**SMART ENERGY**



**Projeto Aplicado**

**Licenciatura em Engenharia de Sistemas Informáticos**

**Regime Pós-Laboral**

**2022/2023**

**Alunos**

Nuno Mendes – Nº 2727

Tiago Azevedo - Nº 21153

Francisco Pereira – Nº 21156

**Orientação**

Profº Eduardo Peixoto

Índice

[1. Introdução 1](#_Toc119701218)

[2. Proposta de sistema 2](#_Toc119701219)

[2.1. Requisitos Funcionais 2](#_Toc119701220)

[2.2. Requisitos Não Funcionais 2](#_Toc119701221)

----- 2.3 tecnologias utilizadas:

hardware: arduino, shield wifi, raspberry, switch, ponto de acesso

software: arduino ide, sistema operativo debian, apache, php, mariadb

[3. Prototipagem 4](#_Toc119701222)

[4. Organização de Grupo 7](#_Toc119701223)

[4.1. Regulamento interno 7](#_Toc119701224)

[4.2. Cronograma 9](#_Toc119701225)

[4.3. Sistema de Avaliação Interno 10](#_Toc119701226)

----- 4.4 atas

[5. Levantamento de Requisitos 11](#_Toc119701227)

[5.1. Casos de uso 11](#_Toc119701228)

[5.2. Diagrama de Modelo de Dados (ER) 12](#_Toc119701229)

[6. Conclusão 14](#_Toc119701230)

[7. Bibliografia 15](#_Toc119701231)

Em cada foto colocar um resumo explicativo

Utilizar o business model canvas

Índice de ilustrações

[Figura 1 - Esboço do Sistema em papel 4](https://d.docs.live.net/dd9d8dc938d9c9fa/Universidade/3_Ano/1_Semestre/SMARTCAMPUS/PA/Relatorio_2727_21153_21156_Fase2.docx#_Toc119701232)

[Figura 2 - Diagrama de Hardware de UM poste de iluminação 5](#_Toc119701233)

[Figura 3 - Diagrama de Hardware de TODO o Sistema 5](#_Toc119701234)

[Figura 4 - Protótipo da Maquete do Sistema (visão de cima) 6](#_Toc119701235)

[Figura 5 - Protótipo da Maquete do Sistema (visão lateral) 6](https://d.docs.live.net/dd9d8dc938d9c9fa/Universidade/3_Ano/1_Semestre/SMARTCAMPUS/PA/Relatorio_2727_21153_21156_Fase2.docx#_Toc119701236)

[Figura 6 - Cronograma do Projeto 9](https://d.docs.live.net/dd9d8dc938d9c9fa/Universidade/3_Ano/1_Semestre/SMARTCAMPUS/PA/Relatorio_2727_21153_21156_Fase2.docx#_Toc119701237)

[Figura 7 - Diagrama de Casos de Uso 11](#_Toc119701238)

[Figura 8 - Diagrama Entidade-Relação 12](#_Toc119701239)

# Introdução

No âmbito da UC de Projeto Aplicado, em coligação com as unidades curriculares de Inteligência Artificial; Programação de Dispositivos Móveis; Sistemas Embebidos e de Tempo Real e Integração de Sistemas de Informação, pretende-se desenvolver um projeto que visa usar os meios tecnológicos da atualidade para tornar um dos setores do nosso campus mais eficiente, tanto em termos funcionais como económicos.

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.

