

Automação Residencial

[Criando seus próprios
projetos de automação
residencial]

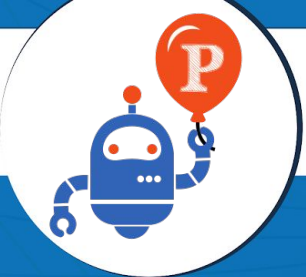
Ministrante: Guilherme Augusto Bileki (bileki@usp.br)

Monitor: Breno Cunha Queiroz

Universidade de São Paulo

ROBÔS NA ESCOLA



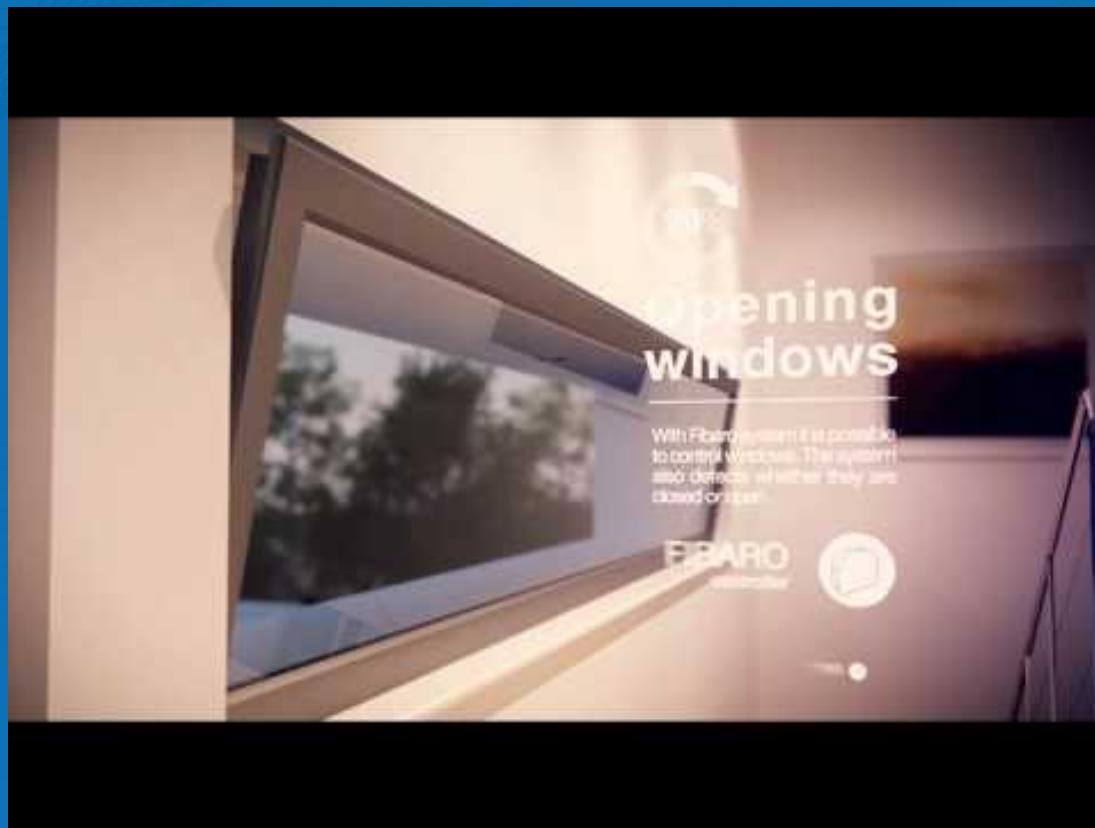


Tópicos

- Motivação
- Introdução
- Arduino
 - Instalação da IDE
 - Bibliotecas
- Componentes
- Aplicativo de controle
- Exemplos
 - Pisca-pisca
 - Analógico/digital
 - Sensores
 - Comunicação
- Mãos na massa
- Discussão



Motivação





Introdução

Automação

- É a tecnologia pela qual um processo ou procedimento é realizado com assistência humana mínima
- Possui como objetivo aumentar a eficiência dos processos, maximizar a produção com o menor consumo de energia, menor emissão de resíduos e melhores condições de segurança, seja material, humana ou das informações
- É um passo além da mecanização, onde operadores humanos dispõem de máquinas para auxiliá-los em seus trabalhos



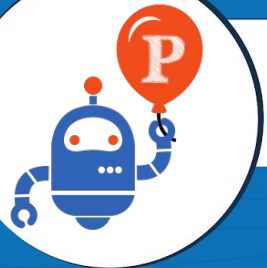
Introdução

Laços de controle

- *Aberto*: independe da saída do processo, ex: janela controlada por temporizador para abrir de manhã e fechar de noite
- *Fechado*: depende da saída do processo, ex: luz controlada por sensor de presença

Ações de controle

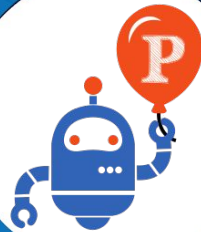
- *Discreto*: liga/desliga
- *PID*: ajuste baseado nos componentes proporcional, derivativo e integrativo
- *Estados*: depende do estado do sistema atual para tomada de decisão
- *Computador*: pode ser programado para agir de todas as formas acima



