

# SMART ENERGY



**Programação de Dispositivos Móveis**  
**Licenciatura em Engenharia de Sistemas Informáticos**  
**Regime Pós-Laboral**  
**2022/2023**

## **Alunos**

Nuno Mendes – N° 2727  
Tiago Azevedo - N° 21153  
Francisco Pereira – N° 21156

## **Orientação**

Profº Lourenço Gomes

## Índice

|      |  |    |
|------|--|----|
| 1.   | Introdução.....  | 1  |
| 2.   | Proposta de sistema .....                                | 2  |
| 2.1. | Requisitos Funcionais .....                              | 2  |
| 2.2. | Requisitos Não Funcionais.....                           | 3  |
| 3.   | Levantamento de Requisitos .....                         | 4  |
| 3.1. | Diagrama de Casos de uso.....                            | 4  |
| 3.2. | Diagrama de Modelo de Dados (ER).....                    | 5  |
| 3.3. | Diagramas de Atividade.....                              | 6  |
| 3.4. | Diagrama de Estados .....                                | 7  |
| 3.5. | Diagramas de Sequência .....                             | 8  |
| 4.   | Prototipagem .....                                       | 9  |
| 4.1. | Tecnologias utilizadas.....                              | 9  |
| 4.2. | Mockups .....  | 13 |
| 5.   | Código implementado .....                                | 18 |
| 5.1. | Código de SETR .....                                     | 18 |
| 5.2. | Integração dos postes na aplicação móvel e servidor..... | 27 |
| 5.3. | Código de PDM .....                                      | 36 |
| 5.4. | Resultado obtido .....                                   | 36 |
| 6.   | Conclusão.....   | 38 |
| 7.   | Bibliografia .....                                       | 39 |

## Índice de ilustrações

|  |    |
|--|----|
| Figura 1 - Diagrama de Casos de Uso.....                                   | 4  |
| Figura 2 - Diagrama Entidade-Relação.....                                  | 5  |
| Figura 3 - Diagrama de atividade de Inicialização.....                     | 6  |
| Figura 4 - Diagrama de atividade de Detecção de Movimento.....             | 6  |
| Figura 5 - Diagrama de estados .....                                       | 7  |
| Figura 6 - Diagrama de sequência do Movimento Não Ativa a Luz.....         | 8  |
| Figura 7 - Diagrama de sequência do Movimento Ativa a Luz .....            | 8  |
| Figura 8 - Circuito eletrónico no TinkerCad .....                          | 11 |
| Figura 9 - Esquema do circuito eletrónico .....                            | 12 |
| Figura 10 - Esboço do Sistema em papel .....                               | 13 |
| Figura 11 - Diagrama de Hardware de UM poste de iluminação .....           | 14 |
| Figura 12 - Diagrama de Hardware de TODO o Sistema.....                    | 15 |
| Figura 13 - Protótipo da Maquete do Sistema (visão de cima) .....          | 16 |
| Figura 14 - Protótipo da Maquete do Sistema (visão lateral).....           | 16 |
| Figura 15 - Mockups da App Móvel .....                                     | 17 |
| Figura 16 - Serviço Web de um Poste de Iluminação #1 .....                 | 25 |
| Figura 17 - Serviço Web de um Poste de Iluminação #2 .....                 | 26 |
| Figura 18 - Serviço Web de um Poste de Iluminação #3 .....                 | 26 |
| Figura 19 - Printscreen da App: Listagem de dispositivos .....             | 36 |
| Figura 20 - Printscreen da App: Login .....                                | 36 |
| Figura 21 - Printscreen da App: Detalhes de um Poste conectado #2 .....    | 37 |
| Figura 22 - Printscreen da App: Detalhes de um Poste conectado #1 .....    | 37 |
| Figura 23 - Printscreen da App: Detalhes de um Poste sem comunicação ..... | 37 |
| Figura 24 - Printscreen da App: Detalhes de um Poste conectado #3 .....    | 37 |

## 1. Introdução

No âmbito da UC de Programação de Dispositivos Móveis pretende-se desenvolver uma aplicação móvel que possa ser utilizado como resposta a uma necessidade específica. Este tema provém da proposta apresentada na unidade curricular de Projeto Aplicado, que pretende interligar várias UC's com o mesmo projeto.

A nossa equipa optou por escolher o setor energético considerando que o mesmo tem uma boa base para melhorias em termos de eficiência e, por conseguinte, em termos económicos. Um caso prático do nosso projeto será, por exemplo, otimizar a utilização das lâmpadas dos postes de iluminação das estradas com o objetivo de rentabilizar e prolongar o tempo de vida útil das lâmpadas, evitando assim, manutenções desnecessárias e, por consequência, reduzir a mão de obra de manutenção.

Neste momento não existe qualquer tipo de automatização no que toca ao controlo das luzes nos parques de estacionamento e vias públicas. Uma grande preocupação nos dias de hoje é o gasto excessivo de eletricidade, até porque grande parte dessa energia provém de combustíveis fósseis, levando a uma pegada de carbono significativa.

Este projeto poderá também contribuir para as metas da Comissão Europeia para as Smart Cities em 2030, que atualmente se encontram em risco de não serem alcançadas.

## 2. Proposta de sistema

Abaixo poderá ser observado o ponto da situação atual, em termos dos Requisitos Funcionais e Não Funcionais que foram prontamente levantados pela nossa equipa para obter, desta forma, a fundação daquilo que é esperado, uma vez que o projeto esteja na fase de lançamento.

### 2.1. Requisitos Funcionais

É sabido que os Requisitos Funcionais são a definição daquilo que o sistema poderá fazer, ou seja, a materialização de uma ou várias necessidades realizadas em prol do sistema. Estes requisitos terão um impacto substancial no sucesso do projeto. É essencial identificar essas regras o quanto antes, de modo que não haja falhas de comunicação. É importante frisar que estas regras podem sofrer alterações consoante as necessidades/adversidades que eventualmente possam surgir.

- **RF01: Gestão de Energia** – Através de sensores de luz, determinar qual o momento em que os postes de iluminação devem ser ligados/desligados mediante a hora e nível de luminosidade do momento (dia/noite);
- **RF02: Iluminação mediante movimento** – Desligar a iluminação ou reduzir a intensidade para um valor mínimo em caso de ausência de movimento;
- **RF03: Dados em Tempo Real** – Comunicação centralizada dos dados atuais (ID do Arduino; valor do input da quantidade de luz; valor do output de iluminação e Input de deteção de movimento).

## 2.2. Requisitos Não Funcionais

Por outro lado, os Requisitos Não Funcionais definem o que é que o sistema fará mais concretamente. São premissas, restrições técnicas e necessidades que não podem ser atendidas através de funcionalidades. Estes Requisitos Não Funcionais (RNF) inerentes ao projeto, estão associados à qualidade e segurança da aplicação que garante o funcionamento otimizado de todo o sistema.

- **RNF01: Base de dados** – O armazenamento de dados deverá ser efetuado recorrendo a uma linguagem de base de dados que envolva SQL;
- **RNF02: Aplicação móvel** – A gestão do sistema, desde consultas à base de dados até ao estado de manutenção e monitorização do mesmo, deverá ser acedido através de uma App;
- **RNF03: Idioma da aplicação** – O sistema deverá ser capaz de ser totalmente traduzido do português para o inglês e vice-versa, de modo a qualquer pessoa ser capaz de compreender todas as funcionalidades do sistema;
- **RNF04: Conta de utilizador** – Qualquer utilizador autorizado a fazer uso do sistema terá de possuir uma conta feita de forma manual pelo responsável da segurança da infraestrutura.

### 3. Levantamento de Requisitos

#### 3.1. Diagrama de Casos de uso

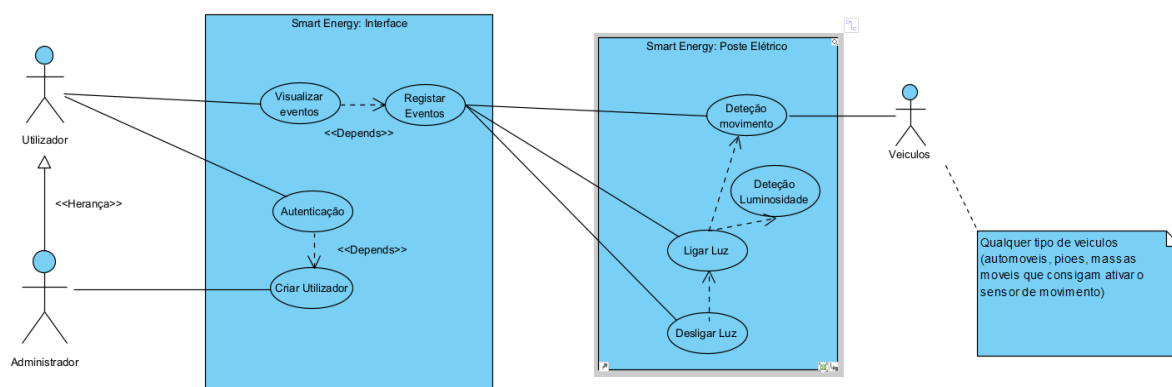


Figura 1 - Diagrama de Casos de Uso

Este diagrama foi concebido para demonstrar os casos de uso presentes no nosso projeto. Este, encontra-se dividido em dois sistemas, que se encontram interligados entre si. O primeiro sistema, “Smart Energy: Interface” é responsável pela visualização de eventos, pela autenticação e pela criação de utilizadores. O segundo sistema, “Smart Energy: Poste Elétrico”, é responsável pela deteção do movimento e pela deteção da luminosidade. Até ao momento, existem três atores: **Administrador, Utilizador e Veiculos**.

O Administrador será responsável por criar utilizadores , bem como tudo o que o utilizador é capaz de fazer, existindo aqui uma relação de herança para com o Utilizador.

Já o Utilizador poderá fazer o uso supervisionado do sistema, nomeadamente para recolha de dados e manutenção da aplicação, visualizando os eventos existentes até à data.

O ator Veiculos, representa qualquer tipo de veículos, sejam eles automóveis, sejam peões, entre outros, desde que consigam ativar o sensor de movimento.

Este diagrama poderá ser alterado mais tarde mediante as necessidades inerentes ao projeto.

