C-BOT: Um protótipo de chatterbot para o ensino de programação

João Eduardo Seffrin Soares[1], **Larissa Astrogildo de Freitas**[1]

[1]Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPel) – Pelotas, RS – Brazil

{jessoares,larissa}@inf.ufpel.edu

Abstract. *One factor that explains students' difficulties in programming is the abstract nature of programming concepts. In addition, students need to adapt to the different syntaxes of programming languages and understand how do code snippets relate to each other. Therefore, alternative approaches to teaching programming are necessary, especially in the early stages of learning. With this context in mind, this work aims to create a chatterbot prototype to assist in teaching programming in C. The chatterbot prototype developed, called C-BOT, is capable of sharing knowledge about programming concepts and test students' knowledge about the functionality of some algorithms.*

Resumo. *A natureza abstrata dos conceitos de programação se mostra como um dos fatores que explicam as dificuldades que muitos estudantes encontram na absorção destes conteúdos. Além disso, os estudantes precisam se adaptar as diferentes sintaxes das linguagens de programação e entender como trechos de código se relacionam. Logo, abordagens alternativas de ensino de programação, principalmente em estágios iniciais, são necessárias. Tendo em vista este contexto, o objetivo deste trabalho é desenvolver um protótipo de chatterbot para auxiliar no ensino de programação na linguagem C. O protótipo de chatterbot desenvolvido, denominado C-BOT, é capaz de fornecer conhecimento sobre programação e testar os conhecimentos do usuário sobre o funcionamento de alguns algoritmos.*

1. Introdução

A crescente importância da área tecnológica em nossa sociedade demanda por profissionais qualificados para atuar em diversos espaços do mercado de trabalho [Technologies 2016], sendo a criação e a compreensão de códigos-fonte fundamental dentro desta qualificação. Em contrapartida, é frequentemente reportado que os cursos relacionados a área de tecnologia enfrentam altas taxas de desistências e reprovações, tanto no exterior [Eom and Ashill 2016], como no Brasil [Saccaro et al. 2019].

Um dos principais fatores por trás do alto índice de evasão está relacionado com a atividade de programar, geralmente precedido da presença de frustração e da falta de motivação com a programação em si [Bennedsen 2008], percebida pelos estudantes como complexa e de difícil aproximação [Daradoumis et al. 2016].

Jenkins [Jenkins 2002] descreve que diversos fatores podem influenciar negativamente o aprendizado da programação, podendo ser: o nível abstrato dos conceitos de programação; diferentes competências necessárias na resolução de problemas; pouco ou

DOI: 10.5753/sbie.2022.225711

nenhum contato anterior com o pensamento computacional ou a necessidade de aprender diversas sintaxes, semânticas e estruturas particulares de cada linguagem de programação.

Já, Hobert [Hobert 2019] destaca que é essencial que o aluno seja encorajado a praticar o ato de programar para sedimentar os conceitos aprendidos em aulas teóricas e práticas. O processo criativo de programar demanda muito dos estudantes, essa situação se agrava quando a orientação por parte do professor carece de alguma maneira, por exemplo: as restrições do tempo destes profissionais e consequentemente a impossibilidade de acompanhar de forma contínua o progresso individual de cada aluno.

Os trabalhos de [Crow et al. 2018] e [Sim and Lau 2018] destacam que uma gama de ferramentas que auxiliam o aprendizado da programação no contexto extracurricular existem na literatura, particularmente na forma de tutores virtuais. Essas ferramentas buscam trazer um nível de suporte aos alunos e também diminuir a quantidade de suporte necessário vindo dos professores. Porém, segundo os autores, esses sistemas não oferecem um nível de suporte considerado tão eficiente quanto o aprendizado em classe, com contato direto com o professor. Na literatura, uma parte desses sistemas são conhecidos como chatterbots [Hobert and von Wolff 2019].

Chatterbots geralmente buscam conduzir conversações de uma maneira proativa ou reativa, a partir da interação com o usuário [Hobert 2019]. A utilização de chatterbots em um contexto educacional abrange diversos espectros, como: suporte pedagógico [Clarizia et al. 2018] e sistemas de perguntas e respostas [Sinha and Basak 2020]. Chatterbots educacionais mostram-se efetivos em auxiliar no aprendizado de conteúdos difíceis [Clarizia et al. 2018], já que, além de possuirem a capacidade de adaptação a individualidades, eles permitem que os estudantes aprendam em seu próprio ritmo [Su et al. 2020] de forma ágil, motivadora e inspiradora [Ruan and Kun 2019].

