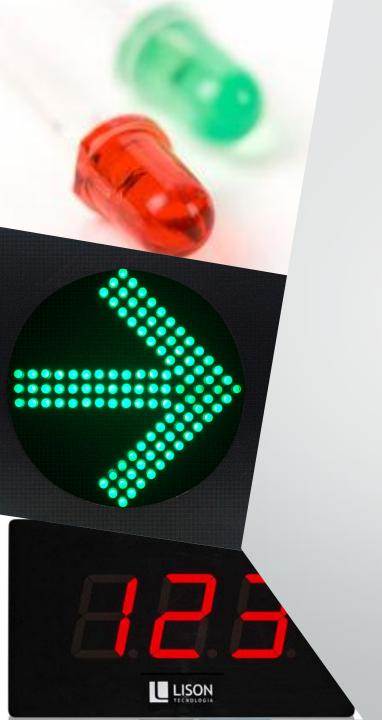




# Objetivo Geral

- Criar uma maquete que exemplifique o funcionamento
- Conceber inovações de modo a criar um sistema melhor
- Estudar e propor um sistema para determinar o estado do lugar
- Permitir uma gestão melhorada do espaço e trazer vantagens para quem o utiliza



# Funcionalidades a Implementar

- LED Verde/Vermelho consoante a disponibilidade
- Mostrar nº de lugares livres/ocupados do Parque
- Guardar o estado da última hora do lugar que foi modificado
- Implementar login no website
- Implementar um sensor de temperatura com ventoinha automática

# Funcionalidades a Implementar

- App/Website de gestão do Parque (Cliente + Administrador)
  - Utilizador terá acesso aos lugares disponíveis antes de se deslocar
- Ventoinha conforme temperatura



## <u>Vantagens</u>

- Otimização do Parque de Estacionamento
- Custo Reduzido comparado a outras soluções
- Redução da Poluição
- Redução do Tráfego (até 30%)
- Números de lugares disponíveis
- Controlo geral do Parque por uma aplicação
- Possibilidade de verificar disponibilidade em casa

# Sensores de veículos

- Indução
- Magnéticos
- Ultrassom
- Óticos- infravermelhos ou laser
- Baseados em vídeo e imagem



#### Sensor mais indicado --> €€€

- "Ultrassom HC SRo4"
- Facilmente programável com o Arduíno



Inúmeras vantagens tal como saber a distância a um obstáculo

• <a href="https://www.arduinoportugal.pt/como-ligar-o-sensor-distancia-ultrassom-hc-sro4-ao-arduino/">https://www.arduinoportugal.pt/como-ligar-o-sensor-distancia-ultrassom-hc-sro4-ao-arduino/</a>

2



## <u>Como</u> <u>implementar</u>

 O Sensor de Ultrassom será colocado no solo ou sobre o veículo



# 60,00 mm -|20,00 mm|<del>-</del>• 100,00 mm 53,84 mm

#### <u>Dimensões do</u> <u>estacionamento</u>

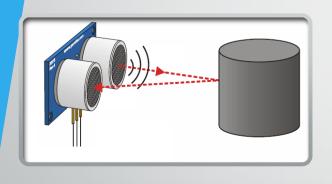
- A sua instalação deve ser feita o mais perpendicularmente possível com o alvo
- São sensíveis às mudanças de temperatura e a fortes ventos.



# Ultrassom "HC - SRo4"

- Transmitem ondas sonoras entre 25 kHz a 50 kHz
- Utilizam a <u>energia refletida</u> para analisar e detetar o estado do lugar
- Ondas ultrassónicas são emitidas do detetor a cada <u>1000</u>
   <u>milissegundos</u>
- Presença/ausência de veículos é determinada