### 3.2. Diagrama de Modelo de Dados (ER)

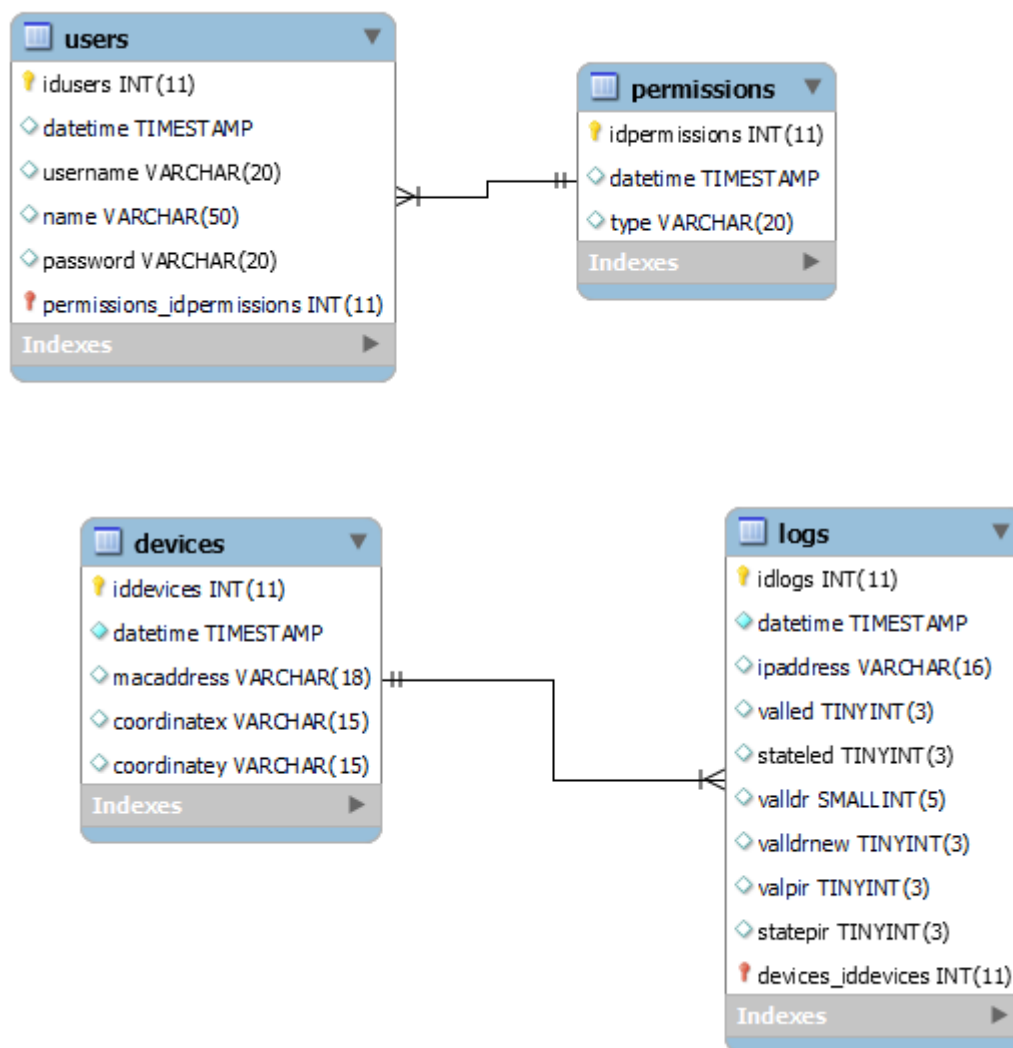


Figura 2 - Diagrama Entidade-Relação

O diagrama Entidade-Relação foi construído já com a concessão da base de dados em mente para que todo o projeto seja alvo de análise posterior. De modo a garantir isso, foram criadas quatro tabelas:

- *Users & Permissions*: Permitem guardar permanentemente os Utilizadores do sistema, assim como as suas respetivas permissões (ex: Administrador);
- *Devices & Logs*: Nestas duas tabelas interligadas, são guardados todos os dados relativamente aos dispositivos que fazem parte do sistema e, de



forma a obter dados concretos da análise dos mesmos, são guardados dados de diagnóstico ou históricos para poderem ser consultados na eventualidade de ser necessário, mediante uma data ou até mesmo um código de dispositivo.

Em suma, o desenvolvimento deste diagrama e a respetiva base de dados são um paço crucial para garantir um sistema seguro e viável de ser expandido sem a necessidade de guardar dados noutro tipo de formato.

### 3.3. Diagramas de Atividade

Os diagramas de atividade são um tipo de diagrama que mostra o fluxo de atividades ou ações dentro do nosso sistema. É uma representação visual da sequência de atividades que devem ser realizadas para concluir os processos ou fluxos de trabalho.

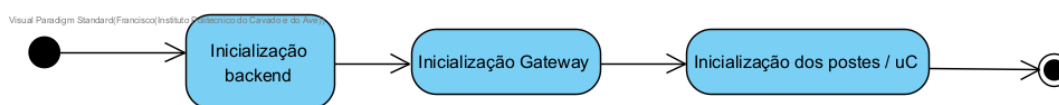


Figura 3 - Diagrama de atividade de Inicialização

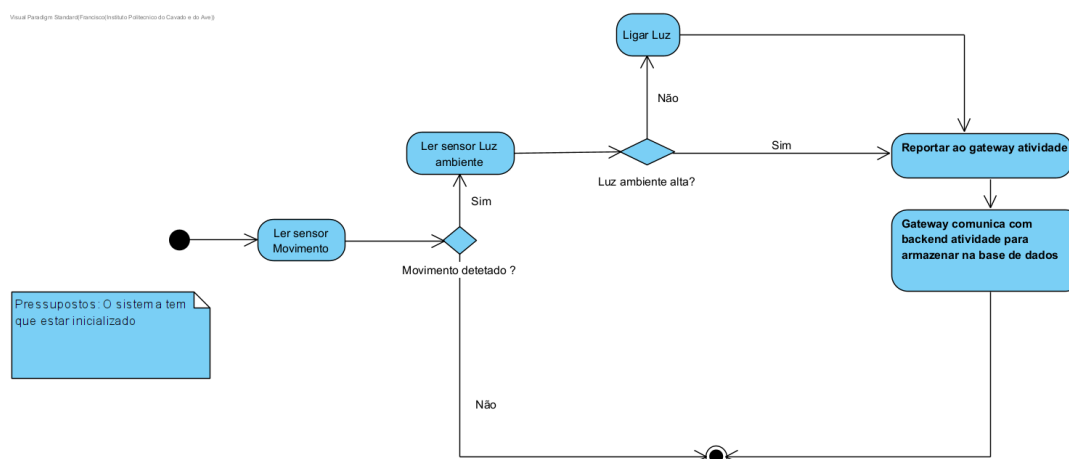


Figura 4 - Diagrama de atividade de Detecção de Movimento

### 3.4. Diagrama de Estados

Os diagramas de estados são um tipo de diagrama que mostra o comportamento do nosso sistema ao longo do tempo. Eles são usados para modelar a dinâmica do sistema e mostrar como ele muda de um estado para outro ao longo do tempo.

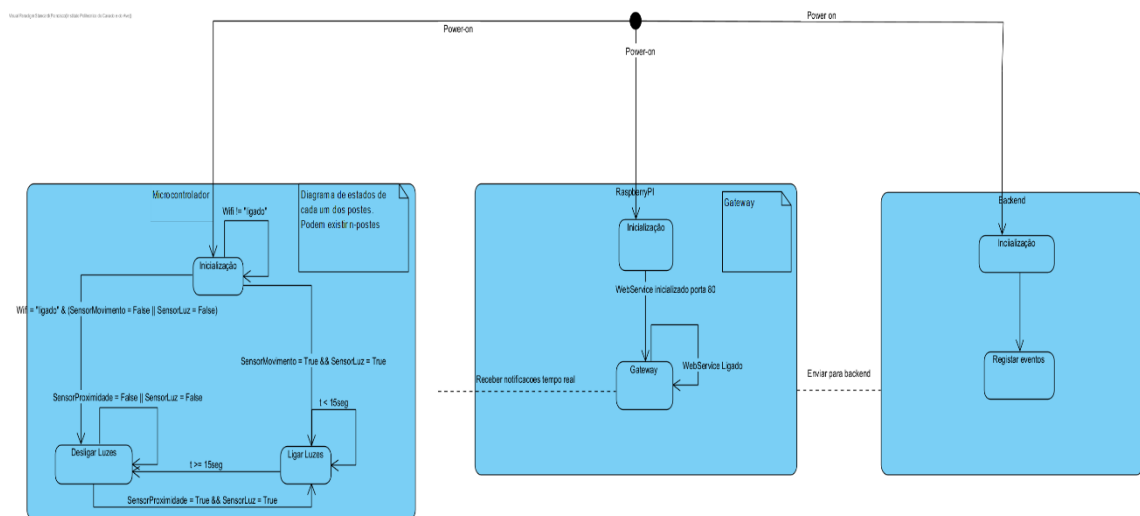


Figura 5 - Diagrama de estados

Este diagrama foi concebido para demonstrar os possíveis estados do poste. Dessa forma, faz sentido começar a explicar os estados do poste. O poste tem três estados: os:

- Inicialização : Objetivo de ligar ao gateway via wifi e inicializar os sensores.
- Desligar Luzes: Serve para desligar a luz do poste.
- Ligar Luzes: Serve para ligar a luz do poste.

Dentro destes três estados, as transições são feitas através dos dados dos sensores maioritariamente. Por exemplo, no caso de o sistema estar no estado "Desligar Luzes", apenas pode transitar para "Ligar Luzes" se detetar movimento e a intensidade de Luz ambiente for baixa.

Acerca do Gateway e backend, estes apenas dependem maioritariamente de notificações enviadas pelo poste, sendo que, não existem mais estados relevantes a identificar, além de, armazenamento da informação.

### 3.5. Diagramas de Sequência

Os diagramas de sequência são um tipo de diagrama que mostra a interação entre diferentes entidades no nosso sistema. Ele é usado para modelar a sequência de mensagens trocadas entre as entidades no nosso sistema, bem como a ordem em que essas mensagens são trocadas.

