

Avaliação de Usabilidade do CoderBot como Recurso Pedagógico no Ensino de Programação

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

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

²PPGES (UNIPAMPA - Campus 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 – (IME - USP), São Paulo, SP, Brasil

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

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

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

⁵pedrohenriquevalle@usp.br, ^{1,2}williamson.silva@gmail.com

Abstract. *Programming is one of the first subjects taught in software engineering courses. However, students often perceive these subjects as complex. In the face of advances in technology-mediated learning, new potential solutions have emerged, such as text-based conversational agents or chatbots. In this work, we present CoderBot, an educational pedagogical agent based on Example-Based Learning, designed to help beginner students understand programming content. We performed an exploratory study to evaluate the usability of CoderBot by using the System Usability Scale (SUS) questionnaire. The results indicate that the level of satisfaction when using CoderBot is acceptable.*

Resumo. *A programação é um dos primeiros conteúdos ensinados nos cursos de Engenharia de Software. No entanto, esses conteúdos frequentemente são percebidos como complexos pelos estudantes. Diante dos avanços na aprendizagem mediada pela tecnologia, surgiram novas soluções potenciais, como os agentes conversacionais baseados em texto, ou chatbots. Neste trabalho, apresentamos o CoderBot, um agente pedagógico educacional fundamentado na Aprendizagem Baseada em Exemplos, projetado para auxiliar estudantes iniciantes na compreensão de conteúdos de programação. Realizamos um estudo exploratório para avaliar a usabilidade do CoderBot utilizando o questionário System Usability Scale (SUS). Os resultados indicam que o nível de satisfação na utilização do CoderBot é aceitável.*

1. Introdução

A programação é um dos primeiros conteúdos ensinados para os estudantes que ingressam no curso de Engenharia de Software. A partir destes conteúdos, os estudantes começam a compreender os elementos básicos de programação, estruturas condicionais, laços de repetição, dentre outros. Há relatos de que os conteúdos relacionados à programação de computadores apresentam um alto nível de complexidade tanto no ensino quanto na aprendizagem. Consequentemente, as taxas de aprovação são baixas [Alves et al. 2019], resultando em altas taxas de retenção que podem implicar diretamente na evasão dos estudantes [Penney et al. 2023]. Manter os estudantes motivados e engajados nas disciplinas de programação tem sido uma tarefa complexa e árdua para os docentes [Robins 2019].

Uma tecnologia emergente que ganhou notoriedade nos últimos anos são os *chatbots*. *Chatbots* são aplicações de software que interagem com os usuários em diversos aspectos da vida diária por meio de linguagem natural e/ou dados predefinidos [Mageira et al. 2022]. *Chatbots* também têm sido adotados como agentes pedagógicos, especialmente no ensino de Computação (como Teste de Software e Pensamento Computacional), possibilitando que os estudantes sejam mais ágeis no processo de aprendizagem, tirem dúvidas específicas e obtenham um *feedback* imediato e adequado sem uma forte dependência por parte dos professores [Ruan et al. 2019].

O objetivo principal deste trabalho é apresentar o CoderBot, um agente pedagógico de apoio aos estudantes de graduação no ensino de conteúdos de programação. O CoderBot visa proporcionar um ambiente de aprendizagem em que os estudantes possam praticar exercícios de programação. O CoderBot é fundamentado na Aprendizagem Baseada em Exemplos (ABE), uma estratégia pedagógica recomendada e utilizada quando se deseja promover a aquisição inicial de habilidades cognitivas, especialmente, quando os estudantes estão aprendendo a resolver certos tipos de problemas, encorajando os estudantes a explicar a lógica por trás das soluções para si mesmos [McLaren et al. 2016]. Durante a resolução de exercícios usando o CoderBot, o estudante averigua as explicações detalhadas sobre o funcionamento do código e testes para verificar se o resultado obtido está correto. A fim de avaliar a usabilidade do CoderBot conduziu-se um estudo exploratório com os 103 estudantes de graduação em disciplinas introdutórias de programação. A avaliação foi conduzida por meio do *System Usability Scale* (SUS).

