobre os comentários dos estudantes coletados a partir dos questionários. As respostas dos questionários foram analisadas qualitativamente seguindo os procedimentos de codificação. O objetivo da análise qualitativa foi codificar, categorizar e sintetizar dados, a fim de identificar as dificuldades e os benefícios percebidos pelos estudantes após o uso do CoderBot. Esta análise foi inspirada nos Silva *et al.* (2019), que estabelece um procedimento de análise qualitativa em quatro etapas.

Na **primeira etapa**, foram realizadas leituras de todos os comentários dos estudantes. Este momento foi importante para obter uma visão ampla dos dados fornecidos pelos estudantes durante o período de experimentação do CoderBot. Também foi realizada uma filtragem para remover comentários sem resposta ou com respostas fora do contexto, que tratavam de aspectos não relacionados ao CoderBot. Na **segunda etapa**, realizou-se a codificação aberta, na qual foram criados códigos (conceitos relevantes para a compreensão da percepção sobre o CoderBot) a partir das respostas dos participantes

(citações dos estudantes). Na **terceira etapa**, realizou-se a codificação axial, agrupando os códigos de acordo com suas propriedades, formando conceitos que representam subcategorias e categorias. Por fim, na **quarta etapa**, realizou-se uma avaliação completa da análise final para verificar a consistência dos resultados gerados. Ressalta-se que um pesquisador conduziu a análise e discutiu-a com outros pesquisadores experientes em análise qualitativa em sucessivas reuniões, realizadas para mitigar qualquer possível viés no processo de codificação. Os resultados da análise qualitativa e as discussões realizadas forneceram *insights* para melhoria do CoderBot.

5. Resultados da Análise Qualitativa

Após uma análise qualitativa dos resultados, foi possível identificar quatro principais categorias que abordam variados aspectos das visões dos envolvidos.

A primeira categoria que emergiu foi: **CODERBOT AJUDOU A MELHORAR A APRENDIZAGEM**. Esta categoria trata da assistência proporcionada pelo CoderBot, abordando como ele auxiliou no aprendizado e em quais aspectos específicos ele foi útil.

A subcategoria CoderBot ajudou a compreender a linguagem ensinada abrange comentários que declararam como o CoderBot auxiliou na aprendizagem da linguagem de programação utilizada nos exercícios. Por exemplo, E01 mencionou: “*o CoderBot me ajudou a realizar os exercícios passados e me ajudou a compreender melhor a linguagem C*”. Outro estudante complementou: “*tenho familiaridade com outra linguagem de programação, que não é C, e o Coderbot me ajudou a ter um início rápido com a sintaxe e a estrutura de um programa em C*” – E31. Os estudantes ressaltaram ainda que o CoderBot facilitou no entendimento dos exercícios, com um deles observando: “*ele facilitou na resolução dos exercícios, pois, ele otimiza o pensamento, através de um código bem escrito, fica fácil de entender qual a lógica para aquela questão*” – E08. E83 destacou que o CoderBot auxiliou na compreensão da lógica de cada exercício: “*ele ajuda no aprendizado de programação, por trazer exemplos práticos que são vistos durante as aulas de Introdução à Programação e Estruturas de Dados, etc., fazendo com que o aluno entenda sobre a lógica de programação que cada um dos exercícios propõem.*”

A subcategoria CoderBot ajudou na resolução de problemas inclui citações sobre como CoderBot auxiliou na resolução de problemas e no desenvolvimento dos exercícios. E16 relatou: “*o uso do CoderBot dá algumas luzes para a resolução de problemas*”), enquanto E14 complementou: “*o CoderBot me auxiliou na resolução dos problemas, mesmo não sendo o problema exato que eu estava tendo, mas a ferramenta me mostrou passo-a-passo para resolver o problema maior*”. Além disso, foi percebido que o CoderBot ajudou os iniciantes em programação, com estudantes destacando o apoio fornecido para quem está começando a aprender programação. O estudante E67 mencionou: “*CoderBot é uma ferramenta útil para quem está iniciando a programação*”, e outro estudante adicionou: “*serviu para auxiliar em questões básicas e para quem está começando uma linguagem é bastante interessante*” – E60.

