universidade de aveiro deti departamento de electrónica, telecomunicações e informática

41488 – Projeto Industrial

Documentação:Cliente/Servidor

Nome do Projeto:	Utilização de tecnologia Bluetooth Low Energy para rastreamento de pessoas numa loja física
Cliente:	Entropic Ventures Unip Lda (David Carvalhão)
Membros da equipa:	Coordenador: Arnaldo Oliveira arnaldo.oliveira@ua.pt Contato principal: João Domingues joaofdomingues@ua.pt (JD) Membros: Rui Lemos rjbs.lemos@ua.pt (RL) Manuel Silva manuelnps@ua.pt (MS) José Santos jmpsantos@ua.pt (JS) Bruno Ravara brunoravara@ua.pt (BR) João Viegas jpviegas@ua.pt (JV)
Data:	

Descrição do módulo:

Neste modulo iremos entrar em detalhe da estrutura do nosso sistema desde a ligação ao servidor web até ao envio para a base de dados dos valores da posição de cada utilizador.

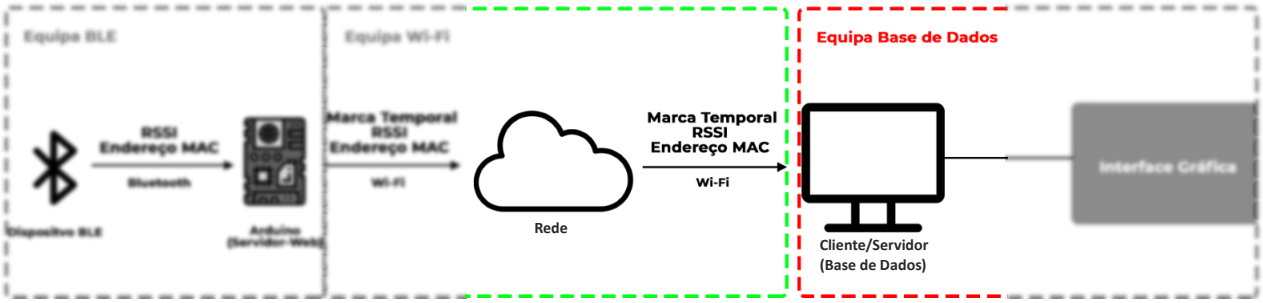

Figura 1 - Arquitetura do sistema

Este módulo tem como objetivo resgatar dados dos vários servidores web (criados por cada Arduíno [Ver Documentação: Arduino"]), agrupar valores de RSSI do mesmo dispositivo obtidos pelos três Arduinos, depois iremos traduzir os valores de RSSI para metros e assim poderemos finalmente calcular a posição de cada dispositivo, esta que irá ser guardada em tabelas na base de dados para posteriormente ser exposto na interface gráfica [Ver *"Documentação: Interface Gráfica"*].

Instruções de Instalação

Quanto à instalação, para poder correr o código distribuído é necessário primeiramente instalar *python (versão 3.9)* e um IDE que suporte *python* (Visual Studio versão 16.9.5), de seguida será necessário verificar se todas as bibliotecas usadas no nosso código estão instaladas sendo estas:

Bibliotecas Usadas:			
Math	Time	Random	Spicy.optimize
random	matplotlib.animation	matplotlib.pyplot	itertools
numpy	requests	bs4	re

Outra ferramenta essencial para o funcionamento do código é a instalação do serviço de base de dados MySQL, em que é necessário instalar o MySQL server e a *Workbench* (versão 8.0.25) para podermos operar na base de dados sem ser pelo código principal. No setup de instalação, ao escolher os produtos a instalar deverá ter uma janela com o aspeto da Figura 2.

