

Avaliando a Autoeficácia e a Aceitação do CoderBot em Cursos Introdutórios de Programação: um estudo exploratório

André Miranda¹, Renato Garcia^{1,2}, João Villa¹, Ana Oran³, Brenda S. Santana⁴
Gilleanes T. A. Guedes^{1,2}, 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

{¹andremiranda.aluno,¹joaovilla,^{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 contents are considered complex from the student's perspective. Chatbots are an emerging technology that has stood out as a pedagogical agent in teaching programming. In this context, we propose the CoderBot as an educational pedagogical agent based on Example-Based Learning. We designed CoderBot to help students understand programming content. We conducted an exploratory study with 103 undergraduate students in introductory programming courses to assess their self-efficacy and acceptance of CoderBot. We highlight the ease of use of CoderBot, improvements in understanding concepts, and the positive impact on students' motivation and self-confidence.*

Resumo. *Os conteúdos de programação são considerados complexos de ser aprendidos do ponto de vista dos estudantes. Um agente pedagógico que tem se destacado no ensino de programação são os chatbots. Neste sentido, desenvolvemos o CoderBot, um agente pedagógico educacional fundamentado na Aprendizagem Baseada em Exemplos, projetado para auxiliar estudantes iniciantes na compreensão de conteúdos de programação. Para avaliar a autoeficácia e aceitação do CoderBot, conduziu-se um estudo com 103 estudantes de graduação de disciplinas introdutórias de programação. Os resultados evidenciam a facilidade de uso do CoderBot, bem como melhorias na compreensão dos conceitos e melhora na motivação e autoconfiança dos estudantes.*

1. Introdução

Os conteúdos relacionados à programação de computadores apresentam um alto nível de complexidade tanto no ensino quanto na aprendizagem. Consequentemente, as taxas de aprovação em cursos da Computação são baixas [Alves et al. 2019], resultando

em altas taxas de retenção que podem implicar diretamente na evasão dos estudantes [Penney et al. 2023]. A dificuldade de ensinar esses conteúdos tem sido constantemente relatada por diversos professores que ministram disciplinas na área [Robins 2019]. Nesse cenário, uma importante tecnologia emergente de grande potencial para apoiar o ensino são os agentes conversacionais baseados em texto, *chatbots*. Os *chatbots* possibilitam que os estudantes avancem mais rapidamente no processo de aprendizagem, tirem dúvidas específicas [Clarizia et al. 2018, Ruan et al. 2019] e obtenham *feedback* imediato e adequado sem depender do suporte contínuo dos professores [Smutny e Schreiberova 2020].

Nesse contexto, o objetivo principal deste trabalho é a proposição de uma ferramenta educacional de apoio aos estudantes de graduação no ensino de conteúdos de programação por meio de um agente pedagógico denominado *CoderBot*. O propósito geral do *CoderBot* é proporcionar um ambiente de aprendizagem em que os estudantes possam praticar exercícios de programação. O *CoderBot* é fundamentado na Aprendizagem Baseada em Exemplos (ABE), pois sem o uso de esquemas ou estruturas adequadas, isto é, modelo instrucional embasado em teorias de aprendizagem reconhecidas, como a teoria da carga cognitiva e da aprendizagem, os estudantes podem enfrentar dificuldades na compreensão de novos conteúdos apresentados em sala de aula [Atkinson et al. 2000]. A fim de avaliar o *CoderBot* conduziu-se um estudo exploratório com os 103 estudantes de graduação em disciplinas introdutórias de programação, de diferentes regiões do Brasil. A avaliação foi conduzida por meio de um questionário com base nos indicadores do Modelo de Aceitação de Tecnologia (do inglês, *Technology Acceptance Model – TAM*) [Venkatesh e Davis 2000], e de um questionário para avaliar a *self-efficacy* (auto eficiência) percebida pelos estudantes no uso do *CoderBot* [Raiche et al. 2023].

