

Automação Residencial

[Criando seus próprios
projetos de automação
residencial]

Ministrante: Guilherme Augusto Bileki
Email: bileki@usp.br
Universidade de São Paulo

ROBÔS NA ESCOLA



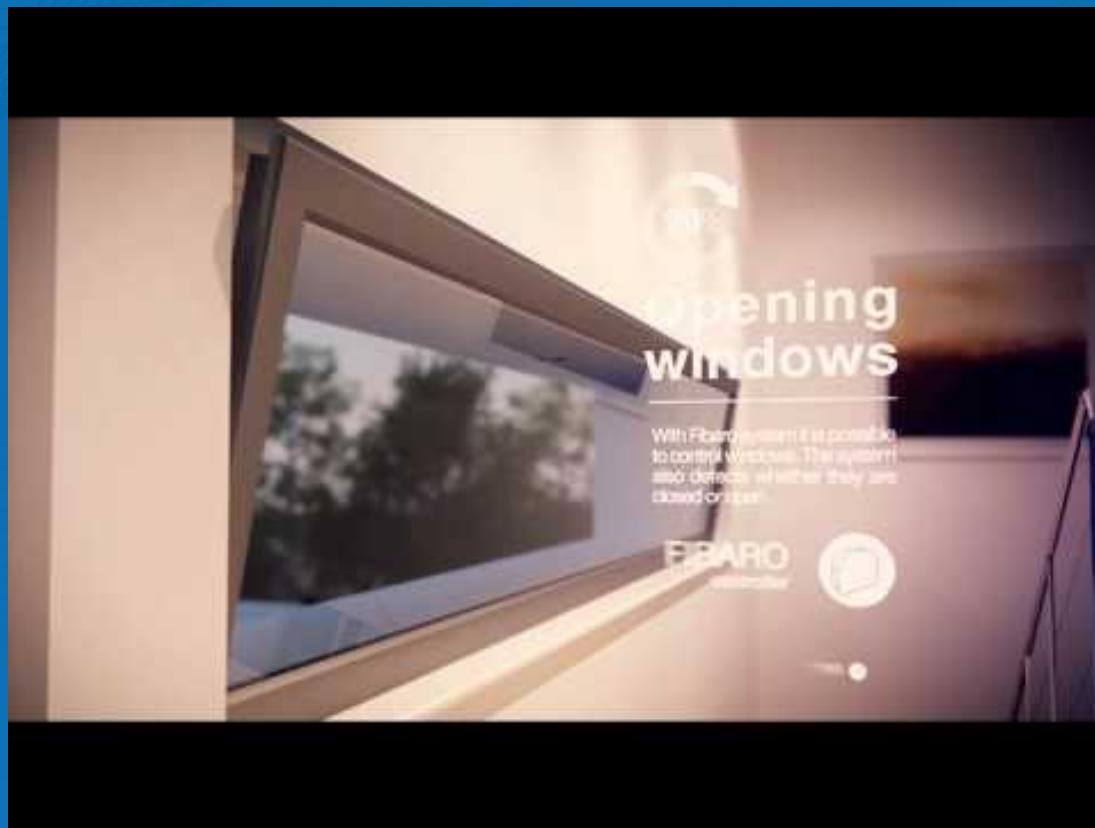


Tópicos

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



Motivação

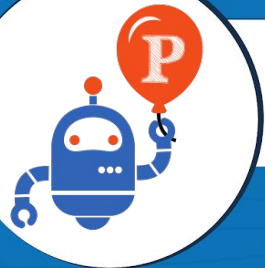




Introdução

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

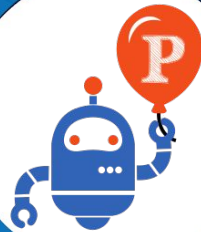
- *Discreto*: liga/desliga
- *PID*: ajuste baseado nos componentes proporcional, derivativo e integrativo
- *Estados*: depende do estado do sistema atual para tomada de decisão
- *Computador*: pode ser programado para agir de todas as formas acima