2. Fundamentação Teórica

2.1. Aprendizagem Baseada em Exemplos

Os exemplos são ferramentas úteis, para docentes e discentes, pois eles podem ser moldados para se adequarem a diversas realidades educacionais. Nesse contexto, surge a Aprendizagem Baseada em Exemplos (ABE), fundamentada na Teoria da Carga Cognitiva, 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]. O uso de exemplos auxilia na redução da carga cognitiva, fornecendo uma estrutura a ser seguida para alcançar o resultado esperado, o que beneficia a compreensão e o entendimento do conteúdo em questão. Há abordagens para realizar demonstrações de exemplos no processo de aprendizagem e algumas desviaram-se da apresentação de uma solução didática e usaram exemplos errôneos com uma instrução para os estudantes encontrarem e corrigirem os erros, ou com *feedback* adicional estimulando os estudantes a processar

os exemplos mais profundamente [McLaren et al. 2016]. O estudo da ABE é um tema relevante para compreender e aprimorar as práticas de ensino e aprendizagem.

2.2. Ensino por meio de *chatbots*

Ondáš *et al.* (2019) indicam que um dos principais focos de *chatbots* concentra-se em reduzir carga humana destinada a suporte de aplicações, permitindo que muitas soluções para problemas corriqueiros de software sejam definidas por meio de um passo a passo, que o usuário pode seguir ou verificar para obter informações úteis.

Há diversos registros de *chatbots* sendo usados como agentes pedagógicos. Wollny *et al.* (2021) demonstram uma grande quantidade de *chatbots* destinados à aprendizagem de idiomas, matemática, psicologia, administração, dentre outros. Em especial, ao ensino de programação, Hoberto (2019), por exemplo, desenvolveram o Coding Tutor, um sistema de aprendizado baseado em *chatbot* projetado para auxiliar programadores novatos em ambientes de aprendizagem formais. O autor realizou uma avaliação do Coding Tutor e os resultados evidenciaram que a interação com o Coding Tutor demonstrou ser eficaz na facilitação do entendimento e na resolução de tarefas de programação, evidenciando melhorias na motivação e autoconfiança dos estudantes.

Carreira *et al.* (2022) desenvolveram e avaliaram o *chatbot* educacional Pyo. Pyo foi projetado para ajudar estudantes iniciantes no aprendizado da linguagem Python. Os autores avaliaram a eficácia do Pyo com 22 estudantes de um curso de programação introdutória. Os resultados indicaram que o Pyo facilitou a aprendizagem dos estudantes, proporcionando suporte contínuo e assistência personalizada, embora os estudantes tenham solicitado explicações mais claras sobre suas funcionalidades.

Winkler *et al.* (2020) desenvolveram a Sara, um agente conversacional que proporciona *scaffolding* durante vídeo-aulas *online*, usando tanto entrada de voz quanto texto, para apoiar a retenção de informações e a capacidade de transferir o conhecimento adquirido para novos problemas de programação. A avaliação da Sara, com 182 estudantes, demonstrou melhorias significativas na aprendizagem em tarefas de programação em comparação com agentes conversacionais tradicionais que não utilizam *scaffolding*.

Além dos trabalhos mencionados acima, há vários estudos explorado o uso de *chatbots* para melhorar a aprendizagem em ambientes educacionais [Pérez et al. 2020], especialmente na programação. Contudo, há uma lacuna na literatura no que diz respeito ao desenvolvimento de *chatbots* fundamentados em abordagens pedagógicas, como a Aprendizagem Baseada em Exemplos. *Chatbots* educacionais que integram exemplos corretos e incorretos pode proporcionar orientações didáticas aos estudantes, promovendo uma compreensão mais profunda dos conceitos de programação e uma aprendizagem mais significativa [Adams et al. 2014, McLaren et al. 2016]. Diante disso, evidencia-se a necessidade do desenvolvimento *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 pedagógico desenvolvido para auxiliar estudantes de graduação no aprendizado de programação, fundamentado em princípios da Teoria da Carga