#### Funcionamento do Ultrassom

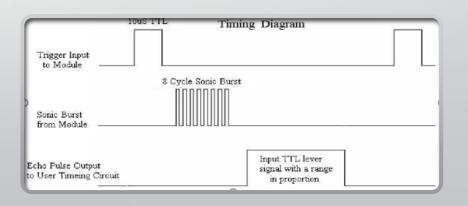


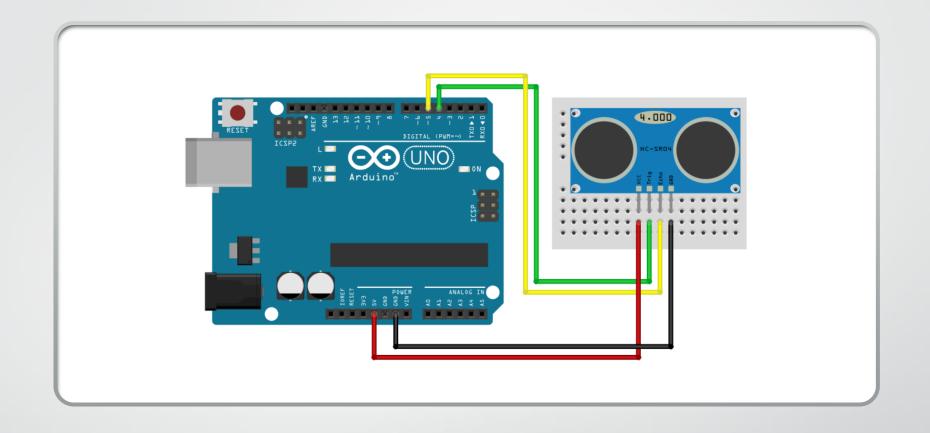
- O sensor baseia-se na velocidade com que o som viaja no ar (34300 cm / s).
- 1 Ele emite uma <u>explosão de ultrassons (emissor)</u> e um dos seus pinos é definida para <u>um</u>.
- 2 <u>Espera pelo eco (recetor)</u> a chegada do sinal e quando o eco vem coloca o <u>pino a zero</u>.
- 3 No microcontrolador quando se detetar a 1 no pino de entrada começa uma <u>contagem de tempo</u>.

#### Datasheet do Ultrassom

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- oV Ground







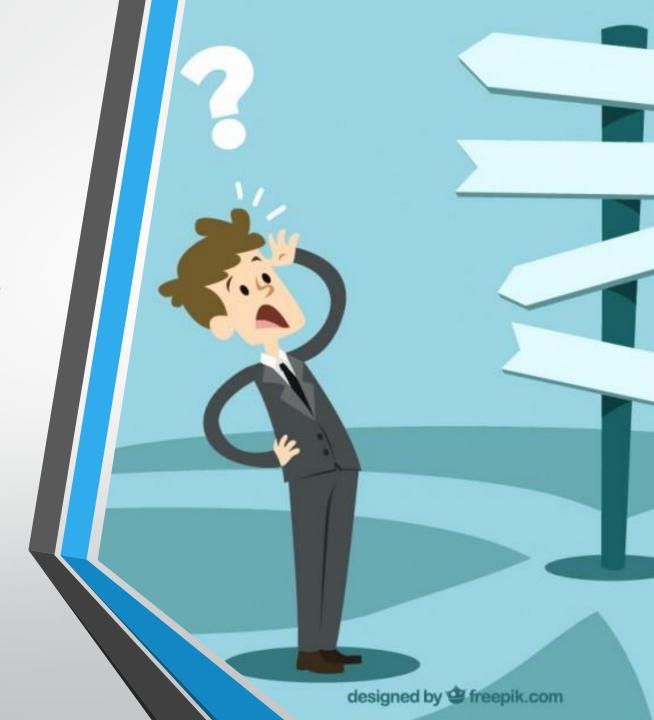
Ligação ao Arduino

#### Problemas encontrados!

- Informação é mostrada numa máquina!
- O microcontrolador tem de estar ligado ao computador!