Outra subcategoria evidenciada foi que o CoderBot provê uma explicação didática e eficaz dos exercícios, com estudantes ressaltando a clareza e a eficácia das explicações fornecidas pelo CoderBot. E57 comentou que o CoderBot disponibiliza “*um conteúdo claro e direto facilitando a leitura. Os tópicos do passo a passo da execução do código me ajudaram a entender como funciona cada linha de comando*”. E73

ainda destacou: *“foi muito mais claro em visualizar o código e em questão de aprendizado me ajudou muito, com o passo a passo fica mais fácil de compreender cada linha”*. Isso evidencia que a explicação clara e direta, com instruções passo-a-passo, pode ter facilitado a aprendizagem e compreensão dos estudantes. Por fim, a subcategoria que aborda como o CoderBot ajudou a relembrar conteúdos previamente aprendidos também foi identificada. E82 mencionou o seguinte: *“o uso do CoderBot me ajudou a melhorar, consegui me lembrar de como se utiliza o Array (aprendi no semestre retrasado), confesso que me esclareceu muitas coisas”*. E15 complementou dizendo: *“ele foi bom para algumas questões, por exemplo, quando eu precisei lembrar de alguns algoritmos que eu já tinha feito na lista, como as do somatório”*.

A segunda categoria emergente foi: **EXEMPLOS PRESENTES NO CODERBOT AJUDARAM NO ENTENDIMENTO**. Esta categoria abrange comentários dos estudantes que relataram como os exemplos fornecidos pelo CoderBot facilitaram a compreensão dos exercícios e de maneira isso ocorreu.

Uma subcategoria importante que se destacou foi CoderBot apresenta exemplos corretos e incorretos. Com base nos resultados, percebeu-se que essa abordagem melhora a compreensão dos estudantes sobre a resolução de problemas. Por exemplo, o estudante E49 afirmou que o CoderBot apresenta *“uma maneira correta e errada de fazer o exercício, assim, o aluno também aprende qual caminho não trilhar no desenvolvimento do código”*. Esta perspectiva enfatiza a importância de mostrar ambos os exemplos, corretos e incorretos, durante a explicação de um exercício, ajudando os alunos a entenderem claramente as duas formas de abordagem, a correta e a incorreta. Outra subcategoria identificada foi CoderBot apresenta a resolução do exercício detalhadamente. Os estudantes destacaram que a resolução detalhada dos exercícios pelo CoderBot os ajudou a entender melhor como resolver as questões. O estudante E57 mencionou que: *“os tópicos do passo a passo da execução do código me ajudaram a entender como funciona cada linha de comando.”* Similarmente, E73 observou: *“com o passo a passo fica mais fácil de compreender cada linha, foi muito mais claro em visualizar o código e em questão de aprendizado me ajudou muito.”*

Além disso, foi notado que CoderBot apresenta uma simplicidade na explicação dos exemplos de códigos, especificamente na forma como as resoluções dos exercícios foram exibidas. Isso é exemplificado pelo comentário de E18, que disse ter conseguido resolver os códigos dos exercícios devido à explicação clara e simplificada fornecida pelo CoderBot: *“acredito que o CoderBot colaborou na resolução de alguns exercícios pela simplicidade na explicação e na estrutura dos exemplos.”* Finalmente, houve relatos de que os exemplos auxiliaram a iniciar o código, indicando que os exemplos ajudaram os estudantes a darem o primeiro passo na resolução dos exercícios. Por exemplo, E76 mencionou que *“o CoderBot foi útil em auxiliar o início do código dando uma breve ideia de como o código poderia ser.”* O estudante E66 disse que *“a gama de dados para exemplos em demonstração é pequena, porém mostrou o caminho inicial para resolução das questões.”*

Os códigos relacionados a esta categoria indicam que, embora muitos estudantes tenham tido uma boa percepção do CoderBot, alguns encontraram dificuldades significativas em sua utilização. A categoria emergente foi: **CODERBOT NÃO AJUDOU A MELHORAR A APRENDIZAGEM**, abordando os motivos pelos quais a ferramenta não foi

útil para determinados estudantes.

