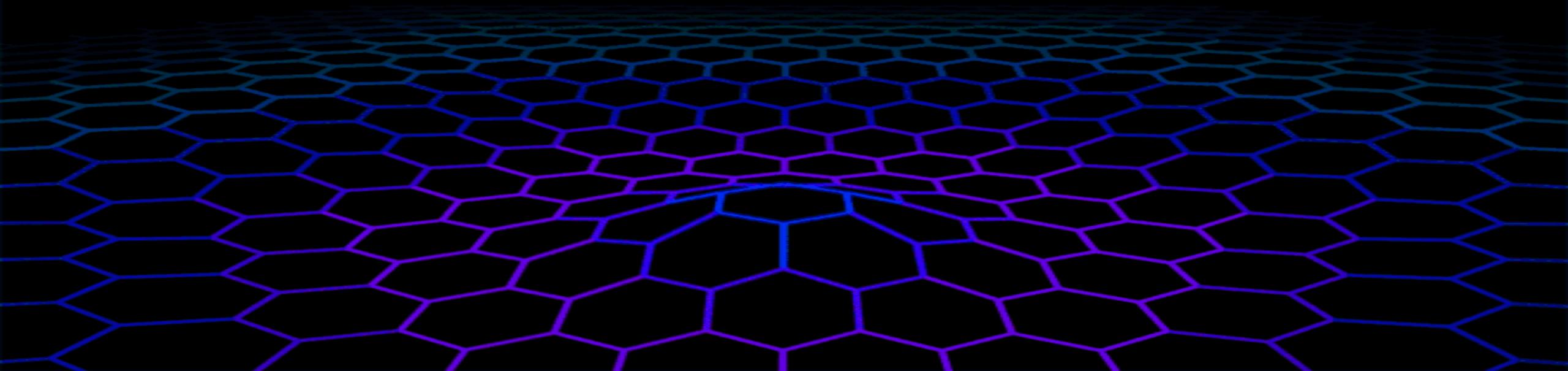
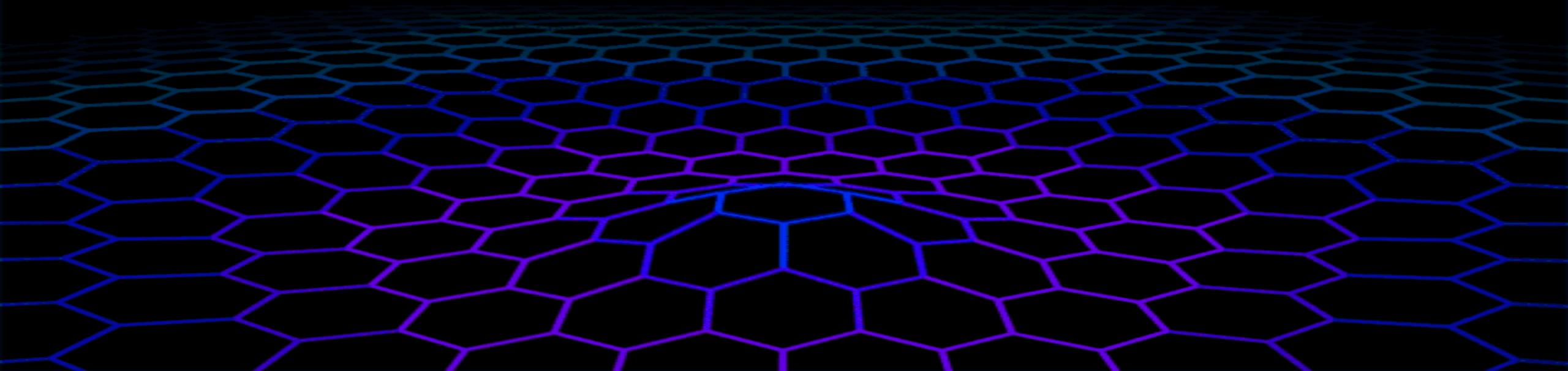


IOT

Prepare-se!



Prepare-se!



Bit

ENIDOS

# Quem é Gladimir?

Mais de 35 anos de experiência profissional

Mais de 25 anos lecionando em ensino superior

Mestre em Educação e Tecnologia pelo IFSul

Leciona em todos os cursos da Faculdade Senac

Prefere Internet das Coisas do que as Coisas da Internet.

Matou mais de 1.000 no Battlefield 2 (todos na faca)

