



Mestrado em Engenharia Mecânica

Desenvolvimento de um Andarilho Instrumentado para apoio a Pacientes com Mobilidade Reduzida

Trabalho Projeto
apresentado para a obtenção do grau de Mestre em Engenharia Mecânica
Especialização em Construção e Manutenção de Equipamentos Mecânicos

Autor

Stéphane Pereira Lopes

Orientadores

Prof. Doutora Cândida Maria dos Santos Pereira Malça

Professora do Departamento de Engenharia Mecânica
Instituto Superior de Engenharia de Coimbra

Prof. Doutor Luis Manuel Ferreira Roseiro

Professor do Departamento de Engenharia Mecânica
Instituto Superior de Engenharia de Coimbra

Coimbra, Junho, 2020

Dedico este trabalho projeto à minha mãe, irmã e namorada

AGRADECIMENTOS

Ao professor e orientador Doutor Luís Manuel Ferreira Roseiro, pela sua orientação, por tornar este trabalho possível, pelas ideias e saberes que transmitiu e pelas contribuições para o trabalho.

À co-orientadora Doutora Cândida Malça, pela sua orientação, pelas opiniões e críticas e igualmente pelo seu apoio na elaboração deste trabalho projeto.

Ao Doutor Frederico Santos, pela total disponibilidade para auxiliar na elaboração da programação, solucionar problemas e dúvidas que foram surgindo.

À minha mãe, pelo enorme esforço e dedicação para tornar o meu percurso académico realidade.

À minha irmã, pela influência e inspiração, pelo apoio e dedicação.

À minha namorada Rita, por todas as palavras de incentivo, pela sua paciência e ajuda prestada durante a elaboração do presente trabalho projeto.

Aos meus amigos que contribuíram, de uma forma direta ou indiretamente, para elaboração deste presente trabalho, mas mais importante, obrigado por todos os momentos passados.

À empresa MDN – Serviços de Manutenção Industrial, mais propriamente ao meu amigo Eng. Kevin Neves pela ajuda na maquinação de peças em alumínio.

À empresa onde estou empregado, PRF – Gás, Tecnologia e Construção por toda a compreensão e flexibilidade de tempo para a realização deste trabalho.

A todos aqueles que ajudaram de livre e espontânea vontade.

RESUMO

As projeções feitas pelo Instituto Nacional de Estatística (INE) revelam um aumento crescente do envelhecimento da população Portuguesa. Uma das principais consequências do envelhecimento da população passa pelo aumento do nível de dependência dos idosos, havendo cada vez mais a necessidade de recorrer a dispositivos de compensação que lhes permitam uma vida mais ativa. Uma das principais limitações associada ao envelhecimento está relacionada com a locomoção, sendo o andarilho um dos dispositivos mais utilizados para apoio na marcha. Existem atualmente diversos modelos mecânicos de andarilho para utilização pela população idosa. No entanto, existe ainda pouca oferta ao nível de andarilhos instrumentados, que permitam uma monitorização e apoio na sua utilização. Este trabalho de projeto pretendeu desenvolver um conjunto de dispositivos de instrumentação que possam ser aplicados num andarilho, permitindo monitorizar a marcha e garantir uma aprendizagem mais eficaz associada com a sua utilização. Este trabalho descreve os dispositivos desenvolvidos para aplicação num andarilho e os testes funcionais que foram implementados. Este será um andarilho instrumentado, ou seja, equipado com a conjugação de diferentes sistemas que o fabricante não tem disponível.

Palavras-chave: Andarilho, instrumentação, sensores

ABSTRACT

The projections made by Instituto Nacional de Estatística (National Institute of Statistic) show an increase in the aging of the Portuguese population. One of the main consequences is the increase in the level of dependence of elderly people, with a cumulative need of use of compensation devices, allowing a more active life. One of the main limitations associated with aging is relate to locomotion, being the walker the most used device for gait support. There are currently several mechanical models of walker for use by the elderly population. However, there is still little supply at the level of instrumented walkers, which allow monitoring and support in their use. The aim of this project work was the development of a set of instrumentation to be apply on a walker, allow monitoring gait and ensuring more effective learning associated with its use. The written work describes the devices developed for application on a walker and the functional tests that have been implement. This will be an instrumented walker, that is, equipped with the combination of different systems that the manufacturer does not have available.

Keywords: Walker, instrumentation, sensors.

ÍNDICE

AGRADECIMENTOS	v
RESUMO.....	vii
ABSTRACT	ix
ÍNDICE.....	xi
ÍNDICE DE FIGURAS	xiii
LISTA DE ABREVIATURAS.....	xv
1. INTRODUÇÃO	1
2. ESTADO DA ARTE.....	3
2.1. Os andarilhos Comercializados	5
2.2. Andarilhos com Biofeedback	8
2.3. Trabalhos biomecânicos envolvendo andarilhos.....	11
3. ANDARILHO INSTRUMENTADO	15
3.1. Rodas e Pés para Andarilho.....	16
3.2. Unidade de processamento	24
3.3. UMI (Unidade de Medição Inercial)	25
3.4. Led's (Light Emitting Diode).....	27
3.5. Interface com utilizador.....	29
4. CONCLUSÃO	35
5. REFERÊNCIAS BIBLIOGRÁFICAS	37
6. ANEXOS	41

6.1.	Anexo 1 – Programação em arduino do andarilho instrumentado 1.....	41
6.2.	Anexo 2 – Programação em arduino do andarilho instrumentado 2.....	56

ÍNDICE DE FIGURAS

Figura 1 – Ciclo da marcha (Sousa, 2017).	4
Figura 2 – (a) Andarilho simples e fixo, (b) Andarilho simples e articulado e (c) Andarilho simples e articulado com dois apoios.	6
Figura 3 – (a) Andarilho com rodas dianteiras dobráveis e fixo e (b) Andarilho com rodas dianteiras e fixo.	7
Figura 4 – (a) Andarilho com 3 rodas e (b) Andarilho com 4 rodas.	8
Figura 5 – Vista de lado do andarilho motorizado (Alves <i>et al.</i> ,2015).	9
Figura 6 – Vista de frente do andarilho motorizado (Alves <i>et al.</i> ,2015).	9
Figura 7 – Punho instrumentado com sensores de pressão e alarme vibratório (Bhutada <i>et al.</i> , 2017).	10
Figura 8 – Correção da postura durante a utilização do andarilho (Bhutada <i>et al.</i> , 2017).	10
Figura 9 – (a) Andarilho FriWalk e (b) Tablet do andarilho FriWalk (Siemens, 2016).	11
Figura 10 – Andarilho inteligente que analisa a condição física do utilizador (Sciencedaily, 2015).	12
Figura 11 – SmartWalker (Rusakov, 2016).	13
Figura 12 – Tablet do <i>SmartWalker</i> (Rusakov, 2016).	13
Figura 13 – Andarilho instrumentado desenvolvido.	15
Figura 14 – Roda projetada.	16
Figura 15 – Princípio de funcionamento da roda.	17
Figura 16 – Roda adquirida.	17
Figura 17 – Posição da mola.	18
Figura 18 – (a) Célula carga à compressão e (b) seu principio de funcionamento.	19
Figura 19 – (a) Célula carga à tração/compressão e (b) seu principio de funcionamento.	19
Figura 20 – (a) Célula carga à tração e (b) seu principio de funcionamento.	19
Figura 21 – Célula carga à compressão FC22.	20
Figura 22 – (a) Suporte inferior e (b) suporte superior do pé instrumentado.	21
Figura 23 – (a) Montagem dos suportes dos pés e (b) vista em corte da montagem dos pés.	21
Figura 24 – (a) Suporte inferior impresso e (b) suporte superior impresso.	22

Figura 25 – Montagem dos suportes impressos.....	22
Figura 26 – (a) Suporte inferior maquinado e (b) suporte superior maquinado	23
Figura 27 – Montagem dos suportes maquinados	23
Figura 28 – Suportes montados	24
Figura 29 – Arduino Mega.....	24
Figura 30 – Arduíno Mega instalado.	25
Figura 31 – MPU 6050.	26
Figura 32 – Acelerómetro instalado.....	26
Figura 33 – Medição do acelerómetro.....	27
Figura 34 – Constituição de um <i>led</i>	27
Figura 35 – Princípio de funcionamento de um <i>led</i>	28
Figura 36 – Led's aplicados.....	28
Figura 37 – Shield RTC para Arduíno.	29
Figura 38 – Shield RTC instalado.....	30
Figura 39 – Shield SD Card.....	30
Figura 40 – Shield SD card instalado no andarilho.	31
Figura 41 – Exportação de dados	31
Figura 42 – Exemplo de tratamento de dados obtidos dos sensores dos pés do andarilho.....	32
Figura 43 – Exemplo de tratamento de dados obtidos das inclinações de lado e de frente.	32
Figura 44 – Pilha e interruptor.....	33

LISTA DE ABRREVIATURAS

DEIUM – Departamento de Eletrónica Industrial da Universidade do Minho

INE – Instituto Nacional de Estatística

FriWalk – Friendly Robot Walker

FSR – Force sensitive sensor

LED – Light Emitting Diode

MEMS – Microelectromecânicos

UMI – Unidade de Medição Inercial

Rx – Rotação no eixo x

Rz – Rotação no eixo z

1. INTRODUÇÃO

O andarilho pode ser classificado como um dispositivo biomecânico de compensação, que permite aumentar a independência e a qualidade de vida em indivíduos com limitações físico-motoras. É um sistema mecânico cuja devida utilização garante a melhoria da capacidade de locomoção, restituindo assim alguma mobilidade e autonomia a quem o utiliza.

O andarilho, tal como qualquer dispositivo de compensação, para permitir a obtenção de benefícios, tem de ser utilizado de modo adequado. No entanto, em muitas situações é empregue de modo incorreto por parte dos utilizadores, o que leva por vezes ao abandono da sua utilização, em muitos casos com transição para cadeira de rodas. Uma utilização incorreta do andarilho causa um efeito negativo na marcha do paciente tornando o seu processo de recuperação mais demorado ou mesmo desadequado.

Assim, é importante desenvolver e implementar recursos e metodologias que tornem a utilização deste dispositivo mais apelativa, permitindo ajudar as pessoas a readquirir uma marcha correta e evitando o seu abandono devido ao seu uso incorreto. Este tipo de ajuda pode acontecer tanto pela interface direta com o utilizador como pelo registo e avaliação que venha a ser feita por técnico especializado que acompanhe o processo de reabilitação de um paciente ou idoso.

Os andarilhos atualmente disponibilizados são, do ponto de vista estrutural, equipamentos bastante avançados, que recorrem a construções leves e seguras. Enquanto estruturas mecânicas, desempenham na sua utilização o papel que é requerido, garantindo a resistência e rigidez mecânica adequadas. No entanto, aquando da sua utilização, a informação do comportamento estrutural dos seus componentes pode ser utilizada, com vista a interagir com o utilizador ou com o clínico que o acompanha, numa perspetiva de ajuda, assente numa linha de biofeedback, ou seja, num conjunto de ferramentas que fornecem informações com objetivo das pessoas, desenvolverem uma capacidade autorregulação. Ora, estes tipos de soluções são ainda escassos no mercado, havendo necessidade de proceder a desenvolvimentos que tenham este tipo de enquadramento.

Assim, foi identificada a possibilidade de contribuir neste domínio de investigação aplicada, desenvolvendo sistemas biomecânicos que possam ser acoplados a um andarilho, e que permitam tanto a aprendizagem e interação direta com o utilizador, como o registo e avaliação da sua utilização.

Neste trabalho foram desenvolvidos alguns sistemas biomecânicos para acoplar num andarilho, que contribuem para atingir estes objetivos, e que se encontram descritos neste projeto. Estruturalmente, o trabalho incorpora um segundo capítulo com um enquadramento inicial e o estado da arte relativamente aos andarilhos atualmente disponibilizados aos utentes.

Posteriormente, no terceiro capítulo, descrevem-se todos os desenvolvimentos que conduziram aos dispositivos desenvolvidos, assim como os testes realizados.

Este trabalho projeto é uma continuação à evolução do andarilho.

2. ESTADO DA ARTE

O andarilho é um dispositivo mecânico utilizado em indivíduos com mobilidade reduzida, de modo a apoiar e garantir a segurança durante a marcha. Trata-se de um tipo de dispositivo de apoio que pode integrar o quotidiano do indivíduo que perdeu a capacidade motora de modo permanente ou ser usada em processos de reabilitação físico-motora. O andarilho reduz a descarga do peso corporal nos membros inferiores, principalmente na fase de apoio do ciclo de marcha, permitindo assim a diminuição da dor nos pacientes ou a garantia do seu equilíbrio.

O andarilho surgiu no início dos anos cinquenta do século XX. O primeiro registo de propriedade associado ao andarilho, foi concedida no ano de 1953 ao inventor *William Cribbes Robb*, de *Stretford*, Reino Unido, com a designação de “*Walking Aid*” (Ajuda para caminhar). Surgiu com o objetivo de ajudar os indivíduos que tivessem sofrido algum acidente ou doença, tendo ficado com mobilidade reduzida ou limitada, em particular ao nível da locomoção. Este dispositivo destinava-se a indivíduos que fossem capazes de se manter de pé com um suporte adequado, ou seja, em que as mãos conseguissem ser usadas para auxiliar as pernas (*Robb*, 1953).

A primeira patente com o termo andarilho (*walker*) foi concedida a *Elmer F. Ries* de *Cincinnati, Ohio* no ano de 1965. Esta invenção foi criada com o objetivo de melhorar e corrigir alguns aspetos da invenção referida anteriormente, pelo facto de algumas pessoas apresentarem elevada instabilidade em posição ortostática, não conseguindo movimentar o dispositivo. Assim, foi desenvolvido sem a necessidade de deslizar ou levantar o andarilho, permitindo ao utilizador ultrapassar obstáculos e elevações ligeiramente altas sem retirar o apoio entre este e o andarilho (*Ries*, 1965).

O primeiro andarilho mais parecido com os andarilhos modernos foi patenteado no ano de 1970 por *Alfred A. Smith of Van Nuys*, na Califórnia, Estados Unidos da América. Este inventor criou um andarilho que foi de encontro a algumas deficiências e necessidades dos dispositivos até a data criados. Uma das grandes necessidades passava por encontrar um adequado balanço entre a rigidez e a rentabilidade do quadro de suporte do andarilho, de modo a encontrar um auto nivelamento em resposta ao peso imposto pelo utilizador (*Nuys*, 1970).

O princípio do andarilho continuou igual, no entanto, foram surgindo atualizações de acordo com as necessidades das pessoas, ou seja, dispositivos com e sem rodas, diferentes quantidades de rodas, com banco, com bolsas, entre outras. Dependendo da necessidade do utilizador, a escolha de um andarilho deve ser efetuada sempre com a supervisão de um profissional de saúde, dado que uma escolha adequada é crucial para a terapêutica ser bem-sucedida.

Ao longo da história foram surgindo diversas atualizações relativamente ao andarilho inicial, sendo cada vez mais possível aliar a tecnologia à medicina, permitindo tirar um maior partido deste dispositivo.

A marcha é constituída por duas etapas, sendo que a conjugação de vários ciclos de marcha se traduz no processo da caminhada (Sousa, 2017). Cada indivíduo tem características de marcha distintas, contudo existem padrões característicos que lhe estão associados. A caminhada requer equilíbrio sobre um membro inferior enquanto o outro se move para frente. Para além do movimento dos membros inferiores, também é essencial a coordenação dos movimentos do tronco e dos membros superiores. O ciclo da marcha pode ser dividido em duas fases (Figura 1):

- fase de apoio - é a etapa que ocorre quando o pé de apoio está em contato com o solo. Essa fase inicia-se quando o calcanhar de um pé toca o solo e termina quando esse mesmo pé se eleva do solo;
- fase do balanço - esta etapa ocorre quando o pé não está em contato com o solo. Inicia-se assim que o pé sai do solo e termina quando o calcanhar do mesmo pé toca o solo novamente.

