

# **Algoritmos e Estrutura de Dados II**

## **Jogo de 20 Perguntas**

**Joana D'arc P. Barbosa<sup>1</sup>, Antônio Frederico<sup>1</sup>, Túlio Faria<sup>1</sup>**  
**Departamento de Ciência da Computação – Instituto Federal de Ciência e Tecnologia**  
**do Sudeste de Minas Gerais - Campus Rio Pomba (IFSEMG)**  
**Av. Dr. José Sebastião da Paixão s/no – Bairro Lindo Vale – CEP 36180-000**  
**Rio Pomba – MG – Brasil**

1

### **1. Introdução**

trabalho foi desenvolvido um programa que se utiliza de uma árvore binária para representar uma base de conhecimento. No programa implementamos um jogo chamado "Jogo de 20 Perguntas". O programa interage com o usuário criando uma árvore de perguntas e respostas. No começo do jogo, o jogador inicia a rodada pensando em alguma coisa e em seguida o computador tenta adivinhar o que o usuário pensou.

O computador faz perguntas ao usuário guardando as informações que ele recebe, com estas perguntas ele dá um palpite para o usuário tentando adivinhar o que ele pensou, caso esteja certo o computador ganha o jogo. A cada rodada o computador utiliza as informações dos jogos anteriores e ganha mais conhecimento, para isso tivemos que guardar as informações em um arquivo para que elas não fossem perdidas. O usuário é responsável por informar se o computador deve ou não usar essas informações.

O programa foi desenvolvido na linguagem Python, por possuir estruturas de alto nível, como as listas ligadas que tem a mesma construção de uma árvore pelo uso de células, pois uma árvore binária também possui uma célula auxiliar que referencia outras duas células, chamadas sub-árvores esquerda e direita. Neste programa a sub-árvore esquerda representa "verdadeiro" e a sub-árvore direita representa "falso", passos que o computador acompanha após as respostas do usuário.

## 2. Implementação

Buscou-se, neste trabalho, implementar um programa chamado ,Jogo de 20 Perguntas. Inicialmente, foi criado uma classe chamada Arvore que tem como parâmetro os nós esquerda e direita da arvore, e também foi adicionado uma variável chamada conteúdo, que representa as informações contidas dentro dos nós.

```
class Arvore:
    def __init__(self, conteudo, esquerda=None, direita=None):
        self.conteudo = conteudo
        self.esquerda = esquerda
        self.direita = direita

    def __str__(self):
        return str(self.conteudo)

    def getEsquerda(self):
        return self.esquerda

    def getDireita(self):
        return self.direita
```

Figura 1.

O programa também contém uma função chamada verdadeiro, usada durante o funcionamento do jogo ,quando o usuário digita Sim para alguma pergunta ela devolve um valor verdadeiro.

```
def verdadeiro(decisao):
    from string import lower
    resposta = lower(raw_input(decisao))
    return (resposta[0] == 's' or resposta[0] == 'S')
```

Figura 2.

A função principal do programa, chamada main inicia interagindo com o usuário e fazendo a primeira pergunta, a partir da resposta do usuário o laço while caminha pela árvore de cima para baixo. A condição do laço é 'continuar', isso significa que o programa continuará rodando até que o usuário não esteja pensando em nada, ou seja, até que encontre o comando break.

---

```
def main():
    print "Bem-vindo ao jogo de 20 Perguntas!"
    print "Pense em alguma coisa e eu vou adivinhar"
    if verdadeiro("Devo me lembrar de nossos jogos anteriores?"):
        lerArquivo()

    cont = 0
    vitorias = 0
    continuar = True
    raiz = Arvore("pessoa")
    while continuar:
        cont = cont+1
        if not verdadeiro("Esta pensando em alguma coisa? "): break
        arvore = raiz
        while arvore.esquerda!= None:
            if verdadeiro("Q:" + arvore.conteudo + "? "):
                arvore = arvore.esquerda
            else:
                arvore = arvore.direita

        pessoa = arvore.conteudo
        if verdadeiro("Q:E um(a) " + pessoa + "? "):
            vitorias = vitorias +1
            imprimeArvore(arvore)
            print "Eu ganhei de voce amigo >_<!"
            if verdadeiro(" Deseja jogar de novo?"):
                continue
        print " Jogadas realizadas:" ,cont
        print " Eu ganhei",vitorias
```

**Figura 3.**

A cada jogada que o usuário faz, o computador guarda as informações obtidas nas jogadas anteriores e as adiciona à árvore.

```
#pegando novas informacoes do usuario
novo = raw_input("Eu perdi amigo(a) -.-. O que voce havia pensado? ")
informacao = raw_input("Faca uma pergunta que diferencia um(a) " + pessoa + " do que voce t
diferenca = "Qual e a resposta <s/n> para essa pergunta? "
#adicionando novas informacoes na arvore
arvore.conteudo = informacao
if verdadeiro(diferenca):
    arvore.esquerda = Arvore(pessoa)
    arvore.direita = Arvore(novo)
else:
    arvore.direita = Arvore(pessoa)
    arvore.esquerda = Arvore(novo)
imprimeArvore(arvore)

return 0

if __name__ == '__main__':
    main()
```