Introdução

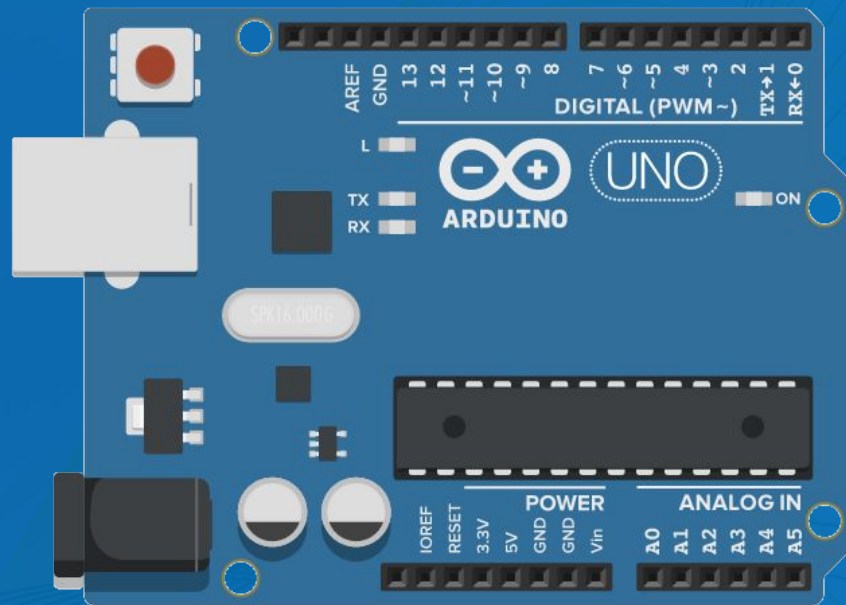
Exemplos de automação na indústria

- *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

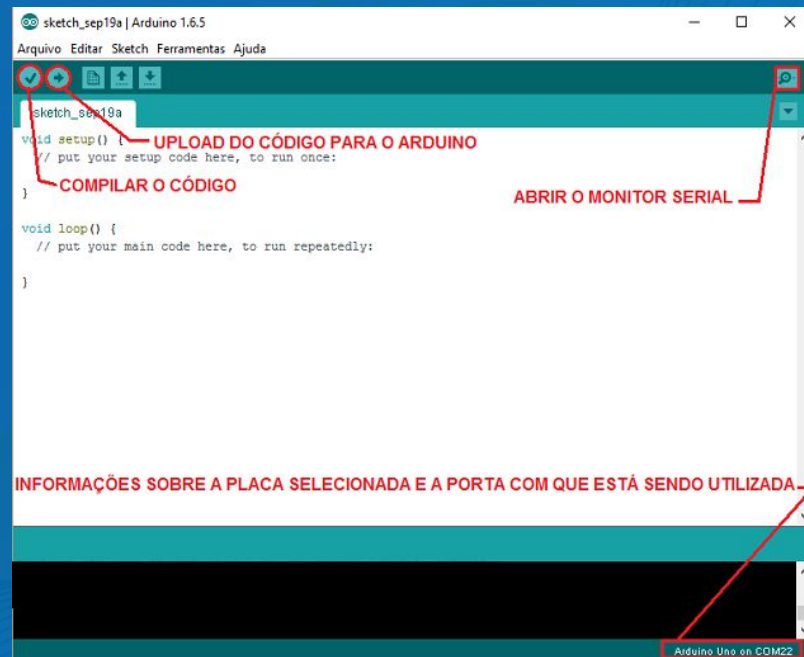
- Plataforma de prototipagem eletrônica de hardware livre e de placa única
- Projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido
- O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais





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





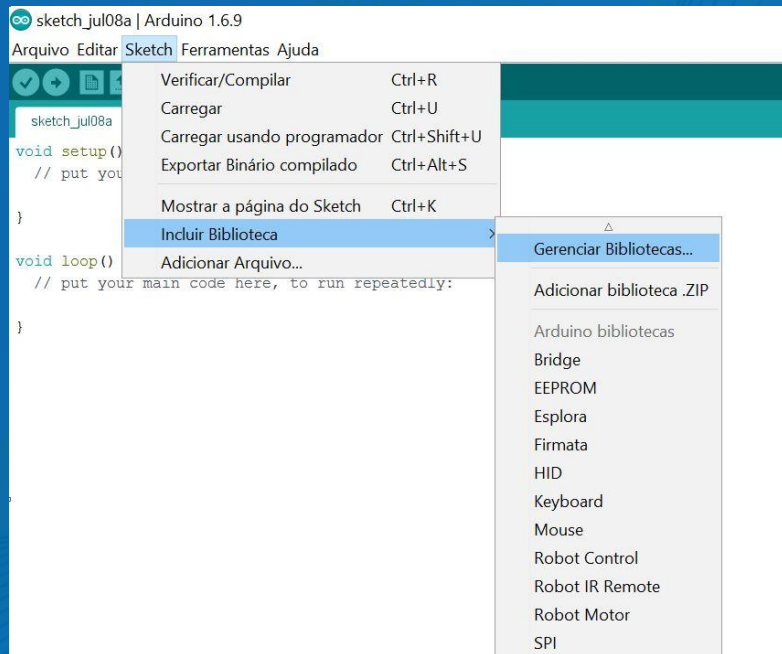
Arduino - Bibliotecas

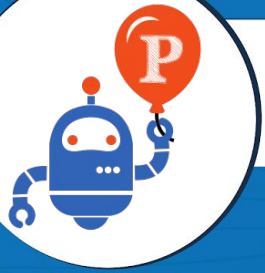
Instalação pode ser feita:

- Diretamente pela IDE
- Pacotes zipados

O mesmo serve para adicionar placas

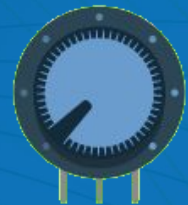
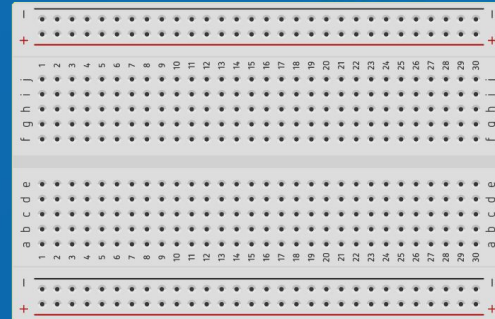
- Ferramentas > Placas > Gerenciar Placas





Componentes - Básicos

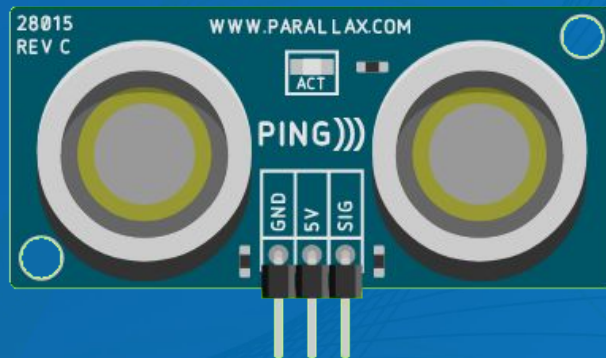
- *Protoboard*: utilizada para prototipar circuitos sem a necessidade de soldas ou emendas de fios
- *Jumpers*: fios com conectores soldados
- *Leds*: é um componente capaz de emitir luz
- *Resistor*: utilizado para limitar, diminuir ou dividir a corrente ou tensão do circuito
- *Botão*: utilizado para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





Componentes - Sensores

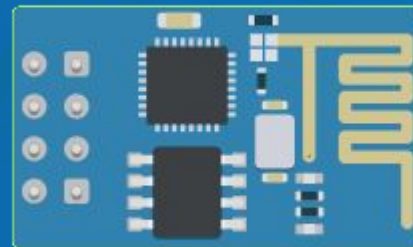
- *Sensor de luz*: é um uma resistência sensível a luz, que gera um sinal analógico baseado na quantidade de luz captada
- *Sensor de temperatura*: é um transistor sensível a temperatura, que gera um sinal analógico baseado na quantidade de calor captado
- *Sonar*: é um instrumento que calcula a distância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

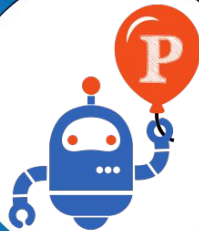




Componentes - Comunicação

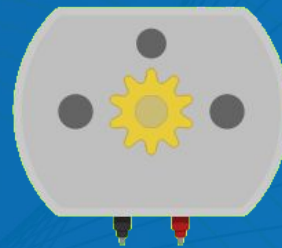
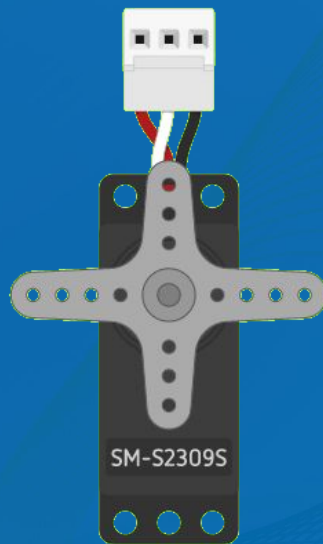
- *Módulo Bluetooth:* é um módulo de comunicação sem fio de alcance aproximado de 10m que pode ser definido como mestre ou escravo e se conecta diretamente a outro módulo Bluetooth
- *Módulo Wi-Fi:* é um módulo de comunicação sem fio que pode ser conectado por meio de um roteador a outros componente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





