



C-BOT: um chatterbot para o ensino da programação

João Eduardo Seffrin Soares

Orientadora: Larissa A. de Freitas

Sumário

1. Introdução
2. Referencial Teórico
3. Trabalhos Relacionados
4. Chatterbot Proposto
5. Resultados Obtidos
6. Considerações Finais
7. Trabalhos Futuros
8. Referências

Introdução

Problemática: Taxa alta de evasão em cursos da computação.

- Brasil: 27,47% nos dois semestres de 2009 em turmas de ingressantes (A. SACCARO, 2019).
- Taxas elevadas também foram reportadas em outras partes do mundo (N. ASHILL, 2016).

Em contrapartida...

Introdução

Segundo pesquisa da empresa Burning Glass Technologies (BURNING GLASS, 2021):

- Em 2015, nos EUA, sete milhões de vagas de emprego que necessitavam de habilidades de programação surgiram.
- A média de empregos na área de programação pode chegar a 50% a mais do que em outras áreas do mercado.

Evasão X Carência no Mercado

Introdução

Possível fator: A programação em si.

- Programação é geralmente considerada de difícil aproximação (BENNEDNSEN, 2008).
- Os fatores que podem dificultar o aprendizado são:
 - Pouco contato anterior com conceitos e práticas.
 - **Conceitos abstratos!**
 - **Diferentes sintaxes e semânticas para cada funcionalidade em cada linguagem!**

(JENKINS, 2002)

Introdução

Segundo (HOBERT, 2019), na estrutura de ensino:

- Além de aulas teóricas e práticas, é essencial o contato extraclasse do aluno com a programação.
- Dificuldades já existentes podem ser agravadas pela carência do contato com um tutor.

Introdução

- Ferramentas que ensinam programação buscam auxiliar o professor e suprir eventuais carências (CROW, 2018).
- Chatterbots educacionais
 - Se mostram efetivos em auxiliar estudantes em conteúdos difíceis (CLARIZIA, 2018).
 - Permitem que estudantes aprendam no seu próprio ritmo (M. SU, 2020).
 - Acessíveis, ágeis e flexíveis (B. J. KUN, 2019).

Introdução

Presente trabalho: Chatterbot que ensina programação

- Disserta sobre aspectos teóricos e práticos de programação.
- “Quiz”.

Referencial Teórico

- Processamento de Língua Natural (PLN)
 - Área de conhecimento de Inteligência Artificial que engloba técnicas de manipulação e processamento de texto ou fala (L. MOUSSIADES, 2020).

Referencial Teórico

- Chatterbots
 - Segundo (K. RAMESH, 2017), chatterbots:
 - Buscam simular uma conversa humana
 - Utilizam técnicas de PLN e/ou AM
 - Segundo (HOBERT, 2019), chatterbots na computação:
 - Simulam o papel de um tutor
 - Utilizam explicações e quizzes
 - Exemplos de plataformas de desenvolvimento de chatterbots:
 - Dialogflow¹ e Rasa²

1 <https://rasa.com/>

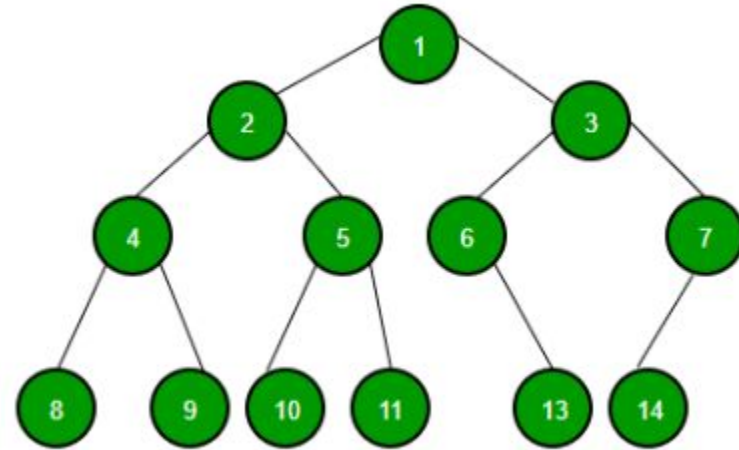
2 <https://dialogflow.cloud.google.com/>

Referencial Teórico

- Linguagem de Programação C
 - Segundo (A. PEARS, 2009), a linguagem de programação C:
 - É largamente adotada em cursos introdutórios de programação.
 - Exemplos de características da linguagem de programação C:
 - Imperativa
 - Linguagem estruturada

Referencial Teórico

- Linguagem de Programação C
 - Conceitos de Programação
 - Variáveis
 - Recursões
 - Estruturas de Dados
 - Árvores Binárias, Grafos, etc.

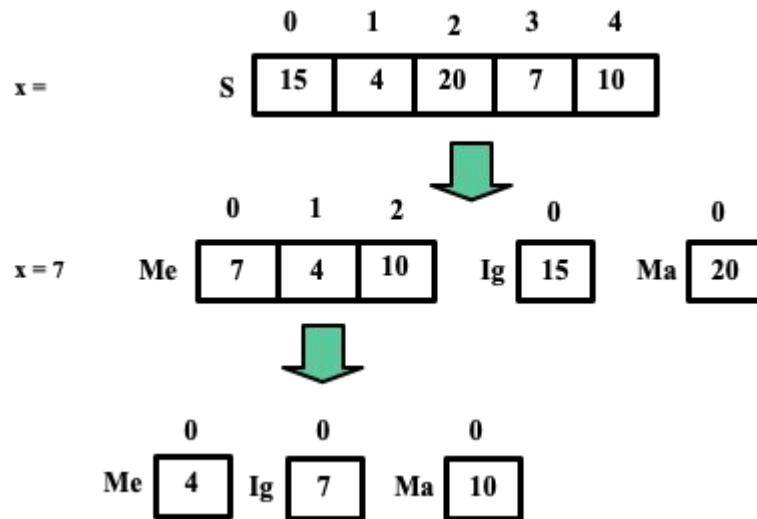


Fonte¹

¹ <https://www.geeksforgeeks.org/>

Referencial Teórico

- Linguagem de Programação C
 - Algoritmos Populares
 - Busca Binária
 - Busca em Profundidade
 - Algoritmos de Ordenação
 - Quicksort, Bubble Sort, etc.