É um cara sério, mas faz cosplay de Stormtrooper

# Adaptação à mudança?



# Adaptação à velocidade da mudança!



**50 milhões de usuários**





50 milhões de usuários

38 anos

**50 milhões de usuários**

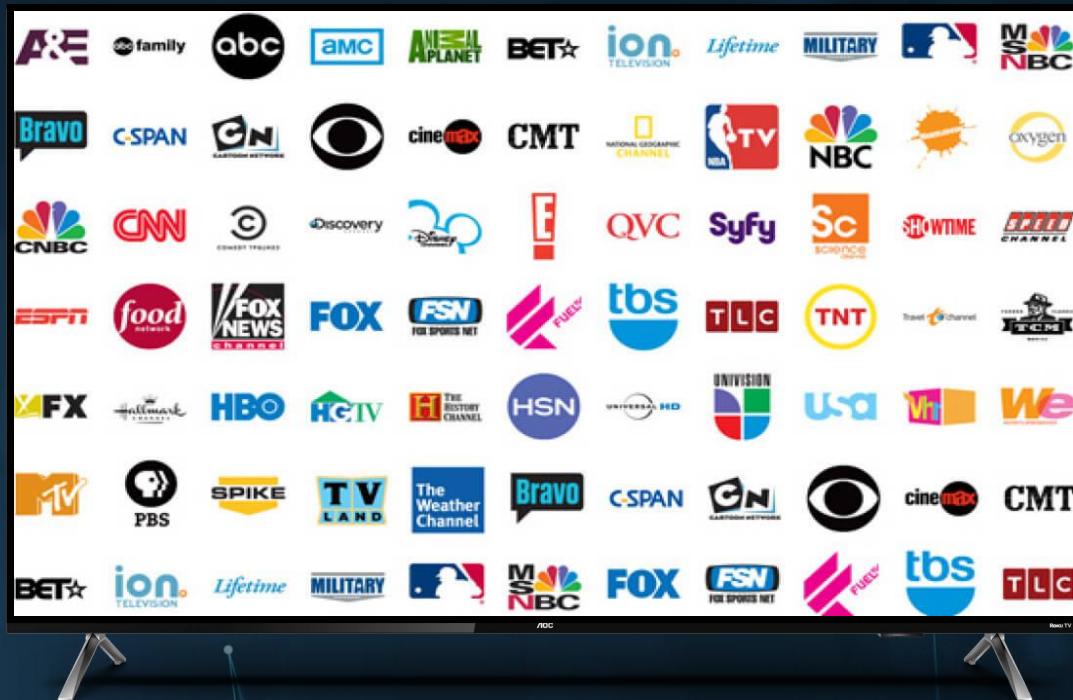


**50 milhões de usuários**



**16 anos**

# 50 milhões de usuários



50 milhões de usuários



10 anos

**50 milhões de usuários**



**50 milhões de usuários**

**5 anos**



**50 milhões de usuários**



**50 milhões de usuários**



**3 anos**

50 milhões de usuários



**50 milhões** de usuários

**2 anos**



50 milhões de usuários



**50 milhões de usuários**



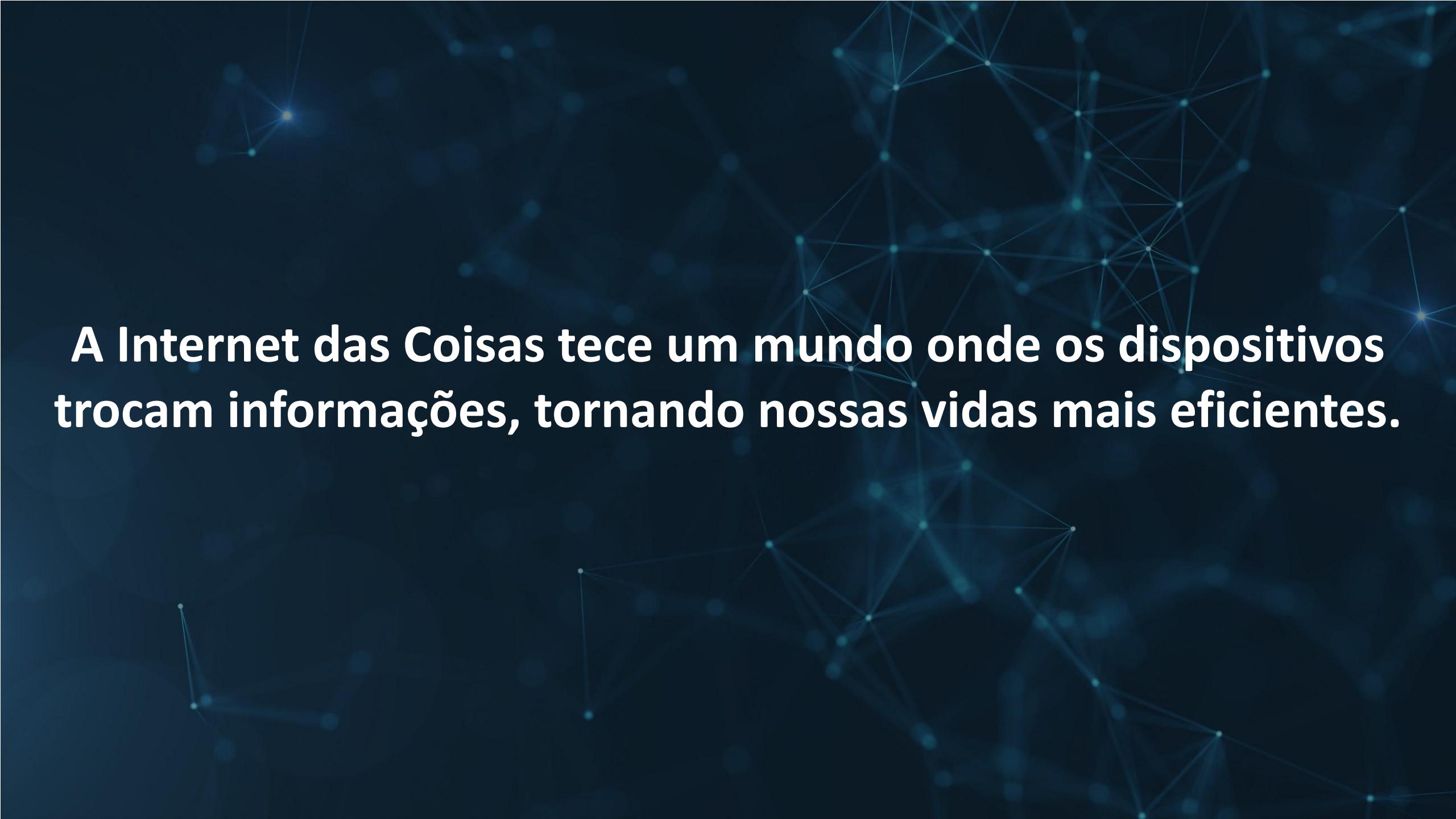
**19 dias**

# IoT - Internet of Things Internet das Coisas

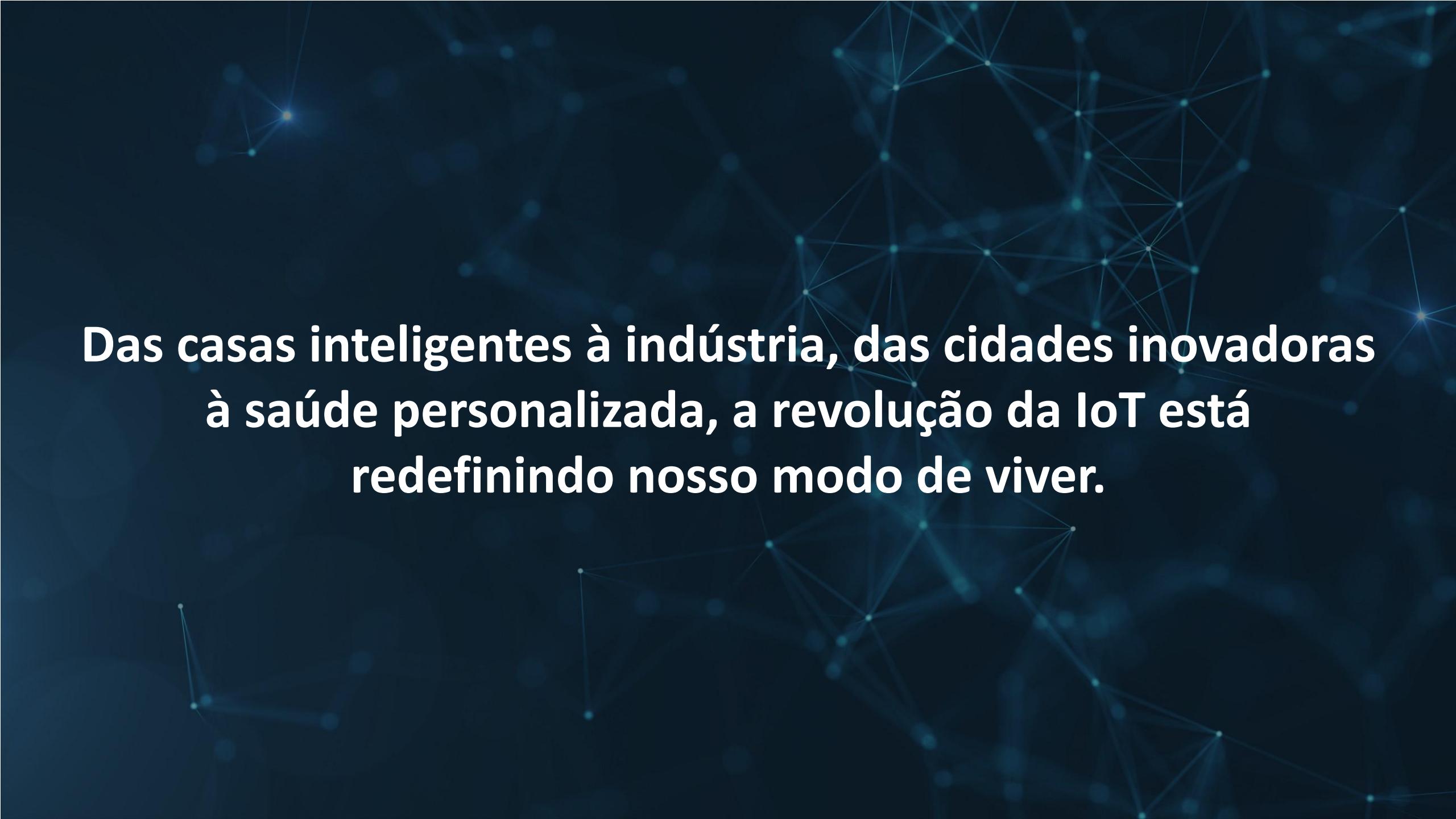
Nós produzimos mais transistores do que grãos de arroz e a um custo menor



O mundo agora produz mais transistores em um segundo que produziu em um ano em 1980

The background of the slide features a dark blue gradient with a subtle, glowing network of interconnected nodes and lines. This network is more dense in the upper right and lower right areas, while the lower left and center are more sparse, creating a sense of depth and connectivity.

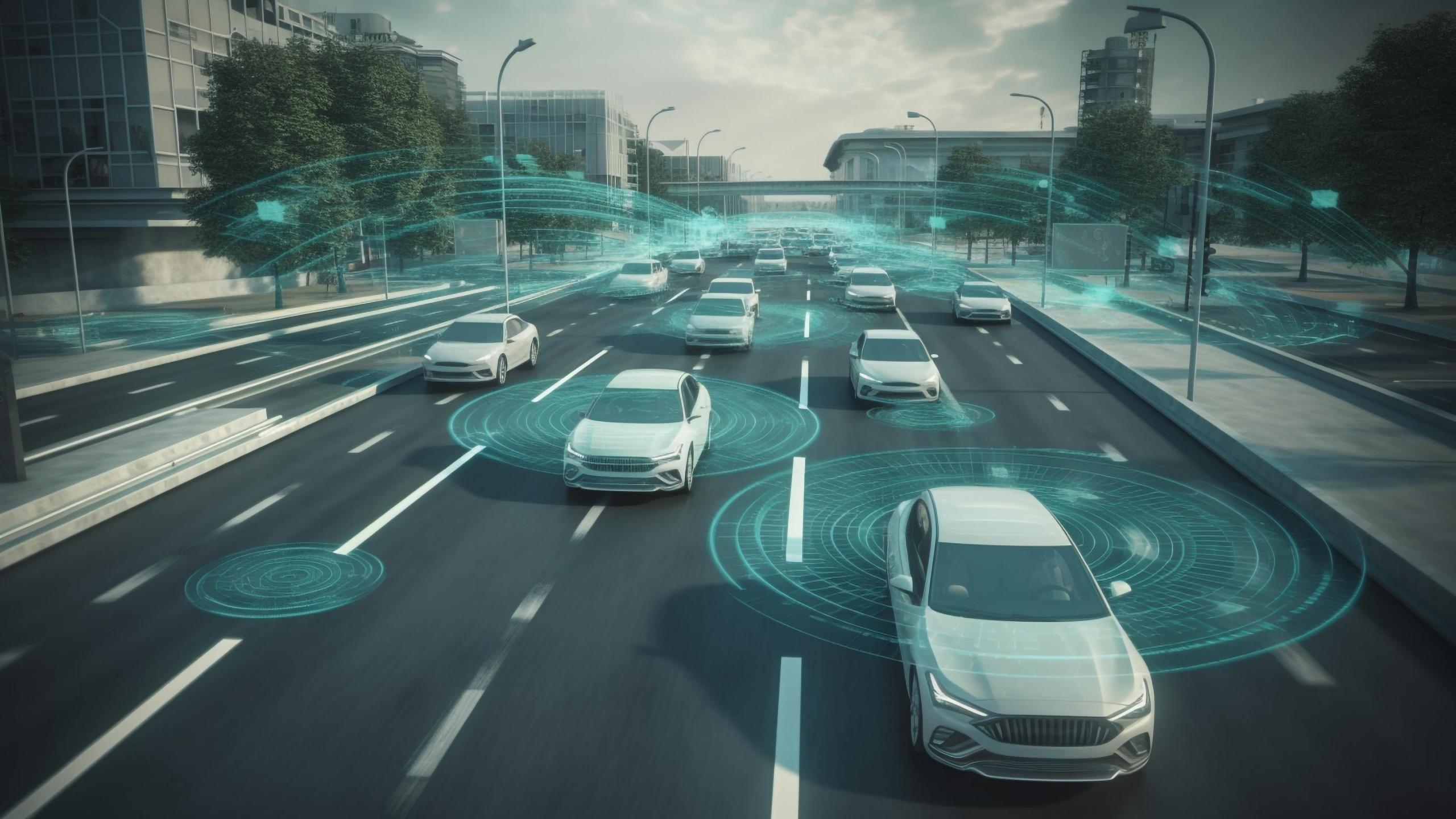
**A Internet das Coisas tece um mundo onde os dispositivos  
trocam informações, tornando nossas vidas mais eficientes.**

A dark blue background featuring a complex network of glowing blue and white dots connected by thin lines, resembling a molecular or neural network.

**Das casas inteligentes à indústria, das cidades inovadoras  
à saúde personalizada, a revolução da IoT está  
redefinindo nosso modo de viver.**









Long Range Wide Area Network



Quer acompanhar de perto?

Baixa o slide ;)



