

PROJETO ARDROBOTICA


PROGRAMAÇÃO COM ARDUINOS



Atividade 5 – Led ativado por um botão de pressão (Push-Button)

Material necessário:

Botão  (Input Digital)

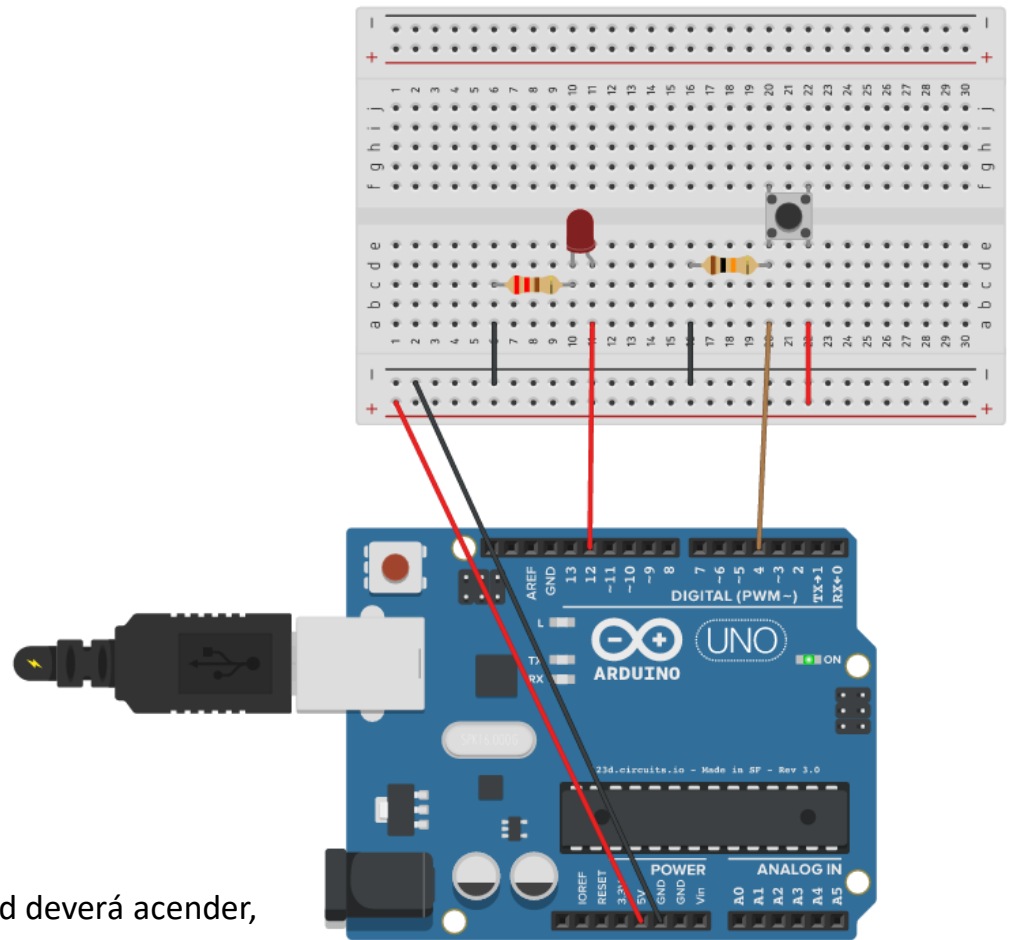
Led  (Output Digital)

2 Resistências  220Ω
 10kΩ

Fios 

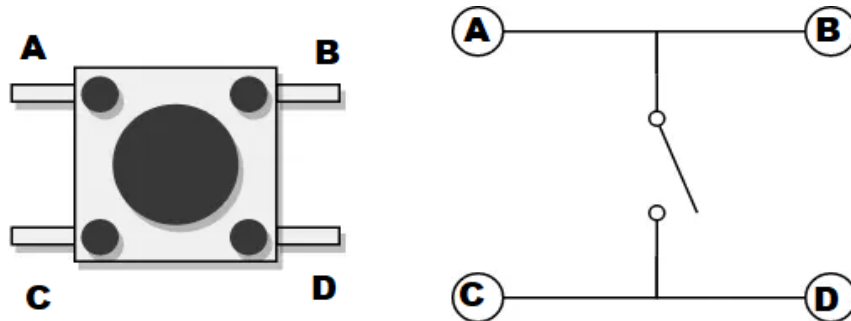
Nota:

Quando o botão de pressão for pressionado o led deverá acender, caso contrário deverá permanecer apagado.