Introdução

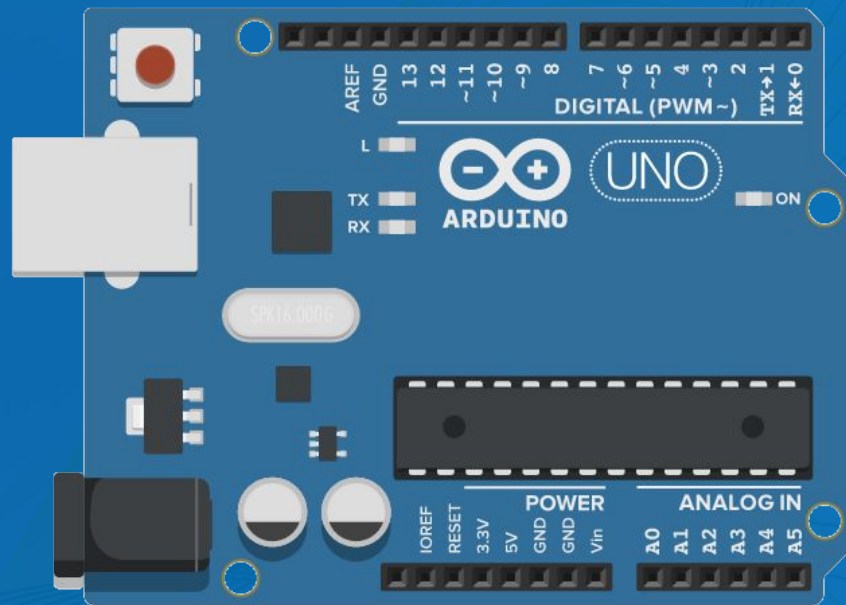
Exemplos de automação na indústria

- *Indústria automobilística*
 - Máquinas de solda
 - Processos de pintura
- *Indústria química*
 - Dosagem de produtos para misturas
 - Estações de tratamento de efluentes
- *Indústria de mineração*
 - Britagem de minérios
 - Carregamento de vagões
- *Indústria de papel e celulose*
 - Corte e descascamento de madeira
 - Corte e embalagem



Arduino

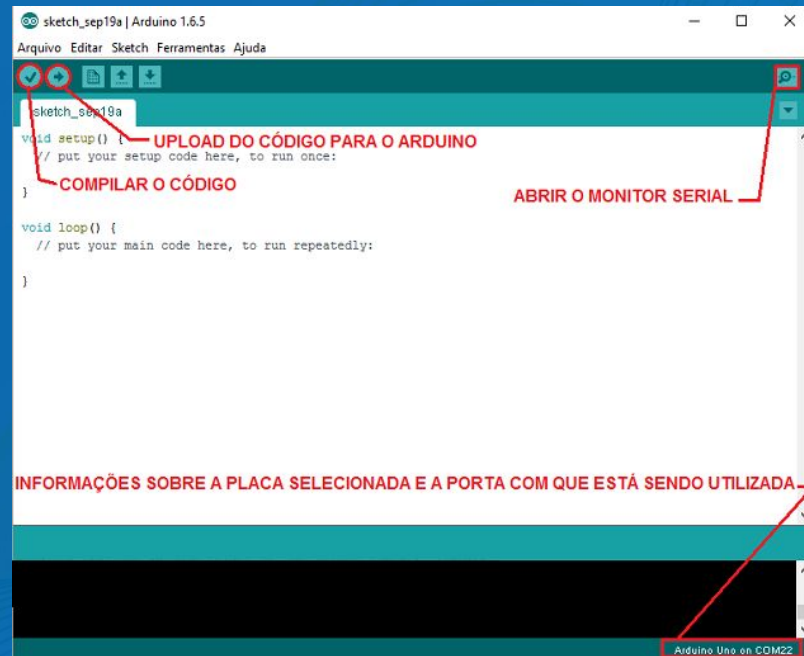
- Plataforma de prototipagem eletrônica de hardware livre e de placa única
- Projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido
- O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais





Arduino - IDE

- Linguagem de programação é essencialmente C/C++
- Versões para Windows, Linux, Mac OSX e online
- Suporte para diversas placas
- Suporte a diversas bibliotecas
- Dicas:
 - No Linux não é necessário instalar nenhum driver, entretanto normalmente é necessário dar permissão de acesso a porta USB
 - `sudo usermod -a -G dialout <username>`
 - `sudo chmod a+rw /dev/tty[ACM0 ou USB0]`
 - No Windows é necessário fazer a instalação do driver específico para seu Arduino