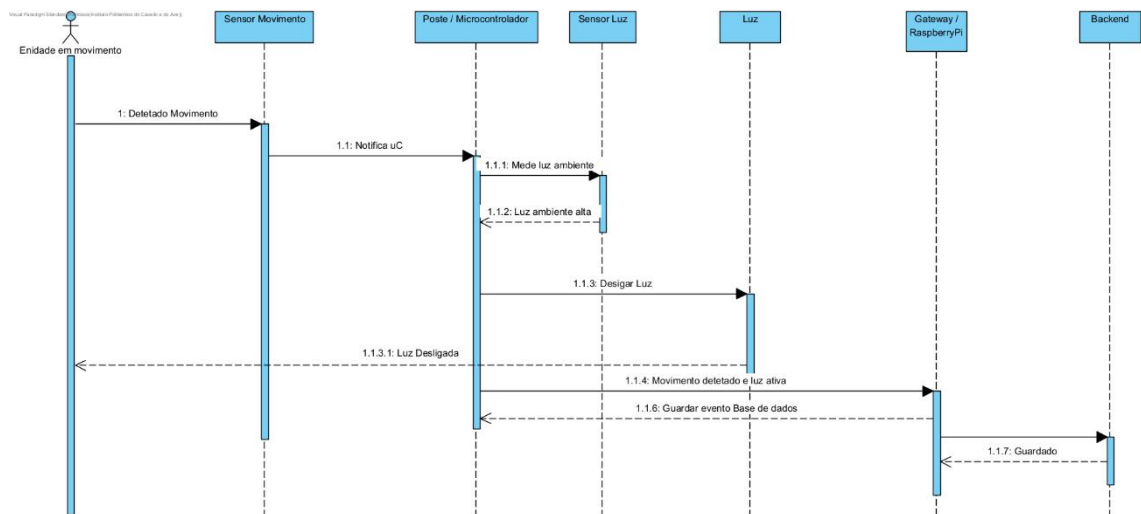


Figura 6 - Diagrama de sequência do Movimento Não Ativa a Luz

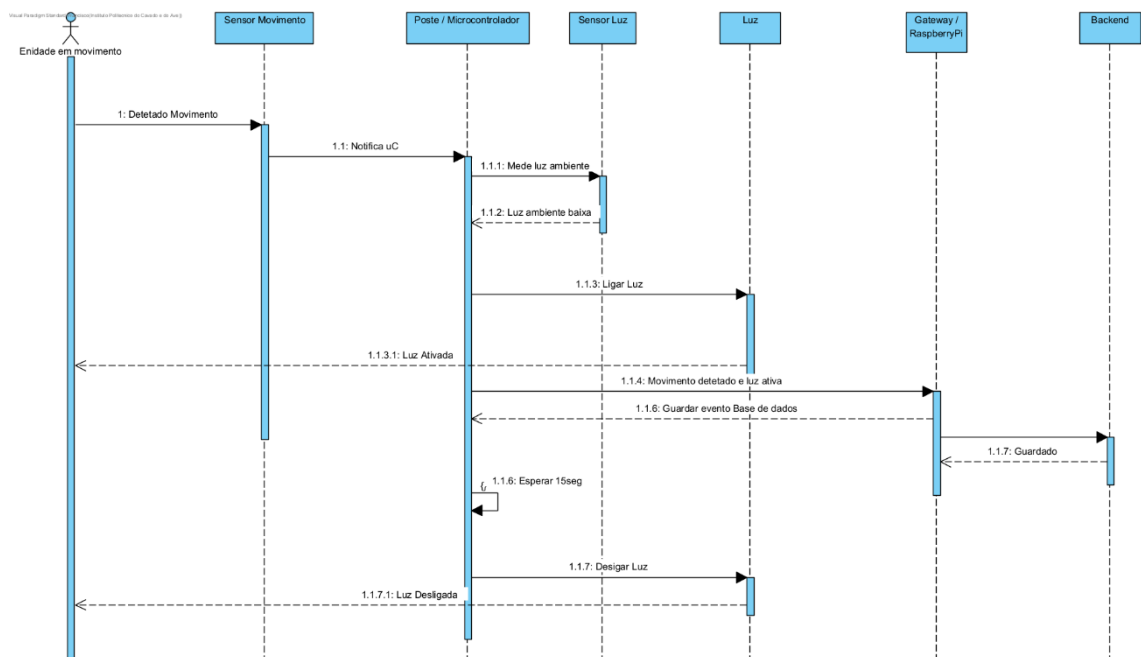


Figura 7 - Diagrama de sequência do Movimento Ativa a Luz

## 4. Prototipagem

Tendo em vista o plano final do projeto, já foram efetuados alguns protótipos daquilo que o nosso projeto visa atingir. Para o efeito foi feito um esboço em papel, seguido de uma transição para um ambiente mais gráfico e, por fim, efetuou-se uma maquete com o Requisito funcional **#RF02 – Iluminação mediante movimento**.

É de frisar que tanto a maquete como o respetivo código não é final e poderá sofrer várias alterações até atingir o seu estado final.

### 4.1. Tecnologias utilizadas

Para o constante progresso deste projeto e, de forma a elaborar uma prova de conceito, foram utilizadas várias tecnologias que nos permitiram estabelecer o que foi planeado através dos diversos conceitos previamente discutidos.

- **HW01: Raspberry Pi 3 Model B+, 1.4GHz, 1GB, cartão SD de 16GB** - O servidor foi implementado num Raspberry Pi para fins de protótipo. É um sistema de baixo custo, consumo reduzido e de grande capacidade para efeitos de prototipagem e desenvolvimento.
- **HW02: Ponto de Acesso de redes sem fios** - Neste equipamento foi configurada uma rede wireless com *ssid smartenergy* segura com encriptação para permitir a comunicação dos *shield wifi* dos *arduinos* com o servidor.
- **HW03: Switch Ethernet** - Este dispositivo foi adicionado à maquete do protótipo com a finalidade de facilitar e interligar diversos computadores portáteis a fim de se poder desenvolver, analisar, etc...
- **HW04: OSOYOO WiFi IoT Learning Kit For Arduino** - Este kit *Arduino* é um clone fiel ao original, além de ser mais económico, inclui os dispositivos necessários para implementação do protótipo do projeto numa maquete. Os componentes essenciais para conceber o protótipo são o *Arduino Uno R3*, o *shield ESP8266 WIFI*, o sensor de movimento por infravermelhos, o sensor

LDR foto resistor, resistências e leds. Para efeitos do projeto foram adquiridos 3 conjuntos para simular 3 postes de iluminação público e poder-se obter dados em situações diferentes e mais realísticas.

- **SW01: Debian OS** - O sistema operativo escolhido para a instalação e desenvolvimento no Raspberry Pi.
- **SW02: Web Services** - Foram incluídos alguns pacotes para disponibilização de serviços web, sendo eles o Apache2 e o PHP 8.1.
- **SW03: Base de dados** - Para a realização de toda a estrutura de dados, foi escolhida uma base de dados simples mas eficiente, sendo ela a MariaDB 10.5.15.

Desenvolveu-se na linguagem de programação PHP um serviço web para que o poste de iluminação possa comunicar com o servidor e, por sua vez, os dados transmitidos possam ser guardados na base de dados. Existe um ficheiro no servidor que é invocado pelos Arduinos com a passagem de determinados parâmetros para suportar esta comunicação. O ficheiro em questão denomina-se por *webservices.php*, que pode ser encontrado no diretório: */var/www/html*.

Para o desenvolvimento do código nos Arduinos, foi escolhida a plataforma open source do fabricante, o Arduino IDE v2.03. A linguagem de programação do Arduino é o C++. Muitas das vezes, estes blocos de código são muito semelhantes à linguagem de programação C, no entanto, é um ambiente relativamente fácil de trabalhar e desenvolver.

Em relação aos componentes eletrónicos utilizados, podem ser consultados na tabela seguinte:

| Nome     | Quantidade | Componente                  |
|----------|------------|-----------------------------|
| U1       | 1          | Arduino Uno R3              |
| W1       | 1          | ESP8266 WIFI                |
| PIR1     | 1          | Sensor PIR                  |
| D1<br>D2 | 2          | LED Branco                  |
| D3       | 1          | LED Vermelho                |
| D4       | 1          | LED Verde                   |
| R1       | 1          | Resistência de 3 k $\Omega$ |
| R6       | 1          | Foto resistor (LDR)         |

Na próxima figura, efetuado na plataforma *TinkerCad*, podemos verificar uma demonstração visual de todos os componentes eletrónicos para efeitos de prototipagem.

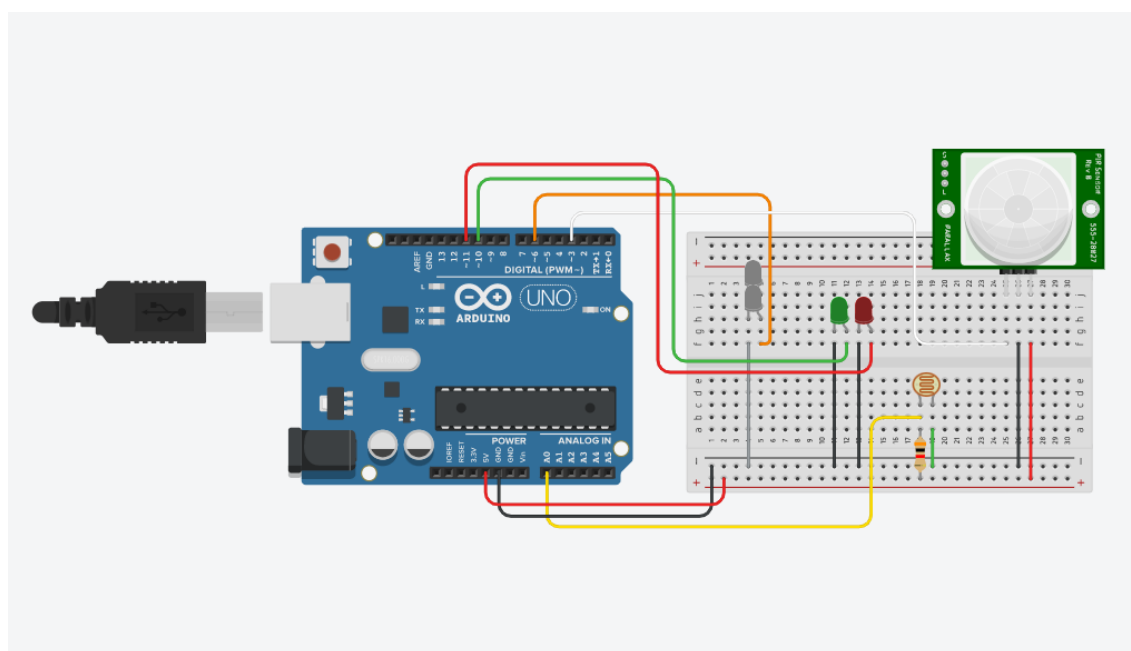


Figura 8 - Circuito eletrónico no TinkerCad

The diagram shows an Arduino Uno (U1) connected to a PIR sensor (PIR1) and four LEDs (D1-D4). The PIR sensor's VCC pin is connected to the 5V supply (U1\_5V) and its GND pin is connected to the common ground (U1\_GND). The PIR sensor's OUT pin is connected to digital pin D9. The four LEDs are connected to digital pins D2, D1, D4, and D3. Each LED has a series resistor (R1, R6, and two unlabeled resistors) connected to the 5V supply. The common ground (U1\_GND) is connected to the GND pin of the Arduino Uno and the GND pins of all LEDs.

