

Padaria Inteligente

Bruno Miguel da Maia Silva

João Rodrigo Lopes Gonçalves

Tiago Miguel Marques Amaro

Trabalho de Projeto da unidade curricular de Introdução às Redes e Sistemas Operativos
TeSP PSI

Leiria, janeiro de 2022

Lista de Figuras

Figura 1 - Esboço	2
Figura 2 - Diagrama para representação do Detetor de Fumo.....	3
Figura 3 - Diagrama para representação da Ventilação.....	4
Figura 4 - Diagrama para representação do Ralo de água	4
Figura 5 - Diagrama para representação da Iluminação	5
Figura 6 - Diagrama para representação da Campainha.....	5
Figura 7 - Diagrama para representação do Sistema de Segurança.....	6
Figura - Cenário de Teste	7
Figura - MCU e alguns dos sensores.....	8
Figura - SBC nº1 e atuadores	8
Figura - SBC nº2 e atuadores	8

Lista de siglas e acrónimos

API	<i>Application Programing Interface</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
ESTG	Escola Superior de Tecnologia e Gestão
HTML	<i>Hypertext Markup Language</i>
IPLeia	Instituto Politécnico de Leiria
MCU	Microcontrolador
PHP	<i>PHP Hypertext Preprocessor</i>
SBC	<i>Single Board(ed) Computer</i>

Índice

Lista de Figuras	iv
Lista de siglas e acrónimos.....	v
1. Introdução.....	1
2. Arquitetura	2
3. Implementação	3
4. Cenário de Teste	7
5. Resultados obtidos.....	9
6. Conclusão	10
7. Bibliografia	11
8. Anexos	12
8.1. Código do SBC Número 1.....	13
8.2. Código do SBC Número 2.....	15
8.3. Código do MCU	17
8.4. Código da página index.php.....	20
8.5. Código da página sensor.php	25
8.6. Código do ficheiro função_calculo.php	30
8.7. Código do ficheiro api.php	32
8.8. Código do ficheiro upload.php.....	35
8.9. Código do ficheiro script.js.....	37
8.10. Código do ficheiro camera.py	39

1. Introdução

Neste trabalho tentamos recriar uma padaria com uma nova vida, com isto quero dizer que vamos usar a massa e a farinha com a tecnologia para a padaria ganhar um pouco de vida e não ser apenas mais uma padaria, com isto tudo podemos dizer que o tema do nosso trabalho é uma Padaria Inteligente.

A escolha deste tema foi feita devido a todos os elementos do grupo terem uma mínima noção como é que funciona uma padaria ajudando a perceber e a facilitar o trabalho a todos os elementos do grupo. Os objetivos principais deste trabalho são usar sensores a interagir com atuadores e enviar a informação de cada sensor existente no nosso projeto do *Cisco Packet Tracer* para o website através de programação em PHP e Python. No projeto foram usadas várias aplicações tais como o Cisco Packet Tracer para a criação de um ambiente virtual e o Visual Studio Code para a edição de código. Usamos também linguagens de programação como o PHP, Python, JavaScript e a linguagem de marcação HTML.

2. Arquitetura

Na arquitetura do nosso projeto idealizamos usar um sensor de água que vai ativar-se caso o nível de água esteja acima do normal, um sensor de movimento que assim que detetar algum movimento vai ligar a câmara e o televisor irá mostrar a imagem em tempo real da câmara, um interruptor que serve para ligar a luz, um sensor de fumo que vai ligar um alarme e vai se também ligar os chuveiros caso se detete demasiado fumo, no sensor de temperatura se for detetado calor a ventoinha irá ligar-se e um botão que serve para tocar á campainha .

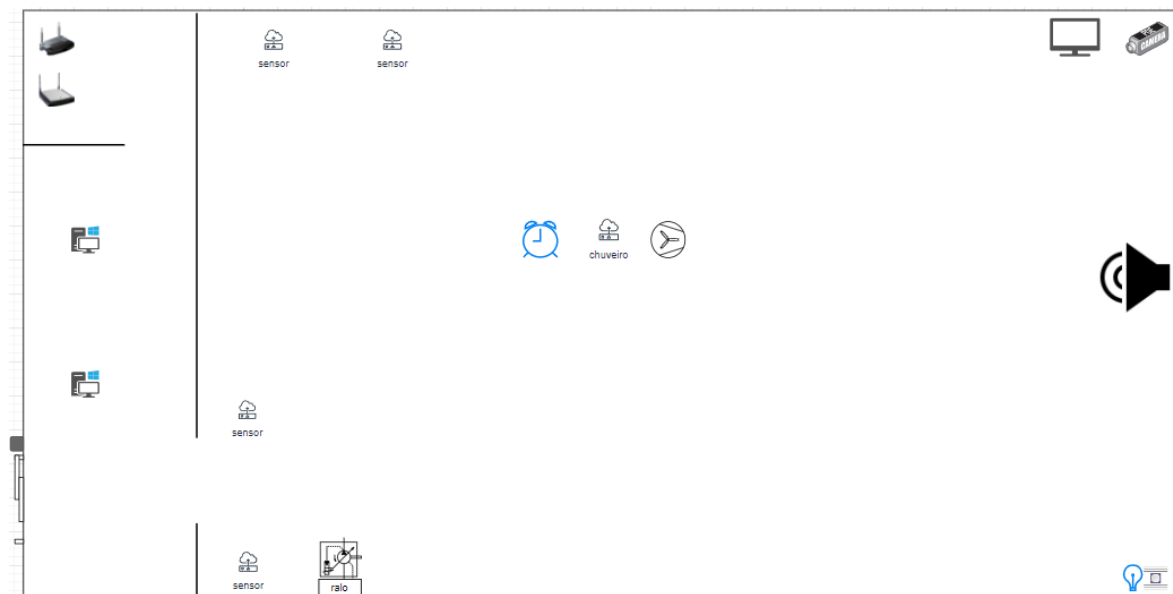


Figura 1 - Esboço

3. Implementação

Os aspetos mais importantes do projeto, foram o detetor de fumo, de temperatura, de movimento, de nível de água; a iluminação e a campainha. Cada um com o propósito de proteger não só o estabelecimento, mas também as pessoas dentro dele.

Detetor de Fumo:

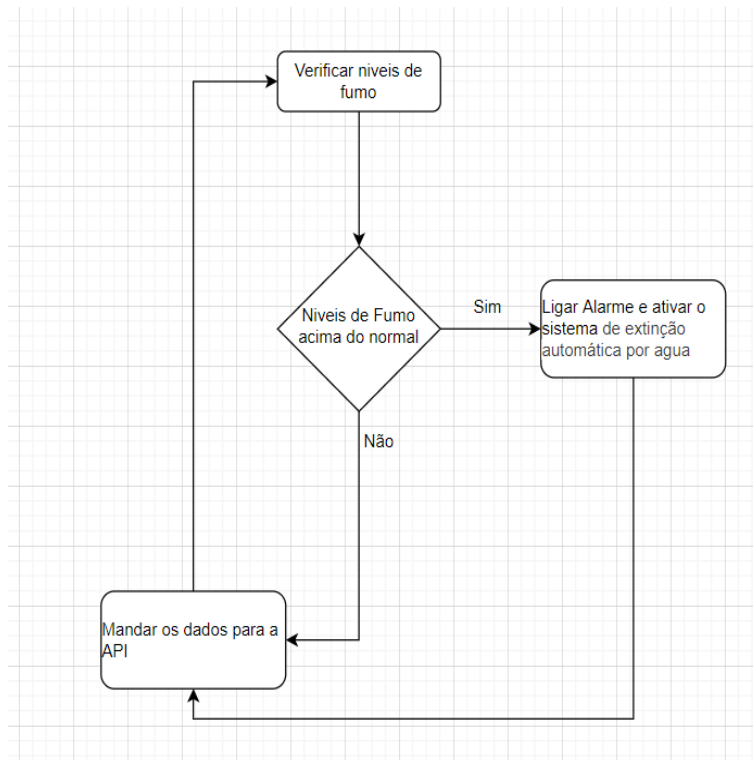


Figura 2 - Diagrama para representação do Detetor de Fumo

Ventilação:

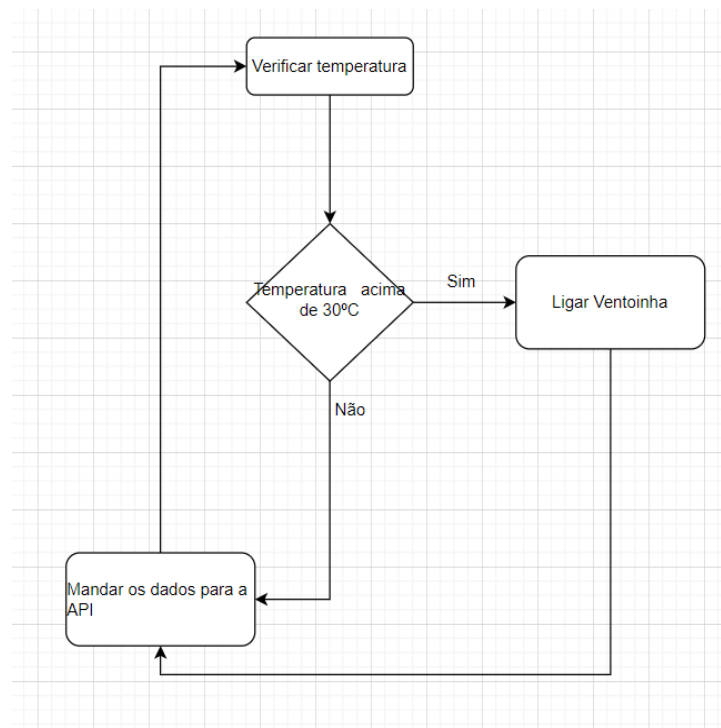


Figura 3 - Diagrama para representação da Ventilação

Ralo de água:

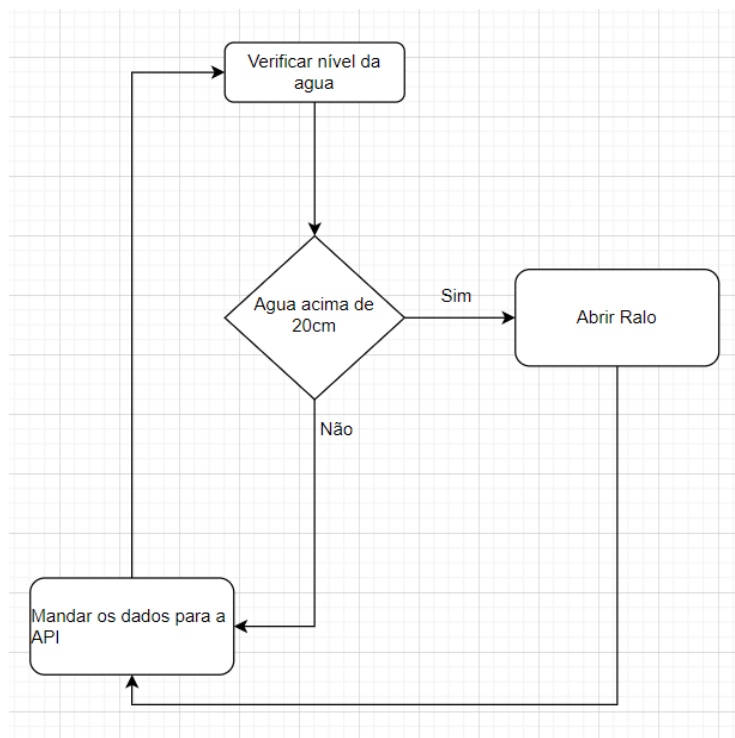


Figura 4 - Diagrama para representação do Ralo de água

Iluminação:

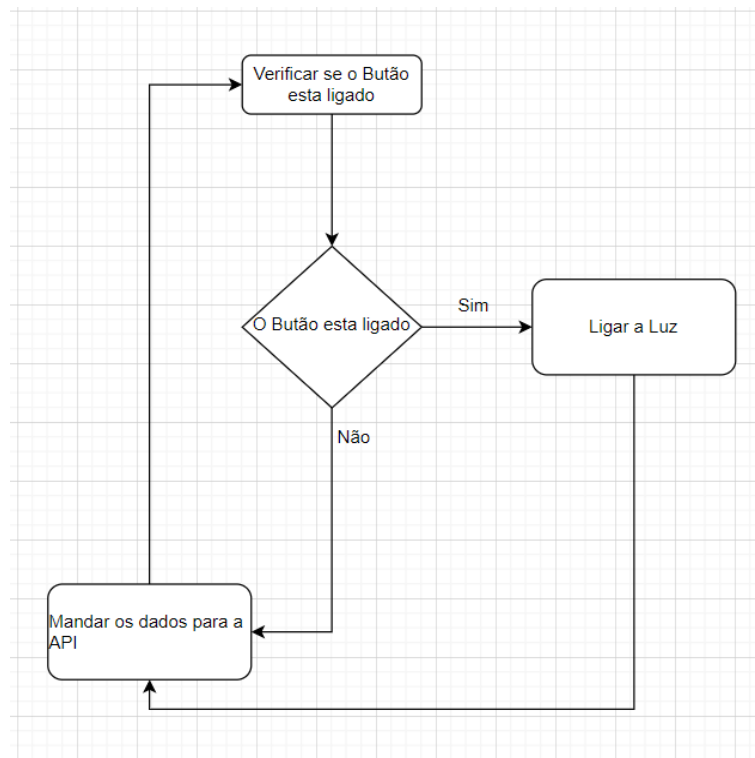


Figura 5 - Diagrama para representação da Iluminação

Campainha:

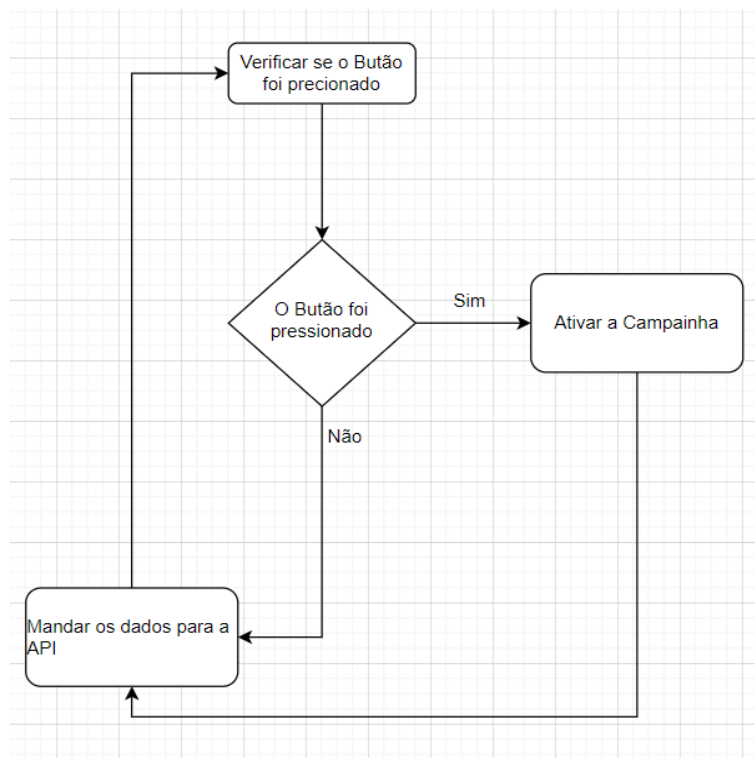


Figura 6 - Diagrama para representação da Campainha

Sistema de Segurança:

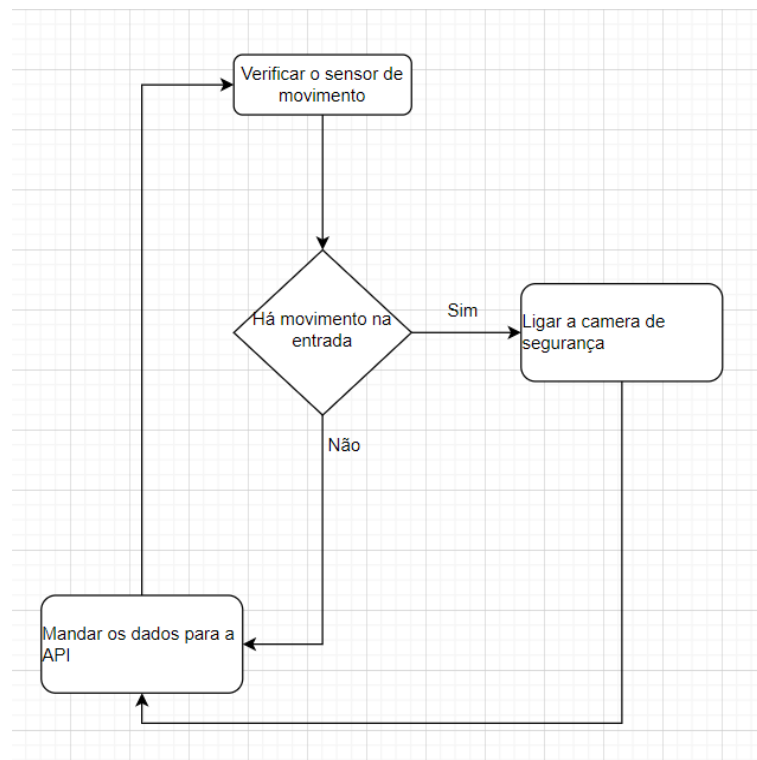


Figura 7 - Diagrama para representação do Sistema de Segurança

4. Cenário de Teste

O nosso cenário de teste é constituído por 1 MCU, 2 SBC, 6 sensores, 8 atuadores, 2 Computadores, 1 *Switch* e 1 *Home Router* que funciona como *Router*, *Access Point* e disponibiliza o serviço de DHCP.

As variáveis de ambiente foram ainda configuradas de forma a que, por exemplo, a ventoinha apenas comece a trabalhar por volta das 12 horas, visto que a temperatura sobe acima dos 30C°.

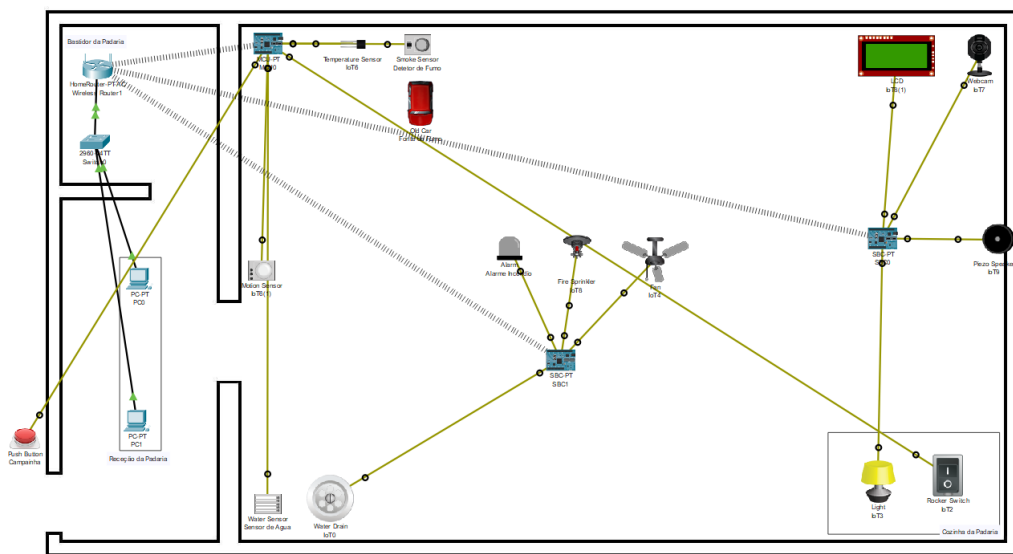


Figura 8 - Cenário de Teste

A parte da rede em si não é nada de especial. É apenas um *Home Router* configurado com uma rede wireless 2.5Ghz à qual o MCU e os SBC's se iram ligar para aceder à internet e comunicar com a API.

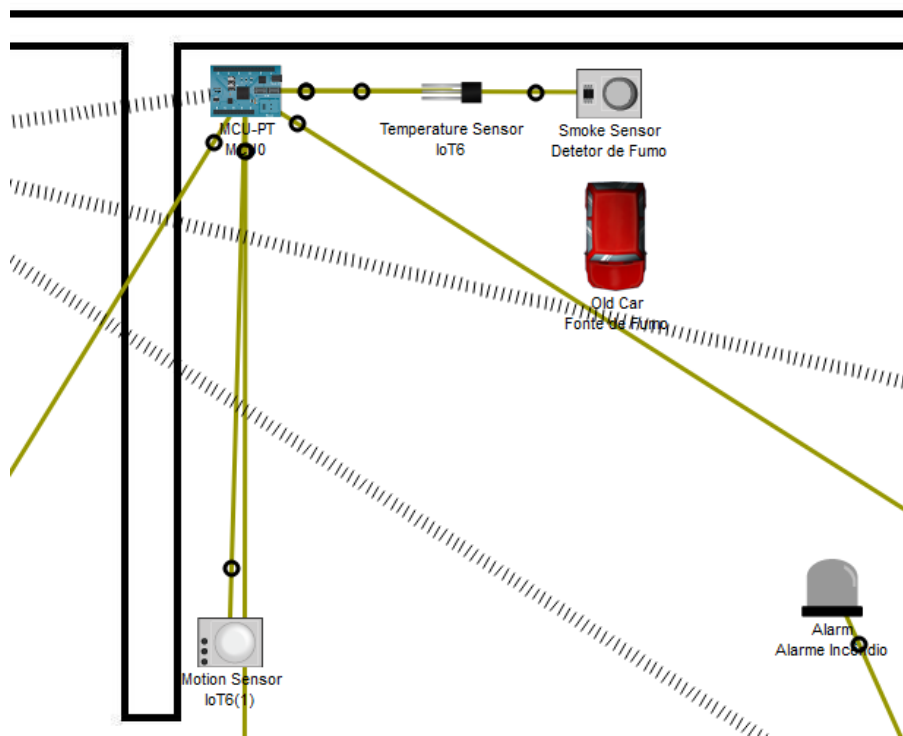


Figura 9 - MCU e alguns dos sensores

O MCU é a placa a que todos os sensores estão ligados. Isto faz com que seja este equipamento a recolher e enviar os estados de cada um dos sensores para a API, onde serão apresentados ao utilizador através do painel de controlo online.