Cognitiva e da Aprendizagem Baseada em Exemplos (ABE). A Teoria da Carga Cognitiva postula que a capacidade cognitiva dos aprendizes é limitada, e que a eficácia do aprendizado pode ser comprometida quando essa capacidade é sobrecarregada [Sweller et al. 1998]. ABE oferece aos estudantes exemplos práticos que reduzem a carga cognitiva ao demonstrar explicitamente as etapas necessárias para resolver um problema [Adams et al. 2014, McLaren et al. 2016]. Isso permite aos estudantes focar em partes específicas do problema de cada vez, facilitando não apenas a compreensão, mas também o desenvolvimento de esquemas de aprendizagem mais generalizados e aplicáveis a diferentes contextos de ensino [McLaren et al. 2016]. Diante disso, o CoderBot utiliza uma combinação de exemplos corretos e errôneos para enriquecer as concepções mentais dos estudantes, proporcionando uma experiência de aprendizado mais eficaz.

O CoderBot foi integrado a um site, possibilitando a apresentação exemplos de códigos corretos, que guia os estudantes passo a passo, e códigos errôneos, que incentiva os estudantes a se questionarem sobre o motivo do erro, a localizá-lo, compreendê-lo, explicá-lo e corrigi-lo de maneira adequada. A interface do usuário é simples e intuitiva, operando por meio de botões, facilitando a navegação e a personalização da experiência de aprendizado. Para demonstrar a aplicação prática do CoderBot, será utilizado o tema de Vetores (*Arrays*), ver Figura 2, mas é possível habilitar a exibição de outros conteúdos (Matrizes, Listas, Filas, entre outros) e o tipo de linguagens de programação que se deseja apresentar os exemplos. Essa customização é realizada pelo docente no painel administrativo do docentes antes do início da aula.



Figura 1. CoderBot - Descrição do *Worked Example*

Ao acessar o site do CoderBot, o estudante é recebido com uma mensagem de boas-vindas, seguida pela apresentação dos conteúdos de programação disponíveis. Ao selecionar o botão de conteúdo “1. Vetores (Arrays)”, são apresentados os exemplos disponíveis para este tópico. Cada exemplo é representado por um título, e a escolha de um exemplo específico direciona o estudante para uma tela onde são apresentados os elementos essenciais dos exemplos, incluindo a descrição do exemplo, o resultado esperado após a execução correta do algoritmo, um aspecto reflexivo que incentiva a reflexão sobre o exemplo e o problema associado, e informações sobre como realizar os testes para verificar a implementação do estudante (Figura 2).

Após a leitura e análise dos termos apresentados (Figura 2), o estudante acessa

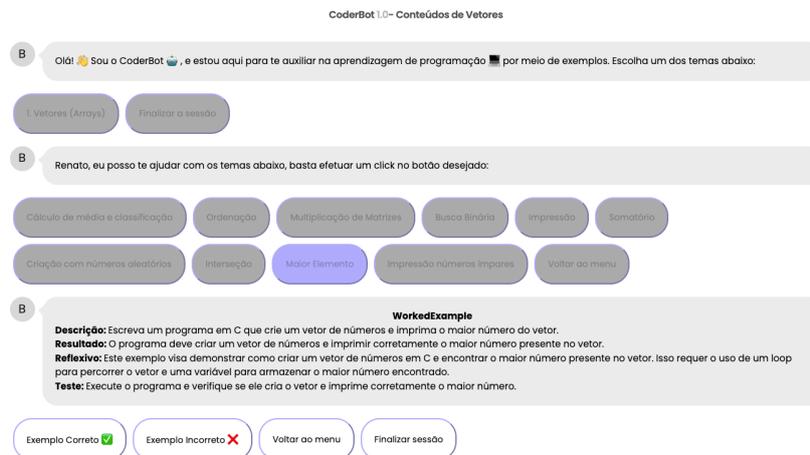


Figura 2. CoderBot - Descrição do *Worked Example*

um menu com as opções **Exemplo Correto**, **Exemplo Incorreto**, além das opções de retornar ao menu inicial de conteúdos e finalizar a sessão. Ao escolher a opção de **Exemplo Correto**, o estudante é redirecionado para uma tela onde são fornecidas informações passo-a-passo para a elaboração do código correto, bem como o próprio código correto (Figura 3). Caso o estudante opte por visualizar um **Exemplo Incorreto**, o CoderBot