# 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.



Figura 1 - Esboço do Sistema em papel



Figura 2 - Diagrama de Hardware de UM poste de iluminação

Uma imagem com texto, céu, itens, diferente

Descrição gerada automaticamente

Figura 3 - Diagrama de Hardware de TODO o Sistema

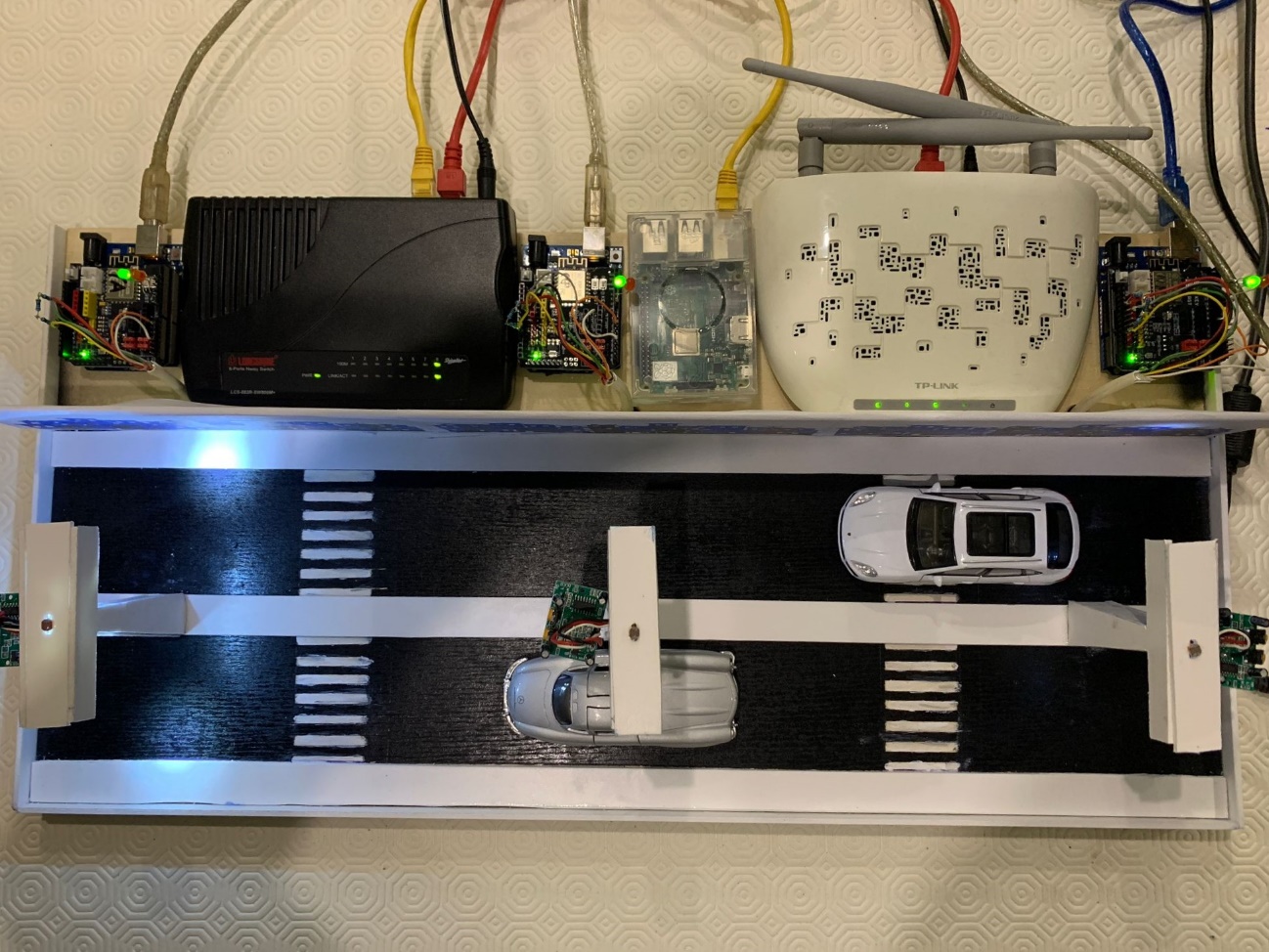


Figura 4 - Protótipo da Maquete do Sistema (visão de cima)

Uma imagem com parede, interior, chão, teto

Descrição gerada automaticamente

Figura 5 - Protótipo da Maquete do Sistema (visão lateral)

# Organização de Grupo

## Regulamento interno

**Artigo 1º - Âmbito do Documento**

Esta parte do documento será utilizada para partilhar a constituição e funcionamento do nosso grupo na elaboração do projeto proposto pelo Professor Eduardo Peixoto na Unidade Curricular de Projeto Aplicado. Poderá também ser observado os deveres e obrigações de todos os elementos deste grupo. Esta documentação poderá estar sujeita a alterações ao longo das diversas fases de entrega, pelo que deve ser verificada pontualmente.

**Artigo 2º - Constituição do Grupo**

O nosso grupo é constituído por três elementos, nomeadamente: Nuno Mendes; Tiago Azevedo e Francisco Pereira. Os nossos contactos são, respetivamente: [a2727@alunos.ipca.pt](mailto:a2727@alunos.ipca.pt) ; [a21153@alunos.ipca.pt](mailto:a21153@alunos.ipca.pt) ; [a21156@alunos.ipca.pt](mailto:a21156@alunos.ipca.pt) .

