

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARÁ
Campus Tianguá



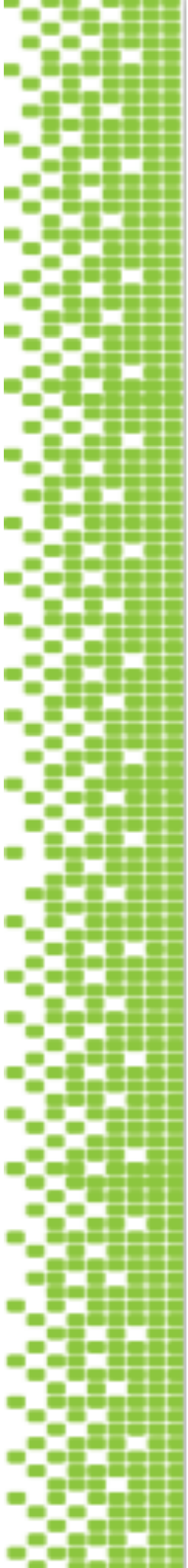
Estruturas de Repetição

Prof. David de Miranda Rodrigues





Conceito

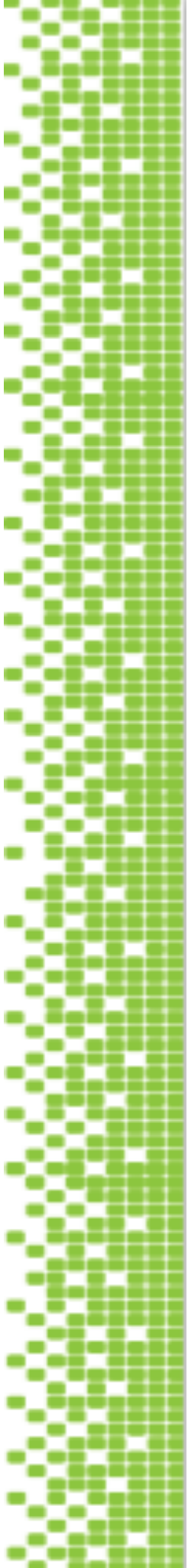
- Comandos que permitem a execução repetida de um determinado trecho de código do programa.
 - Com esse recurso o programador não precisa duplicar as linhas de código em virtude da necessidade de executá-las várias vezes.
 - São chamados de laço ou loops.
- 

Estruturas de Repetição

- Para()
- Enquanto()
- Faca{...}Enquanto()

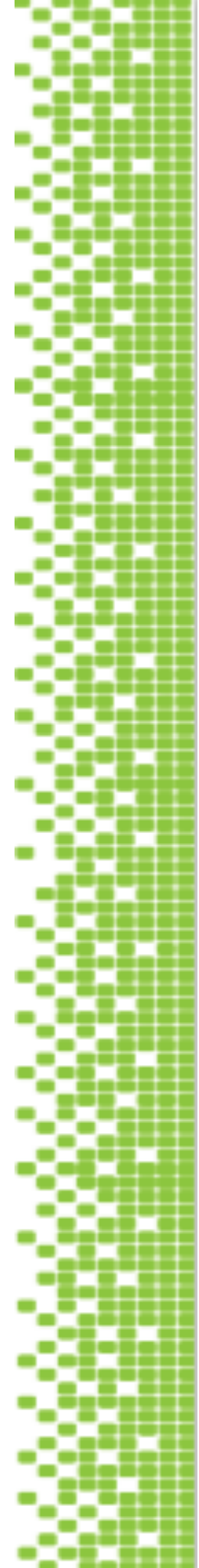


PARA

- Executa um trecho de código um número fixo de vezes que é determinado pelo programador.
 - É o tipo de estrutura mais simples entre todos e deve ser utilizado quando já sabemos antecipadamente quantas vezes as linhas de código devem ser executadas.
- 



PARA

- utiliza uma variável para controlar a contagem do número de vezes que o código foi executado.
 - Sintaxe da estrutura:
 - **PARA** (Inicialização; Condição; Incremento)
 - **PARA** (Inicialização; Condição; Decremento)
- 

Exemplo

```
1 programa
2 {
3     funcao inicio()
4     {
5         //inteiro i
6         para(inteiro i = 0; i < 10; i++){
7             escreva(i, "-")
8         }
9     }
10 }
```



0-1-2-3-4-5-6-7-8-9-

Exemplo

- O algoritmo que permite a montagem de uma tabuada de multiplicação de um número fornecido pelo usuário.

- Dicas:

- Identifique o que se repete.
- Identifique um valor que evolui em um certo passo (de 1 em 1 por exemplo)
- Tente transformar a repetição em uma única expressão com variáveis.

