



Sistema de Monitorização em Tempo Real da Pressão Plantar e Dorsal do Pé

SARA PATRÍCIA DE MEDEIROS LACERDA LOPES

Outubro de 2021

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistema de Monitorização em Tempo Real da Pressão Plantar e Dorsal do Pé

Sara Patrícia de Medeiros Lacerda Lopes

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Telecomunicações



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Outubro, 2021

Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Telecomunicações.

Candidato: Sara Patrícia de Medeiros Lacerda Lopes, Nº 1190148,
1190148@isep.ipp.pt

Orientação Científica: Lino Figueiredo, lbf@isep.ipp.pt

Coorientação Científica: Arcelina Marques, mmr@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Outubro, 2021

Agradecimentos

Desejo exprimir os meus agradecimentos a todos aqueles que, de alguma forma, permitiram que esta tese se concretizasse.

Em primeiro lugar, quero agradecer ao Instituto Superior de Engenharia do Porto (ISEP), por permitir a pessoas, tal como eu, graduadas noutras áreas que não a Engenharia, ingressar num Mestrado nesta instituição.

Em seguida, agradeço a todos os professores que fui tendo ao longo destes dois anos, por toda a ajuda e incentivo que me deram. Gostaria de agradecer especialmente aos meus orientadores, Doutor Lino Figueiredo e Doutora Arcelina Marques, por toda a disponibilidade demonstrada, pelos inúmeros debates e trocas de ideias durante a realização deste trabalho.

Agradeço também aos meus colegas de projeto, Carolina Silva e Pedro Martins, pelo trabalho em grupo, pelas partilha de ideias e, por ao fim destes meses os poder chamar de amigos.

Como não podia deixar de ser, a todos os meus colegas do ramo de Telecomunicações, por todos os momentos de convívio, incentivo e estudo em conjunto. Destes em especial quero agradecer ao Alexandre Amorim pelos momentos de estudo, as múltiplas conversas e desabafos; ao Filipe Chen, pelo sentido de humor e inúmeras partidas ao longo das aulas; ao Lucas Tucker pela disponibilidade em debater ideias e, à Sandra Silva, minha parceira de trabalho de grupo na maioria das cadeiras. Sem o acolhimento de todos a minha passagem por este mestrado teria sido, com toda a certeza, menos profícua.

Gostaria também de agradecer ao Luís Cruz, pela disponibilidade imediata em emprestar diverso material eletrónico para que mais rapidamente pudesse passar para a implementação física do sistema.

Quero deixar um agradecimento especial ao meu marido, João Cacheira, sem o seu constante apoio, incentivo e disponibilidade para me ajudar em todas as tarefas de casa, a realização desta tese e a frequência deste mestrado não teria sido, de todo possível. Muito obrigado por teres sido e continuares a ser o meu pilar e o meu porto seguro.

Por fim, quero agradecer à minha filha, Joana Cacheira, por ter sido um anjinho, por me ter deixado estudar quando era necessário, por ter compreendido todos os momentos conjuntos que lhe roubei pelo fato de ter ousado sonhar em tirar um novo curso. A ela dedico este trabalho!

Resumo

A diabetes é a doença crónica mais frequente no mundo ocidental sendo já considerada uma epidemia a nível mundial. Estima-se que afete aproximadamente 415 milhões de pessoas em todo o mundo, prevendo-se que, daqui a 20 anos, esse número possa ascender aos 642 milhões. A diabetes está associada a inúmeras complicações clínicas ao nível dos olhos, rins, sistema vascular, sistema nervoso e coração.

Uma das complicações mais comuns e graves da diabetes é a ulceração do pé, sendo esta de complexo tratamento bem como com custos elevados associados, quer para o paciente quer para o sistema de saúde em geral. Sabe-se que cerca de 50% das úlceras em diabéticos infectam e cerca de 20% das infeções provocam a amputação de partes ou de todo o pé.

Um dos fatores que leva à ulceração são as hiperpressões frequentes e repetitivas nas mesmas áreas do pé. Assim, acredita-se que uma melhor redistribuição das pressões plantares e constante monitorização das pressões em zonas tidas como de risco ao desenvolvimentos de úlceras, poderá revelar-se um factor preponderante na prevenção da formação destas e, conseqüentemente, poderá diminuir o risco de desenvolvimento lesões no pé.

Torna-se por isso de extrema importância o desenvolvimento de sistemas de monitorização de pressões no pé, nomeadamente ao nível da zona plantar e dorsal, por serem estas as identificadas como zonas de risco à formação de úlceras.

Esta dissertação apresenta o desenvolvimento de um protótipo que permite a monitorização das pressões na zona plantar e na zona dorsal do pé, adaptado a pessoas diabéticas. Deste modo, parte do trabalho consistiu no estudo de sensores que pudessem ser acoplados na palmilha de um sapato bem como na zona da gáspea (parte superior do sapato) tendo estes de ser devidamente adaptado à utilização por diabéticos. Este facto pressupõe que todo o sistema a ser implementado tenha de obedecer a padrões que continuem a permitir uma marcha confortável do paciente, o mais natural possível e sem que o sistema seja considerado um corpo estranho no interior do sapato, e como tal, uma fonte na formação de úlceras. Outros dos focos deste trabalho foi o estudo da melhor localização dos sensores e de todo o processo que leva a uma correta aquisição de dados. Por fim, foi validado o protótipo

desenvolvido através de testes e consequente análise do resultados obtidos.

Palavras-Chave: Pé diabético, pressões plantares, pressão zona dorsal do pé, sistema ne monitorização sensores FSR

Abstract

Diabetes is the most frequent chronic disease in the western world and is already considered a worldwide epidemic. The estimation is that this disease affects approximately 415 million people, and there are predictions that this number may rise to 642 million in the next 20 years.

Diabetes is associated with numerous clinical complications in the eyes, kidneys, heart, vascular and nervous system. One of the most common and significant complications of diabetes is foot ulceration, which is difficult to treat and has huge associated costs, both for the patient and the health system. About 50% of ulcers in diabetics become infected, 20% of those cause amputation of parts or all the foot.

One of the factors that lead to ulceration is frequent and repetitive high pressure in the same areas of the foot. So a better redistribution of plantar pressures and constant monitoring of pressures in high-risk ulcers development areas is a crucial factor to prevent their formation and, consequently, to reduce foot injuries development risk. Thus, with ulceration prevention, it is estimated that amputation in more than 50% of patients, can be avoided.

It is, therefore, crucial to develop systems for monitoring foot pressure, particularly in the plantar and dorsal areas, as these are known to be the risky points for ulcers formation.

This dissertation presents the development of a prototype that allows the monitoring of pressures in the plantar and dorsal area of the foot, directed to diabetic people. Part of the work consists of sensor research to be attached to the insole as well to the upper part of a medical diabetic shoe. The goal was that the entire system must allow a comfortable gait for the patient, as natural as possible and without the feel of a foreign body inside the shoe which is usually the source of ulcers. Another focus of this work was studying the best location of sensors and the entire process that leads to correct data acquisition. Finally, prototype validation was done, through tests, analysis and results obtained.

Keywords: Diabetic foot, plantar pressures, dorsal foot pressure, monitoring system, FSR sensors

Índice

Lista de Figuras	ix
Lista de Tabelas	xiii
Lista de Acrónimos	xv
1 Introdução	1
1.1 Contextualização	1
1.2 Objectivos	5
1.2.1 Requisitos	6
1.3 Plano	6
1.4 Estrutura da dissertação	7
2 Estado de Arte	9
2.1 Prospecção de Wearables	9
2.1.1 Sistemas comerciais com sensores	11
2.1.2 Sistemas experimentais com sensores	16
2.1.3 Localização dos sensores	17
2.2 Revisão tecnológica sobre sensores de força	20
2.2.1 Sensores Capacitivos	21
2.2.2 Sensores Resistivos	22
2.2.3 Sensores piezoresistivos e piezoelétricos	23
2.2.4 Sensores Têxteis	24
2.3 Conclusão	24
3 Arquitectura do Sistema	27
3.1 Arquitectura geral do sistema	27
3.2 MCU	28
3.2.1 BLE	29
3.3 Sensores de força	32
3.3.1 Estudo da gama de medição dos sensores	33
3.4 IMU	36
3.5 Conclusão	37

4	Desenvolvimento do Hardware	39
4.1	Hardware implementado	39
4.2	Conclusão	44
5	Desenvolvimento do Software	47
5.1	Firmware	47
5.2	Software	55
5.2.1	App Visual Basic	55
5.2.2	App telemóvel	59
5.3	Conclusão	64
6	Análise dos Resultados	65
6.1	Testes e Calibração de Sensores	65
6.1.1	Sensores FSR	66
6.1.2	Sensores Piezofabric	68
	Palmilha Azul	68
	Palmilha Verde	69
6.1.3	Sensores PiezoInk	71
6.2	Testes com palmilha e Gáspea sensorizadas	74
6.3	Conclusão	78
7	Conclusão	79
7.1	Trabalho Futuro	81
	Referências	83
	Anexo A Código Firmware	93
A.1	Inicialização de variáveis	93
A.2	Inicialização do IMU	94
A.3	Inicialização BLE	96
A.4	Handler interrupção	97
A.5	Função Setup	97
A.6	Função loop	98
A.7	Leitura do IMU	100
A.8	Leitura dos sensores plantares	100
A.9	Leitura dos sensores da gáspea	101
A.10	Envio de dados por BLE	102
	Anexo B Código aplicação Visual Basic	105
B.1	Inicialização do programa e de variáveis	105
B.2	Definição de botões que permitem mostrar e ocultar gráficos	106
B.3	Configuração dos gráficos	110

B.4	Menu Connection	115
B.5	Menu Data Record	117
B.5.1	Preenchimento da tabela	121
B.5.2	Guardar dados no ficheiro Excel	122
B.6	Recepção e tratamento dos dados	123
Anexo C Blocos MiT App Inventor		135
C.1	Inicialização variáveis	135
C.2	Seleção e visualização de dispositivos BLE	136
C.3	Escuta por bytes recebidos por BLE	136
C.4	Definições de botões e de procedimento de descodificação	137
C.5	Tratamento dos bytes recebidos por BLE	138
C.5.1	UUID Plantar	138
C.5.2	UUID Gaspea	139
C.5.3	UUID IMU	139
C.5.4	Cálculo e apresentação de valores dos sensores no ecrã	140

Lista de Figuras

1.1	Esquema ilustrativo da formação de úlceras no pé [6]	2
1.2	Fatores de risco associados à reincidência da úlcera do pé [6]	3
1.3	Exemplo de constituição de sapato para diabético [15]	4
1.4	Sapato para diabético da marca (a) Ortomedical; (b) Dr Comfort; (c) OrthoFeet	4
2.1	Exemplo de <i>smart socks</i> da marca <i>Sensoria</i> [26]	10
2.2	Exemplo da palmilha F-Scan da marca <i>Tekscan</i> [27]	10
2.3	Exemplo de <i>smart shoe</i> [28]	11
2.4	Sistema de palmilha Pedar [33]	12
2.5	Software F-Scan [27]	12
2.6	Palmilha Orpyx [35]	13
2.7	Palmilha Salted [38]	14
2.8	Exemplo de palmilha sensora [39]	15
2.9	Palmilha Lechal [40]	15
2.10	Software da meia Sensoria [26]	16
2.11	Áreas anatómicas do pé [65]	19
2.12	Zonas de maior risco de úlceração do pé diabético [5]	19
2.13	(a)Localização de cinco sensores na zona plantar [66]; (b) Localização de seis sensores na zona plantar [47]; (c) Localização de oito sensores na zona plantar [62]	20
2.14	Esquema ilustrativo do mecanismo representativo de funcionamento de um dispositivo capacitivo [68]	22
2.15	Camadas de constituição de um sensor capacitivo e respetiva aplicação numa palmilha [64]	22
2.16	Camadas de constituição de um sensor Force Sensing Resistors (FSR)	23
2.17	Esquema ilustrativo do mecanismo representativo de funcionamento de um dispositivo Piezoelétrico e Piezoresistivo [68]	23
2.18	Meia tricotada com sensores têxteis [51]	24
3.1	Arquitectura do sistema	28
3.2	Pilha protocolar Bluetooth Low Energy (BLE) [75]	30
3.3	Serviços e características BLE [77]	31

3.4	Zonas de medição da área do pé [82]	34
4.1	Localização dos sensores na região plantar e na gáspea	40
4.2	(a) Matriz 3X3 de alimentação e ligação ao Analog-to-Digital Converter (ADC) para a palmilha; (b) Matriz 2X2 de alimentação e ligação ao ADC para a gáspea	41
4.3	Divisor de tensão	41
4.4	Pinout da ESP33 DEVKitC [89]	42
4.5	Esquema elétrico dos componentes do sistema	43
4.6	Protótipo do sistema de sensorização	44
4.7	Pormenor das ligações efectuadas entre sensores e placa de desenvolvimento	44
5.1	Fluxograma da aquisição e envio de dados aplicado no dispositivo	48
5.2	Fluxograma da interrupção	49
5.3	Fluxograma da função de leitura dos sensores plantares	51
5.4	Fluxograma da função de leitura sensores gáspea	52
5.5	Fluxograma da função de envio de dados BLE	53
5.6	Características BLE com trama de dados	54
5.7	Trama de dados enviada para a porta serie	55
5.8	Aspecto gráfico da Aplicação Visual Basic	56
5.9	Fluxograma geral da App em Visual Basic	57
5.10	Janela de seleção do dispositivo BLE	60
5.11	Vista da aplicação para <i>Smartphone</i> (a) inicial; (b) Gáspea e IMU	61
5.12	Fluxograma da App para <i>smartphone</i>	62
6.1	Gráfico Condutância em função da Força para R= 1200 Ω para FSR convencionais	66
6.2	Gráfico Condutância em função da Força para R= 4300 Ω para FSR convencionais	67
6.3	Gráfico Condutância em função da Força para R= 10000 Ω para FSR convencionais	67
6.4	Palmilha Azul SensingTex	68
6.5	Gráfico Condutância em função da Força obtido via multímetro para a palmilha azul	68
6.6	Gráfico Condutância em função da Força para R= 1200 Ω para a palmilha azul	69
6.7	Palmilha verde SensingTex	70
6.8	Gráfico Condutância em função da Força obtido via multímetro para a palmilha verde	70

6.9	Gráfico Condutância em função da Força para $R= 1200 \Omega$ para a palmilha verde	71
6.10	Matriz branca SensingTex	72
6.11	Gráfico Condutância em função da Força obtido via Multímetro para a matriz branca	72
6.12	Gráfico Condutância em função da Força para $R= 1200 \Omega$ para a matriz branca	73
6.13	Gráfico Condutância em função da Força para $R= 4300 \Omega$ para a matriz branca	73
6.14	Gráfico Condutância em função da Força para $R= 10000 \Omega$ para a matriz branca	74
6.15	Aplique da gáspea com os sensores aplicados	75
6.16	Sapato para diabéticos com palmilha de sensores e aplique na gáspea	75
6.17	Execução dos testes com os sensores	76
6.18	Gráfico de forças plantares	76
6.19	Gráfico de forças na gáspea	77
6.20	Gráfico da relação da soma das forças plantares e da soma das forças na gáspea	77
B.1	Exemplo do funcionamento do algoritmo de desmultiplexagem dos dados enviados por porta série	128

Lista de Tabelas

1.1	Programação do trabalho	6
2.1	Características de sistemas desenvolvidos para medição de forças/- pressões no pé	18
3.1	Comparação de características de placas de desenvolvimento	29
3.2	Comparação entre Bluetooth clássico e BLE [74]	30
3.3	Características de sensores resistivos	32
3.4	Medidas da área média da sola do pé, por zona [82]	34
3.5	Resumo das forças e pressões estimadas que se prevêem aplicadas em cada sensor	36
3.6	Características de Inertial Measurement Unit (IMU)	36
6.1	Variação do valor detetado no ADC e do valor da tensão com a resis- tência	67
6.2	Variação do valor detetado no ADC e do valor da tensão com a resis- tência para a matriz branca	74

Lista de Acrónimos

ADC	Analog-to-Digital Converter
AIoT	Artificial Intelligence of Things
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
DAC	Digital-to-Analog Converter
DLE	Data Length Extension
DMP	Digital Motion Processor
EVA	Ethylene-Vinyl Acetate
FSR	Force Sensing Resistors
GATT	Generic Attribute Profile
GND	Ground
GPIO	General-purpose input/output
I²C	Inter-Integrated Circuit
I²S	Integrated Interchip Sound
IMU	Inertial Measurement Unit
ISEP	Instituto Superior de Engenharia do Porto
ISR	Integrated Switching Regulator
LSB	Least Significant Bit
MCU	Microcontroller Unit
MSB	Most Significant Bit
PET	Polyethylene terephthalate
RF	Radio Frequency

ROM	Read-only memory
SCL	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TPU	Thermo-plastic polyurethane
UART	Universal Asynchronous Receiver/Transmitter
UUID	Universally Unique Identifier
VCC	Common Collector Voltage
Wi-Fi	Wireless Fidelity
WPF	Windows Presentation Foundation

Capítulo 1

Introdução

Neste capítulo pretende-se realizar uma breve introdução ao tema desta dissertação, definir os principais objetivos e requisitos envolvidos, apresentar a planificação para a execução da mesma bem com a estrutura e o conteúdo de cada capítulo

1.1 Contextualização

A Diabetes Mellitus inclui um grupo de patologias metabólicas caracterizadas pela hiperglicémia como consequência da deficiente produção de insulina, pela sua atuação ineficaz ou pela combinação de ambos [1].

A insulina, hormona produzida pelo pâncreas, permite que a glicose passe da corrente sanguínea para as células que a utilizam como fonte de energia. Num organismo saudável, as células pancreáticas produzem insulina em quantidades suficientes face aos níveis de glicose no sangue. Esta doença pode ser dividida, entre outras, em duas subclasses:

- Diabetes tipo 1 - ocorre quando o pâncreas não produz insulina suficiente e o sistema imunitário destrói as células beta (produtoras de insulina) [1].
- Diabetes tipo 2 - surge quando o pâncreas não é capaz de produzir insulina suficiente ou quando esta não é utilizada de forma eficaz pelo nosso organismo [1].

A hiperglicémia crónica está associada a múltiplas complicações clínicas ao nível dos olhos, rins, sistema vascular, sistema nervoso e coração.

O desenvolvimento de neuropatia periférica, doença vascular periférica, perda de sensibilidade cutânea, diminuta mobilidade articular e lesões por hiperpressões plantares, leva a que pessoas com diabetes tenham um risco acrescido de desenvolver problemas nos membros inferiores, nomeadamente ulceração do pé [2–5].

A ulceração do pé em diabéticos é uma patologia de ocorrência extremamente comum, de complexo tratamento bem como com custos elevados associados, quer para o paciente quer para o sistema de saúde em geral [6]. Segundo alguns estudos, mais de metade das úlceras em diabéticos infetam e cerca de 20% das infecções severas irão provocar algum tipo de amputação, seja de apenas alguns dedos ou mesmo de todo o pé [6, 7].



Figura 1.1: Esquema ilustrativo da formação de úlceras no pé [6]

Na Figura 1.1 pode ser percebido um resumo do modo de formação de

úlceras do pé, provocadas pela diabetes. A ulceração do pé diabético, é normalmente provocada pelo *stress* efetuado numa determinada área sujeita a tensões verticais ou a cisalhamento [8]. A ulceração do pé ocorre principalmente na região plantar dianteira e, com menor prevalência, na região dorsal do pé [9, 10]. Também a zona do retropé, nomeadamente a zona do calcanhar, pode ser afetada por úlceras [11]. Sabe-se que existe uma relação causa-efeito entre a existência de hiperpressões nas zonas afetadas e a formação de úlceras nas mesmas zonas [12].

Por forma a prevenir o aparecimento de lesões, é necessário que haja um bom conhecimento dos fatores que poderão levar à ocorrência da ulceração. Sendo a ulceração do pé diabético uma patologia recorrente, estudos em pacientes com reincidência desta doença mostram que a abundância de calosidades, bolhas ou hemorragia estão entre as mais fortes causas da recorrência de úlcera, tal como indicado na Figura 1.2. Ainda pela análise da Figura 1.2 verifica-se que fatores biomecânicos como a aderência do calçado e a pressão plantar são também fatores importantes na recorrência de úlceras na superfície plantar do pé [6].

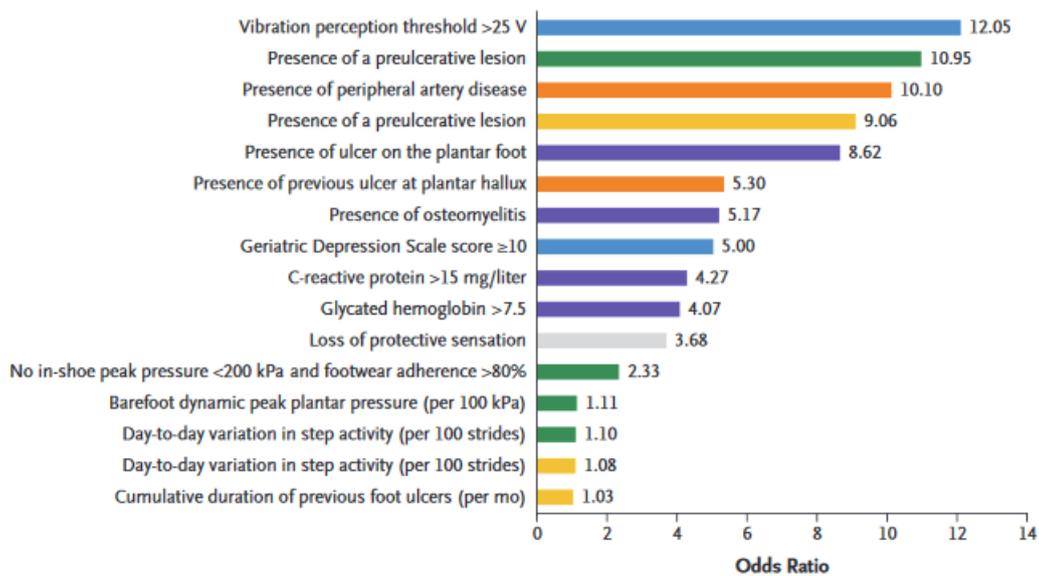


Figura 1.2: Fatores de risco associados à reincidência da úlcera do pé [6]

Contudo, esses fatores biomecânicos são corrigíveis. O tratamento e/ou prevenção pode passar pela utilização de calçado adequado que tenha um efeito demonstrado no alívio da pressão plantar [13, 14]. A utilização de calçado especialmente concebido para diabéticos, poderá promover uma melhor circulação sanguínea e reduzir o risco de lesões.

Ferimentos e alterações na forma e estrutura do pé podem tornar desconfortáveis os sapatos convencionais não adaptados. O calçado para diabéticos deve conter

características específicas e ser projetado bem como construído de forma a aliviar algumas complicações nos pés, a que os diabéticos estão propensos. Os sapatos para diabéticos incluem as seguintes características e qualidades especiais [15]:

- Interior protetor feito de material macio sem costura ou com costura suavizada para não causar irritação.
- Parte superior não vinculativa que garanta que não haja pressão no pé na parte superior do pé.
- Design que permita acomodar inserções ortóticas.
- Uma biqueira profunda que permita espaço extra para os movimentos dos dedos e evite a pressão sobre os mesmos.
- Sola leve e almofadada que facilite a mobilidade, a estabilidade e reduza os impactos nos pés.

A Figura 1.3 mostra um exemplo de requisitos de constituição do sapato para diabéticos.

Hoje em dia são diversas as marcas que comercializam este tipo de calçado, como a Ortomedical (Figura 1.4a), Dr. Comfort (Figura 1.4b), OrthoFeet (Figura 1.4c), Calzamedi, Propét, entre outras.



Figura 1.3: Exemplo de constituição de sapato para diabético [15]



(a)

(b)

(c)

Figura 1.4: Sapato para diabético da marca (a) Ortomedical; (b) Dr Comfort; (c) OrthoFeet

Também as novas tecnologias poderão ter um papel crucial na identificação precoce do risco de ulceração. É importante diagnosticar problemas nos pés numa fase ainda inicial por forma a prevenir o aparecimento de lesões e, em geral, de modo a potenciar o bem estar. Uma das abordagens, que se encontra a ser mais amplamente utilizada de modo a aferir a saúde do pé, é determinar as características de pressão plantar.

Cada vez mais, a utilização de sistemas de medição incorporados no interior do calçado, para monitorização de pressões plantares nos pés, tem vindo a ganhar terreno estando a ser adoptado quer em aplicações na área da saúde quer em aplicações do foro lúdico/desportivo [16].

Uma área que tem atraído especial atenção é a utilização da monitorização de pressões plantares na prevenção da ulceração do pé diabético.

Sabe-se que há zonas do pé onde é maior o risco de ulceração, e que estas se situam maioritariamente na zona plantar. Contudo, a zona dorsal do pé, nomeadamente na zona das falanges, é também uma zona onde se sabe existir um risco acrescido de ulceração [5].

A tensão repetitiva num determinado ponto pode ser detetada com um sistema de medição de pressão colocada no interior do calçado [17]. Tais medições podem também ser utilizadas para melhorar as propriedades de redução de tensões, inclusivamente em calçado terapêutico e diminuir o risco de desenvolvimento de calo e, conseqüentemente, a formação de úlcera [13].

Relativamente à zona dorsal há estudos, onde se pretendia avaliar o conforto do sapato, que identificam valores de pressão entre os 10 kPa e os 70 kPa, dependendo estes da zona e da localização dos sensores [18–20]. Para a zona plantar, existe uma disparidade nos valores mencionados em artigos variando estes entre os 100 kPa e os 950 kPa dependendo estes da localização dos sensores [19, 21, 22]. Porém está também referido que reduções de pressões plantares para valores menor que 200 kPa revelam-se favoráveis à prevenção da reincidência na formação de úlceras, quando medidas em sensores com uma área de 2 cm^2 [23].

Sabendo-se que cerca de 34% dos pacientes diabéticos desenvolvam úlceras no pé ao longo da sua vida [24] e que esta é uma das principais causas de amputação do pé ou de partes do pé de diabéticos [6, 7], é crucial que se desenvolvam sistemas de monitorização fiáveis que permitam a prevenção da ulceração do pé dos diabéticos.

1.2 Objectivos

Face ao exposto na secção anterior, pretende-se desenvolver um sistema para monitorização da pressão plantar e dorsal do pé durante a marcha humana. Este sistema deverá ser incorporado num sapato concebido para diabéticos, garantindo que obedece aos requisitos funcionais deste tipo de sapato. Assim os sensores e os cabos

1.4 Estrutura da dissertação

Esta dissertação é composta por sete capítulos organizados em secções e subsecções.

Após o presente capítulo de introdução, no capítulo 2 apresenta-se um levantamento do estado de arte relativamente a sistemas de sensorização de pressões aplicados ao pé, seja a nível comercial, seja a nível de investigação. Faz-se, ainda, uma breve abordagem à tecnologia dos sensores mais utilizados e à localização mais frequente destes, na zona plantar, nos sistemas já implementados.

No capítulo 3 apresenta-se a arquitetura geral do sistema implementado, tentando-se justificar algumas das escolhas realizadas quanto aos componentes utilizados.

Relativamente ao capítulo 4, este contém a descrição, a nível do hardware, do sistema implementado

Já no capítulo 5, encontra-se expresso o desenvolvimento efetuado a nível do software. Este capítulo contém três secções que convém realçar, a descrição do *firmware* e uma breve abordagem a duas aplicações criadas por forma a realizar testes e obter resultados experimentais.

No capítulo 6 são analisados alguns dados obtidos nos ensaios.

No capítulo 7 apresenta-se as principais conclusões, algumas das restrições do projeto e ideias para a sua futura evolução.

Capítulo 2

Estado de Arte

A monitorização de pacientes diabéticos, no que se refere às hiperpressões plantares, tem sido objecto de estudo e de desenvolvimento recentes. No estado de arte desta dissertação, pretende-se realizar a introdução de conceitos-chave bem como a revisão da literatura, no que concerne a sistemas sensorizados, dando especial relevo aos que monitorizam pressões plantares e na gáspea para aplicação em pacientes diabéticos.

2.1 Prospecção de Wearables

Wearable technology e *wearable devices* são termos que pretendem descrever dispositivos elétricos aplicados em roupas e outros acessórios que podem ser utilizados confortavelmente no corpo. Nos últimos anos, estes dispositivos evoluíram de tecnologias volumosas, grandes e desconfortáveis para itens cada vez mais modernos, leves, de menores dimensões e portáteis [25]. Além disso, o mercado destes dispositivos está em crescimento existindo um cada vez maior número destes para as mais diversas aplicações.

Nesta secção vai-se realizar uma pequena revisão tecnológica a alguns *wearables*, sistemas comerciais que podem ser utilizados em calçado. Também se irá efetuar uma análise aos sistemas experimentais de deteção de pressão, aplicados a pé diabético referenciados em artigos científicos e finalmente, far-se-á uma breve abordagem aos vários tipos de sensores de força existentes no mercado.

Assim, importa salientar a existência de diferentes tipos de *wearable* aplicáveis a medições a partir do interior do calçado: as *smart socks*, o sistema de palmilha

e as sapatilhas integradas. As *smart socks* são sistemas baseados em têxteis com sensores integrados, tal como o exemplo indicado na Figura 2.1. Estas integram um ou mais tipos de sensores, módulos de aquisição e transmissão de dados, juntamente com a implementação ou adoção de protocolos de comunicação de dados, de modo a garantir a comunicação entre a meia sensora e outros dispositivos capazes de processar os sinais por ela gerados [16].



Figura 2.1: Exemplo de *smart socks* da marca *Sensoria* [26]

O sistema de palmilha, Figura 2.2, possui funcionalidades semelhantes às *smart socks* estando contudo os sensores alojados numa palmilha, que poderá ser colocada no interior de qualquer calçado. Existem contudo, outros sistemas de palmilha que, para além de conterem sensores contêm actuadores como motores de vibração.

As sapatilhas integradas, Figura 2.3 vêm de fábrica com todos os sensores montados no interior da estrutura do sapato.

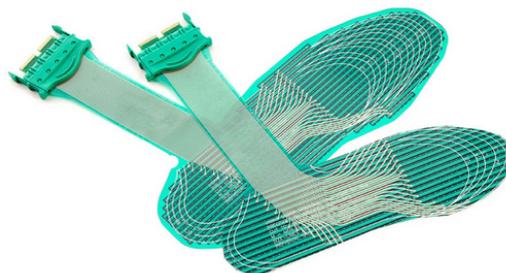


Figura 2.2: Exemplo da palmilha F-Scan da marca *Tekscan* [27]

No que concerne a efetuar medições biomecânicas, todos os dispositivos devem ser otimizados para a sua aplicação específica por forma a assegurar medições fiáveis. Qualquer medição, em tempo real, relativamente à marcha humana requer que os sensores utilizados possuam alguma mobilidade, permitam a sua colocação no



Figura 2.3: Exemplo de *smart shoe* [28]

interior do sapato e possam eficazmente realizar medições na zona pretendida [29]. Para tal, os sistemas montados devem cumprir alguns requisitos como:

- Utilização de sensores finos, leves e flexíveis de modo a permitir a sua colocação e acomodação eficaz ao interior do sapato, não alterando significativamente a forma da marcha do utilizador [30–32].
- Cablagem limitada e fina para garantir uma marcha confortável, segura e natural [29, 31].
- Baixo custo para que seja acessível a uma utilização generalizada [29].
- Baixo consumo de energia de forma que a energia de uma bateria pequena seja suficiente para dar autonomia ao sistema e lhe permita recolher os dados pretendidos [29].

2.1.1 Sistemas comerciais com sensores

Após pesquisa na internet, foram encontrados vários sistemas de monitorização baseados em sensores a serem correntemente comercializados. Dos encontrados destacam-se: Pedar Insole, o F-Scan, o Orpyx SI, o Salted, o Insole, a Lechal e Sensoria Socks.

Pedar

O Pedar Insole é um sistema de palmilha para medição de pressão plantar e é um dos sistemas com maior aceitação no mercado [16]. Possui 99 sensores distribuídos pela palmilha, permite a medição de 20000 sensores/segundo e é adaptável a qualquer tamanho do calçado. O Pedar Insole pode ser ligado a um computador via USB, funciona também via Bluetooth e possui um cartão de memória SD que permite a recolha de dados e posterior download dos mesmos [33].

Já o Pedar Pad trata-se de um sistema, com 56 sensores, que possibilita monitorizar pressões nas áreas dorsal, medial ou lateral do pé. Este dispositivo é um dos únicos do mercado que permite a medição de pressões na zona dorsal do pé.

Ambos os sistemas funcionam em conjunto com um software de aquisição de dados que possibilita uma rápida aquisição e tratamento dos dados recolhidos. O sistema Pedar Insole, o Pedar Pad, o módulo de comunicação destes e o software utilizado podem ser visualizados na Figura 2.4 [33].

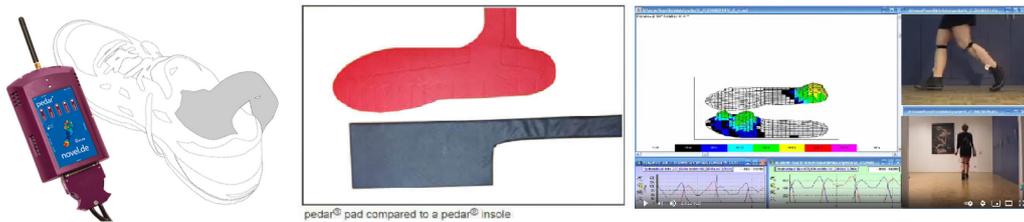
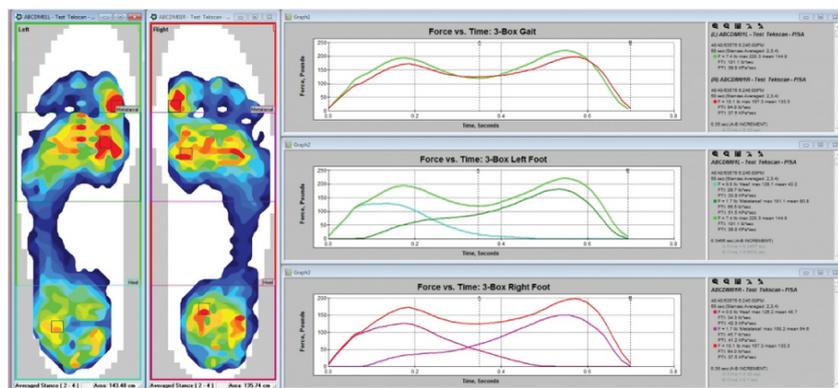


Figura 2.4: Sistema de palmilha Pedar [33]

F-Scan

Outro sistema de medição plantar são as palmilhas F-Scan da Tekscan, Figura 2.2. Tratam-se das palmilhas mais utilizadas em pesquisas e em aplicações clínicas. Possuem cerca de 960 sensores distribuídos ao longo da palmilha. São palmilhas extremamente finas, com apenas 0,15 mm de espessura e o módulo de electrónica para aquisição dos dados pode ser acoplado quer no tornozelo quer na cintura do utilizador [34].

O software F-Scan, Figura 2.5, permite o tratamento dos dados adquiridos pelo sistema de palmilhas e fornece informações que permitem a análise de marcha do utilizador, a visualização das curvas de pressão em tempo real, exportar relatórios de picos de pressão plantar, entre outras [27].



F-Scan 3-Box Analysis segments the foot into three key regions.

Figura 2.5: Software F-Scan [27]

Orpyx SI

As palmilhas Orpyx SI, Figura 2.6, foram criadas exclusivamente para diabéticos. A marca assume-se como a única a realizar múltiplas medições de diversos parâmetros fisiológicos de relevo para a prevenção, e consequente diminuição, do risco de ulceração do pé em diabéticos [35]. Possui oito sensores de pressão, sensor de temperatura e comunica com a App via *Bluetooth* [34]. Permitem a monitorização contínua da pressão plantar e notificam o paciente de forma visual, com um *feedback* vibratório e áudio caso a pressão plantar exceda um limite predefinido [36]. Esta tecnologia pode ser utilizada para educar pacientes a evitar condições que levem ao aumento da pressão plantar [37].

As palmilhas são produzidas por encomenda e personalizadas para cada doente. O seu design permite monitorizar a pressão plantar, a temperatura e ainda registar os passos dados ao longo do dia. Fornece alertas audiovisuais, para além de instruções, quando deteta elevados níveis de pressão ou aumento anormal da temperatura plantar [35].

Os dados recolhidos pelos múltiplos sensores podem ser partilhados com os médicos, através de aplicações próprias, podendo estes ajustar o plano de tratamento prescrito para o seu paciente [35].

Cada conjunto de palmilhas é fornecida com um carregador magnético bem como cabos USB. Já se encontra aprovado nos Estados Unidos da América pela FDA (*Food and Drug Administration*) [35].



Figura 2.6: Palmilha Orpyx [35]

Salted

As palmilhas *Smart Insole* da Salted, Figura 2.7, destinam-se essencialmente à área do *fitness*, são fabricada em Ethylene-Vinyl Acetate (EVA) e tem vários sensores incorporados que analisam padrões de caminhada, de marcha e do equilíbrio do utilizador. Possuem 4 sensores de pressão e ainda motores de vibração que permitem o *feedback* a partir de 12 padrões de vibração diferentes, comunicando via Bluetooth com as App. Funciona em conjunto com quatro aplicações para *smartphone*, destinadas a públicos-alvo distintos, que trata as informações recolhidas pela palmilha, de modo diferenciado [38].

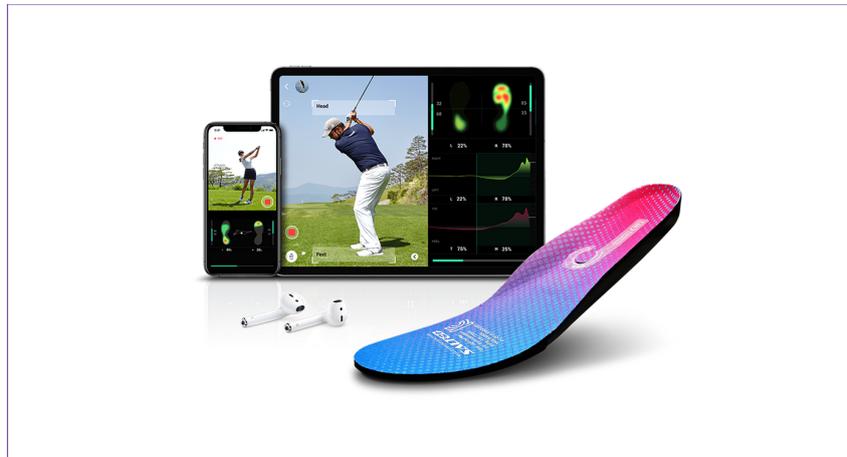


Figura 2.7: Palmilha Salted [38]

Insole

A palmilha Insole da Flextrapower, Figura 2.8, destina-se a doentes diabéticos e tem incorporados vários sensores impressos em grafeno que podem medir e detectar diferenças na temperatura e pressão do pé. A palmilha permite monitorizar continuamente indicadores da saúde do pé. Funciona em conjunto com uma aplicação para *smartphone*, que traduz em tempo real as informações recolhidas pela palmilha. Se for detectada a formação de uma úlcera no pé diabético, o utilizador será notificado automaticamente para que seja possível atuar de modo a reduzir os possíveis danos. Deste modo, estas palmilhas tornam-se num mecanismo preventivo contra o desenvolvimento de uma úlcera no pé [39].