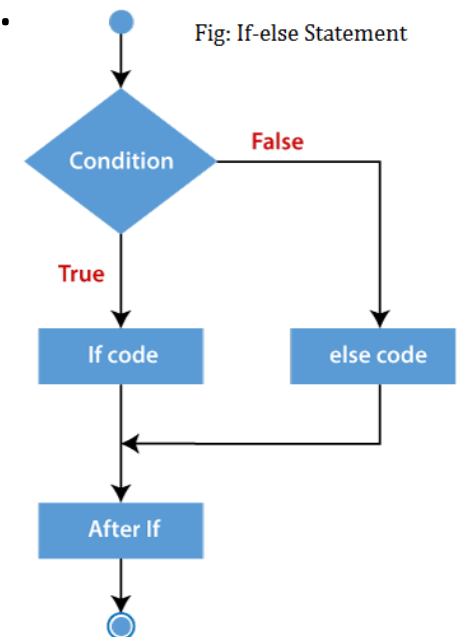
Atividade 5 – Led ativado por um botão de pressão (Push-Button)

- O push button (ou botão de pressão) é um componente que conecta dois pontos de um circuito quando é pressionado.
- Tem o mesmo funcionamento elétrico que um interruptor, fechando ou abrindo o circuito elétrico.
- Embora tenha quatro terminais, apenas tem apenas duas conexões AB e CD.
- Quando o botão é pressionado, passa a existir uma ligação entre AB e CD.



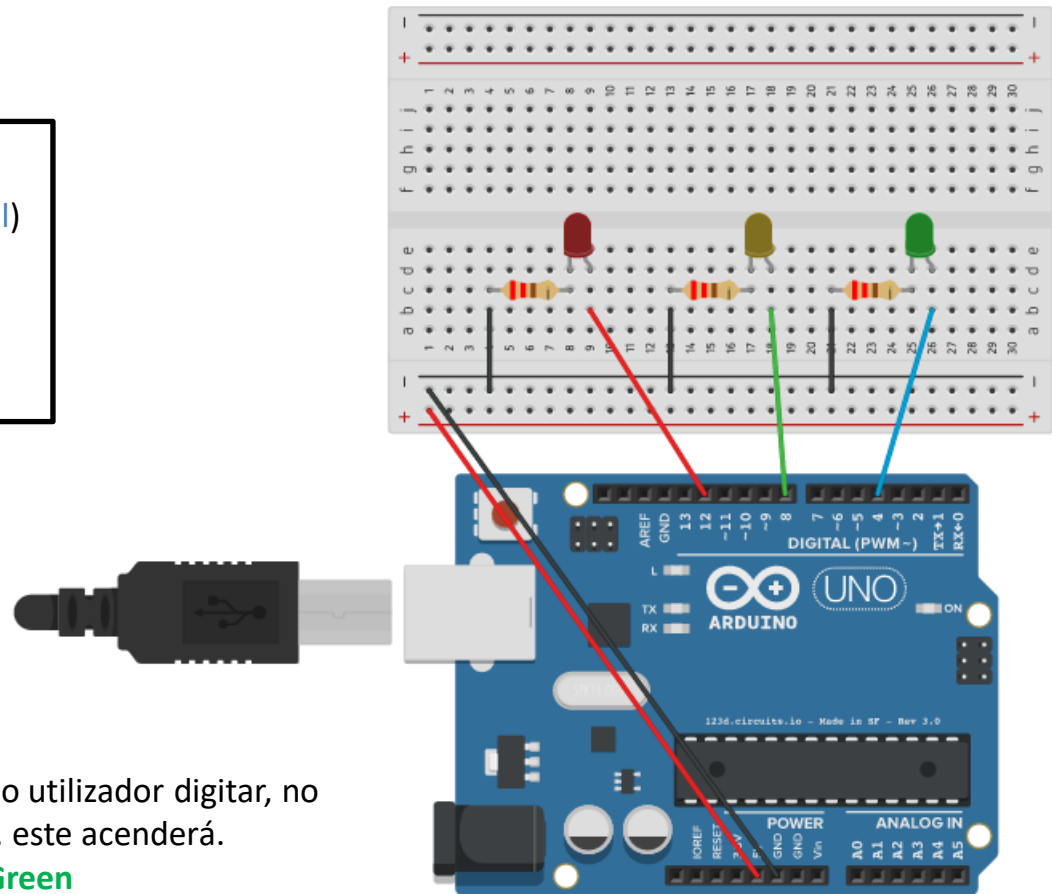
Atividade 5 – Led ativado por um botão de pressão (Push-Button)