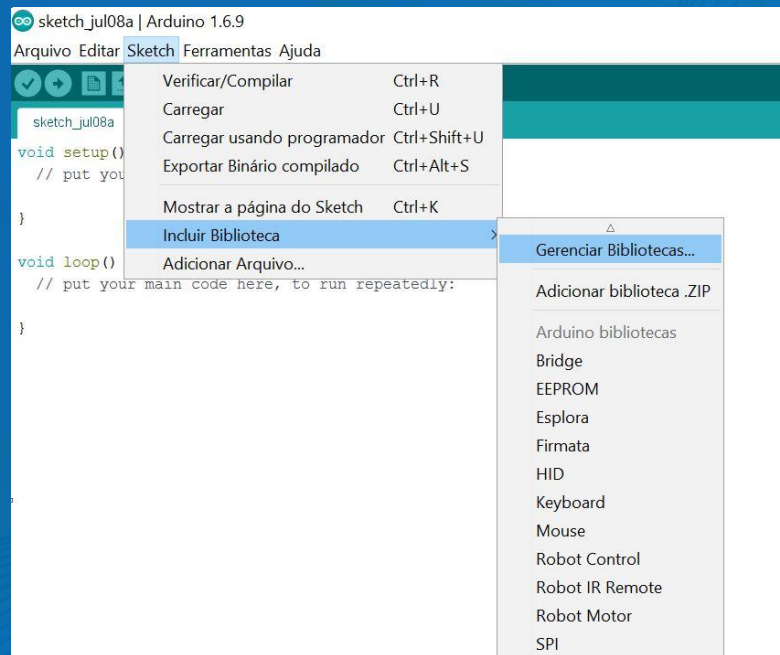
Arduino - Bibliotecas

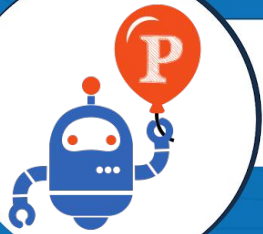
Instalação pode ser feita:

- Diretamente pela IDE
- Pacotes zipados

O mesmo serve para adicionar placas

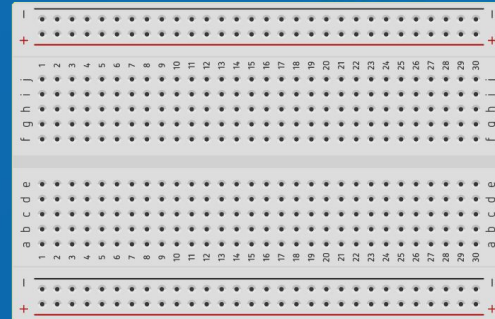
- Ferramentas > Placas > Gerenciar Placas





Componentes - Básicos

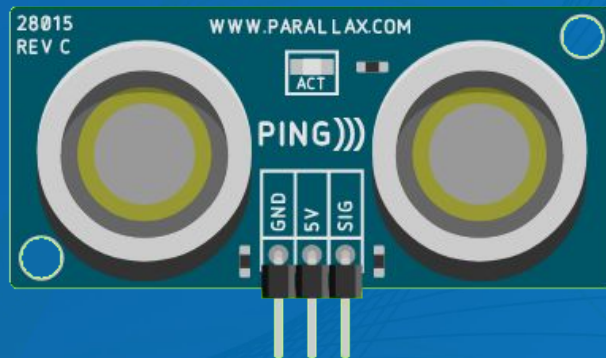
- *Protoboard*: utilizada para prototipar circuitos sem a necessidade de soldas ou emendas de fios
- *Jumpers*: fios com conectores soldados
- *Leds*: é um componente capaz de emitir luz
- *Resistor*: utilizado para limitar, diminuir ou dividir a corrente ou tensão do circuito
- *Botão*: utilizado para controlar o circuito, é análogo a um interruptor
- *Potenciômetro*: possui resistência variável que pode ser controlada mecanicamente





Componentes - Sensores

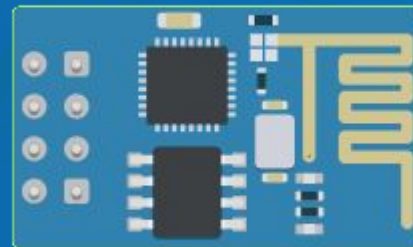
- *Sensor de luz*: é um uma resistência sensível a luz, que gera um sinal analógico baseado na quantidade de luz captada
- *Sensor de temperatura*: é um transistor sensível a temperatura, que gera um sinal analógico baseado na quantidade de calor captado
- *Sonar*: é um instrumento que calcula a distância baseado no tempo em que uma onda sonora emitida demora para ser recebida





Componentes - Comunicação

- *Módulo Bluetooth:* é um módulo de comunicação sem fio de alcance aproximado de 10m que pode ser definido como mestre ou escravo e se conecta diretamente a outro módulo Bluetooth
- *Módulo Wi-Fi:* é um módulo de comunicação sem fio que pode ser conectado por meio de um roteador a outros componente que utilizam a mesma rede

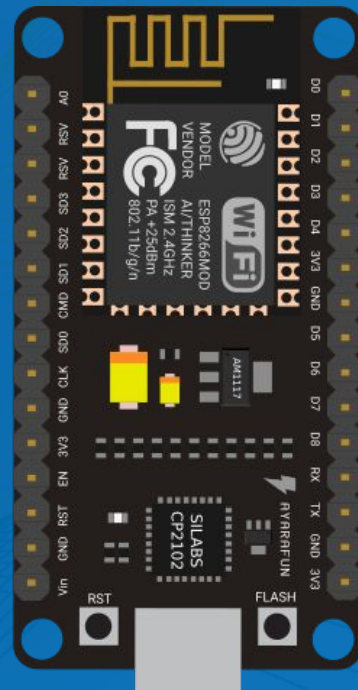


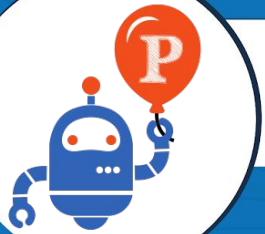


Componentes - NodeMCU

NodeMCU:

