

ESTRUTURA DE REPETIÇÃO ENQUANTO

MÓDULO DA DIVISÃO

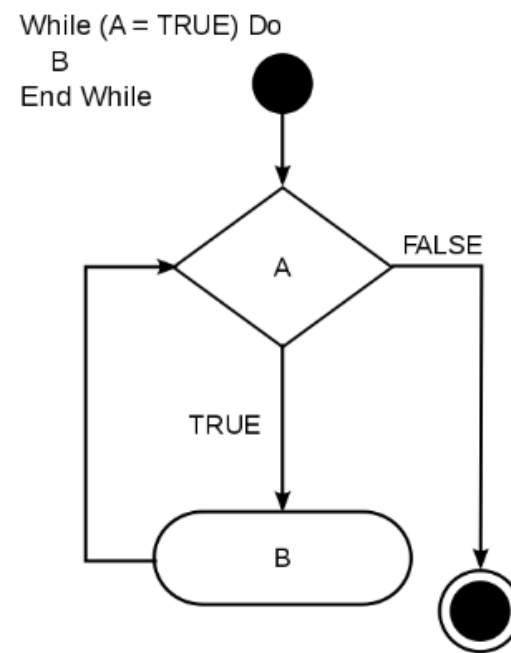
As Estruturas de REPETIÇÃO

- ▶ **O que é estrutura de repetição?**
- ▶ Dentro da lógica de programação é uma estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.
- ▶ São utilizadas, por exemplo, para repetir ações semelhantes que são executadas para todos os elementos de uma lista de dados, ou simplesmente para repetir um mesmo processamento até que a condição seja satisfeita.
- ▶ Existem 4 estruturas de repetição básica para praticamente todas as linguagens de programação, seja C ou javascript.

As Estruturas de REPETIÇÃO

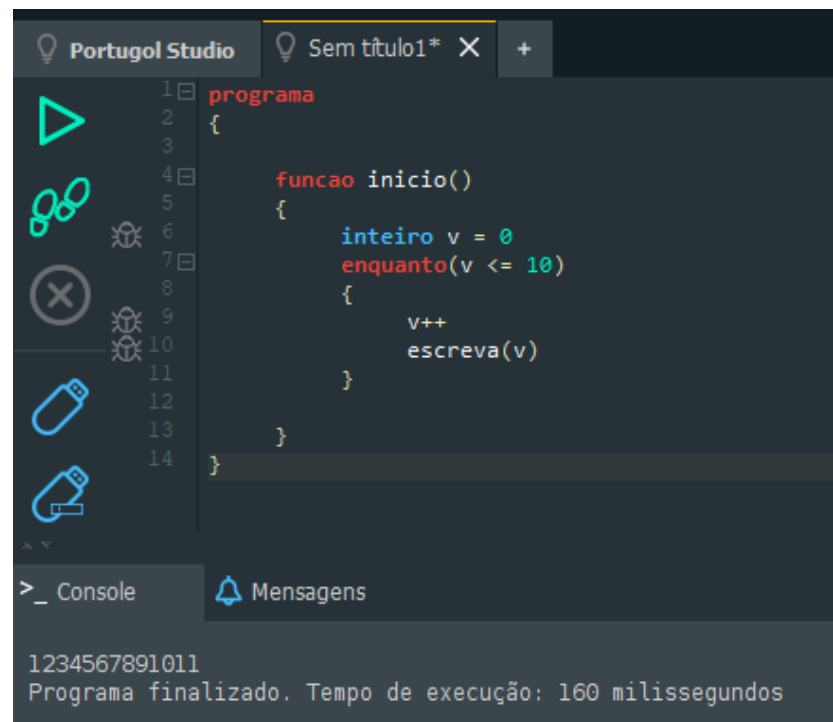
- ▶ É dentre as 3 a mais simples.
- ▶ Repete um bloco de código enquanto uma condição permanecer verdadeira
- ▶ Caso a condição seja falsa, os comandos dentro do while não serão executados e a execução continuará com os comandos após o while
- ▶ A repetição do while é controlada por uma condição que verifica alguma variável. Porém para que o while funcione corretamente é importante que essa variável sofra alteração dentro do while. Ex: um contador.
- ▶ Após entrar dentro da repetição, o bloco de comandos sempre será executado, mesmo que dentro do bloco a variável que está controlando a execução seja alterada.