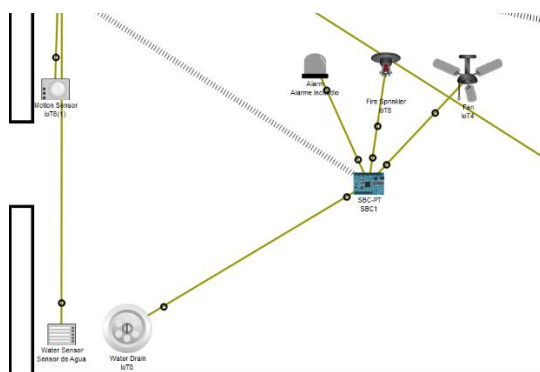


Figura 10 - SBC n°1 e atuadores

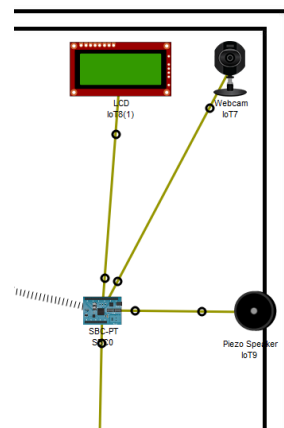


Figura 11 - SBC n°2 e atuadores

Os SBCs são as placas responsáveis por efetuar os pedidos de dados à API, interpretar o seu significado e executar as ações necessárias. Por exemplo, estas são responsáveis por ler os dados relativos à temperatura registada e ligar ou desligar a ventoinha de acordo com os parâmetros programados.

5. Resultados obtidos

Detetor de fumo: utilizando um carro velho como simulador de uma fonte de fumo, e consequentemente, simulando um incêndio, o MCU envia os dados do sensor de fumo constantemente para a API, ativando um alarme e um sistema de extinção automática por água quando a concentração de fumo no ar excede os 39%.

Ventilação: As variáveis de ambiente foram configuradas de tal forma a que a partir de certas horas do dia, a temperatura ultrapasse os 30°C, ativando a ventoinha na potência máxima.

Ralo de água: O sistema de combate a incêndio ativado pelo detetor de fumo inclui um aspersor de supressão de incêndios. Sendo assim, é necessário que exista um ralo de água que permita que esta escoe de forma a prevenir danos causados pela água. Desta forma, o sensor de água deteta o nível da mesma e o ralo é aberto quando este chega a cerca de 15cm.

Iluminação: O SBC identifica que o botão para acender a luz foi ativo e esta acende-se até o mesmo voltar a ser desligado. Estas informações são partilhadas com a API.

Campainha: Ao pressionar um botão, o MCU atualiza o estado da campainha na API para “A tocar...” e o SBC transforma esse estado no toque da campainha.

Sistema de segurança: Quando o sensor de movimento é ativado, a API é notificada e o SBC liga uma câmara de segurança e apresenta uma mensagem no painel LCD informando que a camara se encontra ligada.

6. Conclusão

Para concluir, o projeto da criação de uma padaria inteligente, foi completado, podendo ter sido melhorado no sentido de ter mais ações não só relacionadas com a defesa do estabelecimento, mas também a satisfação do cliente. Futuras alterações podem implementadas de forma a aproveitar melhor o potencial da API através de aplicações móveis ou outros serviços de alerta.

7. Bibliografia

Material disponibilizado pelo docente na plataforma *Moodle*.

8. Anexos

8.1. Código do SBC Número 1

```

1 from realhttp import *
2 from time import *
3 from gpio import *
4 from json import *
5
6 http = RealHTTPClient()
7
8 #D0 -> Ralo
9 #D1 -> Alarme
10 #D2 -> Sprinkler
11 #D3 -> Ventoinha
12
13 def onHTTPOne(status, data):
14     if status != 200:
15         print("fail")
16     else:
17         dados = loads(data)
18         print(dados)
19
20         if(int(dados['detetor_fumo']) > 39):
21             digitalWrite(1, HIGH) # ligar alarme
22             customWrite(2, 1) #ligar aspressor
23         else:
24             digitalWrite(1, LOW) # desligar alarme
25             customWrite(2, 0) #desligar aspressor
26
27         if(int(dados['temperatura']) > 30):
28             customWrite(3, 2) #ligar fan em velocidade maxima
29         else:
30             customWrite(3, 0)
31
32         if(int(dados['ralo_agua']) > 4):
33             customWrite(0, 1) #abrir ralo
34         else:
35             customWrite(0, 0) #fechar ralo
36
37 def main():
38
39     pinMode(0, OUT)
40     pinMode(1, OUT)
41     pinMode(2, OUT)
42     pinMode(3, OUT)
43
44     while True:
45         http.onDone(onHTTPOne)
46         http.get("http://127.0.0.1/api_padaria/api/api.php")
47         sleep(2)
48
49 if __name__ == "__main__":
50     main()

```

8.2. Código do SBC Número 2

```

1 from realhttp import *
2 from time import *
3 from gpio import *
4 from json import *
5
6 http = RealHTTPClient()
7
8 def onHTTPOne(status, data):
9     if status != 200:
10         print("fail")
11     else:
12         dados = loads(data)
13         print(dados)
14
15         if(int(dados['movimento']) == 1):
16             customWrite(6, 1) #ligar camara
17             customWrite(7, "CAMARA ON")
18         else:
19             customWrite(6, 0) #desligar camara
20             customWrite(7, "CAMARA OFF")
21
22         if(int(dados['luz']) == 1):
23             customWrite(5, 2) #ligar luz no maximo
24         else:
25             customWrite(5, 0) # apagar luz
26
27         if(int(dados['campainha']) == 1):
28             digitalWrite(4, HIGH) # tocar campainha
29         else:
30             digitalWrite(4, LOW)
31
32 def main():
33     pinMode(4, OUT) #campainha
34     pinMode(5, OUT) #Luz
35     pinMode(6, OUT) #camara
36     pinMode(7, OUT) #LCD
37
38     while True:
39         http.onDone(onHTTPOne)
40         http.get("http://127.0.0.1/api_padararia/api/api.php")
41         sleep(2)
42
43 if __name__ == "__main__":
44     main()

