



Desenvolvimento de Módulo de Localização para Transportes

Diogo Filipe Ferreira Martins

Dissertação do Mestrado em Engenharia Industrial apresentado à Escola Superior de Tecnologia e de Gestão do Instituto Politécnico de Bragança.

Trabalho orientado por:

Prof. José Luís Lima

Prof. Ana Isabel Pereira

Prof. Sérgio Ivan Lopes

Bragança

2021



Desenvolvimento de Módulo de Localização para Transportes

Diogo Filipe Ferreira Martins

Dissertação do Mestrado em Engenharia Industrial apresentado à Escola Superior de
Tecnologia e de Gestão do Instituto Politécnico de Bragança.

Trabalho orientado por:

Prof. José Luís Lima

Prof. Ana Isabel Pereira

Prof. Sérgio Ivan Lopes

Bragança

2021

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Agradecimentos

Nesta secção queria aproveitar para agradecer a todas as pessoas que, direta ou indiretamente, contribuíram para o meu percurso pessoal e académico, tendo-me ajudado a superar este desafio. Primeiramente quero aqui expressar a minha gratidão aos meus pais por tudo que fizeram por mim, e por me darem a possibilidade de terminar este mestrado. Quero agradecer aos meus amigos que me ajudaram e estiveram sempre presentes. Agradecer ao orientador Prof. José Lima e à coorientadora Prof. Ana Isabel Pereira não só pelas orientações e ajuda durante esta dissertação, mas também pela sua excelente forma de ensino e relação com os alunos. Por fim agradecer também ao Prof. Sérgio Ivan Fernandes Lopes pelas sugestões de melhoria e apoio durante o desenvolvimento desta dissertação.

Abstract

The term smart city has become very prominent in recent years, especially in terms of public mobility.

The primary objective of this project is to develop a modular system for monitoring and tracking public transport. The module, named SMARTMOV device, will be installed in different types of transport and will have the function of collecting the location sent by the GPS integrated in the module and send it through WIFI to a server where it will happen for a possible optimization of the services offered, analyze the flow of public transport users.

The developed module consists of an ESP32 that specifies that everything works together to grant the desired data and send them to a database through the WIFI function, a GPS that is responsible for sending information regarding your location, a laser sensor to collect data regarding the flow of passengers. So this dissertation goes against the ideology of *IoT*, which is nothing more than physical objects linked to exchanging data with other devices and systems over the Internet.

One of the future objectives of the work developed in this dissertation is to use all the data collected, and make a flow forecast, making it possible to know if a particular stop or even a route is necessary for a change, and it is even possible to create new routes.

Keywords: SMARTMOV device, ESP32, GPS, WIFI, IoT, flow, Data Base;

Resumo

O termo cidade inteligente tornou-se muito proeminente nos últimos anos, especialmente na vertente da mobilidade pública.

O objetivo primordial desse projeto visa desenvolver um sistema modular de monitorização e rastreamento de transportes públicos. O módulo, de nome Dispositivo SMARTMOV, será instalado em diferentes tipos de transportes e terá a função de recolher a localização enviada pelo GPS integrado no módulo e envia-o através de WIFI para um servidor onde serão analisados para uma possível otimização dos serviços oferecidos, análise de fluxo de utilizadores de transportes públicos.

O módulo desenvolvido é constituído por um ESP32 que assegura que tudo trabalhe em conjunto para a recolha dos dados pretendidos e envio dos mesmos para a base de dados através da função WIFI, um GPS que é responsável pelo envio das informações relativamente à sua localização, um sensor laser para recolher os dados relativamente ao fluxo de passageiros. Assim, esta dissertação vai de encontro à ideologia da *IoT*, que é nada mais que objetos físicos conectados a trocar dados com outros dispositivos e sistemas pela Internet.

Um dos objetivos futuros do trabalho desenvolvido nesta dissertação é utilizar todos os dados recolhidos, e fazer uma previsão de fluxo, sendo possível saber se uma determinada paragem ou até mesmo rota, necessita de uma alteração, sendo possível até criar novas rotas.

Palavras-chave: Dispositivo SMARTMOV, ESP32, GPS, WIFI, IoT, fluxo, Base de Dados;

Lista de Acrónimos

GPS - Global Positioning System.

STCP - Sociedade de Transportes Colectivos do Porto.

IoT - Internet of Things.

IIoT - Industrial Internet of Things.

OBD - On-Board Diagnostic

MCU - Multipoint Control Unit.

Conteúdo

Agradecimentos	vii
Abstract	ix
Resumo	xi
Lista de Acrónimos	xii
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	3
1.3 Estrutura do Documento	3
2 Estado da Arte	5
2.1 Internet of Things (IoT)	5
2.2 IoT Industrial	7
2.3 Exemplos de Aplicações da IoT e IIoT	7
2.4 Projetos/Empresas Existentes	9
2.4.1 Projetos na Área Industrial	9
2.4.2 Projetos na Área de Mobilidade	10
2.5 Módulos Existentes	11
2.6 Sociedade de Transportes Colectivos do Porto	13
2.7 Proposta de Solução	13

3	Implementação	17
3.1	Hardware	17
3.1.1	ESP32	18
3.1.2	LM2596S DC-DC	19
3.1.3	GPS GY-NEO6MV2	20
3.1.4	Sensor Laser - Referência :SEN05037	21
3.2	Base de Dados	22
4	Desenvolvimento Dispositivo SMARTMOV	25
4.1	Arquitetura do sistema	25
4.2	Esquema de Ligações	26
4.3	Placa de Circuito Impresso Desenvolvida	28
4.4	Programação	30
4.5	Base de Dados	32
4.5.1	Base de dados Relacional	33
4.5.2	MySQL	34
4.5.3	Criação da base de dados	35
4.6	MySQL Workbench	38
4.7	Servidor Local	39
4.7.1	XAMPP e WAMP	39
4.7.2	Configuração XAMPP	40
5	Resultados e Validação	43
5.1	Teste Sensor Laser	43
5.2	Teste GPS	47
5.2.1	Validação dos Dados Recolhidos	48
6	Conclusões e Trabalho Futuro	59
A	Código Arduino e PHP	62

Lista de Tabelas

4.1	Tabela de custo dos componentes	29
5.1	Tabela de valor recolhido de fluxo a altura de 89cm	44
5.2	Tabela de valor recolhido de fluxo a altura de 60 cm	46
5.3	Tabela de valor recolhido de fluxo a altura de 60 cm com alteração no código Arduino	46
5.4	Tabela valores recolhidos primeiro teste, trajeto número 1.	51
5.5	Tabela valores recolhidos primeiro teste, trajeto número 2.	52
5.6	Tabela valores recolhidos segundo teste, trajeto número 1.	57
5.7	Tabela valores recolhidos segundo teste, trajeto número 2.	58

Lista de Figuras

1.1	Diagrama da ideia simplificada do projeto.	3
2.1	Representação Interligações da IoT [3].	6
2.2	Imagem do PT 600x.	11
2.3	Imagem do LT-168W 2G-3G Sinocastel.	12
3.1	ESP32.	18
3.2	Imagem do LM2596 DC-DC.	20
3.3	Imagem doo GPS neo6.	21
3.4	Imagem laser sensor.	22
4.1	Diagrama do dispositivo SMARTMOV	26
4.2	Esquema de ligações do dispositivo SMARTMOV	27
4.3	Board do dispositivo SMARTMOV.	28
4.4	Fotogradia do dispositivo SMARTMOV.	30
4.5	Fluxograma do código arduino.	32
4.6	Exemplo tabelas BD.	36
4.7	Ambiente de trabalho do MySQL Workbench.	38
4.8	Painel de controlo XAMPP com Apache e MySQL ativos.	41
5.1	Fotografia do teste a 89 cm de altura.	44
5.2	Fotografia do teste a 60 cm de altura.	45
5.3	Fotografia do dispositivo SMARTMOV no tabelier do carro no decorrer do primeiro teste.	47

5.4	Fotografia do dispositivo SMARTMOV no porta luvas do carro no decorrer do segundo teste.	48
5.5	Dados de localização recolhidos pelo dispositivo SMARTMOV	49
5.6	Percurso número 1 realizado nos testes ao dispositivo SMARTMOV.	50
5.7	Percurso número 2 realizado nos testes ao dispositivo SMARTMOV.	50
5.8	Site BatchGeo utilizado para fazer o pinpoint dos valores recolhidos.	53
5.9	Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no primeiro percurso com o dispositivo no tabelier	53
5.10	Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no segundo percurso com o dispositivo no tabelier	54
5.11	Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no primeiro percurso com o dispositivo no porta luvas	55
5.12	Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no segundo percurso com o dispositivo no porta luvas	55

Capítulo 1

Introdução

No presente documento é apresentado o trabalho desenvolvido na unidade curricular Dissertação, projeto de final de curso do Mestrado de Engenharia Industrial da Escola Superior de Tecnologia e de Gestão do Instituto Politécnico de Bragança.

Os transportes públicos numa cidade providenciam um meio de deslocamento de pessoas de um ponto a outro na área dessa cidade. A grande maioria das áreas urbanas de médio e grande porte possui algum tipo de transporte público urbano. Estes transportes são, geralmente, de responsabilidade municipal, embora o município possa conceder licenças, às vezes acompanhadas de subsídios, a companhias particulares. Os transportes públicos são um meio de transporte essencial nas cidades e por isso têm de ter um funcionamento eficiente e de fácil acesso às diversas pessoas, mas por vezes isso não acontece resultando num conjunto de problemas como: atrasos, rotas pouco eficientes e pouco contacto com o seu utilizador. Este projecto irá focar-se num dos meios de transportes públicos que todas as cidades têm, os autocarros e irá tentar resolver os problemas acima mencionados.

1.1 Enquadramento

Os transportes públicos são dos meios de transporte mais utilizados por todo o mundo e de acordo com a experiência pessoal e de alguns colegas na utilização dos transportes

públicos chegou-se à conclusão de que sofrem de alguns problemas, como atrasos, rotas pouco eficientes e pouco contacto com o seu utilizador, como referido anteriormente, que necessitam de solução. Para tentar resolver estes problemas optou-se por começar por desenvolver um dispositivo de localização baseado num módulo *GPS*, ao qual foi dado o nome de *Dispositivo SMARTMOV*. Assim, é possível saber onde se encontra o autocarro em tempo real a qualquer hora do dia. O objetivo é tentar identificar e resolver o problema de possíveis atrasos, identificando em tempo real, atrasos do autocarro. Um excelente parâmetro para completar a localização do autocarro seria a recolha do fluxo de cada paragem. O objetivo seria saber qual paragem que tem mais ou menos movimento de pessoas, sendo assim possível fazer uma previsão e comparação de fluxo entre paragens.

Assim, foram colocados no dispositivo dois sensores para saber quantas pessoas entram e saem do autocarro. Todos estes dados serão armazenados numa base de dados para depois serem processados, como dito anteriormente o fluxo será utilizado para fazer previsões e comparações de fluxo nas paragens e assim recomendar uma possível mudança daquela paragem ou até mesmo de uma rota. Isto resolveria o problema de rotas pouco eficientes.

Futuramente e para resolver o problema de pouco contacto com os utilizadores dos transportes públicos, será desenvolvida uma aplicação android ou webapp onde os utilizadores saberão onde o autocarro se encontra em tempo real, qual a paragem mais perto de si e qual o melhor autocarro para o seu destino. Toda esta ideia do projeto foi simplificada na Figura 1.1 para ajudar a uma melhor precessão de todo o projeto.

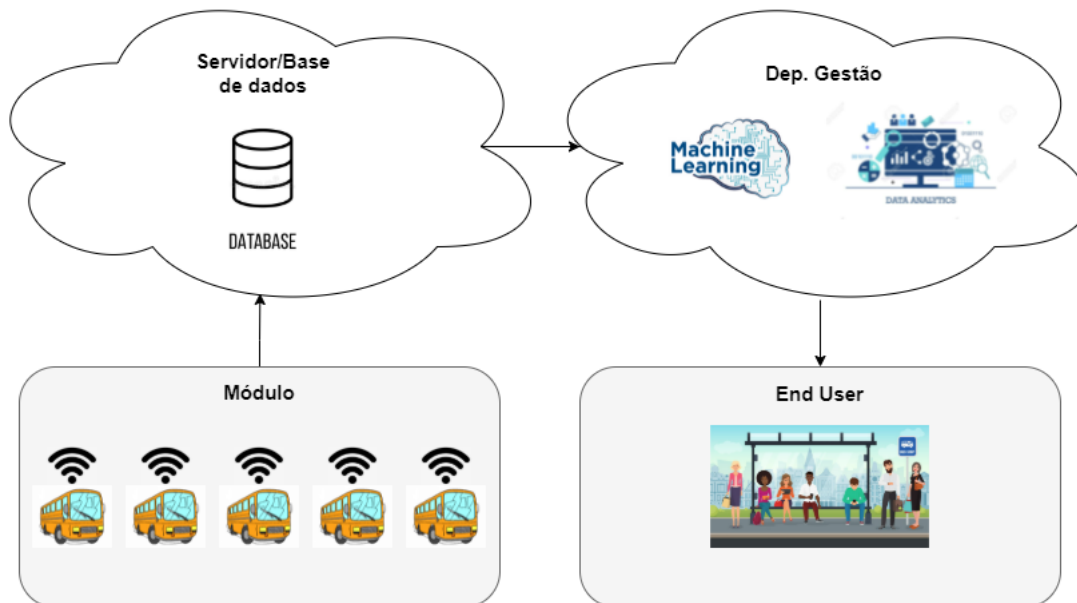


Figura 1.1: Diagrama da ideia simplificada do projeto.

1.2 Objetivos

Com a realização desta dissertação pretende-se resolver os problemas anteriormente descritos da seguinte forma:

- Construção de dispositivo baseado em GPS para localização do autocarro;
- Recolha do fluxo de cada paragem através de um sensor;
- Criação de uma Base de Dados para organização dos dados recolhidos e posteriormente análise de dados para atuação (ex: otimização de rotas existentes ou criação de novas).

1.3 Estrutura do Documento

O presente documento está organizado da seguinte forma.

No **Capítulo 1** é feita uma pequena descrição ao tema deste trabalho, sendo identificados os problemas gerais. É também apresentado os objetivos principais deste trabalho e a organização deste relatório.

No **Capítulo 2** encontra-se o *Estado da Arte* onde é descrita uma das bases em que este projeto se insere chamado, *IoT*. É também realizado um estudo de empresas em varias áreas da *IoT*, bem como a análise de dispositivos existentes no mercado.

No **Capítulo 3** é discutido qual o hardware utilizado para a realização deste projeto, bem como o porquê do mesmo ter sido escolhido. É também descrito a forma e os critérios utilizados para a criação da base de dados.

No **Capítulo 4** é discutido todo o processo de montagem e custo do dispositivo SMARTMOV e da base de dados, desde a criação de uma placa de circuito integrado, código Arduino e *PHP* à organização da base de dados.

No **Capítulo 5** apresentam-se todos os resultados obtidos dos testes realizados ao dispositivo SMARTMOV, como margem de erro do *GPS* e sensor laser, leitura e envio dos dados recolhidos para a base de dados.

As conclusões são apresentadas no **Capítulo 6**, onde se faz uma discussão dos objetivos cumpridos.

Capítulo 2

Estado da Arte

Depois de alguma pesquisa chegou-se à conclusão de que a solução apresentada nesta dissertação se inseria muito na *Internet of Things* (IoT), isto porque estamos a utilizar um método de troca de dados entre dispositivos, recolhidos por sensores e módulos *GPS*, deste modo foi efectuada uma pesquisa mais aprofundada sobre a *Internet of Things* e possíveis projetos dentro deste tema que poderiam ser úteis para o desenvolvimento desta dissertação [1].