12

## 4.2. Mockups

Como indicado anteriormente, o primeiro passo para a concessão das maquetes foi o desenho de um esboço em papel, criado na primeira reunião oficial do nosso grupo. Aqui podem ser vistos os vários componentes que fazem parte do nosso sistema, desde o caso de uso (uma estrada e respetiva iluminação) até aos componentes que manipulam o sistema (Arduinos, Raspberry Pi, etc...)

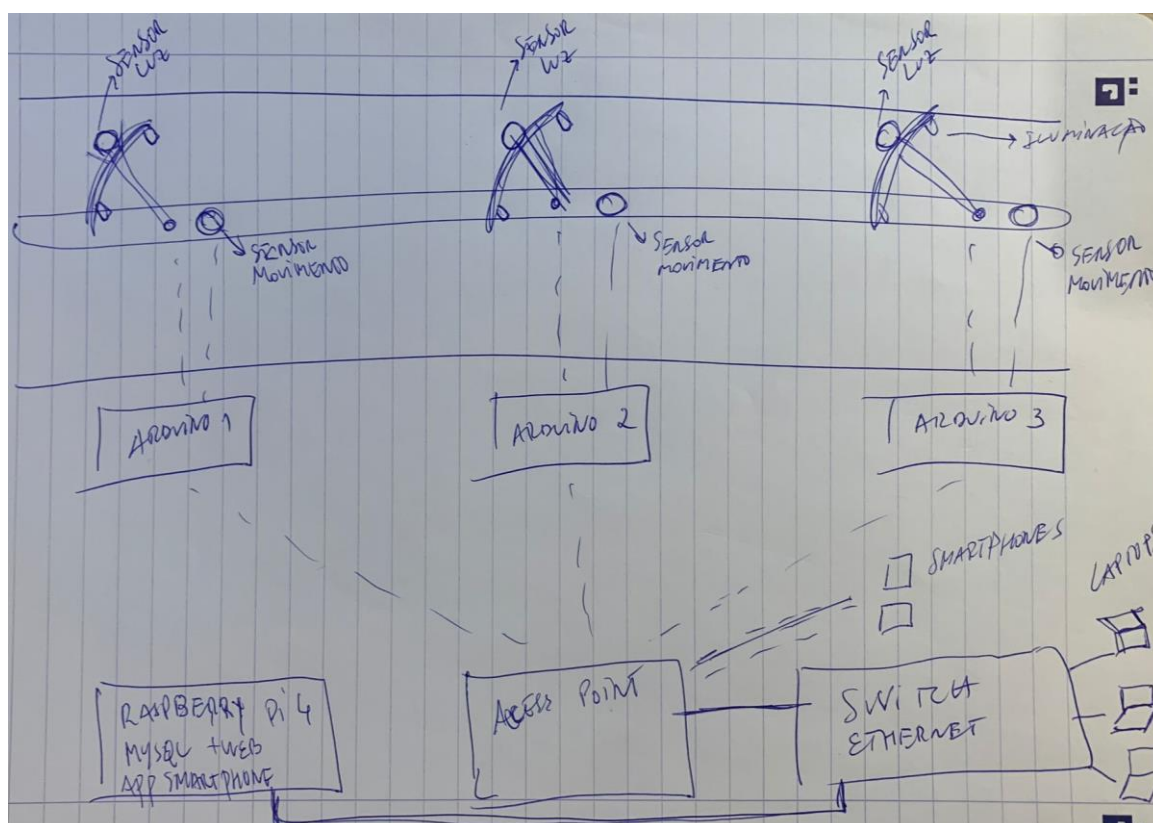
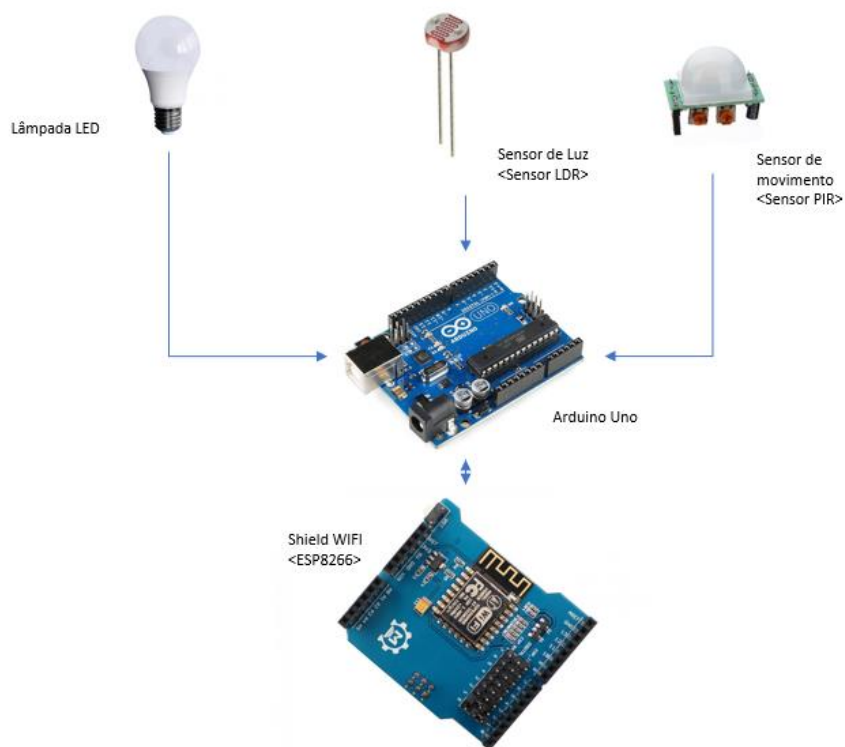


Figura 10 - Esboço do Sistema em papel



Após o esboço em papel ser alvo de análise e aprovação de todos os elementos do grupo, foi passado para um sistema mais organizado e conciso. Este é um diagrama de Hardware que representa o que cada poste necessita de ter de forma a fazer parte do sistema final, nomeadamente:

- 1 Lâmpada LED;
- 1 Sensor de Luz (LDR);
- 1 Sensor de Movimento (PIR);
- 1 Arduíno;
- 1 Shield WiFi (ESP8266).



*Figura 11 - Diagrama de Hardware de UM poste de iluminação*

No próximo esquema conseguimos analisar, de forma completa, todos os componentes que de uma forma ou de outra, irão fazer parte do nosso sistema final.

Aqui estão representados três postes de iluminação, como forma de representação de uma estrada. Na totalidade, a nossa maquete física será constituída por:

- 3 Postes de Iluminação com os componentes descritos acima;
- 1 Access Point;
- 1 Switch Ethernet;
- 1 Raspberry Pi.

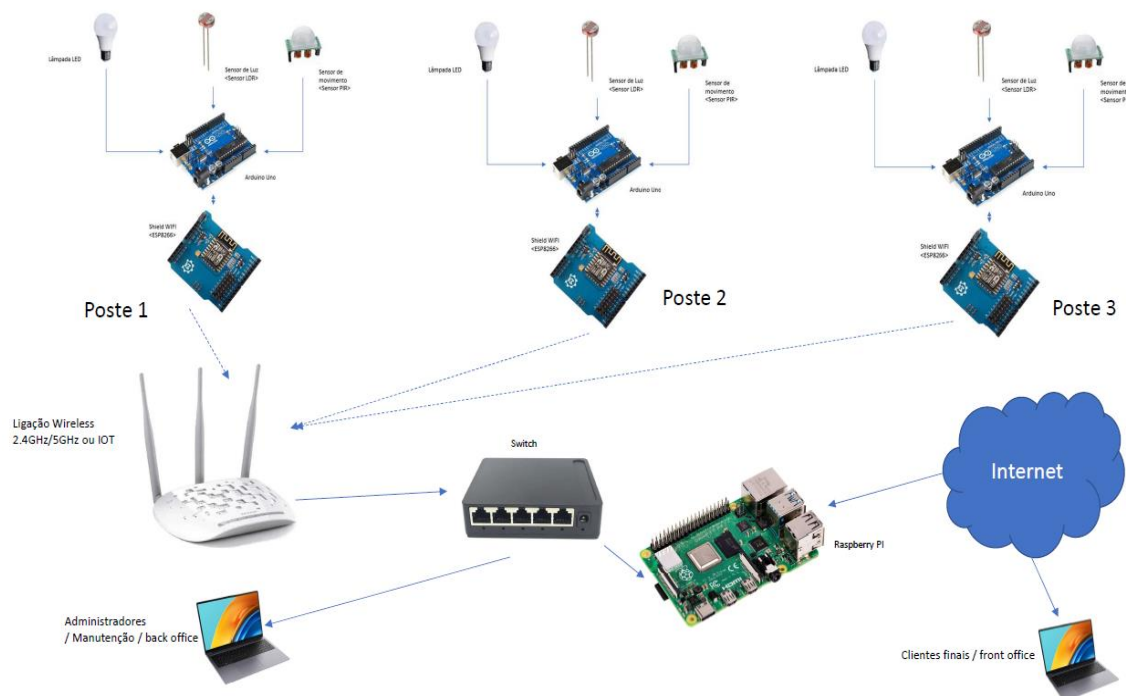


Figura 12 - Diagrama de Hardware de TODO o Sistema

Nas duas próximas figuras conseguimos observar a maquete que já foi prontamente produzida conforme as instruções dos esquemas anteriores.