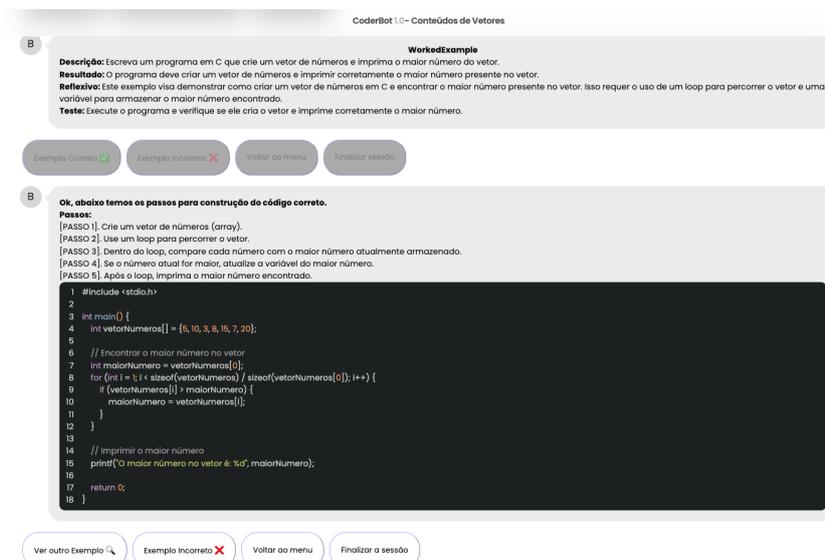


Figura 3. CoderBot - *Worked Example* - Correto

apresenta uma solução incorreta e solicita que o estudante identifique o erro. A pergunta oferece cinco opções de resposta: quatro alternativas específicas e uma para informar que não conseguiu identificar o erro a partir do exemplo (ver Figura 4). A escolha de uma opção leva a três possibilidades:

- Se o estudante escolher a opção correta, ele é parabenizado e recebe um feedback explicativo sobre o erro e o código correto é apresentado;
- Se o estudante escolha a opção incorreta, ele recebe feedback informando que a alternativa escolhida está errada, com indicação do ponto específico do erro. Posteriormente, são apresentados os passos corretos para a solução correta, seguidos pelo código correto.

- Se o estudante escolher a opção “Não Sei Identificar”, ele recebe feedback indicando onde ocorre o erro, e o assistente fornece os passos para desenvolver uma solução correta, seguidos pelo código correto.

Em ambas as telas de Exemplo Correto e Exemplo Incorreto, há opções para alternar entre os exemplos, retornar ao menu inicial ou finalizar a sessão.

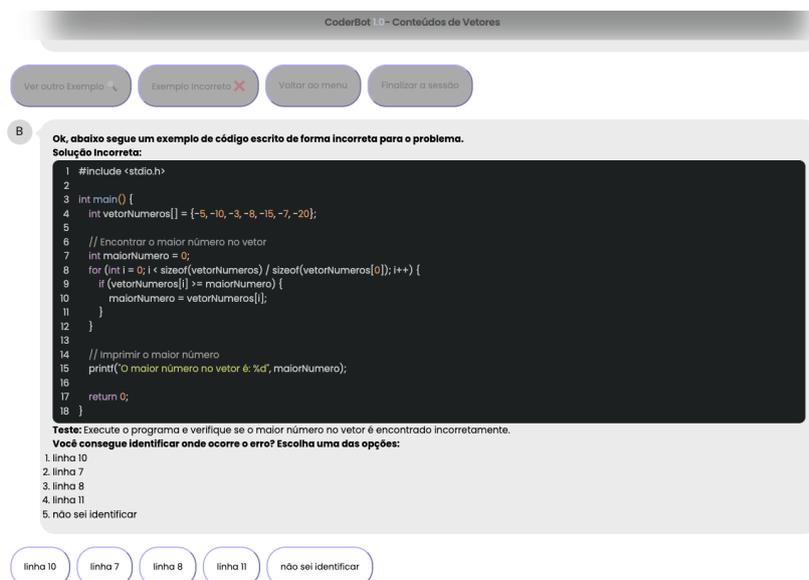


Figura 4. CoderBot - *Worked Example* - Incorreto

A adoção do CoderBot em sala de aula não só facilita a aprendizagem, mas também promove a autonomia dos alunos. Em vez de simplesmente fornecer respostas prontas, o CoderBot guia os alunos através de um processo de descoberta, incentivando-os a desenvolver suas próprias soluções para os problemas de codificação. Essa abordagem é consistente com a literatura que destaca a importância da autonomia e da reflexão crítica no aprendizado de programação.

