



G2 – LÓGICA DE PROGRAMAÇÃO

NOME: FERNANDA AYRES RUARO
DISCIPLINA: LÓGICA DE PROGRAMAÇÃO
DATA: 10/04/2021

- **Passo 1 (Identificar o problema):** Fazer um jogo de pedra, papel e tesoura.
- **Passo 2 (Escrever a historinha do problema):**
 Cliente pediu um jogo simples para entreter seus filhos na pandemia. Conversamos bastante sobre o que seus filhos gostam e chegamos à conclusão de um jogo de pedra, papel e tesoura. Como o cliente tem 2 filhos pediu que o jogo tenha opção para dois jogares, assim seus filhos podem brincar juntos e em cada partida um pode ser vitorioso. Quer que o jogo informe a cada rodada quem ganhou e que no final possa informar quem é o vencedor de cada partida. Para que eles possam jogar novamente a cada final de partida terá um pergunta "querem jogar novamente?", assim eles podem escolher quantas vezes querem jogar, e para ter um pouco mais de interatividade com o jogo a cada partida eles podem decidir quantas rodadas terá cada partida.
Regras do jogo:
 - tesoura vence papel / tesoura perde para pedra / tesoura empata com tesoura
 - papel vence pedra / papel perde para tesoura / papel empata com papel
 - pedra perde para papel / pedra vence tesoura / pedra empata com pedra.
- **Passo 3 (Identificar as variáveis):**
 - qntdRodadas, aux: inteiro
 - nome1, nome2: caractere
 - jogador1, jogador2: inteiro // controla a jogada de cada jogador em cada rodada
 - vitoriaJog1, vitoriaJog2: inteiro //controla as vitórias de cada jogador em cada rodada
- **Passo 4 (Desenhar um Fluxograma do algoritmo):**
 Utilizei programa de fluxograma (draw.io) → anexado no final deste pdf.
- **Passo 6 (Escrever o programa em Portugol):** Utilizei o Portugol Studio
 - anexeí fotos do código no Portugol Studio no final deste pdf
 - anexeí o código a seguir também:

programa {

inclua biblioteca Util --> u

funcao inicio() {
 inteiro qntdRodadas, aux
 inteiro jogador1, jogador2, vitoriaJog1 = 0, vitoriaJog2 = 0
 cadeia nome1, nome2

 //laço de repetição faça para pelo menos uma vez rodar o jogo, e no final o jogador decide se quer jogar novamente

faça {
 escreva("Informe nome do jogador 1: ")
 leia(nome1)
 escreva("Informe o nome do jogador 2: ")
 leia(nome2)
 limpa()