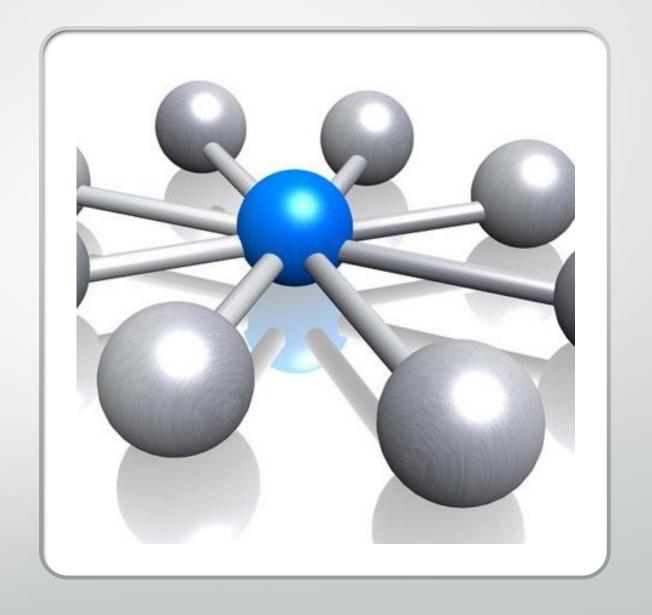
# Então e se houver uns 1000 ou mais lugares de estacionamento?

- Como suportar tantas entradas analógicas/digitais
- Sistema isolado?



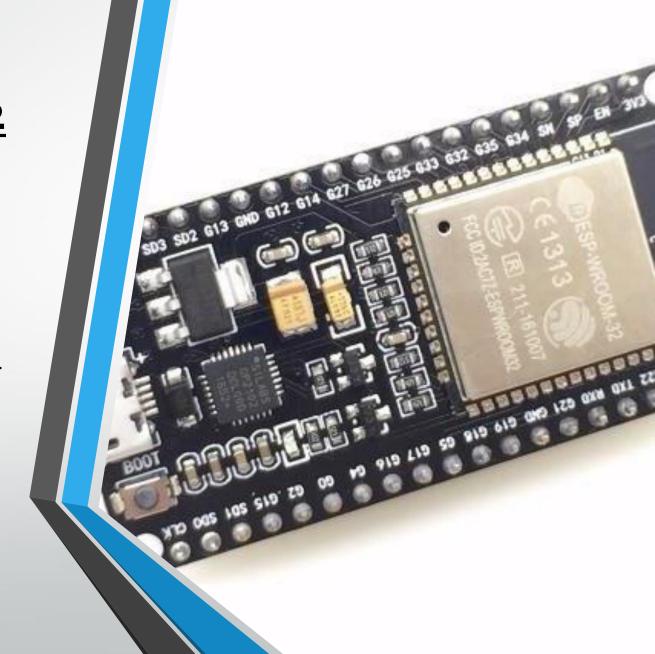
# <u>Servidor</u> <u>Centralizado</u>

- Arduino com Wi-Fi ou Ethernet
- Lugares de estacionamento isolados e independentes
- Porque não o ESP32?



## Arduino Uno VS ESP32

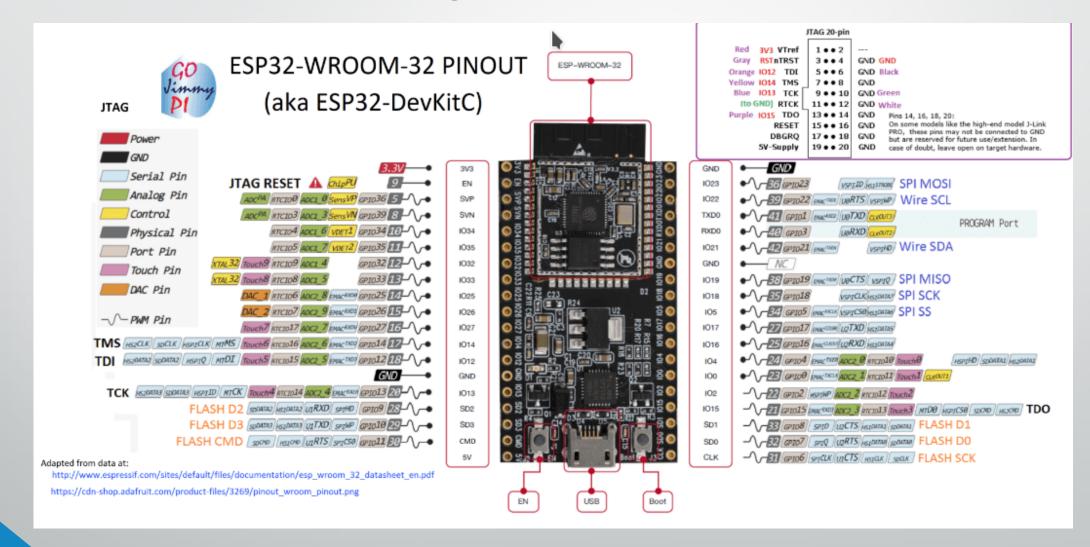
- É uma série de microcontroladores de <u>baixo</u> <u>custo e baixo consumo de energia</u>
- Espressif Sistemas Empresa Chinesa
- Wi-Fi e Bluetooth
- Programável com ARDUINO IDE

