

João Rafael Pereira Costa

Smart Cities - Monitorização Ambiental e de Tráfego Rodoviário



João Rafael Pereira Costa

Smart Cities - Monitorização Ambiental e de Tráfego Rodoviário

Tese de Mestrado

Engenharia Eletrotécnica - Energia e Automação Industrial

Professor Doutor Paulo Rogério Perfeito Tomé

Professor Doutor Daniel Filipe Albuquerque



“Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time.”

Thomas Edison

NOTA PRÉVIA

Este trabalho deu origem ao artigo: Arquitetura de Sistema LowCost para Monitorização de Parâmetros Ambientais em Vias de Circulação Automóvel, (Costa, J., Wanzeller, C., Albuquerque, D., Caldeira, F., Tomé, P. e Martins, P., "LowCost System Architecture for Monitoring Environmental Parameters in Circulating Automotive Roads,", 2019, 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-5).

RESUMO

A qualidade do ar é uma preocupação não só dos cidadãos, mas também, de diversas entidades que procuram, sempre que acham relevante, alertar as pessoas para a baixa qualidade do mesmo. Um sistema com capacidade para monitorizar a qualidade do ar e o volume de tráfego, poderia proporcionar às autarquias uma melhor visão destes dados em determinadas zonas da cidade.

O presente trabalho teve como propósito o desenvolvimento de um sistema de baixo custo capaz de monitorizar o volume de tráfego rodoviário e a medição de alguns dos parâmetros ambientais, tais como: temperatura, humidade relativa, pressão atmosférica, monóxido de carbono, entre outros.

O sistema desenvolvido faz a contagem e classificação de veículos numa via de trânsito através de uma câmara e a medição de parâmetros ambientais com recurso a sensores. Os quais são enviados periodicamente para um servidor, existindo também um sistema de registo local.

Foi também desenvolvida uma aplicação móvel para Android que permite a consulta dos valores armazenados no servidor. Desta forma, é possível a um utilizador consultar os parâmetros da qualidade do ar ou o número de veículos que circularam numa dada via. É ainda permitida, na aplicação, que um utilizador reporte a baixa qualidade do ar.

O sistema desenvolvido provou ser capaz de contar e classificar os veículos, assim como medir os parâmetros ambientais. Quanto à aplicação móvel, apresenta-se numa interface simples e intuitiva para o utilizador.

ABSTRACT

Air quality concerns not only the citizens but also several entities who look, when they find it important, to alert the people to its low quality. A system with the capacity to monitor air quality and traffic volume could give local authorities a better view of this data in certain areas of a city.

The presented work had the goal of developing a low-cost system capable of monitoring road traffic volume and measuring some of the environmental parameters, which include temperature, relative humidity, barometric pressure, carbon monoxide, and others.

The system that was developed counts and classifies vehicles in a traffic lane through a camera and measures environmental parameters using sensors. The collected data is periodically sent to a server, while also being registered locally.

There was also the development of a mobile application for the Android operating system, which allows the consultation of the data stored in the server. This way, it is possible for a user to consult the air quality parameters or the number of vehicles that circulated through a specific roadway. It is also possible for a user to report the low air quality, through the application.

The developed system proved its capabilities in counting and classifying vehicles as well as measuring the environmental parameters. As to the app, it presents itself in a simple and intuitive user interface.

PALAVRAS CHAVE

Aplicação Android
Cidades Inteligentes
Monitorização de Tráfego
Monitorização Ambiental
Processamento de Imagem
Qualidade do Ar

KEY WORDS

Android App
Smart Cities
Traffic Monitoring
Environmental Monitoring
Image Processing
Air Quality

AGRADECIMENTOS

Em primeiro lugar, um especial agradecimento aos meus orientadores, Professor Doutor Paulo Tomé e Professor Doutor Daniel Albuquerque, por todo o acompanhamento e dedicação ao longo deste trabalho.

Agradecimento ao projeto CityAction CENTRO-01-0247-FEDER-017711, apoiado pelo Programa Operacional Regional do Centro de Portugal (CENTRO 2020), ao abrigo do Acordo de Parceria Portugal 2020, através do Fundo Europeu de Desenvolvimento Regional (FEDER), e também financiado por fundos através da FCT - Fundação para a Ciência e Tecnologia, I.P., sob o projeto UID / Multi / 04016/2016.

O meu muito obrigado a todo o Departamento de Engenharia Eletrotécnica, por tudo o que me proporcionaram ao longo destes anos e ao Departamento de Informática, em particular nas pessoas do Professor Doutor Filipe Caldeira, da Professora Doutora Cristina Wanzeller e do Professor Doutor Pedro Martins, pelo apoio no projeto CityAction, e ainda ao Professor Doutor Filipe Cabral Pinto da Altice Labs.

Não poderei esquecer os meus amigos que nos momentos certos estiveram presentes e o meu maior reconhecimento e gratidão à minha família, especialmente aos meus pais e irmãos que, nos vários momentos, me apoiaram sem qualquer reserva.

ÍNDICE GERAL

ÍNDICE GERAL	xv
ÍNDICE DE FIGURAS	xvii
ÍNDICE DE TABELAS	xxi
ABREVIATURAS E SIGLAS	xxiii
1. Introdução	1
1.1 As Cidades Inteligentes e o Ambiente	2
1.2 Objetivos	4
1.3 Estrutura e Organização do Documento	4
2. Sistemas de Monitorização Ambiental e de Tráfego	7
2.1 Sistemas de Aquisição de Dados	7
2.2 Sistemas de Monitorização Ambiental	9
2.2.1 Análise aos Sistemas de Monitorização Ambiental	14
2.3 Sistemas de Monitorização de Tráfego	16
2.3.1 Análise aos Sistemas de Monitorização de Tráfego	19
3. Sistema Desenvolvido	23
3.1 Introdução	23
3.2 Computador	24
3.3 Medição de Parâmetros Ambientais	26
3.3.1 Sensor de Temperatura, Humidade e Pressão	27
3.3.2 Sensor de Luminosidade	28
3.3.3 Sensor de CO ₂ e TVOC	28
3.3.4 Sensor de CO, NO ₂ , NH ₃ , C ₃ H ₈ , C ₄ H ₁₀ , CH ₄ , H ₂ e C ₂ H ₅ OH	29
3.3.5 Sensor de O ₂	31
3.3.6 Sensor de Ruído	32
3.3.7 Conversor Analógico-Digital (ADC)	33
3.4 Câmara	33
3.5 Detalhes do Sistema Desenvolvido	34

ÍNDICE GERAL

3.6	Processamento de Dados	34
3.6.1	Leitura dos Dados	36
3.7	Processamento de Imagem	39
3.7.1	Bloco da Câmara para Raspberry Pi	40
3.7.2	Bloco de Região de Interesse (ROI)	41
3.7.3	Bloco de Detecção de Primeiro Plano	42
3.7.4	Bloco de Filtro de Mediana.....	42
3.7.5	Bloco de BLOB.....	43
3.7.6	Bloco de Contagem e Classificação de Veículos.....	44
3.7.7	Comunicação do Sistema.....	48
3.7.8	Funções Adicionais	49
3.8	Envio e Backup dos Dados.....	50
3.8.1	SmartIoT (AlticeLabs).....	51
3.8.2	Servidor na ESTGV	53
3.8.3	Registo de Dados Local	55
3.9	Consulta e Envio de Dados por Aplicação Móvel	56
4.	Validação e Testes	63
4.1	Hardware do Sistema.....	63
4.2	Software do Sistema.....	66
4.3	Aplicação Móvel	74
5.	Conclusão e Trabalhos Futuros.....	79
5.1	Conclusão	79
5.2	Trabalhos Futuros.....	80
	Referências.....	83
	Apêndice I – Configurações dos Blocos	89
	Anexo A – Informações dos Sensores	91

ÍNDICE DE FIGURAS

Figura 1-1: Média anual de concentrações de NO ₂ observadas em estações de tráfego.	3
Figura 2-1: Diagrama de um DAS.....	8
Figura 2-2: Libelium Air Quality Index IoT Vertical Kit.	9
Figura 2-3: Air Quality Egg.	10
Figura 2-4: Aeroqual AQM65.	11
Figura 2-5: Array of Things.	12
Figura 2-6: DecentLab Air Quality Station.	12
Figura 2-7: PurpleAir PA-II.	13
Figura 2-8: AirPi.....	13
Figura 2-9: Altitude Technology Sensly HAT.	14
Figura 2-10: Solução VT da MetroCount.....	17
Figura 2-11: Solução VP da MetroCount.	17
Figura 2-12: Solução VL da MetroCount.....	17
Figura 2-13: Solução SDR da Urban Aspect.....	18
Figura 3-1: DAS desenvolvido.	24
Figura 3-2: Sensor BME280.....	28
Figura 3-3: Sensor TSL2561.	28
Figura 3-4: Sensor SGP30.	29
Figura 3-5: Sensor MiCS-6814.....	31
Figura 3-6: Sensor ME2-O2-Φ20.....	32
Figura 3-7: Sensor DFR0034.....	32
Figura 3-8: Conversor ADS1115.....	33
Figura 3-9: Câmara Raspberry Pi V2.	34
Figura 3-10: Sistema Desenvolvido.	34
Figura 3-11: Diagrama do software desenvolvido para o Raspberry Pi.....	35
Figura 3-12: Exemplo de código para leitura do sensor.	37
Figura 3-13: Gráfico com 100 leituras do valor de NO ₂	37
Figura 3-14: Gráfico com as médias referentes ao número de leituras do valor de NO ₂	38
Figura 3-15: Funcionamento do <i>pipe</i>	39
Figura 3-16: Blocos do sistema desenvolvido.....	40
Figura 3-17: Aplicação de ROI a uma imagem.....	41
Figura 3-18: Detecção de primeiro plano aplicada.	42
Figura 3-19: Filtro de mediana aplicado.....	43
Figura 3-20: Valores das saídas do bloco de BLOB.	44
Figura 3-21: Pilhas invertidas, vazias, no instante $t = 0$	45
Figura 3-22: Pilhas invertidas preenchidas com dois objetos no instante $t = 1$	45
Figura 3-23: Pilhas invertidas preenchidas com três objetos no instante $t = 2$	46

ÍNDICE DE FIGURAS

Figura 3-24: Pilhas invertidas preenchidas com dois objetos no instante $t = 3$.	46
Figura 3-25: Pilhas invertidas preenchidas com três objetos no instante $t = 4$.	47
Figura 3-26: Diagrama resumo do bloco de classificação e contagem.	48
Figura 3-27: Diagrama resumo do bloco de comunicação.	49
Figura 3-28: Visualização de controlo de contagem.	50
Figura 3-29: Fluxograma de envio e <i>backup</i> dos dados.	51
Figura 3-30: Exemplo de uso da SmartIoT.	52
Figura 3-31: Autenticação da SmartIoT.	53
Figura 3-32: Endereço exemplo de envio dos dados.	53
Figura 3-33: Endereço exemplo de consulta dos dados.	53
Figura 3-34: Resposta do servidor com os dados em JSON.	54
Figura 3-35: Verificação para agregação de dados.	54
Figura 3-36: <i>Backup</i> dos dados em ficheiro.	55
Figura 3-37: Conteúdo de um ficheiro de <i>backup</i> .	55
Figura 3-38: Exemplo de código JSON.	56
Figura 3-39: Atividade principal da aplicação.	57
Figura 3-40: Atividade de consulta dos dados.	58
Figura 3-41: Formas de consulta dos dados.	60
Figura 3-42: Envio de avaliação da qualidade do ar.	61
Figura 4-1: Humidade relativa ao longo de 24 horas.	64
Figura 4-2: Luminosidade visível ao longo de 24 horas.	64
Figura 4-3: Problemas do sensor MiCS-6814.	65
Figura 4-4: CO ao longo de 24 horas.	65
Figura 4-5: Temperatura ao longo de 72 horas.	66
Figura 4-6: Tempo de aprendizagem baixo provoca má leitura do primeiro plano.	67
Figura 4-7: Peões a atravessar a faixa, não sendo contabilizados.	67
Figura 4-8: Bicicletas na via, não sendo contabilizadas.	68
Figura 4-9: Estacionamento paralelo à via, não sendo contabilizado.	69
Figura 4-10: Paragem na via, aguarda ser contabilizado.	70
Figura 4-11: Fila de trânsito parado, aguarda ser contabilizado.	71
Figura 4-12: Veículo ligeiro absorvido por veículo pesado.	72
Figura 4-13: Sistema cego sem conseguir identificar veículos.	73
Figura 4-14: Consulta em equipamento Android de 4 polegadas com API 22.	74
Figura 4-15: Envio de dados para o servidor com sucesso.	75
Figura 4-16: Atividade principal vista em tablet na horizontal.	75
Figura 4-17: Gráfico de barras visto em tablet na horizontal.	76
Figura 4-18: Gráfico de linhas visto em tablet na horizontal.	76
Figura 4-19: Consulta de agregação horária com o tablet na vertical.	77
Figura A-1: Acesso aos endereços de memória do BME280.	91
Figura A-2: Acesso aos endereços de calibração do BME280.	92
Figura A-3: Acesso aos endereços de memória do TSL2561.	93

Figura A-4: Acesso aos endereços de memória do SGP30.	94
Figura A-5: Acesso aos endereços de memória do MiCS-6814.....	95
Figura A-6: Acesso aos endereços de memória do MiCS-6814 (continuação).....	96
Figura A-7: Acesso aos endereços de memória da ADS1115.....	97
Figura A-8: Acesso aos endereços de memória da ADS1115 (continuação).....	98

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

Tabela 2-1: Resumo dos sistemas existentes.....	14
Tabela 2-2: Análise dos sistemas existentes de monitorização de tráfego.	19
Tabela 3-1: Análise dos microcomputadores.	24
Tabela 3-2: Características do Raspberry Pi 3 B+.....	25
Tabela 3-3: Parâmetros analisados e respetivos sensores.....	26
Tabela 3-4: Sensores utilizados e respetivos endereços.	36
Tabela 3-5: Número de leituras e respetiva média.	38
Tabela 4-1: Testes ao sistema de contagem de veículos, por vídeo.	73

ÍNDICE DE TABELAS

ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i>
API	Interface de Programação de Aplicação
BLE	<i>Bluetooth Low Energy</i>
BLOB	<i>Binary Large Object</i>
CE	Comissão Europeia
CPU	Unidade Central de Processamento
CSI	<i>Camera Serial Interface</i>
CSV	<i>Comma-Separated Values</i>
DAS	<i>Data Acquisition System</i>
ESTGV	Escola Superior de Tecnologia e Gestão de Viseu
FIFO	<i>First In, First Out</i>
Gás C ₂ H ₅ OH	Etanol (ou C ₂ H ₆ O)
Gás C ₃ H ₈	Propano
Gás C ₄ H ₁₀	Butano
Gás CH ₄	Metano
Gás CO	Monóxido de Carbono
Gás CO ₂	Dióxido de Carbono
Gás H ₂	Hidrogénio
Gás H ₂ S	Sulfureto de Hidrogénio
Gás NH ₃	Amoníaco
Gás NO	Monóxido de Azoto
Gás NO ₂	Dióxido de Azoto
Gás NO _x	Óxidos de Azoto
Gás O ₂	Oxigénio
Gás O ₃	Ozono
Gás SO ₂	Dióxido de Enxofre
GPIO	<i>General Purpose Input/Output</i>
GPU	Unidade de Processamento Gráfico
I ² C	<i>Inter-Integrated Circuit</i>
IEEE	Instituto de Engenheiros Eletrotécnicos e Eletrónicos
ISO	Organização Internacional de Normalização
JSON	<i>JavaScript Object Notation</i>
LAN	<i>Local Area Network</i>
LDR	Fotorresistência
NB-IoT	<i>Narrowband Internet of Things</i>
OMS	Organização Mundial de Saúde
REST	<i>Representational State Transfer</i>

ABREVIATURAS E SIGLAS

RGB	Vermelho, Verde e Azul
ROI	Região de Interesse
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SoC	<i>System-on-a-Chip</i>
SPI	<i>Serial Peripheral Interface</i>
TVOC	Compostos Voláteis Orgânicos Totais
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
UE	União Europeia
USB	<i>Universal Serial Bus</i>

1. Introdução

Em resultado das inúmeras inovações tecnológicas que temos vindo a assistir, também as cidades evoluíram e estão cada vez mais a tornarem-se cidades inteligentes. Na maioria dos casos o processo iniciou-se por coisas tão simples como a iluminação pública ou a rega dos jardins, mas a cada dia que passa, os sistemas tornam-se cada vez mais completos e complexos com uma monitorização e controlo de um número maior de parâmetros.

O presente trabalho descreve o processo de desenvolvimento de um sistema de monitorização de tráfego rodoviário e parâmetros ambientais como temperatura, humidade relativa, pressão atmosférica, dióxido de carbono, entre outros.

O sistema desenvolvido, apresentado no presente documento, procura, de alguma forma, servir para um aumento da “inteligência” das cidades, permitir a um desportista aceder à aplicação e consultar as zonas que se encontram menos poluídas para a prática de desporto, consciencializar os cidadãos para uma mudança no paradigma do uso dos transportes pessoais, auxiliar as autarquias na gestão de recursos e consciencializar para a redução da poluição atmosférica.

Neste capítulo serão apresentados os aspetos que motivaram e levaram à elaboração do presente documento, os objetivos de estudo e, a finalizar, o resumo de cada um dos capítulos deste trabalho.

1.1 As Cidades Inteligentes e o Ambiente

Embora o conceito de Cidade Inteligente não seja recente, só nesta última década se tem feito notar com maior expressão. Tem-se vindo a assistir a projetos de cidades inteligentes em todo o mundo, tendo como génese a utilização de tecnologias de informação e comunicação para promover a competitividade económica, a sustentabilidade ambiental e a qualidade de vida (INTELI, 2012).

Saber que existe uma rotura numa conduta de água, o local onde a mesma rotura ocorreu e a quantidade de água que se está a desperdiçar era uma tarefa que, num passado não muito distante, poderia demorar dias ou semanas. Atualmente, quase em tempo real, é possível saber da existência da rotura, do local aproximado da mesma e as perdas daí resultantes. A gestão do recurso hídrico tornou-se mais eficiente e otimizou todo o processo que envolve o transporte de água até cada um dos municípios de uma cidade.

Utilizando como exemplo as condutas de água e fazendo um paralelismo para as vias de tráfego, rapidamente podemos concluir que as mesmas podem também gozar de alguma da inteligência das cidades ao serviço dos cidadãos.

Zonas de maior poluição em resultado de um maior fluxo de tráfego, podem servir de alerta para alteração de hábitos dos cidadãos, servir de impulsionador para um maior uso de transportes não poluentes ou para a alteração de hábitos, como por exemplo, partilha de carro ou uso de transportes públicos nos centros urbanos.

Para além das utilidades já referidas de um sistema desta natureza, há ainda a destacar o facto de existir um indicador da Organização Internacional de Normalização (International Organization for Standardization – ISO) ISO 37122:2019 referente às Cidades Inteligentes que avalia o número de estações de monitorização da qualidade do ar em tempo real por quilómetro quadrado (ISO 37122:2019, 2019).

“A poluição atmosférica afeta a nossa saúde e o nosso ambiente. Provém principalmente da indústria, dos transportes, da produção de energia e da agricultura.” (Ohliger, 2018). Foi esta a citação que desencadeou toda a pesquisa e trabalho desenvolvido que serão abordados ao longo deste documento.

De acordo com uma estimativa da Organização Mundial de Saúde (2018), a poluição do ar exterior é a principal causa de morte de 4,2 milhões de pessoas por ano. O que se traduz num número realmente preocupante e que merece a atenção de todos.

De acordo com Silva e Mendes (2006), o tráfego rodoviário é a principal fonte de emissão de poluentes para o ar nos meios urbanos, sendo que as concentrações de partículas, o dióxido de

azoto (NO_2) e o monóxido de carbono (CO) encontram-se com valores mais elevados nas imediações das vias de maior tráfego. Ainda de acordo com um relatório da Agência Europeia do Ambiente (2018), o transporte rodoviário é uma das principais fontes de emissão de CO.

De acordo com a mais recente estatística da Eurostat (2016), Portugal encontra-se no segundo lugar, à frente da Lituânia, no que diz respeito ao uso do automóvel como meio de transporte principal.

Segundo um relatório do Observatório Nacional das Doenças Respiratórias (2018), em Portugal a qualidade do ar é em geral boa, muito embora exista um número significativo de dias em que a qualidade do ar é média ou mesmo fraca.

Ainda de acordo com o mesmo relatório, a Comissão Europeia (CE) recordou que nas zonas do Porto, Braga e Lisboa a concentração de NO_2 está acima da recomendada pelas normas da União Europeia (UE), aconselhando alterações na política de transportes urbanos de forma a reduzir as emissões de óxidos de azoto (NO_x) (monóxido de azoto (NO) e NO_2) e ozono (O_3). Na Figura 1-1 podem ser observados os valores médios de NO_2 nas estações de tráfego da Europa.

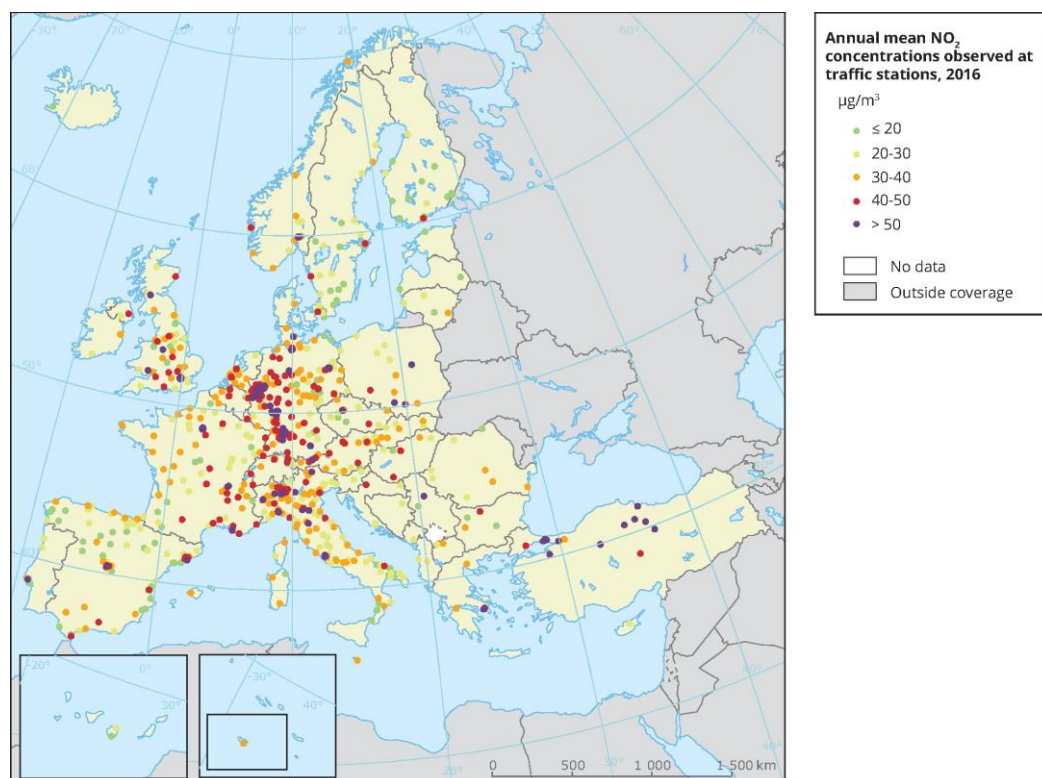


Figura 1-1: Média anual de concentrações de NO_2 observadas em estações de tráfego (Retirado de (European Environment Agency, 2018)).

1.2 Objetivos

O presente trabalho tem como objetivo principal o desenvolvimento de um sistema de monitorização ambiental e tráfego rodoviário de baixo custo, para aplicação no âmbito das cidades inteligentes.

O sistema a desenvolver constitui uma solução completa de monitorização ambiental e de tráfego rodoviário, que inclui todo o hardware e software necessários.

Esta solução possibilita não só a aquisição, mas também, o envio dos dados para uma base de dados, com o objetivo de fazer uma consulta em tempo real ou posteriormente, através de uma aplicação móvel para Android.

A consulta e a visualização dos dados poderão ser realizadas em modo de texto e/ou em modo gráfico.

Com o objetivo de tornar o sistema comercialmente viável, o mesmo deve ser de fácil instalação e uso, com um baixo custo e facilmente escalável, tanto em relação aos sensores como ao sistema total final.

O desenvolvimento da solução pretendida, envolve as seguintes tarefas principais:

- Levantamento dos sensores, câmara, unidade principal e restante hardware necessário;
- Definição dos parâmetros ambientais a medir;
- Desenvolvimento do hardware;
- Desenvolvimento do software de controlo da unidade principal;
- Desenvolvimento da aplicação móvel para sistema Android;
- Instalação do sistema desenvolvido na Escola Superior de Tecnologia e Gestão de Viseu (ESTGV);
- Testes com o sistema desenvolvido implementado.

1.3 Estrutura e Organização do Documento

O presente documento está organizado em 5 capítulos. Seguidamente, enuncia-se para cada um deles a sua finalidade e contributo.

No primeiro capítulo é feita a abordagem a alguns conceitos introdutórios, seguindo-se os objetivos associados à elaboração do documento. Por último, é apresentada uma breve descrição da estrutura e organização adotada.

O capítulo 2 aborda os principais constituintes de um sistema de aquisição de dados, onde são também apresentados e estudados alguns dos diferentes sistemas já existentes.

No capítulo 3 estão presentes os tópicos relativos ao sistema de aquisição de dados desenvolvido, desde a leitura dos sensores ao processamento dos dados lidos, processamento de imagem utilizado, envio para um servidor e *backup* dos dados. Neste capítulo será também feita uma abordagem à aplicação móvel desenvolvida.

O quarto capítulo é reservado ao teste e validação dos resultados obtidos no processo de teste do sistema, que foi efetuado. Neste capítulo, são validados os dados obtidos pelo sistema desenvolvido.

No quinto e último capítulo, é apresentada a conclusão do trabalho realizado, assim como as perspectivas para trabalhos futuros.

2. Sistemas de Monitorização Ambiental e de Tráfego

A aquisição de dados é algo que ultrapassa a existência Humana, uma vez que todo esse processo não é mais do que captar através de sensores um determinado parâmetro e converter o mesmo para sinal elétrico, a fim de ser processado.

Neste capítulo será feita uma abordagem aos sistemas de aquisição de dados com maior incidência para os sistemas de monitorização ambiental e os sistemas de monitorização de tráfego disponíveis comercialmente. Serão feitas algumas comparações com vantagens e desvantagens, assim como as características de alguns dos sistemas disponíveis comercialmente.

2.1 Sistemas de Aquisição de Dados

O corpo humano tem vários sensores, como por exemplo na audição, onde as ondas de um sinal acústico são convertidas para um sinal elétrico, sinal esse enviado pelos neurónios até ao nosso sistema de processamento, o cérebro.

O sistema de aquisição de dados (DAS) é o conjunto do sensor/transdutor, que converte o sinal físico em sinal elétrico, e a unidade principal de processamento (microprocessadores, microcomputadores, etc.). Pode ter ainda outros constituintes, para além dos referidos, conforme Figura 2-1.

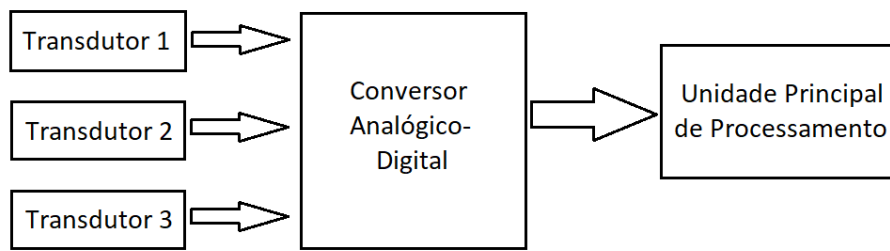


Figura 2-1: Diagrama de um DAS.

De acordo com Keithley Instruments (2001), embora o conceito de aquisição de dados possa ser difícil de definir completamente, a maioria dos utilizadores de computador, engenheiros e cientistas, concorda que existem vários elementos comuns:

- Um computador usado para programar os equipamentos de teste e manipular ou armazenar dados, também pode ser usado para suporte de funções, como produção de relatórios ou gráficos em tempo real. Pode não estar necessariamente em controlo constante do equipamento de aquisição de dados ou mesmo manter-se conectado ao equipamento de aquisição de dados permanentemente;
- O equipamento de teste pode consistir em placas *plug-in* de aquisição de dados, placas externas ou sensores.

Os DAS são equipamentos usados para converter, numa primeira fase, os valores reais em sinais elétricos e, posteriormente, estes em valores numéricos digitais, que podem ser armazenados ou não, para análise e tratamento futuro.

Um sistema de aquisição de dados pode ser constituído por um conjunto de componentes principais:

- Sensores/transdutores que fazem a leitura e conversão de valores reais em sinais elétricos;
- Condicionadores de sinal que colocam o sinal elétrico recebido do sensor/transdutor num sinal adequado para o conversor analógico-digital (ADC). Este conversor transforma o sinal elétrico condicionado num valor de sinal digital (Sankar, Devaraju, Kumar e Sudharshan, 2017) que representa o valor real medido. O valor digital obtido é depois enviado, através de uma interface de comunicação, para um computador ou algum dispositivo de armazenamento de dados.

Para além do já referido hardware, faz parte do DAS o software de *driver*, que permite que o software de aplicação interaja com um dispositivo, e de aplicação, que facilita a interação entre o computador e o utilizador na análise e apresentação dos dados.

2.2 Sistemas de Monitorização Ambiental

Quando se fala de estações de monitorização ambiental existe um conjunto de nomes que se destacam, entre os quais está a empresa espanhola Libelium (Libelium, 2019), conhecida por desenvolver continuamente soluções para as cidades inteligentes, onde a monitorização ambiental se inclui.

A solução da Libelium tem a designação de Air Quality Index IoT Vertical Kit, (Libelium - Air Quality Index IoT Vertical Kit, 2019), apresentando um conjunto de sensores: temperatura, humidade relativa do ar, pressão atmosférica, CO, O₃, NO₂, dióxido de enxofre (SO₂) e partículas, podendo ser adquiridos separadamente outros sensores de entre o catálogo da Libelium, estando, no entanto, limitado o número de entradas disponíveis. A referida solução pode ser vista na Figura 2-2.



Figura 2-2: Libelium Air Quality Index IoT Vertical Kit (Adaptado de (Libelium - Air Quality Index IoT Vertical Kit, 2019)).

Para a interligação e leitura dos valores dos sensores, a solução dispõe de um bloco denominado de Waspote Plug & Sense, (Libelium - Waspote Plug & Sense, 2019).

No conjunto vendido pela Libelium estão ainda incluídos um painel solar para alimentar o sistema, o *gateway* e os acessórios de fixação. O valor proposto no catálogo da empresa é de 5 350,00€.

A Air Quality Egg, (Air Quality Egg, 2019) é uma empresa que disponibiliza uma solução *open source* para a medição de parâmetros ambientais, dispondo de soluções para interior e exterior.

A unidade exterior, ao centro na Figura 2-3, incorpora, além do bloco de interligação e leitura dos valores dos sensores, o *gateway* que permite, por exemplo, estabelecer uma ligação Wi-Fi para envio dos dados para uma *cloud*.



Figura 2-3: Air Quality Egg (Adaptado de (Air Quality Egg, 2019)).

O valor base deste equipamento, segundo o site da marca, é de 140,00€, incluindo apenas a medição de temperatura, humidade relativa do ar e pressão atmosférica. Os restantes sensores são adquiridos separadamente até um máximo de três.

Cada sensor extra tem um custo que varia entre os 35,00€ para o sensor de compostos voláteis orgânicos e os 70,00€ para o sensor de dióxido de carbono (CO₂), existindo ainda os sensores de SO₂, NO₂, O₃, CO e partículas.

