

3. Linguagem de programação

3.1 – Funções Base

Vamos chamar de programa uma sequência de ações programadas passadas previamente à placa Arduino. Toda vez que ela for ligada ela seguirá aquela programação de forma sequencial.

Um programa possui um corpo, uma **função principal**, onde ele será rodado. Em Arduino, tomamos como funções base as funções:

```
void setup() {
```

```
}
```

```
void loop() {
```

```
}
```

A função `void setup()` é a função que inicializa. Ela é rodada apenas uma vez ao alimentar a placa Arduino e nela contém as informações necessárias para a programação feita na função `void loop()`, que por sua vez roda até que a alimentação seja cortada. Dentro da `void loop()` contém as ações principais do programa.

Uma das coisas que devemos informar na função é a **configuração dos pinos** que vamos usar como **pinos de entrada** (INPUT) ou **pinos de saída** (OUTPUT). Fazemos isso da seguinte maneira:

```
pinMode(pino, MODO);
```

Onde *pino* é o número indicativo da porta que está sendo configurada e *MODO* é a configuração que ela está recebendo (entrada ou saída).

Observação:

Repare que cada placa possui a indicação dos pinos. Os pinos de entrada analógica são indicados por A0, A1, A2, A3, A4 e A5, no caso da placa Arduino UNO, Duemilanove e semelhantes. Essas portas não precisam ser identificadas na função de `setup` se estiverem sendo usadas para **leitura de dados analógicos**. Porém, essas portas também podem ser usadas como pinos digitais, e nesse caso eles devem ser identificados e a numeração deles é dada de **14** (no caso do pino A0) até **19** (no caso do pino A5).

Para tratar os dados de forma adequada devemos usar alguns comandos específicos. A tabela abaixo mostra comandos de portas digitais e suas funções:

Comando	Função
<code>digitalRead(pino*);</code>	Esse comando é responsável pela leitura de dados digitais. Ele tem o valor de HIGH ou LOW dependendo do estado do dado de entrada.
<code>digitalWrite(pino*, VALOR);</code>	Esse comando define o estado daquela porta previamente configurada na função <code>void setup()</code> em alta (HIGH) ou baixa (LOW). Em que HIGH significa ligada e LOW significa desligada.