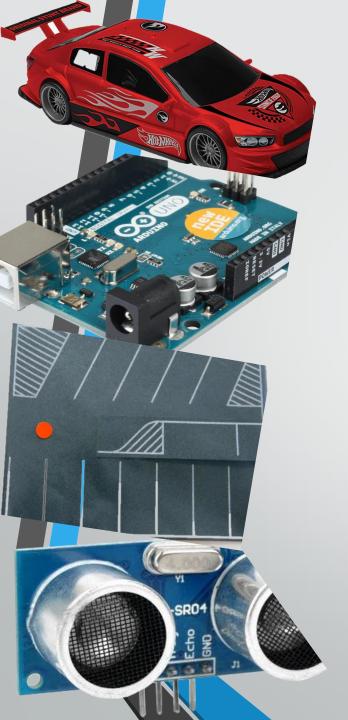


# Arduino Uno VS ESP32

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

## ESP32 Pinout





# Materiais a utilizar:

- 1 Microcontrolador p/ lugar
- 2 Led's vermelhos e 2 verdes
- 2 Sensores Ultrassom
- 2 Barreiras de entrada
- 2 Carrinhos



#### Conexão Wi-Fi

- É feita de <u>forma automática</u> cada vez que o dispositivo é ligado
- Password é enviada juntamente com o código



# Base de Dados – Pedidos HTTP com integração PHP

- O <u>ESP vai ler os dados</u> do sensor e <u>enviá-los</u> através da conexão Wi-Fi a um <u>servidor WEB</u> via requisição HTTP (GET ou POST);
- No servidor um script PHP vai receber os dados enviados e gravá-los na base de dados



#### **GET vs POST**

- POST é usado em <u>formulários</u> feito por inputs.
- **GET** é usado para passar valores pela <u>URL</u>.

- Vantagens de GET: Permite enviar dados rapidamente sem formulários
- Desvantagens de GET: Permite apenas 255 caracteres

- Vantagens de POST: É mais seguro e permite o envio de mais informações
- Desvantagens de POST: Necessita da criação de formulário, e inputs.

## Passo 1 - Criação da base de dados

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra
1	id 🔑	int(10)			Não	None		AUTO_INCREMENT
2	date_updated	timestamp		on update CURRENT_TIMESTAMP	Não	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP
3	mac	varchar(20)	latin1_swedish_ci		Não	None		
4	name	varchar(80)	latin1_swedish_ci		Não	None		
5	state	enum('free', 'busy')	latin1_swedish_ci		Sim	None		