```

8.3. Código do MCU

```

1 from gpio import *
2 from realhttp import *
3 from time import *
4 import math
5
6 http = RealHTTPClient()
7 url = "http://127.0.0.1/api_padaria/api/api.php"
8
9 def js_map(x, inMin, inMax, outMin, outMax):
10     return (x - inMin) * (outMax - outMin) / (inMax - inMin) + outMin
11
12 def lerFumo(slot):
13     return math.floor(js_map(analogRead(slot), 0, 255, 0, 100) + 0.5)
14
15 def lerTemperatura(slot):
16     return math.floor(js_map(analogRead(slot), 0, 1023, -100, 100) + 0.5)
17
18 def converterLeituraAgua(slot):
19     return math.floor(js_map(analogRead(slot), 0, 255, 0, 20) + 0.5)
20
21 def dataHora():
22     return strftime("%Y/%m/%d %H:%M:%S")
23
24 def onHTTPDone(status, data, replyHeader):
25     if status != 200:
26         print("fail")
27
28 def main():
29     #A0 -> Detetor de Fumo
30     #A1 -> Sensor de Temperatura
31     #A2 -> Sensor de Água
32     #D0 -> Sensor de Movimento
33     #D1 -> Botão Campainha
34     #D2 -> Interruptor
35
36     pinMode(0, IN)
37     pinMode(1, IN)
38     pinMode(2, IN)
39
40     while True:
41         # sensor de fumo
42         detetor_fumo = {'nome': 'detetor_fumo', 'valor': lerFumo(A0)*10, 'hora':
dataHora()}
43         http.post(url, detetor_fumo)
44         http.onDone(onHTTPDone)
45
46         #sensor de temperatura
47
48         temperatura = {'nome': 'temperatura', 'valor': lerTemperatura(A1), 'hora':
dataHora()}
49         http.post(url, temperatura)
50         http.onDone(onHTTPDone)
51
52         #sensor de água
53
54         sensor_agua = {'nome': 'ralo_agua', 'valor': converterLeituraAgua(A2), 'hora':
dataHora()}
55         http.post(url, sensor_agua)
56         http.onDone(onHTTPDone)
57
58         #sensor de movimento
59
60         if(digitalRead(0) == HIGH):
61             valor_movimento = 1
62         else:
63             valor_movimento = 0

```

```
64
65 movimento = {'nome': 'movimento', 'valor': valor_movimento, 'hora': dataHora()}
66 http.post(url, movimento)
67 http.onDone(onHTTPDone)
68
69 #botao campainha
70
71 if(digitalRead(1) == HIGH):
72     valor_campainha = 1
73 else:
74     valor_campainha = 0
75
76 campainha = {'nome': 'campainha', 'valor': valor_campainha, 'hora': dataHora()}
77 http.post(url, campainha)
78 http.onDone(onHTTPDone)
79
80 #interruptor
81
82 if(digitalRead(2) == HIGH):
83     valor_luz = 1
84 else:
85     valor_luz = 0
86
87 luz = {'nome': 'luz', 'valor': valor_luz, 'hora': dataHora()}
88 http.post(url, luz)
89 http.onDone(onHTTPDone)
90
91 sleep(2)
92
93 if __name__ == "__main__":
94     main()
```


8.4. Código da página index.php

```

1 <!DOCTYPE html>
2 <html lang="pt">
3   <head>
4     <title>Padaria Inteligente - Painei de Controlo IOT</title>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
8     <link rel="stylesheet" href="https://cdn.jsdelivrivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
9     <link rel="icon" href="./images/logo.png" />
10    <meta http-equiv="refresh" content="15" />
11  </head>
12  <body>
13    <?php require("./funcao_calculo.php"); ?>
14    <nav class="navbar navbar-dark bg-primary navbar-expand-lg">
15      <div class="container">
16        <a class="navbar-brand" href=".">
17          
18          Padaria Inteligente
19        </a>
20        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNavDropdown" aria-
controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
21          <span class="navbar-toggler-icon"></span>
22        </button>
23        <div class="collapse navbar-collapse" id="navbarNavDropdown">
24          <ul class="navbar-nav">
25            <li class="nav-item">
26              <a class="nav-link active" aria-current="page"
href=".">Inicio</a>
27            </li>
28            <li class="nav-item">
29              <a class="nav-link" id="sensor_detetor_fumo"
href="./sensor.php?nome=detetor_fumo">Detetor de Incêndio</a>
30            </li>
31            <li class="nav-item">
32              <a class="nav-link" id="sensor_temperatura"
href="./sensor.php?nome=temperatura">Ventilação</a>
33            </li>
34            <li class="nav-item">
35              <a class="nav-link" id="sensor_ralo_agua"
href="./sensor.php?nome=ralo_agua">Ralo contra Inundação</a>
36            </li>
37            <li class="nav-item">
38              <a class="nav-link" id="sensor_movimento"
href="./sensor.php?nome=movimento">Sensor Movimento</a>
39            </li>
40            <li class="nav-item">
41              <a class="nav-link" id="sensor_luz" href="./sensor.php?
nome=luz">Iluminação</a>
42            </li>
43            <li class="nav-item">
44              <a class="nav-link" id="sensor_campainha"
href="./sensor.php?nome=campainha">Campainha</a>
45            </li>
46          </ul>
47        </div>
48      </div>
49    </nav>
50    <div class="container">
51      <div class="row">

```

```

52         <div class="col border m-2" id="estado_sistema">
53             <div class="row">
54                 <div class="col">
55                     <div class="text-center">
56                         <h1 class="display-1"><i id="icon_estado_sistema">
</i></h1>
57                     </div>
58                 </div>
59                 <div class="col-10">
60                     <p class="lead" id="titulo_estado_sistema">Estado do
sistema.</p>
61                     <small id="descricao_estado_sistema">Mensagem</small>
62                 </div>
63             </div>
64         </div>
65     </div>
66     <div class="row">
67         <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
68             <div class="card">
69                 <div class="card-header text-center">
70                     Detetor de Incêndio
71                 </div>
72                 <ul class="list-group list-group-flush">
73                     <li class="list-group-item"><?=
file_get_contents("./dados/detetor_fumo/valor.txt") > 39 ? "<span style=\"color:
red;\>Ativo</span>" : "<span style=\"color: green;\>A monitorizar</span>" ?></li>
74                     <li class="list-group-item">Nível de fumo: <?=
file_get_contents("./dados/detetor_fumo/valor.txt") ?>%</li>
75                     <li class="list-group-item"><?=
file_get_contents("./dados/detetor_fumo/hora.txt") ?></li>
76                 </ul>
77                 <div class="card-body">
78                     <a href="./sensor.php?nome=detetor_fumo" class="card-
link">Detalhes</a>
79                 </div>
80             </div>
81         </div>
82         <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
83             <div class="card">
84                 <div class="card-header text-center">
85                     Ventilação
86                 </div>
87                 <ul class="list-group list-group-flush">
88                     <li class="list-group-item"><?=
file_get_contents("./dados/temperatura/valor.txt") > 30 ? "<span style=\"color:
green;\>Ligado</span>" : "<span style=\"color: red;\>Desligado</span>" ?></li>
89                     <li class="list-group-item">Temperatura: <?=
file_get_contents("./dados/temperatura/valor.txt") ?>Cº</li>
90                     <li class="list-group-item"><?=
file_get_contents("./dados/temperatura/hora.txt") ?></li>
91                 </ul>
92                 <div class="card-body">
93                     <a href="./sensor.php?nome=temperatura" class="card-
link">Detalhes</a>
94                 </div>
95             </div>
96         </div>
97         <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
98             <div class="card">
99                 <div class="card-header text-center">
100                     Ralo contra Inundação
101                 </div>
102                 <ul class="list-group list-group-flush">
103                     <li class="list-group-item"><?=
file_get_contents("./dados/ralo_agua/valor.txt") > 4 ? "<span style=\"color:
red;\>Aberto</span>" : "<span style=\"color: green;\>Fechado</span>" ?></li>