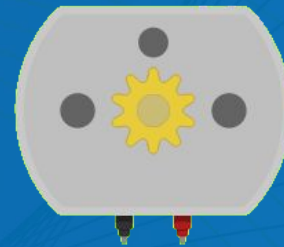
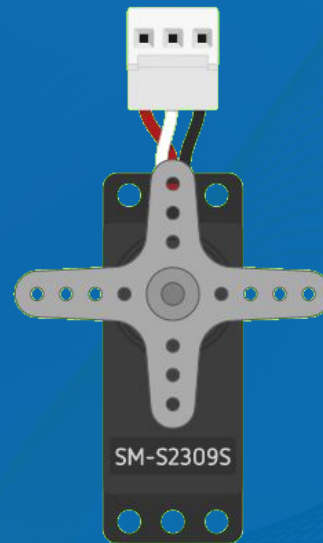
- Um Arduino com Wi-Fi integrado
- Difere um pouco na programação e bibliotecas
- Para adicionar a placa na IDE:
 - Arquivo > Preferências > Additional Boards Manager URLs:
 - http://arduino.esp8266.com/stable/package_esp8266com_index.json





Componentes - Atuadores

- *Servo-motor*: é uma máquina eletromecânica que apresenta movimento proporcional a um comando
- *Motor DC*: é uma máquina eletromecânica que apresenta movimento contínuo, recebendo apenas o comando de liga e desliga





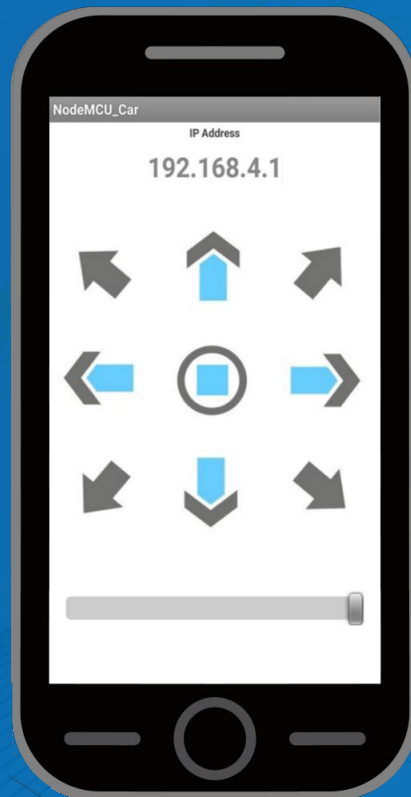
Aplicativo de controle

Exemplo baixado do GitHub do Google e modificado para conexão com módulo de Bluetooth genérico:

- Modificação da UUID de característica para componentes genéricos
- Modificação na função de leitura de características
- Adição da função de escrita de características
- Adaptação do layout

<https://www.youtube.com/watch?v=mJEftoeSBOU>

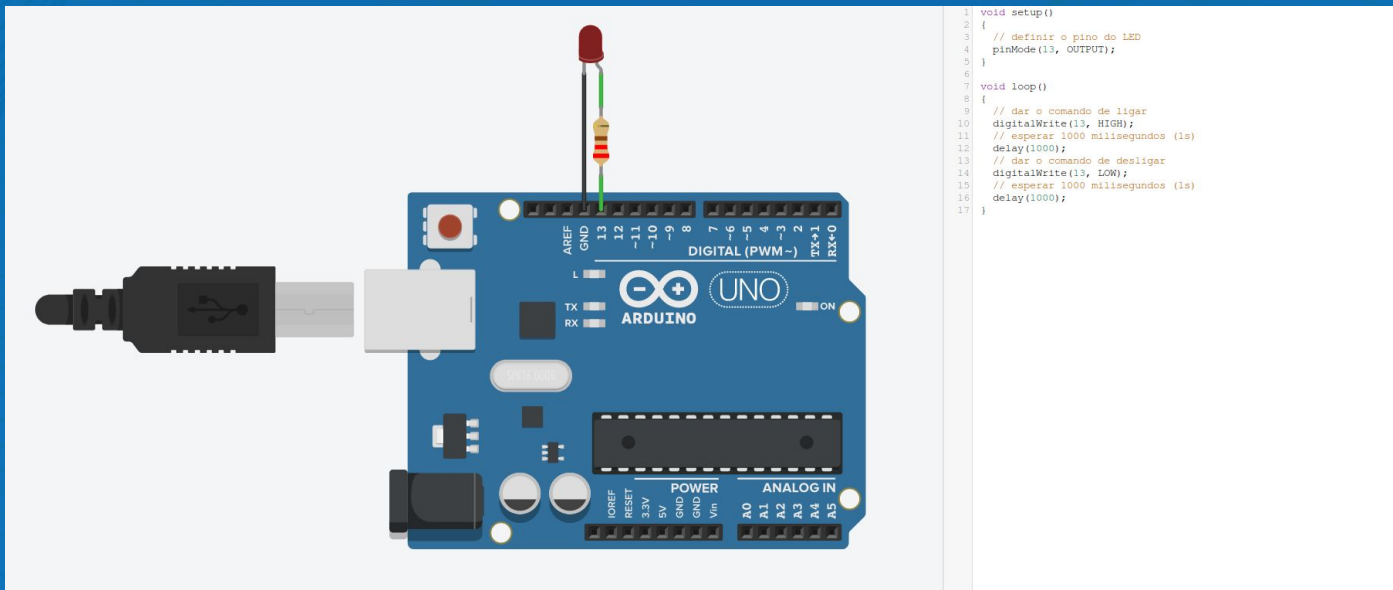
<https://play.google.com/store/apps/details?id=io.dabbleapp>