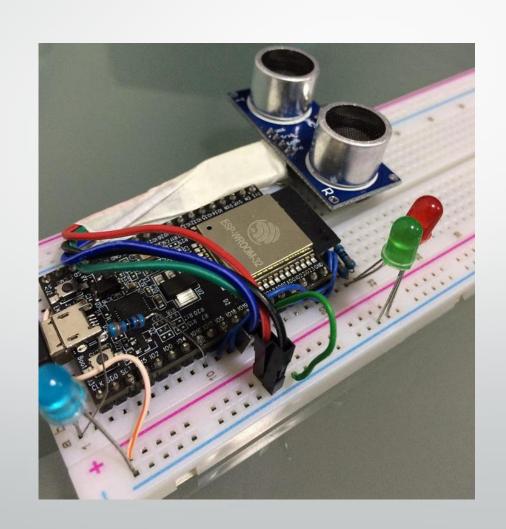
- O campo id (chave primária) será incrementado automaticamente a cada inserção;
- A mesma coisa acontecerá o campo "date\_updated" que guardará a data e hora em que o sinal foi alterado;
- Os campos "mac", "name" e "state" vão armazenar os dados do "mac address", nome e estado do lugar, enviados pelo ESP.

## Passo 2: Script PHP

- O script PHP ficará encarregado de:
  - Ler os dados dos sensores por GET
  - Fazer a conexão com a base de dados
  - Gravar os dados através de um comando UPDATE

```
🔚 insert_parking.php 🔣
        $mac = filter input(INPUT GET, 'mac', FILTER SANITIZE STRING);
        $state = filter input(INPUT GET, 'state', FILTER SANITIZE NUMBER FLOAT);
      if (is null($mac)) {
          //Gravar log de erros
          die ("Dados inválidos");
        $servername = "localhost";
        $username = "root";
        $password = ""
        $dbname = "parking";
        $conn = new mysqli($servername, $username, $password, $dbname);
     if ($conn->connect error)
         //Gravar log de erros
15
          die ("Não foi possível estabelecer conexão com o BD: " . $conn->connect error);
16
17
     if ($state == 1)
19
            $state = "busy";
            $state = "free";
21
22
23
      if ($stmt = $conn->prepare("UPDATE parkings SET state = ? WHERE mac = ?")) {
25
            $stmt->bind param("ss", $state, $mac);
26
        if ($stmt = $conn->prepare("INSERT INTO parkings (namep, state) VALUES (?, ?)")) {
            $stmt->bind param("sd", $namep);
29
            $stmt->bind param("d", $state);
30
31
32
            if (!$stmt->execute())
33
            //Gravar log de erros
34
                die ("Erro na gravação dos dados no BD");
35
36
```

# Passo 3: Protótipo ESP32



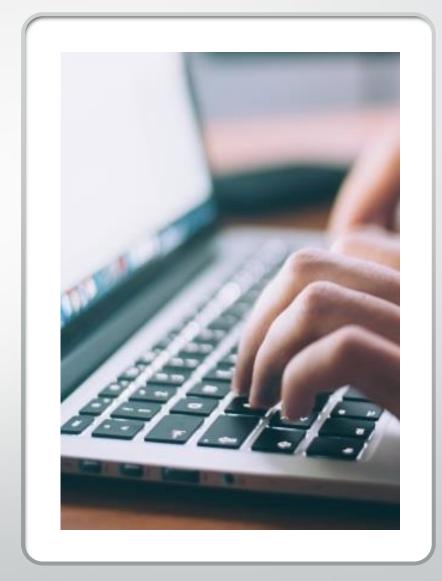
### Passo 4: XAMPP

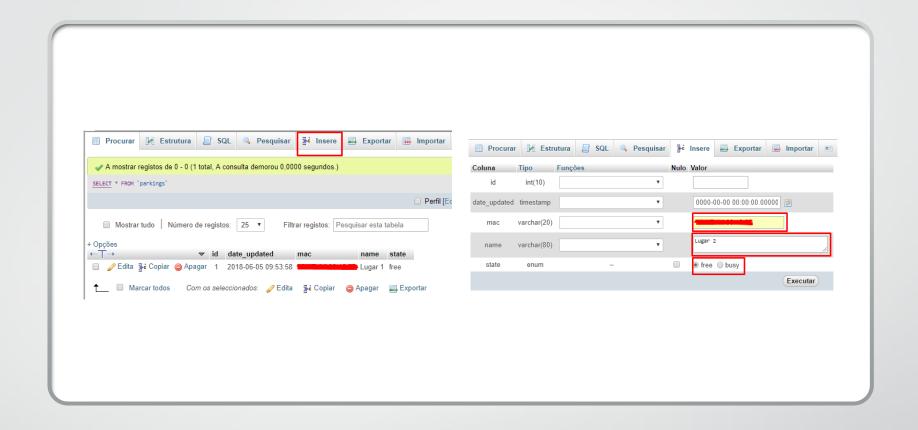
- Depois de criar a base de dados e carregar o código para o ESP:
  - Dar "Start" no "Apache"
     e "MySQL" através da
     ferramenta Xampp.
  - Criar o <u>index.php</u>