2.1 Internet of Things (IoT)

A Internet das Coisas (*IoT*) descreve a rede de objetos físicos - “coisas” - que são incorporados a sensores, software e outras tecnologias com a finalidade de conectar e trocar dados com outros dispositivos e sistemas pela Internet. Esses dispositivos variam de objetos domésticos comuns a ferramentas industriais sofisticadas, na Figura 2.1 é possível observar uma representação das várias interligações.

Nos últimos anos, a *IoT* tornou-se uma das tecnologias mais importantes da atualidade. Agora que podemos conectar objetos do dia-a-dia - eletrodomésticos, carros, termostatos, monitores de bebés - à Internet por meio de dispositivos incorporados, a comunicação perfeita é possível entre pessoas, processos e coisas [2].

Devido à computação de baixo custo, cloud, big data, análises e tecnologias móveis,

coisas físicas podem compartilhar e coletar dados com o mínimo de intervenção humana. Neste mundo hiperconectado, os sistemas digitais podem registrar, monitorizar e ajustar cada interação entre as coisas conectadas.

A razão da *Iot* ser cada vez mais utilizada deve-se ao fato de:

- **Acesso a sensores de baixo custo e de baixa consumo:** Sensores acessíveis e confiáveis estão tornando a tecnologia *IoT* possível para mais fabricantes.
- **Conectividade:** Uma série de protocolos de rede para a Internet facilitou a conexão de sensores à nuvem e a outras “coisas” para uma transferência de dados eficiente.
- **Plataformas de computação em nuvem:** estas plataformas incluem servidores, base de dados, armazenamento, etc. Estas plataformas são flexíveis e podem ser dimensionadas de acordo com o pretendido por cada empresa.
- **Machine Learning:** Com o desenvolvimento do *Machine Learning* e o acesso a grandes quantidades de dados, permite que as empresas adquiram "conhecimento" com mais rapidez e facilidade. Isto permite expandir os limites da *IoT*.

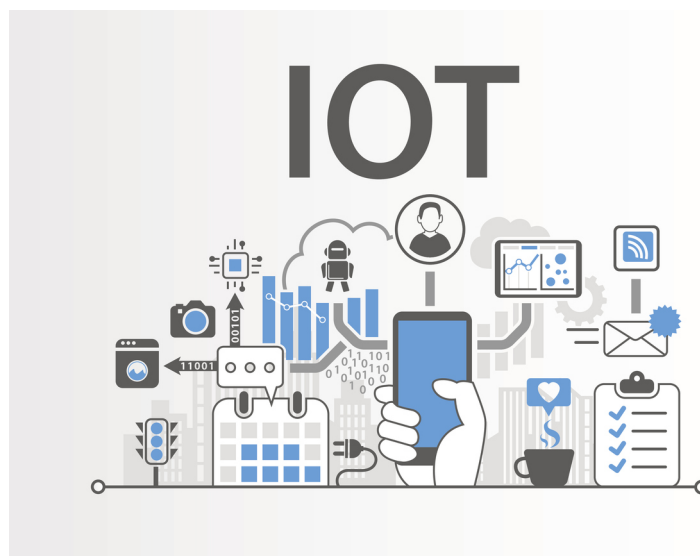


Figura 2.1: Representação Interligações da IoT [3].

2.2 IoT Industrial

A *IIoT* (Industrial Internet of Things) consiste em máquinas conectadas à internet e em plataformas de análises avançadas que processam os dados produzidos. Os dispositivos de *IIoT* variam de minúsculos sensores ambientais a complexos robôs industriais. A palavra “industrial” pode induzir a erro e pensar em armazéns, estaleiros e instalações industriais, mas as tecnologias de *IIoT* também podem fazer muito por uma vasta gama de setores, incluindo agricultura, assistência médica, serviços financeiros, cidades inteligentes, redes eléctricas inteligentes, etc.

2.3 Exemplos de Aplicações da IoT e IIoT

Como foi dito anteriormente a *IoT* e *IIoT* apresenta múltiplas aplicações desde área industrial à área da saúde, sendo que alguns exemplos das mesmas são: previsão de manutenção, automação, optimização, saúde, monitorização e rastreamento. Estes exemplos de aplicações serão descritos seguidamente [4].

Previsão de Manutenção

A monitorização de ativos industriais envolve medidas tomadas para melhorar a integridade do dispositivo e da máquina para assim obter uma maior produção. É um dos principais exemplos de *IIoT* na configuração industrial. A falta de manutenção adequada está entre as principais causas para a redução da vida útil das máquinas, sejam elas indústrias ou até mesmo veículos. A aplicação da *IoT* na manutenção envolve o uso de sensores para prever se uma falha é provável ou não. Com base em certos limites predefinidos, os sensores indicam até que ponto a manutenção pode ser necessária.

Automatização/Optimização

A maioria dos processos industriais pode ser automatizada remotamente sem a necessidade de estar perto das máquinas reais. Dispositivos equipados com sensores *IoT* detectam

sinais em redes sem fio e começam a trabalhar precisamente na tarefa que foi atribuída.

É possível gerir a energia consumida por dispositivos, especialmente na indústria de manufatura, onde grandes quantidades de energia são utilizadas. Quando usados apropriadamente, os dispositivos *IoT* podem ajudar a prever a utilização de energia dos dispositivos individuais e ajudar a reduzir a sobrerutilização, usando-os criteriosamente ou em modo de poupança de energia.

Um dos recursos mais desperdiçados é a água e com o *IoT* é possível, através de sensores, fazer uma leitura dos níveis de água seja em reservas ou construir um sistema de rega inteligente. É neste último que a *IoT* pode ajudar e muito na agricultura, em que um sensor pode determinar a quantidade de humidade no solo e as condições climáticas. Com base nesses parâmetros, este sensor determina com precisão a quantidade de água necessária para as plantações. Isto permite que os agricultores economizem água e cultivem de maneira mais saudável.

Saúde

Frigoríficos de fármacos têm de estar em conformidade regulamentar e segurança. Os frascos de vacinas e medicamentos podem frequentemente estragar-se se não forem mantidos nas temperaturas correctas. Ter sensores *IoT* instalados nestes frigoríficos pode permitir que sejam monitorizados remotamente e que sua temperatura seja alterada de acordo com a necessidade.

Monitorização e Rastreamento

A *IoT* pode também ser utilizada no rastreamento de frotas, desde transporte de mercadorias a transportes públicos. Através de um *GPS* e um conjunto de sensores, é possível saber a localização em tempo real de qualquer veículo, bem como a sua velocidade e possíveis problemas com o mesmo, como problemas no motor ou problemas electrónicos bem como aviso de necessidade de manutenção.

2.4 Projetos/Empresas Existentes

Dentro das diferentes áreas de aplicação da *IoT* existem já algumas empresas/projetos estabelecidos tanto na *IoT* como na *IIoT*.

2.4.1 Projetos na Área Industrial

Existem várias empresas que oferecem diversas soluções para facilitar o trabalho de manutenção e monitorização de todo o tipo de equipamentos numa fabrica industrial, sendo que algumas dessas soluções oferecidas serão seguidamente apresentadas.

AMPER TECHNOLOGIES

Localização: Chicago, Illinois [5].

Aplicação da IoT: a *Amper Technologies* desenvolveu um sistema operativo para fábricas que faz uso eficiente de sensores conectados à internet e análise de dados para aprimorar o processo de fabrico e assim aumentar a produtividade de fabrico.

Impacto na indústria: Os sensores registam uma variedade de dados importantes, incluindo uso de energia e tempo de inatividade, para que os proprietários e supervisores da fábrica possam planear melhor as linhas de produção, cortar custos e apontar áreas de crescimento.

AXZON

Localização: Austin, Texas[6].

Aplicação da IoT: a *Axzon* utiliza sensores que fornecem dados cruciais em tempo real durante o processo de fabrico automatizado, esta tecnologia de manutenção monitoriza a condição do equipamento da fábrica para ajudar a evitar quebras na produção e reparos demorados.

Impacto na indústria: manter as máquinas de qualquer fabrica sobre constantes monitorização é uma mais valia para a deteção de qualquer erro mínimo que pode provocar danos quer na maquina, quer na linha de produção.

2.4.2 Projetos na Área de Mobilidade

Como temos empresas dedicadas à indústria, temos também empresas dedicadas à mobilidade, área na qual se insere esta dissertação.

AIRBIQUITY

Localização: Seattle, Washington [7].

Aplicação da IoT: a *Airbiquity* é uma empresa de software e engenharia que fornece atualizações "over-the-air"(OTA) e serviços que permitem gerir dados para carros conectados.

Impacto na indústria: os utilizadores da *Airbiquity* podem optar por notificar a família ou amigos de forma automática e imediata em caso de acidente.

Zubie

Localização:Bloomington, Minnesota [8].

Aplicação da IoT: A *Zubie* oferece rastreamento de frota por GPS em tempo real enquanto monitoriza a saúde do veículo e o desempenho do motorista.

Impacto na indústria: Se um motorista tem o hábito de travar ou acelerar rapidamente, essa informação pode ser usada para evitar potenciais acidentes e aumentar a eficiência do combustível.

Dash

Localização:New York, New York.

Aplicação da IoT: A app gratuita do *Dash* para iOS e Android recebe dados de uma variedade de sensores que o mantêm informado sobre o estado atual do carro e necessidades de manutenção imediatas ou futuras.

Impacto na indústria: O aplicativo da empresa tem uma função de controlo parental que permite aos pais ou responsáveis receber alertas por e-mail se um jovem motorista exceder os limites estabelecidos ou tentar contorná-los.

2.5 Módulos Existentes

Para se analisar a viabilidade deste projeto no mercado, foi realizada uma pesquisa em relação a empresas existentes que oferecem módulos de rastreamento. Depois de realizada essa pesquisa, foram selecionados dois módulos que tentam oferecer uma solução de rastreamento de veículos, os mesmos serão seguidamente apresentados.

O primeiro da empresa Bofan, PT 600X © [9], oferece:

- Localização em tempo-real dos veículos através de GPS;
- Sistema de videovigilância;
- Tempo de chegada estimado do veículo;
- Poupança de combustível (monitorização de combustível gasto).

Preço (site não oficial): 139.67€



Figura 2.2: Imagem do PT 600x.

A segunda opção é da empresa Sinocastel, LT-168W 2G-3G © [10], oferece:

- Localização em tempo-real dos veículos através de GPS;
- Controlo remoto do motor do veículo (é possível desligar o motor do veículo através de um relé);
- Poupança de combustível;
- Monitorização de possíveis problemas no motor do veículo (isto é possível pois o módulo encontra-se conectado ao veículo através da interface OBD);

- Alerta se o módulo for removido.

Preço (site não oficial): 83.80€



Figura 2.3: Imagem do LT-168W 2G-3G Sinocastel.

Estas duas ofertas no mercado oferecem uma solução de rastreamento dos veículos e monitorização dos mesmos, focando-se exclusivamente no próprio veículo e na recolha da sua posição, não recolhendo nenhum dado relevante para melhorias de rotas ou tempos de chegada estimados em tempo real.

2.6 Sociedade de Transportes Colectivos do Porto

O STCP, é a empresa responsável pela rede de autocarros no concelho do Porto e também várias linhas em concelhos do Grande Porto. Como todas as empresas da área o objetivo da STCP é prestar um bom serviço de transporte dentro das cidades em alternativa ao veículo privado, assim quando se deparou com alguns dos problemas descritos anteriormente, foi necessário encontrar uma solução [11].

A solução encontrada pelos STCP foi entregar o projeto a uma empresa chamada OPT, SA, esta empresa dedica-se à gestão de transportes públicos e à sua optimização, sendo uma empresa com várias parcerias incluindo a Carris. Assim a OPT desenvolveu uma aplicação na qual é possível encontrar informações como localização das paragens e fazer o cálculo de uma rota desejada. Foram também instalados painéis informativos das diferentes rotas [12]. Infelizmente não é possível ter acesso à informação de como é recolhida a localização do autocarro em tempo real, assim foi feita uma pesquisa das críticas à aplicação na app store para tentar concluir se a funcionalidade do tempo real estaria a ser exacta, sendo que as maiores queixas são mesmo a falta de precisão ou mesmo o não funcionamento da localização dos autocarros em tempo real, o que pode revelar que isto é somente uma estimativa do tempo de espera e não envolve qualquer dispositivo de localização.

2.7 Proposta de Solução

Neste projeto procura-se encontrar uma solução para os problemas descritos anteriormente no Capítulo 1 como atrasos, rotas pouco eficientes e pouco contacto com o seu

utilizador, com uma solução mais barata mas igualmente fiável ou superior ao existente no mercado, começando por desenvolver um dispositivo, de nome dispositivo SMARTMOV, de localização baseado em GPS que irá realizar o rastreamento do veículo e que será acompanhado futuramente, por uma aplicação (app) que irá informar os utilizadores dos transportes públicos da localização do veículo através de um mapa e irá dar uma estimativa do seu tempo de chegada. Pretende-se também através dos dados recolhidos dar sugestões às empresas de transportes, como otimizações das rotas, diminuição nos custos de manutenção e de combustível dos seus veículos.

O dispositivo SMARTMOV irá incluir então um sistema de posicionamento global (GPS) que será responsável pela recolha da latitude e longitude do autocarro sabendo assim o seu posicionamento. Assim estes dois dados recolhidos, latitude e longitude, serão os dados utilizados para futuramente se realizar uma estimativa do tempo de chegada, sendo que os valores da latitude e longitude das paragens de autocarro serão valores fixos. Existem vários tipos de módulos GPS no mercado, como o foco do dispositivo SMARTMOV será saber a posição em tempo real do veículo, este terá de ser muito preciso o que pode levar a um preço alto do módulo GPS.

Para fazer previsões e recomendações de novas rotas, foi escolhido fazer a recolha do fluxo de passageiros que entra e sai em cada paragem, para isto será utilizado um sensor laser que irá detectar o movimento de pessoas, foram considerados ainda outros métodos de contagem de pessoas como contagem de sinais WIFI dos telemóveis que se tentariam ligar á rede do autocarro mas sendo que os transportes públicos são muito utilizados por pessoas mais idosas que não possuem um *smartphone* este método só seria viável se utilizado em adição aos sensores escolhidos.

O módulo GPS solicitará a localização do veículo e o sensor irá recolher o fluxo, depois dos dados serem recolhidos será necessário efetuar o seu envio para a base de dados. Assim será necessário um módulo com uma função que permita o envio dos dados, para isto foi escolhido o *ESP32* que tem uma função WIFI que será utilizada para conectar o dispositivo SMARTMOV à internet [13], este foi o meio de comunicação escolhido pois os autocarros possuem uma rede de *WIFI*, sendo assim fácil utilizar a Internet para o envio

de dados. Na base de dados os dados serão organizados e categorizados, ficando prontos a ser utilizados quer para uma futura App ou para o estudo de otimização de rotas.

Como melhoria do dispositivo SMARTMOV futuramente poderá ser adicionada a monitorização do veículo através do On-Board Diagnostics (OBD), dentro do protocolo OBD-II, assim o módulo terá de ser conectado à entrada Data Link Connector (DLC) do veículo. O On-Board Diagnostics (OBD) é um sistema computadorizado que faz o diagnóstico de todo o veículo. Inicialmente foi desenvolvido para controlar as emissões de gases dos automóveis e ajudar a combater a poluição do ar, mas é também possível aceder e recolher informação de praticamente todos os componentes de um automóvel.

Capítulo 3

Implementação

Neste projeto procura-se encontrar uma solução para os problemas descritos anteriormente no Capítulo 1, começando por desenvolver um módulo de localização por GPS, que irá enviar para uma base de dados a posição do autocarro, bem como a data/hora e o número de pessoas que entram/saem em cada paragem (fluxo). O fluxo de passageiros será recolhido através de um Sensor Laser. Os dados recolhidos serão enviados para uma base de dados, para mais tarde serem utilizados para a previsão do fluxo de passageiros, recomendação de novas rotas e monitorização de atrasos dos autocarros.