Uma das dificuldades percebidas foi que é necessário realizar muita leitura a partir do CoderBot. O estudante E02 expressou que *“na minha impressão inicial, a ferramenta tem muita leitura e pouca digitação”* e também destacou que poderia utilizar outras ferramentas disponíveis para realizar a mesma função, considerando o CoderBot limitado. Esta percepção sugere que o excesso de leitura pode ser um obstáculo para estudantes que preferem uma abordagem mais prática. Outra dificuldade percebida foi que o CoderBot requer conhecimento prévio para ser efetivamente utilizado. Nesse sentido, o estudante E37 comentou: *“se quem o utilizar não tiver o mínimo de conhecimento em programar não ajudaria muito na resolução do problema.”* E52 complementou dizendo: *“ele é bem intuitivo e claro para quem já conhece um pouco sobre programação, como ainda estou no início, tive dificuldade em entender a interseção, lista, ponteiros e algumas características mais específicas.”* Esses relatos evidenciam que estudantes em estágio inicial de aprendizagem podem enfrentar dificuldades ao desenvolver os exercícios propostos com o CoderBot.

Os estudantes também relataram que o CoderBot complicou a resolução da atividade em vez de ajudar, deixando-os com mais dúvidas. E05 declarou: *“em outras questões do exercício, ele parcialmente acabou complicando o processo desenvolvimento da questão.”* P75 também corroborou, dizendo: *“às vezes, ele deixou ainda mais confuso as lógicas mais simples.”*. Esses comentários indicam que, para alguns, o CoderBot introduziu mais complexidade do que clareza. Uma quarta dificuldade mencionada foi a falta de clareza nos exemplos do CoderBot. Este problema foi destacado por comentários que mencionam a falta de clareza nos exemplos exibidos, dificultando o desenvolvimento dos exercícios. Nesse quesito, E38 destacou: *“não ajudou muito porque quando eu não entendia a lógica eu precisava de uma explicação”*, sugerindo que apenas o conteúdo apresentado pelo CoderBot não era suficiente para resolver as questões. Somando a esse cenário, E88 comentou: *“o Coderbot apresenta códigos prontos com poucos comentários, deveria apresentar de maneira mais clara a lógica por trás das questões, já que a lógica de programação é mais importante que a codificação em si.”* Esses relatos indicam a necessidade de exemplos mais detalhados e explicativos.

A última categoria emergente, **SUGESTÕES DE MELHORIAS PARA O CODER-BOT**, explora os comentários dos estudantes que propuseram aprimoramentos e ajustes para tornar o CoderBot ainda mais eficaz.

Um dos principais pontos ressaltados foi a falta de mais exemplos práticos para auxiliar no aprendizado e no desenvolvimento dos exercícios. O estudante E29 destacou que o CoderBot deveria exibir mais exemplos de programação com possíveis soluções: *“acho que deveria ter mais exemplos, mostrar mais formas de como se pode fazer uma mesma questão, não me recordo qual, mas alguma das questões ele usou boolean sendo que existe outras maneiras que não precisa usar, então acho que seria cabível ter pelo menos uns 3 exemplos de cada coisa (caso seja um pouco mais complexo).”* E77 complementou, dizendo: *“a disponibilidade de exemplos oferecida é baixa em problemas mais diversos; o Coderbot sairá para trás em relação a outros chatbots.”* Esses comentários sugerem que uma variedade maior de exemplos poderia melhorar a compreensão e a versatilidade da ferramenta. Outro ponto mencionado foi a limitação das funcionalidades limitadas do CoderBot. E01 disse: *“por mais que ele te-*

nha sido de muita ajuda, o seu estado atual é consideravelmente limitado, principalmente quanto a dúvidas mais complexas.” E41 complementou dizendo que a ferramenta foi útil, porém ainda é limitada: *“serviu para ajudar em questões básicas, mas ainda é um sistema limitado.”* E46 sugeriu o adicionar um botão de copiar e colar o código de exemplo exibido no CoderBot, para poupar tempo dos estudantes: *“inserir um botão de copiar o código seria ótimo, para poupar tempo e apenas editar o que precisa.”* Esses feedbacks indicam a necessidade de funcionalidades adicionais e maior flexibilidade no uso do CoderBot.