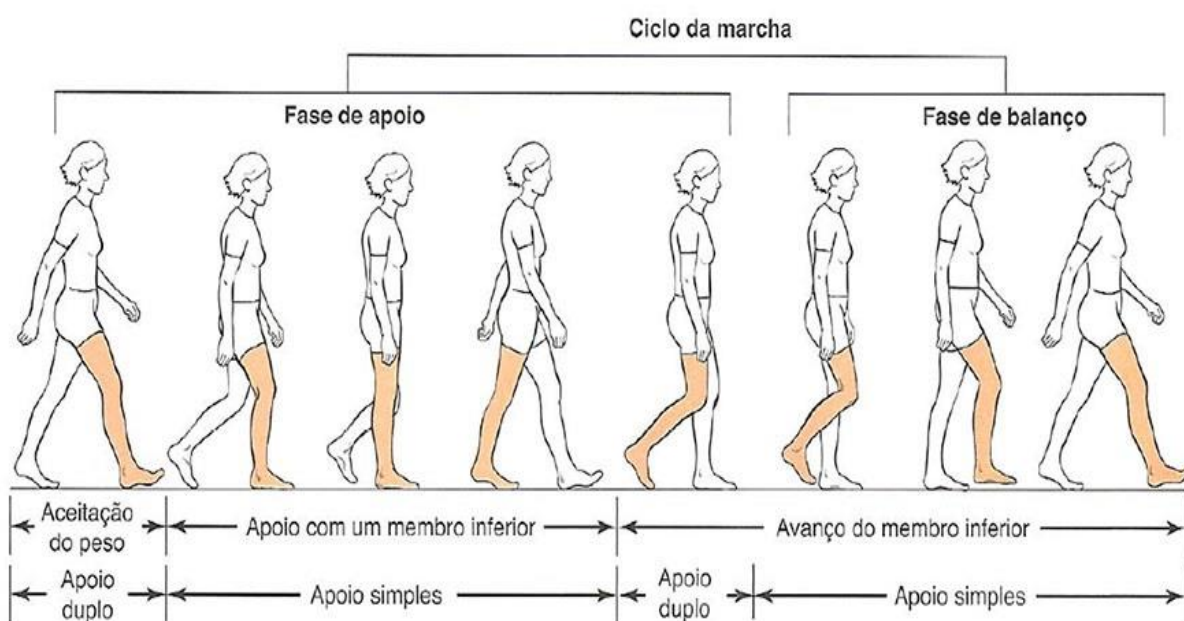


Figura 1 – Ciclo da marcha (Sousa, 2017).

2.1. Os andarilhos Comercializados

Atualmente podem ser adquiridos no mercado diferentes tipos de andarilho, que podem ser divididos em três grandes grupos:

- Simples/sem rodas;
- Com rodas dianteiras e apoios traseiros antiderrapantes;
- Com 3 ou 4 rodas e travões.

Normalmente, os andarilhos são construídos em liga de alumínio, que pelas suas características garante uma boa relação entre o peso, a resistência e a rigidez mecânica. Podem ser equipados com extras personalizados fornecidos pelo fabricante, como por exemplo, bolsas para facilitar o transporte de pertences do utilizador.

2.1.1. Andarilho simples/sem rodas

É o tipo mais leve de andarilho, com quatro pés de apoio, equipados com ponteiros antiderrapantes e com regulação discreta de altura. Este modelo é indicado para pacientes que estejam numa fase inicial do processo de recuperação ou reeducação da marcha. Pelas suas características de leveza, transforma-se num excelente auxiliar para o treino da marcha com maior segurança.

O princípio de utilização deste tipo de andarilho consiste em levantar a sua estrutura, fazendo-a avançar uma certa distância, servindo de apoio no ciclo de marcha seguinte. Dentro deste tipo do andarilho distinguem-se 3 grupos: simples e fixo (Figura 2 - a); simples e articulado (Figura 2 - b); articulado com dois apoios de mãos (Figura 2 - c). O andarilho articulado é semelhante ao simples, sendo, contudo, mais fácil de arrumar ou de transportar, já que incorpora um sistema de articulação. O andarilho com apoios de mãos incorpora dois apoios complementares, posicionados a uma altura inferior de modo a permitir que o paciente se coloque de pé mais facilmente.



Figura 2 – (a) Andarilho simples e fixo¹, (b) Andarilho simples e articulado² e (c) Andarilho simples e articulado com dois apoios³.

2.1.2. Andarilho com rodas dianteiras e apoios traseiros antiderrapantes

Este tipo de andarilho é muito semelhante aos anteriores, tendo incorporado em cada pé dianteiro uma roda de pequeno diâmetro, normalmente em material polimérico. Nas pernas traseiras estão instaladas normalmente ponteiros antiderrapantes. Este tipo de andarilho promove uma marcha mais natural, pois permite ao utilizador progredir no movimento sem ter que parar. Este tipo de andarilho também pode ser articulado (Figura 3 - a) ou fixo (Figura 3 - b), de acordo com o descrito anteriormente. Estes andarilhos são mais indicados para terreno liso e com poucos obstáculos devido às rodas instaladas à frente.

Estes andarilhos são habitualmente utilizados por pacientes que já têm alguma mobilidade, pois permitem que o utilizador empurre o andarilho continuamente, obtendo assim passadas mais fluidas e contínuas, e por isso um ciclo de marcha mais uniforme e rápido. Têm um custo mais elevado quando comparados com os andarilhos referidos anteriormente pelo facto de serem equipados com um par de rodas e também por existir a possibilidade de os equipar com mais extras como por exemplo, um assento. A desvantagem é que são inapropriados para uso em terreno irregular.

¹ Mais que cuidar. (2017). Andarilho simples e fixo. Consultado a 22 de novembro de 2017. Disponível em <https://www.maisquecuidar.com/produto/andarilho-fixo/>.

² Ortopedia Senhor Andarilho. (2017). Andarilho simples e articulado. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortopedia-sr-andarilho.pt/auxiliares-de-marcha>.

³ Ortopedia Marques. (2017). Andarilho articulado com dois apoios de mãos. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortopediamarques.pt/mobilidade-reduzida/514-andarilho-articulado-dois-apoios.html>.



(a)



(b)

Figura 3 – (a) Andarilho com rodas dianteiras dobráveis e fixo⁴ e (b) Andarilho com rodas dianteiras e fixo⁵.

2.1.3. Andarilho com 3 ou 4 rodas e travões (carrinhos de apoio)

Este é o tipo de andarilho mais complexo e também o que tem um maior número de configurações. Apesar de mais fáceis de utilizar e promoverem uma marcha mais natural, impõem menor estabilidade ao utilizador. Dentro deste tipo de andarilho, podem-se distinguir os que integram 3 ou 4 rodas (Figura 4). São apropriados para pacientes com poucas limitações ao nível da sua mobilidade, mas que necessitem de apoio complementar, como por exemplo para atividade fora de edifícios.

As rodas neste tipo de andarilho são de maior diâmetro do que as rodas referidas anteriormente e podem ser rígidas ou de pneu com ar pressurizado. As rodas traseiras são fixas e as rodas dianteiras são direcionais. Possui sistema de travagem simples, por cabos com acionamento ao nível dos punhos, e podem estar equipados com cestas, assentos e tabuleiros amovíveis.

⁴ Elos de Ternura. (2017). Andarilho com rodas dianteiras e apoios traseiros antiderrapantes articulado. Consultado a 22 de novembro de 2017. Disponível em <http://www.elosdetermina.pt/loja/mobilidade-transferencia/andarilho-biort-com-duas-rodas-frontais/>.

⁵ Ortoluz. (2017). Andarilho com rodas dianteiras e apoios traseiros antiderrapantes fixo. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortoluz.com/product/andarilho-com-duas-rodas-actio-invacare>.



(a)



(b)

Figura 4 – (a) Andarilho com 3 rodas⁶ e (b) Andarilho com 4 rodas⁷.

2.2. Andarilhos com Biofeedback

2.2.1. Andarilho Motorizado e Inteligente

O Departamento de Eletrónica Industrial (DEI) da Universidade do Minho desenvolveu um andarilho motorizado e inteligente, que se adapta e auxilia o paciente, através da aquisição e processamento de parâmetros de marcha, quando este se desloca com o andarilho (Alves et al.,2015).

Este andarilho, que pode ser visualizado na Figura 5 e Figura 6, integra um guiador instrumentado para avaliação da força exercida pelo utilizador, um sistema visual e de laser para avaliar a marcha e postura do utilizador e um sistema de motorização nas rodas traseiras que limitam a velocidade e aumentam a segurança. O funcionamento pode ser feito em 4 modos distintos:

- Autónimo: é introduzida a velocidade e o ponto de destino do andarilho pelo especialista e autonomamente, irá para tal destino, evitando todos os obstáculos;
- Manual: o próprio utilizador é quem determina a direção e a velocidade a que deseja caminhar e o especialista é quem define a velocidade máxima;
- Segurança: processo igual ao modo manual, mas alerta o paciente de obstáculos que existam no percurso e pára o andarilho caso seja detetada uma situação de perigo;
- Controlo remoto: o especialista tem o controlo total sobre o movimento e velocidade do andarilho.

⁶ Loja Ortopedica. (2017). Andarilho com três rodas. Consultado a 22 de novembro de 2017. Disponível em <https://lojaortopedica.pt/pt/andarilho-com-3-rodas>.

⁷ Alsomedic. (2017). Andarilho com quatro rodas. Consultado a 22 de novembro de 2017. Disponível em <http://www.alsomedic.pt/andarilhos/448-andarilho-4-rodas-c-assento-e-cesto.html>.

De acordo com os investigadores que desenvolveram este andarilho, o objetivo passa por providenciar assistência na mobilidade em ambientes hospitalares e clínicas de reabilitação, auxiliar no diagnóstico do paciente, assim como no planeamento do seu tratamento.



Figura 5 – Vista de lado do andarilho motorizado (Alves *et al.*,2015).



Figura 6 – Vista de frente do andarilho motorizado (Alves *et al.*,2015).

2.2.2. Andarilho equipado com punhos instrumentados que avisa da má postura de pacientes com doença de Parkinson

Na Universidade do Estado de Arizona, um grupo de investigadores (*Bhutada et al.*, 2017), desenvolveu um punho instrumentado, que se instala no andarilho e alerta o utilizador relativamente à sua postura. O sistema incorpora um sensor piezoresistivo que permite monitorizar o peso exercido sobre o punho. O limite máximo é colocado pelo terapeuta e depende do peso do paciente. Quando o limite é ultrapassado, o sistema emite um sinal vibratório ao utilizador para o alertar que está com uma postura incorreta.



Figura 7 – Punho instrumentado com sensores de pressão e alarme vibratório (*Bhutada et al.*, 2017).

Quando o utilizador inicia a marcha e se encontrar com uma postura incorreta, o sensor emite uma vibração para o alertar. Assim obriga o utilizador a manter uma postura correta e direita, de modo, que não coloque peso sobre o andarilho.



Figura 8 – Correção da postura durante a utilização do andarilho (*Bhutada et al.*, 2017).

2.3. Trabalhos biomecânicos envolvendo andarilhos

2.3.1. Andarilho equipado com câmaras à frente e monitor

A empresa *Siemens* desenvolveu um andarilho que designou por *Friendly Robot Walker* ou *FriWalk* que recorre a um conjunto de câmaras de deteção de profundidade para detetar obstáculos ou terrenos perigosos, e um ecrã onde comunica com o utilizador, alertando-o para potencial perigo (*Siemens*, 2016). Permite a gravação dos movimentos do utilizador todo o dia, durante vários dias, assim como a análise da expressão facial e do batimento cardíaco. De acordo com *Josef Birchbauer*, este andarilho processa 15 a 20 frames por segundo, que produz um caminho virtual onde se consegue ver as pegadas no chão com a correspondente distribuição de pressão e também um caminho de movimento com os pés no ar.

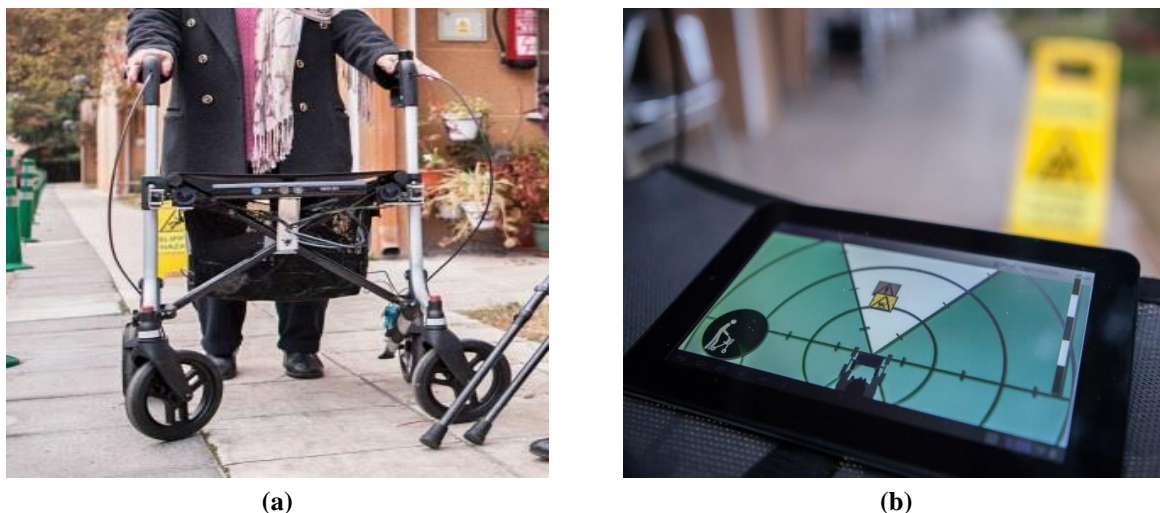


Figura 9 – (a) Andarilho FriWalk e (b) Tablet do andarilho FriWalk (*Siemens*, 2016).

2.3.2. Andarilho que analisa a condição física e regista atividades diárias

O Centro Técnico de Pesquisa da Finlândia (VTT) desenvolveu um andarilho que permite analisar a condição física e registar atividades diárias do utilizador através de um grupo de sensores, cujo sinal é processado através de um software digital (*Kuusisto*, 2015). O dispositivo tem a capacidade de guardar diversas informações do utilizador, tais como, ritmo diário, distância percorrida e tempo de caminhada, velocidade da passada e força de preensão das mãos. Esta informação é essencial para os especialistas analisarem a condição física do paciente e procederem a eventuais ajustes no seu tratamento de reabilitação ou rotina diária (*Sciencedaily*, 2015).



Figura 10 – Andarilho inteligente que analisa a condição física do utilizador (Sciencedaily, 2015).

2.3.3. Andarilho robot - SmartWalker

O andarilho designado por SmartWalker foi desenvolvido e construído por Andrey Rusakov na Universidade da Ciência e Tecnologia de Zurique (Rusakov, 2016). Está equipado com camaras 3D que rodam 360° para analisar a envolvente e medir a distância entre o utilizador e o andarilho, assim como sensores laser que medem a distância das pernas do utilizador. O dispositivo monitoriza se existem obstáculos no caminho a percorrer, incorporando também um grupo de motores elétricos instalados nas rodas traseiras que eliminam o esforço do utilizador ao utilizar o andarilho. O andarilho tem ainda giroscópios para medir o declive e um ecrã no guiador para informação ao utilizador. O sistema pode alternar entre dois modos:

- Modo de assistência: o andarilho ajuda o utilizador a mudar de direção e a percorrer as subidas, diminuindo a quantidade de energia gasta;
- Modo automático: o andarilho atua como um robot automático, que responde a gestos e anda sem nenhuma assistência.

A Figura 11 apresenta visualização do Smartwalker e a Figura 12 a interface com o utilizador disponibilizada por este dispositivo.



Figura 11 – SmartWalker (Rusakov, 2016).

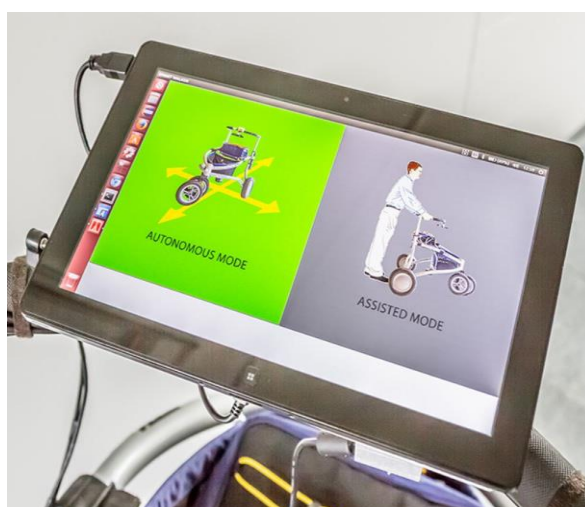


Figura 12 – Tablet do SmartWalker (Rusakov, 2016).

Esta pesquisa vem comprovar a falta de andarilhos tecnológicos no mercado. Esta é ainda uma área pouco desenvolvida, mas com enorme potencial de desenvolvimento. Pois um equipamento que devolva informação relativa ao seu utilizador, irá ajudar os terapeutas numa melhor escolha de terapias a aplicar, o que se traduz numa aceleração da recuperação por parte dos pacientes.

