

Guia Rápido do LabVad



Laboratório Virtual de Atividades Didáticas
com Robótica - LabVad

Login

Email:

Senha:

ACESSAR [Recuperar Senha](#)

Introdução

Bem vindo ao o Laboratório de Atividades Didáticas para Robótica Educativa. O LabVad é de acesso livre, multiplataforma e não necessita de extensões ou plugins para ser executado no seu computador. Agora é necessário fazer um agendamento para executar seus experimentos.

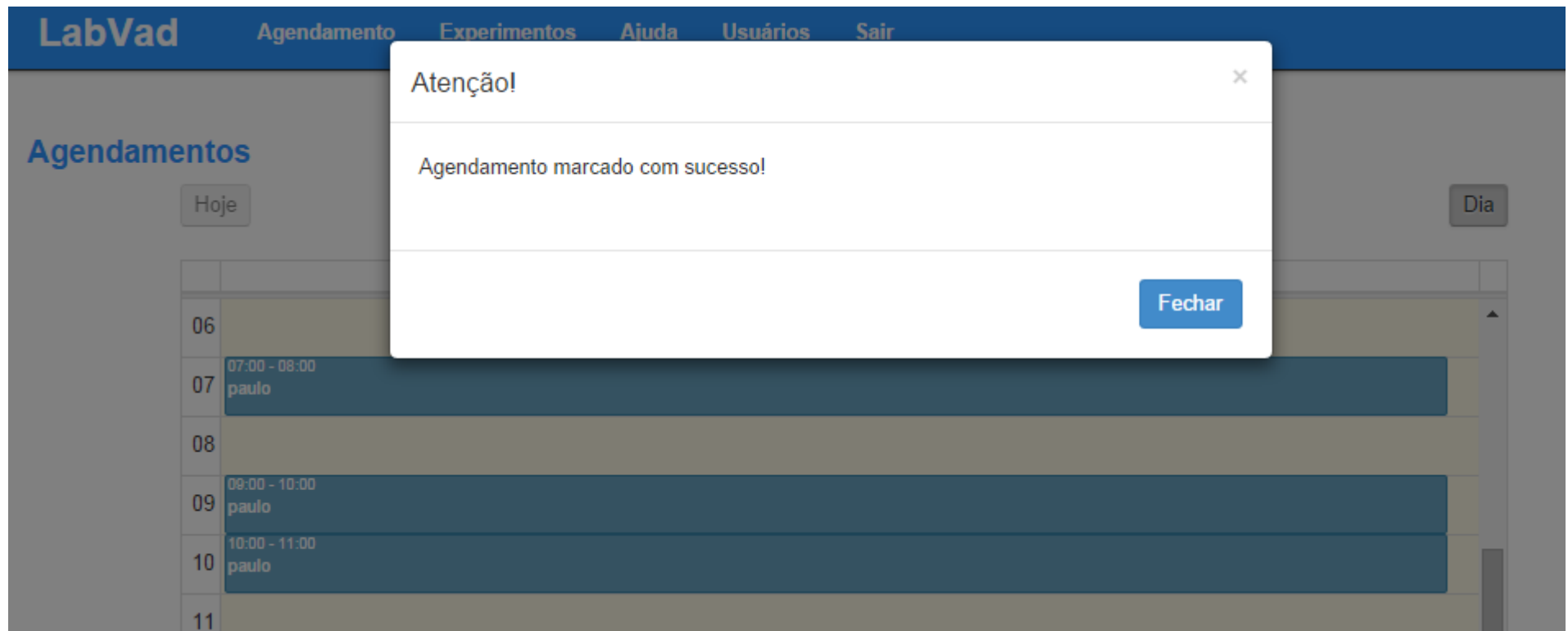
Como agendar um horário?

Basta clicar em Agendamento no menu superior você visualizará um calendário.