Componentes - 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

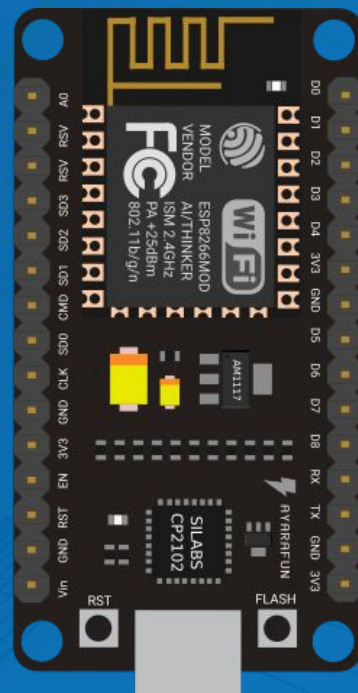


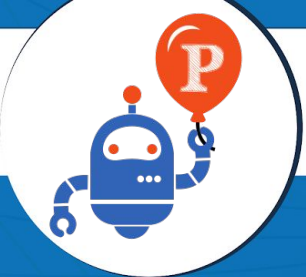


Componentes - NodeMCU

NodeMCU:

- Um Arduino com Wi-Fi e Bluetooth integrado
- Difere um pouco na programação e bibliotecas



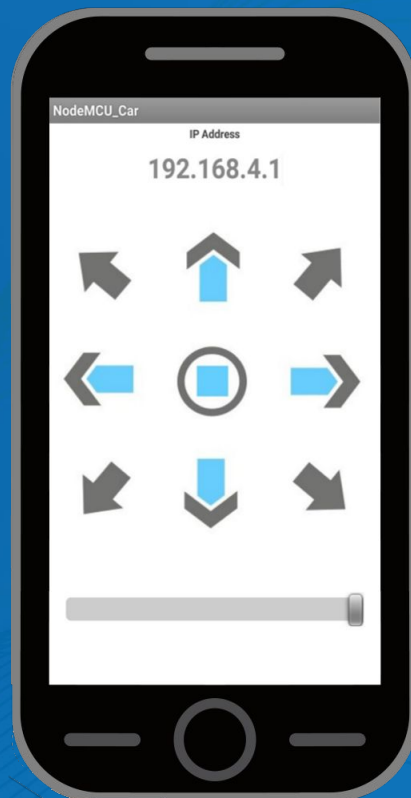


Aplicativo de controle

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>





Instalação - Arduino

Linux:

Baixar pacote do site do Arduino (32 ou 64 bits)
Descompactar e entrar na pasta descompactada
Executar o comando `./install.sh`

Testando:

Conectar o Arduino no PC
Abrir a IDE do Arduino e carregar o blink
Menu Arquivo > Exemplos > 01.Basic > Blink
Configurar no menu Ferramentas a placa e porta
Upar o código blink para a placa

Em caso de erro de permissão:

Executar os seguintes comandos
`"sudo usermod -a -G dialout <username>"`
`"sudo chmod a+rw /dev/tty[ACM0 ou USB0]"`

Windows:

Baixar instalador do site do Arduino ou o .zip
Instalar ou descompactar o zip
Executar a IDE

Testando:

Conectar o Arduino no PC
Abrir a IDE do Arduino e carregar o blink
Menu Arquivo > Exemplos > 01.Basic > Blink
Configurar no menu Ferramentas a placa e porta
Upar o código blink para a placa

Em caso de a porta não aparecer:

Ir até o gerenciador de dispositivos
No item Portas(COM & LPT) > botão direito >
atualizar driver > adicionar pasta do zip do Arduino



Instalação - NodeMCU

Linux e Windows:

Abrir a IDE do Arduino

Menu Arquivo > Preferências

Adicionar a URL abaixo nas URLs adicionais

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Menu Ferramentas > Placas > Gerenciar Placas

Pesquisar por "ESP" e instalar a biblioteca ESP8266

Testando:

Menu Arquivo > Exemplos > 01.Basic > Blink

Menu Ferramentas > escolher NodeMCU 1.0

Upar o código blink para a placa

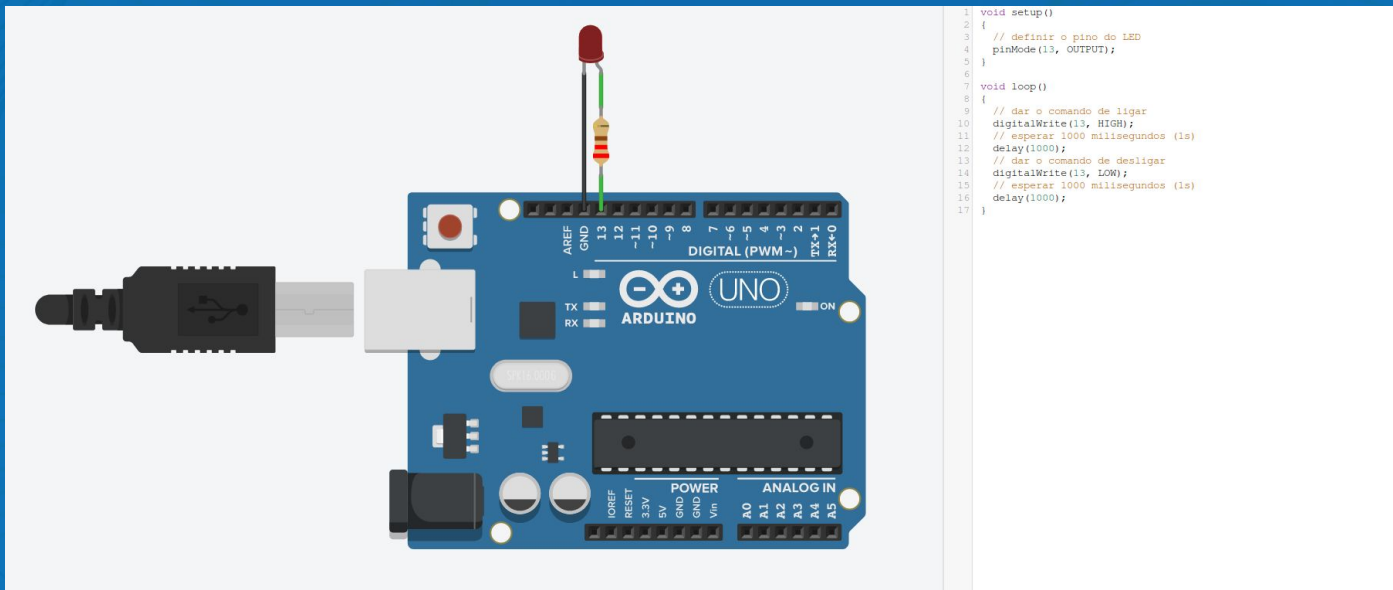
Em caso de erro a porta não aparecer (WINDOWS):

Instalar o driver necessário indicado na placa



Exemplos - Pisca-pisca

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



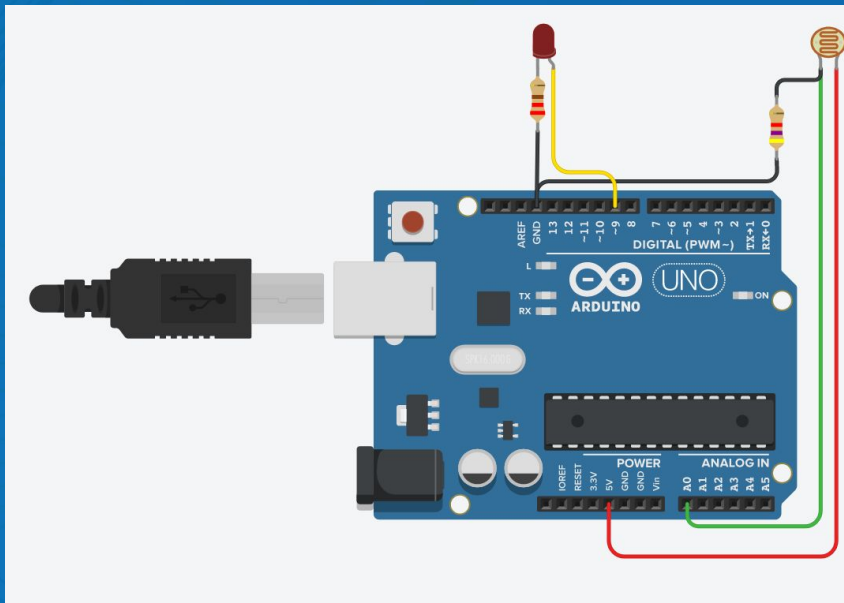


- [illegible]