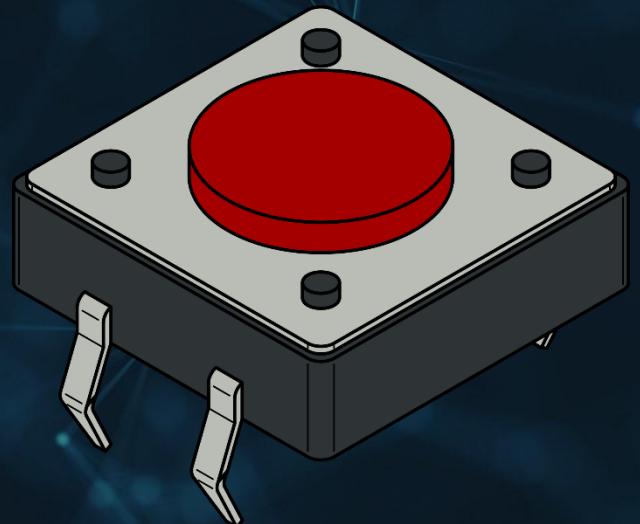
<http://bit.ly/3FCpl0K>

# Placas de desenvolvimento

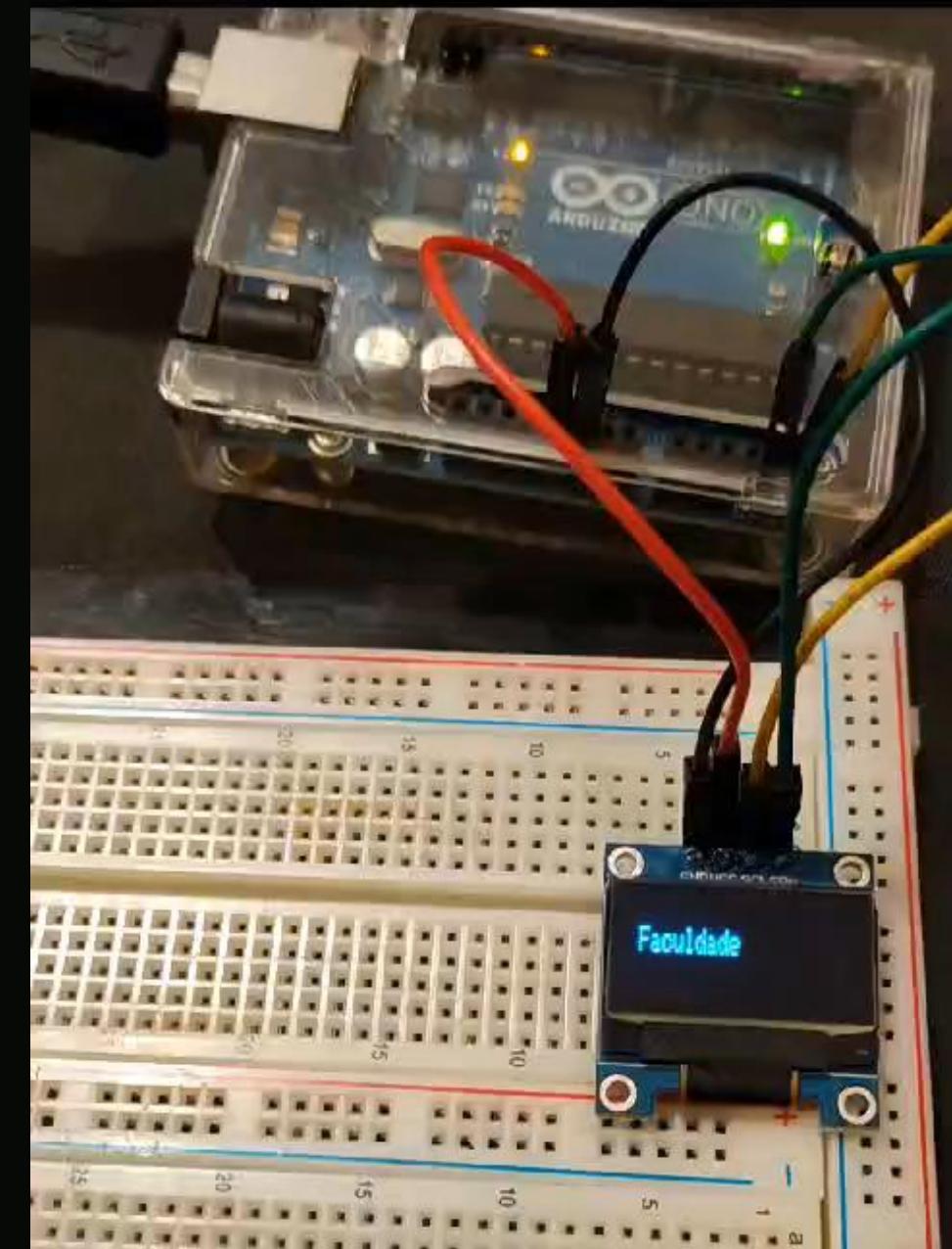
# Eletrônica básica



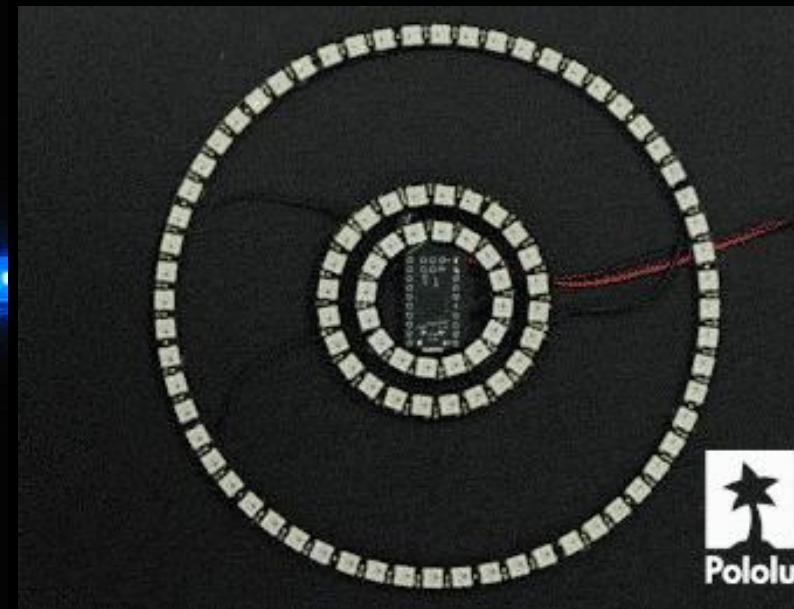
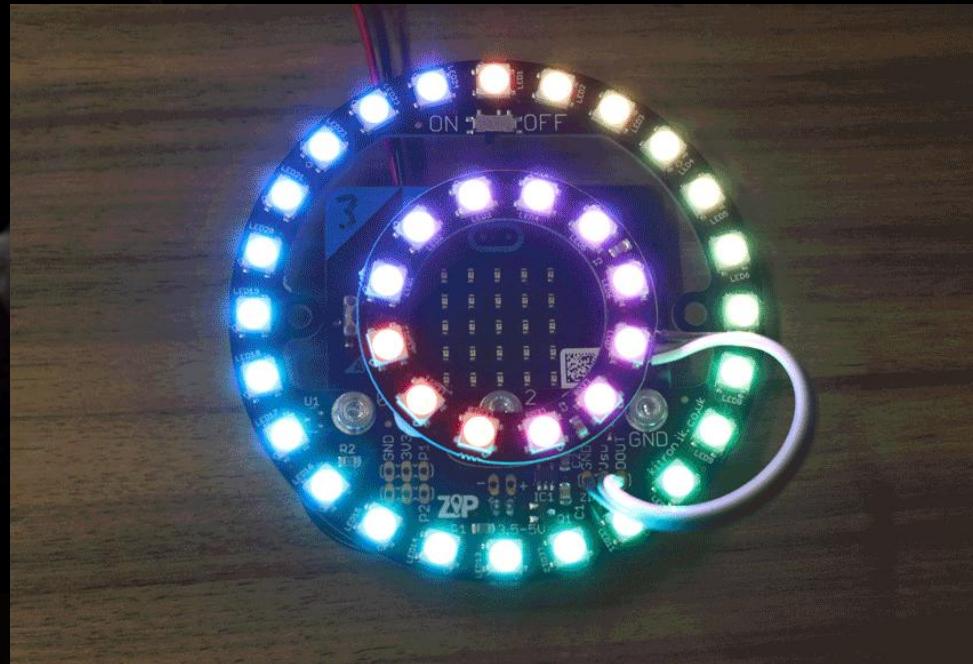
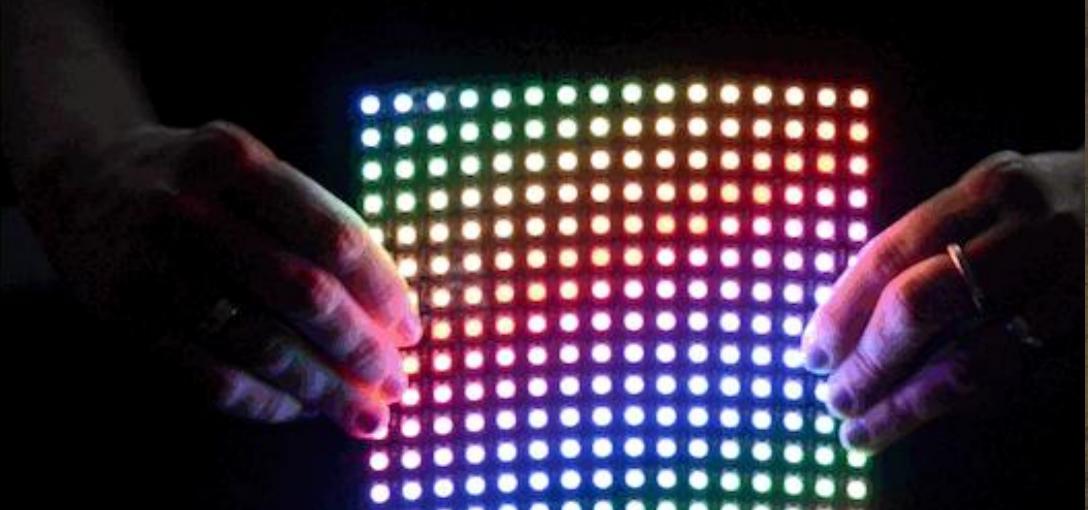
# Sensores

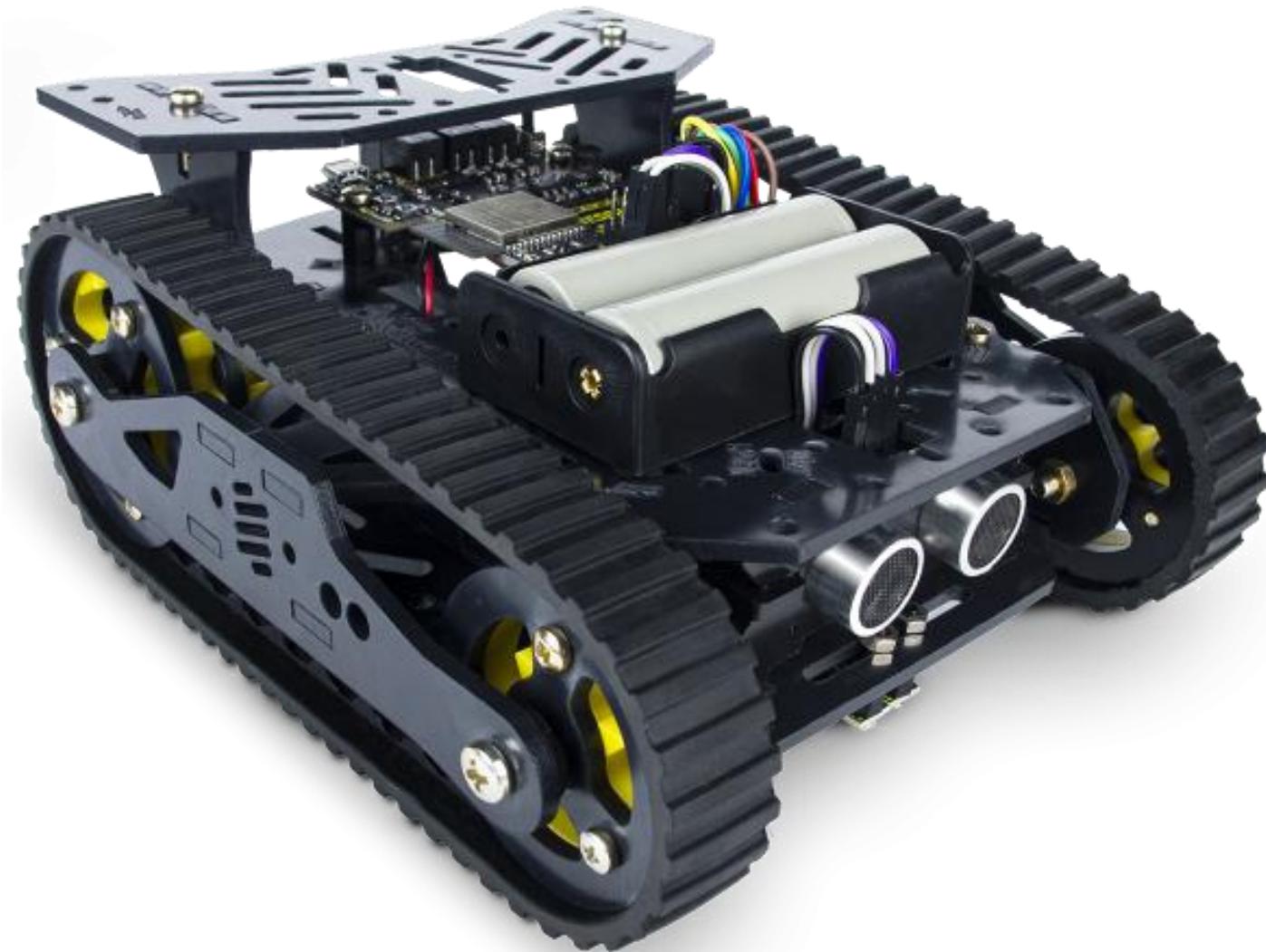


# Atuadores



# LED RGB ENDEREÇÁVEL

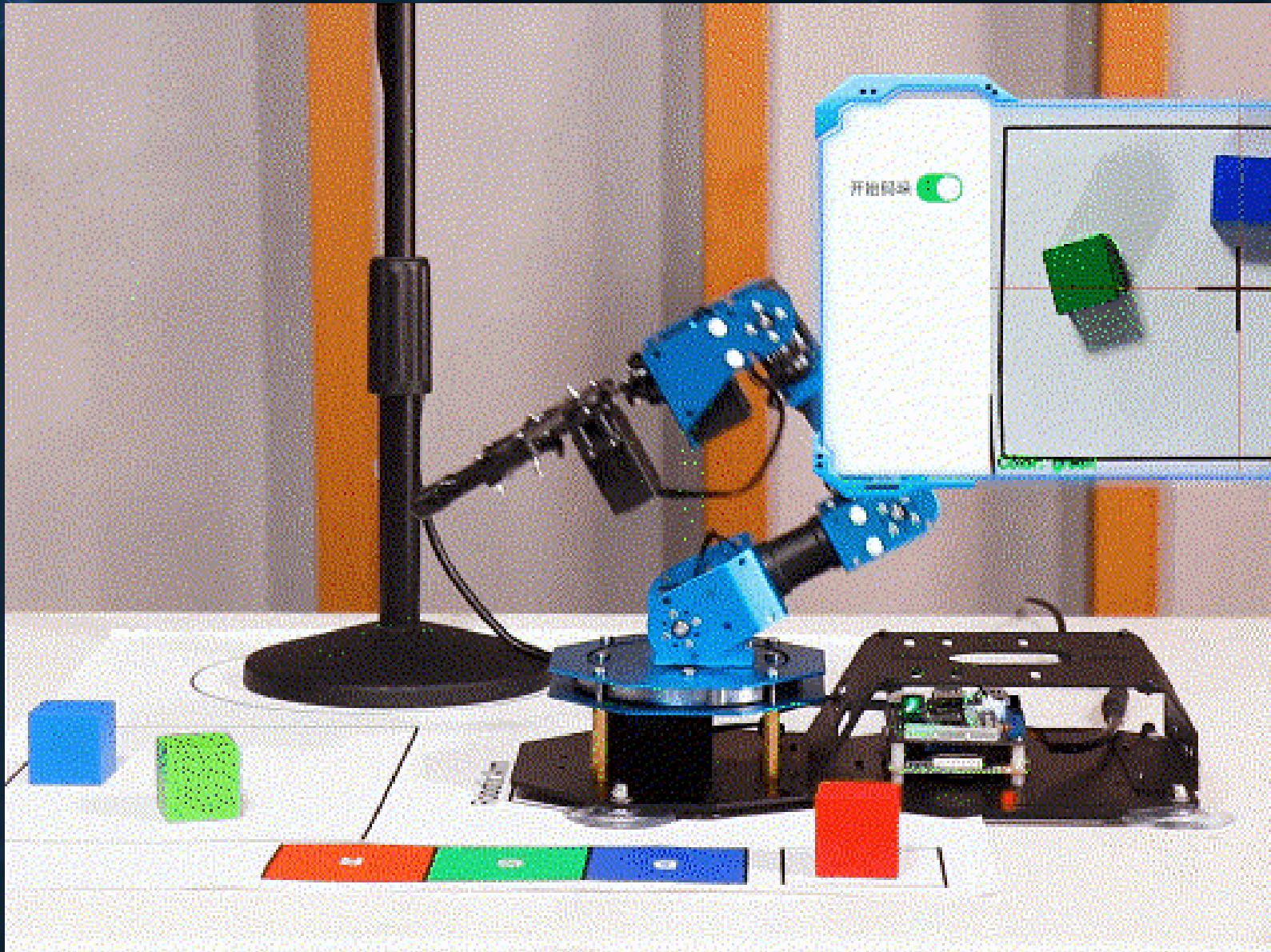








# Braço robótico







# Arduino - UNO



<b>Microcontroller</b>	<b>ATmega328</b>
<b>Operating Voltage</b>	<b>5V</b>
<b>Input Voltage (recommended)</b>	<b>7-12V</b>
<b>Input Voltage (limits)</b>	<b>6-20V</b>
<b>Digital I/O Pins</b>	<b>14 (of which 6 provide PWM output)</b>
<b>Analog Input Pins</b>	<b>6</b>
<b>DC Current per I/O Pin</b>	<b>40 mA</b>
<b>DC Current for 3.3V Pin</b>	<b>50 mA</b>
<b>Flash Memory</b>	<b>32 KB (ATmega328) of which 0.5 KB used by bootloader</b>
<b>SRAM</b>	<b>2 KB (ATmega328)</b>
<b>EEPROM</b>	<b>1 KB (ATmega328)</b>
<b>Clock Speed</b>	<b>16 MHz</b>