3. ANDARILHO INSTRUMENTADO

Neste capítulo descrevem-se as soluções desenvolvidas, e que passaram numa primeira fase pela conceção de um sistema de rodas autoblocantes durante a marcha e posteriormente pela implementação de uma componente de biofeedback no andarilho, tanto para o utilizador como para o terapeuta. O objetivo passou por instrumentar um andarilho para o tornar mais seguro, auxiliar o utilizador na marcha e ajudar os especialistas na identificação de características associadas à marcha, de modo a tornar o diagnóstico, processo de reabilitação ou dia-dia do paciente mais fácil e mais eficaz.

O andarilho instrumentado desenvolvido é resumidamente apresentado na Figura 13. Este andarilho pode ser descrito por um conjunto de unidades que incorpora. Assim:

- A - Unidade estrutural. A unidade estrutural do andarilho tem como base o andarilho de 4 pés, da marca Invacare, modelo Brio, com diâmetro do tubo com 22mm;
- B - Pé Instrumentado. Sistema de medição de força aplicada em cada pé, desenvolvido, e que incorpora uma célula de carga e conjunto de elementos estruturais de acoplamento ao andarilho de base;
- C – Sistema autoblocante. Rodas equipadas com um sistema autoblocante que incluem uma mola conectada à roda.
- D – Luzes LED. Sistema de visualização com luzes do tipo LED, para biofeedback, em tempo real da inclinação do andarilho durante a sua utilização;
- E – Unidade de Aquisição. Sistema de aquisição e registo de dados, que incorpora uma unidade de massa inercial com acelerómetro e giroscópio, assim como todo o hardware do sistema.

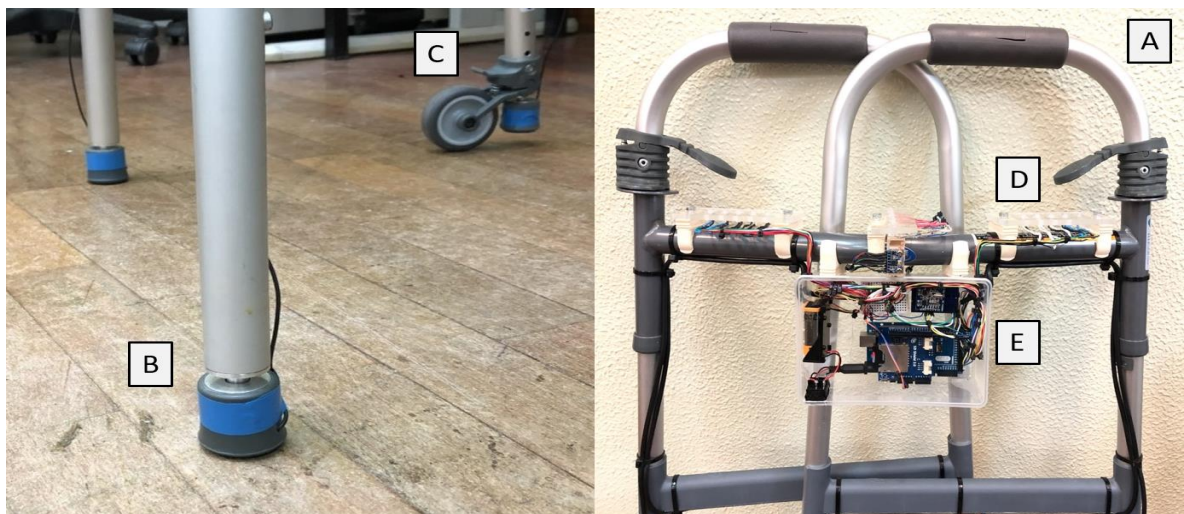


Figura 13 – Andarilho instrumentado desenvolvido.

A solução final do andarilho instrumentado tem a capacidade de auxiliar o utilizador na sua marcha, indicar a inclinação do andarilho através de um sistema de luzes do tipo *LED* e de um giroscópio com acelerómetro, quantificar a distribuição de peso do utilizador em cada um dos pés, e registar todos os dados adquiridos para posterior análise.

A descrição que se desenvolve neste capítulo incorpora todos os passos que conduziram ao desenvolvimento do andarilho final.

3.1. Rodas e Pés para Andarilho

3.1.1. Rodas com sistema autoblocante

O primeiro objetivo deste trabalho de projeto envolveu o desenvolvimento de um sistema de rodas com sistema autoblocante. A ideia de base passou por desenvolver um sistema simples, em que uma roda estaria conectada a uma mola de tração-compressão. O sistema desenvolvido é baseado numa mola com constante de rigidez ajustável através de um parafuso instalado num pé de um andarilho, tal como se pode visualizar na Figura 14 e Figura 15.

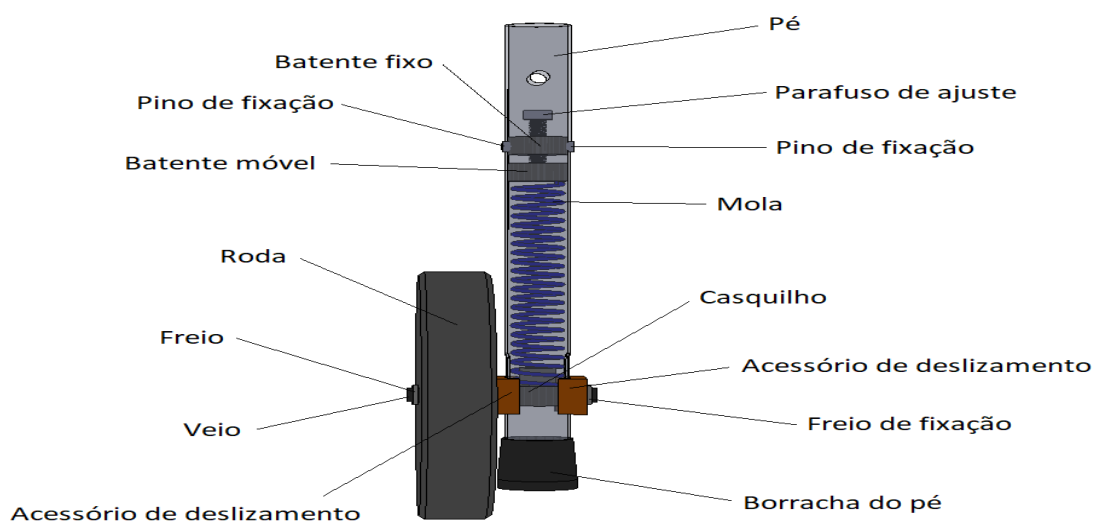


Figura 14 – Roda projetada.

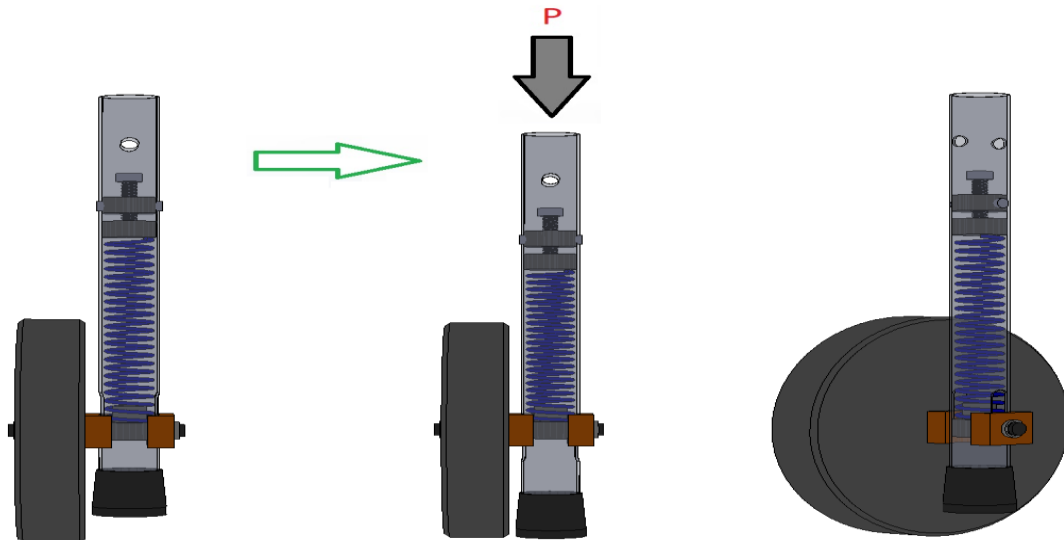


Figura 15 – Princípio de funcionamento da roda.

A instalação deste sistema de roda num andarilho permitiria ao utilizador realizar uma passada mais fluida, no entanto, se em algum momento se sentisse inseguro, ao aplicar força descendente, a mola comprimida e em vez da roda estar em contacto com o chão, estaria uma borracha antiderrapante, immobilizando assim o andarilho.

No processo de desenvolvimento das rodas autoblocantes, foi identificado no mercado um sistema de rodas, também autoblocantes (Figura 16), que permitiriam assegurar parte dos objetivos do presente trabalho. O único inconveniente associado a estas rodas assenta no facto de não permitir o ajuste da rigidez da mola (Figura 17). Assim, optou-se por não avançar para a construção física do sistema de rodas autoblocantes desenvolvido, direcionando o trabalho para a instrumentação do andarilho, tal como se descreve na seção seguinte.



Figura 16 – Roda adquirida.



Figura 17 – Posição da mola.

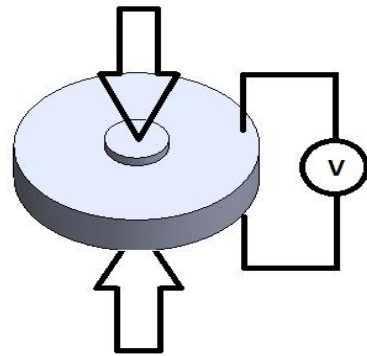
3.1.2. Pé de Andarilho com Sistema de Avaliação de Força Aplicada

O desenvolvimento de um sistema de medição da força aplicada no pé do andarilho pode desempenhar um papel relevante no contexto da instrumentação do mesmo para monitorização da sua utilização. Tendo em vista a implementação de um pé instrumentado, foi desenvolvido um dispositivo, que funciona por encaixe, e que permite atingir estes objectivos. De seguida descrevem-se os passos implementados até à obtenção da solução final.

As células de carga são transdutores de força que abrangem uma vasta gama de aplicações, desde balanças de precisão até, automatismos e controladores industriais entre outras aplicações. As principais razões da sua utilização estão relacionadas com a fiabilidade de resposta, a um custo acessível, com dimensões reduzidas mesmo para aplicações de grandes cargas, permitindo ainda fazer medições dinâmicas. O principal objetivo de uma célula de carga é fornecer um sinal elétrico sempre que uma carga for aplicada sobre ela. Existem três grandes tipos de células de carga, e que dependem do comportamento estrutural que lhe está associado: à compressão (Figura 18), à tração/compressão (Figura 19) e à flexão (Figura 20). A célula de carga à compressão, como o nome indica, só mede forças quando a carga aplicada está a comprimir (Figura 18), a célula de carga à tração/compressão permite quantificar forças quando a carga está a tracionar ou a comprimir a estrutura (Figura 19) e a célula de carga à flexão mede a força quando esta provoca um efeito de flexão, por isso é necessário que esta célula esteja fixa numa das extremidades (Figura 20).



(a)

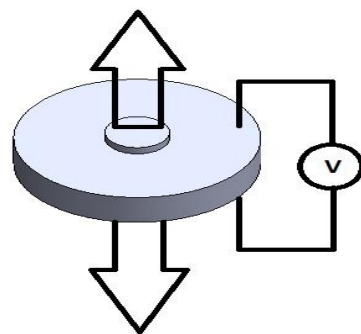


(b)

Figura 18 – (a) Célula carga à compressão⁸ e (b) seu princípio de funcionamento.



(a)

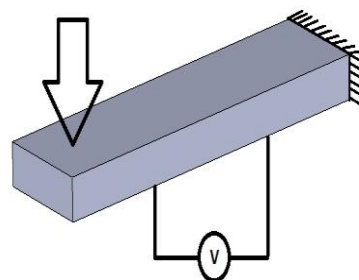


(b)

Figura 19 – (a) Célula carga à tração/compressão⁹ e (b) seu princípio de funcionamento.



(a)



(b)

Figura 20 – (a) Célula carga à tração¹⁰ e (b) seu princípio de funcionamento.

⁸ ACR. (2019). Célula de carga à compressão. Consultado a 10 de março de 2019. Disponível em <http://www.acr1.com.br/produtos/divisao-pesagem-industrial/celulas-de-carga/dk-bp-2093-compressao>.

⁹ ACR. (2019). Célula de carga à tração. Consultado a 10 de março de 2019. Disponível em <http://www.acr1.com.br/produtos/divisao-pesagem-industrial/celulas-de-carga/ctc-tracao-compressao>.

¹⁰ Portal da célula de carga. (2019). Célula de carga à flexão. Consultado a 10 de março de 2019. Disponível em <http://www.celuladecarga.com.br/portal/?p=112>.

Para este projeto, foi selecionada uma célula de carga à compressão visto que o objetivo seria medir a força exercida pelo utilizador no pé do andarilho, que tem solicitação apenas numa direção e sentido. Para este projeto, foi selecionada uma célula de carga à compressão visto que o objetivo seria medir a força exercida pelo utilizador no pé do andarilho, que tem solicitação apenas numa direção e sentido. A célula de carga escolhida foi a FC22 da empresa TE Connectivity com uma capacidade de medição entre 4.5 kg a 45.3 kg de força. Este foi o modelo selecionado pelo facto de ter uma grande capacidade de medição e pelo seu reduzido tamanho, tornando assim possível adapta-lo no pé do andarilho.



Figura 21 – Célula carga à compressão FC22¹¹.

Considerando que a estrutura do andarilho não está preparada para a instalação de células de carga e no sentido de ser possível a medição da força exercida, foram desenvolvidos suportes que ao serem aplicados na borracha de proteção dos pés do andarilho e no tubo da estrutura permitissem auxiliar na medição da força exercida.

A Figura 22 mostra o modelo 3D dos suportes desenvolvidos para acoplamento da célula de carga. Com o intuito de evitar o atrito da borracha de proteção dos pés do andarilho e por consequência influenciar a leitura dos valores da força exercida, foram desenhados dois tipos de suportes: um suporte inferior que mantivesse a célula de carga alinhada e a borracha no lugar e um suporte superior que transmitisse a força exercida no andarilho à célula de carga.

¹¹ TE Connectivity. (2019). Compression load cell. Consultado a 20 de março de 2019. Disponível em <https://www.te.com/global-en/product-CAT-FSE0001.html>.

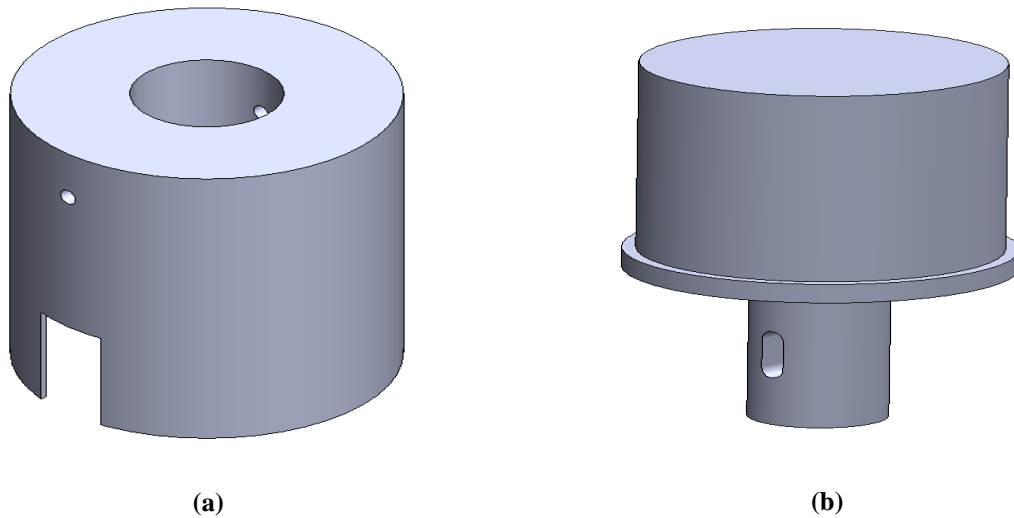


Figura 22 – (a) Suporte inferior e (b) suporte superior do pé instrumentado.

