

## **Veículo/Carro inteligente**

André Santos Pinto

Micael Ferreira Marques

Trabalho de Projeto da unidade curricular de Tecnologias de Internet

Leiria, junho de 2021



# Lista de Figuras

Figura 1 - Imagem representativa da arquitetura de solução .....	5
Figura 2 – Esquema de associação da temperatura ambiente à ventoinha do carro .....	6
Figura 3 - Ventoinha controlada pelo Servidor IoT .....	7
Figura 4 - RFID card reader .....	8
Figura 5 – Esquema associado às portas .....	9
Figura 6 - Sensor de Luminosidade.....	10
Figura 7 - Sensor de movimento .....	11
Figura 8 - Sensor de CO2.....	12
Figura 9 - Sensor de ruído branco .....	13
Figura 10 – Esquema relacionado ao nível de bateria.....	14
Figura 11 – Esquema relacionado a luz de leitura.....	15
Figura 12 – Esquema do cenário de teste .....	18

## Lista de siglas e acrónimos

CO2	<i>Carbon dioxide</i>
CPT	Cisco Packet Tracer
ESTG	Escola Superior de Tecnologia e Gestão
IoT	<i>Internet of Things</i>
IPLeiria	Instituto Politécnico de Leiria
LCD	<i>Liquid crystal display</i>
LED	<i>Light Emitting Diode</i>
MCU	<i>Microcontrollers</i>
RFID	<i>Radio Frequency identification</i>
SBC	<i>Single Board Computer</i>

# Índice

<b>Lista de Figuras .....</b>	<b>iv</b>
<b>Lista de siglas e acrónimos.....</b>	<b>v</b>
<b>1. Introdução.....</b>	<b>1</b>
<b>2. Arquitetura .....</b>	<b>3</b>
<b>3. Implementação .....</b>	<b>6</b>
<b>4. Cenário de Teste .....</b>	<b>16</b>
<b>5. Resultados obtidos.....</b>	<b>19</b>
<b>6. Conclusão .....</b>	<b>23</b>
<b>7. Bibliografia .....</b>	<b>24</b>
<b>8. Anexos .....</b>	<b>25</b>
<b>8.1. Código do Microcontrolador – Sensores.....</b>	<b>25</b>
<b>8.2. Código do Microcontrolador – Sensores genéricos.....</b>	<b>28</b>
<b>8.3. Código do Microcontrolador – Alterar estado portas .....</b>	<b>29</b>
<b>8.4. Código do SBC – Mostrar estado Portas .....</b>	<b>31</b>
<b>8.5. Código do SBC – Atuadores2.....</b>	<b>34</b>
<b>8.6. Código do SBC – Atuadores1.....</b>	<b>37</b>
<b>8.7. Código do PHP Atuadores.php .....</b>	<b>43</b>
<b>8.8. Código do PHP Dashboard.php.....</b>	<b>54</b>
<b>8.9. Código do PHP Historico.php .....</b>	<b>64</b>
<b>8.10. Código do PHP Imagens.php .....</b>	<b>68</b>
<b>8.11. Código do PHP Login.php.....</b>	<b>71</b>
<b>8.12. Código do PHP Logout.php.....</b>	<b>74</b>

<b>8.13.</b>	<b>Código do PHP Navbar.php .....</b>	<b>75</b>
<b>8.14.</b>	<b>Código do PHP Pannel.php .....</b>	<b>93</b>
<b>8.15.</b>	<b>Código do PHP Perfil.php .....</b>	<b>96</b>
<b>8.16.</b>	<b>Código do PHP Perfil.php .....</b>	<b>102</b>
<b>8.17.</b>	<b>Código do PHP Upload.php.....</b>	<b>111</b>
<b>8.18.</b>	<b>Código do PHP UploadFoto.php.....</b>	<b>112</b>
<b>8.19.</b>	<b>Código de Python capturaWebcamOpenCV.py .....</b>	<b>114</b>
<b>8.20.</b>	<b>Código de Python tocaPorta.py .....</b>	<b>115</b>
<b>8.21.</b>	<b>Código de Python percentagemBateria.py .....</b>	<b>117</b>
<b>8.22.</b>	<b>Código de Python luzLeitura.py .....</b>	<b>118</b>



# 1. Introdução

O tema do trabalho é o veículo/carro inteligente, em que neste projeto será um carro totalmente elétrico, que responde à necessidade de atuar em tempo real com certos dispositivos em função de dados obtidos por sensores.

Decidimos escolher este tema, pois, dentro dos temas que foram disponibilizados, é um assunto que nos despertou interesse e que tínhamos algumas ideias previamente definidas. Adicionalmente, é um tema em que julgamos conseguir mais facilmente adaptar ao nosso quotidiano em comparação com a maioria dos restantes temas disponibilizados. Além disso, dá-nos a possibilidade de imaginar uma marca de carros elétricos, a AllElectric, podendo personalizar tudo o que é inerente à marca, nomeadamente o estilo do logótipo e do website apresentado para os possíveis utilizadores.

Quanto ao objetivo do trabalho, sendo sobre um carro elétrico, o nosso primeiro objetivo será colocar uma bateria representativa de um carro elétrico, recarregável e portas automáticas, que não precisem de interação direta com o equipamento para abrirem, testando ainda a capacidade dos vários sensores e atuadores comunicarem e agirem entre si, automaticamente. Idealizámos também, que como é um carro elétrico, teríamos um sistema de autorização para poder usufruir do carro.

O carro elétrico tem uma página web própria para cada utilizador que dispõe de várias informações, caso seja necessário saber alguma informação quando não se está perto do mesmo, como saber se tem portas abertas, que temperatura de ambiente existe...

Ao longo deste relatório, vamos abordar no capítulo 2 a arquitetura da solução implementada, onde descrevemos os equipamentos utilizados na resolução, no capítulo 3 abordamos a implementação do projeto, em que referimos os algoritmos mais relevantes nos diferentes softwares utilizados. Já no capítulo 4, falamos sobre o cenário de teste, em que explicamos mais detalhadamente o funcionamento do projeto, os procedimentos realizados e a sua estrutura. No capítulo 5 abordamos os resultados que obtivemos na realização do projeto, fazendo a conclusão do trabalho no capítulo 6.

No final do projeto temos como objetivo principal ter, portas que sejam possíveis abrir e fechar à distância, medir temperaturas, humidade, níveis de luz, níveis de CO<sub>2</sub> e níveis de



barulho e agir conforme estas. Graças a um sensor de movimento detetar qualquer tipo de movimento e pedir à camera captar uma imagem enquanto sentir movimento, para prevenir de roubos ou danos no veículo (estes sensores estariam idealmente a rodear o carro). Todas estas informações, também vão estar disponíveis no website do carro, contendo um histórico para todos os valores anteriormente obtidos e com a hora do acontecimento.

São utilizados ainda os laboratórios das aulas práticas que aproveitámos e adaptámos em algumas partes do projeto, usando-se várias ferramentas, como HTML, PHP, CSS, JavaScript e Python, tal como o programa CPT – Cisco Packet Tracer.

O trabalho está estruturado então, em 3 grandes partes: o cenário do Cisco Packet Tracer, onde temos quase todos os sensores (excetuando a bateria) e alguns dos atuadores, o dashboard / website, em que se atualiza todos os conteúdos e se tem 2 atuadores específicos apoiados em ficheiros php, imagens, e ficheiros relativos a uma API, e ainda ficheiros python, executados através da linha de comandos.

## 2. Arquitetura

Sendo o tema um carro inteligente, que no nosso caso é um veículo totalmente elétrico, todos os sensores, exceto o que capta o nível da bateria, vão captar informações sobre o ambiente. Usando o CPT, temos os vários equipamentos virtuais contando com alguns **microcontroladores**: 3 MCU e 3 SBC; com **sensores**: um sensor de movimento, um sensor de temperatura, um sensor de humidade, um sensor de luminosidade, 2 sensores genéricos personalizados por nós, sendo que um serve para obter o CO<sub>2</sub> ambiente (visto que o próprio carro não emite CO<sub>2</sub>, por ser um carro totalmente elétrico), e o outro para obter o ruído branco existente no ambiente; **atuadores**: 2 portas (esquerda e direita), um LED de aviso sobre o estado das portas, um LED que funciona como luz de leitura, uma luz de aviso sobre os níveis do CO<sub>2</sub>, e um tejadilho (representado por uma porta no CPT); um monitor de temperatura, um servidor IoT, um Switch, um Access Point, um RFID Card Reader e 2 cartões RFID, sendo que um cartão é o correto e o outro é incorreto. Usamos ainda 2 LCD para mostrar informações diversas e um monitor de temperatura ambiente, cuja temperatura é obtida no Servidor IoT. Utilizamos também equipamento real, como a bateria do computador (que neste caso tem que se usar um computador portátil para testar), a webcam do computador, e as colunas de som do computador (sendo que tanto a webcam como as colunas de som estão representadas no CPT, enquanto o nível de bateria é mostrado num LCD).

Os sensores genéricos criados (para o CO<sub>2</sub> e para o Ruído Branco) estão ligados a um MCU exclusivo para estes, que está sempre a enviar, de 1 em 1 segundo essa informação para a API. Com essa informação, dependendo dos valores, existe uma resposta por parte de atuadores, em que é ligada uma luz de aviso, com 3 estados de alerta, de níveis elevados de CO<sub>2</sub>, caso estes existam, e caso haja um nível elevado de ruído branco, o tejadilho do carro fecha-se, até esses níveis baixarem.

Todos os outros sensores estão todos ligados a um mesmo MCU, que atualiza os valores obtidos para a API de 2 em 2 segundos. Optámos por ter um sensor de temperatura, que dependendo dos valores da temperatura, faz com que se ligue a ventoinha do carro (no caso de a temperatura ultrapassar os 15°C), que não vai baixar a temperatura por si só, mas sim contrariar a sensação de calor sentida dentro do carro. Temos também o sensor de humidade, que vai ser mostrada essa informação no LCD do carro. O sensor de movimento, ativado no

CPT, vai através do MCU enviar a informação se existe ou não um objeto próximo, sendo que se existir, vai ser tirada uma foto, de 5 em 5 segundos, enquanto ainda existir esse objeto nas proximidades. Por fim, usamos o sensor de luminosidade que vai detetando se existe um nível baixo ou elevado de luminosidade no ambiente, em que vão ser ligados os faróis do carro caso se detete que a luminosidade seja baixa.

A nível de atuadores, optámos por colocar botões (tanto no CPT - ligados a MCU - como no website) para abrir/fechar as portas, sendo que quando estas estão abertas, é emitido um som. Também utilizamos um botão no website (e também através de um script no Python) para ligar/desligar a luz de leitura. Usámos também um RFID reader ligado diretamente ao Switch que irá analisar se o RFID Card inserido é o correto, e se este assim o for, irá destrancar as portas. Estas condições estão definidas no servidor IoT.

Temos ainda 2 LCD, sendo que um está ligado ao MCU das portas do carro e indica se as portas estão abertas ou não, e um outro que está ligado a um outro MCU e indica diversas informações sobre o carro, tal como o nível da bateria, se a ventoinha do carro está ou não ligada e o nível da humidade de ambiente. Ligado a este último MCU, colocámos 3 LED, sendo que 2 deles são os faróis do carro e um deles que liga quando os níveis de bateria estão baixos, e ainda um speaker, que emite um som quando as portas estão abertas.

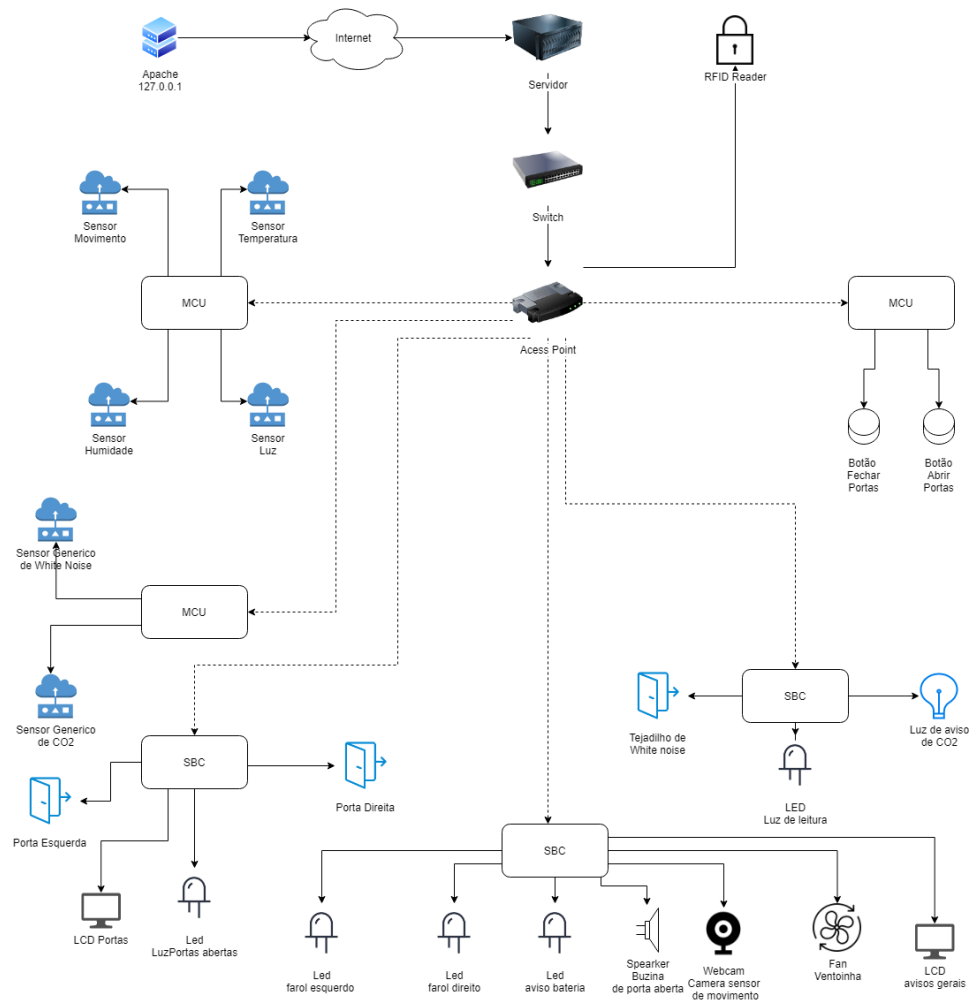


Figura 1 - Imagem representativa da arquitetura de solução

### 3. Implementação

O sensor de temperatura está constantemente a medir a temperatura ambiente. De seguida envia essa informação para o MCU, que vai calcular o valor obtido e converter para graus celsius, sendo essa informação enviada para a API, voltando a repetir o ciclo. (Figura 2)

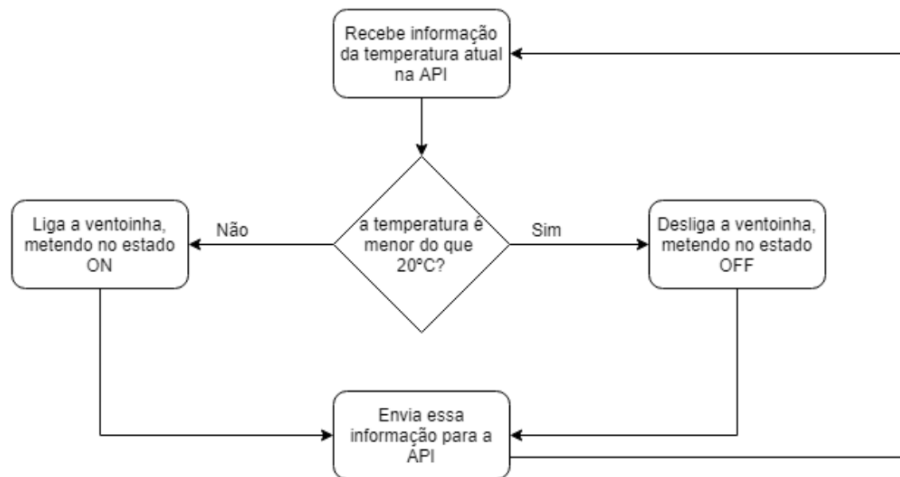
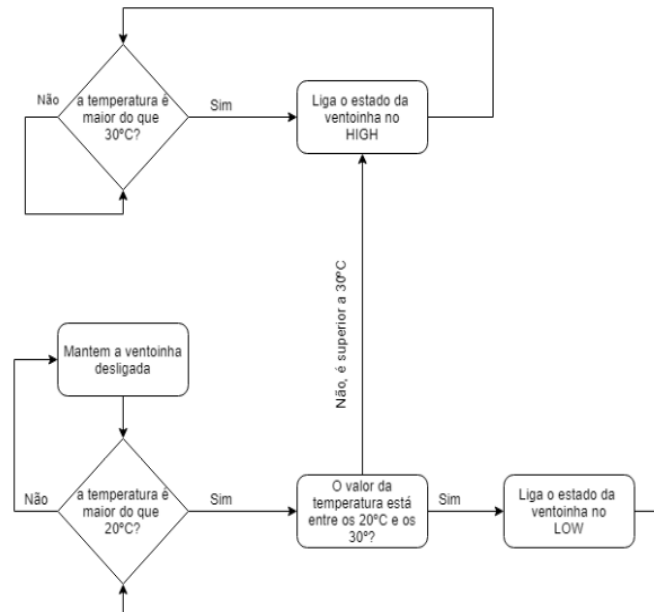


Figura 2 – Esquema de associação da temperatura ambiente à ventoinha

Quando o valor da temperatura é recolhido, e este estiver abaixo dos 20°C a ventoinha ira manter o seu estado desligado. Porem, quando este valor for igual ou superior a 20°C mas inferior a 30°C, a ventoinha ira ser colocada num estado chamado de LOW, através do servidor IoT do CPT. (ver Figura 3)



**Figura 3 - Ventoinha controlada pelo Servidor IoT**

Também no servidor IoT do CPT temos um RFID, quando a chave é válida e for mantida em cima do RFID as portas irão ser destrancadas, no entanto se for retirado, as portas são imediatamente trancadas de novo (ver Figura 4). Se o RFID não detetar uma chave valida, este vai manter o estado das portas trancado (o que impossibilita abrir as portas). Também disponível temos dois botões, um para abrir e outro para fechar as portas. Se o botão de fechar as portas for pressionado, envia o valor “0” para a API e fecha as portas, é desligado o LED de aviso, é escrito no LCD que estão 0 portas abertas, sendo que todas estas informações são enviadas para a API. Por outro lado, se for pressionado o botão de abrir as portas, envia o valor “1” para a API e abre as portas, é ligado um LED de aviso, escreve-se no LCD que estão 2 portas abertas, e procede a enviar estas informações para a API. Mais ainda, já com a informação na API o microcontrolador associado ao buzzer vai verificar se as portas estão abertas ou não. Se estas estiverem abertas vai acionar o buzzer de forma a avisar o utilizador que tem as portas abertas, também mandando esta informação para a API. (ver Figura 5 onde está representado o esquema referido).

