

Linguagem Portugol

Uma linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

O Portugol é uma representação que se assemelha bastante com a linguagem C, porém é escrito em português. A ideia é facilitar a construção e a leitura dos algoritmos usando uma linguagem mais fácil aos alunos.

Tipos

Quais são os tipos de dados que o computador pode armazenar?

Se pararmos para pensar que tudo que o computador compreende é representado através de Zeros e Uns. Então a resposta é Zero e Um. Certo? Certo! Mas como então o computador pode exibir mensagens na tela, apresentar ambientes gráficos cheios de janelas, compreender o significado das teclas do teclado ou dos cliques do mouse.

Bom tudo começa com a definição de uma série de códigos. Por exemplo. A letra "a" do teclado é representada pela seguinte sequência de zeros e uns "01000001". O número 22 é representado por "00010110". E assim todos os dados que são armazenados pelo computador podem ser representados em zeros e uns.

Sendo assim, existem alguns tipos básicos de dados nos quais valores podem ser armazenados no computador. O Portugol exige que o tipo de dado de um valor seja do mesmo tipo da variável ao qual este valor será atribuído.

Nesta seção, serão abordados os seguintes tópicos:

- Tipo Cadeia
- Tipo Character
- Tipo Inteiro
- Tipo Lógico
- Tipo Real
- Tipo Vazio

Tipo Cadeia

Em algumas situações precisa-se armazenar em uma variável, um texto ou uma quantidade grande de caracteres. Para armazenar este tipo de conteúdo, utiliza-se uma variável do tipo cadeia. Cadeia é uma sequência ordenada de caracteres (símbolos) escolhidos a partir de um conjunto pré-determinado.

A sintaxe é a palavra reservada **cadeia** seguida do nome da variável

Exemplo de Sintaxe

<i>cadeia</i> nome_da_variavel

O valor que essa variável assumirá poderá ser especificado pelo programador, ou solicitado ao usuário (ver Operação de Atribuição). Caso seja especificado pelo programador, o conteúdo deve estar acompanhado de aspas duplas.

Exemplo

```
programa
{
    funcao início()
    {
        cadeia nome1, nome2

        nome1 = "Variável declarada através de atribuição"
        //variável declarada através de atribuição do programador
        escreva ("Digite seu nome: ")
        leia (nome2)
        //variável declarada através de entrada do usuário
        escreva ("\\nOlá ", nome2)
    }
}
```

Tipo Character

Em determinadas situações faz-se necessário o uso de símbolos, letras ou outro tipo de conteúdo. Por exemplo, em um jogo da velha, seriam necessárias variáveis que tivessem conteúdos de 'X' e 'O'. Para este tipo de situação, existe a variável do tipo **character**. A variável do tipo **character** é aquela que contém uma informação composta de apenas UM carácter alfanumérico ou especial. Exemplos de caracteres são letras, números, pontuações e etc.

A sintaxe é a palavra reservada **character** e em seguida um nome para variável

Exemplo de Sintaxe

```
character nome_da_variavel
```

O valor que essa variável assumirá poderá ser especificado pelo programador ou solicitado ao usuário (ver Operação de Atribuição). Caso seja especificado pelo programador, o conteúdo deve estar acompanhado de aspas simples.

Exemplo de Sintaxe

```
programa
{
    funcao inicio()
    {
        character vogal, consoante
        vogal = 'a'
        //variável declarada através de atribuição do programador

        escreva ("Digite uma consoante: ")
        leia (consoante)
        //variável declarada através de entrada do usuário

        escreva ("Vogal: ", vogal, "\n", "Consoante: ", consoante)
    }
}
```

Tipo Inteiro

Em determinadas situações faz-se necessário a utilização de valores inteiros em um algoritmo. Como faríamos, por exemplo, uma simples soma entre dois números pertencentes ao conjunto dos números inteiros? Simples. Utilizando variáveis do tipo inteiro. Uma variável do tipo inteiro pode ser entendida como uma variável que contém qualquer número que pertença ao conjunto dos números inteiros. Podem ser positivos, negativos ou nulos.