Fonte¹

¹ <http://www.universidadejava.com.br/>

Trabalhos Relacionados

- Coding Tutor (HOBERT, 2019)
 - Editor de código + Chatterbot
 - Aplica exercícios em etapas
 - Avaliação positiva em classe
- e-Java (S.H.M. DAUD, 2020)
 - Perguntas e respostas sobre conceitos de seleção e repetição
 - Fornece exemplos de códigos-fonte
- Python-bot (A. ADE-IBIJOLA, 2021)
 - Explica conceitos de programação
 - Corrige código-fonte
 - Grande índice de aceitação em classe

Chatterbot Proposto

- C-BOT
 - Framework: Rasa
 - Funcionamento Geral:
 - Interação através do app Messenger¹
 - Mensagem do Usuário -> Servidor -> Modelo em máquina local
 - A intenção da mensagem é extraída
 - Escolhe qual fluxo de conversação seguir
 - Uma ação em resposta é tomada

¹ <https://www.messenger.com/>

Chatterbot Proposto

- Metodologia de ensino
 - Duas funcionalidades principais:
 - Explicações
 - Quiz
 - Códigos-fonte do website [geeksforgeeks](https://www.geeksforgeeks.org/)¹
 - Usuário acessa as funcionalidades através de uma requisição

¹ <https://www.geeksforgeeks.org/>

Chatterbot Proposto

- Metodologia de ensino
 - **Explicações** de conceitos de programação
 - Variáveis
 - Impressão
 - Operações aritméticas
 - Recursão
 - Estrutura de Dados (árvores binárias)
 - Exemplo de requisição: “Como realizar uma recursão?”
 - Uso de imagem (código-fonte) e texto

Chatterbot Proposto

- Metodologia de ensino
 - **Quiz** sobre algoritmos populares
 - Busca Binária
 - Busca em Profundidade
 - Algoritmo de Ordenação (Quicksort)
 - Divide a lógica do algoritmo em etapas
 - Uma trecho de código-fonte incompleto deve ser preenchido a cada etapa
 - Um quiz pode ser pausado e retomado em qualquer ponto
 - Um quiz precisa de uma confirmação inicial após a introdução geral do algoritmo

Chatterbot Proposto

- Banco de dados de treino
 - **nlu.yml**
 - Intenções definem qual fluxo de conversação será tomado

```
nlu:  
- intent: intro  
  examples: |  
    - oi  
    - olá  
    - bom dia  
    - boa tarde  
    - boa noite  
    - Vamos lá  
    - Eai cara  
    - oii  
  
- intent: adeus  
  examples: |  
    - tchau  
    - adeus  
    - até mais  
    - até mais tarde  
    - tenha um bom dia  
    - bye bye  
    - não obrigado  
  
- intent: exercicio  
  examples: |  
    - exercicio  
    - quero fazer exercicio  
    - quais exercicios posso fazer?  
    - exercicios  
    - quero fazer um quiz  
    - quiz  
    - fazer quiz  
    - fazer exercicio
```

Chatterbot Proposto

- Banco de dados de treino
 - **stories.yml**
 - Histórias definem os caminhos (fluxos) de conversação a partir das intenções
 - Permite variações e mudanças entre histórias

```
- story: tree
  steps:
    - intent: tree
    - action : utter_tree
    - action : utter_tree_dois
    - action : utter_continue
    - intent : afirmação
    - action : utter_tree_tres

- story: tree deny
  steps:
    - intent: tree
    - action : utter_tree
    - action : utter_tree_dois
    - action : utter_continue
    - intent : negação
    - action : utter_okay

- story: Activate binary search form
  steps:
    - intent: exercicio
    - action: utter_exes
    - intent: bs_exe
    - action: utter_bs_exe
    - action: utter_bs_exe_um_um
    - action: utter_bs_exe_um_dois
    - action: utter_bs_exe_dois
    - action: utter_exe_ask
    - action: utter_exe_ask_dois
    - intent: afirmação
    - action: utter_bs_exe_tres
    - action: utter_bs_exe_quatro
    - action: bs form
```

Chatterbot Proposto

- Banco de dados de treino
 - **rules.yml**
 - Não permite variações e mudanças entre histórias

rules:

- rule: ajuda bfs
condition:
 - active_loop: bs_formsteps:
 - intent: ajuda
 - action: action_deactivate_loop
 - active_loop: null
 - action: utter_exe_pause
- rule: ajuda dfs
condition:
 - active_loop: dfs_formsteps:
 - intent: ajuda
 - action: action_deactivate_loop
 - active_loop: null
 - action: utter_exe_pause
- rule: ajuda qs
condition:
 - active_loop: qs_formsteps:
 - intent: ajuda
 - action: action_deactivate_loop
 - active_loop: null
 - action: utter_exe_pause