# Arduino – Alimentação

Tensão de funcionamento da placa: **5V**

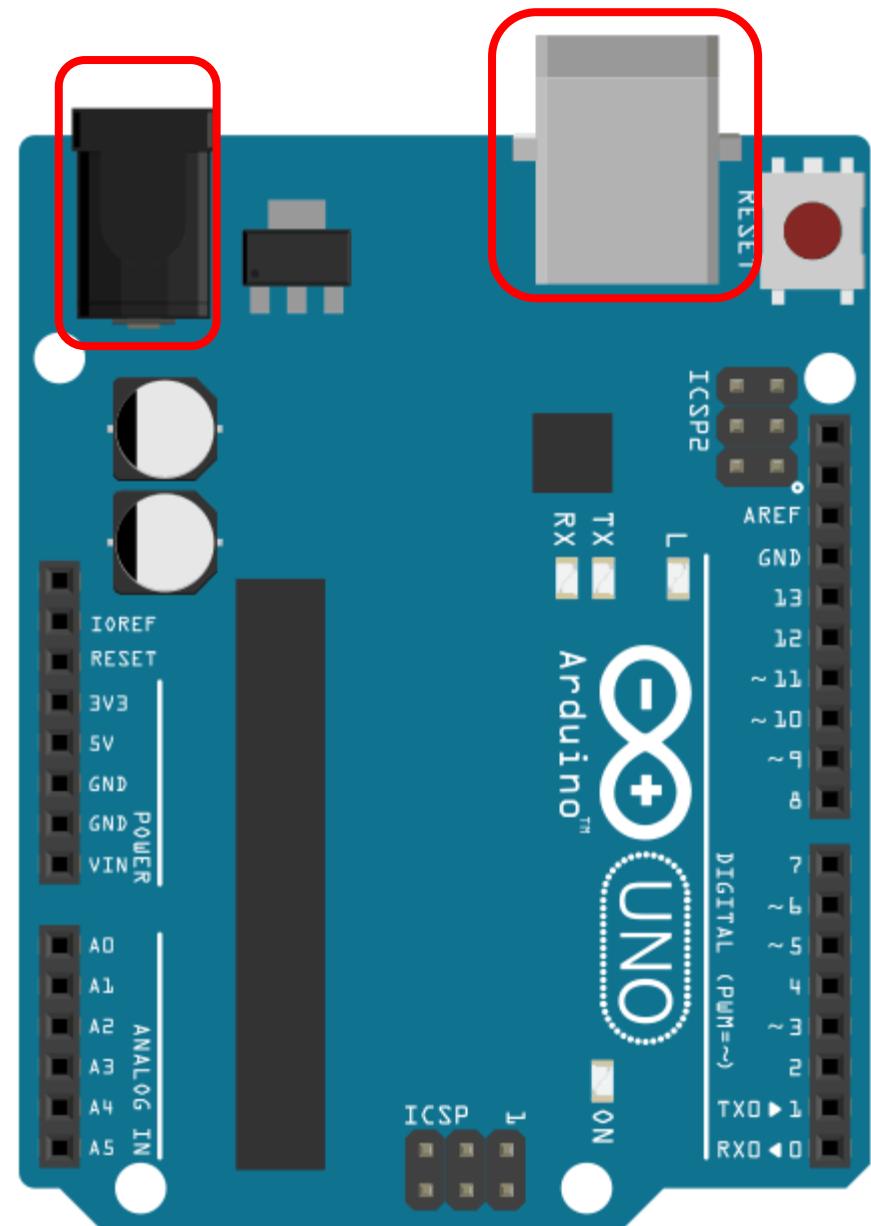
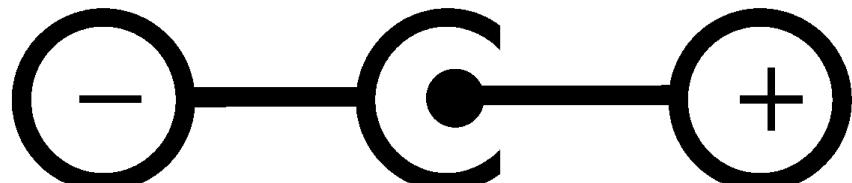
Alimentação feita através de conector Jack 6V a 20V

**Obs.:**

Alimentada abaixo de 7V pode ficar instável.

Alimentada acima de 12V, o regulador de tensão da placa pode sobreaquecer.

**Recomendado fonte externa entre **7V** e **12V****



# Arduino – Pinos de alimentação

**IREF** - Fornece uma tensão de referência para que *shields* possam selecionar o tipo de interface apropriada, dessa forma *shields* que funcionam com a placas Arduino que são alimentadas com 3,3V. podem se adaptar para ser utilizados em 5V e vice-versa.

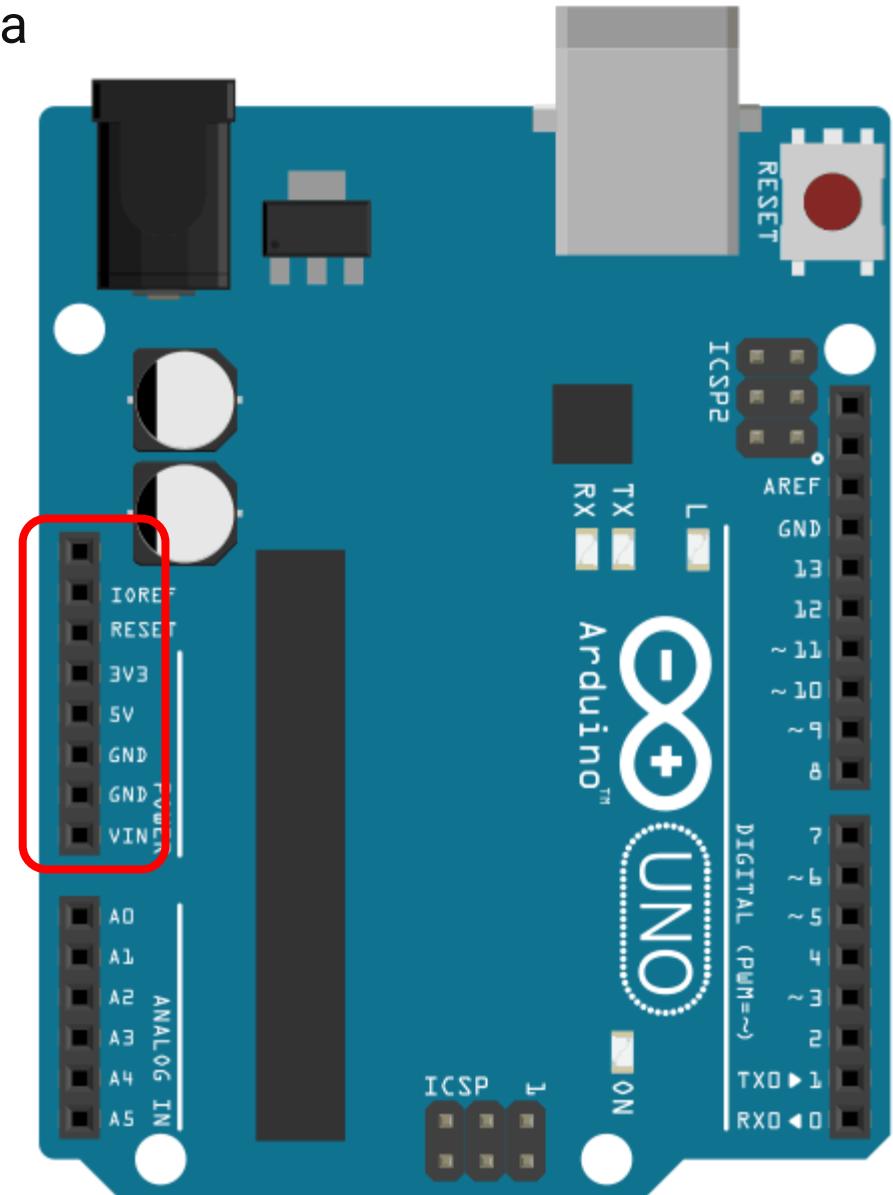
**RESET** - Pode ser utilizado para um reset externo da placa Arduino.

**3,3V** - Fornece tensão de 3,3V. Corrente máxima de 50 mA.

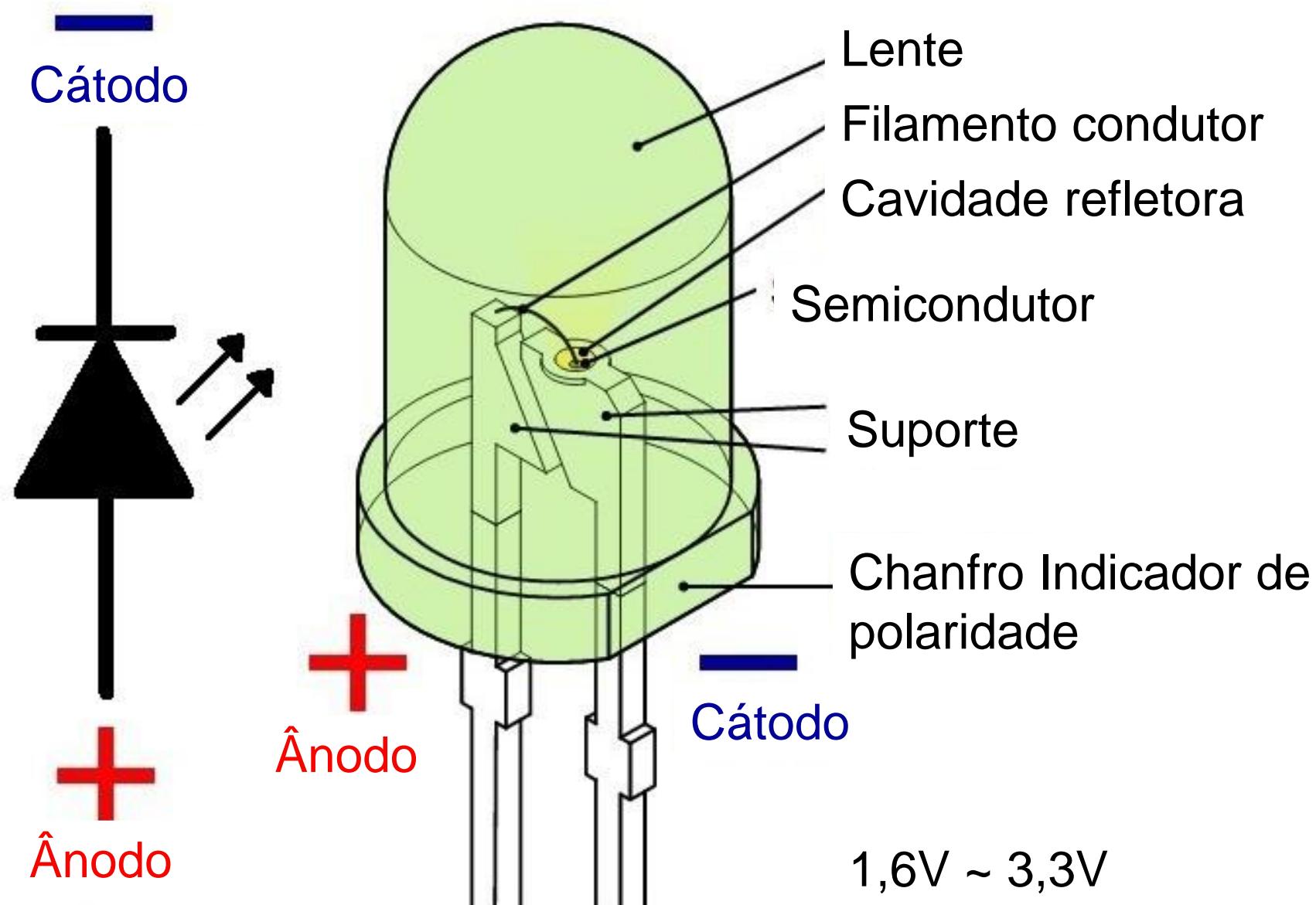
**5V** - Fornece tensão de 5 V para alimentação de *Shields* e circuitos externos.