O nosso orientador é o Professor Eduardo Peixoto que, de forma regular, tem auxiliado na construção e organização do projeto.

**Artigo 3º - Cargos e regularidade da mudança dos mesmos**

O Project Manager tem como principal função assumir a liderança e gestão da equipa, de forma a obter os melhores resultados possíveis mediante os pontos fortes dos restantes elementos. Deverá fazer o agendamento de reuniões, assim como a atribuição de tarefas pendentes.

O Secretário deverá garantir a preparação dos documentos a serem abordados nas reuniões, assim como efetuar as atas das mesmas.

O Lead Developer fica responsável por liderar o elemento técnico deste projeto, garantindo o funcionamento das novas *features* que serão adicionadas ao longo do tempo.

Os cargos apresentados aqui, poderão estar sujeitos a alteração, principalmente a cada entrega ou, em último caso, poderão ser trocados mediante a necessidade de ajuda extra num dos setores do projeto.

**Artigo 4º - Reuniões**

As reuniões dão lugar nas aulas de Projeto Aplicado, ou seja, duas vezes por semana. Desta forma é possível garantir a presença e disponibilidade de todos os elementos, tal como a presença do nosso Orientador.

É de frisar que o ponto inframencionado poderá não ser cumprido mediante compromissos de cariz pessoal ou escolar. Nesse caso as reuniões serão marcadas para uma data a definir através dos nossos canais de comunicação direta.

Mais se informa que no final de cada reunião, deverá ser elaborada a ata da mesma de forma a todos os elementos poderem consultar o conteúdo da mesma à posteriori.

**Artigo 5º - Atas e Convocatórias**

As Convocatórias poderão ser efetuadas por qualquer elemento do grupo, caso o mesmo ache apropriado uma reunião. Esta regra não invalida o dever do elemento em questão de avisar o Project Manager de tal intenção.

Como referido em cima, a ata será responsabilidade do Secretário, sendo que deverá ser validada por todos os elementos do grupo presentes na reunião.

## Cronograma

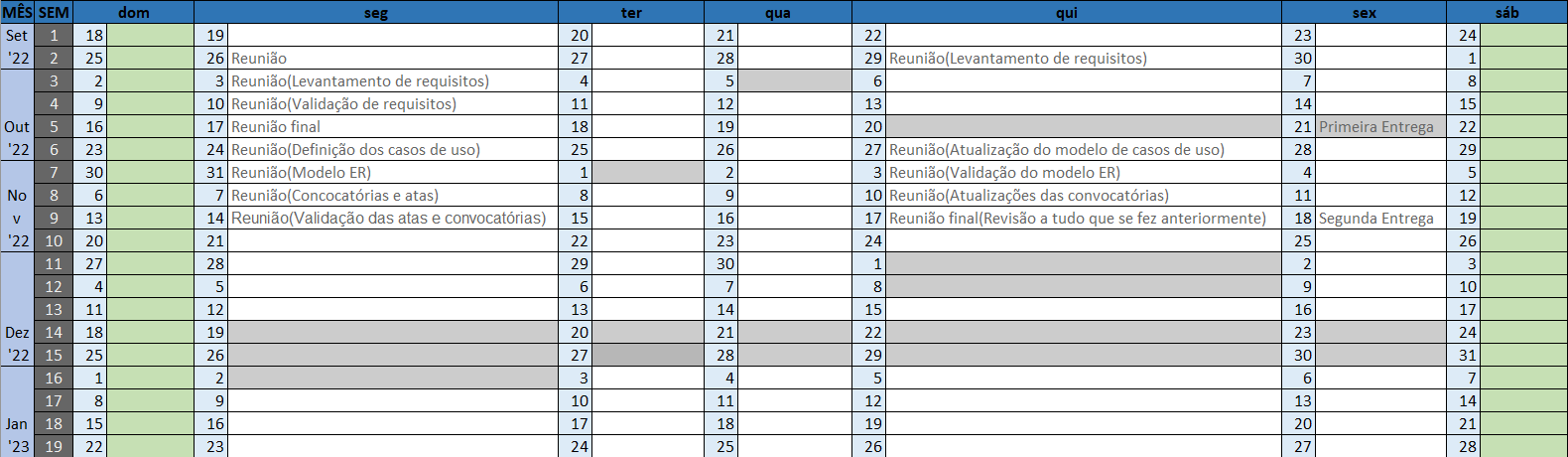
De forma a exacerbar a organização interna, foi efetuado um cronograma com as datas relevantes para o melhoramento contínuo do nosso projeto. Este setor poderá, ao longo do tempo, ser alterado mediante as necessidades e prazos de entrega do nosso trabalho contínuo.