# Registo de utilizador em PHP e MySQLi

- Criar uma base de dados
- Criar uma tabela de base de dados
- Criar um formulário de registo
- Criar um formulário de login
- Conexão à base de dados usando PHP
- Autenticar utilizador conectado
- Criar página de índice
- Criar página do painel
- Criar um logout
- Criar um arquivo CSS





Registar ESP's na Base de Dados



#### Log In

Username

Password

Login

Não registado? Registe-se aqui

Humidade	Temperatura
46 %	21 °C



Smart
Parking
Solution
Miguel Oliveira

# Layout

# Registo



#### **Funcionalidades**

- Disponibilidade do lugar
- Lugares do Parque
- Lugares disponíveis

#### Olá admin!

Logout



#### Lugares de estacionamento

ID	Data	MAC	Name	State
1	2018-06-18 14:20:32	30:AE:A4:00:43:08	Lugar 1	free
2	2018-06-18 14:20:33	30:AE:A4:10:A1:7C	Lugar 2	free

Lugares do parque 2 lugares

Lugares disponíveis 2 lugares

# Implementação da Ventoinha

- Ligou-se a ventoinha ao "ESP32".
- Faltava resolver a questão de conseguir controlá-la de modo a desligar-se e ligar-se
- Sem que tivesse de se mexer nos cabos.



# Não existe transitor disponível --> Utiliza-se RELÉ??

Consiste num <u>dispositivo</u>

 <u>eletromecânico</u>, formado por um magneto móvel, que se desloca unindo dois contatos metálicos

 Uma espécie de <u>antepassados dos</u> <u>transístores</u>

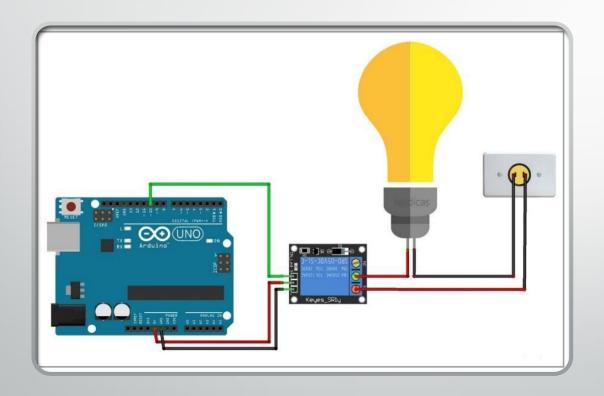


# Como assim dispositivo eletromecânico?

- Basicamente é uma válvula hidráulica
- Por exemplo a do chuveiro
- Ela tem a função de <u>controlar o fluxo de água</u> que sairá pelo chuveiro