A Aeroqual é uma empresa que se impõe pela qualidade e pela variedade dos sensores disponíveis, sendo que o modelo da Figura 2-4, o AQM 65, (AQM 65, 2019) conta com os seguintes: temperatura, humidade relativa do ar, pressão atmosférica, ruído, CO₂, CO, sulfureto de hidrogénio (H₂S), NO₂, NO_x, O₃, SO₂, compostos voláteis orgânicos e partículas.



Figura 2-4: Aeroqual AQM65 (Adaptado de (Aeroqual, 2019)).

O custo de aquisição poderá variar entre os 18 000,00€ e os 44 000,00€, sendo que por este valor, além dos sensores, a unidade já conta com sistema de comunicação, interligação e controlo, tendo ainda instalado no seu interior uma unidade de aquecimento/arrefecimento para permitir o funcionamento em pleno nos locais onde tal se torna necessário.

A Array of Things, (Array of Things, 2019) é um projeto interessante, contando com um sistema bastante completo e compacto, podendo ser facilmente instalado num poste de iluminação pública.

No que diz respeito aos sensores, pode ter os seguintes: temperatura, humidade relativa do ar, pressão atmosférica, ruído, NO₂, O₃, CO, H₂S e SO₂, estando previsto incluir mais sensores para além dos já disponíveis.

A grande vantagem relativamente aos sistemas já referidos reside na câmara que pode ser instalada e que permite uma monitorização do volume de tráfego, embora ainda não esteja disponível e se encontre em fase de avaliação, esta pode ser vista na parte inferior da Figura 2-5.

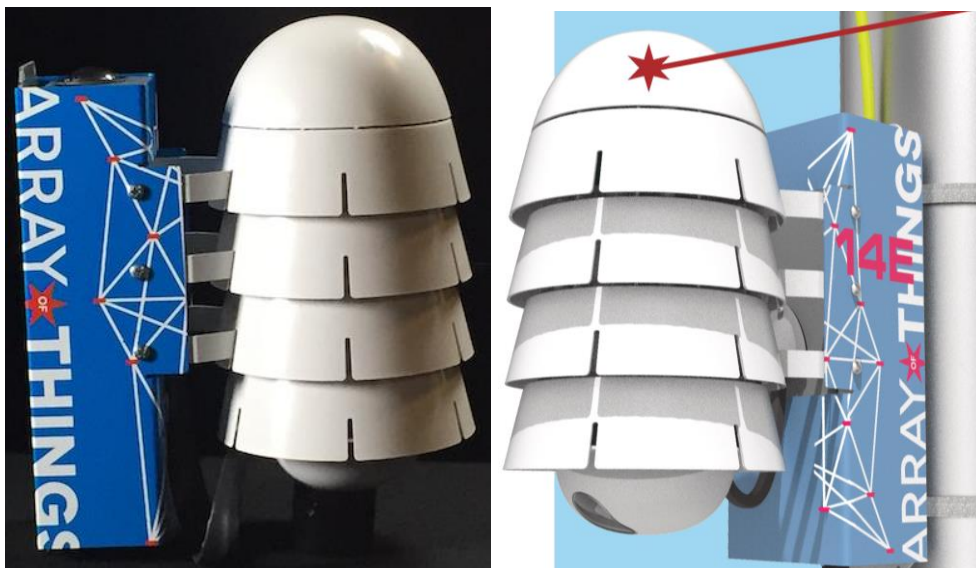


Figura 2-5: Array of Things (Adaptado de (Array of Things, 2019)).

O custo de aquisição de uma unidade completa não é revelado, uma vez que ainda está em fase experimental. Há previsão de receber mais funcionalidades e a implementação de novos sensores, o que pode influenciar o valor. Uma vez que permite selecionar os componentes e sensores de forma individual, isso também se refletirá no valor de aquisição.

A DecentLab, (DecentLab, 2019) apresenta uma solução, conforme Figura 2-6, que permite monitorizar a temperatura, a humidade relativa do ar, o NO₂, o NO, o CO, o H₂S, o O₃ e o SO₂, com a transmissão dos dados a poder ser feita por LoRaWAN. O custo de aquisição de cada unidade poderá variar entre os 1 800,00€ para um sistema mais básico e os 3 500,00€ para um sistema mais completo.



Figura 2-6: DecentLab Air Quality Station (Retirado de (DecentLab - Air Quality Station, 2019)).

A PurpleAir, (PurpleAir, 2019) tem um sistema com um custo de aquisição baixo, quando comparado com as propostas anteriores, no entanto, apenas tem sensores capazes de medir temperatura, humidade relativa do ar, pressão atmosférica e partículas, estando, por isso, este sistema muito limitado, embora conte com uma ligação Wi-Fi para envio dos dados. A solução descrita pode ser vista na Figura 2-7.



Figura 2-7: PurpleAir PA-II (Retirado de (PurpleAir, 2019)).

Na divisão mais económica, encontram-se o sistema AirPi, (AirPi, 2019) e o sistema Sensly HAT, (Altitude, 2019). Ambos necessitam de um Raspberry Pi para funcionarem, uma vez que será este a unidade principal de interligação.

Relativamente ao AirPi, na Figura 2-8, é um equipamento desenvolvido para monitorizar a temperatura, a humidade relativa do ar, a pressão atmosférica, a luminosidade, o ruído, o CO e o NO₂. Dispõe ainda da possibilidade de ligar outros sensores, através das 8 entradas analógicas. O custo de aquisição é de 80,00€ aos quais há ainda que adicionar o custo do Raspberry Pi.



Figura 2-8: AirPi (Adaptado de (AirPi PCB, 2019)).

O Sensly HAT da Altitude Technology pode ser adquirido por 100,00€ na versão base com 3 sensores de gases, aos quais devem ser somados o custo do sensor de temperatura, humidade relativa do ar e pressão atmosférica pretendido (entre 16,00€ e 24,00€) e ainda o custo do Raspberry Pi. Na Figura 2-9 é possível ver a solução instalada num Raspberry Pi.



Figura 2-9: Altitude Technology Sensly HAT (Adaptado de (Altitude Tech, 2019)).

2.2.1 Análise aos Sistemas de Monitorização Ambiental

Anteriormente foram referidas algumas das empresas que se encontram no mercado com soluções para a monitorização de parâmetros ambientais, a Tabela 2-1 procura resumir algumas e as suas características. Não foram avaliadas todas as soluções disponíveis, uma vez que algumas não disponibilizam qualquer informação online ou apenas desenvolvem soluções à medida do cliente, não fazendo qualquer referência aos sistemas em termos de características e custo de aquisição.

Tabela 2-1: Resumo dos sistemas existentes.

Sistema	Sensores	Personalizável	Custo
Libelium Air Quality Index IoT Vertical Kit	Temperatura, Humidade, Pressão, CO, O ₃ , NO ₂ , SO ₂ e Partículas	Com sensores e módulos proprietários	5 350,00€
Air Quality Egg	Temperatura, Humidade e Pressão	Com sensores proprietários (compostos voláteis orgânicos, CO ₂ , SO ₂ , NO ₂ , O ₃ , CO e Partículas)	140,00€
Aeroqual AQM65	Temperatura, Humidade, Pressão, Ruído, CO ₂ , CO, H ₂ S, NO ₂ , NO _x , O ₃ , SO ₂ , compostos voláteis orgânicos e Partículas	Com módulos proprietários (comunicação, aquecimento e arrefecimento)	18 000,00€ (base)

Sistema	Sensores	Personalizável	Custo
Array of Things	Temperatura, Humidade, Pressão, Ruído, NO ₂ , O ₃ , CO, H ₂ S e SO ₂	Com sensores e módulos proprietários (novos sensores e câmara para monitorização do volume de tráfego)	Não revelado (em fase de avaliação)
DecentLab Air Quality Station	Temperatura, Humidade, NO ₂ , NO, CO, H ₂ S, O ₃ e SO ₂	Com módulos proprietários (LoRaWAN)	1 800,00€ (base)
PurpleAir PA-II	Temperatura, Humidade, Pressão e Partículas	Não	205,00€
AirPi	Temperatura, Humidade, Pressão, Luminosidade, Ruído, CO e NO ₂	Totalmente personalizável	80,00€ (sem o valor do Raspberry Pi)
Altitude Technology Sensly HAT	Temperatura, Humidade, Pressão	Com sensores proprietários ou compatíveis com o Raspberry Pi	100,00€ (sem o valor do Raspberry Pi)

As várias soluções presentes na Tabela 2-1 podem ser divididas em dois grupos pelo valor de aquisição, as de baixo custo e as de custo mais elevado. Aquelas que se apresentam com um maior custo de aquisição são também as mais completas e que permitem uma maior personalização, muito embora tal seja conseguido com equipamentos proprietários.

Ficar dependente de servidores de terceiros ou de sensores proprietários, com custos de aquisição mais elevados e com uma oferta mais reduzida, poderá traduzir-se numa grande desvantagem a médio-longo prazo, no entanto, a curto prazo torna-se uma solução mais vantajosa, por não apresentar risco de incompatibilidades e requerer menos configurações.

Relativamente às soluções com um custo de aquisição mais baixo, as soluções da AirPi e da Altitude Technology são as que se apresentam mais personalizáveis. Os sistemas mais personalizáveis são também aqueles onde se podem encontrar maiores incompatibilidades entre os sensores e onde também são requeridas mais configurações, o que se pode traduzir numa desvantagem para quem procure um sistema de fácil e rápida instalação.

Como se pode concluir, um sistema que se aplique de forma perfeita a um caso, poderá não ser a melhor solução para outro, como tal, para cada situação, devem ser analisadas todas as soluções disponíveis.

2.3 Sistemas de Monitorização de Tráfego

Um maior volume de tráfego rodoviário estará diretamente ligado a uma maior poluição do ar ou pelo menos a uma degradação do ar comparativamente à não presença de veículos, para um mesmo local num mesmo instante.

Perceber de que forma é que os veículos circulam poderá dar uma mais adequada ideia de que forma os níveis de poluição podem ser diminuídos ou condicionados. Sabemos que algumas das horas do dia terão, em condições normais, um volume de tráfego superior. Por exemplo, as horas do início e final da manhã e início e final da tarde em condições habituais e durante a semana terão um volume de tráfego consideravelmente superior ao volume registado de madrugada.

Ter o registo de parâmetros ambientais e a evolução destes ao longo do dia e com uma análise direta ao volume e tipo de tráfego rodoviário, será útil para perceber de que forma os cidadãos estão expostos aos poluentes atmosféricos emitidos pelos veículos (Rakowska et al., 2014).

Existem várias soluções para a monitorização do tráfego rodoviário, umas tecnologicamente mais evoluídas que outras, umas mais eficazes do que outras, no entanto, a divisão é feita em dois tipos, os intrusivos e os não intrusivos. Importa então perceber as soluções que existem e qual a tecnologia mais adequada em cada situação.

Ver alguém com um caderno e uma caneta junto a uma estrada a fazer a contagem de veículos parece algo do passado, embora possa ser uma solução para um pequeno período de tempo, até mesmo para servir de comparativo para os sistemas tecnologicamente mais avançados, contudo, não é desejável ter uma contagem manual durante um ano, um mês ou até mesmo uma semana.

Para reduzir o número de situações como a descrita, existem no mercado algumas soluções, mas mesmo assim, algumas destas ainda recorrem a um processo manual à posteriori para fazer o registo dos valores para um suporte perdurável.

A contagem de veículos por tubo pneumático instalado na via é uma dessas soluções e tem sido o método mais popular durante estes anos. Para além do referido tubo, tem instalado um contador, mecânico ou eletromecânico. Um contador mecânico necessitará sempre que alguém faça a leitura do valor do contador e passe o mesmo para um outro suporte, papel ou informático. Já o contador eletromecânico pode guardar os dados na memória interna ou enviar para algum servidor através de um módulo de comunicação. A MetroCount apresenta uma dessas soluções com o modelo VT, conforme Figura 2-10, (MetroCount RoadPod VT, 2019), entre outras empresas como a Diamond Traffic (Diamond Traffic Road Tube, 2019) e a Urban Aspect (Urban Aspect eTube, 2019).



Figura 2-10: Solução VT da MetroCount (Retirado de (MetroCount RoadPod VT, 2019)).

Outro tipo de sistema utilizado recorre a um sensor piezoelétrico. Quando um veículo atravessa o local causa uma deformação, o sensor piezoelétrico mede essa deformação permitindo fazer a contagem do veículo e mesmo a sua classificação, uma vez que um veículo ligeiro causará uma deformação inferior a um veículo pesado. O modelo VP da MetroCount apresenta-se como uma das soluções no mercado (MetroCount RoadPod VP, 2019). Podem ser vistas na Figura 2-11 as marcas do sensor no asfalto e a caixa com a componente eletrónica.



Figura 2-11: Solução VP da MetroCount (Adaptado de (MetroCount RoadPod VP, 2019)).

O sensor indutivo é outra das formas de contagem de veículos, contudo, por necessitar de estar debaixo do alcatrão, nem sempre é escolhido, uma vez que caso a estrada já exista, obriga a obras que podem ser evitadas comparativamente com outras formas de contagem. Será o mais adequado para sistemas definitivos e concebidos antes da construção da via. A Diamond Traffic e a MetroCount têm soluções com este tipo de sensor, a primeira com dois modelos (Diamond Traffic Inductive Loop Counters, 2019) e a última com o modelo VL (MetroCount RoadPod VL, 2019). Na Figura 2-12 podem-se observar as marcas dos sensores instalados na via.



Figura 2-12: Solução VL da MetroCount (Retirado de (MetroCount RoadPod VL, 2019)).

Detetar as variações do campo magnético é uma forma de contagem de veículos utilizada, recorrendo a um sensor magnético. Tal como o sensor indutivo, também pode estar embebido no asfalto, não sendo obrigatório que seja instalado desta forma.

O sensor acústico deteta energia acústica ou sons audíveis produzidos pelo tráfego de veículos e da interação dos pneus de um veículo com o pavimento. Quando um veículo passa através da zona de deteção, existe um aumento da energia sonora e é gerado um sinal da sua presença. Quando o veículo deixa a zona de deteção, o nível de energia do som cai abaixo do limiar de deteção.

Pela temperatura que os veículos irradiam, também é possível fazer a contagem do seu número recorrendo a um sensor de infravermelhos passivo.

Os sensores de micro-ondas de Doppler e radar são uma solução para a contagem de veículos, pois emitem um sinal e fazem a leitura da reflexão do mesmo. Quando existe um veículo em movimento, a frequência do sinal refletido será diferente. Desta forma, o sensor consegue detetar a presença de um veículo. A Urban Aspect tem comercialmente disponível um sistema, o SDR (Urban Aspect SDR Traffic, 2019), conforme Figura 2-13.



Figura 2-13: Solução SDR da Urban Aspect (Adaptado de (Urban Aspect SDR Traffic, 2019)).

A deteção por sensores de ultrassons é também uma solução para a contagem de veículos, trabalhando a uma frequência entre 25 kHz e 50 kHz, o que é acima da gama audível por humanos. A maioria dos sensores de ultrassons opera com formas de onda de pulso e possibilita a contagem de veículos.

Outra das formas de contagem de veículos é feita com sensores de radar a laser, transmitindo energia no espectro infravermelho, podendo mesmo, fazer a contagem de mais do que uma via de trânsito.

A deteção de veículos por processamento de imagem é outra das soluções para a contagem e classificação dos mesmos. Estes sistemas conseguem detetar os veículos em várias direções, algo que nem todos os sistemas permitem, apenas necessitando que a câmara instalada consiga captar a imagem das várias direções.

Ao nível do processamento tem mais exigências do que qualquer outro sistema, sendo que não tem grandes requisitos de estrutura, não necessitando de alterações no pavimento nem de estar muito próximo da via de trânsito.

Para além da câmara, é necessário um computador e associar ao mesmo uma licença com o software que permite a contagem dos veículos. A Roadometry tem disponível um software capaz disso, o VTC (Roadometry VTC, 2019), com um custo de 2 020,00€ para uma licença de um computador.

2.3.1 Análise aos Sistemas de Monitorização de Tráfego

Como foi possível perceber no ponto 2.3, a variedade de sistemas de contagem e veículos é vasta, cada um com as suas vantagens e desvantagens. A Tabela 2-2 pretende fazer essa análise de forma sucinta.

Tabela 2-2: Análise dos sistemas existentes de monitorização de tráfego (Federal Highway Administration, 2016), (Leduc, 2008).

Forma de Contagem	Vantagens	Desvantagens
Manual	<ul style="list-style-type: none"> • Pode ser usado para validação das restantes formas de contagem; • Muito portátil; • Facilidade de contagem em cruzamentos e entroncamentos. 	<ul style="list-style-type: none"> • Dispendioso; • Impreciso para contagens de maior duração.
Tubo Pneumático	<ul style="list-style-type: none"> • Uso comum para contagem e classificação em aplicações portáteis. 	<ul style="list-style-type: none"> • Instalação pode exigir encerramento temporário das faixas; • Não funciona bem em elevado volume de tráfego ou trânsito lento ou parado.
Sensor Piezoelétrico	<ul style="list-style-type: none"> • Alta precisão na classificação dos veículos; • Insensível ao mau tempo como chuva, nevoeiro e neve; • Uso comum para contagem e classificação. 	<ul style="list-style-type: none"> • A instalação requer corte do pavimento; • A instalação inadequada diminui a vida útil do pavimento; • Instalação e manutenção requerem encerramento da via; • Piezo sensível à temperatura; • Não funciona bem em trânsito lento ou parado.

Forma de Contagem	Vantagens	Desvantagens
Sensor Indutivo (Loop)	<ul style="list-style-type: none"> • Design flexível para satisfazer grande variedade de aplicações; • Grande base de experiência; • Fornece outros indicadores básicos de tráfego (por exemplo, o volume, a velocidade, etc.); • Insensível ao mau tempo como chuva, nevoeiro e neve; • Fornece melhor precisão para a contagem dados em comparação com outras técnicas vulgarmente utilizadas; • Uso comum para a obtenção de medições precisas; • Modelos de excitação de alta frequência com processamento de sinal avançado podem fornecer dados de classificação. 	<ul style="list-style-type: none"> • A instalação requer corte do pavimento; • A instalação inadequada diminui a vida útil do pavimento; • Instalação e manutenção requerem encerramento da via; • Múltiplos loops geralmente necessários; • A precisão da deteção pode diminuir quando é requerida a deteção de uma grande variedade de classes de veículos.
Sensor Magnético	<ul style="list-style-type: none"> • Insensível ao mau tempo tais como a neve, chuva e nevoeiro; • Alguns modelos são instalados sob o pavimento sem necessidade de cortes no pavimento. 	<ul style="list-style-type: none"> • A instalação requer corte do pavimento; • A instalação inadequada diminui a vida útil do pavimento; • Instalação e manutenção requerem encerramento da via; • Modelos com zonas de deteção pequenas podem exigir várias unidades para deteção total da via; • Não é possível detetar veículos parados.
Sensor Acústico	<ul style="list-style-type: none"> • Deteção passiva; • Insensível à precipitação; • Operação para várias faixas disponível em alguns modelos. 	<ul style="list-style-type: none"> • As temperaturas frias podem afetar a contagem com precisão; • Alguns modelos não são compatíveis com veículos em filas; • Não é possível detetar veículos parados.
Infravermelhos Passivo	<ul style="list-style-type: none"> • Possível obter a velocidade com sensores de múltiplas zonas; • Instalação não-intrusiva. 	<ul style="list-style-type: none"> • Sensor passivo pode perder sensibilidade ao veículo sob forte chuva, neve e nevoeiro denso; • Requer uma limpeza periódica da lente.
Radar e Doppler Micro-ondas	<ul style="list-style-type: none"> • Tipicamente insensível a influências atmosféricas; • Medição direta da velocidade. 	<ul style="list-style-type: none"> • Não é possível detetar veículos parados.

Forma de Contagem	Vantagens	Desvantagens
Ultrassons	<ul style="list-style-type: none"> • Com capacidade de operação de múltiplas faixas. 	<ul style="list-style-type: none"> • As condições ambientais, tais como mudança de temperatura e turbulência do ar extrema podem afetar o desempenho (compensação da temperatura é possível); • Não é possível detetar veículos parados.
Radar laser	<ul style="list-style-type: none"> • Transmite múltiplos feixes para medição precisa da posição do veículo, a velocidade e classe; • Com capacidade de operação de múltiplas faixas; • Instalação não-intrusiva. 	<ul style="list-style-type: none"> • A operação pode ser afetada por nevoeiro quando visibilidade é fraca ou por neve; • Instalação e manutenção, incluindo limpeza periódica da lente; • Deteção difícil com vias com separador central alto.
Deteção por processamento de imagem de vídeo	<ul style="list-style-type: none"> • Fácil instalação, fácil manutenção, fácil classificação de veículos, medição de distâncias e de velocidades; • Fácil monitorizar múltiplas vias e várias zonas de deteção; • Fácil de adicionar e modificar zonas de deteção; • Podem ser obtidos vários dados do tráfego (velocidade, por exemplo); • Geralmente com custo económico quando são necessárias muitas zonas de deteção, dentro do campo de visão da câmara. 	<ul style="list-style-type: none"> • Instalação e manutenção, incluindo limpeza periódica da lente, podem exigir encerramento da via, quando a câmara está montada sobre o pavimento (encerramento das faixas pode não ser necessário quando a câmara está montada na lateral da faixa de rodagem); • Desempenho afetado pelo mau tempo tais como neblina, chuva e neve, assim como sombras de veículos, obstrução e ausência de iluminação noturna.

Pela análise à Tabela 2-2, pode-se concluir que são muitas as alternativas para sistemas de monitorização e tráfego, todas elas com pontos positivos e pontos menos positivos. Nem todas as soluções são adequadas para um mesmo caso, seja por questões climáticas onde a neve esteja muitas vezes presente, seja por motivos de manutenção onde o encerramento da via provocará enormes constrangimentos do trânsito, seja por razões de ordem económica.

Para o caso concreto de uma cidade, onde é previsível a existência de trânsito lento ou com filas paradas, as soluções com uso de tubo pneumático ou sensor piezoelétrico não serão adequadas. Para o mesmo caso, também serão de excluir as soluções que obrigam a cortes no pavimento ou que o mesmo seja asfaltado, uma vez que é comum as cidades terem estradas em calçada, o que inviabiliza a instalação de sistemas com sensores indutivos ou magnéticos. De entre as restantes soluções que podem ser aplicadas numa cidade, a deteção por processamento de imagem de vídeo é a que se apresenta mais flexível, uma vez que permite

de forma rápida alterar zonas de deteção, sentidos de trânsito e excluir ou adicionar tipos de veículos na contagem.

3. Sistema Desenvolvido

Para dar resposta à falta de soluções de baixo custo para monitorização de parâmetros ambientais e de tráfego rodoviário facilmente personalizáveis, e com a possibilidade de integração com servidores não proprietários para uma melhor interligação entre as cidades inteligentes e a monitorização da qualidade do ar e tráfego, foi desenvolvido o sistema de aquisição em tempo real de dados que a seguir se apresenta. Além de permitir a interligação com câmaras e variados tipos de sensores, não se encontra refém de um servidor proprietário e sem possibilidade de personalização.

Neste capítulo será abordado o sistema de aquisição de dados desenvolvido, os seus constituintes, a forma como os mesmos interagem e a sua influência em cada parte do sistema. Para além do hardware, será abordada de forma resumida a parte de software e programação feitas.

3.1 Introdução

Neste trabalho fazem parte do DAS quatro componentes:

- Os sensores e a câmara que têm a função de ler os valores reais, parâmetros ambientais (sensores) e imagem (câmara), convertendo estes valores em sinais digitais. Neste sistema os ADC estão aplicados nos sensores.
- O computador a funcionar como unidade principal, recebe os sinais digitais por barramento I²C no caso dos sensores e CSI para a câmara. Depois de receber os sinais, converte os mesmos em dados reais para envio para o servidor/base de dados.
- O servidor/base de dados tem como função receber os dados enviados pela unidade principal, guardando-os e disponibilizando um conjunto de ferramentas e de acesso aos mesmos.

- A aplicação móvel traduz os valores lidos em grafismo, tendo como função ler os valores do servidor e fazer a representação dos mesmos facilitando a sua leitura por parte do utilizador.

A Figura 3-1 representa as ligações entre os vários componentes do sistema desenvolvido. Onde o bloco dos sensores é constituído por todos os sensores ambientais utilizados, ligados por I²C ao computador, o bloco da câmara, ligado por CSI, capta a imagem que permitirá ao computador fazer a contagem e classificação dos veículos. Ao computador está destinada a tarefa de converter os dados dos sensores para valores numéricos e classificar e contar os veículos. Estes dados são depois enviados para o servidor/base de dados. Na base de dados encontram-se guardados e para consulta todos os valores enviados pelo computador, assim como os enviados pela aplicação móvel. A aplicação móvel terá como função estabelecer a ligação entre o utilizador desta e a base de dados, permitindo uma consulta dos valores armazenados.

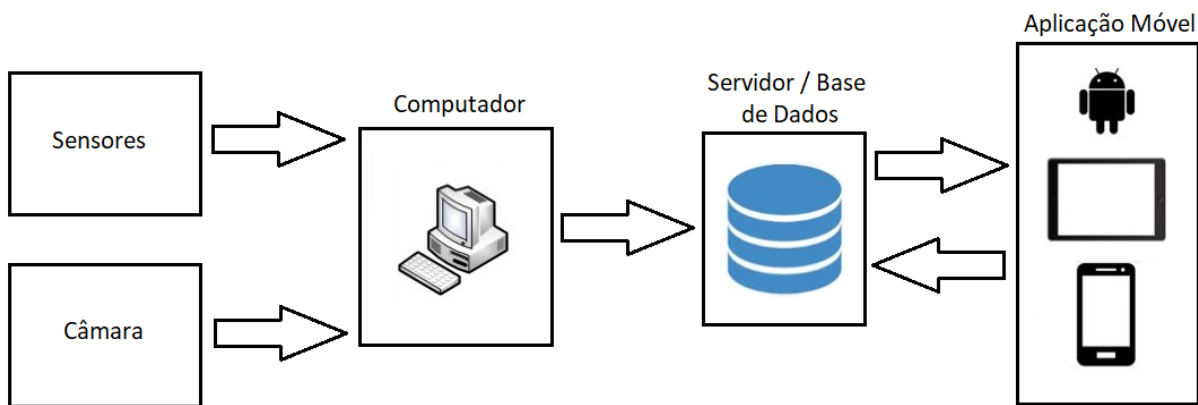


Figura 3-1: DAS desenvolvido.

3.2 Computador

Cada vez mais presentes nos dias de hoje, os microcomputadores acabam por conjugar numa só placa um conjunto de componentes dos DAS. São exemplo destes microcomputadores, o Raspberry Pi, o Arduino e o BeagleBone. O primeiro acaba por ser dos mais económicos e aquele em que existe maior partilha e apoio da comunidade online (Ferdoush, 2014). A Tabela 3-1 resume algumas das características dos três microcomputadores referidos anteriormente.

Tabela 3-1: Análise dos microcomputadores.

	Raspberry Pi 3 B+	Arduino Uno Rev3	BeagleBone
Custo	37,00€	20,00€	60,00€
Memória	MicroSD	32 KB	4 GB + MicroSD
GPIO	40	14	69
Entrada analógica	Não	6	7

	Raspberry Pi 3 B+	Arduino Uno Rev3	BeagleBone
I²C	Sim	Sim	Sim
Ethernet	Sim	Não	Sim
Wi-Fi	Sim	Não	Não
USB	4	0	1

Para o desenvolvimento do DAS foi utilizado como microcomputador um Raspberry Pi 3 B+, com um sistema operativo baseado em Debian (Linux) ARM, o Raspbian. Este equipamento já apresenta alguns protocolos de comunicação para uso com sensores, nomeadamente I²C e SPI, assim como para a câmara CSI e/ou USB. Algumas das características deste microcomputador encontram-se resumidas na Tabela 3-2.

Tabela 3-2: Características do Raspberry Pi 3 B+.

SoC	Broadcom BCM2837B0
CPU	Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
GPU	Broadcom Videocore-IV
SDRAM	1GB LPDDR2
Armazenamento	Micro-SD
Rede	Ethernet 330Mbit, 2.4GHz e 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
Periféricos	40 pinos GPIO 4 x USB 2.0 CSI camera port DSI display port
Alimentação	5V/2,5A DC

Todos estes motivos levaram a que fosse escolhido como microcomputador a usar, uma vez que os restantes tinham um custo de aquisição superior sem grandes vantagens ou um valor de compra inferior, contudo, necessitavam de módulos adicionais.

Segundo Rodrigues (2014) existem diversos protocolos de comunicação no mercado, tais como, UART, SPI, I²C e 1-Wire. O I²C é um protocolo de comunicação que foi desenvolvido pela divisão de semicondutores da Philips, no início da década de 80, este protocolo apresenta inúmeras vantagens, entre as quais se destacam (Veronesi, 2005):

- O endereçamento e protocolo de transferência de dados totalmente definido via software;
- A possibilidade de inclusão ou exclusão de dispositivos no barramento sem afetá-lo ou outros dispositivos conectados a este;

- O diagnóstico de falhas extremamente simples. O mau funcionamento é imediatamente detetado.

A interligação de vários sensores (*slaves*) e várias unidades principais (*masters*) é feita de forma bastante simples, através da partilha do barramento de comunicação composto pelo *Serial Data* (SDA) e pelo *Serial Clock* (SCL), uma vez que cada dispositivo tem um endereço de 7 bits (Veronesi, 2005).

3.3 Medição de Parâmetros Ambientais

Para a implementação do sistema, cujo objetivo é medir parâmetros ambientais, foi necessário adquirir sensores, tendo sido dada atenção aos que foram escolhidos. Eram condições, o baixo custo de aquisição e determinados fornecedores.

Optou-se por sensores compatíveis com o Raspberry Pi ou que de alguma forma fosse possível assegurar a sua compatibilidade, recorrendo, por exemplo, a conversores.

Uma vez que o Raspberry Pi não tem entradas analógicas nativamente, optou-se pelo uso da ADS1115 que, desta forma, permitiu ligar dois sensores analógicos ao Raspberry Pi, como se analisará mais à frente. Na Tabela 3-3 encontram-se reunidos os parâmetros analisados, a unidade de medida, os sensores utilizados e o respetivo endereço.

Tabela 3-3: Parâmetros analisados e respetivos sensores.

Parâmetro	Unidades	Sensor	Endereço
Temperatura	°C	BME280	0x77
Humidade relativa	%		
Pressão atmosférica	hPa		
Luz total	lx	TSL2561	0x39
Luz não visível	lx		
Dióxido de Carbono (CO ₂)	ppm	SGP30	0x58
Compostos voláteis orgânicos (TVOC)	ppm		
Monóxido de Carbono (CO)	ppm	MiCS-6814	0x04
Dióxido de Azoto (NO ₂)	ppm		
Amoníaco (NH ₃)	ppm		
Propano (C ₃ H ₈)	ppm		
Butano (C ₄ H ₁₀)	ppm		
Metano (CH ₄)	ppm		
Hidrogénio (H ₂)	ppm		
Etanol (C ₂ H ₅ OH)	ppm		
Oxigénio (O ₂)	ppm	ADS1115 – ME2 - O2	0x48 – A1
Ruído (Sensor Analógico Som)	dB	ADS1115 – DFRobot	0x48 – A0

3.3.1 Sensor de Temperatura, Humidade e Pressão

O valor da temperatura não influencia diretamente a qualidade do ar, no entanto, é um dado importante uma vez que alguns sensores necessitam de ser compensados em função da temperatura do ar.

Tendo em conta os valores de temperatura que se verificam habitualmente em Portugal e considerando os extremos registados no país (-16 °C e +47,3 °C), considerou-se que um sensor com uma cobertura de temperaturas entre os -15 °C e os +50 °C seria adequado (IPMA Extremos Climatológicos, 2019).

Tal como é referido, a temperatura não influencia diretamente a qualidade do ar, mas sim indiretamente. É facilmente perceptível o porquê desta influência, já que no inverno, o uso de equipamentos de combustão para aquecimento de águas e edifícios, assim como um maior uso de transportes, origina uma maior produção de gases poluentes.