Chatterbot Proposto

- Banco de dados de treino
 - **domain.yml**
 - Ações são declaradas e definidas
 - Quizzes e suas etapas são definidas
 - Intenções são definidas

```
intents:  
  - intro  
  - adeus  
  - afirmação  
  - negação  
  - declara  
  - condicional  
  - loop  
  - ajuda  
  - teste  
  - dica  
  - challenge  
  - exercicio  
  - bs_exe  
  - dfs_exe  
  - qs_exe  
  - operar  
  - recursar  
  - print  
  - tree  
  - resume  
  - code  
  - unknown
```

```
responses:  
  utter_intro:  
    - text: "Olá! Eu sou C-BOT e posso lhe ensinar sobre alguns conceitos"  
  
  utter_intro_dois:  
    - text: "Você deseja saber detalhes sobre o que posso oferecer?"
```

Chatterbot Proposto

- Banco de dados de treino
 - **actions.py**
 - Ações personalizadas são implementas
 - Linguagem de implementação: Python
 - Validação de cada resposta no quiz

```
class ValidateBSForm(FormValidationAction):
    def name(self) -> Text:
        return "validate_bs_form"
    @staticmethod
    def bs_um_db() -> List[Text]:
        return ["meio=e+(d-ea)/2;", "meio=e+(d-e)/2;"]
    @staticmethod
    def bs_dois_db() -> List[Text]:
        return ["return buscaBinaria(arr,meio+1,dir,x);", "buscaBinaria(arr,meio-1,dir,x);"]
    def validate_bs_um(
        self,
        slot_value: Any,
        dispatcher: CollectingDispatcher,
        tracker: Tracker,
        domain: DomainDict,
    ) -> Dict[Text, Any]:
        if slot_value in self.bs_um_db():
            dispatcher.utter_message(text="Certo! Agora p")
            dispatcher.utter_message(text="Como você pode")
            dispatcher.utter_message(text="Mas e a chamad")
            dispatcher.utter_message(image="https://i.img")
            return {"bs_um": slot_value}
        else:
            dispatcher.utter_message(text="Tente de novo")
            return {"bs_um": None}
```

Chatterbot Proposto

- Banco de dados de treino
 - **config.yml**
 - Componentes PLN utilizados no treinamento
 - Políticas:
 - MemoizationPolicy
 - RulePolicy
 - TEDPolicy
 - UnexpectedTEDIntentPolicy
 - Componentes:
 - WhitespaceTokenizer
 - CountVectorsFeaturizer
 - ResponseSelector
 - EntitySynonymMapper
 - FallbackClassifier

pipeline:

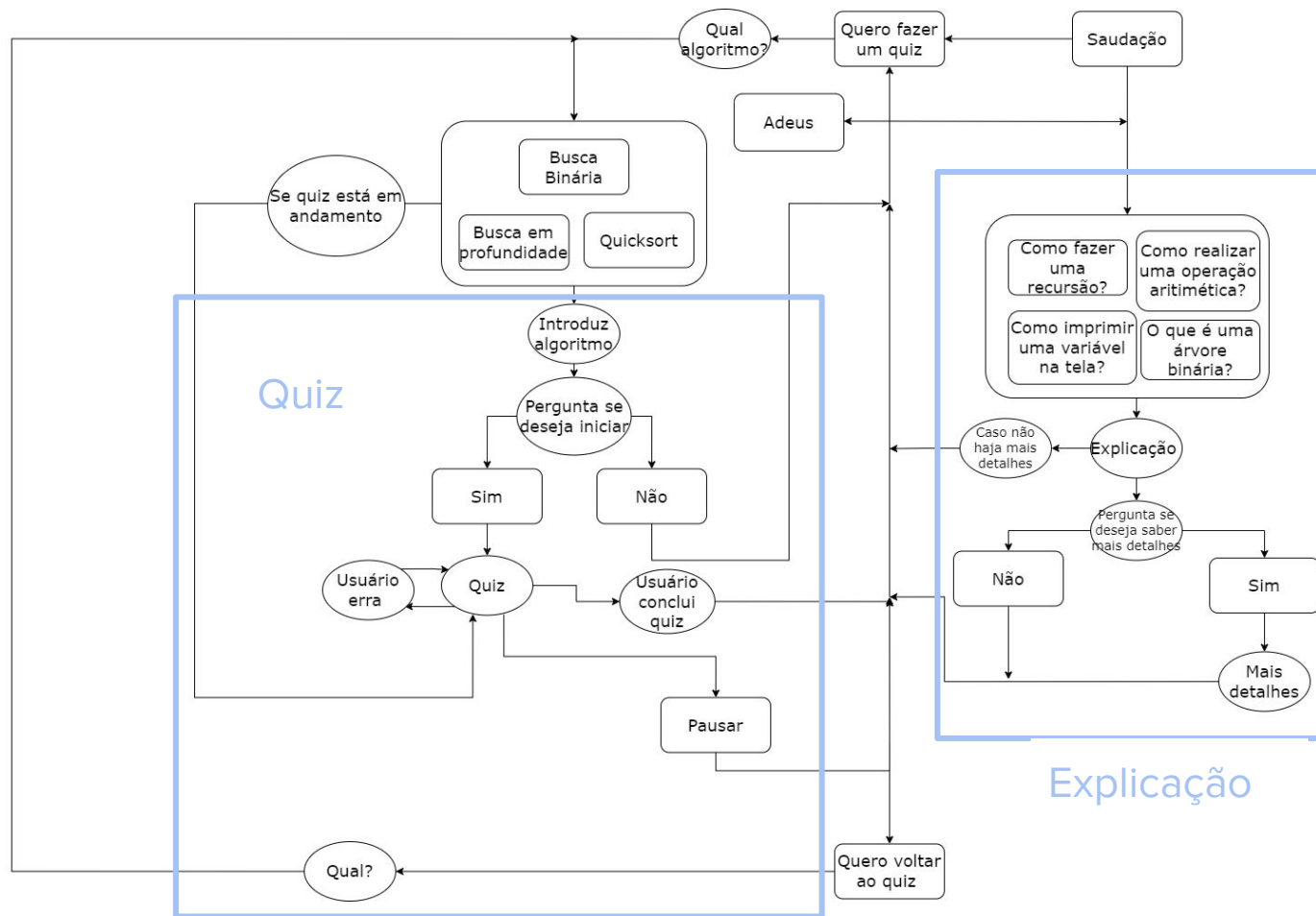
```
- name: WhitespaceTokenizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100
  constrain_similarities: true
- name: EntitySynonymMapper
- name: ResponseSelector
  epochs: 100
  constrain_similarities: true
- name: FallbackClassifier
  threshold: 0.91
  ambiguity_threshold: 0.1
```

