



```
programa
{
    funcao inicio()
    {
        escreva("Olá Mundo")
    }
}
```

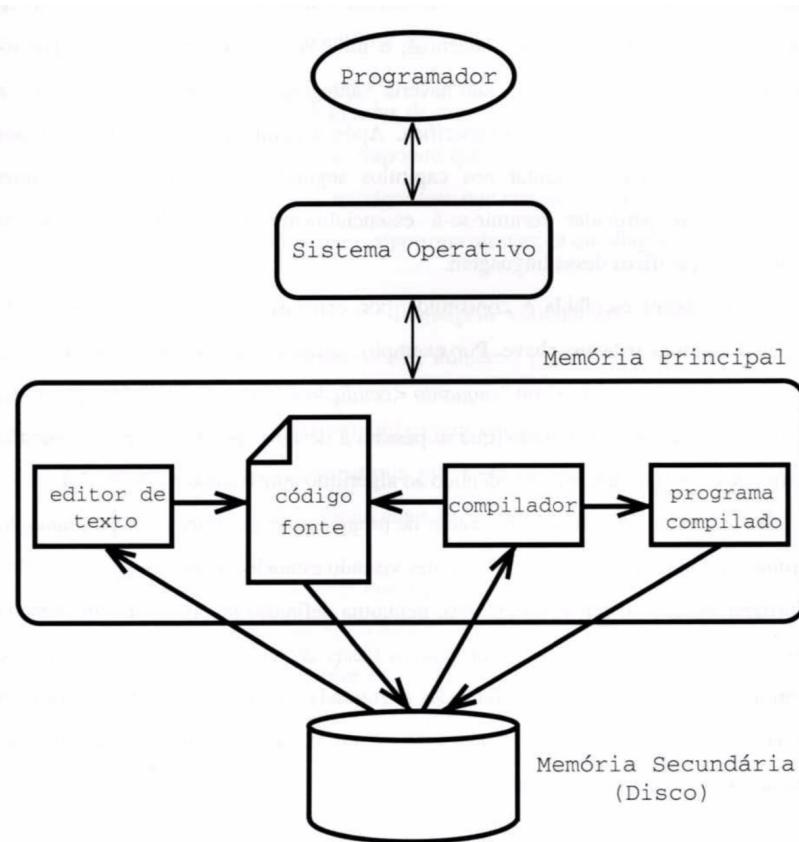
Português Estruturado (ou *pseudo-código*)

- Linguagem algorítmica com uma sintaxe e uma estrutura semelhante a uma linguagem de programação real;
- Escrever um algoritmo em Português Estruturado (PE) não é mais do que recorrer a um texto escrito em Português mas sujeito a um conjunto de regras e com uma estrutura próxima do que seria o algoritmo desenvolvido com uma linguagem de programação real;
- Em PE, os algoritmos são representado de forma textual “traduzindo” por palavras os símbolos gráficos das representações através de fluxogramas;
- A versão de PE a usar é o Portugol, existente na plataforma Portugol Studio.

Linguagem de Programação

- As linguagens de programação são formas de **alto nível** (significa que estão próximas dos utilizadores/programadores) de comunicar instruções a um computador.
- Uma **linguagem de programação** é um conjunto de palavras reservadas, sujeitas a um conjunto de regras sintáticas e semânticas, que indicam ao computador como os dados de um problema são lidos, armazenados e processados de forma a produzir uma solução para o problema.
- O programa escrito numa linguagem de alto nível designa-se por **código-fonte**.
- Para poder ser executado, o código-fonte tem de ser traduzido para **linguagem binária (código binário ou código executável)**.

Desenvolvimento de Programas



- O código-fonte é escrito com um **editor de texto**.
- O código-fonte é traduzido para código binário (código executável) através de um programa especial designado por **compilador**.
- O programa compilado é o código binário que pode ser executado no computador.
- Existem também **interpretadores** que são programas que traduzem e executam as instruções do código-fonte, uma a uma, sem produzir código executável.

Portugol

- O Portugol é uma linguagem algorítmica, uma forma de Português Estruturado, próxima da linguagem de programação C e não muito distante de outras linguagens como Java ou C#.



