

---

```
const int lm35 = A0;
```

Essa linha define que o **lm35** (vulgo sensor de temperatura), terá a saída no **A0**, além disso define que **lm35 = A0 é um const** (ou seja, uma constante logo essa variável não se altera) e é um **int** (int define que o dado recebido será um valor numérico).

```
float temperatura;
```

Essa linha define que 'Temperatura' é um ponto flutuante, ou seja, não é um valor fixo, além disso utilizando **float** pode-se usar números decimais (números que possuem virgula Ex: 10,21).

```
void setup() {  
    Serial.begin(9600);  
}
```

**Void setup** é basicamente uma função que será executada **UMA VEZ**, e **Serial.begin(9600)** está definindo a velocidade de parâmetro.

```
void loop() {  
    int leitura = analogRead(lm35);  
    temperatura = (5.0 / 1023) * leitura * 100;  
  
    Serial.print("Temperatura: ");  
    Serial.println(temperatura);  
    delay(2000);  
}
```

Primeiro vou falar da função em geral **void loop** que após o final do comando nesse caso **delay(2000)** essa função será executada **INFINITAS VEZES**, ou seja essa função vai ficar se repetindo.

```
int leitura = analogRead(lm35) ;
```

Essa linha define que **leitura** irá fazer uma leitura analógica (**analogRead**), ou seja, irá captar os dados como uma leitura analógica, e (**lm35**) é o sensor de temperatura que irá realizar essa leitura, além disso **int** define que o resultado dessa leitura será um número.

```
temperatura = (5.0 /1023) * leitura * 100;
```

Essa linha está definindo que **temperatura** é uma variável.

Está definindo que (**5.0/1023**) esses parênteses define o desempenho do sensor.

**Leitura** é a variável que foi definida na linha anterior.

**100** ele está multiplicando o resultado que vem em milivolts e transformando em Célsius.

```
Serial.print("Temperatura: ");
```

Essa linha define que será impresso no terminal "**Temperatura:**" como está entre aspas é igual em algoritmo ele está imprimindo um texto.

```
Serial.println(temperatura);
```

A diferença é que **println** define que a próxima vez que esse código for executado no terminal ele aparecerá em uma nova linha, ou seja esse **println** é como usar <br> em HTML o código seguinte aparecer na próxima linha.

```
delay(2000);
```

Está definindo que é para esperar 2000 milissegundos, ou seja, 2 segundos até que a função **void loop** seja executada novamente.

---

```
const int lm35 = A0;
float temperatura;
void setup() {
    Serial.begin(9600);
}
void loop() {
    int leitura = analogRead(lm35);
    temperatura = (5.0 / 1023) * leitura * 100;

    Serial.print("Temperatura: ");
    Serial.println(temperatura);
    delay(2000);
}
```