N = 5

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

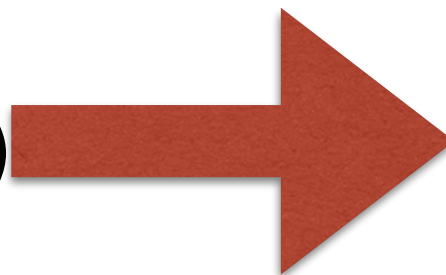
$$5 \times 10 = 50$$

Exemplo

- O algoritmo que permite a montagem de uma tabuada de multiplicação de um número fornecido pelo usuário.

- Tente transformar a repetição em uma única expressão com variáveis.

- **$N \times Mult = (N * Mult)$**



N = 5

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

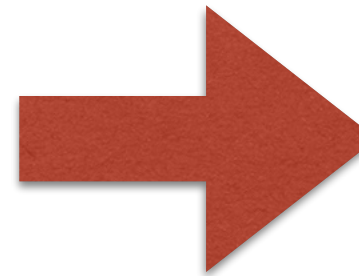
$$5 \times 10 = 50$$

Exemplo

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro n
6         escreva("Digite o número para gerar a calculadora: ")
7         leia(n)
8         para (inteiro mult=1; mult <= 10; mult++){
9             escreva(n, " X ", mult, "\t= ", (n*mult), "\n")
10        }
11    }
12 }
```

> Console Mensagens

Digite o número para gerar a calculadora: 5_



5 X 1 = 5

5 X 2 = 10

5 X 3 = 15

5 X 4 = 20

5 X 5 = 25


5 X 6 = 30

5 X 7 = 35

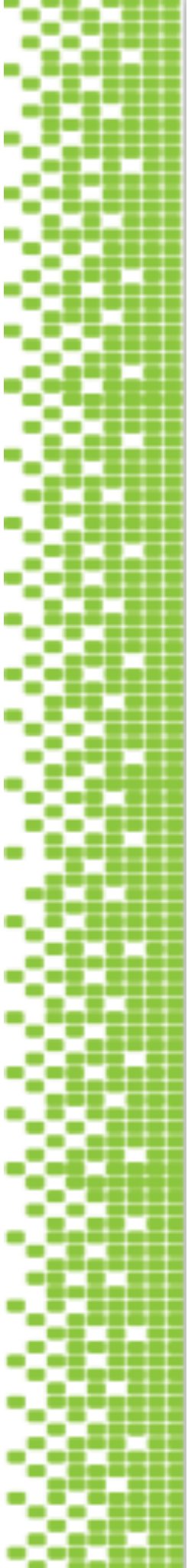
5 X 8 = 40


5 X 9 = 45

5 X 10 = 50

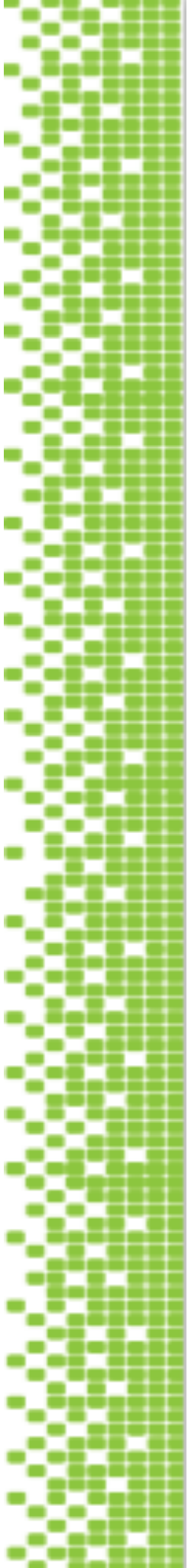


Enquanto

- Deve ser utilizada quando não sabemos antecipadamente a quantidade de vezes que o código deve ser executado.
 - O que encerra a repetição é:
 - uma ação do usuário (evento);
 - uma condição que o programa encontrou.
 - Enquanto o **resultado da condição** for **verdadeiro** o código é executado.
- 



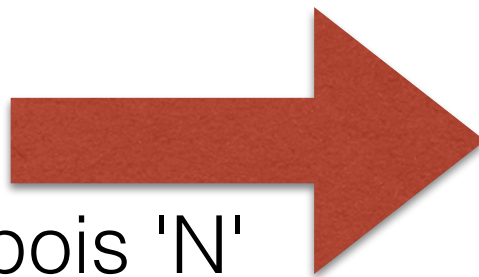
Enquanto

- A estrutura **Enquanto** analisa primeiramente a **expressão** antes de executar o código ou linhas de código. Se a expressão retornar um valor **falso** logo na primeira avaliação, o código não é executado nenhuma vez.
 - Sintaxe:
 - `while(expressão){...}`
- 

Exemplo