Os estudantes comentaram sobre a falta de comentários nos códigos. E88 observou que *“o Coderbot apresenta códigos prontos com poucos comentários, deveria apresentar de maneira mais clara a lógica por trás das questões, já que a lógica de programação é mais importante que a codificação em si.”* Comentários detalhados poderiam fornecer um melhor entendimento dos exemplos de códigos apresentados, beneficiando especialmente os iniciantes. Outro aspecto discutido foi a adição de novas funcionalidades. P85 sugeriu melhorias na interface do chat e a opção de alterar a linguagem usada nas respostas: *“acredito que a interface do chat poderia ser um pouco melhor e uma opção para alterar a linguagem usada na resposta também seria legal.”* E72 destacou a necessidade de descrições dos comandos utilizados nos exemplos: *“alguns comandos não são explicados para que servem... seria legal se tivesse a descrição dos comandos... se é para iniciantes, então eles deveriam pelo menos saber a descrição de cada comando.”* Essas sugestões indicam que melhorias na interface e nas funcionalidades do CoderBot poderiam torná-lo mais acessível e útil para todos os níveis de usuários.

6. Considerações Finais e Trabalhos Futuros

Este artigo apresentou um estudo exploratório qualitativo cujo objetivo foi identificar as percepções dos estudantes sobre CoderBot como uma ferramenta educacional no ensino de programação. Fundamentado em princípios da Aprendizagem Baseada em Exemplos, o CoderBot emprega exemplos corretos e errôneos para que os estudantes possam aprender a construção correta e identificar erros nos códigos. Atuando como um facilitador de aprendizagem, o CoderBot proporciona exemplos estruturados e personalizados, ampliando o suporte à construção de mapas mentais para os discentes que o utilizam e estimulando uma reflexão contínua. Assim, o CoderBot promove uma abordagem reflexiva e participativa no processo de aprendizagem.

Os resultados da análise qualitativa dos *feedbacks* dos estudantes revelaram percepções variadas, estruturadas em quatro categorias. Na primeira categoria (CoderBot ajudou a melhorar a aprendizagem), foram evidenciados diferentes relatos dos estudantes de como o CoderBot contribuiu na aprendizagem de programação. Os relatos dos estudantes indicam o CoderBot facilitou a compreensão da linguagem C (linguagem de programação adotada por um dos docentes durante o experimento), acelerando o entendimento da sintaxe e estrutura da linguagem adotada nos exercícios. Para alguns estudantes, o CoderBot foi especialmente útil, proporcionando suporte básico e ajudando a criar uma base sólida de conhecimento. Além disso, o CoderBot ajudou os estudantes a revisarem conteúdos previamente aprendidos, fortalecendo conceitos importantes.

A segunda categoria (Exemplos presentes no CoderBot ajudaram no entendimento) revelou que os exemplos fornecidos pelo CoderBot foram importante para aju-

dar os estudantes a entender melhor os exercícios de programação. A disponibilização de exemplos corretos e errôneos, fornecendo um ponto de partida claro para a resolução dos problemas, ajudou os estudantes a desenvolver um pensamento crítico e a aprender com os erros. As explicações detalhadas e a linguagem simples facilitaram a assimilação do conteúdo, enquanto os exemplos iniciais incentivaram o engajamento dos alunos nos exercícios. Além disso, a explicação clara e eficaz dos exemplos, com explicações passo a passo, foi importante para a compreensão dos exercícios. Esses resultados ressaltam a importância de integrar de forma correta uma abordagem pedagógica clara, completa e variada no ensino de programação.