Figura 6 - Cronograma do Projeto

## Sistema de Avaliação Interno

A avaliação interna do grupo tem o intuito de fazer um ponto de situação do desempenho individual de cada elemento envolvido no projeto. Esta avaliação será efetuada ao fim de cada fase de entrega do projeto, sendo que as notas serão dadas de forma individual por cada elemento do grupo aos seus respetivos colegas. Por sua vez, esse documento será entregue, de forma individual, por cada elemento do grupo ao orientador do projeto via e-mail.

Todos os membros começam com uma nota inicial de **0** valores, sendo descontado/adicionado valores conforme os seguintes tópicos:

1. **Reuniões (8 valores)**
   * 1. Assiduidade/Pontualidade
     2. Participação
     3. Comunicação
     4. Respeito/Relação entre colegas
2. **Projeto (8 valores)**
   * 1. Cumprimento de prazos
     2. Organização
     3. Iniciativa
     4. Qualidade do trabalho
     5. **Autonomia (4 valores)**

# Levantamento de Requisitos

## Casos de uso

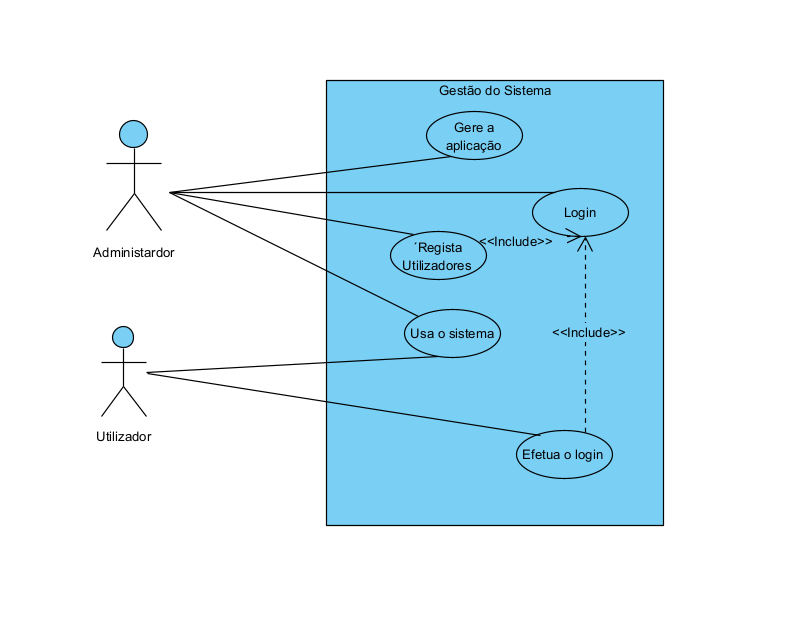


Figura 7 - Diagrama de Casos de Uso

Este diagrama foi concebido para demonstrar os casos de uso presentes no nosso projeto. Até ao momento, existem apenas dois atores: **Administrador** e **Utilizador**.

O papel do Administrador será gerir toda a aplicação, em questões de funcionamento e segurança da mesma. Este ator tem o poder para registar novas contas de utilizador, assim como o uso completo da aplicação.

Já o Utilizador poderá fazer o uso supervisionado do sistema, nomeadamente para recolha de dados e manutenção da aplicação. Este diagrama poderá ser alterado mais tarde mediante as necessidades inerentes ao projeto.

## Diagrama de Modelo de Dados (ER)



Figura 8 - 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.

# 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).

----- 2.3 tecnologias utilizadas:

Hardware

Raspberry Pi 3 Model B+, 1.4GHz, 1GB, cartão SD de 16GB

<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>

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.

Ponto de Acesso de redes sem fios

Neste equipamento foi configurada uma rede wireless com *essid* *smartenergy* segura com encriptação para permitir a comunicação dos *shield wifi* dos *arduinos* com o servidor.