3.1 Hardware

Todo o hardware utilizado para a construção do módulo foi seleccionado tendo em vista o seu melhor funcionamento, mas também dependendo da relação qualidade/preço, eficiência e dimensões. Pois um dos objectivos deste módulo é ser de baixo custo mas preciso e sem avarias. Terá também de ter um consumo de energia o mais baixo possível devido à fonte de alimentação, que será o autocarro ou uma bateria. O módulo terá de ser o mais compacto possível para ser colocado em qualquer espaço, assim todo o hardware teria de ter dimensões mínimas.

3.1.1 ESP32

O *ESP32* será o "cérebro" de todo o dispositivo, é ele que irá fazer com que tudo trabalhe em conjunto para toda a recolha de dados pretendido. Será também o *ESP32* que irá realizar o envio de dados para a base de dados através da sua função WIFI. Todo o restante hardware terá de ter uma alimentação de 3.3 V pois o *ESP32* é alimentado a 5 V, mas só tem de saída 3.3 V.

Design O *ESP32* é capaz de funcionar de forma confiável em ambientes industriais, com uma temperatura de operação variando de -40°C a $+125^{\circ}\text{C}$.

Baixo consumo de energia Projetado para dispositivos móveis e IoT, o *ESP32* atinge um consumo de energia, com pelo menos um microcontrolador ligado, que pode chegar até 2 mA no modo econômico. O *ESP32* também inclui recursos de última geração, como "fine-grained clock gating", vários modos de energia e escalonamento de energia dinâmico.

Hybrid Wi-Fi e Bluetooth Chip O *ESP32* pode funcionar como um sistema autónomo completo ou como um dispositivo escravo para um MCU host, reduzindo a sobrecarga do *stack* de comunicação no processador da aplicação principal. O *ESP32* pode ser utilizado para fazer interface com outros sistemas para fornecer Wi-Fi e Bluetooth por meio das suas interfaces SPI / SDIO ou I2C / UART [14].

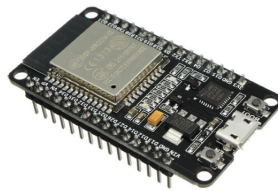


Figura 3.1: ESP32.

3.1.2 LM2596S DC-DC

O módulo terá de ser alimentado a uma tensão de 5 V assim foi necessário implementar um conversor DC-DC para que converta qualquer alimentação entre os 3.2 V - 35 V em 5 V. Assim foi escolhido o *LM2596S* que é um conversor DC-DC passo baixo com faixas de conversão de 3,2-35 V a 1,25-32 V e que trabalha com corrente 3 A. Este é um conversor de pequenas dimensões e tem baixas perdas na conversão, tendo uma eficiência de até 92% (valor dado pelo fabricante). Fornece a possibilidade de permitir ajustar a tensão, assim se necessário uma maior alimentação não é necessário trocar este componente. Existem no mercado outros conversores adequados para este projeto como por exemplo: o *Conversor DC-DC Step-down XL4005*, mas sendo o *Conversor LM2596S* dos mais utilizados no mercado existindo já no nosso stock foi decidido utilizá-lo.

Características:

- Tensão de saída: 1,25 V - 32 V DC (ajustável)
- Tensão de entrada: 3.2 V - 35 V DC
- Corrente de saída Max: 3 A (com dissipador de calor)
- Ondulação em tensão de saída: 30 mV
- Frequência: 150 kHz
- Eficiência de Conversão Max: 92
- Medidas: 43 x 21 x 14 mm

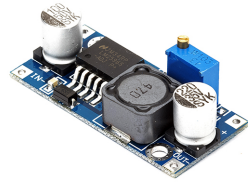


Figura 3.2: Imagem do LM2596 DC-DC.

3.1.3 GPS GY-NEO6MV2

O módulo GPS GY-NEO6MV2 oferece uma solução de baixo custo e baixo consumo, tendo uma precisão de medida do posicionamento de 5m o que em relação ao preço/qualidade é muito boa. Tem dimensões pequenas, e consegue funcionar com uma alimentação de apenas 2.7 V o que é ainda mais baixo do que a saída do *ESP32* de 3.3 V. O módulo GPS é dos componentes mais importantes deste projeto existindo várias ofertas no mercado, o escolhido já estava em stock e poderá mais tarde ser dado o upgrade para um GPS mais caro e mais preciso se necessário, como por exemplo o *MÓDULO GPS L76X* ou *Grove - GPS* que oferecem maior precisão e baixos consumos, mas com a desvantagem de ser muito mais caros.

Especificações:

- Alimentação: 2.7 V a 5 V DC
- Bateria para backup de dados
- Nível de sinal compatível 3.3 V e 5 V
- Taxa de comunicação padrão: 9600 bps
- Precisão: 5 m
- Dimensões: 35 x 25 x 25 mm

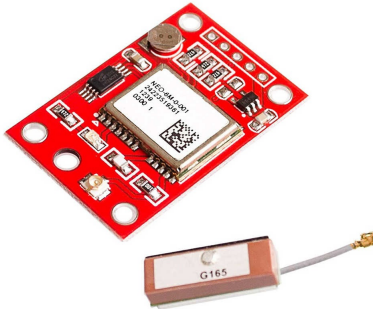


Figura 3.3: Imagem doo GPS neo6.

3.1.4 Sensor Laser - Referência :SEN05037

Para recolher os dados do fluxo foi escolhido o sensor laser de referência :SEN05037, devido ao seu baixo consumo e tendo um alcance de medida suficiente para o problema em questão. A dimensão deste sensor não é tão importante como a de outro hardware pois este sensor ficará fora do módulo, não tendo impacto nas dimensões do mesmo. A sua alimentação por outro lado terá de ser no mínimo 3.3 V. O sensor laser escolhido é já um dos melhores no mercado sendo que o seu upgrade não será necessário.

Especificações:

- Alcance: 0.8 m (típico) - 1.5 m(max);
- Power: 2.5 V 5.0 V;
- Dimensão: 47.7 mm * 17.9 mm



Figura 3.4: Imagem laser sensor.

3.2 Base de Dados

Uma base de dados é um simples repositório de informação relacionado com determinado assunto ou finalidade, ou seja, é uma colecção de dados ou itens de informação estruturados de determinada maneira que permite a sua consulta, actualização e outros tipos de operações processados por meios informáticos.

Assim a base de dados será onde todos os dados recolhidos serão guardados e organizados, logo a sua criação tem de ser bem estruturada, assim para a criação da base de dados foram seguidos os seguintes passos:

1. Planeamento

Levantamento das necessidades, organizar e planear;

2. Recolha de requisitos

Elaboração dos objectivos que o projecto visa atingir;

3. Desenho das tabelas

Desenho do formato das tabelas pretendidas;

4. Interligação

Desenho das interligações entre as varias tabelas e suas colunas;

5. Construção

Construção da base de dados utilizando um programa próprio neste caso o MySQL Workbench;

6. Implementação

Colocar a base de dados num servidor;

7. Manutenção

Começar o envio de dados para a base de dados e resolver quaisquer situações de anomalias ou erros, quer a nível do código arduino quer ao nível da base de dados.

Capítulo 4

Desenvolvimento Dispositivo

SMARTMOV

Depois de toda a pesquisa e escolha de hardware e software a utilizar, chegou-se ao desenvolvimento de todo o projeto em si, tema que se vai discutir neste capítulo.

4.1 Arquitetura do sistema

Chegou-se a uma organização do dispositivo de acordo como se pode ver no esquema da Figura 4.1. O *ESP32* será o componente que lê os dados obtidos tanto pelo GPS como pelo Sensor laser, sendo que é nele que se encontra o código Arduino que comanda todo o processo de recolha dos dados e parte do envio para a base de dados.

O GPS será o responsável da localização do dispositivo identificando a latitude e longitude, bem como a recolha da data e hora. A variável fluxo será da responsabilidade do Sensor Laser, que ao ser interrompido, neste caso por uma pessoa, irá fazer uma soma de mais um ao valor anterior, tendo assim o valor total de fluxo, esperando somente depois pela leitura do *ESP32*.

Para o envio de dados o *ESP32* utilizará a sua função WIFI para fazer a ligação a um ponto de ligação à internet previamente guardado na memória do *ESP32*, tendo assim acesso ao servidor onde se encontra a base de dados. Depois de estar ligado ao servidor,

será requerido um HTTP post a cada 20 segundos para o servidor, de seguida através de um phpscript serão publicados todos os dados recolhidos na tabela pretendida da base de dados, onde os dados ficarão prontos a ser consultados e utilizados.

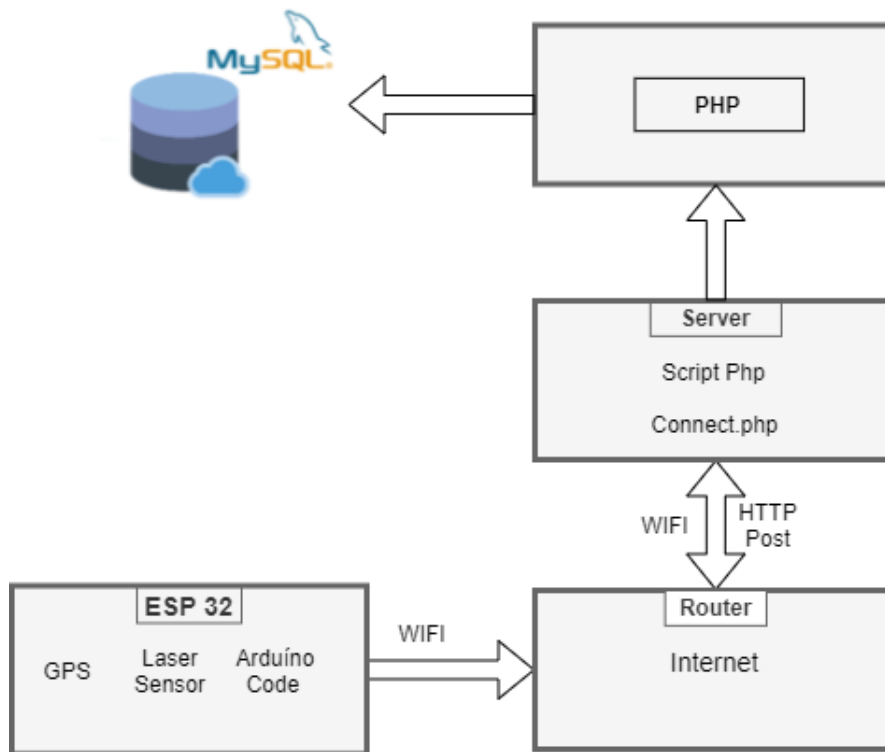


Figura 4.1: Diagrama do dispositivo SMARTMOV

4.2 Esquema de Ligações

O *ESP32* só pode ser alimentado com uma tensão de 5V, assim, e como pode ser verificado na Figura 4.2, foi necessário colocar um conversor DC-DC no circuito para este passar qualquer tensão de alimentação superior à capacidade do *ESP32* para 5 V, sendo que com

o conversor LM 2596S escolhido a tensão de alimentação pode ser alterada para outro valor, sempre superior ao de entrada, se necessário. Como segurança extra foi adicionado um fusível de 1.5A entre o conversor e o *ESP32*, caso exista alguma anomalia na conversão DC-DC e passem mais de 5 V o fusível irá queimar cortando assim a passagem de qualquer tensão e impedindo que o *ESP32* seja colocado em perigo. Foram adicionadas duas saídas extra, para além das duas necessárias uma para cada sensor, para possíveis adições futuras ao circuito como por exemplo um maior número de sensores para a recolha do fluxo.

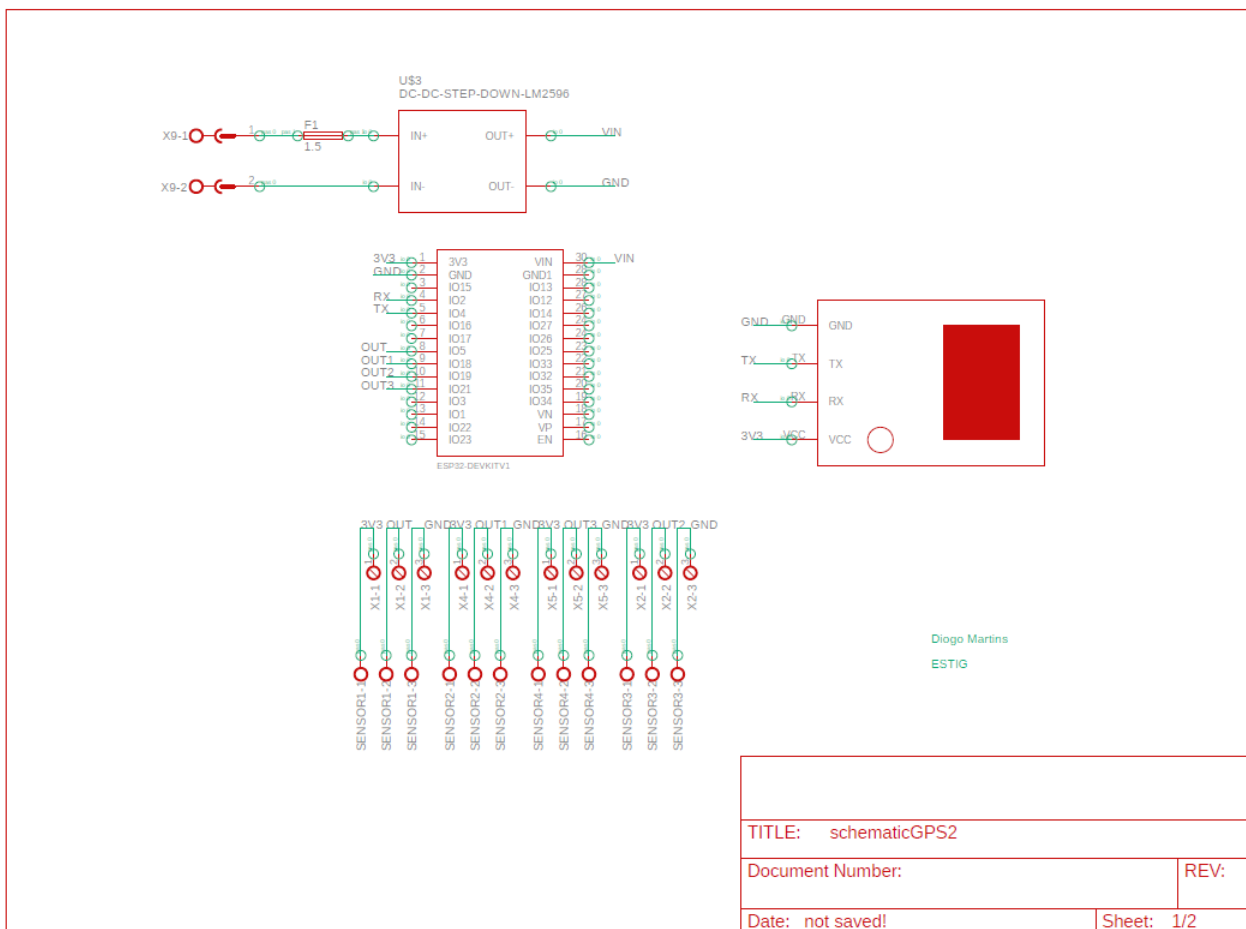


Figura 4.2: Esquema de ligações do dispositivo SMARTMOV

4.3 Placa de Circuito Impresso Desenvolvida

Com o objetivo principal de ser uma placa compacta, foi desenhado o circuito impresso que se pode observar na Figura 3.1 que será futuramente colocada dentro de uma caixa, que servirá para sua proteção tanto do ambiente, como de possíveis alterações ilícitas ao dispositivo SMARTMOV. Esta placa será toda ela alimentada a 12v sendo que o conversor DC-DC integrado na mesma fará a conversão para os 5 V necessários para o seu funcionamento.