- O Portugol Studio é um **ambiente integrado de desenvolvimento** (IDE – *Integrated Development Environment*) que junta numa só aplicação um editor de texto (para escrever o código-fonte), o tradutor do código-fonte para código binário, o ambiente para executar e ver os resultados do programa e ainda funcionalidades para pesquisa e deteção de erros, execução passo a passo, gestão de ficheiros, etc.

Portugol

- **Portugol Studio IDE:** <https://univali-lite.github.io/Portugol-Studio/>

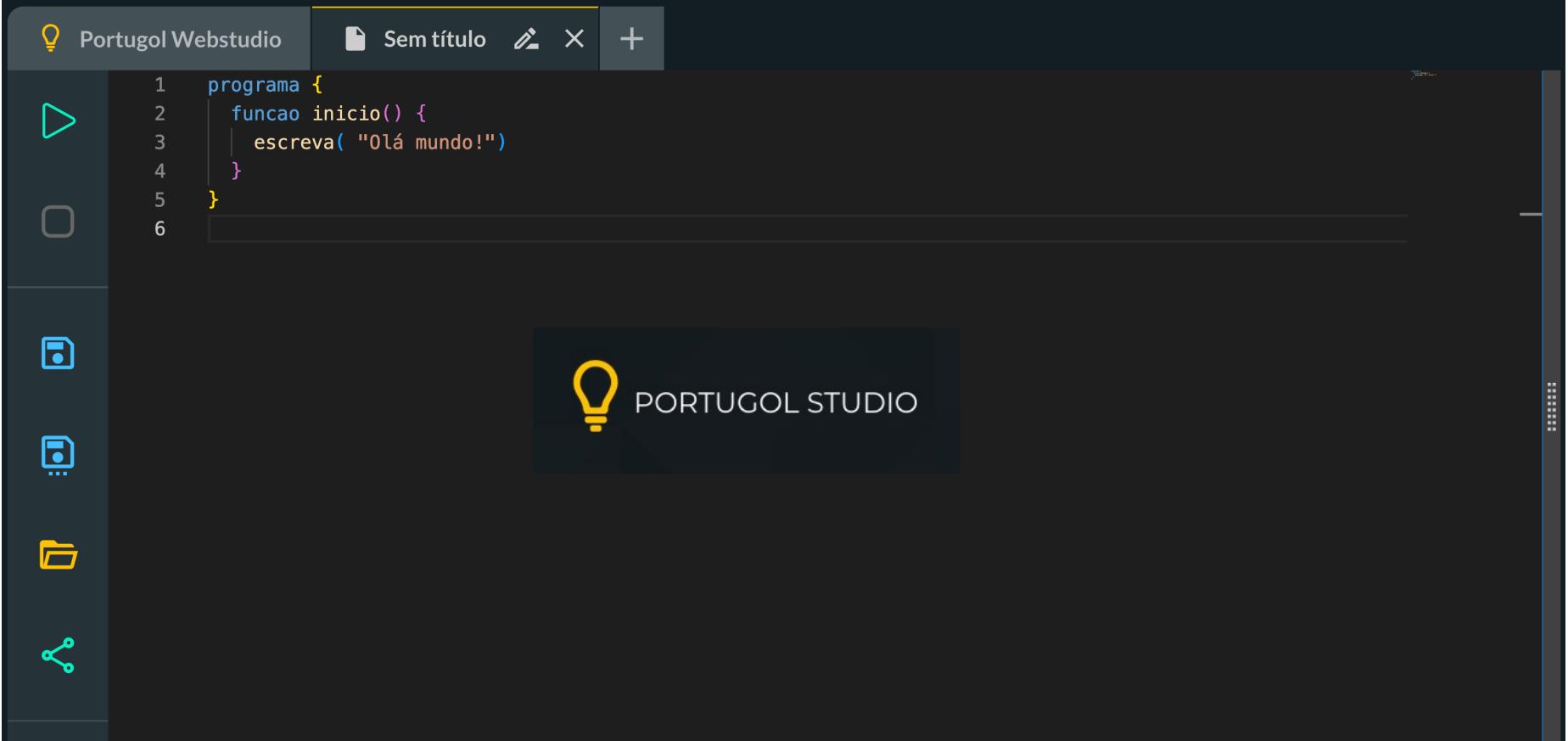


- **Portugol Webstudio:** <https://portugol.dev/>





<https://portugol.dev/>



The image shows a screenshot of the Portugol Webstudio interface. The top bar includes a logo, the text "Portugol Webstudio", a file icon, the text "Sem título", a pencil icon, a close button, and a plus sign button. The main area is a code editor with the following text:

```
1  programa {
2    funcao inicio() {
3      escreva( "Olá mundo!")
4    }
5  }
```

The sidebar on the left contains several icons: a play button, a square, a file with a dot, a file with three dots, a folder, and a share icon. A central banner displays the "PORTUGOL STUDIO" logo with a yellow lightbulb icon.

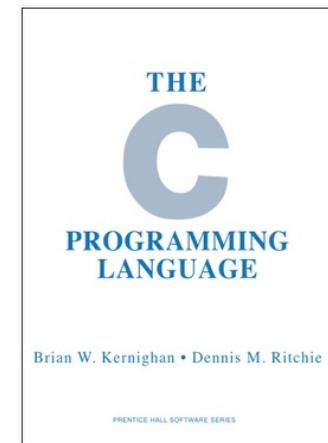
"Hello World"

A "Hello, World!" is a simple program that outputs `Hello, World!` on the screen. Since it's a very simple program, it's often used to introduce a new programming language to a newbie.

```
main()
{
    printf("hello, world\n");
}
```



"Hello, World!" program by [Brian Kernighan](#) (1978)



"Hello World"

```
programa
{
    funcao inicio()
    {
        escreva("Olá Mundo")
    }
}
```

```
HelloWorld.java x
1  package helloworld;
2
3  public class HelloWorld {
4
5      public static void main(String[] args) {
6          System.out.println("Hello World!");
7      }
8
9  }
10
11
12
13
```

```
hello x
#!/usr/bin/python3

print("Hello, World!")
```

```
Hello_World.c x
1  #include <stdio.h>