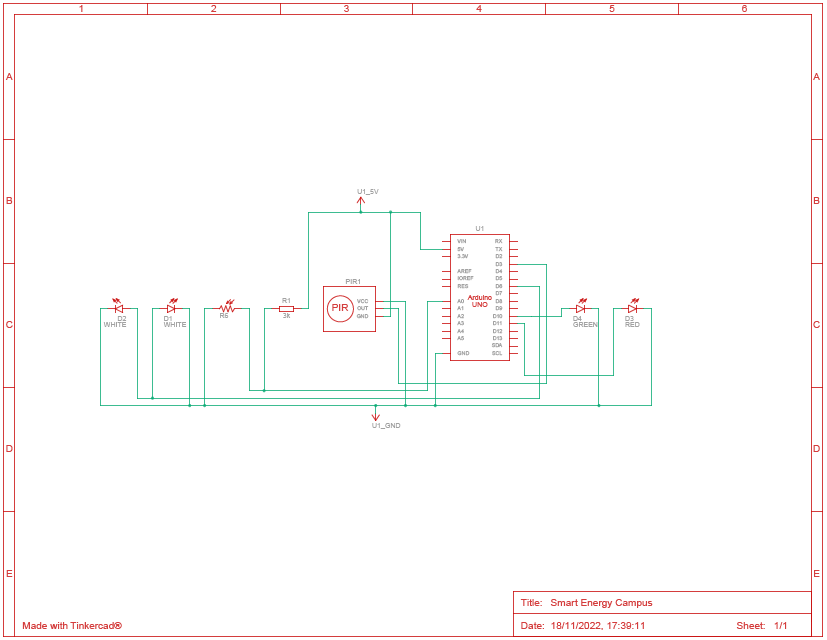
*Switch ethernet*

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,…

OSOYOO WiFi Internet of Things Learning Kit For Arduino

<https://osoyoo.com/2020/05/30/wifi-iot-learning-kit-for-arduino/>

Este kit *Arduino* é um clone fiel do original e mais económico na aquisição, incluiu o dispositivos necessários para implementação do protótipo projeto numa maquete, os componentes essenciais foram 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.



Esquema do circuito eletrónico



Montagem dos componentes em *bread board*

Tabela de componentes eletrónicos necessários:

|  |  |  |
| --- | --- | --- |
| **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 kW |
| R6 | 1 | Foto resistor (LDR) |

Software:

Sistema operativo escolhido para a instalação e desenvolvimento no Raspberry Pi foi o Debian e incluídos os pacotes para disponibilização de serviços web o Apache2 e o PHP 8.1, para a base de dados foi o 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 dos dados transmitidos possam ser guardados na base de dados, o ficheiro no servidor que é invocado pelos arduinos com a passagem de parâmetros para suportar esta comunicação denomina-se *webservices.php*, alojado no servidor em */var/www/html*.

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

**Para abranger a cadeira de SETR incluir também:**

Contextualização e motivação do projeto;

Descrição da arquitetura do sistema; (creio que esteja no capitulo tecnologias utilizadas hardware e software)

Código(s) desenvolvido(s);

ANEXOS??

**Código fonte do projeto**

Código fonte dos ficheiros existentes no serviços localizados em */var/www/html*:

*config.php*

|  |
| --- |
| <?php  flush();  // Configurações gerais  $pasta="/var/www/html/";   // localizacao do site;  // Configuração do servidor mysql  $bd\_server = "10.10.10.2";    // 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  ?> |

*webservices.php*

|  |
| --- |
| <?php  // ficheiro com as todas as configuracoes 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...";  }  ?>  Código fonte desenvolvido no IDE do Arduino: |