2. Fundamentação Teórica e Trabalhos Relacionados

Diversos estudos têm investigado o uso de chatbots no apoio 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, denominado Pyo, projetado ajudar estudantes iniciantes no aprendizado da linguagem Python. Os autores avaliaram a eficácia do Pyo com 22 estudantes de um curso *online* de programação introdutória. Os resultados da avaliação indicaram que o Pyo facilitou a aprendizagem dos alunos, 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. A avaliação da Sara com 182 estudantes demonstrou melhorias na aprendizagem em tarefas de programação em comparação com agentes conversacionais tradicionais que não utilizam *scaffolding*. Nota-se que 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 sig-

nificativa [Große e Renkl 2007, 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]. A Aprendizagem Baseada em Exemplos, por sua vez, reduz a carga cognitiva ao demonstrar explicitamente as etapas necessárias para resolver um problema por meio de exemplos [Adams et al. 2014, McLaren et al. 2016].

O CoderBot combina de exemplos corretos e errôneos para enriquecer as concepções mentais dos estudantes, proporcionando uma experiência de aprendizado mais eficaz [Große e Renkl 2007, Adams et al. 2014, McLaren et al. 2016]. O CoderBot foi integrado a um site, possibilitando a apresentação de 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 1, mas é possível habilitar a exibição de outros conteúdos (Matrizes, Listas, Filas, entre outros) e o tipo de linguagem de programação para apresentar os exemplos.



Figura 1. CoderBot - Descrição do *Worked Example*

Essa customização é realizada pelo docente no painel administrativo do docentes antes do início da aula. 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 1). Após a leitura e análise dos termos apresentados (Figura 1), o estudante acessa 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 2).

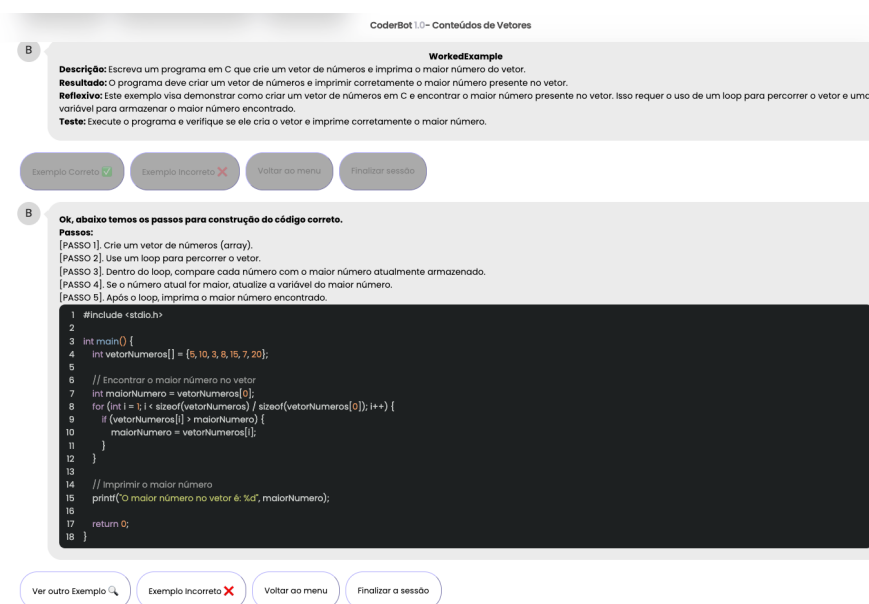


Figura 2. CoderBot - Worked Example - Correto

Caso o estudante opte por visualizar um **Exemplo Incorreto**, o CoderBot 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 3): (i) Se o estudante escolher a opção correta, ele é parabenizado e recebe um feedback explicativo sobre o erro e o código correto é apresentado; (ii) Se o estudante escolher a opção incorreta, ele recebe feedback informando que a alternativa escolhida está errada, com indicação do ponto específico do erro, e são apresentados os passos para a solução correta, seguidos pelo código correto. (iii) 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. 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 [Garces et al. 2023].