*Figura 13 - Protótipo da Maquete do Sistema (visão de cima)*



*Figura 14 - Protótipo da Maquete do Sistema (visão lateral)*

As próximas figuras são maquetes que foram efetuadas fazendo uso da plataforma FIGMA, sendo que estas maquetes irão fazer parte dos recursos a utilizar para a criação da nossa aplicação de gestão do sistema.

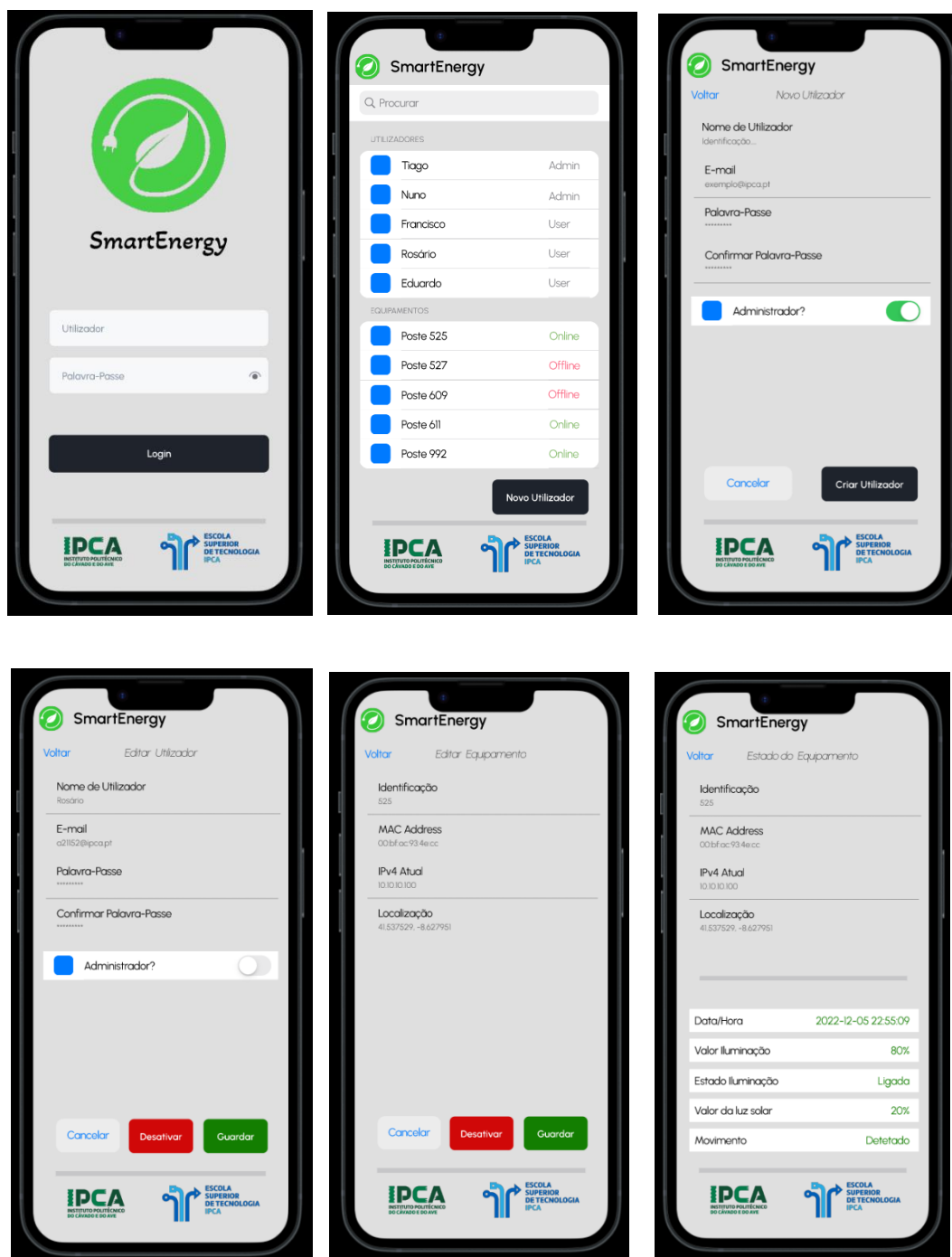


Figura 15 - Mockups da App Móvel

## 5. Código implementado

### 5.1. Código de SETR

Em relação ao código desenvolvido, estará presente na sua totalidade em anexo.

De seguida, está uma explicação mais detalhada sobre as funcionalidades implementadas com o respetivo trecho de código.

#### A. Interrupt ativado por um sensor:

```
#define interruptPin 3 // pino de input do sensor de movimento para a
funcao de interrupt via detecao de movimento
// inicializacao do Interrupt atraves da detecao de movimento com o sensor PIR
attachInterrupt(digitalPinToInterrupt(interruptPin), detectionPIR, CHANGE);
void detectionPIR() {
  if (statePIR == LOW) {
    Serial.print("\nInterrupt via detecao de movimento ATIVADO!");
    statePIR = HIGH;          // estado de detecao de movimento passa a TRUE
    sendData = HIGH;          // estado de envio de dados para o servidor passa a
    TRUE
  }
}
```

Este código tem como finalidade a execução de um Interrupt, ou seja, o objetivo será executar automaticamente as instruções para enviar os dados em tempo real para o servidor, quando é detetado movimento através do sensor PIR.

#### B. Interrupt através de temporizador:

```
int counter = 0; // armazena a quantidade de segundos passados para o
interrupt atraves do timer
```

Primeiramente, é definido um contador para armazenar a quantidade de segundos que o Interrupt timer atuou.

```
// inicializacao do Interrupt atraves de um Timer
Timer1.initialize(500000);
Timer1.setPeriod(1000000);           // definido para periodos de 1
segundo
Timer1.attachInterrupt(periodic);     // funcao que invoca quando e'
atingido o periodo
```

De seguida, é efetuada a inicialização do Interrupt através de um temporizador e este é definido para invocar a função *periodic* a cada segundo.

```
void periodic() {
  if (counter >= periodo) {
    Serial.print("\nInterrupt via timer1 a cada ");
    Serial.print(periodo);
    Serial.print(" segundos ATIVADO!");
    sendData= HIGH;           // estado de envio de dados para o
servidor passa a TRUE
    counter = 0;
  } else {
    ++counter;
  }
}
```

Este trecho de código é referente às instruções da função *periodic*, nomeadamente incrementa o contador *counter* até atingir os 120 segundos, que são definidos na variável *periodo*. Atingindo esta condição, os dados atuais são enviados para o servidor.

### C. Implementação do Serviço Web HTTP:

```
/* inicio: serviço http do proprio poste de iluminacao*/

// fica a espera de ligacoes de clientes
WiFiEspClient client = server.available();
if (client) {
  Serial.println("Nova ligacao http");
  // um pedido http termina com uma linha em branco
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      // se chegou ao fim da linha (recebeu um caracter de nova linha) e a
linha está em branco,
      // o pedido http terminou, entao ja se pode enviar uma resposta
```

```
if (c == '\n' && currentLineIsBlank) {
    Serial.println("Enviado resposta http");

    // envia um cabeçalho de resposta http padrão
    // use \r\n em vez de muitas instruções println para acelerar o
    envio de dados
    client.print(
        "HTTP/1.1 200 OK\r\n"
        "Content-Type: text/html\r\n"
        "Connection: close\r\n" // a ligação será fechada após a
    conclusão da resposta
        "Refresh: 5\r\n" // recarrega a página
    automaticamente a cada 5 segundos
        "\r\n");
    client.print("<!DOCTYPE HTML>\r\n");
    client.print("<html>\r\n");
    client.print("<h4>Smart Energy Campus</h4>\r\n");
    client.print("<h1>Lamp Post</h1>\r\n");
    client.print("<h2>Network</h2>\r\n");
    client.print("Mac Address: ");
    client.print(mac[5],HEX);
    client.print(":");
    client.print(mac[4],HEX);
    client.print(":");
    client.print(mac[3],HEX);
    client.print(":");
    client.print(mac[2],HEX);
    client.print(":");
    client.print(mac[1],HEX);
    client.print(":");
    client.print(mac[0],HEX);
    client.print("<br>\r\n");
    client.print("IP Address: ");
    client.print(ip);
    client.print("<br>\r\n");
    client.print("<h2>Status</h2>\r\n");
    client.print("Light value: ");
    client.print(valLED);
    client.print("<br>\r\n");
    client.print("Light state: ");
    client.print(stateLED);
    client.print("<br>\r\n");
    client.print("LDR value: ");
    client.print(valLDR);
    client.print("<br>\r\n");
    client.print("LDR %: ");
    client.print(valLDRnew);
}
```



```
        client.print("<br>\r\n");
        client.print("PIR value: ");
        client.print(valPIR);
        client.print("<br>\r\n");
        client.print("PIR state: ");
        client.print(statePIR);
        client.print("<br>\r\n");
        client.print("Timer: ");
        client.print(timer);
        client.print("<br>\r\n");
        client.print("<br>\r\n");
        client.print("Requests received: ");
        client.print(++reqCount);
        client.print("<br>\r\n");
        client.print("</html>\r\n");
        break;
    }
    if (c == '\n') {
        // Iniciando uma nova linha
        currentLineIsBlank = true;
    } else if (c != '\r') {
        // Obteve um caracter na linha atual
        currentLineIsBlank = false;
    }
}
}
// Dando tempo ao navegador da web para receber os dados
delay(10);

// Terminar a ligacao
client.stop();
Serial.println("Cliente desconetado");
}
/* fim: serviço http do proprio poste de iluminacao*/
```

Foi implementado um serviço HTTP em cada poste de iluminação, permitindo que externamente com um equipamento com um navegador acede-se através do endereço IP ao posto e visualiza-se o estado atual do sistema. Durante o desenvolvimento e implementação de novas funcionalidades este código deixou de funcionar por motivos de incompatibilidades do sistema, está previsto numa futura versão a correção e reimplementação deste serviço.