A declaração de uma variável do tipo inteiro é simples.

A sintaxe é a palavra reservada **inteiro** e em seguida um nome para variável

Exemplo de Sintaxe

<i>inteiro</i> nome_da_variavel
--

O valor que essa variável assumirá poderá ser especificado pelo programador ou solicitado ao usuário (ver Operação de Atribuição).

Exemplo

```
programa
{
    funcao inicio()
    {
        inteiro num1, num2
        num1 = 5
        num2 = 3
        escreva (num1 + num2)
    }
}
```

Tipo Lógico

Em determinadas situações faz-se necessário trabalhar com informações do tipo verdadeiro e falso. Este tipo de necessidade aparece muito em operações relacionais para exibir se determinada condição é verdadeira ou falsa. Por exemplo: como poderíamos verificar se um número digitado pelo usuário é maior que zero? Através de uma variável do tipo logico. Uma variável do tipo logico é aquela que contém um tipo de dado, usado em operações lógicas, que possui somente dois valores, que são consideradas pelo Portugol como verdadeiro e falso.

A declaração de uma variável do tipo logico é simples. A sintaxe é a palavra reservada **logico** seguida do nome da variável.

Exemplo de Sintaxe

logico *nome_da_variavel*

O valor que essa variável assumirá poderá ser especificado pelo programador ou solicitado ao usuário (ver Operação de Atribuição). Lembrando que em ambos os casos a variável só assume valores verdadeiro ou falso.

Exemplo

```
programa
{
    funcao inicio()
    {
        logico teste
        inteiro num

        escreva ("Digite um valor para ser comparado :")
        leia (num)

        teste = (num>0)

        escreva ("O número digitado é maior que zero? ", teste)
    }
}
```

Tipo Real

Em algumas situações é necessário armazenar valores que não pertencem aos números inteiros. Por exemplo, se quiséssemos armazenar o valor da divisão de 8 por 3, como faríamos? Este problema pode ser solucionado com uma variável do tipo real. Uma variável do tipo real armazena um número real como uma fração decimal possivelmente infinita, como o número PI 3.1415926535. Os valores do tipo de dado real são números separados por pontos e não por virgulas.

A sintaxe para a declaração é a palavra reservada real junto com o nome da variável.

Exemplo de Sintaxe

<i>real</i> nome_da_variavel

O valor que essa variável assumirá poderá ser especificado pelo programador ou solicitado ao usuário (ver Operação de Atribuição).

Para melhor compreensão deste conceito, confira o exemplo abaixo.

Exemplo

<pre>programa { funcao inicio() { real div div = 8.0/3.0 escreva (div) } }</pre>
--

Entrada e Saída

Entrada/saída é um termo utilizado quase que exclusivamente no ramo da computação (ou informática), indicando entrada (inserção) de dados por meio de algum código ou programa, para algum outro programa ou hardware, bem como a sua saída (obtenção de dados) ou retorno de dados, como resultado de alguma operação de algum programa, consequentemente resultado de alguma entrada.

A instrução de entrada de dados possibilita que o algoritmo capture dados provenientes do ambiente externo (fora da máquina) e armazene em variáveis. Assim um algoritmo consegue representar e realizar operações em informações que foram fornecidas por um usuário tais como: nome, idade, salário, sexo, etc. A forma mais comum de capturar dados é através do teclado do computador. Por meio dele o usuário pode digitar números, palavras, frases etc.

A instrução de saída de dados permite ao algoritmo exibir dados na tela do computador. Ela é utilizada para exibir mensagens, resultados de cálculos, informações contidas nas variáveis, etc.

Nesta seção, serão abordados os seguintes tópicos:

- *Escreva*
- *Leia*

Escreva

Em determinadas situações precisamos mostrar ao usuário do programa alguma informação. Para isso, existe um comando na programação que exibe dados ao usuário. No português a instrução de saída de dados para a tela é chamada de "escreva", pois segue a ideia de que o algoritmo está escrevendo dados na tela do computador.

O comando escreva é utilizado quando deseja-se mostrar informações no console da IDE, ou seja, é um comando de saída de dados.