Figura 2.8: Exemplo de palmilha sensora [39]

Lechal

As palmilhas da marca Lechal, destinam-se ao mercado de *fitness*. A tecnologia desenvolvida por esta empresa encontra-se embutida numa pequena cápsula, que será alojada na parte inferior de uma palmilha [40].

O sistema assume-se com o primeiro dispositivo da classe com *feedback* haptico, dando indicações através de vibrações. Para além de ser um dispositivo de navegação, emitindo vibrações indicativas de mudança de direção, também funciona como um *fitness tracker* onde os dados recolhidos e gravados pelas cápsulas, podem ser transferidos para aplicações como o Google Fit ou a Apple Health [40].

Funciona mesmo na ausência do *smartphone*, guardando dados e transferindo-os assim que se encontre na proximidade desse dispositivo. Regista, o número de passos dados, a distância percorrida, a quantidade de calorias despendidas. Tendo em conta que permite navegação sem recurso ao *feedback* visual de um *smartphone*, uma das valências destas palmilhas é a sua utilização por parte de invisuais [40].

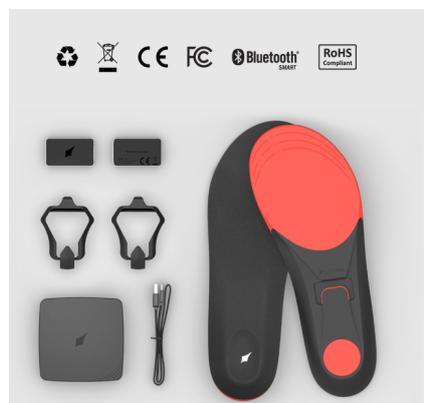


Figura 2.9: Palmilha Lechal [40]

Sensoria Socks

As meias Sensoria, Figura 2.1, funcionam como um monitor da pressão plantar permitindo medições em três regiões plantares de interesse: calcanhar, cabeça do primeiro metatarso e cabeça do quinto metatarso. As meias incluem um alicate para a zona do cóis onde irá encaixar o sistema de transmissão de dados (que também contém um IMU bem como uma bateria). Através de uma App, Figura 2.10, pode-se visualizar, em tempo real, a pressão plantar e cadência, entre outras funcionalidades [26, 37].

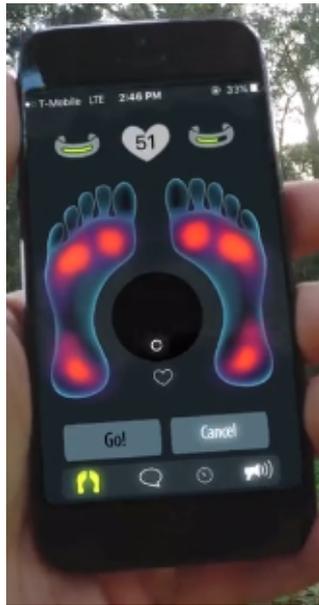


Figura 2.10: Software da meia Sensoria [26]

2.1.2 Sistemas experimentais com sensores

Para além do casos comerciais que possuem sensores, existem vários outros sistemas sensores com a elaboração de um protótipo, referenciados em artigos científicos. Efectivamente, os sistemas de medição de pressão no interior do calçado têm sido amplamente utilizados seja em pesquisas científicas, na área clínica ou a nível desportivo, por forma a avaliar a marcha do utilizador.

Entre o vários artigos analisados verifica-se que, a maioria das aplicações de medições de forças ou de pressões ao nível do pé, são desenvolvidas para medições plantares [21, 41–57]. Porém existem algumas abordagens para medições na zona dorsal do pé [19, 20, 58–60] e apenas foi encontrado um artigo onde tentavam combinar a monitorização da zona dorsal e plantar do pé em simultâneo [61]. Contudo, este último estudo incidia sobre as diferentes pressões obtidas com o uso de sapatilhas adaptadas a diversos desportos, tentando provar que a utilização de sapatilhas

personalizadas melhoram o conforto e previnem lesões. Dentre os artigos encontrados, nenhum fazia referência a medições de pressões plantares e pressões na zona da gáspea em simultâneo e em pacientes diabéticos.

Quanto ao tipo de sensores utilizados, a maioria utiliza sensores resistivos [41–59], vários utilizam sensores têxtil piezoresistivo [20, 51–54] e um estudo indica a utilização de sensores piezoresistivos flexíveis em borracha [62]. Outros dos sensores mais utilizados para medições de forças ao nível do pé são os piezoelétricos [55–57, 60, 61]. Dois dos artigos encontrados referem que utilizam sensores indutivos [62, 63] e um utiliza sensores capacitivos [64].

Todos os sistemas necessitam de uma unidade de controlo, que irá servir para controlar todo o sistema de aquisição de dados pelo sensores. A maioria dos estudos não refere a taxa de aquisição e os que referem indicam que esta se situa entre 10 e 250 Hz. Estes sistemas de aquisição são, normalmente, fixados na perna, cintura ou mesmo em mochilas às costas dos utilizadores. Apesar de prevalecerem os estudos cuja aquisição de dados ocorre por ligação com fios, já existem alguns sistemas que utilizam Bluetooth ou Wireless Fidelity (Wi-Fi) [51, 52, 54–56], um utiliza Zigbee [50] e outro Radio Frequency (RF) [53]. De todos os artigos explorados, apenas um sugere que se encontra a implementar a unidade de controlo no interior da sola do sapato, ficando esta totalmente integrada no mesmo [62]. A tabela 2.1, mostra uma síntese de algumas das características de sistemas desenvolvidos para medição de forças ou pressões no pé.

2.1.3 **Localização dos sensores**

Os sistemas de pressão plantar disponíveis no mercado ou utilizados em pesquisas científicas variam no número de sensores e na sua configuração de modo a cumprirem diferentes requisitos. Existem estudos em que a sola do pé pode ser dividida em 15 áreas diferentes calcânhar (área 1-3), mediopé (área 4-5), metatarso (área 6-10) e dedos do pé (área 11-15), como ilustrado na Figura 2.11 [65].

Essas áreas suportam a maior parte do peso corporal e são ajustadas pelo equilíbrio do corpo [29]. Deste modo, idealmente, seriam necessários 15 sensores para monitorizar alterações de forças plantares, tendo apenas por base a anatomia.

Contudo, no que se refere ao risco de ulceração no pé de diabéticos, são conhecidas as zonas onde é maior este risco, tendo em conta o tipo de pé, estando estas indicadas na Figura 2.12.

A maioria dos estudos efetuados propõem a utilização de cinco, seis ou oito pontos de medição na zona plantar [47, 48, 55, 57, 62, 64, 66, 67], sendo as configurações mais utilizadas as de oito pontos de medição. A localização dos sensores abrange

Tabela 2.1: Características de sistemas desenvolvidos para medição de forças/pressões no pé

Marca dos sensores/ Tipo de sensor	Zona de aplicação dos sensores	Tipo de sistema	Número de sensores	Localização	Tipo de comunicação	Taxa de aquisição	Autor	
F-Scan / Resistivos	Plantar	Palminha	954 (mas variável com o número da palminha)	Ao longo de toda a zona plantar	Com fios	100 Hz	Lord et al., 2000 [41]	
		Palminha		Nas cabeças dos metatarsos: hálux e calcâneo	Com fios	50 Hz	Mori et al., 2012 [42]	
		Palminha		metatarsos: hálux, metópico, lateral e calcâneo	Com fios	50 Hz	Takano et al., 2014 [43]	
Interlink Electronics/ Resistivos	Plantar	Palminha	7		Com fios	35 Hz	Zhu et al., 1993 [45]	
		Palminha	8	metatarsos: hálux, metópico, lateral e calcâneo	Com fios	35 Hz	Wertsch et al., 1995 [46]	
		Palminha	6	Cabeças dos 1 ^o , 2 ^o , 4 ^o e 5 ^o metatarsos; hálux, metópico, calcâneo	Com fios	100 Hz	Fennery et al., 2004 [47]	
		Palminha	7	Cabeças dos 1 ^o , 2 ^o , 3 ^o , 4 ^o e 5 ^o metatarsos; base do 5 ^o metatarsos; hálux e calcâneo	Com fios	60 Hz	Brown et al., 2004 [48]	
		Palminha	4	Cabeças dos 1 ^o e 5 ^o metatarsos; hálux e calcâneo	Com fios	10 Hz	Aaes, 2014 [49]	
		Dorsal	8	Ao longo da zona dorsal do pé				Olaso et al., 2007 [58]
		Plantar	5	Cabeças dos 1 ^o , 3 ^o e 5 ^o metatarsos; metópico e calcâneo	Zigbee			Dabiri et al., 2008 [50]
Flexforce, Tekscan/ Resistivos	Plantar	Palminha	5	Cabeças dos 1 ^o , 2 ^o , 3 ^o , 4 ^o e 5 ^o metatarsos; hálux e calcâneo	Com fios	250 Hz	Ostadabadi et al., 2012 [21]	
		Dorsal	14	Distribuídos pela zona dorsal	Sem fios		Rupérez et al., 2012 [19]	
		Dorsal	3	1 ^o junta metatarso-falange, cuneiforme intermédio e calcâneo			Takessue et al., 2019 [59]	
		Dorsal	8	Cabeça dos metatarsos, metópico e calcâneo	Bluetooth		Perrier et al., 2014 [51]	
Textil piezoresistivo	Plantar	Meia	4	Cabeça dos metatarsos e peito do pé	Com fios	30 Hz	Harbaut et al., 2016 [20]	
		Meia	1	A colocar onde o utilizador pretende	Bluetooth		Ravighione et al., 2017 [52]	
		Meia	4	Cabeças dos 1 ^o , 3 ^o e 5 ^o metatarsos e calcâneo	RF		Lin et al., 2017 [53]	
		Meia	6	Cabeças dos 1 ^o e 4 ^o metatarsos; hálux; metópico e calcâneo	Bluetooth		Tan et al., 2021 [54]	
		Meia	54	Ao longo da dorsal do pé			Hégen et al., 2010 [60]	
Sensor piezoeletrico	Plantar e dorsal	Palminha	8	Cabeças dos 1 ^o , 3 ^o e 5 ^o metatarsos; 4 ^o dedo; hálux; metópico e calcâneo	Wi-Fi Bluetooth	100 Hz	Mei et al., 2014 [61]	
		Palminha	6	Região plantar e região dorsal	Bluetooth	100 Hz	Arenuzua et al., 2016 [56]	
		Palminha	8	Cabeças dos metatarsos; hálux e calcâneo	Com fios		Rajala et al., 2017 [57]	

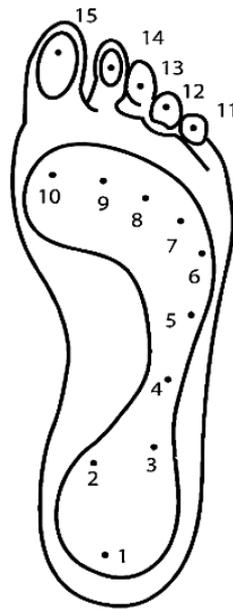


Figura 2.11: Áreas anatômicas do pé [65]

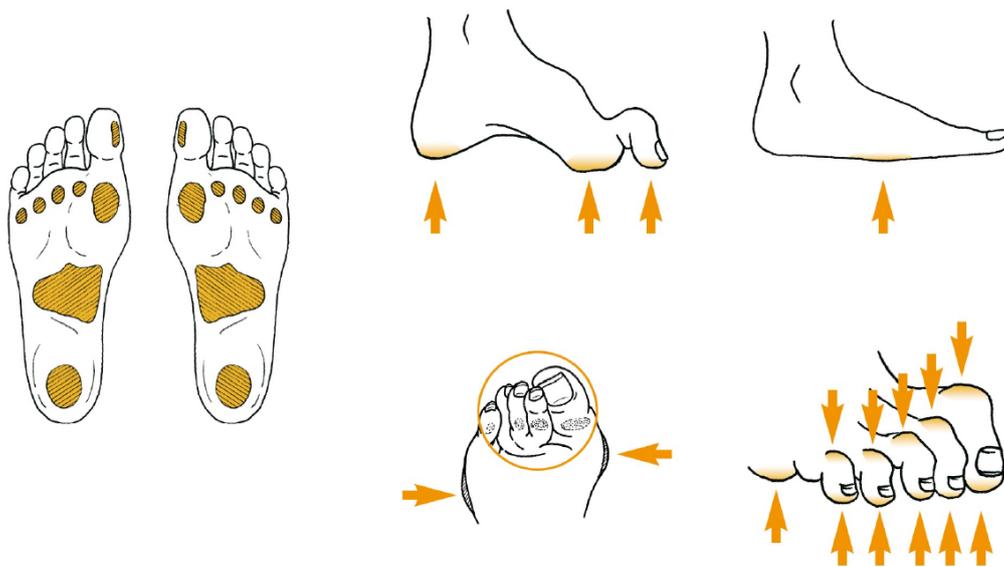


Figura 2.12: Zonas de maior risco de úlceração do pé diabético [5]

os principais pontos anatômicos da região plantar do pé onde é maior o risco de ulceração, existindo sempre sensores localizados na zona do hálux, metatarsos e calcanhar. Nos sistemas onde estão aplicados um maior número de sensores optam por medir valores também na zona do mediopé. As figuras seguintes ilustram as localizações mais utilizadas na região plantar com 5 pontos (Figura 2.13a), 6 pontos (Figura 2.13b) e 8 pontos de medição plantar (Figura 2.13c).

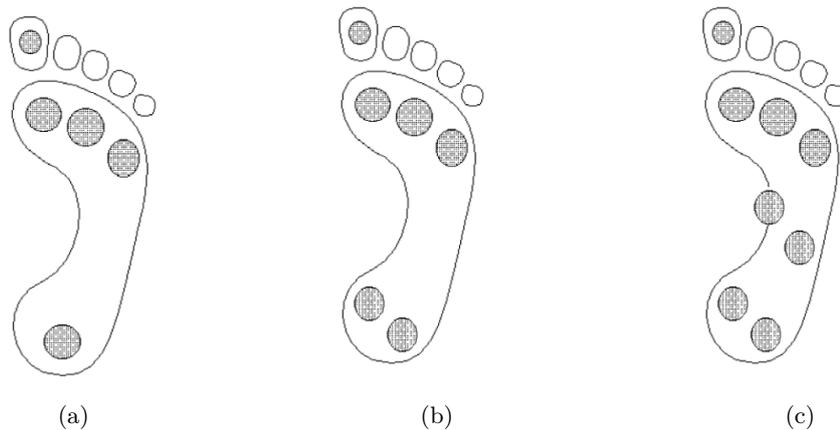


Figura 2.13: (a) Localização de cinco sensores na zona plantar [66]; (b) Localização de seis sensores na zona plantar [47]; (c) Localização de oito sensores na zona plantar [62]

Relativamente à localização dos sensores na zona dorsal do pé, trata-se de uma abordagem ainda pouco explorada. Existem poucos estudos que permitam avaliara as forças exercidas nessa zona do pé, não existindo por isso uma localização recomendável e padronizada a este nível. Na maioria dos casos, os sensores são distribuídos por toda a região dorsal [19, 58–60]. Contudo, face às zonas de maior risco de ulceração, evidenciadas na Figura 2.12, estas serão, provavelmente, na zona superior das falanges.

2.2 Revisão tecnológica sobre sensores de força

Atualmente existe uma vasta gama de sensores que poderão ser utilizados por forma a monitorizar forças e, por consequência, pressões plantares. As principais especificações para um bom desempenho do sensor incluem a sua linearidade, histerese, área da zona de deteção, pressão/força máxima admitida e sensibilidade à temperatura [30, 32].

- Histerese - Quando sobre um sensor é aplicada uma força e quando essa mesma força deixa de ser exercida sobre o sensor, este tem respostas diferentes face ao valor gerado. Idealmente, um sensor é capaz de medir as variações do parâmetro de entrada, independentemente do sinal dessas variações (diminuição ou aumento da entrada). A histerese é uma medida desta propriedade.
- Linearidade - A linearidade de um sensor expressa o quanto a sua curva característica, que é actualmente medida, se afasta da curva tida como ideal. No caso dos sensores de força, importa então que ocorra uma relação linear entre uma característica do sinal por ele gerado quando sujeito a uma força e a força sobre ele aplicada.

- Sensibilidade à temperatura - Os sensores podem gerar valores de força diferentes consoante a temperatura a que estão sujeitos. Tal deve-se aos materiais que fazem parte da constituição do sensor, pois respondem de maneira diferente à mudança de temperatura. Deve-se selecionar um sensor com baixa sensibilidade a alterações de temperatura [29,32].
- Força máxima admitida - A principal especificação de um sensor de força, que poderá ser utilizado para medir pressão plantar, é a sua gama de operação. Como diferentes aplicações requerem o registo de diferentes valores é necessário optar por sensores que permitam medições entre os valores máximos e mínimos que se pretende medir [29]
- Área de deteção do sensor - O tamanho e a colocação do sensor também são críticos. Um sensor com elevada área de deteção, pode fornecer indicações imprecisas de pressão quando ocorre um pico de força. Assim é recomendável áreas de deteção de 5 mm x 5 mm, no mínimo, para aplicação em pontos de pressão plantares específicos [29]
- Tolerância à fadiga e repetibilidade - Quando submetidos a pressões constantes e também devido à flexão do pé durante o ciclo de marcha humana, os sensores localizados na zona plantar têm tendência a sofrer danos, pelo que devem ser resistentes à deformação e permitir a obtenção de resultados com confiabilidade ao longo do tempo [29,32]

Seguidamente far-se-á uma abordagem aos diferentes tipos de sensores, mais utilizados para medição de forças plantares, separando-os de acordo com as suas características.

2.2.1 Sensores Capacitivos

Estes sensores, tal com evidenciado na Figura 2.14, são compostos por dois condutores elétricos separados por uma camada dielétrica elástica. Devido à elasticidade dessa camada, a distância entre os dois materiais elétricos poderá mudar, quando neles são aplicadas forças, produzindo, como consequência, variações na diferença de potencial medida entre os condutores [16,34,68,69].

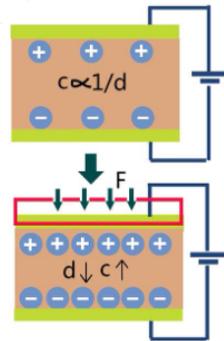


Figura 2.14: Esquema ilustrativo do mecanismo representativo de funcionamento de um dispositivo capacitivo [68]

Estes sensores tem a vantagem de possuírem elevada sensibilidade, uma resposta rápida, baixo consumo e amplo alcance [68, 69]. Porém são susceptível a interferências eletromagnéticas [69].

Na Figura 2.15, encontra-se um diagrama exemplificativo da constituição de um sensor capacitivo em cobre e a respetiva aplicação destes numa palmilha.

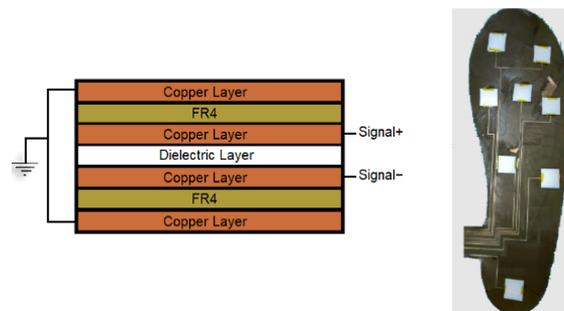


Figura 2.15: Camadas de constituição de um sensor capacitivo e respetiva aplicação numa palmilha [64]

2.2.2 Sensores Resistivos

Os sensores resistivos são produzidos a partir de um polímero condutor, que possui a capacidade de variar a sua resistência, quando sujeito a forças externas. Neste tipo de sensores, usualmente quando a força aplicada aumenta, e conseqüentemente a pressão exercida também aumenta, a resistência do material diminui [16, 34, 69].

Os sensores resistivos, possuem uma estrutura simples, têm baixo custo, são facilmente integrados num sistema e a aquisição do sinal gerado é relativamente simples [69]. São sensores amplamente estudados pelo que o seu comportamento é

bem conhecido, apresentando uma resposta linear dentro de determinados valores de pressão [16].

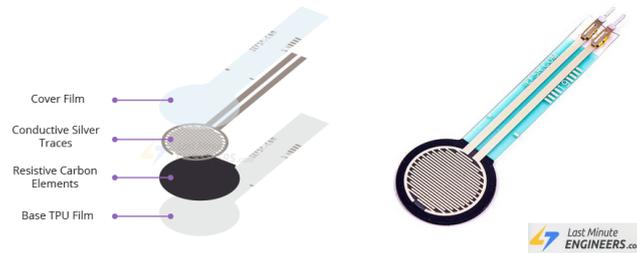


Figura 2.16: Camadas de constituição de um sensor FSR

Na Figura 2.16, encontra-se um diagrama exemplificativo da constituição de um sensor FSR.

2.2.3 Sensores piezoresistivos e piezoelétricos

Nos sensores piezoelétricos (Figura 2.17a) a variação da força aplicada é convertida em carga elétrica podendo esta ser medida. Já nos sensores piezoresistivos (Figura 2.17b), quando o material sofre estiramento, a variação da sua resistência elétrica pode ser mensurada [16, 34, 68, 69].

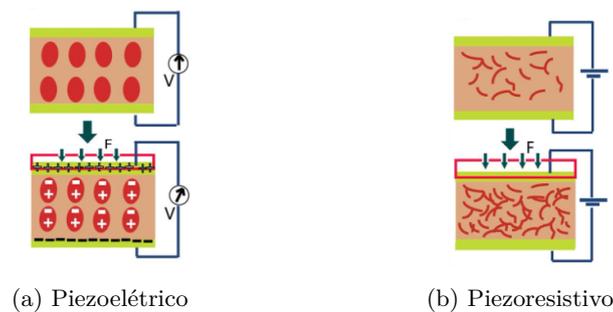


Figura 2.17: Esquema ilustrativo do mecanismo representativo de funcionamento de um dispositivo Piezoelétrico e Piezoresistivo [68]

De um modo geral, os sensores de pressão piezoresistivos, são mais utilizados e mais estudados do que os sensores de pressão piezoelétricos. Os sensores piezoresistivos possuem baixo consumo elétrico, tem um processo de fabrico relativamente simples e baixo custo [68]. Os sensores piezoelétricos possuem alta sensibilidade e geram espontaneamente sinal elétrico. Contudo como os materiais piezoelétricos possuem propriedades piroelétricas são afetados pela temperatura e o sinal produzido poderá sofrer desvios com o passar do tempo [69].

2.2.4 Sensores Têxteis

No caso dos sensores têxteis, estes podem ter duas formas diferentes de fabrico. Numa, sobre uma matriz têxtil é utilizada tinta condutora de modo a criar um material que seja simultaneamente fino mas sensível à pressão. Outra opção é impregnar o fio têxtil com substâncias que, após devidamente entrançado, tornem a matriz formada sensível à pressão.

Este tipo de sensores está correntemente a sofrer um enorme desenvolvimento visto que, a sua maioria, tem comportamentos não lineares além de sofrerem de uma histerese significativa [16, 34].

Contudo, estas desvantagens estão a ser colmatadas com processos inovadores de fabrico pelo que estão a surgir no mercado tecidos sensores com um bom padrão de qualidade.

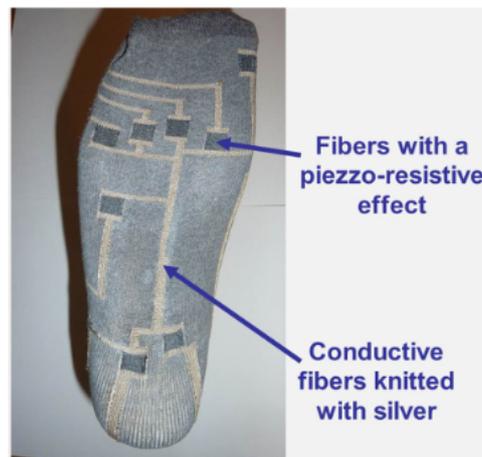


Figura 2.18: Meia tricotada com sensores têxteis [51]

A Figura 2.18 mostra uma aplicação de uma meia sensora tricotada onde o sensor é constituído por fio têxtil entrelaçado com a matriz da meia.

2.3 Conclusão

Neste capítulo pretendeu-se elaborar um breve levantamento sobre o estado de arte, no que se refere a sistemas de sensorização plantares e na zona dorsal do pé.

Verificou-se que, ao longo dos anos tem sido desenvolvidos vários estudos onde se produziram protótipos com sensorização plantar. Nestes, a maioria utiliza oito sensores distribuídos em zonas onde é conhecido o risco acrescido de ulceração do pé.

Relativamente à zona dorsal, existem poucos estudos sendo uma área onde importa obter-se um conhecimento mais alargado dos valores das forças aplicados.

Quanto aos sensores mais utilizados, estes são maioritariamente do tipo resistivo. Porém estão a surgir no mercado novos sensores, com uma base têxtil, que parecem ser promissores, para aplicação na zona plantar.

Para além dos sistemas experimentais de medições de pressão plantares existem já soluções comerciais vocacionadas para o pé diabético. Contudo estas são soluções que ainda não estão ao alcance da maioria da população.

Capítulo 3

Arquitectura do Sistema

Neste capítulo pretende-se fazer uma abordagem à arquitetura geral do sistema implementado, tentando justificar as opções no que concerne aos dispositivos seleccionados (unidade de microcontrolador, sensores de força e unidade de medição inercial).

3.1 Arquitectura geral do sistema

O sistema que se pretende desenvolver, irá permitir medir forças aplicadas na região plantar bem como na zona dorsal do pé. Como tal, um dos componentes base deste sistema será, obrigatoriamente, dispositivos que, quando sobre eles é exercida uma força, gere uma variação de uma grandeza que possa ser mensurada.

Além dos sensores de força, considerou-se útil a utilização de um IMU, que permitisse recolher dados inerciais do movimento do utilizador do sistema.

Contudo, estes sensores necessitam de uma unidade que controle e monitorize os seus sinais tendo, obrigatoriamente, de estar associados a um MCU. As principais funções do MCU serão gerir a alimentação dos sensores, converter o sinal analógico emanado pelos sensores num sinal digital (no caso dos sensores de força) e armazenar os dados em memória.

Outro dos requisitos iniciais foi que todos os dados terão de ser enviados a partir uma ligação sem fios para um *smartphone*.

Tendo por base, o levantamento efectuado no Capítulo 2, chegou-se à conclusão que o sistema a ser implementado teria de possuir no mínimo 8 sensores de força na

zona plantar e 4 na zona da gáspea. Deste modo, a unidade de MCU teria de ter, pelo menos, um conversor ADC com 12 canais.

Por outro lado, é sabido que a maioria dos IMU disponíveis no mercado comunicam por Serial Peripheral Interface (SPI) ou Inter-Integrated Circuit (I²C) pelo que o MCU deveria possuir a capacidade de estabelecer estas formas de comunicação.

Outro dos requisitos essenciais deste sistema está relacionado com o consumo energético, visto pretender-se que este sistema venha a integrar um *wearable*. Uma parte considerável do consumo energético está relacionado com a transmissão de dados. Assim, como este sistema irá necessitar de uma transferência de dados constante, assumiu-se como requisito que o MCU possuísse a capacidade de transmitir dados por BLE.

A Figura 3.1 indica um diagrama simplificado da arquitectura do sistema implementado. Nas secções seguintes ir-se-á abordar as características de cada um dos componentes implementados no sistema.



Figura 3.1: Arquitectura do sistema

3.2 MCU

Tendo em conta os requisitos iniciais do sistema, procurou-se encontrar um MCU que permitisse receber e controlar sinais dos sensores bem como transmiti-los para outros dispositivos. Considerou-se assim três opções de placas de desenvolvimento com microcontroladores diferentes, cujas características encontram-se expressas na Tabela 3.1.

Analisando as características de cada placa de desenvolvimento, a Arduino Uno foi excluída visto não possui módulo Bluetooth integrado e o ADC possui apenas 8 canais, não cumprido por isso os requisitos iniciais do sistema. A placa nRF5232,

Tabela 3.1: Comparação de características de placas de desenvolvimento

Dispositivo	Wireless	ADC	Interfaces de comunicação	Consumo
Arduino Uno [70]	—	8 canais 10bit	1 X I2C 1 X SPI USART	TX (3V) 1.5mA
ESP32 DevKitC V4 [71]	Bluetooth LE v4.2	2 X 12bit 16 canais	4 X SPI 2 X I2S 2 X I2C 3 X UART	TX (3,3V) 130 mA
nRF5232 [72]	Bluetooth 5	12 bit 8 canais	3 X SPI I2S 2 X I2C UART	TX (3V) 5,3 mA

foi também ela excluída, pois apesar de possuir Bluetooth 5 e ter baixo consumo energético, o seu ADC só possuía 8 canais. Assim, apesar de ser a que possui maior consumo, optou-se pela utilização da ESP32 DevKitC V4, por ser a única que, para além de permitir comunicar por BLE possuía dois ADC com 16 canais no total.

Analisando com mais pormenor as características da placa verifica-se que esta tem por base o módulo ESP32-WROOM-32D [71]. Tem integrado um oscilador de 40MHz, opera a uma tensão entre 3,0 V e 3,6 V com uma corrente média na ordem dos 80 mA [71, 73]. Funciona entre -40 °C e 85 °C e possui 448 KB de Read-only memory (ROM) e 520 MB de Static Random Access Memory (SRAM). Tem incluído dois ADC de 12 bit, possuindo o ADC1 6 canais e o ADC2 10 canais. Tem integrado ainda dois Digital-to-Analog Converter (DAC) de 8 bit. Permite a comunicação por múltiplos protocolos como SPI, Integrated Interchip Sound (I²S); I²C, Universal Asynchronous Receiver/Transmitter (UART), Wi-Fi e Bluetooth v4.2 BR/EDR com BLE. Possui ainda, 2 *timers* de 64 bit que permitem a utilização de interrupções. As interrupções possibilitam que um programa possa ser interrompido por um dado evento, sendo chamada uma rotina específica de serviço à interrupção, Integrated Switching Regulator (ISR), por forma a tratar o evento que levou à interrupção [71, 73].

3.2.1 BLE

O Bluetooth Low Energy, BLE, é uma tecnologia que pretende permitir comunicação compatível com o Bluetooth clássico, mas com um consumo energético consideravelmente mais baixo. Emite na mesma frequência de 2,4 GHz do Bluetooth, utilizando um sistema de modulação mais simples. Existem enormes diferenças entre o BLE e o Bluetooth clássico, nomeadamente no que se refere ao tipo de aplicação a que cada um se destina [74]. A Tabela 3.2 refere as principais diferenças entre estas duas tecnologias.

Tabela 3.2: Comparação entre Bluetooth clássico e BLE [74]

Bluetooth Clássico	BLE
Utilizado para aplicações que necessitem de streaming, como streaming de áudio e transferência de ficheiros	Utilizado para recolha de dados de sensores, controlar dispositivos e aplicações que exijam baixa largura de banda.
Não é otimizado para baixo consumo (1 W) mas possui uma taxa mais elevada de transferência de dados (3 Mbps)	Criado especificamente para ter baixo consumo energético (0,01 a 0,5 W) e para envios de poucos dados, taxa de transferência de dados menor (1 Mbps)
Opera em 79 canais de RF	Opera em 40 canais de RF
Possui 32 canais de discovery	Possui 3 canais de discovery

Apesar de existirem outras tecnologias cujo consumo energético é igualmente otimizado para envios de dados de pequena dimensão, tal como o ZigBee, o BLE tem a grande vantagem de ser compatível nativamente com a maioria dos *smartphones* existentes no mercado. O BLE é amplamente utilizado para transferências de dados em dispositivos *Wearables* [74].

A arquitetura do BLE, apresentada na Figura 3.2 apesar de mais simples do que o Bluetooth clássico, ainda é complexa considerando-se que a sua descrição excede o âmbito do trabalho desenvolvido. Contudo, existem conceitos que se revelam essenciais, como o Generic Attribute Profile (GATT) e o Attribute Protocol (ATT), ambos da camada *Host*.

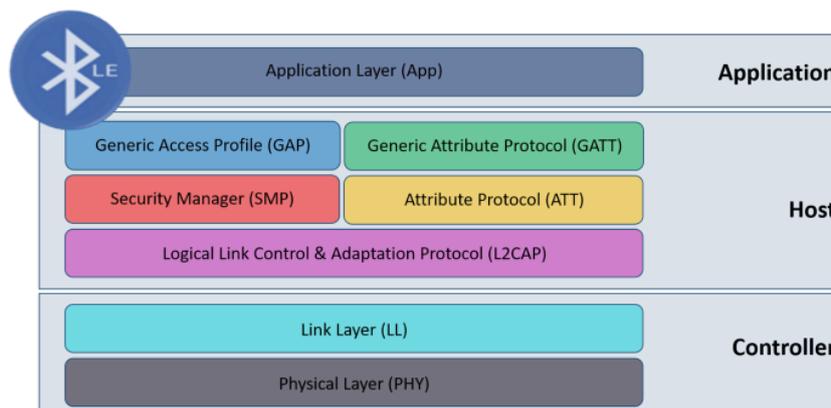


Figura 3.2: Pilha protocolar BLE [75]

O Attribute Protocol (ATT) define o modo como o servidor irá apresentar os dados ao cliente e o modo como esses dados se encontram estruturados, definindo-se este por atributo. Por exemplo os serviços e as suas características são tipos de

atributos. Associado a um atributo existe sempre um Universally Unique Identifier (UUID), um número de 16 bit, no caso de se utilizar atributos definidos pelo Bluetooth SIG, ou um número de 128 bit que poderá ser gerado aleatoriamente [74].

O GATT define o formato dos serviços, das suas características, e os procedimentos para interagir com estes atributos, tal como o *service discovery*, *characteristic reads*, *characteristic writes*, *notifications* e *indications*. O GATT é utilizado aquando do estabelecimento de uma ligação entre dois dispositivos BLE [74, 76].

Um serviço é um grupo de um ou mais atributos, alguns dos quais são características. Um serviço pode conter mais do que uma característica, para além de outros atributos que não são características mas que ajudam a estruturar os dados num serviço. A Figura 3.3 evidencia a estrutura de um serviço com múltiplas características.

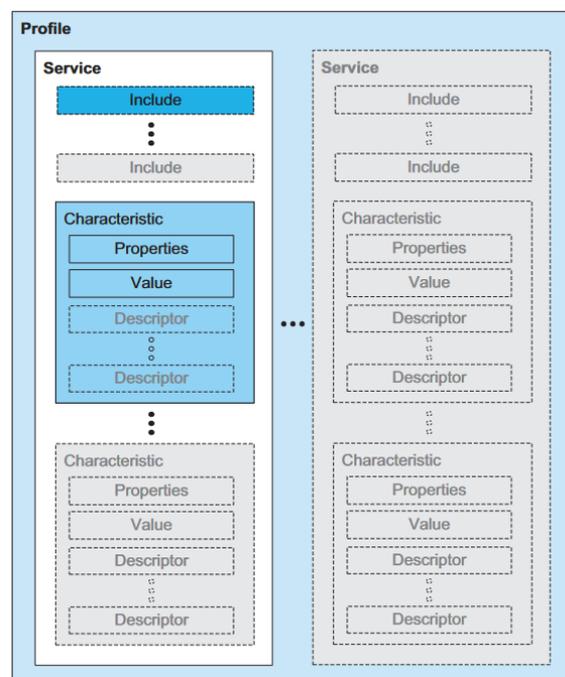


Figura 3.3: Serviços e características BLE [77]

Por outro lado, as características fazem sempre parte de um serviço e representam uma informação ou um dado que o servidor quer fornecer ao cliente. A propriedade de uma característica define a forma como o valor da característica pode ser utilizado: pode tomar o valor de *read*, *write*, *write without response*, *notify* ou *indicate* [76]. Dentre os valores das propriedades convém clarificar a diferença entre o *notify* e o *indicate*. A propriedade *indicate* é uma operação assíncrona de notificação, inicializada pelo servidor para o cliente. Requer que, após receber a notificação, o cliente envie um *acknowledgement* ao servidor. Por outro lado o *notify* é em tudo semelhante ao *indicate*, simplesmente não é exigido ao cliente um

acknowledgement, indicando que recebeu a informação [76].

Um descritor pode ser utilizado para armazenar informação relacionado com o valor da característica, por exemplo a indicação da unidade do valor da característica [74, 76].

Um dos parâmetros definidos no estado *connection* poderá ser o Data Length Extension (DLE), que quando está no modo *enable*, permite que a quantidade de dados transmitidos seja até 251 bytes e até 27 bytes quando está no modo *disable*. Esta funcionalidade foi introduzida nas especificações do Bluetooth 4.2. Porém, por defeito, esta encontra-se no estado *disable*, e como existem outras informações obrigatoriamente a serem transmitidas a cada conjunto de dados (tal como o cabeçalho) implica que, na realidade, apenas estejam disponíveis 20 bytes para envio de dados, no máximo, a cada transmissão [74]. Apesar de ser possível aumentar o tamanho do pacote de dados a ser enviado, tal terá influência nos tempos de transmissão de dados podendo tornar o sistema de envio mais lento.

3.3 Sensores de força

Existe, hoje em dia, uma variedade enorme de sensores de força, produzidos pelas mais variadas marcas.

Após análise de diversas *datasheets* e de se ter decidido utilizar sensores resistivos, consideram-se como sendo os mais promissores os elencados na Tabela 3.3 apresentando esta o resumo das principais características destes sensores. Apesar de os sensores utilizados serem, na realidade sensores de força, são diversas vezes referenciados ao longo do texto como sensores de pressão, visto existir uma relação matemática entre estas duas grandezas física e, na maioria dos estudos encontrar-se valores obtidos para pressões plantares e não para forças plantares.

