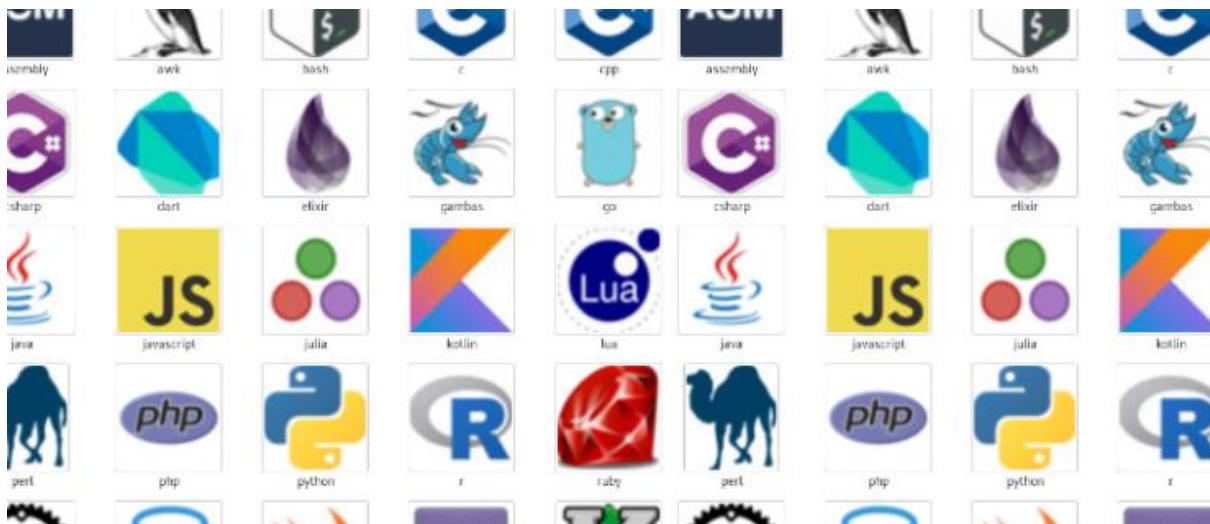


## Linguagens de Programação

Podemos imaginar o computador como uma super calculadora, capaz de fazer cálculos muito mais rápido que nós, mas para isso devemos dizer para o computador o que deve ser calculado e como deve ser calculado. A função das linguagens de programação é exatamente essa, ou seja, servir de um meio de comunicação entre computadores e humanos.



Existem dois tipos de linguagens de programação: as de **baixo nível** e as de **alto nível**. Os computadores interpretam tudo como números em base binária, ou seja, só entendem zero e um. As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido, porém é muito difícil e incômodo se trabalhar com elas. Exemplos de linguagens de baixo nível são a linguagem binária e a linguagem **Assembly**.

### Exemplo de Código em Assembly:

MOV r0, #0C ;load base address of string into r0

LOAD: MOV r1,(r0) ;load contents into r1

CALL PRINT ; call a print routine to print the character in r1

INC r0 ;point to next character

JMP LOAD ;load next character

Como pode-se notar, é uma linguagem bastante complicada.

Já as linguagens de alto nível são mais fáceis de se trabalhar e de entender, as ações são representadas por palavras de ordem (exemplo faça, imprima, etc) geralmente em inglês, foram feitas assim para facilitar a memorização e a lógica. Elas não são interpretadas diretamente pelo computador, sendo necessário traduzi-las para linguagem binária utilizando-se de um programa chamado **compilador**.

Quando programamos em uma linguagem de programação de alto nível primeiramente criamos um arquivo de texto comum contendo a lógica do programa, ou seja, é onde falamos ao computador como deve ser feito o que queremos. Este arquivo de texto é chamado de **código-fonte**, cada palavra de ordem dentro do código-fonte é chamada de instrução. Após criarmos o código-fonte devemos traduzir este arquivo para linguagem binária usando o compilador correspondente com a linguagem na qual estamos programando. O compilador irá gerar um segundo arquivo que chamamos de executável ou programa, este arquivo gerado é interpretado diretamente pelo computador.

Existem algumas linguagens de programação que não necessitam de compiladores, como o PHP, uma linguagem dedicada à produção de websites dinâmicos, como o InfoEscola. As instruções em PHP são compiladas e executadas ao mesmo tempo.

Cada linguagem de programação é diferente da outra, contendo palavras-chave próprias. Exemplos de linguagens de alto nível são C++, Java, C#, Delphi (Pascal), PHP, Visual Basic, etc.