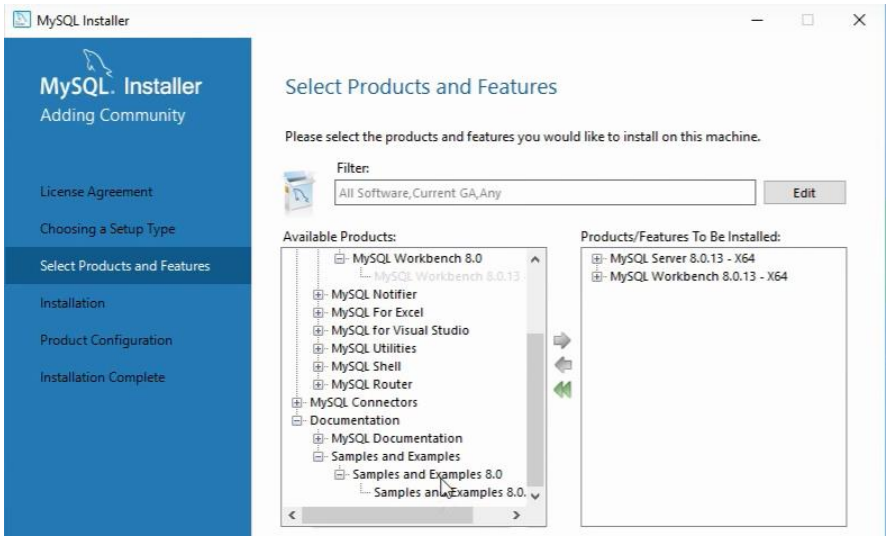

Figura 2 - Instalação de MySQL Server

Outro passo importante na instalação do MySQL server é quando nos é pedido a introdução de uma palavra passe inserir "BLElocation!" sem as aspas, isto irá servir para não ser necessária qualquer alteração no código principal. Iremos ter então uma janela com o aspeto da Figura 3.

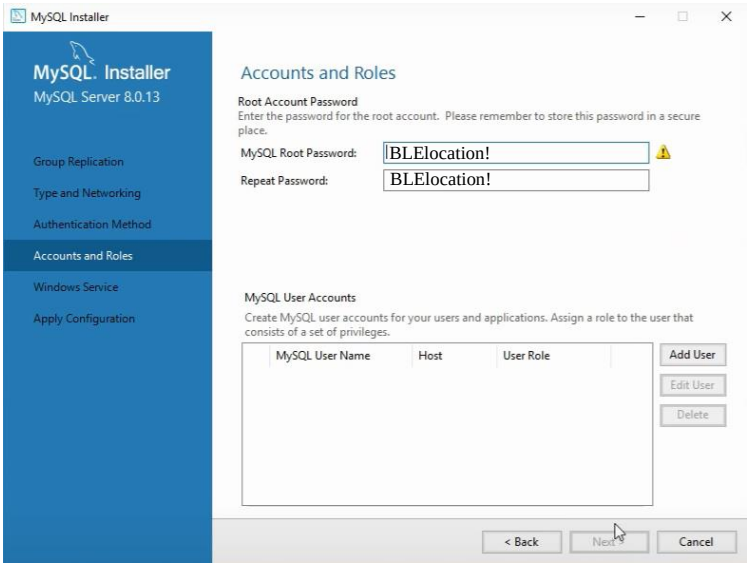

Figura 3 - Introdução da palavra passe

Após as instalações anteriores é necessário fazer o *setup* apropriado a base de dados,criando na MySQL Workbench uma schema com o nome “testpi” como vemos na Figura 4 .

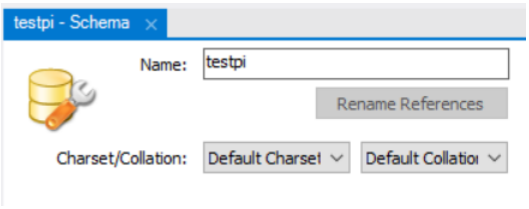

Figura 4 - Criação do Schema

Se no passo anterior tiver escrito uma palavra passe diferente terá de alterar a linha de código 30 a frase PASSWORD para a nova palavra passe como podemos ver na Figura 5, no lugar da linha 31 irá estar o nome da schema criada neste passo (“testpi”), se não escolheu outro nome não é necessário alterar mais nada neste ponto.

```
25  #-------------- 1:Initialize BD ---------------#
26  import mysql.connector
27  mydb = mysql.connector.connect(
28      host="localhost",
29      user="root",
30      password="PASSWORD!",
31      database="NOME_DA_BASE_DE_DADOS"
32      )
33  mydb.autocommit = True
34  mycursor = mydb.cursor()
35
```

Figura 5 - Secção de código a alterar

Também será necessário criar as tabelas com os nomes e variáveis específicos para não haver conflito no código, estas definições das tabelas iram ser disponibilizadas juntamente com o código principal.

Por fim é necessário introduzir os ips a qual os arduinos estão a usar [Ver Documentação: Arduino] como podemos ver na Figura 6.

```
80      url0 = 'http://192.168.1.104/'  #Arduino na posição 1
81      url1 = 'http://192.168.1.103/'  #Arduino na posição 2
82      url2 = 'http://192.168.1.101/'  #Arduino na posição 3
```

Figura 6 - Ips dos Arduinos

Arquitetura do código

Para a arquitetura do código este foi dividido em 10 módulos, 4 dos mesmos fazem parte do setup inicial e os restantes fazem parte de um Loop a qual cada um tem o seu papel a desempenhar como podemos ver na Figura 2. De seguida iremos analisar cada bloco individualmente para perceber o funcionamento de cada um.