4. Estudo Exploratório

Para avaliar o CoderBot, foi realizado um estudo exploratório com o objetivo de compreender a percepção dos estudantes sobre CoderBot como uma ferramenta auxiliar no ensino de programação.

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 e quando os pesquisadores identificavam algum problema, uma nova versão do artefato era elaborada.

4.2. Participantes

Participaram deste estudo estudantes de diferentes Instituições de Ensino Superior (IES) que estavam cursando disciplinas iniciais de programação de três diferentes regiões do Brasil. No total, o experimento envolveu 103 estudantes. Mais detalhes na Tabela 1.

Tabela 1. Visão geral dos participantes do estudo.

IES	Curso	Docentes	Disciplina	Conteúdo	#Estudantes
UNIPAMPA	Ciência da Computação	D1	Algoritmos e Programação para Computação	Funções	8
UNIPAMPA	Engenharia de Software	D2	Algoritmos e Programação	Arrays	11
IFMS	Tecnologia em Análise e Desenvolvimento de Sistemas	D3	Programação de Computadores	Listas	15
IFMS	Tecnologia em Análise e Desenvolvimento de Sistemas	D4	Linguagem de Programação 1	Arrays	18
UFAM	Engenharia de Software	D5	Algoritmos e Estruturas de Dados 1	Funções	51

4.3. Execução do Experimento

O experimento foi realizado em três IES, com a participação de 103 alunos de cinco turmas. Inicialmente, um estudo piloto foi conduzido para avaliar o roteiro, que se mostrou satisfatório, não necessitando de ajustes. Os docentes das IES foram convidados via e-mail, que detalhava o objetivo do estudo e fornecia orientações. Após a aceitação, um pesquisador agendava a data para a condução do estudo e enviava um e-mail com o roteiro de preparação, enfatizando a necessidade de realizar o estudo presencialmente em laboratórios de informática, com o apoio dos docentes para tirar dúvidas dos estudantes.

No dia agendado, o estudo foi realizado como uma atividade prática avaliativa, prevista na ementa das disciplinas. Os docentes atuaram como moderadores, transmitindo as informações sobre as atividades para os estudantes. Inicialmente, os estudantes assinaram um termo de consentimento, concordando em participar do estudo e ceder seus dados para análise. Todos os estudantes concordaram e assinaram o termo.

Em seguida, os estudantes responderam a um formulário de caracterização sobre sua experiência em programação, que demonstrou que a maioria não possuía experiência prévia. Após isso, os moderadores realizaram um treinamento sobre o CoderBot, suas funcionalidades e modo de uso. Posteriormente, distribuíram as atividades de programação, que deveriam ser resolvidas utilizando o CoderBot. O tempo médio para realização das atividades foi de 136 minutos (mínimo 40 minutos e máximo 150 minutos), indicando a viabilidade de uso do CoderBot durante as aulas.

Após a conclusão das atividades, foi realizada a avaliação de usabilidade com o método *System Usability Scale* (SUS). Desenvolvido por Brooke *et al.* (1996), o SUS é um questionário validado e comumente usado para mensurar a capacidade de aprendizagem, eficiência, capacidade de memorização, minimização de erros e satisfação de aplicações. O SUS é composto por dez questões (ver Tabela 2), acompanhadas de opção de resposta em escala Likert variando de 1 (Discordo Fortemente) a 5 (Concordo Fortemente). O *score* do SUS é feito a partir de cálculos que permitem ter uma visão geral do grau de usabilidade do sistema avaliado. Para obter o *score* é realizado o cálculo de cada um dos dez itens, em que, nas questões ímpares (1, 3, 5, 7, 9), a nota recebida pelo usuário na escala Likert é subtraída de 1 (nota do usuário - 1). Já para os itens pares (2, 4,

6, 8, 10), o cálculo é 5 menos a nota recebida pelo usuário (5 - nota do usuário). O resultado final é dado pela soma dessas questões após o cálculo e pela multiplicação por 2,5 [Brooke et al. 1996]. A pontuação do SUS varia de 0 a 100, onde pontuações abaixo de 60 são classificadas como ruins, entre 60 e 69 como fracas, entre 70 e 79 como média, entre 80 e 89 como boas e 90 e acima como excelentes [Brooke et al. 1996].