policies:

```
- name: MemoizationPolicy
  max_history: 2
- name: RulePolicy
- name: UnexpectedTEDIntentPolicy
  max_history: 5
  epochs: 100
- name: TEDPolicy
  max_history: 5
  epochs: 100
  constrain_similarities: true
```


Chatterbot Proposto

- Fluxo de conversação



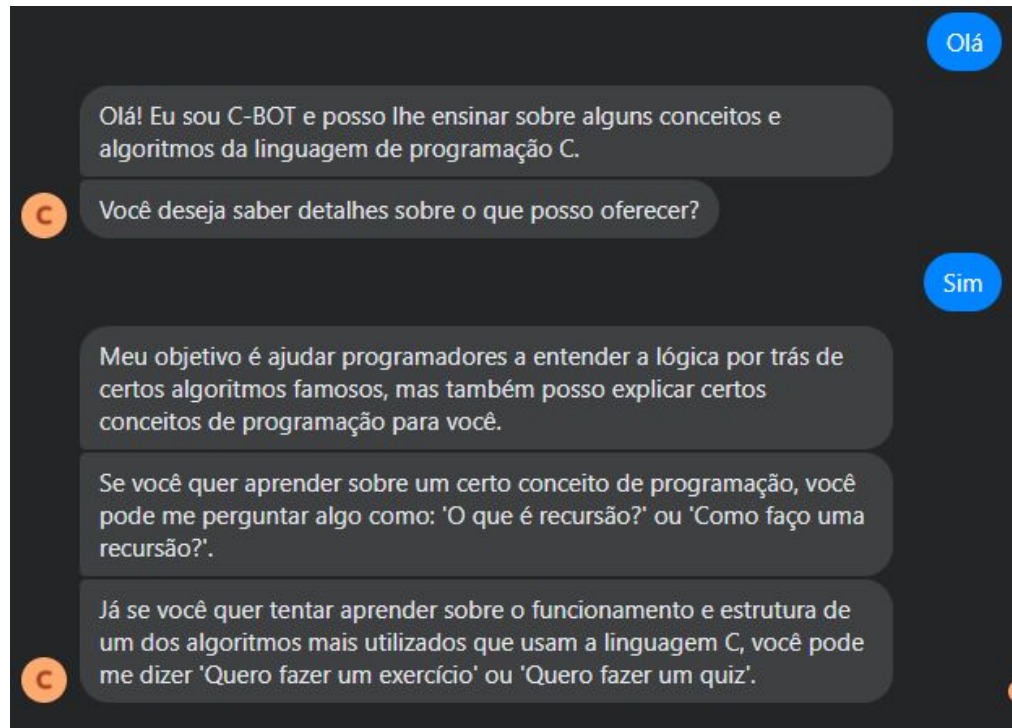
Resultados Obtidos

- Conversação através da plataforma Messenger
- Servidor é aberto através de uma máquina local
 - Utiliza aplicativo de terceiros, ngrok¹
 - Limite de duas horas

¹ <https://ngrok.com/>

Resultados Obtidos

- Saudação
 - Explicação sobre o funcionamento do chatterbot mediante confirmação.



Resultados Obtidos

- Explicação
 - Explicação conceitual sobre recursão.
 - Exemplo de código-fonte (Fatorial).

Como realizar uma recursão?