Tal como a temperatura, a humidade relativa do ar não influencia a qualidade do mesmo, no entanto, é um dado importante uma vez que a concentração de alguns gases (como por exemplo o Ozono) aumenta com menores valores de humidade (e maiores de luminosidade).

O valor da pressão atmosférica tem influência na medição da qualidade do ar, uma vez que em situações de elevadas pressões os gases poluentes tendem a concentrar-se mais junto ao solo.

Em Portugal, o valor mais baixo de pressão atmosférica registado é de 931 hPa (Taborda, 2012), sendo que o máximo foi de 1050,3 hPa (Boletim Climatológico Sazonal IPMA, 2015).

Tendo em conta que os valores mínimos e máximos a registar não serão muito diferentes dos extremos registados até à presente data, foram analisadas algumas hipóteses de sensores que conseguissem dar resposta.

Alguns dos sensores disponíveis, tais como o BME280, o BMP280, o DHT22, o SHT10, entre outros, não tinham capacidade de medir os três parâmetros, temperatura, humidade relativa e pressão atmosférica, o que obrigaria a ter pelo menos dois sensores.

Optou-se pelo sensor BME280, conforme Figura 3-2, uma vez que o mesmo se apresenta como capaz de medir temperaturas entre os -40 °C e os 85 °C, humidades relativas entre 0% e 100% e pressões atmosféricas entre 300 hPa e 1100 hPa. Tem ainda como grande vantagem o protocolo I²C já estar instalado, que tornaria toda a implementação mais simples (Bosch BME280, 2019).



Figura 3-2: Sensor BME280 (Adaptado de (Adafruit BME280, 2019)).

3.3.2 Sensor de Luminosidade

A luminosidade não é um parâmetro ambiental que afeta diretamente a qualidade do ar, sabe-se, no entanto, que promove a alteração da constituição dos gases na atmosfera. As plantas através da fotossíntese são exemplo disso, onde o Oxigênio (O₂) é produzido com recurso ao CO₂ que é consumido (Moreira, 2013).

Entre as várias alternativas para medir a luminosidade, existiam a foto-resistência (LDR), o sensor BH1750 e o TSL2561. A LDR foi excluída pela baixa resolução e comportamento não linear. Optou-se pelo TSL2561 dado que o tem uma ampla faixa dinâmica de lux (de 0,1 lx a 40000 lx), permitindo ainda medir separadamente infravermelho e espectro total, uma vez que possui na sua constituição dois díodos (SparkFun TSL2561, 2019).

Este sensor, que se encontra na Figura 3-3, tem ainda a vantagem de ter implementado o protocolo I²C, o que torna a comunicação e implantação mais simples.

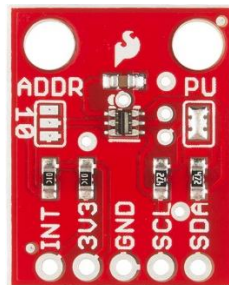


Figura 3-3: Sensor TSL2561 (Retirado de (SparkFun TSL2561, 2019)).

3.3.3 Sensor de CO₂ e TVOC

De acordo com Comissão Europeia (2019) em 2016, o tráfego automóvel foi responsável por 21% das emissões de CO₂ da União Europeia, um dos principais gases com efeito de estufa que contribui diretamente para o aquecimento global, traduzindo-se num aumento da

temperatura que, de forma indireta, pode trazer consequências à saúde humana (Soares, 2015).

Atendendo à grande diversidade de Compostos Orgânicos Voláteis existente (tendo sido identificados mais de 900), à sua variabilidade e capacidade de interação com os outros poluentes, torna-se inviável a sua medição e determinação individual, pelo que o parâmetro geralmente medido é o Total de Compostos Orgânicos Voláteis (TVOC) (Coelho, 2014).

De acordo com Mishra (2015) o tráfego é dos maiores contribuidores para as emissões de compostos voláteis orgânicos, particularmente dos veículos a gasolina e a evaporação de combustíveis.

O sensor SGP30, na Figura 3-4, embora não tenha capacidade de medir diretamente o CO₂, consegue-o fazer de forma indireta, desta forma o valor é referido como eqCO₂, ou seja, CO₂ equivalente obtido com base nos níveis de Hidrogénio (H₂) (Adafruit SGP30, 2019). Relativamente aos VOC, como já foi referido, é feita a medição total, não se sabendo o valor de cada composto.

O parâmetro objetivo era o CO₂ e como tal existiam outras alternativas como, por exemplo, os sensores SEN0159 e SEN0220. No entanto, a saída dos mesmos era analógica, o que obrigaria a um ADC. O SGP30 tem algumas vantagens como o I²C implementado e o custo de aquisição mais baixo. A precisão dos sensores é muito próxima, 50 ppm + 5% do valor lido para o SEN0220 e 15% para o SGP30 (Adafruit SGP30, 2019).



Figura 3-4: Sensor SGP30 (Adaptado de (Adafruit SGP30, 2019)).

3.3.4 Sensor de CO, NO₂, NH₃, C₃H₈, C₄H₁₀, CH₄, H₂ e C₂H₅OH

O CO é um gás inodoro e incolor. Os seus maiores níveis de concentração devem-se à queima de combustíveis, sendo nas cidades o tráfego rodoviário (Guri, 2016) o principal responsável.

Trata-se de um gás que no ser humano origina problemas cardiovasculares e do sistema nervoso, podendo provocar morte por envenenamento em concentrações extremas (Guri, 2016).

O NO₂ é um gás facilmente detetável pelo odor, muito oxidante e corrosivo. Tal como outros gases já abordados, nos dias de hoje, a sua origem resulta essencialmente da queima de combustíveis, onde, mais uma vez, se incluem muitos dos meios de transporte (Kuriyama, Moreira e Silva, 1997).

Quando atinge concentrações elevadas pode provocar problemas respiratórios ou acentuar os já existentes, especialmente em grupos de risco como idosos e crianças. Este gás é também um dos responsáveis pela existência das chuvas ácidas que corroem edifícios e elementos naturais (Kuriyama et al., 1997).

Este é um dos parâmetros referidos pela ISO 37120:2018 que foi considerado para análise pelo sistema desenvolvido (ISO 37120:2018, 2018).

O Amoníaco (NH₃) é um gás muito utilizado em equipamentos de refrigeração, embora seja facilmente absorvido pelas plantas. Em elevadas concentrações deste gás as consequências podem ser graves para os humanos, uma vez que se trata de um gás extremamente irritante e corrosivo (Agency for Toxic Substances and Disease Registry, 2004).

A principal fonte deste gás é a agricultura (Oenema, Velthof, Amann, Klimont e Winiwarter, 2012) pelo que não foi objetivo principal considerar a medição do mesmo, mas uma vez que o sensor tinha essa capacidade, foi feita a leitura.

O Metano (CH₄) é um gás incolor e quando se encontra no ar com alguma concentração, torna-se altamente inflamável. A sua produção é essencialmente causada pela agricultura, embora também esteja associada ao tráfego automóvel em percentagem insignificante (Nam, Jensen e Wallington, 2004).

O Hidrogénio, sendo um elemento químico (H), geralmente apresenta-se sob a forma molecular (H₂). Nos dias de hoje tem sido muito associado aos veículos como combustível, e não, como resultado do uso de outros combustíveis.

É um gás que resulta mais do uso de veículos ligeiros do que pesados, estando relacionado com um maior uso da gasolina pelos ligeiros e maior uso do gasóleo pelos pesados (K. Vollmer et al., 2007).

Propano (C₃H₈), Butano (C₄H₁₀) e Etanol (C₂H₅OH ou C₂H₆O), embora sejam diferentes, são três dos vários VOC existentes (Szulczyński e Gębicki, 2017).

O sensor MiCS-6814 é responsável pela medição de todos os parâmetros referidos anteriormente, importando salientar que foi escolhido pela medição do CO e do NO₂. Todos os restantes parâmetros são lidos, uma vez que o fabricante disponibiliza a função de leitura dos mesmos (Grove Multichannel Gas Sensor, 2019), embora não fosse o objetivo principal.

Entre as alternativas ao MiCS-6814 estavam o MiCS-2714, o MiCS-4514 e o TGS5042-A00. Optou-se pelo MiCS-6814, conforme Figura 3-5, por já se encontrar comercialmente disponível com o protocolo I²C implementado, sendo que nos restantes sensores obrigaria a acrescentar um ADC.

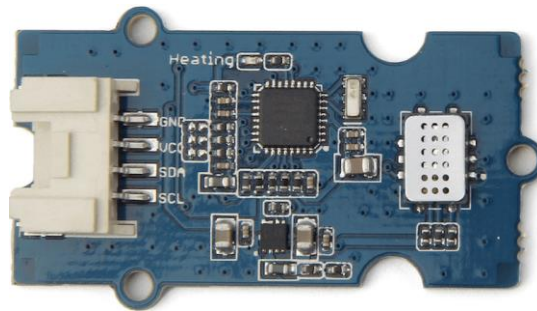


Figura 3-5: Sensor MiCS-6814 (Retirado de (Grove Multichannel Gas Sensor, 2019)).

3.3.5 Sensor de O₂

O O₂ é o segundo maior constituinte da atmosfera terrestre, sendo um gás de extrema importância para a existência de vida e utilizado pelos motores de combustão presentes nos veículos.

Com uma maior afluência de trânsito, verifica-se que os níveis de oxigénio diminuem, essencialmente devido à elevada concentração de monóxido de carbono (N. A. S. Martins, 2018).

Para a medição deste gás existiam algumas alternativas, por exemplo, o sensor SK-25F e o ME2-O2-Φ20. Não estava disponível qualquer alternativa com o protocolo I²C, sendo aqueles dois sensores com saída analógica. Optou-se pelo ME2-O2-Φ20, representado na Figura 3-6, por uma questão de disponibilidade.

Uma vez que a saída deste sensor é analógica, tornou-se obrigatório o uso de um ADC, de forma a que Raspberry Pi consiga fazer a leitura dos valores transmitidos pelo sensor. Este ADC será abordado no ponto 3.3.7.



Figura 3-6: Sensor ME2-O2-Φ20 (Retirado de (Grove Oxygen Sensor, 2019)).

3.3.6 Sensor de Ruído

O ruído é definido como sendo um som desagradável ou indesejável para o Ser Humano. O som gerado pelo tráfego nas cidades é exemplo disso e é um problema que vem sendo considerado como uma das prioridades na definição de políticas de garantia das condições de bem-estar e saúde pública (M. M. de A. Martins, 2015), sendo um dos indicadores referido no ponto ambiente e alterações climáticas da ISO 37120:2018 (ISO 37120:2018, 2018).

O ruído pode ter efeitos de natureza diversa e intensidade variável sobre a saúde da população exposta. Estes efeitos, normalmente adversos, podem-se classificar segundo o tipo de repercussão que apresentam no organismo, como efeitos sobre o aparelho auditivo e efeitos não auditivos (Carvalho e Rocha, 2008).

Para dar resposta à medição deste parâmetro, optou-se pelo uso do sensor DFR0034 (Analog Sound Sensor V2), representado na Figura 3-7, um sensor de som de saída analógica e que por este motivo foi necessário interligar um ADC que será abordado no ponto 3.3.7.



Figura 3-7: Sensor DFR0034 (Adaptado de (DFRobot Analog Sound Sensor, 2019)).

3.3.7 Conversor Analógico-Digital (ADC)

Conforme foi referido anteriormente, o uso de ADC torna-se necessário para interligar alguns sensores com determinadas placas controladoras. Para o presente caso, o ADC foi necessário para interligar os sensores ME2-O2- Φ 20 e DFR0034 com o Raspberry Pi. Os restantes sensores usam conversores já implementados pelos fabricantes.

Existiam algumas alternativas para o ADC a usar, como por exemplo, o ADS1015, o ADS1115 ou o ADC1256. Optou-se pelo ADS1115, uma vez que é um conversor de baixa potência, com 16 bits de resolução, com interface I²C (Texas Instruments ADS1115, 2019) e com um baixo custo de aquisição. Na Figura 3-8 é possível observar a ADS1115 com as respetivas entradas analógicas, de A0 a A3.

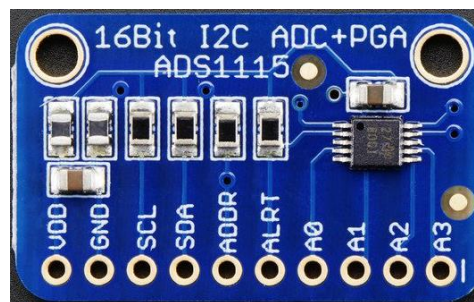


Figura 3-8: Conversor ADS1115 (Adaptado de (Adafruit ADS1115, 2019)).

3.4 Câmara

A relação entre o número e tipo de veículos e a alteração de parâmetros ambientais poderá ser importante para encontrar uma correlação entre estes. Para fazer a classificação e a contagem do número de veículos existiam várias alternativas, já referidas no ponto 2.3.1, com as vantagens e desvantagens de cada um dos sistemas.

Pelas vantagens que apresentava, optou-se por fazer a contagem e classificação de veículos por processamento de imagem, com desenvolvimento do respetivo algoritmo.

Antes do processamento de imagem é necessário que esta seja captada por um sensor. Para tal, recorreu-se a uma câmara com um custo relativamente baixo e que dava garantias de resposta ao problema. Assim, a câmara utilizada foi a Raspberry Pi Camera Module V2, conforme a Figura 3-9, utiliza Interface Série para Câmaras (CSI), sendo garantida a total compatibilidade com o Raspberry Pi utilizado.

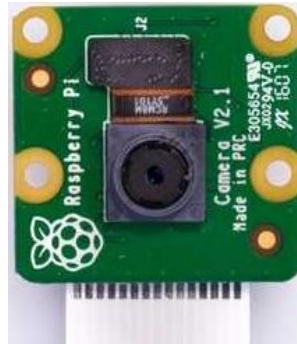


Figura 3-9: Câmera Raspberry Pi V2 (Adaptado de (Raspberry Pi Camera, 2019)).

3.5 Detalhes do Sistema Desenvolvido

Após a apresentação de todo o hardware, importa referir alguns detalhes do sistema desenvolvido e a forma como o mesmo é ligado. A ligação à rede de internet é feita por ethernet, podendo ser utilizada em alternativa a ligação por Wi-Fi. A comunicação com a câmara é feita pelo barramento CSI. Os sensores fazem ligação direta por I²C, exceção feita para o sensor de O₂ e de ruído que ligam por intermédio do ADC (ADS1115) que por sua vez liga por I²C. A Figura 3-10 representa as ligações descritas anteriormente.

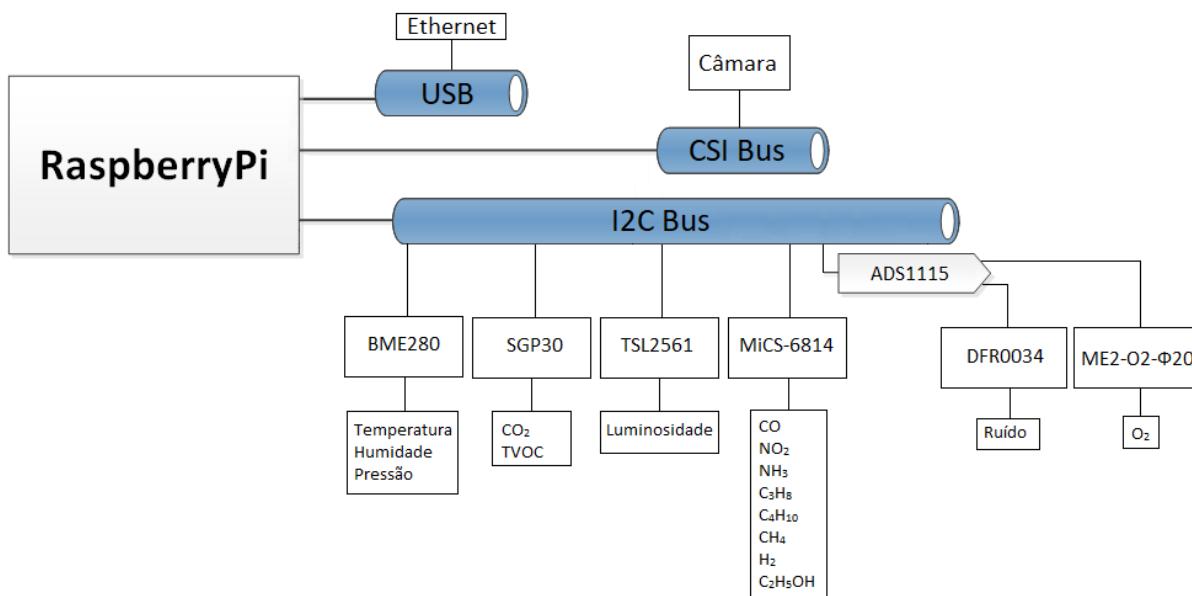


Figura 3-10: Sistema Desenvolvido.

3.6 Processamento de Dados

Depois do sistema se encontrar definido ao nível dos parâmetros a medir, dos sensores a utilizar e das funções que o mesmo tem de realizar, seguiu-se o desenvolvimento da parte de

software destinada ao Raspberry Pi. O diagrama da Figura 3-11 representa o funcionamento sequencial do software desenvolvido para o Raspberry Pi.

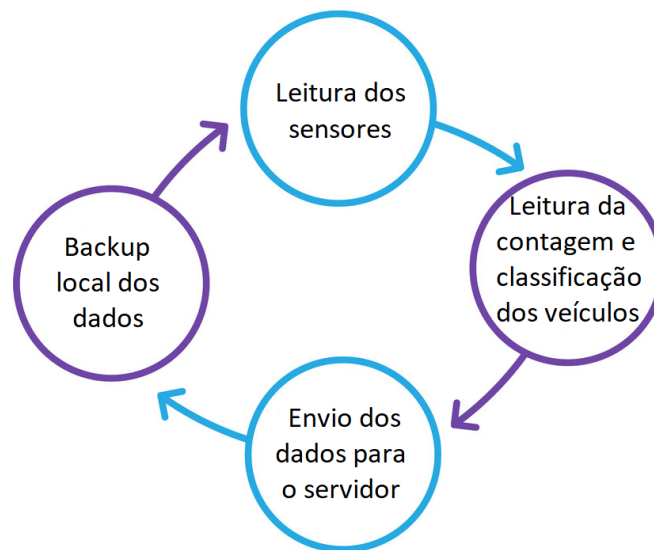


Figura 3-11: Diagrama do software desenvolvido para o Raspberry Pi.

Por ser utilizado protocolo I²C para interligação do Raspberry Pi com os sensores e o ADC, a escolha da linguagem de programação não foi objeto de grande detalhe na seleção e aquisição dos equipamentos já referidos. Assim, no início de configuração dos sensores, recorreu-se ao MATLAB/Simulink, sendo uma forma simples de interligação e a escolha certa por ser utilizado para o processamento de imagem na contagem e classificação de veículos.

Com os sensores já a funcionar de forma separada, passou-se à fase seguinte e começaram-se a juntar em blocos, de forma a poder-se testar o seu funcionamento em conjunto, embora o protocolo I²C não levantasse qualquer objeção nesse sentido, seria mais fácil perceber se existiria algum problema e identificar o mesmo.

Quando praticamente já todos os sensores estavam a funcionar, restava juntar apenas um, o MiCS-6814, que segundo o fabricante (Grove Multichannel Gas Sensor, 2019), teoricamente era compatível com o Raspberry Pi, embora não fosse disponibilizada de forma oficial qualquer biblioteca ou código de demonstração. Este sensor já estava a funcionar de forma isolada e, por isso, não mostrava qualquer indício da sua não compatibilidade, mas após feitas todas as ligações e quando foi tudo junto ao mesmo bloco, começaram a surgir as incompatibilidades, pois quando o código referente a este sensor estava inserido, deixava de ser possível obter leituras dos valores lidos pelo sensor.

Optou-se então por alterar a linguagem principal de programação do sistema, escolhendo-se Python, uma vez que alguns dos sensores tinham sido inicialmente testados nesta linguagem de programação, o que permitiria ganhar algum tempo e seria em princípio uma garantia de

funcionamento do sistema. Recorreu-se à biblioteca `smbus` mas, após diversas abordagens, o sensor MiCS-6814 não permitia fazer leituras dos parâmetros. Era detetado pelo Raspberry Pi, sendo possível fazer-se a alteração do endereço I²C, uma vez que se pensava que pudesse ser esta a causa. Através de pesquisas, tentou-se perceber se existia já código Python desenvolvido para o sensor em questão, chegando-se à existência de código desenvolvido por uma empresa, a Dexter Industries, que utiliza este mesmo sensor para uso com a placa de interface desenvolvida por eles, a GrovePi. No entanto, esse código Python também não estava a funcionar e tal era referido pela empresa a 15 de abril de 2016 (DexterInd/GrovePi, 2016).

Uma vez que todas as abordagens anteriores não tinham resultado, optou-se por alterar para linguagem C, a forma de leitura dos valores do sensor MiCS-6814, tendo o mesmo respondido sem qualquer problema. Tal facto está relacionado com a forma de ler valores do sensor, uma vez que neste sensor a forma de leitura é diferente de todos os restantes sensores usados. A leitura deste sensor é feita como se de um ficheiro se tratasse. Posto isto, decidiu-se que por uma questão de coerência, faria todo o sentido alterar-se os restantes sensores para linguagem C, o que será abordado com mais pormenor no ponto 3.6.1.

3.6.1 Leitura dos Dados

Como já foi referido anteriormente, a linguagem C não era a que estava inicialmente definida, mas por problemas com um sensor noutras linguagens e por esta se ter revelado totalmente compatível com os sensores, optou-se por ser a escolhida.

O protocolo I²C facilita o uso dos sensores e, por estar muito difundido, especialmente na comunidade do Raspberry Pi, existe um grande número de bibliotecas a que se pode recorrer, tal como a `WiringPi` (WiringPi, 2019). Cada sensor tem o respetivo endereço I²C. No presente trabalho foram utilizados cinco endereços conforme Tabela 3-4.

Tabela 3-4: Sensores utilizados e respetivos endereços.

Dispositivo	Endereço
BME280	0x77
TSL2561	0x39
SGP30	0x58
MiCS-6814	0x04
ADS1115	0x48

Antes de se proceder à leitura dos sensores, é habitual que se tenha que enviar para os mesmos um determinado valor, depois disso, devolvem uma resposta e deve ser enviado um novo valor para se começar a fazer a leitura dos parâmetros disponibilizados pelo sensor. Conforme o exemplo da Figura 3-12 demonstra, onde 0x20 é referente ao endereço do sensor

e para se fazer a leitura do valor é necessário escrever o valor 0x03 no registo 0x30. Em relação à leitura, a mesma é feita no endereço 0x3A. Para o sensor MiCS-6814, a função `wiringPiI2CReadReg8()` é substituída pela função `read()`, uma vez que, tal como já foi referido, não era possível fazer a leitura como nos restantes sensores.

```
sensor = wiringPiI2CSetup(0x20);
wiringPiI2CWriteReg8(sensor, 0x30, 0x03);
valor = wiringPiI2CReadReg8(sensor, 0x3A);
```

Figura 3-12: Exemplo de código para leitura do sensor.

Este exemplo pretende ser genérico, e como tal, devem ser consultadas as folhas de especificações dos sensores, uma vez que cada um deles tem determinados valores a serem escritos num registo específico. Por norma, tal é referido pelo fabricante, pois para além do valor a escrever, também é indicado o endereço para leitura dos dados e a respetiva conversão desses dados em informação recolhida pelos sensores.

Os sensores com registos da calibração ou os de múltiplos parâmetros são aqueles que apresentam fórmulas de cálculo mais elaboradas. Para alguns, foi necessário recorrer à biblioteca externa `math`, uma vez que de base a programação em C não dispunha de soluções de cálculo.

Antes de ser efetivado o envio dos dados, são feitas trinta leituras por minuto aos sensores, para que desta forma se tenha um valor mais próximo do valor real e diminuir o peso de uma possível falsa leitura ou de alguma contaminação instantânea que pudesse condicionar o valor da leitura. Os gráficos das Figura 3-13 e Figura 3-14 demonstram o efeito conseguido com trinta leituras por minuto, comparativamente com números de valores de leitura superiores e inferiores.

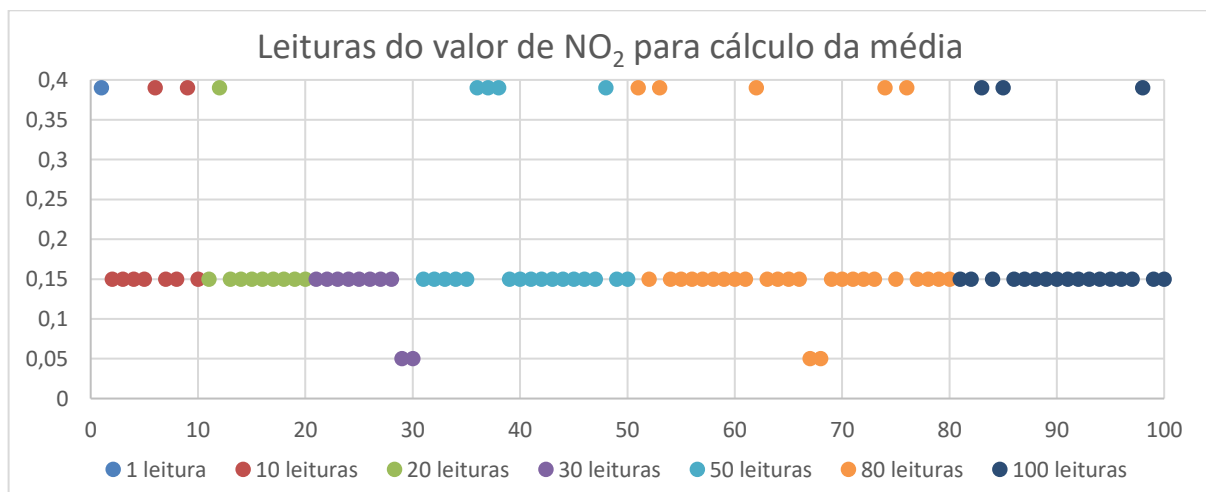
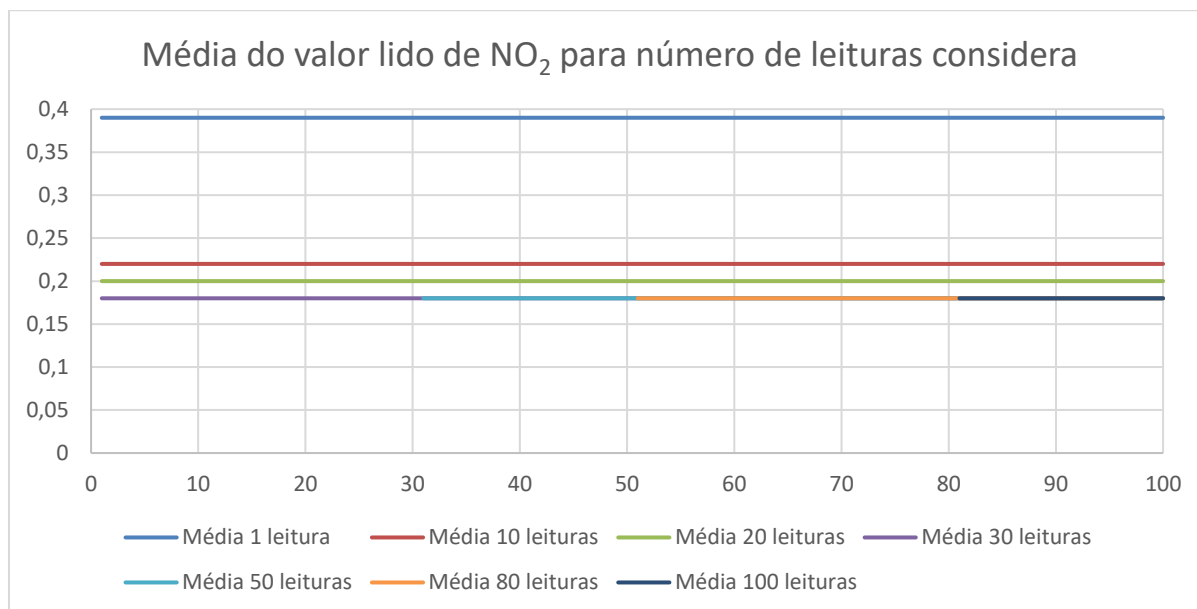


Figura 3-13: Gráfico com 100 leituras do valor de NO₂.

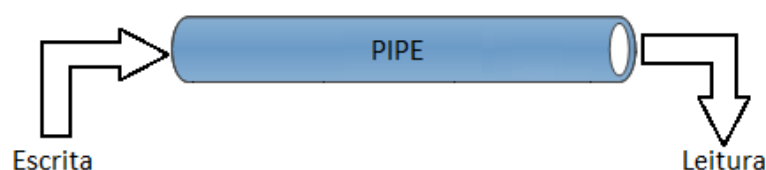
Depois de obtidos os valores de 100 leituras do valor de NO₂, foi calculada a média para cada caso, arredondada a duas casas decimais. Conclui-se que a diferença, após o arredondamento, era nula entre 30, 50, 80 ou 100 leituras, conforme se pode observar na Tabela 3-5 e na Figura 3-14.

Tabela 3-5: Número de leituras e respetiva média.

Número de Leituras	Média
1	0,39
10	0,22
20	0,20
30	0,18
50	0,18
80	0,18
100	0,18

Figura 3-14: Gráfico com as médias referentes ao número de leituras do valor de NO₂.

Uma vez que o sistema desenvolvido está continuamente a correr dois programas distintos, um relativo à recolha de parâmetros ambientais e outro relativo à contagem e classificação de veículos, tornou-se necessário desenvolver uma comunicação entre ambos. Inicialmente optou-se por utilizar um ficheiro intermédio, um *buffer*, onde os dados eram escritos, lidos, enviados e apagados. A solução era pouco eficiente e poderia levar a perda ou sobreposição de dados. Assim, fez-se uma outra abordagem ao problema, utilizando *named pipes* do tipo FIFO, primeiro a entrar, primeiro a sair (*first in, first out*). Desta forma o sistema tornar-se-ia mais eficiente e com riscos quase nulos de perda ou adulteração de dados. A Figura 3-15 representa um *pipe*, onde a escrita é feita pelo programa de processamento de imagem, já a leitura é realizada pelo programa dos sensores.

Figura 3-15: Funcionamento do *pipe*.

3.7 Processamento de Imagem

Conforme já foi referido anteriormente, fazer a contagem e a classificação dos veículos a circular era um dos pontos de interesse do presente trabalho. Assim, face às vantagens já apresentadas no ponto 2.3.1 optou-se por fazer a contagem e classificação dos veículos por processamento de imagem.

Depois de definido que a contagem e classificação de veículos seria por processamento de imagem, tornou-se necessário determinar de que forma o sistema iria ser implementado. Por uma questão de familiaridade com o processamento de imagem em MATLAB, optou-se por esta solução. A MathWorks é a empresa responsável por este poderoso software.

Para o desenvolvimento do processo de identificação e contagem de veículos, tornou-se necessário não só o software MATLAB/Simulink (MATLAB, 2019), como a *toolbox* de processamento de imagem e ainda o pacote gratuito de suporte para o Raspberry Pi (MATLAB Raspberry Pi Support, 2019).

Nos seguintes pontos deste capítulo, será feita uma explicação mais detalhada de todo o processo de desenvolvimento, assim como o modo de funcionamento do processamento de imagem na contagem e classificação de veículos.

Conforme já foi referido anteriormente, por uma questão de familiarização com o MATLAB em processamento de imagem, optou-se por este software para o desenvolvimento do sistema de contagem e classificação de veículos. Para além do já referido software, respetiva *toolbox* e pacote de suporte, são necessários uma câmara e um Raspberry Pi. O sistema pode funcionar num computador, no entanto, considerou-se que faria todo o sentido o mesmo ser implementado num Raspberry Pi, juntamente com o sistema de recolha de parâmetros ambientais, uma vez que torna o sistema mais eficiente, mais compacto e menos dependente de comunicações ou cablagens longas.