O presente trabalho tem como objetivo apresentar o C-BOT, um protótipo de chatterbot capaz de auxiliar o aluno a programar na linguagem de programação C. Ele é capaz de compartilhar conhecimentos sobre fundamentos da programação e sobre a construção de alguns algoritmos. Para isso, foi modelada uma base de dados contendo algoritmos relevantes para a formação do aluno. Por fim, buscamos elaborar um modelo de conversas de entendimento simples para o auxílio do aprendizado em um contexto extracurricular.

Este documento é organizado da seguinte forma: a seção 2 descreve o referencial teórico; a seção 3 descreve os trabalhos relacionados; a seção 4 descreve o desenvolvimento do C-BOT; a seção 5 descreve os resultados obtidos; por fim, a seção 6 apresenta as considerações finais e os trabalhos futuros.

2. Referencial Teórico

Nesta seção os principais conceitos para o entendimento deste trabalho são descritos, são eles: Chatterbots e Linguagem de Programação C.

2.1. Chatterbots

Um chatterbot tem como objetivo conduzir conversações de maneira similar a seres humanos [Hobert 2019]. Esse objetivo é atingido através da aplicação de metodologias de diferentes áreas como Processamento de Língua Natural (PLN) ou Aprendizado de Máquina (AM). A entrada de um usuário na forma linguagem natural é processada por

um componente presente na arquitetura do chatterbot, após isso é computada uma resposta apropriada a partir de uma base de dados em que o chatterbot é treinado, assim o componente de geração de linguagem natural escolhe uma forma de saída que é mostrada ao usuário [Ramesh et al. 2017].

Atualmente, a evolução no campo de PLN e o crescimento do interesse comercial em chatterbots permitiu que esses espalhassem-se para diversos domínios, chatterbots modernos notáveis são assistentes virtuais, como Cortana[1] e Alexa[2] [Adamopoulou and Moussiades 2020]. Segundo [Hobert 2019] o interesse em chatterbots pedagógicos também tem se intensificado. Estudos relatam que este tipo de chatterbot cumpre o seu objetivo de auxiliar no ensino [Clarizia et al. 2018]. Outro fator positivo é que certos estudantes relatam que se sentem mais confortáveis ao interagir com um chatterbot em comparação a um amigo ou a um professor [Fryer and Carpenter 2006]. Chatterbots pedagógicos também mostram-se capazes de auxiliar professores em ensinar, ao engajar estudantes e simplificar certos passos do aprendizado [Okonkwo and Ade-Ibijola 2021].

No campo da computação, os chatterbots fornecem conteúdos para o aprendizado de programação básica ou fornecem uma maneira de testar conhecimentos, como quizzes [Hobert 2019], e buscam empenhar um papel similar ao de um professor, através de interações baseadas em linguagem natural.

O desenvolvimento de chatterbots geralmente se dá através de linguagens de programação como C, Java e Python, ou através da utilização de plataformas estado-da-arte. As plataformas permitem a construção de chatterbots de uma maneira rápida ao oferecer métodos predefinidos de PLN, além da integração com plataformas externas, facilitando a adição de recursos ao chatterbot [Xuan et al. 2018]. Elas geralmente possuem funcionalidades em comum como funcionamento na nuvem e a compatibilidade com diferentes linguagens de programação. Porém, podem divergir em certas características. São exemplos de plataformas para construção de chatterbots: Rasa[3] e Dialogflow[4].

2.2. Linguagem de Programação C

Linguagens de programação são notações para a construção de algoritmos que empenham certo tipo de tarefa. Uma linguagem de programação popular é a linguagem de programação C [5]. Ela é classificada como uma linguagem de programação estruturada pois utiliza estruturas de controle básicas em seu design. Linguagens de programação estruturadas são largamente adotadas no ensino de programação introdutória [Pears et al. 2009].

Dentro do escopo do C-BOT, estão os conceitos fundamentais tanto para a linguagem de programação C como para outras linguagens de programação, entre eles estão: variáveis, que armazenam valores na memória [Knuth 1997]; funções, que embalam uma sequência de código para uso pontual [Stroustrup 2013] e sua recursão, em que se chama uma função dentro de sua própria sequência de códigos, a fim de resolver problemas com

[1]https://www.microsoft.com/en-us/cortana
[2]https://developer.amazon.com/pt-BR/alexa
[3]https://rasa.com/
[4]https://dialogflow.cloud.google.com/
[5]https://www.tiobe.com/tiobe-index

execuções repetidas [Epp 2019]; estruturas de dados, que são variáveis ou dados organizados em um formato para acesso ou modificação facilitada [H and Ronald 2009]; árvores binárias, que são estruturas de dados organizados em níveis em que nodos são conectados entre si porém conectados a não mais que dois nodos em um nível inferior [Garnier 2009].