Tabela 3.3: Características de sensores resistivos

Sensor	Princípio de funcionamento	Diâmetro (mm)	Área da zona ativa (mm^2)	Alcance	Comprimento total (mm)
Sensor Interlink FSR UX 400 [78]	FSR	5,08	20,3	até 150 N	38
Sensor Interlink FSR UX 402 [78]	FSR	14,68	169,2	até 150 N	56
Sensor Flexiforce A201 [79]	FSR	9,53	71,3	4N 111N 445N	50,8 101,6 152,4 190,5
Sensor Flexiforce A401 [80]	FSR	25,4	506,7	111N	56,9
Sensores têxtil SensingTex [81]	Têxtil Resistivo	10	78,53	até 25 kg/cm^2	Não aplicável

Os sensores têm de possuir a capacidade de resistir a sucessivas forças/pressões podendo estas ocorrerem em situações extremas, pelo que se considerou importante

realizar um estudo de modo a verificar se a sua utilização poderá ser compatível com o fim a que se destinam. Assim a secção seguinte resume várias hipóteses de utilização extrema bem como valores expectáveis de pressão exercidas sobre um sensor nesses casos.

3.3.1 Estudo da gama de medição dos sensores

Para se prever o valor limite de pressão máxima a que um determinado sensor FSR ou têxtil (SensingTex) poderá estar sujeito, quando colocado na região plantar do pé, equacionou-se uma pessoa em postura anatómica unipedal (postura corporal vertical sobre um único pé) na seguintes situações:

- apoio em bicos de pé;
- apoio sobre o calcanhar;
- apoio em pontas de ballet;
- apoio durante uma corrida.

Destas, algumas situações podem ocorrer naturalmente, mas outras, são consideradas extremas e pouco prováveis de ocorrer. O objetivo foi identificar pressões limites que deverão ser suportadas pelos sensores. Com estes dados, será possível atender ao alcance do sensor e identificar qual o mais indicado para esta aplicação.

Nas hipóteses exploradas será considerada uma pessoa com 100 kg de massa e os sensores com as características indicadas na Tabela 3.3. Os valores de pressão a apurar nesta abordagem serão usados para se ter uma ideia inicial da ordem de grandeza das pressões limite envolvidas.

Área mínima de contacto do pé com o solo e respetiva pressão

Tendo, por hipótese, a pessoa uma massa de 100 kg, em repouso esta irá exercer sobre o chão uma força igual ao seu peso (F_g) e portanto com uma intensidade $F_g = 980 N$.

Contudo, para avaliar a pressão plantar e tendo em consideração que a Pressão (P) é igual à razão entre a força exercida numa determinada área, $P = F/A$, é necessário avaliar qual a área onde estará a ser exercida o peso em cada situação. Deste modo, procurou-se encontrar medidas da área de pé.

Tendo em conta que o tamanho do pé poderá variar, optou-se por utilizar uma medida média, obtida a partir de oito indivíduos (quatro de cada sexo) com tamanho do pé compreendidos entre o número 36 e o número 46. Estas medidas encontram-se expressas na Tabela 3.4 e de acordo com as zonas indicadas na Figura 3.4 [82], onde *GT* indica a área do hálux, *Toes*, a área dos dedos dois a cinco, *LatMet*, *MidMet*, e *MedMet* correspondem às áreas das zonas do metatarso lateral, médio e medial

respectivamente; *LatArch*, *MidArch*, e *MedArch*, às áreas das zonas do arco lateral, médio e medial respectivamente e *Heel*, à área da zona do calcâneo.

Tabela 3.4: Medidas da área média da sola do pé, por zona [82]

Área média (cm ²)	Regiões da sola do Pé								
	GT	Toes	MedMet	MidMet	LatMed	MedArch	MidArch	LatArch	Heel
	13,2	16,9	17,2	16,6	14,2	19,3	20,4	21,6	38,8

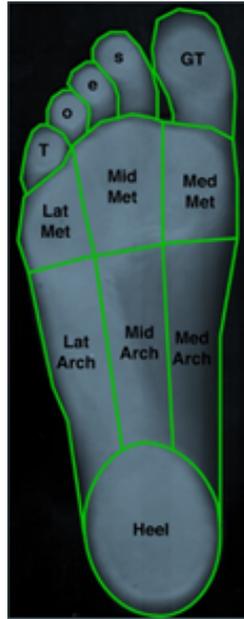


Figura 3.4: Zonas de medição da área do pé [82]

Hipótese 1: Pessoa em “bicos de pé”

Nesta hipótese pretende-se determinar a pressão plantar de uma pessoa em “bicos de pé”. Assim, a área de contacto com o solo (em posição unípede) para esta situação será a soma das áreas identificadas como GT, Toes, LatMet, MidMet e MedMet na Figura 3.4. De acordo com a Tabela 3.4 essa área será:

$$A_{BicosPe} = 78,1 \text{ cm}^2$$

Hipótese 2: Pessoa em apoio sobre o calcanhar

Nesta hipótese pretende-se determinar a área de contacto com o solo de uma pessoa apenas apoiada sobre o calcanhar (em posição unípede). De acordo com a Tabela 3.4 essa área será:

$$A_{calcanhar} = 38,8 \text{ cm}^2$$

Hipótese 3: Pessoa em pontas de ballet (extremo)

Nesta hipótese pretende-se determinar força exercida na área de contacto com o solo de uma pessoa apoiada sobre a biqueira de uma sabrina de pontas para ballet

(uma situação completamente extrema). Apesar da área da plataforma de um sapato de pontas de ballet poder variar, a sua área média pode ser considerada próxima de 15 cm^2 [83].

Hipótese 4: Pessoa a realizar uma corrida

Neste caso, pretende-se obter o valor da força exercida na área de contacto com o solo, de uma pessoa que se encontre a realizar uma corrida. Sendo este movimento complexo não se pretendendo efetuar cálculos biomecânicos, mas apenas avaliara o adequação dos sensores a este caso, optou-se por utilizar uma aproximação do valor máximo da força de reação vertical que a superfície exerce no pé, utilizando-se a equação 3.1.

$$F_z = m \times g \frac{t_c + t_a}{t_c} \quad (3.1)$$

Nesta equação, F_z representa a força de reação vertical, g a aceleração da gravidade, t_c o tempo de contacto de pé com o chão e t_a o tempo aéreo. Para que esta equação seja válida considera-se que a velocidade do corredor é constante e que não existe deslocamento do centro de massa do corpo em cada passada. Cada passo pode ser traduzido pela soma de duas componentes, uma onde o tempo de contacto com o solo, t_c , ocorre na presença de uma força de reacção vertical e outra durante a fase aérea (t_a) onde a força será nula. Nestas condições, um corredor irá suportar em média, o peso do seu corpo ($m \times g$) durante o tempo de uma passada ($t_c + t_a$) [84]. Utilizando-se uma nova aproximação onde $t_c=t_a=0,25$ s, o valor de F_z será de aproximadamente 2000 N. Supôs-se ainda que esta força será exercida apenas no calcanhar, por ser a zona de menor área e, como tal, irá gerar maiores valores de pressão. Não foi utilizada a zona do antepé, zona em que durante a corrida ocorre um maior apoio do pé visto esta possuir uma maior área.

Cálculos efetuados

Por forma a aferir se os sensores seleccionados possuíam um alcance que permitissem a sua utilização, foram efetuados cálculos, supondo que existe uma distribuição homogénea do peso na área de contacto descrita em cada uma das hipótese anteriores. Assim, determinou-se quantos sensores de cada tipo caberiam nessa área e determinou-se qual a força exercida em cada sensor. Na Tabela 3.5 encontra-se um resumo dos cálculos efetuados para cada situação bem como para cada sensor considerado.

Tabela 3.5: Resumo das forças e pressões estimadas que se prevêem aplicadas em cada sensor

	Sensor FSR UX 400		Sensor FSR UX 402		Sensor A201		Sensor A401		SensingTex		
	Gama de medição: 0,5N até 150 N		Gama de medição: 0,5N até 150 N		Gama de medição: 0 N até 4448 N		Gama de medição : 0 N até 31138 N		Gama de medição: 0 e 25 kg/cm ²		
	F (N)	P (kPa)	F (N)	P (kPa)	F (N)	P (kPa)	F (N)	P (kPa)	F (N)	P (kPa)	P (kg/cm ²)
Bicos de pés	2,55	125,6	20,85	123,2	8,91	124,9	61,25	120,8	9,8	124,8	1,27
Apoio calcanhar	5,10	251,4	42,61	251,8	17,8	249,9	122,5	241,8	19,6	249,6	2,55
Pointe ballet	13,24	652,4	108,98	643,3	46,7	654,5	326,7	644,7	51,6	656,8	6,7
Corrida calcanhar	10,4	513,1	86,9	513,9	36,4	510	250	493,7	40	509,3	5,2

Após a análise da Tabela 3.5, considera-se que, qualquer um dos sensores analisados poderá ser utilizado na aplicação que se pretende desenvolver, visto as situações extremas encontrarem-se dentro da gama de medição de cada sensor. Para o sistema implementado optou-se pela utilização de sensores têxteis da marca SensingTex para a zona plantar e sensores Interlink FSR UX 402 para a zona da gáspea. Nas secções 4.1 e 6.1 irão ser analisados, com maior detalhe, os vários sensores estudados assim como os factores que levaram à sua selecção.

3.4 IMU

Os acelerómetros poderão permitir analisar o tipo de movimento realizado em cada um dos pés dos utilizadores o que poderá vir a traduzir-se numa vantagem. Contudo a opção a implementar será a utilização de um IMU (Inertial Measurement Unit) que integra num único periférico um acelerómetro e um giroscópio. Este poderá fornecer um maior número de informações sobre a dinâmica da passada do utilizador, da sua cadência, entre outros parâmetros importantes que poderão ajudar o sistema a melhor monitorizar o utilizador em causa. Dentre os vários IMU existentes no mercado, analisaram-se algumas hipóteses cujas características encontram-se descritas na Tabela 3.6

Tabela 3.6: Características de IMU

Dispositivo	Tensão de funcionamento (V)	Interfaces de comunicação	Sensores integrados	Consumo (mA)
MPU 6050 [85]	2,375 – 3,46	I ² C	Acelerómetro Giroscópio Temperatura	3,9
BHI 160(B) [86]	1,71 – 3,6	I ² C	Acelerómetro Giroscópio	1,5
ICM-20689 [87]	1,71 – 3,45	SPI I ² C	Acelerómetro Giroscópio	3

Na solução implementada optou-se por utilizar o MPU 6050 por ser único imediatamente disponível para utilização. O MPU 6050 é um dispositivo do tipo *MotionTracking* de 6 eixos, que combina no mesmo dispositivo um acelerómetro de 3 eixos, um giroscópio de 3 eixos, um Digital Motion Processor (DMP) para além de possuir ainda sensor de temperatura [85]. A comunicação é feita por I²C e possibilita entradas diretas de outros dispositivos e sensores externos. Devido ao facto de interligar 6 eixos, DMP e *firmware* de calibração, este IMU irá permitir um óptimo desempenho na detecção de movimento. O giroscópio tem um alcance programável de ± 250 , ± 500 , ± 1000 e ± 2000 °/s (dps). Já o acelerómetro pode ser programável na escala ± 2 g, ± 4 g, ± 8 g e ± 16 g. O MPU 6050 possui ainda um filtro programável passa-baixo por forma a reduzir o ruído para o giroscópio, acelerómetro e sensor de temperatura. A sensibilidade inicial calibrada de fábrica nos sensores reduz os requisitos de calibração [85]. Estando os sensores directamente ligados a ADCs de 16 bit, analisando-se a tabela de registos verifica-se que, a cada um dos 3 eixos de cada sensor, estão alocados dois registos de oito bits cada, contendo em conjunto os dados de saída para cada eixo [88]. Os dados são apenas de leitura, sendo necessário concatenar o Least Significant Bit (LSB) e o Most Significant Bit (MSB) de cada eixo de modo a obter os valores medidos por cada sensor.

3.5 Conclusão

O sistema implementado é composto por uma unidade MCU, um conjunto de sensores de força na gáspea e na zona plantar e um IMU.

Tendo em conta os requisitos iniciais do sistema, a unidade de MCU teria de possuir pelo menos 12 canais de ADC, permitir comunicação I²C e possuir um modulo de comunicação BLE.

Selecionou-se para MCU a placa de desenvolvimento ESP32 DevKitC V4, por esta possuir 16 canais de ADC para além de cumprir os restantes requisitos do sistema.

Para sensores de força, foi efetuado um estudo dos existentes no mercado e da gama de medição necessária para a implementação do sistema. Optou-se pela utilização de sensores de força resistivos, FSR.

Quanto ao IMU, apesar de se ter a noção que o MPU 6050 não deverá ser utilizado para novos desenvolvimentos, este estava disponível para implementação, pelo que foi o efetivamente utilizado.

Capítulo 4

Desenvolvimento do Hardware

O sistema idealizado envolve a construção de um prototipo que realize a aquisição de dados, permita monitorizar hiperpressões plantares e correlacionar a incidência destas com o ciclo de marcha humano. Na construção deste dispositivo, existem alguns aspectos a ter em consideração como a dimensão e o posicionamento dos sensores bem como o condicionamento de sinal necessário à aquisição de dados fiáveis, pelo que ao longo deste capítulo serão abordados os passos necessários que levaram a sua consecução.

4.1 Hardware implementado

Ao longo do desenvolvimento do sistema de aquisição, este foi sofrendo algumas alterações e adaptações.

Apesar de se ter definido que os sensores a serem utilizados seriam resistivos, equacionou-se a utilização de sensores FSR convencionais na zona plantar. Contudo, para que o sistema final fosse uma solução robusta, estes foram descartados. Tal decisão prende-se com o facto de as ligações destes sensores aos fios condutores (soldas) facilmente poderem sofrer danos provocados pelo posicionamento do pé durante o movimento. Além do mais, considerou-se que, tendo em conta que se pretende implantar este sistema num sapato para diabéticos, a presença de todos os componentes tem obrigatoriamente de ser o mais discreta possível, por forma a que não representem corpos estranhos que possam potenciar o risco de lesão por si só.

Equacionou-se a utilização de novos tipos de sensores emergentes no mercado e optou-se pela escolha de sensores têxteis da marca SensingTex. Esta empresa apresentou várias soluções de palmilhas, por eles produzidas, cuja localização dos sensores seriam diferentes das definidas para o projeto. Deste modo, por forma a poder-se validar e testar a aplicação destas palmilhas ao sistema de sensorização a implementar, foram adquiridas algumas amostras. Os resultados dos testes realizados com as amostras encontram-se no Capítulo 6.

As palmilhas testadas vêm montadas de fábrica com sensores ligados na forma matricial 4X4. Tendo em conta o produto da SensingTex que estava a ser estudado, passou a estar previsto a utilização de nove sensores, por forma a obter-se uma matriz de sensores 3X3. O objectivo futuro seria pois, adquirir conjuntos de palmilhas e apliques para a gáspea personalizados. Na região plantar, está prevista a colocação de dois sensores na zona das falanges do primeiro e quinto dedos (sensores 1 e 2), três sensores nas cabeças do primeiro, segundo/terceiro e quarto/quinto metatarsos (sensores 3, 4 e 5), um sensor na lateral do mediopé e três sensores localizados na região do calcanhar (sensores 7, 8 e 9). Já para a gáspea seria fabricado um alicate com 4 sensores, no formato de matriz 2X2. Para a região dorsal do pé está prevista a colocação de três sensores na zona das falanges do primeiro, segundo/terceiro e quarto/quinto dedos e um quarto sensor na zona do aperto do sapato. Na Figura 4.1 está representada a localização prevista para os sensores que serão acoplados na palmilha e na gáspea do calçado.



Figura 4.1: Localização dos sensores na região plantar e na gáspea

Prevendo-se que os sensores sejam colocados na região plantar num sistema de matriz, 3 sensores serão alimentados em simultâneo e a leitura da tensão, de cada um, será detetada em 3 canais de ADC diferentes. Por exemplo, os sensores 1,

2 e 3 serão alimentados pelo IN1, o valor do sensor 1 será lido canal analógico Out1, o sensor 2 no canal Out2 e o sensor 3 na saída Out3. Após serem lidos os sensores alimentados por IN1, serão alimentados os sensores ligados a IN2 e lidos nos canais correspondente bem como os alimentados por IN3 serão lidos pelos respectivos ADCs. Na figura 4.2a e 4.2b encontram-se a matriz de alimentação e ligação ao ADC para a palmilha e para a gáspea respectivamente.



Figura 4.2: (a) Matriz 3X3 de alimentação e ligação ao ADC para a palmilha; (b) Matriz 2X2 de alimentação e ligação ao ADC para a gáspea

No sistema implementado, serão necessários 5 General-purpose input/output (GPIO) de alimentação e 5 canais ADC, três para a zona plantar e dois para a gáspea de cada tipo respectivamente, para efetuar a leitura obtida pelo conjunto de 13 sensores de força.

Após vários testes às palmilhas, cujos resultados encontram-se no Capítulo 6, verificou-se que como forma de condicionamento de sinal bastaria a utilização de um divisor de tensão, tal como indicado na Figura 4.3.

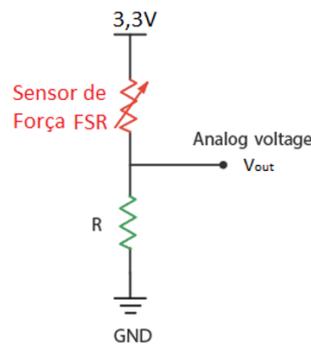


Figura 4.3: Divisor de tensão

Neste caso, a tensão V_{out} será obtida pela equação 4.1.

$$V_{out} = 3,3 \times \frac{R}{R + FSR} \quad (4.1)$$

Por forma a ajustar a sensibilidade dos sensores à zona pretendida de medição, optou-se por utilizar valores de resistências, no divisor de tensão, diferenciadas para a zona plantar e para a gáspea. Para a zona plantar utilizaram-se resistências de 1200Ω e para a zona dorsal resistências de 4300Ω , de forma a ter uma maior sensibilidade, mas com penalização no alcance.

Tal como já foi referido, a solução que se pretende implementar irá necessitar de 5 GPIO de alimentação e 5 GPIO associados a 5 canais diferentes de um conversor ADC. Atendendo à Figura 4.4, que indica o *pinout* da placa de desenvolvimento indicando os periféricos associados a cada GPIO, foram atribuídos para alimentação da palmilha os GPIO 25, 26 e 27 e para alimentação da gáspea os GPIO 18 e 19. Relativamente ao ADC, e após se ter encontrado referências a incompatibilidades entre o ADC2 da placa ESP32 e os módulos Wi-Fi e Bluetooth, optou-se por utilizar apenas canais referentes ao ADC1. Assim, para a zona plantar utilizou-se os GPIO 34, 35 e 32, que correspondem ao ADC1 canal 6, ADC1 canal 7 e ADC1 canal 4 respetivamente, e para a zona da gáspea utilizou-se o GPIO 36 (VP) e o 39 (VN) que correspondem ao ADC1 canal 0 e ADC1 canal 3 respetivamente.

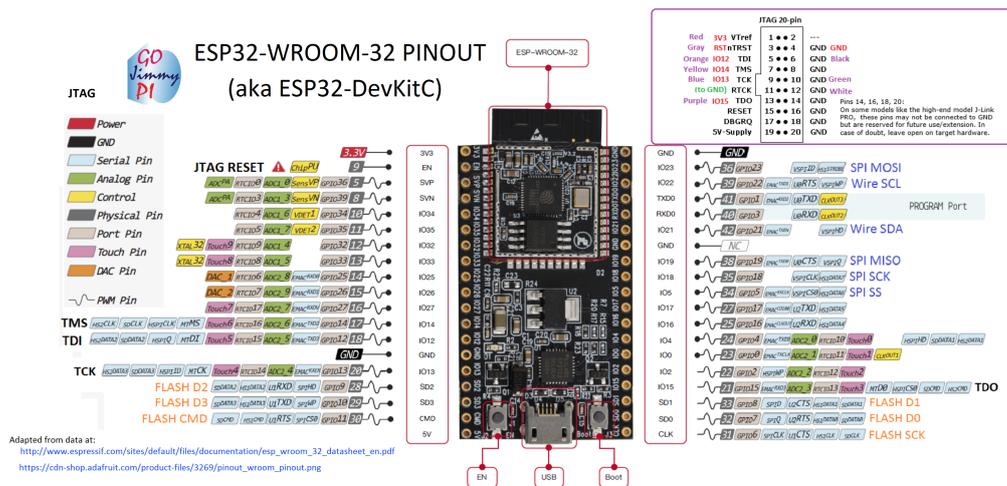


Figura 4.4: Pinout da ESP33 DEVKitC [89]

Relativamente ao MPU 6050 este irá comunicar por I²C com a placa de desenvolvimento, necessitando por isso de 4 ligações: Common Collector Voltage (VCC), Ground (GND), Serial Clock (SCL) e Serial Data (SDA). Assim, ligou-se o VCC do MPU ao pino 3v3 da ESP32 e o GND a um dos pinos referenciados como tal na Figura 4.4. Estabeleceu-se ainda a ligação entre o SCL com o GPIO 22 e o SDA com o GPIO 21. O esquema elétrico dos componentes do sistema encontra-se indicado na Figura 4.5.

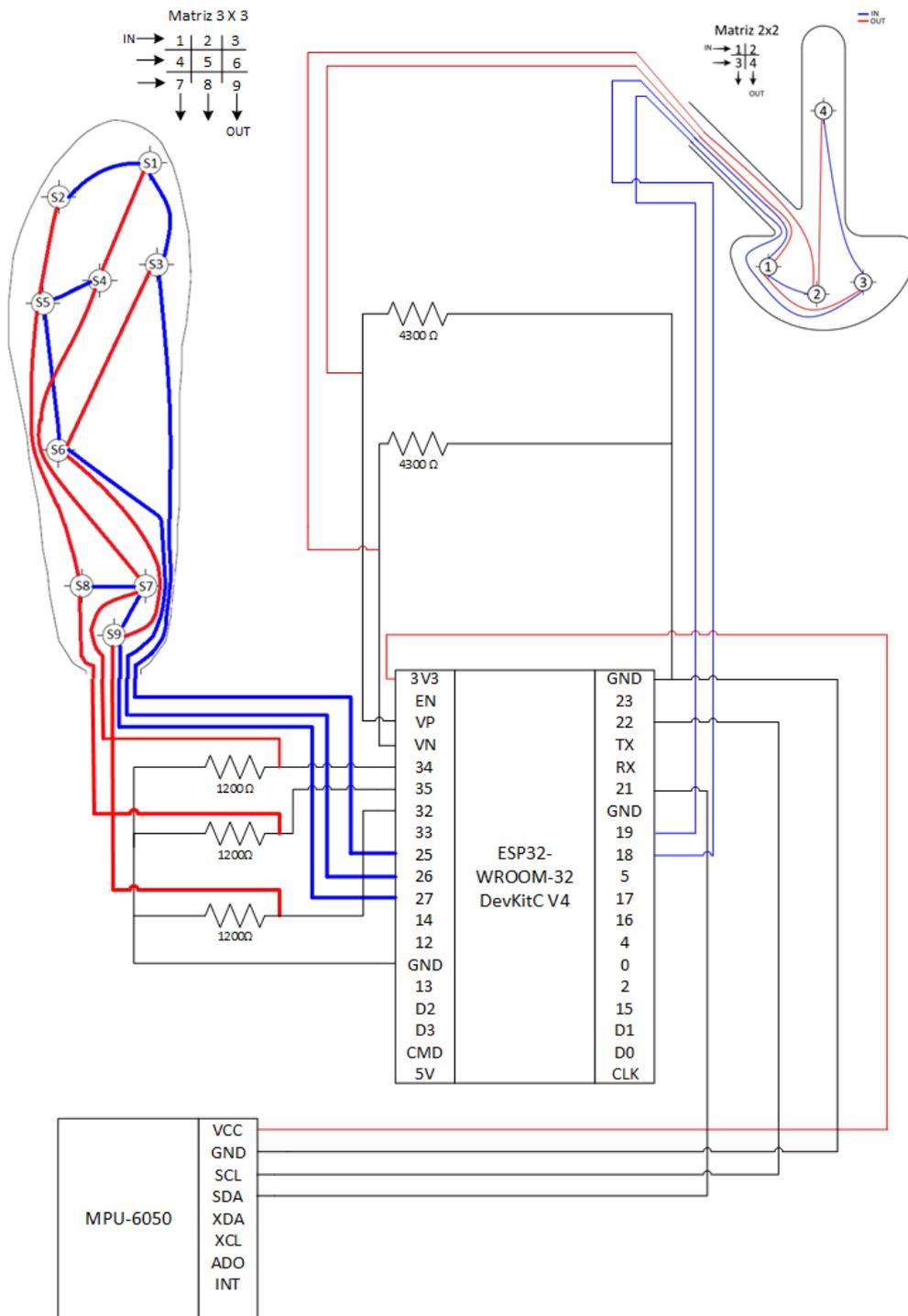


Figura 4.5: Esquema elétrico dos componentes do sistema

No que concerne ao protótipo desenvolvido, este encontra-se na Figura 4.6 e na Figura 4.7 encontra-se o pormenor das ligações efetuadas entre sensores e placa ESP32.

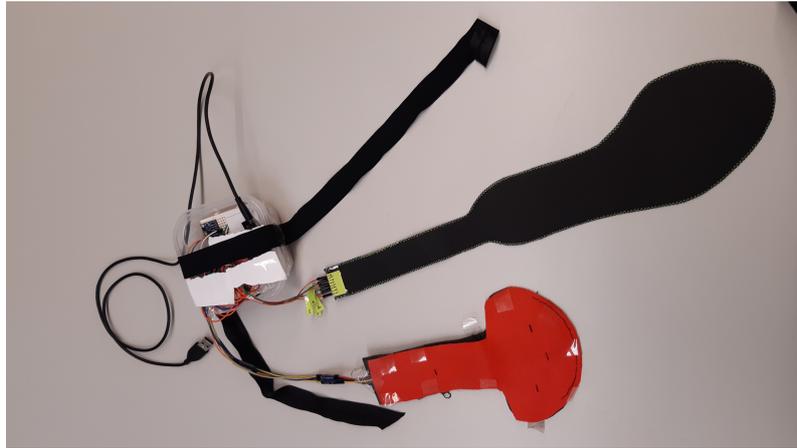


Figura 4.6: Protótipo do sistema de sensorização

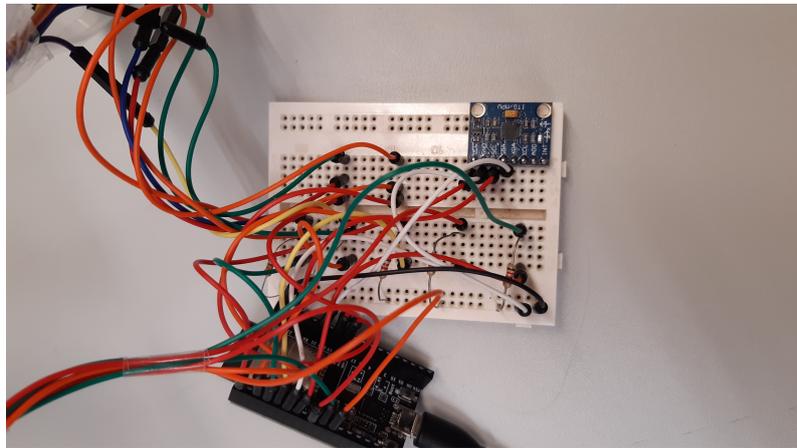


Figura 4.7: Pormenor das ligações efectuadas entre sensores e placa de desenvolvimento

De ressaltar que devido ao facto de, em tempo útil, saber-se que seria impossível a aquisição e utilização do aplico previsto para a zona dorsal, optou-se por nesta fase utilizar-se sensores FSR convencionais da marca Interlink, estando contudo estes montados sob a forma da matriz apresentada na Figura 4.2b. De igual modo optou-se por utilizar uma das palmilhas em matriz 4X4, estando apenas ligados 3 pinos de alimentação e 3 pinos ligados aos canais de ADC já mencionados.

4.2 Conclusão

O prototipo desenvolvido foi sofrendo alterações na sua concepção, ajustando-se às necessidade levantadas, ao material disponível para utilização e a ideias inovadoras que se pretenderam ver implementas. Destas destacam-se a utilização de sensores

têxteis resistivos montados numa palmilha em matriz 3X3. Para a zona dorsal do pé, neste momento está a ser implementado o uso de sensores FSR convencionais.

No que respeita ao IMU utilizou-se o MPU 6050, que comunica via I²C com a placa de desenvolvimento utilizada.

Capítulo 5

Desenvolvimento do Software

O desenvolvimento do sistema de sensorização compreendeu várias etapas, tendo inicialmente sido realizado uma parte de programação do MCU (firmware) e posteriormente a programação de duas pequenas aplicações que permitissem testar o firmware desenvolvido. Ao longo deste capítulo vão ser abordados os passos que permitiram a obtenção da solução final

5.1 Firmware

O kit de desenvolvimento Esp32 pode ser programado com recurso a um software de código aberto, tendo-se optado pela utilização do Arduino IDE pela sua simplicidade e por ter bibliotecas estáveis e compatíveis com a placa selecionada.

O fluxograma implantado na aplicação encontra-se expresso na Figura 5.1. No Anexo A, pode ser encontrado todo o código referente à programação do *firmware*.

Após serem inicializadas algumas variáveis do sistema, foram definidos os UUID do serviço e de três características referentes ao BLE tendo sido estes gerados aleatoriamente (Anexo A.1).

Na função *Setup*, após ser definida o *Baud Rate* a 115200, irá ser inicializado um sub processo, *initImu()*, cujo código pode ser encontrado no Anexo A.2. Este, irá proceder à inicialização do IMU, nomeadamente a definição da sensibilidade do acelerómetro e do giroscópio. Utiliza a biblioteca *Adafruit_MPU6050.h*, que possui funcionalidades já estabelecidas que permitem a comunicação via I²C, acesso aos registos, formas de alterar a sensibilidade dos sensores, entre outras, e a biblioteca

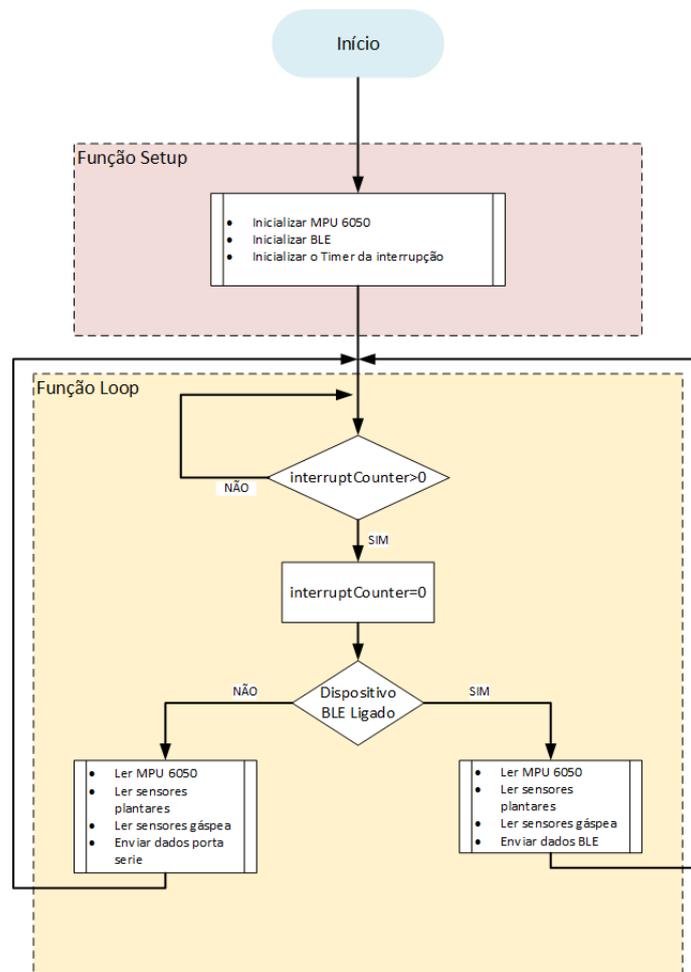


Figura 5.1: Fluxograma da aquisição e envio de dados aplicado no dispositivo

Adafruit_Sensor.h que retorna os valores obtidos em diversos sensores num formato comum.

Seguidamente, será inicializada a comunicação por BLE no sub processo *myBLEInit()*, cujo código poderá ser visualizado no Anexo A.3. Este utiliza as bibliotecas *BLEDevice.h*, *BLEUtils.h*, *BLEServer.h* e *BLE2902.h*, que servem de suporte para a utilização do BLE com a ESP32 em conjunto com o Arduino IDE. Estas bibliotecas irão configurar os parâmetros e protocolos de comunicação entre cliente e servidor, possuem as configurações de todos os *handlers* para gestão de eventos e outras funcionalidades. No *myBLEInit()*, optou-se por alterar o nome do dispositivo para "Pe_Diabetico" de forma a que este fosse facilmente reconhecível. Foram definidas três características para o mesmo serviço de BLE, uma afeta aos dados dos sensores plantares, outra afeta aos dados dos sensores da gáspea e a última aos dados do MPU 6050, de acordo com o descrito na secção 3.2.1.

Recorreu-se a uma interrupção como mecanismo para garantir o envio de dados a

uma frequência constante, 10 Hz. Para tal, definiu-se uma variável a *interruptCounter* que irá ser ativada com o nível lógico 1, cada vez que é gerada uma interrupção pelo *handler* (a cada 100000 pulsos) (Anexo A.4). Visto que o oscilador da ESP32 funciona a 40MHz definiu-se, na função *setup* (Anexo A.5), o *prescaler* para 40, recorrendo à função *timerBegin*, de modo a obter-se um sinal com uma frequência de 1 MHz e como tal facilitar o algoritmo de divisão de relógio. Seguidamente, gerou-se uma interrupção a cada 0,1 s, ou seja a cada 100000 flancos ascendentes do relógio, recorrendo à função *timerAlarmWrite*. Assim que se atinge os 100000 flancos ascendentes, irá ser desencadeada a função *IRAM_ATTR onTimer ()*. Como função *timerAlarmWrite* encontra-se com o modo *autoreload* ativo, será realizado um *reset* ao *Timer* do alarme cada vez que este atinge o valor estipulado. A Figura 5.2 indica o fluxograma da interrupção que incrementa a *flag interruptCounter*.

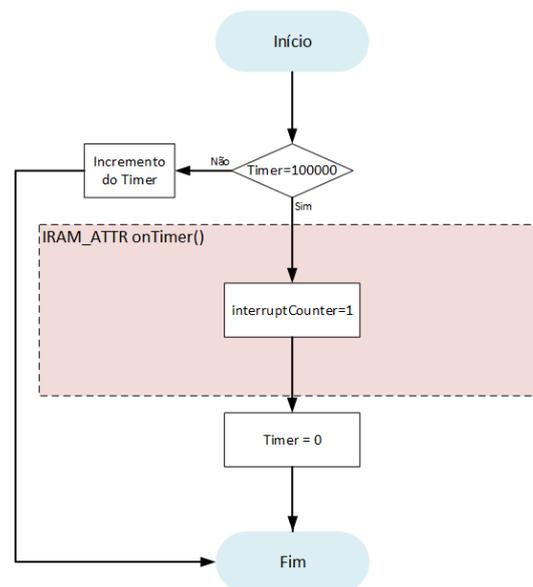


Figura 5.2: Fluxograma da interrupção

Relativamente à função *Loop* (Anexo A.6), esta inicia-se com o teste à *flag interruptCounter*. Se a variável *interruptCounter* tiver sido ativada pelo ISR, ou seja se for superior a "0" será processado o código do *handling* da interrupção e o valor dessa variável inativado (colocado a zero). Como já referido, a cada 100000 flancos ascendentes do relógio, será chamada a função *IRAM_ATTR onTimer ()* que irá incrementar novamente o *interruptCounter* garantindo que a função *loop* seja executada a cada 0,1 s.

Seguidamente será verificado se o dispositivo está devidamente ligado por BLE a outro dispositivo (*flag deviceConnected = True*) ou não (*flag deviceConnected = False*) (Anexo A.3).

Caso a transmissão seja efetuada por BLE, os dados adquiridos pelos sensores são

tratados de forma a serem enviados por esta via para outro dispositivo. Assim, serão chamadas funções que permitem efetuar a leitura dos sensores, *readImu()* (Anexo A.7), *readSensorPlantar()* (Anexo A.8) e a *readSensorGaspea()* (Anexo A.9).

A leitura dos sensores do IMU é executada pela função *readImu()* (Anexo A.7), que irá utilizar funcionalidades definidas nas bibliotecas *Adafruit_MPU6050.h* e *Adafruit_Sensor.h*. Assim foram definidas variáveis do tipo *sensors_event_t* para que o método *mpu.getEvent* possa armazenar valores obtidos nos sensores. De referir que, apesar do MPU 6050 ter incorporado um sensor de temperatura, optou-se por não efetuar a leitura do mesmo, visto ser um dado não relevante para a aplicação pretendida.

De forma a minimizar a quantidade de dados enviados, as leituras dos sensores do IMU foram convertidas para números inteiros e positivos e armazenados em variáveis de 16bits. Ou seja, tomando como exemplo a coordenada do acelerómetro segundo o eixo os *xx* (*aceX*) o valor lido pelo sensor (a. acceleration .x) será primeiramente multiplicado por 1000, para que o seu valor seja inteiro e ao resultado será adicionado 10000, para garantir que seja positivo, tal como descrito na equação 5.1. A mesma equação aplicou-se a todos os dados recebidos do IMU.

$$aceX = (a.acceleration.x \times 1000) + 10000 \quad (5.1)$$

A leitura dos sensores plantares é executada pela função *readSensorPlantar()* (Anexo A.8), que irá efetuar a leitura, em matriz, dos sensores plantares. Para tal, é inicialmente ativado o GPIO correspondente, de forma a injectar 3.3V na entrada comum dos três sensores. A saída de cada um destes 3 sensores estará ligada a 1 canal de ADC e a leitura efectuada em cada um deles será armazenada numa posição de um *array 2D* de 9 entradas de 16 bit. Seguidamente, a alimentação dos primeiros 3 sensores será cortada e passará a ser alimentada a entrada seguinte do conjunto de outros 3 sensores, repetindo-se este ciclo até terem sido lidos os nove sensores plantares, de acordo com o expresso na Figura 4.5.