**GND** - pinos de referência, terra.

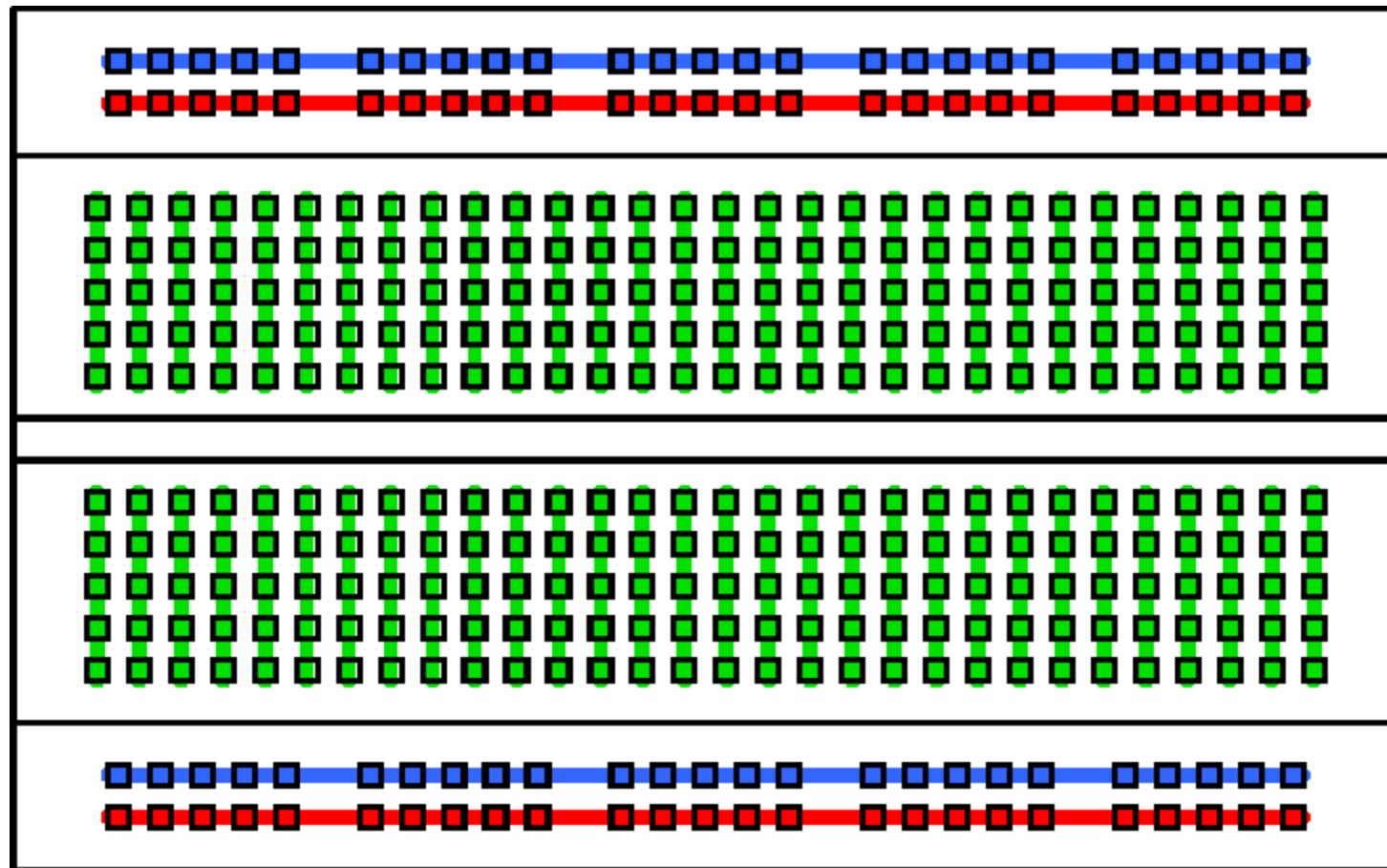
**VIN** - pino para alimentar a placa através de shield ou bateria externa. Quando a placa é alimentada através do conector Jack, a tensão da fonte estará nesse pino.

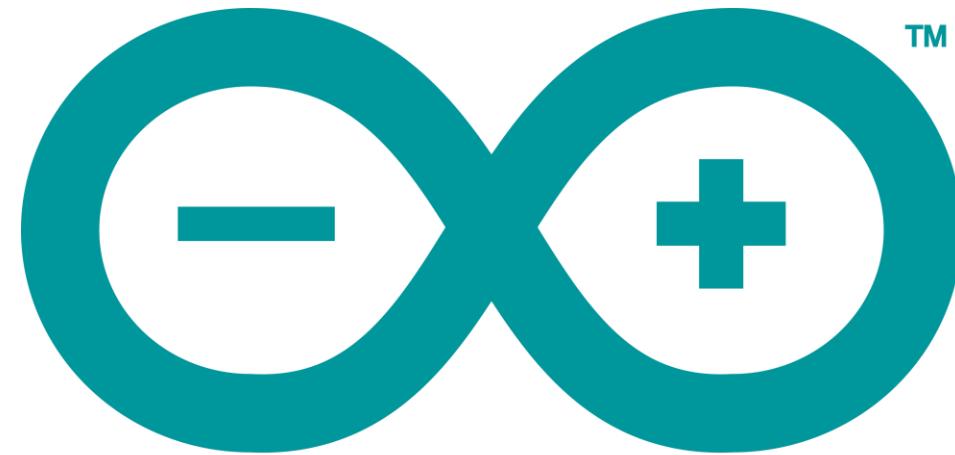


# LED - Light Emitting Diode (Diodo Emissor de Luz)



# Protoboard





# ARDUINO

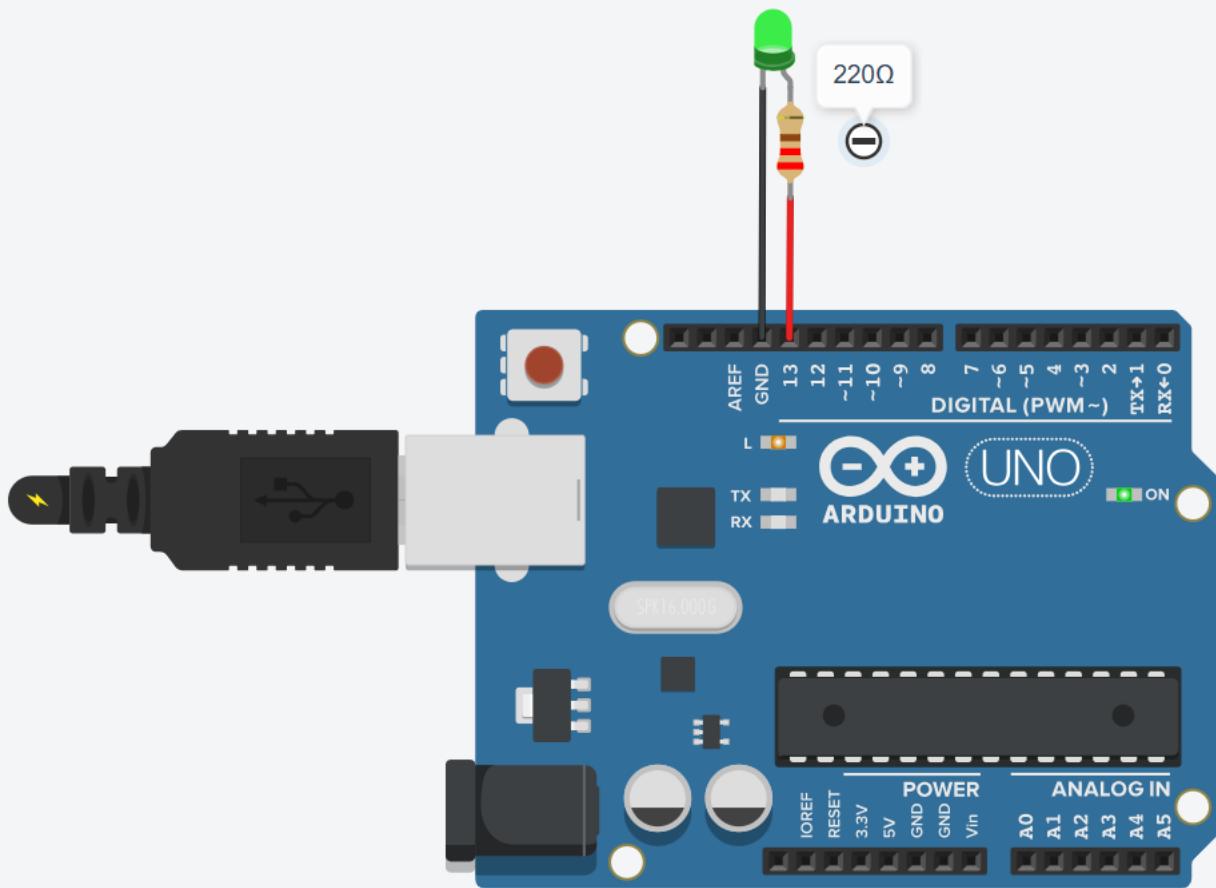
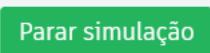
&

TIN  
KER  
CAD

AUTODESK®  
TINKERCAD®



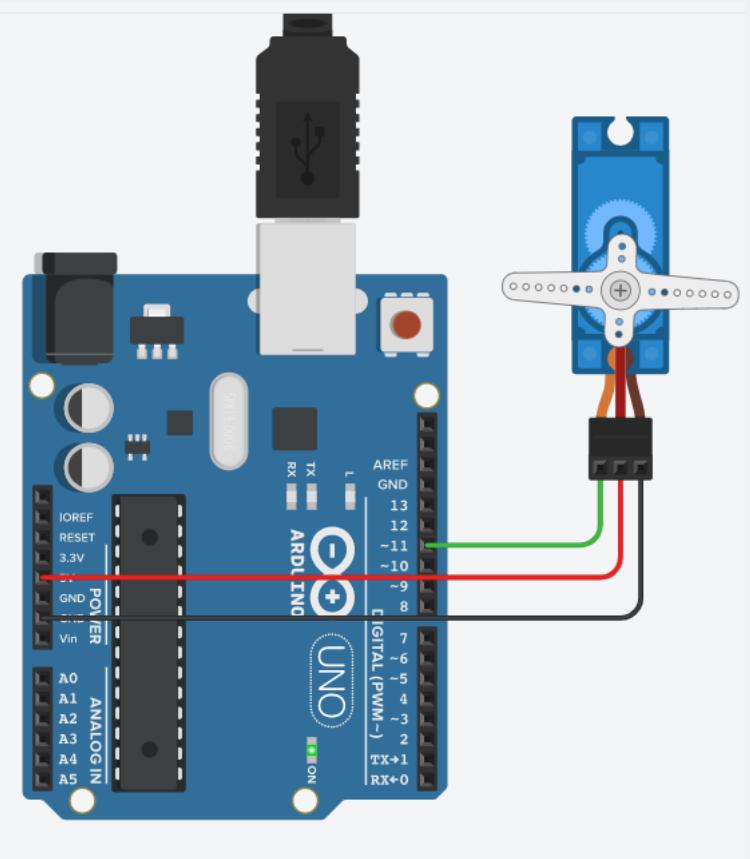
Hora do simulador: 00:00:01



1 (Arduino Uno R3)

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, 1);  
    delay(2000);  
    digitalWrite(13, 0);  
    delay(1000);  
}
```

# Arduino + Servo (Tinkercad)



```
#include <Servo.h>

int posicao = 0;
int incremento = 5; // valor do incremento
Servo servo1;