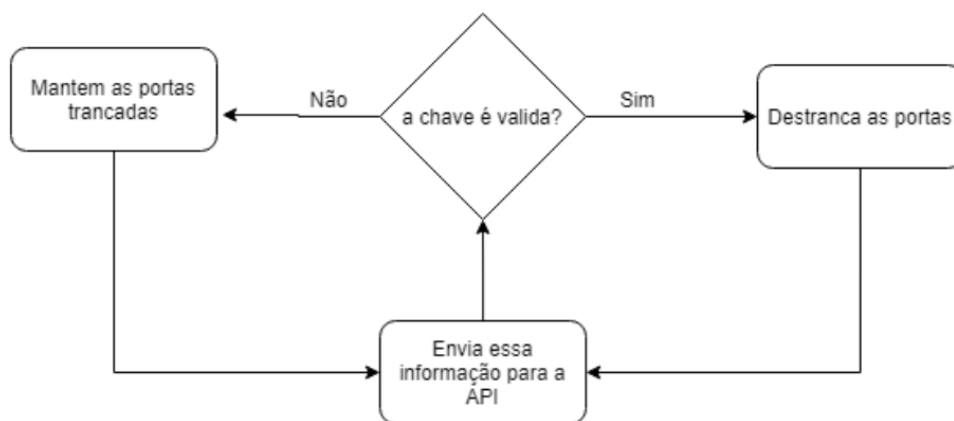


Figura 4 - RFID card reader

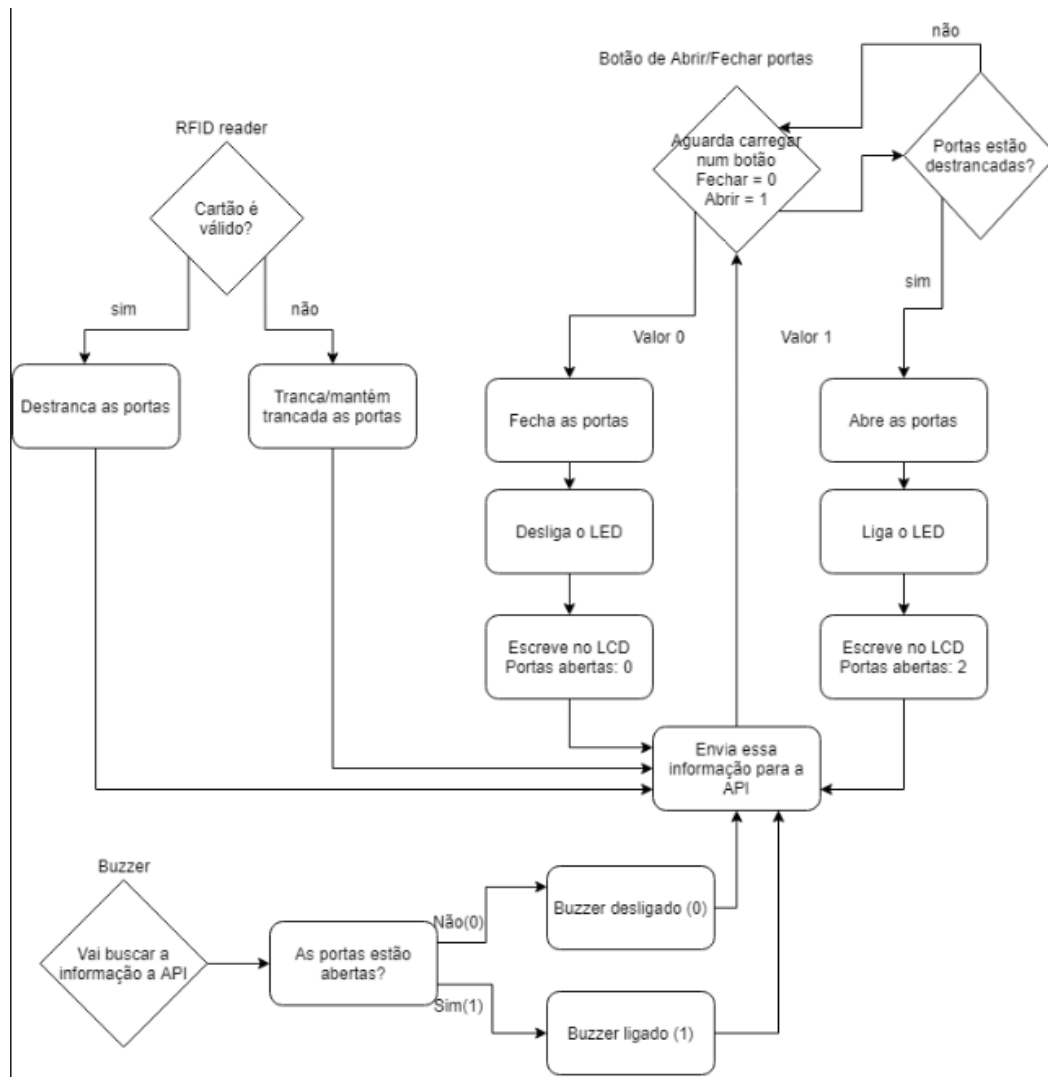


Figura 5 – Esquema associado às portas



O sensor de luz mede a luminosidade existente no ambiente, e verificar se está superior a 20%. Se assim estiver, envia para o MCU o valor HIGH”, caso seja inferior a 20% envia LOW. De seguida envia a informação obtida para a API, que dependendo do recebido liga ou desliga os faróis.. (Figura 6)

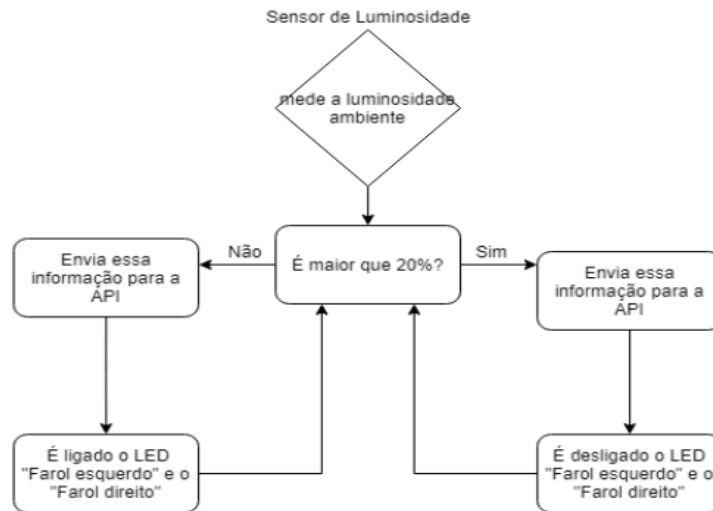


Figura 6 - Sensor de Luminosidade

Quando o sensor de movimento deteta movimento, este vai mandar essa informação para a API, a partir desse momento, sendo que um script em Python está periodicamente a pedir essa informação à API, a camera é ligada e envia-se essa informação para a API, sendo depois tirada uma foto que vai ser descarregada para uma pasta, onde depois é disponibilizado uma página de histórico no website. Este ciclo é repetido de 5 segundos em 5 segundos. (ver Figura 7)

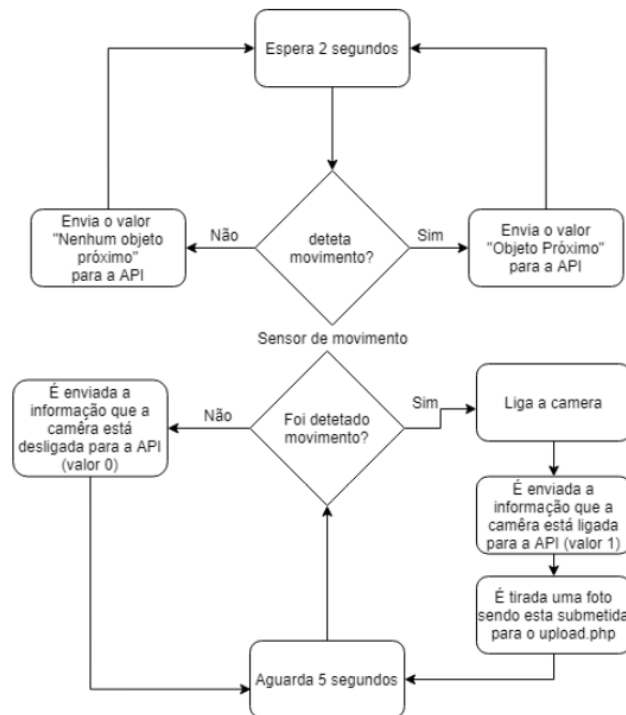


Figura 7 - Sensor de movimento

O sensor de CO2 vai buscar o valor em percentagem do “Environments” do CPT, vai enviar esse valor para a API, depois a API verifica se esse valor é superior a 40%. Se for vai mandar ligar a “Luz de aviso CO2” no estado 2 (Luz mais intensa). Se o valor não for superior a 40%, será colocada outra pergunta, desta vez se o valor é superior a 20%, mas inferior a 40%, e se for, vai mandar ligar a “Luz de aviso CO2” no estado 1 (Luz normal). Se o valor inicialmente obtido não for superior a 20%, vai desligar a “Luz de aviso CO2” metendo o estado a 0 (Desligado). (ver Figura 8)

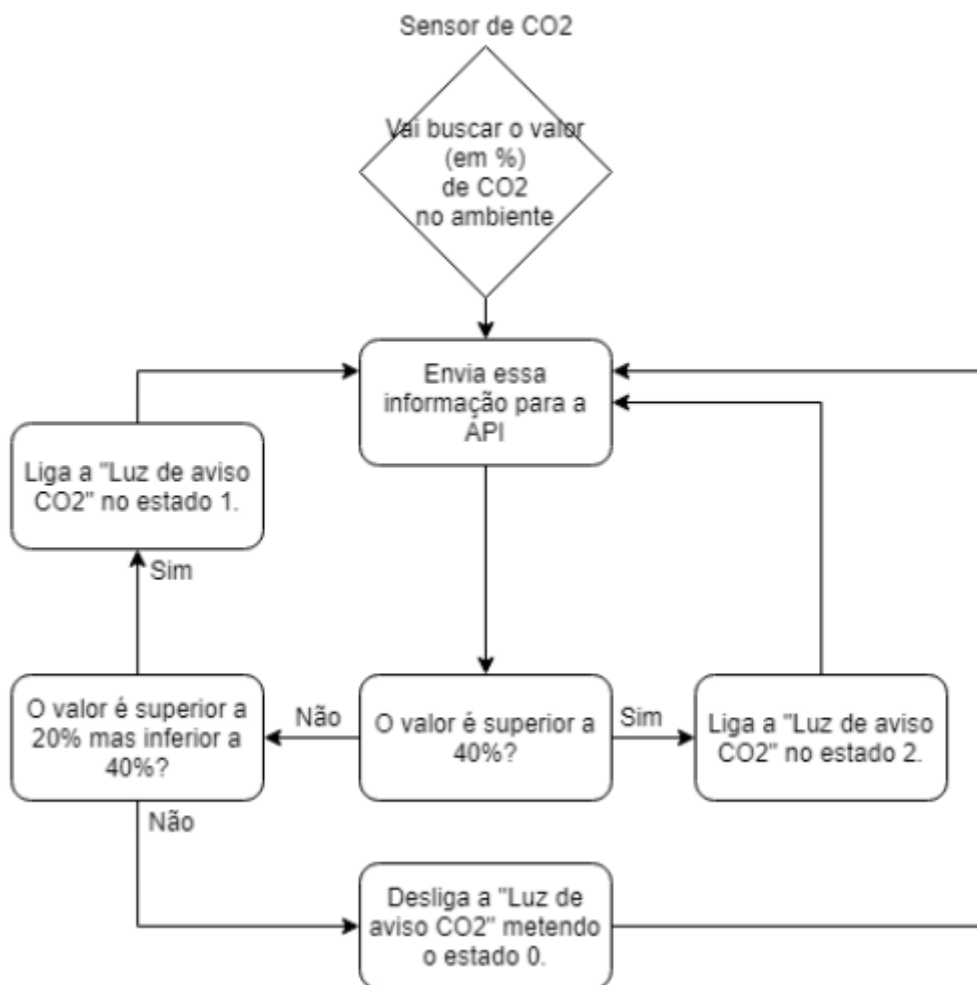


Figura 8 - Sensor de CO2

O sensor de Ruído branco vai buscar o valor em percentagem do “Environments” do CPT, envia essa informação para a API, e de seguida verificar se o valor é superior a 15%. Se assim for, fecha o tejadilho como método de proteção. Se não for superior a 15%, abre o tejadilho. (Figura 9)

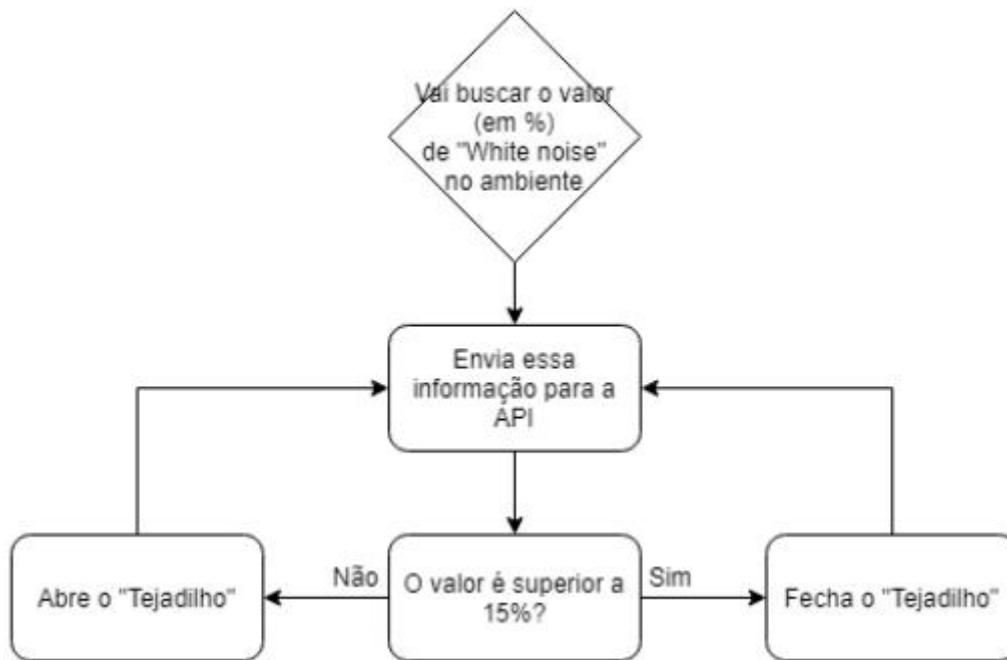


Figura 9 - Sensor de ruído branco

Em relação ao LED da bateria, verifica se o nível da bateria é menor do que 30%, se for, vai ligar o LED de aviso de bateria, e escreve no LCD “*Battery: Low*”. No entanto se a bateria for superior a 30%, vai verificar de novo, se a bateria é inferior a 60% mas superior a 30%, se assim for vai desligar o LED de aviso de bateria, e escrever no LCD “*Battery: Medium*”. Se não for o caso de estar entre 30% e 60%, vai desligar o LED de aviso de bateria, e escrever no LCD “*Battery:High*”. (Figura 10)

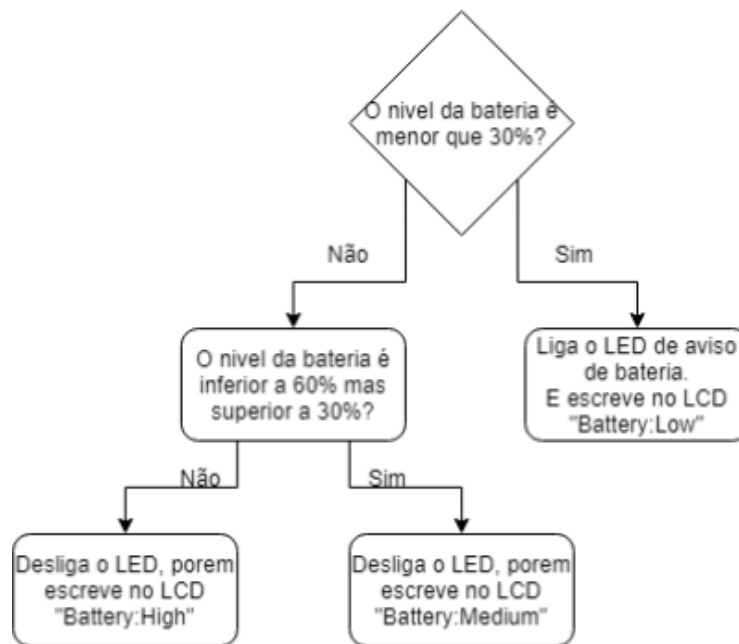


Figura 10 – Esquema relacionado ao nível de bateria

Por fim, acerca do atuador da luz de leitura, esta é única que é controlada exclusivamente fora do cenário CPT, em que através de um script em Python, ao carregar em 1 ou 0, vai ligar ou desligar a luz, respetivamente (enviando automaticamente essa informação para a API) ou, através da dashboard, num botão disponibilizado no website, carregando no botão para ligar/desligar a luz (que também envia automaticamente esse informação para a API). (figura 11)

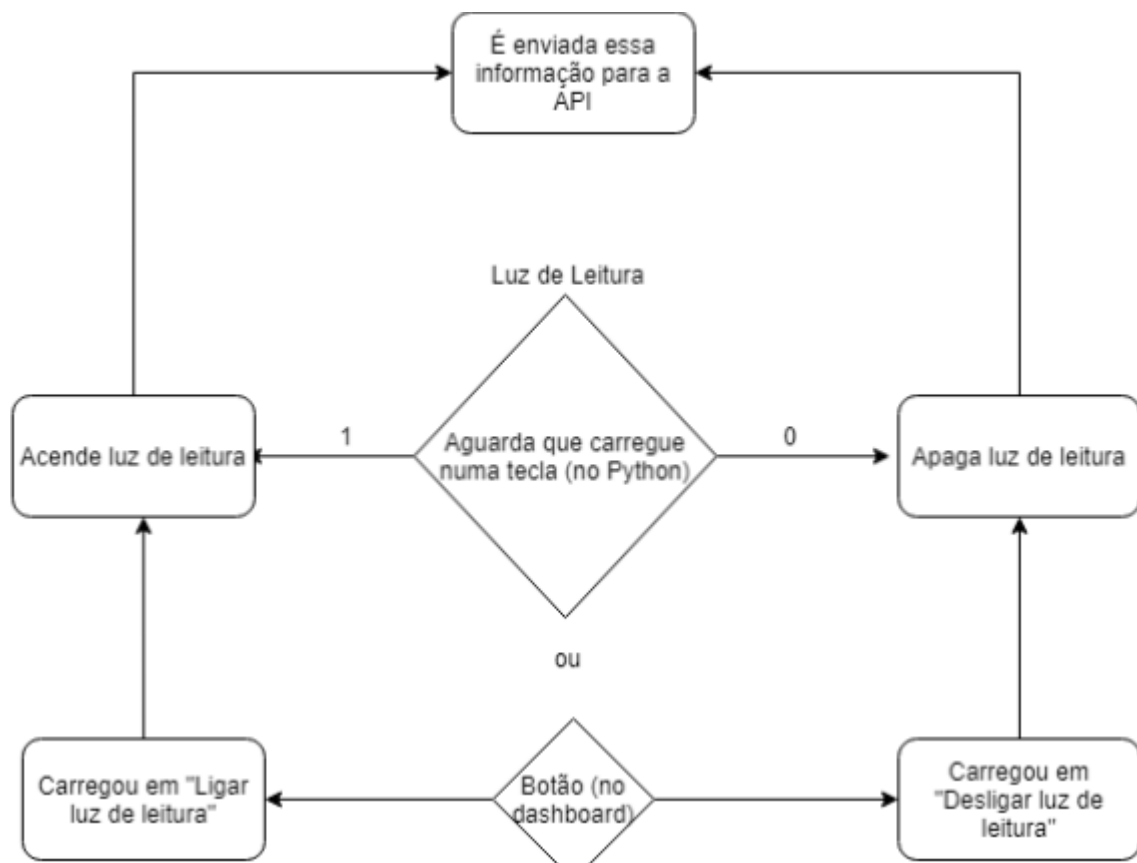


Figura 11 – Esquema relacionado a luz de leitura

## 4. Cenário de Teste

Começando pelo cenário CPT: usamos um “servidor IoT” (Server IoT), com o IP 192.168.0.2. Este usa, para se conectar a outros dispositivos o username: cisco, e password: cisco. Permite então, criar condições para alguns equipamentos utilizados. Optámos por criar uma ventoinha adicional em que, dependendo da temperatura, liga a “ventoinha server” (Fan), que tem o IP 192.168.0.18, que ligámos também ao Access Point (este tem como username “AllElectric”, e como password “ae123456”). Adicionalmente, criámos condições para que quando fosse detetado o “cartão correto” (RFID card) no “leitor de cartões” (RFID Reader), destrancasse as portas “Porta Esquerda” e “Porta Direita”. Para isso acontecer, também conectámos as portas ao “servidor IoT”, tendo a “Porta Esquerda” o IP 192.168.0.34 e “Porta Direita” 192.168.0.32, estando também conectadas ao Access Point.

As portas “Porta Esquerda” e “Porta Direita” estão ambas conectadas ao microcontrolador SBC-PT “Mostrar Estado Portas”, de IP 192.168.0.7 (estando também este conectado ao Access Point, recebendo a informação do estado das portas pela API), por portas digitais, D0 em que através da função digitalWrite() enviam o estado da porta para as portas D1 e D2. Adicionalmente, a “Porta Direita” tem ainda mais uma conexão digital, a sua porta D4, conectada ao SBC-PT na porta D2, que envia a informação se as portas estão trancadas ou não. Ainda com este SBC-PT temos o LED “luz portas abertas”, em que através de portas digitais, o SBC-PT através de um digitalWrite() indica ao LED para se acender, quando recebe a informação que as portas estão abertas. Adicionalmente, através de customWrite(), este SBC-PT envia, da sua porta D0 para a porta D0 de um LCD “LCD portas” a informação sobre portas abertas (2 ou 0). No MCU-PT “Alterar estado portas”, de IP 192.168.0.8 também este conectado ao Access Point, está conectado a 2 push buttons, “ABRIR PORTAS” e “FECHAR PORTAS”, através de ligações digitais, conectando-se pelo D1 e pelo D0 respetivamente, que através de digitalRead() lê se os botões foram pressionados ou não, enviando essa informação para a API para abrir ou fechar as portas.