Cada valor lido pelo ADC será então mapeado de modo a converter o valor digital de 12 bit para uma escala cujo máximo é de 3300 mV, V_{out} . Aplicando a equação 4.1 e tendo em conta que a resistência R no divisor de tensão tem o valor 1200 Ω , o valor da resistência do sensor, FSR, será obtida pela equação 5.2.

$$FSR = \frac{3300 - V_{out}}{V_{out}} \times 1200 \quad (5.2)$$

Para os sensores, existe uma relação linear entre a condutância e a força aplicada, estando o valor da condutância expresso em $\mu\Omega^{-1}$ na reta de calibração, pelo que o valor da condutância do sensor, G_{FSR} é obtida pela equação 5.3.

$$G_{FSR} = \frac{1000000}{FSR} \quad (5.3)$$

Tendo em conta a reta de calibração dos sensores plantares, neste caso $G_{FSR} = 360F_{FSR} - 2210$, será calculado o valor correspondente da força aplicada F_{FSR} tal como expresso na equação 5.4.

$$F_{FSR} = \frac{G_{FSR} + 2210}{360} \quad (5.4)$$

Porém esse valor, F_{FSR} , será multiplicado por 100 de modo a passar de decimal para inteiro, tal como descrito na equação 5.5.

$$F_{sensor} = F_{FSR} \times 100 \quad (5.5)$$

A Figura 5.3 mostra o fluxograma da função de leitura dos sensores plantares. Nesta os GPIOs que injectam os 3.3V nas entradas dos conjuntos de sensores na posição i são os já mencionados na secção 4.1 e correspondem, para a palmilha, aos GPIO 25, 26 e 27. Já os GPIOs que lêem as saídas dos sensores na posição j correspondem aos os GPIO 34, 35 e 32 respectivamente.

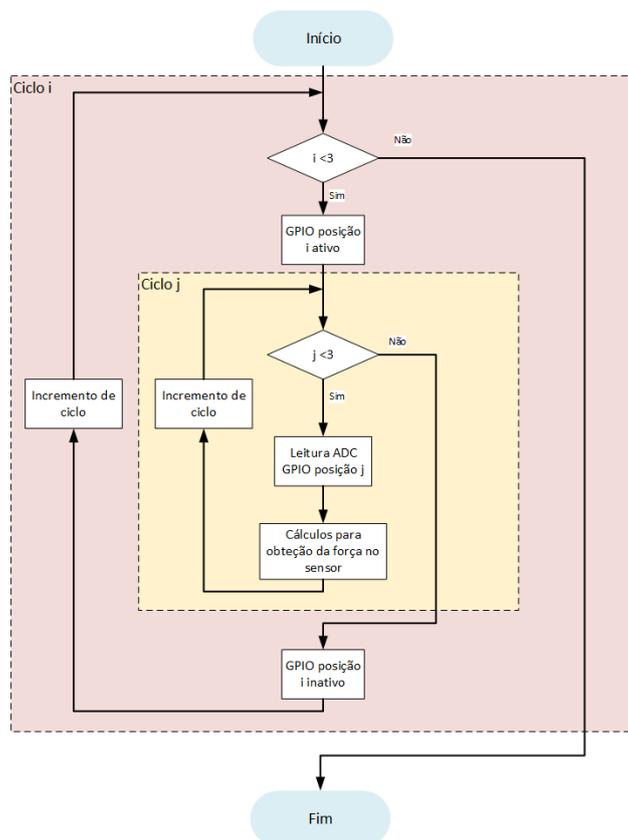


Figura 5.3: Fluxograma da função de leitura dos sensores plantares

No que concerne aos sensores da gáspea, estes são lidos recorrendo à função `readSensorGaspea()` (Anexo A.9) que é em tudo semelhante à `readSensorPlantar()`,

existindo apenas um menor número de sensores a ser lido (4 sensores).

Contudo, como a resistência R aplicada no divisor de tensão é de 4300Ω , a equação que permite o cálculo do valor da resistência do sensor da gáspea será dada pela equação 5.6. A equação 5.3 é também aplicada a estes sensores, mas como a reta de calibração dos sensores da gáspea é diferente, o valor da força aplicada será obtida pela equação 5.7.

$$FSR = \frac{3300 - V_{out}}{V_{out}} \times 4300 \quad (5.6)$$

$$F_{FSR} = \frac{G_{FSR} + 150}{90} \quad (5.7)$$

Também esse valor, F_{FSR} , será multiplicado por 100 de modo a passar de decimal para inteiro, tal como descrito na equação 5.5.

A Figura 5.4 mostra o fluxograma da função de leitura dos sensores da gáspea. Neste caso os GPIOs que injectam os 3.3 V nas entradas dos conjuntos de sensores na posição i são os mencionados na secção 4.1 e correspondem, para a gáspea, aos GPIO 18 e 19. Já os GPIOs que lêem as saídas dos sensores na posição j correspondem aos GPIO 36 e 39 respectivamente.

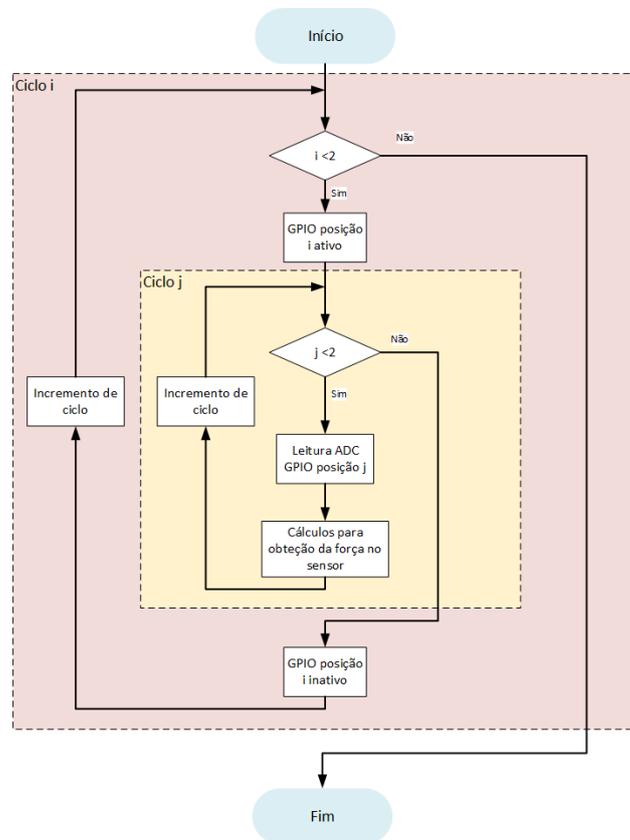


Figura 5.4: Fluxograma da função de leitura sensores gáspea

Após todos os sensores terem sido lidos, os seus valores terão que ser enviados por BLE. Este protocolo está limitado a 20 Bytes por transmissão [74] e como tal foi necessário reduzir o volume de dados a transmitir. Recorre-se, então, à função *sendBLEData()* (Anexo A.10), que irá partir os valores armazenados, todos eles de 16 bits, em dois conjuntos de 8 bits. Estes bits serão concatenados num *array* de Bytes cuja dimensão irá depender do número de sensores lidos. Assim, aos sensores do IMU irá corresponder um *array* de 12 Bytes (12 conjuntos de 8 bits), aos sensores plantares irá corresponder um *array* de 18 Bytes (18 conjuntos de 8 bits) e aos sensores da gáspea um *array* de 8 Bytes (8 conjuntos de 8 bits). Cada um dos 3 *arrays* de bytes será então atribuído ao *value* da respectiva característica do serviço BLE, procedendo-se finalmente ao seu envio através da propriedade *Notify*. Esta, como já foi referido na secção 3.2.1, trata-se de uma operação assíncrona de notificação, inicializada pelo servidor que não necessita de um *acknowledgement* por parte do cliente, indicando que recebeu a informação. A Figura 5.5 mostra o fluxograma da função *sendBLEData()*.

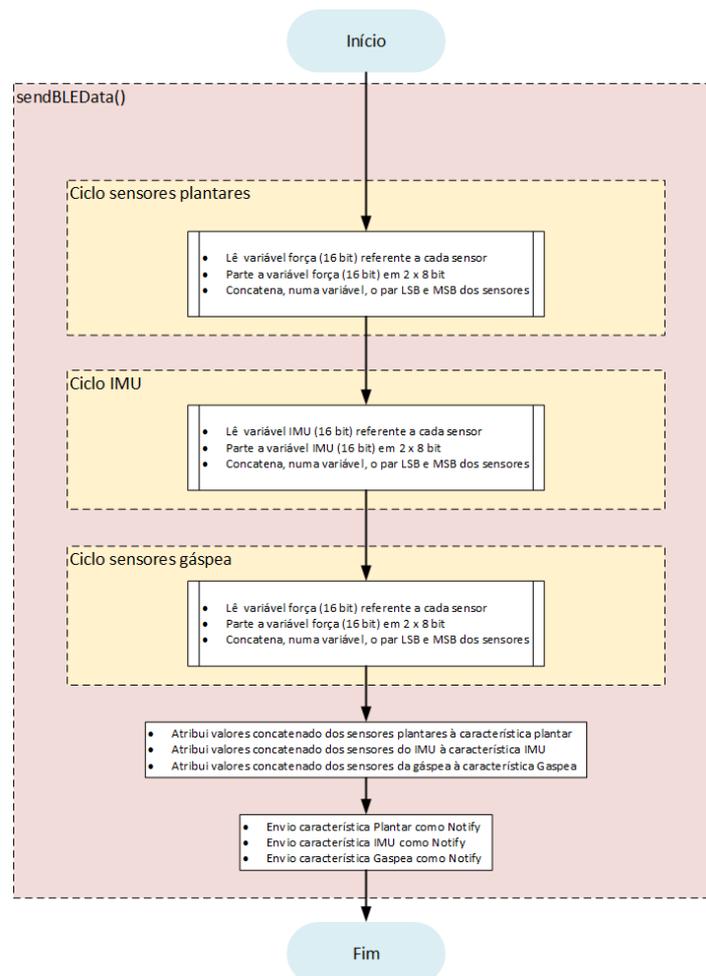


Figura 5.5: Fluxograma da função de envio de dados BLE

A Figura 5.6 indica a trama de dados a ser enviada em cada característica do BLE. Tal como pode ser observado, a cada 0,1 s serão enviados 38 Bytes de dados, correspondentes a dados dos sensores, e tendo em conta que a taxa de transferência de dados por BLE é de 1 Mbps, este envio teoricamente demorará 0,3 ms, valor muito abaixo do ciclo de aquisição de dados (0,1 s). De salientar que o tempo total de envio será muito superior visto que em cada transmissão, para além dos dados dos sensores, são enviados cabeçalhos, UUID do serviço e da característica, entre outras informações.

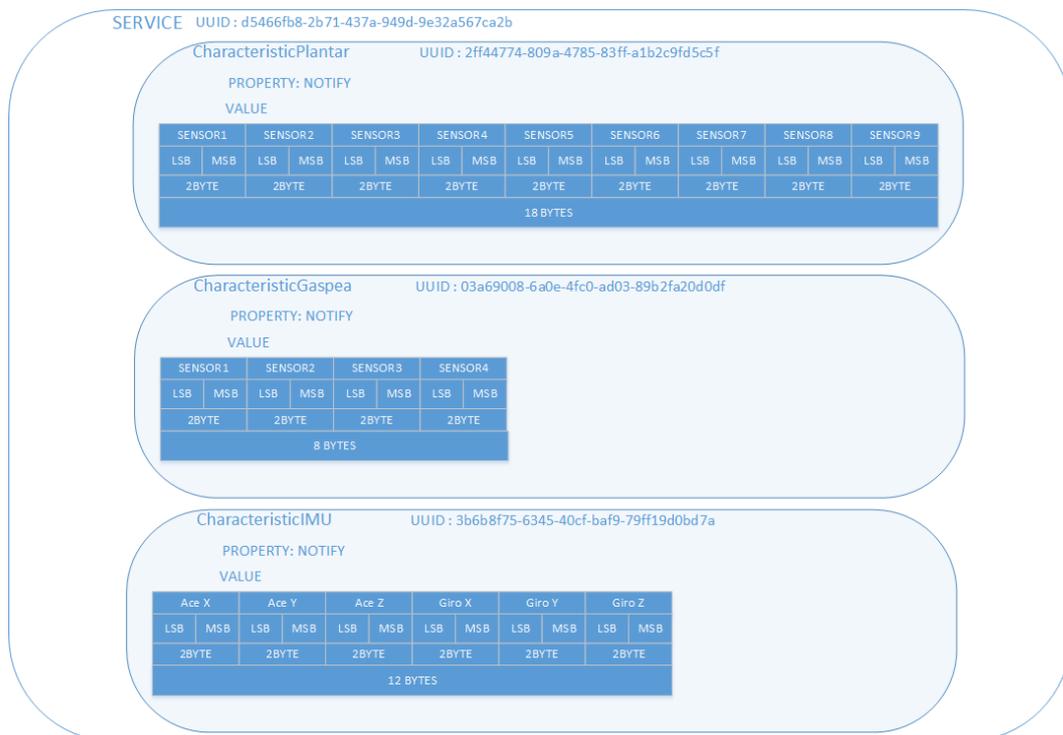


Figura 5.6: Características BLE com trama de dados

Caso a transmissão se efetue por porta serie, estando a *flag deviceConnected* a *False*, os dados serão adquiridos aos sensores através das funções *readImu()*, *readSensorPlantar()* e *readSensorGaspea()* que irão funcionar de igual modo ao já descrito.

Contudo os dados dos sensores armazenados nas variáveis serão concatenados a uma letra maiúscula (de A a S) e enviados como sequência de caracteres (Anexo A.6). Os caracteres de "A" a "D" antecedem os dados dos sensores 1 a 4 da gáspea. Os caracteres de "E" a "M" antecedem os dados dos sensores 1 a 9 plantares. Os caracteres "N" a "S" irão corresponder aos valores dos acelerómetro (x, y, z) seguidos dos valores do giroscópio (x, y, z). A Figura 5.7 indica a trama de dados a ser enviada pela porta serie. No máximo, a cada dado referente a um sensor irá corresponder 6 Bytes (máximo de 6 caracteres enviados). Tendo em conta que o *Baud Rate* definido

é de 115200 bits por segundo, irá corresponder a um tempo de envio teórico de 7,9 ms, valor muito inferior ao ciclo de aquisição de dados definido 0,1 s.

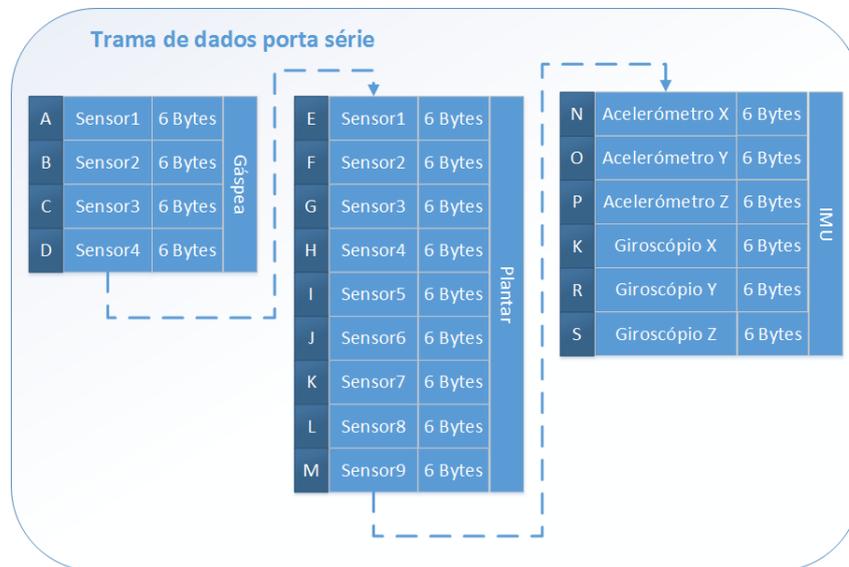


Figura 5.7: Trama de dados enviada para a porta serie

5.2 Software

Por forma a facilitar a análise de dados recolhidos pelos sensores, bem como realizar testes ao *firmware* desenvolvido, foram criadas duas aplicações, uma em Visual Basic que permite a aquisição de dados através da porta série, e outra recorrendo ao MiT App Inventor que permite a recepção de dados num *smartphone*. Nas subsecções seguintes ir-se-á realizar a descrição destas aplicações bem como das suas principais funcionalidades.

5.2.1 App Visual Basic

A aplicação realizada em Visual Basic recebe os dados dos sensores enviados pela porta serie, mostra-os em tempo real, regista o valor máximo atingido por cada sensor e permite visualizar a variação da força ao longo do tempo nos gráficos referentes a cada sensor plantar ou da gáspea, bem como um gráfico da soma das forças dos sensores plantares. Além destas funcionalidades, permite registar os dados adquiridos numa tabela e caso se pretenda, esses dados poderão ser guardados num ficheiro excel. Os gráficos poderão ser apagados a qualquer altura, bem como a tabela com os dados recolhidos. O aspecto gráfico da aplicação encontra-se apresentado na Figura 5.8.

Para se iniciar a aplicação, a placa Esp32 deverá estar ligada por USB ao computador onde se encontra esta aplicação, pressiona-se o botão "*Scan Port*" que irá

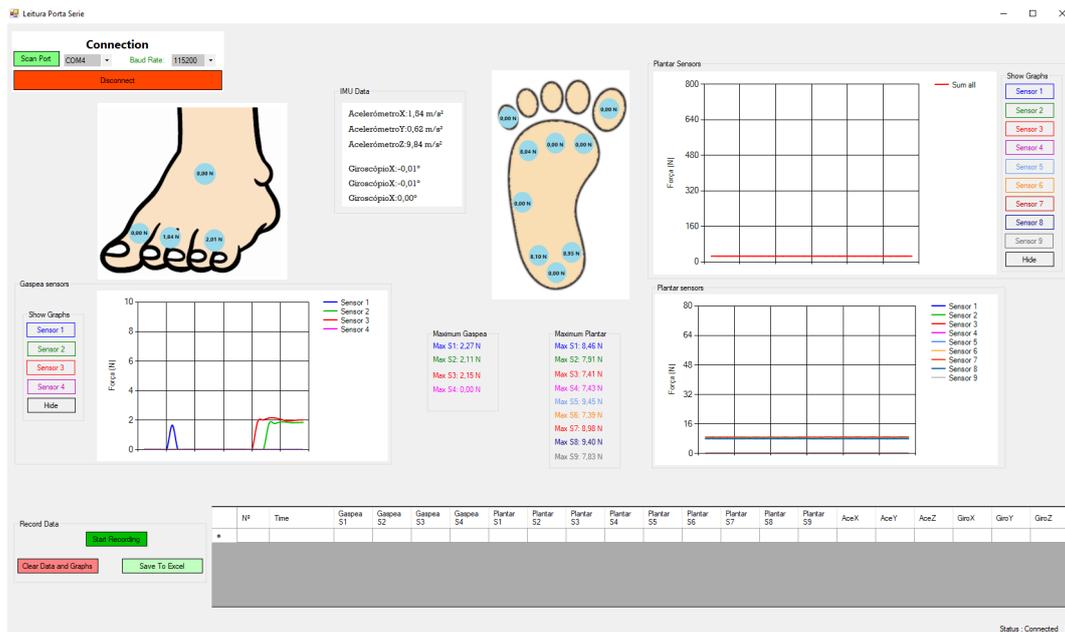


Figura 5.8: Aspecto gráfico da Aplicação Visual Basic

detectar qual a porta serie (COM) a que a placa estará ligada. O *Baud Rate* está definido a 115200, pelo que pressionando no botão *Connect*, a aplicação irá confirmar a validade da ligação. Caso não detecte nenhum erro estabelece comunicação com a placa e os dados adquiridos irão começar a ser mostrados. Os gráficos, por defeito, mostram os dados das forças aplicadas em todos os sensores, em simultâneo, no mesmo gráfico. Caso se pretenda analisar algum sensor em particular, poderá ser pressionado o botão correspondente a esse sensor e, irá ser desenhado o gráfico das forças exercidas neste sensor, em conjunto com o gráfico das forças aplicadas em todos os sensores em simultâneo, mas em tamanho mais reduzido.

Pressionando o botão *"Start Recording"*, os dados de todos os sensores passam a ser mostrados na tabela, com o respetivo *"Timestamp"*. Pressionando em *"Stop Recording"*, botão que surge aquando do início do preenchimento da tabela, esta parará de ser preenchida, dando ao utilizador a possibilidade de guardar os resultados pressionando *"Save to Excel"*. Em qualquer momento o utilizador pode pressionar em *"Clean Data and Graphs"* fazendo um *reset* a todos os dados recebidos.

No Anexo B, pode ser encontrado todo o código referente à App em Visual Basic e o fluxograma principal encontra-se na Figura 5.9.

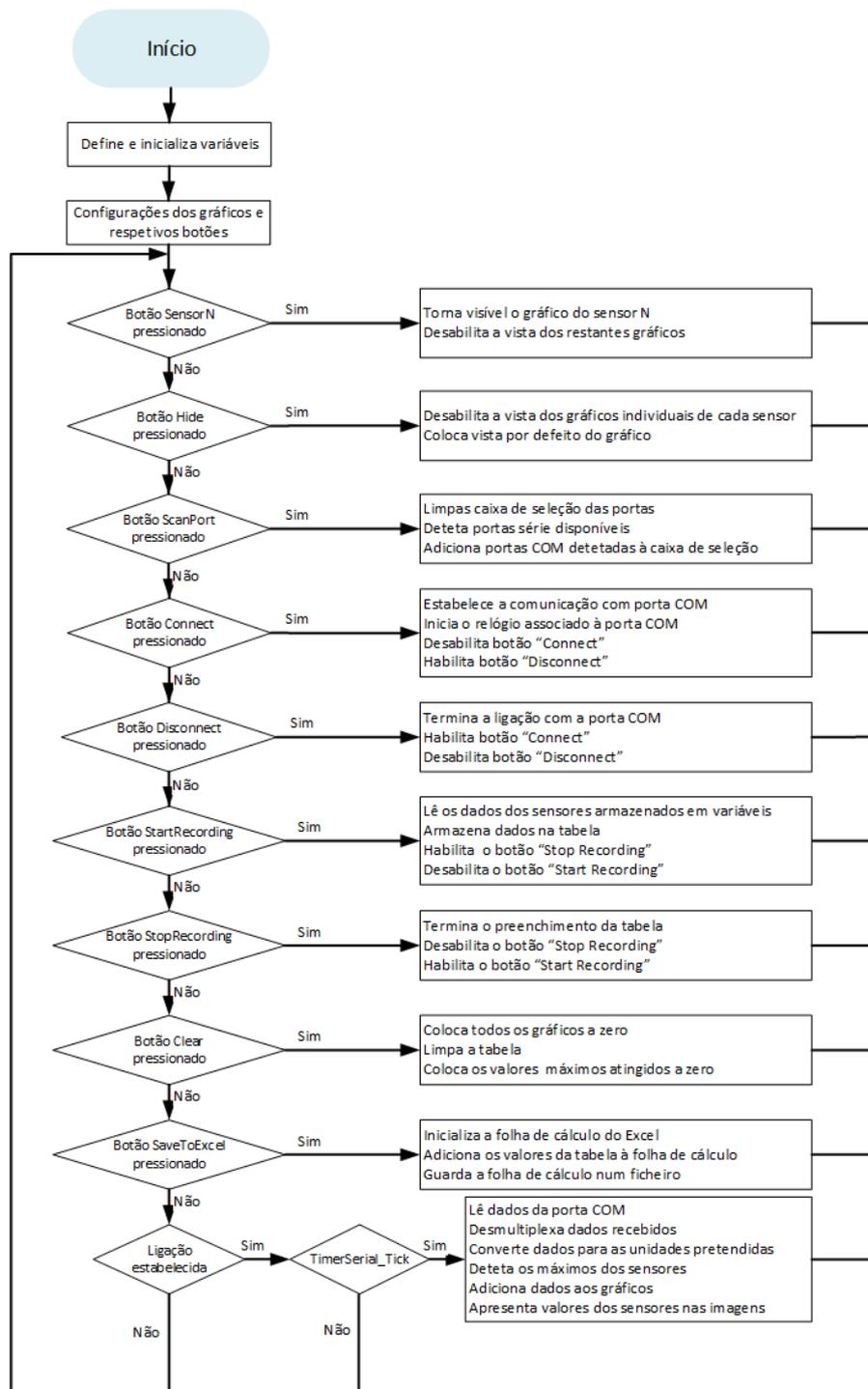


Figura 5.9: Fluxograma geral da App em Visual Basic

Assim que a aplicação é iniciada, são definidas várias variáveis bem como os botões laterais dos gráficos que irão permitir que, quando pressionados, se altere os gráficos visualizados. O código referente a estas definições encontram-se nos Anexos B.1 e B.2.

Seguidamente, são configurados os gráficos e respetivas dimensões, cujo código encontra-se no Anexo B.3.

Após terem sido definidas estas configurações iniciais, são configuradas todas as funcionalidades para estabelecer a ligação entre o computador e a placa ESP32 pela porta COM (Anexo B.4). Quando o botão *ButtonScanPort* é pressionado, desencadeia uma série de funcionalidades que irão permitir apresentar, numa *ComboBox*, todas as portas séries detetadas. Após selecionada a porta COM onde a placa ESP32 está ligada, e tendo a *ComboBoxBaudRate*, por defeito, a apresentar o taxa de 115200, poderá ser acionado o botão *Connect*. Deste modo, quando o *ButtonConnect* é pressionado, irá inicializar a ligação à porta serie selecionada, com o respetivo *Baud Rate*, e ativar o temporizador adicionado a esta porta. Neste caso, o botão *Disconnect* passará a estar visível. Caso se tente pressionar o botão *Connect*, tendo um *Baud Rate* ou porta serie inválidos, será gerada uma mensagem de erro. No caso do botão *Disconnect* ser pressionado, irá ser terminada a ligação à porta serie, bem como o respetivo temporizador. Este passará a não estar visível e voltará a surgir o botão *Connect*.

De referir ainda que é necessário definir o relógio associado à porta serie neste software, com um intervalo igual ao da interrupção definida no *firmware*, para evitar erros de comunicação entre a placa e a App.

Estando a ligação estabelecida, a cada 100 ms (valor definido no temporizador da porta serie) será iniciada a leitura dos dados que estão a ser recebidos, com a função *TimerSerial_Tick*, cujo código encontra-se no Anexo B.6. Os dados recebidos serão colocados numa tabela, seguidamente analisados linha a linha e atribuídos ao respetivo sensor.

Como referido na secção 5.1, os dados dos sensores enviados pela porta serie são antecidos por uma letra maiúscula. Assim, será necessário neste programa desmultiplexar os dados e efetuar os cálculos necessários de modo a que possam ser obtidos os valores referentes a cada um dos sensores, nas unidades pretendidas. Ou seja, por exemplo o valor do sensor 1 da gáspea virá concatenado com a letra A, pelo que primeiramente é necessário obter apenas o valor numérico e posteriormente dividi-lo por 100 de modo a obter o valor expresso em Newton.

Assim, tendo em conta a trama de dados presente na Figura 5.7, será analisado o primeiro caractere de cada linha, o dado recebido será colocado na variável correspondente a cada sensor, sendo removido o caractere identificador e restando apenas o valor numérico. Seguidamente, serão executados os cálculos necessários por forma a guardar este valor numa variável, com a unidade pretendida. A Figura B.1, que se encontra no Anexo B.6 evidencia o modo como foi realizado a desmultiplexagem dos dados recebidos. Os dados referentes a cada sensor da gáspea ou plantar serão então comparados com o valor máximo já atingido nesse sensor e, caso seja obtido um novo máximo, será apresentado esse novo valor. Posteriormente, cada dado dos

sensores plantares e da gáspea será adicionado ao respectivo gráfico e passará a estar também visível sobre a imagem indicativa da localização do sensor. No caso dos dados referentes ao IMU, estes serão apenas apresentados numericamente na aplicação. Novamente, o programa voltará a estar à espera da recepção de dados repetindo todo este procedimento.

Quando é pressionado o botão *Start Recording*, cujo código encontra-se no Anexo B.5, este irá tornar visível o botão *Stop Recording*, tornar inativo o botão *Save to Excel* e ativar a função *TimerDataLogRecord*. Esta função irá permitir o preenchimento de uma tabela, com o respetivo *Timestamp*, sendo lidas as variáveis armazenadas referentes aos dados dos sensores e passando estas a estarem visíveis para o utilizador. O código referente ao preenchimento da tabela através da função *TimerDataLogRecord_Tick*, pode ser encontrado no Anexo B.5.1.

Pressionado o Botão *Stop Recording*, este botão passará a estar inativo, o botão *Start Recording* passará a estar novamente ativo assim como o botão *Save to Excel* e a função *TimerDataLogRecord* irá terminar de ser executada.

Caso o utilizador assim pretenda, poderá pressionar o botão *Save to Excel*, cujo código se encontra no Anexo B.5.2. Este irá efetuar procedimentos por forma a gerar e guardar um ficheiro excel com os dados constantes na tabela. Para tal, irão ser chamadas funcionalidade do excel de modo a ler os dados constantes na tabela e coloca-los nas mesmas posições numa folha de excel. O ficheiro gerado será automaticamente guardado na pasta onde foi lançada a aplicação, com o nome "*dia_hora_minuto.xlsx*" correspondente ao instante em que este ficheiro é criado. Será então exibida uma mensagem informativa indicando que o ficheiro foi guardado com sucesso.

A qualquer momento, poderá ser pressionado o botão *ButtonClear*, cujo código poderá ser encontrado no Anexo B.5. Este botão irá colocar a zero todos os valores representados nos gráficos assim como os valores máximos atingidos por cada sensor, para além de apagar a tabela.

5.2.2 App telemóvel

Um dos requisitos deste sistema é que os dados dos sensores deverão ser enviados, por BLE, para uma App de telemóvel, App essa que ainda não está disponível para testes com o hardware já implementado. Face à necessidade de testar esta funcionalidade num *smartphone*, foi elaborada uma pequena aplicação para esse efeito. Esta aplicação foi realizada com recurso ao Mit App Inventor. Apesar, de se ter a noção que o Mit App Inventor possui inúmeras limitações, o objetivo desta aplicação será apenas testar a transmissão BLE. Deste modo, esta aplicação recebe os múltiplos dados dos sensores, que são enviados em *raw data*, tratar-os esses dados de forma a atribuir um valor decimal a cada sensor e apresenta-os na aplicação.

Assim poderemos ter a noção do funcionamento do BLE e de erros que possam surgir.

Após o início da aplicação, pressionando o Botão "Scan", surgirá uma nova vista que mostrará todos os dispositivos BLE dentro do alcance do telemóvel. A aplicação só permite a ligação com o dispositivo "Pe_Diabetico" que aparecerá na listagem, tal como se pode visualizar na Figura 5.10.



Figura 5.10: Janela de seleção do dispositivo BLE

De referir que o dispositivo deverá ser previamente emparelhado com o *smartphone* de modo a permitir a ligação com a aplicação. Assim que esteja estabelecida a ligação, o botão *Connected*, irá passar a verde e o *Disconnect* a vermelho. Deste modo, inicialmente irá aparecer no ecrã a vista indicada na Figura 5.11a, onde as imagens apresentadas servirão apenas como forma de legenda da localização dos sensores. Os dados recebidos em cada instante serão mostrados nos locais assinalados por Sensor1, Sensor2, ..., Sensor9. Pressionando o botão *Gáspea e IMU* passará a ser mostrado no *Smartphone* a vista indicada na Figura 5.11b, que neste caso em específico o Sensor1 está com a leitura S1: 1.8 N e para os restantes o valor 0 N. Os

botões *Força* e *Pressão kPa* irão permitir apresentar os valores em Newton (N) ou em kiloPascal (kPa).



Figura 5.11: Vista da aplicação para *Smartphone* (a) inicial; (b) Gáspea e IMU

Um dos cuidados a ter na aplicação é que o relógio definido na aplicação para a recepção de novos eventos terá de ser igual ao tempo de cada interrupção gerada no hardware, de modo a sincronizar os dados.

No Anexo C, pode ser encontrado o programa em forma de diagrama de blocos referente à App realizada no Mit App Inventor e a Figura 5.12, indica o fluxograma da aplicação para *smartphone*.

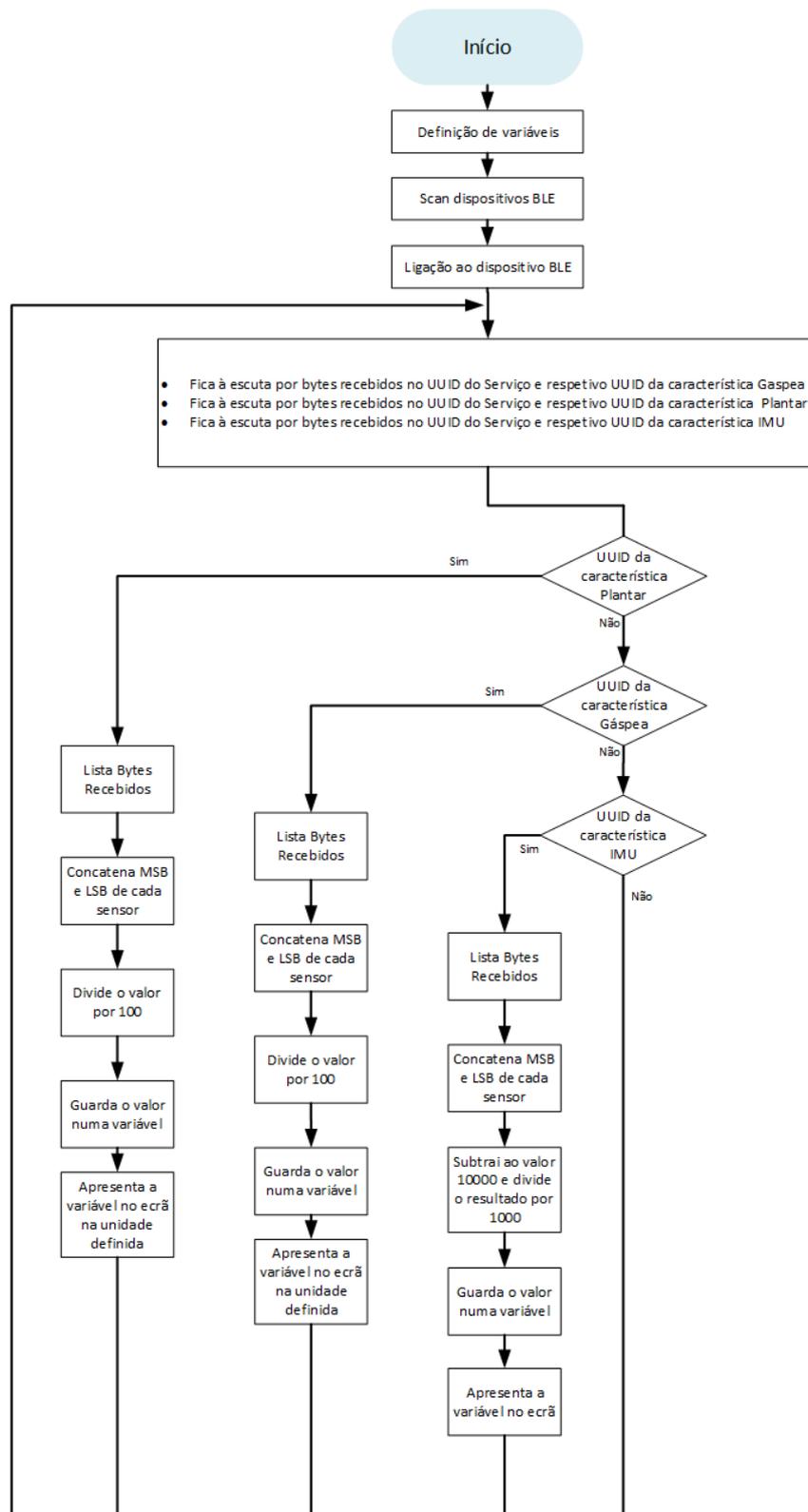


Figura 5.12: Fluxograma da App para *smartphone*

Assim, após serem inicializadas as várias variáveis globais (Anexo C.1) a aplicação é inicializada (*Bloco When Screen1. Initialize*) e esta passa a realizar um *scan* de dispositivos BLE, apresentando-os numa lista (*Bloco When Bluetooth.DeviceFound*). Após a seleção do dispositivo "Pe_Diabetico" (*Bloco When ListPicker1. AfterPicking*) estabelece-se então a ligação entre o telemóvel e a Esp32. Os Blocos que possibilitam estes passos encontram-se no Anexo C.2.

Seguidamente, após se ter definido o tempo do relógio e caso o dispositivo esteja ligado por BLE, é necessário a indicação que o sistema está preparado e à escuta de dados por parte de um serviço que tem associado um UUID, e respectivas características (*Bloco When Clock1.Timer*). Nesta aplicação, têm-se um único serviço com três características diferentes associadas: Plantar, Gaspea e IMU. O diagrama de blocos que executa esta função encontra-se no Anexo C.3.

Como já foi referido, os dados enviados pela ESP32 vêm num array de Bytes. Para desmultiplexar e tratar os dados recebidos foi criado um procedimento o *decodeShort2* (Anexo C.4) que será chamado aquando da recepção de bytes em qualquer uma das características. Cada vez que detecta a recepção de Bytes (*Bloco When Bluetooth1.BytesReceived*) estes serão filtrados pelo UUID da sua característica e colocados na lista referente a cada grupo de sensores. Se os dados estiverem associados ao UUID Plantar, cada byte será colocado na lista *ListView1* (Anexo C.5.1), se o UUID corresponder à Gáspea os dados serão colocados na *ListView2* (Anexo C.5.2) e caso o UUID seja o do IMU serão colocados na lista *ListView3* (Anexo C.5.3).

Tendo em conta que os dados enviados via BLE encontram-se em *raw data* e tendo sido necessário partir os 16 bits correspondentes a cada dado adquirido pelos sensores em conjuntos de 8+8 bits, estes terão de ser tratados e concatenados pela App de acordo com a trama expressa na Figura 5.6. A função *decodeShort2* lê os bits menos significativos (LSB) e bits mais significativos (MSB) referentes a cada sensor, concatena-os e dependendo da característica, trata-os e armazena-os na variável correspondente .

Caso o UUID seja referente à característica Plantar ou à Gaspea o valor obtido para cada sensor será dividido por 100 e armazenado numa variável (Anexo C.5.1 e C.5.2). Caso o UUID seja referente à característica IMU, ao valor obtido para cada sensor será subtraído o valor 10000 e o resultado dividido por 1000, sendo então armazenado numa variável (Anexo C.5.3).

Os valores armazenados nas variáveis poderão então ser mostrados no ecrã (Anexo C.5.4). Como existe a possibilidade mostrar os valores dos sensores plantares e da gáspea em Newton ou em kiloPascal, para finalizar, procedeu-se aos cálculos necessários para tal (Anexo C.5.4).

De referir ainda que, a alternância da vista de valores de sensores plantares e valores dos sensores da gáspea e IMU é efetuada a partir da programação de botões

que, quando accionados, tornam visíveis ou escondem determinados componentes. A programação desses botões encontra-se no Anexo C.4.