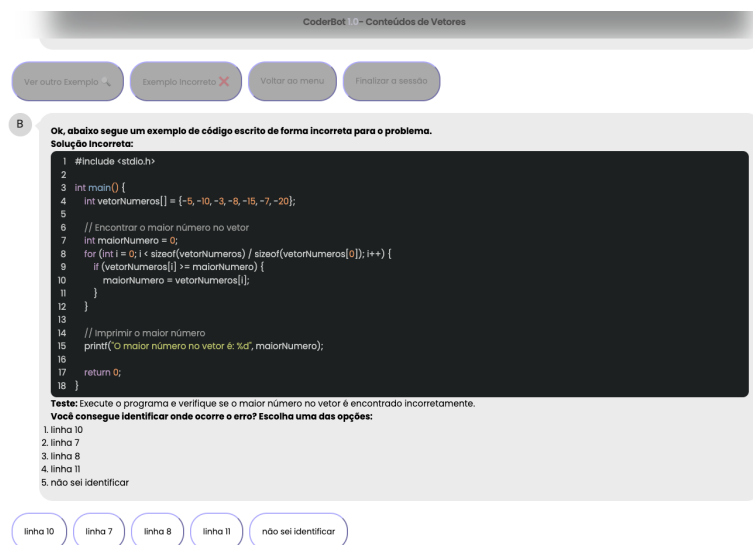


Figura 3. CoderBot - *Worked Example* - Incorreto

4. Estudo Exploratório

4.1. Planejamento do Experimento

O estudo exploratório foi realizado com o objetivo de compreender a percepção dos estudantes sobre CoderBot como uma ferramenta auxiliar no ensino de programação. 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	Semestre	Conteúdo	#Estudantes
IES 1	Ciência da Computação	D1	Algoritmos e Programação para Computação	1	Funções	8
IES 1	Engenharia de Software	D2	Algoritmos e Programação	1	Arrays	11
IES 2	Tecnologia em Análise e Desenvolvimento de Sistemas	D3	Programação de Computadores	2	Listas	15
IES 3	Tecnologia em Análise e Desenvolvimento de Sistemas	D4	Linguagem de Programação 1	2	Arrays	18
IES 4	Engenharia de Software	D5	Algoritmos e Estruturas de Dados 1	2	Funções	51

4.3. Execução do Experimento

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. Na data agendada, o estudo era realizado com os estudantes. 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 a conclusão das atividades, os estudantes responderam a questionários para medir a aceitação e auto-eficácia (*self-efficacy*) percebida após o uso do CoderBot (Tabela 2). O questionário de aceitação foi elaborado com base nos indicadores do Modelo de Aceitação de Tecnologia (TAM) [Venkatesh e Davis 2000] (ver Tabela 2). As respostas foram fornecidas em uma escala Likert de cinco pontos, variando de “Discordo totalmente a “Concordo totalmente, com opção neutra. O questionário de autoeficácia foi baseado nos indicadores propostos por Raiche *et al.*(2023) (ver Tabela 2). As respostas da autoeficácia foram fornecidas também uma escala de cinco pontos, cujas opções foram: Muito Insatisfeito, Insatisfeito, Neutro, Satisfeito e Muito Satisfeito.

Tabela 2. Questionário elaborado para coletar as percepções dos estudantes.