Agendamentos

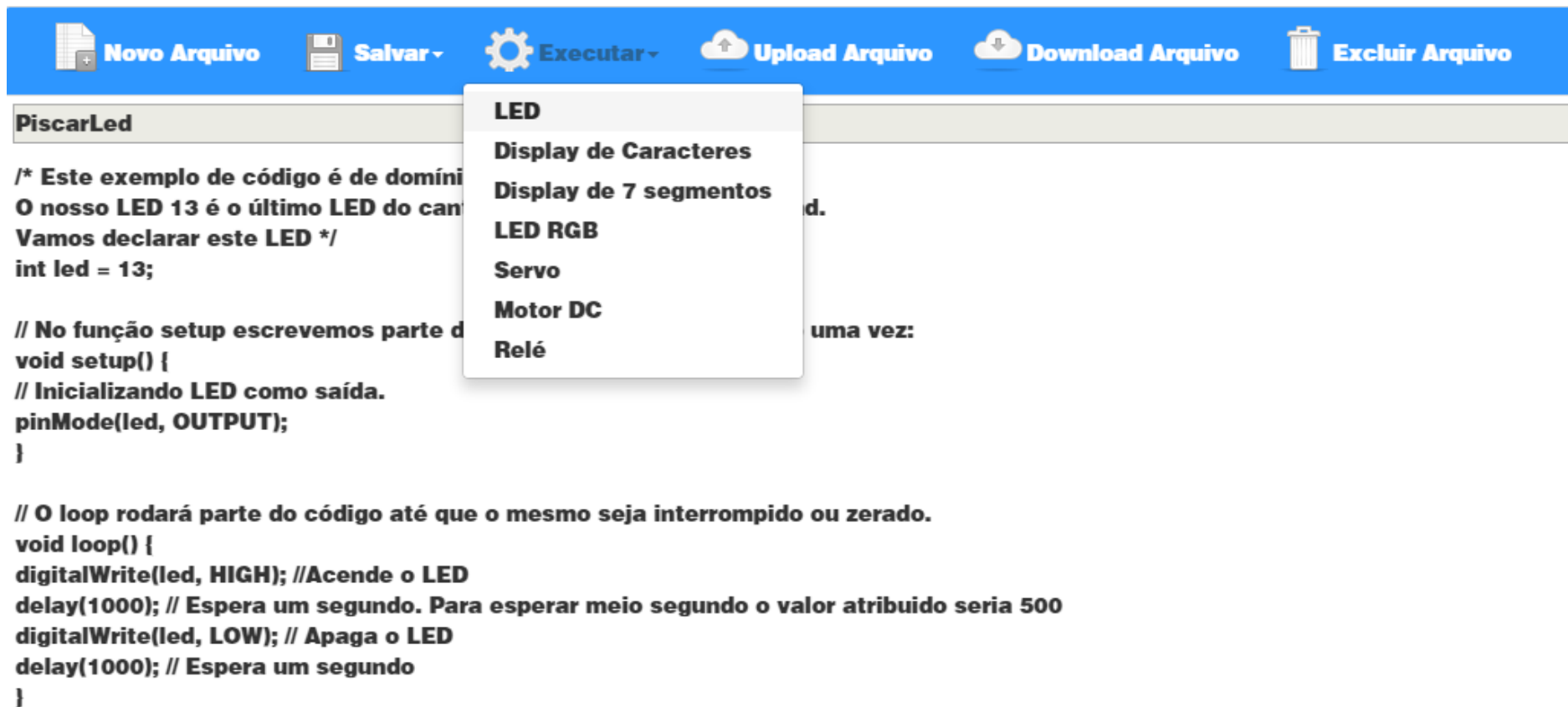
< >		Hoje		setembro 2014		Mês	
dom	seg	ter	qua	qui	sex	sáb	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	1	2	3	4	

Para efetuar o agendamento basta clicar no dia desejado, como na figura abaixo:



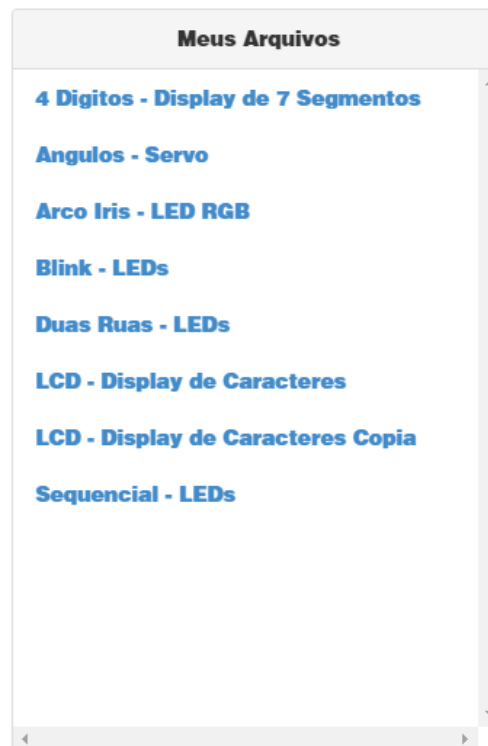
Experimentos

O LabVad possui várias funcionalidades, dispostas de forma bem clara para o usuário. Na função de Executar Código o usuário deve ter atenção para direcionar seu experimento ao dispositivo correto!



Além de conseguir visualizar seus experimentos em tempo real, o LabVad possui um processo muito claro de compilação, retornado para o usuário se o código foi compilado com sucesso ou não. As funções do menu Executar só ficam ativadas para o usuário agendado naquele momento, mas vale ressaltar que o ambiente é multiusuário, ou seja, você pode

acompanhar os experimentos em execução do professor e de outros colegas, bem como: criar, editar, salvar, deletar, subir (upload) ou baixar (download) dos programas na pasta meus códigos.



Lições

Preparamos algumas lições explicando o que é e o que faz cada dispositivo do LabVad. Além disso temos exemplos de códigos para que você construa sua própria biblioteca de códigos no LabVad - Clique em Ajuda.