5.3 Conclusão

Ao longo deste capítulo foram exploradas as soluções de software implementadas no corrente projeto. Relativamente ao *firmware*, o maior desafio terá sido talvez as limitações de envios de dados por parte do BLE. Apesar de existirem dados na literatura em que já é possível enviar mais do que 20 Bytes por cada valor de característica, após alguns testes verificou-se que tal não estava a ser conseguido com a Esp32. O facto de esta placa possuir apenas Bluetooth versão 4.2, um dos protocolos iniciais de BLE, poderá ter sido o motivo deste condicionamento.

Apesar de, face aos objetivos do trabalho, ser de extrema importância a programação da placa de desenvolvimento, ou seja a programação do *firmware*, foi necessário ir um pouco mais além, e arranjar forma de verificar e armazenar os dados obtidos.

Assim foi inicialmente pensado o desenvolvimento de uma pequena aplicação em Visual Basic, que recebesse dados pela porta serie e via BLE. No entanto, após já estar implementada a aplicação com a respectiva aquisição via porta serie, verificou-se que o Visual Basic não funciona nativamente com BLE. Mais, existem alguns problemas de compatibilidade entre BLE e Windows visto as bibliotecas Windows para BLE não estarem suficientemente desenvolvidas. Uma solução possível passaria por refazer a aplicação em Windows Presentation Foundation (WPF) pelo que se optou por manter a aplicação em Visual Basic e construir uma outra em MIT App Inventor. Porém, esta última possui limitações e serve apenas para testar os dados enviados.

Capítulo 6

Análise dos Resultados

Por forma a obter-se resultados fidedignos a partir do sistema desenvolvido, inicialmente teve-se, obrigatoriamente, de proceder a diversos testes de calibração. Estes, para além de servirem para testar o sistema e ajustar o condicionamento do sinal, ajudaram a seleccionar os sensores e serem utilizados. Após todo o sistema estar devidamente acondicionado no interior de um sapato, foram realizados testes cujos resultados serão analisados também neste capítulo.

6.1 Testes e Calibração de Sensores

Com os testes efetuados pretendeu-se aferir a viabilidade de várias tecnologias existentes no mercado, para sensorização de pressões. Deste modo, procedeu-se inicialmente à medição da variação da resistência de vários sensores de cada tecnologia, quando sujeitos a aumentos progressivos de força aplicada na área ativa dos mesmos.

Inicialmente, realizou-se uma medição direta da variação da resistência do sensor em função da força aplicada, recorrendo a um multímetro digital. Foi colocada uma pequena peça, com um diâmetro semelhante ao da área ativa dos sensores e sobre esta, colocadas massas calibradas. Dependendo do sensor, o incremento sucessivo de massa aplicada foi de 100 ou de 200 g.

Seguidamente, procedeu-se de igual modo mas com os sensores ligados a um kit de desenvolvimento ESP32 onde os dados foram adquiridos recorrendo a um divisor de tensão para condicionamento do sinal. Foram realizados testes com diferentes valores de resistência no divisor de tensão, por forma a seleccionar a mais indicada

para a medição de pressão plantar e para a medição da pressão na zona da gáspea. Realizou-se, então, ensaios para três valores de R diferentes, 10000, 4300 e 1200 Ω .

Nas subsecções seguintes serão abordados os resultados obtidos em cada teste realizado.

6.1.1 Sensores FSR

Os sensores FSR utilizados foram os Interlink FSR UX 402 que têm um alcance máximo de 20 N, um diâmetro de 14,68 mm (zona ativa) e uma área de 169,2 mm² (1,692 cm²). De modo a inferir qual a resistência do divisor de tensão que melhor se ajustava às necessidades de valores de força detetadas na zona plantar do pé, realizaram-se ensaios para três valores diferentes de resistência R no divisor de tensão e para dois sensores distintos.

Tal como se pode observar nas Figuras 6.1, 6.2 e 6.3, independentemente do valor da resistência utilizada, mantém-se a linearidade entre a condutância e força aplicada, alterando-se apenas a equação da reta que melhor se ajusta aos pontos experimentais.

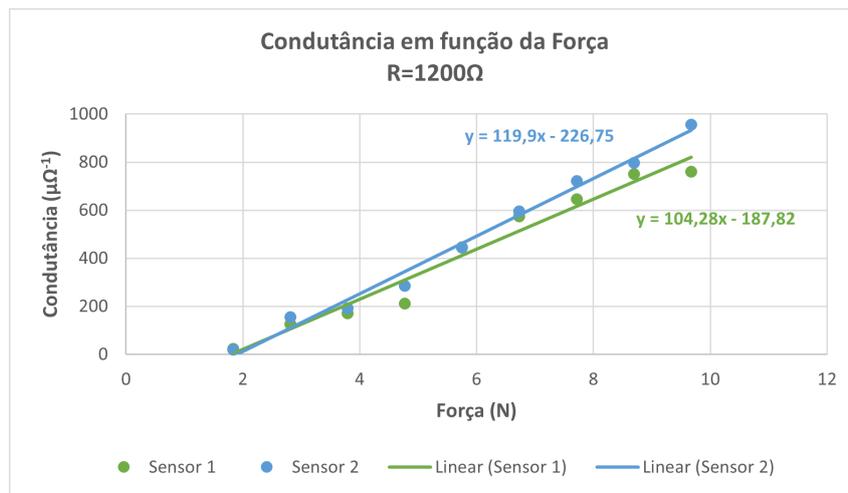


Figura 6.1: Gráfico Condutância em função da Força para R= 1200 Ω para FSR convencionais

Por outro lado, analisando os dados medidos no ADC de 12 bits e o valor de tensão Vout correspondente verifica-se que, quanto maior for a força exercida, maior será o valor Vout e este último terá um valor tanto menor quanto menor for a resistência utilizada. Estes resultados podem ser observados na Tabela 6.1. Assim verifica-se que, para os mesmos sensores, a utilização de resistências maiores levam a que se tenha uma maior sensibilidade mas menor alcance, visto atingir-se o máximo de leitura do ADC de 12 bits (máximo de 4095) com uma menor força aplicada. Por outro lado, a utilização de valores de resistência menores levam a que a gama de medição dos sensores seja maior, mas com pouca sensibilidade.

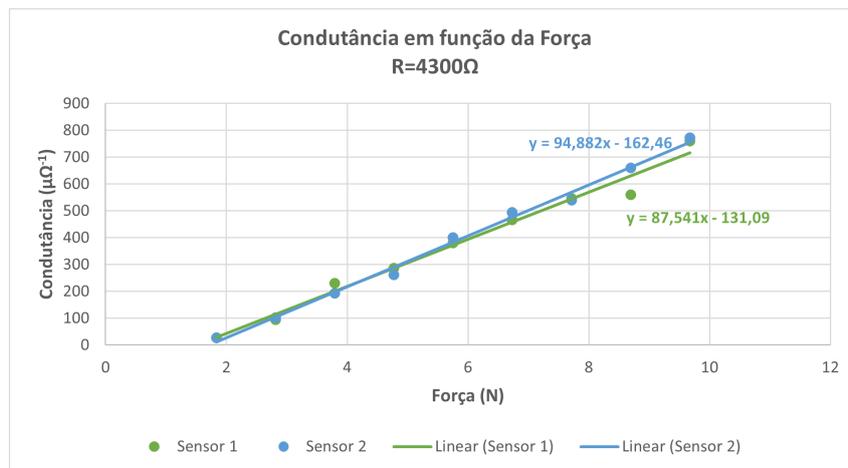


Figura 6.2: Gráfico Condutância em função da Força para R= 4300Ω para FSR convencionais

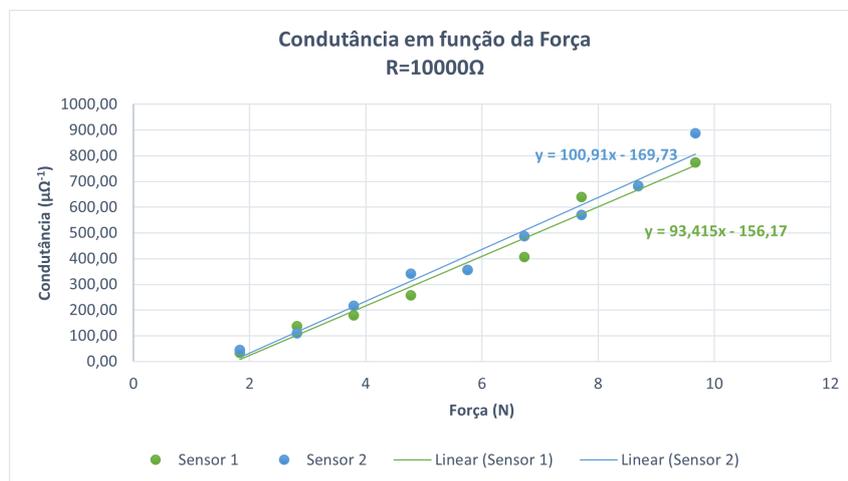


Figura 6.3: Gráfico Condutância em função da Força para R= 10000 Ω para FSR convencionais

Tabela 6.1: Variação do valor detetado no ADC e do valor da tensão com a resistência

Resistência no divisor de tensão (Ω)	Valor máximo registado no ADC	Valor máximo de tensão (mV)
10000	3699	2981
4300	3151	2539
1200	2192	1766

Considerando que a pressão máxima exercida no sensor foi de cerca de 63 kPa e que para a sensorização plantar ir-se-á detetar pressões superiores a 200 kPa, considera-se necessário que a resistência no divisor de tensão seja na ordem dos 1000 Ω, de modo a otimizar o condicionamento do sinal.

6.1.2 Sensores Piezofabric

Relativamente aos sensores do tipo PiezoFabric testados, estes são sensores piezo-resistivos têxteis. Todos possuem o mesmo material sensitivo, contudo variam no material base e/ou dimensões da área sensitiva. Na subsecção seguinte são abordados os resultados obtidos para as duas amostras de palmilha.

Palmilha Azul

As palmilhas azuis da SensingTex (Figura 6.4) possuem um material exterior em Polyethylene terephthalate (PET) e zonas sensíveis do material designado por Piezo Fabric Q8, com um diâmetro de 1 cm, e como tal com uma área sensível de 0,7853 cm². Segundo as especificações do fabricante, este tipo de sensor tem um alcance de 25 kgf/cm².

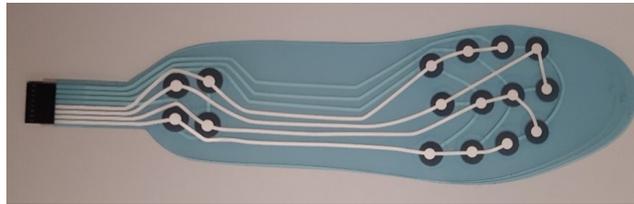


Figura 6.4: Palmilha Azul SensingTex

Foram realizados testes para três sensores desta palmilha e para cada um foram obtidos valores de condutância em função da força, utilizando um multímetro (Figura 6.5) e utilizando o ADC da ESP 32 (Figura 6.6).

Tendo em conta os resultados e conclusões obtido com os sensores FSR convencionais para as medições efetuadas com o ESP32, optou por utilizar como divisor de tensão uma resistência de 1200 Ω .

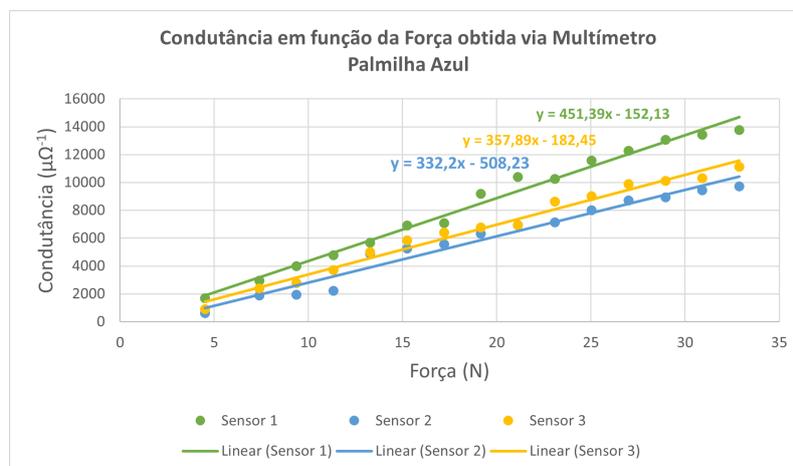


Figura 6.5: Gráfico Condutância em função da Força obtido via multímetro para a palmilha azul

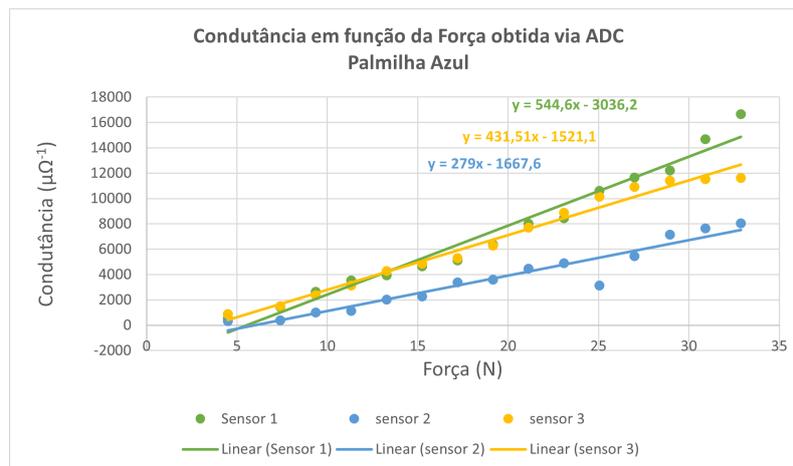


Figura 6.6: Gráfico Condutância em função da Força para $R= 1200 \Omega$ para a palmilha azul

Para estes sensores, verificou-se que existe uma relação linear entre a força aplicada e a condutância do sensor, pelo que um aumento da força aplicada traduz-se num aumento da condutância do sensor, independentemente da utilização do multímetro ou de hardware com condicionamento de sinal. Por outro lado, comparando os gráficos da Figura 6.5 e Figura 6.6 verifica-se que o sensor 1 (representado a verde) é o que possui uma constante de proporcionalidade maior (maior declive da reta) e o sensor 2 (representado a azul) o que possui uma constante de proporcionalidade menor, na relação entre a força aplicada e a condutância do sensor, seja na situação de medição com o multímetro seja na situação com condicionamento do sinal. Para o sensor 1 representado a verde, verifica-se que a partir de forças aplicadas superiores a 30 N, o sinal gerado já não se encontra numa zona de linearidade da condutância em função da força aplicada, pelo que, caso se verifique que é necessário a deteção de forças superiores a esse valor, poderá ser necessário ajustar o condicionamento do sinal.

Palmilha Verde

As palmilhas verdes da SensingTex (Figura 6.7) possuem um material exterior em PET e zonas sensíveis do material designado por Piezo Fabric Q8, com um diâmetro de 0,6 cm, e como tal com uma área sensível de área de $0,283 \text{ cm}^2$. Segundo as especificações do fabricante, este tipo de sensor tem um alcance de 25 kgf/cm^2 .

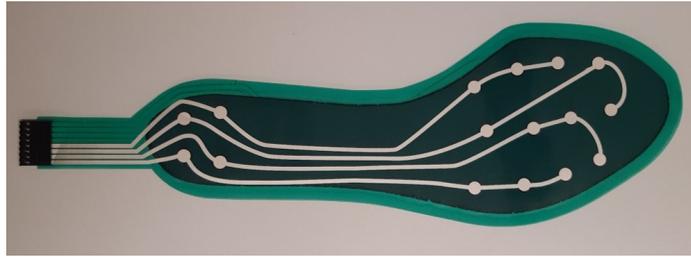


Figura 6.7: Palmilha verde SensingTex

Foram realizados testes para três sensores diferentes. Para cada um foram obtidos valores de condutância em função da força utilizando um multímetro (Figura 6.8) e utilizando o ADC integrado no ESP32 e um divisor de tensão com $R=1200\ \Omega$ (Figura 6.9). Tal como o verificado para a palmilha azul, em todos os ensaios realizados existe uma relação linear entre a força aplicada e a condutância do sensor, pelo que um aumento da força aplicada traduz-se num aumento da condutância do sensor, independentemente da utilização do multímetro ou de hardware com condicionamento de sinal.

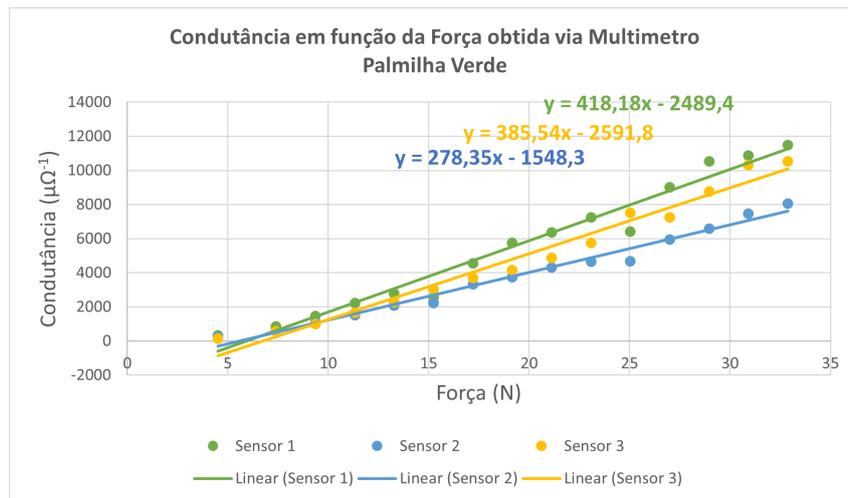


Figura 6.8: Gráfico Condutância em função da Força obtido via multímetro para a palmilha verde

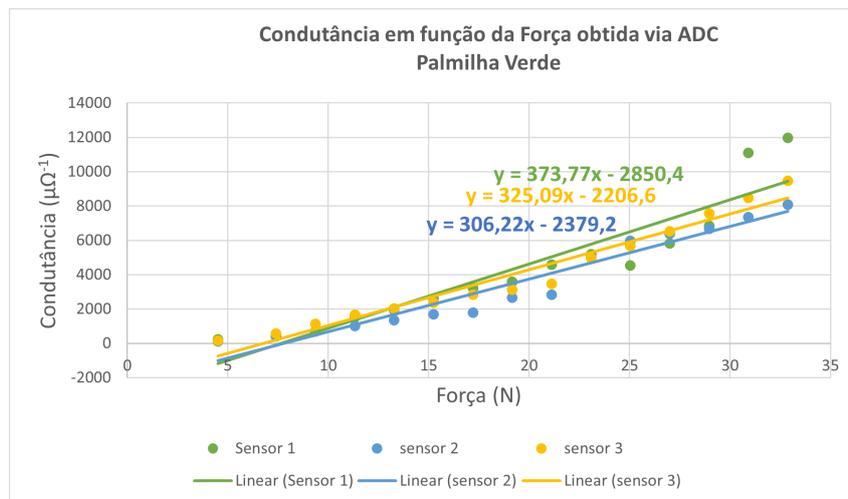


Figura 6.9: Gráfico Condutância em função da Força para $R= 1200 \Omega$ para a palmilha verde

Também nestas palmilhas verifica-se que o sensor cuja reta de calibração possui maior declive é o sensor 1, independentemente de ser obtido por medição direta com o multímetro, ou por medição via ADC com o respetivo condicionamento de sinal. De referir que o sensor 1 começa a sair da zona de linearidade, para medições via ADC, quando a força aplicada é de 25 N enquanto os sensores 2 e 3 permanecem na zona de linearidade.

Tem que se salientar que a força aplicada em cada um destes sensores foi numa área superior à área real dos mesmos. Na realidade, a força foi aplicada numa área próxima de $0,8 \text{ cm}^2$, semelhante à das palmilhas azuis, existindo por isso um erro sistemático na medição da resistência destes sensores.

6.1.3 Sensores PiezoInk

A matriz branca da SensingTex (Figura 6.10) possui um material exterior em Thermoplastic polyurethane (TPU) e zonas sensíveis em PiezoInk, com um diâmetro de 1 cm, e como tal com uma área sensível de área de $0,7853 \text{ cm}^2$.

Foram realizados testes para três sensores desta matriz. Para cada um obtiveram-se valores de condutância em função da força utilizando um multímetro (Figura 6.11) e utilizando o ADC do ESP32.

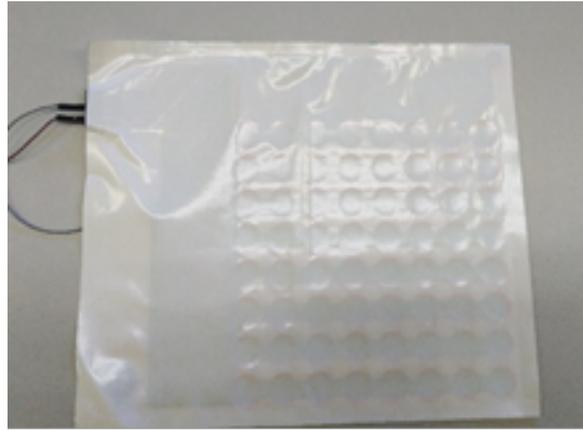


Figura 6.10: Matriz branca SensingTex

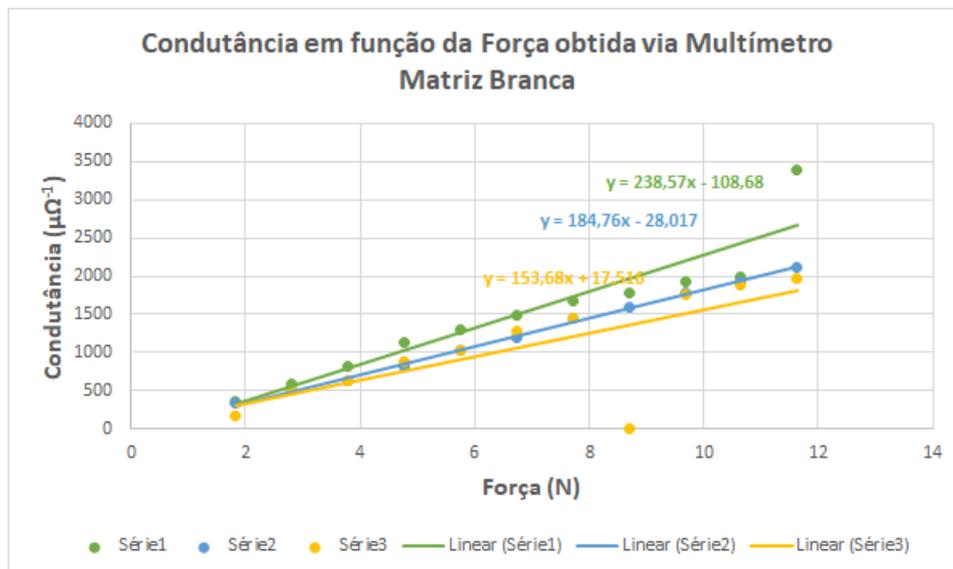


Figura 6.11: Gráfico Condutância em função da Força obtido via Multímetro para a matriz branca

Contudo, tendo em conta que este tipo de sensor tem uma maior sensibilidade, e como tal deverá ser mais adequado à utilização na zona da gáspea, zona onde são espectáveis valores inferiores de pressão, foi realizado um estudo sobre qual a resistência a utilizar no divisor de tensão. Tal como se pode observar na Figura 6.12 e na Figura 6.13, independentemente do valor da resistência utilizada, mantém-se a linearidade entre a condutância e a força aplicada, alterando-se apenas a equação da reta que melhor se ajusta aos pontos experimentais. Já a Figura 6.14, que corresponde a uma resistência de 10 kΩ no divisor de tensão, verifica-se que só existe linearidade até cerca de 7 N de força aplicada. Contudo, para este valor de resistência a sensibilidade do sensor é de tal modo elevada que permite detectar incrementos de 0,01 N na força aplicada.

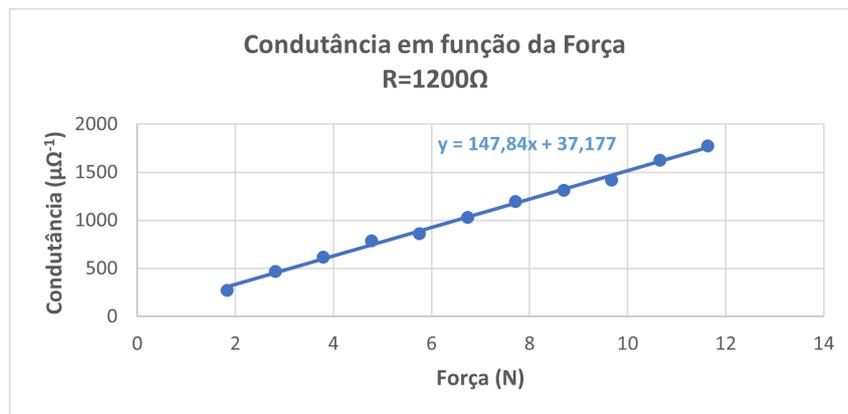


Figura 6.12: Gráfico Condutância em função da Força para R=1200 Ω para a matriz branca

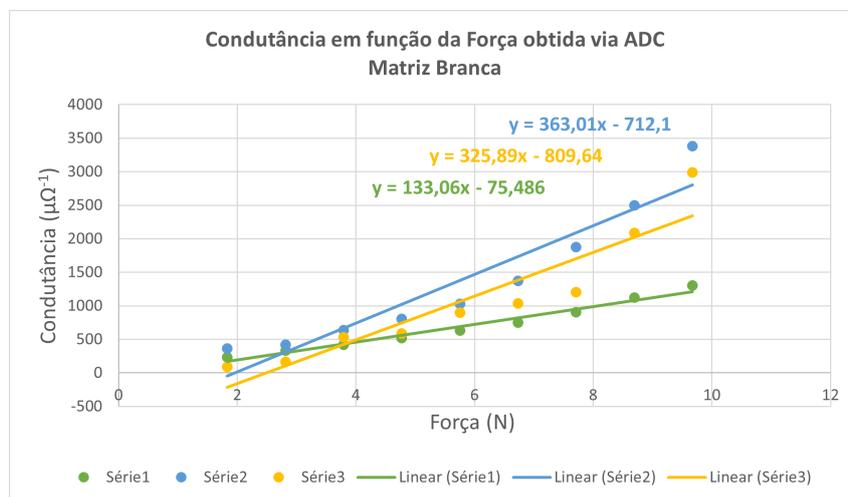


Figura 6.13: Gráfico Condutância em função da Força para R=4300 Ω para a matriz branca

Por outro lado, tal como já descrito para os sensores FSR, analisando os dados medidos no ADC de 12 bits e o valor de tensão V_{out} correspondente verifica-se que, quanto maior for a força exercida, maior será o valor V_{out} e este último terá um valor tanto menor quanto menor for a resistência utilizada. Estes resultados podem ser observados na Tabela 6.2. Assim verifica-se que, para os mesmos sensores, a utilização de resistências maiores levam a que se tenha uma maior sensibilidade mas menor alcance, visto atingir-se o máximo de leitura do ADC de 12 bits (máximo de 4095) com uma menor força aplicada. Por outro lado, a utilização de valores de resistência menores levam a que a gama de medição dos sensores seja maior, mas com pouca sensibilidade.

Atendendo a que ainda estão por definir quais os valores de pressão que se prevê medir na zona da gáspea, considerou-se que, seria mais adequado a utilização de um divisor de tensão com R=4300 Ω modo a otimizar o condicionamento do sinal.



Figura 6.14: Gráfico Condutância em função da Força para R=10000 Ω para a matriz branca

Tabela 6.2: Variação do valor detetado no ADC e do valor da tensão com a resistência para a matriz branca

Resistência no divisor de tensão (Ω)	Valor máximo registado no ADC	Valor máximo de tensão (mV)
10000	3984	3211
4300	3621	2918
1200	2787	2246

6.2 Testes com palmilha e Gáspea sensorizadas

Afim de verificar o comportamento de todo o sistema, foram realizados testes, com gáspea e palmilha sensorizadas. Para a palmilha, foi utilizado um modelo semelhante ao da palmilha verde, Figura 6.7, revestida por um têxtil e com conectores maiores, tendo sido esta última característica o factor que condicionou a sua utilização. Como esta palmilha é constituída por numa matriz 4X4 e como tal possui um total de 16 sensores, foi adaptada para uma matriz 3X3 não se ligando todos os pinos da ficha de ligação. Para a gáspea, foram utilizados sensores FSR (Interlink FSR UX 402). De modo a acomodar convenientemente os sensores na gáspea, foi elaborado um aplique, Figura 6.15, com um formato adaptado à zona onde se pretende colocar e com um local onde se consegue inserir cada um dos sensores.

Por forma a dar-se início aos testes, todo o sistema foi montado num sapato adequado ao pé diabético tal como mostra a Figura 6.16.

Após devidamente calçado (Figura 6.17), permaneceu-se em repouso durante alguns segundos e registou-se o valor medido por cada um dos sensores em situação estática. Seguidamente, foram registados os valores medidos por cada um dos sensores numa deslocação num percurso de 20 metros e, aproximadamente, durante 20 segundos. Foram contabilizados 19 passos durante o percurso.

Da aquisição de dados, foi possível notar que, em repouso os valores obtidos

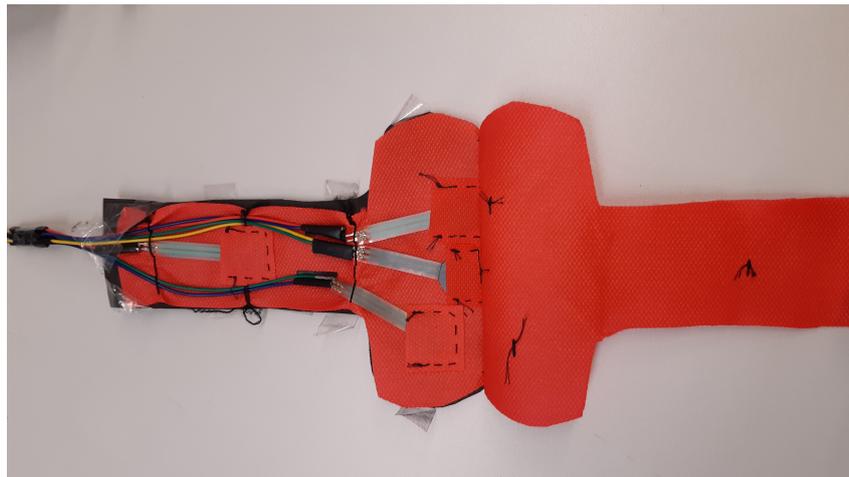


Figura 6.15: Aplique da gáspea com os sensores aplicados



Figura 6.16: Sapato para diabéticos com palmilha de sensores e aplique na gáspea

na zona da gáspea eram nulos para a zona frontal (sensores 1, 2 e 3) e no sensor 4, localizado na zona aperto do cordão, rondavam o valor de 1,9 N. Já na zona plantar, os sensores onde foram registados forças com intensidades superiores foram os localizados no calcanhar (11,6 a 14,3 N).

A partir dos valores registados durante a caminhada obtiveram-se os gráficos evidenciados nas Figura 6.18, Figura 6.19 e Figura 6.20. Pela análise dos mesmos, verificou-se que, tal como previsto, existiam 19 máximos relativos à soma das forças plantares, correspondendo cada máximo a um passo contabilizado durante a aquisição dos dados. É possível ainda verificar que existe um padrão relativamente à variação da soma das forças aplicadas na gáspea ao longo do tempo e que também este possui sensivelmente 19 máximos.



Figura 6.17: Execução dos testes com os sensores

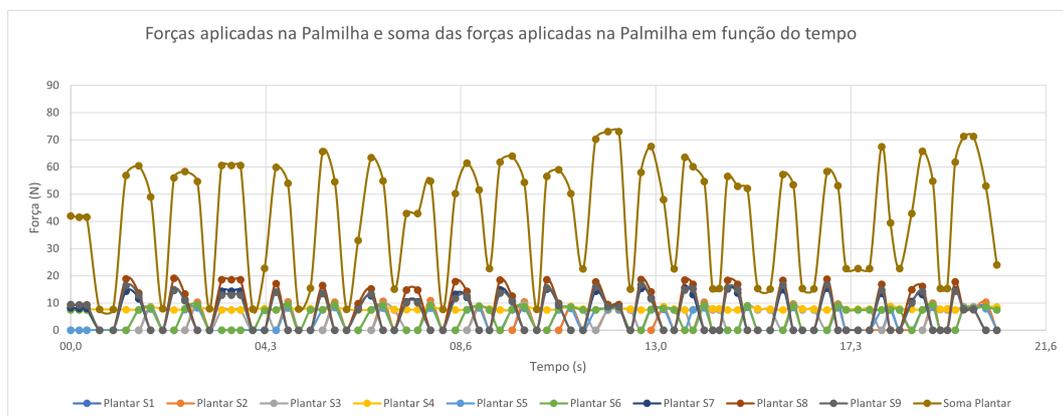


Figura 6.18: Gráfico de forças plantares

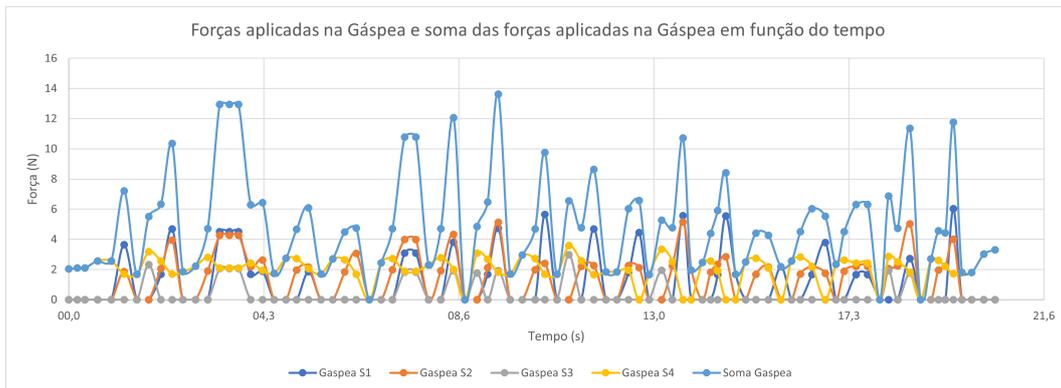


Figura 6.19: Gráfico de forças na gáspea

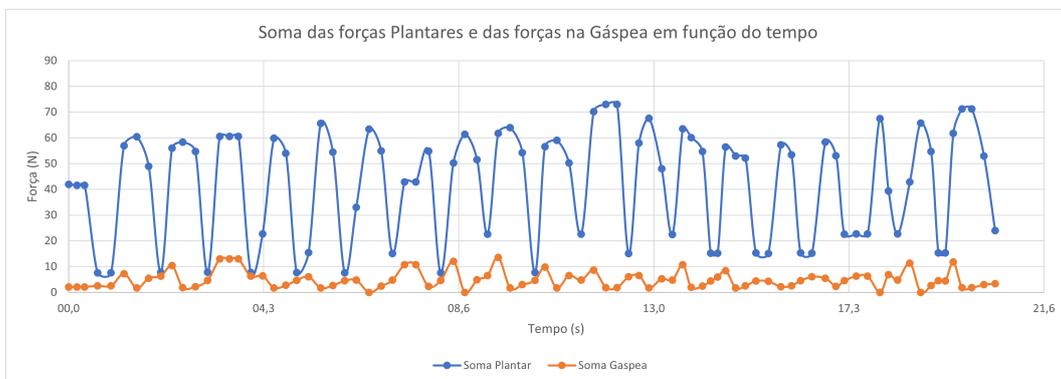


Figura 6.20: Gráfico da relação da soma das forças plantares e da soma das forças na gáspea

Pela análise do gráfico expresso na Figura 6.20 o momento onde ocorre um máximo na soma das forças da gáspea é sempre anterior ao momento em que ocorre um máximo na soma das forças na zona plantar. Atendendo à biomecânica do movimento do pé numa passada, tal facto era espectável e irá corresponder a uma fase que antecede a colocação do pé no chão, onde existirá uma flexão para cima da parte frontal do pé, levando a uma maior força aplicada na zona do aperto do sapato onde se encontra um dos sensores. Durante esse mesmo movimento do pé, há pessoas que têm a tendência de levantar os dedos dos pé, algo que o utilizador do sistema durante o teste parece realizar. Este facto pode ser comprovado pela análise do gráfico da Figura 6.19 onde é visível que os sensores 1 e 2, que se encontram localizados por cima do 1º metatarso e entre o 2º e 3º metatarsos respetivamente atingem máximos nos mesmos instantes.

Analisando os dados obtidos na gáspea, verifica-se que o valor máximo de força detetado ocorreu no sensor 1 e ronda os 6 N, que corresponderá a cerca de 35 kPa. Existem estudos que indicam valores de pressão expectável que poderão atingir 85

kPa [61]. Já na zona plantar, o valor máximo obtido ocorreu nos sensores localizados no calcanhar que rondam os 20 N e irá corresponder a cerca de 250 kPa. Para a zona do calcâneo, estão referenciados valores na ordem dos 170 kPa [61]. Sabendo-se que os valores no referido estudo têm uma grande variação relacionada com o utilizador e com o sapato, realizando-se uma comparação entre os valores obtidos e os valores referidos no artigo, verifica-se que estes encontram-se dentro do expectável.

Contudo, dependendo do utilizador, prevê-se que o padrão de pressões na zona da gáspea possa variar tal como também varia o padrão de valores na zona plantar.

6.3 Conclusão

Entre os sensores analisados, todos possuem uma relação linear entre a condutância e a força aplicada no sensor, sendo apenas necessário ajustar o valor da resistência do divisor de tensão, de modo a ajustar os valores que se pretendem medir à zona de linearidade de medição dos mesmos. Contudo, a linearidade só é mantida até um determinado valor limite, que varia de sensor para sensor.

Relativamente ao condicionamento de sinal, para a palmilha considera-se que será adequado a utilização de um divisor de tensão onde a resistência R deverá rondar os 1000 Ω e para a gáspea um divisor de tensão onde a resistência R deverá rondar os 4500 Ω .

Nos testes realizados verificou-se que os sensores Piezofabric possuíam um limite mínimo de deteção de cerca de 85 g aplicados a área útil do sensor, enquanto que os sensores piezoink possuíam um limite mínimo de deteção inferior a 1 g. Ainda nos testes realizados, o valor de massa máxima aplicada nos sensores piezofabric foi de 3355 g (cerca de 33 N) e no sensores piezoink de 1187 g (cerca de 12 N). As curvas de calibração obtidas não são concordantes com as fornecidas pelo fabricante, que pressupõem a utilização de Hardware por ele fornecido.