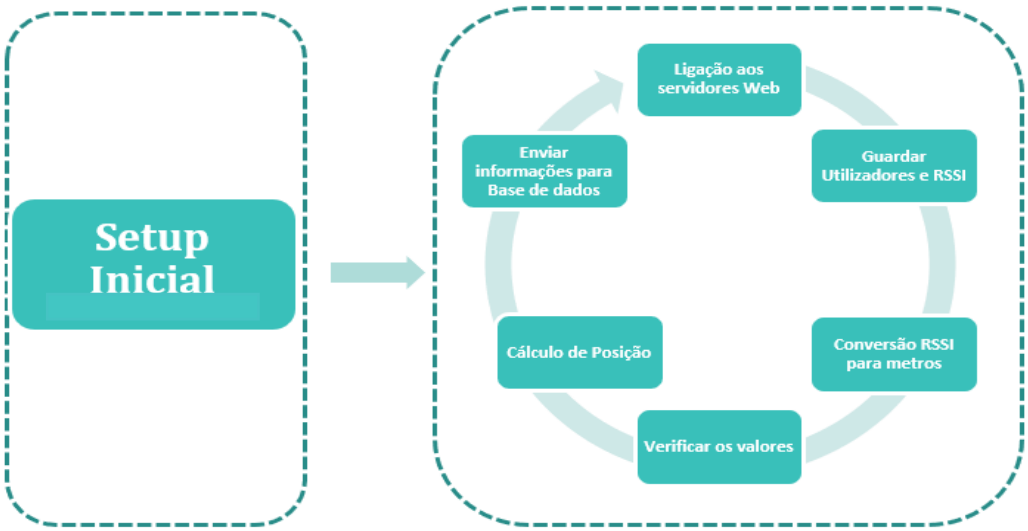

Figura 7 - Ciclo de execução

1. ***Initialize BD***

   Inicializa a base de dados (MySQL);

2. ***Hashing User ID***

   Função para dar *Hash* ao endereço MAC mas que não a chegamos a usar pois o dicionário de *Hashing* mudava a cada compilação;

3. **Inicializar algumas Variaveis**

   Iniciamos a *timestamp* e imprimimos (nesta versão não usamos este valor em nenhuma ocasião);

4. **Configuração Inicial**

   Vamos à base de dados da tabela *saveflag* verificar-se a configuração das posições iniciais foram feitas (1-*True* 0- *False*);

   O código fica preso aqui se a configuração não for feita na interface gráfica (que muda a *flag* para 1 na BD quando se grava);

   Esta posição inicial é essencial para o cálculo da posição;

5. **Adquirir dados do *webserver* (Modulo feito por João Viegas)**

   Vamos nos ligar em primeiro ao *webserver* do arduino 1 com um IP introduzido aqui manualmente (Ainda não conseguimos tornar o IP estático de cada arduino);

   O código fica preso até se ligar ao *webserver1*, depois guarda todos os dados numa *string* 'info' e envia para a tabela *rawdata* 1 o endereço MAC e o valor de RSSI e para a tabela *Users* o endereço MAC;

   Após se ligar ao *webserver1* passa a tentar ligar-se ao *webserver* 2 e 3 e envia os dados agora para a tabela *rawdata2* e *rawdata3* respetivamente (e para a tabela *Users* também)

6. **Conversão de RSSI para Metros**

   Aqui criamos uma função que pega no valor de RSSI e converte para metros;

   Temos duas versões, a comentada é uma versão inicial que tentamos implementar de outros autores, a segunda que é a que estamos a usar atualmente foi criada por um membro do grupo

(João Domingues) que definiu que mais se aproximava aos resultados pretendidos, esta formula pode ser vista na Figura abaixo;

$$Distância = 10^{\frac{RSSI_{1Metro} - RSSI_{Medido}}{10*N}}$$

Figura 8 - Formula de conversão de RSSI para metros

Nesta versão também temos que para valores de RSSI iguais ou maiores que |-120| (valor a qual o erro era bastante significativo >10%) devolve '-1';

7. Ciclo de utilizadores

Esta secção serve para percorremos a tabela *Users* e a cada endereço MAC vamos a cada tabela dos dados *raw* (*rawdata#*) e retirar o valor de RSSI obtido em cada arduino;

8. Verificar BLE's FUNCIONAIS

Aqui fazemos a seleção dos arduinos que deram valores corretos de RSSI (diferente de -1 como vimos no passo 6.)