*SmartEnergyCampus.ino*

|  |
| --- |
| #include <WiFiEsp.h>  #include <WiFiEspClient.h>  #include <WiFiEspServer.h>  #include <WiFiEspUdp.h>  /\*    Project: Smart Energy Campus    Year: 2022 / 2023          LESI @ IPCA    Authors: 2727 Nuno Mendes             21138 Rosario Silva             21153 Tiago Azevedo             21156 Francisco Pereira  \*/  #include "WiFiEsp.h"  #include "WiFiEspClient.h"  //#ifndef HAVE\_HWSERIAL1  #include "SoftwareSerial.h"  SoftwareSerial softserial(4, 5);  // RX, TX  //#endif  // network info variables  byte mac[6];  IPAddress ip;  char serveraddress[] = "10.10.10.2";  // web & database server address  char ssid[] = "smartenergy";   // your network SSID (name)  char pass[] = "20222023lesi";  // your network password  // Initialize the Ethernet client object  WiFiEspClient client;  int status = WL\_IDLE\_STATUS;  int reqCount = 0;  // number of requests received  #define LED 6         // pino de output dos LEDs, porta PWM  #define LDR 0         // pino de input do sensor de luz  #define PIR 3         // pino de input do sensor de movimento  #define LEDWIFION 10  // wireless conetado e a funcionar  #define LEDWIFIOFF 11 // wireless nao conetado  #define MAXLED 24     // LED maximum value during tests  #define LDRmax 1000   // LDR maximum input  #define LDRmin 40     // LDR minimum input  #define LDRmed 600    // 600 para efeitos de testes dentro de casa, 400 no IPCA  #define TIMEmax 15    // tempo maximo LEDs ligados  #define valLEDmin 2   // valor dos LEDs quando ligados mas sem movimentom, em standby  #define valINCREMENT 4 // valor de incremento / decremento na suavizacao de alteracao do valor da iluminacao  int counter = 0;  int valLED = 0;       // valor inicial  int stateLED = LOW;   // valor inicial do estado dos LEDs desligados  int valLDR = 0;       // valor inicial  long valLDRnew = 0;   // valor incicial  int valPIR = 0;       // valor inicial  int statePIR = LOW;   // sem deteção de movimento  uint32\_t timer = 0;   // temporizador para o tempo dos LEDs ligados  uint32\_t timer2 = 0;  WiFiEspServer server(80);  void info() {    // informacao inicial no arranque do sistema    Serial.println("\n\n\n\n");    Serial.println("Smart Energy Campus version 0.3 @ IPCA 2022/2023\n");    Serial.println(" 2727 Nuno Mendes");    Serial.println("21138 Rosario Silva");    Serial.println("21153 Tiago Azevedo");    Serial.println("21156 Francisco Pereira");    delay(1000);  }  void test() {    Serial.println("\n[Testing]");    delay(1000);    Serial.print("\n- Ilumination...");    // aumentar o brilho    Serial.print("\n Bright up ");    for(int i=0; i<=255; i++) {      //if ((i % 2) == 0) Serial.print("+");      ++i;      analogWrite(LED, i);      delay(10);    }    // reduzir o brilho    Serial.print("\n Bright down ");    for(int i=255; i>=0; i--){      //if ((i % 2) == 0) Serial.print("-");      --i;      analogWrite(LED, i);      delay(10);    }    analogWrite(LED, 0);    // desliga os LEDs    Serial.print("\n\n- Light Sensor LDR... ");    for(int i=0; i<=127; i++) {      if ((i % 30) == 0) Serial.print("\n");      valLDR = analogRead(LDR);      Serial.print(" ");      Serial.print(valLDR);      delay(10);    }    valLDR = 0;    Serial.print("\n\n- Motion Sensor PIR... ");    for(int i=0; i<=127; i++) {      if ((i % 60) == 0) Serial.print("\n");      valPIR = digitalRead(PIR);      Serial.print(" ");      Serial.print(valPIR);      delay(10);      //if (i == 255) i=0;    }    valPIR = LOW;    Serial.print("\n\n- Network LEDs... ");    for(int i=0; i<=10; i++) {      analogWrite(LEDWIFION, 0);      analogWrite(LEDWIFIOFF, MAXLED);      delay(100);      analogWrite(LEDWIFION, MAXLED);      analogWrite(LEDWIFIOFF, 0);      delay(100);    }    analogWrite(LEDWIFION, 0);    analogWrite(LEDWIFIOFF, 0);    Serial.println("\n");  }  void setup() {    pinMode(LED, OUTPUT);    pinMode(LEDWIFION, OUTPUT);    pinMode(LEDWIFIOFF, OUTPUT);    pinMode(LDR, INPUT);    pinMode(PIR, INPUT);    analogWrite(LED, 0);    // Desliga os LEDs    analogWrite(LEDWIFION, 0);    analogWrite(LEDWIFIOFF, MAXLED);    // initialize serial for debugging    Serial.begin(9600);    // Info do projeto    info();    // Testando os componentes externos    //test();    // initialize serial for ESP module    analogWrite(LEDWIFIOFF, MAXLED);    Serial.println("\n[Network