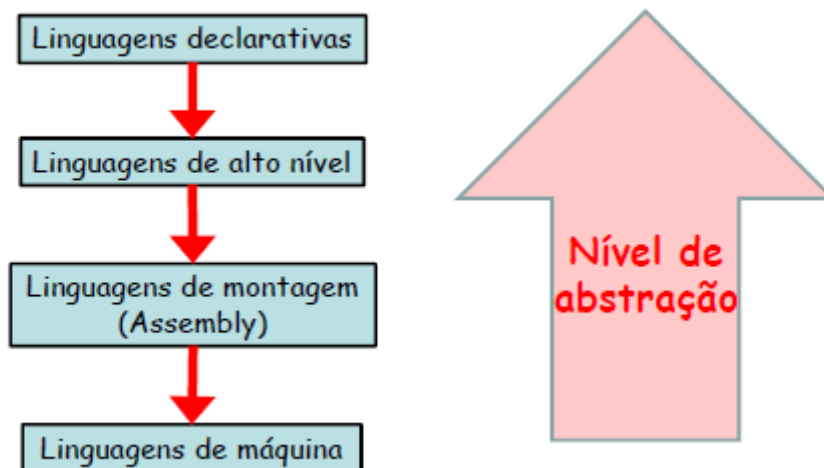


Linguagens de baixo nível x alto nível. O que é isso?

Existem diversas definições a respeito do alto ou baixo nível de uma determinada linguagem, mas desconheço uma que seja universalmente aceita como correta e precisa. Existem algumas opiniões sobre o assunto.

Genericamente dá para dizer que linguagens de alto nível são aquelas que fornecem maior nível de abstração e as de baixo nível são as que fornecem um menor nível, portanto são mais concretas.

Outra forma de explicar é dizer que as linguagens de alto nível são aquelas que estão mais próximas do entendimento humano e as de baixo nível estão mais próximas da operação concreta da máquina. Quanto mais próximo do hardware, onde será implementada e instalada, mais baixo nível a linguagem terá, e necessitará que o programador escreva várias instruções pequenas, que naturalmente serão mais granularizadas, para que obtenha um determinado resultado. De outra forma, quanto mais alto o nível da linguagem, mais próxima e parecida ela tende a ser com a linguagem natural. Linguagens de alto nível permitem que algoritmos sejam expressos em um nível e estilo de escrita fácil para leitura e compreensão por humanos, assim, possuem características de portabilidade, já que podem ser transferidas de uma máquina para outra.



As linguagens de alto nível são mais expressivas, ou seja, o programador expressa melhor qual a intenção do algoritmo. As de baixo nível dizem mais o que fazer sem se preocupar em deixar claro qual é a intenção.

Certamente a falta de sintaxe com melhores construções para expressar de forma intuitiva o que está querendo fazer, de forma estruturada torna uma linguagem de mais baixo nível. Estas linguagens não conseguem definir bem o problema como ele realmente é. Elas se preocupam muito com o mecanismo

para resolver o problema e menos com o conceito dele. Também é certo que há mais controle do que a máquina deve executar quando se usa linguagens de baixo nível.

Então esta classificação se daria pela distância que determinada linguagem tem do trabalho do *hardware*. Em geral as linguagens de alto nível dificultam ou mesmo proíbem o acesso direto ao hardware, que é o oposto do que fazem as linguagens de baixo nível. Mas não é só o acesso que conta, é a forma como este acesso é feito. As linguagens de alto nível não requerem conhecimento profundo sobre o funcionamento do *hardware*.

Alguns exemplos de linguagem com seus níveis:

- **Linguagem de máquina e de assembly:** Linguagens que dependem da arquitetura da máquina (Linguagem de montagem é uma representação simbólica da linguagem de máquina associada).
- **Linguagens de Médio nível:** Apesar de muitos não aceitarem este termo, C pode ser classificada como uma linguagem de médio nível, porque ela pode, por exemplo, acessar os registros do sistema ou trabalhar com endereços de memória, que são características de linguagens de baixo nível, e ao mesmo realizar operações de alto nível.
- **Linguagens de Alto nível:** Java, PHP, C# e muitas outras.
- **Linguagens declarativas (Linguagens expressivas como a linguagem oral expressam o que fazer ao invés de como fazer, pode conter:** programação funcional, programação lógica ou programação restritiva): Haskell (funcional) e Prolog(lógica).

Um adendo: As linguagens podem ser divididas em gerações também:

- Primeira geração: Linguagem máquina.
- Segunda geração: Criaram-se as primeiras linguagens assembly.
- Terceira geração: Criam-se as primeiras linguagens de alto nível. Ex: C, Pascal, Cobol...
- Quarta geração: São linguagens capazes de gerar código por si só, são os chamados RAD, com o qual pode-se realizar aplicações sem ser um expert na linguagem. Aqui também se encontram as linguagens orientadas a objetos, tornando possível a reutilização de partes do código para outros programas. Ex: Visual, Natural Adabas...
- Quinta geração: Aqui se encontram as linguagens orientadas à inteligência artificial. Estas linguagens ainda estão pouco desenvolvidas. Ex: LISP