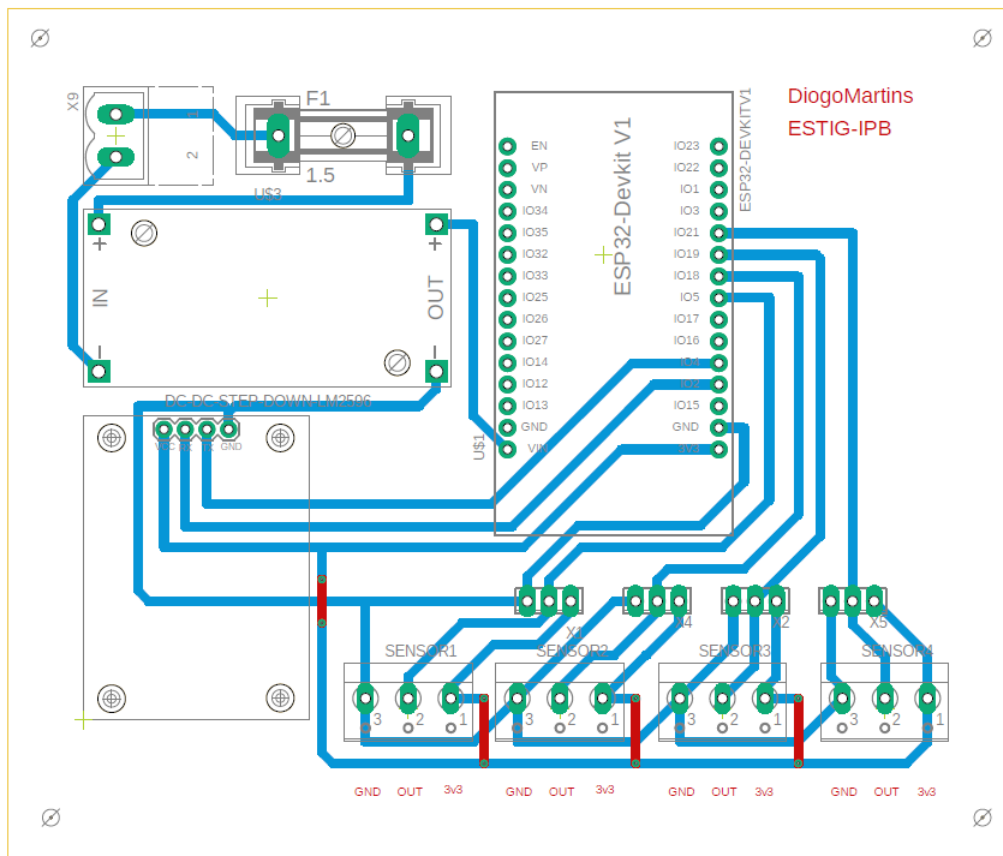


Figura 4.3: Board do dispositivo SMARTMOV.

O circuito impresso é de duas camadas, sendo que as ligações que se podem observar na Figura 4.3 são ligação nas camadas "Bottom" e "Top", as ligações representadas a azul são feitas na camada inferior e a vermelho são as da camada superior.

Um dos objetivos na realização deste projeto é realizar a montagem de todo o módulo por um preço baixo e competitivo, não perdendo qualidade na recolha de dados tanto do GPS, como do sensor laser (fluxo de pessoas). Como se pode ver na Tabela 4.1 encontram-se o preço de cada componente utilizado na montagem do módulo, incluindo o preço de fabrico da placa da Secção 4.3, todos estes preços estão sujeitos a alterações tanto em termos de variações nos valores de mercado, como na escolha do fornecedor ou quantidade de componentes comprados. No preço total do dispositivo SMARTMOV não está contabilizado o custo de montagem de todos os componentes na placa impressa.

Material	Custo
Placa	3.76 €
ESP32	10.00 €
Sensor Laser	11.00 €
Conversor DC-DC	4.00 €
GPS	9.00 €
Fusivel	0.30 €

Tabela 4.1: Tabela de custo dos componentes

Assim e pelos preços de mercado dos componentes na realização deste projeto, o preço total de montagem do módulo será de *38.06 €*, sendo este o custo para a produção de um único dispositivo SMARTMOV. Se compararmos com os preços dos dispositivos da Sinocastel e da Bofan mencionados no capítulo 2 o preço do dispositivo SMARTMOV fica muito abaixo do preço de 83.80€ e 139.67€ da Sinocastel e da Bofan, respetivamente, sendo que estes valores são o preço de venda dos seus dispositivos e não o preço de fabrico que será por suposição mais baixo. Mas em caso de venda do dispositivo SMARTMOV existe uma grande margem para um possível preço de retalho.

Depois da compra de todas as peças e da placa construída, foi realizada a sua montagem o que resultou no dispositivo SMARTMOV que se pode observar na Figura 4.4

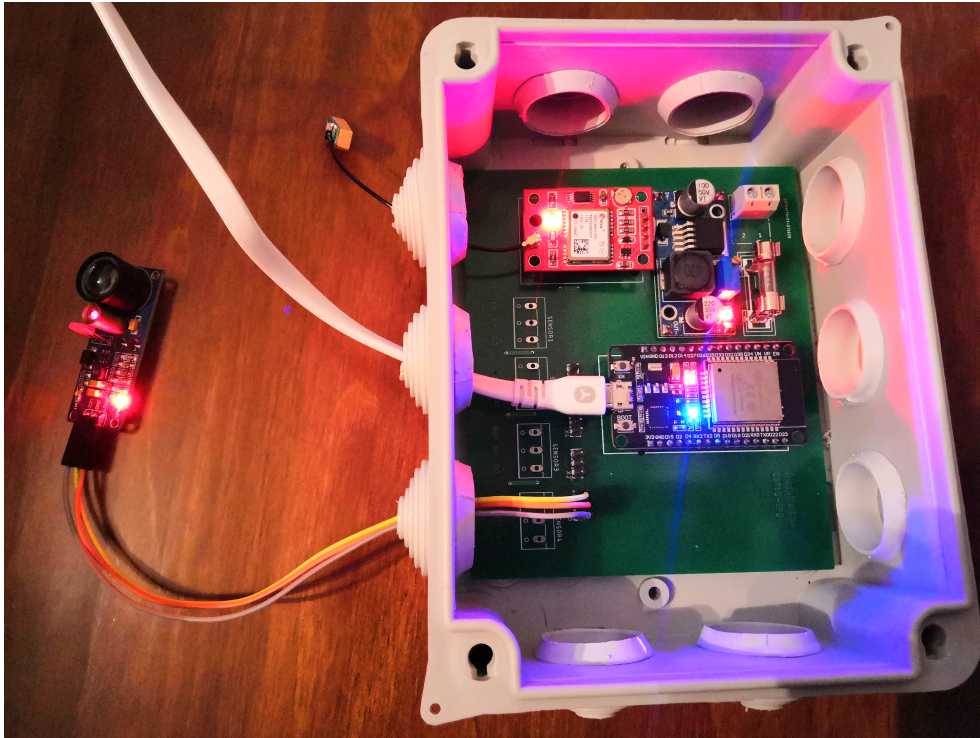


Figura 4.4: Fotografia do dispositivo SMARTMOV.

4.4 Programação

Para o funcionamento de todo o módulo foi necessário criar um código Arduino para ser possível o seu total funcionamento, sendo aqui que se gere os tempos em que é necessário fazer uma leitura do *GPS* e sensor ou fazer o envio dos dados dessa leitura. Relativamente ao envio de dados foi necessário criar um código em *PHP* que juntamente com uma secção no código Arduino irá conectar á base de dados e fazer o envio dos dados, organizando pelas colunas da tabela, que recebe os dados em tempo real, cada dado onde ele pertence: latitude, longitude, data/hora e fluxo, ficando assim os dados prontos a serem utilizados.

Este projeto ficou assim dividido em dois códigos, o código Arduino que será executado no *ESP32* e o código *PHP* que será executado no servidor quando o *ESP32* o requisitar.

Começando pelo código Arduino, primeiramente a ligação ao WIFI seria feita através do SSID e password da rede WIFI, escritas no código raiz. Isto levantou um problema, pois se fosse necessário alterar a rede WIFI teria de ser feita uma alteração no código do

ESP32 diretamente. Depois de alguma pesquisa foi encontrada uma biblioteca Arduino o "AutoConnect". Com esta biblioteca implementada no código não é necessário codificar previamente o SSID e a senha no Sketch para se conectar entre *ESP32* e a rede WIFI. O que isto faz é conectar o *ESP32* a uma rede conhecida guardada posteriormente na flash do *ESP32*, se não for encontrada uma rede conhecida o *ESP32* irá criar uma rede WIFI própria, que pode ser acessada através de um smartphone, depois de conectada irá abrir um interface na web onde se pode inserir o SSID e senha da rede WIFI pretendida. Esta nova rede será guardada na flash do *ESP32*, não sendo necessário inserir novamente as credenciais no futuro.

Depois de feita a ligação à rede, será iniciado o contador que irá dar os tempos para a execução do restante código.

A cada 11 segundos o *ESP32* irá guardar os dados recolhidos pelo GPS: latitude, longitude, data/hora e guardar para o próximo passo que irá acontecer de 20 em 20 segundos. Assim a cada 20 segundos e com o auxílio do código PHP o *ESP32* irá tentar a conexão ao host para fazer o envio dos dados recolhidos para a base de dados.

O sensor laser que está responsável pela contagem do fluxo, está considerado no código como interruptor pois se não fosse assim e no caso prático, se uma pessoa para-se em frente ao sensor ele estaria sempre a contar essa pessoa o que estaria incorreto pois continuava a ser a mesma pessoa, com esta solução mesmo que parem em frente ao sensor ele continuará somente a contar uma pessoa.

O código PHP é o que permite a conexão à base de dados sendo nele que se encontram as credenciais da base de dados para a qual os dados serão enviados, sendo o código PHP que envia uma *Query* para a base de dados para que se distribuam os vários dados para a tabela correta e para a coluna correspondente a cada um.

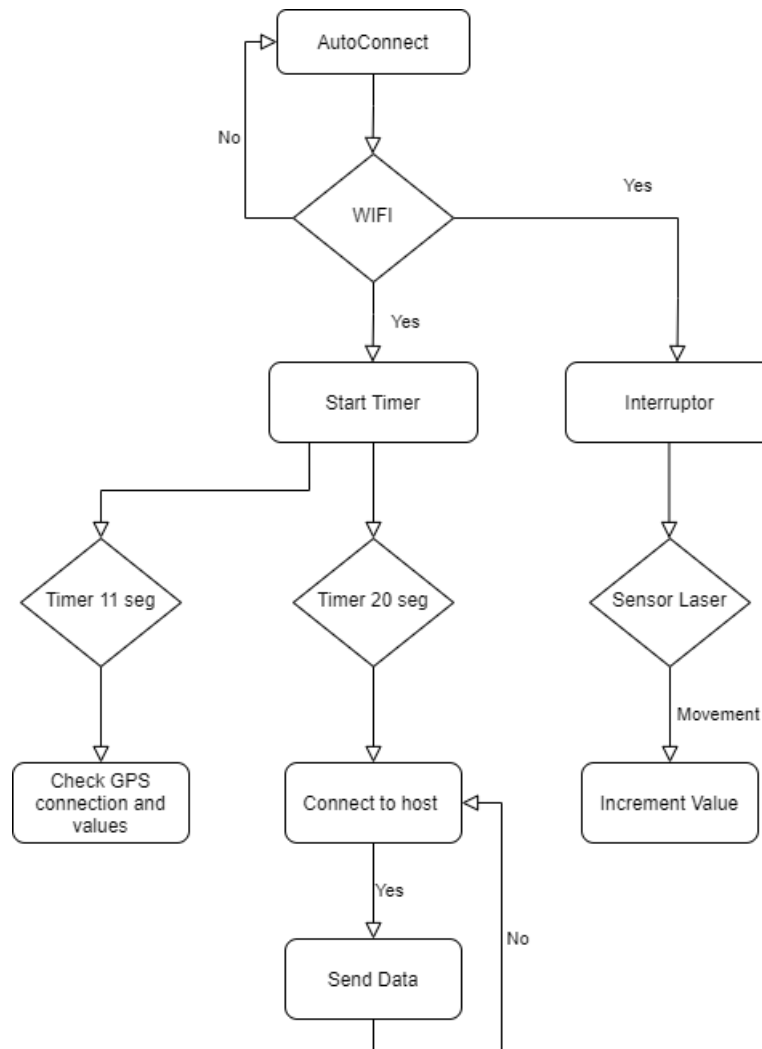


Figura 4.5: Fluxograma do código arduino.

4.5 Base de Dados

Os primeiros sistemas de informação eram baseados em conjuntos de ficheiros que guardavam conjuntos de registos, este sistema tinha limitações sendo que as principais eram:

- Os dados ficavam separados e isolados o que tornava a tarefa de obter dados de mais que um ficheiro em simultâneo, muito difícil;
- A mesma informação existe em mais que um ficheiro tornando complicado os processos de manutenção o que por norma leva a problemas de integridade dos dados.

- Os programas que acediam aos dados eram dependentes do formato dos ficheiros, se os formatos mudassem os programas teriam de ser também alterados.
- Ficheiros muitas vezes incompatíveis uns com os outros.
- Dificuldade em representar os dados na perspectiva do utilizador.

Todos estes problemas ou limitações levaram a uma procura de soluções para uma utilização mais sustentável e facilitada das base de dados, o que levou aos seguintes resultados:

- Os dados são todos guardados na mesma base de dados. Quando são necessários dados de entidades diferentes, o programa apenas especifica como é que os dados são combinados e é o gestor da base de dados que efectua todas as operações para dar a resposta.
- Os dados são guardados apenas num sítio assim a duplicação de dados está limitada e portanto os problemas de integridade são também eles limitados.
- Os programas não acedem directamente aos dados, apenas fazem pedidos ao *Data Base Management System* (DBMS) e é este que fornece os dados, assim os programas ficam menos dependentes dos formatos dos ficheiros.
- É relativamente fácil representar a estrutura de dados

4.5.1 Base de dados Relacional

Existem vários modos de armazenamento de dados em base de dados, um deles é a base de dados relacional. Com uma base de dados relacional, os dados serão divididos em varias tabelas, sem agrupar tudo numa grande unidade de armazenamento e onde é possível criar relações entre tabelas. Sem este modelo de base de dados relacional a construção de qualquer tipo de aplicação que iria utilizar os dados armazenados seria muito mais complexa e ineficiente. O modelo relacional simples, mas eficiente, é utilizado por organizações de todos os tipos e tamanhos para uma ampla variedade de necessidades de

armazenamento de dados. As bases de dados relacionais são usadas para organização de inventários, processar transações de comércio eletrônico, gerir grandes quantidades de informações essenciais sobre o cliente e muito mais [15]. Uma base de dados relacional pode ser considerada para qualquer situação na qual existe necessidade dos vários dados se relacionem entre si e devem ser geridos de forma segura e consistente, com base em regras. Com o desenvolvimento do modelo relacional surgiu a *Linguagem de Consulta Estruturada (SQL)* a linguagem de base de dados relacional simples e fácil de utilizar. Esta linguagem é a base para as linguagens mais utilizadas neste momento, MySQL, PostgreSQL e Microsoft SQL Server.

4.5.2 MySQL

MySQL é um sistema de gestão de base de dados relacional open-source. Como acontece com outras bases de dados relacionais, o *MySQL* armazena dados em tabelas compostas por linhas e colunas, onde os utilizadores podem definir, manipular, controlar e consultar dados [16].

O *MySQL* é conhecido por ser de fácil utilização, mas isso não lhe retira funcionalidades, prova disso é que é utilizado por empresas que trabalham com grandes volumes de dados, tais como, NASA, Bradesco, HP, Sony entre outras grandes empresas de renome. A sua interface extremamente simples e compatibilidade com grande parte dos sistemas operativos, são duas das principais características para o *MySQL* ser tão utilizado atualmente.