Após o ajuste do sistema, foram realizados testes onde foram adquiridos dados de forças plantares e forças na gáspea em simultâneo. Destes, foi possível inferir algumas conclusões como a existência de um padrão na variação da soma das forças na gáspea, tal como existe um padrão nas variações da soma das forças plantares ao longo do tempo. Este poderá alterar-se consoante a anatomia do pé do utilizador e do respectivo posicionamento do pé no solo. Os valores máximos de pressão obtidos encontram-se dentro dos valores expectáveis encontrados na literatura.

Capítulo 7

Conclusão

O trabalho apresentado tinha como objetivo o desenvolvimento de um protótipo de um sistema de sensorização, que auxilie a monitorização de pressões plantares e dorsais do pé, sendo este pensado para a utilização em pacientes que padecem de diabetes. Este sistema poderá ajudar a prevenir a ulceração e consequente amputação do pé ou partes deste.

Inicialmente foi realizado um levantamento de dispositivos semelhantes disponíveis no mercado e verificou-se que existem alguns sistemas que monitorizam as pressões plantares podendo, algumas das soluções encontradas, ser aplicadas aos diabéticos. Contudo, não foi encontrado nenhum dispositivo que monitorize, para além das pressões plantares, pressões na zona dorsal do pé, zona onde também existe o risco de ulceração. Deste modo, um dos principais pontos de inovação neste trabalho é a monitorização conjunta da zona plantar e zona dorsal do pé.

Para o sistema implementado, considerou-se ser mais adequada a utilização de sensores resistivos, pois estes possuem a vantagem de terem baixo custo, são facilmente integráveis no sistema e a aquisição do sinal é relativamente simples. Porém, dentro dos sensores do tipo resistivo, estão a emergir no mercado sensores baseados em têxtil ou mesmo em tintas piezoresistivas. Deste modo, a solução mais atrativa, tendo em conta que se pretende integrar os sensores no interior de uma palmilha de um sapato, foi a utilização de sensores têxteis. Ao contrário dos sensores FSR convencionais que tem sido amplamente estudados e utilizados em sensorização plantar, a utilização de sensores têxteis também são um foco de inovação neste projeto.

Após alguma pesquisa no mercado, optou-se pela utilização de sensores têxteis da marca SensingTex, tendo sido adquiridas amostras. A solução adquirida consistia em palmilhas, já fabricadas, com sensores têxteis ligados sob a forma de matriz 4X4. Foi ainda adquirida uma matriz de sensores em tinta piezoresistiva, de modo validar qual a melhor opção para a aplicação na zona da gáspea.

Porém, o comportamentos destes sensores, sejam eles do tipo têxtil ou de tinta piezoresistiva, era pouco conhecido. Tal implicou um estudo mais aprofundado dos mesmos e a obtenção de várias curvas de calibração com recuso a diferentes condicionamentos de sinal. Os resultados iniciais dos testes efetuados pareceram promissores, pelo que foram iniciadas conversações, com a SensingTex, no sentido de serem produzidas amostras de acordo com as localizações desejadas e definidas para o projeto. Assim, a palmilha pretendida deveria conter 9 sensores, dispostos sob a forma de matriz 3X3, localizados em zonas onde seria maior o risco de ulceração, a saber: na zona das falanges do primeiro e quinto dedos; cabeças do primeiro, segundo/terceiro e quarto/quinto metatarsos; um sensor na lateral do mediopé e três sensores localizados na região do calcanhar. Para zona da gáspea definiu-se a utilização de 4 sensores ligados em matriz 2X2, onde três sensores estariam localizados na zona das falanges do primeiro, segundo/terceiro e quarto/quinto dedos e um quarto sensor na zona do aperto do sapato.

Tendo em conta estas definição, avançou-se com todas as funcionalidades necessárias à implementação do hardware e conseqüente programação do *firmware*. Verificou-se que, relativamente ao condicionamento do sinal, estes sensores permitiam a utilização de um simples divisor de tensão, não sendo necessário, por exemplo, amplificadores operacionais para a obtenção de sinal. Todavia, existiu a necessidade de alterar o valor da resistência do divisor de tensão a fim de ajustar a melhor solução entre alcance e sensibilidade dos sensores, dependendo da zona onde se pretendiam implementar.

Optou-se pela utilização de ESP32 como MCU, visto esta possuir um número suficiente de canais ADC, permitir o envio de dados por BLE e ainda permitir a comunicação por I²C com o MPU 6050. Este último componente, trata-se de um dispositivo que possui giroscópio e acelerómetro integrados. Considera-se que a utilização do IMU poderá trazer vantagens no conhecimento da biomecânica da passada do utilizador.

Na programação do *firmware* foi necessário implementar a comunicação com o IMU, a leitura dos sensores plantares que se encontravam ligados sob a forma de matriz, a leitura do sensores da gáspea ligados sob a forma de matriz, o envio dos dados dos sensores por BLE ou por porta serie a uma determinada frequência.

O BLE é um protocolo que permite a transmissão de dados sem fios e caracteriza-se por baixo consumo, no entanto a taxa de transferência de dados é bastante limitada. Na programação do Firmware, foi necessário ter em atenção à quantidade de

dados a ser enviada (menos de 20 Bytes por transmissão) pelo que se optou pelo envio em *raw data*.

Após todo o sistema estar devidamente montado e o *firmware* programado procederem-se a testes. Rapidamente se percebeu que, para testar a fiabilidade dos valores medidos pelos sensores, seja no envio efetivo de dados por BLE, seja no envio de dados pela porta serie, seria necessário a criação de duas aplicações. Foi elaborada uma App para *Smartphone* criada em MIT APP Inventor, apenas para receber os dados por BLE, proceder ao seu tratamento e mostrar o seu valor. Outra, criada em Visual Basic, que permitisse receber dados via porta série, visualizar os resultados em gráficos desenhados em tempo real e guardar os dados num ficheiro Excel.

Como a discussão sobre a aquisição de amostras personalizadas do sensores pretendidos ainda não estava finalizada e tendo em conta que o tempo estimado para entrega destes excedia tempo previsto para a execução desta primeira fase do projeto, optou-se por adaptar as palmilhas existentes para utilização na zona plantar. Dentre as amostras de palmilhas disponíveis, utilizou-se umas semelhantes às palmilhas verde, que possuíam um maior comprimento das pistas até aos conectores e permitia uma melhor integração no sapato. Relativamente à gáspea, esta foi sensorizada recorrendo a um pequeno aplique que permitia a colocação de sensores FSR convencionais nos locais estipulados.

Foram então efetuados testes que permitissem realizar uma primeira validação do sistema, revelando estes resultados promissores. Os valores máximos de pressão obtidos na zona plantar rondariam os 250 kPa e na zona da gáspea 35 kPa, ou seja, os valores de máximos de pressão na zona da gáspea deverão ser cerca de sete vezes inferiores aos máximos da zona plantar. Mais, para a zona da gáspea parece haver um padrão na soma das forças exercidas nos sensores ao longo do tempo, tal como ocorre na zona plantar. A análise conjunta dos gráficos obtidos permite a retirada de informações relativamente ao tipo de movimentos do pé do utilizador, nomeadamente se ocorre um apoio maior para a zona lateral ou medial do pé ou se o utilizador tem tendência a exercer forças com os dedos do pé na gáspea e em que fase do ciclo de marcha tal ocorre.

O trabalho apresentado dá início do estudo desta aplicação, e consistiu na criação de um primeiro protótipo com algumas lacunas a serem melhoradas. Na secção seguinte serão exploradas algumas ideias e sugestões para um trabalho futuro.

7.1 Trabalho Futuro

Ao longo dos testes efetuados verificou-se que existe uma grande discrepância entre as curvas de calibração de sensores, escolhidos aleatoriamente, na mesma palmilha baseada em sensores em têxteis. A mesma discrepância também é verificada nas curvas de calibração dos sensores em tinta piezoresistiva. Apesar de se ter a noção

que o método utilizado para a obtenção das curvas de calibração foi um método manual, onde os erros de medição poderão ser muitos, todos os dados dos sensores foram obtidos pela mesma via e por isso, todos estariam afetos ao mesmo tipo de erros. Assim, o método utilizado para a obtenção das curvas de calibração não justifica as discrepâncias obtidas nas curvas. Uma solução para a obtenção de dados mais fiáveis, poderá passar pelo abandono da solução da ligação em matriz dos sensores. Sabe-se que este tipo de ligação poderá provocar interferências de *crosstalk* o que poderá ajudar a explicar a discrepância obtida. Contudo, também é sabido que, apesar de os sensores têxteis estarem a sofrer uma grande evolução, ainda podem apresentar problemas na sua linearidade e precisão pelo que, outra opção, poderá passar pela utilização de outro tipo de sensores que garantam resultados mais consistentes.

Outra situação que carece de atenção é a utilização a placa de desenvolvimento Esp32 DevKitC V4 , visto que, à data desta dissertação, esta placa já não está recomendada para aplicação em novas soluções. Num mundo em que começa a ser comum a utilização de recursos como *machine learning* e *cloud computing*, a utilização de uma placa de desenvolvimento que não permita a sua implementação e que para além do mais apenas possui Bluetooth 4.2, torna-se numa enorme desvantagem. Começou a ser comercializado, muito recentemente, uma nova placa de desenvolvimento da Espressif, a ESP32-S3-DevKitC-1, que apresenta 20 canais de ADC, Bluetooth 5 além de ter sido criada tendo em conta o mercado da Artificial Intelligence of Things (AIoT).

Apesar de se ter estabelecido comunicação com um IMU, e de se ter noção das vantagens que o seu estudo poderá ter na monitorização da biomecânica da marcha do utilizador, não foram efetuados quaisquer tipo de cálculos com os dados recolhidos pelo mesmo. Na realidade, está previsto que este componente fique localizado na parte inferior do sapato, segundo uma orientação ainda a definir. Estando o sistema e o IMU montado com recurso a uma *breadboard* e esta acoplada ao tornozelo, não se garantindo sempre a mesma posição, considerou-se que o seu estudo deveria ficar para uma parte mais avançada do projeto em causa.

Relativamente ao envio de dados por BLE, será necessário um estudo mais aprofundado e, provavelmente otimizado o seu envio, por forma a obter-se maiores taxas de amostragem.

Também será ainda necessário, após estarem decididos todos os componentes a serem utilizados, efetuar um estudo do consumo energético do sistema sensorizado e conseqüente escolha da bateria.

Referências

- [1] A. M. and R. Augustine, *Diabetes and healthcare: an overview*, pp. 1–129. Apple Academic Press Inc., 01 2013. [Citado na página 1]
- [2] S. C. Mishra, K. C. Chhatbar, A. Kashikar, and A. Mehndiratta, “Diabetic foot,” *the BMJ*, vol. 359, no. Supp 1, pp. 1–7, 2017. [Citado na página 2]
- [3] R. Waaijman, M. De Haart, M. L. Arts, D. Wever, A. J. Verlouw, F. Nollet, and S. A. Bus, “Risk factors for plantar foot ulcer recurrence in neuropathic diabetic patients,” *Diabetes Care*, vol. 37, no. 6, pp. 1697–1705, 2014. [Citado na página 2]
- [4] A. J. Boulton, L. Vileikyte, G. Ragnarson-Tennvall, and J. Apelqvist, “The global burden of diabetic foot disease,” 11 2005. [Citado na página 2]
- [5] N. Schaper, J. Van Netten, J. Apelqvist, S. Bus, R. Hinchliffe, B. Lipsky, and I. Board, “Practical guidelines on the prevention and management of diabetic foot disease (iwgdf 2019 update),” *Diabetes/Metabolism Research and Reviews*, vol. 36, 03 2020. [Citado nas páginas ix, 2, 5 e 19]
- [6] D. G. Armstrong, A. J. Boulton, and S. A. Bus, “Diabetic foot ulcers and their recurrence,” *New England Journal of Medicine*, vol. 376, no. 24, pp. 2367–2375, 2017. PMID: 28614678. [Citado nas páginas ix, 2, 3 e 5]
- [7] L. Prompers, M. Huijberts, J. Apelqvist, E. Jude, A. Piaggese, K. Bakker, M. Edmonds, P. Holstein, A. Jirkovska, D. Mauricio, G. Tennvall, H. Reike, M. Spraul, L. Uccioli, V. Urbancic, K. Acker, J. Baal, F. Van Merode, and N. Schaper, “High prevalence of ischemia, infection and serious comorbidity in patient with diabetic foot disease in europe. baseline results from the eurodiale study,” *Diabetologia*, vol. 50, pp. 18–25, 02 2007. [Citado nas páginas 2 e 5]
- [8] M. Monteiro-Soares, E. Boyko, J. Ribeiro, I. Ribeiro, and M. Dinis-Ribeiro, “Predictive factors for diabetic foot ulceration: a systematic review,” *Diabetes/metabolism research and reviews*, vol. 28, pp. 574–600, 10 2012. [Citado na página 3]
- [9] J. L. Lázaro-Martínez, J. Aragón-Sánchez, F. J. Álvaro-Afonso, E. García-Morales, Y. García-Álvarez, and R. J. Molines-Barroso, “The best way to reduce

- reulcerations: If you understand biomechanics of the diabetic foot, you can do it,” 12 2014. [Citado na página 3]
- [10] J. J. van Netten, J. G. van Baal, A. Bril, M. Wissink, and S. A. Bus, “An exploratory study on differences in cumulative plantar tissue stress between healing and non-healing plantar neuropathic diabetic foot ulcers,” *Clinical Biomechanics*, vol. 53, pp. 86–92, 03 2018. [Citado na página 3]
- [11] C. J. Bennetts, T. M. Owings, A. Erdemir, G. Botek, and P. R. Cavanagh, “Clustering and classification of regional peak plantar pressures of diabetic feet,” *Journal of Biomechanics*, vol. 46, pp. 19–25, 01 2013. [Citado na página 3]
- [12] P. R. Cavanagh and J. S. Ulbrecht, “Clinical plantar pressure measurement in diabetes: rationale and methodology,” *The Foot*, vol. 4, no. 3, pp. 123–135, 1994. [Citado na página 3]
- [13] A. Wang, G. Lv, X. Cheng, X. Ma, W. Wang, J. Gui, J. Hu, M. Lu, G. Chu, J. Chen, H. Zhang, Y. Jiang, Y. Chen, W. Yang, L. Jiang, G. Houfa, R. Zheng, Y. Li, W. Feng, and Y. Hu, “Guidelines on multidisciplinary approaches for the prevention and management of diabetic foot disease (2020 edition),” *Burns & trauma*, vol. 8, 07 2020. [Citado nas páginas 3 e 5]
- [14] J. Jorgetto, M. Gamba, and D. Kusahara, “Evaluation of the use of therapeutic footwear in people with diabetes mellitus – a scoping review,” *Journal of Diabetes & Metabolic Disorders*, vol. 18, 08 2019. [Citado na página 3]
- [15] “The Difference Between Diabetic Shoes and Regular Shoes.” Disponível em: <https://www.orthofeet.com/blogs/news/the-difference-between-diabetic-shoes-and-regular-shoes>. Último acesso: 2021-02-05. [Citado nas páginas ix e 4]
- [16] A. Dragulinescu, A. Dragulinescu, G. Zinčă, D. Bucur, V. Feies, and M. Neagu, “Smart socks and in-shoe systems: State-of-the-art for two popular technologies for foot motion analysis, sports, and medical applications,” *Sensors*, vol. 20, p. 4316, 08 2020. [Citado nas páginas 5, 10, 11, 21, 22, 23 e 24]
- [17] K. Deschamps, G. Matricali, P. Roosen, K. Desloovere, H. Bruyninckx, P. Spaepen, F. Nobels, J. Tits, M. Flour, and F. Staes, “Classification of forefoot plantar pressure distribution in persons with diabetes: A novel perspective for the mechanical management of diabetic foot?,” *PLoS ONE*, vol. 8, 2013. [Citado na página 5]

- [18] C. Jordan and R. Bartlett, “Perceived comfort and pressure distribution in casual footwear,” *Gait & Posture*, vol. 3, pp. 215–220, 12 1995. [Citado na página 5]
- [19] M. J. Rupérez, J. D. Martín-Guerrero, C. Monserrat, and M. Alcañiz, “Artificial neural networks for predicting dorsal pressures on the foot surface while walking,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5349–5357, 2012. [Citado nas páginas 5, 16, 18 e 20]
- [20] A. Herbaut, E. Simoneau-Buessinger, F. Barbier, F. Cannard, and N. Guéguen, “A reliable measure of footwear upper comfort enabled by an innovative sock equipped with textile pressure sensors,” *Ergonomics*, vol. 59, no. 10, pp. 1327–1334, 2016. [Citado nas páginas 5, 16, 17 e 18]
- [21] S. Ostadabbas, A. Saeed, M. Nourani, and M. Pompeo, “Sensor architectural tradeoff for diabetic foot ulcer monitoring,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2012, pp. 6687–6690, 2012. [Citado nas páginas 5, 16 e 18]
- [22] A. Martinez-Nova, J. P. Huerta, and R. Sanchez-Rodriguez, “Cadence, age, and weight as determinants of forefoot plantar pressures using the Biofoot in-shoe system,” *Journal of the American Podiatric Medical Association*, vol. 98, no. 4, pp. 302–310, 2008. [Citado na página 5]
- [23] S. Ahmed, A. Barwick, P. Butterworth, and S. Nancarrow, “Footwear and insole design features that reduce neuropathic plantar forefoot ulcer risk in people with diabetes: a systematic literature review,” *Journal of Foot and Ankle Research*, vol. 13, 12 2020. [Citado na página 5]
- [24] D. G. Armstrong, A. J. Boulton, and S. A. Bus, “Diabetic Foot Ulcers and Their Recurrence,” *New England Journal of Medicine*, vol. 376, pp. 2367–2375, 06 2017. [Citado na página 5]
- [25] F. Mokhtari, Z. Cheng, R. Raad, J. Xi, and J. Foroughi, “Piezofibers to smart textiles: a review on recent advances and future outlook for wearable technology,” *J. Mater. Chem. A*, vol. 8, pp. 9496–9522, 2020. [Citado na página 9]
- [26] Sensoria, “Sensoria Artificial Intelligence Sportswear.” Disponível em: <https://www.sensoriafitness.com/>. Último acesso: 2021-08-05. [Citado nas páginas ix, 10 e 16]
- [27] Tekscan, “F-Scan System: Ultra-thin, in-shoe sensors capture timing & pressure information for foot function & gait analysis.” Disponível em: <https://www.>

- tekscan.com/products-solutions/systems/f-scan-system. Último acesso: 2021-08-05. [Citado nas páginas ix, 10 e 12]
- [28] D. Smartshoe, “Smartshoe | the world’s first intelligent sneaker.” Disponível em: <https://www.kickstarter.com/projects/141658446/digitsole-smartshoe-the-worlds-first-intelligent-s>. Último acesso: 2021-08-05. [Citado nas páginas ix e 11]
- [29] A. Razak, A. Zayegh, R. Begg, and Y. Wahab, “Foot plantar pressure measurement system: A review,” *Sensors (Basel, Switzerland)*, vol. 12, pp. 9884–912, 12 2012. [Citado nas páginas 11, 17 e 21]
- [30] S. Bamberg, A. Benbasat, D. Scarborough, D. Krebs, and J. Paradiso, “Gait analysis using a shoe-integrated wireless sensor system,” *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 12, pp. 413–23, 07 2008. [Citado nas páginas 11 e 20]
- [31] H. Tanwar, L. Nguyen, and N. Stergiou, “Force sensitive resistor (fsr)-based wireless gait analysis device,” *Proceedings of the 3rd IASTED International Conference on Telehealth*, pp. 1–6, 01 2007. [Citado na página 11]
- [32] N. Lee, R. Goonetilleke, Y. Cheung, and G. So, “A flexible encapsulated mems pressure sensor system for biomechanical applications,” *Microsystem Technologies*, vol. 7, pp. 55–62, 05 2001. [Citado nas páginas 11, 20 e 21]
- [33] Novel.de, “pedar®: Dynamic pressure distribution inside the footwear.” Disponível em: <https://www.novel.de/products/pedar/>. Último acesso: 2021-08-05. [Citado nas páginas ix, 11 e 12]
- [34] J. A. Ramirez-Bautista, J. A. Huerta-Ruelas, S. L. Chaparro-Cárdenas, and A. Hernández-Zavala, “A review in detection and monitoring gait disorders using in-shoe plantar measurement systems,” *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 299–309, 2017. [Citado nas páginas 12, 13, 21, 22, 23 e 24]
- [35] Orpyx, “Orpyx® SI Sensory Insoles Help Prevent Diabetic Foot Ulcers .” Disponível em: <https://www.orpyx.com>. Último acesso: 2021-02-05. [Citado nas páginas ix e 13]
- [36] B. Najafi, E. Ron, A. Enriquez, I. Marin, J. Razjouyan, and D. Armstrong, “Smarter sole survival: Will neuropathic patients at high risk for ulceration use a smart insole-based foot protection system?,” *Journal of Diabetes Science and Technology*, vol. 11, 01 2017. [Citado na página 13]

- [37] B. Najafi, N. D. Reeves, and D. G. Armstrong, “Leveraging smart technologies to improve the management of diabetic foot ulcers and extend ulcer-free days in remission.,” *Diabetes/metabolism research and reviews*, vol. 36 Suppl 1, p. e3239, 03 2020. [Citado nas páginas 13 e 16]
- [38] Salted, “Make Any Footwear Smarter.” Disponível em: <https://www.salted.ltd/insole>. Último acesso: 2021-02-05. [Citado nas páginas ix e 14]
- [39] Flextrapower. Disponível em: <https://www.flextrapower.com/products>. Último acesso: 2021-02-05. [Citado nas páginas ix, 14 e 15]
- [40] Lechal, “Step Into the Footwear of the Future: GPS Navigation and more for your feet .” Disponível em: <https://lechal.com/#Home>. Último acesso: 2021-02-05. [Citado nas páginas ix e 15]
- [41] M. Lord and R. Hosein, “A study of in-shoe plantar shear in patients with diabetic neuropathy,” *Clinical Biomechanics*, vol. 15, no. 4, pp. 278–283, 2000. [Citado nas páginas 16, 17 e 18]
- [42] T. Mori, M. Hamatani, H. Noguchi, M. Oe, and H. Sanada, “Insole-type simultaneous measurement system of plantar pressure and shear force during gait for diabetic patients,” *Journal of Robotics and Mechatronics*, vol. 24, no. 5, pp. 766–772, 2012. [Citado nas páginas 16, 17 e 18]
- [43] M. Takano, H. Noguchi, M. Oe, H. Sanada, and T. Mori, “Development and evaluation of a system to assess the effect of footwear on the in shoe plantar pressure and shear during gait,” *ROBOMECH Journal*, vol. 1, no. 1, 2014. [Citado nas páginas 16, 17 e 18]
- [44] M. Hamatani, T. Mori, M. Oe, H. Noguchi, K. Takehara, A. Amemiya, Y. Ohashi, K. Ueki, T. Kadowaki, and H. Sanada, “Factors Associated with Callus in Patients with Diabetes, Focused on Plantar Shear Stress during Gait,” *Journal of Diabetes Science and Technology*, vol. 10, no. 6, pp. 1353–1359, 2016. [Citado nas páginas 16, 17 e 18]
- [45] H. Zhu, J. J. Wertsch, G. F. Harris, H. M. Alba, and M. B. Price, “Sensate and insensate in-shoe plantar pressures,” *Archives of Physical Medicine and Rehabilitation*, vol. 74, no. 12, pp. 1362–1368, 1993. [Citado nas páginas 16, 17 e 18]
- [46] J. J. Wertsch, L. W. Frank, H. Zhu, M. B. Price, G. F. Harris, and H. M. Alba, “Plantar pressures with total contact casting,” *Journal of Rehabilitation Research and Development*, vol. 32, no. 3, pp. 205–209, 1995. [Citado nas páginas 16, 17 e 18]

- [47] V. G. Femery, P. G. Moretto, J.-M. G. Hespel, A. Thévenon, and G. Lensel, “A real-time plantar pressure feedback device for foot unloading.,” *Archives of physical medicine and rehabilitation*, vol. 85, pp. 1724–1728, 10 2004. [Citado nas páginas ix, 16, 17, 18 e 20]
- [48] D. Brown, J. J. Wertsch, G. F. Harris, J. Klein, and D. Janisse, “Effect of Rocker Soles on Plantar Pressures,” *Archives of Physical Medicine and Rehabilitation*, vol. 85, no. 1, pp. 81–86, 2004. [Citado nas páginas 16, 17 e 18]
- [49] M. N. Anas, “An instrumented insole system for gait monitoring and analysis,” *International Journal of Online Engineering*, vol. 10, no. 6, pp. 30–34, 2014. [Citado nas páginas 16, 17 e 18]
- [50] F. Dabiri, A. Vahdatpour, H. Noshadi, H. Hagopian, and M. Sarrafzadeh, “Electronic orthotics shoe: preventing ulceration in diabetic patients.,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2008, pp. 771–774, 2008. [Citado nas páginas 16, 17 e 18]
- [51] A. Perrier, N. Vuillerme, V. Luboz, M. Bucki, F. Cannard, B. Diot, D. Colin, D. Rin, J.-P. Bourg, and Y. Payan, “Smart Diabetic Socks: Embedded device for diabetic foot prevention,” *IRBM*, vol. 35, no. 2, pp. 72–76, 2014. [Citado nas páginas ix, 16, 17, 18 e 24]
- [52] A. Raviglione, R. Reif, M. Macagno, D. Vigano, J. Schram, and D. Armstrong, “Real-Time Smart Textile-Based System to Monitor Pressure Offloading of Diabetic Foot Ulcers.,” *Journal of diabetes science and technology*, vol. 11, pp. 894–898, 09 2017. [Citado nas páginas 16, 17 e 18]
- [53] X. Lin and B.-C. Seet, “Battery-Free Smart Sock for Abnormal Relative Plantar Pressure Monitoring,” *IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS*, vol. 11, no. 2, pp. 464–473, 2017. [Citado nas páginas 16, 17 e 18]
- [54] Y. Tan, K. Ivanov, Z. Mei, H. Li, H. Li, L. Lubich, C. Wang, and L. Wang, “A soft wearable and fully-textile piezoresistive sensor for plantar pressure capturing,” *Micromachines*, vol. 12, pp. 1–12, 02 2021. [Citado nas páginas 16, 17 e 18]
- [55] E. Klimiec, J. Piekarski, W. Zaraska, and B. Jasiewicz, “Electronic measurement system of foot plantar pressure,” *Microelectronics International*, vol. 31, no. 3, pp. 229–234, 2014. [Citado nas páginas 16, 17 e 18]
- [56] A. Amemiya, H. Noguchi, M. Oe, H. Sanada, and T. Mori, “Establishment of a measurement method for in-shoe pressure and shear stress in specific regions

- for diabetic ulcer prevention.,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2016, pp. 2291–2294, 08 2016. [Citado nas páginas 16, 17 e 18]
- [57] S. Rajala, R. Mattila, I. Kaartinen, and J. Lekkala, “Designing, Manufacturing and Testing of a Piezoelectric Polymer Film In-Sole Sensor for Plantar Pressure Distribution Measurements,” *IEEE SENSORS JOURNAL*, vol. 17, no. 20, pp. 6798–6805, 2017. [Citado nas páginas 16, 17 e 18]
- [58] J. Olaso, J. C. González, S. Alemany, E. Medina, A. López, C. Martín, J. Prat, and C. Soler, “Study of the influence of fitting and walking condition in foot dorsal pressure,” *Proc. 8th Footwear Biomechanics Symposium–Taipei*, p. 41, 2007. [Citado nas páginas 16, 17, 18 e 20]
- [59] S. Takesue, P. Y. Loh, S. Muraki, S. Hamanaka, A. Yamada, K. Ikegami, K. Wada, and H. Furutachi, “Individual differences in contact pressure on the dorsal surface of the foot during gait,” *Advances in Intelligent Systems and Computing*, vol. 824, pp. 216–219, 2019. [Citado nas páginas 16, 17, 18 e 20]
- [60] M. Hagen, A. K. Homme, T. Umlauf, and E. M. Hennig, “Effects of different shoe-lacing patterns on dorsal pressure distribution during running and perceived comfort,” *Research in Sports Medicine*, vol. 18, no. 3, pp. 176–187, 2010. [Citado nas páginas 16, 17, 18 e 20]
- [61] Q. Mei, M. Graham, and Y. Gu, “Biomechanical analysis of the plantar and upper pressure with different sports shoes,” *International Journal of Biomedical Engineering and Technology*, vol. 14, no. 3, pp. 181–191, 2014. [Citado nas páginas 16, 17, 18 e 78]
- [62] D. Wang, J. Ouyang, P. Zhou, J. Yan, L. Shu, and X. Xu, “A Novel Low-Cost Wireless Footwear System for Monitoring Diabetic Foot Patients,” *IEEE Transactions on Biomedical Circuits and Systems*, 2020. [Citado nas páginas ix, 17 e 20]
- [63] L. Du, X. Zhu, and J. Zhe, “An inductive sensor for real-time measurement of plantar normal and shear forces distribution,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 5, pp. 1316–1323, 2015. [Citado na página 17]
- [64] P. Aqueveque, R. Osorio, F. Pastene, F. Saavedra, and E. Pino, “Capacitive Sensors Array for Plantar Pressure Measurement Insole fabricated with Flexible PCB.,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2018, pp. 4393–4396, 07 2018. [Citado nas páginas ix, 17 e 22]

- [65] L. Shu, T. Hua, Y. Wang, Q. Li, D. D. F. Feng, and X. Tao, “In-shoe plantar pressure measurement and analysis system based on fabric pressure sensing array,” *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 14, pp. 767–75, 05 2010. [Citado nas páginas ix, 17 e 19]
- [66] Z. Pataky, L. Faravel, J. Silva, and J.-P. Assal, “A new ambulatory foot pressure device for patients with sensory impairment. a system for continuous measurement of plantar pressure and a feed-back alarm,” *Journal of biomechanics*, vol. 33, pp. 1135–8, 10 2000. [Citado nas páginas ix, 17 e 20]
- [67] H. S. Zhu, J. J. Wertsch, G. F. Harris, J. D. Loftsgaarden, and M. B. Price, “Foot pressure distribution during walking and shuffling,” *Archives of Physical Medicine and Rehabilitation*, vol. 72, no. 6, pp. 390–397, 1991. [Citado na página 17]
- [68] F. Xu, X. Li, Y. Shi, L. Li, W. Wang, L. He, and R. Liu, “Recent developments for flexible pressure sensors: A review,” *Micromachines*, vol. 9, p. 580, 11 2018. [Citado nas páginas ix, 21, 22 e 23]
- [69] M. Cheng, G. Zhu, F. Zhang, W.-L. Tang, S. Jianping, J.-Q. Yang, and L.-Y. Zhu, “A review of flexible force sensors for human health monitoring,” *Journal of advanced research*, vol. 26, 11 2020. [Citado nas páginas 21, 22 e 23]
- [70] Arduino.cc, “Arduino Uno Rev3.” Disponível em: <https://store.arduino.cc/products/arduino-uno-rev3?selectedStore=eu>. Último acesso: 2021-04-08. [Citado na página 29]
- [71] Espressif, “ESP32 DataSheet.” Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 03 2021. Último acesso: 2021-04-01. [Citado na página 29]
- [72] N. Semiconductor, “nRF52832 Product Specificationv1.4.” Disponível em: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_nrf52%2Fstruct%2Fnrf52832.html, 10 2017. Último acesso: 2021-04-05. [Citado na página 29]
- [73] Espressif, “ESP32 WROOM 32D & ESP32 WROOM 32U Datasheet.” Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf, 03 2021. Último acesso: 2021-04-10. [Citado na página 29]
- [74] M. Afaneh, *INTRO TO BLUETOOTH LOW ENERGY*. Novel Bits, 2018. [Citado nas páginas xiii, 29, 30, 31, 32 e 53]

- [75] Olivia'sPc, "BLE Protocol Stack — Controller." Disponível em: <https://pcng.medium.com/ble-protocol-stack-controller-2d2d5371deec>, 09 2019. Último acesso: 2021-09-10. [Citado nas páginas ix e 30]
- [76] M. Bhargava and J. Wilson, *IoT Projects with bluetooth low energy: harness the power of connected things*. Packt Publishing, 2017. [Citado nas páginas 31 e 32]
- [77] B. S. Proprietary, "Bluetooth Core Specification v 5.0." Disponível em: <https://www.mouser.it/pdfdocs/bluetooth-Core-v50.pdf>, 12 2016. Último acesso: 2021-09-27. [Citado nas páginas ix e 31]
- [78] I. Eletronics, "FSR 400 series Data Sheet ." Disponível em: https://cdn2.hubspot.net/hubfs/3899023/Interlinkelectronics%20November2017/Docs/Datasheet_FSR.pdf. Último acesso: 2021-04-08. [Citado na página 32]
- [79] Tekscan, "FlexiForce Standard Model A201 ." Disponível em: <https://www.tekscan.com/sites/default/files/resources/FLX-Datasheet-A201-RevH.pdf>, 03 2021. Último acesso: 2021-04-02. [Citado na página 32]
- [80] Tekscan, "FlexiForce Standard Model A401 ." Disponível em: <https://www.tekscan.com/sites/default/files/resources/FLX-Datasheet-A401-RevH.pdf>, 03 2021. Último acesso: 2021-04-02. [Citado na página 32]
- [81] SensingTex, "PRESSURE MAT ." Disponível em: http://sensingtex.com/wp-content/uploads/2018/05/114_WHITE_PAPER_Pressure_Mat_english_rev_26.pdf, 2017. Último acesso: 2021-04-05. [Citado na página 32]
- [82] N. Strzalkowski, R. Peters, J. Inglis, and L. Bent, "Cutaneous afferent innervation of the human foot sole: What can we learn from single unit recordings?," *Journal of Neurophysiology*, vol. 120, 06 2018. [Citado nas páginas x, xiii, 33 e 34]
- [83] S. Kerr, B. Olivier, A. Green, C. Dafkin, S. Wood, A. Woodiwiss, and W. McKinon, "Analysis of balance and body positioning in ballerinas with different levels of skill," 09 2016. [Citado na página 35]
- [84] K. Clark, L. Ryan, and P. Weyand, "A general relationship links gait mechanics and running ground reaction forces," *The Journal of experimental biology*, vol. 220, 11 2016. [Citado na página 35]
- [85] InvenSense, "MPU-6000 and MPU-6050 Product Specification Revision 3.4." Disponível em: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. Último acesso: 2021-04-08. [Citado nas páginas 36 e 37]

- [86] Bosch, “ Data Sheet BHI160 / BHI160B Ultra low-power sensor hub incl. integrated IMU.” Disponível em: <https://nl.mouser.com/datasheet/2/783/bst-bhi160b-ds000-1842960.pdf>. Último acesso: 2021-04-08. [Citado na página 36]
- [87] T. InvenSense, “ High Performance 6-Axis MEMS MotionTracking™ Device .” Disponível em: <https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/03/DS-000143-ICM-20689-TYP-v1.1.pdf>. Último acesso: 2021-04-08. [Citado na página 36]
- [88] InvenSense, “ MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2 .” Disponível em: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>. Último acesso: 2021-04-08. [Citado na página 37]
- [89] botnroll, “ Placa de desenvolvimento ESP32 c/ conectores fema - ESPRESSIF .” Disponível em: <https://www.botnroll.com/pt/esp/3131-placa-de-desenvolvimento-esp32-c-conectores-femea-espressif.html>. Último acesso: 2021-09-27. [Citado nas páginas x e 42]

Anexo A

Código Firmware

A.1 Inicialização de variáveis

```
1
2
3
4 #define SERVICE_UUID          "d5466fb8-2b71-437a-949d-9e32a567ca2b"
5 #define CHARACTERISTIC_UUID_Plantar "2ff44774-809a-4785-83ff-
   a1b2c9fd5c5f"
6 #define CHARACTERISTIC_UUID_Imu "3b6b8f75-6345-40cf-baf9-79
   ff19d0bd7a"
7 #define CHARACTERISTIC_UUID_Gaspea "03a69008-6a0e-4fc0-ad03-89
   b2fa20d0df"
8
9 uint16_t imu[6];
10 int fsrPinIn[3] = {25, 26, 27};
11 int fsrPinOut[3] = {34, 35, 32};
12 int fsrPinInGas[2] = {18, 19};
13 int fsrPinOutGas[2] = {36, 39};
14 float fsrVoltage[3][3];
15 float fsrResistance[3][3];
16 float fsrConductance[3][3];
17 uint16_t fsrForce[3][3];
18 float fsrVoltageGas[2][2];
19 float fsrResistanceGas[2][2];
20 float fsrConductanceGas[2][2];
```

```

21 uint16_t fsrForceGas [2][2];
22 int const num_line = 3;
23 int const num_col = 3;
24 uint16_t outputData[num_line][num_col];
25 uint16_t outputDataGas [2][2];
26 bool deviceConnected = false;
27 unsigned long startMicros;
28 unsigned long lastsendTime = 0;
29
30
31 BLECharacteristic *pCharacteristicPlantar;
32 BLECharacteristic *pCharacteristicImu;
33 BLECharacteristic *pCharacteristicGaspea;

```

A.2 Inicialização do IMU

```

1
2 void initImu() {
3     // Try to initialize!
4     if (!mpu.begin()) {
5         Serial.println("Failed to find MPU6050 chip");
6         while (1) {
7             delay(10);
8         }
9     }
10    Serial.println("MPU6050 Found!");
11
12    mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
13    Serial.print("Accelerometer range set to: ");
14    switch (mpu.getAccelerometerRange()) {
15        case MPU6050_RANGE_2_G:
16            Serial.println("+-2G");
17            break;
18        case MPU6050_RANGE_4_G:
19            Serial.println("+-4G");
20            break;
21        case MPU6050_RANGE_8_G:
22            Serial.println("+-8G");
23            break;
24        case MPU6050_RANGE_16_G:
25            Serial.println("+-16G");
26            break;
27    }
28    mpu.setGyroRange(MPU6050_RANGE_500_DEG);
29    Serial.print("Gyro range set to: ");
30    switch (mpu.getGyroRange()) {

```

```
31     case MPU6050_RANGE_250_DEG:
32         Serial.println("+ - 250 deg/s");
33         break;
34     case MPU6050_RANGE_500_DEG:
35         Serial.println("+ - 500 deg/s");
36         break;
37     case MPU6050_RANGE_1000_DEG:
38         Serial.println("+ - 1000 deg/s");
39         break;
40     case MPU6050_RANGE_2000_DEG:
41         Serial.println("+ - 2000 deg/s");
42         break;
43 }
44
45 mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
46 Serial.print("Filter bandwidth set to: ");
47 switch (mpu.getFilterBandwidth()) {
48     case MPU6050_BAND_260_HZ:
49         Serial.println("260 Hz");
50         break;
51     case MPU6050_BAND_184_HZ:
52         Serial.println("184 Hz");
53         break;
54     case MPU6050_BAND_94_HZ:
55         Serial.println("94 Hz");
56         break;
57     case MPU6050_BAND_44_HZ:
58         Serial.println("44 Hz");
59         break;
60     case MPU6050_BAND_21_HZ:
61         Serial.println("21 Hz");
62         break;
63     case MPU6050_BAND_10_HZ:
64         Serial.println("10 Hz");
65         break;
66     case MPU6050_BAND_5_HZ:
67         Serial.println("5 Hz");
68         break;
69         Serial.println("");
70         delay(100);
71     }
72 }
```