O sistema de contagem e classificação de veículos está desenvolvido de forma simples e por blocos. A Figura 3-16 é representativa do sistema desenvolvido com os blocos utilizados, numerados de 1 a 7 e que serão descritos mais à frente.

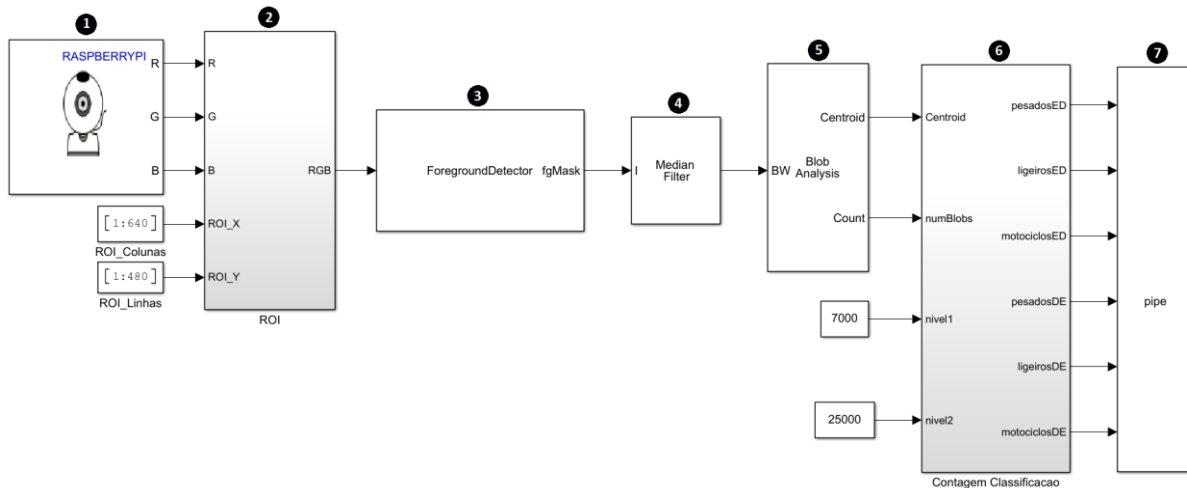


Figura 3-16: Blocos do sistema desenvolvido.

O primeiro bloco (1) diz respeito à captação de imagem pela câmara, sendo uma entrada do sistema com os dados em serem lidos em RGB. O bloco seguinte (2) é opcional, permitindo implementar uma região de interesse (ROI). Foi incluído para permitir eliminar zonas onde não é relevante analisar a imagem. No terceiro bloco (3) encontra-se a identificação do primeiro plano. Este bloco permite distinguir o que será o fundo, casas, árvores e alcatrão, do que são os objetos em primeiro plano, neste caso, veículos, pessoas, sombras entre outros. No bloco seguinte (4) é definido um filtro de mediana, tendo como objetivo definir melhor os contornos do objeto em primeiro plano relativamente ao fundo. O bloco subsequente (5) diz respeito à análise de parâmetros da imagem binária, tendo como função transmitir ao bloco seguinte a informação organizada dos objetos. Depois de recolhida a informação pelo bloco anterior, esta é recebida pelo bloco (6), que diz respeito à classificação e contagem dos veículos. O último bloco (7) é responsável pela comunicação com o sistema dos sensores e de envio para o servidor.

Ao longo do ponto 3.7 serão abordadas configurações nos blocos utilizados, as mesmas podem ser consultadas no Apêndice I do documento.

3.7.1 Bloco da Câmera para Raspberry Pi

O primeiro bloco do sistema desenvolvido é referente à câmara, a entrada principal do sistema. Este bloco faz parte do já referido pacote de suporte para o Raspberry Pi. Nele encontram-se definidos o nome do dispositivo, o tamanho da imagem em pixéis e o formato dos mesmos.

O nome do dispositivo é correspondente ao endereço do ficheiro da câmara no sistema operativo da origem. No presente caso corresponde ao 'video0' da pasta 'dev'. No que se refere ao tamanho da imagem, é possível utilizar vários tipos de resoluções, desde 160x120 a

800x600, existindo ainda a possibilidade de personalizar a resolução a usar. Este ponto tem alguma importância no desempenho do sistema, uma vez que um maior número de pixels significa um maior esforço do processamento mas que pode não se traduzir em ganhos significativos.

3.7.2 Bloco de Região de Interesse (ROI)

O bloco que se segue depois do referente à câmara é facultativo. Foi desenvolvido a pensar na exclusão de áreas que são captadas pela câmara, mas para as quais não existe qualquer interesse em analisar, como por exemplo, a existência de duas ruas paralelas onde apenas exista interesse em trabalhar sobre a primeira, podendo ser interessante remover essa área numa fase inicial do que posteriormente, até por uma questão de gestão de recursos de processamento.

O bloco permite definir a região de interesse, em pixels, que será utilizada no processamento, utilizando por exemplo uma resolução de 480x640 como entrada do bloco. Definir apenas a ROI como [181:480] e [1:640], obtendo-se assim uma imagem de 300x640, de onde foi excluída a região superior correspondente a 180x640. O exemplo descrito pode ser observado na Figura 3-17.

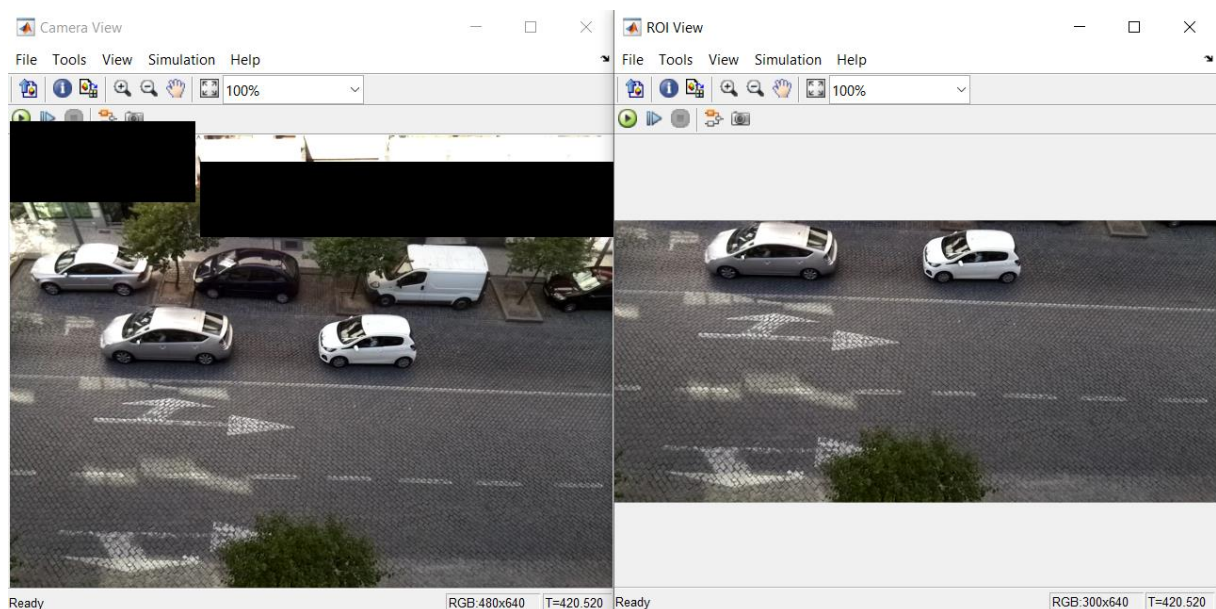


Figura 3-17: Aplicação de ROI a uma imagem.

3.7.3 Bloco de Detecção de Primeiro Plano

O bloco que se segue diz respeito à detecção de primeiro plano, sendo um dos blocos principais do sistema, uma vez que uma má definição do que é objeto em primeiro plano, poderá levar a classificações errôneas e contagens irreais.

O bloco apresenta alguns parâmetros de configuração sendo que um é o número de *frames* iniciais de treino que, tal como indica, corresponde ao número necessário de *frames* que o modelo precisa para identificar corretamente o fundo. Um valor baixo pode levar a erros na detecção do primeiro plano.

Conforme se pode observar na Figura 3-18, a detecção do primeiro plano é feita com sucesso, onde também se pode visualizar a presença de algum ruído nesta detecção, tal acontece por movimentos das folhas da árvore na parte inferior, a presença de sombras dos veículos e reflexos de luz nas vias de trânsito.

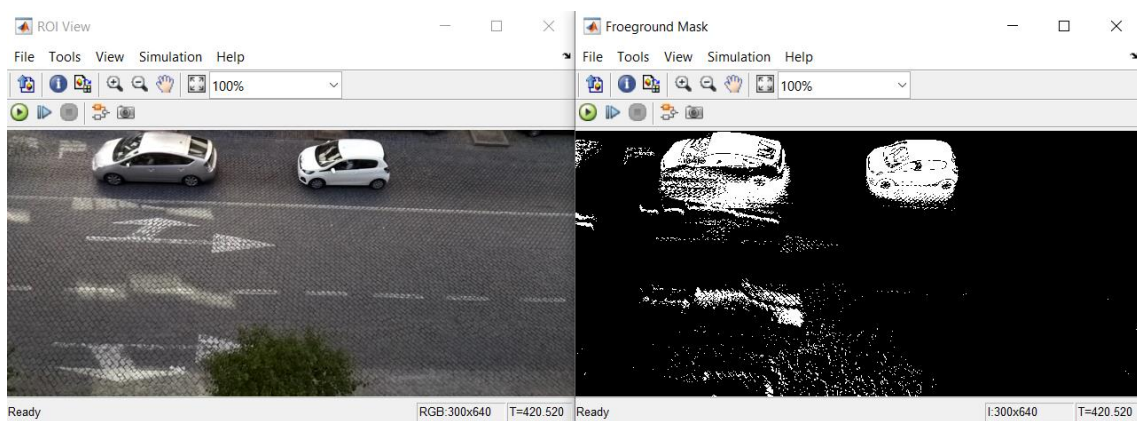


Figura 3-18: Detecção de primeiro plano aplicada.

3.7.4 Bloco de Filtro de Mediana

O filtro de mediana permite filtrar a mediana da matriz de entrada, que corresponde à saída do bloco de detecção de primeiro plano. O valor de cada pixel de entrada será substituído pelo valor da vizinhança N por N , sendo o valor deste parâmetro definido no bloco. O bloco é usado para remover algum ruído que possa estar presente na saída do bloco anterior. Num caso perfeito de identificação do primeiro plano, este bloco seria dispensável.

A Figura 3-19 demonstra a utilização deste filtro para redução de algum do ruído referido no ponto 3.7.3, onde parte da sobra dos veículos foi eliminada, assim como os reflexos nas vias de trânsito, também o movimento das folhas foi reduzido a uma área muito pequena. Desta forma, o ruído eliminado não terá repercussão no bloco sequente.

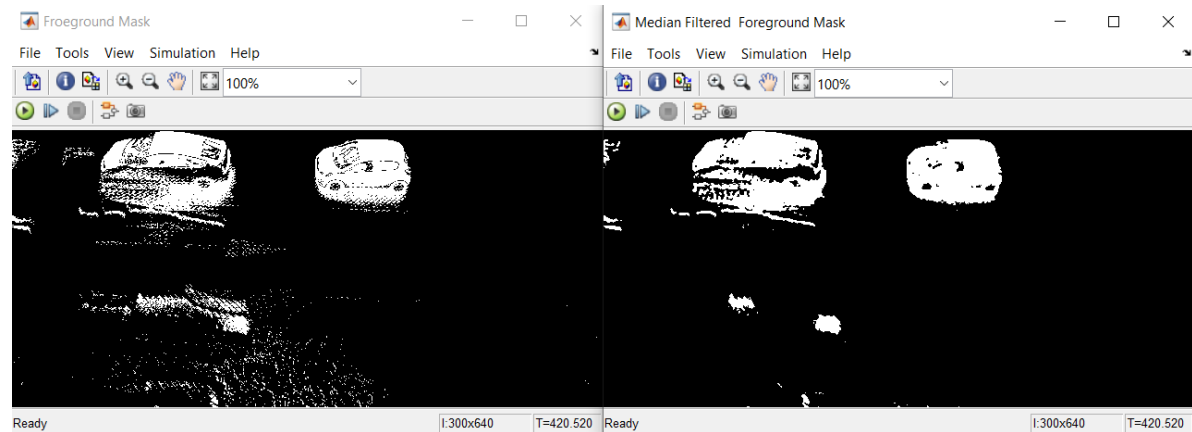


Figura 3-19: Filtro de mediana aplicado.

3.7.5 Bloco de BLOB

BLOB sigla para Binary Large Object é um bloco de análise que permite calcular parâmetros de uma imagem binária. Tem como entrada a imagem a preto e branco, binária, com origem na saída do bloco anterior. Relativamente às saídas, são vários os parâmetros que se podem seleccionar, entre os quais estão a área, o centroide e a caixa delimitadora respeitantes a cada objeto detetado. Já em relação às opções, é possível definir alguns limites, tais como o número máximo de objetos e a dimensão máxima e mínima de cada um.

Para o bloco seguinte é possível trabalhar-se apenas com o centroide, no entanto, para se fazer uma pré-seleção dos dados a tratar, optou-se por limitar as áreas mínima e máxima. Também por uma questão de trabalhar com uma área mais próxima da real do objeto detetado, optou-se por se adicionar a área aos parâmetros de saída, uma vez que pelo centroide apenas se obtinha a localização do veículo, para a classificação é necessário a área que o mesmo ocupa na imagem.

A Figura 3-20 representa os valores das saídas do bloco referido, onde os primeiros dois valores dizem respeito ao centroide do primeiro veículo e o terceiro valor à área do mesmo, este valor diminui por consequência de o mesmo estar a desaparecer da imagem. Os quarto e quinto valores são referentes a um segundo veículo existente na imagem, sendo o último valor a respetiva área.

47 ; 76 ; 9752 ; 568 ; 83 ; 7488
46 ; 76 ; 9384 ; 565 ; 83 ; 7605
43 ; 75 ; 9016 ; 562 ; 82 ; 7605
41 ; 75 ; 8648 ; 559 ; 82 ; 7552
39 ; 75 ; 8188 ; 556 ; 81 ; 7488
38 ; 75 ; 7912 ; 552 ; 81 ; 7616
36 ; 75 ; 7626 ; 548 ; 80 ; 7616
34 ; 75 ; 7161 ; 545 ; 79 ; 7680
32 ; 75 ; 6882 ; 541 ; 79 ; 7497
30 ; 74 ; 6348 ; 537 ; 79 ; 7497
28 ; 74 ; 5980 ; 533 ; 79 ; 7497
26 ; 74 ; 5612 ; 529 ; 79 ; 7316
24 ; 74 ; 5244 ; 525 ; 78 ; 7434
22 ; 73 ; 4968 ; 521 ; 78 ; 7497
20 ; 72 ; 4312 ; 516 ; 78 ; 7434
19 ; 72 ; 3825 ; 511 ; 78 ; 7616
17 ; 71 ; 3362 ; 507 ; 78 ; 7434
15 ; 70 ; 2775 ; 502 ; 78 ; 7616
13 ; 70 ; 2400 ; 498 ; 78 ; 7434

Figura 3-20: Valores das saídas do bloco de BLOB.

3.7.6 Bloco de Contagem e Classificação de Veículos

O bloco de contagem e classificação de veículos foi desenvolvido recorrendo ao bloco de função de MATLAB personalizado. Este tipo de bloco é usado para se definir uma função específica, neste caso em código de MATLAB, recorrendo às funções que esta linguagem permite.

Este bloco apresenta-se com seis entradas. Duas delas são reservadas para o controlo temporal que permite fazer reinicializar os contadores. Outras duas, como já foi referido, estão destinadas para a área do objeto detetado e para o centroide deste. A área será o que permitirá fazer a correta classificação do veículo. O centroide terá como objetivo permitir acompanhar o percurso do veículo, possibilitando desta forma fazer a correta contagem do mesmo e o respetivo sentido, da esquerda para a direita ou da direita para a esquerda.

Através da entrada centroide é possível saber-se a cada instante onde se encontram os objetos. Para se fazer a contagem, optou-se pelo desenvolvimento de um sistema básico de seguimento do objeto, desde o momento em que é detetado, até ao momento em que o deixa de o ser. Existe uma pilha invertida que será ocupada pelos objetos e libertada ao longo do tempo.

Seguidamente é apresentado um exemplo do funcionamento da pilha invertida durante vários instantes temporais. No instante inicial as pilhas invertidas encontram-se vazias, conforme a Figura 3-21 representa.

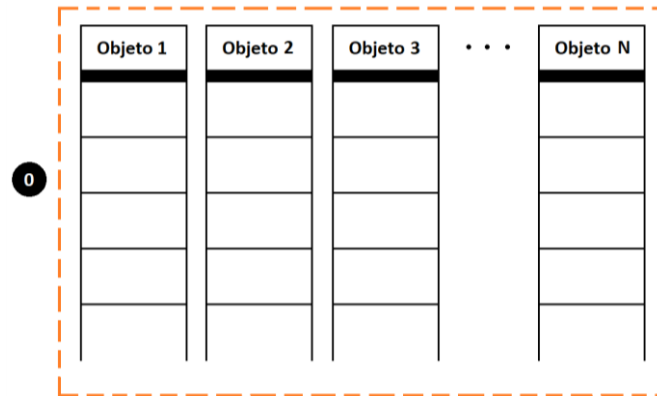


Figura 3-21: Pilhas invertidas, vazias, no instante $t = 0$.

Decorrido o instante inicial, são detetados na imagem pela primeira vez dois objetos, cada um na sua posição para o instante 1, que a Figura 3-22 representa.

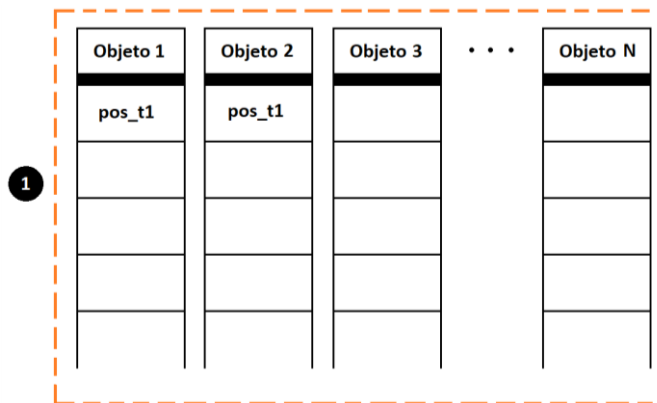


Figura 3-22: Pilhas invertidas preenchidas com dois objetos no instante $t = 1$.

No instante 2, os dois objetos encontrados anteriormente, continuam a ser detetados mas em novas posições e aos quais se junta um terceiro objeto, tal como se representa na Figura 3-23. Primeiramente é verificado se existem objetos nas pilhas, caso existam, são comparadas as distâncias entre posições anteriores e as novas, sendo a nova posição adicionada na coluna do objeto onde a distância entre a posição anterior e a nova forem menores. Caso não existam objetos anteriores ou as novas posições não tenham correspondência com as anteriores, os novos objetos são adicionados às pilhas vazias.

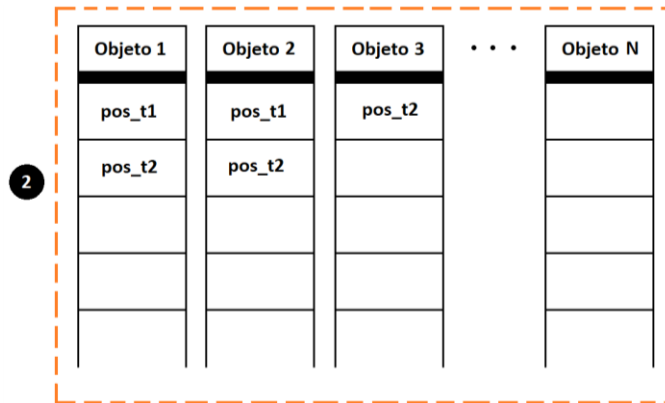


Figura 3-23: Pilhas invertidas preenchidas com três objetos no instante $t = 2$.

No momento seguinte existem duas possibilidades para os objetos, caso se movimentem ou se mantenham imóveis, são adicionados às respectivas pilhas invertidas. No caso de algum desaparecer da imagem, considera-se que o mesmo não era um objeto, no entanto a posição anterior não é eliminada, uma vez que o objeto pode estar presente mas não ter sido detetado neste instante. No instante 3, apenas dois objetos são encontrados, o objeto 1 deixa de ser detetado, como tal, essa posição ficará vazia. Na Figura 3-24, o espaço vazio encontra-se marcado com um X.

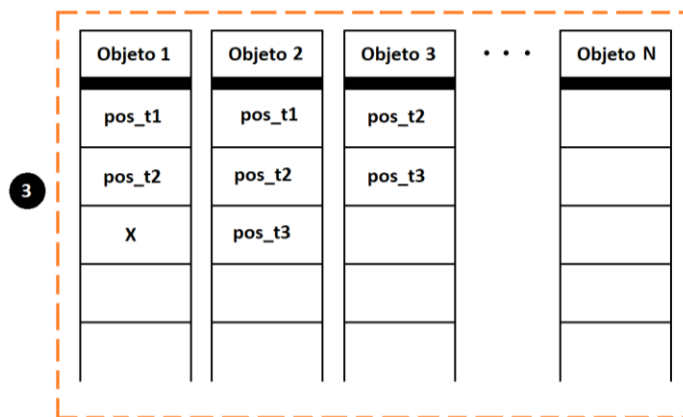


Figura 3-24: Pilhas invertidas preenchidas com dois objetos no instante $t = 3$.

No instante em que deixa de ser detetado o objeto 1, é analisada a distância percorrida para ser validada a contagem, caso essa distância ainda não tenha sido percorrida, o sistema aguarda por nova detecção, no caso que não seja encontrado por vários instantes, o mesmo é descartado e não contabilizado, considerando-se um erro. Por esse motivo, é adicionado um tempo de esquecimento que é usado para libertar colunas da pilha invertida. Na Figura 3-25, o objeto ainda não tinha percorrido a distância suficiente mas no instante 4 volta a ser detetado e adicionado à respectiva pilha.

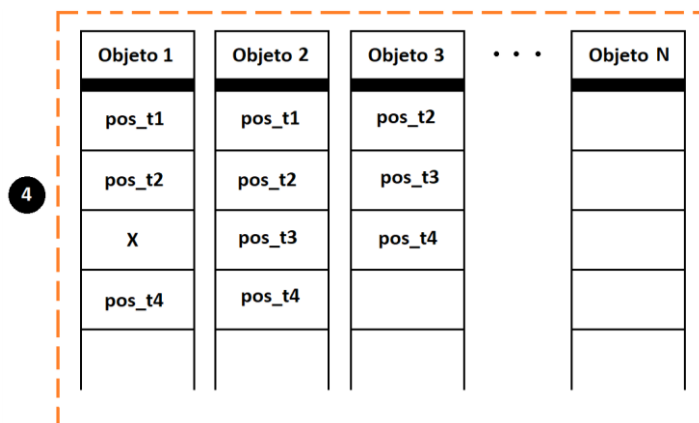


Figura 3-25: Pilhas invertidas preenchidas com três objetos no instante $t = 4$.

Nos momentos seguintes volta-se a repetir o processo onde a pilha vai crescendo. No momento em que o objeto deixe de ser detetado pelo período temporal definido como tempo de esquecimento, é avaliada a distância percorrida por esse objeto, e caso seja superior ao valor definido, considera-se que era um objeto válido para classificação. Caso não tenha percorrido essa distância, considera-se que o mesmo não é válido para classificação. Após ser validado ou não para classificação, a pilha é libertada e colocada vazia. Para definir o sentido do objeto, é utilizado o eixo horizontal. Caso o valor do x do centroide aumente, considera-se que o objeto se desloca da esquerda para a direita e, caso contrário, assume-se que se desloca da direita para a esquerda.

As outras duas entradas, do bloco de contagem e classificação, estão destinadas à separação da classificação dos veículos, sendo o valor mais baixo destinado à separação de motocicletas dos veículos ligeiros e o valor mais alto para distinguir veículos ligeiros de pesados. Depois de o objeto ser detetado, faz-se a média da área detetada e se esta estiver compreendida entre o valor mínimo e o valor máximo, considera-se um veículo ligeiro. Caso tenha uma área inferior ao valor mínimo, assume-se como sendo um motociclo. Se o valor da área detetada for superior ao valor máximo definido, nesse caso o objeto é classificado como veículo pesado. A Figura 3-26 resume todo o processo descrito no ponto 3.7.6.

Depois de feita a contagem e classificação, os valores são enviados para seis saídas, três para cada um dos dois sentidos, onde cada uma dessas três corresponde ao tipo de veículo, motociclo, ligeiro ou pesado. Estes valores são colocados a zero a cada minuto. Considerou-se este espaço temporal como o mais adequado para um futuro tratamento dos dados.

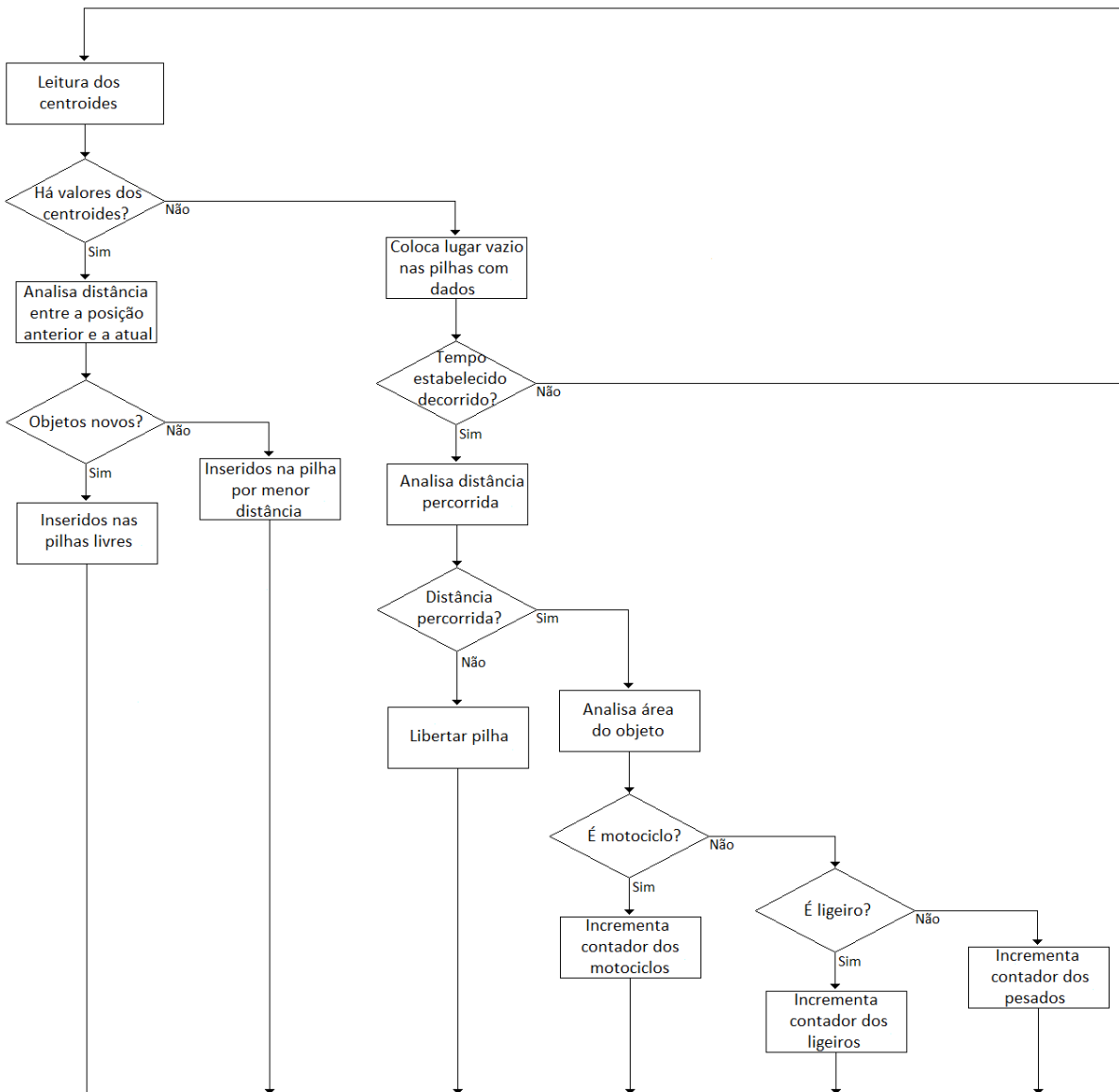


Figura 3-26: Diagrama resumo do bloco de classificação e contagem.

3.7.7 Comunicação do Sistema

O último bloco do sistema corresponde ao de comunicação. Tal como o bloco anterior foi desenvolvido de forma personalizada e assenta num bloco do tipo S (bloco de sistema), com uma grande diferença relativamente ao anterior, uma vez que este utiliza linguagem C. Foi utilizada esta linguagem uma vez que este bloco tem duas funções distintas que não estavam disponíveis em MATLAB: a primeira passa por ler a hora do sistema para ser possível determinar se já decorreu o tempo definido, neste caso um minuto, para que as contagens de veículos sejam enviadas e seja feita a reinicialização dos contadores; a segunda função diz respeito à comunicação, já referida no ponto 3.6.1, uma comunicação de *named pipes* do tipo FIFO, onde o bloco do sistema escreve os valores de contagem respeitantes a cada sentido e a

cada tipo de veículo, sendo posteriormente estes valores lidos pelo sistema de comunicação. A Figura 3-27 resume o bloco descrito.

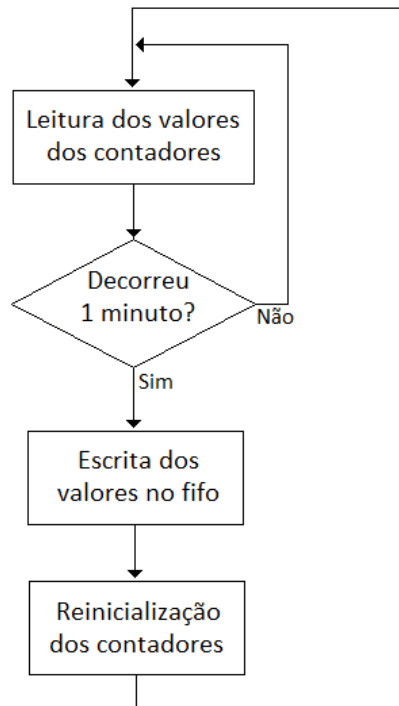


Figura 3-27: Diagrama resumo do bloco de comunicação.

3.7.8 Funções Adicionais

O bloco opcional desenvolvido, permite visualizar em tempo real os resultados das contagens com os veículos devidamente identificados. Este bloco é totalmente prescindível, especialmente numa fase em que o sistema já esteja totalmente configurado e implementado. Até se chegar a esta fase, pode ser de grande importância perceber se o sistema está a funcionar corretamente para melhor corrigir os erros e perceber de que forma os mesmos acontecem. É com recurso ao bloco SDL Display que esta depuração de erros e problemas pode ser feita. O bloco faz exibir uma janela no Raspberry Pi com a imagem captada pela câmara, os veículos destacados com retângulos e os seis contadores distintos, conforme Figura 3-28.