#### D. Inteligência do poste de iluminação

```
void outputs() {
    // LDRmax - pouca iluminacao, sem sol, escuro
    // LDRmin - muita iluminacao, muito sol
    if (valLDR <= LDRmin ) valLDR=LDRmin;
    if (valLDR >= LDRmax) valLDR=LDRmax;
    valLDRnew = (long) (valLDR * 100 / LDRmax );    // converter para
    percentagem 0% a 100%
    int valLEDnew = (int) (255 * valLDRnew / 100); // atribui ao LED o
    valor de iluminacao ideal de acordo com o sensor de input LDR

    if (valLDR >= LDRmed) {

        if (statePIR==HIGH) {    // caso volte a detetar movimento reinicia o
        timer
            timer = TIMEmax;    // o tempo de LEDs ligados volta ao maximo
            stateLED = HIGH;    // liga os LEDs
            sendDataToServer();
            statePIR = LOW;    // estado detecao de movimento passa a FALSO
        }

        if (timer > TIMEmax) timer = TIMEmax;
        if (timer > 0) {
            timer = timer - (millis() - timer2); // atualiza o tempo restante
            guardado na variavel timer
            // Ajusta o valor da iluminacao conforme a intensidade de luz
            "solar", o novo valor que esta guardado em valLEDnew
            if (valLED < valLEDnew) valLED=valLED + valINCREMENT;
            if (valLED > valLEDnew) valLED=valLED - valINCREMENT;
        } else {                // reduz o valor da iluminacao dos LEDs
        ao valor mínimo, iluminacao de presenca
            timer = 0;
            stateLED = HIGH;
            if (valLED > valLEDmin) {    // reduz a iluminacao até ser igual ao
            valor de iluminacao de presenca valLEDmin
                valLED = valLED - valINCREMENT;
            } else {
                valLED = valLEDmin;
            }
        }
    }

    analogWrite(LED, valLED);    // atribui a iluminacao atual aos LEDs
} else {                        // desliga os LEDs
    stateLED = LOW;
    valLED = 0;                // atribui a iluminacao a zero...
    analogWrite(LED, valLED);    // ...e desliga os LEDs
}
```

```
timer = 0;
//sendDataToServer();
}

// envia para a consola os dados atuais de input e output
Serial.print("\nLight actual value: "); Serial.print(valLED);
Serial.print("| Light next value: "); Serial.print(valLEDnew);
Serial.print("| Light state: "); Serial.print(stateLED);
Serial.print("| LDR value: "); Serial.print(valLDR);
Serial.print("| LDR %: "); Serial.print(valLDRnew);
Serial.print("| PIR value: "); Serial.print(valPIR);
Serial.print("| PIR state: "); Serial.print(statePIR);
Serial.print("| Timer: "); Serial.print(timer);
Serial.print("| Counter: "); Serial.print(counter);
delay(75);

timer2 = millis(); // regista o tempo atual
if (statePIR == HIGH && stateLED == LOW) {
    statePIR = LOW;
}
}
```

Este código faz as verificações dos dados de entrada obtidos pelos sensores e, de acordo com a quantidade de luz natural existente e a presença de movimento, faz atuar a iluminação artificial por um curto período, ou seja, apenas o essencial para a iluminar a zona onde se enquadra o poste.

#### E. Envio de dados para o servidor

```
void sendDataToServer() { // funcao que faz o envio dos dados atuais para
o servidor
    Serial.println("\nEnviado dados para o servidor");
    client.stop(); // termina todas as ligacoes e efetua um novo pedido e
liberta o socket do shield WiFi

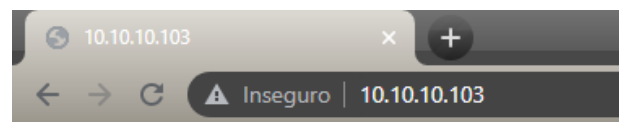
    // verifica se existe conetividade com o servidor
    if (client.connect("10.10.10.2", 80)) {
        // a ligacao com o servidor foi efetuada
        String s1 = "GET /webservices.php?macaddress=";
        s1 += String(mac[5],HEX); s1 += ":";
        s1 += String(mac[4],HEX); s1 += ":";
        s1 += String(mac[3],HEX); s1 += ":";
        s1 += String(mac[2],HEX); s1 += ":";
        s1 += String(mac[1],HEX); s1 += ":";
    }
```

```
s1 += String(mac[0],HEX);
s1 += "&ipaddress=";
s1 += String(ip[0])+String(".")+String(ip[1])
+String(".")+String(ip[2])+String(".")+String(ip[3]); // endereço IP
atual
String s2 = "&valled="; s2 += valLED;
s2 += "&stateled=";      s2 += stateLED;
s2 += "&valldr=";        s2 += valLDR;
s2 += "&valldrnew=";      s2 += valLDRnew;
s2 += "&valpir=";         s2 += valPIR;
s2 += "&statepir=";       s2 += statePIR;
s2 += " HTTP/1.1";
s1 += s2;
Serial.println((s1));
client.println((s1));
client.println(F("Host: 10.10.10.2"));
client.println("Connection: close");
client.println();
}
else {
    // se a ligacao com o servidor nao for efetuada
    Serial.println(F("A ligacao falhou!"));
}
client.stop();
}
```

Nesta função é invocado um webservice do nosso servidor em que passam todos os valores/dados atuais. Estes dados são armazenados numa base de dados (MariaDB) para mais tarde serem tratados. Além disso, na unidade curricular de Programação de Dispositivos Móveis, elaborou-se uma aplicação android para analisar e tratar estes dados de acordo com as maquetes planeadas anteriormente.

#### F. Acesso HTTP aos postes de iluminação

As próximas três figuras representam os dados em tempo real a serem consultados no Serviço Web de cada poste, através do respetivo endereço de IP. Podem ser verificados dados importantes, tais como o nível de iluminação ou até mesmo a percentagem de luz solar que o sistema está a detetar.



Smart Energy Campus

## Lamp Post

### Network

Mac Address: C4:5B:BE:F3:C4:8F  
IP Address: 10.10.10.103

### Status

Light value: 2  
Light state: 0  
LDR value: 461  
LDR %: 46  
PIR value: 0  
PIR state: 0  
Timer: 0

Requests received: 1

*Figura 16 - Serviço Web de um Poste de Iluminação #1*

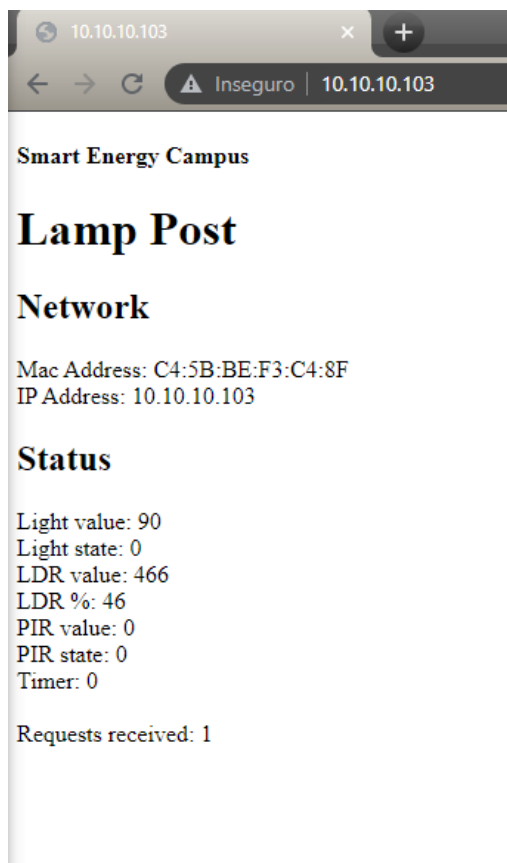


Figura 17 - Serviço Web de um Poste de Iluminação #2

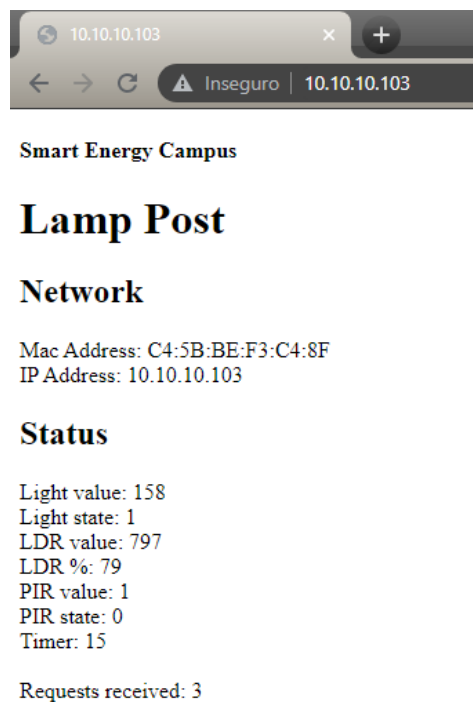


Figura 18 - Serviço Web de um Poste de Iluminação #3

## 5.2. Integração dos postes na aplicação móvel e servidor

### A. Config.php

```
<?php
flush();
// Configurações gerais
$pasta="/var/www/html/"; // localizacao do site;

// Configuração do servidor mysql
$bd_server = "localhost"; // Nome ou IP do servidor
$bd = "smartenergy"; // Nome da Base de Dados

// mysql read & write perms
$bd_user = "se"; // Username
$bd_passwd = "smartenergy"; // Palavra Passe

// tempos
$onlineTime = "00:02:15"; // tempo para considerar um poste de
iluminacao online (hh:mm:ss)
?>
```

