

### 3 - LINGUAGENS DE PROGRAMAÇÃO

Uma linguagem de programação é um conjunto de termo, expressões e símbolos que permitem aos programadores escreverem instruções que serão executadas por um computador.

Existe uma correlação entre a simplicidade de uma linguagem e a suas potencialidades, sendo que quanto mais simples, (normalmente) menos potente é a linguagem.

Existem dezenas de linguagens de programação, algumas são genéricas e outras de uso mais específico. Pode consultar uma lista das linguagens de programação neste artigo da Wikipédia [https://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_programming_languages).

### **3 - LINGUAGENS DE PROGRAMAÇÃO (cont...)**

As linguagens estão divididas em dois grupos principais:

#### **3.1 - LINGUAGENS DE ALTO NÍVEL**

As linguagens de alto nível são linguagens criadas para serem fáceis de usar e não necessitarem de uma compreensão detalhada sobre a componente de hardware.

Estas linguagens têm uma codificação próxima da linguagem humana e gerem algumas componentes dos sistemas de forma automática e normalmente podem ser executadas em sistemas com hardware distinto.

As linguagens de alto nível são:

- Relativamente simples de escrever
- De escrita relativamente rápida
- Fáceis de entender
- Razoavelmente rápidas na sua execução
- Podem ser utilizadas em hardware distinto
- Não necessitam de conhecimento específico sobre a camada de hardware

### 3 - LINGUAGENS DE PROGRAMAÇÃO (cont...)

Exemplos de Linguagens de alto nível:

- Python
- PHP
- Visual Basic
- Java

Mais informação em [https://en.wikipedia.org/wiki/High-level\\_programming\\_language](https://en.wikipedia.org/wiki/High-level_programming_language)

## 3.2 - LINGUAGENS DE BAIXO NÍVEL

As linguagens de baixo nível são linguagens mais próximas do que se chama “código máquina” e mais próximas do hardware. Este tipo de linguagem pode ser mais complexo e mais específico para cada tipo de hardware, sendo que um programa escrito numa destas linguagens pode não poder ser executado num sistema com hardware distinto.

As linguagens de baixo nível:

- Permitem um controlo mais granular do hardware
- São normalmente de execução mais rápida
- Requerem conhecimentos específicos sobre o hardware
- Normalmente só podem ser executadas no mesmo (ou similar) processador

## 3.2 - LINGUAGENS DE BAIXO NÍVEL (cont...)

Exemplos de linguagens de baixo nível:

- Assembly
- Código Máquina

Mais informação em [https://en.wikipedia.org/wiki/Low-level\\_programming\\_language](https://en.wikipedia.org/wiki/Low-level_programming_language)

### 3.4 - O QUE É O CÓDIGO FONTE?

O código fonte é um conjunto de instruções, normalmente contido num ou mais ficheiros, que contém instruções escritas por um programador. O código fonte geralmente é destinado a ser compilado ou interpretado e contém as instruções necessárias para atingir um determinado fim.

### 3.5 - O QUE SÃO PALAVRAS RESERVADAS?

Numa linguagem de programação, as “palavras reservadas” são todas palavras que têm um significado pré-definido, ou seja, estão reservadas para um uso específico e não podem ser usadas para outro fim.

Alguns exemplos desse tipo de palavras reservadas são: Start, End, Read, Write.

Palavras reservadas existem em todas as linguagens de alto-nível e só podem ser usadas para um fim determinado.

### 3.6 - QUAL A DIFERENÇA ENTRE UM SCRIPT E UM PROGRAMA?

Geralmente associa-se um script a um conjunto de instruções que são interpretadas e um programa a um conjunto de instruções compiladas num binário ou executável.

A principal função de um script é controlar outra aplicação, como por exemplo o PHP pode ser utilizado para controlar a resposta de um servidor Web ou criar scripts de linha de comandos que controla e/ou automatiza funções de um sistema operativo. Outro exemplo é o JavaScript, que é usado habitualmente para controlar um browser.

Por outro lado, um programa escrito numa linguagem como o Java, C++, Visual Basic .NET podem ser executados de forma independente e sempre que o utilizador o deseja.



### 3.7 - COMPILADORES E INTERPRETADORES

Os computadores podem executar programas escritos numa determinada linguagem. Não é possível (ainda) escrever programas numa linguagem corrente como o Inglês ou o Português, porque o seu computador não vai entender nem executar.

Mas afinal qual é a linguagem que o computador compreende? Um computador entende uma linguagem de baixo nível específica chamada “código máquina”. Em código máquina, todas as instruções são constituídas por zeros e uns, o chamado código binário. O Exemplo a seguir é código máquina para a soma de dois números:

```
0010 0001 0000 0100
0001 0001 0000 0101
0011 0001 0000 0110
0111 0000 0000 0001
```

Não se preocupe, não vai necessitar de saber programar desta forma. Felizmente já ninguém programa desta forma. Atualmente todos os programadores escrevem programas numa linguagem chamada de alto-nível e utilizam outros programas específicos para traduzir os seus programas em linguagem máquina.

### 3.7 - COMPILADORES E INTERPRETADORES (cont...)

Há dois tipos de programas que podem ser utilizados neste tipo de tradução:

- **Compiladores**

Um compilador é um programa que traduz instruções escritas numa linguagem de alto nível num programa em código máquina. O programa depois de compilado pode ser executado sempre que o desejar. O compilador é apenas utilizado para a tradução do código fonte para um programa executável (compilado) e não é mais necessário para a correta execução do nosso programa. Os programas compilados são geralmente os programas de mais rápida execução.

- **Interpretadores**

Um interpretador é um programa que traduz e executa em simultâneo as instruções escritas numa linguagem de alto nível. Do ponto de vista do funcionamento de um interpretador, cada linha de código é lida e executada sequencialmente. Este processo é executado para cada linha do código fonte até chegar ao fim do código. As linguagens “interpretadas” são normalmente de execução mais lenta dos que as linguagens compiladas.