Em relação ao “MCU sensores genéricos” (MCU-PT), com o IP 192.168.0.22, está também conectado ao Access Point, enviando lhe informações que recebe de 2 sensores genéricos (Generic Environment Sensor), ligados por portas analógicas: ao “WhiteNoise sensor” pela porta A0 e o “CO2 sensor” pela porta A1. Estas informações vão ser recolhidas pelo SBC-PT “Atuadores2”, de IP 192.168.0.17, também conectado ao Access Point, que

vai buscar à API as informações anteriormente referidas para poder alterar os atuadores a que está conectada: “Tejadilho Resposta ao White Noise” (simulada aqui por uma Door), pela porta digital D2, em que realiza um `digitalWrite(D2,HIGH)` quando recebe a informação que o ruído branco de ambiente ultrapassou os 15%, e LOW quando é inferior a esse valor; “Luz Aviso CO2” (Light), pela porta digital D1, em que a acende com diferentes graus de intensidade consoante as condições já referidas sobre os níveis de CO2 ambiente; “Luz Leitura” (LED), pela porta digital D0, em que a acende consoante as informações que lhe foram enviadas pela API.

Em relação aos principais atuadores escolhidos, ligados ao “Atuadores1” (um SBC-PT) com o IP 192.168.0.12, que está ligado via wireless ao Access Point e vai recolher informações à API, conectámos: uma Webcam (“Camera Sensor Movimento”), pela porta D1, em que o microcontrolador realiza um `customWrite(1,1)` quando recebe a informação da API que existe algum objeto próximo do carro e `customWrite(1,0)` em caso contrário; “farol esquerdo” e “farol direito”, conectados às portas D3 e D4 respetivamente, em que estes vão ser ligados quando o microcontrolador recebe da API a informação que a luminosidade ambiente está baixa; uma “Buzina (Porta Aberta)” (um speaker), pela porta D7 à porta A0, em que o liga através de um `analogWrite()` quando existem portas abertas, visto que pelo script em Python, definimos que quando as portas estivessem abertas, que se emitisse um aviso sonoro; um “LED aviso bateria” (LED), pela porta D2, em que o microcontrolador liga este atuador, quando recebe a informação da API que os níveis de bateria são inferiores a 30%, através de um `digitalWrite()`, atualizando na API o estado deste LED; uma “ventoinha” (Fan), pela porta D6, em que quando o microcontrolador recebe a informação pela API que a temperatura excede os 15°C, liga a ventoinha e atualiza este estado da ventoinha na API também; por fim o LCD “LCD avisos gerais”, em que atualiza a informação que este mostra através da função `customWrite()`.

Por fim, acerca do cenário CPT, ainda temos o MCU-PT “Sensores”, de IP 192.168.0.9, conectado via wireless ao Access Point. Este microcontrolador vai atualizar, na API, os valores dos diferentes sensores a que está conectado, nomeadamente: “sensor luz” (Photo Sensor), que se conecta à porta digital D2 do microcontrolador, enviando a informação de luminosidade baixa (quando é inferior a 20% no ambiente) ou alta, caso contrário; “humidade” (Humidity Sensor), ligado à porta analógica A1 do microcontrolador, em que este vai receber em percentagem o valor de humidade do ar, através de um `analogRead()`; “sensor temperatura” (Temperature Sensor) que envia a informação para o microcontrolador,



da sua porta A0 para a porta A0 do MCU-PT, enviando os valores de temperatura ambiente detetados; “sensor movimento” (Motion Sensor), que conecta-se ao microcontrolador na porta digital D5, e envia a informação que existe movimento quando se faz ALT+left click em cima do sensor, simulando um objeto nas proximidades.

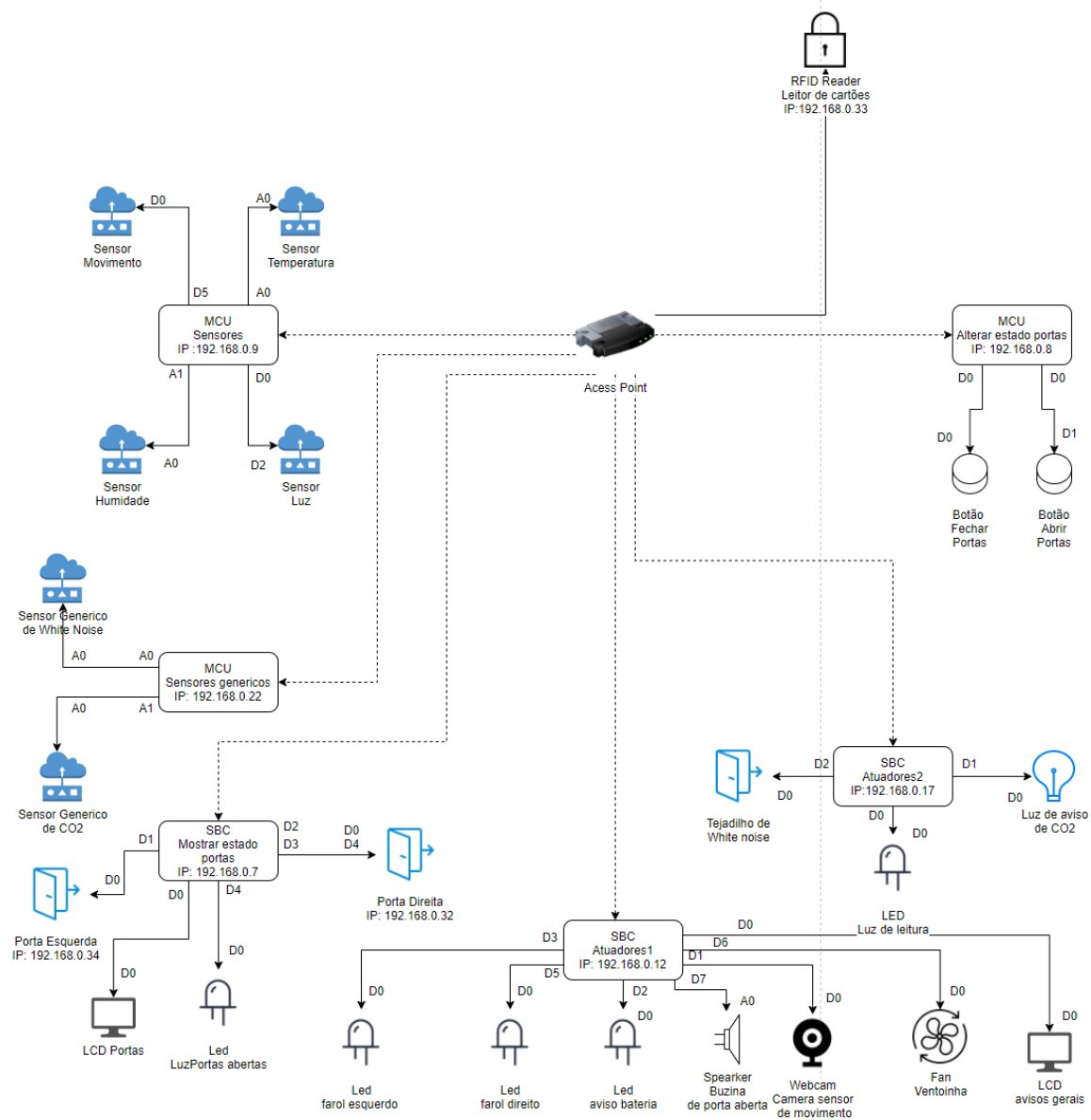


Figura 12 – Esquema do cenário de teste

## 5. Resultados obtidos

No cenário de ambiente do CPT configurámos algumas variáveis de ambiente tais como o “White Noise”, a “Ambient Temperature”, o “Sunlight” e o “CO2”. Optámos por colocar em todas estas um nível ascendente de valores a partir de cerca das 10h, sendo o pique por volta das 12h e descendo gradualmente até por volta das 19h, pois com o nosso tema, colocámos o cenário em que o nosso carro se estivesse num ambiente de cidade, em que acontece tudo por entre estas horas e há mais barulho e poluição, e consequentemente também um aumento dos níveis de temperatura.

Os valores de temperatura estão a ser enviados pelo MCU para a API em intervalos de 2,5 segundos. Um SBC está a pedir informações, de 3 em 3 segundos, à API, sendo que quando a temperatura aumenta dos 15°C, liga a ventoinha e mostra a informação que existe uma temperatura elevada no LCD. O estado da ventoinha (ligado ou desligado) é também enviado pelo SBC, de 3 em 3 segundos, para a API.

Os valores de luminosidade (Sunlight) estão também a ser enviados pelo MCU para a API em intervalos de 2,5 segundos. O mesmo SBC referido anteriormente está a pedir essa informação, de 3 em 3 segundos, à API, e quando deteta que os níveis de luminosidade estão baixos (inferiores a 20% - detetados pelo próprio sensor de luminosidade e enviados para o SBC em modo digital 0), são ligados os faróis do carro, sendo que envia também para a API a informação do estado dos faróis do carro (se eles estão ligados ou desligados).

Os valores de ruído branco (White Noise) são captados por um sensor genérico e são enviados a cada segundo pelo MCU para a API. Um SBC capta essa informação, com pedidos à API de cerca de 3 em 3 segundos, e quando os níveis de ruído ultrapassam os 15% é fechado o tejadilho, que caso contrário, estará aberto. A informação sobre o estado do tejadilho é também atualizada através de envios para a API através de um SBC no cenário CPT.

Os valores de CO2 são captados por um sensor genérico e enviados pelo MCU para a API a cada segundo. Também o mesmo SBC do ruído branco pede à API em intervalos de 3 segundos essa informação, que caso os valores de CO2 ultrapassem os 20%, é ligada uma luz de aviso de níveis elevados de CO2, sendo que essa alteração é também enviada para a API para atualizar o estado da luz.

Em relação aos valores da bateria, o script em Python `percentagemBateria.py` está a obter, de 10 em 10 segundos, a informação dos níveis da bateria (que têm que ser de um computador portátil) e envia esse valor à API. No cenário CPT, colocámos um SBC, que vai buscar de 3 em 3 segundos os valores da bateria e quando os níveis de bateria são inferiores a 30%, no LCD é colocada a informação “BATTERY : LOW”, e acende também uma luz de aviso de bateria baixa, um LED, em que envia para a API a informação que o LED está aceso. Caso seja os valores de bateria sejam superiores a 30%, o LED está apagado (essa informação é enviada para a API), sendo que no cenário CPT ainda fazemos a distinção que caso os níveis de bateria estejam entre os 30% e os 60%, é colocado no LCD a informação “BATTERY : MEDIUM”, e por fim caso seja superior a 60%, é colocado a informação “BATTERY : HIGH”.

Altera-se ainda o estado das portas. Usando o RFID card e o RFID reader, altera-se o “lockState” das portas caso o cartão seja o correto, isto é, se as portas estão trancadas ou não. Caso não seja inserido o cartão correto, não destranca as portas. Foi possível implementar esta funcionalidade pois, usando no CPT a Porta Direita, esta vai atualizando para o SBC se está ou não trancada, que por sua vez vai enviando e atualizando para a API se as portas estão ou não trancadas. Só é possível abrir as portas, tanto no CPT como no website, caso as portas estejam destrancadas (fazem sempre primeiro um pedido à API para verificar esta situação – no CPT através do MCU dos botões de abrir e fechar portas, no website através de um `file_get_contents`). Depois, tanto o dashboard com um botão (só aparece este botão para abrir a porta caso estas estejam destrancadas, ou, se for para fechar a porta, o botão aparece sempre), no ficheiro `atuadores.php` (é enviada a informação que o estado das portas foi alterado para a API instantaneamente) como o cenário CPT, através de dois botões, um para abrir e outro para fechar, estando estes ligados a um MCU e sendo enviada essa informação para a API a cada segundo, conseguem alterar o estado das portas (abertas/fechadas). Se as portas estiverem abertas no momento que se trancou as portas, ainda é possível fechá-las. Ainda no cenário CPT, um SBC vai buscar o valor se as portas estão abertas, sendo que se estiverem, envia também para a API a informação se a luz de portas abertas está ligada ou caso contrário envia a informação para a API que a luz está apagada. Esta informação é demonstrada por um LED no CPT, ligado ao mesmo SBC, que acende quando as portas estão abertas, e que está apagado quando estas estão fechadas. Por outro lado, o script em Python `tocaPorta.py` (**Nota:** para conseguir executar o script `tocaPorta.py` corretamente é necessário que o ficheiro “Alarm.wav” esteja na diretoria

“C:/pythonproj/ ” )está a pedir, de 3 em 3 segundos, à API a informação se as portas estão ou não abertas. Caso as portas estejam abertas, emite um alerta sonoro, até as portas ficarem fechadas. A informação que este som sonoro está a ser emitido ou não está a ser enviada para a API. Ainda no cenário CPT, temos um buzzer conectado a um SBC, em que este vai buscar a informação à API, de 3 em 3 segundos, sobre o som sonoro (se está a ser emitido ou não), sendo a animação também mostrada no buzzer do CPT.

Em relação aos valores da humidade, estes são captados pelo sensor de humidade e enviados para a API através do MCU de 2,5 em 2,5 segundos. Depois um SBC faz um pedido à API, obtém os valores de humidade e mostra-os no LCD.

Sobre a luz de leitura, através do script em Python luzLeitura.py, envia automaticamente à API sempre que se desliga a luz (carregando no 0) ou sempre que se liga a luz (carregando no 1). Esse valor, além de ser mostrado no website, é demonstrado no cenário CPT através de um pedido à API por um SBC, em que depois consoante o valor lido (1 ou 0), acende a luz ou não.

Por fim, em relação aos valores da proximidade, no cenário CPT ao ter o MCU a correr e a fazer ALT+ENTER no sensor de movimento, este deteta que houve movimento, fornecendo a informação ao MCU, sendo que este envia a informação “Objeto próximo” para a API, ou caso não seja detetado movimento, envia também para a API essa informação. Em caso contrário, mantém a webcam desligada. Simultaneamente, o script em Python capturaWebcamOpenCV.py vai buscar à API o valor do estado atual das proximidades, e se esse valor for “Objeto próximo” tira uma foto, fazendo upload para a pasta uploads, esperando 5 segundos para avaliar outra vez o estado da proximidade. O script em Python também envia para a API a informação sobre o estado da camera, enviando o valor “1” caso esteja ligada, e o valor “0” caso esteja desligada. Ainda no cenário CPT, um SBC faz um pedido à API sobre o valor atual da camera, em que liga a webcam do CPT caso a informação obtida na API seja “1”, ou desliga a camera caso o valor obtido seja “0”.

De notar ainda que todas as alterações efetuadas em todos os sensores e atuadores anteriormente referidos são recolhidas nos ficheiros da API pelos ficheiros que constituem o website, sendo que todas as informações são constantemente demonstradas nos respetivos locais, tal como as imagens também são recolhidas e descarregadas no website.

Em termos de funcionalidades opcionais, realizámos diversos tópicos, nomeadamente a autenticação dos pedidos HTTP na API, ao necessitar de ser enviado um parâmetro de segurança adicional quando se realiza um pedido POST para a API; utilizámos um total de 18 sensores e atuadores, tendo mais 8 dos que eram obrigatórios; apresentamos, na dashboard “Sensores” uma parte gráfica que representa alguns dos sensores que considerámos que teriam relevância apresentar; implementámos a plataforma web de forma a que tenha 3 utilizadores diferentes, sendo eles “Micael” (utilizador com mais privilégios, que tem acesso à totalidade das funcionalidades do carro), “Andre” (utilizador que consegue abrir e fechar portas e ligar e desligar a luz de leitura, mas no entanto não consegue aceder nem ao histórico de imagens, nem ao painel de controlo) e por fim o utilizador “carro”, o que tem menos privilégios, conseguindo apenas visualizar os dados dos diferentes sensores e atuadores, não conseguindo alterar estados de portas nem da luz de leitura, nem aceder a nenhum tipo de histórico; utilizámos uma biblioteca Python para interagir com o hardware do PC, da Psutil-master, em que deteta o estado da bateria do computador, sendo necessário usufruir de um computador portátil para poder ver os diferentes valores; criámos sensores genéricos no CPT, sendo estes para detetar o ruído branco e os níveis de CO<sub>2</sub>, o que nos pareceu pertinente visto que idealizamos que o carro elétrico é um carro de cidade populosa, correndo o risco de ser exposto a estes 2 fatores de poluição; criámos uma página no dashboard que permite aceder a um histórico de imagens (imagens estas que são tiradas quando o sensor de movimento do carro deteta um objeto próximo); temos ainda um controlo das imagens recebidas, com a funcionalidade adicional que colocámos ao fazer uma página de perfil para cada utilizador “logado”, sendo que cada perfil pode dar upload de uma imagem diferente (utilizamos assim o ficheiro uploadFoto.php).

## 6. Conclusão

Concluímos assim que julgamos ter conseguido realizar os objetivos que nos tínhamos proposto inicialmente, conseguindo fazer com que toda a informação relacionada com o carro fosse atualizada automaticamente, tendo uma resposta também imediata. Cumprimos com a ideia de ter um carro que fosse exclusivo para certo utilizador, que só mediante uma identificação prévia o poderia aceder. Ainda no dashboard, ao realizar a funcionalidade de ter 3 utilizadores com diferentes privilégios, demonstramos exatamente isso, visto que há um utilizador que apenas pode ver os estados dos sensores e atuadores, e um utilizador principal que tem acesso total a tudo o que o carro oferece.

Julgamos ter tido um método correto e eficaz na resolução deste trabalho, ao aproveitar o que nos foi lecionado ao longo do semestre. Utilizámos o CPT para detetar a maioria dos sensores (excetuando o sensor do nível da bateria que usa equipamento real) e apresentar o estado dos atuadores, para poder ter uma representação gráfica. Usámos ainda alguns scripts em Python e ficheiros com PHP, HTML, CSS e algum JavaScript para poder ter a dashboard funcional e a mostrar a informação pretendida.

Um dos pontos que gostaríamos ainda de ter melhorado seria uma implementação de um sistema de reconhecimento facial, que iria ser uma alternativa adicional à chave de identificação (o RFID card) que implementámos, também permitiria destrancar o veículo, sendo que a camera iria comparar a foto tirada com as imagem das pessoas previamente colocadas como “autorizadas” do veículo, sendo que se não reconhecesse a cara iria mandar um aviso para o dono do mesmo passado 3 ou mais tentativas, guardando num histórico todas as imagens captadas.

De notar que, apesar de julgarmos que o resultado final se apresenta como tínhamos idealizado no início do projeto e de como era pretendido para este trabalho, lamentamos que não tenhamos conseguido testar a 100% e em todas as situações que queríamos pois, mesmo após várias tentativas, notámos que pelo menos nas nossas máquinas, o CPT não se encontrou totalmente funcional a nível da definição das variáveis de ambiente, de alteração dos seus valores para certas horas (por exemplo, os valores da temperatura ambiente muitas vezes chegavam a mais de 90°C apesar de nunca termos colocado esses valores, sendo alguns destes valores muito irregulares).