### B. Engine.php

```
<?php
date_default_timezone_set("Europe/Lisbon");
include 'config.php';
$output = array();
$input = array();
$response = array();

// ligacao a base de dados
$mysqli = new mysqli($bd_server, $bd_user, $bd_passwd, $bd);
// verifica a conetividade com a base de dados
if ($mysqli === false) {
    die("ERROR: Could not connect. ".$mysqli->connect_error);
} else {
    //echo "<br>sql connection sucessfull<br>";
}

//-----
if($_GET['method']=="select"){
    // method: select
}
//-----
```

```

if($_GET['object']=="users"){
    // object: users
    $sql = "select * from users";
    if($_GET['idusers']!="") {
        $sql = $sql . ' where idusers="'.$_GET["idusers"].'";
    }
    $result = $mysqli->query($sql);
    if($result->num_rows > 0){
        echo '{"status":"ok","totalResults":'.$result->num_rows.', "'.$_GET['object'].'":[';
        for ($i=0;$i<mysqli_num_rows($result);$i++) {
            echo ($i>0?',':').json_encode(mysqli_fetch_object($result));
        }
        echo ']]';
    }else{
        $output['response'] = "false";
        $output['userid'] = "no_record_found";
        $output['name'] = "no_record_found";
        $output['username'] = "no_record_found";
        $output['password'] = "no_record_found";
        $output['permission'] = "no_record_found";
        echo json_encode($output);
    }
}

//-----
if($_GET['object']=="devices"){
    // object: devices
    $sql = "select * from devices";
    if($_GET['iddevices']!="") {
        $sql = $sql . ' where iddevices="'.$_GET["iddevices"].'";
    }
    $result = $mysqli->query($sql);
    if($result->num_rows > 0){
        echo '{"status":"ok","totalResults":'.$result->num_rows.', "'.$_GET['object'].'":[';
        for ($i=0;$i<mysqli_num_rows($result);$i++) {
            $res = ($i>0?',':').json_encode(mysqli_fetch_object($result));
            echo substr($res, 0,-1);
            echo ",";

            // status: online / offline
            $value=strstr($res,':');
            $value=strstr($value,'datetime',true);
            $value=ltrim($value, ' ');
            $value=substr($value, 0,-3);
            $sql2 = 'select
if(timediff(now(),logs.datetime)<"'.$onlineTime.'" ,"online","offline") as

```

```
status FROM smartenergy.devices left join logs on
devices.iddevices=devices_iddevices where devices_iddevices =
"'.$value.'" order by logs.datetime desc limit 1';
    //echo $sql2;
    $result2 = $mysqli->query($sql2);
    if (mysqli_num_rows($result2)==0) echo '"status":"offline"';
    for ($j=0;$j<mysqli_num_rows($result2);$j++) {
        $res2 =
($j>0?',':').json_encode(mysqli_fetch_object($result2));
        echo ltrim($res2, '{');
    }
    }
    echo ']]';
}else{
    $output['response'] = "false";
    $output['iddevices'] = "no_record_found";
    $output['macaddress'] = "no_record_found";
    $output['coordinatex'] = "no_record_found";
    $output['coordinatey'] = "no_record_found";
    echo json_encode($output);
}
}
//-----
if($_GET['object']=="logs"){
    // object: logs
    $sql = "select * from logs";
    if($_GET['devices_iddevices']!="") {
        $sql = $sql . ' where
devices_iddevices="'.$_GET["devices_iddevices"].'";
    }
    $sql = $sql. ' order by datetime desc limit 100';
    $result = $mysqli->query($sql);
    if($result->num_rows > 0){
        echo '{"status":"ok","totalResults":'.$result->num_rows.', "'.$_GET['object'].'":[';
        for ($i=0;$i<mysqli_num_rows($result);$i++) {
            echo ($i>0?',':').json_encode(mysqli_fetch_object($result));
        }
        echo ']]';
    }else{
        $output['response'] = "false";
        $output['iddevices'] = "no_record_found";
        $output['macaddress'] = "no_record_found";
        $output['coordinatex'] = "no_record_found";
        $output['coordinatey'] = "no_record_found";
        echo json_encode($output);
    }
}
```



```

}
//-----
if($_GET['object']=="devicestatus"){
    // object: devicestatus, get a device and status
    if($_GET['devices_iddevices']!="") {
        $sql = 'select *,
if(timediff(now(),datetime)<''. $onlineTime.'',"online","offline") as
status from smartenergy.logs';
        $sql = $sql . ' where
devices_iddevices="' . $_GET["devices_iddevices"] . '"';
        $sql = $sql . ' order by idlogs desc limit 1';
    } else {
        $sql = "select * from (select * from smartenergy.logs";
        $sql = $sql. ' order by idlogs desc) as tmp_table group by
devices_iddevices;';
    }

    $result = $mysqli->query($sql);
    if($result->num_rows > 0){
        echo '{"status":"ok","totalResults":"' . $result-
>num_rows . '","' . $_GET['object'] . '":[';
        for ($i=0;$i<mysqli_num_rows($result);$i++) {
            echo ($i>0?',': '').json_encode(mysqli_fetch_object($result));
        }
        echo ']]';
    }else{
        $output['response'] = "false";
        $output['iddevices'] = "no_record_found";
        $output['macaddress'] = "no_record_found";
        $output['coordinatex'] = "no_record_found";
        $output['coordinatey'] = "no_record_found";
        echo json_encode($output);
    }
}

//-----
if($_GET['object']=="devicesstatus"){
    // object: devicesstatus, get all devices and
status
    $sql = "select iddevices, macaddress, detail, coordinatex,
coordinatey from devices";
    if($_GET['iddevices']!="") {
        $sql = $sql . ' where iddevices="' . $_GET["iddevices"] . '"';
    }
    $result = $mysqli->query($sql);
    if($result->num_rows > 0){

```

```

    echo '{"status":"ok","totalResults":"' . $result-
>num_rows.'", "'. $GET['object']. '":[';
    for ($i=0;$i<mysqli_num_rows($result);$i++) {
        $res = ($i>0?', ':'').json_encode(mysqli_fetch_object($result));
        echo substr($res, 0,-1);
        echo ",";

        // status: online / offline
        $value=strstr($res,':');
        $value=strstr($value,'macaddress',true);
        $value=ltrim($value, ':');
        $value=substr($value, 0,-3);
        //$sql2 = 'select
if(timediff(now(),logs.datetime)<"' . $onlineTime.'" ,"online","offline") as
status FROM smartenergy.devices left join logs on
devices.iddevices=devices_iddevices where devices_iddevices =
"' . $value.'" order by logs.datetime desc limit 1';
        $sql2 = 'select *,
if(timediff(now(),datetime)<"' . $onlineTime.'" ,"online","offline") as
status from smartenergy.logs';
        $sql2 = $sql2 . ' where devices_iddevices="' . $value.'"';
        $sql2 = $sql2 . ' order by idlogs desc limit 1';
        //echo $sql2;
        $result2 = $mysqli->query($sql2);
        if (mysqli_num_rows($result2)==0) echo '"status":"offline"';
        for ($j=0;$j<mysqli_num_rows($result2);$j++) {
            $res2 =
($j>0?', ':'').json_encode(mysqli_fetch_object($result2));
            echo ltrim($res2, '{');
        }
    }
    echo ']]';
} else {
    $output['response'] = "false";
    $output['iddevices'] = "no_record_found";
    $output['macaddress'] = "no_record_found";
    $output['coordinatex'] = "no_record_found";
    $output['coordinatey'] = "no_record_found";
    echo json_encode($output);
}
}
}
?>

```

Neste código, foi desenvolvido um webservice para a aplicação movel consultar dados existentes na base de dados

### C. Login.php

```
<?php
date_default_timezone_set("Europe/Lisbon");
include 'config.php';
$output = array();
$input = array();
$response = array();

if($_GET['method']=="login"){
    $username= $_GET["username"];
    $password= $_GET["password"];

    // ligacao a base de dados
    $mysqli = new mysqli($bd_server, $bd_user, $bd_passwd, $bd);

    // verifica a conetividade com a base de dados
    if ($mysqli === false) {
        die("ERROR: Could not connect. ".$mysqli->connect_error);
    } else {
        //echo "<br>sql connection sucessfull<br>";
    }

    // query a base de dados se existe o username com a password
    $sql = "select * from users where username='$username' and
password='$password'";
    $result = $mysqli->query($sql);

    if($result->num_rows > 0){
        // inicializa valores de nao acesso / login invalido
        $output['response'] = "false";
        $output['userid'] = "no_record_found";
        $output['name'] = "no_record_found";
        $output['username'] = "no_record_found";

        // coloca os valores corretos do login, username, name
        $row = $result->fetch_assoc();
        $output["response"] = trim("true");
        $output["userid"] = trim($row["idusers"]);
        //$name = str_replace(" ","_",$row['name']);
        //$output["name"] = trim($name);
        $output["name"] = trim($row['name']);
        $username = str_replace(" ","_",$row['username']);
        $output["username"] = trim($username);
    }else{
        $output['response'] = "false";
        $output['userid'] = "no_record_found";
    }
}
```

```
        $output['name'] = "no_record_found";
        $output['username'] = "no_record_found";
    }
    echo json_encode($output);
}
?>
```

Este código refere-se a um webservice que trata da autenticação para a aplicação móvel. Podemos verificar que o código visa efetuar a ligação à base de dados e fazer uma query para verificar se o utilizador e password existem e coincidem.

