

## FUNDAMENTOS EM LÓGICA DE PROGRAMAÇÃO

**Atividade:** Começando a pensar logicamente e a desenvolver softwares utilizando o Portugol Studio

**Tema:** Estruturas de repetição – *FAÇA ENQUANTO* e *ENQUANTO*

### INDICADORES ASSOCIADOS

- 2 - Analisa e avalia o funcionamento de computadores e periféricos em ambientes computacionais.
- 3 - Codifica programas computacionais utilizando lógica de programação e respeitando boas práticas de programação.
- 5 - Desenvolver capacidades linguísticas de modo a saber usar adequadamente a linguagem oral e escrita em diferentes situações e contextos.
- 8 - Utilizar estruturas de dados definindo-as e aplicando-as adequadamente nos programas.

### A TEORIA

- Debatermos e conhecermos o funcionamento da estrutura de repetição *PARA FAÇA*, certo?
- Agora, vamos avançar ainda mais em nossa aquisição de novos conhecimentos e entraremos em contato com duas novas estruturas: *FAÇA ENQUANTO* e *ENQUANTO*;
- Imaginemos o seguinte: estou com sede, portanto, encherei um copo com água e vou beber;
- Nessa situação hipotética, há duas possibilidades, duas ponderações, duas validações a serem feitas:
  - Antes de beber a água, o copo está cheio?
  - Depois que bebi a água, ainda tenho sede?
- Repare que há dois momentos em que faço essas verificações: uma antes e outra depois;
- É isso que diferencia o funcionamento do *FAÇA ENQUANTO* do modo como age o *ENQUANTO*;

- Apesar de fazerem a mesma coisa, que é repetir uma ação inúmeras vezes, os momentos em que os dados são validados são diferentes;
- Mas, como assim?
- A estrutura *FAÇA ENQUANTO* é normalmente usada:
  - Em casos onde não se sabe a quantidade de vezes que deve ser repetido;
  - Em que conteúdo do bloco interno deve ser executado PELO MENOS UMA VEZ.
- A estrutura *ENQUANTO* é normalmente usada:
  - Em casos onde não se sabe a quantidade de vezes que deve ser repetido;
  - PODE NÃO SER NECESSÁRIO EXECUTAR o conteúdo do bloco interno.
- Pense na seguinte situação:
- Um livro tem ao menos uma página;
- Eu nunca comprei um livro.
- Vamos tentar elucidar essa questão melhor?
- Quantas páginas tem o livro "Harry Potter e a Pedra Filosofal"?
  - Vou folhear o livro até eu achar a última página;
  - Todo o livro tem NO MÍNIMO uma página;
  - No caso desse livro, são 264 páginas;
  - Para esse caso, eu utilizaria o *FAÇA ENQUANTO*.
- Quantos livros eu tenho em minha estante?
  - Vou olhar em minha estante para contar quantos livros tenho;
  - Vou verificar *ENQUANTO* eu achar livros na estante;
  - Em nosso caso, eu não tenho livros;
  - Para esse caso, o prudente seria se utilizar do *ENQUANTO*;
  - Eu posso ou não livros em minha estante.
- Posso inverter? Usar só *ENQUANTO*? Optar somente pelo *FAÇA ENQUANTO*?
- Pode sim, apenas nos preocupamos em trazer a solução mais simples e ideal para você;
- Use aquilo que se sentir mais confortável para chegar no resultado esperado;

ENTENDA QUE A CAMINHADA A SER TRILHADA É SUA E QUE SÓ VOCÊ SABE  
AQUILO QUE MELHOR CONSEGUIU ENTENDER E SE ADAPTAR;

- Bora entender como funcionam os dois casos?

### Usando o *FAÇA ENQUANTO*

- Inicialmente, vamos entender como funciona o desenvolvimento de uma solução simples com o *FAÇA ENQUANTO*;
- Preparamos um novo programa no Portugol Studio, criando duas variáveis: nroContador do tipo inteiro, logRepete do tipo lógico;
  - Além disso, iniciaremos nroContador com o valor 0:

Figura 1 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
programa
{
    funcao inicio()
    {
        inteiro nroContador
        logico logRepete

        nroContador = 0
    }
}
```

Fonte: Autores, 2023.

- Iniciado nosso código, agora passaremos a fazer a estruturação do *FAÇA ENQUANTO*:

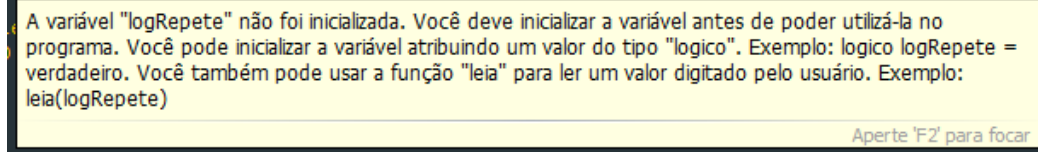
Figura 2 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
faca {
} enquanto(logRepete == verdadeiro)
```

Fonte: Autores, 2023.