Os algoritmos relevantes construídos na linguagem C, presentes no quiz do C-BOT são: algoritmo de Busca Binária, que encontram a posição de um certo valor dentro de uma *array* já ordenada [Weisstein 2021]; algoritmo de Busca em Profundidade, que visita todos os nodos de uma árvore binária apenas uma vez e imprime seus valores na tela na forma que um nodo-pai (nível superior) nunca será impresso antes de seu nodos-filho (nível inferior) [Pfaff 2004]; algoritmo de Ordenação (*Quicksort*), que ordena uma *array* de forma numérica ao trocar valores de posição [Hoare 1962].

3. Trabalhos Relacionados

Nesta seção os trabalhos relacionados são apresentados. A partir de uma análise da literatura, alguns chatterbots com características similares ao protótipo de chatterbot proposto neste trabalho foram encontrados. De maneira geral, é de interesse destacar chatterbots que foram utilizados em um contexto de ensino de programação, tais como: [Hobert 2019], [Daud et al. 2020] e [Okonkwo and Ade-Ibijola 2021].

Hobert [Hobert 2019] apresenta o projeto de um sistema de ensino de programação inteligente, denominado Coding Tutor. O sistema é capaz de comunicar-se com o usuário através de um chatterbot e também conta com um editor de código, buscando auxiliar estudantes individualmente na construção de códigos, especialmente quando há a indisponibilidade de tutores humanos. O artigo apresenta as capacidades do chatterbot, ele é capaz de apresentar os conceitos teóricos por trás dos exercícios; corrigir os exercícios e dar *feedbacks* aos usuários. A metodologia de ensino empenhada pelo chatterbot se define pelos fluxos de conversação em exercícios, esses variam de acordo com a necessidade pelo número de explicações de cada estudante. Isso permite que estudantes com mais experiência não sejam submetidos a mesma concentração de conteúdo que estudantes inexperientes. Uma avaliação foi feita através de um questionário aplicado a 40 estudantes de um curso de programação introdutório com resultados favoráveis.

Daud et al.[Daud et al. 2020] propõem um chatterbot que tem como objetivo auxiliar o aprendizado de conceitos de programação em Java, mais precisamente, conceitos de seleção e repetição. O chatterbot seria capaz de prover explicações em relação aos conteúdos de programação a partir de perguntas dos estudantes pelo módulo de conversação do chatterbot. O chatterbot também é capaz de fornecer a visualização de códigos exemplo que utilizam conceitos de seleção (*if*, *else*) e repetição (*while*, *do while*, *for*). Os autores não tiveram a oportunidade de usar o chatterbot em um contexto prático e planejam melhorar o chatterbot futuramente.

Okonkwo e Ade-Ibijola [Okonkwo and Ade-Ibijola 2021] desenvolveram o Python-bot, o qual é capaz de explicar conceitos de programação a alunos, fornece problemas de programação predefinidos e permite ao usuário entrar em contato com um tutor humano caso necessário. Python-bot começa uma interação ao solicitar ao usuário que ele se apresente, com o objetivo de armazenar o nome do aluno como também o tempo total de cada conversa. Então o chatterbot fica a disposição do aluno para detalhar explicações conceituais de Programação de Python ou o passo-a-passo da construção de códigos. O

uso do chatterbot no aprendizado da linguagem Python foi testado por estudantes da universidade de Johanesburgo, a partir de uma avaliação de opinião do uso do chatterbot, cerca de 80% da turma se mostraram favoráveis ao uso dele.

Pode-se observar que os chatterbots descritos são similares em certos aspectos, por exemplo todos conseguem explicar alguma forma de conhecimento através de textos emitidos para o usuário. No caso de chatterbots voltados para o escopo da programação temos funcionalidades que exploram a introdução ao modo de programar códigos-fontes de maneiras diversas.

Essas funcionalidades são familiares com as funcionalidades oferecidas pelo C-BOT. Destacamos que nosso chatterbot difere dos apresentados em certas características. Ao contrário do Coding Tutor, o C-BOT não tem como objetivo agir diretamente no ato de programar, o foco dele é fornecer um ambiente de acesso rápido para que conhecimentos de programação sejam absorvidos de maneira casual em qualquer momento, dentro do ato de programar ou em outros momentos em que o estudante busca aprender ou testar seus conhecimentos. Ainda, ao contrário do Python-bot, o C-BOT oferece uma funcionalidade quiz que não apenas fala sobre funcionamento de um código como também permite o aprendizado em interação com o usuário na forma de perguntas e respostas.

