**SMART ENERGY**



**Sistemas Embebidos e de Tempo Real**

**Licenciatura em Engenharia de Sistemas Informáticos**

**Regime Pós-Laboral**

**2022/2023**

**Alunos**

Nuno Mendes – Nº 2727

Rosário Silva – Nº 21138

Tiago Azevedo - Nº 21153

Francisco Pereira – Nº 21156

**Orientação**

Profº Pedro Cunha

Índice

[1. Introdução 1](#_Toc123133364)

[2. Proposta de sistema 2](#_Toc123133365)

[2.1. Requisitos Funcionais 2](#_Toc123133366)

[2.2. Requisitos Não Funcionais 3](#_Toc123133367)

[3. Código desenvolvido 4](#_Toc123133368)

[3.1.1. Arduíno: 4](#_Toc123133369)

[3.1.1.1. Interrupt associado ao sensor PIR para que sempre ao detetar movimento enviar os dados para o servidor 4](#_Toc123133370)

[3.1.1.2. Interrupt timer de 120 em 120 segundos envia o estado atual do sistema para o servidor 4](#_Toc123133371)

[3.1.1.3. Durante o desenvolvimento deste projeto na versão 0.1 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. 4](#_Toc123133372)

[3.1.1.4. Fazer capturas de ecrã do acesso HTTP aos postes de iluminação, está na versão v0.1. As capturas de ecrã encontram-se em "projeto\servidor\galeria" denominadas por: 4](#_Toc123133373)

[3.1.1.5. serviço web de um posto de iluminação1.png 4](#_Toc123133374)

[3.1.1.6. serviço web de um posto de iluminação2.png 4](#_Toc123133375)

[3.1.1.7. serviço web de um posto de iluminação3.png 4](#_Toc123133376)

[3.1.1.8. código fonte do Arduíno encontra-se em "projeto\arduino\SmartEnergyCampus\SmartEnergyCampus.ino" 5](#_Toc123133377)

[3.1.2. Maquete: 5](#_Toc123133378)

[3.1.2.1. Referir que o sensor pir em futuros trabalhos será colocado num ponto mais estratégico do poste de iluminação de forma que a área de abrangência seja superior e o poste não interfira na deteção de movimento (colocar as fotos que se encontram em "projeto\maquete\sensorPIR") 5](#_Toc123133379)

[4. Justificação das decisões tomadas 6](#_Toc123133380)

[5. Prototipagem 7](#_Toc123133381)

[5.1. Mockups 7](#_Toc123133382)

[5.2. Tecnologias utilizadas 12](#_Toc123133383)

[6. Conclusão 15](#_Toc123133384)

[7. Bibliografia 16](#_Toc123133385)

Índice de ilustrações

[Figura 1- Cronograma do projeto 4](#_Toc121523074)

[Figura 2 - Diagrama de Casos de Uso 8](#_Toc121523075)

[Figura 3 - Diagrama Entidade-Relação 9](#_Toc121523076)

[Figura 4 - Diagrama de atividade de Inicialização 10](#_Toc121523077)

[Figura 5 - Diagrama de atividade de Deteção de Movimento 10](#_Toc121523078)

[Figura 6 - Diagrama de estados 11](#_Toc121523079)

[Figura 7 - Diagrama de sequência do Movimento Não Ativa a Luz 12](#_Toc121523080)

[Figura 8 - Diagrama de sequência do Movimento Ativa a Luz 12](#_Toc121523081)

[Figura 9 - Circuito eletrónico no TinkerCad 15](#_Toc121523082)

[Figura 10 - Esquema do circuito eletrónico 16](#_Toc121523083)

[Figura 11 - Esboço do Sistema em papel 17](file:///C:\Users\tiago\Documents\SmartEnergy\PA\Entrega3\Relatorio_2727_21153_21156_Fase3.docx#_Toc121523084)

[Figura 12 - Diagrama de Hardware de UM poste de iluminação 18](#_Toc121523085)

[Figura 13 - Diagrama de Hardware de TODO o Sistema 19](#_Toc121523086)

[Figura 14 - Protótipo da Maquete do Sistema (visão de cima) 20](#_Toc121523087)

[Figura 15 - Protótipo da Maquete do Sistema (visão lateral) 20](file:///C:\Users\tiago\Documents\SmartEnergy\PA\Entrega3\Relatorio_2727_21153_21156_Fase3.docx#_Toc121523088)

[Figura 16 - Mockups da App Móvel 21](file:///C:\Users\tiago\Documents\SmartEnergy\PA\Entrega3\Relatorio_2727_21153_21156_Fase3.docx#_Toc121523089)

# Introdução

No âmbito da UC de Sistema Embebidos e de Tempo Real pretende-se desenvolver um projeto que consistirá no desenvolvimento de um sistema em tempo real que possa ser utilizado como resposta a uma necessidade específica.