LED

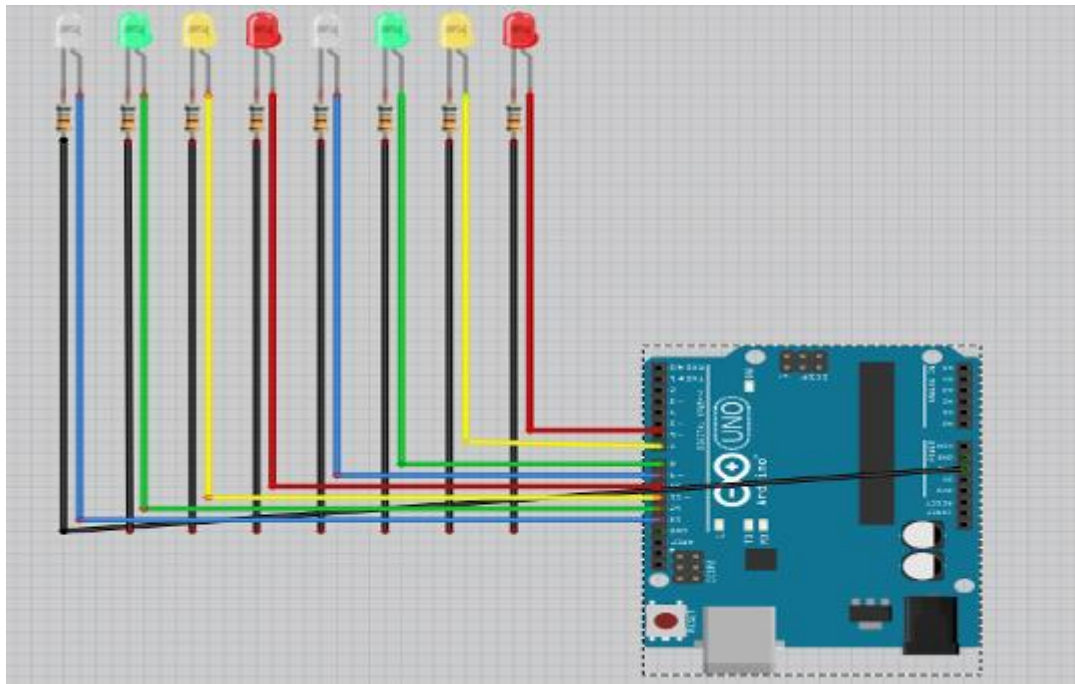


O que é? O que faz?

LED (Light Emitting Diode). Emite uma luz quando uma pequena corrente o aciona.

Como funciona no LabVad?

Os LED estão conectados no LabVad dos pinos 6 ao 13, como na figura abaixo:



Portanto para utilizá-los devemos fazer a declaração, conforme o exemplo abaixo:

```
int ledVermelho1 = 6;
```

```
int ledAmarelo1 = 7;
```

```
int ledVerde1 = 8;
```

```
int ledBranco1 = 9
```



```
int ledVermelho2 = 10;
```

```
int ledAmarelo2 = 11;
```

```
int ledVerde2 = 12;
```

```
int ledBranco2 = 13;
```

Veja um exemplo de código de domínio público:

```
/*  
Picar LED  
Acende o LED por um segundo e depois o apaga por um segundo também.
```

```
  
Este exemplo de código é de domínio publico  
*/
```

```
// O nosso LED 13 é o último LED do canto esquerdo da tela do LabVad.  
// Vamos declarar este LED:  
int led = 13;
```

```
// Na função setup escrevemos parte do código que será executado uma vez:  
void setup() {  
  // Inicializando LED como saída.  
  pinMode(led, OUTPUT);  
}
```

```
// O loop rodará parte do código até que o mesmo seja interrompido ou zerado.  
void loop() {  
  digitalWrite(led, HIGH); //Acende o LED  
  delay(1000); // Espera um segundo. Para esperar meio segundo o valor atribuido seria 500  
  digitalWrite(led, LOW); // Apaga o LED  
  delay(1000); // Espera um segundo  
}
```