Computadores são estúpidos. Isso pode vir como uma surpresa, considerando que eles são capazes de fazer cálculos complexos em poucos segundos, de forma que nenhum humano conseguiria. Mas é verdade! Computadores são extremamente estúpidos e precisam de instruções exatas para tudo o que fazem. O que é óbvio para você, humano, certamente não é óbvio para uma máquina. E se você quer que a máquina faça algo pra você, bem, você precisa, de certa forma, "falar com ela".

Você já deve ter ouvido falar que computadores entendem apenas binário, ou seja, a linguagem de 0 e 1. Todas as instruções que são dadas para eles são traduzidas em sinais elétricos que significam, basicamente, ligado e desligado. Pense em uma pequena lâmpada. Quando ela está ligada, indica 0. Quando desligada, indica 1. Isso é binário. E há uma infinita quantidade de combinações que você pode fazer com isso.

É claro, nenhum - talvez algum - humano na face da terra iria programar um Skyrim ou mesmo um simples site da Web utilizando apenas 0 e 1. Isso seria extremamente cansativo e demorado. Pra isso que servem as linguagens de programação.

Pense em você e sua máquina. Você fala português, mas a sua máquina fala apenas binário. Entre você e a máquina está um intérprete, o compilador, mas ele só sabe binário e uma outra língua - Java, por exemplo. Aprender a "falar" Java é muito mais fácil que aprender a "falar" binário. Isso porque Java possui muitas palavras em comum com o inglês, e você pode escrever instruções que você entende, como "imprima" ou "leia", ao invés de 0010100010101010. Assim, quando você quer se comunicar com a sua máquina e criar uma série de instruções para ela seguir, você escreve em Java para o compilador e ele traduz as instruções para a máquina, em binário. Isso é uma linguagem de programação.

Linguagens de programação podem ser catalogadas em dois tipos: de **baixo nível** e **alto nível**. Isso não quer dizer que uma é mais rica que a outra ou algo do tipo. Significa que uma está muito mais próxima da máquina do que a outra.

Chamamos de **linguagem de programação** (ou informática) uma linguagem destinada a descrever o conjunto das ações consecutivas que um computador deve executar. Desta forma, uma linguagem informática é uma maneira prática para nós, humanos, darmos instruções a um computador. Por outro lado, o termo 'linguagem natural' representa as possibilidades de expressão compartilhadas por um grupo de indivíduos (por exemplo, o inglês ou o português).

As linguagens que servem para que os computadores se comuniquem entre si não têm nada a ver com linguagens informáticas. Neste caso, falamos de protocolos de comunicação, que são duas noções totalmente diferentes. Em uma linguagem de programação rigorosa, cada instrução gera uma ação do processador.

A linguagem utilizada pelo processador chama-se **linguagem das máquinas**. Trata-se de dados tal como chegam ao processador, constituídos por uma sequência de 0 e 1 (dados binários). Assim, ela não é compreensível por um ser humano e é por isso que foram criadas linguagens intermediárias, compreensíveis por homens. O código escrito neste tipo de linguagem é transformado em linguagem das máquinas para poder ser usada pelo processador.

A **linguagem de montagem** foi a primeira linguagem de programação a ser usada. Ela é parecida com a linguagem das máquinas, mas é mais compreensível pelos programadores. Ela é tão parecida com a linguagem das máquinas que depende do tipo de processador utilizado, já que cada um pode ter a sua própria linguagem das máquinas. Assim, um programa desenvolvido para uma máquina não poderá ser aplicado a outro tipo de máquina. O termo **portabilidade** designa a aptidão que um programa informático tem em ser utilizado em máquinas de tipos diferentes. Às vezes, para poder utilizar um programa informático escrito em linguagem de montagem em outro tipo de máquina, o programa deverá ser reescrito completamente.

Uma linguagem informática tem, então, várias vantagens, ou seja, ela é mais fácil de entender se comparada à linguagem das máquinas e, sobretudo, ela permite uma maior portabilidade, isto é, uma maior facilidade de adaptação nas máquinas de tipos diferentes.

## Quais são os Tipos de Linguagem de Programação

As linguagens de programação estão divididas em duas grandes famílias, dependendo da maneira como as instruções são processadas: as **linguagens imperativas** e as **linguagens funcionais**.