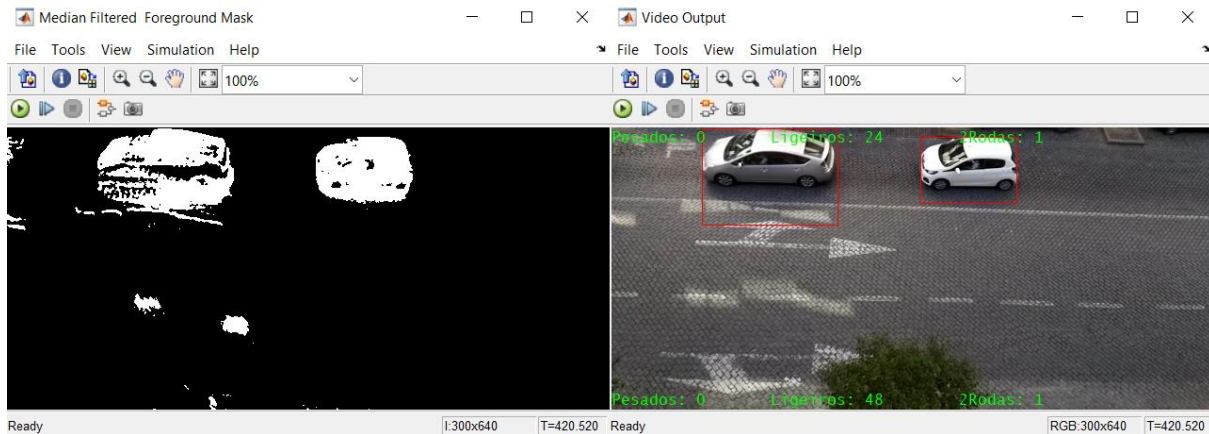


Figura 3-28: Visualização de controlo de contagem.

3.8 Envio e Backup dos Dados

Depois de obtidos os valores dos sensores, era objetivo que os mesmos ficassem de alguma forma disponíveis, não só em tempo real, mas para consulta futura. Com estas finalidades, os dados teriam então de ser enviados para um servidor, onde os mesmos pudessem ficar guardados para consulta futura.

Para dar resposta a este ponto, foram equacionadas algumas opções. A primeira seria deixar os dados guardados no Raspberry Pi e quando fosse necessário o acesso aos mesmos, seria feito um acesso direto ao Raspberry Pi. Tal solução foi afastada, uma vez que num possível caso com vários dispositivos, não se tornava prático aceder diretamente a cada um deles, tendo ainda a desvantagem de, no caso do Raspberry Pi ficar inacessível, deixar-se-ia de ter acesso tanto aos dados em tempo real como aos dados mais antigos.

Ficou assim definido que os dados seriam enviados para um servidor com recurso a *web service*, onde os mesmos ficariam guardados para acesso posterior. Assim, e como ferramenta a biblioteca *cURL*, deu-se resposta ao problema. Os dados ficam, desta forma, acessíveis para consulta posterior.

Por uma questão de redundância e de forma a tornar o sistema compatível com diferentes servidores, foi instalado e utilizado um servidor na ESTGV e o servidor SmartIoT da AlticeLabs (AlticeLabs SmartIoT, 2019). Os envios para cada um destes servidores são feitos de forma diferente, embora em ambos se recorra à biblioteca *cURL* (Curl, 2019).

Os dados são enviados com uma cadência de um minuto, podendo este tempo ser reduzido ou aumentado. Considerou-se que um tempo inferior não traria grande vantagem tendo em conta o tipo de dados. Um tempo superior apenas teria a vantagem de reduzir o tráfego de dados,

embora a largura de banda para envio seja de aproximadamente 34 kbps para o servidor da ESTGV e 65 kbps para o SmartIoT.

Desta forma, a amostragem a cada minuto, permite, no servidor, em momento posterior, fazer as agregações conforme for mais vantajoso para a consulta. O sistema torna-se assim mais flexível a cada caso e não compromete, de alguma forma, a discrepância que seria obtida com tempos de envio superiores.

O fluxograma da Figura 3-29 representa o funcionamento do sistema de envio e *backup* dos dados, são feitas as leituras aos sensores, caso estas tenham sido feitas no número pré-estabelecido, é feita uma verificação se já existem dados no FIFO relativos à contagem e classificação de veículos, caso seja positivo, são lidos e enviados para os servidores, sendo de seguida guardados em *backup*.

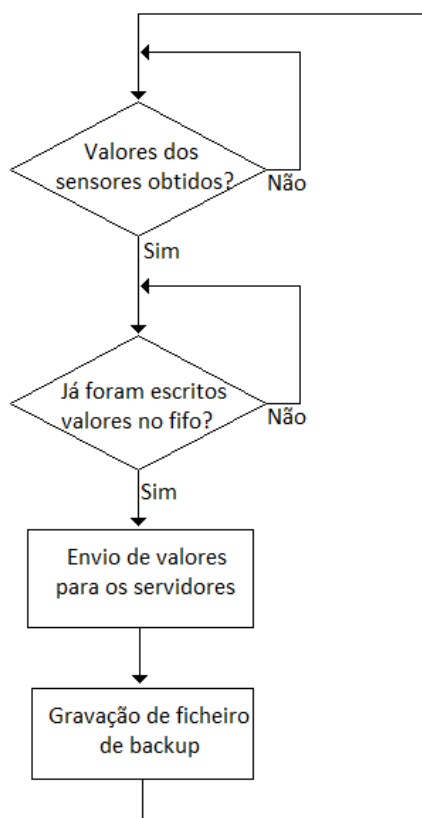


Figura 3-29: Fluxograma de envio e *backup* dos dados.

3.8.1 SmartIoT (AlticeLabs)

Conforme foi referido no ponto 3.8, um dos servidores utilizado para o armazenamento dos dados é o SmartIoT da AlticeLabs, tratando-se de uma API baseada em REST. Tem como

objetivo servir de alojamento para utilizar em conjunto com um centro de governação da cidade, no âmbito das Smart Cities.

O SmartIoT é uma plataforma especializada no fornecimento de mecanismos para integrar dispositivos com aplicações de uma forma ágil e aberta, para que os fornecedores de serviços consigam, facilmente, criar um ecossistema de parceiros e enriquecer a sua oferta, idealizada de acordo com um conjunto de premissas ao nível do desempenho, fiabilidade e capacidade de integração (Reis, 2017).

Com o objetivo de se perceber melhor as possíveis utilidades que esta ferramenta permite, segue-se um pequeno exemplo, que considera dois tipos de dispositivos, um ar-condicionado e o seu controlo remoto. Neste cenário a comunicação e entrega de mensagens entre um aparelho de ar-condicionado e o seu respetivo controlo remoto é feita através da SmartIoT, como a Figura 3-30 representa (AltiCeLabs SmartIoT Quick Start Guide, 2019).

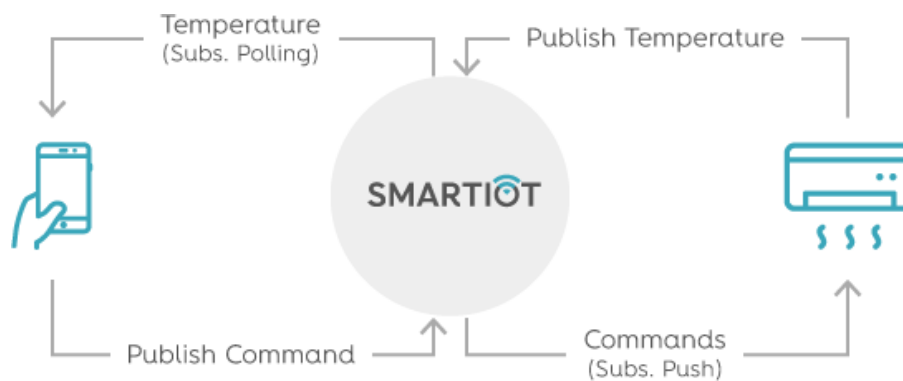


Figura 3-30: Exemplo de uso da SmartIoT (Retirado de (AltiCeLabs SmartIoT Quick Start Guide, 2019)).

Para usar o cenário descrito é necessário (AltiCeLabs SmartIoT Quick Start Guide, 2019):

1. Ter uma conta para que se possa autenticar na plataforma;
2. Registrar ambos os dispositivos ar-condicionado (AC) e controlo remoto (RC) e os fluxos relacionais, ou seja, temperatura e comandos, respetivamente;
3. Criar as inscrições:
 - O controlo remoto subscreve a “temperatura” do AC;
 - O AC subscrever os "comandos" do RC;
4. Os dispositivos autenticam-se dentro da plataforma;
5. Os dispositivos publicam dados na plataforma.

De uma forma muito simples é possível explicar o mecanismo de autenticação que permite a subscrição. O dispositivo comunica com a SmartIoT de forma mais tradicional com as credenciais, utilizador e password, do dispositivo. Caso as credenciais fornecidas estejam corretas, a SmartIoT devolve um token para subscrição, válido por até uma hora (AltiCeLabs SmartIoT Autenticação, 2019). Este método de autenticação está representado na Figura 3-31.

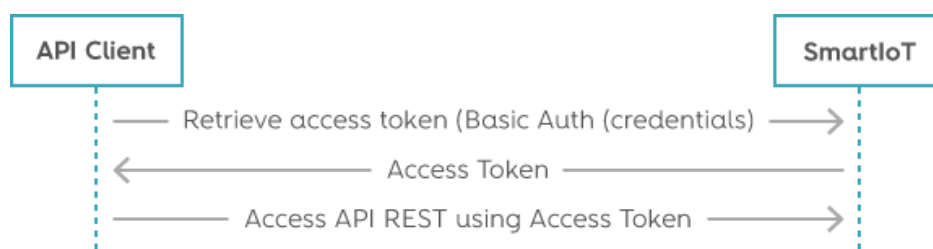


Figura 3-31: Autenticação da SmartIoT (Retirado de (AlticeLabs SmartIoT Autenticação, 2019)).

3.8.2 Servidor na ESTGV

Conforme foi referido no ponto 3.8, um dos servidores utilizado para o armazenamento dos dados encontra-se montado na ESTGV, no mesmo encontra-se instalado o sistema de gestão de base de dados MySQL assim como o administrador de base de dados phpMyAdmin.

O phpMyAdmin é responsável por receber os dados enviados através de um endereço e seguidamente escrever os mesmos na base de dados, os valores são enviados conforme pode ser visto na Figura 3-32, onde a variável é substituída pelo respetivo nome da variável, por exemplo CO, e o valor 123 substituído pelo respetivo valor.

```
http://190.██.██.██/~pdm/get_data.php/?tip=variavel&var1=123
```

Figura 3-32: Endereço exemplo de envio dos dados.

Para além de receber os dados, também tem a função de enviar os dados pedidos pela aplicação móvel. Depois de receber o pedido pela aplicação, faz a consulta na base de dados e retorna os valores pedidos pela aplicação, disponíveis no endereço exemplo da Figura 3-33 e apresentados em formato JSON como a Figura 3-34 representa.

```
http://190.██.██.██/~pdm/consultaval.php
```

Figura 3-33: Endereço exemplo de consulta dos dados.

```
{
  "valores": [
    {"contador": "2605090", "data": "2019-09-19", "hora": "04:01:12", "tipo": "4", "var1": "26.9"},
    {"contador": "2605111", "data": "2019-09-19", "hora": "04:02:12", "tipo": "4", "var1": "26.9"},
    {"contador": "2605132", "data": "2019-09-19", "hora": "04:03:12", "tipo": "4", "var1": "26.9"},
    {"contador": "2605153", "data": "2019-09-19", "hora": "04:04:12", "tipo": "4", "var1": "26.9"},
    {"contador": "2605174", "data": "2019-09-19", "hora": "04:05:13", "tipo": "4", "var1": "26.9"},
    {"contador": "2605195", "data": "2019-09-19", "hora": "04:06:12", "tipo": "4", "var1": "26.9"},
    {"contador": "2605216", "data": "2019-09-19", "hora": "04:07:12", "tipo": "4", "var1": "26.8"}
  ]
}
```

Figura 3-34: Resposta do servidor com os dados em JSON.

3.8.2.1 Agregação de Dados

A agregação dos dados no servidor torna a sua consulta mais rápida, retirando da aplicação móvel esse esforço adicional. Com esse objetivo foi implementada a agregação de valores no servidor da ESTGV.

Sempre que se enviam dados para o servidor, é avaliada a existência ou não de condições para efetuar a agregação dos dados. Tal verificação é feita através de um *trigger* na base de dados, sendo que as agregações são efetuadas por hora, dia, semana e mês.

Quando um valor é recebido na base de dados, é analisado para se apurar se houve uma mudança de hora, dia, semana ou mês. Caso se confirme essa mudança, é feita uma agregação dos dados correspondentes. A Figura 3-35 representa o funcionamento da verificação de condições para agregação.

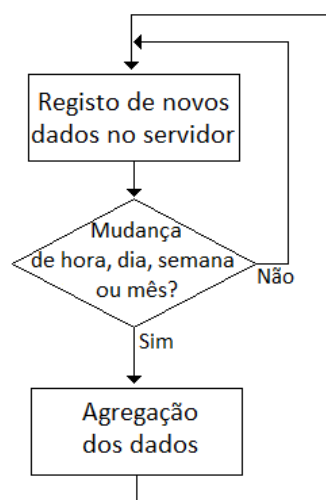


Figura 3-35: Verificação para agregação de dados.

As agregações são realizadas em quatro tipos: hora, dia, semana e mês, onde para cada hora é feita a média dos 60 minutos correspondentes, o mesmo se aplicando às restantes agregações.

3.8.3 Registo de Dados Local

Como foi referido no ponto 3.8, os dados são enviados para dois servidores distintos, mas, desta forma, o sistema necessita de uma ligação à rede de internet, o que por algum motivo pode ficar indisponível ou o local onde vai ficar instalado não ter acesso à internet. Para dar resposta a este problema, foi desenvolvido um sistema de *backup*, onde os dados são guardados a cada minuto, podendo esse tempo ser alterado.

Optou-se então por fazer a gravação dos dados no próprio Raspberry Pi. O volume que ocupa é relativamente reduzido, 153KB por dia, tratando-se de um ficheiro CSV que possibilita mais tarde que possa ser utilizado por qualquer software. Por uma questão de facilidade da consulta, a cada dia é gerado um novo ficheiro cujo nome é a data: ano, mês e dia, conforme Figura 3-36.

```

pi@raspi:~/backupCSV $ ls
backup2019-3-31.csv  backup2019-5-22.csv  backup2019-6-24.csv  backup2019-7-23.csv
backup2019-4-15.csv  backup2019-5-23.csv  backup2019-6-25.csv  backup2019-7-24.csv
backup2019-4-16.csv  backup2019-5-24.csv  backup2019-6-26.csv  backup2019-7-26.csv
backup2019-4-1.csv   backup2019-5-27.csv  backup2019-6-27.csv  backup2019-7-27.csv
backup2019-4-22.csv  backup2019-5-28.csv  backup2019-6-28.csv  backup2019-7-28.csv
backup2019-4-29.csv  backup2019-5-2.csv   backup2019-6-29.csv  backup2019-7-29.csv
backup2019-4-30.csv  backup2019-5-3.csv   backup2019-6-30.csv  backup2019-7-2.csv
backup2019-4-7.csv   backup2019-5-6.csv   backup2019-6-6.csv   backup2019-7-30.csv
backup2019-4-8.csv   backup2019-5-7.csv   backup2019-7-10.csv  backup2019-7-31.csv
backup2019-4-9.csv   backup2019-5-8.csv   backup2019-7-11.csv  backup2019-7-3.csv
backup2019-5-10.csv  backup2019-5-9.csv   backup2019-7-12.csv  backup2019-7-4.csv
backup2019-5-11.csv  backup2019-6-11.csv  backup2019-7-13.csv  backup2019-7-5.csv
backup2019-5-12.csv  backup2019-6-12.csv  backup2019-7-14.csv  backup2019-7-6.csv
backup2019-5-13.csv  backup2019-6-13.csv  backup2019-7-15.csv  backup2019-7-7.csv
backup2019-5-14.csv  backup2019-6-14.csv  backup2019-7-16.csv  backup2019-7-8.csv
backup2019-5-15.csv  backup2019-6-15.csv  backup2019-7-17.csv  backup2019-7-9.csv
backup2019-5-16.csv  backup2019-6-16.csv  backup2019-7-18.csv  backup2019-8-1.csv
backup2019-5-17.csv  backup2019-6-17.csv  backup2019-7-19.csv  backup2019-8-2.csv
backup2019-5-18.csv  backup2019-6-18.csv  backup2019-7-1.csv   backup2019-8-3.csv
backup2019-5-19.csv  backup2019-6-19.csv  backup2019-7-20.csv  backup2019-8-4.csv
backup2019-5-20.csv  backup2019-6-20.csv  backup2019-7-21.csv  backup2019-8-5.csv
backup2019-5-21.csv  backup2019-6-21.csv  backup2019-7-22.csv  backup2019-8-6.csv

```

Figura 3-36: Backup dos dados em ficheiro.

Dentro do ficheiro encontram-se os dados organizados por colunas, as duas primeiras respeitantes a data e hora respetivamente, as restantes são relativas à contagem e classificação de veículos e aos parâmetros ambientais, como a Figura 3-37 demonstra.

```

2019-8-12;13:24:4;0;0;0;26.2;999;45;12;6;6;414;1;4.51;0.16;0.69;577.48;404.00;709.96;0.76;1.67;272957.56;80
2019-8-12;13:25:4;0;0;0;26.2;999;46;12;6;6;414;1;4.60;0.16;0.67;551.56;387.75;799.00;0.78;1.72;272971.62;87
2019-8-12;13:26:4;0;0;0;26.2;999;45;12;6;6;414;1;4.59;0.15;0.73;622.27;429.57;794.57;0.78;1.71;272958.00;73
2019-8-12;13:27:4;0;0;0;26.3;999;45;12;6;6;414;1;4.53;0.16;0.74;650.13;444.65;777.25;0.76;1.69;272991.28;76
2019-8-12;13:28:5;0;0;0;26.2;999;45;12;6;6;414;1;4.47;0.15;0.74;649.96;442.49;630.24;0.74;1.65;272998.62;77
2019-8-12;13:29:4;0;0;0;26.3;999;45;12;6;6;414;1;4.58;0.15;0.71;613.01;421.32;788.59;0.78;1.71;272990.38;82
2019-8-12;13:30:4;0;0;0;26.2;999;45;12;6;6;414;1;4.43;0.15;0.72;623.25;427.73;620.41;0.73;1.63;272986.09;81
2019-8-12;13:31:4;0;0;0;26.2;999;46;12;6;6;414;1;4.43;0.15;0.72;625.75;429.21;620.76;0.73;1.63;273004.34;81
2019-8-12;13:32:4;0;0;0;26.2;1000;45;12;6;6;414;1;4.49;0.15;0.74;650.27;444.82;768.07;0.76;1.67;272987.78;81
2019-8-12;13:33:4;0;0;0;26.2;999;45;12;6;6;414;1;4.53;0.16;0.70;586.67;409.89;779.80;0.76;1.69;273002.59;80
2019-8-12;13:34:4;0;0;0;26.3;999;45;12;6;6;414;1;4.60;0.16;0.74;652.70;446.07;804.20;0.78;1.72;273002.16;78

```

Figura 3-37: Conteúdo de um ficheiro de *backup*.

3.9 Consulta e Envio de Dados por Aplicação Móvel

Depois do sistema se encontrar a enviar os dados, colocava-se a questão de como se iria fazer o acesso aos mesmos. Aceder diretamente ao servidor não era intuitivo para um utilizador com menos conhecimentos, pelo que se optou por se desenvolver uma forma simples de acesso aos dados através do browser.

O acesso através do browser era simples, bastava escrever o endereço, no entanto, os dados eram visualizados de forma pouco intuitiva, tratando-se apenas de exibição de dados em JSON, conforme Figura 3-38. Este processo estava a ser feito pelo servidor da ESTGV e foi uma forma simples de não o sobrecarregar.

```
1 {
2   "valores": [
3     {
4       "data": "2019-08-25",
5       "hora": "11:27:10",
6       "tipo": "temperatura",
7       "var1": "30.2"
8     },
9     {
10      "data": "2019-08-25",
11      "hora": "11:27:10",
12      "tipo": "pressao",
13      "var1": "995"
14    },
15    {
16      "data": "2019-08-25",
17      "hora": "11:27:10",
18      "tipo": "humidade",
19      "var1": "41"
20    }
21  ]
22 }
```

Figura 3-38: Exemplo de código JSON.

Apoiado nas novas tecnologias, definiu-se como um caminho mais adequado uma aplicação móvel em Android, uma vez que é mais provável uma pessoa ter um smartphone do que um computador, mesmo que portátil, com ligação à internet. Esta aplicação desenvolveu-se para o sistema operativo Android, uma vez que é aquele que tem maior quota de mercado, superior a 75% (StatCounter Global Stats, 2019).

A crescente preocupação com a saúde, aliando a prática de desporto ao ar livre com a qualidade do ambiente que a rodeia, foi a premissa base para o desenvolvimento de uma aplicação móvel que possibilitasse que um utilizador se mantivesse não só informado sobre a qualidade do ar, mas que também lhe permitisse emitir uma opinião em relação ao espaço onde o mesmo se encontra.

A aplicação foi desenvolvida com recurso ao Android Studio, software de programação para aplicações móveis Android, disponibilizado pela Google de forma gratuita (Android Studio,

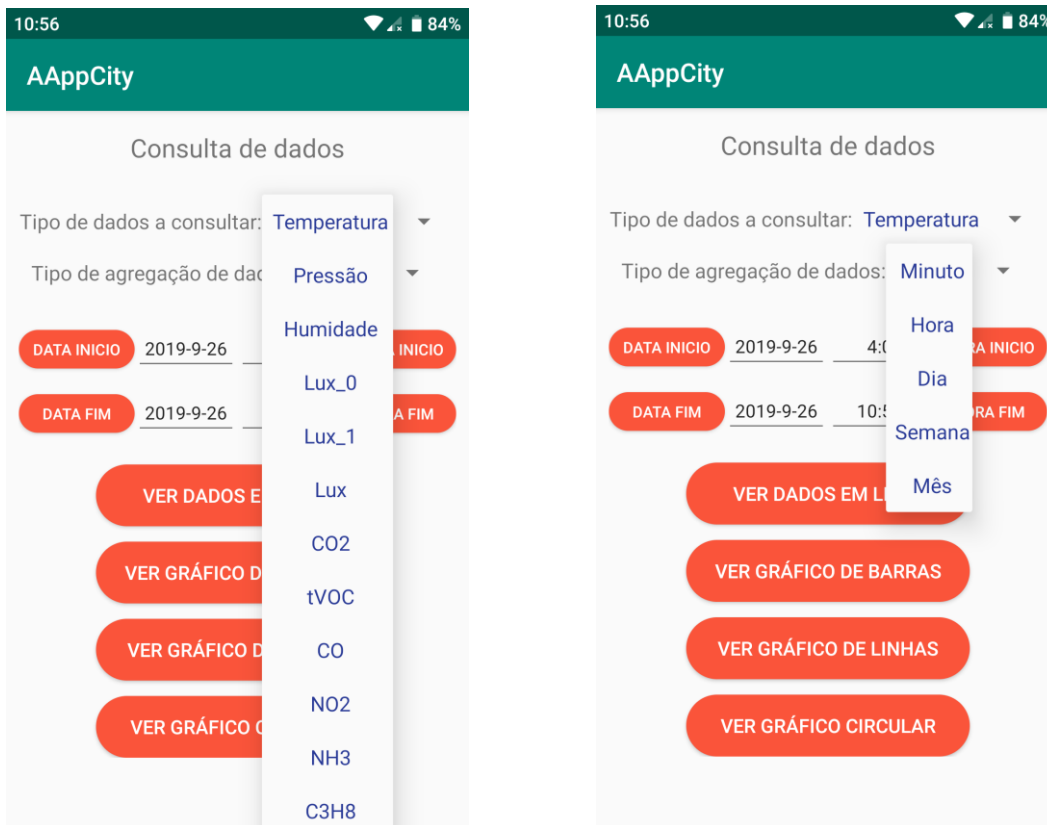
2019). A linguagem de programação utilizada foi JAVA, com recurso a bibliotecas externas, nomeadamente para a visualização de gráficos.

A aplicação foi projetada com o foco na utilidade e não tanto no design, muito embora se pretendesse que fosse de utilização fácil. A consulta de dados permite adicionar filtros de uma forma intuitiva na aplicação, tipo de dados, data, entre outros. Foram desenvolvidas diversas funcionalidades para a aplicação. Na página inicial é exibido o nome da aplicação, assim como as atividades iniciais, despoletadas por botões, que podem ser realizadas na mesma, conforme Figura 3-39.



Figura 3-39: Atividade principal da aplicação.

O primeiro botão, consultar dados, dá acesso à atividade de consulta de dados, presente na Figura 3-40. Nesta atividade é exibido um conjunto de informações que podem ser alteradas e adicionadas. A primeira diz respeito ao tipo de dados a consultar Figura 3-40a, por exemplo a temperatura. A segunda é referente ao tipo de consulta a realizar Figura 3-40b, sendo possível realizar uma consulta com intervalo dos dados de um minuto por exemplo ou agregações de uma hora, um dia, uma semana ou um mês. Seguidamente devem ser selecionadas as datas e horas de início e fim da consulta de dados a realizar. Os botões auxiliam nesse processo com a escolha da data em calendário e a hora em relógio. Ainda dentro desta atividade deve-se carregar no botão correspondente à forma como se pretende realizar a visualização dos dados, se na forma de lista ou se na forma de gráficos, de barras, de linhas ou circular, conforme se pode observar na Figura 3-40.



(a) Lista do tipo de dados

(b) Lista do tipo de agregação

Figura 3-40: Atividade de consulta dos dados.

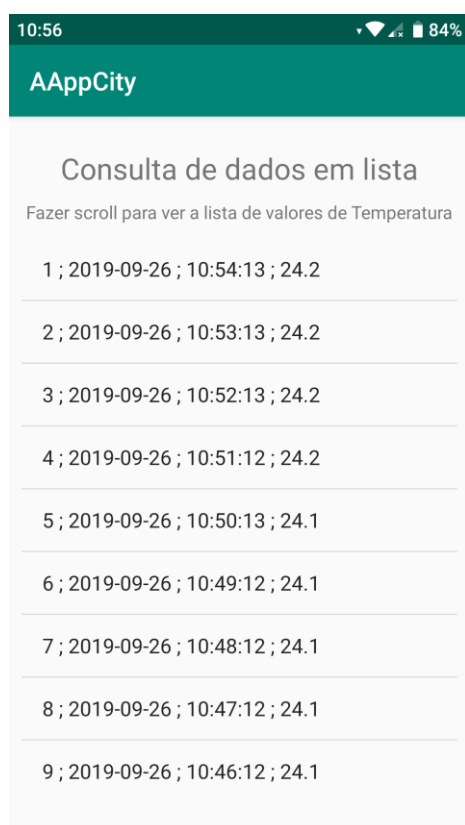
A forma de ler os dados que se encontram no servidor consiste em formular um pedido e analisar a resposta do servidor. No pedido é indicado o tipo de dados que se pretende consultar, o tipo de agregação em que se vai fazer a consulta e as datas e horas de início e fim da consulta. Depois de feito o pedido, é devolvida a resposta pelo servidor com os dados a serem recebidos em formato JSON. Após recebidos os dados, os mesmos são organizados para visualização em lista ou em gráficos.

Os dados recebidos, pelo equipamento Android, são organizados para serem exibidos ao utilizador da forma que este pretenda. Caso seja selecionada a visualização dos dados em lista, os mesmos são colocados numa nova atividade, numa lista organizada e numerada do evento mais recente para o mais antigo na primeira coluna. A segunda e terceira colunas são referentes à data e à hora respetivamente e a última coluna diz respeito ao valor do parâmetro consultado.

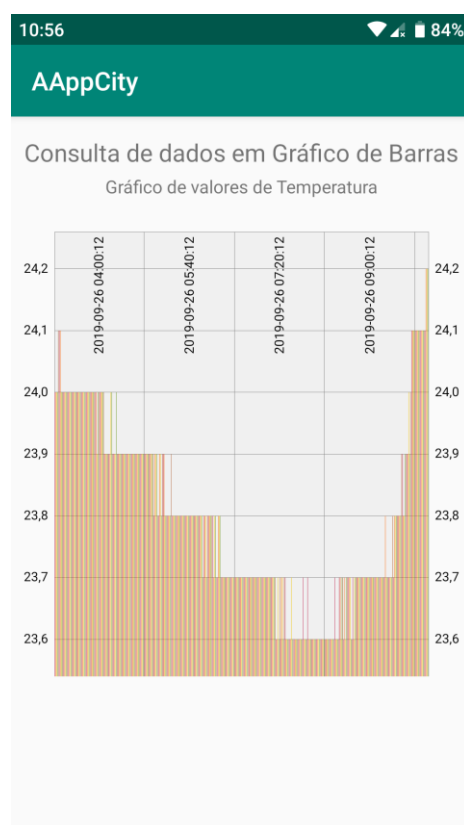
No caso de a visualização ser feita em gráficos de barras, de linhas ou circular (PhilJay/MPAndroidChart, 2019), os dados são recebidos tal como na apresentação por lista, no entanto, são organizados do evento mais antigo para o mais recente, de forma a ser imediatamente visível a evolução temporal. Os dados são, tal como já foi referido, recebidos como na visualização em lista, em formato JSON. De seguida, a aplicação faz a conversão

destes dados para um *array* que depois é enviado para uma atividade a correr em segundo plano. A atividade a correr em segundo plano é responsável por converter os valores do *array* num gráfico.

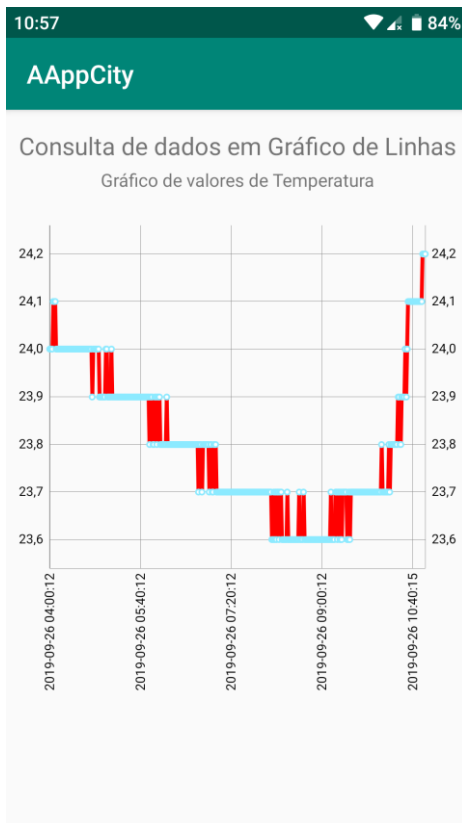
Nas atividades seguintes, desencadeadas pelos botões correspondentes à forma de visualização dos dados, são exibidos os valores que foram previamente lidos do servidor da ESTGV. A Figura 3-41 demonstra as formas de consulta dos dados, pela ordem apresentada nos botões da atividade anterior.



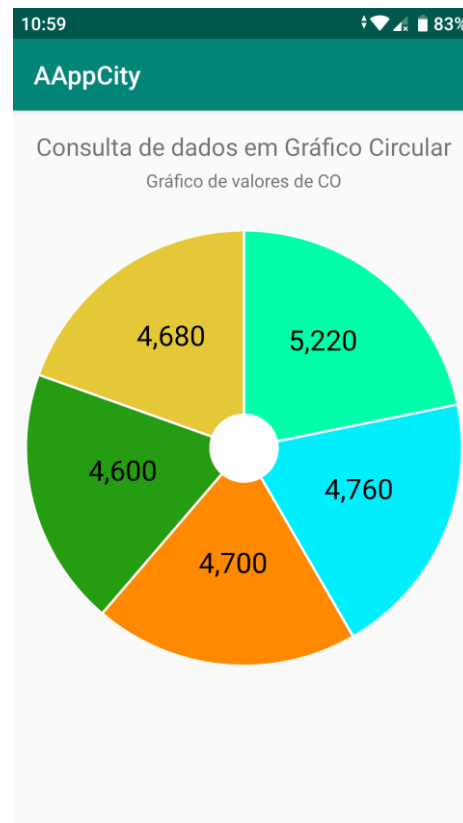
(a) Dados em lista



(b) Dados em gráfico de barras



(c) Dados em gráfico de linhas



(d) Dados em gráfico circular

Figura 3-41: Formas de consulta dos dados.

