

Retrospectiva : O que já aprendemos?

- Até aqui, já vimos
 - Valores
 - Como nos organizamos como turma
 - Valores em trabalho em equipe e desenvolvimento de software
 - Conteúdo
 - O que é um algoritmo
 - O que é um programa
 - Qual ferramenta utilizaremos
 - Operações de entrada e saída
 - O que são variáveis e constantes
 - Desvios condicionais (se e senão)
 - Operadores lógicos (E, OU ...)



Retrospectiva : O que iremos aprender?

- Laços de repetição (enquanto, para ... faça)
- Subrotinas (Funções)
 - Recursividade
 - Bibliotecas
- Estruturas de dados (Vetores, Matrizes, Filas e Pilhas)



Laços de repetições



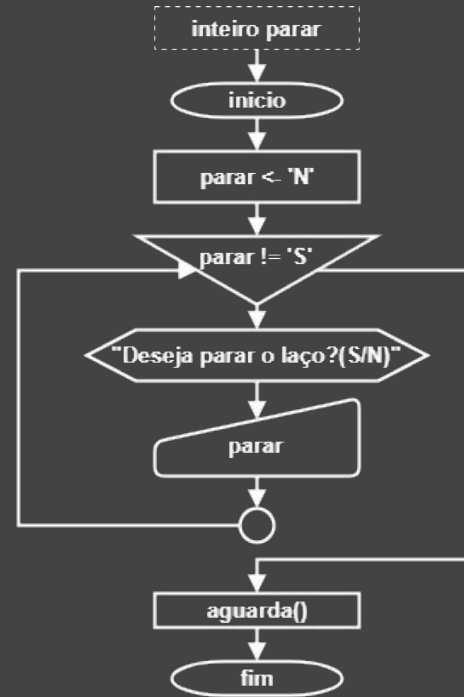
Sobre laços de repetição

- Se uma ação se repete em um algoritmo, em vez de escrevê-la várias vezes, em certos casos podemos resumir anotando uma vez só e solicitando que ela se repita, usando umas das **estruturas de repetição**.
- Podemos pedir que uma ação (ou um conjunto de ações) seja executada um número definido ou indefinido de vezes, ou enquanto um estado permanecer ou até que um estado seja atingido.
- Fora do Portugal, essas estruturas são denominadas do inglês , **while** (enquanto) e **for** (para)

Laço Enquanto (Pré-Testado)

programa

```
{  
  funcao inicio()  
  {  
    caracter parar  
    parar = 'N'  
  
    enquanto (parar != 'S')  
    {  
      escreva ("deseja parar o laço? (S/N)")  
      leia (parar)  
    }  
  }  
}
```

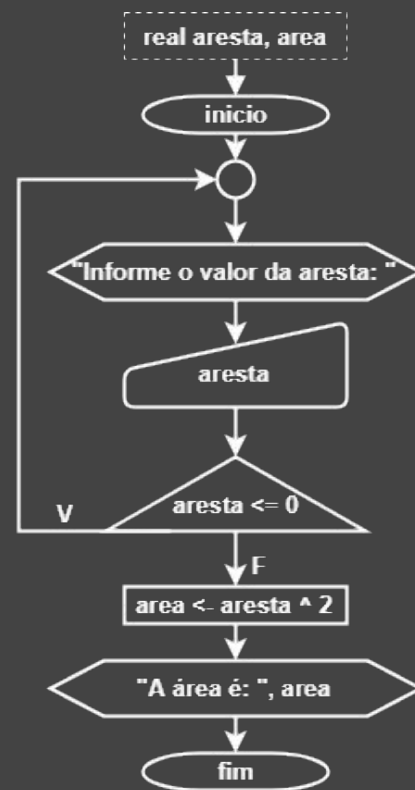


Laço Faça-Enquanto (Pós-Testado)

```
programa
{
  funcao inicio()
  {
    real aresta, area

    faca
    {
      escreva ("Informe o valor da aresta: ")
      leia (aresta)
    } enquanto (aresta <= 0)

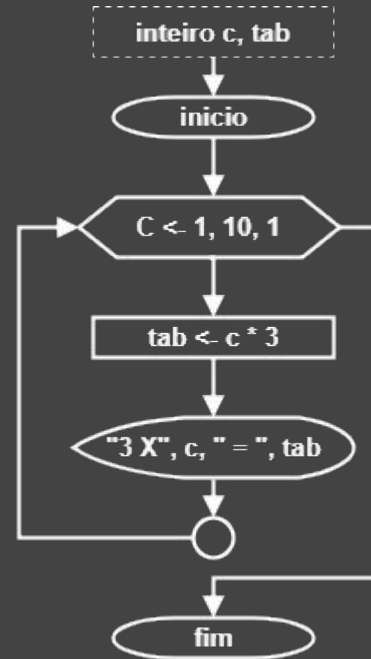
    area = aresta*aresta
    escreva("A área é: ", area)
  }
}
```



Laço Para (Com Variável de Controle)

```
programa
{
  funcao inicio()
  {
    inteiro tab

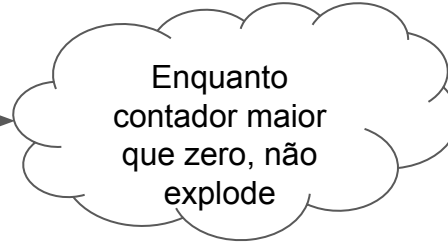
    para (inteiro c=1; c<=10; c++)
    {
      tab=c*3
      escreva ("3 x ", c, " = ", tab, "\n")
    }
  }
}
```