### Como Funciona a Linguagem Imperativa

A **linguagem imperativa** organiza o programa através de uma série de instruções, reunidas por blocos e compreendendo saltos condicionais que permitem retornar a um bloco de instruções se a condição for realizada. Historicamente, são as primeiras linguagens, mesmo se várias linguagens modernas ainda utilizem este princípio de funcionamento. Porém, as linguagens imperativas estruturadas sofrem da falta de flexibilidade, dado o caráter sequencial das instruções.

### Como é Construída a Linguagem Funcional

A **linguagem funcional** tem um nível elevado de abstração, já que elimina muitos detalhes da programação, diminuindo assim a eventualidade de erros. Por quê? Como não depende das operações de atribuição, este tipo de linguagem evita estados ou dados mutáveis.

Ela se baseia na aplicação de funções e, desta forma, qualquer função pode ter ou não regras e um simples valor de retorno. As regras são os valores de entrada da função e o valor de retorno é o resultado desta função. A determinação de uma função vai dizer como ela será avaliada em relação às outras, não exigindo definições adicionais. A linguagem funcional é, pelas suas características, mais simples para tratar das provas e análises matemáticas do que a linguagem imperativa.

### Como é Feita a Interpretação e a Compilação

As linguagens de programação podem ser classificadas em duas categorias: as **linguagens interpretadas** e as **linguagens compiladas**.

#### O Processo da Linguagem Interpretada

Uma linguagem de programação é, por definição, diferente da linguagem das máquinas. Por isso, é necessário traduzi-la para torná-la compreensível para o processador. Um programa escrito numa **linguagem interpretada** precisa de um programa auxiliar (o intérprete) para traduzir progressivamente as instruções recebidas.

#### A Construção da Linguagem Compilada

Um programa escrito numa **linguagem compilada** vai ser traduzido completamente por um programa anexo, chamado compilador, a fim de gerar um novo arquivo autônomo, que não precisará mais de outro programa para ser executado; dizemos, então, que este arquivo é executável.

Um programa escrito em linguagem compilada não precisa, uma vez compilado, de um programa anexo para ser executado. Além disso, como a tradução é feita completamente, a execução é mais rápida. Contudo, ele é menos flexível do que um programa escrito com uma linguagem interpretada, porque a cada modificação do arquivo-fonte (aquele que vai ser compilado) será preciso recopiá-lo para que as alterações tenham efeito.

Por outro lado, um programa compilado tem a vantagem de garantir a segurança do código-fonte. Na verdade, uma linguagem interpretada e compreensível, permite a qualquer programador conhecer os segredos de construção de um programa e, desta forma, copiar ou alterar o código. Por causa disso, existe o risco de desrespeito aos direitos autorais. Certos aplicativos protegidos precisam garantir a confidencialidade do código para evitar o hacking.

#### A Estrutura das Linguagens Intermediárias

Algumas linguagens podem pertencer, de certa forma, às duas categorias (LISP, Java, Python, etc.), já que o programa escrito com estas linguagens pode, em certas condições, sofrer uma fase de compilação intermediária para ser compatível com um arquivo escrito numa linguagem que não é inteligível (logo, diferente do arquivo-fonte) e, por isso, não executável, necessitando de um intérprete. Por

exemplo, os applets Java, pequenos softwares executados na janela de um aplicativo para ampliar as funcionalidades dos navegadores (som, animação, etc.), encontrados nas páginas Web. Eles são arquivos compilados que só podem ser executados a partir de um navegador (arquivos com extensão Class).

### As Linguagens na Prática

Veja, abaixo, a lista das linguagens de programação mais conhecidas:

Linguagem	Domínio de aplicação principal	Compilada/interpretada
ADA	O tempo real	Linguagem compilada
BASIC	Programação básica com objetivos educativos	Linguagem interpretada
C	Programação do sistema	Linguagem compilada
C++	Programação do sistema objeto	Linguagem compilada
Cobol	Gestão	Linguagem compilada
Fortran	Cálculo	Linguagem compilada
Java	Programação orientada para a Internet	Linguagem intermediária
MATLAB	Cálculo matemático	Linguagem interpretada
Matemática	Cálculo matemático	Linguagem interpretada
LISP	Inteligência artificial	Linguagem intermediária
Pascal	Ensino	Linguagem compilada
PHP	Desenvolvimento de sites web dinâmicos	Linguagem interpretada
Prolog	Inteligência artificial	Linguagem interpretada
Perl	Processamento de cadeias de caracteres	Linguagem interpretada