## 7. Bibliografia

Laboratórios práticos da Unidade Curricular de Tecnologias de Internet

Capítulo 4 – JavaScript & Python da Unidade Curricular de Tecnologias de Internet

<https://stackoverflow.com/questions/29110399/how-to-edit-the-contents-of-a-file-on-button-click-php>

[http://www.uniformserver.com/ZeroXI\\_documentation/php.html](http://www.uniformserver.com/ZeroXI_documentation/php.html)

[https://www.w3schools.com/html/html\\_iframe.asp](https://www.w3schools.com/html/html_iframe.asp)

<https://forum.webflow.com/t/force-iframe-links-to-stay-on-the-same-page-within-the-same-iframe/87048>

[https://www.w3schools.com/php/func\\_filesystem\\_glob.asp](https://www.w3schools.com/php/func_filesystem_glob.asp)

<https://stackoverflow.com/questions/34564458/limiting-php-form-to-only-upload-gifs-jpeg-or-pngs-under-a-certain-size>

<https://stackoverflow.com/questions/12801370/count-how-many-files-in-directory-php>

<https://stackoverflow.com/questions/16380394/getting-battery-capacity-in-windows-with-python>

<https://www.youtube.com/watch?v=oXcBN31IMt4&ab>

## 8. Anexos

### 8.1.Código do Microcontrolador – Sensores

```
// Definir variaveis

var url = 'http://127.0.0.1/projeto/api/api.php';

var motion = 5;

var photo = 2;


// Calculo da temperatura para obter o valor em percentagem

function lerTemperatura(slot){

    var aux = analogRead(slot);

    var temp = ((aux/1023)*200)-100;

    temp = temp.toFixed(2);

    return temp;

}

// Calculo da humidade para obter o valor em percentagem

function lerHumidade(slot){

    var au = analogRead(slot);

    var humidade = (au*100)/1023;

    humidade = humidade.toFixed(2);

    return humidade;

}

// Função para mostrar a hora e data

function dataHora(){

    var dataAUX = new Date().toISOString().split("T");

    var data = dataAUX[0];

    var hora = dataAUX[1].split('.')[0];

    var dataHor = data + " " + hora;

    Serial.println(dataHor);

    return dataHor;

}

// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas

function setup(){

    pinMode(A0,INPUT);
```



```
pinMode(A1,INPUT);

pinMode(photo,INPUT);

pinMode(motion,INPUT);

}

function loop(){

    // Lê o valor da humidade e faz um print para a consola

    // Manda os valores obtidos da humidade e hora para a API

    var humidade = lerHumidade(A1);

    Serial.print("Humidade: ");

    Serial.println(humidade);

    var horaHumidade = dataHora();

    var valoresHumidade = { 'nome': 'humidade', 'valor': humidade, 'hora': horaHumidade, 'pass': "ae123" };

    RealHttpClient.post(url, valoresHumidade, function(status, data){

        Serial.println("resposta: " + data);

    });

    // Lê o valor da temperatura e faz um print para a consola

    // Manda os valores obtidos da temperatura e hora para a API

    var temp = lerTemperatura(A0);

    Serial.println(temp);

    var hora = dataHora();

    var valoresTemp = { 'nome': 'temperatura', 'valor': temp, 'hora': hora, 'pass': "ae123" };

    RealHttpClient.post(url, valoresTemp, function(status, data){

        Serial.println("resposta: " + data);

    });

    delay(500);

    // Se o sensor de movimento detetar movimento vai escrever na consola que há movimento

    // E manda a informação para a API

    if(digitalRead(motion)==HIGH){

        Serial.println("Há movimento");

        hora = dataHora();
```

```
var valores = { 'nome': 'proximidade', 'valor': "Objeto Próximo", 'hora': hora, 'pass': "ae123" };

    RealHTTPClient.post(url, valores, function(status, data){

    Serial.println("resposta: " + data);

        }

    );

} else {

    Serial.println("Não há movimento");

    hora = dataHora();

    valores = { 'nome': 'proximidade', 'valor': "Nenhum objeto próximo", 'hora': hora, 'pass': "ae123" };

    RealHTTPClient.post(url, valores, function(status, data){

    Serial.println("resposta: " + data);

        }

    );

}

// Se o sensor de luz detetar que há luz, escreve na consola há luz

// E manda a informação para a API

if(digitalRead(photo)==HIGH){

    Serial.println("Há luz");

    var hora3 = dataHora();

    var valoresLuz = { 'nome': 'luminosidade', 'valor': "Elevada", 'hora': hora3, 'pass': "ae123" };

    RealHTTPClient.post(url, valoresLuz, function(status, data){

    Serial.println("resposta: " + data);

        }

    );

} else {

    Serial.println("Não há luz");

    hora3 = dataHora();

    valoresLuz = { 'nome': 'luminosidade', 'valor': "Baixa", 'hora': hora3, 'pass': "ae123" };

    RealHTTPClient.post(url, valoresLuz, function(status, data){

    Serial.println("resposta: " + data);

        }

    );

}

delay(2000);
```

```
}
```

## 8.2.Código do Microcontrolador – Sensores genéricos

```
// Definir variaveis

var url = 'http://127.0.0.1/projeto/api/api.php';

// Função para mostrar a hora e data

function dataHora(){

    var dataAUX = new Date().toISOString().split('T');

    var data = dataAUX[0];

    var hora = dataAUX[1].split('.')[0];

    var dataHor = data + " " + hora;

    Serial.println(dataHor);

    return dataHor;

}

// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas

function setup(){

    pinMode(A0,INPUT);

    pinMode(A1,INPUT);

}

// Função em modo loop que está continuamente

// a calcular os valores obtidos dos sensores para converter para percentagem e mandar para a API

function loop(){

    var valueNoise= analogRead(A0);

    var valorLidoNoise = valueNoise*105/1023;

    valorLidoNoise = valorLidoNoise.toFixed(2);

    var hora = dataHora();

    var valores = { 'nome': 'whiteNoise', 'valor': valorLidoNoise, 'hora':hora, 'pass':"ae123"};

    RealHttpClient.post(url, valores, function(status, data){

        Serial.println("resposta: " + data);

    }

    );

}
```

```
Serial.print("Valor White Noise: ");
Serial.println(valorLidoNoise);

var valueCO2= analogRead(A1);
var valorLidoCO2 = valueCO2*105/1023;
valorLidoCO2 = valorLidoCO2.toFixed(2);
var horaCO2 = dataHora();
var valoresCO2 = {'nome': 'CO2', 'valor': valorLidoCO2, 'hora':horaCO2, 'pass':"ae123"};
RealHTTPClient.post(url, valoresCO2, function(status, data){
  Serial.println("resposta: " + data);
}
);

Serial.print("Valor CO2: ");
Serial.println(valorLidoCO2);

delay(1000);

}
```

### 8.3.Código do Microcontrolador – Alterar estado portas

```
// Definir variaveis
var url = 'http://127.0.0.1/projeto/api/api.php';
var urlTranca = "http://127.0.0.1/projeto/api/api.php?nome=estado";
var botaoFechar=0, botaoAbrir=1;
var estado_tranca, valor_tranca;

// Função para mostrar a hora e data
function dataHora(){
  var dataAUX = new Date().toISOString().split("T");
  var data = dataAUX[0];
  var hora = dataAUX[1].split('.')[0];
  var dataHor = data + " " + hora;
  Serial.println(dataHor);

  return dataHor;
}
```

```
}

// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas

function setup(){

  pinMode(botaoFechar,INPUT);

  pinMode(botaoAbrir,INPUT);

}

// Manda o estado da porta para a API

function loop(){

  RealHttpClient.get(urlTranca,function(status,data){

    estado_tranca=status;

    valor_tranca=data;

    delay(500);

  });

  // Se o botão de fechar for pressionado

  // fecha as portas e manda a informação para a API

  if(digitalRead(botaoFechar)=="1023"){

    var hora = dataHora();

    var valores = {'nome': 'portas', 'valor': 0, 'hora':hora, 'pass': 'ae123'};

    RealHttpClient.post(url, valores, function(status, data){

      Serial.println("resposta: " + data);

    }

  );

}

// Caso haja tentativa de abrir as portas com elas trancadas

if(valor_tranca == "1"){

  Serial.println("Portas trancadas - não é possível abrir as portas");

// Caso não estejam trancadas, vai abrir as portas e mandar a informação para a API

}else{

  if(digitalRead(botaoAbrir)=="1023"){

    hora = dataHora();
```

```
        valores = { 'nome': 'portas', 'valor': 2, 'hora': hora, 'pass': "ae123" };

        RealHttpClient.post(url, valores, function(status, data){

            Serial.println("resposta: " + data);

        }

    );

    }

}

}

delay(1000);

}
```

## 8.4.Código do SBC – Mostrar estado Portas

```
// Definir variaveis

var urlPortas = "http://127.0.0.1/projeto/api/api.php?nome=portas";

var url = 'http://127.0.0.1/projeto/api/api.php';

var LCD = 0;

var estado_portas, valor_portas;

var portaDireita=2;

var portaEsquerda=1;

var trancado=3;

var led=4;


// Função para mostrar a hora e data

function dataHora(){

    var dataAUX = new Date().toISOString().split('T');

    var data = dataAUX[0];

    var hora = dataAUX[1].split('.')[0];

    var dataHor = data + " " + hora;

    Serial.println(dataHor);

    return dataHor;

}


// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas

function setup(){

    pinMode(LCD,OUTPUT);

    pinMode(portaDireita,OUTPUT);

    pinMode(portaEsquerda,OUTPUT);

}
```

```
pinMode(led,OUTPUT);

pinMode(trancado,INPUT);

}

// Vai verificar qual o estado das portas

// Se o estado for "destrancado" vai mandar essa informação para a API

function loop(){

  var tranca = digitalRead(trancado);

  Serial.print("Porta está trancada? -> ");

  Serial.println(tranca);

  if(tranca == "1023"){

    customWrite(portaDireita,0);

    customWrite(portaEsquerda,0);

    var horaTranca = dataHora();

    var valoresTranca = { 'nome': 'estado', 'valor': 1, 'hora':horaTranca, 'pass': 'ae123'};

    RealHttpClient.post(url, valoresTranca, function(status, data){

      Serial.println("resposta: " + data);

    }

  );

} else{

  // Caso o estado seja "trancado" vai destrancar e mandar a informação para a API

  horaTranca = dataHora();

  valoresTranca = { 'nome': 'estado', 'valor': 0, 'hora':horaTranca, 'pass': 'ae123'};

  RealHttpClient.post(url, valoresTranca, function(status, data){

    Serial.println("resposta: " + data);

  }

);

}

RealHttpClient.get(urlPortas,function(status,data){

  estado_portas=status;

  valor_portas=data;

  delay(500);

});
```

// Se o valor o que recebeu da API foi que há portas abertas, vai abrir aqui as portas, e mandar a informação para a API que a luz das portas está ligada

```
if(valor_portas>0){

    customWrite(portaDireita,1);

    customWrite(portaEsquerda,1);

    var hora = dataHora();

    var valores = { 'nome': 'luzPortas', 'valor': 1, 'hora':hora, 'pass': 'ae123'};

    RealHTTPClient.post(url, valores, function(status, data){

        Serial.println("resposta: " + data);

    }

    );

    digitalWrite(led,HIGH);

// Caso não haja portas abertas, vai fechar e mandar a informação para a API

}else{

    customWrite(portaDireita,0);

    customWrite(portaEsquerda,0);

    hora = dataHora();

    valores = { 'nome': 'luzPortas', 'valor': 0, 'hora':hora, 'pass': 'ae123'};

    RealHTTPClient.post(url, valores, function(status, data){

        Serial.println("resposta: " + data);

    }

    );

    digitalWrite(led,LOW);

}

// Envia a informação de quantas portas estão abertas para o LCD

Serial.println(valor_portas);

customWrite(LCD,"Portas Abertas: ");

delay(1000);

customWrite(LCD,valor_portas);

delay(1000);

}
```



## 8.5. Código do SBC – Atuadores2

```
// Definir variaveis

var urlLuzLeitura = "http://127.0.0.1/projeto/api/api.php?nome=luzLeitura";
var urlWhiteNoise = "http://127.0.0.1/projeto/api/api.php?nome=whiteNoise";
var urlCO2 = "http://127.0.0.1/projeto/api/api.php?nome=CO2";
var urlPOST = 'http://127.0.0.1/projeto/api/api.php';
var luzCO2 = 1, tejadilho =2;

var estado_luz, valor_luz,estado_noise, valor_noise,estado_CO2, valor_CO2;


// Função para mostrar a hora e data
function dataHora(){

    var dataAUX = new Date().toISOString().split("T");

    var data = dataAUX[0];

    var hora = dataAUX[1].split('.')[0];

    var dataHor = data + " " + hora;

    Serial.println(dataHor);

    return dataHor;
}


// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas
function setup() {

    pinMode(0, OUTPUT);

    pinMode(luzCO2, OUTPUT);

    pinMode(tejadilho, OUTPUT);

}


// Vai buscar o valor da luz de leitura a API e guarda numa variavel
function loop() {

    RealHttpClient.get(urlLuzLeitura,function(status,data){

        estado_luz=status;

        valor_luz=data;

        delay(250);

    });

    Serial.println(valor_luz);

    // Se o valor obtido for 0, muda o estado da luz para LOW
```

```
if(valor_luz == "0"){
    digitalWrite(0,LOW);
}
else{
    // Se o for diferente de 0, muda o estado da luz para HIGH
    digitalWrite(0,HIGH);
}

// Vai buscar o valor do CO2 a API e guarda numa variavel
RealHTTPClient.get(urlCO2,function(status,data){
    estado_CO2=status;
    valor_CO2=data;
    delay(250);
});

// Se o valor do CO2 for superior a 45, vai mandar o valor 2 (que liga a luz no HIGH) para a API
if(valor_CO2 >45){
    var horaLuz = dataHora();
    var valoresLuz = { 'nome': 'luzCO2', 'valor': 2, 'hora':horaLuz, 'pass':"ae123"};
    RealHTTPClient.post(urlPOST, valoresLuz, function(status, data){
        Serial.println("resposta: " + data);
    }
    );
    customWrite(luzCO2,2);
}
else{
    //Se o valor do CO2 for superior a 20(mas inferior a 45), vai mandar a informação que a luz está acesa(1) (que liga a luz no
    LOW) para a API

    if(valor_CO2 >20){
        horaLuz = dataHora();
        valoresLuz = { 'nome': 'luzCO2', 'valor': 1, 'hora':horaLuz, 'pass':"ae123"};
        RealHTTPClient.post(urlPOST, valoresLuz, function(status, data){
            Serial.println("resposta: " + data);
        }
        );
        customWrite(luzCO2,1);
    }
    else{
        // Se o valor do CO2 não for superior a 20, vai mandar o valor 0 (que desliga a luz) para a API
        horaLuz = dataHora();
        valoresLuz = { 'nome': 'luzCO2', 'valor': 0, 'hora':horaLuz, 'pass':"ae123"};
```

```

        RealHttpClient.post(urlPOST, valoresLuz, function(status, data){

            Serial.println("resposta: " + data);

        }

    );

    customWrite(luzCO2,0);

}

}

// Imprime o valor do CO2 para a consola
Serial.println(valor_CO2);

// Vai buscar o valor do sensor de ruido branco a API e guarda numa variavel
RealHttpClient.get(urlWhiteNoise,function(status,data){

    estado_noise=status;

    valor_noise=data;

    delay(250);

});

// Imprime o valor do sensor de Ruido branco
Serial.println(valor_noise);

// Se o valor do sensor de ruido for superior a 15
// Vai mandar fechar o tejadilho
if(valor_noise>15){

    var horaTejadilho = dataHora();

    var valoresTejadilho = {'nome': 'tejadilho', 'valor': 1, 'hora':horaTejadilho, 'pass':"ae123"};

    RealHttpClient.post(urlPOST, valoresTejadilho, function(status, data){

        Serial.println("resposta: " + data);

    }

    );

    customWrite(tejadilho,0);

}else{

    // Se for inferior a 15, vai mandar abrir o tejadilho

    horaTejadilho = dataHora();

    valoresTejadilho = {'nome': 'tejadilho', 'valor': 0, 'hora':horaTejadilho, 'pass':"ae123"};

    RealHttpClient.post(urlPOST, valoresTejadilho, function(status, data){

        Serial.println("resposta: " + data);

    }

}

```

```
);  
  
    customWrite(tejadilho,1);  
  
}  
  
  
  
    delay(2500);  
  
}
```

## 8.6.Código do SBC – Atuadores1

```
// Definir variaveis  
  
var urlTemperatura = "http://127.0.0.1/projeto/api/api.php?nome=temperatura";  
var urlBateria = "http://127.0.0.1/projeto/api/api.php?nome=bateria";  
var urlCamera = "http://127.0.0.1/projeto/api/api.php?nome=camera";  
var urlLuminosidade = "http://127.0.0.1/projeto/api/api.php?nome=luminosidade";  
var urlPortas = "http://127.0.0.1/projeto/api/api.php?nome=portas";  
var urlPOST = 'http://127.0.0.1/projeto/api/api.php';  
var urlHumidade = "http://127.0.0.1/projeto/api/api.php?nome=humidade";  
var LCD=0;  
var webcam = 1;  
var bateria = 2;  
var LEDbateria=4;  
var LEDright=5,LEDleft=3;  
var FAN=6;  
var buzina=7;  
  
var valor_bat, valor_temp, valor_prox, valor_luz;  
var estado_bat,estado_temp , estado_prox, estado_luz;  
var estado_humidade, estado_portas;  
var valor_humidade, valor_portas;  
  
  
  
// Função para mostrar a hora e data  
function dataHora(){  
    var dataAUX = new Date().toISOString().split("T");  
    var data = dataAUX[0];  
    var hora = dataAUX[1].split('.')[0];
```

```
    var dataHor = data + " " + hora;

    Serial.println(dataHor);

    return dataHor;
}

// Setup dos sensores como INPUT/OUTPUTs e as respetivas portas
function setup(){

    pinMode(LEDbateria,OUTPUT);

    pinMode(LEDright,OUTPUT);

    pinMode(LEDleft,OUTPUT);

    pinMode(FAN,OUTPUT);

    pinMode(LCD,OUTPUT);

    pinMode(buzina,OUTPUT);

    pinMode(webcam,OUTPUT);

}

function loop(){

    // Vai buscar o valor da temperatura a API e guarda numa variavel

    Serial.println("Read temperatura from server");

    RealHTTPClient.get(urlTemperatura,function(status,data){

        estado_temp=status;

        valor_temp=data;

        delay(500);

    });

    // Imprime o valor da temperatura na consola

    Serial.println("-Temperatura- \nResposta: " + estado_temp);

    Serial.println("Valor: " + valor_temp);

    // Vai buscar o valor da bateria a API e guarda numa variavel

    Serial.println("Read bateria from server");

    RealHTTPClient.get(urlBateria,function(status,data){

        estado_bat=status;

        valor_bat=data;

        delay(500);
```

```
});

// Imprime o valor da bateria na consola

Serial.println("Valor BATERIA: " + valor_bat);

// Vai buscar o valor da camera a API e guarda numa variavel

Serial.println("Read camera from server");

RealHTTPClient.get(urlCamera,function(status,data){

    estado_prox=status;

    valor_prox=data;

    delay(500);

});

// Vai imprimir o valor da camera, se está ligada ou desligada

Serial.println("Camera: ");

if(valor_prox == "0"){

    customWrite(webcam,0);

}else{

    customWrite(webcam,1);

}

// Vai buscar o valor das a API e guarda numa variavel

Serial.println("Read portas from server");

RealHTTPClient.get(urlPortas,function(status,data){

    estado_portas=status;

    valor_portas=data;

    delay(500);

});

// Vai imprimir o valor das portas, se estão abertas ou fechadas

// Se as portas estiverem abertas, vai ligar a buzina.

Serial.println("Portas: ");

if(valor_portas == "0"){

    analogWrite(buzina,0);

}else{
```

```
        analogWrite(buzina,255);
    }

    // Vai buscar o valor da luz a API e guarda numa variavel
    Serial.println("Read luminosidade from server");
    RealHTTPClient.get(urlLuminosidade,function(status,data){
        estado_luz=status;
        valor_luz=data;
        delay(500);
    });
    // Se o valor da luz for elevada, vai desligar os LEDs
    if(valor_luz == "Elevada"){
        digitalWrite(LEDleft,LOW);
        digitalWrite(LEDright,LOW);
        var horaLuz = dataHora();
        var valoresLuz = {'nome': 'farois', 'valor': 0, 'hora':horaLuz, 'pass':"ae123"};
        RealHTTPClient.post(urlPOST, valoresLuz, function(status, data){
        }
    );
    // Senão for elevada, liga as luzes
    }else{
        digitalWrite(LEDleft,HIGH);
        digitalWrite(LEDright,HIGH);
        horaLuz = dataHora();
        valoresLuz = {'nome': 'farois', 'valor': 1, 'hora':horaLuz, 'pass':"ae123"};
        RealHTTPClient.post(urlPOST, valoresLuz, function(status, data){
        }
    );
    }

    // Se o valor da temperatura for menor que 15, desliga a ventoinha e manda essa informação para o website
    if (valor_temp < 15){
        var horaTemp = dataHora();
        var valoresFan = {'nome': 'ventoinha', 'valor': 0, 'hora':horaTemp, 'pass':"ae123"};
```

```
RealHTTPClient.post(urlPOST, valoresFan, function(status, data){
    }
});

customWrite(FAN,"0");

customWrite(LCD,"FAN: OFF");

}else{

// Senão, liga a ventoinha e manda essa informação para o website

    horaTemp = dataHora();

    valoresFan = { 'nome': 'ventoinha', 'valor': 1, 'hora':horaTemp, 'pass':"ae123"};

    RealHTTPClient.post(urlPOST, valoresFan, function(status, data){

        }

    );

    customWrite(FAN,"2");

    customWrite(LCD,"FAN: ON");

}

delay(1000);

// Se o valor da bateria for menor que 30, liga a luz de aviso da bateria

// E escreve no LCD Battery: Low

if (valor_bat < 30){

    var horaLuzBat = dataHora();

    var valoresLuzBat = { 'nome': 'luzBateria', 'valor': 1, 'hora':horaLuzBat, 'pass':"ae123"};

    RealHTTPClient.post(urlPOST, valoresLuzBat, function(status, data){

        }

    );

    digitalWrite(LEDbateria,HIGH);

    customWrite(LCD,"BATTERY: LOW");

}else{

// Se o nivel da bateria for superior a 30%, mas inferior a 60% desliga a luz de aviso da bateria

// E escreve no LCD Battery: Medium

    if(valor_bat < 60){

        horaLuzBat = dataHora();

        valoresLuzBat = { 'nome': 'luzBateria', 'valor': 0, 'hora':horaLuzBat, 'pass':"ae123"};

        RealHTTPClient.post(urlPOST, valoresLuzBat, function(status, data){

            }

        );

    }

}
```



```

        digitalWrite(LEDbateria,LOW);

        customWrite(LCD,"BATTERY: MEDIUM");

    }else{

// Se o nivel de bateria for superior a 60%, desliga a luz de aviso da bateria

// E escreve no LCD Battery:High

        horaLuzBat = dataHora();

        valoresLuzBat = { 'nome': 'luzBateria', 'valor': 0, 'hora':horaLuzBat, 'pass':"ae123"};

        RealHTTPClient.post(urlPOST, valoresLuzBat, function(status, data){

            }

        );

        digitalWrite(LEDbateria,LOW);

        customWrite(LCD,"BATTERY: HIGH");

    }

}

delay(1000);

// Vai buscar o valor da humidade a API e guarda numa variavel

Serial.println("Read humidade from server");

RealHTTPClient.get(urlHumidade,function(status,data){

    estado_humidade=status;

    valor_humidade=data;

    delay(500);

});

// Se o valor da humidade for superior a 20% escreve no LCD Humidity:High

if(valor_humidade>20){

    customWrite(LCD,"HUMIDITY: HIGH");

}else{

// Se for inferior a 20% escreve Humidity:Low

    customWrite(LCD,"HUMIDITY: LOW");

}

delay(500);

}