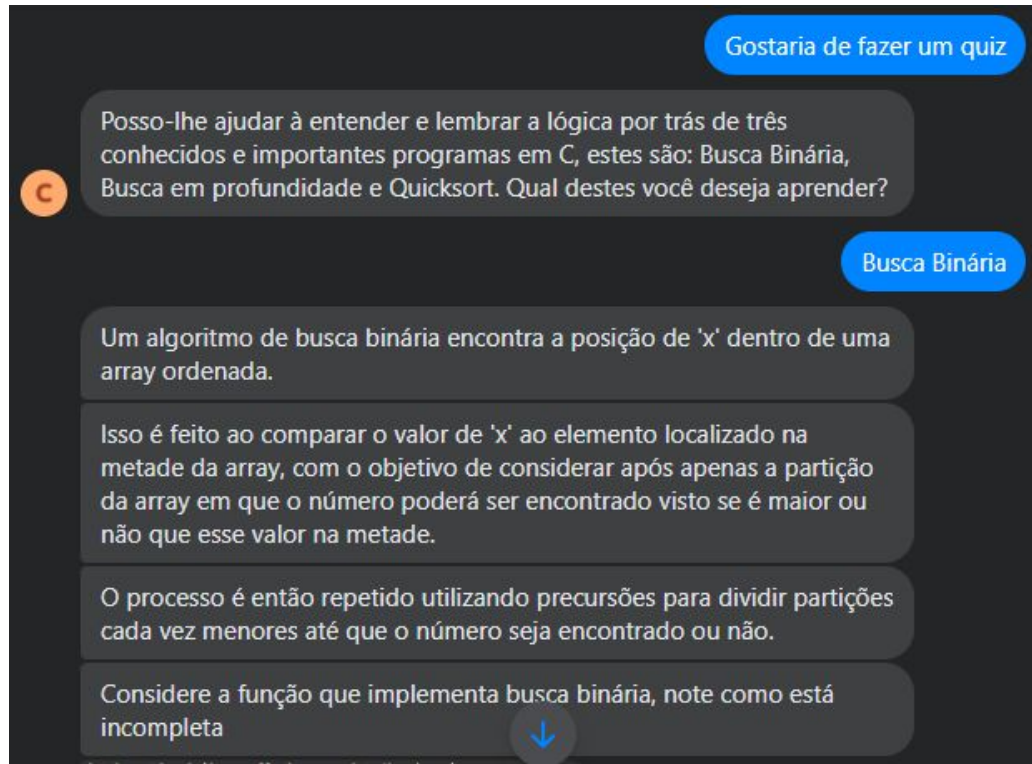
Recursão é um processo em que uma função é chamada dentro de si mesmo. A ideia é resolver um problema maior através de uma ou mais execuções menores iguais.

Um exemplo de chamada recursiva é:

```
int fatorial(int n) // fatorial de n, exemplo: 9! = 9*8*7...*2*1
{
    if (n <= 1) // Se n menor ou igual a 1, chegamos a última recursão
        return 1;
    else
        return n*fatorial(n-1); // operação > n atual * n proxima recursão (n-1)
}
```

Resultados Obtidos

- Quiz (Opções)
 - Usuário deve escolher entre opções:
 - Busca Binária
 - Busca em Profundidade
 - Ordenação (Quicksort)
 - Chatterbot introduz o algoritmo de busca binária.



Resultados Obtidos

- Quiz
 - Código-fonte incompleto sobre busca binária é apresentado.
 - Chatterbot pede confirmação para começar o quiz.

Considere a função que implementa busca binária, note como está incompleta

```
int buscaBinaria(int arr[], int esq, int dir, int x)
{
    //If que verifica se o extremo direito e esquerdo não se encontram
    // o que diz que a busca na partição terminou.
    if (dir >= esq) {

        int meio //posição na metade da partição

        meio = (esq + dir) / 2;
        //Encontrou x se n
        if (arr[meio] == x)
            return meio;

        if (arr[meio] > x)
            return buscaBinaria(arr, esq, meio - 1, x);

        return buscaBinaria(arr, meio + 1, dir, x);
    }
    //Retorna -1 caso o valor X não seja encontrado
    return -1;
}
```

Você gostaria de realizar esse quiz? Lembre que você deve reescrever a linha inteira em cada resposta!

Você também pode pausar o quiz para me perguntar qualquer coisa se assim desejar, basta me pedir isso durante o quiz.

Resultados Obtidos

```
int buscaBinaria(int arr[], int esq, int dir, int x)
{
    //If que verifica se o extremo direito e esquerdo não se encontram
    // o que diz que a busca na partição terminou.
    if (dir >= esq) {

        int meio //posição na metade da partição

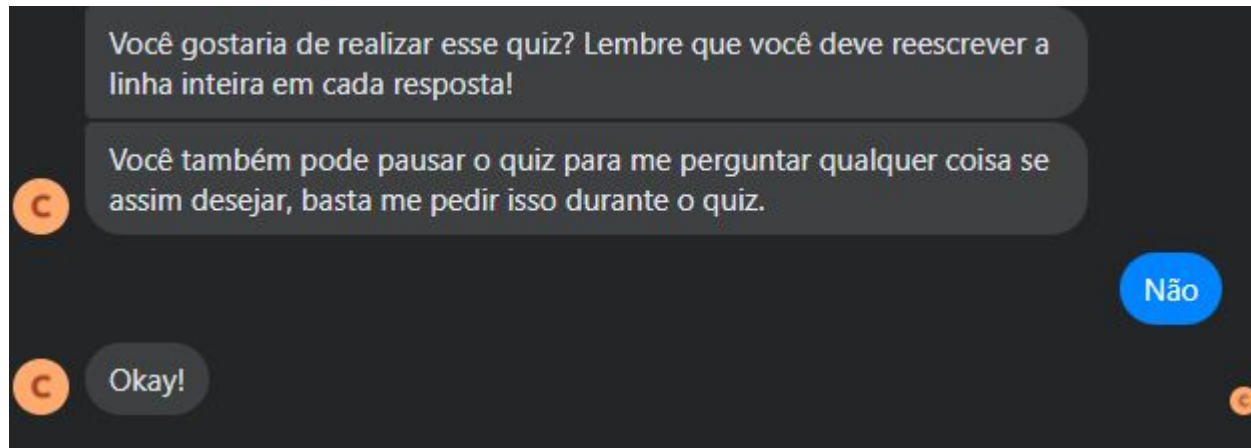
        meio=_( _- _)/2;
        //Encontrou x em m
        if (arr[meio] == x)
            return meio;

        if (arr[meio] > x)
            return buscaBinaria(arr, esq, meio - 1, x);

        return buscaBinaria(arr, __ + 1, __, x);
    }
    //Retorna -1 caso o valor X não seja encontrado
    return -1;
}
```

Resultados Obtidos

- Quiz
 - Confirmação é negada.