#### D. Webservices.php

```
<?php
// ficheiro com as configuracoes todas necessarias
include 'config.php';

$macaddress = "";

// recolha dos parametros/atributos passados ao servico
if ($_GET) {
    //echo "valor passado pela url no endereco";
    $macaddress = $_GET['macaddress'];
    $ipaddress = $_GET['ipaddress'];
    $valled = $_GET['valled'];
    $stateled = $_GET['stateled'];
    $valldr = $_GET['valldr'];
    $valldrnew = $_GET['valldrnew'];
    $valpir = $_GET['valpir'];
    $statepir = $_GET['statepir'];

    if ( $macaddress == "")
    {
        //echo "valor passado como atributo";
        $macaddress = $attrs['macaddress'];
        $ipaddress = $attrs['ipaddress'];
        $valled = $attrs['valled'];
        $stateled = $attrs['stateled'];
        $valldr = $attrs['valldr'];
        $valldrnew = $attrs['valldrnew'];
        $valpir = $attrs['valpir'];
        $statepir = $attrs['statepir'];
    }
}
if ( $macaddress == "")
{
```

```
//echo "valores por defeito";
$macaddress = '';
$ipaddress = '';
$valled = "";
$stateled = "";
$valldr = "";
$valldrnew = "";
$valpir = "";
$statepir = "";
}

echo "<h2>webservices.php</h2>";
echo "<br>macaddress ".$macaddress;
echo "<br>ipaddress ".$ipaddress;
echo "<br>valled ".$valled;
echo "<br>stateled ".$stateled;
echo "<br>valldr ".$valldr;
echo "<br>valldrnew ".$valldrnew;
echo "<br>valpir ".$valpir;
echo "<br>statepir ".$statepir;
echo "<br>";

if ( $macaddress != "") {
    // ligacao a base de dados
    $mysqli = new mysqli($bd_server, $bd_user, $bd_passwd, $bd);

    // verifica a conetividade com a base de dados
    if ($mysqli === false) {
        die("ERROR: Could not connect. ".$mysqli->connect_error);
    } else {
        echo "<br>sql connection sucessfull<br>";
    }

    echo "<br>checking if macaddress ".$macaddress." exists...";

    $sql = "SELECT iddevices FROM devices WHERE macaddress =
'".$macaddress."'";
    echo "<br>mysql: ".$sql;

    $result = $mysqli->query($sql);

    while ($valor = $result->fetch_array(MYSQLI_BOTH)){
        $iddevice = $valor["iddevices"];
    }

    echo "<br>";
}
```

```

echo "<br>populating tables...";

if (mysqli_num_rows($result)>0) { // caso existe
    mysqli_free_result($result);
    echo "<br>iddevice: ".$iddevice;
    echo "<br>";

    $sql = "INSERT INTO logs (ipaddress, valled, stateled,
valldr, valldrnew, valpir, statepir, devices_iddevices)
VALUES('".$ipaddress."', ".$valled.", ".$stateled.", ".$
$valldr.", ".$valldrnew.", ".$valpir.", ".$statepir.", ".$iddevice.")";

    // executa o comando sql gerado anteriormente
    if ($mysqli->query($sql) === true) // sucesso na insercao
    {
        echo "<br>mysql: ".$sql;
        echo "<br>successfully inserted data!";
    }
} else { // caso nao existe
    mysqli_free_result($result);
    echo "<br>macaddress does not exist, let's create a
record...";
    $sql = "INSERT INTO devices (macaddress, coordinatex,
coordinatey)
VALUES('".$macaddress."', 0, 0)";
    // executa o comando sql gerado anteriormente
    if ($mysqli->query($sql) === true) // sucesso na insercao
    {
        echo "<br>mysql: ".$sql;
        echo "<br>successfully inserted macaddress!";
    }
}
} else {
    echo "<br>nothing to do...";
}
?>

```

Este webservice, de uma forma resumida, permite que um poste de iluminação envie todos os dados de input/output para serem armazenados na base de dados.

### 5.3. Código de PDM

Dada a extensão do código desenvolvido para a aplicação móvel nesta unidade curricular, optamos por anexar todo o projeto num documento comprimido. Este poderá ser consultado na mesma pasta deste relatório.

### 5.4. Resultado obtido

Esta secção representa o estado atual do nosso projeto a nível da aplicação móvel. As mockups efetuadas na unidade curricular de Projeto Aplicado foram tomadas em consideração para a realização desta aplicação. De seguida estarão representadas algumas figuras que retratam toda a gestão do nosso sistema.

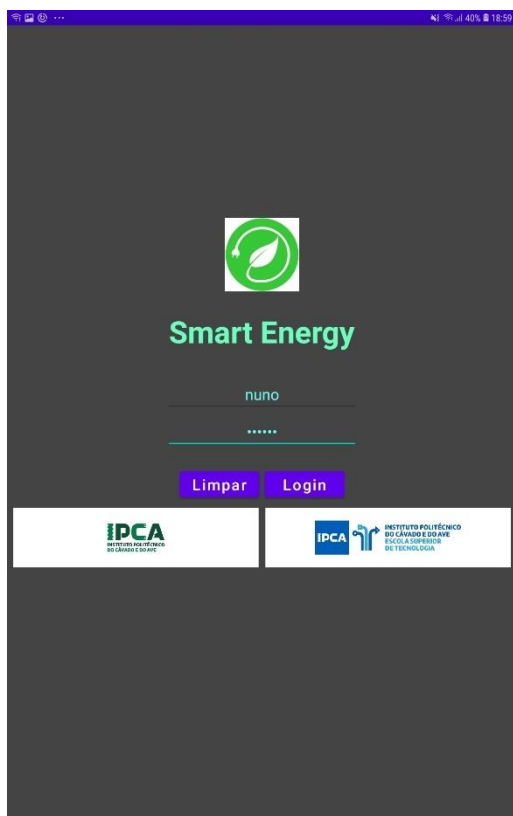


Figura 20 - Printscreen da App: Login



Figura 19 - Printscreen da App: Listagem de dispositivos

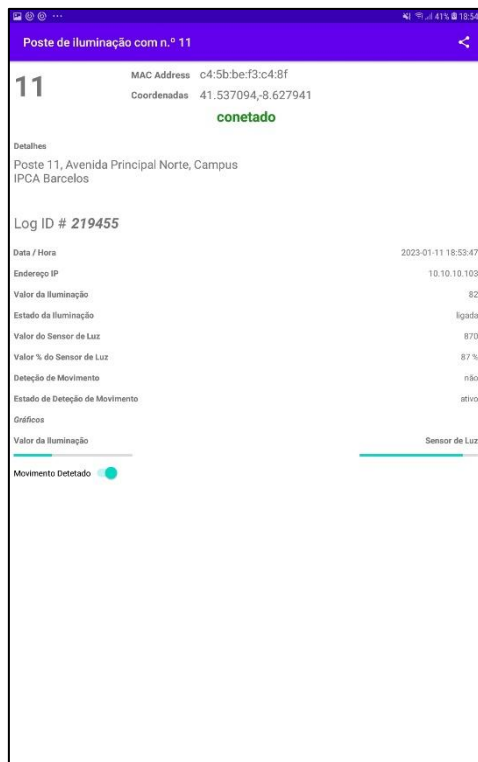


Figura 22 - Printscreen da App: Detalhes de um Poste conectado #1

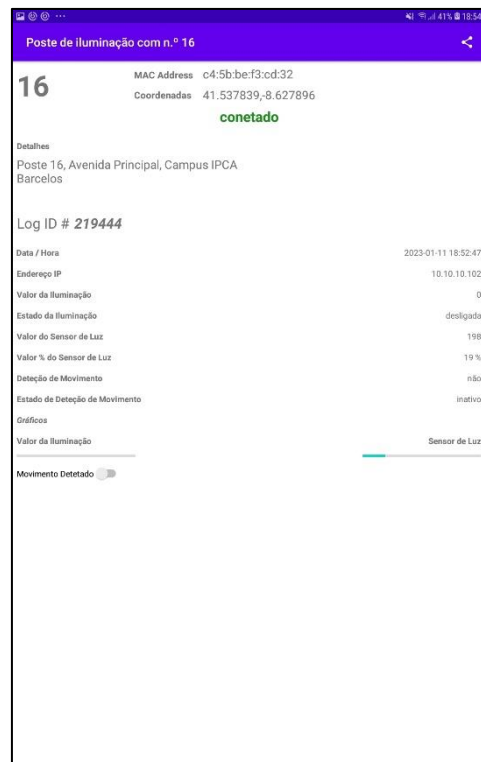


Figura 21 - Printscreen da App: Detalhes de um Poste conectado #2



Figura 24 - Printscreen da App: Detalhes de um Poste conectado #3

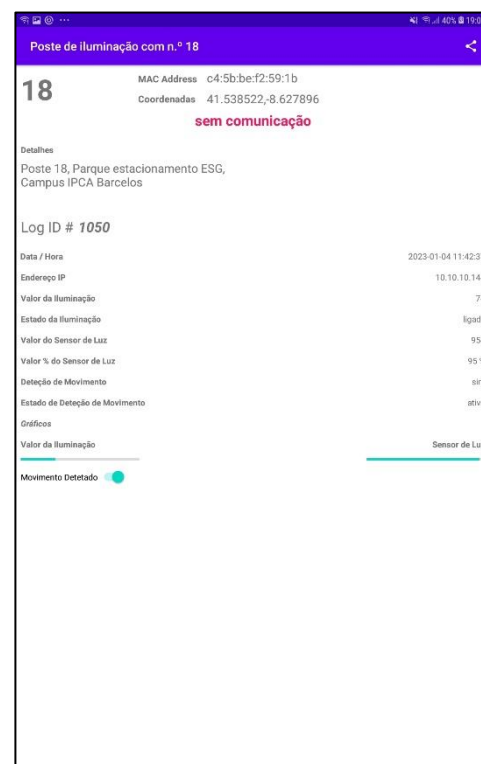


Figura 23 - Printscreen da App: Detalhes de um Poste sem comunicação



## 6. Conclusão

A realização deste trabalho permitiu-nos esmiuçar os conteúdos lecionados na unidade curricular de Programação de Dispositivos Móveis, nomeadamente o uso do *Android Studio* e a linguagem de programação *Kotlin* para a realização de um projeto inovador.

O nosso grupo de trabalho compreende o grande poder e usabilidade de um dispositivo deste género e aprecia o facto de nos serem explicadas todas as dúvidas, sempre que necessário, por parte do docente.

Os resultados obtidos apontam para a redução dos consumos de energia, o que irá levar a uma redução da pegada ambiental causada pelo consumo excessivo de energia.

Uma prova de protótipo conceptual já está totalmente funcional e pronta para ser apresentada, tanto a nível de aplicação móvel como a nível de maquete física.

Com base na nossa investigação, a nossa equipa reconhece o problema emergente das emissões de CO<sub>2</sub> em Portugal e espera que projetos como este possam proporcionar consciência e inspiração para criar ideias mais inovadoras no futuro e que possam ajudar a comunidade de uma forma positiva.

## 7. Bibliografia

Raspberry Pi:

<https://www.raspberrypi.com/>

Sistema Operativo Debian:

<https://wiki.debian.org/RaspberryPi>

Programação em PHP:

<https://www.php.net/>

Base de dados MariaDB:

<https://mariadb.org/>

Clone Arduino OSOYOO:

<https://osoyoo.com/>

Documentação sobre Arduinos e programação:

[https://docs.arduino.cc/software/ide-v2?\\_gl=1\\*1x87tgx\\*\\_ga\\*MjUzNjY5NjAxLjE2NjQzMDI4ODQ.\\*\\_ga\\_NEXN8H46L5\\*MTY3MDQwODU4Ni4xMy4xLjE2NzA0MDg2MDkuMC4wLjA](https://docs.arduino.cc/software/ide-v2?_gl=1*1x87tgx*_ga*MjUzNjY5NjAxLjE2NjQzMDI4ODQ.*_ga_NEXN8H46L5*MTY3MDQwODU4Ni4xMy4xLjE2NzA0MDg2MDkuMC4wLjA)