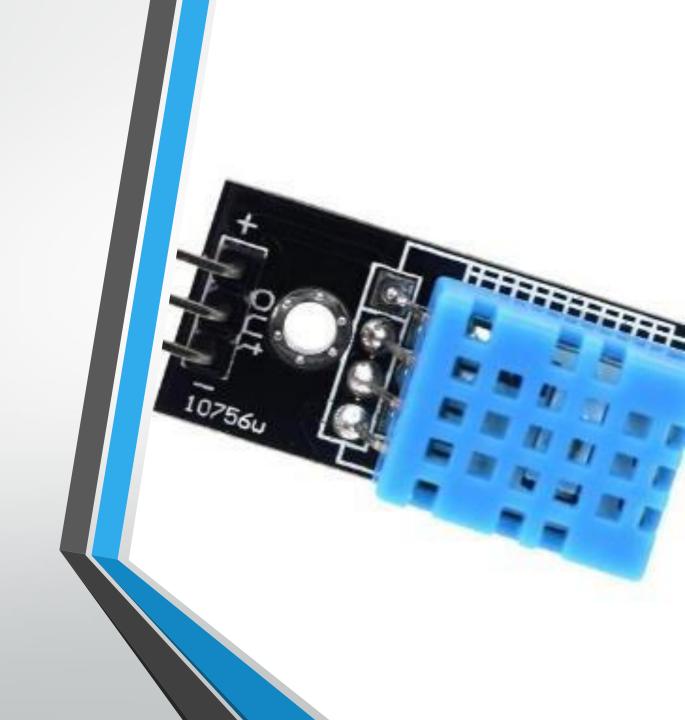
#### Interligação do Relé



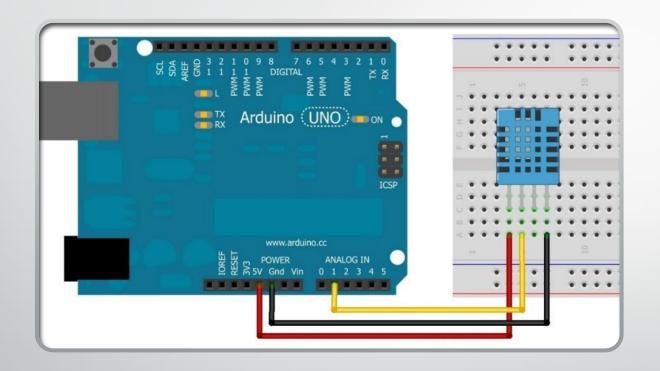
```
* Firmware para controlar uma lâmpada
 * com um módulo de relé
 * Ana Paula Messina - Tec Dicas
 * 13/11/2017
void setup()
 pinMode(10, OUTPUT); // Porta 10
void loop()
  digitalWrite(10, HIGH); // Liga/Fecha
 delay(2000); // 2 seg
  digitalWrite(10, LOW); // Desliga/Abre
 delay(2000); // 2 seg
```

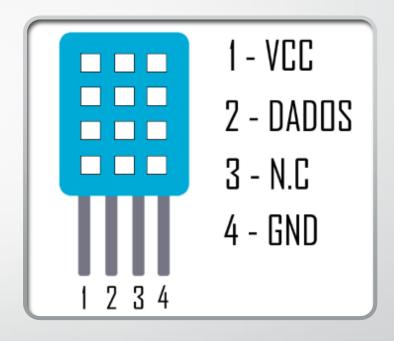
# DHT11 – Sensor de Temperatura

- É um sensor de <u>temperatura e humidade</u>
- Permite fazer leituras de temperaturas
  - entre o a 50 Celsius
  - humidade entre 20 a 90%,



#### Ligações/Pinos





```
DHTxx test!

Humidity: 32.70 % Temperature: 25.70 *C

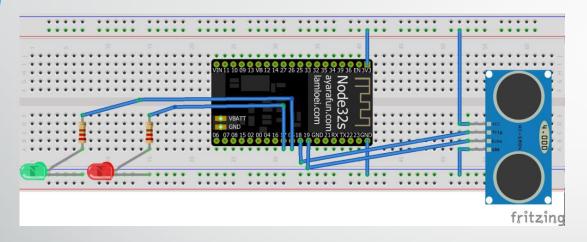
Humidity: 32.60 % Temperature: 25.70 *C
```