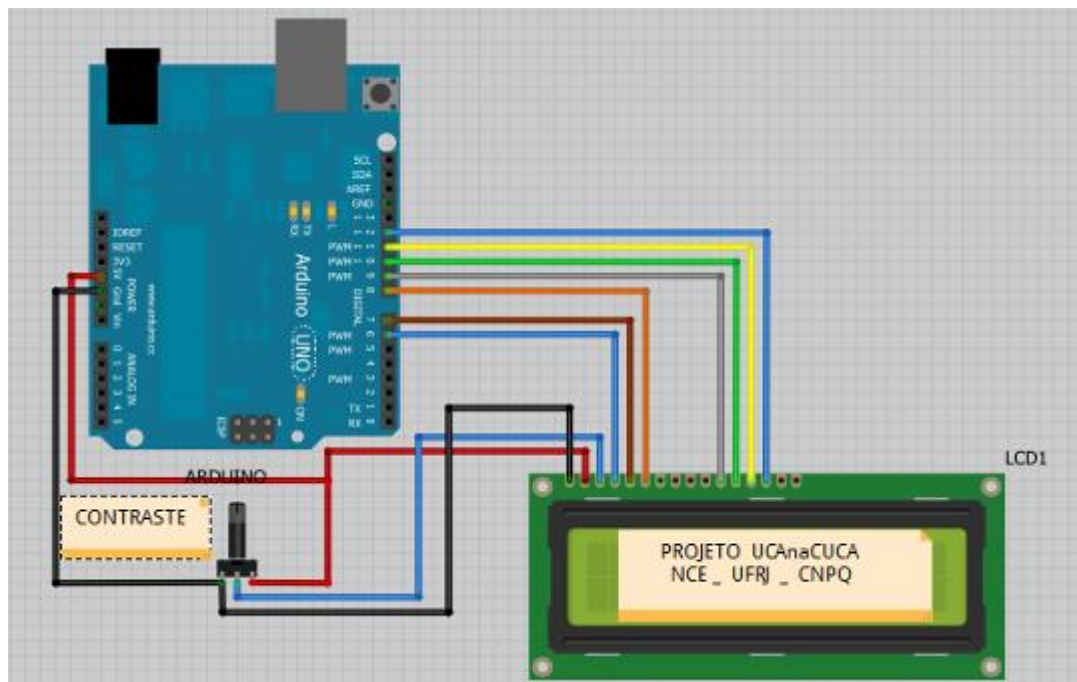
Display de Caracteres

O que é? O que faz?

Um **display de cristal líquido**, acrônimo de **LCD** (em inglês *liquid crystal display*). Mostra dados lidos pelo LabVad em letras e números, de uma forma inteligível.

Como funciona no LabVad?

Os display está conectado ao LabVad nos pinos 6, 7, 8, 9, 10, 11, 12. Veja na imagem abaixo:



Vale lembrar que chamamos o este display de LCD de 16x2, pois ele tem 16 colunas de dígitos e duas linhas, ou seja, você tem duas linhas para escrever até 16 caracteres. Isto vai ficar mais óbvio quando veremos o dispositivo funcionando no LabVad.

Use nosso exemplo para imprimir na tela o que quiser!

```
//*****
// GERADOR de MENSAGENS com Display 16x2
// By S.Brandao
```

```
// 25/07/2014 - 3792 bytes
// Esta experiencia usa o DISPLAY de CARACTERES
//*****
```

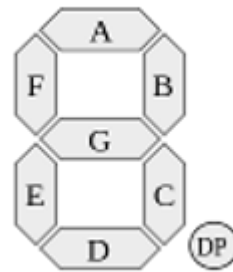
```
#include <LiquidCrystal.h>
```

```
// LiquidCrystal display with:
// rs on pin 6
// rw on pin 7
// enable on pin 8
// d4, d5, d6, d7 on pins 9, 10, 11, 12
LiquidCrystal lcd(6, 7, 8, 9, 10, 11, 12);
int brightness = 255;
```

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

```
void loop()
{
  // Print a message to the LCD.
  lcd.clear();
  delay(500);
  lcd.begin (16, 2); // Define o tamanho do DISPLAY
  lcd.print("* NCE * LABVAD *");
  delay(1500);
  lcd.setCursor(0,1); // Escreve na Linha 1
  lcd.print("Proj: UCAnaCUCA");
}
```

```
brightness = 100;
delay(1500);
lcd.setCursor(0,1); // Escreve na linha 1
lcd.print("* AGOSTO__2014 *");
delay(1500);
lcd.setCursor(0,0); // Escreve na linha 0
lcd.print("* VISITANTES *");
delay(1500);
lcd.setCursor(0,1); // Escreve na linha 1
lcd.print("Sejam Bem-vindos");
delay(1500);
}
```