A.3 Inicialização BLE

```
1 class MyServerCallbacks: public BLEServerCallbacks {
2     void onConnect(BLEServer* pServer) {
3         deviceConnected = true;
4     };
5
6     void onDisconnect(BLEServer* pServer) {
7         deviceConnected = false;
8     }
9 };
10 void myBLEInit() {
11
12     BLEDevice::init("Pe_Diabetico");
13     BLEServer *pServer = BLEDevice::createServer();
14     pServer->setCallbacks(new MyServerCallbacks());
15     BLEService *pService = pServer->createService(SERVICE_UUID);
16     //create characteristic plantar sensores
17
18     pCharacteristicPlantar = pService->createCharacteristic(
19         CHARACTERISTIC_UUID_Plantar,
20         BLECharacteristic::PROPERTY_NOTIFY
21     );
22     pCharacteristicPlantar->addDescriptor(new BLE2902());
23
24     //create characteristic IMU
25     pCharacteristicImu = pService->createCharacteristic(
26         CHARACTERISTIC_UUID_Imu,
27         BLECharacteristic::PROPERTY_NOTIFY
28     );
29     pCharacteristicImu->addDescriptor(new BLE2902());
30
31     //create characteristic gaspea sensores
32     pCharacteristicGaspea = pService->createCharacteristic(
33         CHARACTERISTIC_UUID_Gaspea,
34         BLECharacteristic::PROPERTY_READ |
35         BLECharacteristic::PROPERTY_WRITE |
36         BLECharacteristic::PROPERTY_NOTIFY |
37         BLECharacteristic::PROPERTY_INDICATE
38     );
39     pCharacteristicGaspea->addDescriptor(new BLE2902());
40
41     pService->start();
42     pServer->getAdvertising()->start();
43     Serial.println("Characteristic defined! Now you can read it in
44         your phone!");
45 }
```

A.4 Handler interrupção

```
1
2 volatile int interruptCounter;
3 hw_timer_t * timer = NULL;
4 portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
5
6
7
8 void IRAM_ATTR onTimer() {
9     portENTER_CRITICAL_ISR(&timerMux);
10    interruptCounter++;
11    portEXIT_CRITICAL_ISR(&timerMux);
12
13 }
```

A.5 Função Setup

```
1 void setup(void) {
2     Serial.begin(115200); // We'll send debugging information via
3     the Serial monitor
4     pinMode(25, OUTPUT); //it Declares pin 25 as an output
5     pinMode(26, OUTPUT);
6     pinMode(27, OUTPUT);
7     pinMode(18, OUTPUT);
8     pinMode(19, OUTPUT);
9     pinMode(34, INPUT); //it Declares pin 34 as an input
10    pinMode(35, INPUT);
11    pinMode(32, INPUT);
12    pinMode(36, INPUT);
13    pinMode(39, INPUT);
14    startMicros = micros();
15    initImu();
16    myBLEInit();
17    timer = timerBegin(0, 40, true);
18    timerAttachInterrupt(timer, &onTimer, true);
19    timerAlarmWrite(timer, 100000, true);
20    timerAlarmEnable(timer);
21 }
```

A.6 Função loop

```
1 void loop(void) {
2   int i, j;
3   int totalTimeSensorRead;
4   int totalTimeBLESend;
5   unsigned long readSensortime;
6   unsigned long finalMicros;
7   unsigned long inicialMicros;
8
9   if (interruptCounter > 0) {
10
11     portENTER_CRITICAL(&timerMux);
12     interruptCounter--;
13     portEXIT_CRITICAL(&timerMux);
14
15     if (deviceConnected) {
16
17
18       inicialMicros = micros();
19       readImu();
20       readSensorPlantar();
21       readSensorGaspea();
22       readSensortime = micros();
23       sendBLEData();
24       lastsendTime = micros();
25       totalTimeSensorRead = readSensortime - inicialMicros;
26       totalTimeBLESend = lastsendTime - readSensortime;
27       Serial.print("Tempo leitura sensores:");
28       Serial.println(totalTimeSensorRead);
29
30       Serial.print("Tempo envio BLE:");
31       Serial.println(totalTimeBLESend);
32
33     } else {
34       readSensorPlantar();
35       readSensorGaspea();
36       readImu();
37
38       Serial.print("A");
39       Serial.println(fsrForceGas [0] [0]);
40
41
42       Serial.print("B");
43       Serial.println(fsrForceGas [0] [1]);
44
45
46       Serial.print("C");
```

```
47     Serial.println(fsrForceGas [1] [0]);
48
49
50     Serial.print("D");
51     Serial.println(fsrForceGas [1] [1]);
52
53
54     Serial.print("E");
55     Serial.println(fsrForce [0] [0]);
56
57     Serial.print("F");
58     Serial.println(fsrForce [0] [1]);
59
60     Serial.print("G");
61     Serial.println(fsrForce [0] [2]);
62
63     Serial.print("H");
64     Serial.println(fsrForce [1] [0]);
65
66     Serial.print("I");
67     Serial.println(fsrForce [1] [1]);
68
69     Serial.print("J");
70     Serial.println(fsrForce [1] [2]);
71
72     Serial.print("K");
73     Serial.println(fsrForce [2] [0]);
74
75     Serial.print("L");
76     Serial.println(fsrForce [2] [1]);
77
78     Serial.print("M");
79     Serial.println(fsrForce [2] [2]);
80
81     Serial.print("N");
82     Serial.println(imu [0]);
83
84     Serial.print("O");
85     Serial.println(imu [1]);
86
87     Serial.print("P");
88     Serial.println(imu [2]);
89
90     Serial.print("Q");
91     Serial.println(imu [3]);
92
93     Serial.print("R");
94     Serial.println(imu [4]);
95
```

```
96     Serial.print("S");
97     Serial.println(imu[5]);
98   }
99 }
100
101 }
```

A.7 Leitura do IMU

```
1 void readImu() {
2   /* Get new sensor events with the readings */
3   int i, aceX, aceY, aceZ, rotX, rotY, rotZ;
4   sensors_event_t a, g, temp;
5   mpu.getEvent(&a, &g, &temp);
6
7   aceX = (a.acceleration.x * 1000) + 10000;
8   aceY = (a.acceleration.y * 1000) + 10000;
9   aceZ = (a.acceleration.z * 1000) + 10000;
10  rotX = (g.gyro.x * 1000) + 10000;
11  rotY = (g.gyro.y * 1000) + 10000;
12  rotZ = (g.gyro.z * 1000) + 10000;
13
14  imu[0] = (uint16_t)aceX;
15  imu[1] = (uint16_t)aceY;
16  imu[2] = (uint16_t)aceZ;
17  imu[3] = (uint16_t)rotX;
18  imu[4] = (uint16_t)rotY;
19  imu[5] = (uint16_t)rotZ;
20 }
```

A.8 Leitura dos sensores plantares

```
1 void readSensorPlantar() {
2   int i, j, k;
3
4   for (i = 0; i < num_line; i++) {
5
6     digitalWrite(fsrPinIn[i], HIGH);
7
8     for (j = 0; j < num_col; j++) {
9
10      outputData[i][j] = analogRead(fsrPinOut[j]);
```

```

11
12
13     fsrVoltage[i][j] = map(outputData[i][j], 0, 4095, 0, 3300);
        // analog voltage reading ranges from about 0 to 4095
        // which maps to 0V to 3,3V (= 3300mV)
14     if (fsrVoltage[i][j] == 0) {
15         fsrResistance[i][j] = 0;
16         fsrConductance[i][j] = 0;
17         fsrForce[i][j] = 0;
18
19
20     } else {
21
22         fsrResistance[i][j] = 3300 - fsrVoltage[i][j]; // The
            voltage = Vcc * R / (R + FSR) where R = 1.2K and Vcc =
            3,3V so FSR = ((Vcc - V) * R) / V ; fsrVoltage is in
            millivolts so 3,3V = 3300mV
23         fsrResistance[i][j] *= 1200;
24         fsrResistance[i][j] /= fsrVoltage[i][j];
25         fsrConductance[i][j] = 1000000; // we measure in
            microohms
26         fsrConductance[i][j] /= fsrResistance[i][j];
27         fsrForce[i][j] = (fsrConductance[i][j] + 2210) / 3, 6;
28
29     }
30 }
31
32     digitalWrite(fsrPinIn[i], LOW);
33 }
34 }

```

A.9 Leitura dos sensores da gáspea

```

1 void readSensorGaspea() {
2     int i, j, k;
3
4     for (i = 0; i < 2; i++) {
5
6         digitalWrite(fsrPinInGas[i], HIGH);
7
8         for (j = 0; j < 2; j++) {
9
10            outputDataGas[i][j] = analogRead(fsrPinOutGas[j]);
11

```

```

12     fsrVoltageGas[i][j] = map(outputDataGas[i][j], 0, 4095, 0,
13                               3300); // analog voltage reading ranges from about 0 to
14                               4095 which maps to 0V to 3,3V (= 3300mV)
15
16     if (fsrVoltageGas[i][j] == 0) {
17         fsrResistanceGas[i][j] = 0;
18         fsrConductanceGas[i][j] = 0;
19         fsrForceGas[i][j] = 0;
20
21     } else {
22
23         fsrResistanceGas[i][j] = 3300 - fsrVoltageGas[i][j]; //
24         // The voltage = Vcc * R / (R + FSR) where R = 4.3kOhm
25         // and Vcc = 3,3V so FSR = ((Vcc - V) * R) / V ;
26         // fsrVoltage is in millivolts so 3,3V = 3300mV
27         fsrResistanceGas[i][j] *= 4300;
28         fsrResistanceGas[i][j] /= fsrVoltageGas[i][j];
29         fsrConductanceGas[i][j] = 1000000; // we measure
30         // in microohms
31         fsrConductanceGas[i][j] /= fsrResistanceGas[i][j];
32         fsrForceGas[i][j] = ((fsrConductanceGas[i][j] + 150) / 90)
33         * 100;
34     }
35 }
36
37 digitalWrite(fsrPinInGas[i], LOW);
38 }
39 }

```

A.10 Envío de datos por BLE

```

1
2 void sendBLEData() {
3     int i, j, k, z, x;
4     uint8_t joinVariables[18];
5     uint8_t joinvariablesImu[12];
6     uint8_t joinVariablesGas[8];
7
8     k = 0;
9     z = 0;
10    for (i = 0; i < 3; i++) {
11        for (j = 0; j < 3; j++) {
12            joinVariables[k + 1] = fsrForce[i][j] >> 8;
13            joinVariables[k] = fsrForce[i][j];

```

```
14     k = k + 2;
15   }
16 }
17
18 for (i = 0; i < 6; i++) {
19     joinvariablesImu[z + 1] = imu[i] >> 8;
20     joinvariablesImu[z] = imu[i];
21     z = z + 2;
22 }
23
24 for (i = 0; i < 2; i++) {
25     for (j = 0; j < 2; j++) {
26         joinVariablesGas[x + 1] = fsrForceGas[i][j] >> 8;
27         joinVariablesGas[x] = fsrForceGas[i][j];
28         x = x + 2;
29     }
30 }
31
32 pCharacteristicPlantar->setValue(joinVariables, 18);
33 pCharacteristicImu->setValue(joinvariablesImu, 12);
34 pCharacteristicGaspea->setValue(joinVariablesGas, 8);
35
36
37 pCharacteristicGaspea->notify();
38 pCharacteristicPlantar->notify();
39 pCharacteristicImu->notify();
40
41
42 }
```

Anexo B

Código aplicação Visual Basic

B.1 Inicialização do programa e de variáveis

```
1
2 Imports System
3 Imports System.IO.Ports
4 Imports System.Linq
5 Imports System.Data.SqlClient
6 Imports System.Data.OleDb
7 Imports Microsoft.Office.Core
8 Imports Excel = Microsoft.Office.Interop.Excel
9 Imports ExcelAutoFormat = Microsoft.Office.Interop.Excel.
    XlRangeAutoFormat
10 Imports Microsoft.Office.Interop
11 Imports System.IO
12 Imports System.Xml.XPath
13 Imports System.Data
14 Imports System.Xml
15
16
17 Public Class Form1
18
19     Dim S1L, S2L, S3L, S4L, SP1L, SP2L, SP3L, SP4L, SP5L, SP6L,
        SP7L, SP8L, SP9L, AXL, AYL, AZL, GXL, GYL, GZL As Integer
```

```

20 Dim S1Result, S2Result, S3Result, S4Result, SP1Result,
    SP2Result, SP3Result, SP4Result, SP5Result, SP6Result,
    SP7Result, SP8Result, SP9Result, AXResult, AYResult,
    AZResult, GXResult, GYResult, GZResult As Decimal
21
22 Dim S1, S2, S3, S4 As String
23 Dim SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9 As String
24 Dim AX, AY, AZ, GX, GY, GZ As String
25 Dim SensorSum As Decimal
26 Dim ChartLimit As Integer = 30
27 Dim StrSerialIn, StrSerialInRam As String
28 Dim FilePathAndName As String
29 Dim MaxS1, MaxS2, MaxS3, MaxS4, MaxSP1, MaxSP2, MaxSP3, MaxSP4
    , MaxSP5, MaxSP6, MaxSP7, MaxSP8, MaxSP9 As Decimal

```

B.2 Definição de botões que permitem mostrar e ocultar gráficos

```

1
2
3
4 Private Sub ButtonShowPlantarHide_Click(sender As Object, e As
    EventArgs) Handles ButtonShowPlantarHide.Click
5     Chart7S1.Visible = False
6     Chart8S2.Visible = False
7     Chart9S4.Visible = False
8     Chart10S3.Visible = False
9     Chart12S5.Visible = False
10    Chart13S6.Visible = False
11    Chart14S7.Visible = False
12    Chart15S8.Visible = False
13    Chart16S9.Visible = False
14    Chart11.Height = 317
15 End Sub
16
17 Private Sub ButtonShowPlantarS9_Click(sender As Object, e As
    EventArgs) Handles ButtonShowPlantarS9.Click
18    Chart7S1.Visible = False
19    Chart8S2.Visible = False
20    Chart9S4.Visible = False
21    Chart10S3.Visible = False
22    Chart12S5.Visible = False
23    Chart13S6.Visible = False
24    Chart14S7.Visible = False
25    Chart15S8.Visible = False

```

```
26         Chart16S9.Visible = True
27         Chart11.Height = 189
28     End Sub
29
30     Private Sub ButtonShowPlantarS8_Click(sender As Object, e As
31         EventArgs) Handles ButtonShowPlantarS8.Click
32         Chart7S1.Visible = False
33         Chart8S2.Visible = False
34         Chart9S4.Visible = False
35         Chart10S3.Visible = False
36         Chart12S5.Visible = False
37         Chart13S6.Visible = False
38         Chart14S7.Visible = False
39         Chart15S8.Visible = True
40         Chart16S9.Visible = False
41         Chart11.Height = 189
42     End Sub
43
44     Private Sub ButtonShowPlantarS7_Click(sender As Object, e As
45         EventArgs) Handles ButtonShowPlantarS7.Click
46         Chart7S1.Visible = False
47         Chart8S2.Visible = False
48         Chart9S4.Visible = False
49         Chart10S3.Visible = False
50         Chart12S5.Visible = False
51         Chart13S6.Visible = False
52         Chart14S7.Visible = True
53         Chart15S8.Visible = False
54         Chart16S9.Visible = False
55         Chart11.Height = 189
56     End Sub
57
58     Private Sub ButtonShowPlantarS6_Click(sender As Object, e As
59         EventArgs) Handles ButtonShowPlantarS6.Click
60         Chart7S1.Visible = False
61         Chart8S2.Visible = False
62         Chart9S4.Visible = False
63         Chart10S3.Visible = False
64         Chart12S5.Visible = False
65         Chart13S6.Visible = True
66         Chart14S7.Visible = False
67         Chart15S8.Visible = False
68         Chart16S9.Visible = False
69         Chart11.Height = 189
70     End Sub
71
72     Private Sub ButtonShowPlantarS5_Click(sender As Object, e As
73         EventArgs) Handles ButtonShowPlantarS5.Click
74         Chart7S1.Visible = False
```

```
71         Chart8S2.Visible = False
72         Chart9S4.Visible = False
73         Chart10S3.Visible = False
74         Chart12S5.Visible = True
75         Chart13S6.Visible = False
76         Chart14S7.Visible = False
77         Chart15S8.Visible = False
78         Chart16S9.Visible = False
79         Chart11.Height = 189
80     End Sub
81
82     Private Sub ButtonShowPlantarS4_Click(sender As Object, e As
83         EventArgs) Handles ButtonShowPlantarS4.Click
84         Chart7S1.Visible = False
85         Chart8S2.Visible = False
86         Chart9S4.Visible = True
87         Chart10S3.Visible = False
88         Chart12S5.Visible = False
89         Chart13S6.Visible = False
90         Chart14S7.Visible = False
91         Chart15S8.Visible = False
92         Chart16S9.Visible = False
93         Chart11.Height = 189
94     End Sub
95
96     Private Sub ButtonShowPlantarS3_Click(sender As Object, e As
97         EventArgs) Handles ButtonShowPlantarS3.Click
98         Chart7S1.Visible = False
99         Chart8S2.Visible = False
100        Chart9S4.Visible = False
101        Chart10S3.Visible = True
102        Chart12S5.Visible = False
103        Chart13S6.Visible = False
104        Chart14S7.Visible = False
105        Chart15S8.Visible = False
106        Chart16S9.Visible = False
107        Chart11.Height = 189
108    End Sub
109
110     Private Sub ButtonShowPlantarS2_Click(sender As Object, e As
111         EventArgs) Handles ButtonShowPlantarS2.Click
112         Chart7S1.Visible = False
113         Chart8S2.Visible = True
114         Chart9S4.Visible = False
115         Chart10S3.Visible = False
116         Chart12S5.Visible = False
117         Chart13S6.Visible = False
118         Chart14S7.Visible = False
119         Chart15S8.Visible = False
```

```
117         Chart16S9.Visible = False
118         Chart11.Height = 189
119     End Sub
120
121     Private Sub ButtonShowPlantarS1_Click(sender As Object, e As
        EventArgs) Handles ButtonShowPlantarS1.Click
122         Chart7S1.Visible = True
123         Chart8S2.Visible = False
124         Chart9S4.Visible = False
125         Chart10S3.Visible = False
126         Chart12S5.Visible = False
127         Chart13S6.Visible = False
128         Chart14S7.Visible = False
129         Chart15S8.Visible = False
130         Chart16S9.Visible = False
131         Chart11.Height = 189
132     End Sub
133
134     Private Sub ButtonShowGaspeaS1_Click(sender As Object, e As
        EventArgs) Handles ButtonShowGaspeaS1.Click
135         PanelConnection.Focus()
136         Chart1.Visible = True
137         Chart2.Visible = False
138         Chart3.Visible = False
139         Chart4.Visible = False
140         Chart5.Height = 115
141     End Sub
142
143     Private Sub ButtonShowGaspeaS2_Click(sender As Object, e As
        EventArgs) Handles ButtonShowGaspeaS2.Click
144         PanelConnection.Focus()
145         Chart1.Visible = False
146         Chart2.Visible = True
147         Chart3.Visible = False
148         Chart4.Visible = False
149         Chart5.Height = 115
150     End Sub
151
152     Private Sub ButtonShowGaspeaS3_Click(sender As Object, e As
        EventArgs) Handles ButtonShowGaspeaS3.Click
153         PanelConnection.Focus()
154         Chart1.Visible = False
155         Chart2.Visible = False
156         Chart3.Visible = True
157         Chart4.Visible = False
158         Chart5.Height = 115
159     End Sub
160
```

```
161 Private Sub ButtonShowGaspeaS4_Click(sender As Object, e As
      EventArgs) Handles ButtonShowGaspeaS4.Click
162     PanelConnection.Focus()
163     Chart1.Visible = False
164     Chart2.Visible = False
165     Chart3.Visible = False
166     Chart4.Visible = True
167     Chart5.Height = 115
168 End Sub
169
170 Private Sub ButtonHideGaspea_Click(sender As Object, e As
      EventArgs) Handles ButtonHideGaspea.Click
171     PanelConnection.Focus()
172     Chart1.Visible = False
173     Chart2.Visible = False
174     Chart3.Visible = False
175     Chart4.Visible = False
176     Chart5.Height = 267
177 End Sub
```

B.3 Configuração dos gráficos

```
1
2 Private Sub Form1_Load(sender As Object, e As EventArgs)
      Handles MyBase.Load
3     Me.CenterToScreen()
4     PanelConnection.Focus()
5     ComboBoxBaudRate.SelectedIndex = 5
6
7     For i = 0 To 30 Step 1
8         Chart1.Series("Sensor 1").Points.AddY(0)
9         If Chart1.Series(0).Points.Count = ChartLimit Then
10            Chart1.Series(0).Points.RemoveAt(0)
11        End If
12
13        Chart2.Series("Sensor 2").Points.AddY(0)
14        If Chart2.Series(0).Points.Count = ChartLimit Then
15            Chart2.Series(0).Points.RemoveAt(0)
16        End If
17
18        Chart3.Series("Sensor 3").Points.AddY(0)
19        If Chart3.Series(0).Points.Count = ChartLimit Then
20            Chart3.Series(0).Points.RemoveAt(0)
21        End If
22
23        Chart4.Series("Sensor 4").Points.AddY(0)
```

```
24         If Chart4.Series(0).Points.Count = ChartLimit Then
25             Chart4.Series(0).Points.RemoveAt(0)
26         End If
27
28
29         Chart5.Series("Sensor 1").Points.AddXY(DateTime.Now.
30             ToLongTimeString, 0)
31         If Chart5.Series(0).Points.Count = ChartLimit Then
32             Chart5.Series(0).Points.RemoveAt(0)
33         End If
34
35         Chart5.Series("Sensor 2").Points.AddY(0)
36         If Chart5.Series(1).Points.Count = ChartLimit Then
37             Chart5.Series(1).Points.RemoveAt(0)
38         End If
39
40         Chart5.Series("Sensor 3").Points.AddY(0)
41         If Chart5.Series(2).Points.Count = ChartLimit Then
42             Chart5.Series(2).Points.RemoveAt(0)
43         End If
44
45         Chart5.Series("Sensor 4").Points.AddY(0)
46         If Chart5.Series(3).Points.Count = ChartLimit Then
47             Chart5.Series(3).Points.RemoveAt(0)
48         End If
49
50         'Plantar Sensors Graphs'
51         Chart7S1.Series("Sensor 1").Points.AddY(0)
52         If Chart7S1.Series(0).Points.Count = ChartLimit Then
53             Chart7S1.Series(0).Points.RemoveAt(0)
54         End If
55
56         Chart8S2.Series("Sensor 2").Points.AddY(0)
57         If Chart8S2.Series(0).Points.Count = ChartLimit Then
58             Chart8S2.Series(0).Points.RemoveAt(0)
59         End If
60
61         Chart10S3.Series("Sensor 3").Points.AddY(0)
62         If Chart10S3.Series(0).Points.Count = ChartLimit Then
63             Chart10S3.Series(0).Points.RemoveAt(0)
64         End If
65
66         Chart9S4.Series("Sensor 4").Points.AddY(0)
67         If Chart9S4.Series(0).Points.Count = ChartLimit Then
68             Chart9S4.Series(0).Points.RemoveAt(0)
69         End If
70
71         Chart12S5.Series("Sensor 5").Points.AddY(0)
```

```
72     If Chart12S5.Series(0).Points.Count = ChartLimit Then
73         Chart12S5.Series(0).Points.RemoveAt(0)
74     End If
75
76     Chart13S6.Series("Sensor 6").Points.AddY(0)
77     If Chart13S6.Series(0).Points.Count = ChartLimit Then
78         Chart13S6.Series(0).Points.RemoveAt(0)
79     End If
80
81     Chart14S7.Series("Sensor 7").Points.AddY(0)
82     If Chart14S7.Series(0).Points.Count = ChartLimit Then
83         Chart14S7.Series(0).Points.RemoveAt(0)
84     End If
85
86     Chart15S8.Series("Sensor 8").Points.AddY(0)
87     If Chart15S8.Series(0).Points.Count = ChartLimit Then
88         Chart15S8.Series(0).Points.RemoveAt(0)
89     End If
90
91     Chart16S9.Series("Sensor 9").Points.AddY(0)
92     If Chart16S9.Series(0).Points.Count = ChartLimit Then
93         Chart16S9.Series(0).Points.RemoveAt(0)
94     End If
95
96
97     Chart6.Series("Sensor 1").Points.AddXY(DateTime.Now.
98         ToLongTimeString, 0)
99     If Chart6.Series(0).Points.Count = ChartLimit Then
100         Chart6.Series(0).Points.RemoveAt(0)
101     End If
102
103     Chart6.Series("Sensor 2").Points.AddY(0)
104     If Chart6.Series(1).Points.Count = ChartLimit Then
105         Chart6.Series(1).Points.RemoveAt(0)
106     End If
107
108     Chart6.Series("Sensor 3").Points.AddY(0)
109     If Chart6.Series(2).Points.Count = ChartLimit Then
110         Chart6.Series(2).Points.RemoveAt(0)
111     End If
112
113     Chart6.Series("Sensor 4").Points.AddY(0)
114     If Chart6.Series(3).Points.Count = ChartLimit Then
115         Chart6.Series(3).Points.RemoveAt(0)
116     End If
117
118     Chart6.Series("Sensor 5").Points.AddY(0)
119     If Chart6.Series(4).Points.Count = ChartLimit Then
120         Chart6.Series(4).Points.RemoveAt(0)
```

```
120         End If
121
122         Chart6.Series("Sensor 6").Points.AddY(0)
123         If Chart6.Series(5).Points.Count = ChartLimit Then
124             Chart6.Series(5).Points.RemoveAt(0)
125         End If
126
127         Chart6.Series("Sensor 7").Points.AddY(0)
128         If Chart6.Series(6).Points.Count = ChartLimit Then
129             Chart6.Series(6).Points.RemoveAt(0)
130         End If
131         Chart6.Series("Sensor 8").Points.AddY(0)
132         If Chart6.Series(7).Points.Count = ChartLimit Then
133             Chart6.Series(7).Points.RemoveAt(0)
134         End If
135
136         Chart6.Series("Sensor 9").Points.AddY(0)
137         If Chart6.Series(8).Points.Count = ChartLimit Then
138             Chart6.Series(8).Points.RemoveAt(0)
139         End If
140
141         'sum of all plantar forces'
142         Chart11.Series("Sum all").Points.AddY(0)
143         If Chart11.Series(0).Points.Count = ChartLimit Then
144             Chart11.Series(0).Points.RemoveAt(0)
145         End If
146     Next
147
148     Chart1.ChartAreas(0).AxisY.Maximum = 10
149     Chart1.ChartAreas(0).AxisY.Minimum = 0
150     Chart1.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
151         False
152     Chart2.ChartAreas(0).AxisY.Maximum = 10
153     Chart2.ChartAreas(0).AxisY.Minimum = 0
154     Chart2.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
155         False
156     Chart3.ChartAreas(0).AxisY.Maximum = 10
157     Chart3.ChartAreas(0).AxisY.Minimum = 0
158     Chart3.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
159         False
160     Chart4.ChartAreas(0).AxisY.Maximum = 10
161     Chart4.ChartAreas(0).AxisY.Minimum = 0
162     Chart4.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
163         False
164     Chart5.ChartAreas(0).AxisY.Maximum = 10
165     Chart5.ChartAreas(0).AxisY.Minimum = 0
166     Chart5.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
167         False
168     Chart1.Visible = False
```

```
164     Chart2.Visible = False
165     Chart3.Visible = False
166     Chart4.Visible = False
167     Chart5.Height = 267
168
169     Chart6.ChartAreas(0).AxisY.Maximum = 80
170     Chart6.ChartAreas(0).AxisY.Minimum = 0
171     Chart6.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled =
        False
172     Chart7S1.ChartAreas(0).AxisY.Maximum = 80
173     Chart7S1.ChartAreas(0).AxisY.Minimum = 0
174     Chart7S1.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled
        = False
175     Chart8S2.ChartAreas(0).AxisY.Maximum = 80
176     Chart8S2.ChartAreas(0).AxisY.Minimum = 0
177     Chart8S2.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled
        = False
178     Chart9S4.ChartAreas(0).AxisY.Maximum = 80
179     Chart9S4.ChartAreas(0).AxisY.Minimum = 0
180     Chart9S4.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled
        = False
181     Chart10S3.ChartAreas(0).AxisY.Maximum = 80
182     Chart10S3.ChartAreas(0).AxisY.Minimum = 0
183     Chart10S3.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
184     Chart12S5.ChartAreas(0).AxisY.Maximum = 80
185     Chart12S5.ChartAreas(0).AxisY.Minimum = 0
186     Chart12S5.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
187     Chart13S6.ChartAreas(0).AxisY.Maximum = 80
188     Chart13S6.ChartAreas(0).AxisY.Minimum = 0
189     Chart13S6.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
190     Chart14S7.ChartAreas(0).AxisY.Maximum = 80
191     Chart14S7.ChartAreas(0).AxisY.Minimum = 0
192     Chart14S7.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
193     Chart15S8.ChartAreas(0).AxisY.Maximum = 80
194     Chart15S8.ChartAreas(0).AxisY.Minimum = 0
195     Chart15S8.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
196     Chart16S9.ChartAreas(0).AxisY.Maximum = 80
197     Chart16S9.ChartAreas(0).AxisY.Minimum = 0
198     Chart16S9.ChartAreas("ChartArea1").AxisX.LabelStyle.
        Enabled = False
199
200     Chart11.ChartAreas(0).AxisY.Maximum = 800
201     Chart11.ChartAreas(0).AxisY.Minimum = 0
```

```
202         Chart11.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled
           = False
203
204
205         Chart7S1.Visible = False
206         Chart8S2.Visible = False
207         Chart9S4.Visible = False
208         Chart10S3.Visible = False
209         Chart12S5.Visible = False
210         Chart13S6.Visible = False
211         Chart14S7.Visible = False
212         Chart15S8.Visible = False
213         Chart16S9.Visible = False
214         Chart11.Height = 317
215
216     End Sub
```

B.4 Menu Connection

```
1     Private Sub ComboBoxBaudRate_SelectedIndexChanged(sender As
           Object, e As EventArgs) Handles ComboBoxBaudRate.
           SelectedIndexChanged
2           PanelConnection.Focus()
3     End Sub
4
5
6     Private Sub ComboBoxBaudRate_DropDown(sender As Object, e As
           EventArgs) Handles ComboBoxBaudRate.DropDown
7           PanelConnection.Focus()
8     End Sub
9
10
11    Private Sub ComboBoxBaudRate_Click(sender As Object, e As
           EventArgs) Handles ComboBoxBaudRate.Click
12        If LabelStatus.Text = "Status : Connected" Then
13            MsgBox("Connction in progress, please Disconnect to
                   change Baud Rate.", MsgBoxStyle.Critical, "Warning
                   !!!")
14            Return
15        End If
16    End Sub
17
18
19    Private Sub ComboBoxPort_SelectedIndexChanged(sender As Object
           , e As EventArgs) Handles ComboBoxPort.SelectedIndexChanged
20        PanelConnection.Focus()
```

```
21     End Sub
22
23     Private Sub ComboBoxPort_DropDown(sender As Object, e As
24         EventArgs) Handles ComboBoxPort.DropDown
25         PanelConnection.Focus()
26     End Sub
27
28     Private Sub ComboBoxPort_Click(sender As Object, e As
29         EventArgs) Handles ComboBoxPort.Click
30         If LabelStatus.Text = "Status : Connected" Then
31             MsgBox("Connction in progress, please Disconnect to
32                 change COM.", MsgBoxStyle.Critical, "Warning !!!")
33             Return
34         End If
35     End Sub
36
37     Private Sub ButtonScanPort_Click(sender As Object, e As
38         EventArgs) Handles ButtonScanPort.Click
39         PanelConnection.Focus()
40         If LabelStatus.Text = "Status : Connected" Then
41             MsgBox("Connction in progress, please Disconnect to
42                 scan the new port.", MsgBoxStyle.Critical, "Warning
43                 !!!")
44             Return
45         End If
46         ComboBoxPort.Items.Clear()
47         Dim myPort As Array
48         Dim i As Integer
49         myPort = IO.Ports.SerialPort.GetPortNames()
50         ComboBoxPort.Items.AddRange(myPort)
51         i = ComboBoxPort.Items.Count
52         i = i - 1
53         Try
54             ComboBoxPort.SelectedIndex = i
55             ButtonConnect.Enabled = True
56         Catch ex As Exception
57             MsgBox("Com port not detected", MsgBoxStyle.Critical,
58                 "Warning !!!")
59             ComboBoxPort.Text = ""
60             ComboBoxPort.Items.Clear()
61             Return
62         End Try
63         ComboBoxPort.DroppedDown = True
64     End Sub
65
```

```
62 Private Sub ButtonConnect_Click(sender As Object, e As
    EventArgs) Handles ButtonConnect.Click
63     PanelConnection.Focus()
64     Try
65         SerialPort1.BaudRate = ComboBoxBaudRate.SelectedItem
66         SerialPort1.PortName = ComboBoxPort.SelectedItem
67         SerialPort1.Open()
68         TimerSerial.Start()
69
70         LabelStatus.Text = "Status : Connected"
71         ButtonConnect.SendToBack()
72         ButtonDisconnect.BringToFront()
73         PictureBoxStatusConnection.BackColor = Color.Green
74     Catch ex As Exception
75         MsgBox("Please check the Hardware, COM, Baud Rate and
            try again.", MsgBoxStyle.Critical, "Connection
            failed !!!")
76     End Try
77 End Sub
78
79
80 Private Sub ButtonDisconnect_Click(sender As Object, e As
    EventArgs) Handles ButtonDisconnect.Click
81     PanelConnection.Focus()
82     TimerSerial.Stop()
83     SerialPort1.Close()
84     ButtonDisconnect.SendToBack()
85     ButtonConnect.BringToFront()
86     LabelStatus.Text = "Status : Disconnect"
87     PictureBoxStatusConnection.Visible = True
88     PictureBoxStatusConnection.BackColor = Color.Red
89 End Sub
```

B.5 Menu Data Record

```
1 Private Sub ButtonStartRecording_Click(sender As Object, e As
    EventArgs) Handles ButtonStartRecording.Click
2     ButtonStartRecording.Enabled = False
3     ButtonStopRecording.Enabled = True
4     ButtonSaveToExcel.Enabled = False
5     TimerDataLogRecord.Start()
6     ButtonStartRecording.SendToBack()
7     ButtonStopRecording.BringToFront()
8 End Sub
9
10
```

```
11 Private Sub ButtonStopRecording_Click(sender As Object, e As
    EventArgs) Handles ButtonStopRecording.Click
12     ButtonStartRecording.Enabled = True
13     ButtonStopRecording.Enabled = False
14     ButtonSaveToExcel.Enabled = True
15     TimerDataLogRecord.Stop()
16     ButtonStopRecording.SendToBack()
17     ButtonStartRecording.BringToFront()
18 End Sub
19
20
21 Private Sub ButtonClear_Click(sender As Object, e As EventArgs
    ) Handles ButtonClear.Click
22     For i = 0 To 30 Step 1
23         Chart1.Series("Sensor 1").Points.AddY(0)
24         If Chart1.Series(0).Points.Count = ChartLimit Then
25             Chart1.Series(0).Points.RemoveAt(0)
26         End If
27
28         Chart2.Series("Sensor 2").Points.AddY(0)
29         If Chart2.Series(0).Points.Count = ChartLimit Then
30             Chart2.Series(0).Points.RemoveAt(0)
31         End If
32
33         Chart3.Series("Sensor 3").Points.AddY(0)
34         If Chart3.Series(0).Points.Count = ChartLimit Then
35             Chart3.Series(0).Points.RemoveAt(0)
36         End If
37
38         Chart4.Series("Sensor 4").Points.AddY(0)
39         If Chart4.Series(0).Points.Count = ChartLimit Then
40             Chart4.Series(0).Points.RemoveAt(0)
41         End If
42
43
44         Chart5.Series("Sensor 1").Points.AddXY(DateTime.Now.
            ToLongTimeString, 0)
45         If Chart5.Series(0).Points.Count = ChartLimit Then
46             Chart5.Series(0).Points.RemoveAt(0)
47         End If
48
49         Chart5.Series("Sensor 2").Points.AddY(0)
50         If Chart5.Series(1).Points.Count = ChartLimit Then
51             Chart5.Series(1).Points.RemoveAt(0)
52         End If
53
54         Chart5.Series("Sensor 3").Points.AddY(0)
55         If Chart5.Series(2).Points.Count = ChartLimit Then
56             Chart5.Series(2).Points.RemoveAt(0)
```

```
57         End If
58
59         Chart5.Series("Sensor 4").Points.AddY(0)
60         If Chart5.Series(3).Points.Count = ChartLimit Then
61             Chart5.Series(3).Points.RemoveAt(0)
62         End If
63
64
65         'Plantar Sensors Graphs'
66         Chart7S1.Series("Sensor 1").Points.AddY(0)
67         If Chart7S1.Series(0).Points.Count = ChartLimit Then
68             Chart7S1.Series(0).Points.RemoveAt(0)
69         End If
70
71         Chart8S2.Series("Sensor 2").Points.AddY(0)
72         If Chart8S2.Series(0).Points.Count = ChartLimit Then
73             Chart8S2.Series(0).Points.RemoveAt(0)
74         End If
75
76         Chart10S3.Series("Sensor 3").Points.AddY(0)
77         If Chart10S3.Series(0).Points.Count = ChartLimit Then
78             Chart10S3.Series(0).Points.RemoveAt(0)
79         End If
80
81         Chart9S4.Series("Sensor 4").Points.AddY(0)
82         If Chart9S4.Series(0).Points.Count = ChartLimit Then
83             Chart9S4.Series(0).Points.RemoveAt(0)
84         End If
85
86         Chart12S5.Series("Sensor 5").Points.AddY(0)
87         If Chart12S5.Series(0).Points.Count = ChartLimit Then
88             Chart12S5.Series(0).Points.RemoveAt(0)
89         End If
90
91         Chart13S6.Series("Sensor 6").Points.AddY(0)
92         If Chart13S6.Series(0).Points.Count = ChartLimit Then
93             Chart13S6.Series(0).Points.RemoveAt(0)
94         End If
95
96         Chart14S7.Series("Sensor 7").Points.AddY(0)
97         If Chart14S7.Series(0).Points.Count = ChartLimit Then
98             Chart14S7.Series(0).Points.RemoveAt(0)
99         End If
100
101         Chart15S8.Series("Sensor 8").Points.AddY(0)
102         If Chart15S8.Series(0).Points.Count = ChartLimit Then
103             Chart15S8.Series(0).Points.RemoveAt(0)
104         End If
105
```

```
106     Chart16S9.Series("Sensor 9").Points.AddY(0)
107     If Chart16S9.Series(0).Points.Count = ChartLimit Then
108         Chart16S9.Series(0).Points.RemoveAt(0)
109     End If
110
111
112     Chart6.Series("Sensor 1").Points.AddXY(DateTime.Now.
        ToLongTimeString, 0)
113     If Chart6.Series(0).Points.Count = ChartLimit Then
114         Chart6.Series(0).Points.RemoveAt(0)
115     End If
116
117     Chart6.Series("Sensor 2").Points.AddY(0)
118     If Chart6.Series(1).Points.Count = ChartLimit Then
119         Chart6.Series(1).Points.RemoveAt(0)
120     End If
121
122     Chart6.Series("Sensor 3").Points.AddY(0)
123     If Chart6.Series(2).Points.Count = ChartLimit Then
124         Chart6.Series(2).Points.RemoveAt(0)
125     End If
126
127     Chart6.Series("Sensor 4").Points.AddY(0)
128     If Chart6.Series(3).Points.Count = ChartLimit Then
129         Chart6.Series(3).Points.RemoveAt(0)
130     End If
131
132     Chart6.Series("Sensor 5").Points.AddY(0)
133     If Chart6.Series(4).Points.Count = ChartLimit Then
134         Chart6.Series(4).Points.RemoveAt(0)
135     End If
136
137     Chart6.Series("Sensor 6").Points.AddY(0)
138     If Chart6.Series(5).Points.Count = ChartLimit Then
139         Chart6.Series(5).Points.RemoveAt(0)
140     End If
141
142     Chart6.Series("Sensor 7").Points.AddY(0)
143     If Chart6.Series(6).Points.Count = ChartLimit Then
144         Chart6.Series(6).Points.RemoveAt(0)
145     End If
146     Chart6.Series("Sensor 8").Points.AddY(0)
147     If Chart6.Series(7).Points.Count = ChartLimit Then
148         Chart6.Series(7).Points.RemoveAt(0)
149     End If
150
151     Chart6.Series("Sensor 9").Points.AddY(0)
152     If Chart6.Series(8).Points.Count = ChartLimit Then
153         Chart6.Series(8).Points.RemoveAt(0)
```

```
154         End If
155
156         'sum of all plantar forces'
157         Chart11.Series("Sum all").Points.AddY(0)
158         If Chart11.Series(0).Points.Count = ChartLimit Then
159             Chart11.Series(0).Points.RemoveAt(0)
160         End If
161
162     Next
163     DataGridView1.Rows.Clear()
164     MaxS1 = 0.00
165     MaxS2 = 0.00
166     MaxS3 = 0.00
167     MaxS4 = 0.00
168     MaxSP1 = 0.00
169     MaxSP2 = 0.00
170     MaxSP3 = 0.00
171     MaxSP4 = 0.00
172     MaxSP5 = 0.00
173     MaxSP6 = 0.00
174     MaxSP7 = 0.00
175     MaxSP8 = 0.00
176     MaxSP9 = 0.00
177
178 End Sub
```