Exemplos - Sensores

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

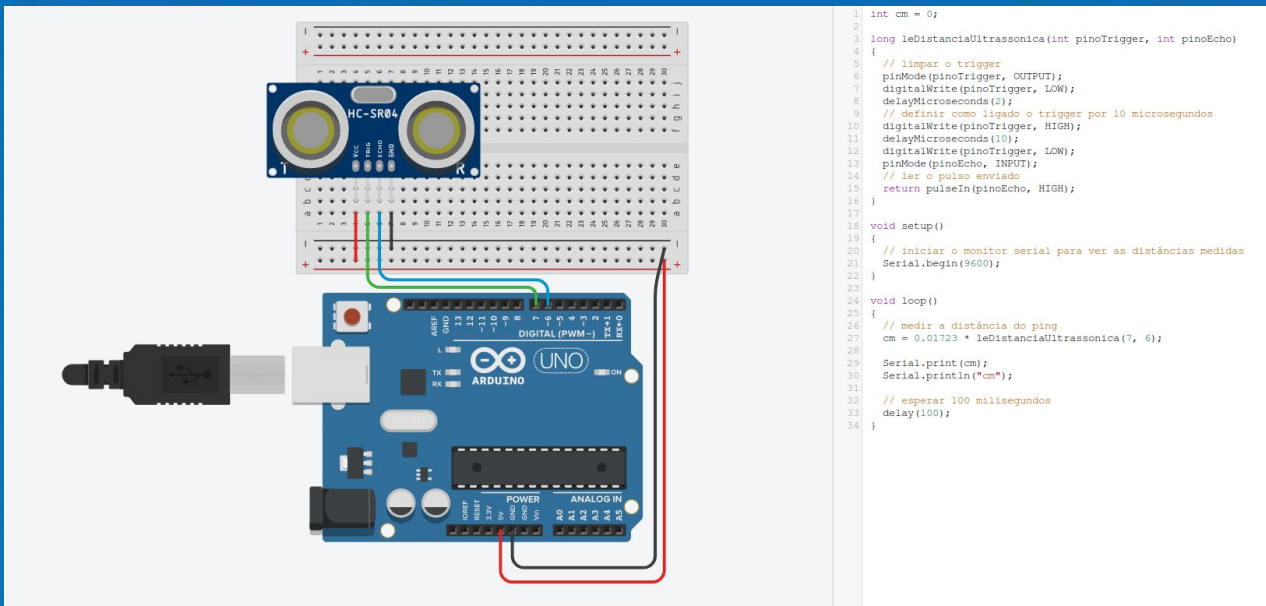


```
1 // variável para ler o valor do sensor de luz
2 int valorSensor = 0;
3
4 void setup()
5 {
6   // definir pino A0 como entrada
7   pinMode(A0, INPUT);
8   // iniciar monitor serial para ver o valor lido
9   Serial.begin(9600);
10
11   // definir pino do LED
12   pinMode(9, OUTPUT);
13 }
14
15 void 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