4. Abordagem Proposta

Nesta seção detalhes referentes a abordagem proposta são explicados. Primeiramente será detalhado o funcionamento geral do C-BOT, logo após será apresentada a estrutura das conversas com C-BOT e por fim será apresentado o banco de dados de treinamento do C-BOT.

4.1. Funcionamento Geral do C-BOT

Para o desenvolvimento do protótipo de chatterbot, foi escolhido a framework Rasa. Rasa conta com um módulo de PLN, que extrai a intenção da mensagem do usuário, ("*Request*" e "*Interpreter*") [Bocklisch et al. 2017].

De forma geral, o funcionamento do protótipo de chatterbot começa no envio de uma mensagem de texto pelo aluno para uma entidade que representa o ele e que foi registrada no aplicativo. Esta entidade está conectada a uma máquina que hospeda o servidor responsável pela conexão. Esta máquina também contém a implementação do protótipo de chatterbot propriamente dita. Após estabelecida a conexão, a mensagem de texto é recebida pelo protótipo de chatterbot.

O próximo estágio do funcionamento é o processamento do texto. O protótipo de chatterbot possui em sua base de dados, exemplos de texto de entrada e seus respectivos grupos de intenção do texto. Assim, o módulo de PLN emite uma previsão sobre qual intenção o texto de entrada atual se encaixa, e o nível de confiança por trás desta previsão, numa escala de 0 a 1.

Após a intenção ser prevista, o framework Rasa deve agir de acordo com esta intenção, para isso, ele conta com um tipo de dado que dita um fluxo de conversação a partir de cada intenção ou das suas ramificações, as quais podem existir em cada fluxo, chamadas de "*stories*". Cada "*story*" possui a ordem de ações que o protótipo de chatterbot deve tomar dada uma combinação de intenções. Estas ações podem ser respostas em texto ou imagem. As imagens são enviadas pelo ele através de "*links*".

As ações possíveis pelo protótipo de chatterbot estão inseridas em seu banco de dados. Exitem tipos de ações já implementadas pelo framework Rasa, como por exemplo, a ação de enviar um trecho de texto como resposta para o aluno. Esta ação deve ser definida em um arquivo e inserida em uma "*story*" para ser executada no momento definido. Porém, ações personalizadas podem ser construídas pelo desenvolvedor através da linguagem de programação Python.

4.2. Estrutura de conversa do C-BOT

A estrutura das conversas com o C-BOT pode ser dividida em duas funcionalidades, são elas: (1) parágrafos sobre conceitos de programação; (2) aplicação de quizzes sobre o funcionamento de certos algoritmos. Ambas funcionalidades são brevemente introduzidas pelo protótipo de chatterbot após uma mensagem de saudação.

Todos os parágrafos sobre conceitos de programação e as imagens dos códigos-fonte utilizados pelo C-BOT foram extraídas do *website* "*GeeksforGeeks*"[6]. Este *website* é uma plataforma voltada para o ensino de conceitos de programação e de algoritmos.

No C-BOT optou-se por tratar sobre conceitos de programação em parágrafos em texto e introduzir o funcionamento de algoritmos em quizzes. Os conceitos de programação escolhidos para estarem no escopo deste protótipo foram: variáveis, operações aritméticas, recursão e estrutura de dados (árvore binária). Já os algoritmos escolhidos foram: Algoritmo de Busca Binária, Algoritmo de Busca em Profundidade e Algoritmo de Ordenação (*Quicksort*).

Ambas funcionalidades do protótipo de chatterbot solicitam do aluno algum tipo de requisição. A parte de conceitos em texto é geralmente feita pelo C-BOT a partir de uma pergunta direta do aluno, o texto é processado pelo módulo de PLN do C-BOT e a predição em relação a qual a intenção da mensagem é realizada, essa predição toma em conta as mensagens exemplo dentro da base de dados do treino do C-BOT, são exemplos: "Como funciona uma recursão?" ou apenas uma mensagem contendo a palavra "Recursão", mensagens que derivam das mensagens dentro da base de dados podem ser feitas, se estas se encaixam nos padrões estabelecidos nos banco de treinamento para uma intenção, esta mensagem será reconhecida. Em resposta o C-BOT envia um texto sobre a natureza de uma recursão em programação, a sua utilidade e exibirá uma imagem com um trecho de código que utiliza uma recursão.