2  int main() {
3      printf("Hello, World!\n");
4      return 0;
5
6 }
```

"Hello World"

```
programa
{
    funcao inicio()
    {
        escreva("Olá Mundo")
    }
}
```



PORTUGOL STUDIO

Estrutura de um programa

//O comando programa é obrigatório
programa//Inclusões de bibliotecas
// - Quando houver a necessidade de utilizar
// uma ou mais bibliotecas, as inclusões
// devem aparecer antes de qualquer declaração
/*
* Dentro do programa é permitido declarar
* variáveis globais, constantes globais e
* funções em qualquer ordem.
*///Declarações de funções somente
//só são permitidas dentro do programa.
funcao inicio()/*
* Declarações de variáveis locais,
* constantes locais, estruturas de
* controle e expressões.
*/{}



(fluxogramas)

```
//O comando programa é obrigatório
programa{//Inclusões de bibliotecas
// - Quando houver a necessidade de utilizar
// uma ou mais bibliotecas, as inclusões
// devem aparecer antes de qualquer declaração
/*
 * Dentro do programa é permitido declarar
 * variáveis globais, constantes globais e
 * funções em qualquer ordem.
 *///Declarações de funções somente
//são permitidas dentro do programa.
funcãoinicio()/*
 * Declarações de variáveis locais,
 * constantes locais, estruturas de
 * controle e expressões.
 */}
```

```
programa () {  
    ...  
}  
}
```

(livro “Introdução à Programação”)

//O comando programa é obrigatório
programa//Inclusões de bibliotecas
// - Quando houver a necessidade de utilizar
// uma ou mais bibliotecas, as inclusões
// devem aparecer antes de qualquer declaração
/*
 * *Dentro do programa é permitido declarar*
 * *variáveis globais, constantes globais e*
 * *funções em qualquer ordem.*
 *///Declarações de funções somente
//são permitidas dentro do programa.
funcãoinicio()/*
 * *Declarações de variáveis locais,*
 * *constantes locais, estruturas de*
 * *controle e expressões.*
 */}

```
programa {  
  
    funcao inicio() {  
  
        // instruções do programa  
  
    }  
}
```