O objetivo passou por encaixar o suporte superior no suporte inferior e com o auxílio de um escatel permitir que estes fiquem conectados de modo a que nunca se separem. O suporte superior tem um canal próprio para que haja alguma mobilidade entre os suportes.

Entre os dois suportes existe uma folga possibilitando que estes nunca toquem um no outro, apenas no encaixe, pois assim proporciona a transmissão de toda a força exercida no andarilho (Figura 23).

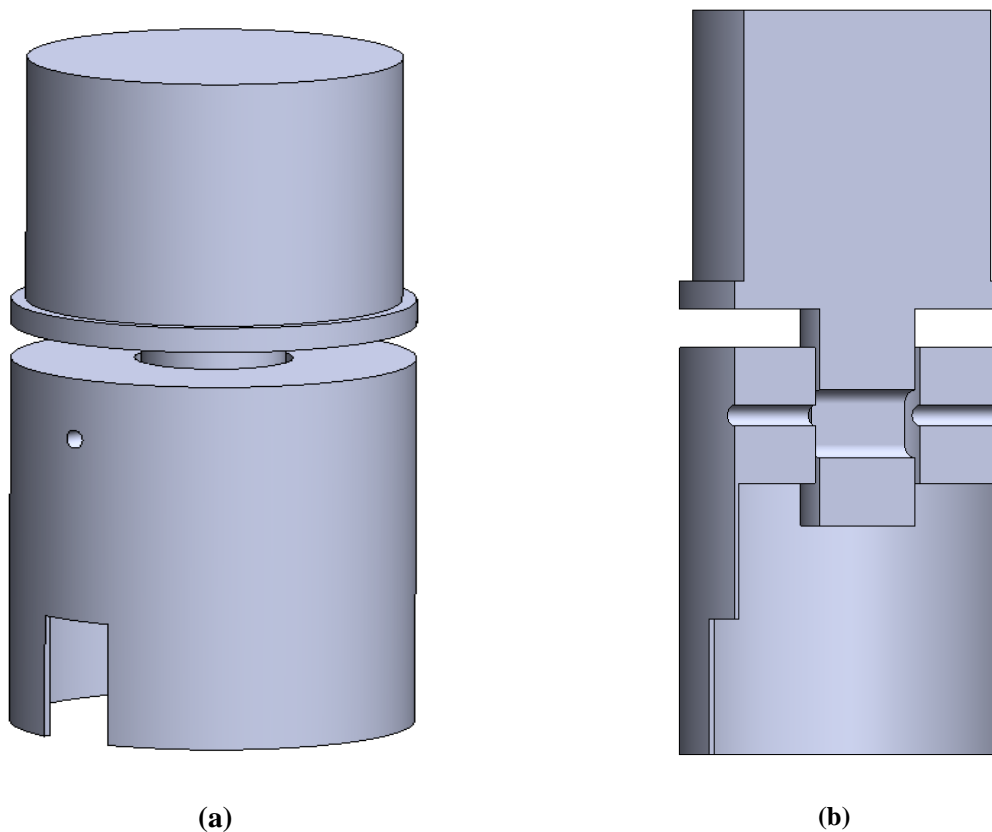
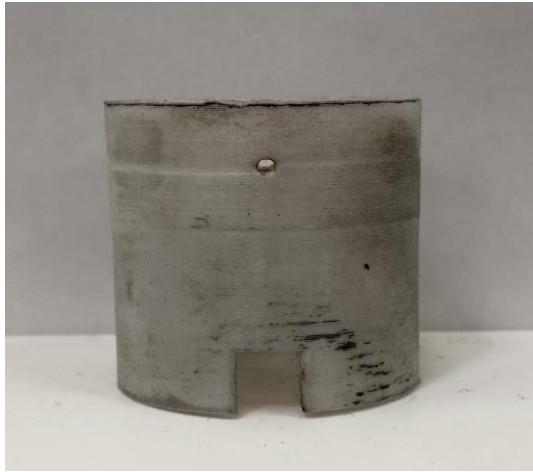


Figura 23 – (a) Montagem dos suportes dos pés e (b) vista em corte da montagem dos pés.

Inicialmente, os suportes desenvolvidos foram produzidos através de fabrico aditivo, numa impressora 3D da marca ProJet, modelo 1500, de modo a ser possível testar o seu princípio de funcionamento. O suporte inferior foi produzido em ABS (Acrilonitrila butadieno estireno) Figura 24a e o suporte superior em PLA (poliácido láctico) Figura 24b.



(a)



(b)

Figura 24 – (a) Suporte inferior impresso e (b) suporte superior impresso.

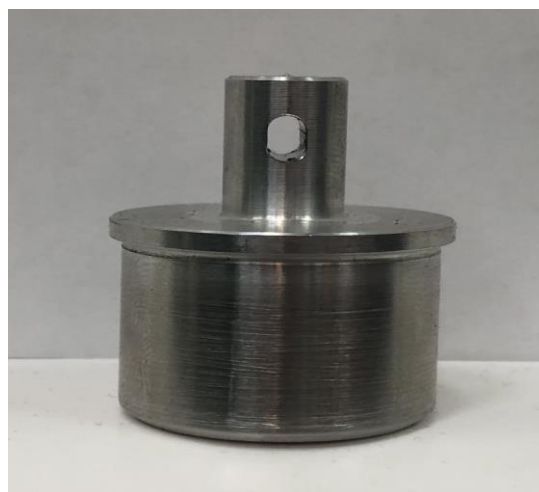


Figura 25 – Montagem dos suportes impressos

Após a realização de alguns testes com os componentes em material polimérico, foi possível aferir a funcionalidade do sistema, assim como aumentar a sua resistência mecânica. Assim, optou-se pela maquinação dos suportes em liga de alumínio (Figura 26)



(a)



(b)

Figura 26 – (a) Suporte inferior maquinado e (b) suporte superior maquinado



Figura 27 – Montagem dos suportes maquinados

Após a realização de alguns testes com os suportes maquinados, foi possível concluir que no que diz respeito ao princípio de funcionamento dos suportes o objetivo foi eficazmente cumprido. A espessura da parede do suporte inferior e a resistência de ambos os suportes se revelou adequada com a maquinação dos suportes em alumínio.

Deste modo, foi possível adaptar a estrutura do andarilho para a instalação de células de carga permitindo a medição da força exercida no andarilho pelo seu utilizador. Na Figura 28 é possível observar os suportes, maquinados em alumínio, inseridos no andarilho, na sua versão final.



Figura 28 – Suportes montados

3.2. Unidade de processamento

O Arduíno é uma plataforma eletrônica *open-source* que permite a implementação de *hardware* e *software*. Os microcontroladores Arduíno permitem entradas digitais ou analógicas. Existem vários tipos de microcontroladores Arduíno, cada um com um diferente tipo de aplicação. A linguagem de programação das placas Arduíno é fundamentada no *Wiring* e o *software* é baseado no *Processing*. Foi selecionada a placa Arduíno Mega devido a ser aquela com mais entradas e saídas em relação a todas as outras da vasta lista existente do Arduíno. O microcontrolador Arduíno Mega tem 54 entradas e saídas digitais, 16 entradas analógicas e uma grande memória de armazenamento.



Figura 29 – Arduino Mega.

Neste projeto, o Arduino Mega instalado permite a leitura dos valores fornecidos pelo acelerómetro e giroscópio, ligar e desligar os LED's, adquirir os dados das células de carga instaladas nos quatro pés do andarilho e registar a hora e data fornecidos pelo relógio instalado.

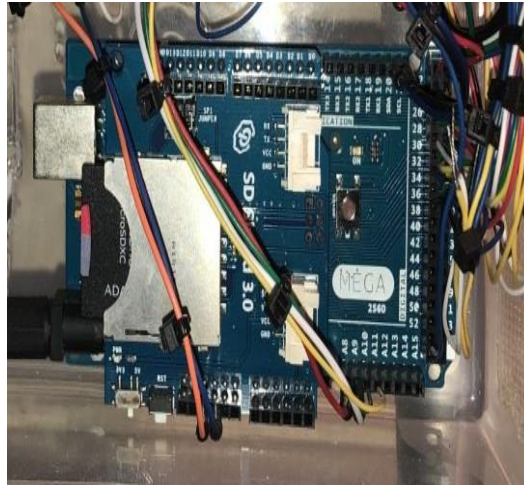


Figura 30 – Arduino Mega instalado.

3.3. UMI (Unidade de Medição Inercial)

O acelerómetro é um dispositivo que mede a aceleração própria no sistema em que está instalado. É muito utilizado e a sua aplicação depende do tamanho, da massa, sensibilidade, eixos de medição, faixas de amplitude e frequência. Existem no mercado diversos tipos de acelerómetros, sendo os mais comuns:

- **Piezoelétricos:** é designado por este nome pois é constituído por cristais que ao sofrerem uma deformação, geram uma tensão elétrica. No seu interior, está uma massa, que ao sofrer uma aceleração e como está em contacto com um cristal piezoelétrico, faz com que este gere uma carga elétrica. Esta carga elétrica é proporcional a aceleração, ou seja, quanto maior a aceleração maior a carga elétrica que o cristal produz;
- **Piezoresistivos:** o princípio de funcionamento é muito idêntico ao do extensómetro, ou seja, varia a sua resistência com a deformação que foi sujeito. É necessário que esteja ligado com uma configuração de meia ponte de Wheatstone ou de ponte completa;
- **Capacitivos:** é constituído por uma massa sísmica e duas placas, uma unida à estrutura do acelerómetro e outra presa à massa. A capacitância varia conforme a massa sísmica e é proporcional a distância entre as placas;
- **Microelectromecânicos (MEMS):** os acelerómetros MEMS podem ser dos tipos referidos anteriormente mas com a particularidade de serem de tamanho microscópico, na sua maioria, constituídos de silício.

O giroscópio é um dispositivo cujo eixo de rotação mantém sempre a mesma direção na ausência de forças que o perturbem. O giroscópio veio substituir a bússola na navegação marítima, na aviação auxilia o piloto automático, permitindo o voo em condições de visibilidade zero e no espaço é utilizado para guiar os vaivéns.

O acelerómetro com giroscópio instalado no andarilho é do tipo MEMS com a designação MPU 6050, com 3 eixos e 6 graus de liberdade. Foi seleccionado devido ao seu baixo custo e alta precisão (Figura 31).

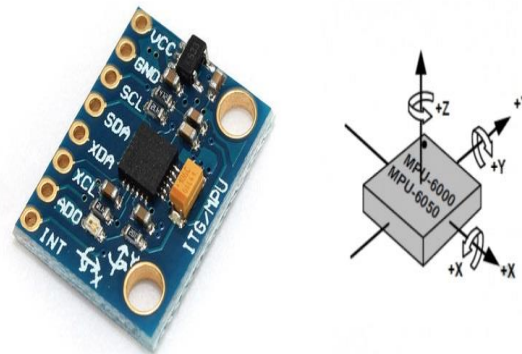


Figura 31 – MPU 6050.

A instalação do acelerómetro com giroscópio surge com a necessidade de avaliar a inclinação em que o andarilho se encontra. Este está aplicado na parte central do andarilho (Figura 32). Tal como se pode verificar na Figura 33, o acelerómetro foi programado para medir a rotação em x (R_x) e a rotação em z (R_z) pois é segundo estes dois eixos que o utilizador tem mais propensão a acidentes.

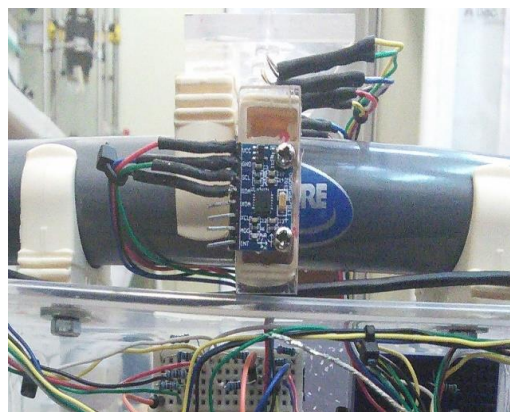


Figura 32 – Acelerómetro instalado.

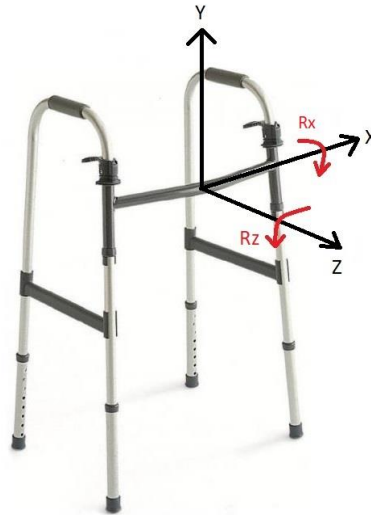


Figura 33 – Medição do acelerómetro.

3.4. Led's (Light Emitting Diode)

Os *led's* são díodos semicondutores que emitem luz quando são alimentados com energia elétrica. São aplicados em dispositivos onde a sua utilização é mais conveniente do que as tradicionais lâmpadas fluorescentes, já que têm baixo custo, baixo consumo de energia, grande fluxo luminoso para uma baixa potência consumida e maior durabilidade. Os *led's* utilizam um nível de tensão de 1,6 a 3,3 V e seguem um código de identificação para a determinação dos terminais A (ânodo) e K (cátodo) para uma correta polarização, sendo o terminal mais curto, o cátodo e o mais comprido, o ânodo (Figura 34).

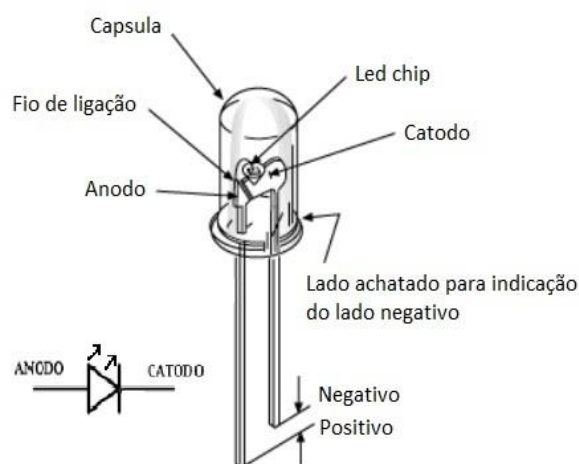


Figura 34 – Constituição de um *led*.

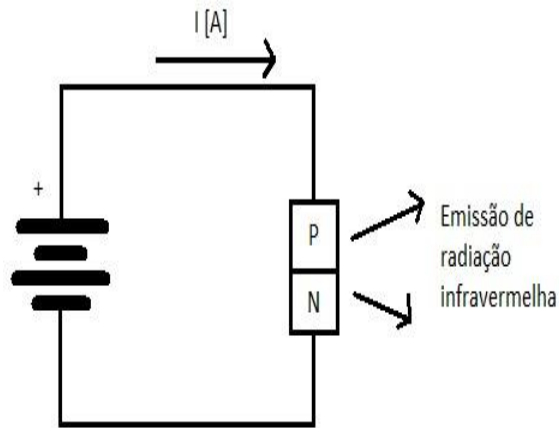


Figura 35 – Princípio de funcionamento de um led.

Selecionou-se o led devido ao seu pequeno tamanho e sua versatilidade de instalação. Na parte superior do andarilho, foram instaladas três barras com *led's*, duas segundo o eixo y e uma segundo o eixo z (Figura 36), para auxiliar o utilizador e/ou o terapeuta relativamente à identificação visual da inclinação do andarilho. Segundo uma inclinação de segurança predefinida, os *led's* são ativados, um a um (os dois primeiros são verdes, os três restantes são vermelhos) com o aumento da inclinação. Se a inclinação começar a reduzir, os *led's* desactivam-se um a um até que a inclinação seja zero.