 //o jogador informa quantas rodadas vai querer jogar em cada partida
 escreva("Informe a quantidade de rodadas: ")

```
leia(qntdRodadas)
limpa()
```

```
//laço para repetir até a quantidade de rodadas informadas pelos jogadores
para(inteiro i=0; i<qntdRodadas; i++){
```

```
    //bloco que válida e armazena a jogado do jogador 1 a cada partida
    faca{
        escreva("RODADA ", i+1, "\n\n")
        escreva("Vez do jogador 1:")
        escreva("\n1- Pedra \n2- Papel \n3- Tesoura \nEscolha sua jogada: ")
        leia(jogador1)
        se(jogador1 < 1 ou jogador1 > 3){
            escreva("\n\nValor informado inválido!")
        }
        u.aguarde(1000)
        limpa()
    } enquanto (jogador1 < 1 ou jogador1 > 3)
```

```
    //bloco que válida e armazena a jogado do jogador 2 a cada partida:
    faca{
        u.aguarde(1000)
        escreva("RODADA ", i+1, "\n\n")
        escreva("Vez do jogador 2:")
        escreva("\n1- Pedra \n2- Papel \n3- Tesoura \nEscolha sua jogada: ")
        leia(jogador2)
        se(jogador2 < 1 ou jogador2 > 3){
            escreva("\n\nValor informado inválido!")
        }
        u.aguarde(1000)
        limpa()
    } enquanto (jogador2 < 1 ou jogador2 > 3)
```

```
    //desvios condicionais com as regras do jogo para comparar a cada
rodada quem é o vencedor e acrescenta um ponto para o jogador vitorioso da rodada
```

```
    se(jogador1 == jogador2 ) {
        escreva("EMPATE")
    }
    senao se(jogador1 == 1 e jogador2 == 2) {
        escreva("VITÓRIA DO JOGADOR 2")
        vitoriaJog2++
    }
    senao se(jogador1 == 1 e jogador2 == 3) {
        escreva("VITÓRIA DO JOGADOR 1")
        vitoriaJog1++
    }
    senao se(jogador1 == 2 e jogador2 == 1) {
        escreva("VITÓRIA DO JOGADOR 1")
        vitoriaJog1++
    }
    senao se(jogador1 == 2 e jogador2 == 3) {
        escreva("VITÓRIA DO JOGADOR 2")
    }
```

```

        vitoriaJog2++
    }
    senao se(jogador1 == 3 e jogador2 == 1) {
        escreva("VITÓRIA DO JOGADOR 2")
        vitoriaJog2++
    }
    senao se(jogador1 == 3 e jogador2 == 2) {
        escreva("VITÓRIA DO JOGADOR 1")
        vitoriaJog1++
    }
    u.aguarde(1000)
    limpa()
}

//desvios condicionais para verificar quem ganhou o jogo
se(vitoriaJog1 == vitoriaJog2){
    escreva("EMPATE!\n")
}
senao se(vitoriaJog1 < vitoriaJog2){
    escreva("PARABÉNSS ",nome2 , "!! VITÓRIA DO JOGADOR 2!\n")
}
senao{
    escreva("PARABÉNSS ",nome1 , "!! VITÓRIA DO JOGADOR 1!\n")
}