- Para se poder determinar se o botão de pressão foi ou não pressionado, vamos recorrer a uma estrutura de decisão – o **if else**.
- A instrução “**if ... else ...**” é uma estrutura de controlo que permite determinar se uma condição é ou não verdadeira e consoante o resultado, executar o respetivo código associado.
- A palavra **if** assinala o início da estrutura de decisão; a **condição** é uma expressão do tipo lógico ou booleano, portanto devolve um valor de verdadeiro ou falso;
- O **else** é uma parte da estrutura de decisão que é opcional. O respetivo código associado ao else apenas é executado se a condição for falsa.



Atividade 6 – Leds controlados por uma letra

(R – Red, Y – Yellow, G – Green)



Nota:

Para cada led é atribuída uma letra. Quando o utilizador digitar, no monitor série, a letra correspondente ao led, este acenderá.

R – Red Y – Yellow G – Green

Se o utilizador digitar a letra A, todos os leds se apagam.

Se o utilizador digitar a letra L, todos os leds se ligam.

Atividade 6 – Leds controlados por uma letra

(R – Red, Y – Yellow, G – Green)

- Para se poder definir as ações a realizar, consoante a letra digitada no monitor série, podemos utilizar a estrutura de decisão **switch**.

Exemplo:

```
switch (operador) {  
    case '+': resultado = a + b;  
        break;  
    case '-': resultado = a - b;  
        break;  
    case '*': resultado = a * b;  
        break;  
    case '/': resultado = a / b;  
        break;  
    default : println("Operação não permitida!");  
}
```

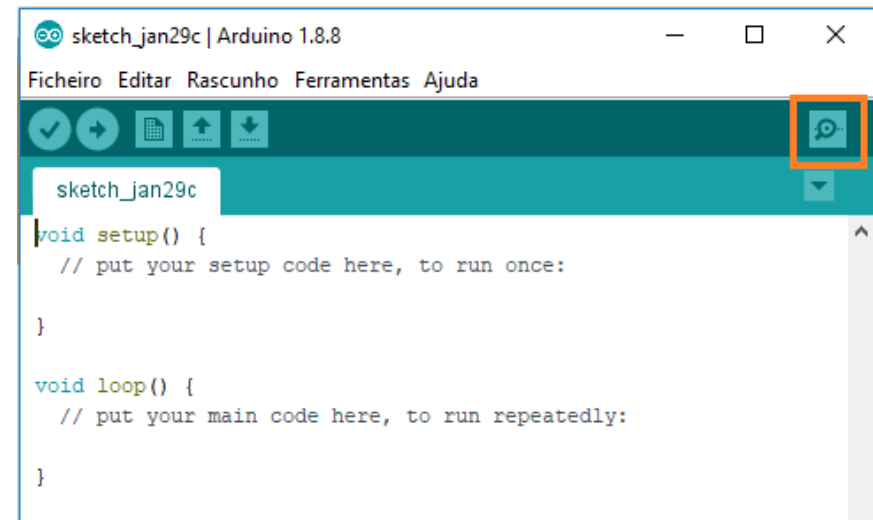

Atividade 6 – Leds controlados por uma letra

(R – Red, Y – Yellow, G – Green)

- Mediante o carácter digitado pelo utilizador, o programa irá realizar a operação pretendida. Para cada caso previsto, é indicada a sequência de instruções a realizar.
- A palavra break termina a sequência de instruções para cada opção.
- A opção default é realizada sempre que nenhuma das condições anteriores tenha sido satisfeita.