Voltando à primeira atividade, atividade principal da aplicação, o segundo botão permite enviar dados para o servidor, de forma a tornar o utilizador também numa fonte de dados. A aplicação possibilita que o utilizador emita uma opinião pessoal, por exemplo, em relação à qualidade do ar, quantitativa e não valores concretos, uma vez que não é previsível que tenha tecnologia ao seu alcance para reportar valores credíveis e com a cadência que se espera de todo o sistema. A Figura 3-42 demonstra uma avaliação da qualidade do ar.

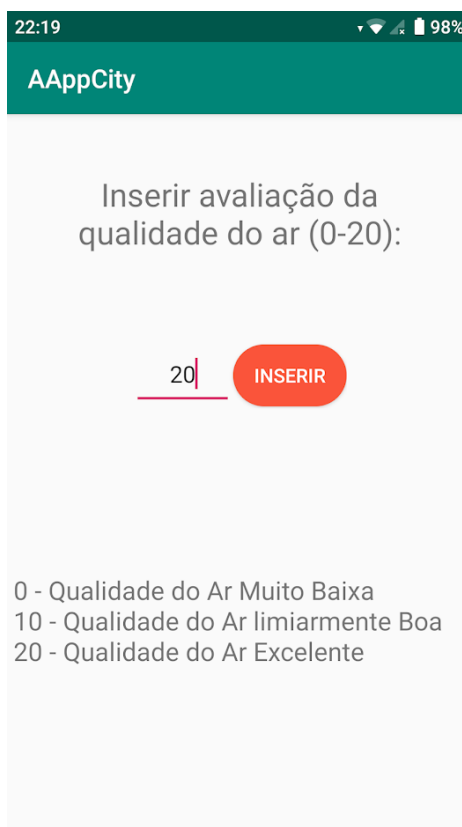


Figura 3-42: Envio de avaliação da qualidade do ar.

Para além dos elementos já referidos relativamente à visualização gráfica, existem ainda outros como os botões, que foram adaptados, tanto em forma como em cor, com o objetivo de tornar a aplicação mais apelativa ao uso e gerar uma maior harmonia gráfica. A aplicação adapta-se a qualquer dispositivo e tanto pode ser usada na vertical como na horizontal, adaptando-se totalmente e de forma automática. A aplicação é desenhada para funcionamento em equipamentos Android desde a API 19, Android 4.4, o que segundo o software Android Studio, torna a aplicação compatível com mais de 95% dos equipamentos Android.

4. Validação e Testes

A fase de testes de desenvolvimento de um sistema procura validar e apresentar evidências da confiabilidade deste, demonstrando as capacidades do mesmo. Nem sempre este processo é interrompido, levando a sucessivos testes e melhorias.

Este capítulo tem como objetivo detalhar os testes realizados ao sistema, tanto ao nível do hardware como do software, assim como partilhar os resultados obtidos.

4.1 Hardware do Sistema

Muito embora os testes ao sistema sejam um conjunto de hardware com software, optou-se por fazer a separação dos mesmos, sendo este ponto dedicado apenas ao hardware e às possíveis implicações do mesmo no software.

Sendo vários os componentes de hardware e de diferentes fabricantes, testar o funcionamento do conjunto seria o desejável e recomendável. Assim, numa primeira fase, optou-se por colocar o sistema com o Raspberry Pi e alguns sensores ligados por períodos de 24 horas, em ambiente interior, como a Figura 4-1 e a Figura 4-2 representam. Desta forma seria possível detetar possíveis falhas que surgissem ainda antes de todos os sensores ligados. Com o envio dos dados já implementado, era possível acompanhar a evolução dos valores lidos. O sistema revelou-se capaz e ultrapassou o teste com sucesso.

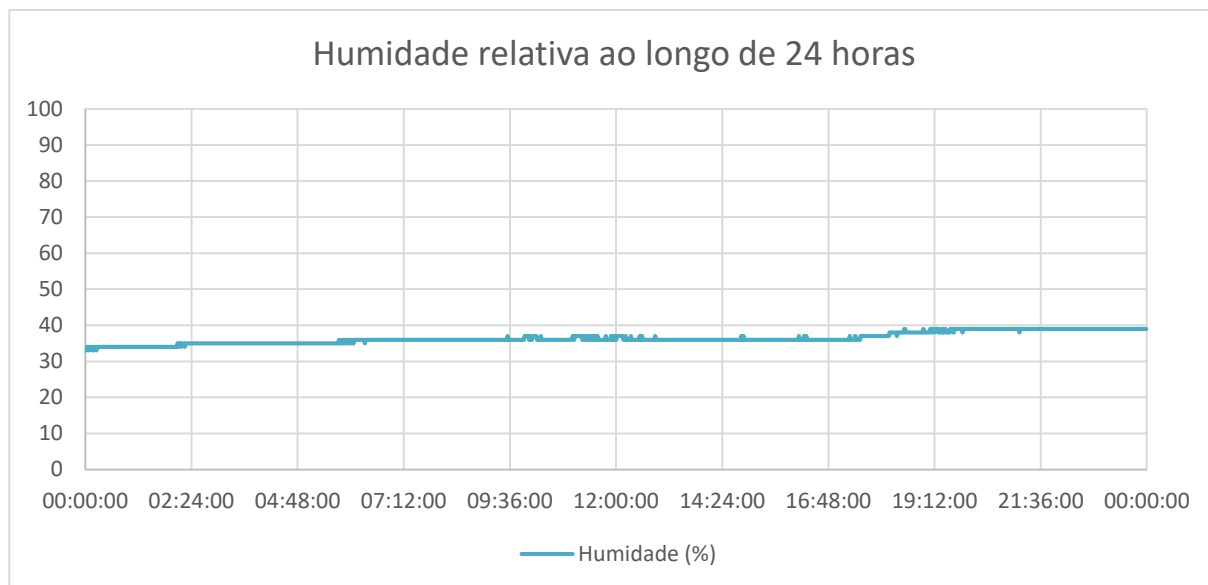


Figura 4-1: Humidade relativa ao longo de 24 horas.

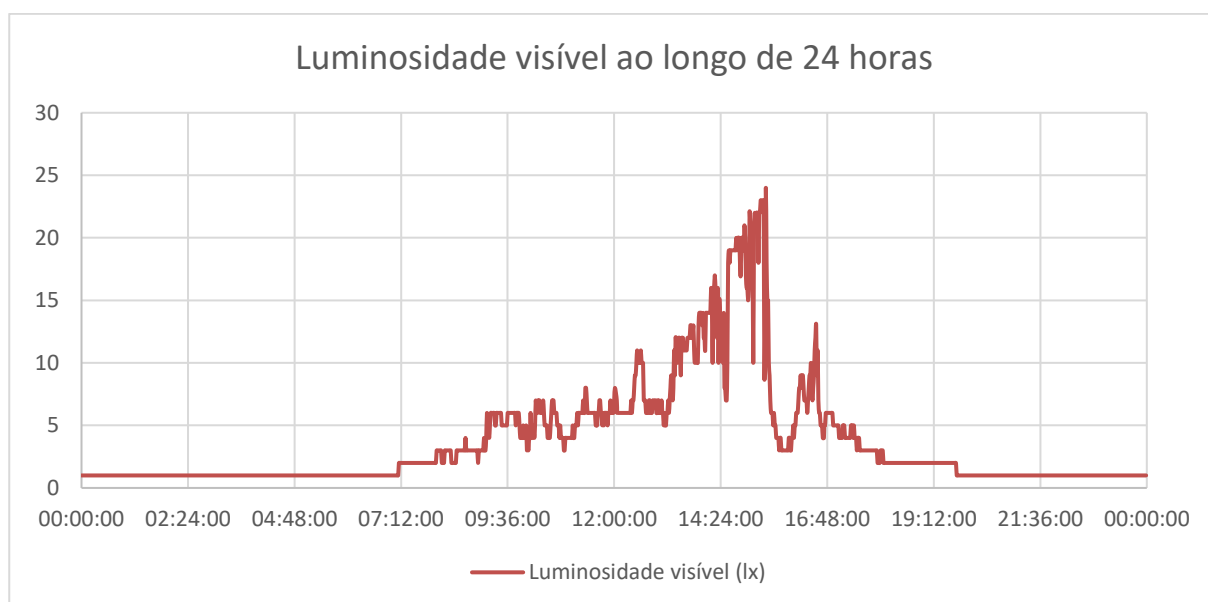


Figura 4-2: Luminosidade visível ao longo de 24 horas.

Depois dos primeiros testes, passou-se a uma segunda fase dos mesmos, onde o sistema foi deixado ligado de forma contínua, com todos os sensores e com a câmara a fazer contagem de objetos aleatórios. Após esta segunda fase de testes e com um acompanhamento contínuo dos valores lidos, chegou-se à conclusão que, de forma esporádica, dois dos sensores apresentavam leituras erróneas, dado que não era um problema cíclico e que uma reinicialização do sistema, por vezes, não resolvia, decidiu-se perceber de que forma poderia ser algum problema de hardware. Na Figura 4-3 estão representados alguns dos dados obtidos que permitiram perceber que existia um problema com o sensor MiCS-6814, os valores ficavam constantes e sempre com os mesmos valores, esporadicamente e no espaço de 1 minuto existiam grandes variações.

```

22/05/2019;00:00:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
22/05/2019;00:01:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
...
22/05/2019;08:26:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
22/05/2019;08:27:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
22/05/2019;08:28:03;332.48;0.09;4.54;25249.45;7366.28;2.67454E+12;1703.42;956.15
22/05/2019;08:29:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
22/05/2019;08:30:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
...
22/05/2019;09:41:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68
22/05/2019;09:42:03;4.53;0.15;0.7;589.17;411.38;651.99;0.75;1.68

```

Figura 4-3: Problemas do sensor MiCS-6814.

Os dois sensores em questão tinham formas de funcionamento diferentes, um era analógico e o outro tinha protocolo I²C implementado diretamente, apenas partilhavam a alimentação que era comum a todos os sensores. Após uma análise à parte física dos dois sensores, percebeu-se que os mesmos utilizavam o mesmo tipo de cabo e fichas. Eram os dois da mesma marca, um mau contacto nas fichas provocava leituras erróneas dos valores dos sensores, sendo essa a falha dos sensores.

Resolvido o problema acima descrito, conforme demonstra a Figura 4-4, efetuaram-se novos testes com o sistema a fazer contagem e classificação de veículos e com todos os sensores ligados, em ambiente interior. Foi deixado ligado por longos períodos, não apresentando qualquer problema, como se representa pela Figura 4-5.

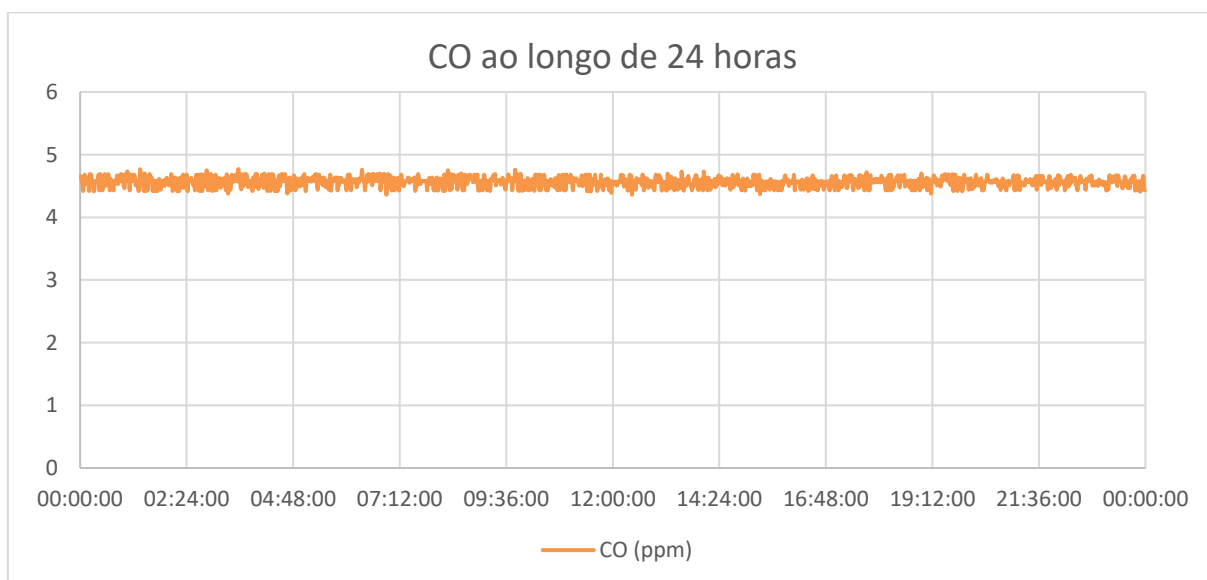


Figura 4-4: CO ao longo de 24 horas.

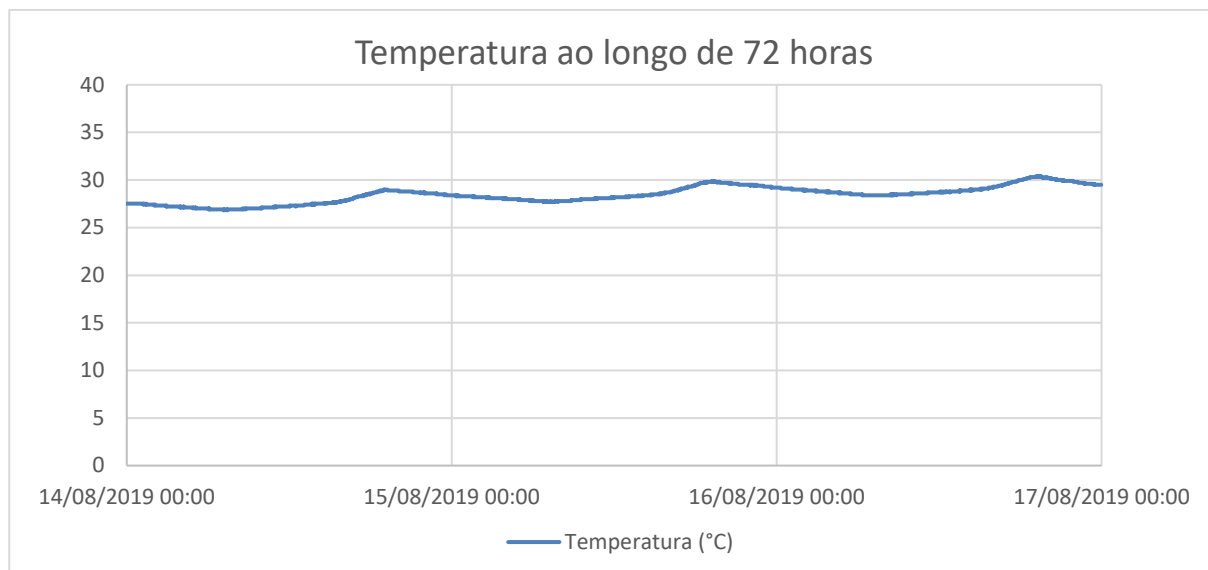


Figura 4-5: Temperatura ao longo de 72 horas.

4.2 Software do Sistema

Os testes iniciais ao software foram realizados de forma não coincidente com os testes ao hardware. Desta forma tornou-se mais fácil diagnosticar algum mau funcionamento e despistar possíveis leituras errôneas dos sensores que fossem por má programação e não por problema físico do sensor. A primeira fase de testes teve início quando surgiram as primeiras incompatibilidades dos sensores, conforme foi relatado no ponto 3.6 deste documento.

Numa segunda fase de testes, foram colocados à prova, a leitura dos sensores e a contagem de objetos aleatórios pela câmara, onde não foi diagnosticado qualquer incidente de software.

Uma vez que da parte de software não existiam problemas, considerou-se que já existia maturidade suficiente para os testes reais para se diagnosticar melhor as falhas. A captação de imagem em tempo real foi substituída por um vídeo, gravado através do Raspberry Pi com a câmara usada no processamento de imagem e a utilizar a mesma resolução. Este teste será mais detalhado em seguida.

Relativamente à classificação de veículos, foi necessária a calibração do sistema, para ser possível distinguir as três classes de veículos. Após este passo, foi deixado o sistema a analisar o vídeo para serem detetados os pontos a melhorar.

Um primeiro ponto que mereceu a atenção foi o referente ao tempo de aprendizagem do bloco de deteção do primeiro plano, pois por ser um tempo demasiado curto não permitia detetar os veículos, conforme se pode ver na Figura 4-6.

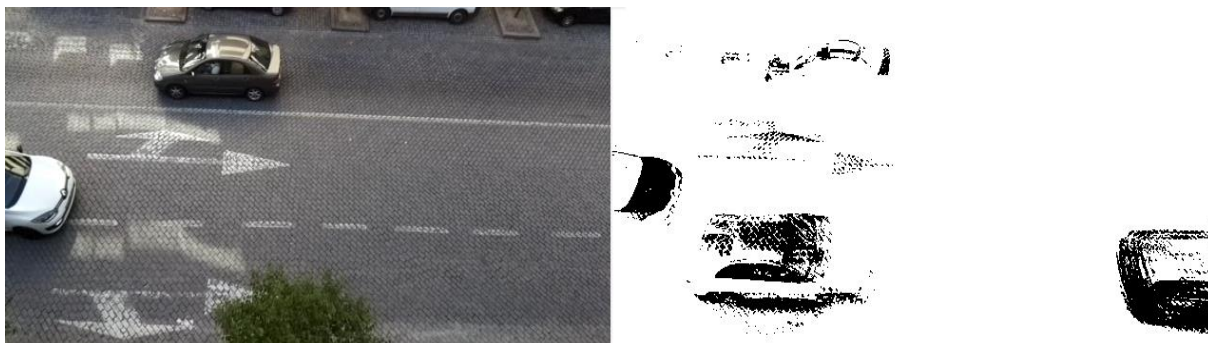


Figura 4-6: Tempo de aprendizagem baixo provoca má leitura do primeiro plano.

Após alteração ao tempo de aprendizagem deste bloco para um valor em que se considerou que estaria funcional, colocou-se o sistema novamente em teste. Neste ensaio foram analisadas algumas situações que se consideraram que poderiam levar mais facilmente a leituras erróneas, sendo elas:

- Situação de pessoas ou bicicletas na via;
- Situação de estacionamento ou paragem;
- Situação de fila com trânsito parado ou em marcha lenta;
- Situação de veículo pesado coincidente com ligeiro ou motociclo.

Para cada um dos pontos anteriores, serão apresentadas *frames* que procuram dar resposta à situação referida. Na Figura 4-7 estão duas pessoas a atravessar a via, ainda que de forma perpendicular ao trânsito, não sendo classificadas para contagem, uma vez que a área que ocupam é inferior à mínima considerada para identificação.

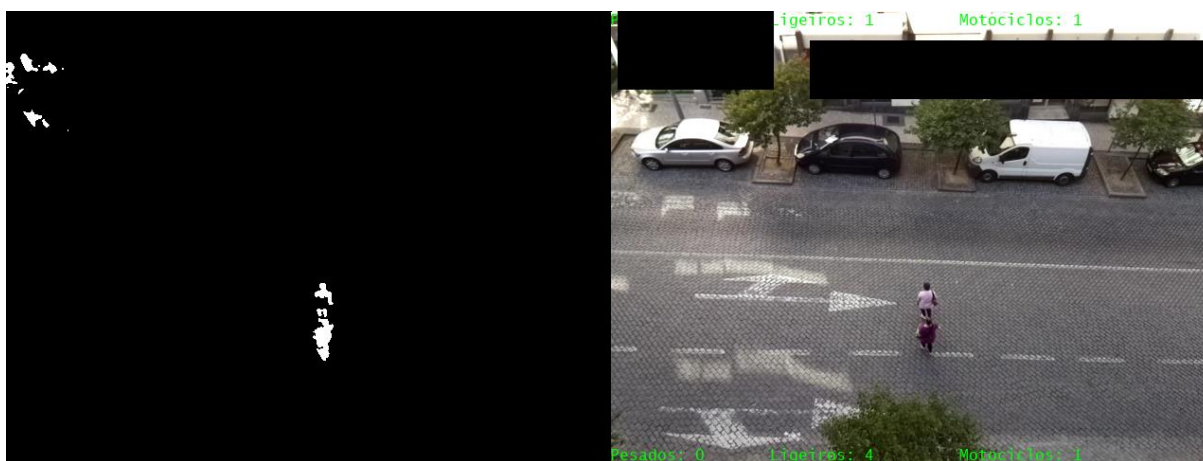


Figura 4-7: Peões a atravessar a faixa, não sendo contabilizados.

Ainda dentro do conjunto de situações a não considerar para contagem, estão as bicicletas, uma vez que são veículos não poluentes na sua utilização, fazendo todo o sentido ficarem excluídos da contagem, tal como se pode observar na Figura 4-8.

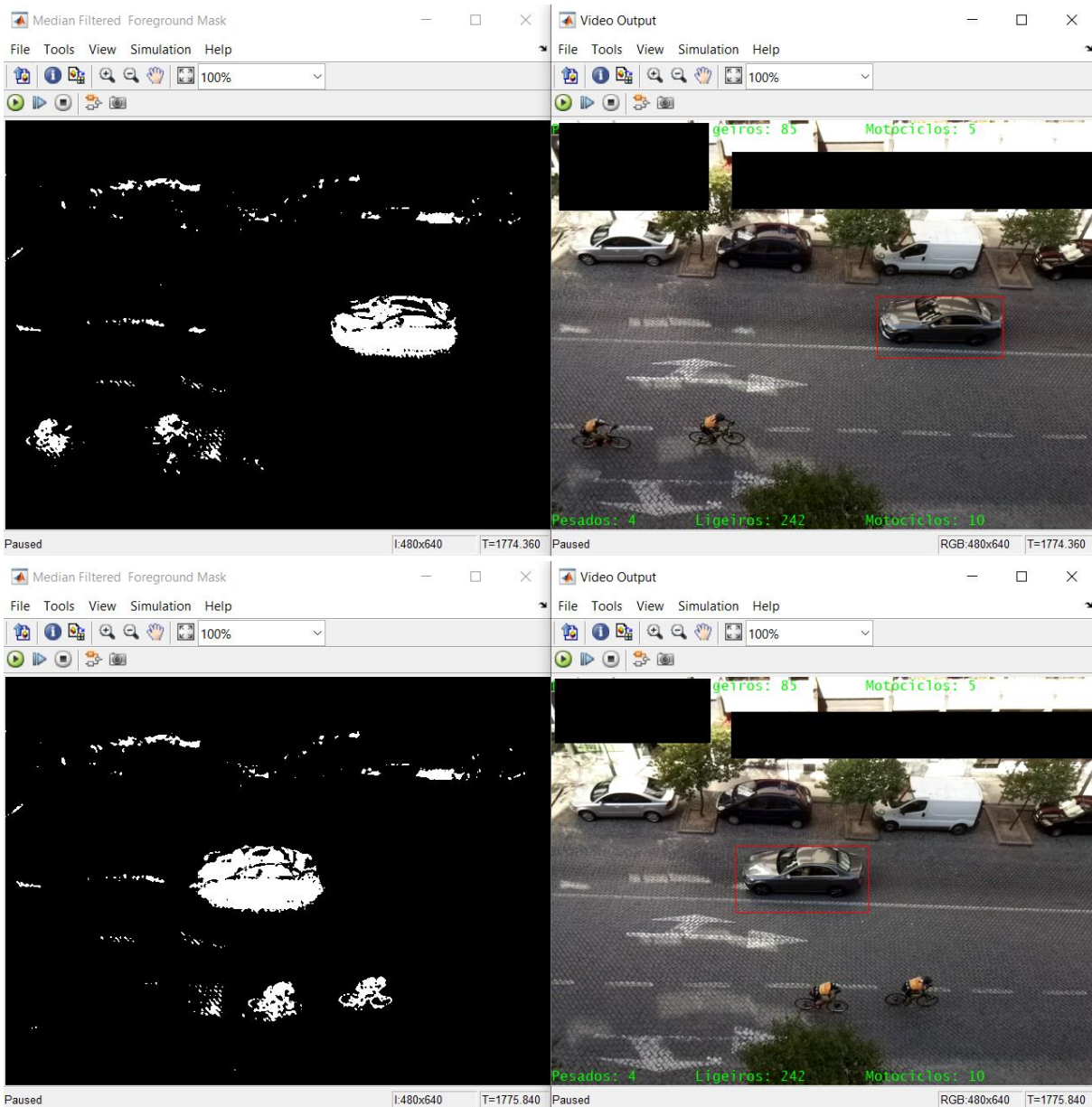


Figura 4-8: Bicicletas na via, não sendo contabilizadas.

Numa situação de estacionamento, se o veículo não percorrer mais do que determinada distância, não é contabilizado, sendo considerado apenas mais tarde, quando sair do lugar de estacionamento. Caso a distância percorrida seja superior ao valor pré-determinado, o veículo será contabilizado antes de estar estacionado, sendo que no momento em que se ausentar do lugar de estacionamento, não deverá ser contabilizado, uma vez que já não terá distância suficiente a percorrer para que seja contabilizado.

Na Figura 4-9 é possível se observar um carro em manobra de estacionamento, como o espaço percorrido não foi maior do que o definido, o veículo não foi contabilizado.

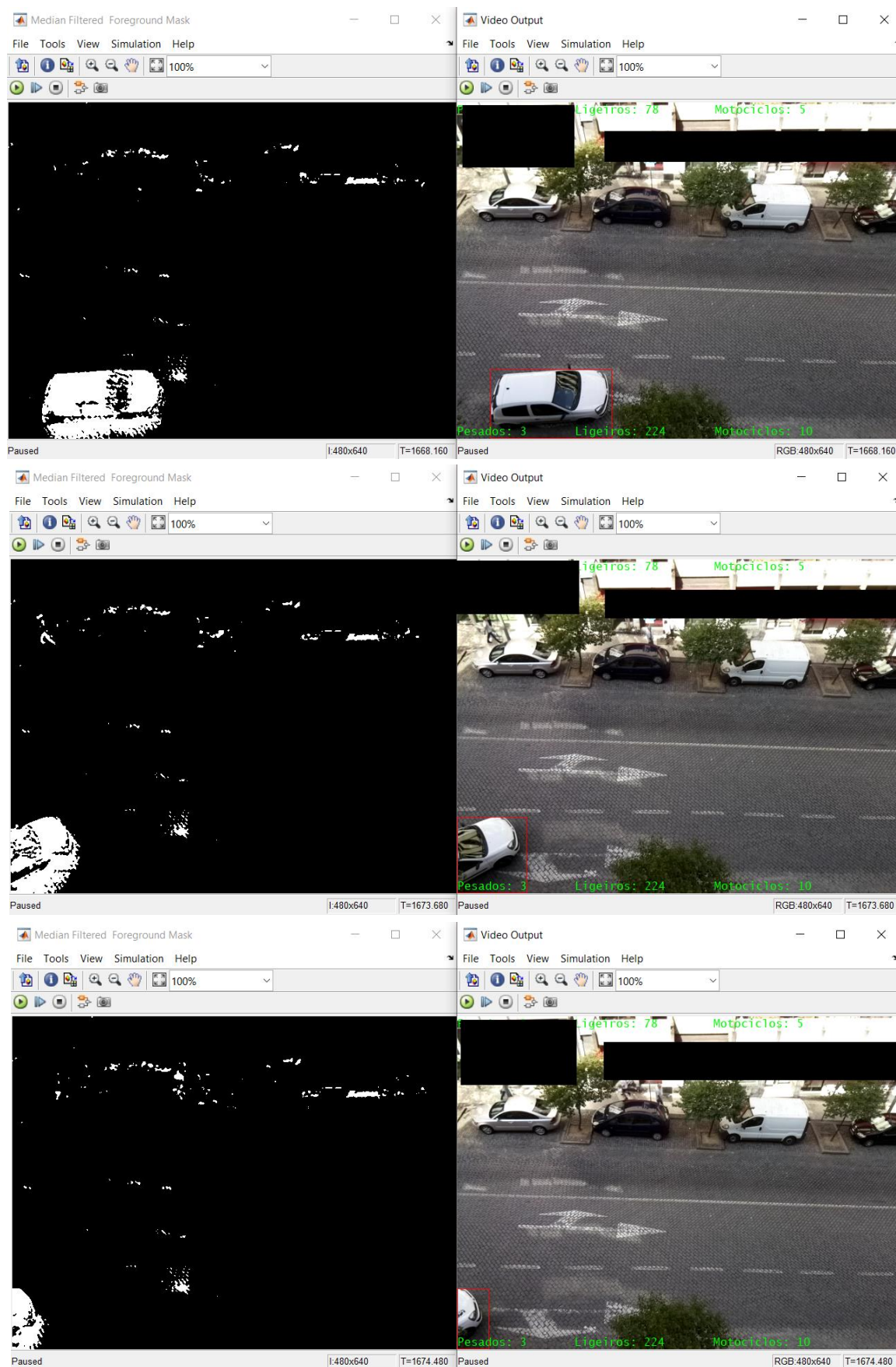


Figura 4-9: Estacionamento paralelo à via, não sendo contabilizado.

Relativamente a uma situação de paragem na via, conforme a Figura 4-10, seja prolongada ou apenas provocada por fila de trânsito ou marcha lenta, como o veículo não desaparece da imagem, o mesmo não é contabilizado. Sendo a contagem feita quando desaparece da imagem, uma situação diferente da referida anteriormente em que o estacionamento se encontra fora do campo de visão da câmara.

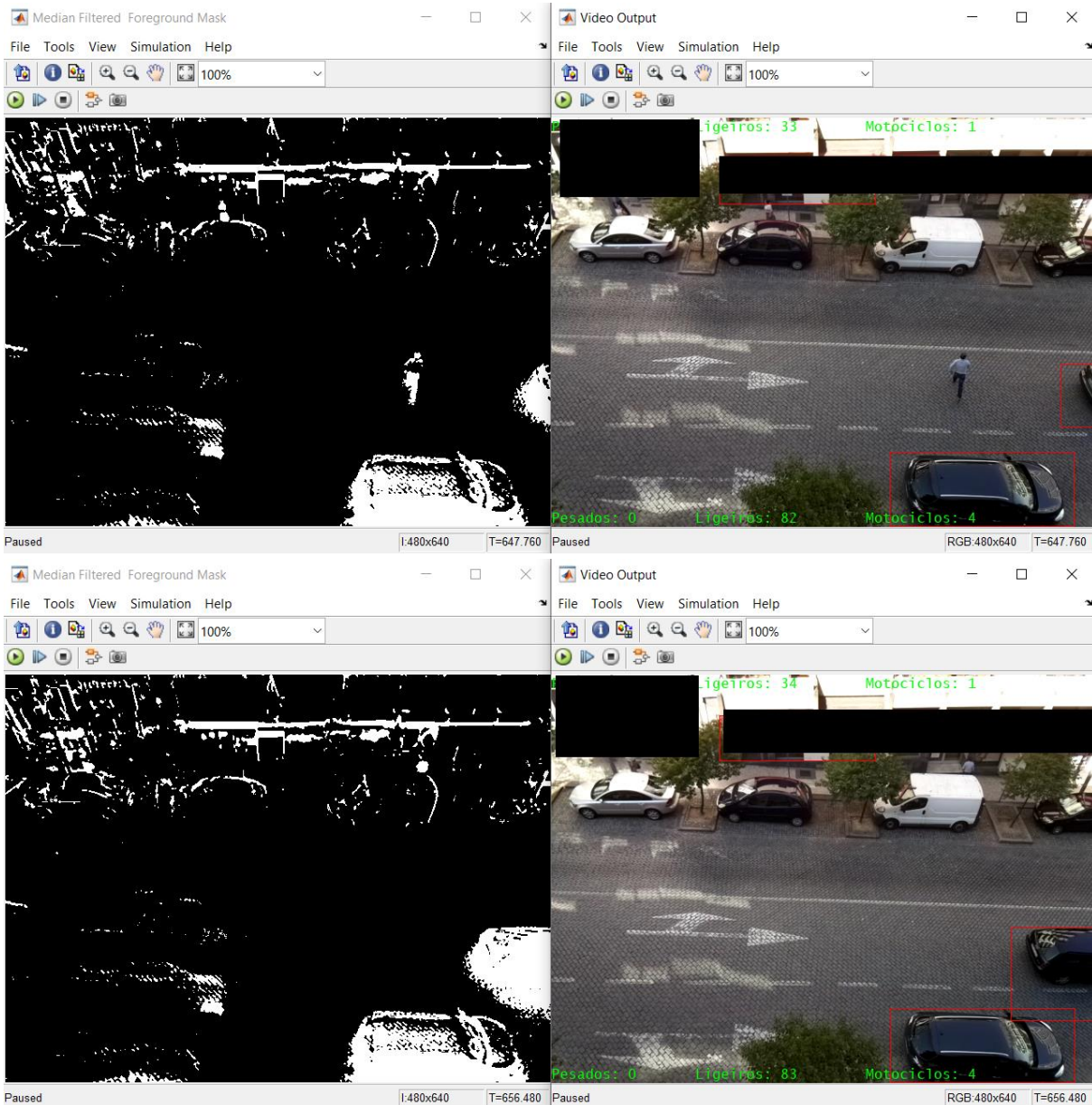


Figura 4-10: Paragem na via, aguarda ser contabilizado.