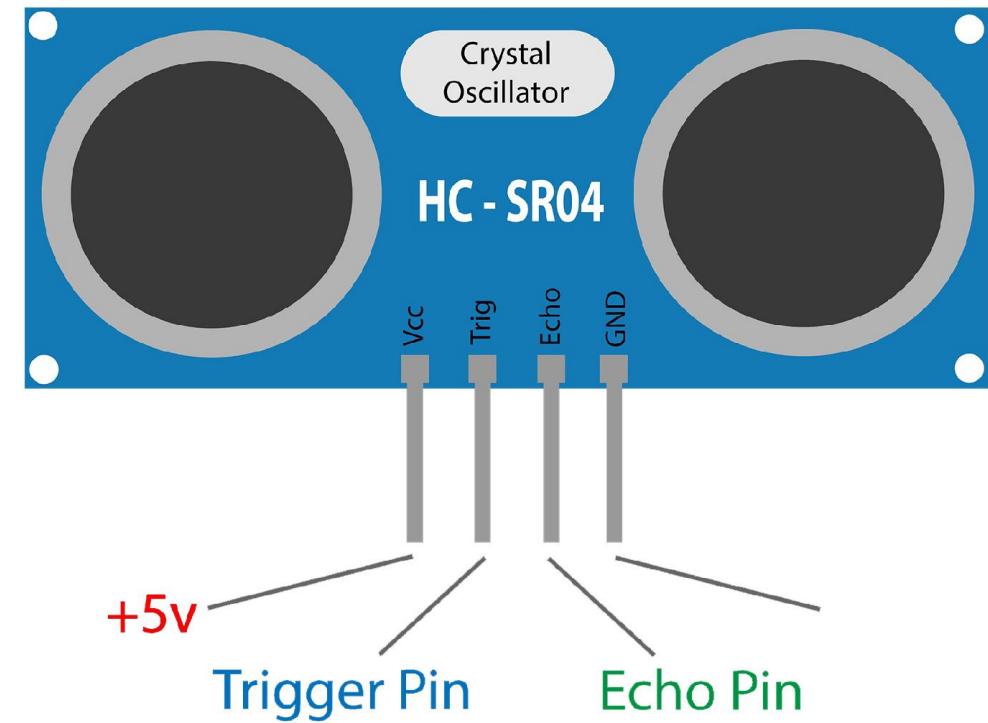
void setup()
{
    servo1.attach(11);
    Serial.begin(9600);
}

void loop()
{
    servo1.write(posicao);
    Serial.println(posicao);
    delay(500);

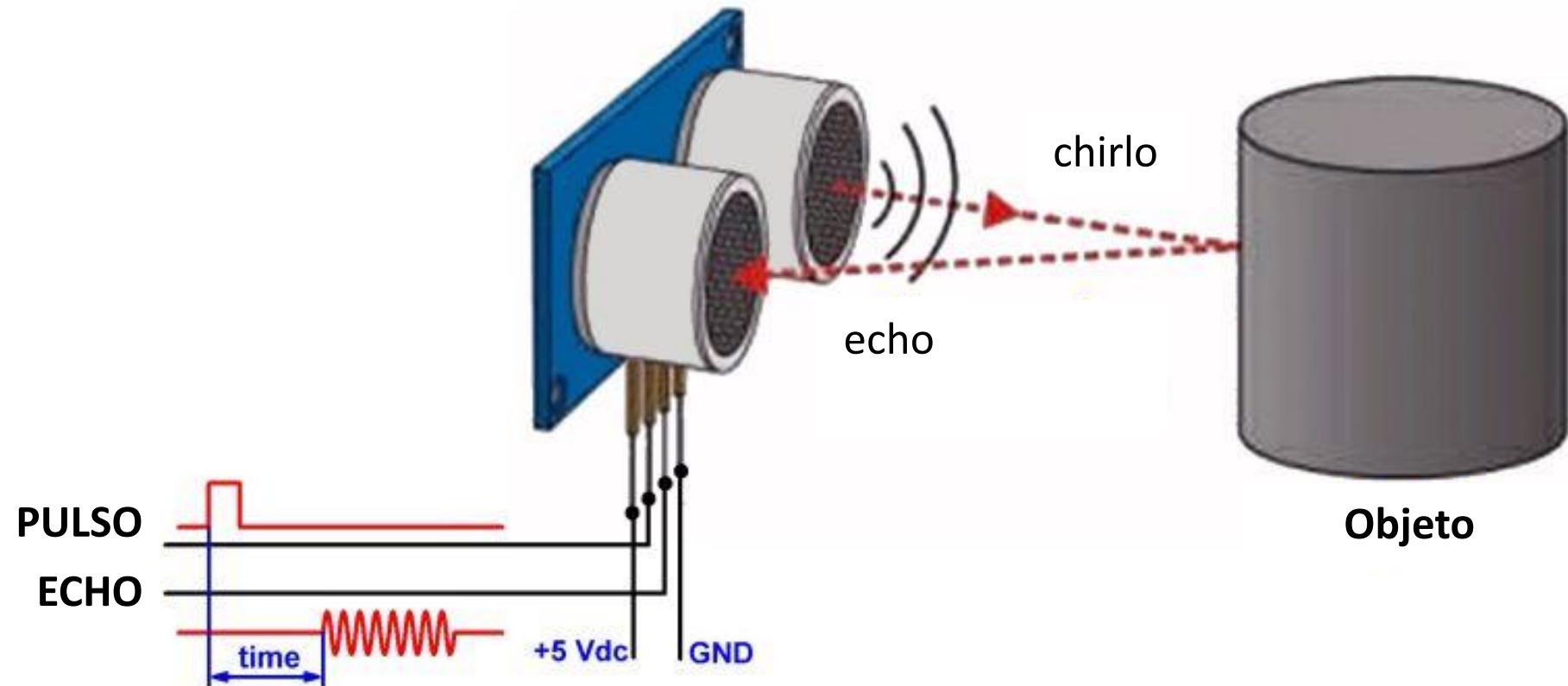
    posicao = posicao + incremento;

    if (posicao >= 180 || posicao <= 0) {
        incremento = -incremento; // inverte o sentido
    }
}
```

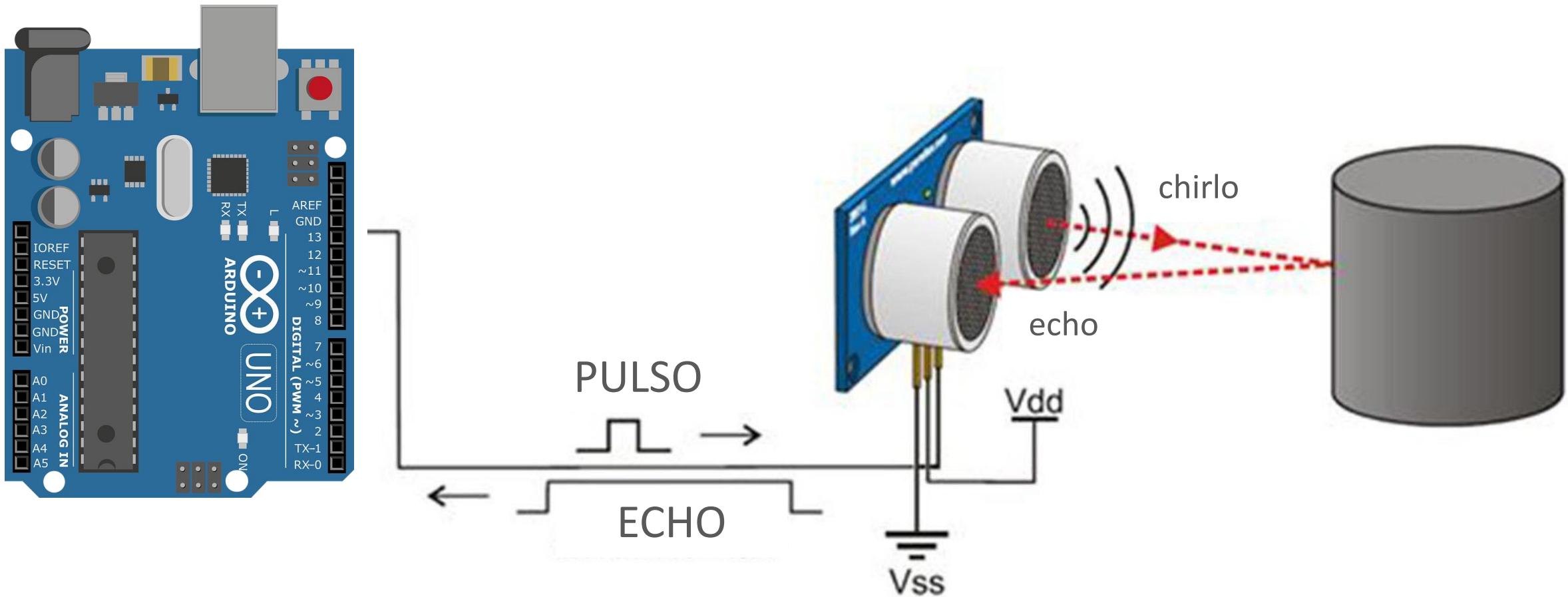
# Sensor Ultrassônico HC-SR04

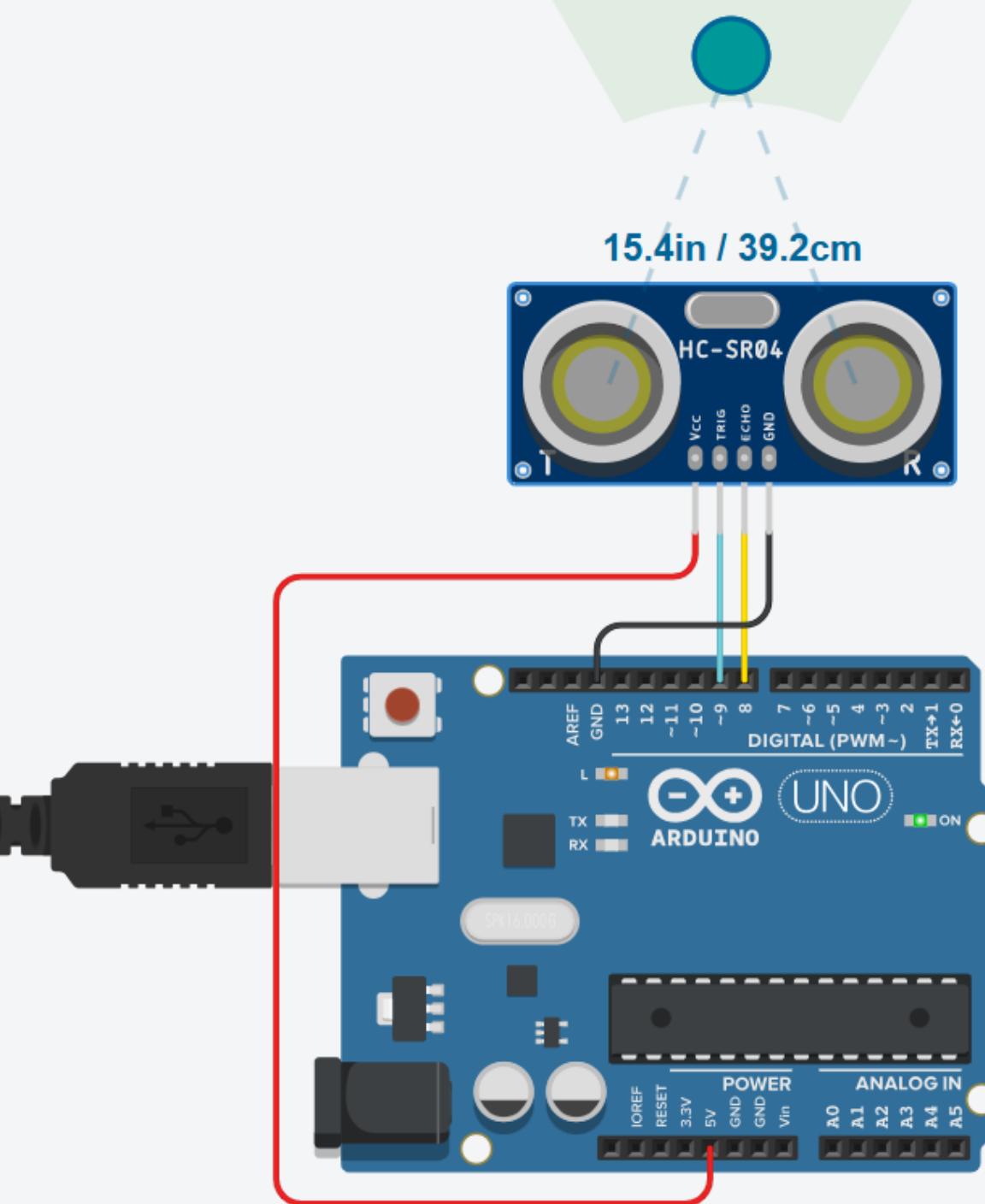


# Sensor Ultrassônico HC-SR04



# Sensor Ultrassônico 28015





```
// Pinos usados no Arduino
const int pinoTrigger = 9;
const int pinoEcho    = 8;
const int ledInterno = 13; // LED interno

// Variáveis para armazenar duração e distância
long duracao;
float distancia;

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

  // Configuração dos pinos
  pinMode(pinoTrigger, OUTPUT);
  pinMode(pinoEcho, INPUT);
  pinMode(ledInterno, OUTPUT);
}

void loop() {
  // Gera pulso ultrassônico
  digitalWrite(pinoTrigger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinoTrigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinoTrigger, LOW);

  // ... Código completo no próximo slide
```

```
// Pinos usados no Arduino
const int pinoTrigger = 9;
const int pinoEcho    = 8;
const int ledInterno  = 13; // LED interno

// Variáveis para armazenar duração e distância
long duracao;
float distancia;