```

## 8.7.Código do PHP Atuadores.php

```
<?php

//criação das variáveis para ir buscar os dados aos ficheiros


//para a ventoinha

$estado_ventoinha= file_get_contents("api/files/ventoinha/valor.txt");

$hora_ventoinha = file_get_contents("api/files/ventoinha/hora.txt");

$nome_ventoinha = file_get_contents("api/files/ventoinha/nome.txt");

//fim das variaveis para a ventoinha


//para a porta

$estado_portas = file_get_contents("api/files/portas/valor.txt");

$hora_portas = file_get_contents("api/files/portas/hora.txt");

$nome_portas = file_get_contents("api/files/portas/nome.txt");

//fim das variaveis para a porta


//para farois

$estado_farois = file_get_contents("api/files/farois/valor.txt");

$hora_farois = file_get_contents("api/files/farois/hora.txt");

$nome_farois = file_get_contents("api/files/farois/nome.txt");

//fim das variaveis para farois


//para a camera

$estado_camera = file_get_contents("api/files/camera/valor.txt");

$hora_camera = file_get_contents("api/files/camera/hora.txt");

$nome_camera = file_get_contents("api/files/camera/nome.txt");

//fim das variaveis para a camera


//para o buzzer

$estado_buzzer = file_get_contents("api/files/buzzer/valor.txt");

$hora_buzzer = file_get_contents("api/files/buzzer/hora.txt");

$nome_buzzer = file_get_contents("api/files/buzzer/nome.txt");

//fim das variaveis para o buzzer
```

```
$valor_luzLeitura = file_get_contents("api/files/luzLeitura/valor.txt");

$hora_luzLeitura = file_get_contents("api/files/luzLeitura/hora.txt");

$nome_luzLeitura = file_get_contents("api/files/luzLeitura/nome.txt");

$valor_luzCO2 = file_get_contents("api/files/luzCO2/valor.txt");

$hora_luzCO2 = file_get_contents("api/files/luzCO2/hora.txt");

$nome_luzCO2 = file_get_contents("api/files/luzCO2/nome.txt");

$valor_luzBateria = file_get_contents("api/files/luzBateria/valor.txt");

$hora_luzBateria = file_get_contents("api/files/luzBateria/hora.txt");

$nome_luzBateria = file_get_contents("api/files/luzBateria/nome.txt");

$valor_luzPortas = file_get_contents("api/files/luzPortas/valor.txt");

$hora_luzPortas = file_get_contents("api/files/luzPortas/hora.txt");

$nome_luzPortas = file_get_contents("api/files/luzPortas/nome.txt");

$valor_tejadilho = file_get_contents("api/files/tejadilho/valor.txt");

$hora_tejadilho = file_get_contents("api/files/tejadilho/hora.txt");

$nome_tejadilho = file_get_contents("api/files/tejadilho/nome.txt");

$valor_trancado = file_get_contents("api/files/estado/valor.txt");

$hora_trancado = file_get_contents("api/files/estado/hora.txt");

$nome_trancado = file_get_contents("api/files/estado/nome.txt");


$url = "http://127.0.0.1/projeto/api/api.php";

//inicia a sessão no ficheiro

session_start();


//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login

if(!isset($_SESSION['username'])){

    header("refresh:3;url=login.php");

    die("Acesso restrito.");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Logótipo no separador -->
```

```
<link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

<link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />

<meta charset="UTF-8">


<!-- Página faz refresh de 10 em 10 segundos-->

<!-- <meta http-equiv="refresh" content="2"> -->


<!-- Referência ao ficheiro css utilizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUfPcJcEUQouq2+1" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css2.css">


<!-- Título da página -->

<title>allEletric Atuadores</title>


<!-- links para o Dropdown usando javascript -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Fim das referencias aos links do dropdown-->


</head>


<body>


<!-- Navbar -->

<?php include('navbar.php'); ?>


<br>


<div class="container">

    <div class="card">


        <!-- Alternativa ao jumbotron -->

        <div class="class-header p-3 mb-2 bg-danger text-dark"> <b>Tabela de Atuadores</b></div>
```

```
<div class="card-body" >

    <table class="table">

        <!-- Cria o table header -->

        <thead>

            <tr>

                <th scope="col">Tipo de Atuador IoT</th>

                <th scope="col">Estado</th>

                <th scope="col">Data de Atualização</th>

            </tr>

        </thead>

        <!-- Fim do table header -->

        <tbody>

            <tr>

                <!-- Apresenta a informação do Ar Condicionado -->

                <td>Ventoinha</td>

                <!-- Caso esteja ligado ou desligado apresenta uma
mensagem -->

                <?php if($estado_ventoinha==1){

                    success'>Ligado</span></td>";

                    }else

                    {

                        danger'>Desligado</span></td>";

                    }

                <?>

                <!-- Apresenta a hora da ultima atualização -->

                <td><?php echo $hora_ventoinha?></td>

            </tr>

            <tr>

                <!-- Apresenta a informação das Portas -->

                <td>Portas</td>
```

```

        <!-- Apresenta a quantidade de portas abertas/fechadas -->

        <?php

        //faz envio POST para a API com a informação do botão em
que se carregou (se abriu ou fechou as portas)

        if($estado_portas==0){

            if($utilizador != "carro"){ //pois o utilizador
"carro" não tem controlo sobre nada, é apenas um modo de visualização

            if($valor_trancado==1){ //se o carro
estiver trancado, apenas mostra essa informação, não permitindo abrir as portas

                echo "

                <td><span class='badge
badge-pill badge-primary'> Portas trancadas </span></td>

                <td>". $hora_trancado."</td>";

            }else{

                echo "

                <td><span class='badge
badge-pill badge-primary'> Fechou 2 portas </span></td>

                <td>". $hora_portas."</td>

                <form method='post'

                action='http://127.0.0.1/projeto/api/api.php' target='ifrm1'>

                <input name='nome'

                type='hidden' value='portas'></input>

                <input name='valor'

                type='hidden' value='2'></input>

                <input name='hora'

                type='hidden' value='".date("Y-m-d H:i:s")."'></input>

                <input name='pass'

                type='hidden' value='ae123'></input>

                <input type='submit'

                value='Abrir Portas'></input>

                </form>

                <iframe id='ifrm1'

                name='ifrm1' style='display:none'></iframe>;

                }

            }

        }

    }else{ //se o carro estiver trancado, permite na mesma fechar as portas

        echo "<td><span class='badge badge-pill badge-primary'>

        Abriu 2 portas </span></td>";

        echo "<td>". $hora_portas."</td>";

        if($utilizador != "carro"){

```

```

        action='http://127.0.0.1/projeto/api/api.php' target='ifrm1'>
        value='portas'</input>
        value='0'</input>
        value='".date("Y-m-d H:i:s")."'</input>
        value='ae123'</input>
        Portas'</input>

        </form>
        <iframe
        id='ifrm1'
        name='ifrm1'

        }
    }
?>

</tr>
<tr>
    <!--Apresenta a informação da Lampada -->
    <td>Faróis</td>
    <!-- Caso esteja ligado ou desligado apresenta uma mensagem -->
    <?php if($estado_farois==1){
        echo "
        <td><span
        class='badge
        badge-pill
        badge-
        success'>Ligados</span></td>";

    }else
    {
        echo "
        <td><span
        class='badge
        badge-pill
        badge-
        danger'>Desligados</span></td>";

    }
?>
    <!-- Apresenta a hora da ultima atualização -->
    <td><?php echo $hora_farois?></td>

```

```

        </tr>

        <tr>

            <!-- Apresenta a informação da Camera - neste caso a camera é um
            atuador pois apenas se liga/desliga consoante a distância a objetos -->

            <td>Camera</td>

            <!-- Caso esteja ligado ou desligado apresenta uma mensagem -->

            <?php if($estado_camera==1){

                echo    "<td><span    class='badge    badge-pill    badge-
success'>Ligada</span></td>";

            }else

            {

                echo    "<td><span    class='badge    badge-pill    badge-
danger'>Desligada</span></td>";

            }

            ?>

            <!-- Apresenta a hora da ultima atualização -->

            <td><?php echo $hora_camera?></td>

        </tr>

        <tr>

            <!-- Apresenta a informação do Buzzer -->

            <td>Buzzer</td>

            <!-- Caso esteja ligado ou desligado apresenta uma mensagem -->

            <?php if($estado_buzzer==1){

                echo    "<td><span    class='badge    badge-pill    badge-
success'>Ligado</span></td>";

            }else

            {

                echo    "<td><span    class='badge    badge-pill    badge-
danger'>Desligado</span></td>";

            }

            ?>

            <!-- Apresenta a hora da ultima atualização -->

            <td><?php echo $hora_buzzer?></td>

```



```

</tr>

<tr>
    <td>Luz de Leitura</td>

<?php
//envia pedido post para a API com a informação do botão em que se
carregou

if($valor_luzLeitura==0){

    if($utilizador != "carro"){ //pois o utilizador
"carro" não tem controlo sobre nada, é apenas um modo de visualização

        echo      "<form      method='post'
action='http://127.0.0.1/projeto/api/api.php' target='ifrm1'>

        <input      name='nome'      type='hidden'
value='luzLeitura'></input>

        <input      name='valor'      type='hidden'
value='1'></input>

        <input      name='hora'      type='hidden'
value='".date("Y-m-d H:i:s")."></input>

        <input      name='pass'      type='hidden'
value='ae123'></input>

        <input type='submit' value='Ligar Luz de
Leitura'></input>

        </form>

        <iframe      id='ifrm1'      name='ifrm1'
style='display:none'></iframe>";

    }

}

}else{

    if($utilizador != "carro"){

        echo      "<form      method='post'
action='http://127.0.0.1/projeto/api/api.php' target='ifrm1'>

        <input      name='nome'      type='hidden'
value='luzLeitura'></input>

        <input      name='valor'      type='hidden'
value='0'></input>

        <input      name='hora'      type='hidden'
value='".date("Y-m-d H:i:s")."></input>

        <input      name='pass'      type='hidden'
value='ae123'></input>

        <input type='submit' value='Desligar Luz de
Leitura'></input>

        </form>

```

```

style='display:none'></iframe>";

                                <iframe                id='ifrm1'                name='ifrm1'

                                }

                                }

                                ?>

                                <!-- Apresenta o valor da ultima atualização -->

                                <?php if($valor_luzLeitura==0){

                                echo "<td><span class='badge badge-pill badge-danger'>

Luz apagada! </span></td>";

                                }else

                                {

                                echo "<td><span class='badge badge-pill badge-success'>

Luz acesa </span></td>";

                                }

                                ?>

                                <!-- Apresenta a hora da ultima atualização -->

                                <td><?php echo $hora_luzLeitura?></td>

                                </tr>

                                <tr>

                                <!--Apresenta a informação da Proximidade: valor e hora da ultima

atualização -->

                                <td>Luz de Aviso CO2</td>

                                <!-- Apresenta o valor da ultima atualização -->

                                <?php if($valor_luzCO2==0){

                                echo "<td><span class='badge badge-pill badge-success'>

Luz apagada </span></td>";

                                }else

                                {

                                if($valor_luzCO2==1){

                                echo "<td><span class='badge badge-pill badge-

danger'> Luz acesa (Aviso Nível Médio) </span></td>";

                                }else{

                                echo "<td><span class='badge badge-pill badge-

danger'> Luz acesa (Aviso Nível Alto) </span></td>";

                                }

                                }

                                ?>

```

```

        <!-- Apresenta a hora da ultima atualizaç o -->

        <td><?php echo $hora_luzCO2?></td>

    </tr>

    <tr>

        <!--Apresenta a informa  o da Proximidade: valor e hora da ultima
atualiza  o -->

        <td>Luz de Aviso Bateria</td>

        <!-- Apresenta o valor da ultima atualiza  o -->

        <?php if($valor_luzBateria==0){

            echo "<td><span class='badge badge-pill badge-success'>

        }else

        {

            echo "<td><span class='badge badge-pill badge-danger'>

Luz apagada </span></td>";

Luz acesa </span></td>";

        }

    ?>

    <!-- Apresenta a hora da ultima atualiza  o -->

    <td><?php echo $hora_luzBateria?></td>

    </tr>

    <tr>

        <!--Apresenta a informa  o da Proximidade: valor e hora da ultima
atualiza  o -->

        <td>Luz de Aviso Portas</td>

        <!-- Apresenta o valor da ultima atualiza  o -->

        <?php if($valor_luzPortas==0){

            echo "<td><span class='badge badge-pill badge-success'>

Portas fechadas </span></td>";

        }else

        {

            echo "<td><span class='badge badge-pill badge-danger'>

Portas abertas </span></td>";

        }

    ?>

    <td><?php echo $hora_luzPortas?></td>

    </tr>
    
```

```

    }
    ?>

    <!-- Apresenta a hora da ultima atualização -->
    <td><?php echo $hora_luzPortas?></td>

</tr>

<tr>

    <!-- Apresenta a informação da Proximidade: valor e hora da ultima
atualização -->

    <td>Tejadilho</td>

    <!-- Apresenta o valor da ultima atualização -->

    <?php if($valor_tejadilho==0){

Fechado </span></td>";

        echo "<td><span class='badge badge-pill badge-danger'>

    }else

    {

        echo "<td><span class='badge badge-pill badge-success'>

Aberto </span></td>";

    }

    ?>

    <!-- Apresenta a hora da ultima atualização -->

    <td><?php echo $hora_tejadilho?></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<!-- JavaScript para fazer com que a animação do dropdown funcione -->

<script>

```

```
$(document).ready(function(){

    $('<div>dropdown-submenu a.test</div>').on("click", function(e){

        $(this).next('ul').toggle();

        e.stopPropagation();

        e.preventDefault();

    });

});

</script>

<!-- Fim do código em JavaScript-->

</body>

</html>
```

## 8.8.Código do PHP Dashboard.php

```
<?php

// Dados a autenticação dos utilizadores

$_SESSION["username"] = "andre" ou "michael" ou "carro"

$_SESSION["password"] = "1234" ou "leiria" ou "carro"


//para a temperatura

$valor_temperatura = file_get_contents("api/files/temperatura/valor.txt");

$hora_temperatura = file_get_contents("api/files/temperatura/hora.txt");

$nome_temperatura = file_get_contents("api/files/temperatura/nome.txt");

//fim das variaveis da temperatura


//para a bateria

$valor_bateria = file_get_contents("api/files/bateria/valor.txt");

$hora_bateria = file_get_contents("api/files/bateria/hora.txt");

$nome_bateria = file_get_contents("api/files/bateria/nome.txt");

//fim das variaveis da bateria


//para as portas

$valor_portas = file_get_contents("api/files/portas/valor.txt");

$hora_portas = file_get_contents("api/files/portas/hora.txt");

$nome_portas = file_get_contents("api/files/portas/nome.txt");

//fim das variaveis das portas
```

```
//para a humidade

$valor_humidade = file_get_contents("api/files/humidade/valor.txt");

$hora_humidade = file_get_contents("api/files/humidade/hora.txt");

$nome_humidade = file_get_contents("api/files/humidade/nome.txt");

//fim das variaveis da humidade


//para a proximidade

$valor_proximidade = file_get_contents("api/files/proximidade/valor.txt");

$hora_proximidade = file_get_contents("api/files/proximidade/hora.txt");

$nome_proximidade = file_get_contents("api/files/proximidade/nome.txt");

//fim das variaveis da proximidade


//para a luminosidade

$valor_luminosidade = file_get_contents("api/files/luminosidade/valor.txt");

$hora_luminosidade = file_get_contents("api/files/luminosidade/hora.txt");

$nome_luminosidade = file_get_contents("api/files/luminosidade/nome.txt");

//fim das variaveis da luminosidade


//inicia a sessão no ficheiro

session_start();


//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login

if(!isset($_SESSION['username'])){

    header("refresh:3;url=login.php");

    die("Acesso restrito.");

}

?>


<!DOCTYPE html>


<html lang="en">


<head>
```

```
<!-- Logótipo no separador -->

<link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

<link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />


<!-- definição da codificação dos caracteres utilizados -->

<meta charset="UTF-8">


<!-- Página faz refresh de 15 em 15 segundos-->

<meta http-equiv="refresh" content="15">


<!-- Titulo da página -->

<title>allEletric Home</title>


<!-- Referência ao ficheiro css utlizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+1" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css2.css">


<!-- links para o Dropdown usando javascript -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Fim das referencias aos links do dropdown-->


</head>


<body>

<!-- Navbar -->

<?php include('navbar.php'); ?>


<!-- jumbotron da dashboard -->

<div class="jumbotron">

    <?php if($utilizador != "carro"){ echo "<h1 class='text-center' style='font-weight: bold'>Bem-vindo,".
$_SESSION['username']. "</h1>";} else { echo "<h1 class='text-center' style='font-weight: bold'> Modo Visualização </h1>";}?>
```

```
->
    <!--<h1 class="text-center" style="font-weight: bold">Bem-vindo, <?php echo $_SESSION['username'] ?></h1> -
    <!--<p class="text-center">Aqui pode ver todas as informações sobre o seu carro elétrico</p> -->

</div>

<!-- Início do carousel slide -->

<div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel" style="background-color: grey">

    <ol class="carousel-indicators">

        <li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>

        <li data-target="#carouselExampleCaptions" data-slide-to="1"></li>

        <li data-target="#carouselExampleCaptions" data-slide-to="2"></li>

    </ol>

    <!-- Início dos tabs do carousel slide-->

    <div class="carousel-inner" >

        <!-- Primeiro "tab" do carousel slide -->

        <div class="carousel-item active">

            <!-- imagem em branco para colocar as dimensões pretendidas-->

            <!-- div para colocar o texto no carousel slide -->

            <div class="carousel-caption d-none d-md-block">

                <h5 style="color: red"><b>allEletric</b></h5>

                <p>Muito mais do que um simples meio de transporte</p>

            </div>

        </div>

        <!-- Fim do primeiro "tab" do carousel slide-->

        <!-- Segundo "tab" do carousel slide -->

        <div class="carousel-item">

            <!-- imagem em branco para colocar as dimensões pretendidas-->

            <!-- div para colocar o texto no carousel slide -->

            <div class="carousel-caption d-none d-md-block">

                <p>Aqui pode ver todas as informações sobre o seu carro elétrico</p>

            </div>

        </div>

    </div>

</div>
```



```
</div>

<!-- Fim do segundo "tab" do carousel slide -->

<!-- Terceiro "tab" do carousel slide -->
<div class="carousel-item">
    <!-- imagem em branco para colocar as dimensões pretendidas-->
    
    <!-- div para colocar o texto no carousel slide -->
    <div class="carousel-caption d-none d-md-block">
        <p>Mais novidades em breve...</p>
    </div>
</div>

<!-- Fim do terceiro "tab" do carousel slide -->
</div>

<!-- Fim dos tabs do carousel slide -->

<!-- botão para ir para o carousel slide anterior ao atual -->
<a class="carousel-control-prev" href="#carouselExampleCaptions" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>

<!-- botão para ir para o carousel slide posterior ao atual-->
<a class="carousel-control-next" href="#carouselExampleCaptions" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>

</div>

<br>

<!-- Fim do carousel slide -->

<!-- início dos cards -->
<div class="container-fluid text-center container-sm">
    <div class="row">
        <!-- divide na horizontal a página em 3 divisões de dimensão 4, em "small" -->
```

```

<!-- inicio da primeira divisão -->

<div class="col-sm-4">

    <div class="card border-danger mb-3" >

        <div class="card-body">

            <h5 class="card-title">Bateria: <?php echo $valor_bateria?> % </h5>

            <p class="card-text">Kms restantes: <?php echo
($valor_bateria*300)/100?>km</p>

            <p class="card-text">Última atualização do sensor a <?php echo
$hora_bateria?></p>

        </div>

    </div>

</div>

<!-- fim da primeira divisão -->

<!-- inicio da segunda divisão -->

<div class="col-sm-4">

    <div class="card border-danger mb-3" >

        <div class="card-body">

            <h5 class="card-title">Luminosidade: <?php echo
$valor_luminosidade?></h5>

            <p class="card-text">Última atualização do sensor a <?php echo
$hora_luminosidade?></p>

        </div>

    </div>

</div>

<!-- fim da segunda divisão -->

<!-- inicio da terceira divisão -->

<div class="col-sm-4">

    <div class="card border-danger mb-3" >

        <div class="card-body">

            <h5 class="card-title">Temperatura: <?php echo
$valor_temperatura?> °C</h5>

            <p class="card-text">Última atualização do sensor a <?php echo
$hora_temperatura?></p>

        </div>

    </div>

</div>