```

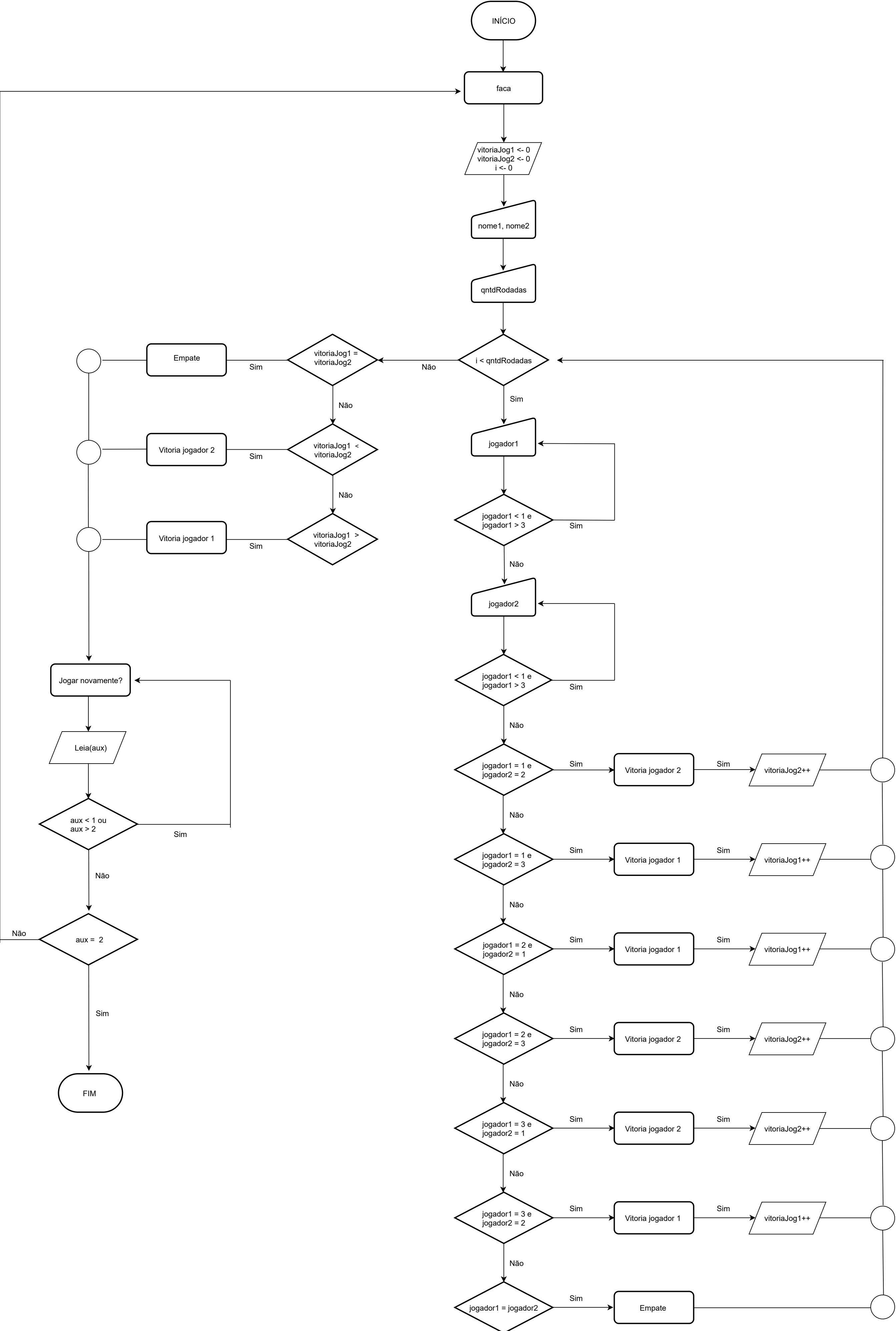
//bloco de condição que controla se o jogador quer jogar novamente e válida a resposta

```

faca{
    escreva("\nGostaria de jogar novamente? \n1- SIM \n2- NÃO\n")
    leia(aux)
    limpa()
    se(aux < 1 ou aux > 2){
        //caso jogador informe um valor sem ser 1 e 2 aparece esta mensagem
        escreva("Valor informado inválido!")
    }
} enquanto (aux < 1 ou aux > 2)
//quando jogador informar um valor válido, sai do loop

} enquanto(aux == 1)
//se o jogador responder 1 (SIM) a condição será verdadeira então irá
repetir o jogo
//se o jogador responder 2 (NÃO) a condição será falsa e sairá do loop de
repetição e não irá repetir o jogo
}
}

```



```
1 - programa {
2
3     inclui biblioteca Util --> u
4
5 - funcao inicio() {
6     inteiro qntdRodadas, aux
7     inteiro jogador1, jogador2, vitoriaJog1 = 0, vitoriaJog2 = 0
8     cadeia nome1, nome2
9
10    //laço de repetição faca para pelo menos uma vez rodar o jogo, e no final o jogador decide se quer jogar novamente
11 -   faca {
12       escreva("Informe nome do jogador 1: ")
13       leia(nome1)
14       escreva("Informe o nome do jogador 2: ")
15       leia(nome2)
16       limpa()
17
18       //o jogador informa quantas rodadas vai querer jogar em cada partida
19       escreva("Informe a quantidade de rodadas: ")
20       leia(qntdRodadas)
21       limpa()
22
23       //laço para repetir até a quantidade de rodadas informadas pelos jogadores
24 -   }
25 }
```