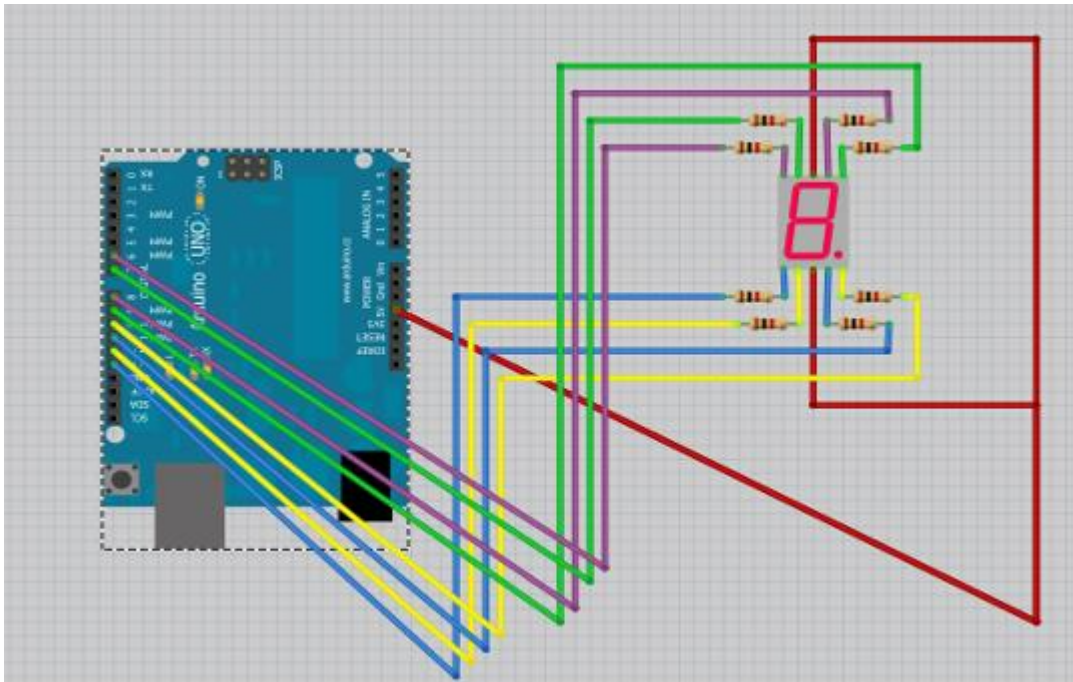


Display de 7 Segmentos

O que é? O que faz?

É um **circuito encapsulado LEDs interligados**. Este tipo de display é amplamente utilizado eletrodomésticos, relógios e equipamentos industriais. Trata-se de 8 LEDs ligados com um ponto em comum. Para entender melhor o funcionamento veja a imagem autoexplicativa acima.

Como funciona no LabVad?



Veja um exemplo onde fica bem claro como você irá programá-lo.

```
/*  
* // DISPLAY 7 SEGMENTOS *  
*/
```

```

*
* // By S.Brandao * *
* // 25/07/2014 - 1670 bytes ***G***
* // Experiencia com o Display de 7 Segmentos * *
*
*
*****
*/

```

```

int timer = 2000; // Tempo = 2 segundos.

```

```

int Seg_F = 7; // Segmento F.
int Seg_G = 6; // Segmento G.
int Seg_E = 13; // Segmento E.
int Seg_D = 12; // Segmento D.
int Seg_A = 8; // Segmento A.
int Seg_B = 9; // Segmento B.
int Seg_C = 11; // Segmento C.
int PD = 10; // Ponto Decimal.

```

```

void setup()
{
  pinMode(Seg_F,OUTPUT); // Segmento F.
  pinMode(Seg_G,OUTPUT); // Segmento G.
  pinMode(Seg_E,OUTPUT); // Segmento E.
  pinMode(Seg_D,OUTPUT); // Segmento D.
  pinMode(Seg_A,OUTPUT); // Segmento A.
  pinMode(Seg_B,OUTPUT); // Segmento B.

```

```

pinMode(Seg_C,OUTPUT); // Segmento C.
pinMode(PD,OUTPUT); // Ponto Decimal.

}
void loop()

{
// DIGITO 1
digitalWrite(Seg_F,LOW); // Segmento F OFF
digitalWrite(Seg_G,LOW); // Segmento G OFF
digitalWrite(Seg_E,LOW); // Segmento E OFF
digitalWrite(Seg_D,LOW); // Segmento D OFF
digitalWrite(Seg_A,LOW); // Segmento A OFF
digitalWrite(Seg_B,HIGH); // Segmento B ON
digitalWrite(Seg_C,HIGH); // Segmento C ON
digitalWrite(PD,HIGH); //PONTO DEC. ON
delay(timer);

// -----

// DIGITO 2
digitalWrite(Seg_F,LOW); // Segmento F OFF
digitalWrite(Seg_G,HIGH); // Segmento G ON
digitalWrite(Seg_E,HIGH); // Segmento E ON
digitalWrite(Seg_D,HIGH); // Segmento D ON
digitalWrite(Seg_A,HIGH); // Segmento A ON
digitalWrite(Seg_B,HIGH); // Segmento B ON
digitalWrite(Seg_C,LOW); // Segmento C OFF