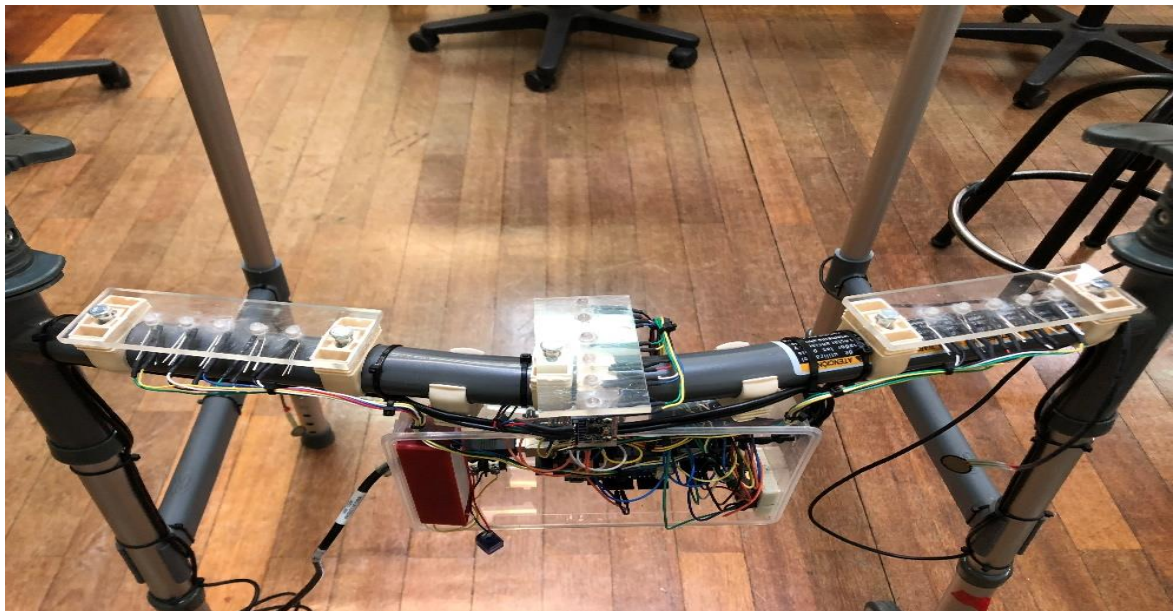


Figura 36 – Led's aplicados.

3.5. Interface com utilizador

3.5.1. Relógio

Como referido anteriormente, na gama Arduino existem muitos microcontroladores (placas) para diferentes aplicações e de acordo com os objetivos a atingir. Para complementar os diferentes microcontroladores existentes, a Arduino criou placas de circuito modular, que se conectam ao microcontrolador de modo fácil e permitem acrescentar uma funcionalidade extra. Estas chamam-se de *shields*.

Uma das imensas *shields* que existem é a *shield RTC*, que não passa de um relógio com uma pilha, que guarda a data e hora na sua memória e fornece-a à placa Arduino sempre que for necessário, dependendo do que o programador desejar. De modo a auxiliar o tratamento dos dados, foi introduzido um relógio no Arduino Mega para que, cada vez que forem gravados dados, estes sejam gravados com data e hora a que estes foram adquiridos. Portanto possibilita fazer diferenciação entre pacientes e também comparação de dados, acompanhando assim a evolução dos pacientes.

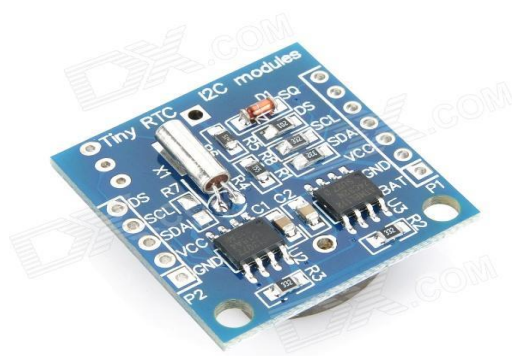


Figura 37 – Shield RTC para Arduino.

A conexão ao Arduino Mega é simples, através de quatro conectores elétricos, sendo dois cabos para alimentação e os outros dois cabos para passagem de informação para o microcontrolador. Na primeira conexão ao computador, a RTC vai buscar a informação da data e da hora de onde está ligado e guarda na sua memória. Como tem uma pilha inserida, mesmo que deixe de estar conectado ao computador, a *shield RTC* continua sempre a contar, para assim que seja necessária a informação, forneça uma correta data e hora.



Figura 38 – Shield RTC instalado.

3.5.2. Armazenamento de dados

Aproveitando a vasta lista de *shields* para Arduino e para complementar este projeto na questão de armazenamento de dados, foi adquirida uma *shield SD Card* (Figura 39). Esta placa é um adaptador de cartão de memória para o microcontrolador e permite um grande armazenamento de dados obtidos. Para um tratamento dos dados obtidos posteriormente à utilização do andarilho por parte do paciente, a *shield* com um cartão de memória SD no Arduino grava todos os valores adquiridos de inclinação e de pressão através do acelerómetro e dos sensores de pressão, respetivamente. Este cartão de memória grava de segundo a segundo, cada vez que o Arduino seja ligado e pára de gravar assim que este seja desligado. Será possível ao terapeuta retirar o cartão de memória da *shield* e introduzi-lo num computador habilitado a ler cartões de memória e efetuar o tratamento dos dados como desejar (Figura 41).



Figura 39 – Shield SD Card.

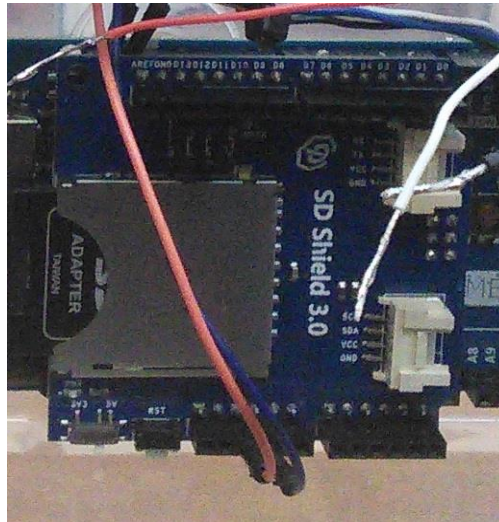


Figura 40 – Shield SD card instalado no andarilho.

	A	B	C	D	E	F	G	H
1	ANDARILHO							
2								
3	Hora	Data	Apoio da direita traseiro: [N]	Apoio da esquerda traseiro: [N]	Apoio da esquerda frente: [N]	Apoio da direita frente: [N]	Angulo de lado: [Âº]	Angulo de frente: [Âº]
4	18:51:09	30/06/2019	0	0	0	0	0	0
5	18:51:11	30/06/2019	0	0	0	0	0	0
6	18:51:12	30/06/2019	0	0	0	0	0	0
7	18:51:13	30/06/2019	0	0	0	0	0	0
8	18:51:15	30/06/2019	0	0	0	0	0	0
9	18:51:16	30/06/2019	0	0	0	0	0	0
10	18:51:17	30/06/2019	0	0	0	0	0	0
11	18:51:18	30/06/2019	0	0	0	0	0	0
12	18:51:20	30/06/2019	0	0	0	0	0	0
13	18:51:21	30/06/2019	0	0	0	0	0	0
14	18:51:22	30/06/2019	0	0	0	0	0	0
15	18:51:24	30/06/2019	0	0	0	0	0	0
16	18:51:25	30/06/2019	0	0	0	0	0	0
17	18:51:26	30/06/2019	0	0	0	0	0	0
18	18:51:28	30/06/2019	1	0	0	0	0	0
19	18:51:29	30/06/2019	1	0	0	0	0	0
20	18:51:30	30/06/2019	2	0	0	0	0	0
21	18:51:31	30/06/2019	2	0	0	0	0	0
22	18:51:33	30/06/2019	2	0	0	0	0	0
23	18:51:34	30/06/2019	2	0	0	0	0	0
24	18:51:35	30/06/2019	2	0	0	0	0	0
25	18:51:37	30/06/2019	26	8	0	0	0	0
26	18:51:38	30/06/2019	14	8	0	0	0	0
27	18:51:39	30/06/2019	12	7	0	0	0	1
28	18:51:41	30/06/2019	2	16	0	0	0	0
29	18:51:42	30/06/2019	5	0	0	0	-5	1
30	18:51:43	30/06/2019	2	0	39	21	0	4
31	18:51:44	30/06/2019	2	0	49	0	-1	4
32	18:51:46	30/06/2019	0	0	0	39	6	6
33	18:51:47	30/06/2019	1	3	0	0	2	3
34	18:51:48	30/06/2019	38	47	0	0	-1	0
35	18:51:50	30/06/2019	2	0	0	0	0	0
36	18:51:51	30/06/2019	2	0	0	0	0	0

Figura 41 – Exportação de dados

3.5.3. Tratamento de dados

Com os dados adquiridos (Figura 41) foi possível criar uma macro em Microsoft Excel para selecionar e tratar os valores adquiridos e guardados no cartão de memória do andarilho instrumentado. Foi criado um gráfico de barras (Figura 42) para ser mais fácil e intuitivo analisar e comparar os valores obtidos pelos sensores nos pés do andarilho. É possível observar os máximos em cada sensor. Com o mesmo objetivo referido anteriormente, foi criado um gráfico de linhas (Figura 43) para os valores de inclinação de lado e para a frente.

Estes dois tipos de gráficos servem apenas como exemplo, permitindo elucidar o leitor das possibilidades no que diz respeito ao tratamento dos dados adquiridos, sendo possível adaptar consoante os utilizadores do andarilho e objetivos do terapeuta.

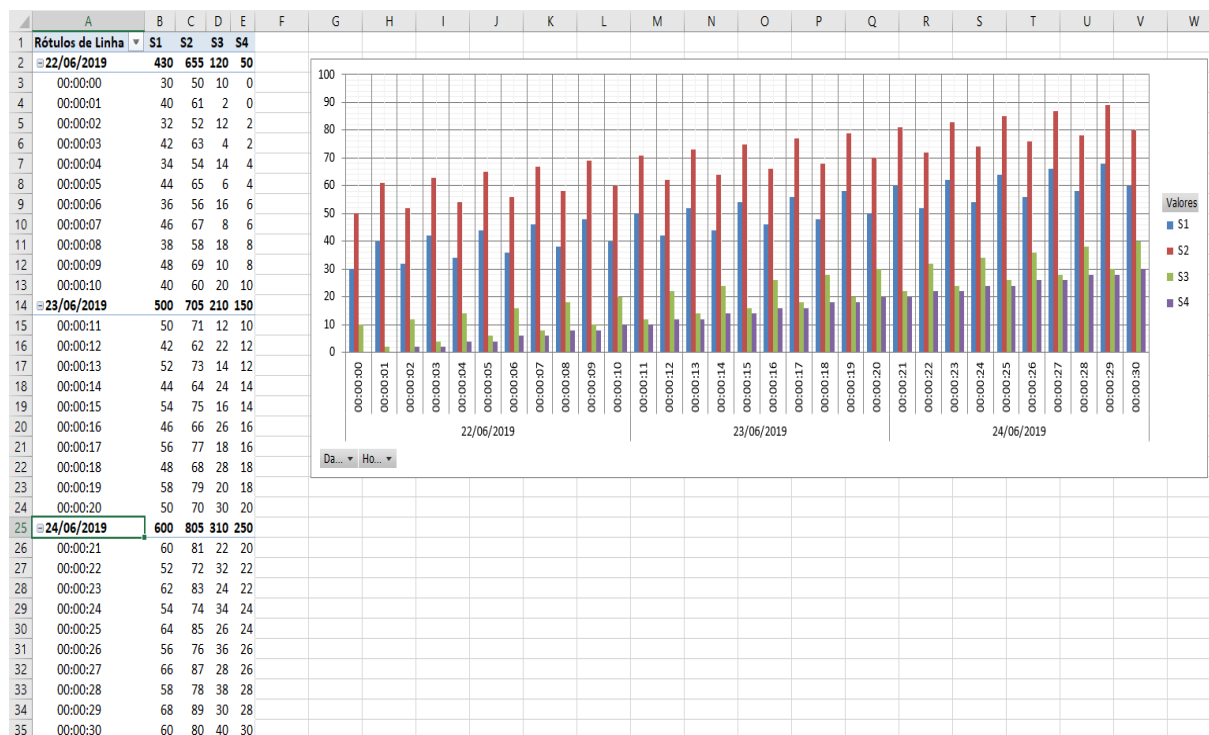


Figura 42 – Exemplo de tratamento de dados obtidos dos sensores dos pés do andarilho.

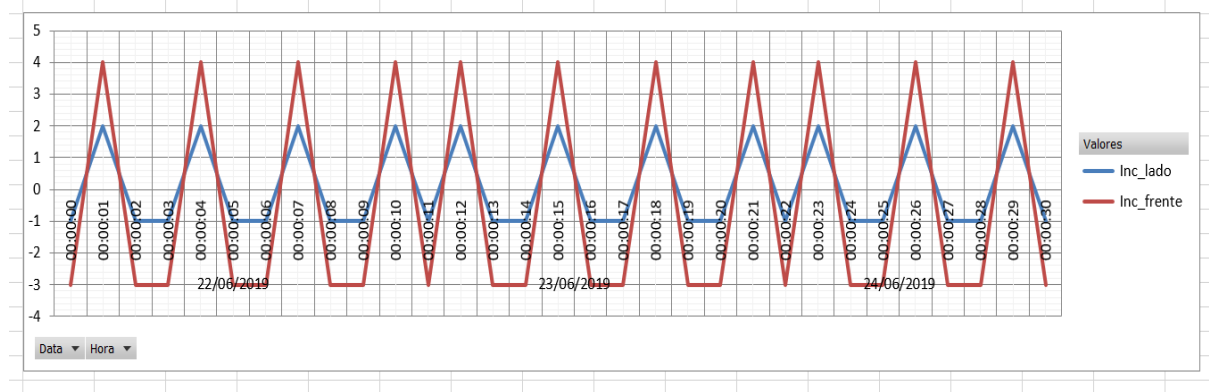


Figura 43 – Exemplo de tratamento de dados obtidos das inclinações de lado e de frente.

3.5.4. Pilha e interruptor ON/OFF

De maneira a não limitar a área de uso do andarilho instrumentado, ou seja, não existir a necessidade de este estar conectado a um computador por um fio aquando a sua utilização, foram instalados os seguintes componentes, uma pilha de 9v e um interruptor, para alimentar de energia o Arduino colocado no andarilho. A pilha pode ser carregada com um carregador

próprio. O interruptor foi instalado para não haver necessidade do Arduíno estar sempre ligado, esgotando assim a pilha muito mais depressa.

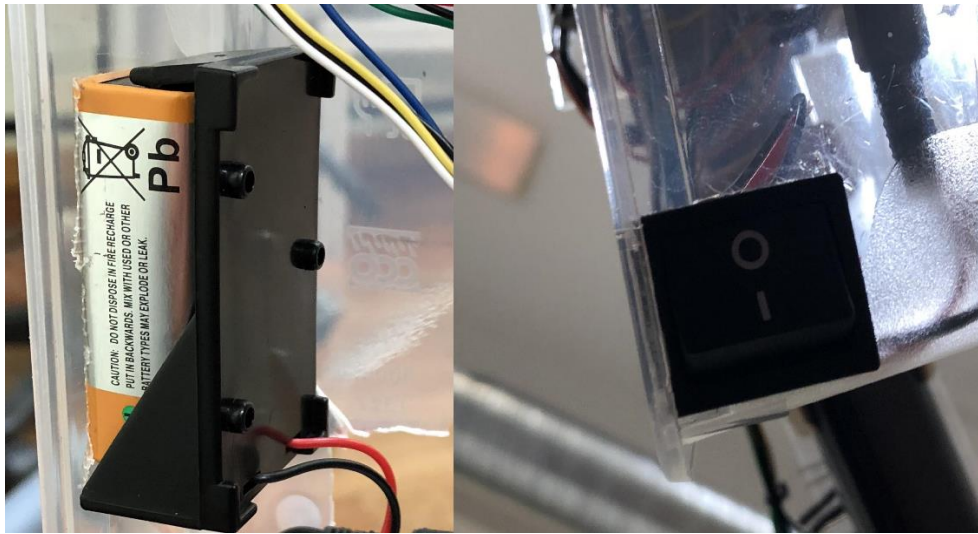


Figura 44 – Pilha e interruptor.

4. CONCLUSÃO

O presente trabalho projeto pretendeu desenvolver sistemas eletromecânicos que possam ser acoplados a um andarilho, tornando-o com características de monitorização e biofeedback, de modo a auxiliar utilizadores com dificuldades associadas à marcha no processo de aprendizagem da sua utilização, e ao mesmo tempo, ajudando os terapeutas a adquirir dados para uma maior precisão na caracterização da marcha dos seus utentes, que pode resultar numa melhor escolha ou ajuste no tipo de tratamento.

O trabalho envolveu em primeiro lugar uma revisão dos trabalhos académicos e de investigação que apresentavam inovações em qualquer tipo de andarilhos, que mostrou a existência de uma lacuna relativamente à existência deste tipo de dispositivos para acoplamento em andarilhos.