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.

# 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.

## 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 Arduíno; valor do input da quantidade de luz; valor do output de iluminação e Input de deteção de movimento).

## 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.

# Código desenvolvido

## Arduíno:

### Interrupt associado ao sensor PIR para que sempre ao detetar movimento enviar os dados para o servidor

#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

  }

}

### Interrupt timer de 120 em 120 segundos envia o estado atual do sistema para o servidor

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

// 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

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;

  }

}

### Durante o desenvolvimento deste projeto na versão 0.1 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.

/\* 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 ligacao será fechada após a conclusão da resposta

            "Refresh: 5\r\n"         // recarrega a pagina 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\*/

### Fazer capturas de ecrã do acesso HTTP aos postes de iluminação, está na versão v0.1. As capturas de ecrã encontram-se em "projeto\servidor\galeria" denominadas por:

### serviço web de um posto de iluminação1.png

### serviço web de um posto de iluminação2.png

### serviço web de um posto de iluminação3.png

### código fonte do Arduíno encontra-se em "projeto\arduino\SmartEnergyCampus\SmartEnergyCampus.ino"

## Maquete:

### Referir que o sensor pir em futuros trabalhos será colocado num ponto mais estratégico do poste de iluminação de forma que a área de abrangência seja superior e o poste não interfira na deteção de movimento (colocar as fotos que se encontram em "projeto\maquete\sensorPIR")

# Justificação das decisões tomadas

# 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.

## 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…)

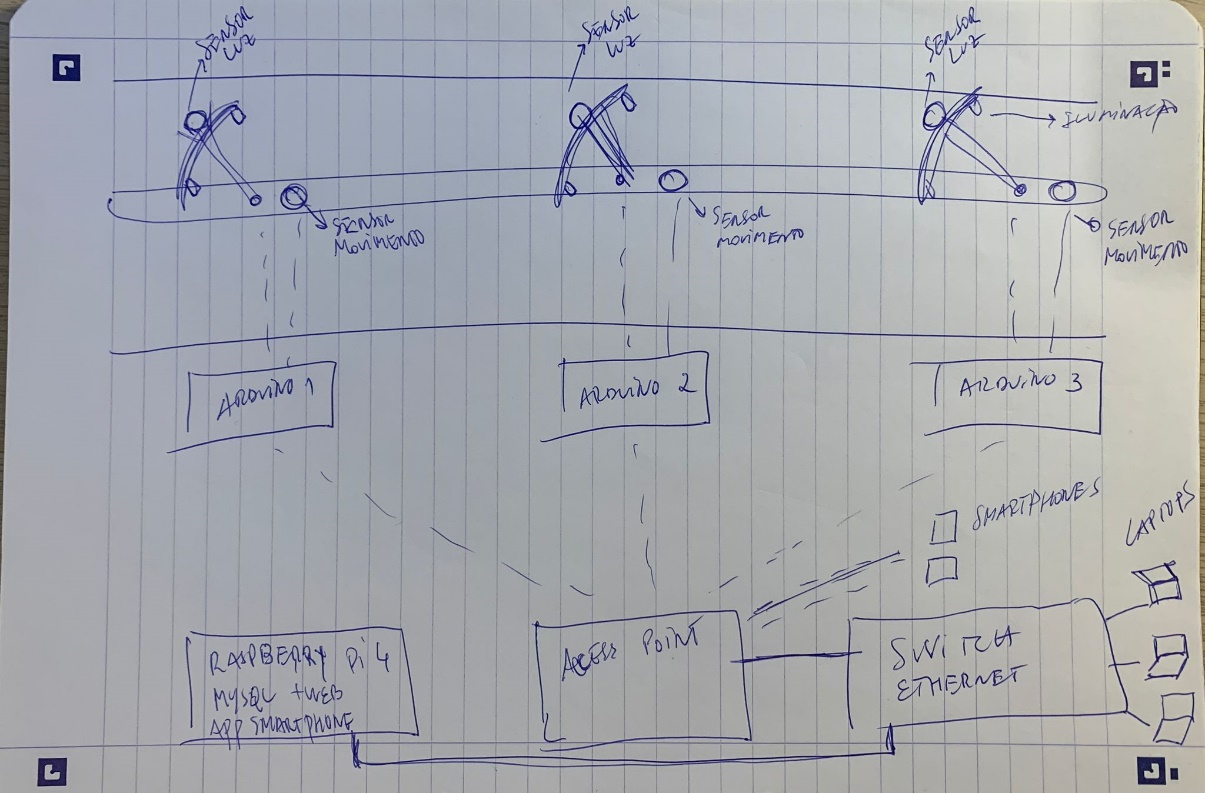


Figura 11 - Esboço do Sistema em papel

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 12 - 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.