Comentários num programa:

// representa um **comentário de linha**; tudo o que estiver após // não é traduzido para código binário nem executado.

```
/*  
    comentário com várias linhas  
*/
```

Tipos de Dados:

inteiro

real

caracter

cadeia

logico

Declaração de Variável:

<tipo_de_dados> <nome_variável>

Exemplo: inteiro idade

Declaração de Constante:

const <tipo_de_dados> <nome_constante> = <valor_constante>

Exemplo: const real PI = 3.1416

```
programa
{
    funcao inicio()
    {
        inteiro x
        real y, z
        caracter c = 'A'
        cadeia s = "Olá Mundo"
        const inteiro N = 10
    }
}
```

Leitura e Escrita:

leia() - lê um ou mais valores

escreva() - escreve um ou mais valores



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

        escreva("Insira um número: ")

        leia( x )

        escreva( "O número lido foi ", x )
    }
}
```

>_ Console

Mensagens

Insira um número inteiro:4

0 número lido foi 4

Programa finalizado. Tempo de execução: 6322 milissegundos

Operadores aritméticos:

Operação	Símbolo	Prioridade
Adição	+	1
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Resto da divisão inteira	%	2

Operadores relacionais:

Operação	Símbolo
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

Operadores lógicos:

Operador	Prioridade
ou	1
e	2
nao	3

Operadores aritméticos:

```
programa
{
    funcao inicio()
    {
        inteiro metade_inteira, resto, valor

        escreva("Digite um valor: ")
        leia(valor)

        metade_inteira = valor / 2 // Calcula a metade inteira do valor
        resto = valor % 3 // Calcula o resto da divisão do valor por 3

        escreva("\nA metade inteira do numero é: ", metade_inteira)
        escreva("\nO resto (mod) da divisão por 3 é: ", resto, "\n")
    }
}
```

\n – força a mudança de linha

Biblioteca Matematica:

```
programa
{
    inclua biblioteca Matematica --> mat // Inclui a biblioteca Matemática

    funcao inicio()
    {
        real valor, potencia, raiz_quadrada

        escreva("Digite um valor: ")
        leia(valor)

        potencia = mat.potencia(valor, 3.0)      // Calcula o valor elevado ao cubo
        raiz_quadrada = mat.raiz (valor, 2.0)      // Calcula a raiz quadrada do valor

        // Exibe os resultados

        escreva("\nO número ao cubo é: ", potencia, "\n")
        escreva("A raiz quadrada do número é: ", raiz_quadrada, "\n")
    }
}
```

Biblioteca Matematica:

```
programa
{
    inclua biblioteca Matematica --> mat // Inclui a biblioteca Matemática

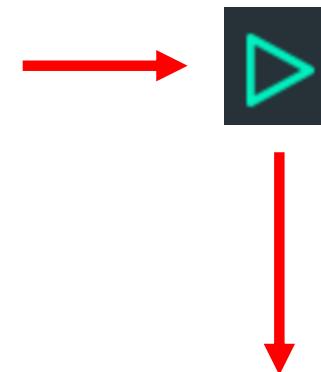
    funcao inicio()
    {
        real valor, potencia, raiz_quadrada

        escreva("Digite um valor: ")
        leia(valor)

        potencia = mat.
        raiz_quadrada =
        // Exibe os res

        escreva("\nO nú
        escreva("A raiz

    }
}
```



```
Digite um valor: 4
O número ao cubo é: 64.0
A raiz quadrada do número é: 2.0
```

```
Programa finalizado.
```



- Expressões: operadores aritméticos, relacionais e lógicos;
- Instruções de Controlo de Fluxo:
 - **Instruções Condicionais** (desvios condicionais) – permitem executar um conjunto de ações (instruções) ou outro conjunto de ações, alternativamente, sob o controlo de uma condição (teste de condição).
 - **Instruções Cíclicas** (ciclos; laços de repetição) – permitem executar repetidamente (em ciclo) um conjunto de ações (instruções) sob o controlo de uma condição (teste de condição).