```

```

104         <li class="list-group-item">Nível da água: <?=
file_get_contents("./dados/ralo_agua/valor.txt") ?>cm</li>
105         <li class="list-group-item"><?=
file_get_contents("./dados/ralo_agua/hora.txt") ?></li>
106     </ul>
107     <div class="card-body">
108         <a href="./sensor.php?nome=ralo_agua" class="card-
link">Detalhes</a>
109     </div>
110 </div>
111 </div>
112 <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
113     <div class="card">
114         <div class="card-header text-center">
115             Sensor Movimento
116         </div>
117         <ul class="list-group list-group-flush">
118             <li class="list-group-item"><?=
file_get_contents("./dados/movimento/valor.txt") == 1 ? "<span style=\"color:
red;\>Movimento detetado</span>" : "<span style=\"color: green;\>A
monitorizar</span>" ?></li>
119             <li class="list-group-item"><?=
file_get_contents("./dados/movimento/hora.txt") ?></li>
120         </ul>
121         <div class="card-body">
122             <a href="./sensor.php?nome=movimento" class="card-
link">Detalhes</a>
123         </div>
124     </div>
125 </div>
126 <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
127     <div class="card">
128         <div class="card-header text-center">
129             Iluminação
130         </div>
131         <ul class="list-group list-group-flush">
132             <li class="list-group-item"><?=
file_get_contents("./dados/luz/valor.txt") == 1 ? "<span style=\"color:
green;\>Ligado</span>" : "<span style=\"color: red;\>Desligado</span>" ?></li>
133             <li class="list-group-item"><?=
file_get_contents("./dados/luz/hora.txt") ?></li>
134         </ul>
135         <div class="card-body">
136             <a href="./sensor.php?nome=luz" class="card-
link">Detalhes</a>
137         </div>
138     </div>
139 </div>
140 <div class="col-12 col-sm-6 col-md-4 mt-2 text-center">
141     <div class="card">
142         <div class="card-header text-center">
143             Campanha
144         </div>
145         <ul class="list-group list-group-flush">
146             <li class="list-group-item"><?=
file_get_contents("./dados/campanha/valor.txt") == 1 ? "<span style=\"color:
blue;\>A Tocar</span>" : "<span style=\"color: green;\>Pronta</span>" ?></li>
147             <li class="list-group-item"><?=
file_get_contents("./dados/campanha/hora.txt") ?></li>
148         </ul>
149         <div class="card-body">
150             <a href="./sensor.php?nome=campanha" class="card-
link">Detalhes</a>
151         </div>
152     </div>
153 </div>

```

```

154         </div>
155         <div class="row mt-2">
156             <div class="col-5 mt-2 text-center">
157                 Última imagem captada pela camara de videovigilância.
158                 <br>Última Atualização: <?=
file_get_contents("./dados/camara/hora.txt") ?>
159             </div>
160             <div class="col-7 mt-2">
161                 <div class="text-center">
162                     
163                 </div>
164             </div>
165         </div>
166         <footer class="d-flex flex-wrap justify-content-between align-items-
center py-3 my-4 border-top">
167             <div class="col-md-4 d-flex align-items-center">
168                 <a href="/" class="mb-3 me-2 mb-md-0 text-muted text-decoration-
none lh-1">
169                     <svg class="bi" width="30" height="24">
170                         <use xlink:href="#bootstrap"></use>
171                     </svg>
172                 </a>
173                 <span class="text-muted">© 2022 Bruno Silva, João Gonçalves,
Tiago Amaro</span>
174             </div>
175             <ul class="nav col-md-4 justify-content-end list-unstyled d-flex">
176                 <li class="ms-3">
177                     <a class="text-muted"
href="https://github.com/JoaoRodrigoGoncalves/projeto_IRSO">
178                         <i class="bi bi-github"></i>
179                     </a>
180                 </li>
181             </ul>
182         </div>
183     </div>
184     <script src="./js/jquery.min.js"></script>
185     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
187     <script src="./js/script.js"></script>
188     <script>
189         $(document).ready(() => {
190             acionarEstadoSistema(<?= calcularEstadoSistema() ?>);
191         });
192     </script>
193 </body>
194 </html>

```

8.5. Código da página sensor.php

```

1 <!DOCTYPE html>
2 <html lang="pt">
3     <head>
4         <title>Padaria Inteligente - Detalhes</title>
5         <meta charset="utf-8" />
6         <meta name="viewport" content="width=device-width, initial-scale=1" />
7         <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFSpd3yD65VohhpucOmlASjC"
crossorigin="anonymous">
8         <link rel="stylesheet" href="https://cdn.jsdelivrivr.net/npm/bootstrap-
icons@1.7.2/font/bootstrap-icons.css">
9         <link rel="icon" href="./images/logo.png" />
10        <meta http-equiv="refresh" content="15" />
11    </head>
12    <body>
13        <?php
14            if (!isset($_GET['nome'])) {
15                header("Location: ./"); // voltar para trás caso não exista nenhum
sensor especificado
16            }
17
18            require("../funcao_calculo.php");
19
20            /* Nome Sensores */
21
22            $nome_sensor['detetor_fumo'] = "Detetor de Incêndio";
23            $nome_sensor['temperatura'] = "Ventilação";
24            $nome_sensor['ralo_agua'] = "Ralo contra Inundação";
25            $nome_sensor['movimento'] = "Sensor Movimento";
26            $nome_sensor['luz'] = "Iluminação";
27            $nome_sensor['campanha'] = "Campanha";
28
29            /* Unidades por Sensor*/
30
31            //      Nome do Sensor
32            $unidade['detetor_fumo'] = "%";
33            $unidade['temperatura'] = "C°";
34            $unidade['ralo_agua'] = "cm";
35            $unidade['movimento'] = " (0 - LOW / 1 - HIGH)";
36            $unidade['luz'] = " (0 - LOW / 1 - HIGH)";
37            $unidade['campanha'] = " (0 - LOW / 1 - HIGH)";
38
39
40
41            /* Estados dos Sensores */
42
43            $estado_sensor['detetor_fumo']['trip_value'] = 39;
44            $estado_sensor['detetor_fumo']['on'] = "<span style=\"color:
red;\>Alarme Ativo</span>";
45            $estado_sensor['detetor_fumo']['off'] = "<span style=\"color: green;\>A
monitorizar...</span>";
46
47            $estado_sensor['temperatura']['trip_value'] = 30;
48            $estado_sensor['temperatura']['on'] = "<span style=\"color:
green;\>Ligado</span>";
49            $estado_sensor['temperatura']['off'] = "<span style=\"color:
red;\>Desligado</span>";
50
51            $estado_sensor['ralo_agua']['trip_value'] = 4;
52            $estado_sensor['ralo_agua']['on'] = "<span style=\"color:
green;\>Aberto</span>";
53            $estado_sensor['ralo_agua']['off'] = "<span style=\"color:
red;\>Fechado</span>";
54