Posteriormente, o trabalho desenvolvido passou por responder às necessidades das pessoas que utilizam o andarilho diariamente, tendo sido desenvolvidos alguns componentes, nomeadamente: pés instrumentados, sistema autoblocante, luzes LED e unidade de aquisição. O estudo da funcionalidade dos dispositivos desenvolvidos mostrou que os mesmos podem ser aplicados num andarilho, tornando-o um dispositivo de compensação mais eficaz, tanto no processo de aprendizagem associada à utilização do andarilho como no acompanhamento do procedimento de reabilitação físico-motora.

Em termos de trabalho futuro, será necessário proceder a um conjunto mais alargado de estudos envolvendo a utilização de um andarilho com os sistemas desenvolvidos, de modo a comprovar a sua eficácia tanto na fase de aprendizagem de utilização como no acompanhamento dos procedimentos de reabilitação.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- (1) ACR. (2019). Célula de carga à compressão. Consultado a 10 de março de 2019. Disponível em <http://www.acr1.com.br/produtos/divisao-pesagem-industrial/celulas-de-carga/dk-bp-2093-compressao>.
- (2) ACR. (2019). Célula de carga à tração. Consultado a 10 de março de 2019. Disponível em <http://www.acr1.com.br/produtos/divisao-pesagem-industrial/celulas-de-carga/ctc-tracao-compressao>.
- (3) Alsomedic. (2017). Andarilho com quatro rodas. Consultado a 22 de novembro de 2017. Disponível em <http://www.alsomedic.pt/andarilhos/448-andarilho-4-rodas-c-assento-e-cesto.html>.
- (4) Arduino. (2018). Arduino Mega 2560. Consultado a 22 de abril de 2018. Disponível em <https://www.arduino.cc/en/Guide/ArduinoMega2560>.
- (5) Departamento de Eletrónica Industrial da Universidade do Minho. (2015). Andarilho Motorizado e Inteligente. Consultado a 06 de abril de 2018. Disponível em <https://www.dei.uminho.pt/wp-content/uploads/2015/12/andarilho.pdf>.
- (6) Elos de Ternura. (2017). Andarilho com rodas dianteiras e apoios traseiros antiderrapantes articulado. Consultado a 22 de novembro de 2017. Disponível em <http://www.elosdeternura.pt/loja/mobilidade-transferencia/andarilho-biort-com-duas-rodas-frontais/>.
- (7) ETH Zurich. (2016). SmartWalker. Consultado a 06 de abril de 2018. Disponível em <https://www.ethz.ch/en/news-and-events/eth-news/news/2016/02/smart-walking-aid%20.html>.
- (8) Filipeflop. (2018). Real time clock. Consultado a 22 de abril de 2018. Disponível em <https://www.filipeflop.com/produto/real-time-clock-rtc-ds1307/>.
- (9) HBM. (2019). Como uma célula de carga trabalha. Consultado a 10 de março de 2019. Disponível em <https://www.hbm.com/pt/6768/como-uma-celula-de-carga-trabalha/>.
- (10) Instituto Newton C. Braga. (2011). Como funcionam os leds. Consultado a 27 de julho de 2018. Disponível em <http://www.newtoncbraga.com.br/index.php/como-funciona/733-como-funcionam-os-leds-art096>.
- (11) Instituto Superior de Economia e Gestão. (2018). O envelhecimento da população portuguesa. Consultado a 11 de agosto de 2018. Disponível em <https://medium.com/@isegjbc/o-envelhecimento-da-popula%C3%A7%C3%A3o-portuguesa-8426628cf14c>.
- (12) ITEAD. (2018). Stackable SD card shield 3.0. Consultado a 22 de abril de 2018. Disponível em https://www.itead.cc/wiki/Stackable_SD_Card_shield_V3.0.

-
- (13) Loja Ortopedica. (2017). Andarilho com três rodas. Consultado a 22 de novembro de 2017. Disponível em <https://lojaortopedica.pt/pt/andarilho-com-3-rodas>.
 - (14) Mais que cuidar. (2017). Andarilho simples e fixo. Consultado a 22 de novembro de 2017. Disponível em <https://www.maisquecuidar.com/produto/andarilho-fixo/>.
 - (15) Ortoluz. (2017). Andarilho com rodas dianteiras e apoios traseiros antiderrapantes fixo. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortoluz.com/product/andarilho-com-duas-rodas-actio-invacare>.
 - (16) Ortopedia Marques. (2017). Andarilho articulado com dois apoios de mãos. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortopediamarques.pt/mobilidade-reduzida/514-andarilho-articulado-dois-apoios.html>.
 - (17) Ortopedia Senhor Andarilho. (2017). Andarilho simples e articulado. Consultado a 22 de novembro de 2017. Disponível em <https://www.ortopedia-sr-andarilho.pt/auxiliares-de-marcha>.
 - (18) Portal da célula de carga. (2019). Célula de carga à flexão. Consultado a 10 de março de 2019. Disponível em <http://www.celuladecarga.com.br/portal/?p=112>.
 - (19) RESNA Student Design Competition. (2017). Andarilho equipado com punhos que avisa da má postura do paciente. Consultado a 06 de abril de 2018. Disponível em <http://aac-lerc.psu.edu/wordpressmu/RESNA-SDC/2017/06/18/biofeedback-walker-handle-weight-distribution-monitoring-system-for-patients-with-parkinsons-disease-arizona-state-university/>.
 - (20) Ries, E. F. (1965). U.S. Patent No. 3,165,112. Washington, DC: U.S. Patent Office.
 - (21) Robb, W. C. (1953). U.S. Patent No. 2,656,874. Washington, DC: U.S. Patent Office.
 - (22) SILVA, André; LOPES, Artur. (2014). Kit para andarilhos simples. Relatório da unidade curricular de projeto. Coimbra: Instituto Superior de Engenharia de Coimbra.
 - (23) Sciencedaily. (2015). Andarilho que analisa a condição física e regista atividades diárias. Consultado a 06 de abril de 2018. Disponível em <https://www.sciencedaily.com/releases/2015/10/151001095419.htm>.
 - (24) Smith, A. A. (1970). U.S. Patent No. 3,517,677. Washington, DC: U.S. Patent Office.
 - (25) Sousa, J. (2017). Cinesiologia e Biomecânica do ciclo da marcha. Consultado a 19 de Outubro de 2019. Disponível em <https://www.treinoemfoco.com.br/qualificando-seu-treino/cinesiologia-biomecanica-da-corrida/>
 - (26) Tassino. (2012). O que é biofeedback? Consultado a 6 de Novembro de 2018. Disponível em http://www.tassino.com.br/biofeedback_oquee.htm.
 - (27) Techcrunch. (2016). Andarilho equipado com câmaras à frente e monitor. Consultado a 06 de abril de 2018. Disponível em <https://techcrunch.com/2016/05/31/depth-sensing-walker-monitors-and-connects-the-elderly-and-mobility-impaired/?guccounter=1>.

-
- (28) TE Connectivity. (2019). Compression load cell. Consultado a 20 de março de 2019. Disponível em <https://www.te.com/global-en/product-CAT-FSE0001.html>.
- (29) Tecmundo. (2009). Acelerómetro. Consultado a 05 de junho de 2018. Disponível em <https://www.tecmundo.com.br/curiosidade/2652-o-que-e-um-acelerometro-.htm>.
- (30) Electronica. (2015). Diodo emissor de luz. Consultado a 29 de julho de 2018. Disponível em <https://www.electronica-pt.com/led>.

6. ANEXOS

6.1. Anexo 1 – Programação em arduino do andarilho instrumentado 1

```
// Stéphane Lopes
// Andarilho instrumentado
// Trabalho projeto de mestrado em Engenharia Mecânica
// a21220030@isec.pt
// stephanelopes22@hotmail.com

// bibliotecas
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <RTClib.h>
#include <MPU6050.h>
#include <I2Cdev.h>

// Incluir relógio
RTC_DS1307 RTC;
char buf1[20];
char buf2[20];
const int RTC_ADDR = 0x50; // Endereço I2C do relógio

const int MPU_ADDR = 0x69; // Endereço I2C do MPU-6050 (acelerómetro)
int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variáveis do acelerómetro para memória
int16_t gyro_x, gyro_y, gyro_z; // variables do giroscópio para memória
int16_t temperature; // variables da temperatura para memória
MPU6050 mpu(0x69); // Endereço do acelerómetro+giroscópio

int accx; // limites para os ângulos de oscilação lateral
int accz; // limites para os ângulos de oscilação frente

// Led's nas portas digitais de 22 a 31 (Led's do lado esquerdo e lado direito)
#define led1 31
#define led2 30
#define led3 29
#define led4 28
#define led5 27
#define led6 22
#define led7 23
```

```
#define led8 24
#define led9 25
#define led10 26

// Led's nas portas digitais de 34 a 39 (Led's do meio)
#define led11 37
#define led12 36
#define led13 34
#define led14 35
#define led15 33
#define led16 32

// Níveis de inclinação do eixo x (do lado esquerdo e lado direito)
#define LEVEL1 300
#define LEVEL2 600
#define LEVEL3 900
#define LEVEL4 1200
#define LEVEL5 1500

// Níveis de inclinação do eixo z (ao meio)
#define LEVEL6 14000
#define LEVEL7 15000
#define LEVEL8 16000
#define LEVEL9 17000
#define LEVEL10 18000
#define LEVEL11 19000

char tmp_str[7]; // Variável temporária usada na função de converter

char* convert_int16_to_str(int16_t i)
{ // converte int16 em string
  sprintf(tmp_str, "%6d", i);
  return tmp_str;
}

// sensor de pressão no apoio esquerdo e atrás no pino 9
float s_esq_atras=A11;
float s_esq_atras1;
float s_esq_atras2;
float v_esq_atras;
float v_esq_atras1;
float f_esq_atras;
```

```
float f_esq_atras1;
float f_esq_atras2;
float f_esq_atras3;

// sensor de pressão no apoio direito e atras no pino 10
float s_dir_atras=A9;
float s_dir_atras1;
float s_dir_atras2;
float v_dir_atras;
float v_dir_atras1;
float f_dir_atras;
float f_dir_atras1;
float f_dir_atras2;
float f_dir_atras3;

// sensor de pressão no apoio esquerdo e a frente no pino 11
float s_esq_frente=A10;
float s_esq_frente1;
float s_esq_frente2;
float v_esq_frente;
float v_esq_frente1;
float f_esq_frente;
float f_esq_frente1;
float f_esq_frente2;
float f_esq_frente3;

// sensor de pressão no apoio direito e a frente no pino 12
float s_dir_frente=A8;
float s_dir_frente1;
float s_dir_frente2;
float v_dir_frente;
float v_dir_frente1;
float f_dir_frente;
float f_dir_frente1;
float f_dir_frente2;
float f_dir_frente3;

// Cartão de memória SD no pino 4
int SDcard=4;
File ANDARILHO;

int x;
```

```
void setup() {
  Serial.begin(9600); // ligar a porta
  Wire.begin();

  pinMode(10, OUTPUT); // definição da porta 10
  SD.begin(SDcard); // ligar o cartão de memoria

  // ligar relógio
  //RTC.begin();
  //RTC.adjust(DateTime(F(__DATE__),F(__TIME__)));

  // definição das portas dos LED's do lado esquerdo e lado direito
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  pinMode(led5,OUTPUT);
  pinMode(led6,OUTPUT);
  pinMode(led7,OUTPUT);
  pinMode(led8,OUTPUT);
  pinMode(led9,OUTPUT);
  pinMode(led10,OUTPUT);
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,HIGH);
  digitalWrite(led7,HIGH);
  digitalWrite(led8,HIGH);
  digitalWrite(led9,HIGH);
  digitalWrite(led10,HIGH);

  // definição das portas dos LED's do meio
  pinMode(led11,OUTPUT);
  pinMode(led12,OUTPUT);
  pinMode(led13,OUTPUT);
  pinMode(led14,OUTPUT);
  pinMode(led15,OUTPUT);
  pinMode(led16,OUTPUT);
  digitalWrite(led11,HIGH);
```

```

digitalWrite(led12,HIGH);
digitalWrite(led13,HIGH);
digitalWrite(led14,HIGH);
digitalWrite(led15,HIGH);
digitalWrite(led16,HIGH);

s_esq_atras1 = analogRead(s_esq_atras);
v_esq_atras = ((s_esq_atras1*4)/1023); // Tensão = (valorsensor * (4 / 1023.0) (vai desde 0 -
1023) numa tensão (0.5 - 4.5V):
f_esq_atras = ((112.23*v_esq_atras)-56.651); // Equação da recta do sensor Força = Tensão -
b (Recta retirada da calibração do sensor)

s_dir_atras1 = analogRead(s_dir_atras);
v_dir_atras = ((s_dir_atras1*4)/1023);
f_dir_atras = ((112.65*v_dir_atras)-61.383);

s_esq_frente1 = analogRead(s_esq_frente);
v_esq_frente = ((s_esq_frente1*4)/1023);
f_esq_frente = ((112.14*v_esq_frente)-55.45);

s_dir_frente1 = analogRead(s_dir_frente);
v_dir_frente = ((s_dir_frente1*4)/1023);
f_dir_frente = ((112.41*v_dir_frente)-58.791);

Wire.begin();
Wire.beginTransmission(MPU_ADDR); // Começa a transmissão para o endereço I2C
Wire.write(0x6B); // registo PWR_MGMT_1
Wire.write(0); // liga o MPU-6050
Wire.endTransmission(true);
}

void loop() {

Wire.beginTransmission(MPU_ADDR);
Wire.write(0x3B); // Começa o registo 0x3B
Wire.endTransmission(false); //
Wire.requestFrom(MPU_ADDR, 7*2, true); // requiere um total de 7*2=14 registos

// Criação de offsets no acelerómetro + giroscópio para estar sempre calibrado
mpu.setXAccelOffset(-167);
mpu.setYAccelOffset(-3590);
mpu.setZAccelOffset(3873);

```

```

mpu.setXGyroOffset(-38);
mpu.setYGyroOffset(-16);
mpu.setZGyroOffset(20);

// Leitura dos valores do acelerómetro + giroscópio
accelerometer_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B
(ACCEL_XOUT_H) and 0x3C (ACCEL_XOUT_L)
accelerometer_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D
(ACCEL_YOUT_H) and 0x3E (ACCEL_YOUT_L)
accelerometer_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F
(ACCEL_ZOUT_H) and 0x40 (ACCEL_ZOUT_L)
gyro_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x43 (GYRO_XOUT_H) and
0x44 (GYRO_XOUT_L)
gyro_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x45 (GYRO_YOUT_H) and
0x46 (GYRO_YOUT_L)
gyro_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x47 (GYRO_ZOUT_H) and
0x48 (GYRO_ZOUT_L)

// Conversão dos valores obtidos em string para serem visualizados
Serial.println(" ");
Serial.print("aX = "); Serial.print(convert_int16_to_str(accelerometer_x));
Serial.print(" | aY = "); Serial.print(convert_int16_to_str(accelerometer_y));
Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(accelerometer_z));
Serial.print(" | gX = "); Serial.print(convert_int16_to_str(gyro_x));
Serial.print(" | gY = "); Serial.print(convert_int16_to_str(gyro_y));
Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(gyro_z));
Serial.println();

// comparações para acender e/ou desligar os led's consoante a inclinação
if (accelerometer_x < -LEVEL1)
{
  if(accelerometer_x > -LEVEL2) // acende led 5
  {
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
  }
}