Além da recursão, a versão atual de C-BOT também fala sobre o uso de variáveis, operações aritméticas e estruturas de dados (árvore binária) na linguagem de programação C. Algumas destas opções de conteúdos são introduzidas pelo C-BOT a partir de uma requisição. Espera-se do aluno o interesse em realizar perguntas sobre diversos assuntos, buscando tornar a busca de conhecimento sobre a linguagem de programação C a partir do uso do C-BOT mais instigante e não tão robótica, mesmo que o banco de dados atual do protótipo não possua um vasto escopo.

A parte de quiz é composta de perguntas e respostas. Um quiz é iniciado após a requisição do aluno, em seguida, o C-BOT fala as opções de quiz, existindo um para cada algoritmo (Busca Binária, Busca em Profundidade e Quicksort). A escolha é feita através de um texto de entrada do aluno e a partir daí o C-BOT descreve brevemente o algoritmo

[6]https://www.geeksforgeeks.org/

que vai ser trabalhado e exibe o algoritmo em uma imagem. Após, é perguntado ao aluno se o mesmo deseja começar o quiz. Caso o aluno concorde em começar o quiz, a execução do quiz se inicia. Caso o aluno não concorde em começar o quiz, a execução do quiz se encerra.

A estrutura da execução do quiz se dá por uma série de textos em etapas. Cada etapa é composta por uma pergunta que deve ser respondida corretamente para o prosseguimento do quiz. Cada pergunta refere-se a uma linha de código que empenha uma certa ação no código-fonte final. Portanto, cabe ao aluno completar as partes do algoritmo. A visualização do código-fonte final se dá por uma imagem que é enviada no começo do quiz. Nela, as linhas de código que integram o quiz estão incompletas, contudo, optamos por permitir a permanência de certos detalhes em cada linha (como: variáveis). Esse formato se assemelha a resolução de um quebra-cabeça e busca tornar a construção de um algoritmo mais flexível, trazendo foco para o "como" um algoritmo é construído, assim como estimulando a memória visual dos alunos ao fornecer a imagem de um código-fonte corretamente compilado a cada acerto.

4.3. Banco de Dados de Treinamento do C-BOT

Para que o C-BOT tenha a capacidade de realizar o papel de um tutor virtual de acordo com o formato descrito, um banco de dados de treinamento foi criado. O banco de dados encontra-se no formato ".yml".

No arquivo "nlu.yml" são inseridas todas intenções esperadas por parte do aluno em uma conversa juntos de seus respectivos exemplos de mensagens. Algumas intenções previstas pelo C-BOT são ações fundamentais à uma conversa, como uma saudação ou adeus. Outras, são ações personalizadas, como o interesse em aprender recursão ou o desejo de realizar um quiz sobre ordenação.

Nos arquivos "rules.yml" e "stories.yml" são inseridas regras e histórias, respectivamente. As regras permitem que o protótipo de chatterbot siga fluxos de conversação a partir de intenções do aluno, também ditam que tipo de resposta deve ser dada, em cada etapa de um determinado fluxo, ou seja, regras ditam fluxos que são imutáveis. Enquanto que, as histórias dão um fluxo generalizado permitindo variações ou mudanças entre fluxos, caso haja um desvio de intenção.

No arquivo "domain.yml" a estrutura e as formas de resposta são declaradas, o arquivo contém declarações dos seguintes componentes do protótipo de chatterbot: as respostas do dele para cada interação com o aluno; as estruturas dos quizzes, que são declarados como "forms", sendo formulários em que cada resposta é um "slot" ativado e preenchido pelo aluno; ações personalizadas que validam se um "slot" foi respondido corretamente; e as intenções no escopo do C-BOT para um texto de entrada do aluno.

No arquivo "actions.py" foram construídas as ações personalizadas de validação de cada resposta dos quizzes, elas analisam o texto de entrada do aluno e verificam se este condiz com a linha de código resposta.

Por fim, no arquivo "config.yml" são especificadas e personalizadas as políticas e componentes de PLN que serão utilizados pelo C-BOT no processamento de textos, este processo é denominado *pipeline*.

No C-BOT, foram utilizadas as políticas como: *RulePolicy* que indica o uso de re-

gras para fluxos fixos de conversação; *TEDPolicy* e *UnexpecTEDIntentPolicy* que auxilia na predição de ações do protótipo de chatterbot; *TED* (*Transformer Embedding Dialogue*) que utiliza *transformers* para atribuir diferentes pesos para partes de um texto e *MemoizationPolicy* que considera as histórias estabelecidas nos dados de treino para a predição de ações [Vaswani et al. 2017].