- Podemos colocar condições dentro da estrutura **enquanto**

programa

```
{
    inclua biblioteca Util
    funcao inicio() {
        inteiro contador = 10
        enquanto (contador > 0)
        {
            limpa()
            escreva ("Detonação em: ", contador)
            contador = contador - 1
            aguarde(1000) // Aguarda 1000 milisegundos (1 segundo)
        }
        limpa()
        escreva ("Booom!\n")
    }
}
```



Outro exemplo

Faça um programa usando o enquanto que escreva na tela números de 1 a 100.

```
programa
{
    funcao inicio()
    {
        inteiro numero
        numero = 1
        enquanto(numero<=100){
            escreva(numero +",")
            numero++
        }
        escreva("Fim")
    }
}
```

Faça o mesmo exercício usando o para.

```
programa
{
    funcao inicio()
    {
        para(inteiro numero = 1; numero<=100;numero++){
            escreva(numero +",")
        }
        escreva("Fim")
    }
}
```

Exercício

Escrever um programa de computador que leia números inteiros e ao final, apresente a soma de todos os números lidos até que o valor digitado seja zero.

```
programa
{

    funcao inicio()
    {

        inteiro numero, total = 0
        escreva("Digite o número:")
        leia(numero)
        enquanto(numero != 0){
            total = total + numero
            escreva("Digite o número:")
            leia(numero)
        }
        escreva("Total:" + total)

    }

}
```

O Enquanto é muito utilizado quando precisamos fazer testes e não sabemos quantas vezes será realizado. No exemplo anterior precisamos fazer a leitura do número duas vezes porque a estrutura enquanto testa no início, neste caso podemos utilizar o **faça enquanto** que testa no final.

```
programa
{
    funcao inicio()
    {
        inteiro numero, total=0
        faca{
            escreva("Digite o número:")
            leia(numero)
            total = total + numero
        }
        enquanto(numero !=0)
            escreva("Total:" + total)

    }
}
```

Além do **enquanto**, temos o **para... até ... faça**

- Imagine que queremos saber a tabuada de um número.
 - Quais são os requisitos?
 - Escolher um número
 - Multiplicar o número escolhido por 1 até 10
- Então **para 1 até 10 multiplique** o número escolhido.



Como fica o código??

Tabuada usando laços de repetição

O Para possui uma variável de controle, a qual podemos repetir um conjunto de instruções até um determinado número de vezes. A variável de controle é chamada de contador.

programa

```
{
    funcao inicio()
    {
        inteiro numero, resultado, contador

        escreva("Informe um número para ver sua tabuada: ")
        leia(numero)

        limpa()

        para (contador = 1; contador <= 10; contador++)
        {
            resultado = numero * contador
            escreva (numero, " X ", contador, " = ", resultado , "\n")
        }
    }
}
```

Note que ao usar o "para" temos uma estrutura facilitada para intervalos de repetição



Leia a idade de uma determinada quantidade de pessoas que também deverá ser informada pelo usuário e diga no final quantos são de maior e menor idade.

```
programa
{
    funcao inicio()
    {
        inteiro quantPessoas, idade, totalMaior18=0, totalMenor18=0
        escreva("Digite a quantidade de pessoas:")
        leia(quantPessoas)
        para(inteiro i=0; i<quantPessoas; i++){
            escreva("Digite a idade da pessoa:")
            leia(idade)
            se(idade >= 18){
                totalMaior18 ++
            }senao{
                totalMenor18 ++
            }
        }
        escreva("Total Maior de idade:" + totalMaior18, "\n")
        escreva("Total Menor de idade:" + totalMenor18)
    }
}
```

Exercícios

Faça um programa que leia um número e apresente como resultado a multiplicação de 10 até 0.

Exemplo: $3 \times 10 = 30$

$3 \times 9 = 27$

```
programa
{

    funcao inicio()
    {

        inteiro numero
        escreva("Digite o número:")
        leia(numero)
        para(inteiro i=10; i>=0;i=i-1){
            escreva("\n",numero,"x",i,"=",numero * i)
        }
    }
}
```

Exercícios

- 1) Escreva um programa que sirva como um cronômetro. O usuário deve digitar um número em segundos e o programa deve contabilizar os segundos digitados. O programa deve receber somente números entre 1 e 59, do contrário ele retornar um erro e solicitar que o usuário digite um número válido.
- 2) Desenvolva um programa que receba um número do usuário e escreva a tabuada dele.
- 3) Aproveite o código dos exercícios anteriores e escreva um código que receba dois números inteiros e escreva todos os números pares entre eles.
- 4) Um cliente que promove eventos e solicitou um programa que seja capaz de identificar se uma pessoa é maior de idade. Pessoas com menos de 16 anos não podem entrar nos eventos. Entre 16 e 18 anos somente acompanhado pelos responsáveis. Maiores de 18 podem entrar normalmente. (O programa deve executar até que o usuário digite 0)
- 5) Crie um algoritmo que receba login e senha e verifique as credenciais. Caso algum deles estiver errado o programa deve retornar ao usuário quais das opções está errada, se é o login ou a senha. O programa deve bloquear o acesso após 3 tentativas erradas. Quando for a última tentativa ele deve emitir um alerta:
"Última tentativa, mais um erro seu acesso será bloqueado!"
- 6) Escreva um programa onde o computador escolhe um número entre 1 e 100. O usuário deve tentar adivinhar o número, e o programa deve informar se o palpite está acima, abaixo ou correto. O jogo continua até que o usuário acerte.
- 7) Desenvolva um programa que solicite ao usuário que digite números positivos. O programa deve somar todos os números inseridos até que o usuário digite um número negativo. Ao final, o programa exibe a soma total.