connection]");    softserial.begin(115200);    softserial.write("AT+CIOBAUD=9600\r\n");    softserial.write("AT+RST\r\n");    softserial.begin(9600);    // initialize ESP module    WiFi.init(&softserial);    // check for the presence of the shield    if (WiFi.status() == WL\_NO\_SHIELD) {      Serial.println("WiFi shield not present");      // don't continue      while (true)        ;    }    // attempt to connect to WiFi network    while (status != WL\_CONNECTED) {      analogWrite(LED, 255);      analogWrite(LEDWIFION, MAXLED);      analogWrite(LEDWIFIOFF, 0);      delay(100);      analogWrite(LED, 64);      analogWrite(LEDWIFION, 0);      analogWrite(LEDWIFIOFF, MAXLED);      delay(100);      analogWrite(LED, 255);      analogWrite(LEDWIFION, MAXLED);      analogWrite(LEDWIFIOFF, 0);      delay(100);      analogWrite(LED, 64);      analogWrite(LEDWIFION, 0);      analogWrite(LEDWIFIOFF, MAXLED);      delay(100);      Serial.print("Attempting to connect to WPA SSID: ");      Serial.println(ssid);      // Connect to WPA/WPA2 network      status = WiFi.begin(ssid, pass);      analogWrite(LED, 0);    }    Serial.println("You're connected to the network");    printWifiStatus();    analogWrite(LEDWIFION, MAXLED);    analogWrite(LEDWIFIOFF, 0);    // start the web server on port 80    server.begin();  }  void inputs() {    valLDR = analogRead(LDR);             // leitura do valor do sensor LDR    valPIR = digitalRead(PIR);            // leitura do valor do sensor de movimento PIR    if (valPIR == HIGH) {      statePIR = HIGH;  // estado de detecao de movimento passa a TRUE      sendDataToServer();    }  }  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(100);    timer2 = millis();    // regista o tempo atual    if (statePIR == HIGH && stateLED == LOW) {      statePIR = LOW;    }  }  void loop() {    // funcao para fazer a leitura de todos os inputs (sensores)    inputs();   // leitura de sensores    outputs();  // comandos para os atuadores    if (counter > 200) {         sendDataToServer();         counter = 0;    }    ++counter;  void printWifiStatus() {    // print the SSID of the network you're attached to    Serial.print("SSID: ");    Serial.println(WiFi.SSID());    // print your WiFi shield's IP address    ip = WiFi.localIP();    Serial.print("IP Address: "); Serial.println(ip);    WiFi.macAddress(mac);    Serial.print("MAC: ");    Serial.print(mac[5],HEX);  Serial.print(":");    Serial.print(mac[4],HEX);  Serial.print(":");    Serial.print(mac[3],HEX);  Serial.print(":");    Serial.print(mac[2],HEX);  Serial.print(":");    Serial.print(mac[1],HEX);  Serial.print(":");    Serial.println(mac[0],HEX);    // print where to go in the browser    Serial.println();    Serial.print("To see this page in action, open a browser to http://");    Serial.println(ip);    Serial.println();  }  void sendDataToServer() {    analogWrite(LEDWIFION, 0);    analogWrite(LEDWIFIOFF, 0);    Serial.println("\n\nSending data to server\n");  // close any connection before send a new request    // this will free the socket on the WiFi shield    client.stop();    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);    s1 += String(ip[0])+String(".")+String(ip[1]) +String(".")+String(ip[2])+String(".")+String(ip[3]);    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));    // if there's a successful connection    if (client.connect("10.10.10.2", 80)) {      // pisca led VERDE para sinalizar comunicacao com o servidor com sucesso      analogWrite(LEDWIFION, MAXLED); delay(100);      client.println( (s1));      analogWrite(LEDWIFION,0); delay(100);      client.println(F("Host: 10.10.10.2"));      analogWrite(LEDWIFION,MAXLED); delay(100);      client.println("Connection: close");      analogWrite(LEDWIFION,0); delay(100);      client.println();      analogWrite(LEDWIFION,MAXLED); delay(100);      // note the time that the connection was made    }    else {      // if you couldn't make a connection      // picas led VERMELHO para sinalizar erro de comunicacao com o servidor      Serial.println(F("Connection failed"));      analogWrite(LEDWIFIOFF, MAXLED); delay(100);      analogWrite(LEDWIFIOFF, 0); delay(100);      analogWrite(LEDWIFIOFF, MAXLED); delay(100);      analogWrite(LEDWIFIOFF, 0); delay(100);    }    analogWrite(LEDWIFION,MAXLED);    analogWrite(LEDWIFIOFF, 0);  } |