- Repare duas coisas:
  - Conforme informado em nossa regra, sempre que logRepete for verdadeiro, o *FAÇA ENQUANTO* será repetido
  - A variável logRepete está sublinhada em vermelho;
    - Isso significa que há um erro abaixo da variável logRepete;
    - O erro que você repara está aqui embaixo:

Figura 3 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*



Fonte: Autores, 2023.

- Mas, por qual motivo isso acontece?
- Simples a variável não foi iniciada:
  - Ou seja, logRepete não possui valor algum;
  - Por enquanto, não faça nada para corrigir;
  - Porém, saiba que não inicializar variáveis não é parte de uma boa prática de programação;
  - Ou seja, sempre tentamos definir um valor base (dar um valor básico “sem importância” como zero, texto em branco, falso);
- Você pode alterar a última linha do seguinte trecho do início do código-fonte:

Figura 4 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
inteiro nroContador
logico logRepete

nroContador = 0
logRepete = falso
```

Fonte: Autores, 2023.

- Beleza, problema resolvido!
- Agora vamos criar uma regrinha para ser seguida pelo código-fonte;
- Então o que faremos?
- A nossa regra fará o seguinte:
  - Toda a vez que for executado o *FAÇA ENQUANTO*, será somado 1 no nroContador;
  - Em seguida, será perguntado se o usuário deseja prosseguir ou não:
    - Aqui é importante apontar que só será aceito os termos: verdadeiro ou falso;
    - Qualquer valor diferente disso, o programa deixará de funcionar
    - Mais a frente, você terá oportunidades para tentar corrigir esse problema.
  - Por fim, a resposta informada pelo usuário na pergunta, será armazenado na variável logRepete.

- Perceba que útil, sempre que logRepete for verdadeiro o programa seguirá no *FAÇA ENQUANTO*, correto?
- Nada mudou de antes, exceto o fato de que agora quem informa se quer ou não prosseguir é o próprio usuário:
  - O logRepete recebe o valor informado pelo usuário e só depois valida se prossegue ou não no *FAÇA ENQUANTO*.

Figura 5 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
faca {
    nroContador += 1

    escreva("Deseja continuar? Digite verdadeiro para SIM E falso para NÃO\n")
    leia(logRepete)
} enquanto(logRepete == verdadeiro)
```

Fonte: Autores, 2023.

- Pois bem, nosso código-fonte completo é finalizado com a apresentação do valor final do nroContador;

Figura 6 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
escreva("\nLembre-se que o contador se iniciou com a quantidade ZERO")
escreva("\nO contador executou " + nroContador + " o FAÇA ENQUANTO")
```

Fonte: Autores, 2023.

- Como sempre, salve, verifique se tudo está correto, e execute o código-fonte;
- A versão final está aqui embaixo:

Figura 7 - Desenvolvimento do código-fonte do *FAÇA ENQUANTO*

```
programa
{
    funcao inicio()
    {
        inteiro nroContador
        logico logRepete

        nroContador = 0
        //logRepete = falso

        faca {
            nroContador += 1

            escreva("Deseja continuar? Digite verdadeiro para SIM E falso para NÃO\n")
            leia(logRepete)
        } enquanto(logRepete == verdadeiro)

        escreva("\nLembre-se que o contador se iniciou com a quantidade ZERO")
        escreva("\nO contador executou " + nroContador + " o FAÇA ENQUANTO")
    }
}
```

Fonte: Autores, 2023.

- Resumindo toda essa brincadeira, quando usamos o *FAÇA ENQUANTO*, não queremos validar nada inicialmente;
  - Nossa intenção é que o programa rode SEMPRE pelo menos uma vez.

### Usando o *ENQUANTO*

- Agora, vamos ver o funcionamento do *ENQUANTO*?
- Para isso, novamente utilizaremos um exemplo, como temos feito desde nosso segundo documento;
- Vamos começar fazendo a cópia desse pequeno trecho:

Figura 8 - Desenvolvimento do código-fonte do ENQUANTO