As Estruturas de Repetição



While Loop

As Estruturas de Repetição



The screenshot shows the Portugol Studio IDE with a single window titled "Sem título1* X +". The main editor displays a program in a flowchart-like syntax. The code is as follows:

```
1 programa
2 {
3
4     funcao inicio()
5     {
6         inteiro v = 0
7         enquanto(v <= 10)
8         {
9             v++
10            escreva(v)
11        }
12    }
13
14 }
```

On the left side of the editor, there is a vertical toolbar with icons for running (a green play button), debugging (a magnifying glass over a bug), and other development tools. Below the editor, there are two panels: "_ Console" and "Mensagens". The "_ Console" panel shows the output of the program:

```
>_ Console
1234567891011
Programa finalizado. Tempo de execução: 160 milissegundos
```

As Estruturas de Repetição

► Interagindo com o usuário



The screenshot displays the Portugol Studio IDE interface. The main editor window shows a program with the following code:

```
1 programa
2 {
3     inteiro num
4
5     funcao inicio()
6     {
7         escreva("Digite um numero até 5 a 10")
8         leia(num)
9
10        enquanto(num <= 100) {
11            num++
12            escreva(num)
13        }
14    }
15 }
```

The left sidebar contains icons for running, debugging, and other IDE functions. The bottom panel is split into 'Console' and 'Mensagens' tabs. The Console tab shows the output of the program:

```
Digite um numero até 5 a 1010
111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485
Programa finalizado. Tempo de execução: 5849 milissegundos
```