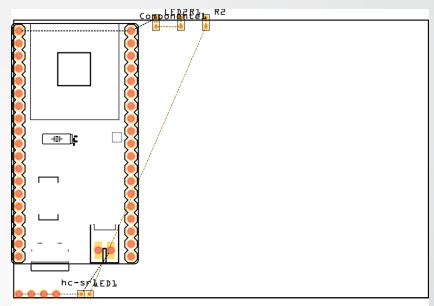
# Envio para base de dados



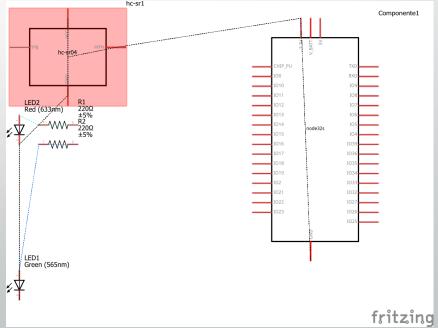


# Esquema dos Circuitos Estacionamento

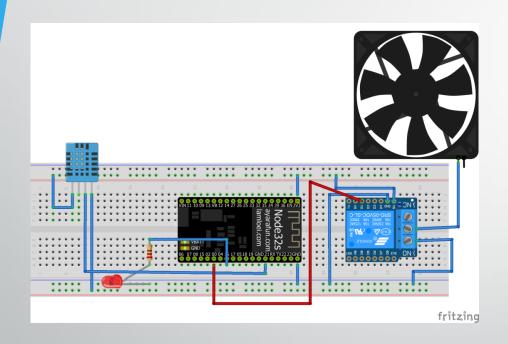


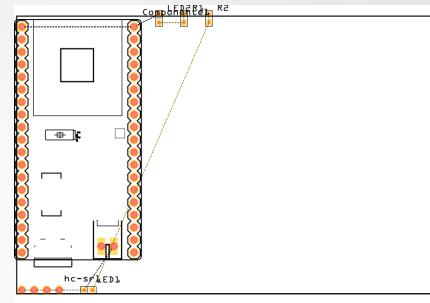


fritzing

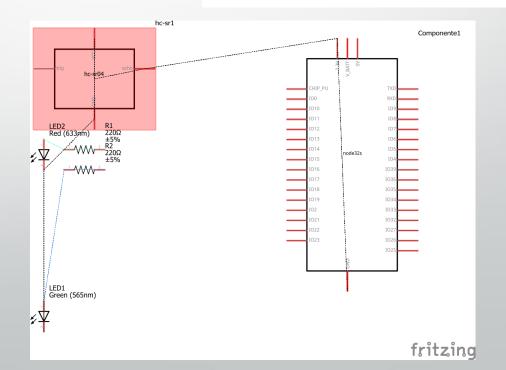


# Esquema dos Circuitos Temperatura





fritzing



#### Rede Wi-Fi

- Para que tudo funcione sem necessidade de usar outras redes, podemos utilizar um router Wi-Fi
- De seguida, na zona das reservas do DHCP, selecionamos o nome do computador desejado (Servidor), colocamos o IP a fixar e o MAC Address correspondente ao Servidor



```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix :
Link-local IPv6 Address . . : fe80::a1f4:1b28:d107:9312%7
IPv4 Address . . . : 192.168.0.100
Subnet Mask . . . . : 255.255.255.0
Default Gateway . . : 192.168.0.1
```

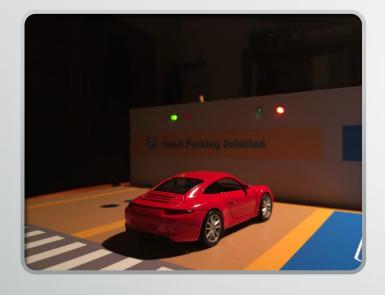
Wireless Network Name parking (SSID):

Security Mode 1: Auto (WPA or WPA2) - Personal

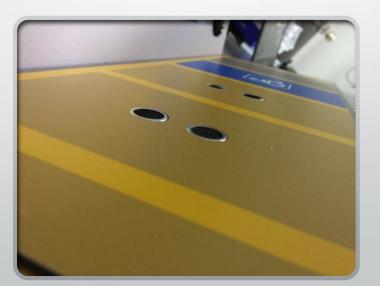
Cipher Type: TKIP and AES

Pre-Shared Key: parking1234

## **Maquete**









# Video de Demonstração