Para utilizar o comando escreva, você deverá escrever este comando e entre parênteses colocar a(s) variável(eis) ou texto que você quer mostrar no console. Lembrando que quando você utilizar textos, o texto deve estar entre aspas. A sintaxe para utilização deste comando está demonstrada a seguir:

Exemplo de Sintaxe

<i>escreva</i> ("Escreva o texto a ser digitado aqui")

Note que quando queremos exibir o valor de alguma variável não utilizamos aspas. Para exibição de várias mensagens em sequência, basta separá-las com vírgulas.

Existem duas ferramentas importantes que auxiliam a organização e visualização de textos exibidos na tela. São elas: o quebra-linha e a tabulação.

O quebra-linha é utilizado para inserir uma nova linha aos textos digitados. Sem ele, os textos seriam exibidos um ao lado do outro. Para utilizar este comando, basta inserir "\n". O comando de tabulação é utilizado para inserir espaços maiores entre os textos digitados. Para utilizar este comando, basta inserir "\t".

Exemplo

```
programa
{
    funcao inicio()
    {
        inteiro variavel=5

        //escreve no console um texto qualquer
        escreva ("Escreva um texto aqui.\n")

        //escreve no console o valor da variável "variável"
        escreva (variavel, "\n")

        //escreve no console o resultado da operação
        escreva (variavel+variavel, "\n")

        //escreve no console o texto digitado, e o valor contido na variável
        escreva ("O valor da variável é: ", variavel)

        //escreve no console o texto com quebra de linha
        escreva ("Texto com\n", "quebra-linha")

        //escreve no console o texto com espaço de tabulação
        escreva ("Texto com\t tabulação")
    }
}
```

Leia

Em alguns problemas, precisamos que o usuário digite um valor a ser armazenado. Por exemplo, se quisermos elaborar um algoritmo para calcular a média de nota dos alunos, precisaremos que o usuário informe ao algoritmo quais as suas notas. No português a instrução de entrada de dados via teclado é chamada de "leia", pois segue a ideia de que o algoritmo está lendo dados do ambiente externo (usuário) para poder utilizá-los.

O Comando leia é utilizado quando se deseja obter informações do teclado do computador, ou seja, é um comando de entrada de dados. Esse comando aguarda um valor a ser digitado e o atribui diretamente na variável.

Para utilizar o comando leia, você deverá escrever este comando e entre parênteses colocar a(s) variável (eis) que você quer que recebam os valores a serem digitados. A sintaxe deste comando está exemplificada a seguir:

Exemplo de Sintaxe

inteiro *x*

cadeia *y*

real *z*

//chamando o comando leia

leia(*x*)

leia(*y*, *z*)

//No final as variáveis irão possuir o valor digitado pelo usuário.