Exemplos - Pisca-pisca

- Arduino IDE > Arquivo > Exemplos > 1.Basics > Blink
- <https://www.tinkercad.com/things/0jaMCzY8l8s-blink>





- ```

1 void setup() {
2 // iniciar o monitor serial
3 Serial.begin(9600);
4 // definir o botão de pressão
5 pinMode(2, INPUT_PULLUP);
6 // definir o LED do Arduino como saída
7 pinMode(13, OUTPUT);
8 }
9
10
11 void loop() {
12 // ler o valor do botão
13 int valorBotao = digitalRead(2);
14
15 int valorPotenciometro = analogRead(A0);
16 //print out the value of the pushbutton
17 Serial.print("Botao: ");
18 Serial.print(valorBotao);
19 Serial.print(", Potenciometro: ");
20 Serial.println(valorPotenciometro);
21
22 // verificar se o botão está pressionado ou não para ligar o LED
23 if (valorBotao == HIGH) {
24 digitalWrite(13, LOW);
25 } else {
26 digitalWrite(13, HIGH);
27 }
28 }

```

Serial Monitor

```

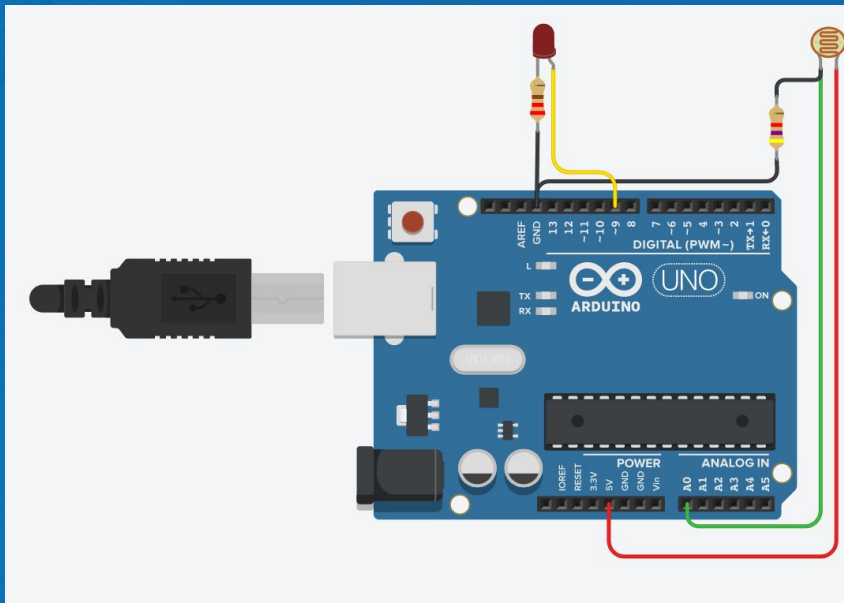
Botao: 1, Potenciometro: 61
Botao: 1, Potenciometro: 61
Botao: 1, Potenciometro: 61
Botao: 1, Potenciometro: 61
Botao: 1, Potenciometro: 61
Botao: 1, Potenciometro: 61

```



# Exemplos - Sensores

- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada sensor
- <https://www.tinkercad.com/things/hZCVqkOKq3-sensor-de-luz/>