Também foram utilizados os componentes como: *WhitespaceTokenizer* que separa o texto por espaços em branco, unidades denominadas *tokens*; *CountVectorsFeaturizer* que agrupa os *tokens* a fim de classificar a intenção da mensagem, como também selecionar a resposta apropriada junto do componente *ResponseSelector*; *DIETClassifier* que classifica as intenções dada uma sequência de tokens; *FallbackClassifier* que customiza o nível de confiança que deve ser prevista para uma intenção.

5. Resultados Obtidos

O framework Rasa conta com funcionalidades de conexão com diversos canais de texto, entre eles estão: o Telegram, o Slack, o Messenger e também interfaces de chat construídas em HTML. Com o intuito de testar o uso do C-BOT por alunos de programação, optou-se por disponibilizar o protótipo de chatterbot através do Messenger levando em consideração a natureza acessível dele, em navegadores ou aplicativos de celular.

A integração do C-BOT com algum meio na internet se dá através de um servidor aberto em uma máquina local que faz a comunicação entre o sistema online e o protótipo de chatterbot. Abaixo alguns testes realizados e os resultados obtidos com o uso do C-BOT.

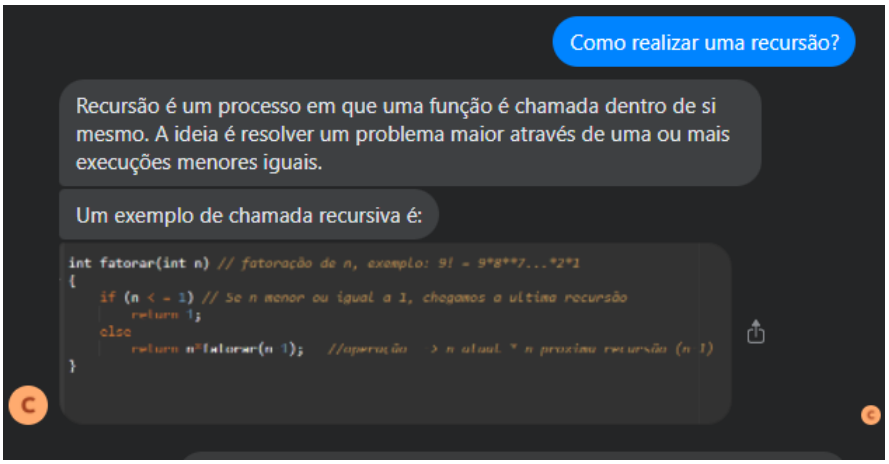

Figura 1. Texto sobre recursão. Fonte: Própria.

Na Figura 1, o aluno pergunta “Como realizar uma recursão?” e o C-BOT fornece um texto o sobre recursão em linguagem natural. Em seguida, ele apresenta um trecho de código ilustrando uma chamada recursiva. Neste trecho de código, uma função que realiza um fatorial de forma recursiva é mostrado. Um fatorial de um número inteiro positivo ‘n’ consiste em uma série de produtos para todos números inteiros positivos menores que ‘n’ [Kuhail et al. 2021].

Na Figura 2, o C-BOT diz que o primeiro passo a ser realizado no código-fonte fornecido é encontrar o número de posições no meio entre dois extremos. Para isso, deve-se fazer um cálculo utilizando uma fórmula contendo as posições já conhecidas, ou seja,