```
1 programa
2 {
3     funcao inicio()
4     {
5         caracter resp = 'S'
6         cadeia nome
7         enquanto(resp=='S'ou resp == 's'){
8             escreva("Digite um nome: ")
9             leia(nome)
10            escreva(nome, "\n")
11            escreva("Deseja informar novamente outro nome: [S/N]")
12            leia(resp)
13        }
14    }
15 }
```

Quando o usuário
responde 'S' e depois 'N'

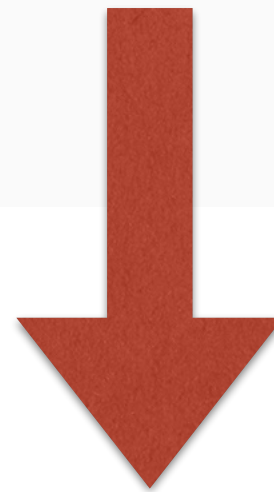


David de Miranda

David de Miranda

Exemplo

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro i=0
6
7         enquanto(i < 11){
8             escreva(i, "-")
9             i++ //incrementando uma unidade ao i
10        }
11    }
12 }
```



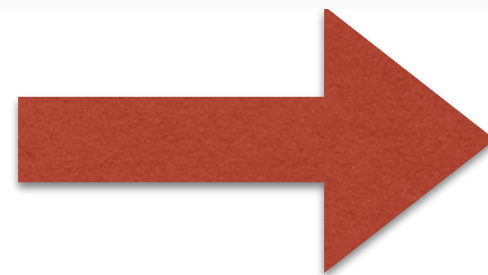
0-1-2-3-4-5-6-7-8-9-10-

Faca{...}Enquanto

- Difere um pouco da anterior pelo fato de efetuar a avaliação da expressão no fim do laço.
- O código sempre é executado pelo menos uma vez.
- Repete o código enquanto a condição dada for **verdadeira**.
- Empregada em situações onde não conhecemos a quantidade de repetições que um código deve ser executado, mas sim um evento que poderá parar a repetição.

Exemplo

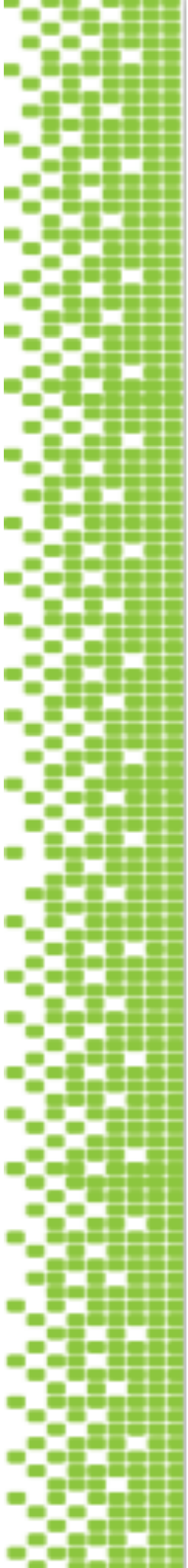
```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro i=0
6
7         faca{
8             escreva(i, "_")
9             i++ //incrementando uma unidade ao i
10        }enquanto(i < 11)
11    }
12 }
```



0-1-2-3-4-5-6-7-8-9-10-



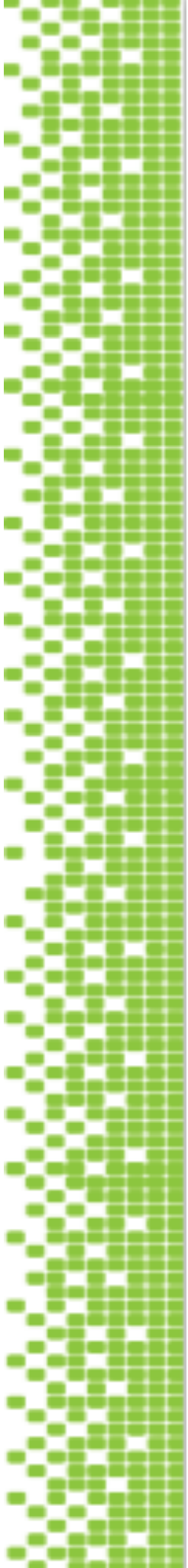
Exercícios

- **Para**
 - Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem decrescente.
 - Escreva um algoritmo para imprimir os 10 primeiros números inteiros maiores que 100.
- 



Exercícios

- **Enquanto**

- Escreva um algoritmo para ler 2 valores e se o segundo valor informado for ZERO, deve ser lido um novo valor, ou seja, para o segundo valor não pode ser aceito o valor zero e imprimir o resultado da divisão do primeiro valor lido pelo segundo valor lido.
 - Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem crescente.
- 



Exercícios

- **Faca{...}Enquanto**

- Escreva um algoritmo para ler 2 valores e mostre para o usuário o valor da soma desses valores. Depois questione ao usuário se este deseja fazer uma nova soma, para isso registre a resposta do usuário com valor 'S' ou 'N'. Caso ele responda 'S', todo processo inicial deverá se repetir.
 - Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem decrescente.
- 