```



```
digitalWrite(PD,LOW); // PONTO DEC. OFF  
delay(timer);
```

```
// DIGITO 3
```

```
digitalWrite(Seg_F,LOW); // Segmento F OFF  
digitalWrite(Seg_G,HIGH); // Segmento G ON  
digitalWrite(Seg_E,LOW); // Segmento E OFF  
digitalWrite(Seg_D,HIGH); // Segmento D ON  
digitalWrite(Seg_A,HIGH); // Segmento A ON  
digitalWrite(Seg_B,HIGH); // Segmento B ON  
digitalWrite(Seg_C,HIGH); // Segmento C ON  
digitalWrite(PD,HIGH); // PONTO DEC. ON  
delay(timer);
```

```
// DIGITO 4
```

```
digitalWrite(Seg_F,HIGH); // Segmento F ON  
digitalWrite(Seg_G,HIGH); // Segmento G ON  
digitalWrite(Seg_E,LOW); // Segmento E OFF  
digitalWrite(Seg_D,LOW); // Segmento D OFF  
digitalWrite(Seg_A,LOW); // Segmento A OFF  
digitalWrite(Seg_B,HIGH); // Segmento B ON  
digitalWrite(Seg_C,HIGH); // Segmento C ON  
digitalWrite(PD,LOW); // PONTO DEC. OFF  
delay(timer);
```

```
// DIGITO 5
```

```
digitalWrite(Seg_F,HIGH); // Segmento F ON  
digitalWrite(Seg_G,HIGH); // Segmento G ON  
digitalWrite(Seg_E,LOW); // Segmento E OFF  
digitalWrite(Seg_D,HIGH); // Segmento D OFF
```

```
digitalWrite(Seg_A,HIGH); // Segmento A OFF  
digitalWrite(Seg_B,LOW); // Segmento B ON  
digitalWrite(Seg_C,HIGH); // Segmento C ON  
digitalWrite(PD,HIGH); // PONTO DEC. OFF  
delay(timer);  
  
} // THE END
```



LED RGB

O que é? O que faz?

Um LED RGB incorpora três LEDs juntos, um vermelho (RED), um verde (GREEN) e um azul (BLUE). Podemos codificar as mais diversas cores que as combinações de vermelho, verde e azul podem nos oferecer.

Como funciona no LabVad?


```
int timer = 5000;
int LED_VM = 9;
int LED_VD = 10;
int LED_AZ = 11;

void setup()
{ // Obs: Todos os LEDs tem que estarem conectados em PINOS PWM
  pinMode(LED_VM, OUTPUT); // Pino_9
  pinMode(LED_VD, OUTPUT); // Pino_10
  pinMode(LED_AZ, OUTPUT); // Pino_11
}
void loop()

{
  // Cor 1 - APAGADO
  analogWrite(LED_VD, 0); // LED
  analogWrite(LED_AZ, 0); // LED
  analogWrite(LED_VM, 0); // LED VM
  delay(timer);
  // Cor 2 - VERMELHO
  analogWrite(LED_VM, 255); // LED VM
  analogWrite(LED_AZ, 0); // LED AZ
  analogWrite(LED_VD, 0); // LED VD
  delay(timer);
  // Cor 3 - VERDE
  analogWrite(LED_VM, 0); // LED VM
  analogWrite(LED_AZ, 0); // LED AZ
```

```
analogWrite(LED_VD, 255); // LED VD
delay(timer);
// Cor 4 - AZUL
analogWrite(LED_VM, 0); // LED VM
analogWrite(LED_AZ, 255); // LED AZ
analogWrite(LED_VD, 0); // LED VD
delay(timer);
// Cor 5 - Amarelo
analogWrite(LED_VD, 255); // LED VD
analogWrite(LED_VM, 255); // LED VM
analogWrite(LED_AZ, 0); // LED AZ
delay(timer);
// Cor 6 - Magenta
analogWrite(LED_VM, 255); // LED VM
analogWrite(LED_AZ, 255); // LED AZ
analogWrite(LED_VD, 0); // LED VD
delay(timer);
// Cor 7 - Ciano
analogWrite(LED_VM, 0); // LED VM
analogWrite(LED_AZ, 255); // LED AZ
analogWrite(LED_VD, 255); // LED VD
delay(timer);
// Cor 8 - BRANCO
analogWrite(LED_VM, 255); // LED VM
analogWrite(LED_AZ, 255); // LED AZ
analogWrite(LED_VD, 255); // LED VD
delay(timer);
}
```