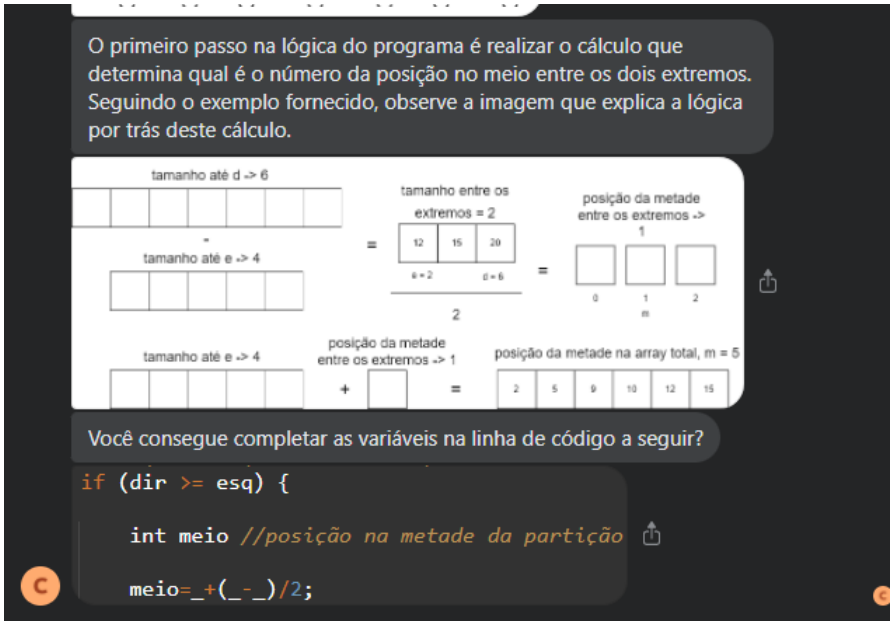

Figura 2. Texto da lógica do cálculo e parte de trecho de código oculto. Fonte: Própria.

o começo e o fim da partição. A imagem ilustra o uso das posições em cálculo, resultando na fórmula "e +(d-e)/2", sendo 'e' o limite inferior da partição e 'd' o limite superior da partição, logo, espera-se que o aluno insira a linha de código que processa essa operação aritmética corretamente, levando em conta seus conhecimentos sobre programação e as partes de trechos de código não ocultos.

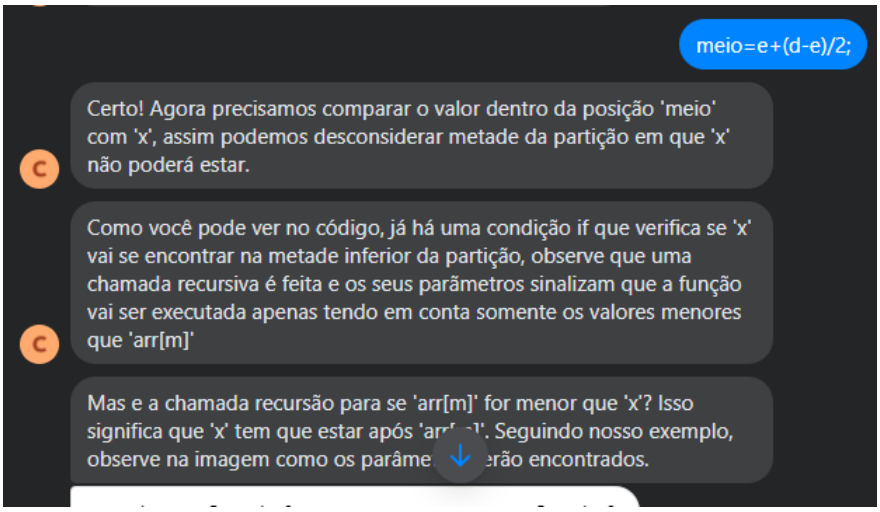

Figura 3. O aluno acertou parte de trecho de código oculto. Fonte: Própria.

Na Figura 3, o C-BOT mostra que após a linha de código ser inserida corretamente, o aluno recebe uma parabenização e o quiz prossegue para a próxima etapa.

6. Considerações Finais e Trabalhos Futuros

Neste trabalho, um protótipo de chatterbot para auxiliar o ensino de programação na linguagem C foi desenvolvido, o C-BOT. Assim como diversos chatterbots, o fluxo con-

versacional do C-BOT pode ser incrementado com novas funcionalidades que buscam auxiliar o ensino de programação, tais como: responder à perguntas mais específicas referentes a um certo conteúdo geral, customizando o aprendizado de cada conteúdo a partir das necessidades do aluno e adaptar o fluxo de cada conversa ao registrar que dificuldades o aluno tem em um quiz, sugerindo textos referentes aos conceitos que o aluno tem mais dificuldade. A implementação do C-BOT está disponível no repositório GitHub *omitted due to blind review*.

Como trabalhos futuros pretendemos integrar o protótipo de chatterbot com outras plataformas de comunicação, em sua versão atual, C-BOT foi testado apenas no aplicativo de comunicação Messenger. Ainda, a aplicação do C-BOT em um curso/disciplina de programação introdutório é essencial para o refinamento do formato de conversa dele. Assim como foi feito nos trabalhos propostos por [Hobert 2019] e [Okonkwo and Ade-Ibijola 2021], os quais avaliaram os chatterbots aplicando questionários de satisfação nas turmas que fizeram uso deste recurso.

Referências

Adamopoulou, E. and Moussiades, L. (2020). An overview of chatbot technology. *Artificial Intelligence Applications and Innovations*, pages 1–3.