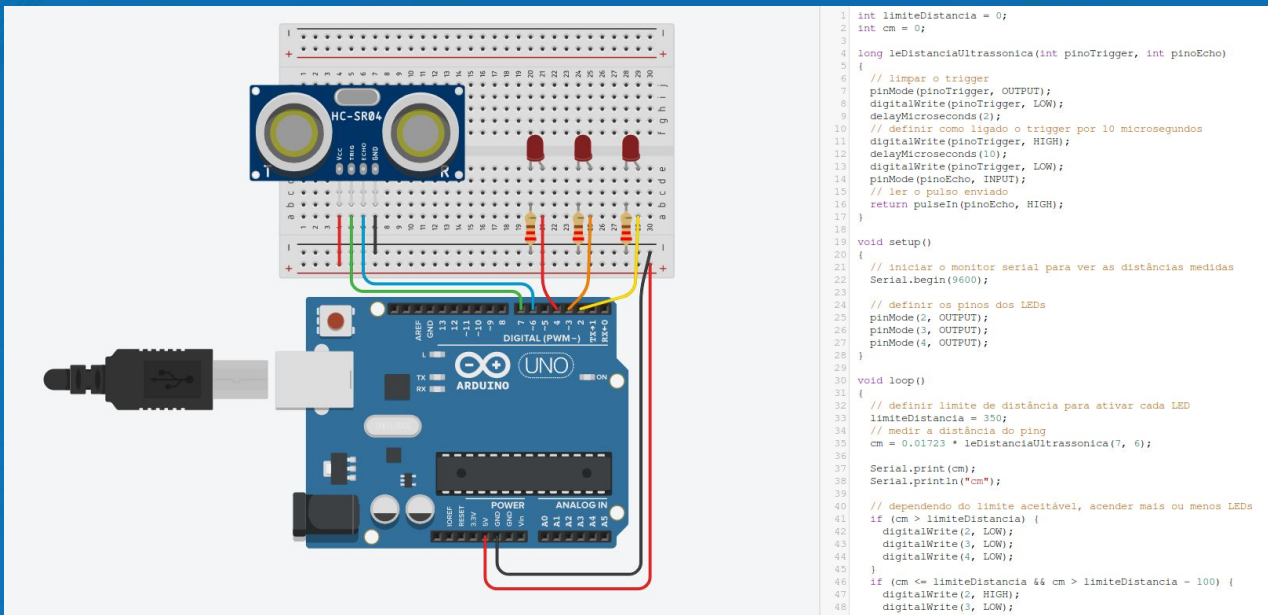
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada sensor
- <https://www.tinkercad.com/things/0Q20aht4wiF-sonar/>





Exemplos - Sensores

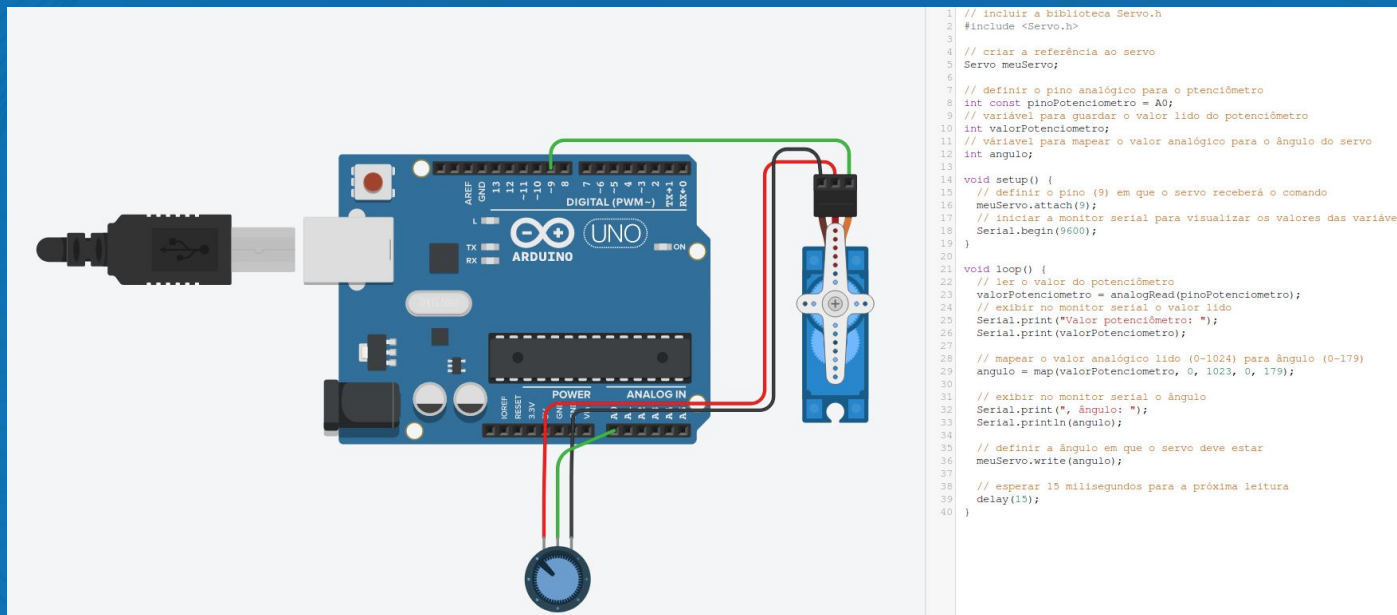
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada sensor
- <https://www.tinkercad.com/things/3fSIMyId4Cm-sonar-com-leds/>

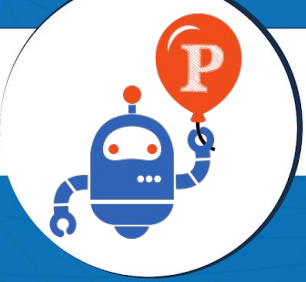




Exemplos - Atuadores

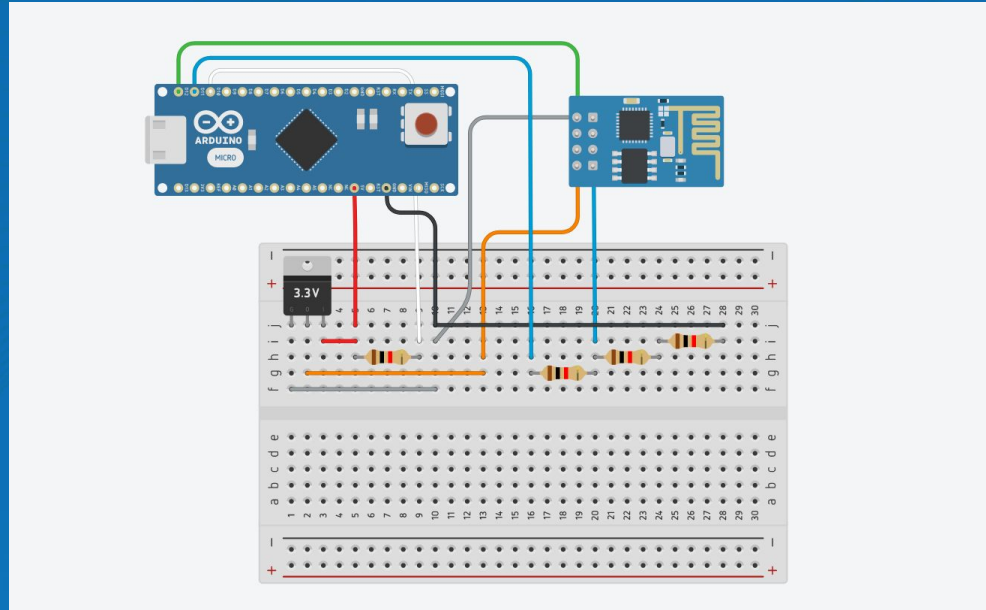
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de cada atuador