//
// THE END



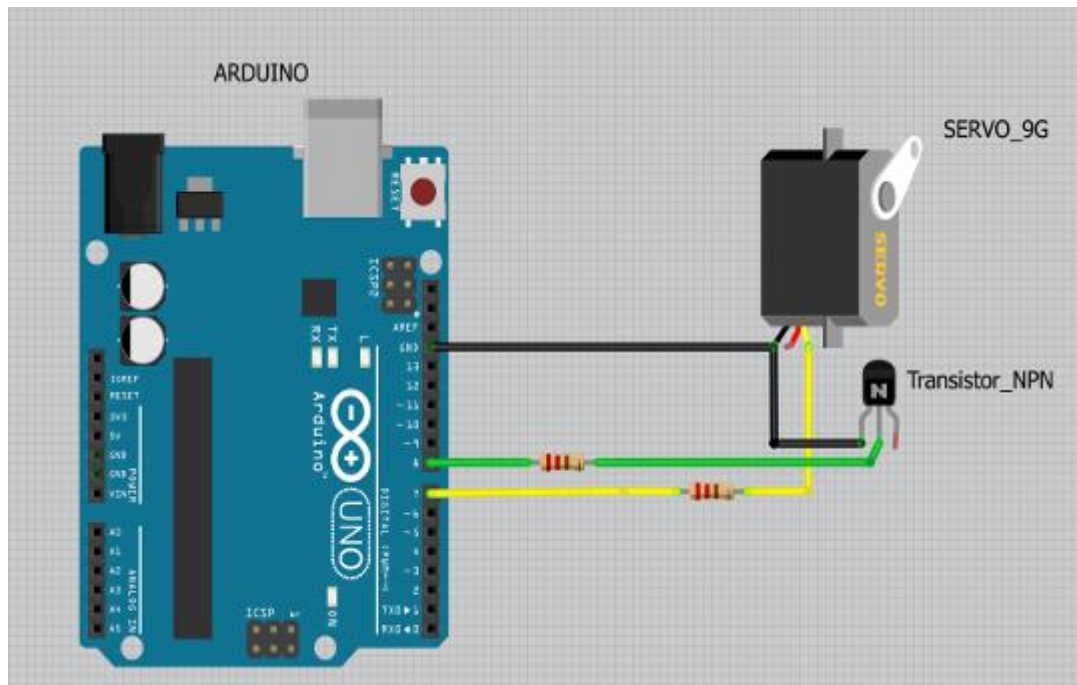
Micro Servo Motor

O que é? O que faz?

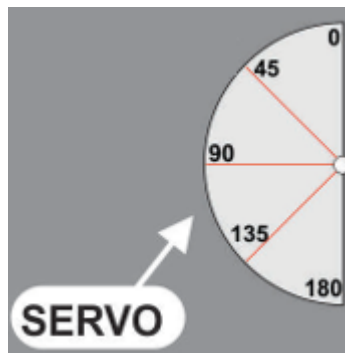
É uma máquina eletromecânica que apresenta **movimento proporcional a um comando**. Geralmente, recebem um sinal de controle, verificam a posição atual e atuam no sistema indo para a posição desejada.

Como funciona no LabVad?

Como podemos ver na figura abaixo, o servo motor do LabVad está conectado na entrada 7.



O movimento do servo motor do LabVad vai de 0° a 180°. Para auxiliar a programação temos a imagem abaixo no LabVad.



Veja um exemplo onde fica bem claro como você irá programá-lo.

```
//*****  
// S E R V O gira modulos de 45 graus *  
// By S.Brandao *  
// Em: 21/07//2014 - *  
// Experiencia com o Servo *  
//*****  
#include <Servo.h>  
  
Servo Meu_servo;  
  
void setup()  
{  
Meu_servo.attach(7); // Servo está conectado no I/O(7)  
}  
  
void loop()  
{  
  
Meu_servo.write(0);  
delay(5000); // tempo para o SERVO se posicionar.  
  
// Posiciona o SERVO a 45 GRAUS da origem.  
Meu_servo.write(45);  
delay(5000);
```



```
// Posiciona o SERVO a 90 GRAUS da origem.  
Meu_servo.write(90);  
delay(5000);  
  
// Posiciona o SERVO a 135 GRAUS da origem.  
Meu_servo.write(135);  
delay(5000);  
  
// Posiciona o SERVO a 180 GRAUS da origem.  
Meu_servo.write(180);  
delay(5000);  
  
// Posiciona o SERVO na origem.  
Meu_servo.write(0);  
delay(5000);  
  
}
```



Relé

O que é? O que faz?

O que é? O que faz?

O relé é um dispositivo eletromecânico, que serve para ligar ou desligar outros dispositivos. Podemos utilizar um pequeno sinal de corrente contínua de 5V para acionar uma lâmpada de 110V. Podemos elaborar projetos mais robustos como automatizar toda uma residência com uma simples placa Arduino. Por isso, no LabVad, utilizamos a ilustração de uma cafeteira.