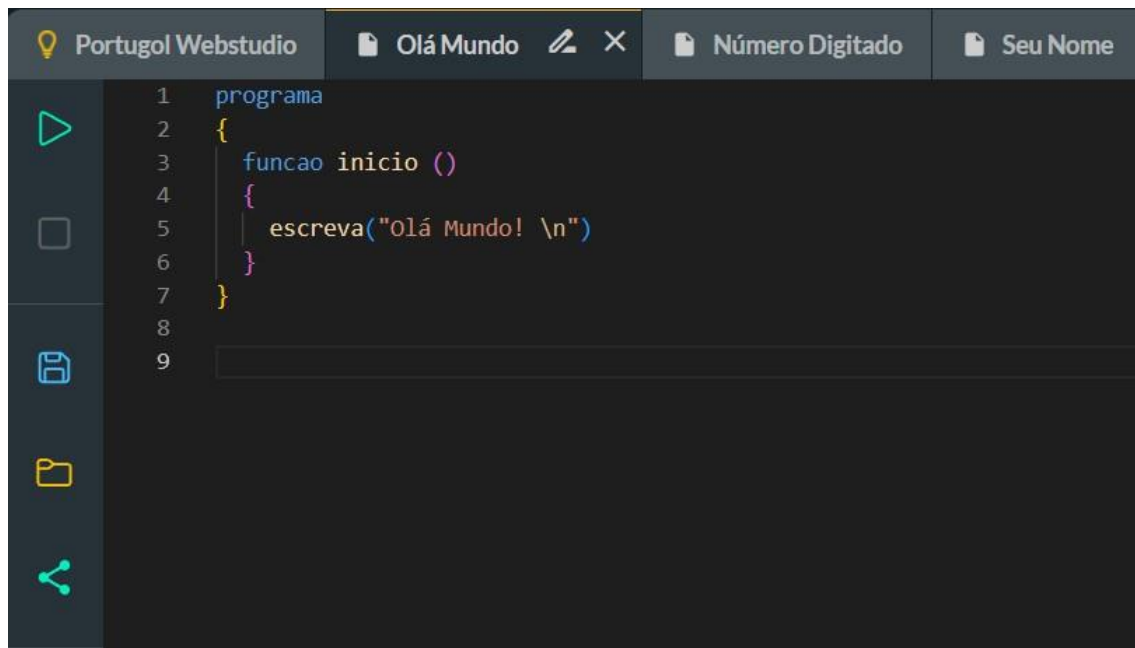
Note que para armazenar um valor em uma variável, é necessário que a mesma já tenha sido declarada anteriormente. Assim como no comando escreva, se quisermos que o usuário entre com dados sucessivos, basta separar as variáveis dentro dos parênteses com vírgula.

Exemplo

```
programa
{
    funcao inicio()
    {
        inteiro idade
        real salario, nota1, nota2, nota3
        cadeia nome, sobrenome

        escreva("Informe a sua idade: ")
        leia (idade)
        //lê o valor digitado para "idade"
        escreva("Informe seu salario: ")
        leia (salario)
        //lê o valor digitado para "salario"
        escreva("Informe o seu nome e sobrenome: ")
        leia (nome, sobrenome)
        //lê o valor digitado para "nome" e "sobrenome"
        escreva("Informe as suas três notas: ")
        leia (nota1, nota2, nota3)
        //lê o valor digitado para "nota1", "nota2" e "nota3"

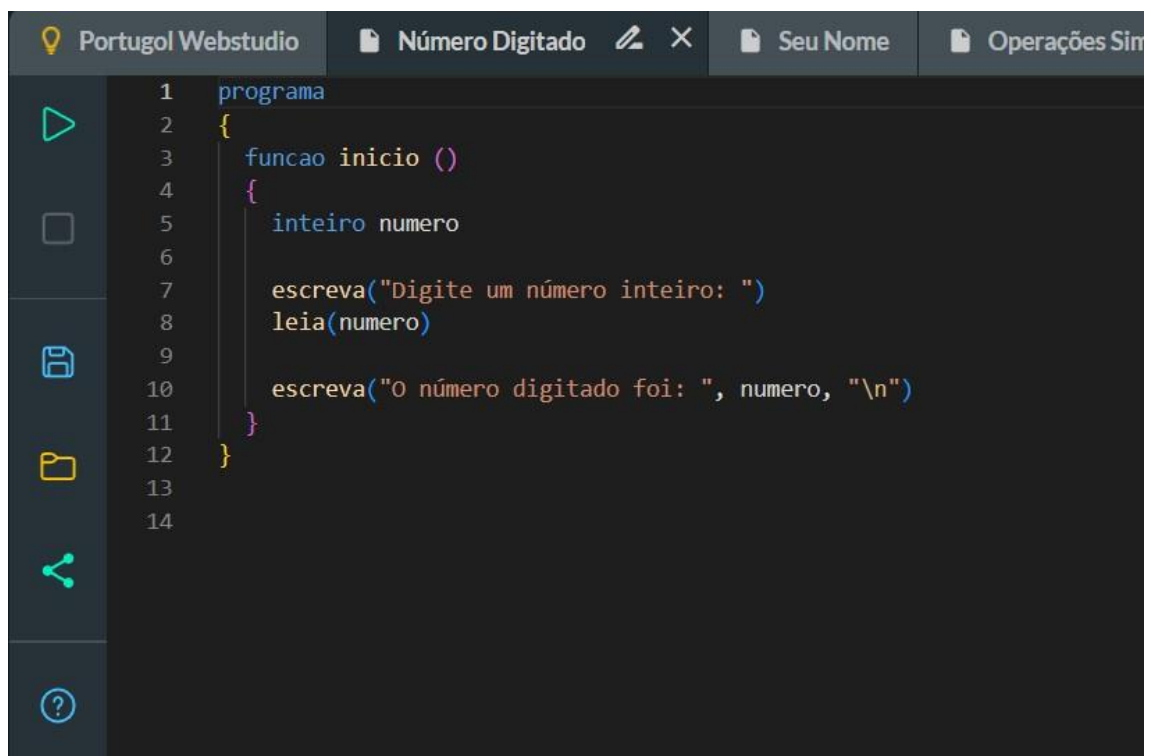
        escreva("Seu nome é:"+nome+" "+sobrenome+"\n")
        escreva("Você tem "+idade+" anos e ganha de salario "+salario+"\n")
        escreva("Suas três notas foram:\n")
        escreva("Nota 1: "+nota1+"\n")
        escreva("Nota 2: "+nota2+"\n")
        escreva("Nota 3: "+nota3+"\n")
    }
}
```