As Estruturas de Repetição

- Interagindo com o usuário

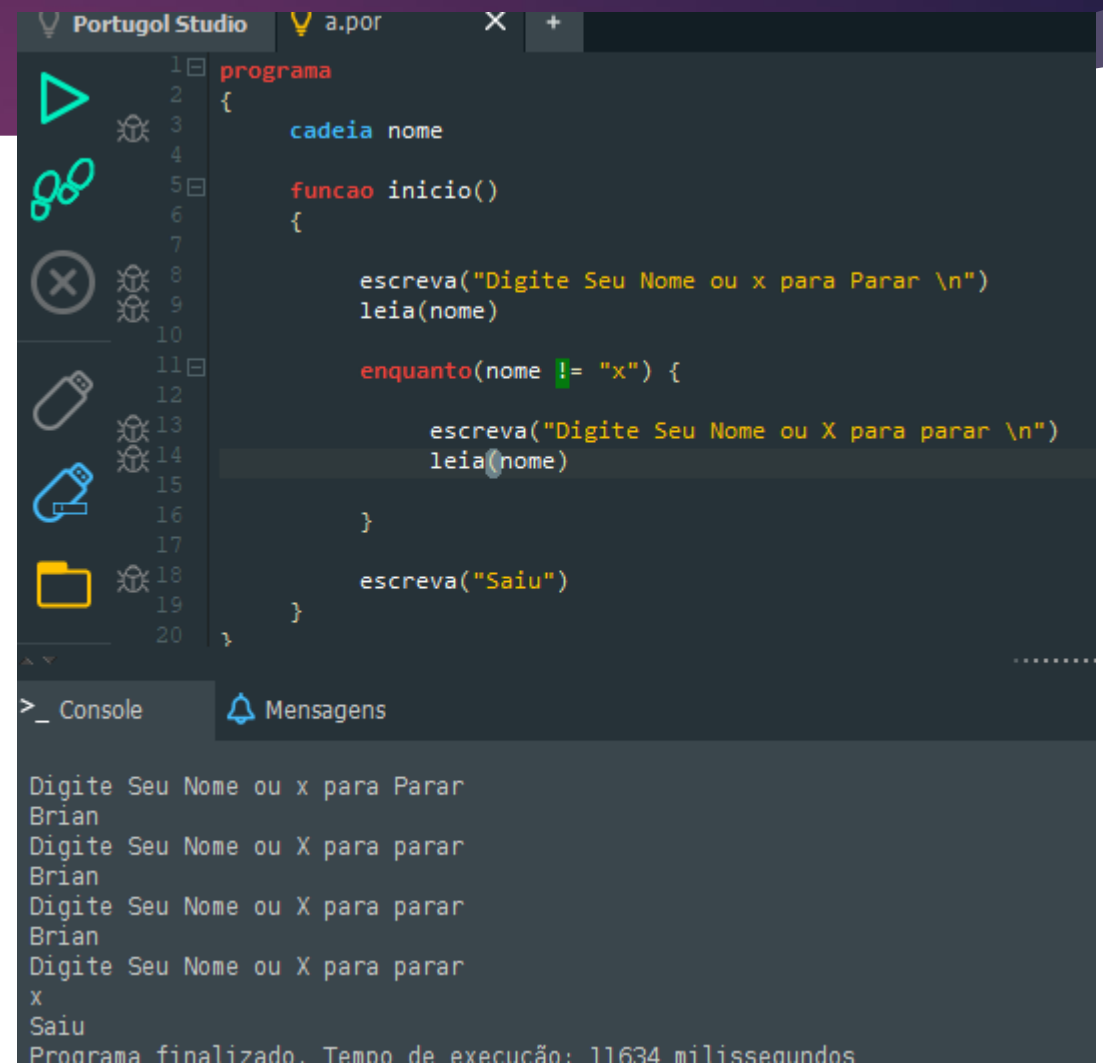


The screenshot shows the Portugol Studio IDE with a file named 'a.ppt'. The code is written in a structured programming style with icons on the left. The code defines a program with a function 'inicio()' that declares a string variable 'nome' and an integer variable 'num' initialized to 0. It then enters a 'while' loop ('enquanto') that runs as long as 'num' is less than or equal to 2. Inside the loop, it prompts the user to enter their name ('Digite Seu Nome \n') and reads the input ('leia(nome)'). After each iteration, it increments 'num' by 1 ('num ++').

```
1 programa
2 {
3
4     funcao inicio()
5     {
6
7         cadeia nome
8         inteiro num = 0
9
10        enquanto(num <= 2) {
11
12            escreva("Digite Seu Nome \n")
13            leia(nome)
14
15            num ++
16
17        }
18    }
19 }
```

As Estruturas de Repetição

- Interagindo com o usuário



The screenshot shows the Portugol Studio IDE with a file named 'a.por' open. The code is written in a flowchart-like syntax. It defines a 'programa' block containing a 'cadeia nome' variable, an 'inicio()' function, and a loop structure. The loop is an 'enquanto' (while) loop that continues as long as 'nome' is not equal to 'x'. Inside the loop, it prompts the user to enter their name and reads the input. After the loop, it prints 'Saiu' (Exit).

```
1 programa
2 {
3     cadeia nome
4
5     funcao inicio()
6     {
7
8         escreva("Digite Seu Nome ou x para Parar \n")
9         leia(nome)
10
11        enquanto(nome != "x") {
12
13            escreva("Digite Seu Nome ou X para parar \n")
14            leia(nome)
15
16        }
17
18        escreva("Saiu")
19    }
20 }
```