Afirmativas do Questionário TAM	Utilidade Percebida (grau em que o estudante acredita que o uso do CoderBot pode melhorar seu desempenho acadêmico)
	UP1 – Usar o CoderBot melhora o meu desempenho ao resolver problemas de programação.
	UP2 – Usar o CoderBot melhora a minha produtividade na resolução de exercícios.
	UP3 – O CoderBot facilita o meu trabalho durante o aprendizado de programação.
	UP4 – Eu acho o CoderBot útil para me auxiliar no aprendizado de programação.
	Facilidade de Uso Percebida (grau em que o estudante acredita que o uso do CoderBot pode ser feito sem muito esforço)
	FUP1 – O CoderBot foi claro e fácil de entender para mim.
	FUP2 – Usar o CoderBot não demandou muito esforço mental.
	FUP3 – Eu acho que o CoderBot é um chatbot fácil de usar.
	FUP4 – Eu acho fácil lembrar como executar as tarefas usando o CoderBot.
Perguntas do Questionário da Auto-eficácia	Intenção de Uso Percebida (grau em que o estudante acredita que poderá utilizar o CoderBot no futuro)
	IU1 – Assumindo que eu tenha acesso ao CoderBot, eu pretendo usá-lo futuramente para me apoiar no aprendizado de programação.
	IU2 – Dado que eu tenha acesso ao CoderBot, eu prevejo que eu o usaria futuramente no aprendizado de programação.
	IU3 – Eu pretendo usar o CoderBot para me auxiliar na resolução de exercícios no próximo mês.
	Satisfação (prazer percebido no uso e utilidade do CoderBot)
	S1 – O quanto você gostou de usar este chatbot?
	S2 – Quão útil foi o CoderBot para você no apoio à resolução dos exercícios?
	S3 – Como você avaliaria sua satisfação geral com o CoderBot?
	Usabilidade (facilidade percebida de uso do CoderBot),
	U1 – Quão fácil foi usar o CoderBot?
	U2 – Quão compreensíveis foram as respostas fornecidas pelo CoderBot?
	U3 – Quão aceitável é o tempo gasto interagindo com o CoderBot?
	Benevolência (percepção de que o CoderBot age com intencionalidade e precisão, considerando os interesses do usuário)
	B1 – Você sentiu que suas dúvidas foram corretamente compreendidas pelo CoderBot?
	B2 – Você sentiu que as respostas fornecidas pelo CoderBot foram claras?
	B3 – Você achou a interação com o CoderBot agradável?
	Credibilidade (percepção da habilidade e experiência do CoderBot)
	C1 – O CoderBot deve ser integrado nas práticas de ensino de programação?
	C2 – O CoderBot deve ser utilizado obrigatoriamente em aulas de programação?
	C3 – Você sentiu que o CoderBot era confiável?

5. Resultados

A Figura 4 apresenta os resultados das percepções dos estudantes avaliando o CoderBot utilizando o TAM. Os resultados relativos à Utilidade Percebida, evidenciou-se que 83%

dos estudantes concordaram que o CoderBot melhora o desempenho na resolução de problemas de programação (UP1). O estudante E34 disse que “o CoderBot ajudou, pois demonstrou exemplos de como resolver determinadas questões rápidas que poderiam levar horas tentando.” Além disso, 80% dos estudantes concordaram que o CoderBot pode ser útil no aprendizado de programação (UP4): “ajuda no aprendizado de programação, por trazer exemplos práticos que são vistos durante as aulas, fazendo com que o estudante entenda sobre a lógica de programação proposta pelos exercícios” – E83. Esses comentários indicam que o CoderBot foi bem recebido pelos estudantes como uma ferramenta útil para o ensino de programação, reforçando sua eficácia como suporte educacional.