Monitor Série

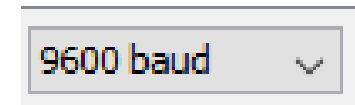
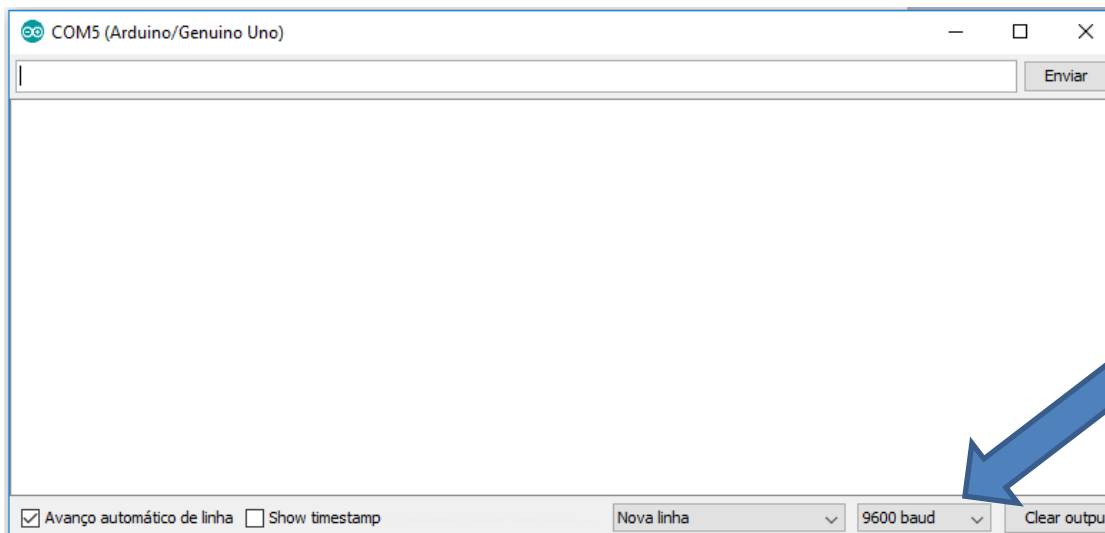
- O Monitor Série é uma ferramenta muito útil, especialmente para efetuar debug (depuração) do código.
- Apresenta os dados transmitidos pelo Arduino através da porta USB.
- Permite:
 - a comunicação entre o Arduino e a porta série.
 - visualizar valores lidos pelos sensores.
 - mostrar textos
 - enviar dados para o Arduino (como se fossem ordens).



Monitor Série

Para trabalhar com o monitor série é necessário:

- Inicializar o monitor série na função setup através da instrução **Serial.begin(x)**
- em que x é a taxa de transferência em bits por segundo:
300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, ou 115200.



Monitor Série

- Inicializar o monitor Série
- Mostrar mensagens

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

```
int i=0;  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.print("Hello World");  
    Serial.print("\t");  
    Serial.print(i);  
    Serial.println("seg");  
    delay(1000);  
    i++;  
}
```

Monitor Série

- `Serial.print("Hello World")`
Imprime a string "Hello World"
- `Serial.print("\t")`
Imprime uma tabulação
- `Serial.print(i)`
Imprime o valor da variável i
- `Serial.println("seg")`
Imprime a string "seg"
Faz mudança de linha

```
int i=0;
void setup() {
  Serial.begin(9600);
}

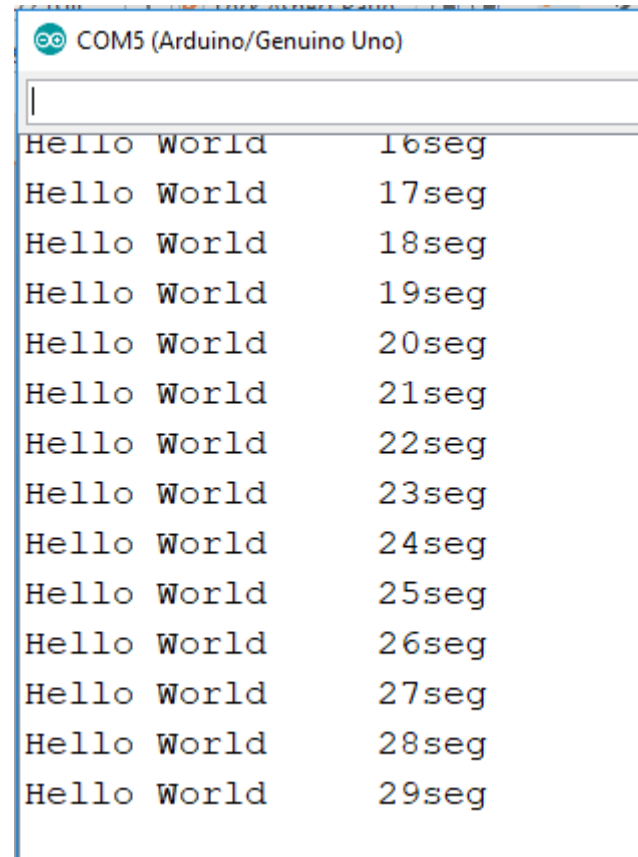
void loop() {
  Serial.print("Hello World");
  Serial.print("\t");
  Serial.print(i);
  Serial.println("seg");
  delay(1000);
  i++;
}
```