Resultados Obtidos

- Quiz
 - Confirmação é afirmada
 - Chatterbot introduz exemplo de busca binária na forma de texto e imagem.

C Você também pode pausar o quiz para me perguntar qualquer coisa se assim desejar, basta me pedir isso durante o quiz.

Sim

C Para ajudar entender a lógica da busca binária considere este exemplo: Buscamos encontrar o número 20. Observe como já estamos apenas considerando a metade superior da array visto que 20 é maior que o elemento na metade da chamada anterior (10), logo, os extremos da partição atual serão um elemento após a metade anterior (posição 4) e o fim da array (posição 6) e o elemento na metade, que será comparado é 15 (posição 5).

array total

$x = 20$

$e = 4$ $m = 5$ $d = 6$

2	5	9	10	12	15	20
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]

Resultados Obtidos

- Quiz (Primeira Etapa)
 - Chatterbot explica a lógica da primeira etapa do algoritmo de busca binária.
 - Chatterbot pede ao usuário para inserir sua resposta, ou seja, completar o trecho de código-fonte incompleto.

O primeiro passo na lógica do programa é realizar o cálculo que determina qual é o número da posição no meio entre os dois extremos. Seguindo o exemplo fornecido, observe a imagem que explica a lógica por trás deste cálculo.

tamanho até d -> 6

tamanho entre os extremos = 2

posição da metade entre os extremos -> 1

tamanho até e -> 4

posição da metade entre os extremos -> 1

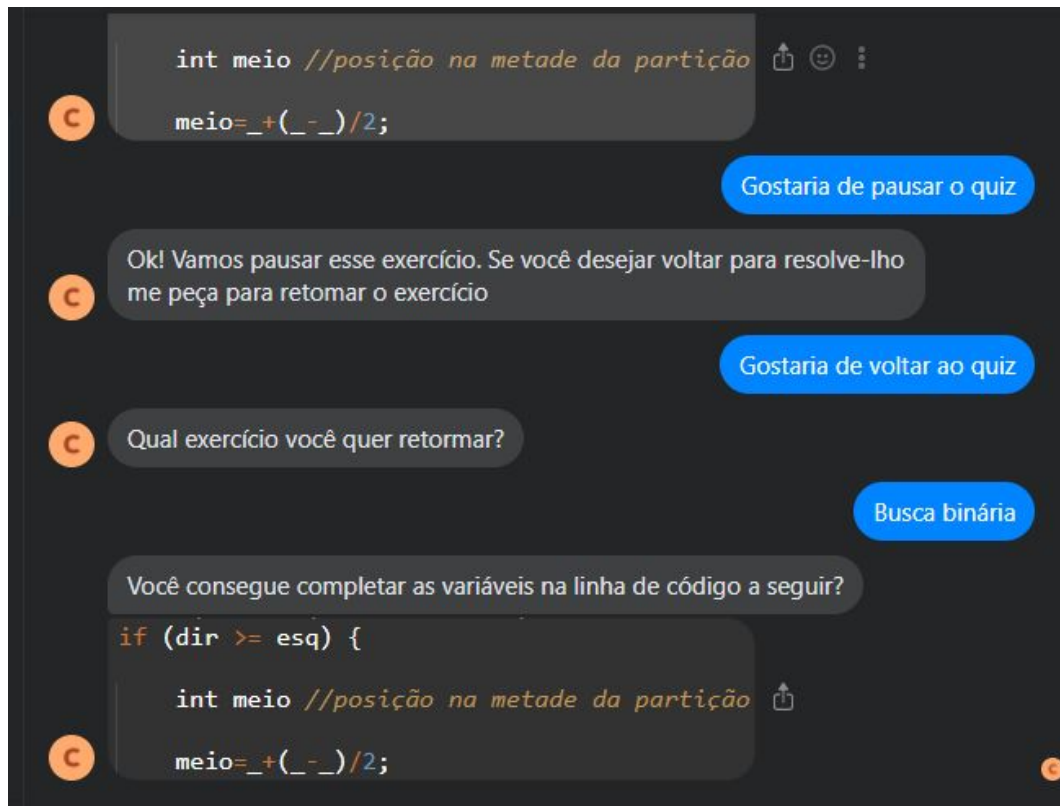
posição da metade na array total, m = 5

Você consegue completar as variáveis na linha de código a seguir?

```
if (dir >= esq) {  
    int meio //posição na metade da partição  
    meio = _ + (_ - _)/2;
```