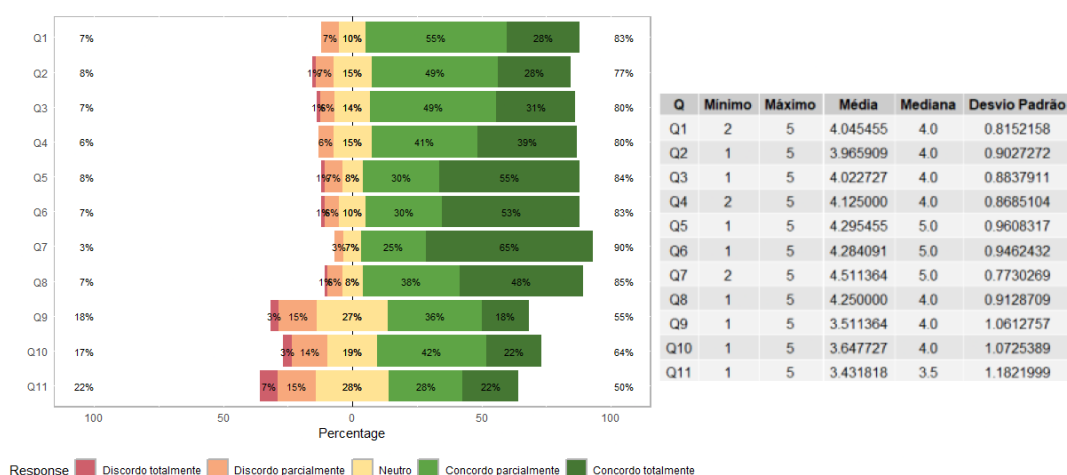


Figura 4. Aceitação dos estudantes.

A análise da Facilidade de uso evidenciou uma concordância acima de 85% neste indicador. No item FUP3, E17 afirmou: “o uso do Coderbot foi fácil, ajudando a resolver exercícios simples e, em alguns momentos, exercícios de dificuldade média”. Portanto, a inclusão do CoderBot no ensino não apresenta maiores dificuldades para os estudantes. Apesar de mais de 56% dos estudantes terem uma percepção positiva sobre a Intenção de Uso Percebida, uma quantidade razoável de estudantes permaneceu neutra (média de 24,7%) e discordando (média de 19%). E33 sugeriu: “para o intuito de aprendizado seria importante adicionar comentários nos códigos para auxiliar no entendimento do mesmo.” Esses dados indicam que, embora haja uma aceitação geral, há espaço para melhorias na funcionalidade e abrangência do CoderBot.

A Figura 5 apresenta os resultados da Autoeficácia percebidas pelos estudantes. No indicador de **Satisfação**, observou-se que a maioria dos estudantes (média de 68% dos estudantes) ficou satisfeita com o CoderBot. Por exemplo, E81 reforçou que ficou satisfeito, pois o CoderBot “ajudou em dar um norte para cada questão, o que economiza tempo, isso é ótimo”. Além disso, 68% dos estudantes concordaram que gostaram do CoderBot. O estudante E18 comentou “mesmo que o exercício não peça exatamente a função que tem no CoderBot, é fácil de utilizar como base para adaptar para o problema real da questão.” No que diz respeito à **Usabilidade**, este obteve uma média de concordância de 79%, a maior entre os indicadores de autoeficácia. O item U1 obteve concordância de 91% dos estudantes, demonstrando a facilidade de uso do CoderBot.

No indicador de **Benevolência**, a concordância positiva média foi de 65%. O item B3, que avaliou se a interação com o CoderBot foi agradável, foi o que obteve os melhores

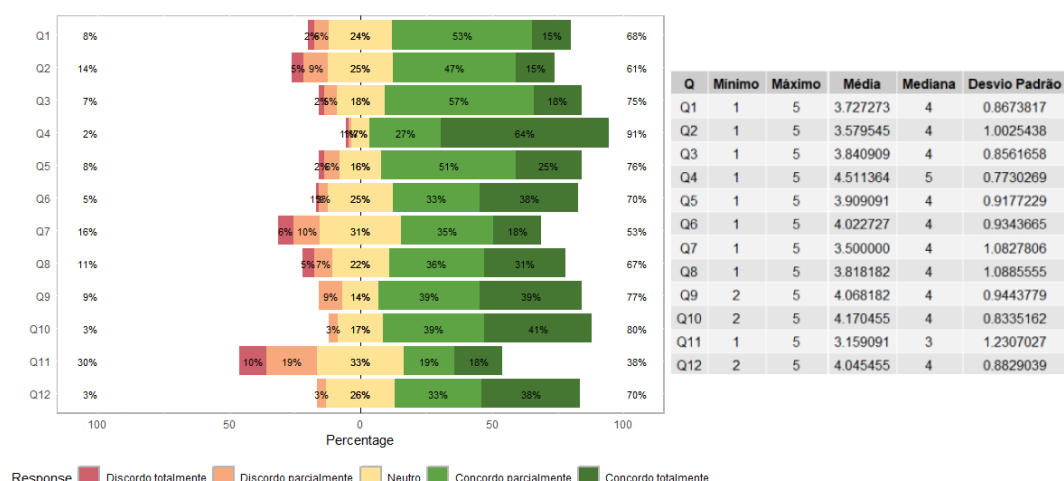


Figura 5. Autoeficácia percebida pelos estudantes.