```

```
</div>

<!-- fim da terceira divisão -->

</div>

</div>

<!-- fim do primeiro set de cards -->

<br>

<!-- início do segundo set de cards -->

<div class="container-fluid text-center container-sm">

    <div class="row">

        <!-- divide na horizontal a página em 3 divisões de dimensão 4, em "small" -->

        <!-- inicio da primeira divisão -->

        <div class="col-sm-4">

            <div class="card border-danger mb-3" >

                <div class="card-body">

                    <h5 class="card-title">Portas abertas: <?php echo $valor_portas
?></h5>

                    <p class="card-text">Última atualização do sensor a <?php echo
$hora_portas ?></p>

                </div>

            </div>

        </div>

        <!-- fim da primeira divisão -->

        <!-- inicio da segunda divisão -->

        <div class="col-sm-4">

            <div class="card border-danger mb-3" >

                <div class="card-body">

                    <h5 class="card-title">Humidade: <?php echo $valor_humidade ?>
%</h5>

                    <p class="card-text">Última atualização do sensor a <?php echo
$hora_humidade ?></p>
```

```

        </div>

    </div>

</div>

<!-- fim da segunda divisão -->


<!-- inicio da terceira divisão -->

<div class="col-sm-4">

    <div class="card border-danger mb-3" >

        <div class="card-body">

            <h5 class="card-title"><?php echo $valor_proximidade ?> </h5>

            <p class="card-text">Última atualização do sensor a <?php echo
$hora_proximidade ?> </p> <!-- - <a href="#">Histórico</a> -->

        </div>

    </div>

</div>

<!-- fim da terceira divisão -->

<div class="col-sm-12">

    <div class="card border-danger mb-3" >

        <div class="card-body">

            <?php

//PARA CONTAR QUANTOS FICHEIROS ESTÃO NA DIRETORIA

$directory = "uploads/";

$filecount = 0;

//vai à path dos uploads, com a função glob, obtem um array com cada ficheiro que existe nos uploads

$files = glob($directory . "*");

if ($files){

    $filecount = count($files)-1;

    //se existirem ficheiros, e como começa a contar do zero, a última imagem que se fez upload tem o nome de
webcam'nº ficheiros - 1'.png

} ?>

            <h5 class="card-title">Camera:</h5>

            <?php

//vai mostrar apenas a última imagem que se fez upload

echo          "<img          src='uploads/webcam'.".$filecount." .jpg'

style='width:25%'>";

            ?>

```

```
</div>

<!-- mostra a hora que foi tirada a última foto -->

<div class="card-footer"><p>Atualização: <?php echo date("d-m-Y
H:i:s",filemtime('uploads/webcam'. $filecount.'.jpg')) ); ?> </p></div>

</div>

</div>

</div>

</div>

<!-- fim dos cards-->

<br>

<!-- Dá espaçamento entre os diferentes elementos-->

<br>

<!-- Início do footer -->

<footer id="colophon" class="site-footer">

    <!-- Inicio das imagens com links para os websites-->

    <div class="social-wrapper">

        <!-- Link com imagem para o twitter -->

        <a href="https://www.twitter.com" target="_blank">

        </a>

        <!-- Fim do link com imagem para o twitter -->

        <!-- Link com imagem para o instagram -->

        <a href="https://www.instagram.com" target="_blank">

        </a>

        <!-- Fim do link com imagem para o instagram -->

        <!-- Link com imagem para o linkedin -->

        <a href="https://www.linkedin.com" target="_blank">
```

```


</a>

<!-- Fim do link com imagem para o linkedin -->


<!-- Link com imagem para o facebook -->

<a href="https://www.facebook.com" target="_blank">

</a>

<!-- Fim do link com imagem para o facebook -->


<!-- Link com imagem para o google plus -->

<a href="https://www.google.com" target="_blank">

</a>

<!-- Fim do link com imagem para o google plus -->


</div>


<nav class="footer-nav">

    <!-- Informação com a criação da marca e o ano atual -->

    <a>

        Copyright &copy; 2020 -

        <?php echo date("Y"); ?> allEletric. Todos os direitos reservados.

    </a>

    <!-- Fim da informação -->

</nav>


</footer>

<!--Fim do footer-->


<!-- Início de código em JavaScript para o botão do histórico na navbar funcionar corretamente -->

<script>

$(document).ready(function(){

    $('<div>.dropdown-submenu a.test').on("click", function(e){

        $(this).next('ul').toggle();

        e.stopPropagation();

    });

});

</script>
```

```
        e.preventDefault();

    });

});

</script>

<!-- Fim do código em JavaScript-->

</body>

</html>
```

## 8.9.Código do PHP Historico.php

```
<?php

//coloca em variáveis os valores passados pelo URL

$sensor=$_REQUEST['sensor'];

$tipo=$_REQUEST['tipo'];

//coloca dentro de uma variavel o conteúdo do ficheiro log

$log = file_get_contents("api/files/".$sensor."/log.txt");

//converter o $log num string pois o 2º parametro do explode() necessita de ser uma string

$logString = strval($log);

$linhas = explode("|", $logString); //1º elemento do vetor é o que está antes do primeiro "|", 2º elemento do vetor é o que está entre
o primeiro "|" e o segundo "|" ...etc... (Divide os diferentes valores)

//inicia a sessão no ficheiro

session_start();

//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login

if(!isset($_SESSION['username'])) {

    header("refresh:3;url=login.php");

    die("Acesso restrito.");

}

?>

<!DOCTYPE html>
```

```

<html lang="en">

<head>

    <!-- links para o Dropdown funcionar-->

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>


    <!-- Logótipo no separador -->

    <link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

    <link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />


    <meta charset="UTF-8">


    <meta http-equiv="refresh" content="10"> <!--para fazer refresh de 10 em 10 seg-->


    <!-- Referência ao ficheiro css utilizado -->

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUfPcJjEUQouq2+1" crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="Css2.css">


    <!-- Colocar o caminho do sensor/atuador que se está a ver o histórico -->

    <?php echo "<title> Histórico/" . $sensor . "</title>" ?>


</head>


<body>

    <!-- inclui a navbar -->

    <?php include('navbar.php'); ?>


    <!-- separa a navbar do restante código -->

    <br>


    <!-- início do container utilizado para mostrar a tabela de atualizações do ficheiro log.txt do sensor/atuador selecionado -->

```



```

<div class="container">

    <!-- divisão no início da página a mostrar informações sobre o sensor/atuador -->

    <div class="p-3 mb-2 bg-danger text-white">

        <!-- imagem do sensor/atuador em 64px -->

        <?php echo "<img style='float:right;' src='".$sensor."'64.png' alt='imagem do sensor/atuador'" ?>

        <?php echo "<h1 class='text-left'" . $sensor . "</h1'" ?>

        <!-- header para indicar o caminho utilizado no histórico (Histórico/Sensor ou Histórico/Atuador)-->

        <h6 class="text-left">Histórico/<?php echo $tipo?></h6>

    </div>

    <!-- Inicio da table -->

    <div class="card">

        <div class="card-header" style="font-weight: bold">Atualizações - <?php echo $sensor ?> : </div>

        <div class="card-body">

            <table class="table">

                <!-- Inicio do table header -->

                <thead>

                    <tr>

                        <th scope="col">Data de Atualização</th>

                        <th scope="col">Valor</th>

                    </tr>

                </thead>

                <!-- Fim do table header -->

                <tbody>

                    <?php

                        // Para cada linha do ficheiro log, vai separar a hora do
                        // respectivo valor com o explode

                        foreach ($linhas as $valores) {

                            $separar = explode(";", $valores);

                            $hora = $separar[0];

                            // caso não tenha sido introduzido um valor,
                            // coloca-o a null

                            if (!isset($separar[1])){

                                $separar[1] = null;

                            }

                        }

                    </?php

```

```

$valor=$separar[1];

// Cria uma entrada na tabela por cada Data e Valor
encontrado

echo
"<tr>

<td>".$hora."</td>

<td>".$valor."</td>

</tr>";
}
?>

<!-- Fim do foreach e da criação de entradas na tabela-->

</tbody>

<!-- Fim do table body -->

</table>

<!-- Fim do table header-->

</div>

</div>

<!-- fim do container principal -->

<!-- Início de código em JavaScript para o botão do histórico na navbar funcionar corretamente -->
<script>
$(document).ready(function(){
    $('#dropdown-submenu a.test').on("click", function(e){
        $(this).next('ul').toggle();
        e.stopPropagation();
        e.preventDefault();
    });
});
</script>

<!-- Fim do código em JavaScript-->

</body>

```

```
</html>
```

## 8.10. Código do PHP Imagens.php

```
<?php
```

```
//coloca dentro de uma variavel o conteúdo do ficheiro log
```

```
//$log = file_get_contents("/uploads");
```

```
//inicia a sessão no ficheiro
```

```
session_start();
```

```
//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login
```

```
if(!isset($_SESSION['username'])){
```

```
    header("refresh:3;url=login.php");
```

```
    die("Acesso restrito.");
```

```
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <!-- links para o Dropdown funcionar-->
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
```

```
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

```
    <!-- Logótipo no separador -->
```

```
    <link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />
```

```
    <link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />
```

```
    <meta charset="UTF-8">
```

```
    <!--<meta http-equiv="refresh" content="10"> para fazer refresh de 5 em 5 seg-->
```

```

<!-- Referência ao ficheiro css utilizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+1" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css2.css">

<!-- Colocar o caminho do sensor/atuador que se está a ver o histórico -->

<?php echo "<title> Imagens </title>" ?>

</head>

<body>

<!-- inclui a navbar -->

<?php include('navbar.php'); ?>

<!-- separa a navbar do restante código -->

<br>

<!-- início do container utilizado para mostrar a tabela de atualizações do ficheiro log.txt do sensor/atuador selecionado -->

<div class="container">

    <!-- divisão no início da página a mostrar informações sobre o sensor/atuador -->

    <div class="p-3 mb-2 bg-danger text-white">

        <!-- imagem do sensor/atuador em 64px -->

        <?php echo "<img style='float:right;' src='imagens64.png' alt='imagem do sensor/atuador'>" ?>

        <?php echo "<h1 class='text-left'>Imagens</h1>" ?>

        <!-- header para indicar o caminho utilizado no histórico (Histórico/Sensor ou Histórico/Atuador)-->

        <h6 class="text-left">Histórico de imagens</h6>

    </div>

    <!-- Inicio da table -->

    <div class="card">

        <div class="card-header" style="font-weight: bold" > Atualizações : </div>

        <div class="card-body" >

            <table class="table">

```

```

<!-- Inicio do table header -->

<!-- Fim do table header -->

<!-- Inicio do table body-->
<tbody>
    <?php
        //vai à path dos uploads, com a função glob, obtem um array com cada
ficheiro que existe nos uploads

        $directory = "uploads/";
        $filecount = 0;
        $files = glob($directory . "*"); //vai buscar todos os ficheiros existentes
        if ($files){ //se existirem ficheiros
            $filecount = count($files) - 1 ;
            //se existirem ficheiros, e como começa a contar do zero, a
última imagem que se fez upload tem o nome de webcam'nº ficheiros - 1'.png
        }

        for ($i = 0; $i <= $filecount; $i++) { //mostra no historico de imagens,
para cada imagem existente. Começando na foto mais antiga
            echo "
                <img
                src='uploads/webcam'.". $i . ".jpg'
style='width:25%'>"; //no historico, fazer foreach imagem no uploads

            echo "
                <div class='card-footer'><p>Atualização: ". date('d-
m-Y H:i:s',filetime('uploads/webcam'.". $i . ".jpg')) )." </p></div>";

            }

        ?>

        <!-- Fim do foreach e da criação de entradas na tabela-->

    </tbody>

    <!-- Fim do table body -->

</table>

<!-- Fim do table header-->

</div>

```

```
</div>

<!-- fim do container principal -->

<!-- Início de código em JavaScript para o botão do histórico na navbar funcionar corretamente -->

<script>

    $(document).ready(function(){

        $('.dropdown-submenu a.test').on("click", function(e){

            $(this).next('ul').toggle();

            e.stopPropagation();

            e.preventDefault();

        });

    });

</script>

<!-- Fim do código em JavaScript-->

</body>

</html>
```

## 8.11. Código do PHP Login.php

```
<?php

    //para aceder -> http://127.0.0.1/projeto/login.php

    session_start();

    //credenciais de autenticação para o primeiro utilizador - o que tem mais permissões

    $username1="Andre";

    $password1="1234";

    //credenciais de autenticação para o segundo utilizador

    $username2="Micael";

    $password2="leiria";

    //credenciais de autenticação para o terceiro utilizador - o que tem menos permissões (apenas consegue visualizar alguns dados)

    $username3="carro";

    $password3="carro";

    //verificar se as variáveis existem
```

```
if (isset($_POST['username']) && isset($_POST['password'])) {

    //verificar se o login foi bem sucedido (password certa para o respetivo username)

    if( ($_POST['username'] != $username1 || $_POST['password'] != $password1) && ($_POST['username'] != $username2 ||
$_POST['password'] != $password2) && ($_POST['username'] != $username3 || $_POST['password'] != $password3)) {

        echo "<div class='alert alert-danger alert-dismissible'>

        <button type='button' class='close' data-dismiss='alert'>&times;</button>

        <strong>Credenciais erradas!</strong>

        </div>" ;

    }

    else {

        echo "<div class='alert alert-success alert-dismissible'>

        <button type='button' class='close' data-dismiss='alert'>&times;</button>

        <strong>Sucesso!</strong> A redireccionar...

        </div>" ;

        $_SESSION['username'] = $_POST['username'];

        header('refresh:1; url=dashboard.php?opcao=dashboard');

    }

}

?>

<!-- Início da parte HTML -->

<!doctype html>

<html lang="en">

<head>

    <!-- Logótipo no separador -->

    <link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

    <link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />

    <!-- definição da codificação dos caracteres utilizados -->

    <meta charset="utf-8">

    <!-- instruções para controlar as dimensões/scaling da página no browser -->
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Referência ao ficheiro css utilizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-
B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUfPcJEUQouq2+I" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css.css">

<!-- Título da página -->

<title>allEletric Car</title>

</head>

<body>

<!-- container para introduzir as credenciais -->

<div class="container">

    <form class="formProj" method="post">

        <div class="form-group">

            <label for="username" id="usr">Username:</label>

            <input type="text" class="form-control" id="username" name="username"
placeholder="Escreva o username" required>

        </div>

        <div class="form-group">

            <label for="pwd">Password:</label>

            <input name="password" type="password" class="form-control" id="pwd"
placeholder="Escreva a password" required>

        </div>

        <input type="submit" class="btn btn-info" value="Submeter" style="background-color:#D6292B;">

    </form>

</div>

<!-- fim do container para introduzir as credenciais -->

<!-- Início do footer -->

<footer id="colophon" class="site-footer">

    <div class="social-wrapper">

        <a href="https://www.twitter.com" target="_blank">

        </a>

    </div>

</footer>
```



```

<a href="https://www.instagram.com" target="_blank">

</a>

<a href="https://www.linkedin.com" target="_blank">

</a>

<a href="https://www.facebook.com" target="_blank">

</a>

<a href="https://www.google.com" target="_blank">

</a>

</div>


<nav class="footer-nav">

    <a>

        Copyright &copy; 2020 -

        <?php echo date("Y"); ?> allEletric. Todos os direitos reservados.

    </a>

    <!--<p style="text-align: right" > <a href="#top" >Back to top of page</a></p>-->

</nav>

</footer>

<!-- fim do footer -->

</body>

</html>

```

## 8.12. Código do PHP Logout.php

```

<?php

session_start();    // diz ao browser para utilizar variáveis de sessão;

    session_unset(); // remove todas as variáveis de sessão;

session_destroy(); // destrói a sessão

header("Location: login.php");

?>

```

## 8.13. Código do PHP Navbar.php

```

<?php

$opcao=$_REQUEST['opcao'];

$utilizador=$_SESSION['username'];

?>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

    <div class="container">

        <!-- Logótipo da marca na navbar -->

        <!-- Nome da marca na navbar -->

        <a class="navbar-brand" style="color: red" href="dashboard.php?opcao=dashboard">&nbsp;allEletric&nbsp;</a>

        <!-- Início das opções na navbar, com condições para saber em qual página se carregou -->

    <?php

//quando está na página dashboard.php, coloca a "Página Principal" como activa para distinguir das outras páginas, fazendo uma animação
quando se faz hover nestas -> tabindex=-1 no dropdown para poder navegar dentro do dropdown(e não consegue ser acedido pelo teclado)

    if($opcao=="dashboard"){

        echo "

        <ul class='navbar-nav mr-auto'>

            <li class='nav-item'>

                <a class='nav-link active' href='dashboard.php?opcao=dashboard'>Página Principal<span class='sr-
only'>(current)</span></a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>

            </li>

            <li>"/>se o utilizador

            if ($utilizador=="Micael"){

                echo "<li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

                </li>

                <li class='nav-item'>

```

```

Imagens</a>
<a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
</li>";
}

do historico
if($utilizador != "carro"){ //no modo visualização, apenas permite-se ver os valores atuais, e não os dados

    echo "
    <li class='nav-item'>
    <a class='nav-link hvr-underline-from-center'
href='perfil.php?opcao=".$utilizador."'>Perfil</a>
    </li>

    </ul>

    <div class='nav-item dropdown'>
    <div class='dropdown'>
    <a class='nav-link dropdown-toggle hvr-underline-from-center' href=' # '
id='navbarDropdown' role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>
    Histórico
    </a>

    <div class='dropdown-menu' aria-labelledby='navbarDropdown'>
    <ul>
    <li class='dropdown-submenu'>
    <a class='test dropdown-item' tabindex='-1' href=' # ' >Sensores <span
class='caret'></span></a>

    <ul class='dropdown-menu'>
    <li>
    <a class='dropdown-item'
href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a></li>
    <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>

```

```

</ul>

<a class='test dropdown-item' tabindex='-1' href=' # '>Atuadores <span
class='caret'></span></a>

<ul class='dropdown-menu'>

<li>

<a
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>

<li><a
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>

<li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>

<li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>

<li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>

<li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>

<li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>

<li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>

<li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>

<li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>";

//se o utilizador for o admin

if ($utilizador=="Micael"){

echo "
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";

}

echo "

</ul>

</li>

</ul>

</div>

</div>

```

```
        ";
    }

}

}else

//quando está na página atuadores.php, coloca "Atuadores" como activa para distinguir das outras páginas, fazendo
uma animação quando se faz hover nestas

if($opcao=="atuadores")
{
    echo "