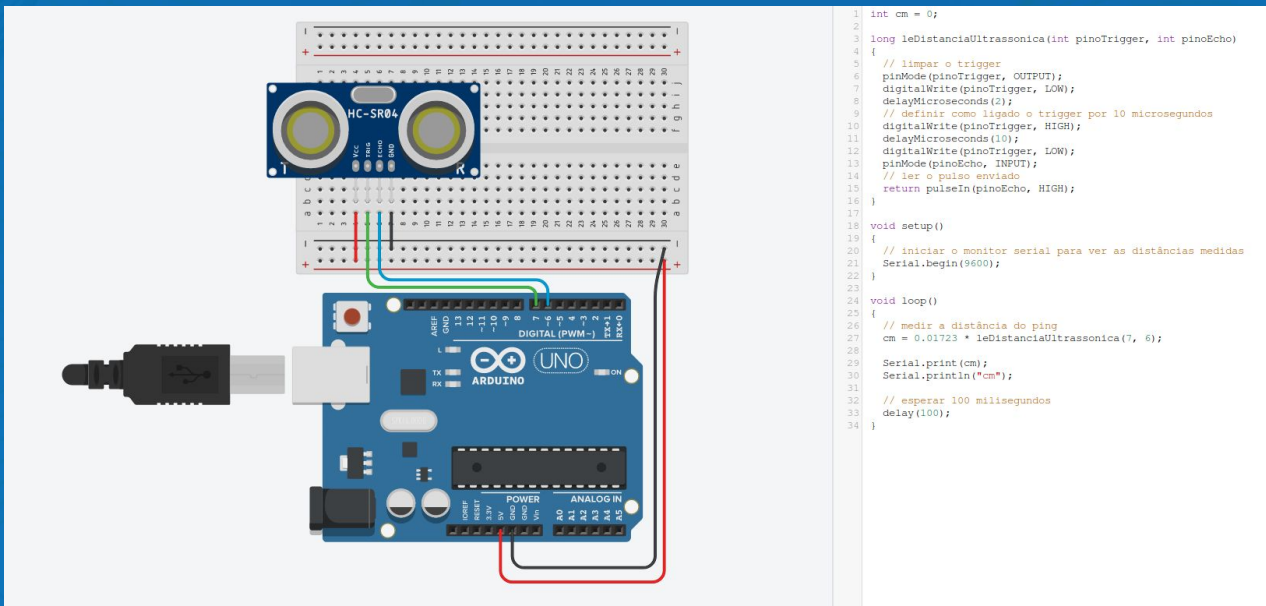
```
1 // variável para ler o valor do sensor de luz
2 int valorSensor = 0;
3
4 void setup()
5 {
6 // definir pino A0 como entrada
7 pinMode(A0, INPUT);
8 // iniciar monitor serial para ver o valor lido
9 Serial.begin(9600);
10
11 // definir pino do LED
12 pinMode(9, OUTPUT);
13 }
14
15 void loop()
16 {
17 // ler o valor registrado pelo sensor
18 valorSensor = analogRead(A0);
19 // exibir no monitor serial o valor obtido
20 Serial.println(valorSensor);
21 // mapear o valor do sensor (0-1023) para a intensidade do LED (0-255)
22 analogWrite(9, map(valorSensor, 0, 1023, 0, 255));
23 // esperar 100 milissegundos
24 delay(100);
25 }
```





# Exemplos - Sensores

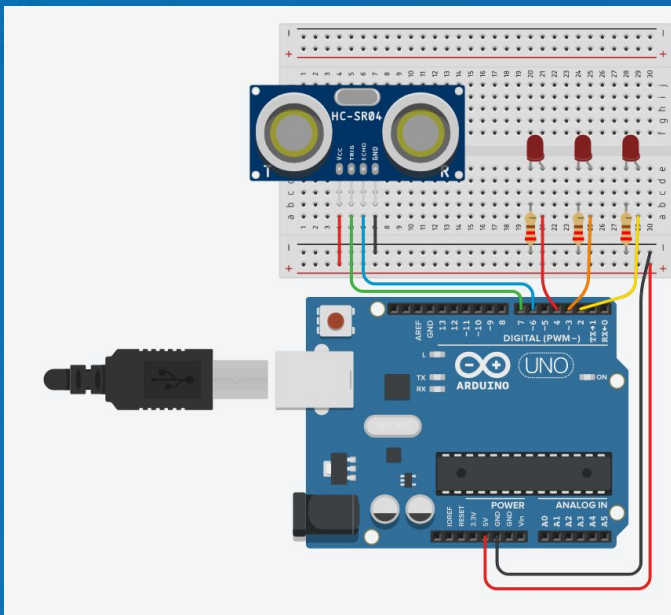
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada sensor
- <https://www.tinkercad.com/things/0Q20aht4wiF-sonar/>





# Exemplos - Sensores

- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada sensor
- <https://www.tinkercad.com/things/3fSIMyId4Cm-sonar-com-leds/>

