

- **LÓGICO:** Este tipo de dados pode assumir apenas dois valores VERDADEIRO ou FALSO. Também é conhecido como **booleano**. Reserva apenas um bit na memória, onde o valor 1 representa VERDADEIRO e o valor 0 representa FALSO. Por exemplo uma variável *status_da_lampada* (acesa ou apagada).
- **TEXTO:** Tipo de dados para variáveis que armazenam textos. Por exemplo uma variável *nome*.

Algumas linguagens de programação dividem esses tipos primitivos de acordo com o espaço necessário para os valores daquela variável. Na linguagem Java por exemplo, o tipo de dados inteiro é dividido em 4 tipos primitivos: *byte*, *short*, *int* e *long*. A capacidade de armazenamento de cada um deles é diferente.

- **byte:** é capaz de armazenar valores entre -128 até 127.
- **short:** é capaz de armazenar valores entre - 32768 até 32767.
- **int:** é capaz de armazenar valores entre -2147483648 até 2147483647.
- **long:** é capaz de armazenar valores entre -9223372036854775808 até 9223372036854775807.

Mas essa divisão é uma particularidade da linguagem de programação. O objetivo é otimizar a utilização da memória. Em algumas linguagens de programação não é necessário especificar o tipo de dados da variável, eles são identificados dinamicamente. Porém, é necessário informar o tipo de dados de cada variável em algoritmos.

Tipos de dados customizados

A partir dos tipos de dados primitivos podemos criar outros tipos de dados utilizando uma combinação de variáveis. São estruturas de dados, classes, vetores, matrizes, etc.

Por exemplo, uma classe chamada *Carro* é um tipo de dados que agrupa outras variáveis básicas como **marca**, **cor**, **ano**, **modelo**, etc.

Um *vetor* é um agrupamento de variáveis do mesmo tipo, uma *matriz* é um agrupamento de vetores. Enfim, a base de todos os tipos de dados são os tipos de dados primitivos, independente da linguagem de programação.

```
class Carro {  
  
    String tipo;  
    String cor;  
    String placa;  
    int numPortas;  
  
    void setCor(String c){  
        cor = c;  
    }  
  
    String getCor(){  
        return cor;  
    }  
}
```