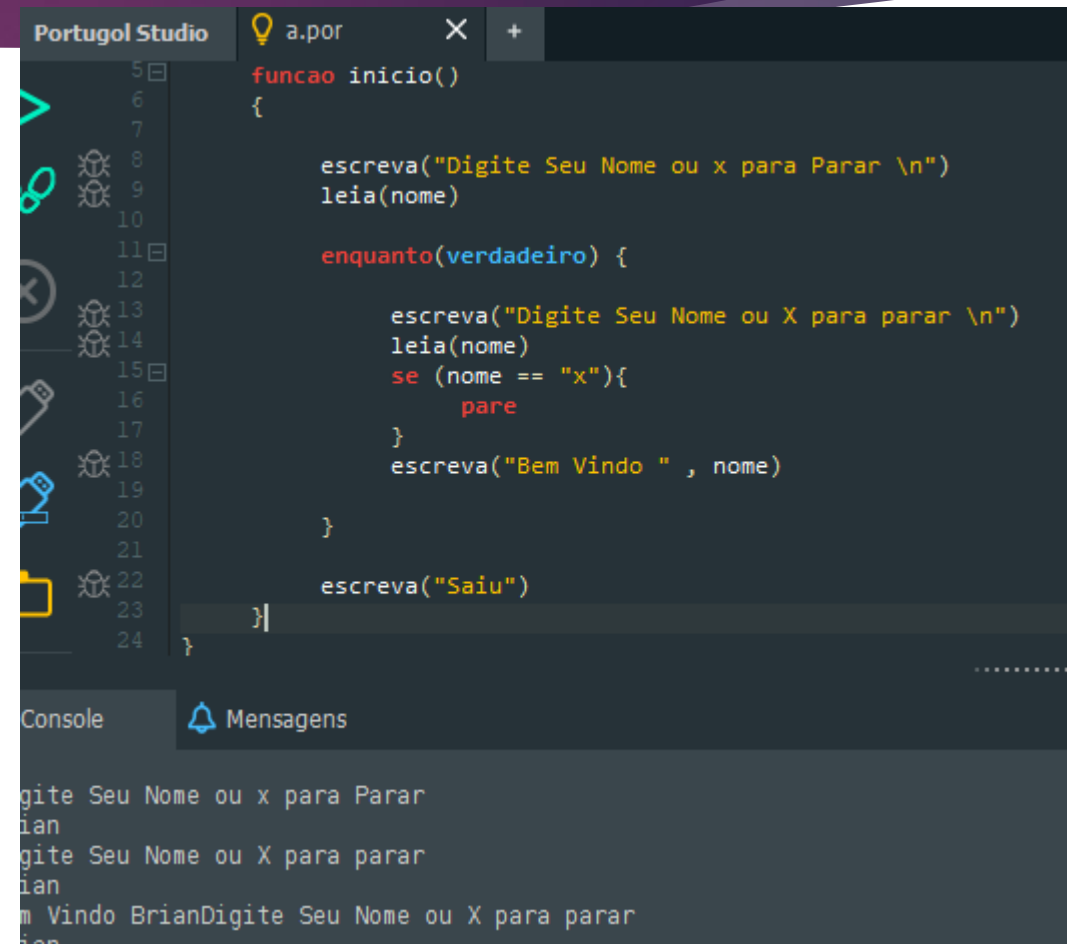
The console output shows the program's execution. It prompts the user to enter their name, and the user enters 'Brian'. This process repeats three times. Finally, the user enters 'x', which triggers the 'Saiu' message and ends the program. The execution time is 11634 milliseconds.