    <ul class='navbar-nav mr-auto'>

    <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>

    </li>

    <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>

    </li>

    <li class='nav-item'>

        <a class='nav-link active' href='atuadores.php?opcao=atuadores'>Atuadores</a>

    </li>";

    if ($utilizador=="Micael"){

        echo "<li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
Imagens</a>

        </li>";

    }

    if($utilizador != "carro"){

        echo "

        <li class='nav-item'>

            <a class='nav-link hvr-underline-from-center' href='perfil.php?opcao=".$utilizador."'>Perfil</a>

        </li>

    }
```

```

</ul>

<div class='nav-item dropdown'>
  <div class='dropdown'>
    <a class='nav-link dropdown-toggle hvr-underline-from-center' href='#'
id='navbarDropdown' role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>
      Histórico
    </a>

    <div class='dropdown-menu' aria-labelledby='navbarDropdown'>
      <ul>
        <li class='dropdown-submenu'>
          <a class='test dropdown-item' tabindex='-1' href='#'>Sensores <span
class='caret'></span></a>

          <ul class='dropdown-menu'>

            <li><a
href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>

          </ul>

          <a class='test dropdown-item' tabindex='-1' href='#'>Atuadores <span
class='caret'></span></a>

          <ul class='dropdown-menu'>

            <li><a
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>

            <li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>

```

```

        <li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>;

        //se o utilizador for o admin

        if ($utilizador=="Micael"){

            echo "
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";

        }

        echo "

</ul>

</li>

</ul>

</div>

</div>

";

    }

}

}else

if($opcao=="sensores")

{

    echo "

<ul class='navbar-nav mr-auto'>

<li class='nav-item'>

```

```

        <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>

    </li>

    <li class='nav-item'>

        <a class='nav-link active' href='sensores.php?opcao=sensores'>Sensores</a>

    </li>

    <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>

    </li>";

    if ($Utilizador=="Micael"){

        echo "<li class='nav-item'>

            <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
Imagens</a>

                </li>";

        }

    if($Utilizador != "carro"){

        echo "

            <li class='nav-item'>

                <a
                    class='nav-link
                    hvr-underline-from-center'
href='perfil.php?opcao=".$Utilizador."'>Perfil</a>

                </li>

            </ul>

            <div class='nav-item dropdown'>

                <div class='dropdown'>

                    <a class='nav-link dropdown-toggle hvr-underline-from-center' href=' # '
id='navbarDropdown' role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>

                        Histórico

                    </a>

                    <div class='dropdown-menu' aria-labelledby='navbarDropdown'>

                        <ul>

                            <li class='dropdown-submenu'>

                                <a class='test dropdown-item' tabindex='-1' href=' # ' >Sensores <span
class='caret'></span></a>

                                <ul class='dropdown-menu'>

```



```

        <li><a
href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>
class='dropdown-item'

    </ul>

    <a class='test dropdown-item' tabindex='-1' href=' # ' >Atuadores <span
class='caret'></span></a>

    <ul class='dropdown-menu'>

        <li><a
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>;
class='dropdown-item'

        //se o utilizador for o admin

        if ($utilizador=="Micael"){

            echo "
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";
class='dropdown-item'

        }
    
```

```

        echo "

        </ul>

        </li>

        </ul>

        </div>

        </div>

        </div>";
    }

}

else // Quando a opção da navbar seleccionada foi "Histórico" coloca-a como activa para distinguir das outras páginas,
fazendo uma animação quando se faz hover nestas

if($opcao=="historico"){

    echo "

    <ul class='navbar-nav mr-auto'>

    <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>

        </li> ";

    if ($Utilizador=="Micael"){

        echo "<li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
Imagens</a>

        </li>";

    }
}

```

```

echo "

<li class='nav-item'>

<a class='nav-link hvr-underline-from-center' href='perfil.php?opcao=".$utilizador."'>Perfil</a>

</li>

</ul>

<div class='nav-item dropdown'>

<div class='dropdown'>

<a class='nav-link dropdown-toggle active' href=' # ' id='navbarDropdown' role='button' data-
toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:white !important;'>

Histórico

</a>

<div class='dropdown-menu' aria-labelledby='navbarDropdown'>

<ul>

<li class='dropdown-submenu'>

<a class='test dropdown-item' tabindex='-1' href=' # '>Sensores <span class='caret'></span></a>

<ul class='dropdown-menu'>

<li><a href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a></li>

<li><a href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a></li>

<li><a href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a></li>

<li><a href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a></li>

<li><a href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>

<li><a href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a></li>

<li><a href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>

</ul>

<a class='test dropdown-item' tabindex='-1' href=' # '>Atuadores <span class='caret'></span></a>

<ul class='dropdown-menu'>

<li><a href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>

<li><a href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>

```

```

        <li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>";

//se o utilizador for o admin

        if ($utilizador=="Micael"){

            echo
            "
            <li><a
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";
            class='dropdown-item'

        }

        echo "

        </ul>

        </li>

        </ul>

        </div>

        </div>

        </div>

        ";

    }else

        // entra nesta parte do código se o utilizador for "Micael", que é o utilizador com mais permissões

        if($opcao == "painel")

        {

            echo "

            <ul class='navbar-nav mr-auto'>

```

```

<li class='nav-item'>
    <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>
</li>
<li class='nav-item'>
    <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>
</li>
<li class='nav-item'>
    <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>
</li>
<li class='nav-item'>
    <a class='nav-link active' href='painel.php?opcao=painel'>Painel de controlo</a>
</li>
<li class='nav-item'>
    <a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
Imagens</a>
</li>
<li class='nav-item'>
    <a class='nav-link hvr-underline-from-center' href='perfil.php?opcao=". $utilizador.">Perfil</a>
</li>
</ul>

<div class='nav-item dropdown'>
    <div class='dropdown'>
        <a class='nav-link dropdown-toggle hvr-underline-from-center' href=' # ' id='navbarDropdown'
role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>
            Histórico
        </a>

        <div class='dropdown-menu' aria-labelledby='navbarDropdown'>
            <ul>
                <li class='dropdown-submenu'>
                    <a class='test dropdown-item' tabindex='-1' href=' #'>Sensores <span class='caret'></span></a>
                    <ul class='dropdown-menu'>
                        <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a>
                        <li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a>
                    </ul>
                </li>
            </ul>
        </div>
    </div>

```

```

        <li><a
href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>
class='dropdown-item'

    </ul>

    <a class='test dropdown-item' tabindex='-1' href='#' >Atuadores <span class='caret'></span></a>

    <ul class='dropdown-menu'>

        <li><a
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>";

    if ($utilizador=="Micael"){

        echo
        "
        <li><a
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";
        class='dropdown-item'

    }

    echo "

    </ul>

    </li>

    </ul>

    </div>

```

```

        </div>

        </div>";

    }else

        if($opcao == "imagens"){ //entra aqui se quisermos mostrar o Historico de Imagens (apenas permitido para os
        utilizadores Micael e Andre)

            echo "

            <ul class='navbar-nav mr-auto'>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>

            </li> ";

            if ($Utilizador=="Micael"){

                echo "<li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

                </li>

                <li class='nav-item'>

                <a class='nav-link active' href='imagens.php?opcao=imagens'>Histórico Imagens</a>

                </li>";

            }

            echo "

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='perfil.php?opcao=".$Utilizador.">Perfil</a>

            </li>

```

```

</ul>

<div class='nav-item dropdown'>

<div class='dropdown'>

<a class='nav-link dropdown-toggle hvr-underline-from-center' href=' # ' id='navbarDropdown'
role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>

Histórico

</a>

<div class='dropdown-menu' aria-labelledby='navbarDropdown'>

<ul>

<li class='dropdown-submenu'>

<a class='test dropdown-item' tabindex='-1' href=' # '>Sensores <span class='caret'></span></a>

<ul class='dropdown-menu'>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>

</ul>

<a class='test dropdown-item' tabindex='-1' href=' # '>Atuadores <span class='caret'></span></a>

<ul class='dropdown-menu'>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>

<li><a class='dropdown-item'
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>

```



```

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>";
class='dropdown-item'

//se o utilizador for o admin

if ($utilizador=="Micael"){

        echo
        "
        <li><a
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>";
class='dropdown-item'

        }

        echo "

</ul>

</li>

</ul>

</div>

</div>

</div>";

}

}

}else{ //entra aqui se for para mostrar o perfil, apenas visivel para os utlizadores Micael e Andre

        echo "

        <ul class='navbar-nav mr-auto'>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='dashboard.php?opcao=dashboard'>Página
Principal<span class='sr-only'>(current)</span></a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='sensores.php?opcao=sensores'>Sensores</a>

        </li>

        <li class='nav-item'>

        <a class='nav-link hvr-underline-from-center' href='atuadores.php?opcao=atuadores'>Atuadores</a>

        </li> ";

```

```

        if ($Utilizador=="Micael"){

            echo "<li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='painel.php?opcao=painel'>Painel de
controlo</a>

            </li>

            <li class='nav-item'>

                <a class='nav-link hvr-underline-from-center' href='imagens.php?opcao=imagens'>Histórico
Imagens</a>

            </li>";

        }

        echo "

        <li class='nav-item'>

            <a class='nav-link active' href='perfil.php?opcao=".$Utilizador."'>Perfil</a>

        </li>

    </ul>

    <div class='nav-item dropdown'>

        <div class='dropdown'>

            <a class='nav-link dropdown-toggle hvr-underline-from-center' href='#' id='navbarDropdown'
role='button' data-toggle='dropdown' aria-haspopup='true' aria-expanded='false' style='color:#b3b5b4;'>

                Histórico

            </a>

            <div class='dropdown-menu' aria-labelledby='navbarDropdown'>

                <ul>

                    <li class='dropdown-submenu'>

                        <a class='test dropdown-item' tabindex='-1' href='#'>Sensores <span class='caret'></span></a>

                        <ul class='dropdown-menu'>

                            <li><a

                                href='historico.php?opcao=historico&sensor=bateria&tipo=Sensores'>Bateria</a>
                                class='dropdown-item'

                            </li>

                            <li><a

                                href='historico.php?opcao=historico&sensor=luminosidade&tipo=Sensores'>Luminosidade</a>
                                class='dropdown-item'

                            </li>

                            <li><a

                                href='historico.php?opcao=historico&sensor=temperatura&tipo=Sensores'>Temperatura</a>
                                class='dropdown-item'

                            </li>

                            <li><a

                                href='historico.php?opcao=historico&sensor=whiteNoise&tipo=Sensores'>White Noise</a>
                                class='dropdown-item'

                            </li>

                            <li><a

                                href='historico.php?opcao=historico&sensor=humidade&tipo=Sensores'>Humidade</a></li>
                                class='dropdown-item'

                            </li>

                            <li><a

                                href='historico.php?opcao=historico&sensor=proximidade&tipo=Sensores'>Estado das proximidades</a>
                                class='dropdown-item'

                            </li>

                        </ul>

                    </li>

                </ul>

            </div>

        </div>

```

```

        <li><a
href='historico.php?opcao=historico&sensor=CO2&tipo=Sensores'>Níveis de CO2</a></li>
class='dropdown-item'

    </ul>

    <a class='test dropdown-item' tabindex='-1' href=' #'>Atuadores <span class='caret'></span></a>

    <ul class='dropdown-menu'>

        <li><a
href='historico.php?opcao=historico&sensor=arCondicionado&tipo=Atuadores'>Ar Condicionado</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=farois&tipo=Atuadores'>Faróis</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=buzzer&tipo=Atuadores'>Buzzer</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=camera&tipo=Atuadores'>Camera</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=portas&tipo=Atuadores'>Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzLeitura&tipo=Atuadores'>Luz de leitura</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzCO2&tipo=Atuadores'>Luz aviso CO2</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzBateria&tipo=Atuadores'>Luz aviso Bateria</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=luzPortas&tipo=Atuadores'>Luz aviso Portas</a></li>
class='dropdown-item'

        <li><a
href='historico.php?opcao=historico&sensor=tejadilho&tipo=Atuadores'>Tejadilho</a></li>;
class='dropdown-item'

        //se o utilizador for o admin

        if ($utilizador=="Micael"){

            echo
            " <li><a
href='historico.php?opcao=historico&sensor=estado&tipo=atuador'>On/Off</a></li>;
            class='dropdown-item'

        }

        echo "

    </ul>

</li>

</ul>

</div>

</div>

</div>";

}

?>

</div>

```

```
<!-- botão do logout com a imagem -->  
  
<a href="logout.php" class="btn btn-danger hvr-buzz-out" role="button"> Sair</a>
```

```
</nav>
```

```
<!-- Retorna ao ficheiro que fez include da navbar.php -->
```

## 8.14. Código do PHP Painei.php

```
<?php
```

```
//inicia a sessão no ficheiro
```

```
session_start();
```

```
//para o estado do carro
```

```
$valor = file_get_contents("api/files/estado/valor.txt");
```

```
//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login
```

```
if(!isset($_SESSION['username'])){  
    header("refresh:3;url=login.php");  
    die("Acesso restrito.");  
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<!-- Logótipo no separador -->
```

```
<link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />
```

```
<link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />
```

```
<meta charset="UTF-8">
```

```
<!-- Página faz refresh de 10 em 10 segundos-->

<meta http-equiv="refresh" content="10">


<!-- Referência ao ficheiro css utilizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+1" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css2.css">


<!-- Título da página -->

<title>allEletric Painele de Controlo</title>


<!-- links para o Dropdown usando javascript -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Fim das referencias aos links do dropdown-->

</head>

<body>


<!-- Navbar -->

<?php include('navbar.php'); ?>


<br>


<div class="container">

    <div class="card">


        <!-- Alternativa ao jumbotron -->

        <div class="class-header p-3 mb-2 bg-danger text-dark"> <b>Painele de Controlo</b></div>


        <div class="card-body" >

            <table class="table">

                <!-- Cria o table header -->

                <thead>
```

```

        <tr>

            <th scope="col">Informação: </th>

            <th scope="col"> </th>

            <th scope="col">Estado</th>

        </tr>

    </thead>

    <!-- Fim do table header -->

    <tbody>

        <tr>

            <!-- Apresenta a informação do estado do carro - se está ou
não destrancado -->

            <td>Estado do Carro</td>

            <td> </td>

            <?php if($valor==0){ //está destrancado

                echo "<td><span class='badge badge-pill badge-
success' style='text-align:center'>Destrancado</span></td>";

            }else

            {

                echo "<td><span class='badge badge-pill badge-
danger'>Trancado</span></td>";

            }

        ?>

            </td>

        <tr>

            <td>Modelo do carro</td>

            <td> </td>

            <td>X Æ A-06</td>

        </td>

        <tr>

            <td>Matrícula</td>

            <td> </td>

```

```

        <td>ZK 62 KL</td>

    </tr>

    <tr>

        <td>Ano de fabrico</td>

        <td> </td>

        <td>2019</td>

    </tr>

</tbody>

</table>

</div>

</div>

</div>

<!-- JavaScript para fazer com que a animação do dropdown funcione -->

<script>

    $(document).ready(function(){

        $('<div>dropdown-submenu a.test</div>').on("click", function(e){

            $(this).next('ul').toggle();

            e.stopPropagation();

            e.preventDefault();

        });

    });

</script>

<!-- Fim do código em JavaScript-->

</body>

</html>

```

## 8.15. Código do PHP Perfil.php

```

<?php

session_start();

```

```

if(!isset($_SESSION['username'])){

    header("refresh:3;url=login.php");

    die("Acesso restrito.");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Logótipo no separador -->

    <link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

    <link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />

    <!-- definição da codificação dos caracteres utilizados -->

    <meta charset="UTF-8">

    <!-- Página faz refresh de 15 em 15 segundos-->

    <meta http-equiv="refresh" content="15">

    <!-- Referência ao ficheiro css utilizado -->

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+I" crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="Css.css">

    <!-- links para o Dropdown usando javascript -->

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

    <!-- Fim das referencias aos links do dropdown-->

    <!-- Colocar o caminho do sensor/atuador que se está a ver o histórico -->

    <title> Perfil </title>

```



```

</head>

<body>

    <!-- Navbar -->

    <?php include('navbar.php'); ?>

    <!-- jumbotron da dashboard -->

    <div class="jumbotron">


    </div>

    <div class="container">

        <div class="card">


            <!-- Alternativa ao jumbotron -->

            <div class="class-header p-3 mb-2 bg-danger text-dark"> <b>O seu perfil:</b></div>

            <div class="card-body" >

                <table class="table">

                    <!-- Cria o table header -->

                    <thead>

                        <tr>

                            <th scope="col">Informação:</th>

                        </tr>

                    </thead>

                    <!-- Fim do table header -->

                    <tbody>

                        <!-- mostra a imagem do utilizador que fez login -->

                        <?php echo <img
src='perfil/fotoPerfil'.$_SESSION['username'].'.jpg' alt='imagem de perfil' width='400' class='center'>; ?>

                        <h1 class="text-center" style="font-size:10px">

                            <!-- formulário para fazer upload à foto de perfil -->

                            <form action="uploadFoto.php" method="post"
enctype="multipart/form-data">

                                Select image to upload:

                                <input type="file" name="imagem"

                                <input type="submit" value="Upload Image"
name="submit">

```

```
</form>

</h1>

<?php if($_SESSION['username'] == "Micael"){

    echo "

    <tr>

    <!--Apresenta a informação das Portas -->

    <td>Nome:</td>

    <td> </td>

    <td>Micael</td>

    </tr>

    <tr>

    <!--Apresenta a informação das Portas -->

    <td>E-mail:</td>

    <td> </td>

    <td>2201743@my.ipleiria.pt</td>

    </tr>

    ";

}

} else { //como só mostra o perfil.php aos utilizadores "Micael" e
"Andre", neste else está garantido que o utilizador será o "Andre"

    echo "

    <tr>

    <!--Apresenta a informação das Portas -->

    <td>Nome:</td>

    <td> </td>

    <td>André</td>

    </tr>

    <tr>

    <!--Apresenta a informação das Portas -->

    <td>E-mail:</td>

    <td> </td>

    <td>2201723@my.ipleiria.pt</td>

    </tr>
```

```
        ";
    }
    ?>

</tbody>

</table>

</div>

</div>

</div>

<!-- Início do footer -->

<footer id="colophon" class="site-footer">

    <!-- Inicio das imagens com links para os websites-->

    <div class="social-wrapper">

        <!-- Link com imagem para o twitter -->

        <a href="https://www.twitter.com" target="_blank">

        </a>

        <!-- Fim do link com imagem para o twitter -->

        <!-- Link com imagem para o instagram -->

        <a href="https://www.instagram.com" target="_blank">

        </a>

        <!-- Fim do link com imagem para o instagram -->

        <!-- Link com imagem para o linkedin -->

        <a href="https://www.linkedin.com" target="_blank">

        </a>

        <!-- Fim do link com imagem para o linkedin -->

        <!-- Link com imagem para o facebook -->

        <a href="https://www.facebook.com" target="_blank">
```

```


</a>

<!-- Fim do link com imagem para o facebook -->


<!-- Link com imagem para o google plus -->

<a href="https://www.google.com" target="_blank">

</a>

<!-- Fim do link com imagem para o google plus -->


</div>


<nav class="footer-nav">

    <!-- Informação com a criação da marca e o ano atual -->

    <a>

        Copyright &copy; 2020 -

        <?php echo date("Y"); ?> allEletric. Todos os direitos reservados.