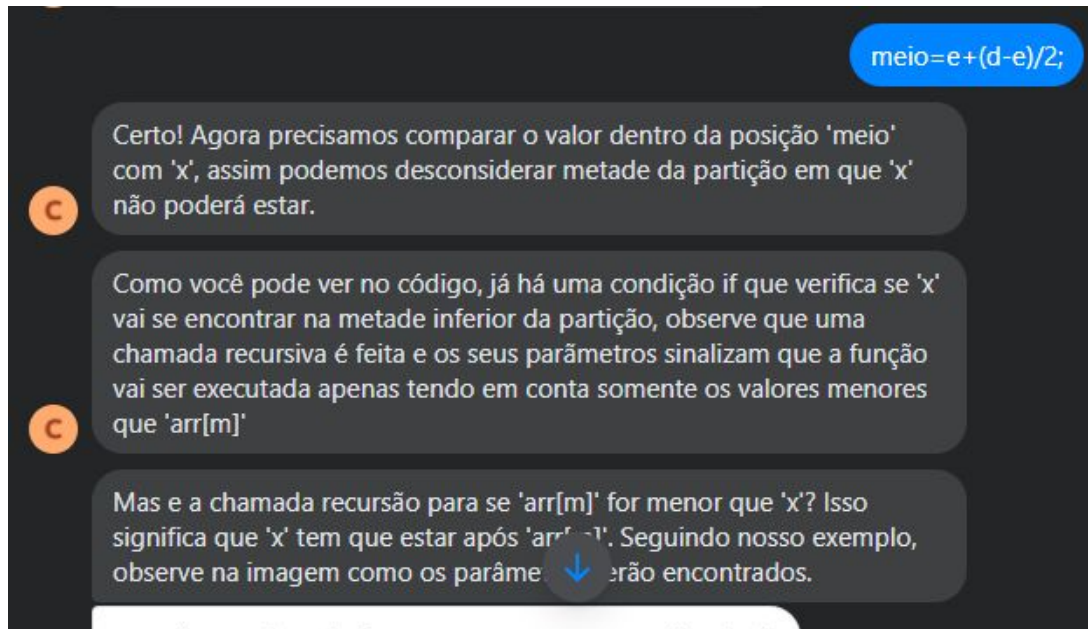
Resultados Obtidos

- Quiz (Pausa)
 - Usuário pode pausar e retornar ao quiz a qualquer momento.
 - Quando desejar retornar, o usuário deve indicar ao chatterbot qual quiz deseja retornar.
 - Chatterbot repete a última etapa do algoritmo não concluída.



Resultados Obtidos

- Quiz
 - Usuário acerta a resposta, ou seja, completa corretamente o trecho de código-fonte incompleto.
 - Chatterbot explica a lógica da próxima etapa do algoritmo de busca binária.



Resultados Obtidos

- Quiz (Segunda Etapa)
 - Chatterbot explica a lógica da segunda etapa do algoritmo de busca binária.
 - Chatterbot pede ao usuário para inserir sua resposta, ou seja, completar o trecho de código-fonte incompleto.

que arr[m]

Mas e a chamada recursão para se 'arr[m]' for menor que 'x'? Isso significa que 'x' tem que estar após 'arr[m]'. Seguindo nosso exemplo, observe na imagem como os parâmetros serão encontrados.

e = 4	m = 5	d = 6
12	15	20
arr[4]	arr[5]	arr[6]

arr[m] = 15

15 < 20 x = 20

e = m + 1

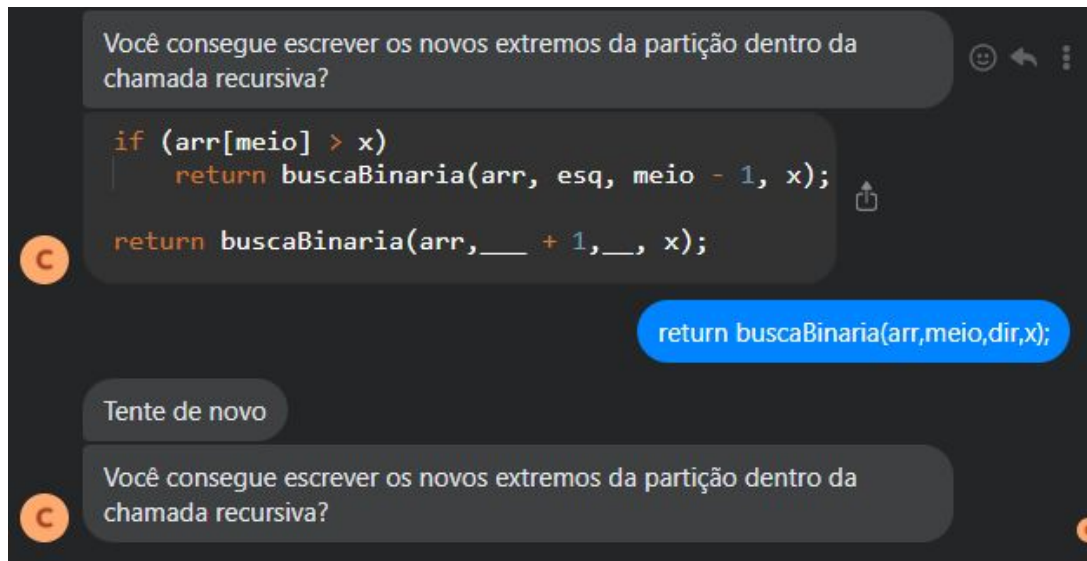
Você consegue escrever os novos extremos da partição dentro da chamada recursiva?

```
if (arr[meio] > x)
    return buscaBinaria(arr, esq, meio - 1, x);

return buscaBinaria(arr, __ + 1, __, x);
```

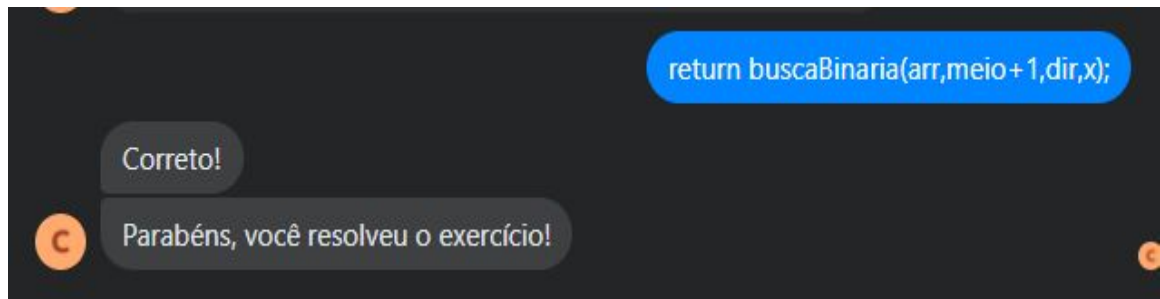
Resultados Obtidos

- Quiz
 - Usuário erra a resposta, ou seja, completa equivocadamente o trecho de código-fonte incompleto.
 - Chatterbot pede que o usuário tente novamente e repete a pergunta anterior.



