

## ✓ RESUMO DE ALGUNS COMANDOS COMUNS NO PORTUGOL

### 1. **escreva()**

- **Saída de dados.**
- **escreva()** imprime algo na área de saída

```
escreva("Digite um número: ")  
escreva("Olá, mundo!")
```

---

### 2. **leia()**

- **Entrada de dados.**
- Recebe valor digitado pelo usuário.

```
leia(numero)
```

---

### 3. **se...senao**

- **Condicional (estrutura de decisão).**
- Estrutura de controle condicional. Ela permite que o programa execute diferentes blocos de código dependendo se uma determinada condição é verdadeira ou falsa.

```
programa {  
  funcao inicio() {  
    inteiro idade  
    escreva ("Digite sua idade: ")  
    leia (idade)  
    se (idade >= 18) {escreva("\nMaior de idade")  
    } senao {  
      escreva("\nMenor de idade")  
    }  
  }  
}
```

---

## 4. enquanto

- **Laço de repetição com condição.**
- Executa uma lista de comandos enquanto uma determinada condição for verdadeira.

```
programa
{
    funcao inicio()
    {
        caracter parar
        parar = 'N'

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

---

## 5. faca . . . enquanto

- **Estrutura de repetição pós-teste.**
- Isso significa que o bloco de código dentro do **faca** será executado **pelo menos uma vez** antes da condição no **enquanto** ser verificada. Se a condição for verdadeira, o bloco de código será executado novamente. Esse processo se repete até que a condição se torne falsa.

```
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)
    }
}
```

## 6. Para

- O laço de repetição com variável de controle facilita a construção de algoritmos com número definido de repetições, pois possui um contador (variável de controle) embutido no comando com o incremento automático.

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

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

Principais Comandos em Portugol		
Comandos de Entrada e Saída		
Comando	Função	Exemplo
<i>leia()</i>	Recebe um valor do usuário	<i>leia(nome)</i>
<i>escreva()</i>	Mostra um valor na tela	<i>escreva("Olá, ", nome)</i>
Comandos de Controle de Fluxo (Decisão)		
<i>se</i>	Inicia uma condição	<i>se (idade &gt;= 18) escreva ("Maior de idade")</i>
<i>senão</i>	Executa outra ação caso a condição seja falsa	<i>senão escreva("Menor de idade")</i>
Comandos de Repetição (Laços)		
<i>enquanto</i>	Repete enquanto a condição for verdadeira	<i>enquanto (x &lt; 10)</i>
<i>faca . . . enquanto</i>	Estrutura de repetição pós-teste.	<i>faca { escreva ("Informe o valor da aresta: ") leia (aresta) } enquanto (aresta &lt;= 0)</i>
<i>para</i>	<i>laço de repetição com variável de controle.</i>	<i>para (inteiro i = 0; i &lt; 8; i++)</i>
Comandos de Atribuição e Declaração		
<i>=</i>	Atribui valor a uma variável	<i>idade = 18</i>
<i>inteiro</i>	Declara variável do tipo inteiro	<i>inteiro idade</i>
<i>real</i>	Declara variável real (número com decimal)	<i>real nota</i>
<i>cadeia</i>	Declara variável de texto	<i>cadeia nome</i>
<i>logico</i>	Declara variável booleana	<i>logico aprovado</i>

Operações Relacionais			
Operação	Símbolo	Operação	Resultado
Maior	>	3 > 4	Falso
Menor	<	7 != 7	Falso
Maior ou igual	>=	9 == 10 - 1	Verdadeiro
Menor ou igual	<=	33 <= 100	Verdadeiro
Igual	=	6 >= 5 + 1	Verdadeiro
Diferente	!=	5 + 4 <= 11 - 2	Verdadeiro

# EXERCÍCIOS DE CÓDIGO EM PORTUGOL PARTE 1

## 1- Aprendendo a declarar variáveis

Variáveis são como caixinhas vazias que servem para guardar determinado objeto ou produto (toda caixinha deve ser etiquetada com um nome e o tipo de coisa que ela guarda).

Na programação, variáveis são espaços na memória que serão preenchidos por um tipo de dado (definido pelo programador ou usuário).

A estrutura básica de uma variável é `<tipo> <nome_da_variavel>` .

**VAMOS ELABORAR UM SISTEMA QUE MOSTRE OS DADOS DE UM ALUNO, SUAS NOTAS, A MÉDIA E SE ELE FOI APROVADO OU REPROVADO**

### CHECKLIST

- ☐ Declarar as variáveis
- ☐ Atribuir valores às variáveis
- ☐ Exibir os dados pessoais do aluno no console
- ☐ Verificar se ele está aprovado ou reprovado

- ESCREVA NO EDITOR DE CÓDIGO AS VARIÁVEIS ABAIXO:

nome

idade

nota1

nota2

media

turno

status (aprovado ou reprovado)

## Regras para dar nome às variáveis

Regra	Exemplo certo	Exemplo errado
Deve começar com uma letra	nome, idade1	1idade, @nome
Não pode usar acentos ou espaços	pesoCorporal	peso corporal
Evite palavras reservadas	inicio (❌ ruim)	iniciar (✅ bom)
Pode usar letras e números	nota1, endereco2	-
Nomes devem ser descritivos	mediaFinal (✅)	x, abc (❌)

- AGORA INDIQUE QUAL TIPO DE PRODUTO A CAIXINHA/VARIÁVEL VAI GUARDAR

inteiro (números sem casas decimais)

real (números com casas decimais)

caracter (letras)

lógico (variável booleana - verdadeiro ou falso)

cadeia (permite a inserção de palavras ou frases)

## 2- Aprendo a atribuir valores às variáveis

Depois de criar a caixinha, etiquetar com nome e o tipo de produto que ela vai guardar, é hora de colocar o produto dentro.

Na programação, é hora de ATRIBUIR VALORES A ESTAS VARIÁVEIS.

No editor de código, vamos atribuir os seguintes valores às variáveis:

```
nome = "Mariana"
idade = 27
nota1 = 10
nota2 = 10
turno = 'M'
media= (nota1+nota2)/2
status = (media>=7)
```

## 3- Interagindo com o console

Para que o aluno (usuário) veja seus dados pessoais, suas notas, suas médias e a situação, precisamos exibir estas informações em um espaço chamado console ou saída de dados.

Para isso, vamos utilizar o comando **ESCREVA**

```
escreva ("Nome: ", nome)
escreva ("\nIdade: ", idade)
escreva ("\nNota 1: ", nota1 )
escreva ("\nNota 2: ",nota2 )
escreva ("\nMédia: ",media )
escreva ("\nTurno: ", turno )
```

## 4- Aprendendo a utilizar os comandos/funções

Vamos verificar o status do aluno, se ele está aprovado ou reprovado

**SE** a média do aluno for MAIOR ou IGUAL a 7, ele está aprovado, **SENÃO** ele está reprovado.

```
se(status){ //também daria para escrever se(media>=7)

    escreva ("\nStatus: Aprovado!")

} senao {

    escreva ("\nStatus: Reprovado!")

}

}
```

### Código final

```
programa {

    funcao inicio() {

        //declaração de variáveis e atribuição de tipo
        cadeia nome
        inteiro idade
        real nota1, nota2, media
        logico status
        caracter turno

        // atribuição de valores as variáveis
        nome = "Mariana"
        idade = 27
        nota1 = 9
        nota2 = 10
        turno = 'M'

        media= (nota1+nota2)/2
        status = (media>=7)


        // exibição de resultado no console
        escreva ("Nome: ", nome)
        escreva ("\nIdade: ", idade, " anos") // \n= quebra de linha
        escreva ("\nNota 1: ", nota1, " pontos" )
        escreva ("\nNota 2: ",nota2, " pontos" )
        escreva ("\nMédia: ",media, " pontos" )
        escreva ("\nTurno: ", turno )