>_ Console

```
Digite Seu Nome ou x para Parar
Brian
Digite Seu Nome ou X para parar
Brian
Digite Seu Nome ou X para parar
Brian
Digite Seu Nome ou X para parar
x
Saiu
Programa finalizado. Tempo de execução: 11634 milissegundos
```


As Estruturas de Repetição

- ▶ Interagindo com o usuário
- ▶ Parando o Laço de Repetição
- ▶ Com comando informado pelo Usuário



The screenshot displays the Portugol Studio interface. The main editor window shows a program named 'a.por' with the following code:

```
5 funcao inicio()
6 {
7
8     escreva("Digite Seu Nome ou x para Parar \n")
9     leia(nome)
10
11     enquanto(verdadeiro) {
12
13         escreva("Digite Seu Nome ou X para parar \n")
14         leia(nome)
15         se (nome == "x"){
16             pare
17         }
18         escreva("Bem Vindo ", nome)
19
20     }
21
22     escreva("Saiu")
23 }
24 }
```

The bottom panel shows the 'Console' output, which includes the prompts and user input from the program's execution:

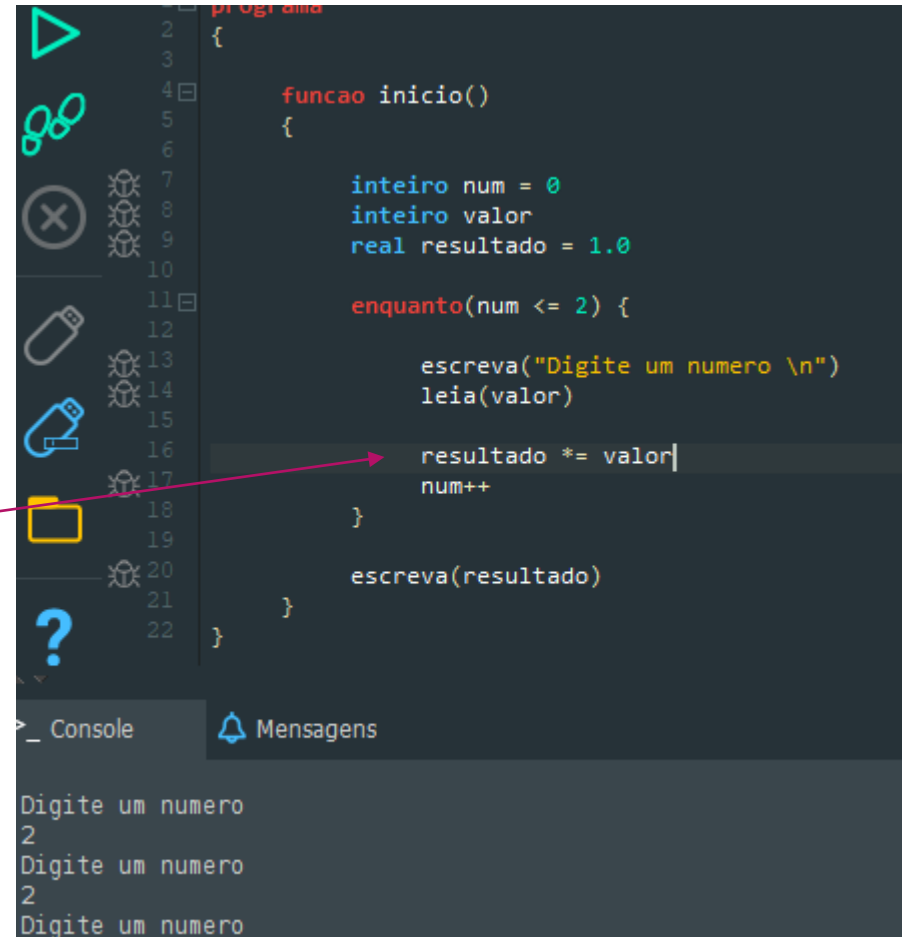
```
gite Seu Nome ou x para Parar
ian
gite Seu Nome ou X para parar
ian
m Vindo Brian
gite Seu Nome ou X para parar
ian
```

As Estruturas de Repetição

- ▶ Multiplicando Números inserido
- ▶ pelo usuário

resultado =
resultado * valor

**Atribuição
composta**



```
1  programa
2  {
3
4      funcao inicio()
5      {
6
7          inteiro num = 0
8          inteiro valor
9          real resultado = 1.0
10
11         enquanto(num <= 2) {
12
13             escreva("Digite um numero \n")
14             leia(valor)
15
16             resultado *= valor
17             num++
18         }
19
20         escreva(resultado)
21     }
22 }
```