Contrapondo a ideia da primeira categoria, a terceira categoria (CoderBot não ajudou a melhorar a aprendizagem) evidenciou que embora o CoderBot tenha potencial para auxiliar na aprendizagem, ele também apresenta alguns desafios para os estudantes. Nesse sentido, os estudantes apontaram a quantidade excessiva de leitura, a exigência de conhecimento prévio e a falta de clareza nos exemplos como principais dificuldades. Ainda, segundo os estudantes, o CoderBot, em alguns casos, tornou a solução das atividades mais complicada. Estes resultados evidenciam a necessidade de aprimoramentos no CoderBot, a fim de torná-lo mais acessível e eficiente para todos os níveis de ensino e aprimorar a eficácia da ferramenta.

Os estudantes também comentaram sobre as limitações e recomendaram sugestões de melhorias para aprimorar o CoderBot. Os estudantes destacaram que uma das limitações do CoderBot é em relação a tirar dúvidas mais complexas, que ele não executa tal ação de uma forma mais autônoma, gerando uma sensação de redundância e inferioridade em comparação com outros *chatbots*, como o ChatGPT, por exemplo. Isso também foi comentado devido à falta de exemplos variados. Os estudantes também recomendaram incluir um botão de copiar código e adicionar mais exemplos variados para cada exercício. A necessidade de comentários detalhados nos códigos dos exemplos foi ressaltada como importante para a compreensão das questões. Outras funcionalidades sugeridas incluíam a possibilidade de alterar a linguagem de programação para um mesmo exercício e a explicação dos comandos utilizados nos exercícios. As limitações e sugestões apresentadas pontuam a necessidade de realizar melhorias contínuas no CoderBot para aumentar a eficácia do CoderBot no suporte ao ensino de programação.

Com base nas evidências qualitativas encontradas no estudo, acredita-se que o CoderBot possui potencial para ser uma ferramenta educacional eficaz no apoio ao aprendizado de programação, principalmente em termos de clareza, acessibilidade e suporte contínuo. No entanto, a necessidade de melhorias podem direcionar em novas versões do CoderBot, visando torná-lo mais acessível e eficiente para todos os níveis de ensino, proporcionando uma experiência de aprendizagem mais produtiva e eficaz. Como trabalhos futuros, pretende-se realizar uma nova versão do CoderBot atendendo os pontos comentados pelos estudantes, bem como aumentar a base de dados dos exemplos disponibilizados pelo CoderBot obtidos de projetos de código aberto, que serão utilizados como modelos auxiliares pelos estudantes na resolução de problemas semelhantes. Por fim, pretende-se realizar um estudo experimental comparativo para analisar o desempenho e a compreensão dos estudantes ao utilizar diferentes ferramentas pedagógicas durante a aprendizagem de programação.

Agradecimentos

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001 e a Universidade Federal do Pampa (Unipampa - Campus Alegrete) pelo apoio. Os autores também agradecem pelo apoio financeiro da FAPERGS (Projeto ARD/ARC – processo 22/2551-0000606-0) e a FAPEMIG (Processo APQ-00743-22).

Referências

- Adams, D. M., McLaren, B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., and Van Velsen, M. (2014). Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior*, 36:401–411.
- Bii, P. (2013). Chatbot technology: A possible means of unlocking student potential to learn how to learn. *Educational Research*, 4(2):218–221.
- Carreira, G., Silva, L., Mendes, A. J., and Oliveira, H. G. (2022). Pyo, a chatbot assistant for introductory programming students. In *2022 International Symposium on Computers in Education (SIIE)*, pages 1–6. IEEE.
- Chaves, A. P. (2023). Desenho de linguagem de chatbots: influência da variação da linguagem na experiência do usuário com chatbot assistente de turismo. In *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas Colaborativos*, pages 42–47. SBC.
- Chen, O., Retnowati, E., Chan, B. K. Y., and Kalyuga, S. (2023). The effect of worked examples on learning solution steps and knowledge transfer. *Educational Psychology*, 43(8):914–928.
- Correia, J., Nicholson, M. C., Coutinho, D., Barbosa, C., Castelluccio, M., Gerosa, M., Garcia, A., and Steinmacher, I. (2024). Unveiling the potential of a conversational agent in developer support: Insights from mozilla’s pdf.js project.
- Dantas, L. G. (2020). Um protótipo de um sistema para fornecer dicas para tarefas de programação em disciplinas de programação introdutória.
- de Sousa Alves, R., do Nascimento, G. M., and de Sousa, R. R. (2021). Elementos do emprego de chatbots para auxílio no ensino de programação: Uma revisão sistemática da literatura. *Brazilian Journal of Development*, 7(5):43908–43927.
- Edwards, J., Ditton, J., Trninic, D., Swanson, H., Sullivan, S., and Mano, C. (2020). Syntax exercises in cs1. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, pages 216–226.
- Große, C. S. and Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and instruction*, 17(6):612–634.
- Hobert, S. (2019). Say hello to ‘coding tutor’! design and evaluation of a chatbot-based learning system supporting students to learn to program.
- Huang, X. (2017). Example-based learning: Effects of different types of examples on student performance, cognitive load and self-efficacy in a statistical learning task. *Interactive Learning Environments*, 25:283–294.