    }

}
```

```

    // teste para determinado a aprovação ou reprovação
(funcionalidade)
    se(status){ //também daria para escrever se(media>=7)
        escreva ("\nStatus: Aprovado!")
    } senao {
        escreva ("\nStatus: Reprovado!")
    }
}
}
//escreva ("Nome: ", nome, "\nIdade: ", idade, "\nNota 1: ",
nota1, "\nNota 2: ", nota2, "\nMédia: ", media, "\nTurno: ", turno )

```

## 5- Explorando a função LEIA

No Portugol, temos uma função muito importante que permite a interação do usuário com o sistema.

Pense em um formulário, o usuário precisa preencher cada campo com seus dados.

Dito isso, vamos adicionar mais uma função ao nosso sistema de notas de alunos.

Vamos supor que o aluno insere seus dados pessoais e suas notas, o sistema calcula a sua média e mostra o status (aprovado ou não).

Primeiramente, vamos declarar nossas variáveis que são as mesma do código anterior (basta copiar e colar).

```

cadeia nome
inteiro idade
real nota1, nota2, media
logico status
caracter turno

```

**Não vamos atribuir valores a estas variáveis, pois os valores serão atribuídos pelo usuário (então o usuário que vai guardar os produtos na caixinha e não o programador)**

Temos que estruturar as mensagens que vão aparecer no console e que o usuário deve preencher (assim como em um formulário)

```

    escreva("Digite seu nome: ")
    escreva("Digite sua idade: ")
    escreva("Digite seu turno (M para manhã, T para tarde, N para
noite): ")

```



```
escreva("Digite sua primeira nota: ")
escreva("Digite sua segunda nota: ")
```

```
Digite seu nome:
Digite sua idade:
Digite seu turno (M para manhã, T para tarde, N para noite):
Digite sua primeira nota:
Digite sua segunda nota:
```

Em seguida, vamos associar a cada questionamento a estrutura **leia(nome\_variável)**.

Isso significa que, a cada informação escrita pelo usuário, o sistema vai ler e guardar dentro da sua respectiva variável.

```
escreva("Digite seu nome: ")
    leia(nome)
    escreva("Digite sua idade: ")
    leia(idade)
    escreva("Digite seu turno (M para manhã, T para tarde, N para
noite): ")
    leia(turno)
    escreva("Digite sua primeira nota: ")
    leia(notas1)
    escreva("Digite sua segunda nota: ")
    leia(notas2)
```

Depois que o aluno digita suas notas, é hora de calcular a média para descobrir se ele está aprovado ou não. Para isso, vamos usar a mesma estrutura da **variável média** usada no código anterior.

```
media = (notas1 + notas2) / 2
```

Logo após, vamos elaborar uma espécie de MENU, onde todas as informações preenchidas pelo usuário serão listadas.

```
// Saída dos dados
    escreva("\n--- Resultado ---\n")
    escreva("Nome: ", nome)
    escreva("\nIdade: ", idade, " anos")
    escreva("\nTurno: ", turno)
    escreva("\nNota 1: ", notas1, " pontos")
    escreva("\nNota 2: ", notas2, " pontos")
    escreva("\nMédia: ", media, " pontos")
```

Por fim, precisamos definir se o aluno foi aprovado ou não, para isso, vamos usar a mesma estrutura **SE ... SENAO** do código anterior.

```
se (media >= 7)
{
    escreva("\nStatus: Aprovado! Parabéns :)")
}
senao
{
    escreva("\nStatus: Reprovado. Estude mais e tente novamente!")
}
```

## CÓDIGO COMPLETO

```
programa
{
    funcao inicio()
    {
        // Declaração de variáveis
        cadeia nome
        inteiro idade
        real nota1, nota2, media
        caracter turno

        // Entrada de dados
        escreva("Digite seu nome: ")
        leia(nome)

        escreva("Digite sua idade: ")
        leia(idade)

        escreva("Digite seu turno (M para manhã, T para tarde, N para noite): ")
        leia(turno)

        escreva("Digite sua primeira nota: ")
        leia(nota1)

        escreva("Digite sua segunda nota: ")
        leia(nota2)