Tabela 2. Questões Escala SUS.

#	Questão
1	Eu acho que eu gostaria de utilizar este sistema frequentemente.
2	Eu achei o sistema desnecessariamente complexo.
3	Eu achei o sistema fácil de usar.
4	Eu acho que eu precisaria da ajuda de uma pessoa com conhecimentos técnicos para eu conseguir utilizar o sistema.
5	Eu acho que as várias funcionalidades do sistema estão bem integradas.
6	Eu achei que tinha muita inconsistência no sistema.
7	Eu acho que a maioria das pessoas iriam aprender facilmente a utilizar esse sistema.
8	Eu achei o sistema muito incômodo de usar.
9	Eu me senti muito confiante em utilizar o sistema.
10	Eu precisei aprender muito antes de utilizar esse sistema.

5. Resultados

Ao todo, participaram do estudo 103 estudantes de graduação, em que 58,26% tinham idade entre 18 e 20 anos, 38,26% tinham entre 21 e 30 anos, 2,61% tinham entre 31 e 15 anos, e apenas 0,87% tinham idade acima de 50 anos. Além disso, 73,03% se declararam como Homem, 23,48% se declararam como Mulher, 2,61% se declaram como Não-Binário e 0,87% preferiram não comentar.

Como informado anteriormente, a análise foi realizada com base no questionário SUS, a qual foi escolhido por se tratar de uma metodologia rápida e eficaz para realizar a avaliação de diferentes tipos e complexidades de sistemas. De acordo com Sauro (2011) para que o sistema avaliado seja considerado de boa usabilidade, o *score* mínimo deve ser 68 pontos. A avaliação dos resultados da escala SUS retornou um *score* de 79,2 pontos, o que pode ser classificado como tendo média usabilidade. Ainda, calculou-se o desvio padrão, indicando que os valores da amostra estão variando 11,7 pontos da média.

Foi utilizada uma questão aberta para que os discentes dissertaram sua opinião sobre o CoderBot. Os participantes concordaram que o CoderBot foi fácil de utilizar. O comentário de E50 foi: *“a parte visual do CoderBot é boa, agradável”*, e o E57 complementou que *“o CoderBot tem uma interface de fácil entendimento.”* P23 destacou que se sentiu livre de dificuldades e que achou a interface intuitiva: *“utilizando o CoderBot, consegui aprimorar minha compreensão de códigos de programação. Sua interface é intuitiva, simplifica a interação, tornando sua utilização livre de dificuldades.”* Assim como ele, P40 também pontuou: *“ajudou, foi simples e rápido e ele é intuitivo também.”* P48, disse que conseguiu entender os trechos de código, e devido a isso, gostou da forma como ocorre a utilização da ferramenta: *“com o uso do CoderBot consegui entender trechos de código. Achei interessante o método de utilização dele.”*

Um dos principais pontos negativos ressaltados foi a **falta de mais exemplos práticos** para auxiliar no aprendizado e no desenvolvimento dos exercícios. O estudante E77 destacou que o CoderBot deveria exibir mais exemplos de programação com possíveis soluções: *“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 tenha 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.