The screenshot shows the Portugol Webstudio interface. The top bar contains the logo and the text 'Portugol Webstudio'. Below it, there are tabs for 'Olá Mundo', 'Número Digitado', and 'Seu Nome'. The main editor area displays a program with the following code:

```
1 programa
2 {
3     funcao inicio ()
4     {
5         escreva("Olá Mundo! \n")
6     }
7 }
8
9
```

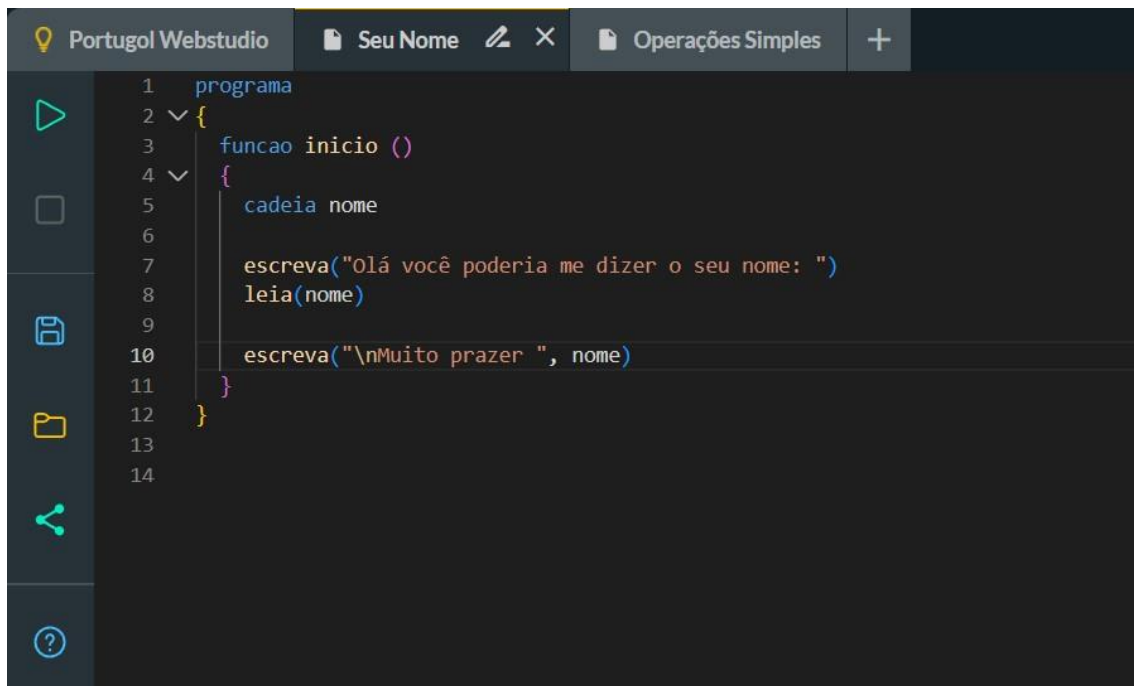
Figura 1: Exemplo de Escreva



The screenshot shows the Portugol Webstudio interface with a different program. The top bar contains the logo and the text 'Portugol Webstudio'. Below it, there are tabs for 'Número Digitado', 'Seu Nome', and 'Operações Sim'. The main editor area displays a program with the following code:

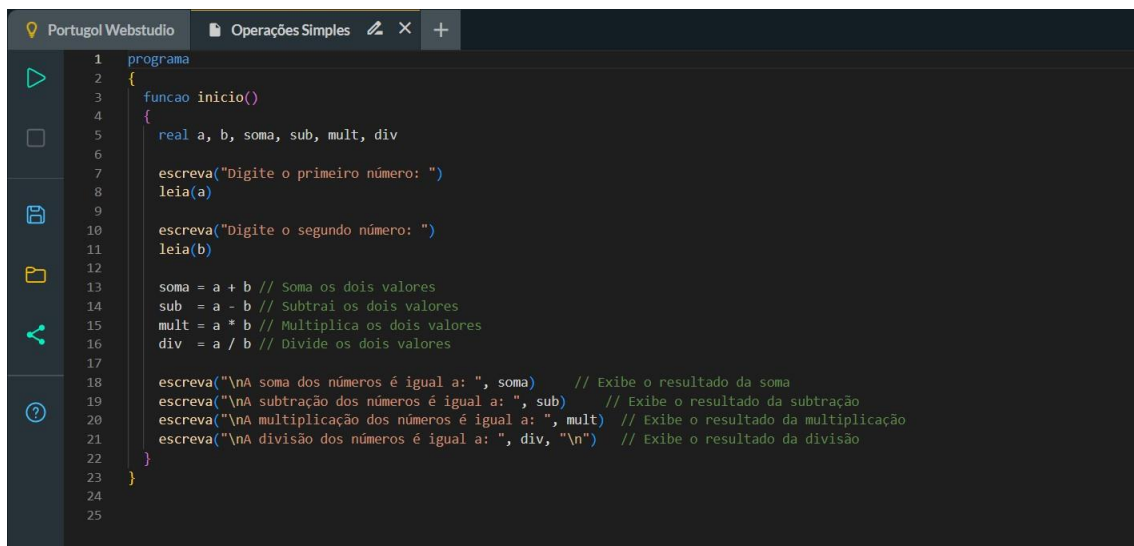
```
1 programa
2 {
3     funcao inicio ()
4     {
5         inteiro numero
6
7         escreva("Digite um número inteiro: ")
8         leia(numero)
9
10        escreva("O número digitado foi: ", numero, "\n")
11    }
12 }
13
14
```

Figura 2: Exemplo de Escreva e Leia



```
1 programa
2 {
3     funcao inicio ()
4     {
5         cadeia nome
6
7         escreva("Olá você poderia me dizer o seu nome: ")
8         leia(nome)
9
10        escreva("\nMuito prazer ", nome)
11    }
12 }
13
14
```

Figura 3: Exemplo de Cadeia



```
1 programa
2 {
3     funcao inicio()
4     {
5         real a, b, soma, sub, mult, div
6
7         escreva("Digite o primeiro número: ")
8         leia(a)
9
10        escreva("Digite o segundo número: ")
11        leia(b)
12
13        soma = a + b // Soma os dois valores
14        sub = a - b // Subtrai os dois valores
15        mult = a * b // Multiplica os dois valores
16        div = a / b // Divide os dois valores
17
18        escreva("\nA soma dos números é igual a: ", soma) // Exibe o resultado da soma
19        escreva("\nA subtração dos números é igual a: ", sub) // Exibe o resultado da subtração
20        escreva("\nA multiplicação dos números é igual a: ", mult) // Exibe o resultado da multiplicação
21        escreva("\nA divisão dos números é igual a: ", div, "\n") // Exibe o resultado da divisão
22    }
23 }
24
25
```

Figura 4: Exemplo de Real e Operações Matemáticas