```

```

55         $estado_sensor['movimento']['trip_value'] = 0; //verificado como maior
que 0 (bool)
56         $estado_sensor['movimento']['on'] = "<span style=\"color:
red;\">Movimento Detetado</span>";
57         $estado_sensor['movimento']['off'] = "<span style=\"color: green;\">A
monitorizar...</span>";
58
59         $estado_sensor['luz']['trip_value'] = 0; //verificado como maior que 0
(bool)
60         $estado_sensor['luz']['on'] = "<span style=\"color:
green;\">Ligada</span>";
61         $estado_sensor['luz']['off'] = "<span style=\"color:
red;\">Desligada</span>";
62
63         $estado_sensor['campinha']['trip_value'] = 0; //verificado como maior
que 0 (bool)
64         $estado_sensor['campinha']['on'] = "<span style=\"color: blue;\">A
tocar...</span>";
65         $estado_sensor['campinha']['off'] = "<span style=\"color: green;\">À
espera</span>";
66     ?>
67     <nav class="navbar navbar-dark bg-primary navbar-expand-lg">
68         <div class="container">
69             <a class="navbar-brand" href="."/>
70                 
71                 Padaria Inteligente
72             </a>
73             <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNavDropdown" aria-
controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
74                 <span class="navbar-toggler-icon"></span>
75             </button>
76             <div class="collapse navbar-collapse" id="navbarNavDropdown">
77                 <ul class="navbar-nav">
78                     <li class="nav-item">
79                         <a class="nav-link" href="."/>Inicio</a>
80                     </li>
81                     <li class="nav-item">
82                         <a class="nav-link" id="sensor_detetor_fumo"
href="./sensor.php?nome=detetor_fumo">Detetor de Incêndio</a>
83                     </li>
84                     <li class="nav-item">
85                         <a class="nav-link" id="sensor_temperatura"
href="./sensor.php?nome=temperatura">Ventilação</a>
86                     </li>
87                     <li class="nav-item">
88                         <a class="nav-link" id="sensor_ralo_agua"
href="./sensor.php?nome=ralo_agua">Ralo contra Inundação</a>
89                     </li>
90                     <li class="nav-item">
91                         <a class="nav-link" id="sensor_movimento"
href="./sensor.php?nome=movimento">Sensor Movimento</a>
92                     </li>
93                     <li class="nav-item">
94                         <a class="nav-link" id="sensor_luz" href="./sensor.php?
nome=luz">Iluminação</a>
95                     </li>
96                     <li class="nav-item">
97                         <a class="nav-link" id="sensor_campinha"
href="./sensor.php?nome=campinha">Campinha</a>
98                     </li>
99                 </ul>
100             </div>
101         </div>
102     </nav>

```



```

103         <div class="container mt-2">
104             <div class="row">
105                 <div class="col border m-2" id="estado_sistema">
106                     <div class="row">
107                         <div class="col">
108                             <div class="text-center">
109                                 <h1 class="display-1"><i id="icon_estado_sistema">
</i></h1>
110                             </div>
111                         </div>
112                         <div class="col-10">
113                             <p class="lead" id="titulo_estado_sistema">Estado do
sistema.</p>
114                             <small id="descricao_estado_sistema">Mensagem</small>
115                         </div>
116                     </div>
117                 </div>
118             </div>
119             <div class="row">
120                 <div class="col-12 col-sm-5">
121                     <div class="card">
122                         <div class="card-header text-center"><? =
$nome_sensor[$_GET["nome"]] ?></div>
123                         <ul class="list-group list-group-flush">
124                             <li class="list-group-item"><? =
file_get_contents("./dados/" . $_GET["nome"] . "/valor.txt") >
$estado_sensor[$_GET['nome']][ 'trip_value' ] ? $estado_sensor[$_GET['nome']][ 'on' ] :
$estado_sensor[$_GET['nome']][ 'off' ] ?></li>
125                             <li class="list-group-item">Valor: <? =
file_get_contents("./dados/" . $_GET["nome"] . "/valor.txt") ?><? =
$unidade[$_GET['nome']] ?></li>
126                             <li class="list-group-item">Última Atualização: <? =
file_get_contents("./dados/" . $_GET["nome"] . "/hora.txt") ?></li>
127                         </ul>
128                     </div>
129                 </div>
130                 <div class="col-12 col-sm-7">
131                     <div class="card">
132                         <div class="card-header">Histórico do Sensor</div>
133                         <ul class="list-group list-group-flush" style="max-height:
54vh; overflow-y: scroll;">
134                             <?php
135                                 $list = file_get_contents("./dados/" . $_GET["nome"]
. "/log.txt");
136                                 $dados = explode("\n", $list);
137                                 if(count($dados) > 1)
138                                 {
139                                     foreach (array_reverse($dados) as $key => $linha)
140                                     {
141                                         if (!empty($linha) && ($key !=
count($dados)-1)) { // primeira linha é uma quebra de linha
142                                             list($data_hora, $valor) = explode(";",
$linha);
143                                             echo "<li class=\"list-group-item\">[" .
$data_hora . "]" Valor: " . $valor . $unidade[$_GET['nome']] . "</li>";
144                                         }
145                                     }
146                                 }
147                                 else
148                                 {
149                                     echo "<li class=\"list-group-item text-
center\">Sem informação a mostrar</li>";
150                                 }
151                             ?>
152                         </ul>
</div>

```

```

153         </div>
154     </div>
155     <footer class="d-flex flex-wrap justify-content-between align-items-
center py-3 my-4 border-top">
156         <div class="col-md-4 d-flex align-items-center">
157             <a href="/" class="mb-3 me-2 mb-md-0 text-muted text-decoration-
none lh-1">
158                 <svg class="bi" width="30" height="24">
159                     <use xlink:href="#bootstrap"></use>
160                 </svg>
161             </a>
162             <span class="text-muted">© 2022 Bruno Silva, João Gonçalves,
Tiago Amaro</span>
163         </div>
164         <ul class="nav col-md-4 justify-content-end list-unstyled d-flex">
165             <li class="ms-3">
166                 <a class="text-muted"
href="https://github.com/JoaoRodrigoGoncalves/projeto_IRSO">
167                     <i class="bi bi-github"></i>
168                 </a>
169             </li>
170         </ul>
171     </footer>
172 </div>
173 <script src="./js/jquery.min.js"></script>
174 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
176 <script src="./js/script.js"></script>
177 <script>
178     $(document).ready(() => {
179         $("#sensor_<?=$_GET['nome'] ?>").addClass("active").attr("aria-
current", "page");
180         acionarEstadoSistema(<?=$_GET['nome'] ?>);
181     });
182 </script>
183 </body>
184 </html>