6. Considerações Finais

Neste trabalho apresentamos o Coderbot, um agente pedagógico fundamentado na ABE, que atua como um facilitador de aprendizagem ao proporcionar exemplos estruturados e personalizados. Além disso, apresentou-se uma avaliação de usabilidade do CoderBot, a partir da coleta de dados quantitativos e qualitativos, com discentes de diferentes instituições de ensino. Os resultados quantitativos, realizados por meio da escala SUS, demonstraram que o CoderBot alcançou o *score* que seria considerado aceitável no quesito usabilidade, ou seja, a pontuação média do SUS foi de 79,2 em 100. Os resultados do estudo indicam que o CoderBot foi percebido com boa usabilidade pelos estudantes, sendo percebido também como uma ferramenta útil e fácil de usar para o aprendizado de programação. No entanto, os principais pontos negativos destacados foram a falta de mais exemplos práticos e as funcionalidades limitadas do CoderBot para resolver dúvidas mais complexas. Esses feedbacks indicam a necessidade de melhorias na variedade de exemplos e na flexibilidade das funcionalidades do CoderBot.

Como trabalhos futuros pretende-se realizar as melhorias apontadas pelos participantes, realizar uma nova versão do CoderBot e realizar novas avaliações de usabilidade, de UX e da experiência de aprendizagem dos estudantes afim de compreender com estes percebem o uso do CoderBot durante o processo de aprendizagem de programação. Como uma perspectiva para trabalhos futuros, planeja-se integrar ao *chatbot* uma inteligência artificial baseada em *Large Language Models*, que possa compreender com maior precisão as dificuldades dos estudantes e interpretar diferentes formas de interação. Isso tornaria o *CoderBot* mais próximo do usuário, permitindo uma interação mais fluida e adaptada às preferências individuais.

Agradecimentos

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001 e a Universidade Tecnológica Federal do Paraná (UTFPR - Cornélio Procópio) 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., e Van Velsen, M. (2014). Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior*, 36:401–411.

- Alves, G., Rebouças, A., e Scaico, P. (2019). Coding dojo como prática de aprendizagem colaborativa para apoiar o ensino introdutório de programação: Um estudo de caso. Em *Anais do XXVII Workshop sobre Educação em Computação*, páginas 276–290. SBC.
- Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Carreira, G., Silva, L., Mendes, A. J., e Oliveira, H. G. (2022). Pyo, a chatbot assistant for introductory programming students. Em *2022 International Symposium on Computers in Education (SIIE)*, páginas 1–6.
- Hobert, S. (2019). Say hello to ‘coding tutor’! design and evaluation of a chatbot-based learning system supporting students to learn to program.
- Mageira, K., Pittou, D., Papasalouros, A., Kotis, K., Zangogianni, P., e Daradoumis, A. (2022). Educational ai chatbots for content and language integrated learning. *Applied Sciences*, 12(7):3239.
- McLaren, B. M., van Gog, T., Ganoë, C., Karabinos, M., e 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.
- Ondáš, S., Pleva, M., e Hládek, D. (2019). How chatbots can be involved in the education process. Em *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, páginas 575–580. IEEE.
- Penney, J., Pimentel, J. F., Steinmacher, I., e Gerosa, M. A. (2023). Anticipating user needs: Insights from design fiction on conversational agents for computational thinking. Em *International Workshop on Chatbot Research and Design*, páginas 204–219.
- Pérez, J. Q., Daradoumis, T., e Puig, J. M. M. (2020). Rediscovering the use of chatbots in education: A systematic literature review. *Computer Applications in Engineering Education*, 28(6):1549–1565.
- Robins, A. V. (2019). 12 novice programmers and introductory programming. *The Cambridge handbook of computing education research*, página 327.
- Ruan, S., Jiang, L., Xu, J., Tham, B. J.-K., Qiu, Z., Zhu, Y., Murnane, E. L., Brunskill, E., e Landay, J. A. (2019). Quizbot: A dialogue-based adaptive learning system for factual knowledge. Em *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, páginas 1–13.
- Sweller, J., Van Merriënboer, J. J., e Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, páginas 251–296.
- Winkler, R., Hobert, S., Salovaara, A., Söllner, M., e Leimeister, J. M. (2020). Sara, the lecturer: Improving learning in online education with a scaffolding-based conversational agent. Em *CHI Conference on Human Factors in Computing Systems, CHI '20*, página 1–14, New York, NY, USA. Association for Computing Machinery.
- Wollny, S., Schneider, J., Di Mitri, D., Weidlich, J., Rittberger, M., e Drachsler, H. (2021). Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.