Motor DC

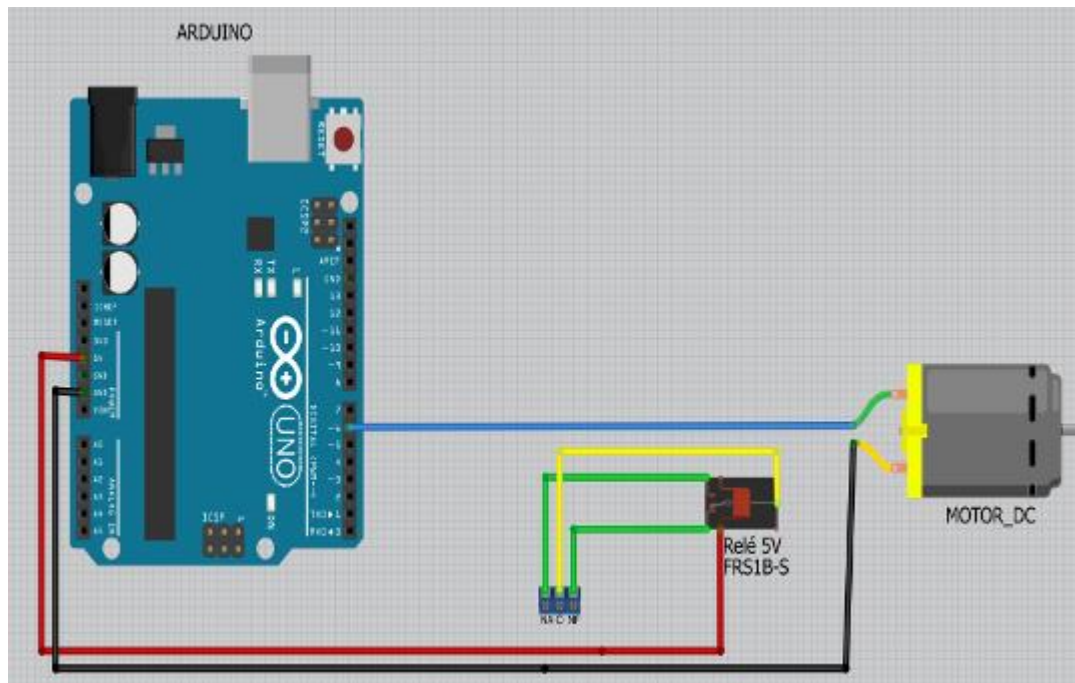
O que é? O que faz?

Motores são usados em aplicações fixas e móveis. São encontrados em processos de fabricação (em esteiras, por exemplo), em sistemas de locomoção de robôs, cadeiras de roda motorizada, além de muitos outros projetos. Sabe aquele projeto bacana com chasis

e robôs segue linha? Isso tudo só é possível através de motores DC. Esperamos que os exercícios feitos no LabVad estimule você a criar muitos robôs que funcionam através de motores.

Veja como o Relé e o Motor DC funcionam no LabVad?

Como podemos ver na figura abaixo, o Motor DC do LabVad está conectado na entrada 6 e o Relé na entrada 8.



Vamos ver um código de como testar nosso Relé.

```
/* Experimento _ RELE - Acionando uma carga com tempo programado
```

```
* By S.Brandao
```

```
* Em: 25/07/2014 -
```

```
*/
```

```
int RELE = 8; // Selecciona o pino da saida 8
```

```
int Timer = 15000; // Tempo de acionamento e desligamento
```

```
// Saídas não utilizadas neste experimento da cat: MISCELANEA
```

```
int saida_6 = 6;
```

```
int saida_7 = 7;
```

```
int LED_VM = 9;
```

```
int LED_VD = 10;
```

```
int LED_AZ = 11;
```

```

void setup() {

    pinMode(RELE, OUTPUT); // configura a porta do RELE como saida.

    // Decodificador do MUX.*****

    pinMode(LED_VM0,OUTPUT); // Pino_3 - Decodificador do Conjunto bit_0
    pinMode(LED_VM1,OUTPUT); // Pino_5 - Decodificador do Conjunto bit_1

    // Habilita MUX - 01 - Conjunto de LEDs - LED0 a LED7

    digitalWrite(LED_VM0, HIGH); // LED VM0
    digitalWrite(LED_VM1, HIGH); // LED VM1

    // *****

    // Resete das saidas NAO utilizadas neste experimento

    pinMode(saida_6,OUTPUT); // Pino_6 MOTOR
    pinMode(saida_7,OUTPUT); // Pino_7 SERVO

```

```
pinMode(LED_VM, OUTPUT); // Pino_9

pinMode(LED_VD, OUTPUT); // Pino_10

pinMode(LED_AZ, OUTPUT); // Pino_11

}

void loop() {

digitalWrite(RELE, LOW);

delay(Timer);

digitalWrite(RELE, HIGH); // Aciona o RELE por um tempo pre-definido

delay(Timer);

// F I M

}
```

Veja um exemplo de como programar o Motor DC.

```
/* Experimento _ Motor DC com velocidade controlada
 * By S.Brandao
```

```

* Em: 25/07/2014 -
*/

int Motor_DC = 6; // Seleciona o pino de saída PWM para o motor
int Valor = 125; // Valor de 0 a 45 (parado) de 50(vel.min.) a 255(vel. max)

void setup() {

  pinMode(Motor_DC, OUTPUT); // configura a porta do motor como saída

  // Resete das saidas NAO utilizadas neste experimento
  pinMode(saida_7,OUTPUT); // Pino_7 SERVO
  pinMode(saida_8,OUTPUT); // Pino_8 RELE
  pinMode(LED_VM, OUTPUT); // Pino_9
  pinMode(LED_VD, OUTPUT); // Pino_10
  pinMode(LED_AZ, OUTPUT); // Pino_11
}

void loop() {

  analogWrite(Motor_DC, Valor); // Aciona o Motor
}

```