```
programa
{
    funcao inicio()
    {
        inteiro nroContador
        logico logRepete

        nroContador = 0
```

Fonte: Autores, 2023.

- Reparou algo diferente?
- Perceba que não inicializamos a variável logRepete;
- Mas por quê?
- Faremos um procedimento um pouquinho diferente com ele, adicionando o trecho abaixo logo depois daquilo que já digitamos:

Figura 9 - Desenvolvimento do código-fonte do ENQUANTO

```
escreva("Deseja iniciar? Digite verdadeiro para SIM E falso para NÃO\n")
leia(logRepete)
```

Fonte: Autores, 2023.

- Repare que não inicializamos a variável em momento algum, porém, a primeira ação é solicitar o valor de logRepete ao usuário;
- Esse procedimento é bastante comum;
- Porém, vamos seguir com as boas práticas, por isso, deixe o seu código-fonte da seguinte forma:

Figura 10 - Desenvolvimento do código-fonte do ENQUANTO

```
inteiro nroContador
logico logRepete

nroContador = 0
logRepete = falso

escreva("Deseja iniciar? Digite verdadeiro para SIM E falso para NÃO\n")
leia(logRepete)
```

Fonte: Autores, 2023.

- Agora, vamos prosseguir;
- Assim como no exemplo que fizemos para o *FAÇA ENQUANTO*, utilizaremos a mesma validação:
  - Ou seja, sempre que o logRepete for informado como verdadeiro, o *ENQUANTO* será executado novamente

Figura 11 - Desenvolvimento do código-fonte do ENQUANTO

```
enquanto (logRepete == verdadeiro) {  
    }  
}
```

Fonte: Autores, 2023.

- Contudo, temos uma diferença: por qual motivo o *ENQUANTO* a gente faz a validação no começo?
- Essa é exatamente a diferença entre o *FAÇA ENQUANTO* e o *ENQUANTO*:
  - *FAÇA ENQUANTO* – sempre executa PELO MENOS UMA VEZ:
    - Ou seja, faz a sua validação APENAS após executar uma vez;
  - *ENQUANTO* – faz sua validação no início:
    - Depende do resultado de uma validação;
    - ELE PODE NEM SER EXECUTADO;
- Pois bem, agora que já sabemos as diferenças entre o *FAÇA ENQUANTO* e o *ENQUANTO*, vamos prosseguir;
- Complemente o seu código-fonte com o seguinte trecho:

Figura 12 - Desenvolvimento do código-fonte do ENQUANTO

```
enquanto (logRepete == verdadeiro) {  
  
    nroContador += 1  
  
    escreva("Deseja continuar? Digite verdadeiro para SIM E falso para NÃO\n")  
    leia(logRepete)  
  
}
```

Fonte: Autores, 2023.

- Retomando, dependendo do que o usuário informar em sua primeira interação, o *ENQUANTO* pode ou não ser executado;
- Ademais, o código-fonte prossegue com um final semelhante ao exemplo anterior:

Figura 13 - Desenvolvimento do código-fonte do ENQUANTO

```
    escreva("\nLembre-se que o contador se iniciou com a quantidade ZERO")  
    escreva("\nO contador executou " + nroContador + " o FAÇA ENQUANTO")  
}  
}
```

Fonte: Autores, 2023.

- Pois bem, a versão final de seu código-fonte será o seguinte:

Figura 14 - Desenvolvimento do código-fonte do ENQUANTO

```
programa  
{  
    funcao inicio()  
    {  
        inteiro nroContador  
        logico logRepete  
  
        nroContador = 0  
        logRepete = falso  
  
        escreva("Deseja iniciar? Digite verdadeiro para SIM E falso para NÃO\n")  
        leia(logRepete)  
  
        enquanto (logRepete == verdadeiro) {  
  
            nroContador += 1  
  
            escreva("Deseja continuar? Digite verdadeiro para SIM E falso para NÃO\n")  
            leia(logRepete)  
  
        }  
  
        escreva("\nLembre-se que o contador se iniciou com a quantidade ZERO")  
        escreva("\nO contador executou " + nroContador + " o FAÇA ENQUANTO")  
    }  
}
```

Fonte: Autores, 2023.



- Se está tudo funcionando perfeitamente, vamos para os exercícios!