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

  // Configuração dos pinos
  pinMode(pinoTrigger, OUTPUT);
  pinMode(pinoEcho, INPUT);
  pinMode(ledInterno, OUTPUT);
}

void loop() {
  // Gera pulso ultrassônico
  digitalWrite(pinoTrigger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinoTrigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinoTrigger, LOW);

  // Mede o tempo do eco de retorno
  duracao = pulseIn(pinoEcho, HIGH);

  // ... Continua ao lado =>
```

```
// ... ← Continuação

// Converte duração em distância (em centímetros)
distancia = (duracao / 2.0) / 29.1;

// Verifica se a leitura
// está dentro do alcance do sensor
if (distancia >= 300 || distancia <= 0) {
  Serial.println("Fora do alcance");
} else {
  Serial.print(distancia);
  Serial.println(" cm");
}

// Aciona o LED conforme proximidade
if (distancia < 40) {
  digitalWrite(ledInterno, HIGH);
} else {
  digitalWrite(ledInterno, LOW);
}

delay(300); // Intervalo entre leituras
}
```

# Servo Motor

# Servo Motor



# Servo Motor

Os Servo motores não pertencerem à uma classe específica de motores, ou seja, eles podem ser tanto motores CA quanto motores CC.

Os servos são atuadores projetados para aplicações onde é necessário fazer o controle de movimento com posicionamento de alta precisão, reversão rápida e de alto desempenho.

Eles são amplamente usados em robótica, sistemas automatizados, máquinas CNC e em outras diversas aplicações.

# Servo Motor

Os servos apresentam várias diferenças em relação aos demais tipos de motores.

Os servos têm incorporado um *encoder* e um controlador.

O *encoder* é um sensor de velocidade que possui a função de fornecer a velocidade e posicionamento do motor.

Para controlar a velocidade e a posição final do motor, o Servo trabalha com servomecanismo que usa feedback (realimentação) de posição.

Um Servo motor combina internamente um motor com circuito de realimentação, um controlador e outros circuitos complementares.

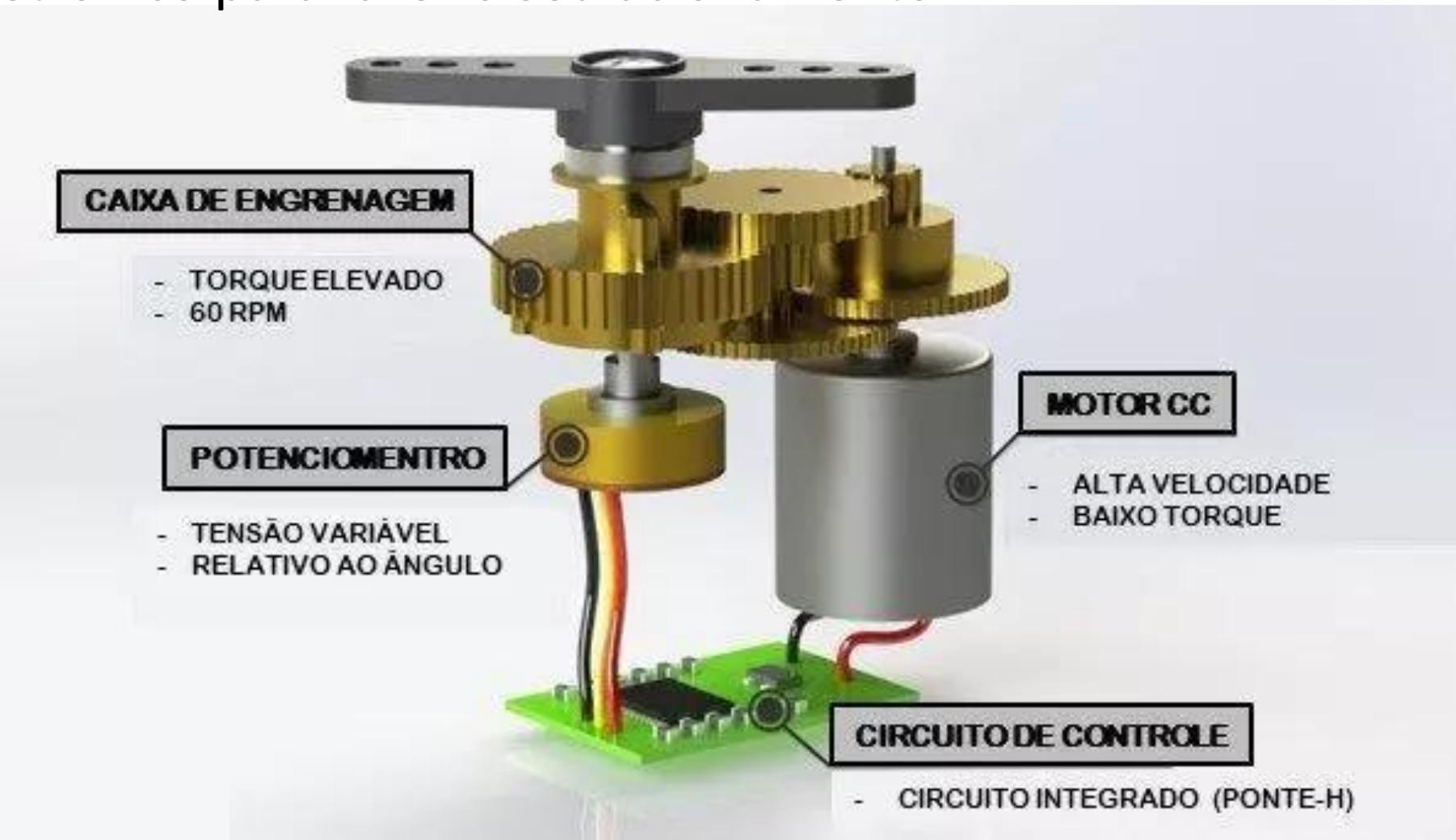
# Servo Motor - Tipos

Os Servo motores CC (de corrente contínua) geralmente são usados em projetos menores, eles possuem um custo relativamente baixo e são eficientes.

Os servo motores CA (de corrente alternada) normalmente são usados em ambientes industriais, pois costuma ser de elevada potência, oferece maior exatidão no seu controle e pouca manutenção.

# Servo Motor - CC

Um Servo motor CC é formado basicamente por um pequeno motor de corrente contínua, um potenciômetro de realimentação, uma caixa de engrenagens e um circuito eletrônico para fazer o seu acionamento.



# Servo SM-S2309S

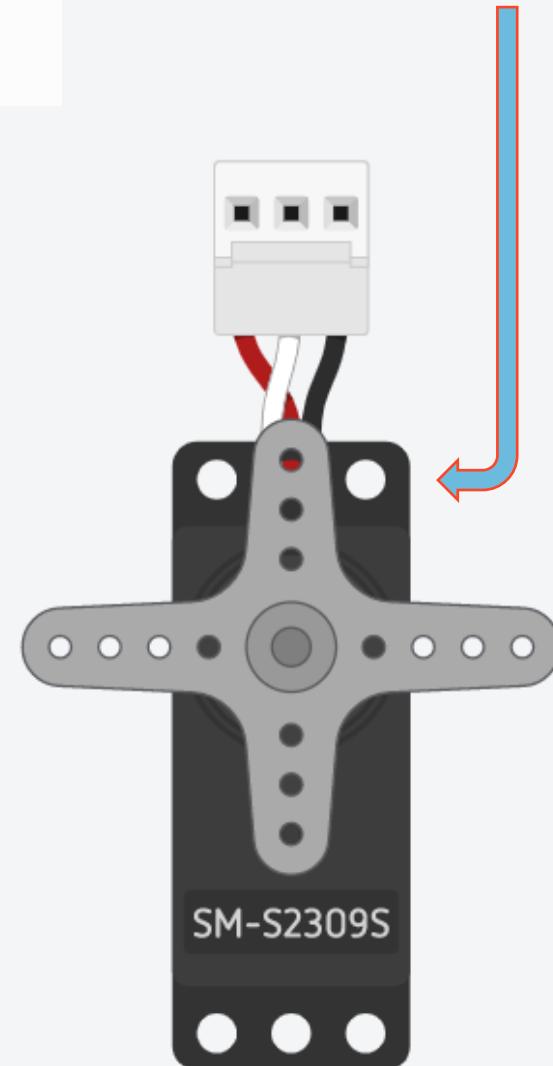


## SM-S2309S

Size	22.2×11.6×21.5 mm	
Weight	9.0 g	0.32 oz
4.8Volts	speed	0.12 s/60°
	Torque	1.0 kg.cm 13.9 oz.in
6.0Volts	speed	0.10 s/60°
	Torque	1.2 kg.cm 16.7 oz.in

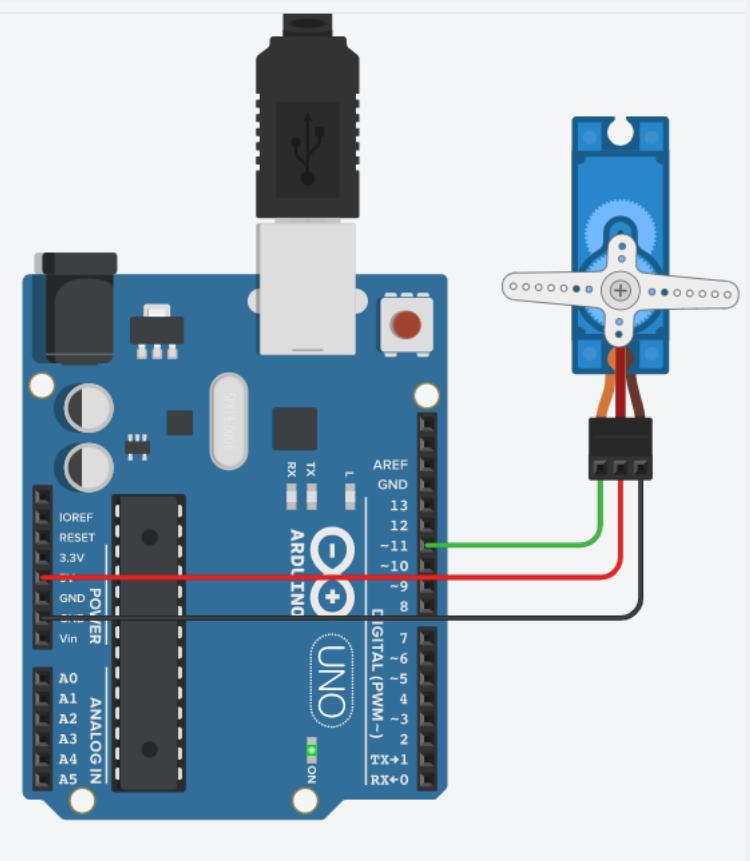
Micro analog servo,plastic gear

TIN  
KER  
CAD



# Exemplos

# Arduino + Servo (Tinkercad)



```
#include <Servo.h>

int posicao = 0;
int incremento = 5; // valor do incremento
Servo servo1;

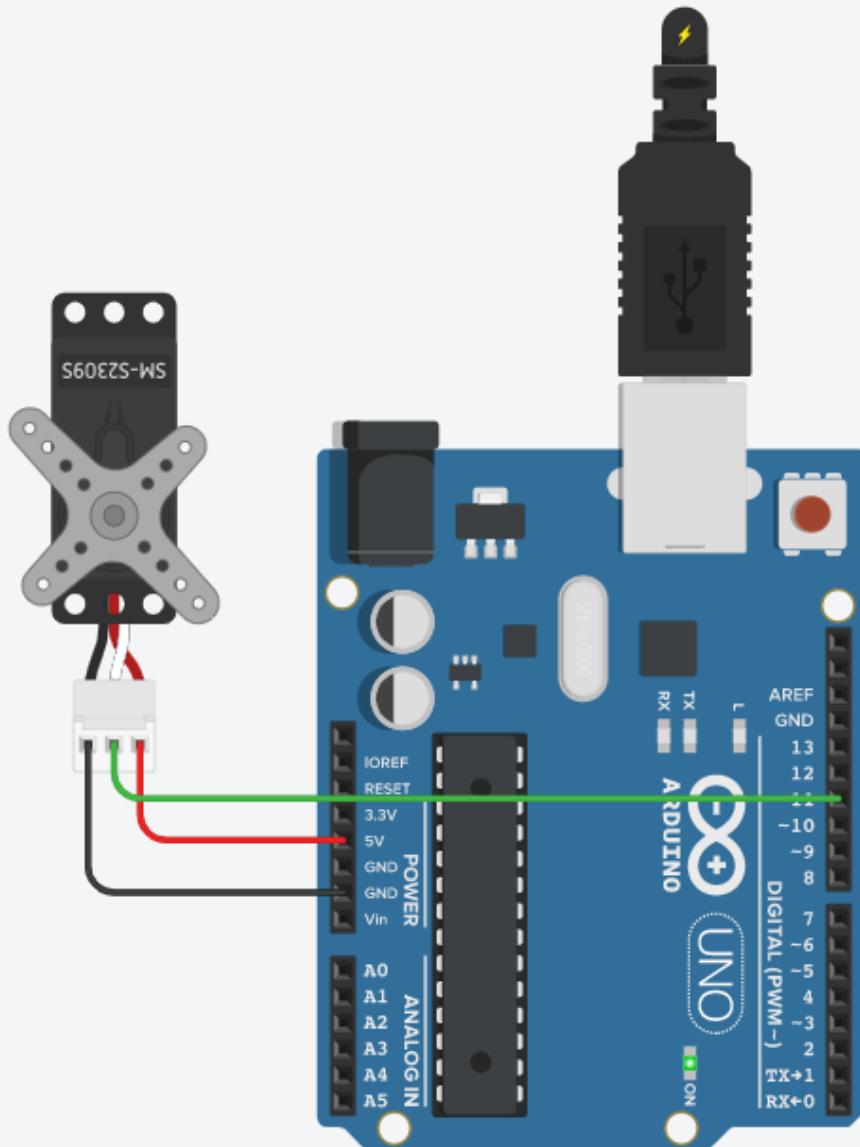
void setup()
{
    servo1.attach(11);
    Serial.begin(9600);
}

void loop()
{
    servo1.write(posicao);
    Serial.println(posicao);
    delay(500);

    posicao = posicao + incremento;

    if (posicao >= 180 || posicao <= 0) {
        incremento = -incremento; // inverte o sentido
    }
}
```

# Arduino + Servo – leitura (Tinkercad)