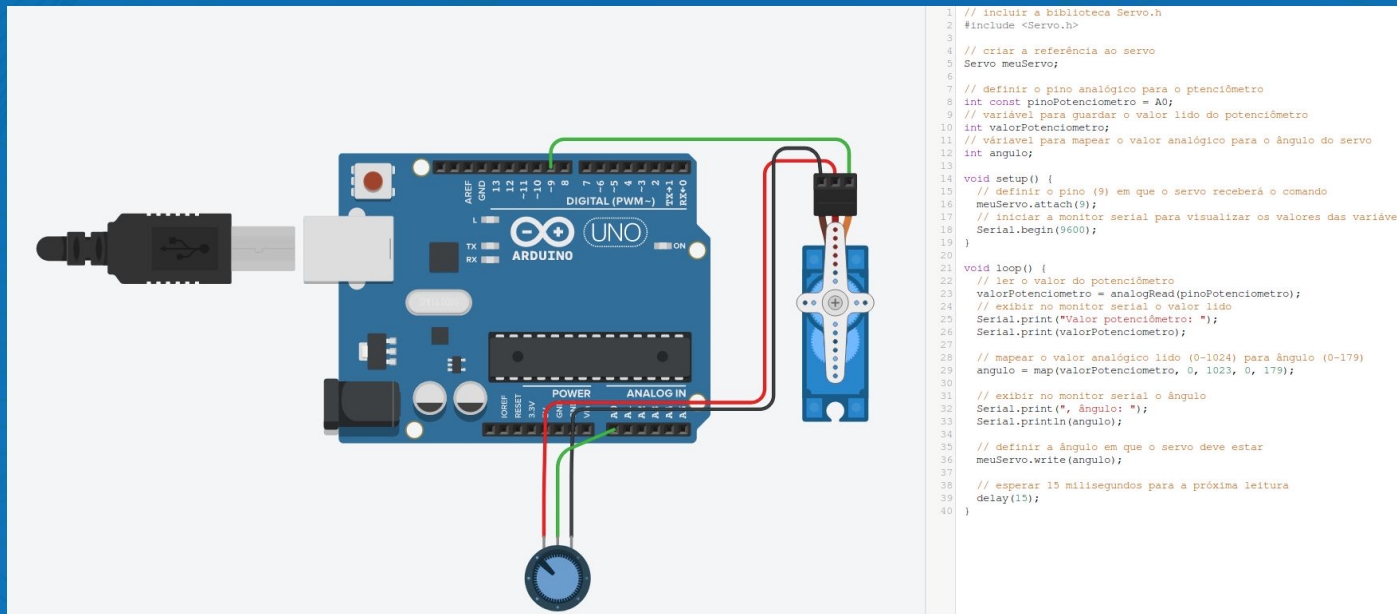


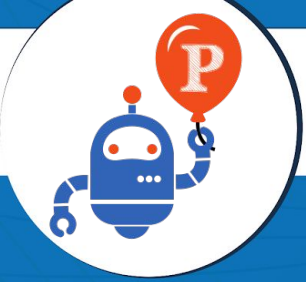
```
1 int limiteDistancia = 0;
2 int cm = 0;
3
4 long leDistanciaUltrasonica(int pinoTrigger, int pinoEcho)
5 {
6 // limpar o trigger
7 pinMode(pinoTrigger, OUTPUT);
8 digitalWrite(pinoTrigger, LOW);
9 delayMicroseconds(2);
10 // definir como ligado o trigger por 10 microsegundos
11 digitalWrite(pinoTrigger, HIGH);
12 delayMicroseconds(10);
13 digitalWrite(pinoTrigger, LOW);
14 pinMode(pinoEcho, INPUT);
15 // ler o pulso enviado
16 return pulseIn(pinoEcho, HIGH);
17 }
18
19 void setup()
20 {
21 // iniciar o monitor serial para ver as distâncias medidas
22 Serial.begin(9600);
23
24 // definir os pinos dos LEDs
25 pinMode(2, OUTPUT);
26 pinMode(3, OUTPUT);
27 pinMode(4, OUTPUT);
28 }
29
30 void loop()
31 {
32 // definir limite de distância para ativar cada LED
33 limiteDistancia = 350;
34 // medir a distância do ping
35 cm = 0.01723 * leDistanciaUltrasonica(7, 6);
36
37 Serial.print(cm);
38 Serial.println("cm");
39
40 // dependendo do limite aceitável, acender mais ou menos LEDs
41 if (cm > limiteDistancia) {
42 digitalWrite(2, LOW);
43 digitalWrite(3, LOW);
44 digitalWrite(4, LOW);
45 }
46 if (cm <= limiteDistancia && cm > limiteDistancia - 100) {
47 digitalWrite(2, HIGH);
48 digitalWrite(3, LOW);
49 }
```



# Exemplos - Atuadores

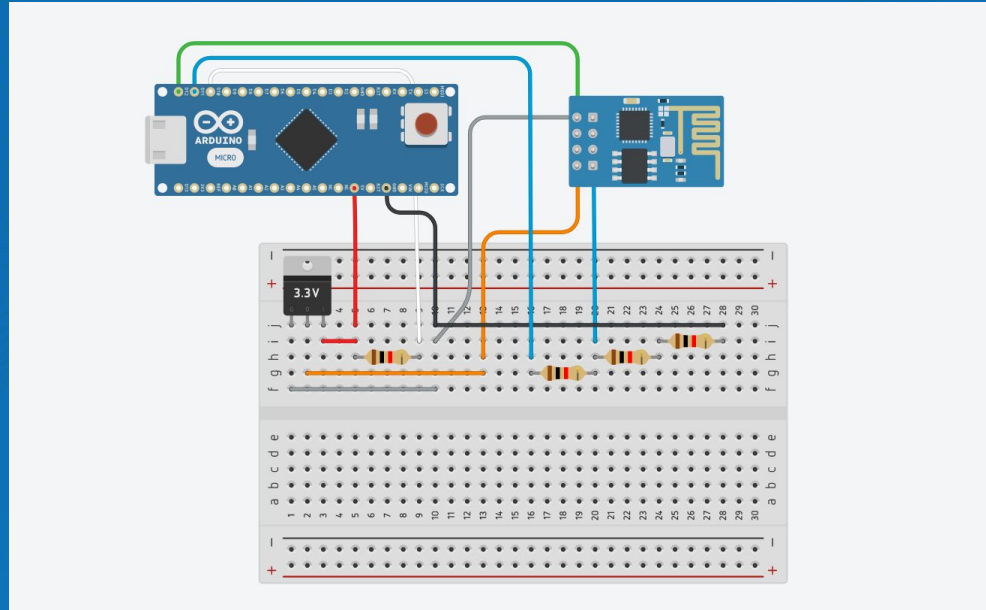
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada atuador
- <https://www.tinkercad.com/things/9CTUzW9i36K-servo/>