Uma imagem com texto, céu, itens, diferente

Descrição gerada automaticamente

Figura 13 - 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.

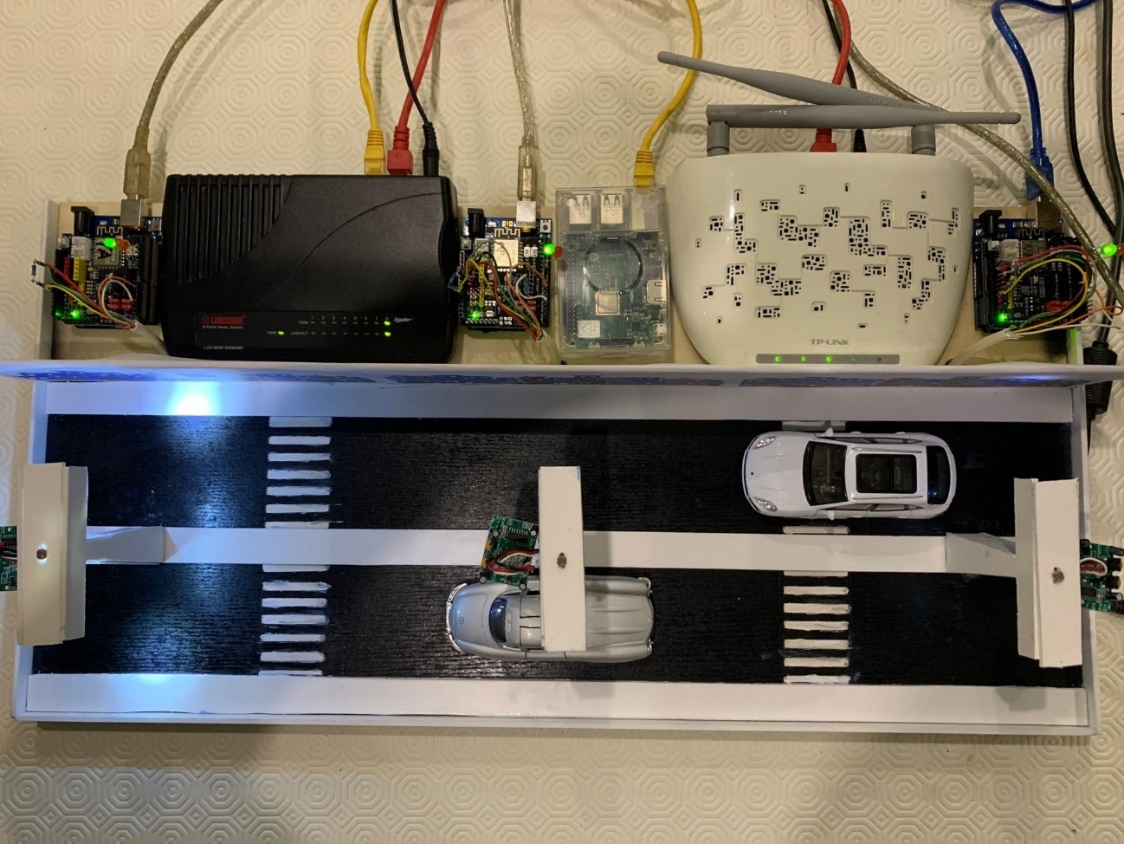


Figura 14 - Protótipo da Maquete do Sistema (visão de cima)

Uma imagem com parede, interior, chão, teto

Descrição gerada automaticamente

Figura 15 - Protótipo da Maquete do Sistema (visão lateral)

Uma imagem com texto, iPod, captura de ecrã

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteAs 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 16 - Mockups da App Móvel

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

## 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 *essidsmartenergy*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  D2 | 2 | LED Branco |
| D3 | 1 | LED Vermelho |
| D4 | 1 | LED Verde |
| R1 | 1 | Resistência de 3 k |
| 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 - Circuito eletrónico no TinkerCad

Este próximo esquema foi também efetuado no TinkerCad, mas desta vez para ter uma representação gráfica e simbológica dos circuitos na implementação do projeto, nomeadamente num poste de iluminação.



Figura - Esquema do circuito eletrónico

# Conclusão

# 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)