---

**Figura 4.**

### 3. Testes Executados

Neste capítulo, serão apresentados os testes executados durante o desenvolvimento do programa.

```
Bem-vindo ao jogo de 20 Perguntas!  
Pense em alguma coisa e eu vou adivinhar  
Devo me lembrar de nossos jogos anteriores?n  
Esta pensando em alguma coisa? s  
Q:E um(a) pessoa? n  
Eu perdi amigo(a) _-_-...O que voce havia pensado? gato  
Faca uma pergunta que diferencia um(a) pessoa do que voce havia pensado ?pode falar  
Qual e a resposta <s/n>para essa pergunta? s  
Esta pensando em alguma coisa? s  
Q:pode falar? n  
Q:E um(a) gato? s  
R: gato  
Eu ganhei de voce amigo >_<!  
Deseja jogar de novo?n  
Jogadas realizadas: 2  
Eu ganhei 1  
Devo lembrar-me desse jogo?n  
>>>
```

Figura 5.

```
Bem-vindo ao jogo de 20 Perguntas!  
Pense em alguma coisa e eu vou adivinhar  
Devo me lembrar de nossos jogos anteriores?n  
Esta pensando em alguma coisa? s  
Q:E um(a) pessoa? n  
Eu perdi amigo(a) _-_-...O que voce havia pensado? gato  
Faca uma pergunta que diferencia um(a) pessoa do que voce havia pensado ?pode falar  
Qual e a resposta <s/n>para essa pergunta? s  
Esta pensando em alguma coisa? s  
Q:pode falar? n  
Q:E um(a) gato? n  
Eu perdi amigo(a) _-_-...O que voce havia pensado? cao  
Faca uma pergunta que diferencia um(a) gato do que voce havia pensado ?pode miar  
Qual e a resposta <s/n>para essa pergunta? s  
Esta pensando em alguma coisa? s  
Q:pode falar? n  
Q:pode miar? n  
Q:E um(a) cao? n  
Eu perdi amigo(a) _-_-...O que voce havia pensado? computador  
Faca uma pergunta que diferencia um(a) cao do que voce havia pensado ?pode latir  
Qual e a resposta <s/n>para essa pergunta? s  
Esta pensando em alguma coisa? s  
Q:pode falar? s  
Q:E um(a) pessoa? s  
R: pessoa  
Eu ganhei de voce amigo >_<!  
Deseja jogar de novo?n  
Jogadas realizadas: 4  
Eu ganhei 1  
Devo lembrar-me desse jogo?n
```

Figura 6.

```

Bem-vindo ao jogo de 20 Perguntas!
Pense em alguma coisa e eu vou adivinhar
Devo me lembrar de nossos jogos anteriores?s
Qual o nome do arquivo?Questao1.txt
Lendo do arquivo...
Q: É um animal?

R: gato

Q: Ele vai em seus pés?

R: sapato

R: computador
Esta pensando em alguma coisa? s
Q:É um(a) pessoa? n
Eu perdi amigo(a) -_-...O que voce havia pensado? animal
Faca uma pergunta que diferencia um(a) pessoa do que voce havia pensado ?pode falar
Qual e a resposta <s/n>para essa pergunta? s
Esta pensando em alguma coisa? s
Q:pode falar? n
Q:É um(a) animal? n
Eu perdi amigo(a) -_-...O que voce havia pensado? sapato
Faca uma pergunta que diferencia um(a) animal do que voce havia pensado ?e irracional
Qual e a resposta <s/n>para essa pergunta? s
Esta pensando em alguma coisa? s
Q:pode falar? n
Q:e irracional? s
Q:É um(a) animal? s
R: animal

```

**Figura 7.**

```

Bem-vindo ao jogo de 20 Perguntas!
Pense em alguma coisa e eu vou adivinhar
Devo me lembrar de nossos jogos anteriores?n
Esta pensando em alguma coisa? s
Q:É um(a) pessoa? n
Eu perdi amigo(a) -_-...O que voce havia pensado? sapato
Faca uma pergunta que diferencia um(a) pessoa do que voce havia pensado ?pode falar
Qual e a resposta <s/n>para essa pergunta? s
Esta pensando em alguma coisa? s
Q:pode falar? n
Q:É um(a) sapato? s
R: sapato
Eu ganhei de voce amigo >_<!
Deseja jogar de novo?n
Jogadas realizadas: 2
Eu ganhei 1
Devo lembrar-me desse jogo?s
Qual o nome do arquivo?Questa2.txt

```

**Figura 8.**

## 4. Conclusão

Este trabalho teve como objetivo a implementação do Jogo de 20 Perguntas na linguagem Python.

O algoritmo do jogo é constituído por uma estrutura de árvore binária e nó (sim e não). Uma função para verificar se a resposta do usuário é um nó sim ou um nó não, este decide para qual direção o algoritmo deverá seguir. Outras funções utilizadas foram para gerar os logs de saída e a entrada de dados externos.

Durante o desenvolvimento do trabalho, as maiores dificuldades enfrentadas pelo grupo ao implementar o algoritmo, foram manipular arquivos de entrada/saída e armazenar externamente o estado da árvore binária, uma vez que exigia percorrer a árvore em ordem correta para obtenção das perguntas e respostas em sua respectiva sequência.

Concluimos que o desenvolvimento deste trabalho foi de grande importância para nossa compreensão e aprofundamento do tema abordado, pois visto que nos permitiu desenvolver um jogo interativo utilizando-se dos conhecimentos adquiridos nas aulas.

## **Referências**

- [1] DOWNEY,A., ELKNER,J., MEYERS,C. *Como pensar como um Cientista da Computação usando Python*.janeiro 2008.
- [2] BORGES,L.E. *Python para Desenvolvedores*. 2 edição.