Monitor Série

Resultado:

```
int i=0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("Hello World");
  Serial.print("\t");
  Serial.print(i);
  Serial.println("seg");
  delay(1000);
  i++;
}
```



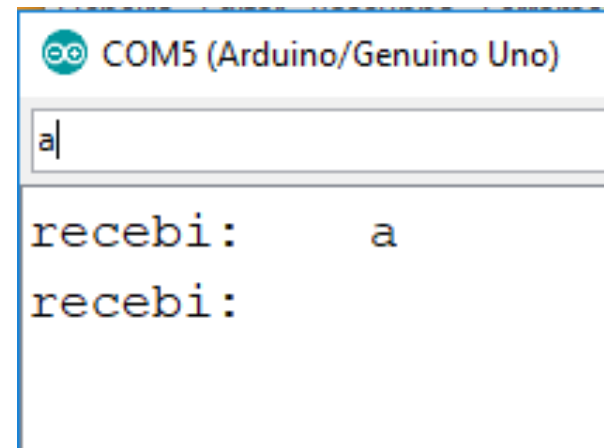
The screenshot shows the Serial Monitor window for COM5 (Arduino/Genuino Uno). The output displays a series of "Hello World" messages, each followed by a tab character and a number representing the iteration count, and then "seg" for seconds. The iteration count increases from 16 to 29 over the course of the output shown.

Hello World	16seg
Hello World	17seg
Hello World	18seg
Hello World	19seg
Hello World	20seg
Hello World	21seg
Hello World	22seg
Hello World	23seg
Hello World	24seg
Hello World	25seg
Hello World	26seg
Hello World	27seg
Hello World	28seg
Hello World	29seg

Monitor Série

`Serial.read()` - permite receber dados através da porta série.


```
char variavel;  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    if (Serial.available() > 0) {  
        variavel = Serial.read();  
        Serial.print("recebi: ");  
        Serial.println(variavel);  
    }  
}
```



Atividade 7 – Led com Fade

Material necessário:

Potenciômetro  (Input Analógico)

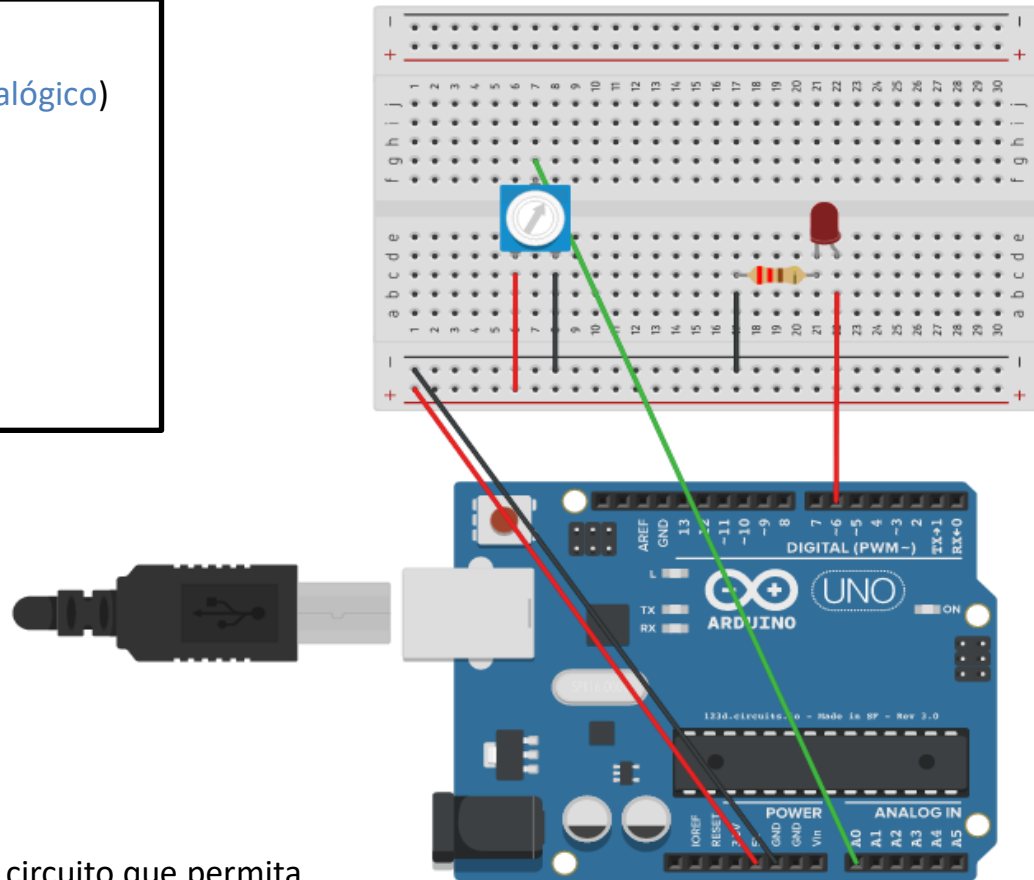
Led  (Output Digital)