Tal como referido, também numa situação de fila de trânsito parado ou em marcha lenta, a forma de contabilizar é a mesma e como tal, o veículo só será incrementado no contador no momento em que sai do campo de visão da câmara. Na Figura 4-11 encontra-se uma fila de trânsito parado, a contabilização do primeiro veículo desta fila só aconteceu no momento em que deixou de aparecer na imagem.

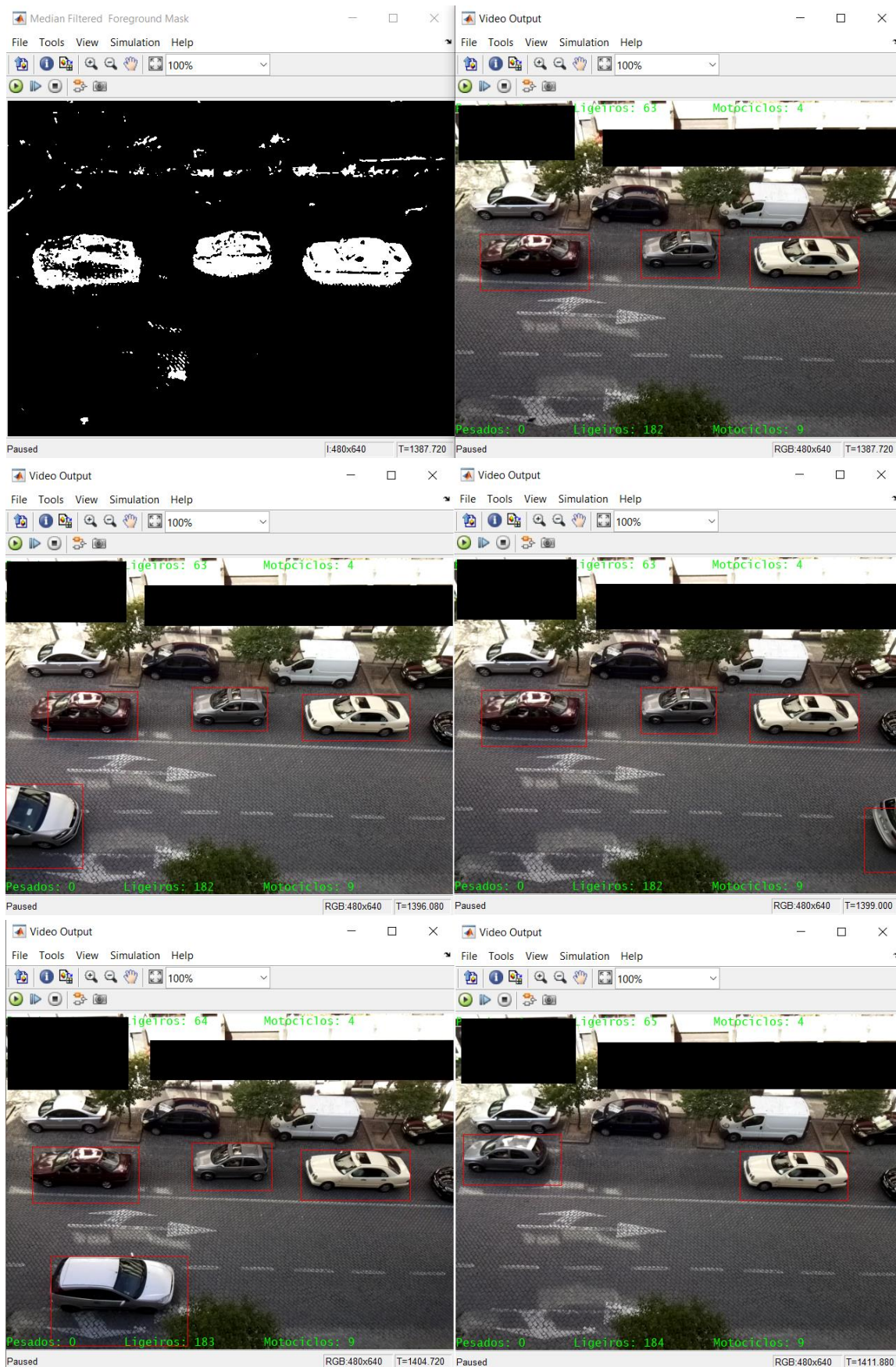


Figura 4-11: Fila de trânsito parado, aguarda ser contabilizado.

Existem situações em que a contagem por processamento de imagem não é totalmente eficaz, seja por objetos que limitam a visão da mesma, por exemplo a árvore que se encontra na parte inferior das imagens anteriores e seguintes, seja por motivos da própria estrutura e forma de instalação. Seguidamente serão apresentados dois exemplos onde o sistema não foi totalmente eficaz, na Figura 4-12, um veículo ligeiro é absorvido por um veículo pesado e dessa forma contabilização é feita de apenas um pesado. Esta falha poderá ser minimizada com uma alteração ao sistema, utilizando também a cor para separar veículos.

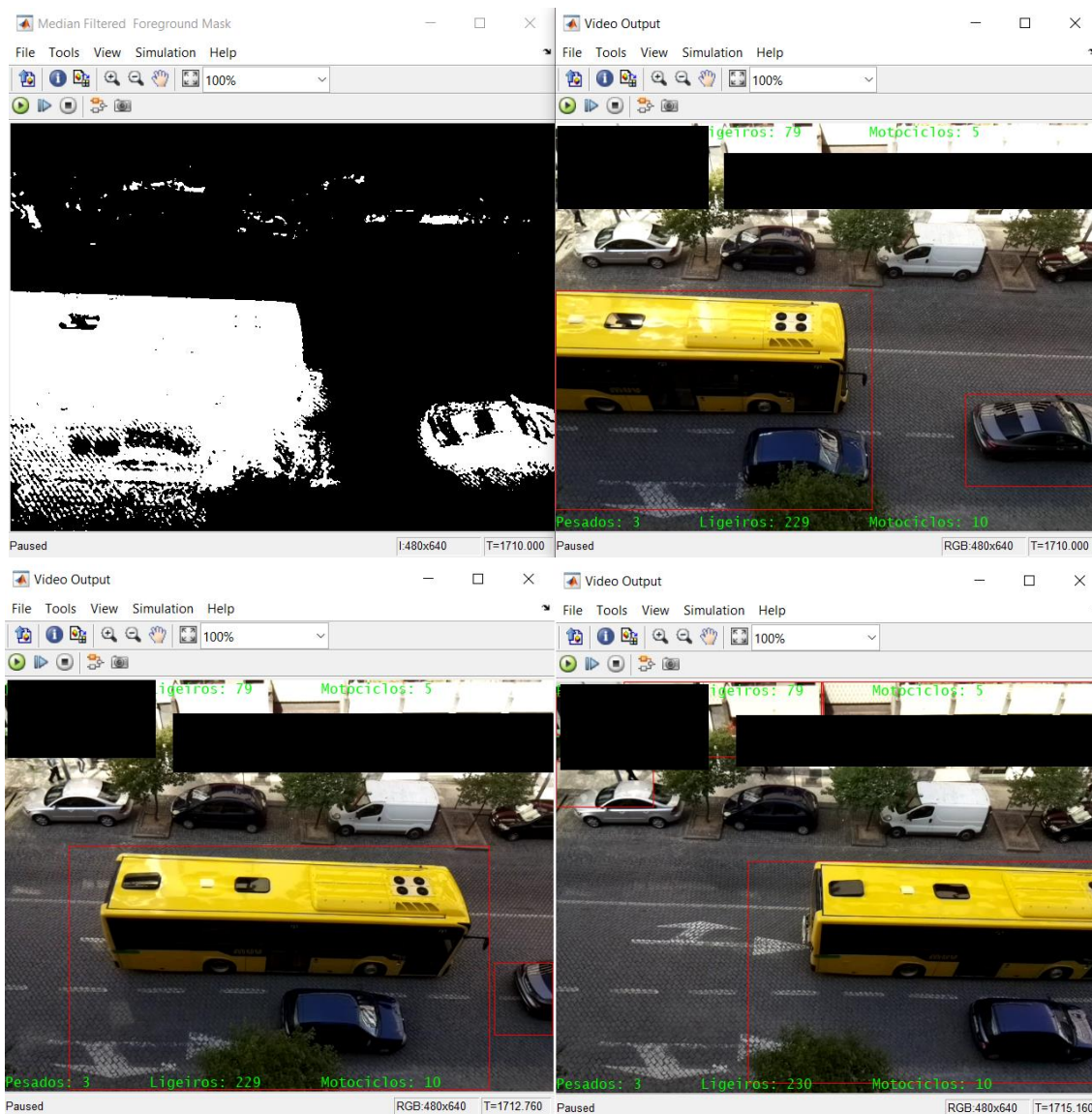


Figura 4-12: Veículo ligeiro absorvido por veículo pesado.

A Figura 4-13 trata-se de uma situação em que por uma alteração da posição da câmara o sistema fica “cego”, não identificando e por consequência não contabilizando qualquer veículo. A solução passaria por fixar a câmara num suporte mais robusto onde esta situação seja minimizada.

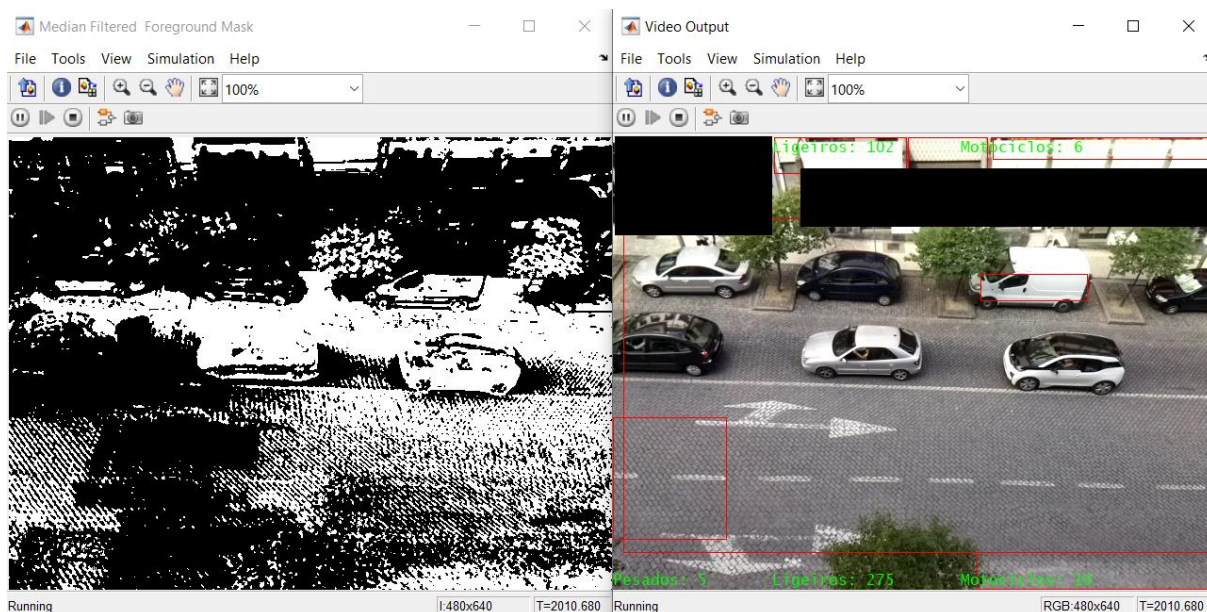


Figura 4-13: Sistema cego sem conseguir identificar veículos.

Foi colocado em teste um vídeo com uma duração de 30 minutos, a Tabela 4-1 apresenta os resultados obtidos que serão alvo de análise mais a frente.

Tabela 4-1: Testes ao sistema de contagem de veículos, por vídeo.

Veículos	Sentido	Contagem real	Contagem do sistema	Diferença	%
Pesados	Esq-Dir	4	4	0	0,00
	Dir-Esq	3	3	0	0,00
	Total	7	7	0	0,00
Ligeiros	Esq-Dir	254	253	-1	-0,39
	Dir-Esq	111	103	-8	-7,21
	Total	365	356	-9	-2,47
Motociclos	Esq-Dir	7	7	0	0,00
	Dir-Esq	4	3	-1	-25,00
	Total	11	11	-1	-9,09
Total	Total	383	373	-10	-2,61

No geral as diferenças não são significativas, especialmente tendo em conta que ainda durante o tempo de adaptação do sistema foram feitas contagens e que num período temporal superior, esse valor seria diluído. Há ainda a considerar que durante algum tempo o sistema ficou parcialmente “cego”, devido a uma deslocação temporária da câmara, não conseguindo contar e classificar corretamente todos os veículos, tendo como consequência a não contabilização de alguns veículos e a má classificação de outros.

4.3 Aplicação Móvel

Finalizados todos os testes feitos ao sistema desenvolvido, importava também provar a eficácia da aplicação móvel desenvolvida para o sistema Android. Numa primeira fase foi instalada a aplicação em dispositivos móveis distintos, com características de hardware diferentes e com sistemas operativos recentes e mais antigos. Foi ainda testada a ligação com uso do Wi-Fi e com uso dos dados móveis, a rotação de ecrã, com o equipamento na vertical e com o mesmo na horizontal. Na Figura 4-14 é possível se observar o funcionamento da leitura dos dados num equipamento Android com API 22, um ecrã de 4 polegadas e ligação por dados móveis.

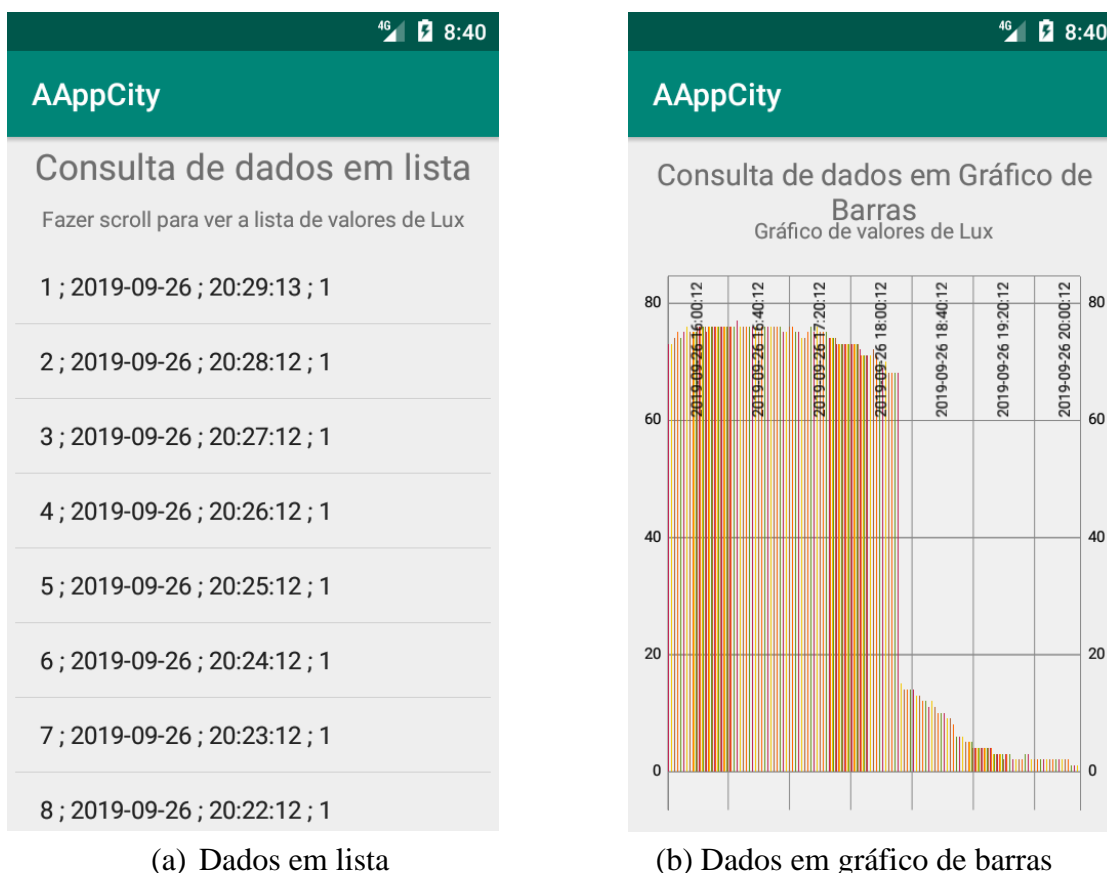


Figura 4-14: Consulta em equipamento Android de 4 polegadas com API 22.

No que respeita ao envio de valores para o servidor, também foram realizados testes com sucesso, conferidos pela resposta deste, visível na Figura 4-15 e no acesso posterior ao servidor.

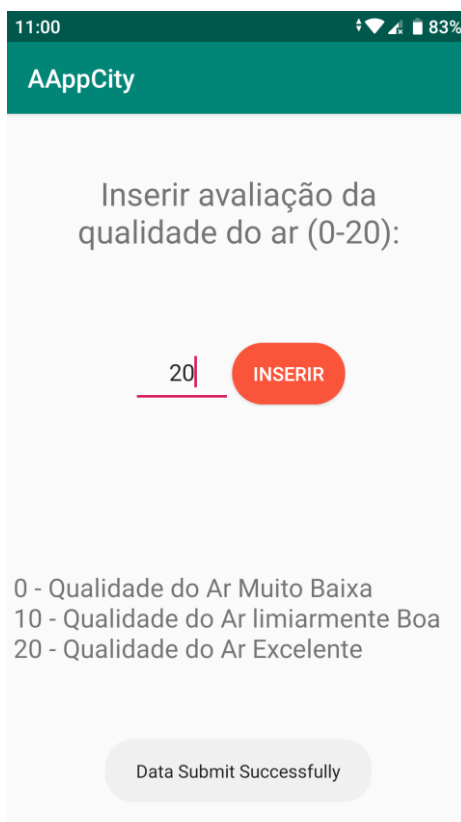


Figura 4-15: Envio de dados para o servidor com sucesso.

Para além dos testes em smartphones, foi incluído um tablet de 10 polegadas para, de igual forma, provar o funcionamento da aplicação nos mais variados tipos de dispositivos Android, com o tablet foi também testada a ligação por Wi-Fi, conforme pode ser visto na Figura 4-16.

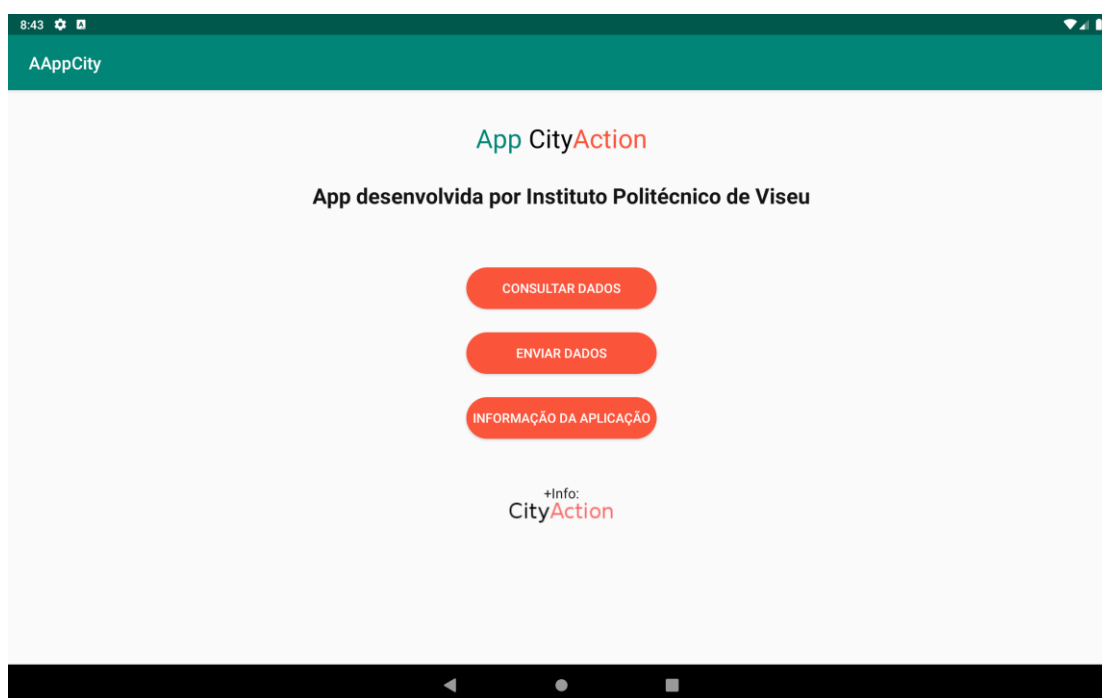


Figura 4-16: Atividade principal vista em tablet na horizontal.

Na visualização de gráficos a aplicação também se mostrou totalmente capaz e tornou a visualização destes mais agradável quando o número de valores é mais elevado, conforme a Figura 4-17 e a Figura 4-18.

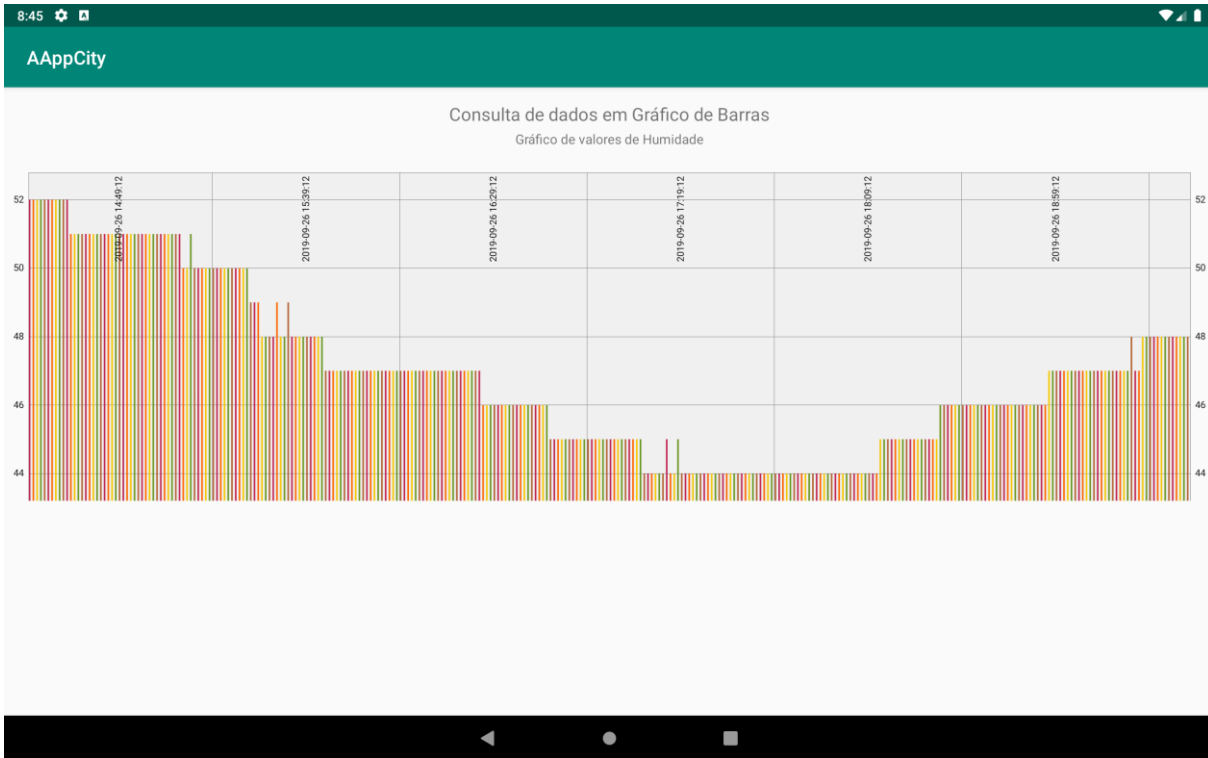
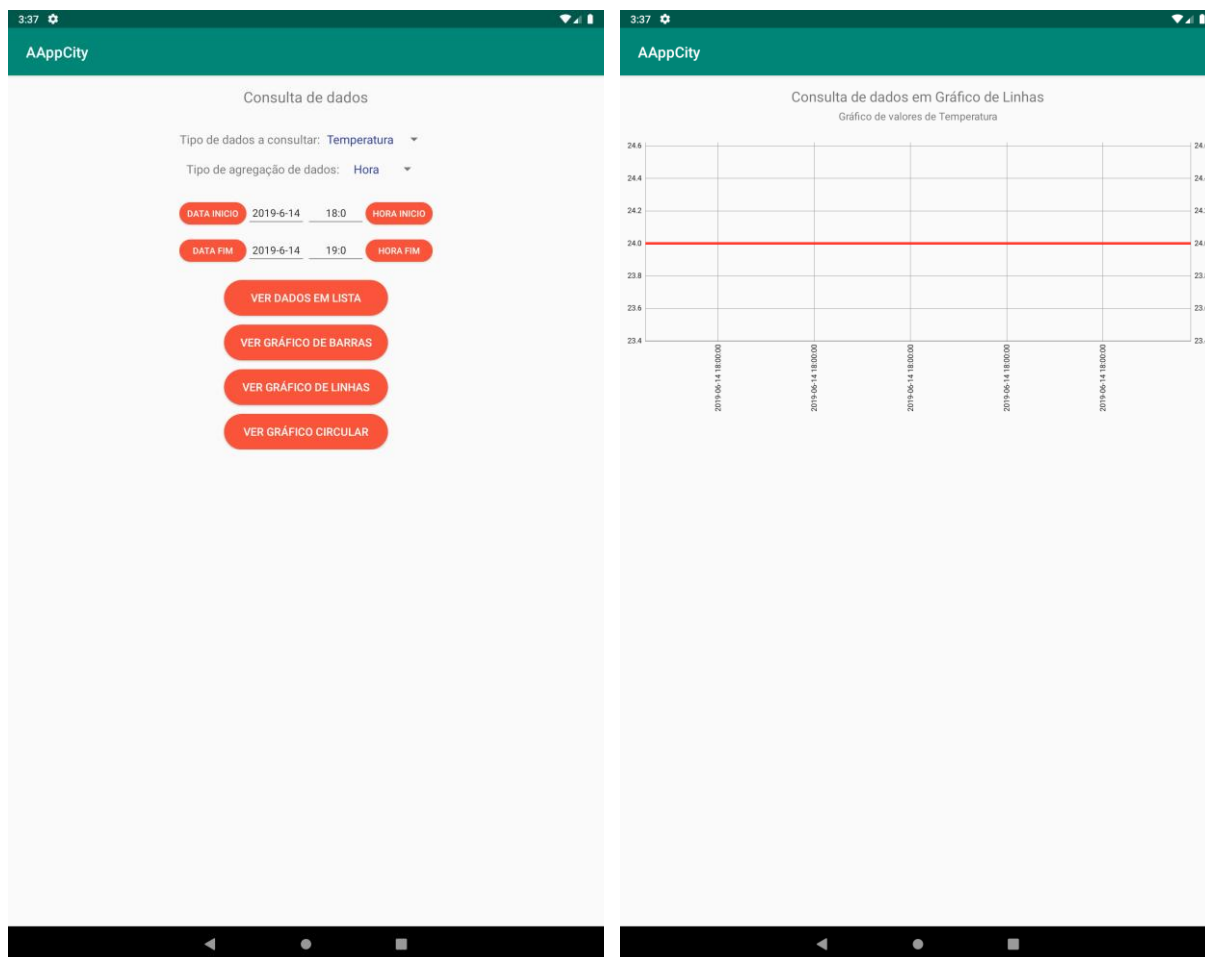


Figura 4-17: Gráfico de barras visto em tablet na horizontal.



Figura 4-18: Gráfico de linhas visto em tablet na horizontal.

Foi ainda colocado o tablet na vertical para testar a adaptação da aplicação a esta posição e ainda realizada a consulta com agregação horária Figura 4-19a, não foi encontrado qualquer problema e a visualização manteve-se confortável, conforme Figura 4-19b.



(a) Seleção para consulta dos dados

(b) Gráfico de linhas da agregação

Figura 4-19: Consulta de agregação horária com o tablet na vertical.

5. Conclusão e Trabalhos Futuros

Tratando-se do último capítulo deste documento, nele serão apresentadas as conclusões e os possíveis trabalhos futuros a serem desenvolvidos.

No primeiro ponto serão abordadas as conclusões gerais no que diz respeito ao desenvolvimento, funcionamento, bem como à temática em si. No ponto 5.2 serão apresentados os possíveis trabalhos futuros a serem desenvolvidos que visam melhorar o funcionamento e potencializar as capacidades do sistema em geral, do ponto de vista funcional e estético.

5.1 Conclusão

Relativamente às cidades inteligentes e a tudo o que as envolve, conclui-se que se trata de uma área ainda em crescimento e maturação. Certamente que nos próximos anos, com a chegada de novas tecnologias e novos desafios, as cidades ficarão ainda “mais inteligentes” e tornar-se-ão uma grande ajuda, não só para os municípios que poderão fazer uma melhor gestão e dinamização das cidades, como para os cidadãos que poderão ver o seu dia-a-dia mais facilitado, tendo um papel ainda mais ativo nas mesmas. Também do ponto de vista da saúde, as pessoas estão cada vez mais preocupadas, como tal, soluções que permitam a estas ter uma informação sobre a qualidade do ar que respiram, serão certamente bem acolhidas.

No que ao conteúdo do presente trabalho diz respeito, foi apresentada uma solução versátil e eficaz para um sistema de monitorização de tráfego e parâmetros ambientais, utilizando tecnologias existentes e que se apresentam fiáveis. Os sensores utilizados mostraram-se capazes de dar resposta à leitura de parâmetros ambientais. Pela forma como o sistema foi projetado e desenvolvido, é possível substituir os sensores utilizados por outros, bastando para

tal que os mesmos tenham implementado I²C ou apresentem uma saída analógica. A monitorização de tráfego revelou-se eficaz e com capacidade para se adaptar a diferentes vias de trânsito, sendo o número de configurações a fazer relativamente baixo e fácil. Para além do sistema físico, a solução contou também com o desenvolvimento de uma aplicação móvel para Android que permite ao utilizador manter-se informado e ter um papel mais ativo na comunidade, permitindo a esta não só consultar os dados recolhidos pelo sistema mas também dar o seu contributo dando a sua opinião, através da aplicação, em relação à qualidade do ar.