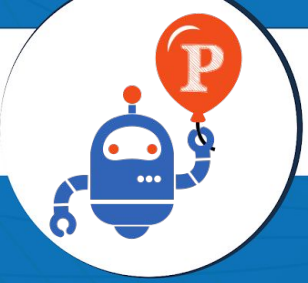


# Exemplos - Comunicação

- Arduino IDE > Arquivo > Exemplos > Bibliotecas de comunicação
- <https://www.tinkercad.com/things/19iiXJ6NPzG-circuito-basico-1-kit-bt/>

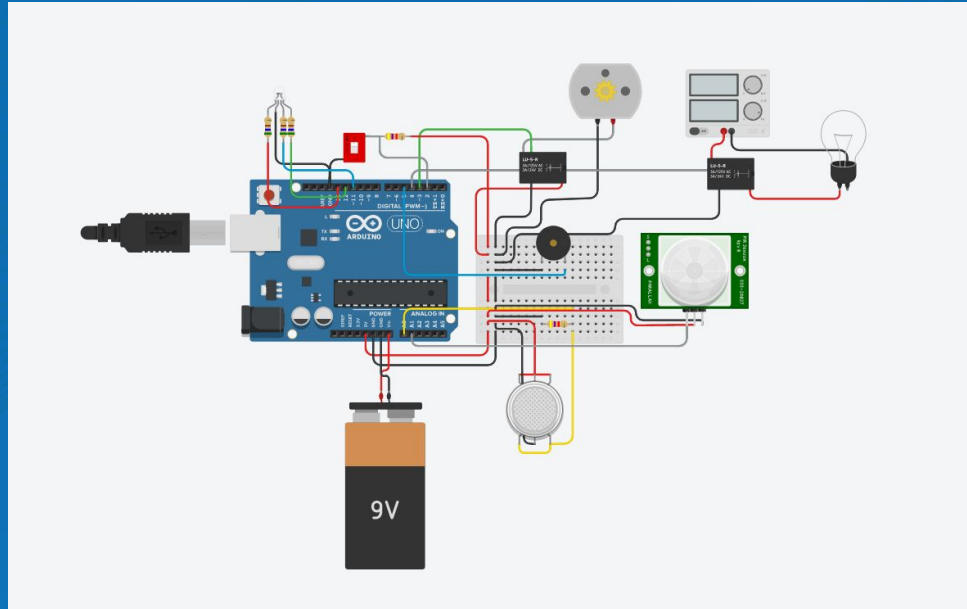






# Exemplos - Juntando Tudo

- <https://www.tinkercad.com/things/3ZxFs6prJjI-automacao-residencial/>







# Mãos na massa

- **Formem grupos de 5 pessoas**
- **Pensem em um projeto para resolver um problema ou para criar uma facilidade na casa de um participante**
- **Quebrem o problema em pequenas partes e separem entre os integrantes**
- **Separem os componentes necessários para desenvolver o projeto**
- **Com a limitação dos componentes disponíveis, simule algumas etapas da solução do problema**
- **Caso o seu problema envolva sinais externos, simule-os**

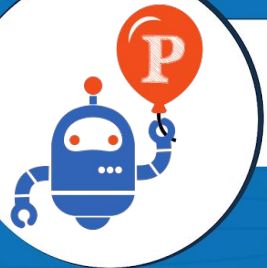
**BOA SORTE!**



# Discussão

## Sobre o vídeo inicial:

- o que nos falta para fazer igual?
- como podemos diminuir os custos?
- o que podemos melhorar?
- quais outras funções poderíamos propor?



# Links

- **Tutorial NodeMCU+Android:** <https://www.youtube.com/watch?v=psqv5z3kYW8>
- **Projeto do APK utilizado:** <https://www.youtube.com/watch?v=d-Ub9-CIEyU>
- **Galeria de circuitos prontos:** <https://www.tinkercad.com/things?type=circuits>
- **Documentação do Arduino:** <https://www.arduino.cc/en/Guide/HomePage>
- **App Inventor do MIT:** <http://ai2.appinventor.mit.edu/>
- **Tutoriais do Dabble:** <https://thetempedia.com/products/dabble-app/>
- **GitHub do Robôs na Escola:** <https://github.com/simoesusp/RobosNaEscola>
- **GitHub com projetos legais:** <https://github.com/simoesusp>
- **GitHub do Bileki:** <https://github.com/bileki>



# Dúvidas





# Agradecimento

# Obrigado!

ROBÔS NA ESCOLA