B.5.1 Preenchimento da tabela

```
1
2 Private Sub TimerDataLogRecord_Tick(sender As Object, e As
3     EventArgs) Handles TimerDataLogRecord.Tick
4
5     Dim DT As DateTime = Now
6
7     DataGridView1.Rows.Add(New String() {DataGridView1.
8         RowCount, DT.ToString("hh:mm:ss.fff tt"), S1Result,
9         S2Result, S3Result, S4Result, SP1Result, SP2Result,
10        SP3Result, SP4Result, SP5Result, SP6Result, SP7Result,
11        SP8Result, SP9Result, AXResult, AYResult, AZResult,
12        GXResult, GYResult, GZResult})
13
14     Me.DataGridView1.FirstDisplayedScrollingRowIndex = Me.
15     DataGridView1.RowCount - 1
16
17 End Sub
```

B.5.2 Guardar dados no ficheiro Excel

```
1
2 Private Sub ButtonSaveToExcel_Click(sender As Object, e As
3     EventArgs) Handles ButtonSaveToExcel.Click
4     ButtonSaveToExcel.Height = 25
5     ButtonSaveToExcel.Text = "Please Wait..."
6     ButtonSaveToExcel.Enabled = False
7     ButtonStartRecording.Enabled = False
8     ProgressBarProcess.Visible = True
9     ProgressBarProcess.Value = 1
10
11 Dim xlApp As Microsoft.Office.Interop.Excel.Application
12 Dim xlWorkBook As Microsoft.Office.Interop.Excel.Workbook
13 Dim xlWorkSheet As Microsoft.Office.Interop.Excel.
14     Worksheet
15 Dim misValue As Object = System.Reflection.Missing.Value
16 Dim i As Integer
17 Dim j As Integer
18
19 ProgressBarProcess.Value = 3
20
21 xlApp = New Microsoft.Office.Interop.Excel.Application
22 xlWorkBook = xlApp.Workbooks.Add(misValue)
23 xlWorkSheet = xlWorkBook.Sheets("Folha1")
24
25 ProgressBarProcess.Value = 5
26
27 For k As Integer = 1 To DataGridView1.Columns.Count
28     xlWorkSheet.Cells(1, k) = DataGridView1.Columns(k - 1)
29     .HeaderText
30 Next
31
32 For i = 0 To DataGridView1.RowCount - 2
33     For j = 0 To DataGridView1.ColumnCount - 1
34         xlWorkSheet.Cells(i + 2, j + 1) = DataGridView1(j,
35             i).Value.ToString()
36     Next
37 Next
38
39 ProgressBarProcess.Value = 8
40
41 FilePathAndName = Application.StartupPath & "\" & Now.Day
42     & "-" & Now.Hour & "-" & Now.Minute & ".xlsx"
43 If File.Exists(FilePathAndName) Then File.Delete(
44     FilePathAndName)
```

```
41
42     xlWorkSheet.SaveAs(FilePathAndName)
43     xlWorkBook.Close()
44     xlApp.Quit()
45
46     releaseObject(xlApp)
47     releaseObject(xlWorkBook)
48     releaseObject(xlWorkSheet)
49
50     ProgressBarProcess.Value = 10
51
52     MsgBox("Successfully saved" & vbCrLf & "File are saved at
53         : " & FilePathAndName, MsgBoxStyle.Information, "
54         Information")
55
56     ProgressBarProcess.Visible = False
57
58     Process.Start(FilePathAndName)
59
60     ButtonSaveToExcel.Height = 30
61     ButtonSaveToExcel.Text = "Save To MS Excel"
62     ButtonSaveToExcel.Enabled = True
63     ButtonStartRecording.Enabled = True
64 End Sub
65
66 Private Sub releaseObject(ByVal obj As Object)
67     Try
68         System.Runtime.InteropServices.Marshal.
69             ReleaseComObject(obj)
70         obj = Nothing
71     Catch ex As Exception
72         obj = Nothing
73     Finally
74         GC.Collect()
75     End Try
76 End Sub
```

B.6 Recepção e tratamento dos dados

```
1     Private Sub TimerSerial_Tick(sender As Object, e As EventArgs)
2         Handles TimerSerial.Tick
3
4         Try
5             StrSerialIn = SerialPort1.ReadExisting '--> Read
6                 incoming serial data'
```

```
6 Dim TB As New TextBox
7 TB.Multiline = True
8 TB.Text = StrSerialIn '--> Enter serial data into
   the textbox'
9
10 If TB.Lines.Count > 0 Then
11     If TB.Lines(0) = "Failed to read sensors!" Then
12         '--> Check Arduino IDE if it fails to read the
13         sensors, if this happens the connection is
14         disconnected'
15         TimerSerial.Stop()
16         SerialPort1.Close()
17         LabelStatus.Text = "Status : Disconnect"
18         ButtonDisconnect.SendToBack()
19         ButtonConnect.BringToFront()
20         PictureBoxStatusConnection.Visible = True
21         PictureBoxStatusConnection.BackColor = Color.
           Red
22         MsgBox("Failed to read sensors !!!, Please
23         check the Hardware and Please connect again
24         .", MsgBoxStyle.Critical, "Connection
25         failed !!!")
26     Return
27 End If
28
29 StrSerialInRam = TB.Lines(0).Substring(0, 1)
30 If StrSerialInRam = "A" Then
31     S1 = TB.Lines(0)
32     S1L = S1.Length
33 Else
34     S1 = S1
35 End If
36
37 StrSerialInRam = TB.Lines(1).Substring(0, 1)
38 If StrSerialInRam = "B" Then
39     S2 = TB.Lines(1)
40     S2L = S2.Length
41 Else
42     S2 = S2
43 End If
44
45 StrSerialInRam = TB.Lines(2).Substring(0, 1)
46 If StrSerialInRam = "C" Then
47     S3 = TB.Lines(2)
48     S3L = S3.Length
49 Else
50     S3 = S3
51 End If
```

```
47
48     StrSerialInRam = TB.Lines(3).Substring(0, 1)
49     If StrSerialInRam = "D" Then
50         S4 = TB.Lines(3)
51         S4L = S4.Length
52     Else
53         S4 = S4
54     End If
55
56     StrSerialInRam = TB.Lines(4).Substring(0, 1)
57     If StrSerialInRam = "E" Then
58         SP1 = TB.Lines(4)
59         SP1L = SP1.Length
60     Else
61         SP1 = SP1
62     End If
63
64     StrSerialInRam = TB.Lines(5).Substring(0, 1)
65     If StrSerialInRam = "F" Then
66         SP2 = TB.Lines(5)
67         SP2L = SP2.Length
68     Else
69         SP2 = SP2
70     End If
71
72     StrSerialInRam = TB.Lines(6).Substring(0, 1)
73     If StrSerialInRam = "G" Then
74         SP3 = TB.Lines(6)
75         SP3L = SP3.Length
76     Else
77         SP3 = SP3
78     End If
79
80
81     StrSerialInRam = TB.Lines(7).Substring(0, 1)
82     If StrSerialInRam = "H" Then
83         SP4 = TB.Lines(7)
84         SP4L = SP4.Length
85     Else
86         SP4 = SP4
87     End If
88
89     StrSerialInRam = TB.Lines(8).Substring(0, 1)
90     If StrSerialInRam = "I" Then
91         SP5 = TB.Lines(8)
92         SP5L = SP5.Length
93     Else
94         SP5 = SP5
95     End If
```

```
96
97     StrSerialInRam = TB.Lines(9).Substring(0, 1)
98     If StrSerialInRam = "J" Then
99         SP6 = TB.Lines(9)
100        SP6L = SP6.Length
101     Else
102        SP6 = SP6
103     End If
104
105     StrSerialInRam = TB.Lines(10).Substring(0, 1)
106     If StrSerialInRam = "K" Then
107        SP7 = TB.Lines(10)
108        SP7L = SP7.Length
109     Else
110        SP7 = SP7
111     End If
112
113     StrSerialInRam = TB.Lines(11).Substring(0, 1)
114     If StrSerialInRam = "L" Then
115        SP8 = TB.Lines(11)
116        SP8L = SP8.Length
117     Else
118        SP8 = SP8
119     End If
120
121     StrSerialInRam = TB.Lines(12).Substring(0, 1)
122     If StrSerialInRam = "M" Then
123        SP9 = TB.Lines(12)
124        SP9L = SP9.Length
125     Else
126        SP9 = SP9
127     End If
128
129     StrSerialInRam = TB.Lines(13).Substring(0, 1)
130     If StrSerialInRam = "N" Then
131        AX = TB.Lines(13)
132        AXL = AX.Length
133     Else
134        AX = AX
135     End If
136
137     StrSerialInRam = TB.Lines(14).Substring(0, 1)
138     If StrSerialInRam = "O" Then
139        AY = TB.Lines(14)
140        AYL = AY.Length
141     Else
142        AY = AY
143     End If
144
```

```
145         StrSerialInRam = TB.Lines(15).Substring(0, 1)
146         If StrSerialInRam = "P" Then
147             AZ = TB.Lines(15)
148             AZL = AZ.Length
149         Else
150             AZ = AZ
151         End If
152
153         StrSerialInRam = TB.Lines(16).Substring(0, 1)
154         If StrSerialInRam = "Q" Then
155             GX = TB.Lines(16)
156             GXL = GX.Length
157         Else
158             GX = GX
159         End If
160
161         StrSerialInRam = TB.Lines(17).Substring(0, 1)
162         If StrSerialInRam = "R" Then
163             GY = TB.Lines(17)
164             GYL = GY.Length
165         Else
166             GY = GY
167         End If
168
169         StrSerialInRam = TB.Lines(18).Substring(0, 1)
170         If StrSerialInRam = "S" Then
171             GZ = TB.Lines(18)
172             GZL = GZ.Length
173         Else
174             GZ = GZ
175         End If
176
177
178         S1Result = Format((Val(Mid(S1, 2, S1L)) / 100),
179             "0.00")
180         S2Result = Format((Val(Mid(S2, 2, S2L)) / 100),
181             "0.00")
182         S3Result = Format((Val(Mid(S3, 2, S3L)) / 100),
183             "0.00")
184         S4Result = Format((Val(Mid(S4, 2, S4L)) / 100),
185             "0.00")
186
187         SP1Result = Format((Val(Mid(SP1, 2, SP1L)) / 100),
188             "0.00")
189         SP2Result = Format((Val(Mid(SP2, 2, SP2L)) / 100),
190             "0.00")
191         SP3Result = Format((Val(Mid(SP3, 2, SP3L)) / 100),
192             "0.00")
```

```

187         SP4Result = Format((Val(Mid(SP4, 2, SP4L)) / 100),
188             "0.00")
189         SP5Result = Format((Val(Mid(SP5, 2, SP5L)) / 100),
190             "0.00")
191         SP6Result = Format((Val(Mid(SP6, 2, SP6L)) / 100),
192             "0.00")
193         SP7Result = Format((Val(Mid(SP7, 2, SP7L)) / 100),
194             "0.00")
195         SP8Result = Format((Val(Mid(SP8, 2, SP8L)) / 100),
196             "0.00")
197         SP9Result = Format((Val(Mid(SP9, 2, SP9L)) / 100),
198             "0.00")
199
200         AXResult = Format((Val(Mid(AX, 2, AXL)) - 10000) /
201             1000, "0.00")
202         AYResult = Format((Val(Mid(AY, 2, AYL)) - 10000) /
203             1000, "0.00")
204         AZResult = Format((Val(Mid(AZ, 2, AZL)) - 10000) /
205             1000, "0.00")
206         GXResult = Format((Val(Mid(GX, 2, GXL)) - 10000) /
207             1000, "0.00")
208         GYResult = Format((Val(Mid(GY, 2, GYL)) - 10000) /
209             1000, "0.00")
210         GZResult = Format((Val(Mid(GZ, 2, GZL)) - 10000) /
211             1000, "0.00")

```

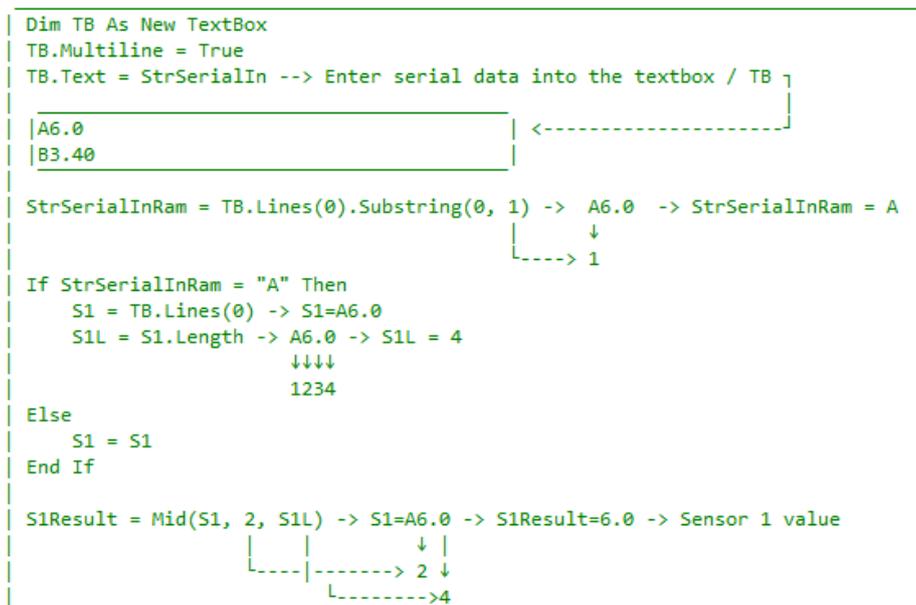


Figura B.1: Exemplo do funcionamento do algoritmo de desmultiplexagem dos dados enviados por porta série

```
1
2     If MaxS1 <= S1Result Then
3         MaxS1 = S1Result
4         LabelMaxS1.Text = "Max S1: " & MaxS1 & " N"
5
6     End If
7
8     If MaxS2 <= S2Result Then
9         MaxS2 = S2Result
10        LabelMaxS2.Text = "Max S2: " & MaxS2 & " N"
11    End If
12
13    If MaxS3 <= S3Result Then
14        MaxS3 = S3Result
15        LabelMaxS3.Text = "Max S3: " & MaxS3 & " N"
16    End If
17
18    If MaxS4 <= S4Result Then
19        MaxS4 = S4Result
20        LabelMaxS4.Text = "Max S4: " & MaxS4 & " N"
21    End If
22
23    If MaxSP1 <= SP1Result Then
24        MaxSP1 = SP1Result
25        LabelMaxPlanS1.Text = "Max S1: " & MaxSP1 & "
26        N"
27    End If
28
29    If MaxSP2 <= SP2Result Then
30        MaxSP2 = SP2Result
31        LabelMaxPlanS2.Text = "Max S2: " & MaxSP2 & "
32        N"
33    End If
34
35    If MaxSP3 <= SP3Result Then
36        MaxSP3 = SP3Result
37        LabelMaxPlanS3.Text = "Max S3: " & MaxSP3 & "
38        N"
39    End If
40
41    If MaxSP4 <= SP4Result Then
42        MaxSP4 = SP4Result
43        LabelMaxPlanS4.Text = "Max S4: " & MaxSP4 & "
44        N"
45    End If
46
47    If MaxSP5 <= SP5Result Then
48        MaxSP5 = SP5Result
```

```
45         LabelMaxPlanS5.Text = "Max S5: " & MaxSP5 & "  
46             N"  
47     End If  
48  
49     If MaxSP6 <= SP6Result Then  
50         MaxSP6 = SP6Result  
51         LabelMaxPlanS6.Text = "Max S6: " & MaxSP6 & "  
52             N"  
53     End If  
54  
55     If MaxSP7 <= SP7Result Then  
56         MaxSP7 = SP7Result  
57         LabelMaxPlanS7.Text = "Max S7: " & MaxSP7 & "  
58             N"  
59     End If  
60  
61     If MaxSP8 <= SP8Result Then  
62         MaxSP8 = SP8Result  
63         LabelMaxPlanS8.Text = "Max S8: " & MaxSP8 & "  
64             N"  
65     End If  
66  
67     If MaxSP9 <= SP9Result Then  
68         MaxSP9 = SP9Result  
69         LabelMaxPlanS9.Text = "Max S9: " & MaxSP9 & "  
70             N"  
71     End If  
72  
73     SensorSum = Val(SP1Result) + Val(SP2Result) + Val(  
74         SP3Result) + Val(SP4Result) + Val(SP5Result) +  
75         Val(SP6Result) + Val(SP7Result) + Val(SP8Result  
76         ) + Val(SP9Result)  
77  
78     '-----Enter the Sensors values into the  
79     chart-----'  
80     Chart1.Series("Sensor 1").Points.AddY(S1Result)  
81     If Chart1.Series(0).Points.Count = ChartLimit Then  
82         Chart1.Series(0).Points.RemoveAt(0)  
83     End If  
84  
85     Chart2.Series("Sensor 2").Points.AddY(S2Result)  
86     If Chart2.Series(0).Points.Count = ChartLimit Then  
87         Chart2.Series(0).Points.RemoveAt(0)  
88     End If  
89  
90     Chart3.Series("Sensor 3").Points.AddY(S3Result)  
91     If Chart3.Series(0).Points.Count = ChartLimit Then  
92         Chart3.Series(0).Points.RemoveAt(0)
```

```
85         End If
86
87         Chart4.Series("Sensor 4").Points.AddY(S4Result)
88         If Chart4.Series(0).Points.Count = ChartLimit Then
89             Chart4.Series(0).Points.RemoveAt(0)
90         End If
91
92         Chart5.Series("Sensor 1").Points.AddXY(DateTime.
93             Now.ToLongTimeString, S1Result)
94         If Chart5.Series(0).Points.Count = ChartLimit Then
95             Chart5.Series(0).Points.RemoveAt(0)
96         End If
97
98         Chart5.Series("Sensor 2").Points.AddY(S2Result)
99         If Chart5.Series(1).Points.Count = ChartLimit Then
100             Chart5.Series(1).Points.RemoveAt(0)
101         End If
102
103         Chart5.Series("Sensor 3").Points.AddY(S3Result)
104         If Chart5.Series(2).Points.Count = ChartLimit Then
105             Chart5.Series(2).Points.RemoveAt(0)
106         End If
107
108         Chart5.Series("Sensor 4").Points.AddY(S4Result)
109         If Chart5.Series(3).Points.Count = ChartLimit Then
110             Chart5.Series(3).Points.RemoveAt(0)
111         End If
112
113         'plantar Sensors'
114
115         Chart7S1.Series("Sensor 1").Points.AddY(SP1Result)
116         If Chart7S1.Series(0).Points.Count = ChartLimit
117             Then
118             Chart7S1.Series(0).Points.RemoveAt(0)
119         End If
120
121         Chart8S2.Series("Sensor 2").Points.AddY(SP2Result)
122         If Chart8S2.Series(0).Points.Count = ChartLimit
123             Then
124             Chart8S2.Series(0).Points.RemoveAt(0)
125         End If
126
127         Chart10S3.Series("Sensor 3").Points.AddY(SP3Result
128             )
129         If Chart10S3.Series(0).Points.Count = ChartLimit
130             Then
131             Chart10S3.Series(0).Points.RemoveAt(0)
132         End If
```

```
129     Chart9S4.Series("Sensor 4").Points.AddY(SP4Result)
130     If Chart9S4.Series(0).Points.Count = ChartLimit
131         Then
132             Chart9S4.Series(0).Points.RemoveAt(0)
133     End If
134
135     Chart12S5.Series("Sensor 5").Points.AddY(SP5Result
136         )
137     If Chart12S5.Series(0).Points.Count = ChartLimit
138         Then
139             Chart12S5.Series(0).Points.RemoveAt(0)
140     End If
141
142     Chart13S6.Series("Sensor 6").Points.AddY(SP6Result
143         )
144     If Chart13S6.Series(0).Points.Count = ChartLimit
145         Then
146             Chart13S6.Series(0).Points.RemoveAt(0)
147     End If
148
149     Chart14S7.Series("Sensor 7").Points.AddY(SP7Result
150         )
151     If Chart14S7.Series(0).Points.Count = ChartLimit
152         Then
153             Chart14S7.Series(0).Points.RemoveAt(0)
154     End If
155
156     Chart15S8.Series("Sensor 8").Points.AddY(SP8Result
157         )
158     If Chart15S8.Series(0).Points.Count = ChartLimit
159         Then
160             Chart15S8.Series(0).Points.RemoveAt(0)
161     End If
162
163     Chart16S9.Series("Sensor 9").Points.AddY(SP9Result
164         )
165     If Chart16S9.Series(0).Points.Count = ChartLimit
166         Then
167             Chart16S9.Series(0).Points.RemoveAt(0)
168     End If
169
170     Chart6.Series("Sensor 1").Points.AddXY(DateTime.
171         Now.ToLongTimeString, SP1Result)
172     If Chart6.Series(0).Points.Count = ChartLimit Then
173         Chart6.Series(0).Points.RemoveAt(0)
174     End If
175
176     Chart6.Series("Sensor 2").Points.AddY(SP2Result)
```

```
166         If Chart6.Series(1).Points.Count = ChartLimit Then
167             Chart6.Series(1).Points.RemoveAt(0)
168         End If
169
170         Chart6.Series("Sensor 3").Points.AddY(SP3Result)
171         If Chart6.Series(2).Points.Count = ChartLimit Then
172             Chart6.Series(2).Points.RemoveAt(0)
173         End If
174
175         Chart6.Series("Sensor 4").Points.AddY(SP4Result)
176         If Chart6.Series(3).Points.Count = ChartLimit Then
177             Chart6.Series(3).Points.RemoveAt(0)
178         End If
179
180         Chart6.Series("Sensor 5").Points.AddY(SP5Result)
181         If Chart6.Series(4).Points.Count = ChartLimit Then
182             Chart6.Series(4).Points.RemoveAt(0)
183         End If
184
185         Chart6.Series("Sensor 6").Points.AddY(SP6Result)
186         If Chart6.Series(5).Points.Count = ChartLimit Then
187             Chart6.Series(5).Points.RemoveAt(0)
188         End If
189
190         Chart6.Series("Sensor 7").Points.AddY(SP7Result)
191         If Chart6.Series(6).Points.Count = ChartLimit Then
192             Chart6.Series(6).Points.RemoveAt(0)
193         End If
194         Chart6.Series("Sensor 8").Points.AddY(SP8Result)
195         If Chart6.Series(7).Points.Count = ChartLimit Then
196             Chart6.Series(7).Points.RemoveAt(0)
197         End If
198
199         Chart6.Series("Sensor 9").Points.AddY(SP9Result)
200         If Chart6.Series(8).Points.Count = ChartLimit Then
201             Chart6.Series(8).Points.RemoveAt(0)
202         End If
203
204         'sum of all plantar forces'
205         Chart11.Series("Sum all").Points.AddY(SensorSum)
206         If Chart11.Series(0).Points.Count = ChartLimit
207             Then
208             Chart11.Series(0).Points.RemoveAt(0)
209         End If
210
211         LabelSensor1.Text = S1Result & " N"
212         LabelSensor2.Text = S2Result & " N"
213         LabelSensor3.Text = S3Result & " N"
214         LabelSensor4.Text = S4Result & " N"
```

```

214
215         LabelPlantarS1.Text = SP1Result & " N"
216         LabelPlantarS2.Text = SP2Result & " N"
217         LabelPlantarS3.Text = SP3Result & " N"
218         LabelPlantarS4.Text = SP4Result & " N"
219         LabelPlantarS5.Text = SP5Result & " N"
220         LabelPlantarS6.Text = SP6Result & " N"
221         LabelPlantarS7.Text = SP7Result & " N"
222         LabelPlantarS8.Text = SP8Result & " N"
223         LabelPlantarS9.Text = SP9Result & " N"
224
225         AcelX.Text = "AcelerometroX:" & AXResult & " m/s
226             ^2"
227         AcelY.Text = "AcelerometroY:" & AYResult & " m/s
228             ^2"
229         AcelZ.Text = "AcelerometroZ:" & AZResult & " m/s
230             ^2"
231
232         GirosX.Text = "GiroscopioX:" & GXResult & ""
233         GirosY.Text = "GiroscopioY:" & GYResult & ""
234         GirosZ.Text = "GiroscopioZ:" & GZResult & ""
235         '-----'
236
237         '-----If the Then connection Is successful
238         And running, PictureBoxStatusConnection will
239         blink----'
240
241         If PictureBoxStatusConnection.Visible = True Then
242             PictureBoxStatusConnection.Visible = False
243         ElseIf PictureBoxStatusConnection.Visible = False
244             Then
245                 PictureBoxStatusConnection.Visible = True
246             End If
247         '-----'
248
249     End If
250
251     Catch ex As Exception
252         TimerSerial.Stop()
253         SerialPort1.Close()
254         LabelStatus.Text = "Status : Disconnect"
255         ButtonDisconnect.SendToBack()
256         ButtonConnect.BringToFront()
257         PictureBoxStatusConnection.BackColor = Color.Red
258         MsgBox("Please check the Hardware and Please connect
259             again." & ex.Message, MsgBoxStyle.Critical, "
260             Connection failed !!!")
261
262     Return
263
264 End Try
265
266 End Sub
267
268 End Class

```

Anexo C

Blocos MiT App Inventor

C.1 Inicialização variáveis

```
initialize global Service_UUID to "d5466fb8-2b71-437a-949d-9e32a567ca2b"  
initialize global Characteristic_UUID_Gaspea to "03a69008-6a0e-4fc0-ad03-89b2fa20d0df"  
initialize global Characteristic_UUID_IMU to "3b6b8f75-6345-40cf-baf9-79ff19d0bd7a"  
initialize global Characteristic_UUID_Plantar to "2ff44774-809a-4785-83ff-a1b2c9fd5c5f"  
  
initialize global sensor1 to "0"  
initialize global sensor2 to "0"  
initialize global sensor3 to "0"  
initialize global sensor7 to "0"  
initialize global sensor9 to "0"  
  
initialize global Gsensor1 to "0"  
initialize global Gsensor3 to "0"  
  
initialize global SAceX to "0"  
initialize global SAceZ to "0"  
initialize global SgirosY to "0"  
  
initialize global sensor4 to "0"  
initialize global sensor5 to "0"  
initialize global sensor6 to "0"  
initialize global sensor8 to "0"  
  
initialize global Gsensor2 to "0"  
initialize global Gsensor4 to "0"  
  
initialize global SAceY to "0"  
initialize global SgirosX to "0"  
initialize global SgirosZ to "0"
```

C.2 Seleção e visualização de dispositivos BLE

```

when Screen1.Initialize
do
  call BluetoothLE1.StartScanning
  set Força.Enabled to true
  set Pressão.Enabled to false
  set Imageplantar.Visible to true
  set Imagegaspea.Visible to false
  set TableArrangement1.Visible to true
  set TableArrangement3.Visible to false
  set TableArrangement4.Visible to false
  set Gaspea.Visible to true
  set Plantar.Visible to false

when ListPicker1.AfterPicking
do
  call BluetoothLE1.Connect
  index ListPicker1.SelectionIndex

when BluetoothLE1.DeviceFound
do
  set ListPicker1.ElementsFromString to BluetoothLE1.DeviceList

when BluetoothLE1.Connected
do
  set ButtonConnect.Text to "Connected"
  set ButtonConnect.BackgroundColor to #00FF00
  set ButtonDisconnect.BackgroundColor to #FF0000
  call BluetoothLE1.StopScanning
  
```

C.3 Escuta por bytes recebidos por BLE

```

when Clock1.Timer
do
  if BluetoothLE1.IsDeviceConnected
  then
    call BluetoothLE1.RegisterForBytes
    serviceUuid get global Service_UUID
    characteristicUuid get global Characteristic_UUID_Gaspea
    signed true
    call BluetoothLE1.RegisterForBytes
    serviceUuid get global Service_UUID
    characteristicUuid get global Characteristic_UUID_Plantar
    signed true
    call BluetoothLE1.RegisterForBytes
    serviceUuid get global Service_UUID
    characteristicUuid get global Characteristic_UUID_IMU
    signed true
    set ButtonConnect.Text to "Connected"
  else
    set ButtonConnect.Text to "Connect"
  
```

C.4 Definições de botões e de procedimento de decodificação

```
when Gaspea . Click
do
  set Imageplantar . Visible to false
  set Imagegaspea . Visible to true
  set TableArrangement1 . Visible to false
  set TableArrangement3 . Visible to true
  set TableArrangement4 . Visible to true
  set Gaspea . Visible to false
  set Plantar . Visible to true
```

```
when Plantar . Click
do
  set Imageplantar . Visible to true
  set Imagegaspea . Visible to false
  set TableArrangement1 . Visible to true
  set TableArrangement3 . Visible to false
  set TableArrangement4 . Visible to false
  set Gaspea . Visible to true
  set Plantar . Visible to false
```

```
when Pressão . TouchDown
do
  set Força . Enabled to true
  set Pressão . Enabled to false
```

```
when Força . TouchDown
do
  set Pressão . Enabled to true
  set Força . Enabled to false
```

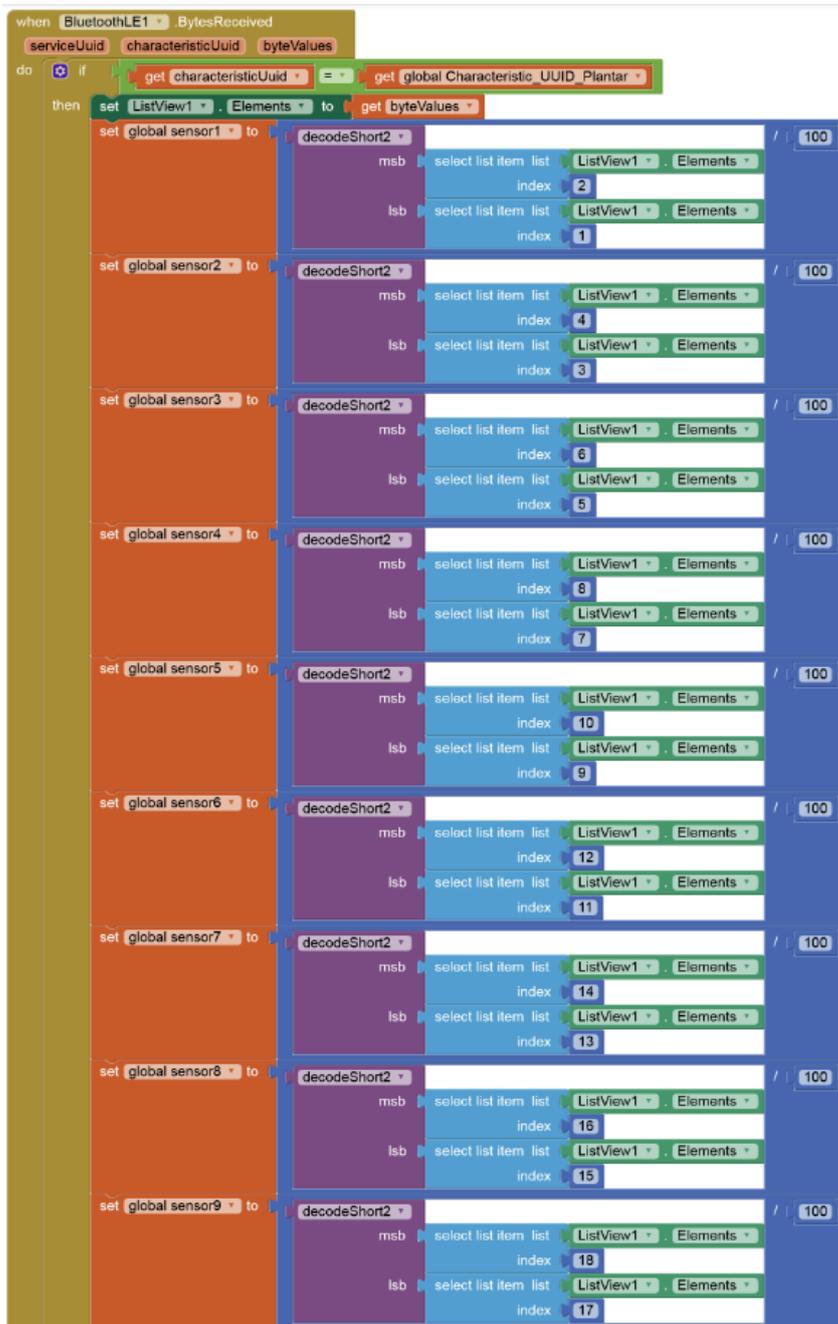
```
when BluetoothLE1 . Disconnected
do
  set ButtonConnect . Text to " Connect "
  set ButtonConnect . BackgroundColor to red
  set ButtonDisconnect . BackgroundColor to gray
```

```
when ButtonDisconnect . Click
do
  if ListPicker1 . SelectionIndex > 0
  then
    call BluetoothLE1 . DisconnectWithAddress
      address call BluetoothLE1 . FoundDeviceAddress
      index ListPicker1 . SelectionIndex
    set ButtonConnect . BackgroundColor to gray
    set ButtonConnect . Text to " Connect "
    set ButtonDisconnect . BackgroundColor to gray
    call BluetoothLE1 . StartScanning
```

```
to decodeShort2 msb lsb
result
  bitwise or
    get msb × 256
    bitwise and
      get lsb
      255
```

C.5 Tratamento dos bytes recebidos por BLE

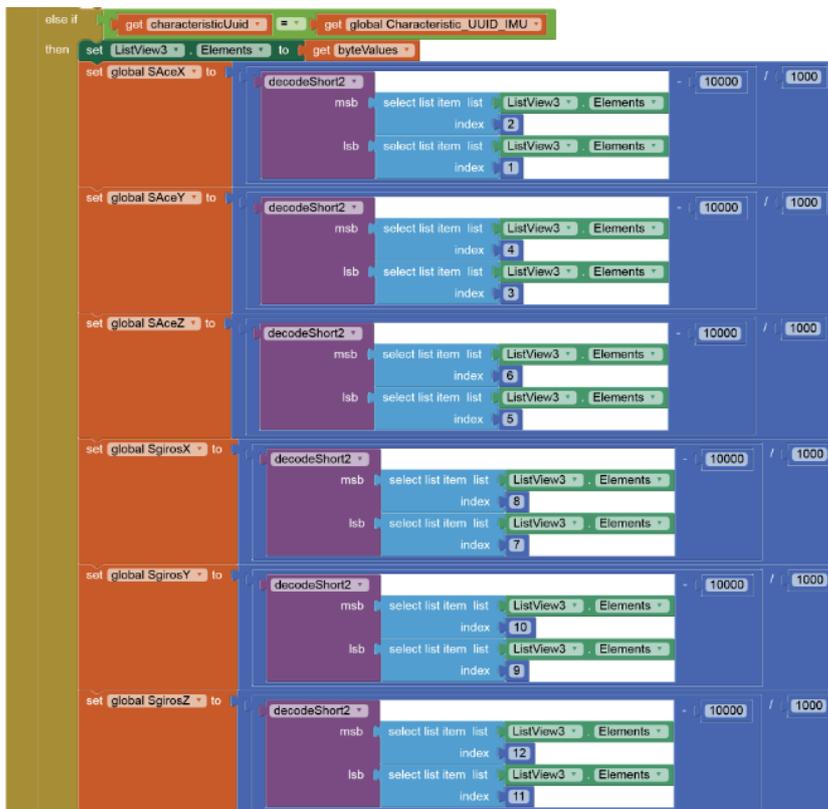
C.5.1 UUID Plantar



C.5.2 UUID Gaspea



C.5.3 UUID IMU



C.5.4 Cálculo e apresentação de valores dos sensores no ecrã