```

8.6. Código do ficheiro função_calculo.php

```
1 <?php
2     function calcularEstadoSistema()
3     {
4         // coloamos os objetos de datas com a timezone correta numa array
5         $sensor[0] = new DateTime(file_get_contents("./dados/detector_fumo/hora.txt"),
new DateTimeZone('Europe/Lisbon'));
6         $sensor[1] = new DateTime(file_get_contents("./dados/temperatura/hora.txt"),
new DateTimeZone('Europe/Lisbon'));
7         $sensor[2] = new DateTime(file_get_contents("./dados/ralo_agua/hora.txt"),
new DateTimeZone('Europe/Lisbon'));
8         $sensor[3] = new DateTime(file_get_contents("./dados/movimento/hora.txt"),
new DateTimeZone('Europe/Lisbon'));
9         $sensor[4] = new DateTime(file_get_contents("./dados/luz/hora.txt"), new
DateTimeZone('Europe/Lisbon'));
10        $sensor[5] = new DateTime(file_get_contents("./dados/campainha/hora.txt"),
new DateTimeZone('Europe/Lisbon'));
11        $sensor[6] = new DateTime(file_get_contents("./dados/camara/hora.txt"), new
DateTimeZone('Europe/Lisbon'));
12
13        $agora = new DateTime();
14        foreach ($sensor as $value) { //loop pela array. caso exista algum sensor que
n o tenha sido atualizado nos  ltimos 10 segundos, devolve a string "false"
15            if($agora->getTimestamp() - $value->getTimestamp() > 10)
16            {
17                return "false";
18            }
19        }
20        return "true";
21    }
22 ?>
```

8.7. Código do ficheiro api.php

```

1 <?php
2     header('Content-Type: text/html; charset=utf-8');
3
4     if($_SERVER['REQUEST_METHOD'] == "POST"){
5         if(isset($_POST['valor']) && isset($_POST['nome']) && isset($_POST['hora']))){
6             switch($_POST['nome']){
7                 case 'detetor_fumo':
8                     file_put_contents("../dados/detetor_fumo/valor.txt",
9 $_POST['valor']);
10                    file_put_contents("../dados/detetor_fumo/hora.txt",
11 $_POST['hora']);
12                    $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
13                    file_put_contents("../dados/detetor_fumo/log.txt", $data,
14 FILE_APPEND);
15                    break;
16
17                    case 'temperatura':
18                        file_put_contents("../dados/temperatura/valor.txt",
19 $_POST['valor']);
20                        file_put_contents("../dados/temperatura/hora.txt",
21 $_POST['hora']);
22                        $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
23                        file_put_contents("../dados/temperatura/log.txt", $data,
24 FILE_APPEND);
25                        break;
26
27                    case 'ralo_agua':
28                        file_put_contents("../dados/ralo_agua/valor.txt",
29 $_POST['valor']);
30                        file_put_contents("../dados/ralo_agua/hora.txt", $_POST['hora']);
31                        $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
32                        file_put_contents("../dados/ralo_agua/log.txt", $data,
33 FILE_APPEND);
34                        break;
35
36                    case 'movimento':
37                        file_put_contents("../dados/movimento/valor.txt",
38 $_POST['valor']);
39                        file_put_contents("../dados/movimento/hora.txt", $_POST['hora']);
40                        $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
41                        file_put_contents("../dados/movimento/log.txt", $data,
42 FILE_APPEND);
43                        break;
44
45                    case 'luz':
46                        file_put_contents("../dados/luz/valor.txt", $_POST['valor']);
47                        file_put_contents("../dados/luz/hora.txt", $_POST['hora']);
48                        $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
49                        file_put_contents("../dados/luz/log.txt", $data, FILE_APPEND);
50                        break;
51
52                    case 'campainha':
53                        file_put_contents("../dados/campainha/valor.txt",
54 $_POST['valor']);
55                        file_put_contents("../dados/campainha/hora.txt", $_POST['hora']);
56                        $data = "\n" . $_POST['hora'] . ";" . $_POST['valor'];
57                        file_put_contents("../dados/campainha/log.txt", $data,
58 FILE_APPEND);
59                        break;
60
61                    default:
62                        http_response_code(400);
63            }
64        }else{
65            http_response_code(400);
66        }
67    }

```

```
55     }else{
56         if($_SERVER['REQUEST_METHOD'] == "GET"){
57             $response = array(
58                 'detetor_fumo' =>
59                 file_get_contents("../dados/detetor_fumo/valor.txt"),
60                 'temperatura' => file_get_contents("../dados/temperatura/valor.txt"),
61                 'ralo_agua' => file_get_contents("../dados/ralo_agua/valor.txt"),
62                 'movimento' => file_get_contents("../dados/movimento/valor.txt"),
63                 'luz' => file_get_contents("../dados/luz/valor.txt"),
64                 'campainha' => file_get_contents("../dados/campainha/valor.txt")
65             );
66             echo json_encode($response);
67         }else{
68             http_response_code(405);
69         }
70     }
71 }>
```

8.8. Código do ficheiro upload.php


```
1 <?php
2     if($_SERVER['REQUEST_METHOD'] == "POST")
3     {
4         if(isset($_FILES['imagen']))
5         {
6             if(move_uploaded_file($_FILES['imagen']['tmp_name'],
7                 "../images/camara.jpg"))
8             {
9                 file_put_contents("../datos/camara/hora.txt", date("Y/m/d H:i:s"));
10                http_response_code(200); // ok
11            }
12            else
13            {
14                http_response_code(500);
15            }
16        }
17        else
18        {
19            http_response_code(400);
20        }
21    }
22    else
23    {
24        http_response_code(405);
25    }
26 ?>
```

8.9. Código do ficheiro script.js

```
1 function acionarEstadoSistema(e)
2 {
3     if(e)
4     {
5         $("#estado_sistema").addClass("border-success");
6         $("#icon_estado_sistema").addClass("bi bi-check-lg");
7         $("#titulo_estado_sistema").text("Estado do Sistema: Tudo OK.");
8         $("#descricao_estado_sistema").text("Tudo está OK.");
9     }
10    else
11    {
12        $("#icon_estado_sistema").addClass("bi bi-exclamation-triangle");
13        $("#estado_sistema").addClass("border-danger");
14        $("#titulo_estado_sistema").text("Estado do Sistema: Alguns Problemas.");
15        $("#descricao_estado_sistema").text("Existem alguns problemas com o sistema.
16        Confirme que todos os sensores estão ligados.");
17    }
18 }
```

8.10. Código do ficheiro camera.py

```

1 #codigo para tirar foto
2 #com um delay de 3 segundos
3
4 import sys
5 from turtle import delay
6 import cv2 as cv
7 from requests import (post, get)
8
9 camera = cv.VideoCapture(0)
10 delay = 3000
11 url = 'http://127.0.0.1/api_padaria/api/upload.php'
12
13 def send_file(file):
14     print("A enviar...")
15     r = post(url, files=file)
16     print(r.status_code, " -- ", r.text)
17
18 try:
19     print("Prima CTRL+C para terminar")
20     while True:
21         print("-----")
22         print("A capturar imagem")
23         ret, image = camera.read()
24         print("Resultado " + str(ret))
25         print("A Guardar")
26         cv.imwrite('camara.jpg', image)
27         file = {'imagem': open('camara.jpg', 'rb')}
28         send_file(file)
29         print("Next in: " + str(delay/1000) + " sec.")
30         cv.waitKey(delay)
31
32 except KeyboardInterrupt:
33     print("Terminado pelo Util")
34
35 except :
36     print("Ocorreu um erro", sys.exc_info())
37
38 finally:
39     print("Fim")
40     camera.release()
41     cv.destroyAllWindows()

```