```
#include <Servo.h>

Servo servo1;
int posicao = 0;
int incremento = 45;

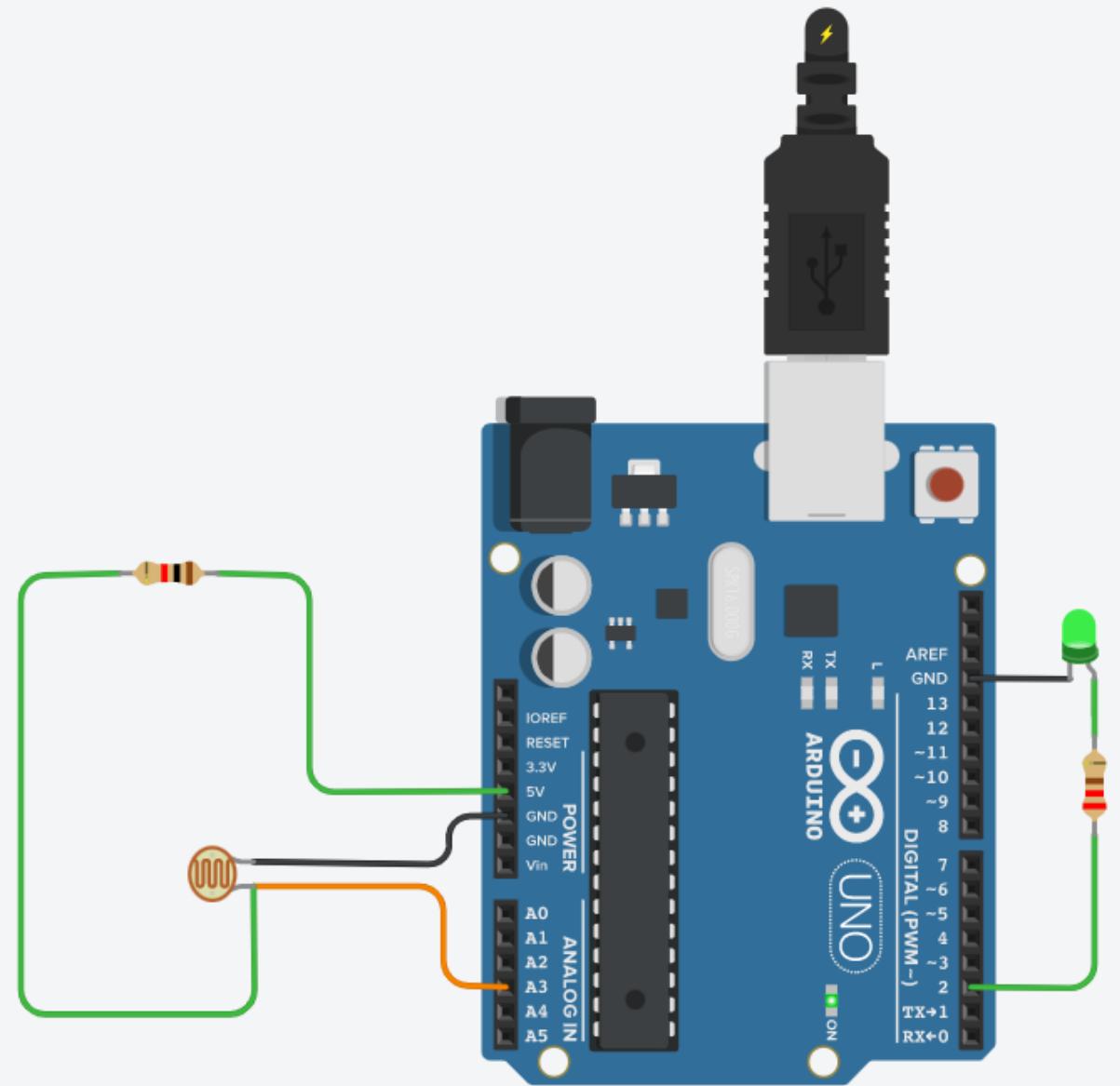
void setup() {
    servo1.attach(11);
    Serial.begin(9600);
}

void loop() {
    servo1.write(posicao);
    delay(1000);
    int posAtual = servo1.read(); // Lê a posição atual do servo
    Serial.print("Posição atual do servo: ");
    Serial.println(posAtual);

    posicao += incremento; // Incrementa a posição

    // Inverte a direção se atingir limites (0 ou 180)
    if (posicao >= 180 || posicao <= 0) {
        incremento = -incremento;
    }
}
```

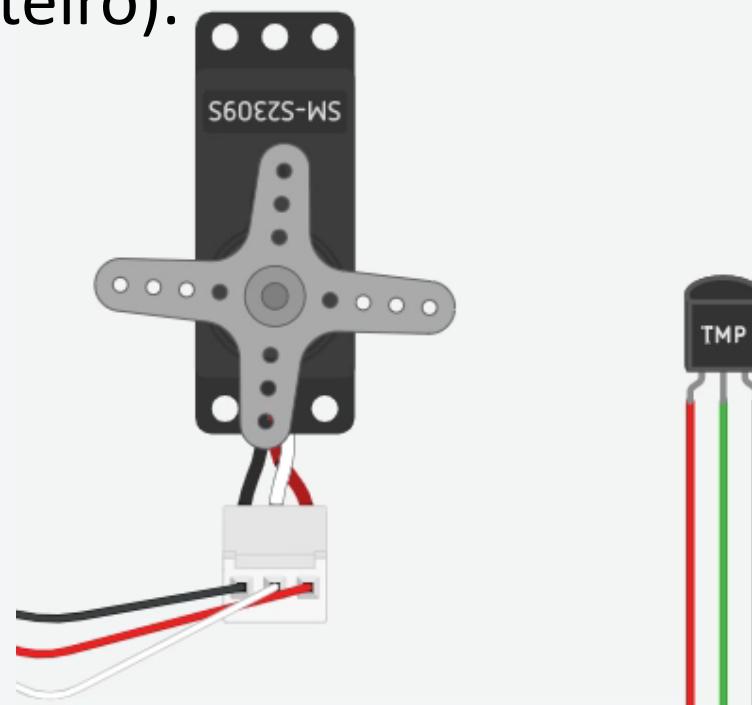
# LDR + LED (projeto visto nas aulas anteriores)



```
int varLDR;  
  
void setup () {  
    pinMode (2, OUTPUT);  
    pinMode (A3, INPUT);  
    Serial.begin(9600);  
}  
  
void loop () {  
    varLDR = analogRead(A3);  
    delay (100);  
    Serial.println(varLDR);  
    if (varLDR <= 488 ) {  
        digitalWrite (2, 0);  
    }  
    else {  
        digitalWrite (2, 1);  
    }  
}
```

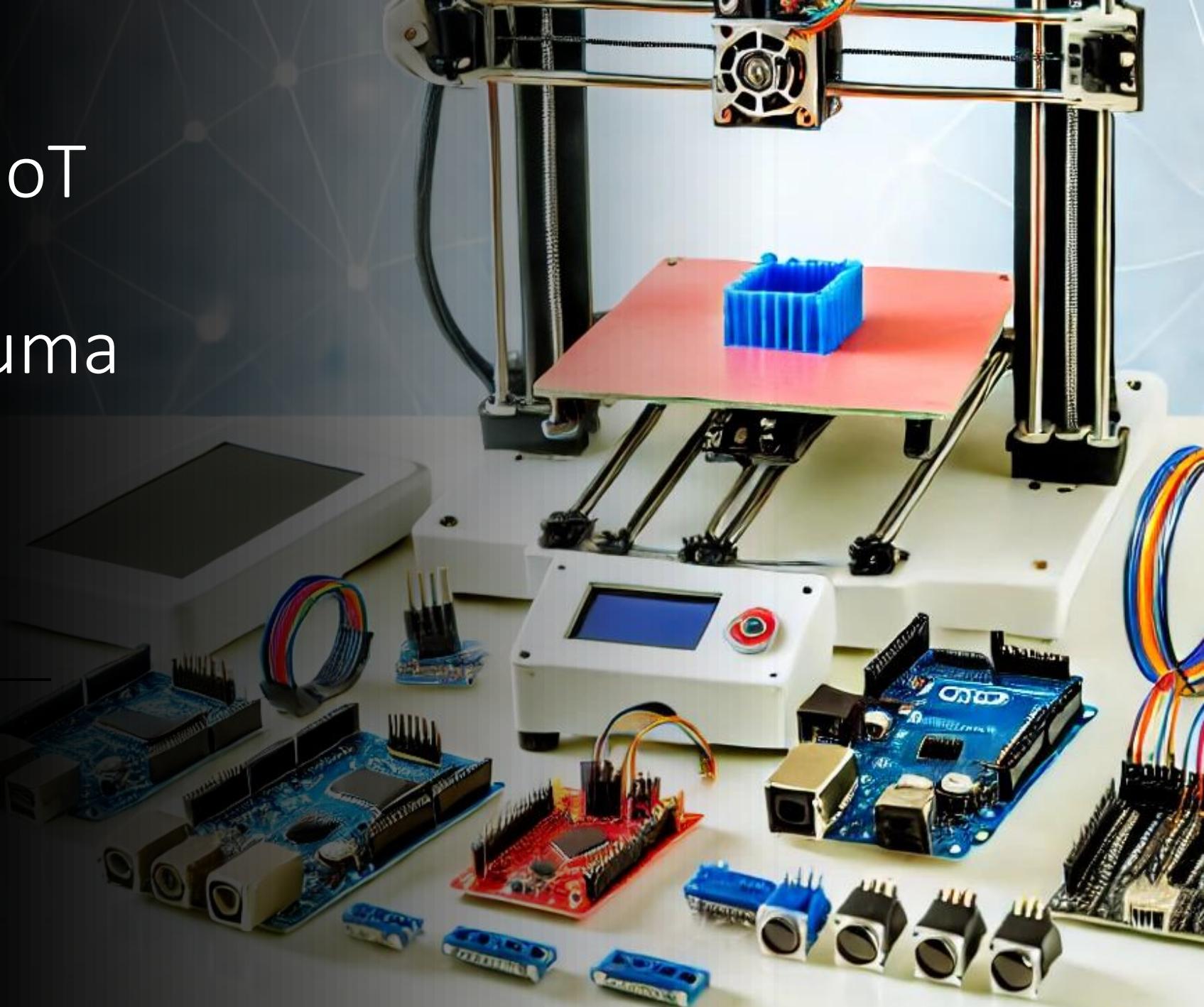
# Desafio

- 1 - Implementar um sistema que leia a temperatura do ambiente e indique se a temperatura está alta ou baixa através da posição de um servo. O servo deve funcionar como um indicador analógico (de ponteiro).
- 2- Implementar um sistema que detecte a luminosidade do ambiente e indique se o ambiente está claro ou escuro através da posição de um servo. O servo deve funcionar como um indicador analógico (de ponteiro).



# Impressão 3D e IoT

Como funciona uma impressora 3D?



A close-up photograph of a 3D printer's extruder assembly. A red filament is being heated by a hot end, which is connected to a yellow gear. The background shows the internal mechanical components of the printer.

how  
it works

# 3D Printer

Obrigado!



<http://bit.ly/3FCpl0K>