Portugol Webstudio



portugol-webstudio.cubos.io/ide



Portugol Webstudio



Jogo.por



22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

```
//laço para repetir até a quantidade de rodadas informadas pelos jogadores
para(inteiro i=0; i<qntdRodadas; i++){

    faca{
        //bloco que válida e armazena a jogado do jogador 1 a cada partida:
        escreva("RODADA ", i+1, "\n\n")
        escreva("Vez do jogador 1:")
        escreva("\n1- Pedra \n2- Papel \n3- Tesoura \nEscolha sua jogada: ")
        leia(jogador1)
        se(jogador1 < 1 ou jogador1 > 3){
            escreva("\n\nValor informado inválido!")
        }
        u.aguarde(1000)
        limpa()
    } enquanto (jogador1 < 1 ou jogador1 > 3)

    //bloco que válida e armazena a jogado do jogador 2 a cada partida:
    faca{
        u.aguarde(1000)
        escreva("RODADA ", i+1, "\n\n")
        escreva("Vez do jogador 2:")
        escreva("\n1- Pedra \n2- Papel \n3- Tesoura \nEscolha sua jogada: ")
        leia(jogador2)
```



```
45     leia(jogador2)
46     se(jogador2 < 1 ou jogador2 > 3){
47         escreva("\n\nValor informado inválido!")
48     }
49     u.aguarde(1000)
50     limpa()
51 } enquanto (jogador2 < 1 ou jogador2 > 3)
52
53 //desvios condicionais com as regras do jogo para comparar a cada rodada quem é o vencedor
54 //e acrescenta um ponto para o jogador vitorioso da rodada
55 se(jogador1 == jogador2 ) {
56     escreva("EMPATE")
57 }
58 senao se(jogador1 == 1 e jogador2 == 2) {
59     escreva("VITÓRIA DO JOGADOR 2")
60     vitoriaJog2++
61 }
62 senao se(jogador1 == 1 e jogador2 == 3) {
63     escreva("VITÓRIA DO JOGADOR 1")
64     vitoriaJog1++
65 }
66 senao se(jogador1 == 2 e jogador2 == 1) {
67     escreva("VITÓRIA DO JOGADOR 1")
68     vitoriaJog1++
```




66

67

68



69

70

71

72



73

74

75

76



77

78

79



80

81



82

83

84

85

86

87

88

89

```
senao se(jogador1 == 2 e jogador2 == 1) {  
    escreva("VITÓRIA DO JOGADOR 1")  
    vitoriaJog1++  
}  
senao se(jogador1 == 2 e jogador2 == 3) {  
    escreva("VITÓRIA DO JOGADOR 2")  
    vitoriaJog2++  
}  
senao se(jogador1 == 3 e jogador2 == 1) {  
    escreva("VITÓRIA DO JOGADOR 2")  
    vitoriaJog2++  
}  
senao se(jogador1 == 3 e jogador2 == 2) {  
    escreva("VITÓRIA DO JOGADOR 1")  
    vitoriaJog1++  
}  
u.aguarde(1000)  
limpa()  
}  
  
//desvios condicionais para verificar quem ganhou o jogo  
se(vitoriaJog1 == vitoriaJog2){  
    escreva("EMPATE!\n")  
}
```

```
90 -      senao se(vitoriaJog1 < vitoriaJog2){
91         escreva("PARABÊNSS ",nome2 , "!! VITÓRIA DO JOGADOR 2!\n")
92     }
93 -      senao{
94         escreva("PARABÊNSS ",nome1 , "!! VITÓRIA DO JOGADOR 1!\n")
95     }
96
97     //bloco de condição que controla se o jogador quer jogar novamente e válida a resposta
98     faca{
99         escreva("\nGostaria de jogar novamente? \n1- SIM \n2- NÃO\n")
100         leia(aux)
101         limpa()
102 -      se(aux < 1 ou aux > 2){
103         //caso jogador informe um valor sem ser 1 e 2 aparece esta mensagem
104         escreva("Valor informado inválido!")
105     }
106     } enquanto (aux < 1 ou aux > 2)
107     //quando jogador informar um valor válido, sai do loop
108
109     } enquanto(aux == 1)
110     //se o jogador responder 1 (SIM) a condição será verdadeira então irá repetir o jogo
111     //se o jogador responder 2 (NÃO) a condição será falsa e sairá do loop de repetição e não irá repetir o jogo
112 }
113 }
```