Resistência  220Ω

Fios 

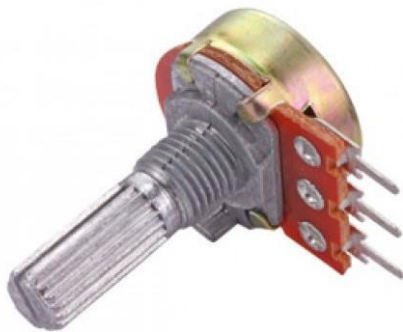
Nota:

Nesta atividade pretende-se construir um circuito que permita controlar a luminosidade de um led através de um potenciômetro.



Potenciômetro

- É um componente eletrônico que possui resistência elétrica ajustável.
- Permite obter uma tensão de saída variável no pino central, a partir de um diferencial de tensão definido entre as duas extremidades do mesmo.



Potenciómetro

- O potenciómetro não tem polo positivo ou negativo, sendo que um terminal externo liga ao positivo e o outro ao negativo.
- O terminal do meio liga a uma porta analógica.
- Os valores obtidos pela leitura do potenciómetro variam entre 0 e 1023.



Portas digitais PWM

- Algumas saídas digitais têm também um tipo de saída específico, uma saída digital PWM.
- A sigla PWM provém do inglês “Pulse Width Modulation” e representa uma técnica que permite por via de um sinal digital, emular um resultado analógico.
- Neste tipo de saída podemos controlar a tensão que sai das portas, ou seja, podemos por exemplo, controlar a velocidade de um motor, ou a luminosidade de um LED.

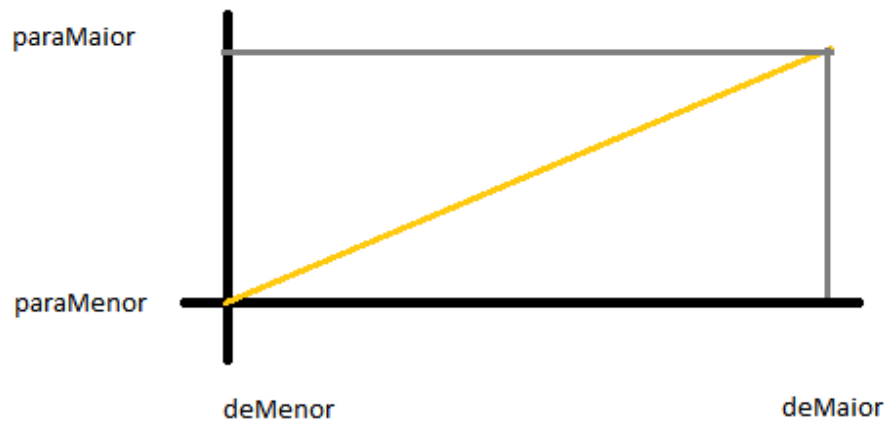
Portas digitais PWM

- O Arduino só tem pinos de Input Analógico (A0...A5)
- O output analógico só pode ser conseguido com PWM, através dos pinos digitais sinalizados com ~.
- O Arduino Uno disponibiliza os pinos 3, 5, 6, 9, 10 e 11 com função PWM.
- Nestes pinos além da função digitalWrite() que permite escrever os valores 0 ou 1, está também disponível a função analogWrite() que permite definir valores numa escala de 0 a 255.

Função map()

- Transforma um valor lido entre um intervalo de valores para outro intervalo de valores

map (valor, deMenor, deMaior, paraMenor, paraMaior)



Sessão 3



The End