```

```
    digitalWrite(led10,HIGH);
  }
else if (accelerometer_x > -LEVEL3) // acende led 4 e 5
  {
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
  }
else if (accelerometer_x > -LEVEL4) // acende led 3, 4 e 5
  {
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
  }
else if (accelerometer_x > -LEVEL5) // acende led 2, 3, 4 e 5
  {
    digitalWrite(led1,HIGH);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
  }
else // acendes todos os led's do lado esquerdo
```

```
{
  digitalWrite(led1,LOW);
  digitalWrite(led2,LOW);
  digitalWrite(led3,LOW);
  digitalWrite(led4,LOW);
  digitalWrite(led5,LOW);
  digitalWrite(led6,HIGH);
  digitalWrite(led7,HIGH);
  digitalWrite(led8,HIGH);
  digitalWrite(led9,HIGH);
  digitalWrite(led10,HIGH);
}
}
else
{
  if (accelerometer_x > LEVEL1)
  {
    if (accelerometer_x < LEVEL2) // acende led 6
    {
      digitalWrite(led1,HIGH);
      digitalWrite(led2,HIGH);
      digitalWrite(led3,HIGH);
      digitalWrite(led4,HIGH);
      digitalWrite(led5,HIGH);
      digitalWrite(led6,LOW);
      digitalWrite(led7,HIGH);
      digitalWrite(led8,HIGH);
      digitalWrite(led9,HIGH);
      digitalWrite(led10,HIGH);
    }
    else if (accelerometer_x < LEVEL3) // acende led 6 e 7
    {
      digitalWrite(led1,HIGH);
      digitalWrite(led2,HIGH);
      digitalWrite(led3,HIGH);
      digitalWrite(led4,HIGH);
      digitalWrite(led5,HIGH);
      digitalWrite(led6,LOW);
      digitalWrite(led7,LOW);
      digitalWrite(led8,HIGH);
      digitalWrite(led9,HIGH);
      digitalWrite(led10,HIGH);
    }
  }
}
```

```
    }
else if (accelerometer_x < LEVEL4) // acende led 6, 7 e 8
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    digitalWrite(led5,HIGH);
    digitalWrite(led6,LOW);
    digitalWrite(led7,LOW);
    digitalWrite(led8,LOW);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
}
else if (accelerometer_x < LEVEL5) // acende led 6, 7, 8 e 9
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    digitalWrite(led5,HIGH);
    digitalWrite(led6,LOW);
    digitalWrite(led7,LOW);
    digitalWrite(led8,LOW);
    digitalWrite(led9,LOW);
    digitalWrite(led10,HIGH);
}
else // acende todos os led's do lado direito
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    digitalWrite(led5,HIGH);
    digitalWrite(led6,LOW);
    digitalWrite(led7,LOW);
    digitalWrite(led8,LOW);
    digitalWrite(led9,LOW);
    digitalWrite(led10,LOW);
}
}
else // desligar LEDs
```

```
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,HIGH);
  digitalWrite(led7,HIGH);
  digitalWrite(led8,HIGH);
  digitalWrite(led9,HIGH);
  digitalWrite(led10,HIGH);
}
```

```
if (accelerometer_z > LEVEL8 && accelerometer_z < LEVEL9) // Comparação para
desligar os led's do meio
```

```
{
  digitalWrite(led11,HIGH);
  digitalWrite(led12,HIGH);
  digitalWrite(led13,HIGH);
  digitalWrite(led14,HIGH);
  digitalWrite(led15,HIGH);
  digitalWrite(led16,HIGH);
}
```

```
else if (accelerometer_z > LEVEL9) // acende led 13
```

```
{
  if (accelerometer_z < LEVEL10)
  {
    digitalWrite(led11,HIGH);
    digitalWrite(led12,HIGH);
    digitalWrite(led13,LOW);
    digitalWrite(led14,HIGH);
    digitalWrite(led15,HIGH);
    digitalWrite(led16,HIGH);
  }
}
```

```
else if (accelerometer_z < LEVEL11) // acende led 12 e 13
```

```
{
  digitalWrite(led11,HIGH);
  digitalWrite(led12,LOW);
  digitalWrite(led13,LOW);
  digitalWrite(led14,HIGH);
}
```

```
    digitalWrite(led15,HIGH);
    digitalWrite(led16,HIGH);
  }
else // acende os 3 led's da frente
  {
    digitalWrite(led11,LOW);
    digitalWrite(led12,LOW);
    digitalWrite(led13,LOW);
    digitalWrite(led14,HIGH);
    digitalWrite(led15,HIGH);
    digitalWrite(led16,HIGH);
  }
}
else
  {
    if (accelerometer_z < LEVEL8)
      {
        if (accelerometer_z > LEVEL7) // acende led 14
          {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,HIGH);
            digitalWrite(led16,HIGH);
          }
        else if (accelerometer_z > LEVEL6) // acende led 14 e 15
          {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,LOW);
            digitalWrite(led16,HIGH);
          }
        else // acende os 3 led's de trás
          {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,LOW);
          }
        }
    }
```

```

        digitalWrite(led16,LOW);
    }
}
}

// Vai buscar a data e hora e imprimi na porta COM
DateTime now=RTC.now();
sprintf(buf1, "%02d:%02d:%02d", now.hour(), now.minute(), now.second());
sprintf(buf2, "%02d/%02d/%02d", now.day(), now.month(), now.year());
Serial.println(" ");
Serial.print(F("Time: "));
Serial.println(buf1);
Serial.print(F("Date: "));
Serial.println(buf2);
Serial.println(" ");

// Imprime os valores do sensor esquerdo e traseiro
s_esq_atras2 = analogRead(s_esq_atras);
v_esq_atras1 = ((s_esq_atras2*4)/1023); // Tensão = (valorsensor * (4 / 1023.0) (vai desde 0
- 1023) numa tensão (0.5 - 4.5V):
f_esq_atras1 = ((112.23*v_esq_atras1)-56.651); // Equação da recta do sensor Força =
Tensão - b (Recta retirada da calibração do sensor)
f_esq_atras2 = (f_esq_atras1-f_esq_atras);
if (f_esq_atras2 < 0)
{
    Serial.print("Apoio da esquerda traseiro: ");Serial.print("0.00");Serial.print(" N ");
    Serial.println(" ");
    f_esq_atras3=0;
}
else
{
    Serial.print("Apoio da esquerda traseiro: ");Serial.print(f_esq_atras2,2);Serial.print(" N
");//Serial.print(f_esq_atras1,2);Serial.print(" ");Serial.print(f_esq_atras,2);
//Serial.print(s_esq_atras1,2);Serial.print(" ");Serial.print(v_esq_atras,3);
    Serial.println(" ");
    f_esq_atras3=(f_esq_atras2);
}

// Imprime os valores do sensor direito e traseiro
s_dir_atras2 = analogRead(s_dir_atras);
v_dir_atras1 = ((s_dir_atras2*4)/1023);

```

```

f_dir_atras1 = ((112.65*v_dir_atras1)-61.383);
f_dir_atras2 = (f_dir_atras1-f_dir_atras);
if (f_dir_atras2 < 0)
{
  Serial.print("Apoio da direita traseiro: ");Serial.print("0.00");Serial.print(" N ");
  Serial.println(" ");
  f_dir_atras3=0;
}
else
{
  Serial.print("Apoio da direita traseiro: ");Serial.print(f_dir_atras2,2);Serial.print(" N
");//Serial.print(f_dir_atras1,2);Serial.print(" ");Serial.print(f_dir_atras,2);
  //Serial.print(s_dir_atras1,2);Serial.print(" ");Serial.print(v_dir_atras,2);
  Serial.println(" ");
  f_dir_atras3=(f_dir_atras2);
}

// Imprime os valores do sensor esquerdo e frontal
s_esq_frente2 = analogRead(s_esq_frente);
v_esq_frente1 = ((s_esq_frente2*4)/1023);
f_esq_frente1 = ((112.14*v_esq_frente1)-55.45);
f_esq_frente2 = (f_esq_frente1-f_esq_frente);
if (f_esq_frente2 < 0)
{
  Serial.print("Apoio da esquerda frente: ");Serial.print("0.00");Serial.print(" N ");
  Serial.println(" ");
  f_esq_frente3=0;
}
else
{
  Serial.print("Apoio da esquerda frente: ");Serial.print(f_esq_frente2,2);Serial.print(" N
");//Serial.print(f_esq_frente1,2);Serial.print(" ");Serial.print(f_esq_frente,2);
  //Serial.print(s_esq_frente1,2);Serial.print(" ");Serial.print(v_esq_frente,2);
  Serial.println(" ");
  f_esq_frente3=(f_esq_frente2);
}

// Imprime os valores do sensor direito e frontal
s_dir_frente2 = analogRead(s_dir_frente);
v_dir_frente1 = ((s_dir_frente2*4)/1023);
f_dir_frente1 = ((112.41*v_dir_frente1)-58.791);
f_dir_frente2 = (f_dir_frente1-f_dir_frente);

```

```

if (f_dir_frente2 < 0)
{
  Serial.print("Apoio da direita frente: ");Serial.print("0.00");Serial.print(" N ");
  Serial.println(" ");
  f_dir_frente3=0;
}
else
{
  Serial.print("Apoio da direita frente: ");Serial.print(f_dir_frente2);Serial.print(" N
");//Serial.print(f_dir_frente1,2);Serial.print(" ");Serial.print(f_dir_frente,2);
  //Serial.print(s_dir_frente1,2);Serial.print(" ");Serial.print(v_dir_frente,2);
  Serial.println(" ");
  f_dir_frente3=(f_dir_frente2);
}

accx=map(accelerometer_x,0,2000,0,7); // limites para os angulos de oscilação lateral
accz=map(accelerometer_z,16300,11500,0,13); // limites para os angulos de oscilação frente

Serial.print("Angulo lateral: ");Serial.print(convert_int16_to_str(accx));Serial.print(" °");
Serial.println(" ");
Serial.print("Angulo frente: ");Serial.print(convert_int16_to_str(accz));Serial.print(" °");
Serial.println(" ");

if (SD.exists("An.csv"))
{
  ANDARILHO=SD.open("An.csv", FILE_WRITE); //Abrir o ficheiro
  if (ANDARILHO)
  {

ANDARILHO.print(buf1);ANDARILHO.print(";");ANDARILHO.print(buf2);ANDARILHO
.print(";");ANDARILHO.print(convert_int16_to_str(f_dir_atras3));ANDARILHO.print(";");
ANDARILHO.print(convert_int16_to_str(f_esq_atras3));ANDARILHO.print(";");

ANDARILHO.print(convert_int16_to_str(f_esq_frente3));ANDARILHO.print(";");ANDARI
LHO.print(convert_int16_to_str(f_dir_frente3));ANDARILHO.print(";");ANDARILHO.print
(convert_int16_to_str(accx));ANDARILHO.print(";");ANDARILHO.print(convert_int16_to_
str(accz));
    ANDARILHO.println(" ");
    ANDARILHO.close();
  }
}
else

```

```

{
  ANDARILHO=SD.open("An.csv", FILE_WRITE); //Abrir o ficheiro
  if (ANDARILHO)
    {
      ANDARILHO.println("ANDARILHO");
      ANDARILHO.println(" ");

ANDARILHO.print("Hora");ANDARILHO.print(";");ANDARILHO.print("Data");ANDARI
LHO.print(";");ANDARILHO.print("Apoio      da      direita      traseiro:
[N]");ANDARILHO.print(";");ANDARILHO.print("Apoio      da      esquerda      traseiro:
[N]");ANDARILHO.print(";");
      ANDARILHO.print("Apoio      da      esquerda      frente:
[N]");ANDARILHO.print(";");ANDARILHO.print("Apoio      da      direita      frente:
[N]");ANDARILHO.print(";");ANDARILHO.print("Angulo      de      lado:
[°]");ANDARILHO.print(";");ANDARILHO.print("Angulo de frente: [°]");
      ANDARILHO.println(" ");

ANDARILHO.print(buf1);ANDARILHO.print(";");ANDARILHO.print(buf2);ANDARILHO
.print(";");ANDARILHO.print(convert_int16_to_str(f_dir_atras3));ANDARILHO.print(";");
ANDARILHO.print(convert_int16_to_str(f_esq_atras3));ANDARILHO.print(";");

ANDARILHO.print(convert_int16_to_str(f_esq_frente3));ANDARILHO.print(";");ANDARI
LHO.print(convert_int16_to_str(f_dir_frente3));ANDARILHO.print(";");ANDARILHO.print
(convert_int16_to_str(accx));ANDARILHO.print(";");ANDARILHO.print(convert_int16_to_
str(accz));
      ANDARILHO.println(" ");
      ANDARILHO.close();
    }
}

// atraso
delay(1000);
}

```

6.2. Anexo 2 – Programação em arduino do andarilho instrumentado 2

```
// Stéphane Lopes
// Andarilho instrumentado
// Trabalho projeto de mestrado em Engenharia Mecânica
// a21220030@isec.pt
// stephanelopes22@hotmail.com

// bibliotecas
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <RTClib.h>
#include <MPU6050.h>
#include <I2Cdev.h>

// Incluir relógio
RTC_DS1307 RTC;
char buf1[20];
char buf2[20];
const int RTC_ADDR = 0x68; // Endereço I2C do relógio

const int MPU_ADDR = 0x69; // Endereço I2C do MPU-6050 (acelerómetro)
int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variáveis do acelerómetro para memória
int16_t gyro_x, gyro_y, gyro_z; // variables do giroscópio para memória
int16_t temperature; // variables da temperatura para memória
MPU6050 mpu(0x69); // Endereço do acelerómetro+giroscópio

int accx; // limites para os ângulos de oscilação lateral
int accz; // limites para os ângulos de oscilação frente

// Led's nas portas digitais de 22 a 31 (Led's do lado esquerdo e lado direito)
#define led1 32
#define led2 31
#define led3 30
#define led4 29
#define led5 28
#define led6 23
#define led7 24
#define led8 25
#define led9 26
```

```
#define led10 27

// Led's nas portas digitais de 34 a 39 (Led's do meio)
#define led11 34
#define led12 35
#define led13 36
#define led14 37
#define led15 38
#define led16 39

// Níveis de inclinação do eixo x (do lado esquerdo e lado direito)
#define LEVEL1 300
#define LEVEL2 600
#define LEVEL3 900
#define LEVEL4 1200
#define LEVEL5 1500

// Níveis de inclinação do eixo z (ao meio)
#define LEVEL6 14000
#define LEVEL7 15000
#define LEVEL8 16000
#define LEVEL9 17000
#define LEVEL10 18000
#define LEVEL11 19000

char tmp_str[7]; // Variavel temporária usada na função de converter

char* convert_int16_to_str(int16_t i)
{ // converte int16 em string
  sprintf(tmp_str, "%6d", i);
  return tmp_str;
}

// sensor de pressão no apoio esquerdo e atras no pino 9
float s_esq_atras=A9;
float s_esq_atras1;
float s_esq_atras2;
float v_esq_atras;
float v_esq_atras1;
float f_esq_atras;
float f_esq_atras1;
float f_esq_atras2;
```

```
float f_esq_atras3;

// sensor de pressão no apoio direito e atras no pino 10
float s_dir_atras=A11;
float s_dir_atras1;
float s_dir_atras2;
float v_dir_atras;
float v_dir_atras1;
float f_dir_atras;
float f_dir_atras1;
float f_dir_atras2;
float f_dir_atras3;

// sensor de pressão no apoio esquerdo e a frente no pino 11
float s_esq_frente=A8;
float s_esq_frente1;
float s_esq_frente2;
float v_esq_frente;
float v_esq_frente1;
float f_esq_frente;
float f_esq_frente1;
float f_esq_frente2;
float f_esq_frente3;

// sensor de pressão no apoio direito e a frente no pino 12
float s_dir_frente=A10;
float s_dir_frente1;
float s_dir_frente2;
float v_dir_frente;
float v_dir_frente1;
float f_dir_frente;
float f_dir_frente1;
float f_dir_frente2;
float f_dir_frente3;

// Cartão de memória SD no pino 4
int SDcard=4;
File ANDARILHO;

int x;

void setup() {
```

```
Serial.begin(9600); // ligar a porta
Wire.begin();

pinMode(10, OUTPUT); // definição da porta 10
SD.begin(SDcard); // ligar o cartão de memória

// ligar relógio
//RTC.begin();
//RTC.adjust(DateTime(F(__DATE__),F(__TIME__)));

// definição das portas dos LED's do lado esquerdo e lado direito
pinMode(led1,OUTPUT);
pinMode(led2,OUTPUT);
pinMode(led3,OUTPUT);
pinMode(led4,OUTPUT);
pinMode(led5,OUTPUT);
pinMode(led6,OUTPUT);
pinMode(led7,OUTPUT);
pinMode(led8,OUTPUT);
pinMode(led9,OUTPUT);
pinMode(led10,OUTPUT);
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
digitalWrite(led3,HIGH);
digitalWrite(led4,HIGH);
digitalWrite(led5,HIGH);
digitalWrite(led6,HIGH);
digitalWrite(led7,HIGH);
digitalWrite(led8,HIGH);
digitalWrite(led9,HIGH);
digitalWrite(led10,HIGH);

// definição das portas dos LED's do meio
pinMode(led11,OUTPUT);
pinMode(led12,OUTPUT);
pinMode(led13,OUTPUT);
pinMode(led14,OUTPUT);
pinMode(led15,OUTPUT);
pinMode(led16,OUTPUT);
digitalWrite(led11,HIGH);
digitalWrite(led12,HIGH);
digitalWrite(led13,HIGH);
```

```

digitalWrite(led14,HIGH);
digitalWrite(led15,HIGH);
digitalWrite(led16,HIGH);

s_esq_atras1 = analogRead(s_esq_atras);
v_esq_atras = ((s_esq_atras1*4)/1023); // Tensão = (valorsensor * (4 / 1023.0) (vai desde 0 -
1023) numa tensão (0.5 - 4.5V):
f_esq_atras = ((112.23*v_esq_atras)-56.651); // Equação da recta do sensor Força = Tensão -
b (Recta retirada da calibração do sensor)

s_dir_atras1 = analogRead(s_dir_atras);
v_dir_atras = ((s_dir_atras1*4)/1023);
f_dir_atras = ((112.65*v_dir_atras)-61.383);

s_esq_frente1 = analogRead(s_esq_frente);
v_esq_frente = ((s_esq_frente1*4)/1023);
f_esq_frente = ((112.14*v_esq_frente)-55.45);

s_dir_frente1 = analogRead(s_dir_frente);
v_dir_frente = ((s_dir_frente1*4)/1023);
f_dir_frente = ((112.41*v_dir_frente)-58.791);

Wire.begin();
Wire.beginTransmission(MPU_ADDR); // Começa a transmissão para o endereço I2C
Wire.write(0x6B); // registo PWR_MGMT_1
Wire.write(0); // liga o MPU-6050
Wire.endTransmission(true);
}

void loop() {

Wire.beginTransmission(MPU_ADDR);
Wire.write(0x3B); // Começa o registo 0x3B
Wire.endTransmission(false); //
Wire.requestFrom(MPU_ADDR, 7*2, true); // requer um total de 7*2=14 registos

// Criação de offsets no acelerómetro + giroscópio para estar sempre calibrado
mpu.setXAccelOffset(-3040);
mpu.setYAccelOffset(-2838);
mpu.setZAccelOffset(2850);
mpu.setXGyroOffset(155);
mpu.setYGyroOffset(25);

