

NOME DA DISCIPLINA	CARGA HORÁRIA	MODALIDADE	PROFESSOR-TUTOR
Raciocínio Computacional	75h	EaD	Galbas Milléo Filho

Algoritmo em pseudocódigo do jogo ZumbieDice

```
função pegarDadosVerde()
    retorne ("C", "P", "C", "T", "P", "C")
função pegarDadosAmarelo():
    retorne ("T", "P", "C", "T", "P", "C")
função pegarDadosVermelhos()
    retorne ("T", "P", "T", "C", "P", "T")

função initDadosCopo(copo)
    para i ← 1 até 6 faça
        copo.inserir(pegarDadosVerde())
    # colocar dados amarelos no copo
    para i ← 1 até 4 faça
        copo.inserir(pegarDadosAmarelo())
    # colocar dados vermelhos no copo
    para i ← 1 até 3 faça
        copo.inserir(pegarDadosVermelhos())
    retorne copo

função pegarDadosCopo(copo)
    # Quantidade de dados no copo
    se tamanho(copo) <> 0
        numDados ← (tamanho(copo) - 1)
        index ← random(0, numDados)
        dado ← copo[index]
        delete (copo[index])
        retorne dado, copo
    senão
        escreva("Copo Vazio!!!")
        retorne -1, copo

função lancarDado(dado)
    faceDado ← random(0, 5)
    se dado[faceDado] = "C"
        escreva("Cerebro!!!")
        retorne 'C'
    senão se dado[faceDado] = "T"
        escreva("Tiro!!!")
        retorne 'T'
    senão
        imprima("Passos!!!")
        retorne "P"
```

```

função mostrarDadosCopo(copo)
  listDado ← []
  para i ← 1 até tamanho(copo) faça
    se copo[i] = ("C", "P", "C", "T", "P", "C")
      listDado.inserir("verde")
    senão se copo[i] = ("T", "P", "C", "T", "P", "C")
      listDado.inserir("amarelo")
    senão
      listDado.inserir("vermelho")
  escreva(listDado)

```

```

função mostrarDado(dado)
  se dado = ("C", "P", "C", "T", "P", "C")
    escreva("verde")
  senão se dado = ("T", "P", "C", "T", "P", "C"):
    escreva("amarelo")
  senão
    escreva("vermelho")

```

```

funcao verificarScore(primeiro, segundo, terceiro)
  tiro ← 0
  cerebro ← 0
  passos ← 0
  se primeiro = "C"
    cerebro += 1
  senão se primeiro = "T"
    tiro += 1
  senão
    passos += 1
  se segundo = "C"
    cerebro += 1
  senão se segundo == "T"
    tiro += 1
  senão
    passos += 1
  se terceiro = "C"
    cerebro += 1
  senão se terceiro == "T"
    tiro += 1
  senão
    passos += 1

  retorne cerebro, tiro, passos

```

```

copo ← []
copo ← initDadosCopo(copo)
listPlayers ← []

```

```

leia(numPlayers)
se numPlayers < 2
  escreva("Sao necessario no minimo dois jogadores!!!\n")
senão:
  para i ← 1 até numPlayers faça
    leia(nome)
    cerebro ← 0
    tiro ← 0
    player ← [ind, nome, cerebro, tiro]
    listPlayers.inserir(player)

```

play ← True

enquanto (play)

para i←1 até listPlayers faça

cod ← player[0]
nome ← player[1]

escreva("##### Player " + nome + "#####")
mostrarDadosCopo(copo)
turno ← Verdadeiro
blockDado1 ← Verdadeiro
blockDado2 ← Verdadeiro
blockDado3 ← Verdadeiro

primeiroDado ← -1
segundoDado ← -1
terceiroDado ← -1

enquanto (turno)

escreva("Continuar ou sair do jogo:: y/yes - continuar ou n/no sair do jogo: ")
leia(playGame)
se playGame = "y" ou playGame = "yes"
continue

senão

turn ← Falso
play ← Falso
sair

escreva("Drawn Dice:")

se blockDado1

primeiroDado, copo ← pegarDadosCopo(copo)
mostrarDado(primeiroDado)

se blockDado2

segundoDado, copo ← pegarDadosCopo(copo)
mostrarDado(segundoDado)

se blockDado3

terceiroDado, copo ← pegarDadosCopo(copo)
mostrarDado(terceiroDado)

escreva("Mostrar Dados Copo:")
mostrarDadosCopo(copo)

one ← " "
two ← " "
three ← " "

se primeiroDado <> -1

one ← lancarDado(primeiroDado)

se segundoDado <> -1

two ← lancarDado(segundoDado)

se terceiroDado <> -1

three ← lancarDado(terceiroDado)

Para os dados com face "passos"

blockDado1 ← Verdadeiro
blockDado2 ← Verdadeiro
blockDado3 ← Verdadeiro

cerebro, tiro, passos ← verificarScore(one, two, three)

```

# Verificar se a vítima escapou
se passos > 0
    se one = "P"
        blockDado1 ← Falso
    se two = "P"
        blockDado2 ← Falso
    se three = "P"
        blockDado3 ← Falso

listPlayers[cod][2] ← player[2] + cerebro
listPlayers[cod][3] ← player[3] + tiro

escreva("Player: " + listPlayers[cod][1])
escreva("Cerebro: " + str(listPlayers[cod][2]))
escreva("Tiro: " + str(listPlayers[cod][3]))

escreva(str(listPlayers[player[0]][3]))

se listPlayers[player[0]][3] > 2
    escreva("Voce morreu!!\n")
    listPlayers[player[0]][2] ← 0
    listPlayers[player[0]][3] ← 0
    copoReset ← []
    copo ← initDadosCopo(copoReset)
    mostrarDadosCopo(copo)

    turno ← Falso

se listPlayers[player[0]][2] > 12
    escreva("Parabens, você venceu!!!")
    play ← Falso
    turno ← Falso

se turno
    escreva("Você deseja continuar? (y - yes - n - not)")
    leia (continueTurno)

    if continueTurno = "y" ou continueTurno = "yes":
        continue
    senão
        listPlayers[player[0]][3] ← 0
        copoReset ← []
        copo ← initDadosCopo(copoReset)
        mostrarDadosCopo(copo)
        turno ← False

```