        // Cálculo da média
        media = (nota1 + nota2) / 2
```

```
// Saída dos dados
escreva("\n--- Resultado ---\n")
escreva("Nome: ", nome)
escreva("\nIdade: ", idade, " anos")
escreva("\nTurno: ", turno)
escreva("\nNota 1: ", nota1, " pontos")
escreva("\nNota 2: ", nota2, " pontos")
escreva("\nMédia: ", media, " pontos")

// Verificação do status
se (media >= 7)
{
    escreva("\nStatus: Aprovado! Parabéns :)")
}
senao
{
    escreva("\nStatus: Reprovado. Estude mais e tente
novamente!")
}
}
```

## 6 EXERCÍCIOS DE CÓDIGO EM PORTUGOL PARTE 2

### 1. Ler um número e exibir o dobro dele

```
programa {  
    funcao inicio() {  
        inteiro numero, dobro  
        escreva("Digite um número: ")  
        leia(numero)  
        dobro = numero * 2  
        escreva("\n O dobro é: ", dobro)  
    }  
}
```

---

### 2. Verificar se o número é par ou ímpar (com entrada do usuário)

```
programa {  
    funcao inicio() {  
        //algoritmo "par_ou_impar"  
        //var  
        inteiro numero  
        //inicio  
        escreva("Digite um número: ")  
        leia(numero)  
        se (numero % 2 == 0) { // condição para verificar se o resto da  
divisão por 2 é igual a 0  
            escreva("Par")  
        } senao {  
            escreva("Ímpar")  
        }  
    }  
}
```

---

### 3. Calcular a média de duas notas e verificar se o aluno foi aprovado ou reprovado (com entrada do usuário)

```
programa {
    funcao inicio() {
        //algoritmo "media_notas"
        //var
        real nota1, nota2, media
        escreva("Digite a primeira nota: ")
        leia(nota1)
        escreva("Digite a segunda nota: ")
        leia(nota2)
        media = (nota1 + nota2) / 2
        escreva("\nMédia: ", media)
        se (media >= 7) {
            escreva("\nAprovado!")
        } senao {
            escreva ("\n Reprovado!")
        }
    }
}
```

---

### 4. Jogo de Adivinhação com Laço **enquanto** (com entrada do usuário)

Objetivo: O usuário deve tentar adivinhar um número secreto entre 1 e 10. O programa continua rodando enquanto o usuário não acertar.

```
programa
{
    funcao inicio()    {
        inteiro numeroSecreto, tentativa
        numeroSecreto = 7 // você pode trocar esse número
        tentativa = 0

        escreva("***Jogo de Adivinhação!***")
        escreva("\nTente adivinhar o número secreto de 1 a 10.")

        enquanto (tentativa != numeroSecreto)
        {
            escreva("\nDigite sua tentativa: ")
            leia(tentativa)
```

```

        se (tentativa != numeroSecreto) {
            escreva("Errado! Tente novamente.")
        } senao {
            escreva("Parabéns! Você acertou o número!")
        }
    }
}
}

```

## 5. Cadastro com Validação de Nome e Idade (com entrada do usuário)

Objetivo: O usuário deve informar:

- Um nome válido
- Uma idade válida (maior que 0)

O programa só continua quando as informações forem corretas.

```

programa
{
    funcao inicio()
    {
        cadeia nome
        inteiro idade

        // Validação básica: não deixar nome em branco
        faca
        {
            escreva("Digite seu nome: ")
            leia(nome)
        } enquanto (nome == "")

        // Validação da idade
        faca
        {
            escreva("Digite sua idade (maior que 0): ")
            leia(idade)
        } enquanto (idade <= 0)

        escreva("\nCadastro realizado com sucesso!")
        escreva("\nNome: ", nome)
    }
}

```

```
        escreva("\nIdade: ", idade)
    }
}
```

---

## 6. Calcular a tabuada

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