9. Algoritmo para o cálculo da Posição

Nesta parte fazemos agora o cálculo da posição de cada *User*, este cálculo é feito pela aproximação inicial da posição da pessoa (no meio do espaço quando não temos um valor de posição anterior ou quando temos tomamos esse mesmo valor) que na Figura abaixo está representado por um Triangulo. O algoritmo irá minimizar (pelo método dos mínimos quadrados) a soma dos erros que temos dessa aproximação e do valor obtido pelo RSSI por cada Arduino (representados por losangos) até chegar a posição marcada por uma Cruz na Figura 9;

$$Minimize\left(\sum R_i - Error_i\right)\ com\ i = 1{,}2{,}3$$

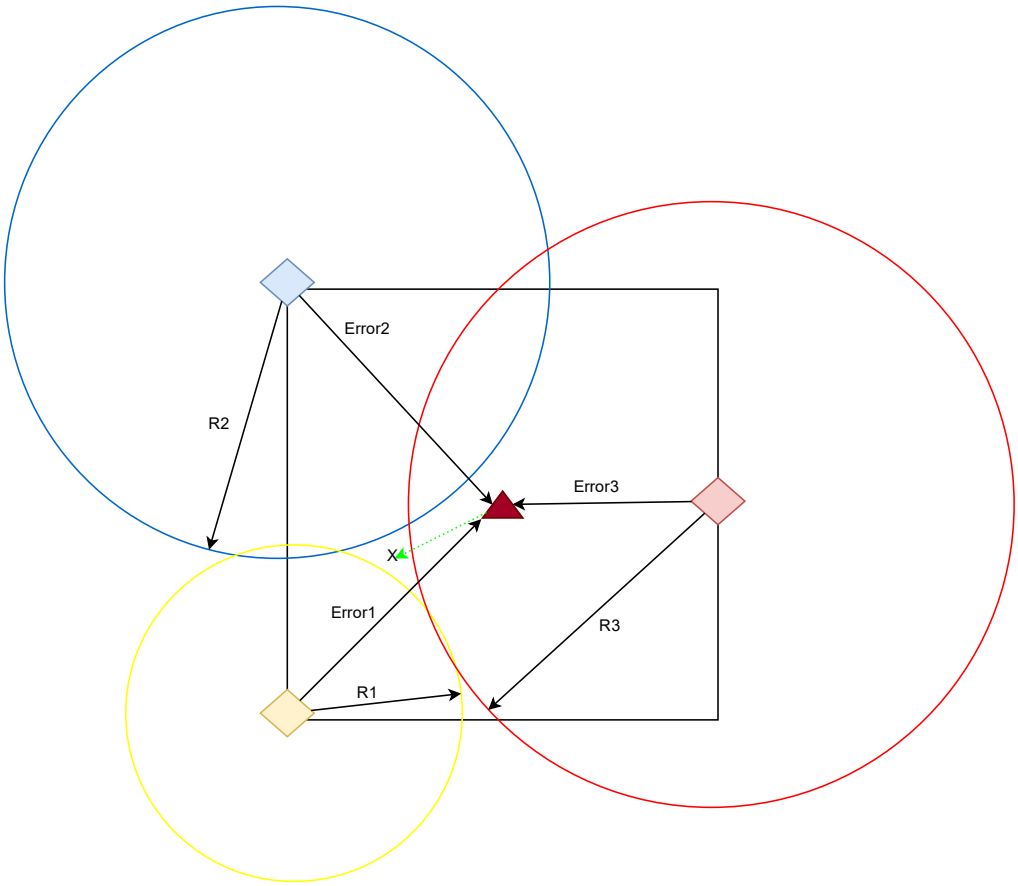

Figura 9 - Representação do cálculo a posição

10. TO DATABASE

Por fim enviamos as posições para a tabela *location* e histórico em que uma tem apenas as posições atuais e a outra tem o histórico das localizações;

Objetivos alcançados

Com este código conseguimos calcular a posição de cada utilizador e enviar para a base de dados com o objetivo da interligação com a interface com o utilizador. O código cumpre os objetivos definidos pela equipa, possíveis melhoramentos seria algum tipo de proteção para se não fosse possível estabelecer ligação a um dos servidores web o código não ficaria preso até a ligação ser feita.