Compatibilidade

MySQL é compatível com a maioria dos sistemas operativos existentes atualmente no mercado [17]. Isto deve-se ao fato de ser desenvolvido em C e C++. Os sistemas operativos compatíveis com *MySQL* são:

- Windows
- Linux (Fedora, Core, Debian, SuSE e RedHat)

- Unix (Solaris, HP-UX, AIX, SCO)
- FreeBSD
- Mac OS X Server

Características

- Suporta diferentes plataformas, interfaces e compiladores, tendo também drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Visual Basic, Python, Perl, PHP, ASP e Ruby
- O *MySQL* possibilita diversos tipos de tabelas para o armazenamento dos dados, enquanto um tipo tem como prioridade a velocidade, outro pode dar prioridade ao volume de dados, entre outras características.
- O Cliente conecta ao MySQL através de conexões TCP/IP
- Possui um sistema de senhas encriptadas seguro
- Capacidade para manipular bancos com até 50 milhões de registos

4.5.3 Criação da base de dados

Depois de tudo descrito na secção 4.5 optou-se então por utilizar uma Base de dados relacional com base em *MySQL*, assim começou-se por fazer um esquema com as tabelas pretendidas e todas as ligações entre elas como se pode ver na Figura 4.6.

O objetivo destas tabelas é armazenar o máximo de dados possível, sejam eles dados recolhidos pelo dispositivo em tempo real ou dados inseridos manualmente.

Tabela *RTdata*

A tabela *RTdata* será a tabela que vai receber os dados em tempo real enviados pelo dispositivo SMARTMOV, ficando aqui armazenados e divididos por colunas os seguintes dados:

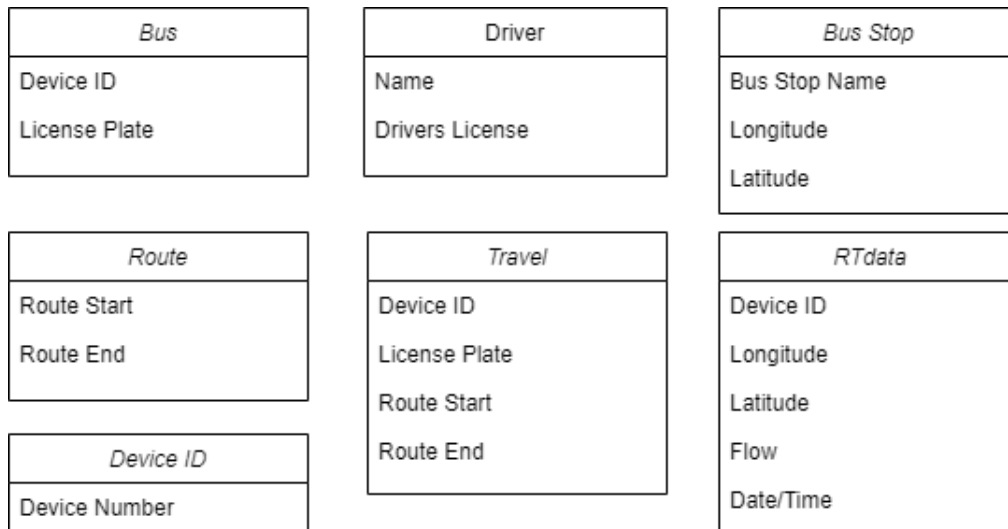


Figura 4.6: Exemplo tabelas BD.

- **Device ID:** O dispositivo que está a enviar os dados, assim saberemos o autocarro ao qual pertencem os dados recebidos;
- **Latitude e Longitude:** para saber a localização do autocarro;
- **Flow:** Número de pessoas que entra e sai em cada paragem;
- **Date/Time:** Data e hora no momento da recolha dos dados.

Tabela Device ID e Tabela Bus

A tabela *Device ID* servirá somente para atribuir a cada dispositivo um número e armazená-lo na base de dados, esta identificação servirá para associar cada dispositivo com o seu respectivo autocarro, isso será feito na tabela "Bus" onde iremos ter a matrícula de cada autocarro com ligação à tabela *DeviceID*, sendo assim possível dizer que, por exemplo, o dispositivo SMARTMOV com *DeviceID* número 1 se encontra no autocarro com matrícula "12-AB-34".

Tabela Route e Tabela Travel

A Tabela *Route* terá duas colunas sendo uma a *Route Start* que indica onde a rota começa e a *Route End* que será onde acaba, esta tabela é necessária para uma fase mais adiantada no projeto que não será explorada no presente trabalho, mas ficando já inserida na base de dados.

Esta tabela será interligada com a tabela *Travel* que será composta somente com dados de outras tabelas sendo estes:

- Dados da tabela *BUS* sendo estes o número do dispositivo SMARTMOV e o número da matrícula do veículo associado à mesma, *License Plate*;
- Dados da tabela *Route* que serão o início e o fim da rota do autocarro.

Com esta tabela é possível associar cada rota a um autocarro e a um dispositivo SMARTMOV, sendo possível identificar que autocarro e que dispositivo está a fazer uma certa rota.

Tabela Bus Stop

Na tabela "Bus Stop" serão armazenados os dados de todas as paragens existentes ficando dividida em três colunas:

- Na coluna *Bus Stop Name* será inserido o nome de cada paragem.
- As colunas *Longitude* e *Latitude* serão armazenadas as coordenadas das paragens

Com isto é possível associar cada paragem a uma localização fixa, esta tabela é importante pois será necessário os seus valores para se conseguir associar o valor correto do fluxo de pessoas à paragem certa. Isto será possível fazendo uma curva de distância entre as coordenadas da paragem e as coordenadas do autocarro, sendo que quando essa atingir o mínimo é retirado o valor do fluxo naquele momento, ficando assim esse valor na sua paragem correspondente.

Tabela Driver

Esta tabela será composta por duas colunas as quais são:

- Coluna *Name* que será o nome completo de um determinado conduto;
- Coluna *Drivers License* que ficará armazenado o numero da carta de condução do condutor.

Estes dados poderiam mais tarde ser utilizados para associar cada condutor a um autocarro e recolher informações de atraso, isto só será possível com autorização dos mesmos.

4.6 MySQL Workbench

O MySQL fornece uma ferramenta que permite gerir uma base de dados, chamada MySQL Workbench. Esta foi a ferramenta utilizada para criar a base de dados com as tabelas descritas na Secção 4.5.3.

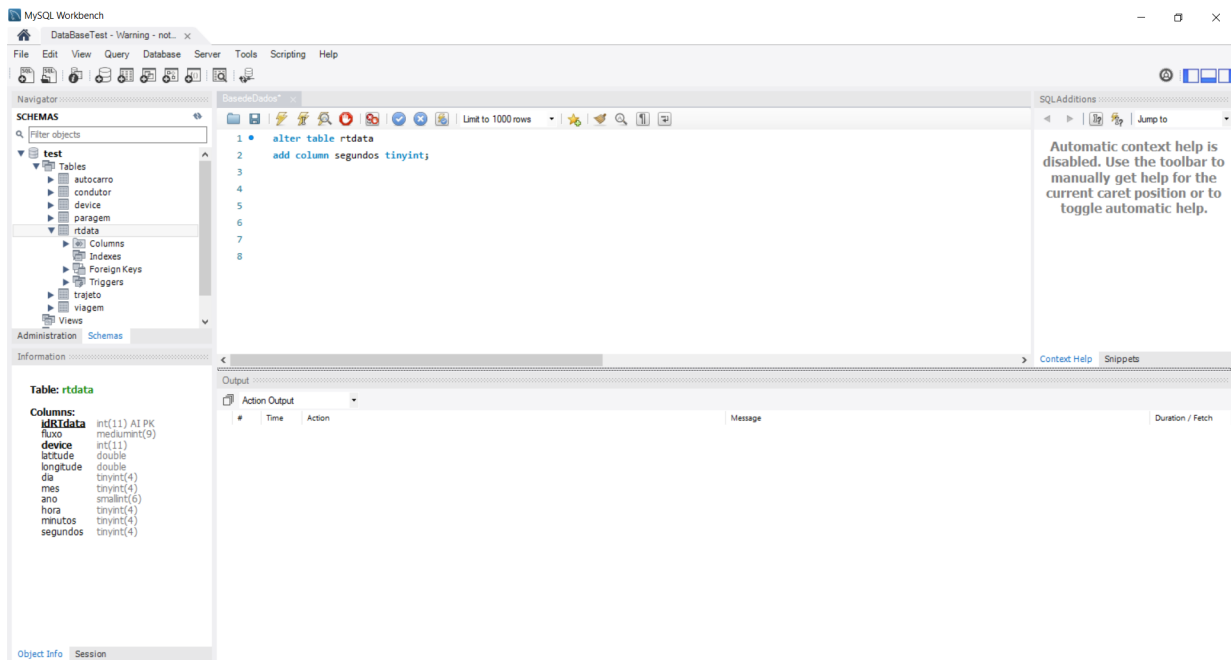


Figura 4.7: Ambiente de trabalho do MySQL Workbench.

A criação de uma base de dados ou tabela no MySQL Workbench não é de dificuldade muito elevada pois os comandos a dar são bastante intuitivos na maioria das vezes, como se pode ver no comando na Figura 4.7 em que o pretendido é alterar a tabela *Rtdata* adicionando uma nova coluna, assim basta escrever:

```
alter table rtdata  
add column segundos tinyint;
```

Assim isto vai alterar a tabela *Rtdata* e adicionar uma coluna segundos [18].

4.7 Servidor Local

Para realizar os testes de envio de dados para a base de dados relacional criada com base em *MySql* era necessário criar um servidor local, isto é, utilizar o próprio computador como se de um servidor remoto se tratasse. Isto levantou o problema de como seria feito e que programas utilizar. Depois de uma pesquisa sobre servidores locais, chegou-se à conclusão que a discussão de que programa a utilizar seria entre *XAMPP* e *WAMP*.

4.7.1 XAMPP e WAMP

XAMPP

O *XAMPP* é uma distribuição Apache pequena e leve que contém as tecnologias de desenvolvimento web mais comuns num único pacote gratuito. Este pacote que o *XAMPP* oferece é o ideal para o desenvolvendo e teste de aplicações em PHP e MySQL [19].

O próprio nome *XAMPP* provem da abreviação dos constituintes do seu pacote:

X – representa a compatibilidade com várias plataformas, como Windows e Linux.

A – Apache HTTP Server, ou mais conhecido somente por Apache.

M – MySQL ou MariaDB, provavelmente duas das base de dados mais utilizadas, pois são as de mais fácil utilização. Sendo a MariaDB uma variação do MySQL.

P – PHP, uma das linguagens de script mais populares.

P – Perl, é uma linguagem de Alto-Nível.

WAMP

Muito parecido com o *XAMPP*, o *WAMP* também é um pacote de servidor de desenvolvimento local que oferece a oportunidade de testar sites e bases de dados localmente.

Como acontece com o *XAMPP*, o próprio nome *WAMP* vem da primeira letra do pacote oferecido [20]:

W – neste caso significa somente o sistema operativo Windows, pois o *WAMP* não é compatível com qualquer outro sistema operativo.

A – Apache HTTP Server.

M – MySQL ou MariaDB, como no anteriormente no *XAMPP*.

P – PHP, uma das linguagens de script mais populares.

O *WAMP* não oferece Pearl por isso a falta do ultimo *P*.

Ambos os produtos são projetados para servir como pacotes que ajudam com o ambiente inicial para configurar um servidor local. Se tudo isto fosse configurado manualmente, teria que se baixar e instalar individualmente os vários componentes necessários. Com *WAMP* ou *XAMPP*, o Apache, MySQL, PHP e alguns outros componentes e serviços personalizados são instalados de uma só vez.

Assim tanto o *XAMPP* como o *WAMP* seriam uma boa escolha para este projeto, mas optou-se por utilizar o *XAMPP*, pois tem uma maior facilidade de instalação e de utilização, bem como um dado muito positivo de se necessário ser utilizado em qualquer sistema operativo no futuro.

4.7.2 Configuração XAMPP

Instalar e configurar o *XAMPP* é fácil e intuitivo. Depois de executar a configuração, o instalador pergunta quais os componentes do pacotes que se pretende instalar, esta é a única parte na instalação do *XAMPP* que é necessário escolher alguma coisa, o resto é feito automaticamente.

Depois de instalado e configurado é só colocar o código PHP, descrito na secção 4.4, na pasta *htdocs* que se encontra na pasta de instalação do *XAMPP*. Depois de tudo feito é só iniciar o painel de controlo do *XAMPP*.

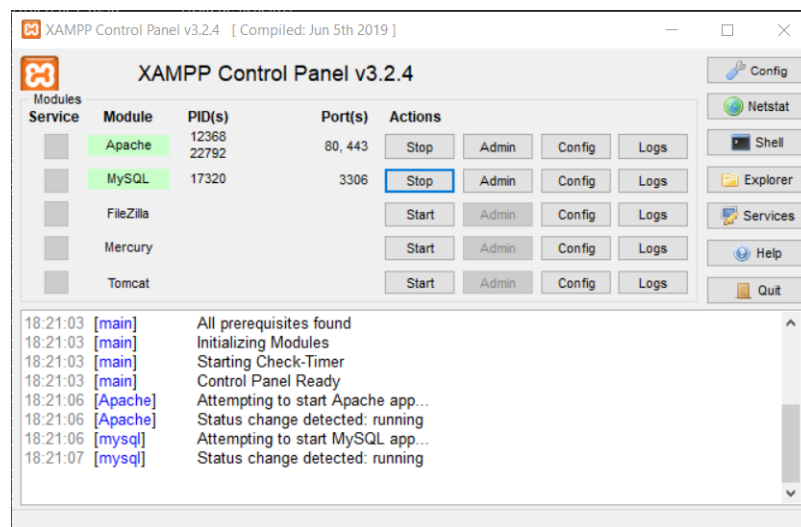


Figura 4.8: Painel de controlo XAMPP com Apache e MySQL ativos.

Para começar a fazer testes com o servidor local basta ativar dois módulos no painel de controlo do *XAMPP* sendo estes o *Apache* e *MySQL* como se pode verificar na Figura 4.8 do painel de controlo, sendo que a porta a utilizar no *Apache* é a porta 80 que será especificada no código Arduino descrito na Secção 4.4.

Clicando no botão *Admin* do painel de controlo do *XAMPP* é uma forma fácil de se aceder ao phpMyAdmin, a partir desta ferramenta é possível criar e remover bases de dados, criar, remover e alterar tabelas, inserir, remover e editar campos, executar códigos SQL e manipular campos chaves, isto tudo através do browser.

Capítulo 5

Resultados e Validação

Depois de desenvolvido o dispositivo SMARTMOV e a base de dados, foram efectuados testes para verificar o seu bom funcionamento.

5.1 Teste Sensor Laser

Como foi escrito no decorrer desta dissertação, o objetivo do sensor laser é contar o número de pessoas que entra e sai de um autocarro, ou seja, o fluxo de pessoas em cada paragem. Sendo assim os valores que são recolhidos de fluxo terão de ser o mais próximo possível do número real ou mesmo o valor exato, pois um valor errado nesta medida pode levar, futuramente, a erros na previsão de fluxo ou até mesmo a erros na recomendação de novas rotas. Assim para tentar diminuir o erro ao mínimo pensou-se em duas hipóteses para o posicionamento do sensor laser, sendo uma delas colocar o sensor a uma altura de tal forma que estivesse ao nível de detetar somente as pernas das pessoas e outra colocar o sensor a uma altura de forma a detetar a cintura das pessoas. Foram então feitos testes, que serão demonstrados em seguida, para perceber qual seria a melhor alternativa. Os testes realizados consistem em colocar o sensor a uma altura de 89 cm e 60 cm e contabilizar as vezes que se entra na sala, comparando depois o número real de pessoas com o número dado pelo sensor.