- <https://www.tinkercad.com/things/9CTUzW9i36K-servo/>

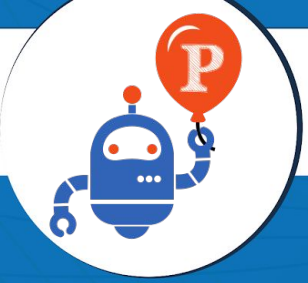




Exemplos - Comunicação

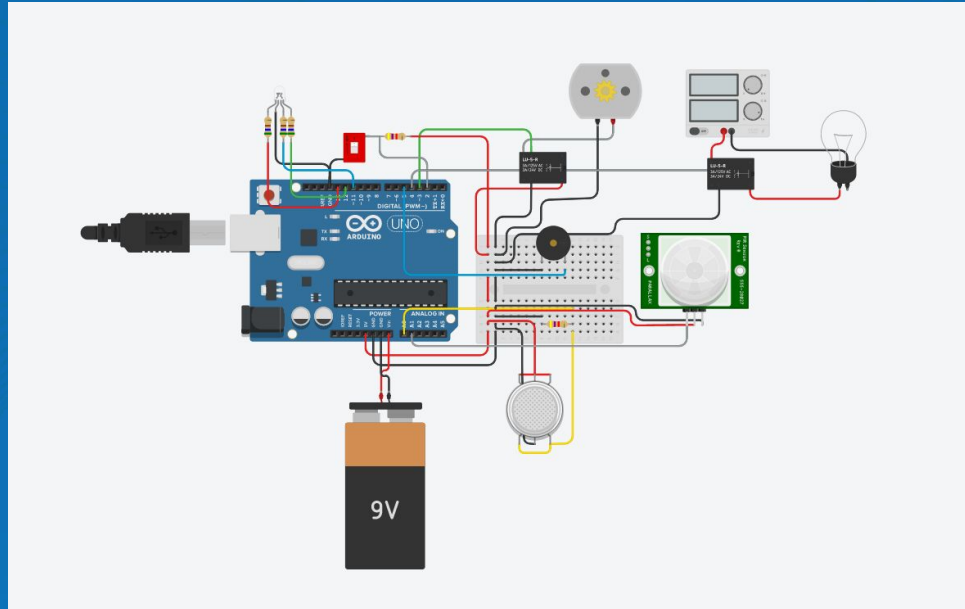
- Arduino IDE > Arquivo > Exemplos > Bibliotecas de comunicação
- <https://www.tinkercad.com/things/19iiXJ6NPzG-circuito-basico-1-kit-bt/>





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**

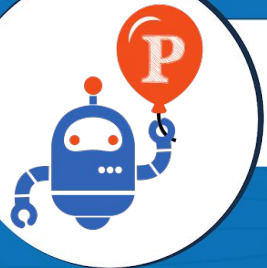
BOA SORTE!



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