resultados (77% de concordância). E23 afirmou: “*utilizando o CoderBot, consegui aprimorar minha compreensão de códigos de programação. Sua interface é intuitiva, simplifica a interação, tornando sua utilização livre de dificuldades.*” Por outro lado, o item B1, que avaliou se as dúvidas foram corretamente compreendidas, mostrou uma quantidade relevante de participantes permanecendo neutros (31% dos respondentes). E01 trouxe a seguinte evidência: “*por mais que o CoderBot tenha sido de muita ajuda, o seu estado atual é consideravelmente limitado, principalmente quanto à dúvidas mais complexas*”.

Por fim, no indicador **Credibilidade**, este foi o que apresentou menor taxa de concordância (média de 62,5%). O item C1, que questionava se o CoderBot deveria ser integrado às práticas de ensino de programação, teve respostas majoritariamente positivas, com 80% de concordância. O estudante E18 ressaltou que o CoderBot “*colaborou na resolução de alguns exercícios pela simplicidade na explicação e na estrutura dos exemplos.*” No entanto, o item C2 mostrou uma neutralidade significativa (33%). O estudante E48 afirmou que o CoderBot é “*muito compreensível e prático, porém, não explica o que iniciantes precisam saber, por isso discordo em utilizar somente ele para os conteúdos dados em aulas.*” Portanto, melhorias na explicação da lógica dos códigos seriam úteis.

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. Os resultados indicam que o CoderBot foi amplamente aceito pelos estudantes, sendo percebido como uma ferramenta útil e fácil de usar para o aprendizado de programação. Quanto a Autoeficácia Percebida, notou-se também um alta taxa de concordância nos indicadores de Satisfação e Usabilidade, sugerindo que o CoderBot cumpre seu papel de assistente educacional eficazmente. Por meio dos comentários, os estudantes destacaram que o *CoderBot* apresentava exemplos muito uniformes, sem uma variedade significativa na apresentação do conteúdo. Além disso, relatou a necessidade da inclusão de mais exemplos e a adição de comentários nos códigos. Essas melhorias podem aumentar a eficácia do CoderBot, tornando-o uma ferramenta ainda mais robusta para o ensino de programação. Como 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.

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.
- Atkinson, R. K., Derry, S. J., Renkl, A., e Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70(2):181–214.
- 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.
- Clarizia, F., Colace, F., Lombardi, M., Pascale, F., e Santaniello, D. (2018). Chatbot: An education support system for student. Em *International Symposium on Cyberspace Safety and Security*, páginas 291–302. Springer.
- Garces, S., Vieira, C., Ravai, G., e Magana, A. J. (2023). Engaging students in active exploration of programming worked examples. *Education and Information Technologies*, 28(3):2869–2886.
- Große, C. S. e 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.
- 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.
- 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. Springer.
- Raiche, A.-P., Dauphinais, L., Duval, M., De Luca, G., Rivest-Hénault, D., Vaughan, T., Proulx, C., e Guay, J.-P. (2023). Factors influencing acceptance and trust of chatbots in juvenile offenders’ risk assessment training. *Frontiers in Psychology*, 14:1184016.
- 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.

- Smutny, P. e Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for the facebook messenger. *Computers Education*, 151:103862.
- Sweller, J., Van Merriënboer, J. J., e Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, páginas 251–296.
- Venkatesh, V. e Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, 46(2):186–204.
- 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 *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, página 1–14, New York, NY, USA. Association for Computing Machinery.