Teste a altura de 89 cm

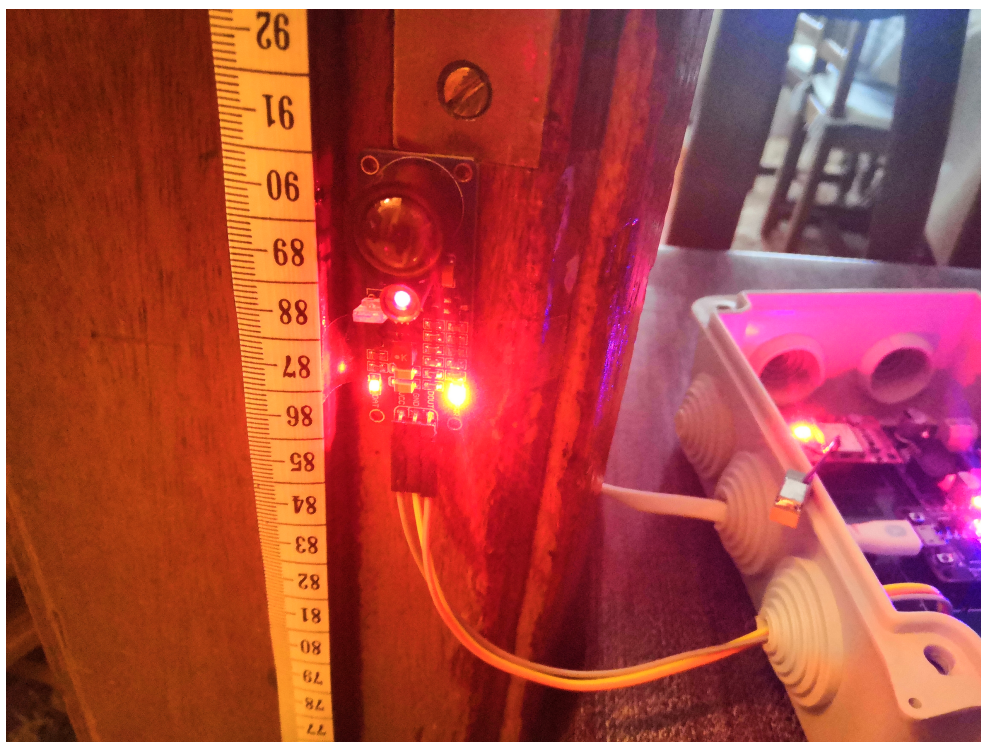


Figura 5.1: Fotografia do teste a 89 cm de altura.

Primeiramente foi realizado um teste com o sensor a uma altura de 89 cm do nível do chão como se pode ver na Figura 5.1, o que resulta que fique à altura da cintura de uma pessoa. Com este teste foram obtidos os dados que se podem observar na Tabela 5.1.

Valor Real	Valor recolhido
10	12
15	19
20	25
30	38
40	51
50	62
60	71

Tabela 5.1: Tabela de valor recolhido de fluxo a altura de 89cm

Como se pode observar os dados recolhidos não são iguais aos dados reais sendo que o valor recolhido é sempre superior. Depois de uma análise aos dados recolhidos e ao posicionamento do sensor laser, chegou-se à conclusão de que a diferença entre o valor

real e o valor recolhido pelo dispositivo SMARTMOV se deve ao facto de como o sensor laser se encontrar ao nível da cintura de uma pessoa, por vezes ele detecta as mãos/braços da pessoa o que leva a um valor errado do fluxo. Assim este posicionamento do sensor laser não será o melhor pois não é possível fazer uma correção fácil do erro dos valores recolhidos, pois os braços não são sempre detectados.

Teste a altura de 60 cm

Depois das conclusões retiradas do primeiro teste a uma altura de 89 cm foi realizado um teste, a uma altura inferior, com o sensor a uma altura de 60 cm do nível do chão como se pode ver na Figura 5.2, o que resulta que fique ao nível das pernas de uma pessoa. Com este teste foram obtidos os dados que se podem observar na Tabela 5.2.

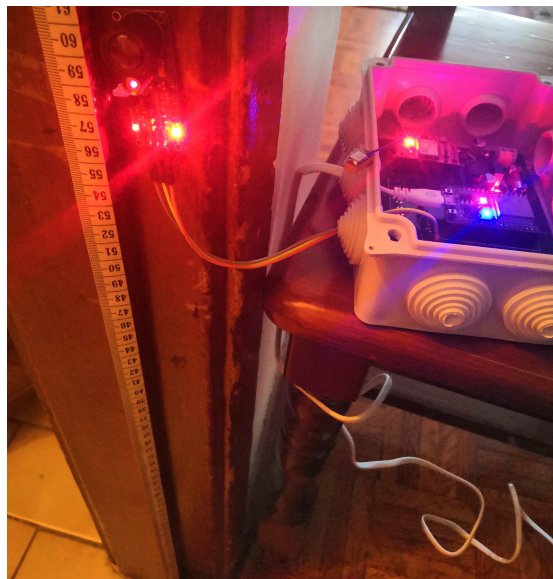


Figura 5.2: Fotografia do teste a 60 cm de altura.

Como se pode observar os dados recolhidos neste teste a 60 cm de altura, são sempre o dobro do valor real. Depois de uma análise aos dados recolhidos e ao posicionamento do sensor laser, chegou-se á conclusão de que a diferença entre o valor real e o valor recolhido pelo dispositivo SMARTMOV se deve ao facto de como o sensor laser se encontrar ao nível das pernas de uma pessoa, o que está a acontecer é que estão a ser detectadas ambas as pernas, levando assim a que o valor recolhido seja sempre o dobro do valor real. Assim

Valor Real	Valor recolhido
10	20
20	40
30	60
40	80
50	100
60	120

Tabela 5.2: Tabela de valor recolhido de fluxo a altura de 60 cm

foi feita uma pequena alteração no código Arduino no qual se divide o valor recolhido por dois, ou seja:

$$Fluxo = \frac{ValorRecolhido}{2} \quad (5.1)$$

Desta forma em teoria o valor recolhido será igual ao valor real, pois estamos a retirar a dupla contagem, uma por cada perna. De forma a comprovar esta teoria foi realizado outro teste á mesma altura de 60 cm mas com esta alteração no código, fora assim obtidos os valores que se podem observar na Tabela 5.3.

Valor Real	Valor recolhido
10	10
20	20
30	30
40	40
50	50
60	60

Tabela 5.3: Tabela de valor recolhido de fluxo a altura de 60 cm com alteração no código Arduino

Como se pode observar os dados recolhidos neste teste a 60 cm de altura, com a alteração no código Arduino efectuada, são exatamente iguais ao dados reais.

Com estes testes realizados ao sensor laser foi possível concluir que o melhor posicionamento do sensor será a uma altura de 60 cm ou inferior de forma a obter os valores mais próximos ou até mesmo exatamente iguais ao valores reais, como foi o caso dos testes realizados a 60 cm de altura.

5.2 Teste GPS

A função de recolha de localização do dispositivo SMARTMOV, através de um módulo GPS, e o envio dos mesmos para a base de dados é a função mais importante do dispositivo, assim é necessário realizar testes de maneira a concluir se existem erros na recolha da latitude e longitude e em que condições esses erros podem ocorrer.

De forma a testar a capacidade do módulo GPS utilizado no dispositivo SMARTMOV, foram realizados dois testes, no primeiro teste o dispositivo encontra-se no tabelier do carro com a antena do GPS sem qualquer obstrução como se pode ver na Figura 5.3. No segundo teste o dispositivo foi colocado no porta luvas do carro como se pode observar na Figura 5.4, onde a antena do GPS, na teoria, teria mais dificuldade de receção. O objetivo destes testes é saber se o local onde se vai colocar o dispositivo SMARTMOV no veiculo, terá influência numa recolha mais correta da latitude e da longitude.



Figura 5.3: Fotografia do dispositivo SMARTMOV no tabelier do carro no decorrer do primeiro teste.

No segundo teste o dispositivo foi colocado no porta luvas do carro como se pode observar na Figura 5.4, onde a antena do GPS, na teoria, teria mais dificuldade de receção.

O objetivo destes testes é saber se o local onde se vai colocar o dispositivo SMARTMOV no veículo, terá influência numa recolha mais correta da latitude e da longitude.

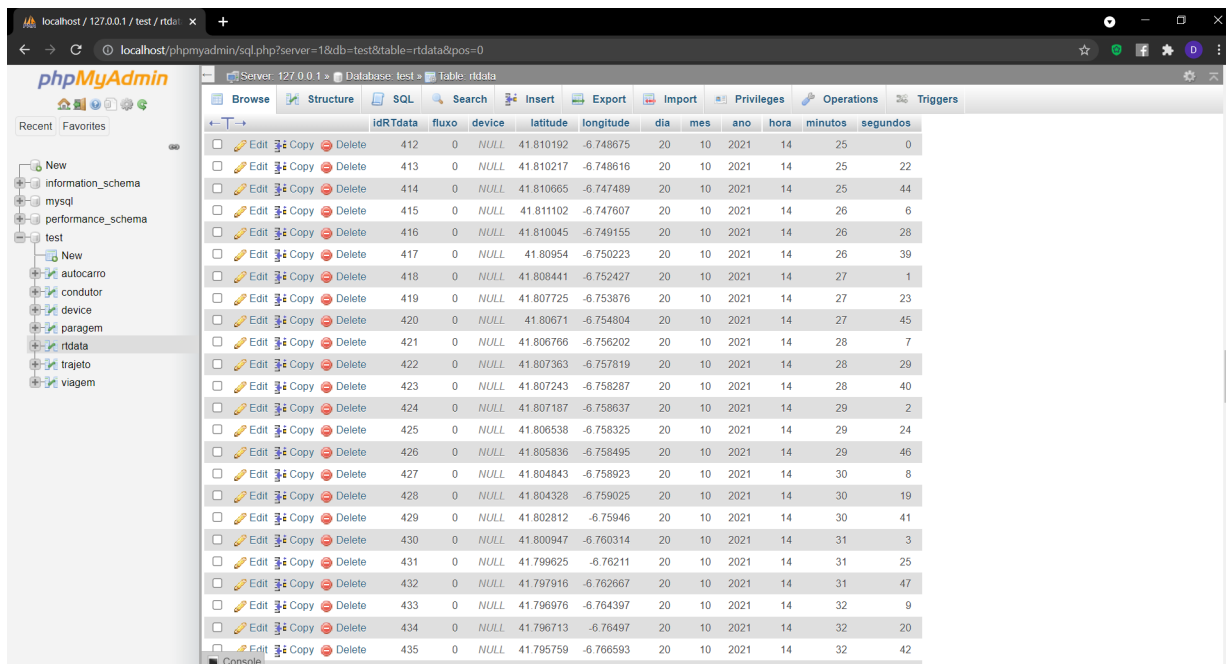


Figura 5.4: Fotografia do dispositivo SMARTMOV no porta luvas do carro no decorrer do segundo teste.

5.2.1 Validação dos Dados Recolhidos

Como foi referido anteriormente na Secção 5.2 foram realizados testes ao dispositivo SMARTMOV de maneira a testar a sua precisão em diferentes locais do veículo, como com a antena sem obstrução no tabelier do carro e com a antena com obstrução no porta luvas. Assim depois de realizados os testes é necessário estudar os dados de maneira a perceber a sua exatidão. Com estes testes foram recolhidos 144 dados de localização sendo 144 dados de latitude e 144 de longitude, na Figura 5.5 é possível visualizar alguns dos dados ainda organizados na Tabela da base de dados, antes de serem retirados para fazer o *pinpoint* no mapa, e ter uma representação da localização resultante dos valores da latitude e longitude.

Para retirar os dados da base de dados basta fazer a sua exportação como ficheiro *CSV* isto através do phpMyAdmin ou de um código python que foi também desenvolvido, de seguida basta abrir o ficheiro *CSV* no *EXCEL* e os dados ficam prontos a ser utilizados para qualquer finalidade.



	idRTdata	fluxo	device	latitude	longitude	dia	mes	ano	hora	minutos	segundos
<input type="checkbox"/>	412	0	NULL	41.810192	-6.748675	20	10	2021	14	25	0
<input type="checkbox"/>	413	0	NULL	41.810217	-6.748616	20	10	2021	14	25	22
<input type="checkbox"/>	414	0	NULL	41.810665	-6.747489	20	10	2021	14	25	44
<input type="checkbox"/>	415	0	NULL	41.811102	-6.747607	20	10	2021	14	26	6
<input type="checkbox"/>	416	0	NULL	41.810045	-6.749155	20	10	2021	14	26	28
<input type="checkbox"/>	417	0	NULL	41.80954	-6.750223	20	10	2021	14	26	39
<input type="checkbox"/>	418	0	NULL	41.808441	-6.752427	20	10	2021	14	27	1
<input type="checkbox"/>	419	0	NULL	41.807725	-6.753876	20	10	2021	14	27	23
<input type="checkbox"/>	420	0	NULL	41.80671	-6.754804	20	10	2021	14	27	45
<input type="checkbox"/>	421	0	NULL	41.806766	-6.756202	20	10	2021	14	28	7
<input type="checkbox"/>	422	0	NULL	41.807363	-6.757819	20	10	2021	14	28	29
<input type="checkbox"/>	423	0	NULL	41.807243	-6.758287	20	10	2021	14	28	40
<input type="checkbox"/>	424	0	NULL	41.807187	-6.758637	20	10	2021	14	29	2
<input type="checkbox"/>	425	0	NULL	41.806538	-6.758325	20	10	2021	14	29	24
<input type="checkbox"/>	426	0	NULL	41.805836	-6.758495	20	10	2021	14	29	46
<input type="checkbox"/>	427	0	NULL	41.804843	-6.758923	20	10	2021	14	30	8
<input type="checkbox"/>	428	0	NULL	41.804328	-6.759025	20	10	2021	14	30	19
<input type="checkbox"/>	429	0	NULL	41.802812	-6.75946	20	10	2021	14	30	41
<input type="checkbox"/>	430	0	NULL	41.800947	-6.760314	20	10	2021	14	31	3
<input type="checkbox"/>	431	0	NULL	41.799625	-6.76211	20	10	2021	14	31	25
<input type="checkbox"/>	432	0	NULL	41.797916	-6.762667	20	10	2021	14	31	47
<input type="checkbox"/>	433	0	NULL	41.796976	-6.764397	20	10	2021	14	32	9
<input type="checkbox"/>	434	0	NULL	41.796713	-6.76497	20	10	2021	14	32	20
<input type="checkbox"/>	435	0	NULL	41.795759	-6.766593	20	10	2021	14	32	42

Figura 5.5: Dados de localização recolhidos pelo dispositivo SMARTMOV

Trajetos realizados

Para ser possível fazer comparações entre um mapa com um trajeto real que se iria fazer na realização dos testes e o mapa resultante dos dados recolhidos pelo dispositivo SMARTMOV, foi delineada uma trajetória a seguir em cada teste como se pode ver nas Figuras 5.6 e 5.7. Sendo que o primeiro trajeto terá início na Avenida Cidade Zamora e terminará na entrada da ESTIG (Escola Superior de Tecnologia e Gestão), o segundo trajeto será o contrário, tendo início na ESTIG e terminando na Avenida Cidade de Zamora, mas passando também pela Avenida Sá Carneiro.

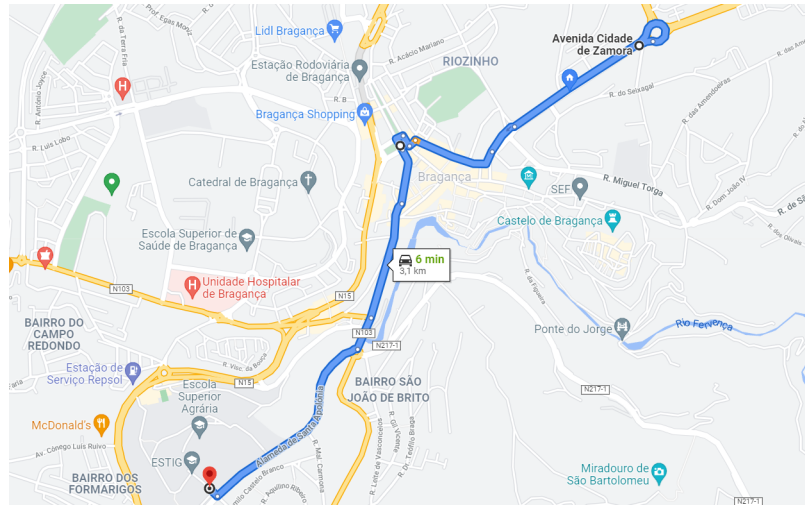


Figura 5.6: Percurso número 1 realizado nos testes ao dispositivo SMARTMOV.