- Ilieva, G., Yankova, T., Klisarova-Belcheva, S., Dimitrov, A., Bratkov, M., and Angelov, D. (2023). Effects of generative chatbots in higher education. *Information*, 14(9):492.
- Iqbal Malik, S., Ashfque, M., Tawafak, R., Alfarsi, G., Usmani, N., and Hamza Khudayer, B. (2022). A chatbot to facilitate student learning in a programming 1 course: A gendered analysis. *International Journal of Virtual and Personal Learning Environments*, 12:1–20.
- Jury, B., Lorusso, A., Leinonen, J., Denny, P., and Luxton-Reilly, A. (2024). Evaluating llm-generated worked examples in an introductory programming course. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 77–86.
- Kasinathan, V., Mustapha, A., Siow, S., and Hopman, M. (2018). Tictad: A chatterbot for learning visual c# programming based on expert system,". *Indonesian Journal of Electrical Engineering and Computer Science*, 11(2):740–746.
- Kerneža, M. (2023). Fundamental and basic cognitive skills required for teachers to effectively use chatbots in education. In *Science And Technology Education: New Developments And Innovations*, pages 99–110. Scientia Socialis, UAB.
- McLaren, B. M., van Gog, T., Ganoë, C., Karabinos, M., and Yaron, D. (2016). The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior*, 55:87–99.
- Morais, C. G. B. (2022). Ensino e aprendizagem de programação: estudo de caso no ensino superior.
- Okonkwo, C. W. and Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033.
- Pane, J. F., Steiner, E. D., Baird, M. D., Hamilton, L. S., and Pane, J. D. (2017). *How Does Personalized Learning Affect Student Achievement?* RAND Corporation, Santa Monica, CA.
- Papakostas, C., Troussas, C., Krouska, A., and Sgouropoulou, C. (2024). A rule-based chatbot offering personalized guidance in computer programming education. In *International Conference on Intelligent Tutoring Systems*, pages 253–264. Springer.
- Penney, J., Pimentel, J. F., Steinmacher, I., and Gerosa, M. A. (2023). Anticipating user needs: Insights from design fiction on conversational agents for computational thinking. In *International Workshop on Chatbot Research and Design*, pages 204–219. Springer.
- Roca, M. D. L., Chan, M. M., Garcia-Cabot, A., Garcia-Lopez, E., and Amado-Salvatierra, H. (2024). The impact of a chatbot working as an assistant in a course for supporting student learning and engagement. *Computer Applications in Engineering Education*, page e22750.
- Silva, W., Steinmacher, I., and Conte, T. (2019). Students' and instructors' perceptions of five different active learning strategies used to teach software modeling. *IEEE Access*, 7:184063–184077.
- Sousa, A. C. S. d. and Fecchio, R. L. (2021). Chatbots no apoio à educação superior: revisão de literatura.

- Sweller, J., Van Merriënboer, J. J., and Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, pages 251–296.
- Van Gog, T. and Rummel, N. (2018). Example-based learning. In *International Handbook of the Learning Sciences*. Routledge.