Quanto ao resultado final global, o sistema revelou ser capaz de dar uma boa resposta ao problema, funcionando de forma contínua e sem necessidade de ajustes constantes. Os sensores usados revelaram-se aptos para realizar a tarefa para a qual foram selecionados, assim como a câmara, que em momento algum demonstraram falta de capacidade para dar resposta ao problema inicial. Relativamente aos sensores, não foi possível concluir se os valores obtidos apresentam um erro elevado, por não ter sido possível aceder a sensores certificados e calibrados. No que diz respeito ao processamento de imagem, conseguiram-se algumas classificações de veículos com erros inferiores a 1%, sendo que o valor final de objetos contabilizados apresenta um erro inferior a 5% face ao valor real, com uma amostra maior seria possível o erro diminuir.

A aplicação móvel provou ser intuitiva na consulta e envio dos dados, compatível com vários equipamentos e capaz de dar uma visão geral não só da aplicação, mas da solução desenvolvida. Foi definitivamente a ponte necessária para ligar o utilizador aos parâmetros avaliados pelo sistema. Destaque ainda para os gráficos gerados na aplicação que se revelaram eficazes, de fácil e agradável leitura dos dados.

Num futuro não muito distante é possível que as cidades apostem em soluções como a desenvolvida, trabalhando continuamente para melhorar a qualidade do ar que as rodeia e garantindo aos habitantes e visitantes um ar de qualidade na manutenção de uma boa saúde.

5.2 Trabalhos Futuros

Tendo em conta que todos os objetivos propostos foram atingidos, não existiria a necessidade da execução de trabalhos futuros, uma vez que o sistema se mostra funcional e com capacidade de resposta para o problema proposto. No entanto, terminado o desenvolvimento e teste das capacidades do sistema, surgiram possíveis inovações e melhorias a aplicar, que são:

1. Substituição de alguns dos sensores;
2. Implementação de um sistema de alimentação independente;
3. Alteração do sistema de comunicação;
4. Melhorar o sistema de contagem e classificação de veículos;
5. Melhorar a aplicação móvel.

Relativamente à substituição de alguns dos sensores não se trata de uma falha do sistema mas a implementação de novos sensores, referenciados na ISO 37120:2018, e com calibração e certificação dos mesmos tornaria o sistema mais completo.

A implementação de um sistema de alimentação independente permitiria autonomia da rede elétrica, uma melhoria a implementar seria o dimensionamento de baterias com o carregamento destas a ser feito por energias renováveis, por exemplo com a instalação de um painel solar fotovoltaico.

Na sequência da autonomia da rede elétrica, surge a alteração do sistema de comunicação, a ligação ethernet implica cabos, a rede Wi-Fi implica uma boa cobertura, desta forma surge a melhoria de alterar o sistema de comunicação utilizando outras tecnologias como por exemplo NB-IoT, LoRaWAN ou mesmo a futura quinta geração de redes móveis (5G).

Relativamente à melhoria do sistema de contagem e classificação de veículos, já foi referido neste documento que o sistema de processamento de imagem pode ganhar com algumas alterações, utilizando por exemplo a cor dos veículos para os separar mais facilmente na imagem. Outra melhoria diz respeito ao funcionamento em condições atmosféricas adversas, recorrendo por exemplo a uma câmara térmica para captação da imagem.

A melhoria da aplicação móvel seria, por um lado, na aplicação já desenvolvida melhorar o aspeto gráfico e a otimização de funções, poderiam ser adicionadas capacidades de leitura dos valores dos sensores do equipamento, por exemplo ruído através do microfone. Outra melhoria dentro das aplicações móveis seria o desenvolvimento de uma aplicação para iOS, de forma a abranger um maior número de utilizadores. A substituição da aplicação por uma Progressive Web App seria a melhoria ideal de forma a abranger mais utilizadores, uma vez que é compatível com os mais variados sistemas operativos, sendo o acesso feito por *browser*.

REFERÊNCIAS

- Adafruit ADS1115. (2019). Adafruit Industries - ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier. Retrieved August 27, 2019, from <https://www.adafruit.com/product/1085>
- Adafruit BME280. (2019). Adafruit BME280 I2C or SPI Temperature Humidity Pressure Sensor. Retrieved August 27, 2019, from <https://www.adafruit.com/product/2652>
- Adafruit SGP30. (2019). Adafruit SGP30 Air Quality Sensor Breakout - VOC and eCO2. Retrieved August 27, 2019, from <https://www.adafruit.com/product/3709>
- Aeroqual. (2019). AQM - Ambient Air Quality Monitoring System | Aeroqual. Retrieved September 10, 2019, from <https://www.aeroqual.com/outdoor-air-quality/aqm-stations>
- Agência Europeia do Ambiente. (2018). *Air quality in Europe — 2018 report. EEA Report No 12/2018*. European Union. Retrieved from <https://www.eea.europa.eu/publications/air-quality-in-europe-2018>
- Agency for Toxic Substances and Disease Registry. (2004). AMMONIA - 3 . HEALTH EFFECTS. Agency for Toxic Substances and Disease Registry.
- Air Quality Egg. (2019). Air Quality Egg - The Egg. Retrieved April 1, 2019, from <https://airqualityegg.com/egg>
- AirPi. (2019). AirPi. Retrieved April 1, 2019, from <http://airpi.es/index.php>
- AirPi PCB. (2019). Airpi | News & Updates. Retrieved September 10, 2019, from <https://airpies.wordpress.com/>
- AlticeLabs SmartIoT. (2019). SmartIoT. Retrieved August 27, 2019, from <https://iot.alticelabs.com/>
- AlticeLabs SmartIoT Autenticação. (2019). SmartIoT Authentication. Retrieved from <https://iot.alticelabs.com/docs/authentication/intro/>
- AlticeLabs SmartIoT Quick Start Guide. (2019). Quick Start Guide. Retrieved September 15, 2019, from <https://iot.alticelabs.com/docs/basics/quick-start/>
- Altitude, T. (2019). Sensly HAT. Retrieved April 1, 2019, from <https://altitude.tech/sensly/>
- Altitude Tech. (2019). SENSLY HAT FOR THE RASPBERRY PI AIR QUALITY & GAS DETECTOR V1.1: 9 Steps. Retrieved September 10, 2019, from <https://www.instructables.com/id/SENSLY-HAT-FOR-THE-RASPBERRY-PI-AIR-QUALITY-GAS-DE-1/>
- Android Studio. (2019). Android Studio and SDK tools | Android Developers. Retrieved August 27, 2019, from <https://developer.android.com/studio>
- AQM 65. (2019). Aeroqual - Ambient Air Monitoring Station. Retrieved April 1, 2019, from <https://www.aeroqual.com/product/aqm-65-air-quality-monitoring-station>
- Array of Things. (2019). Array of Things. Retrieved April 1, 2019, from <https://arrayofthings.github.io/index.html#>
- Boletim Climatológico Sazonal IPMA. (2015). Boletim Climatológico Sazonal - Instituto Português do Mar e da Atmosfera, I.P.
- Bosch BME280. (2019). Bosch BME280. Retrieved August 27, 2019, from https://www.bosch-sensortec.com/bst/products/all_products/bme280
- Carvalho, A. P. O. de, & Rocha, C. A. A. C. da. (2008). *Manual técnico para elaboração de planos municipais de redução de ruído*. Retrieved from <http://repositorio-aberto.up.pt/handle/10216/6639>
- Coelho, P. I. C. (2014). *Exposição aos compostos orgânicos voláteis - Trabalhadores em cozinhas escolares*. Retrieved from <http://repositorio.ipl.pt/handle/10400.21/3877>

REFERÊNCIAS

- Comissão Europeia. (2019). Road transport: Reducing CO2 emissions from vehicles | Climate Action. Retrieved August 27, 2019, from https://ec.europa.eu/clima/policies/transport/vehicles_en
- Curl. (2019). curl. Retrieved August 27, 2019, from <https://curl.haxx.se/>
- DecentLab. (2019). Air quality station. Retrieved April 1, 2019, from <https://www.decentlab.com/products/air-quality-station>
- DecentLab - Air Quality Station. (2019). Air Quality Station NO2 , NO, CO, Ox for LoRaWAN™ — Decentlab. Retrieved September 10, 2019, from <https://www.decentlab.com/products/air-quality-station-no2-no-co-ox-for-lorawan>
- DexterInd/GrovePi. (2016). GrovePi/multichannel_gas_sensor.py at master · DexterInd/GrovePi · GitHub. Retrieved August 27, 2019, from https://github.com/DexterInd/GrovePi/blob/master/Software/Python/grove_i2c_multichannel_gas_sensor/multichannel_gas_sensor.py
- DFRobot Analog Sound Sensor. (2019). Analog Sound Sensor For Arduino - DFRobot. Retrieved August 27, 2019, from <https://www.dfrobot.com/product-83.html>
- Diamond Traffic Inductive Loop Counters. (2019). Inductive Loop Counters | Diamond Traffic Products. Retrieved August 27, 2019, from <https://diamondtraffic.com/productlist/Inductive-Loop-Counters>
- Diamond Traffic Road Tube. (2019). Road Counters | Road Tube Counters | Traffic Counting | Diamond Traffic Products. Retrieved August 27, 2019, from <https://diamondtraffic.com/productlist/Volume-Road-Tube-Counters>
- European Environment Agency. (2018). Annual mean NO2 concentrations observed at traffic stations — European Environment Agency. Retrieved August 27, 2019, from <https://www.eea.europa.eu/data-and-maps/figures/annual-mean-no2-concentration-observed-11/89643-exceedances-of-air-quality.eps>
- Eurostat. (2016). 3.1 Using road and rail. <https://doi.org/10.2785/50487>
- Federal Highway Administration. (2016). Traffic Monitoring Guide. *Fhwa*. Retrieved from <http://www.fhwa.dot.gov/policyinformation/tmguide/>
- Ferdoush, S. M. (2014). *A Low-Cost Wireless Sensor Network System Using Raspberry Pi and Arduino for Environmental Monitoring Applications*. UNIVERSITY OF NORTH TEXAS. UNIVERSITY OF NORTH TEXAS. Retrieved from https://digital.library.unt.edu/ark:/67531/metadc500182/m2/1/high_res_d/thesis.pdf
- Grove Multichannel Gas Sensor. (2019). Grove - Multichannel Gas Sensor - Seeed Wiki. Retrieved August 27, 2019, from http://wiki.seeedstudio.com/Grove-Multichannel_Gas_Sensor/
- Grove Oxygen Sensor. (2019). Grove - Oxygen Sensor(ME2-O2-Φ20). Retrieved August 27, 2019, from <https://www.seeedstudio.com/Grove-Oxygen-Sensor-ME2-O2-f20.html>
- Guri, G. (2016). *Pavement Carbon Monoxide Concentration Prediction: a Prediction Model from Traffic Flow, Fleet Velocity and Sound Pressure*. <https://doi.org/10.13140/RG.2.1.1234.2645>
- INTELI. (2012). *Índice de Cidades Inteligentes - Portugal*. Retrieved from http://www.inteli.pt/uploads/documentos/documento_1357554966_2590.pdf
- IPMA Extremos Climatológicos. (2019). Instituto Português do Mar e da Atmosfera. Retrieved August 27, 2019, from https://www.ipma.pt/pt/oclima/extremos.clima/index.jsp?page=extreme_co.xml
- ISO 37120:2018. (2018). ISO - ISO 37120:2018 - Sustainable cities and communities -- Indicators for city services and quality of life. Retrieved August 27, 2019, from <https://www.iso.org/standard/68498.html>
- ISO 37122:2019. (2019). ISO - ISO 37122:2019 - Sustainable cities and communities -

- Indicators for smart cities. Retrieved August 27, 2019, from <https://www.iso.org/standard/69050.html>
- K. Vollmer, M., Juergens, N., Steinbacher, M., Reimann, S., Weilenmann, M., & Buchmann, B. (2007). Road vehicle emissions of molecular hydrogen (H₂) from a tunnel study. *Atmospheric Environment*, 41, 8355–8369. <https://doi.org/10.1016/j.atmosenv.2007.06.037>
- Keithley Instruments. (2001). *Data Acquisition and Control Handbook* (1^o). Retrieved from https://fenix.tecnico.ulisboa.pt/downloadFile/3779571242401/DataAcquisition_KEYTHLEY.pdf
- Kuriyama, G. S., Moreira, J. C., & Silva, C. R. S. da. (1997). Nitrogen dioxide (NO₂) occupational exposure assessment on Rio de Janeiro 's city traffic wards. *Cad. de Saúde Públ.*, 13(2), 677–683.
- Leduc, G. (2008). Road Traffic Data : Collection Methods and Applications. *EUR Number: Technical Note: JRC 47967*. <https://doi.org/JRC47967-2008>
- Libelium. (2019). Libelium - Connecting Sensors to the Cloud. Retrieved September 10, 2019, from <http://www.libelium.com/>
- Libelium - Air Quality Index IoT Vertical Kit. (2019). Libelium Air Quality Index IoT Vertical Kit. Retrieved April 1, 2019, from <https://www.the-iot-marketplace.com/libelium-air-quality-index-iot-vertical-kit>
- Libelium - Waspote Plug & Sense. (2019). Waspote Plug & Sense. Retrieved April 1, 2019, from <http://www.libelium.com/development/plug-sense/documentation/waspote-plug-sense-technical-guide/>
- Martins, M. M. de A. (2015). *Contribuição para o estudo da atenuação seletiva do ruído de tráfego rodoviário*.
- Martins, N. A. S. (2018). *NÍVEL DE OXIGÉNIO EM ESPAÇOS CONFINADOS - Risco Real e Risco Percecionado*.
- MATLAB. (2019). Pricing and Licensing - MATLAB & Simulink. Retrieved August 27, 2019, from <https://www.mathworks.com/pricing-licensing.html?prodcode=ML&intendeduse=student>
- MATLAB Raspberry Pi Support. (2019). Raspberry Pi Support from MATLAB - Hardware Support - MATLAB & Simulink. Retrieved August 27, 2019, from <https://www.mathworks.com/hardware-support/raspberry-pi-matlab.html>
- MetroCount RoadPod VL. (2019). RoadPod® VL 5805 | Inductive Loop Counter by MetroCount. Retrieved August 27, 2019, from <https://metrocount.com/products/mc5805-inductive-loops-counter/>
- MetroCount RoadPod VP. (2019). RoadPod® VP | Piezoelectric Traffic Monitoring System. Retrieved August 27, 2019, from <https://metrocount.com/products/roadpodvp-permanent-piezo-classifier/>
- MetroCount RoadPod VT. (2019). RoadPod® VT - The most efficient tube counter | MetroCount. Retrieved August 27, 2019, from <https://metrocount.com/products/roadpod-vehicle-tube-classifier/>
- Mishra, N. (2015). *Characterisation, toxicity and source apportionment of atmospheric organic pollutants in urban schools*.
- Moreira, C. (2013). *Fotossíntese*, 1, 1–5.
- Nam, E. K., Jensen, T. E., & Wallington, T. J. (2004). Methane Emissions from Vehicles. *Environmental Science & Technology*, 38(7), 2005–2010. <https://doi.org/10.1021/es034837g>
- Observatório Nacional das Doenças Respiratórias. (2018). 13^oRelatório do Observatório Nacional das Doenças Respiratórias - 2018. Retrieved from

REFERÊNCIAS

- https://www.ondr.pt/files/Relatorio_ONDR_2018.pdf
- Oenema, O., Velthof, G., Amann, M., Klimont, Z., & Winiwarter, W. (2012). Emissions from agriculture and their control potentials. *TSAP Report #3, Version 1.0, DG-Environment of the European Commission*, 34. Retrieved from <http://webarchive.iiasa.ac.at/Admin/PUB/Documents/XO-12-013.pdf>
- Ohliger, T. (2018). *Poluição atmosférica e poluição sonora. União Europeia - Parlamento Europeu*. Parlamento Europeu | União Europeia. Retrieved from http://www.europarl.europa.eu/ftu/pdf/pt/FTU_2.5.5.pdf
- Organização Mundial de Saúde. (2018). OMS | Ambient air pollution. *OMS*. Organização Mundial de Saúde. Retrieved from <https://www.who.int/airpollution/ambient/en/>
- PhilJay/MPAndroidChart. (2019). PhilJay/MPAndroidChart: A powerful Android chart view / graph view library. Retrieved August 27, 2019, from <https://github.com/PhilJay/MPAndroidChart>
- PurpleAir. (2019). PurpleAir sensors. Retrieved April 1, 2019, from <https://www.purpleair.com/sensors>
- Rakowska, A., Wong, K. C., Townsend, T., Chan, K. L., Westerdahl, D., Ng, S., ... Ning, Z. (2014). Impact of traffic volume and composition on the air quality and pedestrian exposure in urban street canyon. *Atmospheric Environment*, 98, 260–270. <https://doi.org/10.1016/j.atmosenv.2014.08.073>
- Raspberry Pi Camera. (2019). Buy a Camera Module V2 – Raspberry Pi. Retrieved August 27, 2019, from <https://www.raspberrypi.org/products/camera-module-v2/>
- Reis, D. C. dos S. (2017). Execução e Gestão de Aplicações Containerizadas, 169. Retrieved from <https://repositorio-aberto.up.pt/handle/10216/109650>
- Roadometry VTC. (2019). Video Traffic Counter. Retrieved August 27, 2019, from <https://roadometry.com/>
- Rodrigues, F. B. (2014). Desenvolvimento de um sistema de supervisão e gestão para sistemas de monitorização de energia elétrica. Retrieved from <https://repositorium.sdum.uminho.pt/handle/1822/41999>
- Sankar, A. M., Devaraju, T., Kumar, M. V., & Sudharshan, P. (2017). Design of High Accurate Data Acquisition System for Real Time Monitoring of Power Grid. *International Journal of Scientific and Research Publications*, 7(7), 610–615. Retrieved from <http://www.ijsrp.org/research-paper-0717/ijsrp-p6775.pdf>
- Silva, L. T., & Mendes, J. F. G. (2006). Determinação do Índice de Qualidade do Ar numa cidade de média dimensão. *2º Congresso Luso-Brasileiro Para o Planeamento Urbano, Regional, Integrado e Sustentável*, 16. Retrieved from http://repositorium.sdum.uminho.pt/bitstream/1822/6333/3/n_27_pag_63-74.pdf
- Soares, J. M. da S. (2015). *Modelação do Impacto da Circulação Rodoviária da NI na Qualidade do Ar em Leiria*.
- SparkFun TSL2561. (2019). SparkFun Luminosity Sensor Breakout - TSL2561. Retrieved August 27, 2019, from <https://www.sparkfun.com/products/retired/12055>
- StatCounter Global Stats. (2019). Mobile Operating System Market Share Worldwide | StatCounter Global Stats. Retrieved August 27, 2019, from <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- Szulczyński, B., & Gębicki, J. (2017). Currently Commercially Available Chemical Sensors Employed for Detection of Volatile Organic Compounds in Outdoor and Indoor Air. *Environments*, 4(1), 21. <https://doi.org/10.3390/environments4010021>
- Taborda, J. P. (2012). O temporal de 3 a 6 de Dezembro de 1739 em Portugal - reconstituição a partir de fontes documentais descritivas. *Finisterra*, 41(82), 73–86. <https://doi.org/10.18055/finis1450>

- Texas Instruments ADS1115. (2019). Texas Instruments - ADS1115 I2C. Retrieved August 27, 2019, from <http://www.ti.com/product/ADS1115>
- Urban Aspect eTube. (2019). Smart Traffic Counter: eTube Traffic - Urban Aspect Inc. Retrieved August 27, 2019, from <https://urbanaspect.ca/product/etube-traffic/>
- Urban Aspect SDR Traffic. (2019). Smart Traffic Counter: SDR Traffic - Urban Aspect Inc. Retrieved August 27, 2019, from <https://urbanaspect.ca/product/sdr-traffic/>
- Veronesi, R. L. M. (2005). *RtrASSoc51 – Módulo de Comunicação I2C Reconfigurável – rI2C*. FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA” - CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” - UNIVEM. <https://doi.org/616.89-008.47:616->
- WiringPi. (2019). WiringPi. Retrieved August 27, 2019, from <http://wiringpi.com/>

REFERÊNCIAS

APÊNDICE I – CONFIGURAÇÕES DOS BLOCOS

Bloco da Câmara

Image size: 640 x 480

Pixel format: RGB

Sample time: 0,1

Bloco de ROI

Colunas: [1:640]

Linhas: [181:480]

Bloco de detecção de primeiro plano

Number of Gaussian modes: 3

Threshold to determine background model: 0,75

Number of initial training frames: 1500

Learning rate for parameter updates: 0.00005

Bloco de filtro de mediana

Neighborhood size: [3 3]

Padding options: Symmetric

Bloco de BLOB

Specify minimum blob area in pixels: 1500

Specify maximum blob area in pixels: 120000

Bloco de contagem e classificação de veículos

Nível 1 (separador entre motocicletas e ligeiros): 7000

Nível 2 (separador entre ligeiros e pesados): 25000

Número máximo de pilhas: 40

Tamanho máximo de cada pilha: 20000

Tempo de esquecimento: 4 segundos

Distância mínima percorrida: 150

Bloco de comunicação

Tempo cíclico de envio: 60 segundos

APÊNDICE I

ANEXO A – INFORMAÇÕES DOS SENSORES

Figura A-1: Acesso aos endereços de memória do BME280.

Table 18: Memory map

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
hum_lsb	0xFE				hum_lsb<7:0>					0x00
hum_msb	0xFD				hum_msb<7:0>					0x80
temp_xlsb	0xFC			temp_xlsb<7:4>		0	0	0	0	0x00
temp_lsb	0xFB				temp_lsb<7:0>					0x00
temp_msb	0xFA				temp_msb<7:0>					0x80
press_xlsb	0xF9			press_xlsb<7:4>		0	0	0	0	0x00
press_lsb	0xF8				press_lsb<7:0>					0x00
press_msb	0xF7				press_msb<7:0>					0x80
config	0xF5		t_sb[2:0]			filter[2:0]			spi3w_en[0]	0x00
ctrl_meas	0xF4		osrs_t[2:0]			osrs_p[2:0]		mode[1:0]		0x00
status	0xF3					measuring[0]			im_update[0]	0x00
ctrl_hum	0xF2							osrs_h[2:0]		0x00
calib26..calib41	0xE1...0xF0				calibration data					individual
reset	0xE0				reset[7:0]					0x00
id	0xD0				chip_id[7:0]					0x60
calib00..calib25	0x88...0xA1				calibration data					individual

Reserved registers	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
do not change	read only	read / write	read only	read only	read only	write only

Registers:

Type:

Figura A-2: Acesso aos endereços de calibração do BME280.

Table 16: Compensation parameter storage, naming and data type

Register Address	Register content	Data type
0x88 / 0x89	dig_T1 [7:0] / [15:8]	unsigned short
0x8A / 0x8B	dig_T2 [7:0] / [15:8]	signed short
0x8C / 0x8D	dig_T3 [7:0] / [15:8]	signed short
0x8E / 0x8F	dig_P1 [7:0] / [15:8]	unsigned short
0x90 / 0x91	dig_P2 [7:0] / [15:8]	signed short
0x92 / 0x93	dig_P3 [7:0] / [15:8]	signed short
0x94 / 0x95	dig_P4 [7:0] / [15:8]	signed short
0x96 / 0x97	dig_P5 [7:0] / [15:8]	signed short
0x98 / 0x99	dig_P6 [7:0] / [15:8]	signed short
0x9A / 0x9B	dig_P7 [7:0] / [15:8]	signed short
0x9C / 0x9D	dig_P8 [7:0] / [15:8]	signed short
0x9E / 0x9F	dig_P9 [7:0] / [15:8]	signed short
0xA1	dig_H1 [7:0]	unsigned char
0xE1 / 0xE2	dig_H2 [7:0] / [15:8]	signed short
0xE3	dig_H3 [7:0]	unsigned char
0xE4 / 0xE5[3:0]	dig_H4 [11:4] / [3:0]	signed short
0xE5[7:4] / 0xE6	dig_H5 [3:0] / [11:4]	signed short
0xE7	dig_H6	signed char

Figura A-3: Acesso aos endereços de memória do TSL2561.

Table 2. Register Address

ADDRESS	REGISTER NAME	REGISTER FUNCTION
--	COMMAND	Specifies register address
0h	CONTROL	Control of basic functions
1h	TIMING	Integration time/gain control
2h	THRESHLOWLOW	Low byte of low interrupt threshold
3h	THRESHLOWHIGH	High byte of low interrupt threshold
4h	THRESHHIGHLOW	Low byte of high interrupt threshold
5h	THRESHHIGHHIGH	High byte of high interrupt threshold
6h	INTERRUPT	Interrupt control
7h	--	Reserved
8h	CRC	Factory test — not a user register
9h	--	Reserved
Ah	ID	Part number/ Rev ID
Bh	--	Reserved
Ch	DATA0LOW	Low byte of ADC channel 0
Dh	DATA0HIGH	High byte of ADC channel 0
Eh	DATA1LOW	Low byte of ADC channel 1
Fh	DATA1HIGH	High byte of ADC channel 1

Figura A-4: Acesso aos endereços de memória do SGP30.

Measure Test

The command "Measure_test" which is included for integration and production line testing runs an on-chip self-test. In case of a successful self-test the sensor returns the fixed data pattern 0xD400 (with correct CRC).

Feature Set	0x0009				
	Command	Hex. Code	Parameter length, including CRC [bytes]	Response length, including CRC [bytes]	Measurement duration [ms]
					Typ. Max.
Init_air_quality	0x2003	-	-	-	1 10
Measure_air_quality	0x2008	-	-	6	40 50
Get_Baseline	0x2015	-	-	6	10 10
Set_Baseline	0x201e	6	-	-	10 10
Measure_test	0x2032	-	-	3	200 220
Get_feature_set_version	0x202f	-	-	3	1 2
Measure_signals	0x2050	-	-	6	180 200

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
 S I2C Address W ACK Command MSB ACK Command LSB ACK
 I2C write header 16-bit command

Table 9 Measurement commands.

Figura A-5: Acesso aos endereços de memória do MiCS-6814.

```

239  /*****
240  ** Function name:          readR0
241  ** Descriptions:         read R0 stored in slave MCU
242  *****/
243  int16_t MutichannelGasSensor::readR0(void)
244  {
245      int16_t rtnData = 0;
246
247      rtnData = readData(0x11);
248
249      if(rtnData > 0)
250          res0[0] = rtnData;
251      else
252          return rtnData;
253
254      rtnData = readData(0x12);
255      if(rtnData > 0)
256          res0[1] = rtnData;
257      else
258          return rtnData;
259
260      rtnData = readData(0x13);
261      if(rtnData > 0)
262          res0[2] = rtnData;
263      else
264          return rtnData;
265
266      return 1;
267  }
268
269  /*****
270  ** Function name:          readR
271  ** Descriptions:         read resistance value of each channel from slave MCU
272  *****/
273  int16_t MutichannelGasSensor::readR(void)
274  {
275      int16_t rtnData = 0;
276
277      rtnData = readData(0x01);
278      if(rtnData >= 0)
279          res[0] = rtnData;
280      else
281          return rtnData;
282
283      rtnData = readData(0x02);
284      if(rtnData >= 0)
285          res[1] = rtnData;
286      else
287          return rtnData;
288
289      rtnData = readData(0x03);
290      if(rtnData >= 0)
291          res[2] = rtnData;
292      else
293          return rtnData;
294
295      return 0;
296  }

```

Figura A-6: Acesso aos endereços de memória do MiCS-6814 (continuação).

```
500  /*****  
501  ** Function name:          powerOn  
502  ** Descriptions:         power on sensor heater  
503  *****/  
504  void MutichannelGasSensor::powerOn(void)  
505  {  
506      if(__version == 1)  
507          sendI2C(0x21);  
508  }
```

Figura A-7: Acesso aos endereços de memória da ADS1115.

Figure 36. Config Register

15	14	13	12	11	10	9	8
OS	MUX[2:0]			PGA[2:0]			MODE
R/W-1h	R/W-0h			R/W-2h			R/W-1h
7	6	5	4	3	2	1	0
DR[2:0]			COMP_MODE	COMP_POL	COMP_LAT	COMP_QUE[1:0]	
R/W-4h			R/W-0h	R/W-0h	R/W-0h	R/W-3h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 8. Config Register Field Descriptions

Bit	Field	Type	Reset	Description
15	OS	R/W	1h	<p>Operational status or single-shot conversion start This bit determines the operational status of the device. OS can only be written when in power-down state and has no effect when a conversion is ongoing.</p> <p>When writing: 0 : No effect 1 : Start a single conversion (when in power-down state)</p> <p>When reading: 0 : Device is currently performing a conversion 1 : Device is not currently performing a conversion</p>
14:12	MUX[2:0]	R/W	0h	<p>Input multiplexer configuration (ADS1115 only) These bits configure the input multiplexer. These bits serve no function on the ADS1113 and ADS1114.</p> <p>000 : AIN_p = AIN0 and AIN_N = AIN1 (default) 001 : AIN_p = AIN0 and AIN_N = AIN3 010 : AIN_p = AIN1 and AIN_N = AIN3 011 : AIN_p = AIN2 and AIN_N = AIN3 100 : AIN_p = AIN0 and AIN_N = GND 101 : AIN_p = AIN1 and AIN_N = GND 110 : AIN_p = AIN2 and AIN_N = GND 111 : AIN_p = AIN3 and AIN_N = GND</p>
11:9	PGA[2:0]	R/W	2h	<p>Programmable gain amplifier configuration These bits set the FSR of the programmable gain amplifier. These bits serve no function on the ADS1113.</p> <p>000 : FSR = ±6.144 V⁽¹⁾ 001 : FSR = ±4.096 V⁽¹⁾ 010 : FSR = ±2.048 V (default) 011 : FSR = ±1.024 V 100 : FSR = ±0.512 V 101 : FSR = ±0.256 V 110 : FSR = ±0.256 V 111 : FSR = ±0.256 V</p>
8	MODE	R/W	1h	<p>Device operating mode This bit controls the operating mode.</p> <p>0 : Continuous-conversion mode 1 : Single-shot mode or power-down state (default)</p>
7:5	DR[2:0]	R/W	4h	<p>Data rate These bits control the data rate setting.</p> <p>000 : 8 SPS 001 : 16 SPS 010 : 32 SPS 011 : 64 SPS 100 : 128 SPS (default) 101 : 250 SPS 110 : 475 SPS 111 : 860 SPS</p>

Figura A-8: Acesso aos endereços de memória da ADS1115 (continuação).

Table 8. Config Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	COMP_MODE	R/W	0h	<p>Comparator mode (ADS1114 and ADS1115 only) This bit configures the comparator operating mode. This bit serves no function on the ADS1113.</p> <p>0 : Traditional comparator (default) 1 : Window comparator</p>
3	COMP_POL	R/W	0h	<p>Comparator polarity (ADS1114 and ADS1115 only) This bit controls the polarity of the ALERT/RDY pin. This bit serves no function on the ADS1113.</p> <p>0 : Active low (default) 1 : Active high</p>
2	COMP_LAT	R/W	0h	<p>Latching comparator (ADS1114 and ADS1115 only) This bit controls whether the ALERT/RDY pin latches after being asserted or clears after conversions are within the margin of the upper and lower threshold values. This bit serves no function on the ADS1113.</p> <p>0 : Nonlatching comparator . The ALERT/RDY pin does not latch when asserted (default). 1 : Latching comparator. The asserted ALERT/RDY pin remains latched until conversion data are read by the master or an appropriate SMBus alert response is sent by the master. The device responds with its address, and it is the lowest address currently asserting the ALERT/RDY bus line.</p>
1:0	COMP_QUE[1:0]	R/W	3h	<p>Comparator queue and disable (ADS1114 and ADS1115 only) These bits perform two functions. When set to 11, the comparator is disabled and the ALERT/RDY pin is set to a high-impedance state. When set to any other value, the ALERT/RDY pin and the comparator function are enabled, and the set value determines the number of successive conversions exceeding the upper or lower threshold required before asserting the ALERT/RDY pin. These bits serve no function on the ADS1113.</p> <p>00 : Assert after one conversion 01 : Assert after two conversions 10 : Assert after four conversions 11 : Disable comparator and set ALERT/RDY pin to high-impedance (default)</p>