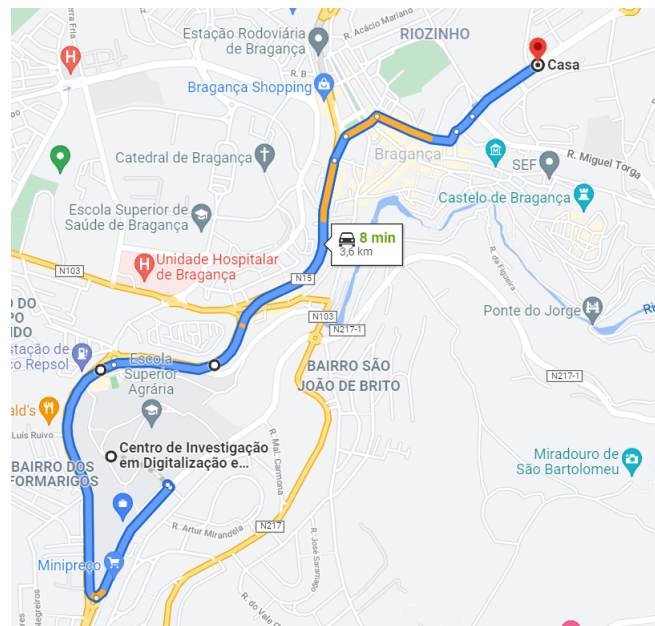


Figura 5.7: Percurso número 2 realizado nos testes ao dispositivo SMARTMOV.

Teste 1: Dispositivo SMARTMOV localizado no tablier

Na realização do Teste 1, o dispositivo SMARTMOV foi então colocado no tablier do carro, ficando sem qualquer obstrução. Desse mesmo teste foram recolhidos os dados que se podem observar na Tabela 5.4 para o trajeto número 1 realizado e os dados da Tabela 5.5 para trajeto número 2.

latitude	longitude	dia	mes	ano	hora	minutos	segundos
41.810192	-6.748675	20.0	10.0	2021.0	14.0	25.0	0.0
41.810217	-6.748616	20.0	10.0	2021.0	14.0	25.0	22.0
41.810665	-6.747489	20.0	10.0	2021.0	14.0	25.0	44.0
41.811102	-6.747607	20.0	10.0	2021.0	14.0	26.0	6.0
41.810045	-6.749155	20.0	10.0	2021.0	14.0	26.0	28.0
41.80954	-6.750223	20.0	10.0	2021.0	14.0	26.0	39.0
41.808441	-6.752427	20.0	10.0	2021.0	14.0	27.0	1.0
41.807725	-6.753876	20.0	10.0	2021.0	14.0	27.0	23.0
41.80671	-6.754804	20.0	10.0	2021.0	14.0	27.0	45.0
41.806766	-6.756202	20.0	10.0	2021.0	14.0	28.0	7.0
41.807363	-6.757819	20.0	10.0	2021.0	14.0	28.0	29.0
41.807243	-6.758287	20.0	10.0	2021.0	14.0	28.0	40.0
41.807187	-6.758637	20.0	10.0	2021.0	14.0	29.0	2.0
41.806538	-6.758325	20.0	10.0	2021.0	14.0	29.0	24.0
41.805836	-6.758495	20.0	10.0	2021.0	14.0	29.0	46.0
41.804843	-6.758923	20.0	10.0	2021.0	14.0	30.0	8.0
41.804328	-6.759025	20.0	10.0	2021.0	14.0	30.0	19.0
41.802812	-6.75946	20.0	10.0	2021.0	14.0	30.0	41.0
41.800947	-6.760314	20.0	10.0	2021.0	14.0	31.0	3.0
41.799625	-6.76211	20.0	10.0	2021.0	14.0	31.0	25.0
41.797916	-6.762667	20.0	10.0	2021.0	14.0	31.0	47.0
41.796976	-6.764397	20.0	10.0	2021.0	14.0	32.0	9.0
41.796713	-6.76497	20.0	10.0	2021.0	14.0	32.0	20.0
41.795759	-6.766593	20.0	10.0	2021.0	14.0	32.0	42.0
41.795721	-6.767102	20.0	10.0	2021.0	14.0	33.0	4.0
41.795719	-6.767136	20.0	10.0	2021.0	14.0	33.0	26.0

Tabela 5.4: Tabela valores recolhidos primeiro teste, trajeto número 1.

Fazer o estudo da exatidão do dispositivo SMARTMOV através dos valores recolhidos no formato de tabela seria uma tarefa muito complicada ou até mesmo impossível, então foi necessário encontrar uma solução para este problema, sendo que a melhor solução seria conseguir fazer o pinpoint destes valores num mapa, de maneira a perceber se formariam o mesmo trajeto que foi realizado na realidade. Depois de uma pesquisa sobre formas de fazer isto, foi encontrado o site da *BatchGeo* [21].

Como se pode ver na Figura 5.8 basta colocar os dados da latitude e longitude que se encontram nas tabelas e depois de uma pequena personalização, irá resultar nos mapas

latitude	longitude	dia	mes	ano	hora	minutos	segundos
41.795739	-6.767137	20.0	10.0	2021.0	14.0	33.0	48.0
41.795726	-6.767125	20.0	10.0	2021.0	14.0	33.0	59.0
41.795679	-6.767195	20.0	10.0	2021.0	14.0	34.0	21.0
41.795662	-6.767215	20.0	10.0	2021.0	14.0	34.0	43.0
41.795436	-6.766997	20.0	10.0	2021.0	14.0	35.0	5.0
41.794606	-6.768059	20.0	10.0	2021.0	14.0	35.0	27.0
41.794008	-6.768735	20.0	10.0	2021.0	14.0	35.0	49.0
41.793303	-6.769118	20.0	10.0	2021.0	14.0	36.0	0.0
41.792058	-6.770217	20.0	10.0	2021.0	14.0	36.0	22.0
41.794419	-6.770383	20.0	10.0	2021.0	14.0	36.0	44.0
41.79597	-6.770771	20.0	10.0	2021.0	14.0	37.0	6.0
41.797414	-6.77131	20.0	10.0	2021.0	14.0	37.0	28.0
41.79865	-6.770799	20.0	10.0	2021.0	14.0	37.0	39.0
41.799413	-6.769574	20.0	10.0	2021.0	14.0	38.0	1.0
41.79943	-6.767086	20.0	10.0	2021.0	14.0	38.0	23.0
41.799327	-6.765248	20.0	10.0	2021.0	14.0	38.0	45.0
41.799669	-6.764496	20.0	10.0	2021.0	14.0	39.0	7.0
41.800863	-6.763709	20.0	10.0	2021.0	14.0	39.0	29.0
41.801559	-6.76265	20.0	10.0	2021.0	14.0	39.0	40.0
41.802087	-6.76124	20.0	10.0	2021.0	14.0	40.0	2.0
41.803411	-6.760341	20.0	10.0	2021.0	14.0	40.0	24.0
41.804645	-6.760333	20.0	10.0	2021.0	14.0	40.0	46.0
41.806248	-6.759803	20.0	10.0	2021.0	14.0	41.0	8.0
41.806748	-6.759426	20.0	10.0	2021.0	14.0	41.0	19.0
41.807109	-6.758632	20.0	10.0	2021.0	14.0	41.0	41.0
41.807046	-6.757712	20.0	10.0	2021.0	14.0	42.0	3.0
41.806824	-6.75682	20.0	10.0	2021.0	14.0	42.0	25.0
41.806822	-6.756428	20.0	10.0	2021.0	14.0	42.0	47.0
41.806765	-6.754798	20.0	10.0	2021.0	14.0	43.0	9.0
41.8071	-6.754355	20.0	10.0	2021.0	14.0	43.0	20.0
41.808262	-6.752684	20.0	10.0	2021.0	14.0	43.0	42.0
41.809321	-6.750478	20.0	10.0	2021.0	14.0	44.0	4.0
41.810175	-6.748755	20.0	10.0	2021.0	14.0	44.0	26.0
41.810197	-6.748715	20.0	10.0	2021.0	14.0	44.0	48.0
41.795739	-6.767137	20.0	10.0	2021.0	14.0	44.0	59.0

Tabela 5.5: Tabela valores recolhidos primeiro teste, trajeto número 2.

da Figura 5.9 e da Figura 5.11, sendo assim possível fazer uma comparação de trajetos, trajeto real e o trajeto criado pelos dados recolhidos.

batchgeo Recursos Precificação Móvel Suporte login da conta Sign Up

Cole seus dados de localização abaixo para mapeá-los:

Example Address	City	State	Zip	Name	Phone Number	Group	URL
1 Crossgates Mall Road	Albany	NY	12203	Apple Store Cross Gates	(518) 869-3192	Example Group 1	http://www.apple.com/retail/cro
Duke Rd & Waiden Ave	Buffalo	NY	14225	Apple Store Waiden Galleria	(716) 685-2762	Example Group 2	http://www.apple.com/retail/wal
630 Old Country Rd	Garden City	NY	11530	Apple Store Garden City	(516) 348-3347	Example Group 3	http://www.apple.com/retail/gar
190 West Whitman Rd	Huntington Station	NY	11746	Apple Store West Whitman	(631) 425-1563	Example Group 3	http://www.apple.com/retail/wst
525 Carlewood Center Drive	Syracuse	NY	13208	Apple Store Carlewood	(315) 433-9454	Example Group 2	http://www.apple.com/retail/car

(Não esqueça de incluir algumas colunas de cabeçalho - Você também pode tentar o nosso [Modelo de planilha \(Excel\)](#), ou clique em "Mapear agora" e experimente com nossos dados de exemplo.)

Validar e definir opções Mapear Agora

Figura 5.8: Site BatchGeo utilizado para fazer o pinpoint dos valores recolhidos.



Figura 5.9: Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no primeiro percurso com o dispositivo no tabelier

Como se pode observar e depois de se fazer uma comparação das duas figuras resultantes do pinpoint dos valores obtidos pelo dispositivo SMARTMOV, com as Figuras 5.6 e 5.7 dos percursos reais efectuados, é possível chegar à conclusão de que o dispositivo SMARTMOV com a antena sem qualquer obstrução terá uma exatidão excelente,



Figura 5.10: Mapa resultante dos dados de localização recolhidos pelo dispositivo SMARTMOV no segundo percurso com o dispositivo no tabelier

pois mostra todos os pontos do percurso até mesmo os pontos de viragem como curvas e rotundas de forma clara.

Teste 2: Dispositivo SMARTMOV localizado no porta luvas

Depois dos excelentes resultados do primeiro teste, decidiu-se fazer um segundo teste onde desta vez o dispositivo SMARTMOV estaria no porta luvas do carro, o pretendido neste teste e como já foi referido anteriormente, é perceber se existirá alguma mudança na recolha dos valores de latitude e longitude, tentando concluir assim se o local onde se coloca o dispositivo SMARTMOV dentro do veículo, terá algum impacto na boa recolha de dados. Assim foram percorridos novamente ambos os percursos das figuras 5.6 e 5.7 onde foram obtidos os dados das Tabelas 5.6 e 5.7 como no teste anterior.

Depois de se utilizar novamente o site da *BatchGeo* para fazer o pinpoint dos novos dados recolhidos, foram obtidos os mapas da Figura 5.11 para o percurso 1 e da Figura 5.12 para o percurso 2.

Desta vez para retirar as conclusões da precisão do dispositivo SMARTMOV dentro

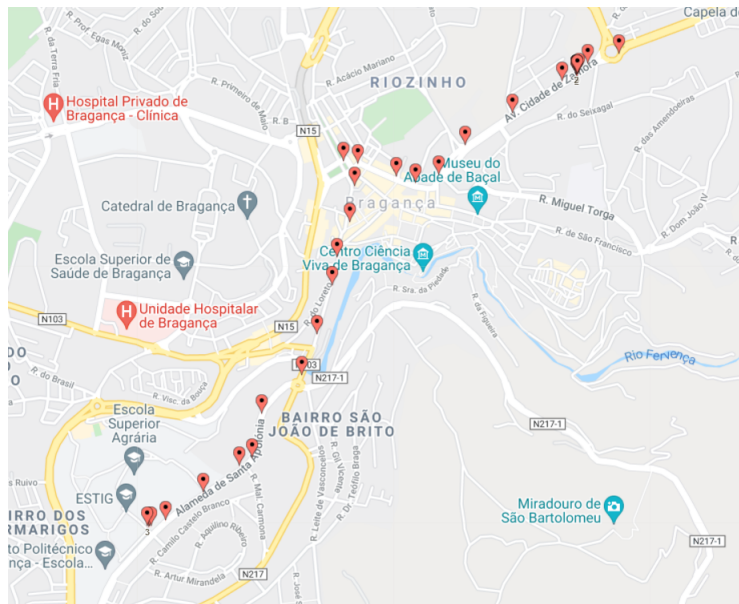


Figura 5.11: Mapa resultante dos dados de localização recolhidos pelo dispositivo SMART-MOV no primeiro percurso com o dispositivo no porta luvas



Figura 5.12: Mapa resultante dos dados de localização recolhidos pelo dispositivo SMART-MOV no segundo percurso com o dispositivo no porta luvas

do porta luvas, é possível fazer uma comparação com os percursos reais, mas também com os percursos resultantes do primeiro teste com o dispositivo no tabelier do carro.