```

```

mpu.setZGyroOffset(-46);

// Leitura dos valores do acelerómetro + giroscópio
accelerometer_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B
(ACCEL_XOUT_H) and 0x3C (ACCEL_XOUT_L)
accelerometer_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D
(ACCEL_YOUT_H) and 0x3E (ACCEL_YOUT_L)
accelerometer_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F
(ACCEL_ZOUT_H) and 0x40 (ACCEL_ZOUT_L)
gyro_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x43 (GYRO_XOUT_H) and
0x44 (GYRO_XOUT_L)
gyro_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x45 (GYRO_YOUT_H) and
0x46 (GYRO_YOUT_L)
gyro_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x47 (GYRO_ZOUT_H) and
0x48 (GYRO_ZOUT_L)

// Conversão dos valores obtidos em string para serem visualizados
Serial.println(" ");
Serial.print("aX = "); Serial.print(convert_int16_to_str(accelerometer_x));
Serial.print(" | aY = "); Serial.print(convert_int16_to_str(accelerometer_y));
Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(accelerometer_z));
Serial.print(" | gX = "); Serial.print(convert_int16_to_str(gyro_x));
Serial.print(" | gY = "); Serial.print(convert_int16_to_str(gyro_y));
Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(gyro_z));
Serial.println();

// comparações para acender e/ou desligar os led's consoante a inclinação
if (accelerometer_x < -LEVEL1)
{
    if(accelerometer_x > -LEVEL2) // acende led 5
    {
        digitalWrite(led1,HIGH);
        digitalWrite(led2,HIGH);
        digitalWrite(led3,HIGH);
        digitalWrite(led4,HIGH);
        digitalWrite(led5,LOW);
        digitalWrite(led6,HIGH);
        digitalWrite(led7,HIGH);
        digitalWrite(led8,HIGH);
        digitalWrite(led9,HIGH);
        digitalWrite(led10,HIGH);
    }
}

```

```
else if (accelerometer_x > -LEVEL3) // acende led 4 e 5
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
}
else if (accelerometer_x > -LEVEL4) // acende led 3, 4 e 5
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
}
else if (accelerometer_x > -LEVEL5) // acende led 2, 3, 4 e 5
{
    digitalWrite(led1,HIGH);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    digitalWrite(led5,LOW);
    digitalWrite(led6,HIGH);
    digitalWrite(led7,HIGH);
    digitalWrite(led8,HIGH);
    digitalWrite(led9,HIGH);
    digitalWrite(led10,HIGH);
}
else // acendes todos os led's do lado esquerdo
{
    digitalWrite(led1,LOW);
```

```
digitalWrite(led2,LOW);
digitalWrite(led3,LOW);
digitalWrite(led4,LOW);
digitalWrite(led5,LOW);
digitalWrite(led6,HIGH);
digitalWrite(led7,HIGH);
digitalWrite(led8,HIGH);
digitalWrite(led9,HIGH);
digitalWrite(led10,HIGH);
}
}
else
{
if (accelerometer_x > LEVEL1)
{
if (accelerometer_x < LEVEL2) // acende led 6
{
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
digitalWrite(led3,HIGH);
digitalWrite(led4,HIGH);
digitalWrite(led5,HIGH);
digitalWrite(led6,LOW);
digitalWrite(led7,HIGH);
digitalWrite(led8,HIGH);
digitalWrite(led9,HIGH);
digitalWrite(led10,HIGH);
}
else if (accelerometer_x < LEVEL3) // acende led 6 e 7
{
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
digitalWrite(led3,HIGH);
digitalWrite(led4,HIGH);
digitalWrite(led5,HIGH);
digitalWrite(led6,LOW);
digitalWrite(led7,LOW);
digitalWrite(led8,HIGH);
digitalWrite(led9,HIGH);
digitalWrite(led10,HIGH);
}
else if (accelerometer_x < LEVEL4) // acende led 6, 7 e 8
```

```
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,LOW);
  digitalWrite(led7,LOW);
  digitalWrite(led8,LOW);
  digitalWrite(led9,HIGH);
  digitalWrite(led10,HIGH);
}
else if (accelerometer_x < LEVEL5) // acende led 6, 7, 8 e 9
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,LOW);
  digitalWrite(led7,LOW);
  digitalWrite(led8,LOW);
  digitalWrite(led9,LOW);
  digitalWrite(led10,HIGH);
}
else // acende todos os led's do lado direito
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
  digitalWrite(led4,HIGH);
  digitalWrite(led5,HIGH);
  digitalWrite(led6,LOW);
  digitalWrite(led7,LOW);
  digitalWrite(led8,LOW);
  digitalWrite(led9,LOW);
  digitalWrite(led10,LOW);
}
}
else // desligar LEDs
{
  digitalWrite(led1,HIGH);
```

```
digitalWrite(led2,HIGH);
digitalWrite(led3,HIGH);
digitalWrite(led4,HIGH);
digitalWrite(led5,HIGH);
digitalWrite(led6,HIGH);
digitalWrite(led7,HIGH);
digitalWrite(led8,HIGH);
digitalWrite(led9,HIGH);
digitalWrite(led10,HIGH);
}
}
```

```
if (accelerometer_z > LEVEL8 && accelerometer_z < LEVEL9) // Comparação para
desligar os led's do meio
```

```
{
digitalWrite(led11,HIGH);
digitalWrite(led12,HIGH);
digitalWrite(led13,HIGH);
digitalWrite(led14,HIGH);
digitalWrite(led15,HIGH);
digitalWrite(led16,HIGH);
}
```

```
else if (accelerometer_z > LEVEL9) // acende led 13
```

```
{
if (accelerometer_z < LEVEL10)
{
digitalWrite(led11,HIGH);
digitalWrite(led12,HIGH);
digitalWrite(led13,LOW);
digitalWrite(led14,HIGH);
digitalWrite(led15,HIGH);
digitalWrite(led16,HIGH);
}
```

```
else if (accelerometer_z < LEVEL11) // acende led 12 e 13
```

```
{
digitalWrite(led11,HIGH);
digitalWrite(led12,LOW);
digitalWrite(led13,LOW);
digitalWrite(led14,HIGH);
digitalWrite(led15,HIGH);
digitalWrite(led16,HIGH);
```

```
    }
else // acende os 3 led's da frente
{
    digitalWrite(led11,LOW);
    digitalWrite(led12,LOW);
    digitalWrite(led13,LOW);
    digitalWrite(led14,HIGH);
    digitalWrite(led15,HIGH);
    digitalWrite(led16,HIGH);
}
}
else
{
    if (accelerometer_z < LEVEL8)
    {
        if (accelerometer_z > LEVEL7) // acende led 14
        {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,HIGH);
            digitalWrite(led16,HIGH);
        }
        else if (accelerometer_z > LEVEL6) // acende led 14 e 15
        {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,LOW);
            digitalWrite(led16,HIGH);
        }
        else // acende os 3 led's de trás
        {
            digitalWrite(led11,HIGH);
            digitalWrite(led12,HIGH);
            digitalWrite(led13,HIGH);
            digitalWrite(led14,LOW);
            digitalWrite(led15,LOW);
            digitalWrite(led16,LOW);
        }
    }
}
```

```

    }
}

// Vai buscar a data e hora e imprimir na porta COM
DateTime now=RTC.now();
sprintf(buf1, "%02d:%02d:%02d", now.hour(), now.minute(), now.second());
sprintf(buf2, "%02d/%02d/%02d", now.day(), now.month(), now.year());
Serial.println(" ");
Serial.print(F("Time: "));
Serial.println(buf1);
Serial.print(F("Date: "));
Serial.println(buf2);
Serial.println(" ");

// Imprime os valores do sensor esquerdo e traseiro
s_esq_atras2 = analogRead(s_esq_atras);
v_esq_atras1 = ((s_esq_atras2*4)/1023); // Tensão = (valorsensor * (4 / 1023.0) (vai desde 0
- 1023) numa tensão (0.5 - 4.5V):
f_esq_atras1 = ((112.23*v_esq_atras1)-56.651); // Equação da recta do sensor Força =
Tensão - b (Recta retirada da calibração do sensor)
f_esq_atras2 = (f_esq_atras1-f_esq_atras);
if (f_esq_atras2 < 0)
{
    Serial.print("Apoio da esquerda traseiro: ");Serial.print("0.00");Serial.print(" N ");
    Serial.println(" ");
    f_esq_atras3=0;
}
else
{
    Serial.print("Apoio da esquerda traseiro: ");Serial.print(f_esq_atras2,2);Serial.print(" N
");//Serial.print(f_esq_atras1,2);Serial.print(" ");Serial.print(f_esq_atras,2);
//Serial.print(s_esq_atras1,2);Serial.print(" ");Serial.print(v_esq_atras,3);
    Serial.println(" ");
    f_esq_atras3=(f_esq_atras2);
}

// Imprime os valores do sensor direito e traseiro
s_dir_atras2 = analogRead(s_dir_atras);
v_dir_atras1 = ((s_dir_atras2*4)/1023);
f_dir_atras1 = ((112.65*v_dir_atras1)-61.383);
f_dir_atras2 = (f_dir_atras1-f_dir_atras);
if (f_dir_atras2 < 0)

```

```

{
  Serial.print("Apoio da direita traseiro: ");Serial.print("0.00");Serial.print(" N ");
  Serial.println(" ");
  f_dir_atras3=0;
}
else
{
  Serial.print("Apoio da direita traseiro: ");Serial.print(f_dir_atras2,2);Serial.print(" N
");//Serial.print(f_dir_atras1,2);Serial.print(" ");Serial.print(f_dir_atras,2);
  //Serial.print(s_dir_atras1,2);Serial.print(" ");Serial.print(v_dir_atras,2);
  Serial.println(" ");
  f_dir_atras3=(f_dir_atras2);
}

// Imprime os valores do sensor esquerdo e frontal
s_esq_frente2 = analogRead(s_esq_frente);
v_esq_frente1 = ((s_esq_frente2*4)/1023);
f_esq_frente1 = ((112.14*v_esq_frente1)-55.45);
f_esq_frente2 = (f_esq_frente1-f_esq_frente);
if (f_esq_frente2 < 0)
{
  Serial.print("Apoio da esquerda frente: ");Serial.print("0.00");Serial.print(" N ");
  Serial.println(" ");
  f_esq_frente3=0;
}
else
{
  Serial.print("Apoio da esquerda frente: ");Serial.print(f_esq_frente2,2);Serial.print(" N
");//Serial.print(f_esq_frente1,2);Serial.print(" ");Serial.print(f_esq_frente,2);
  //Serial.print(s_esq_frente1,2);Serial.print(" ");Serial.print(v_esq_frente,2);
  Serial.println(" ");
  f_esq_frente3=(f_esq_frente2);
}

// Imprime os valores do sensor direito e frontal
s_dir_frente2 = analogRead(s_dir_frente);
v_dir_frente1 = ((s_dir_frente2*4)/1023);
f_dir_frente1 = ((112.41*v_dir_frente1)-58.791);
f_dir_frente2 = (f_dir_frente1-f_dir_frente);
if (f_dir_frente2 < 0)
{
  Serial.print("Apoio da direita frente: ");Serial.print("0.00");Serial.print(" N ");

```

```

    Serial.println(" ");
    f_dir_frente3=0;
}
else
{
    Serial.print("Apoio da direita frente: ");Serial.print(f_dir_frente2);Serial.print(" N
");//Serial.print(f_dir_frente1,2);Serial.print(" ");Serial.print(f_dir_frente,2);
    //Serial.print(s_dir_frente1,2);Serial.print(" ");Serial.print(v_dir_frente,2);
    Serial.println(" ");
    f_dir_frente3=(f_dir_frente2);
}

accx=map(accelerometer_x,0,2000,0,7); // limites para os angulos de oscilação lateral
accz=map(accelerometer_z,16300,11500,0,13); // limites para os angulos de oscilação frente

Serial.print("Angulo lateral: ");Serial.print(convert_int16_to_str(accx));Serial.print(" °");
Serial.println(" ");
Serial.print("Angulo frente: ");Serial.print(convert_int16_to_str(accz));Serial.print(" °");
Serial.println(" ");

if (SD.exists("An.csv"))
{
    ANDARILHO=SD.open("An.csv", FILE_WRITE); //Abrir o ficheiro
    if (ANDARILHO)
    {

ANDARILHO.print(buf1);ANDARILHO.print(";");ANDARILHO.print(buf2);ANDARILHO
.print(";");ANDARILHO.print(convert_int16_to_str(f_dir_atras3));ANDARILHO.print(";");
ANDARILHO.print(convert_int16_to_str(f_esq_atras3));ANDARILHO.print(";");

ANDARILHO.print(convert_int16_to_str(f_esq_frente3));ANDARILHO.print(";");ANDARI
LHO.print(convert_int16_to_str(f_dir_frente3));ANDARILHO.print(";");ANDARILHO.print
(convert_int16_to_str(accx));ANDARILHO.print(";");ANDARILHO.print(convert_int16_to_
str(accz));
        ANDARILHO.println(" ");
        ANDARILHO.close();
    }
}
else
{
    ANDARILHO=SD.open("An.csv", FILE_WRITE); //Abrir o ficheiro
    if (ANDARILHO)

```

```

    {
        ANDARILHO.println("ANDARILHO");
        ANDARILHO.println(" ");

ANDARILHO.print("Hora");ANDARILHO.print(";");ANDARILHO.print("Data");ANDARI
LHO.print(";");ANDARILHO.print("Apoio      da      direita      traseiro:
[N]");ANDARILHO.print(";");ANDARILHO.print("Apoio      da      esquerda      traseiro:
[N]");ANDARILHO.print(";");
        ANDARILHO.print("Apoio      da      esquerda      frente:
[N]");ANDARILHO.print(";");ANDARILHO.print("Apoio      da      direita      frente:
[N]");ANDARILHO.print(";");ANDARILHO.print("Angulo      de      lado:
[°]");ANDARILHO.print(";");ANDARILHO.print("Angulo de frente: [°]");
        ANDARILHO.println(" ");

ANDARILHO.print(buf1);ANDARILHO.print(";");ANDARILHO.print(buf2);ANDARILHO
.print(";");ANDARILHO.print(convert_int16_to_str(f_dir_atras3));ANDARILHO.print(";");
ANDARILHO.print(convert_int16_to_str(f_esq_atras3));ANDARILHO.print(";");

ANDARILHO.print(convert_int16_to_str(f_esq_frente3));ANDARILHO.print(";");ANDARI
LHO.print(convert_int16_to_str(f_dir_frente3));ANDARILHO.print(";");ANDARILHO.print
(convert_int16_to_str(accx));ANDARILHO.print(";");ANDARILHO.print(convert_int16_to_
str(accz));
        ANDARILHO.println(" ");
        ANDARILHO.close();
    }
}
// atraso
delay(1000);
}

```