Resultados Obtidos

- Quiz
 - Usuário acerta a resposta, ou seja, completa corretamente o último trecho de código-fonte incompleto.
 - Chatterbot parabeniza o usuários.



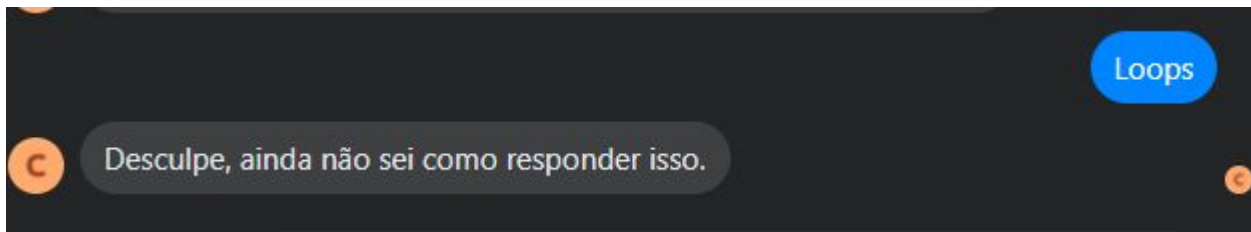
Resultados Obtidos

- Explicação e Quiz
 - Encerramento da conversa



Resultados Obtidos

- Explicação e Quiz
 - Se o assunto ainda não está no escopo do chatterbot ele apresenta uma mensagem de desculpas.
 - Exemplo: Laços (*Loops*)



Considerações Finais

- Neste trabalho foi desenvolvido o C-BOT: uma chatterbot para o ensino da programação.
- Ele é capaz de explicar conceitos de programação (Variáveis, Recursão e Árvores Binárias) e aplicar o quiz sobre algoritmos populares (Busca Binária, Busca em profundidade e Quicksort).
- Os códigos-fonte do C-BOT estão disponíveis no Github : [jessoares/Chatbot](#) ([github.com](#)).

Trabalhos Futuros

- Adaptação às particularidades de cada aluno
- Outras formas de “construir” códigos
- Explicações para dúvidas mais específicas
- Intenções casuais
- Integração com plataforma de uso constante
- Aplicação em um contexto educacional

Referências

- A. SACCARO, P. A. J. e. a. 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), Brasil, p. 1–3, 2019.
- N. ASHILL, S. E. e. The determinants of students' perceived learning outcomes and satisfaction in university online education: An update. Decision Sciences Journal of Innovative Education, USA, p. 3–5, 2016.
- BURNING GLASS, B. Beyond point and click the expanding demand for coding skills. Disponível em: <https://academy.oracle.com/pages/Beyond_Point_Click_final.pdf>. Acesso em: 2021-09-03.
- BENNEDSEN. Teaching and learning introductory programming: a model-based approach. Denmark: Editora desconhecida, 2008.
- T. DARADOUMIS J. M. M. PUIG, M. A. e. a. Analyzing students perceptions to improve the design of an automated assessment tool in online distributed programming. Spain: Computers and Education, 2016.
- JENKINS, T. On the Difficulty of Learning to Program Massachusetts: Computers and Education, 2002.
- HOBERT, S. 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 Germany, p.1–7, 2019.
- T. CROW, A. L.-R. e. B. W. Intelligent tutoring systems for programming education: a systematic review. Australasian Computing Education Conference Australia, p.1–3, 2018
- F. CLARIZIA F. COLACE, M. L. e. a. An education support system for student. In International Symposium on Cyberspace Safety and Security. Cyberspace Safety and Security. Italy, p.2–3, 2018.
- M. SU, C. W. e. H. W. A chatbot using lstm-based multi-layer embedding for elderly care. 2017 International Conference on Orange Technologies, Singapore, p.1–2, 2020.
- B. J. KUN, S. R. e. Quizbot: A dialogue-based adaptive learning system for factual knowledge. CHI Conference on Human Factors in Computing Systems. Brasil, p.2–3, 2019.
- L. MOUSSIADES, E. A. e. An Overview of Chatbot Technology. Artificial Intelligence Applications and Innovations. Greece, p.1–3, 2020.
- A. ADE-IBIJOLA, C. W. O. e. Python-Bot: A Chatbot for Teaching Python Programming. Engineering Letters., South Africa, p.1–6, 2021.
- S. H. M. DAUD, N. H. I. T. e. N. H. M. Z. e-Java Chatbot for Learning Programming Language: A Post-Pandemic Alternative Virtual Tutor. International Journal of Emerging Trends in Engineering Research, Malaysia, p.1–6, 2020.
- K. RAMESH S. RAVISHANKARAN, A. J. e. a. A Survey of Design Techniques for Conversational Agents. Communications in Computer and Information Science, India, p.1–2, 2017.
- A. PEARS S. SEIDMAN, L. M. e. a. A survey of literature on the teaching of introductory programming. ACM SIGCSE, USA, p.1–2, 2009.

Muito obrigado por sua atenção.