Comparando todos os mapas é possível chegar á conclusão de que estando ou não obstruído o dispositivo SMARTMOV terá sempre o mesmo nível de precisão da latitude e longitude. Analisando os mapas é possível perceber que esse nível de precisão é excelente para um dispositivo low-cost. Assim as conclusões finais dos testes realizados são que, o dispositivo SMARTMOV tem a flexibilidade de se poder colocar em qualquer parte do veículo sabendo que terá sempre uma recolha de dados excelente, o que permite uma maior facilidade de instalação pois a boa recolha de dados não estará dependente do local onde o dispositivo de encontra.

latitude	longitude	dia	mes	ano	hora	minutos	segundos
41.810179	-6.748717	20.0	10.0	2021.0	14.0	45.0	21.0
41.810168	-6.748713	20.0	10.0	2021.0	14.0	45.0	43.0
41.810167	-6.74872	20.0	10.0	2021.0	14.0	46.0	5.0
41.810151	-6.748733	20.0	10.0	2021.0	14.0	46.0	27.0
41.81013	-6.748745	20.0	10.0	2021.0	14.0	46.0	49.0
41.810136	-6.748738	20.0	10.0	2021.0	14.0	47.0	0.0
41.81016	-6.748733	20.0	10.0	2021.0	14.0	47.0	22.0
41.810175	-6.748741	20.0	10.0	2021.0	14.0	47.0	44.0
41.810183	-6.74876	20.0	10.0	2021.0	14.0	48.0	6.0
41.810183	-6.748757	20.0	10.0	2021.0	14.0	48.0	28.0
41.810184	-6.748744	20.0	10.0	2021.0	14.0	48.0	39.0
41.810184	-6.748721	20.0	10.0	2021.0	14.0	49.0	1.0
41.810168	-6.748707	20.0	10.0	2021.0	14.0	49.0	23.0
41.810227	-6.748641	20.0	10.0	2021.0	14.0	49.0	45.0
41.810832	-6.746923	20.0	10.0	2021.0	14.0	50.0	7.0
41.810516	-6.74826	20.0	10.0	2021.0	14.0	50.0	29.0
41.809957	-6.749357	20.0	10.0	2021.0	14.0	50.0	40.0
41.808917	-6.751479	20.0	10.0	2021.0	14.0	51.0	2.0
41.807913	-6.753526	20.0	10.0	2021.0	14.0	51.0	24.0
41.80696	-6.75466	20.0	10.0	2021.0	14.0	51.0	46.0
41.8067	-6.755644	20.0	10.0	2021.0	14.0	52.0	8.0
41.806896	-6.756487	20.0	10.0	2021.0	14.0	52.0	19.0
41.807323	-6.758114	20.0	10.0	2021.0	14.0	52.0	41.0
41.807384	-6.758744	20.0	10.0	2021.0	14.0	53.0	3.0
41.806601	-6.758259	20.0	10.0	2021.0	14.0	53.0	25.0
41.805443	-6.758476	20.0	10.0	2021.0	14.0	53.0	47.0
41.804334	-6.759023	20.0	10.0	2021.0	14.0	54.0	9.0
41.803418	-6.759228	20.0	10.0	2021.0	14.0	54.0	20.0
41.801861	-6.759858	20.0	10.0	2021.0	14.0	54.0	42.0
41.800551	-6.76054	20.0	10.0	2021.0	14.0	55.0	4.0
41.799328	-6.762232	20.0	10.0	2021.0	14.0	55.0	26.0
41.797912	-6.76265	20.0	10.0	2021.0	14.0	55.0	48.0
41.797676	-6.763191	20.0	10.0	2021.0	14.0	55.0	59.0
41.79681	-6.764748	20.0	10.0	2021.0	14.0	56.0	21.0
41.795924	-6.766375	20.0	10.0	2021.0	14.0	56.0	43.0
41.795749	-6.766961	20.0	10.0	2021.0	14.0	57.0	5.0
41.795743	-6.767054	20.0	10.0	2021.0	14.0	57.0	27.0
41.795727	-6.767071	20.0	10.0	2021.0	14.0	57.0	49.0

Tabela 5.6: Tabela valores recolhidos segundo teste, trajeto número 1.

latitude	longitude	dia	mes	ano	hora	minutos	segundos
41.795717	-6.767085	20.0	10.0	2021.0	14.0	58.0	0.0
41.795696	-6.767098	20.0	10.0	2021.0	14.0	58.0	22.0
41.795684	-6.767092	20.0	10.0	2021.0	14.0	58.0	44.0
41.795674	-6.767096	20.0	10.0	2021.0	14.0	59.0	6.0
41.795687	-6.767096	20.0	10.0	2021.0	14.0	59.0	28.0
41.795694	-6.767096	20.0	10.0	2021.0	14.0	59.0	39.0
41.795689	-6.7671	20.0	10.0	2021.0	15.0	0.0	1.0
41.795666	-6.767106	20.0	10.0	2021.0	15.0	0.0	23.0
41.795521	-6.766994	20.0	10.0	2021.0	15.0	0.0	45.0
41.795038	-6.767512	20.0	10.0	2021.0	15.0	1.0	7.0
41.794121	-6.768619	20.0	10.0	2021.0	15.0	1.0	29.0
41.794105	-6.768648	20.0	10.0	2021.0	15.0	1.0	40.0
41.793686	-6.769024	20.0	10.0	2021.0	15.0	2.0	2.0
41.792196	-6.770116	20.0	10.0	2021.0	15.0	2.0	24.0
41.794211	-6.770366	20.0	10.0	2021.0	15.0	2.0	46.0
41.79569	-6.770596	20.0	10.0	2021.0	15.0	3.0	8.0
41.796726	-6.771126	20.0	10.0	2021.0	15.0	3.0	19.0
41.798486	-6.77088	20.0	10.0	2021.0	15.0	3.0	41.0
41.799408	-6.769347	20.0	10.0	2021.0	15.0	4.0	3.0
41.799366	-6.766576	20.0	10.0	2021.0	15.0	4.0	25.0
41.800033	-6.764128	20.0	10.0	2021.0	15.0	4.0	47.0
41.801204	-6.763371	20.0	10.0	2021.0	15.0	5.0	9.0
41.80176	-6.762151	20.0	10.0	2021.0	15.0	5.0	20.0
41.802524	-6.760754	20.0	10.0	2021.0	15.0	5.0	42.0
41.803919	-6.760297	20.0	10.0	2021.0	15.0	6.0	4.0
41.805315	-6.760016	20.0	10.0	2021.0	15.0	6.0	26.0
41.806689	-6.7593	20.0	10.0	2021.0	15.0	6.0	48.0
41.806954	-6.758751	20.0	10.0	2021.0	15.0	6.0	59.0
41.807204	-6.757827	20.0	10.0	2021.0	15.0	7.0	21.0
41.806568	-6.755763	20.0	10.0	2021.0	15.0	7.0	43.0
41.806927	-6.754406	20.0	10.0	2021.0	15.0	8.0	5.0
41.808212	-6.752593	20.0	10.0	2021.0	15.0	8.0	27.0
41.809355	-6.750312	20.0	10.0	2021.0	15.0	8.0	49.0
41.809819	-6.749387	20.0	10.0	2021.0	15.0	9.0	0.0
41.81014	-6.748738	20.0	10.0	2021.0	15.0	9.0	22.0
41.810173	-6.748712	20.0	10.0	2021.0	15.0	9.0	44.0
41.81015	-6.748699	20.0	10.0	2021.0	15.0	10.0	6.0
41.810147	-6.748697	20.0	10.0	2021.0	15.0	10.0	28.0

Tabela 5.7: Tabela valores recolhidos segundo teste, trajeto número 2.

Capítulo 6

Conclusões e Trabalho Futuro

Esta dissertação apresenta um dispositivo baseado em *IoT*, intitulado dispositivo SMARTMOV, que quando for combinado com o trabalho realizado em uma outra dissertação, dedicada somente à análise de dados, espera-se que melhore muito a área dos transportes públicos.

Os testes realizados com sucesso ao dispositivo SMARTMOV e aqui demonstrados nesta dissertação, resultados esses atingidos com um dispositivo de baixo custo e fácil montagem, onde se concluiu o melhor posicionamento do sensor laser para a recolha do fluxo, e se comprovou que o dispositivo SMARTMOV pode ser colocado em qualquer parte do veículo, sendo que isso não influencia a boa recolha dos dados de localização do veículo, indica que os componentes escolhidos e utilizados na sua construção, foram de escolha acertada e que a ligação dos mesmos entre si está a funcionar de forma excelente.

Este projeto poderá então agora passar para a próxima fase de interligação com a análise de dados e previsões, oferecendo assim informações que estarão à disposição das empresas com o objetivo de monitorizar a sua frota e ao público com a finalidade de visualizar o horário de chegada dos autocarros, oferecendo assim uma maior aproximação das empresas com as pessoas.

Bibliografia

- [1] R. K. Kodali e A. Valdas, “MQTT Based Monitoring System for Urban Farmers Using ESP32 and Raspberry Pi,” *Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 395–398, 2018.
- [2] J. Okwuibe, J. Haavisto, E. Harjula, I. Ahmad e M. Ylianttila, “SDN Enhanced Resource Orchestration of Containerized Edge Applications for Industrial IoT,” *IEEE Access*, vol. 8, pp. 229 117–229 131, 2020. DOI: 10.1109/ACCESS.2020.3045563. URL: <https://ieeexplore.ieee.org/document/9296769>.
- [3] P. Marmé, *IoT-Internet of Things*, <https://wattson.pt/2018/03/09/o-que-e-a-internet-das-coisas-iot-internet-of-things/>, jan. de 2021.
- [4] Oracle, *What is IoT?* <https://www.oracle.com/pt/internet-of-things/what-is-iot/>, jan. de 2021.
- [5] Amper, *About Amper*, <https://www.amper.xyz/company/>, dezembro de 2020.
- [6] Axzon, *About Axzon*, <https://axzon.com/about-us/>, dezembro de 2020.
- [7] Airbiquity, *About Airbiquity*, <https://www.airbiquity.com/company>, dezembro de 2020.
- [8] Zubie, *Real-Time GPS Tracking for Fleet Vehicles*, <https://zubie.com/zubie-fleet-connect/gps-tracking/>, dezembro de 2020.
- [9] B. Limited, “PT600X User Manual,” 2014.
- [10] sinocastel, *GPS tracking*, <https://www.sinocastel.com/>, dezembro de 2020.

- [11] STCP, *Missão, Visão e Política*, <https://www.stcp.pt/pt/institucional/stcp/missao-visao-e-politica/>, outubro de 2021.
- [12] S. OPT, *MOVE-ME*, <https://www.move-me.mobi/>, setembro de 2021.
- [13] R. Team, *ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE*, <https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php>, fevereiro de 2020.
- [14] Espressif, *ESP32*, <https://www.espressif.com/en/products/socs/esp32>, outubro de 2020.
- [15] E. R. B. Marques, *O Modelo Relacional*, https://www.dcc.fc.up.pt/edrdo/aulas/bd-teoricas/bd_modelo_relacional.pdf, fevereiro de 2021.
- [16] Kinsta, *O Que é MySQL?* <https://kinsta.com/pt/base-de-conhecimento/o-que-e-mysql/relational-database>, fevereiro de 2021.
- [17] MySQL, *Supported Platforms*, <https://www.mysql.com/support/supportedplatforms/database.html>, fevereiro de 2021.
- [18] —, *MySQL Workbench*, <https://www.mysql.com/products/workbench/>, fevereiro de 2021.
- [19] XAMPP, *WHAT IS XAMPP?* <https://www.apachefriends.org/index.html>, mar. de 2021.
- [20] WAMPSEVER, *WAMP*, <https://www.wampserver.com/en/>, mar. de 2021.
- [21] BatchGeo, <https://pt.batchgeo.com/>, agosto de 2021.

Apêndice A

Código Arduino e PHP

Código Arduino

```
1 #include <AutoConnect.h>
2 #include <TinyGPS++.h>
3 #include <SoftwareSerial.h>
4 #include <WiFi.h>
5
6 const char* host = "DESKTOP-9951GHF";
7 const int httpPort = 80;
8
9 static const int RXPin = 4, TXPin = 2;
10 static const uint32_t GPSBaud = 9600;
11
12 int SensorPin = 21;
13 int count = 0;
14 int fluxo;
15 int segundos, minutos, hora, dia, mes, ano;
16
17 const long interval = 30000;
```

```
18 const long T_1 = 11000;
19 const long T_2 = 20000;
20
21
22 long lastT1, lastT2;
23
24 char latitude[15];
25 char longitude[15];
26 double lati;
27 double lon;
28
29 TinyGPSPlus gps;
30 SoftwareSerial ss(RXPin, TXPin);
31
32 AutoConnect      Portal;
33 AutoConnectConfig Config;
34
35
36
37 void IRAM_ATTR detectsMovement() {
38     count++;
39     Serial.print(F("_Fluxo:_"));
40     Serial.println(count/2);
41     fluxo = count/2;
42 }
43
44 void setup() {
45     Serial.begin(115200);
46     ss.begin(GPSBaud);
```

```
47
48 lastT1 = 0;
49 lastT2 = 0;
50
51 pinMode(SensorPin, INPUT_PULLUP);
52 attachInterrupt(digitalPinToInterrupt(SensorPin),
    detectsMovement, RISING);
53
54 Config.autoReset = false;
55 Config.autoReconnect = true;
56 Config.reconnectInterval = 6;
57 Config.retainPortal = true;
58 Portal.config(Config);
59 Portal.begin();
60
61
62 Serial.println("Connecting");
63 while (WiFi.status() != WL_CONNECTED) {
64     delay(500);
65     Serial.print(".");
66 }
67 Serial.println("");
68 Serial.print("Connected to WiFi network with IP Address: ");
69 Serial.println(WiFi.localIP());
70 }
71
72 void func_time1(void)
73 {
74
```

```
75 Serial.println("gps2");
76 {
77
78   if (gps.location.isValid())
79   {
80     lati = gps.location.lat();
81     lon = gps.location.lng();
82
83     sprintf(latitude, "%f", lati);
84     sprintf(longitude, "%f", lon);
85
86     Serial.print(F("Location:_"));
87
88
89     Serial.print(lati);
90     Serial.print(F(", "));
91     Serial.print(lon);
92   }
93   else
94   {
95     Serial.print(F("INVALID"));
96   }
97
98   Serial.print(F("_Date/Time:_"));
99   if (gps.date.isValid())
100  {
101    dia = gps.date.day();
102    mes = gps.date.month();
103    ano = gps.date.year();
```

```
104
105     Serial.print(dia);
106     Serial.print(F("/"));
107     Serial.print(mes);
108     Serial.print(F("/"));
109     Serial.print(ano);
110 }
111 else
112 {
113     Serial.print(F("INVALID"));
114 }
115
116 Serial.print(F("_"));
117 if (gps.time.isValid())
118 {
119     hora = gps.time.hour();
120     minutos = gps.time.minute();
121     segundos = gps.time.second();
122
123
124     if (gps.time.hour() < 10) Serial.print(F("0"));
125     Serial.print(hora);
126     Serial.print(F(":"));
127     if (gps.time.minute() < 10) Serial.print(F("0"));
128     Serial.print(minutos);
129     Serial.print(F(":"));
130     if (gps.time.second() < 10) Serial.print(F("0"));
131     Serial.print(segundos);
132 }
```

```
133     else
134     {
135         Serial.print(F("INVALID"));
136     }
137
138     Serial.println();
139 }
140 }
141
142
143
144 void func_time2(void)
145 {
146     Serial.println("Envio_de_Dados");
147
148     WiFiClient client;
149     if (!client.connect(host, httpPort)) {
150         Serial.println("connection_failed");
151         return;
152     }
153     Serial.println("sending...");
154
155
156     client.print(String("GET_http://localhost/GPSData/connect.php
157         ?") +
158         ("fluxo=") + fluxo +
159        ("&latitude=") + latitude +
160        ("&longitude=") + longitude +
161        ("&dia=") + dia +
```

```
161        ("&mes=") + mes +
162        ("&ano=") + ano +
163        ("&hora=") + hora +
164        ("&minutos=") + minutos +
165        ("&segundos=") + segundos +
166         "_HTTP/1.1\r\n" +
167         "Host:_" + host + "\r\n" +
168         "Connection:_close\r\n\r\n");
169
170     Serial.println("http_send");
171
172     while (client.available()) {
173         String line = client.readStringUntil('\r');
174         Serial.print(line);
175         fluxo = 0;
176     }
177 }
178
179
180 void loop() {
181     unsigned long currentMillis = millis();
182
183     if (ss.available() > 0)
184         gps.encode(ss.read());
185
186     if (currentMillis - lastT1 >= T_1) {
187         lastT1 = currentMillis;
188         func_time1();
189     }
```

```
190
191  if (currentMillis - lastT2 >= T_2) {
192      lastT2 = currentMillis;
193      func_time2();
194  }
195 }
```

Código PHP

```
1 <?php
2  $dbname = 'test';
3      $dbusername = "admin";
4      $dbpassword = "admin";
5      $server = "localhost";
6
7 $connect = @mysqli_connect($server,$dbusername,$dbpassword,
8     $dbname);
9
10     $fluxo = $_GET["fluxo"];
11     $latitude = $_GET["latitude"];
12     $longitude = $_GET["longitude"];
13     $dia = $_GET["dia"];
14     $mes = $_GET["mes"];
15     $ano = $_GET["ano"];
16     $hora = $_GET["hora"];
17     $minutos = $_GET["minutos"];
18     $segundos = $_GET["segundos"];
19
```

```
20 $query = "INSERT INTO rtdata (fluxo, latitude, longitude, dia,
    mes, ano, hora, minutos, segundos) VALUES ('$fluxo',
    '$latitude', '$longitude', '$dia', '$mes', '$ano', '$hora',
    '$minutos', '$segundos') ";
21 $result = mysqli_query($connect, $query);
22
23 ?>
```