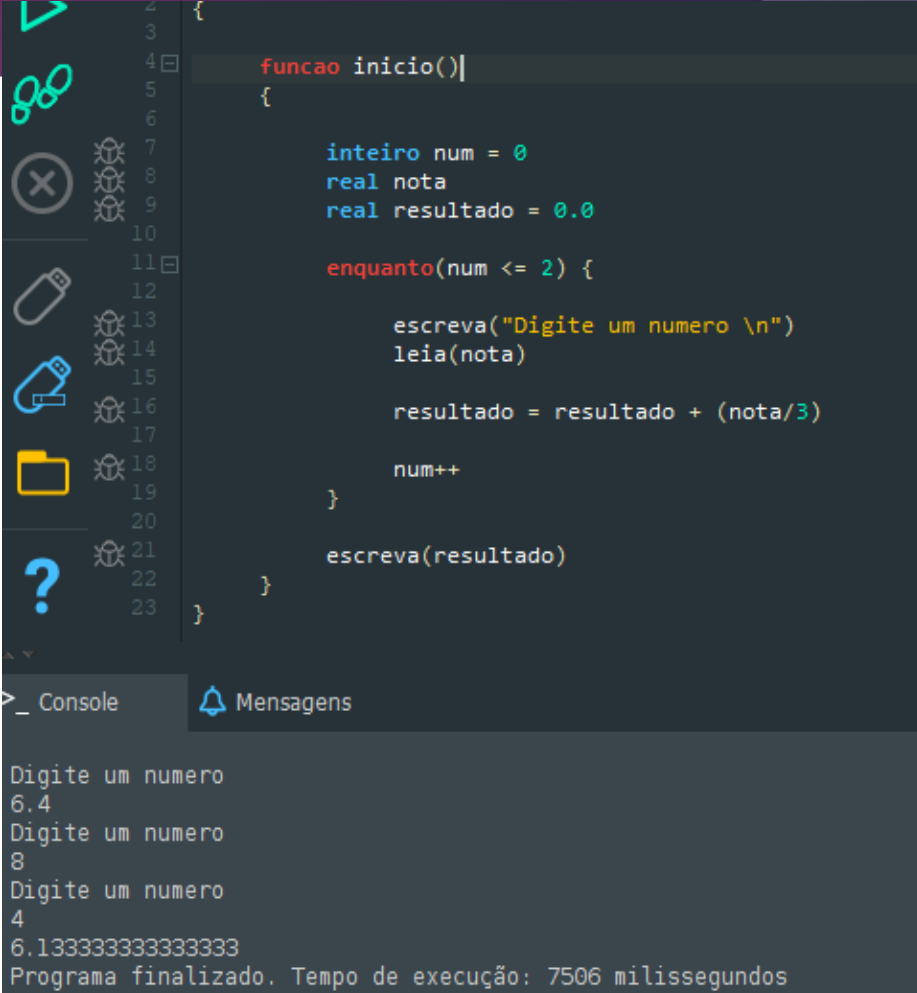
Console

Mensagens

Digite um numero
2
Digite um numero
2
Digite um numero

As Estruturas de Repetição

- ▶ Interagindo com o usuário
- ▶ Somando Números e Dividindo



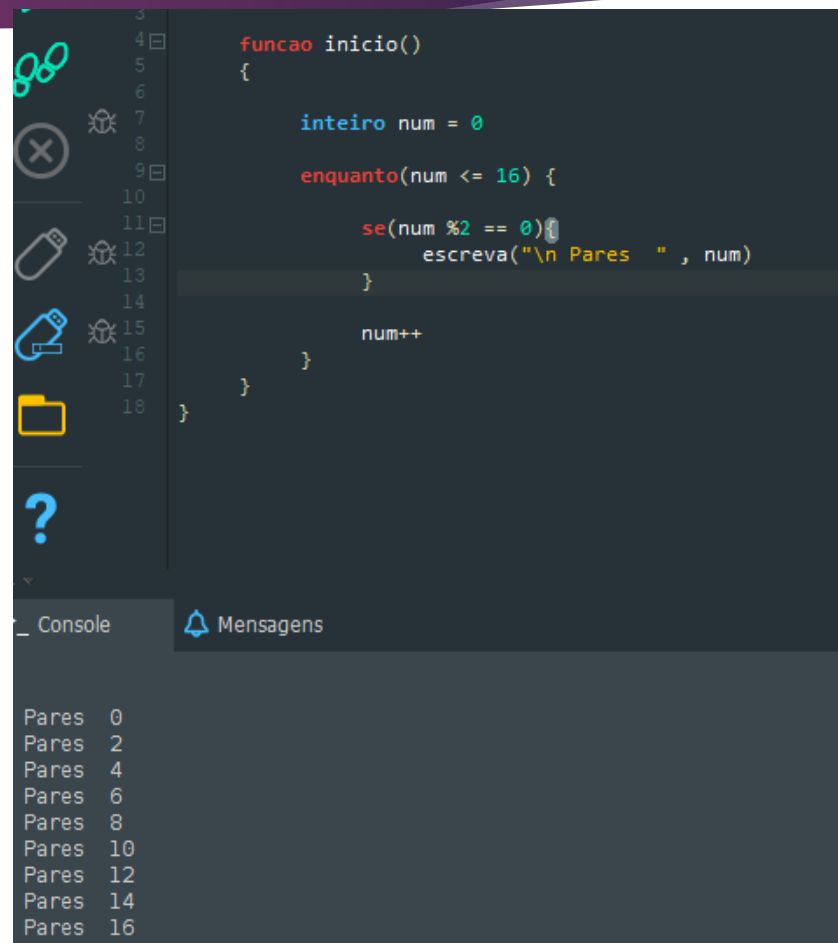
```
1 {  
2  
3  
4  
5 funcao inicio()  
6 {  
7  
8     inteiro num = 0  
9     real nota  
10    real resultado = 0.0  
11  
12    enquanto(num <= 2) {  
13  
14        escreva("Digite um numero \n")  
15        leia(nota)  
16  
17        resultado = resultado + (nota/3)  
18  
19        num++  
20    }  
21  
22    escreva(resultado)  
23 }
```

_ Console Mensagens

Digite um numero
6.4
Digite um numero
8
Digite um numero
4
6.1333333333333333
Programa finalizado. Tempo de execução: 7506 milissegundos

As Estruturas de Repetição

- ▶ Interagindo com o usuário
- ▶ Integrando Estruturas de Decisão



```
3  
4 funcao inicio()  
5 {  
6     inteiro num = 0  
7     enquanto(num <= 16) {  
8         se(num %2 == 0){  
9             escreva("\n Pares  ", num)  
10        }  
11        num++  
12    }  
13 }  
14  
15  
16  
17  
18
```

Console

```
Pares 0  
Pares 2  
Pares 4  
Pares 6  
Pares 8  
Pares 10  
Pares 12  
Pares 14  
Pares 16
```

Exercitando

- ▶ Mostre os números de 350 até 101 na tela.
- ▶ Leia cinco valores com o comando enquanto. No final mostre a soma dos 5 valores.
- ▶ Leia três valores, faça a multiplicação dos valores lidos e mostre o resultado.
- ▶ Desenvolva um algoritmo que mostre os números ímpares de 1 até 15.
- ▶ Leia cinco valores . No final mostre a soma dos 5 valores.
- ▶ Leia 3 valores com o comando enquanto e mostre a média.
- ▶ Determine a diferença de 2 valores. Faça o algoritmo executar 3 vezes.