    </a>

    <!-- Fim da informação -->

</nav>


</footer>

<!--Fim do footer-->


<!-- Início de código em JavaScript para o botão do histórico na navbar funcionar corretamente -->

<script>

    $(document).ready(function(){

        $(''.dropdown-submenu a.test').on("click", function(e){

            $(this).next('ul').toggle();

            e.stopPropagation();

            e.preventDefault();

        });

    });

</script>

<!-- Fim do código em JavaScript-->
```

</body>

</html>

## 8.16. Código do PHP Sensores.php

<?php

//criação das variáveis para ir buscar os dados aos ficheiros

//para a temperatura

\$valor\_temperatura = file\_get\_contents("api/files/temperatura/valor.txt");

\$hora\_temperatura = file\_get\_contents("api/files/temperatura/hora.txt");

\$nome\_temperatura = file\_get\_contents("api/files/temperatura/nome.txt");

//fim das variaveis da temperatura

//para a bateria

\$valor\_bateria = file\_get\_contents("api/files/bateria/valor.txt");

\$hora\_bateria = file\_get\_contents("api/files/bateria/hora.txt");

\$nome\_bateria = file\_get\_contents("api/files/bateria/nome.txt");

//fim das variaveis da bateria

//para a humidade

\$valor\_humidade = file\_get\_contents("api/files/humidade/valor.txt");

\$hora\_humidade = file\_get\_contents("api/files/humidade/hora.txt");

\$nome\_humidade = file\_get\_contents("api/files/humidade/nome.txt");

//fim das variaveis da humidade

//para a proximidade

\$valor\_proximidade = file\_get\_contents("api/files/proximidade/valor.txt");

\$hora\_proximidade = file\_get\_contents("api/files/proximidade/hora.txt");

\$nome\_proximidade = file\_get\_contents("api/files/proximidade/nome.txt");

//fim das variaveis da proximidade

//para a luminosidade

\$valor\_luminosidade = file\_get\_contents("api/files/luminosidade/valor.txt");

\$hora\_luminosidade = file\_get\_contents("api/files/luminosidade/hora.txt");

\$nome\_luminosidade = file\_get\_contents("api/files/luminosidade/nome.txt");

```
//fim das variaveis da luminosidade

//para a whiteNoise

$valor_whiteNoise = file_get_contents("api/files/whiteNoise/valor.txt");

$hora_whiteNoise = file_get_contents("api/files/whiteNoise/hora.txt");

$nome_whiteNoise = file_get_contents("api/files/whiteNoise/nome.txt");

//fim das variaveis da whiteNoise


//para o CO2

$valor_CO2 = file_get_contents("api/files/CO2/valor.txt");

$hora_CO2 = file_get_contents("api/files/CO2/hora.txt");

$nome_CO2 = file_get_contents("api/files/CO2/nome.txt");

//fim das variaveis do CO2


//inicia a sessão no ficheiro

session_start();


//condição para impedir que se aceda ao ficheiro sem antes ter efetuado o login

if(!isset($_SESSION['username'])){

    header("refresh:3;url=login.php");

    die("Acesso restrito.");

}


?>


<!DOCTYPE html>


<html lang="en">


<head>

    <!-- Logótipo no separador -->

    <link rel="icon" type="image/png" href="favicon-32x32.png" sizes="32x32" />

    <link rel="icon" type="image/png" href="favicon-16x16.png" sizes="16x16" />
```

```
<!-- definição da codificação dos caracteres utilizados -->

<meta charset="UTF-8">


<!-- Página faz refresh de 15 em 15 segundos-->

<meta http-equiv="refresh" content="10">


<!-- Referência ao ficheiro css utilizado -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+1" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="Css2.css">


<!-- Título da página -->

<title>allEletric Sensores</title>


<!-- links para o Dropdown usando javascript -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- Fim das referencias aos links do dropdown-->


</head>

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"> </script>

<body>


<!-- Inclui a Navbar -->

<?php include('navbar.php'); ?>


<br>


<div class="container">

    <div class="card">


        <!-- Alternativa ao jumbotron -->

        <div class="class-header p-3 mb-2 bg-danger text-dark"> <b>Tabela de Sensores</b></div>
```

```
<div class="card-body" >

    <table class="table">

        <!-- Cria o table header -->

        <thead>

            <tr>

                <th scope="col">Tipo de Dispositivo IoT</th>

                <th scope="col">Valor</th>

                <th scope="col">Data de Atualização</th>

                <th scope="col">Estado Alertas</th>

            </tr>

        </thead>

        <!-- Fim do table header -->

        <tbody>

            <tr>

                <!-- Apresenta a informação da Bateria: valor e hora da
ultima atualização -->

                <td>Bateria</td>

                <!-- Apresenta o valor da ultima atualização -->

                <td><?php echo $valor_bateria?> %</td>

                <!-- Apresenta a hora da ultima atualização -->

                <td><?php echo $hora_bateria?></td>

                <!-- Dependendo do 'valor_bateria', muda o aspeto da
mensagem do alerta -->

                <?php if($valor_bateria>75){

                    echo "<td><span class='badge badge-pill badge-
success'>Cheia</span></td>";

                }else

                {

                    if($valor_bateria<=75 && $valor_bateria>30){

                        echo "<td><span class='badge badge-
pill badge-primary'>Normal</span></td>";

                    }else

                    {

                        echo "<td><span class='badge badge-pill badge-
danger'>Baixo</span></td>";
```



```

    }

    ?>

    <!-- Fim da informação sobre a bateria -->

</tr>

<tr>

    <!--Apresenta a informação da Luminosidade: valor e hora
da ultima atualização -->

    <td>Luminosidade</td>

    <!-- Apresenta o valor da ultima atualização -->
    <td><?php echo $valor_luminosidade?></td>

    <!-- Apresenta a hora da ultima atualização -->
    <td><?php echo $hora_luminosidade?></td>

    <!-- Dependendo do 'valor_luminosidade', muda o aspeto da
mensagem do alerta -->

    <?php if($valor_luminosidade=="Elevada"){

        echo "<td><span class='badge badge-pill badge-
danger'>Alto</span></td>";

    }else{

        echo "<td><span class='badge badge-pill badge-
success'>Baixo</span></td>";

    }

    ?>

    <!-- Fim da informação sobre a luminosidade -->

</tr>

<tr>

    <!--Apresenta a informação da Temperatura: valor e hora da
ultima atualização -->

    <td>Temperatura</td>

    <!-- Apresenta o valor da ultima atualização -->
    <td><?php echo $valor_temperatura?> °C</td>

    <!-- Apresenta a hora da ultima atualização -->
    <td><?php echo $hora_temperatura?></td>

    <!-- Dependendo do 'valor_temperatura', muda o aspeto da
mensagem do alerta -->

    <?php if($valor_temperatura>30){

        echo "<td><span class='badge badge-pill badge-
danger'>Alto</span></td>";

    }else

```

```

        {
            if($valor_temperatura<=30                &&
$valor_temperatura>10){
                echo "<td><span class='badge badge-
pill badge-success'>Normal</span></td>";
            }else
                echo "<td><span class='badge badge-pill badge-
primary'>Baixo</span></td>";
        }
    ?>

<!-- Fim da informação sobre a temperatura -->

</tr>

<tr>

<!-- Apresenta a informação dos Pneus: valor e hora da
ultima atualização -->

<td>Humidade</td>

<!-- Apresenta o valor da ultima atualização -->
<td><?php echo $valor_humidade?> %</td>

<!-- Apresenta a hora da ultima atualização -->
<td><?php echo $hora_humidade?></td>

<!-- Dependendo do 'valor_pneus', muda o aspeto da
mensagem do alerta -->

<?php if($valor_humidade>70){
    echo "<td><span class='badge badge-pill badge-
danger'>Humidade alta</span></td>";
}
else
{
    if($valor_humidade<=70                &&
$valor_humidade>30){
        echo "<td><span class='badge badge-
pill badge-success'>Humidade normal</span></td>";
    }else
        echo "<td><span class='badge badge-pill badge-
danger'>Humidade baixa</span></td>";
    }
}
?>

<!-- Fim da informação sobre os pneus -->

```

|                        |   |
|------------------------|---|
|                        | </tr>   |
|                        | <tr>  |
| ultima atualização --> | <!--Apresenta a informação da Proximidade: valor e hora da<br><td>Proximidade</td><br><!-- Apresenta o valor da ultima atualização --><br><td><?php echo \$valor_proximidade?> </td><br><!-- Apresenta a hora da ultima atualização --><br><td><?php echo \$hora_proximidade?></td>   |
| mensagem do alerta --> | <!-- Dependendo do 'valor_proximidade', muda o aspeto da<br><?php if(\$valor_proximidade=="Objeto Próximo"){<br>echo "<td><span class='badge badge-pill badge-<br>danger'> Objeto próximo! </span></td>";<br>}else<br>{<br>echo "<td><span class='badge badge-pill badge-<br>success'> Objeto longínquo </span></td>";<br>}<br>?><br><!-- Fim da informação sobre a proximidade --> |
|                        | </tr>   |
|                        | <tr>  |
| ultima atualização --> | <!--Apresenta a informação da Proximidade: valor e hora da<br><td>White Noise</td><br><!-- Apresenta o valor da ultima atualização --><br><td><?php echo \$valor_whiteNoise?> %</td><br><!-- Apresenta a hora da ultima atualização --><br><td><?php echo \$hora_whiteNoise?></td>  |
| mensagem do alerta --> | <!-- Dependendo do 'valor_proximidade', muda o aspeto da<br><?php if(\$valor_whiteNoise>30){<br>echo "<td><span class='badge badge-pill badge-<br>danger'> Níveis elevados </span></td>";<br>}else<br>{   |

```

success'> Níveis normais </span></td>";

echo "<td><span class='badge badge-pill badge-

}

?>

<!-- Fim da informação sobre a proximidade -->

</tr>

<tr>

<!-- Apresenta a informação da Proximidade: valor e hora da

ultima atualização -->

<td>Valores de CO2</td>

<!-- Apresenta o valor da ultima atualização -->

<td><?php echo $valor_CO2?> %</td>

<!-- Apresenta a hora da ultima atualização -->

<td><?php echo $hora_CO2?></td>

<!-- Dependendo do 'valor_proximidade', muda o aspeto da

mensagem do alerta -->

<?php if($valor_CO2>45){

echo "<td><span class='badge badge-pill badge-

}

}

{

if($valor_CO2>20){

echo "<td><span class='badge badge-

}

}

}

}

?>

<!-- Fim da informação sobre a proximidade -->

</tr>

```

```
        </tbody>

    </table>

</div>

</div>

</div>

<br>

<div class="container">

    <canvas id="myChart" style="width:80%;max-width:1100px"></canvas>

    <script>

        //tendo como valores no eixo dos x os sensores, e no eixo dos y os respetivos valores (colocando -10 e
        100 no fim para definir os intervalos do eixo dos y)

        var xValues = ["Humidade (em %)", "Bateria (em %)", "White Noise (em %)", "Valores de CO2 (em
        %)", "Temperatura (em °C)"];

        var yValues = [<?php echo $valor_humidade; ?>, <?php echo $valor_bateria; ?>, <?php echo
        $valor_whiteNoise;?>, <?php echo $valor_CO2;?>, <?php echo $valor_temperatura;?>, -10, 100];

        var barColors = ["red", "green", "blue", "yellow", "grey"]; //definir as cores para cada barra de sensores

        new Chart("myChart", {

            type: "bar",

            data: {

                labels: xValues,

                datasets: [{

                    backgroundColor: barColors,

                    data: yValues

                }]

            },

            options: {

                responsive: true,

                legend: {display: false},

                title: {

                    display: true,

                    text: "Sensores"

                }

            }

        });

    </script>

</div>

</div>
```

```
</script>

</div>

<br>

<br>

<!-- Início de código em JavaScript para o botão do histórico na navbar funcionar corretamente -->

<script>

    $(document).ready(function(){

        $('.dropdown-submenu a.test').on("click", function(e){

            $(this).next('ul').toggle();

            e.stopPropagation();

            e.preventDefault();

        });

    });

</script>

<!-- Fim do código em JavaScript-->

</body>

</html>
```

## 8.17. Código do PHP Upload.php

```
<?php

header('Content-Type: text/html; charset=utf-8');

// Requer um POST do servidor

if($_SERVER['REQUEST_METHOD'] == "POST"){

    if(isset($_FILES['imagem'])){

        //print_r($_FILES['imagem']); mostra os valores todos numa linha

        // Apresenta o nome, tamanho, e o tipo da imagem.

        echo "<br>".$_FILES['imagem']['name'];

        echo "<br>".$_FILES['imagem']['size'];

    }

}
```

```
echo "<br>".$_FILES['imagem']['type']."<br>";

// Inicia-se a variavel a 0, e guarda todas as fotos no diretorio uploads

$directory = "uploads/";

$filecount = 0;

$files = glob($directory . "*"); //ver o que faz este glob

if ($files){

    $filecount = count($files);

}

// Faz upload da imagem submetida para o diretorio uploads com o nome de webcam + "contador".jpg

if(move_uploaded_file($_FILES['imagem']['tmp_name'], "uploads/webcam".$filecount.".jpg")){

    echo "Imagem descarregada com sucesso";

}else{

    echo "Erro ao fazer upload";

}

}else{

    echo "ERROR: Imagem não encontrada";

}

// Se não receber um metodo POST vai mandar mensagem de erro.

}else{

    echo(http_response_code(403));

    echo "Método POST não encontrado";

}

?>
```

## 8.18. Código do PHP UploadFoto.php

```
<?php

session_start();

// Criação de variaveis

$errors = false;

$target_file = "upload/" . basename($_FILES["imagem"]["name"]);

$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
```

```
function save_file($source){  
    //O nome que vai ficar no ficheiro, e a path  
    if(move_uploaded_file($source,"perfil/fotoPerfil".$_SESSION['username'].".jpg")){  
        echo "Imagem de perfil atualizada com sucesso.";  
    }else{  
        echo "Erro ao fazer upload";  
    }  
}  
  
// Aceita como formatos da imagem, jpg e png  
if($imageFileType != "jpg" &&  
    $imageFileType != "png") {  
    echo "Apenas ficheiros JPG e PNG são permitidos.";  
    $errors = true;  
}  
  
//Verifica se o tamanho do ficheiro submetido não é superior a 1mb  
if ($_FILES["imagem"]["size"] > 1048576) {  
    echo "Ficheiro excede o limite maximo de 1mb.";  
    $errors = true;  
}  
  
if ($errors == true){  
    echo "<p>A redirecionar para a pagina.</p>";  
}  
  
// Se não houver errors, vai submeter a imagem  
if ($errors == false){  
    save_file($_FILES['imagem']['tmp_name']);  
}  
  
// Passado 3 segundos, vai redirecionar o utilizador para a pagina que anteriormente estava  
$url="http://127.0.0.1/projeto/perfil.php?opcao=".$_SESSION['username'];  
header("refresh:3;url=".$url);  
exit;
```



## 8.19. Código de Python capturaWebcamOpenCV.py

```
import cv2 as cv

import sys

import time

import requests

url = "http://127.0.0.1/projeto/upload.php"

urlCamera="http://127.0.0.1/projeto/api/api.php?nome=camera"

urlProx = "http://127.0.0.1/projeto/api/api.php?nome=proximidade"

def datahora():

    dataehora=time.strftime("%Y-%m-%d %H:%M:%S",time.gmtime())

    return dataehora;

camera = cv.VideoCapture(0)

try:

    while True:

        #faz um pedido GET para obter a informação se existe um objeto próximo ou não

        r = requests.get(urlProx)

        if r.status_code == 200:

            print(str(r.status_code))

            if r.text == "Objeto Próximo":

                #tira a foto da webcam indicada na variavel camera

                ret, image=camera.read()

                print("Resultado da Camera="+str(ret))

                #guarda a imagem com o nome webcam.jpg

                cv.imwrite('webcam.jpg',image)

                #faz um pedido POST, enviando a imagem para o upload.php

                files = {'imagem': ('webcam.jpg', open('webcam.jpg','rb'), 'image/jpeg')}

                r = requests.post(url, files = files)

                #ao tirar a foto, envia para a API a informação que a camera está ligada

                valores={'nome': 'camera', 'valor': 1, 'hora': datahora(), 'pass': 'ae123'}

                cam = requests.post(urlCamera,valores)
```

```
        if cam.status_code == 200:

            print("OK: POST realizado com sucesso")

        else:

            print("ERRO: Não foi possível realizar o pedido")

        time.sleep(5)

        if r.status_code != 200:

            print("Erro ao fazer o post")

    else:

        print("Nenhum objeto próximo")

        #envia para a API a informação que a camera está desligada

        valores={'nome': 'camera' , 'valor': 0, 'hora': datahora(), 'pass': 'ae123'}

        cam = requests.post(urlCamera,valores)

        if cam.status_code == 200:

            print("OK: POST realizado com sucesso")

        else:

            print("ERRO: Não foi possível realizar o pedido")

        time.sleep(5)

    else:

        print("O pedido HTTP não foi bem sucedido")

except KeyboardInterrupt:

    print("Except")

finally:

    camera.release()

    cv.destroyAllWindows()

    print("Fim do programa")
```

## 8.20. Código de Python tocaPorta.py

```
import sys
```

```
import time

import simpleaudio

import requests

url="http://127.0.0.1/projeto/api/api.php"

def play_sound(nome):

    wave_obj = simpleaudio.WaveObject.from_wave_file("C:/pythonproj/" + nome)

    play_obj = wave_obj.play() # tocar o audio

    play_obj.wait_done() # espera ate o audio terminar

def datahora():

    datahora=time.strftime("%Y-%m-%d %H:%M:%S",time.gmtime())

    return datahora;

def send_to_api(data):

    r=requests.post(url,data)

    if r.status_code == 200:

        print("OK: POST realizado com sucesso")

    else:

        print("ERRO: Não foi possível realizar o pedido")

try:

    print("Prima CTRL+C para terminar")

    while True:

        r=requests.get('http://127.0.0.1/projeto/api/api.php?nome=portas')

        if r.status_code == 200:

            print("Status code: ")

            print(str(r.status_code))

            print("O numero de portas abertas é: ")

            #se o número de portas abertas for 2, então emite o som e envia para a API a informação que o som está
a ser tocado

            if r.text=="2":

                print(r.text)
```

```
        play_sound("Alarm.wav")

        dataehora = datahora()

        print(dataehora)

        valores={'nome':'buzzer' , 'valor':1, 'hora': datahora(), 'pass': 'ae123'}

        send_to_api(valores)

        print("Tem portas abertas!!")

    else:

        #caso nao haja portas abertas, não emite som, enviando para a API a informação que não está
a ser tocado nada

        dataehora = datahora()

        print(dataehora)

        valores={'nome':'buzzer' , 'valor':0, 'hora': datahora(), 'pass': 'ae123'}

        send_to_api(valores)

        print("Nenhuma porta aberta")

    else:

        print("O pedido HTTP não foi bem sucedido")

    time.sleep(3)

except KeyboardInterrupt:

    print("Programa terminado pelo utilizador")

except:

    print("Ocorreu um erro",sys.exc_info())

finally:

    print("Fim do programa")
```

## 8.21. Código de Python percentagemBateria.py

```
import psutil as ps

import time

import requests

url="http://127.0.0.1/projeto/api/api.php"

def datahora():

    dataehora=time.strftime("%Y-%m-%d %H:%M:%S",time.gmtime())

    return dataehora;
```

```
def send_to_api(data):

    r=requests.post(url,data)

    if r.status_code == 200:

        print("OK: POST realizado com sucesso")

    else:

        print("ERRO: Não foi possível realizar o pedido")

try:

    while(True):

        batterystats = dict(ps.sensors_battery())._asdict() #funcao que cria um "dicionário" com os dados da bateria

        timeleft = batterystats.get('secsleft') #indica o tempo restante de bateria

        percentagem = batterystats.get('percent') #indica a percentagem de bateria restante

        ligado = batterystats.get('power_plugged') #indica se a bateria está a ser carregada

        horas = timeleft/3600

        dataehora = datahora()

        print(dataehora)

        #enviamos apenas a percentagem de bateria para a API

        valores={'nome':'bateria' , 'valor':percentagem, 'hora': datahora(), 'pass': 'ae123'}

        send_to_api(valores)

        time.sleep(10)

except KeyboardInterrupt:

    print("A sair..")

finally:

    print("Fim do programa")
```

## 8.22. Código de Python luzLeitura.py

```
import sys

import requests

from time import *

from msvcrt import (kbhit, getch)

url="http://127.0.0.1/projeto/api/api.php"
```

```
def datahora():

    datahora=strftime("%Y-%m-%d %H:%M:%S",gmtime())

    return datahora;

def send_to_api(data):

    r=requests.post(url,data)

    if r.status_code == 200:

        print("OK: POST realizado com sucesso")

    else:

        print("ERRO: Não foi possível realizar o pedido")

try:

    print("Usage:\n [0] Desligar a luz \n [1] Ligar a luz \n CTRL+C Terminar")

    while True:

        if kbhit():

            tecla=getch() #recebe o valor da tecla que foi carregada

            if tecla == b'0': #se a tecla pressionada for o 0, então envia para a API a informação que é para desligar
a luz de leitura

                datahora = datahora()

                print(datahora)

                valores={'nome': 'luzLeitura' , 'valor': 0, 'hora': datahora(), 'pass' : 'ae123'}

                send_to_api(valores)

                print("\nLuz foi desligada")

            elif tecla == b'1': #se a tecla pressionada for o 1, então envia para a API a informação que é para ligar a
luz de leitura

                datahora = datahora()

                print(datahora)

                valores={'nome': 'luzLeitura' , 'valor': 1, 'hora': datahora(), 'pass' : 'ae123'}

                send_to_api(valores)

                print("\nLuz foi ligada ")

            else:

                print("\nOpção Inválida")

except KeyboardInterrupt:

    print("Programa terminado pelo utilizador")

except:
```

```
print("Ocorreu um erro",sys.exc_info())  
  
finally:  
    print("Fim do programa")
```

?>