Bennedsen, J. (2008). *Teaching and learning introductory programming: a model-based approach*. Editora desconhecida, Denmark.

Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *NIPS 2017 Conversational AI workshop*, pages 1–8.

Clarizia, F., Colace, F., Lombardi, M., Pascale, F., and Santaniello, D. (2018). An education support system for student. in international symposium on cyberspace safety and security. *Cyberspace Safety and Security*, pages 2–3.

Crow, T., Luxton-Reilly, A., and Wuensche, B. (2018). Intelligent tutoring systems for programming education: a systematic review. *Australasian Computing Education Conference*, pages 1–3.

Daradoumis, T., Puig, J. M. M., Arguedas, M., and Liñan, L. C. (2016). *Analyzing students perceptions to improve the design of an automated assessment tool in online distributed programming*. Computers and Education, Spain.

Daud, S. H. M., Teo, N. H. I., and Zain, N. H. M. (2020). e-java chatbot for learning programming language: A post-pandemic alternative virtual tutor. *International Journal of Emerging Trends in Engineering Research*, pages 1–6.

Eom, S. and Ashill, N. (2016). The determinants of students' perceived learning outcomes and satisfaction in university online education: An update. *Decision Sciences Journal of Innovative Education*, pages 3–5.

Epp, S. (2019). *Discrete Mathematics with Applications*. ISBN, USA.

Fryer, L. and Carpenter, R. (2006). Bots as language learning tools. *Language, Learning and Technology*, pages 1–4.

Garnier, R. (2009). *Discrete Mathematics:Proofs, Structures and Applications*. CRC Press, USA.

H, H. T. and Ronald, L. (2009). *Introduction to Algorithms*. The MIT Press, USA.

Hoare, C. A. R. (1962). Quicksort. *The Computer Journal*, pages 1–8.

Hobert, S. (2019). Say hello to 'coding tutor'! design and evaluation of a chatbot-based learning system supporting students to learn to program. *Fortieth International Conference on Information Systems*, pages 1–7.

Hobert, S. and von Wolff, R. M. (2019). Say hello to your new automated tutor – a structured literature review on pedagogical conversational agents. *14th International Conference on Wirtschafts informatik*, pages 1–7.

Jenkins, T. (2002). *On the Difficulty of Learning to Program*. Computers and Education, Massachusetts.

Knuth, D. (1997). *The Art of Computer Programming*. American Scientist, USA.

Kuhail, A., Negreiros, M., and Seffah, J. (2021). Teaching recursive thinking using unplugged activities. *Jornal desconhecido*, pages 2–3.

Okonkwo, C. W. and Ade-Ibijola, A. (2021). Python-bot: A chatbot for teaching python programming. *Engineering Letters*, pages 1–6.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., and Paterson, J. (2009). A survey of literature on the teaching of introductory programming. *ACM SIGCSE*, pages 1–2.

Pfaff, B. (2004). *An Introduction to Binary Search Trees and Balanced Trees*. Free Software Foundation, Inc., USA.

Ramesh, K., Ravishankaran, S., Joshi, A., and Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. *Communications in Computer and Information Science*, pages 1–2.

Ruan, S. and Kun, B. J. (2019). Quizbot: A dialogue-based adaptive learning system for factual knowledge. *CHI Conference on Human Factors in Computing Systems*, pages 2–3.

Saccaro, A., França, M. T. A., and Jacinto, P. A. (2019). Fatores associados à evasão no ensino superior brasileiro: um estudo de análise de sobrevivência para os cursos das áreas de ciência, matemática e computação e de engenharia, produção e construção em instituições públicas e privadas. *Estudos Econômicos (São Paulo)*, pages 1–3.

Sim, T. Y. and Lau, S. L. (2018). Online tools to support novice programming: A systematic review. *IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pages 2–3.

Sinha, S. and Basak, S. (2020). An educational chatbot for answering queries. in emerging technology in modelling and graphics. *Springer*, pages 1–5.

Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley, Germany.

Su, M., Wu, C., and Wang, H. (2020). A chatbot using lstm-based multi-layer embedding for elderly care. *2017 International Conference on Orange Technologies*, pages 1–2.

Technologies, B. G. (2016). Beyond point and click: The expanding demand for coding skills.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Cornell University*, pages 1–15.

Weisstein, E. (2021). *Binary search*. MathWorld, USA.

Xuan, P., Pham, T., Quynh, N., Thanh, N., and Thi, C. (2018). Chatbot as an intelligent personal assistant for mobile language learning. *ICEEL 2018*, pages 1–9.