



**Miguel Filipe Batista Prego**

Licenciado em Engenharia Eletrotécnica e de Computadores

## **Deteção e registo automático de irregularidades no asfalto**

Relatório intermédio para obtenção do Grau de Mestre em  
**Engenharia Eletrotécnica e de Computadores**

Orientador: José Manuel Matos Ribeiro da Fonseca, Professor Associado, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

**Setembro, 2016**



# ÍNDICE

<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>1 Estado da arte</b>	<b>1</b>
1.1 Principais metodologias . . . . .	1
1.1.1 Camera vídeo . . . . .	1
1.1.2 Camera vídeo com iluminação artificial . . . . .	3
1.1.3 Sensor ultrassónico . . . . .	5
1.1.4 Acelerómetro . . . . .	6
1.2 Comparação de resultados . . . . .	7
<b>Bibliografia</b>	<b>9</b>



## LISTA DE FIGURAS

1.1	Montagem com duas câmeras . . . . .	2
1.2	Sensor Kinect . . . . .	3
1.3	Matriz de detecção de um buraco . . . . .	4
1.4	Comparação da linha em superfície lisa e num buraco . . . . .	4
1.5	Sensor Ultrassom . . . . .	6
1.6	Direção dos eixos de um acelerómetro . . . . .	7



## LISTA DE TABELAS

1.1	Comparação de resultados de artigos . . . . .	8
1.2	Comparação de resultados com utilização de acelerómetro . . . . .	8





## ESTADO DA ARTE

### 1.1 Principais metodologias

Para que esta dissertação pudesse apresentar vantagens em relação a trabalhos já desenvolvidos, foi necessário fazer uma pesquisa sobre as metodologias propostas para a solução ou diminuição de irregularidades existentes no asfalto, sendo que foram apenas encontrados trabalhos que propõem a detecção de buracos no asfalto, deixando de parte outras irregularidades como juntas de dilatação ou lombas. Após a análise de vários trabalhos relacionados com este problema, foi possível agrupar os trabalhos mais relevantes em categorias referentes às metodologias utilizadas para possíveis soluções, sendo os principais:

1. **Câmera vídeo**
2. **Câmera vídeo com iluminação artificial**
3. **Ultrassom**
4. **Acelerómetro**

Nesta secção serão descritos os trabalhos e as tecnologias utilizadas em cada um deles, bem como as vantagens e desvantagens de cada uma das implementações apresentadas.

#### 1.1.1 Camera vídeo

No que toca às câmeras de vídeo, os trabalhos aqui referidos não são diretamente comparáveis com aquele que foi desenvolvido e é aqui apresentado, uma vez que utiliza uma tecnologia para a detecção e reconhecimento de buracos no asfalto bastante diferente da que foi utilizada nesta dissertação. Ainda assim são trabalhos que se enquadram no tema

e dos quais foram retiradas ideias para metodologias de comunicação entre os vários componentes da montagem. De todos artigos estudados, três são aqueles que mostram abordagens mais interessantes, pois apresentam soluções diferentes entre si. Em Zhang et al. 2014 e Chan et al. 2014 são utilizadas duas câmeras e apresentados algoritmos muito semelhantes, com resultados bastante positivos, sendo possível determinar uma relação entre estes trabalhos devido às várias colaborações já existentes entre os autores. A solução proposta em Zhang et al. 2014 utiliza duas câmeras para criar uma imagem a três dimensões de modo a conseguir detetar um buraco no asfalto mas também calcular a sua profundidade. Cruzando a imagem de ambas as câmeras, é possível encontrar os pontos comuns no asfalto de modo a sobrepor as imagens para o cálculo do desvio que existe entre elas de forma a criar um mapa virtual da superfície e do buraco. Depois da criação do mapa é possível determinar qual o plano da estrada e também os pontos que se encontram abaixo desse plano, onde se encontram localizados os buracos. O sistema é bastante preciso, permitindo a deteção de buracos com profundidades maiores que 4cm mas apresenta problemas de transmissão em tempo real uma vez que é necessário bastante processamento para fazer a sobreposição das imagens e também o mapeamento da superfície.

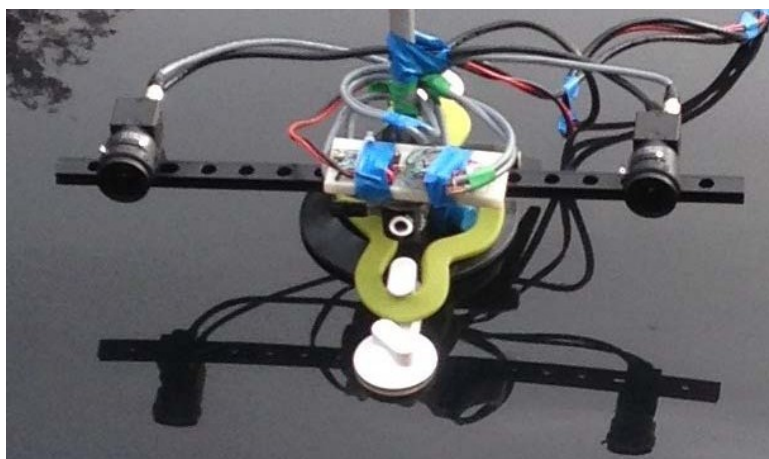


Figura 1.1: Montagem com duas câmeras

Em Chan et al. 2014 é apresentada uma melhoria a este sistema, no que toca ao processamento em tempo real, sendo que foram possíveis avaliar cinco imagens por segundo, uma a cada 200 milissegundos. Juntando estes dois projetos é possível obter-se uma solução viável em termos de processamento e deteção mas que necessita de duas câmeras e um processador com vários núcleos para todo o processamento, tornando assim a sua montagem bastante dispendiosa.

Numa solução diferente, em Moazzam et al. 2013 é descrito um processo de deteção de buracos utilizando um sensor Kinect <sup>1</sup> que só tem uma câmera mas tem um sensor de infravermelhos para determinar a distância a que se encontra de um objeto. O sensor

---

<sup>1</sup><https://developer.microsoft.com/en-us/windows/kinect>

Kinect calcula a distância até um obstáculo e faz o mapeamento dessas distâncias, criando um mapa virtual do solo. De seguida, as coordenadas X e Y do mapa são convertidas para o mundo real, de modo a obter as coordenadas reais. São então feitos diversos cálculos para determinar as áreas de diferentes profundidades do buraco para que seja possível saber o volume que o buraco ocupa no asfalto. Por fim os buracos são classificados com base nas curvas de relação área-profundidade, área ao nível do asfalto e profundidade máxima. Esta solução tem resultados semelhantes aos de Zhang et al. 2014 e Chan et al. 2014 e para a sua implementação são necessários menos cálculos, uma vez que o software Kinect está otimizado. Além disso, embora não seja apresentado, é de esperar que esta opção seja menos dispendiosa que as anteriores graças ao preço do Kinect e que não seja necessário um processador tão potente. Ainda assim existe a grande desvantagem do sistema só ter sido testado de forma imóvel, não existindo dados para a instalação em automóvel.

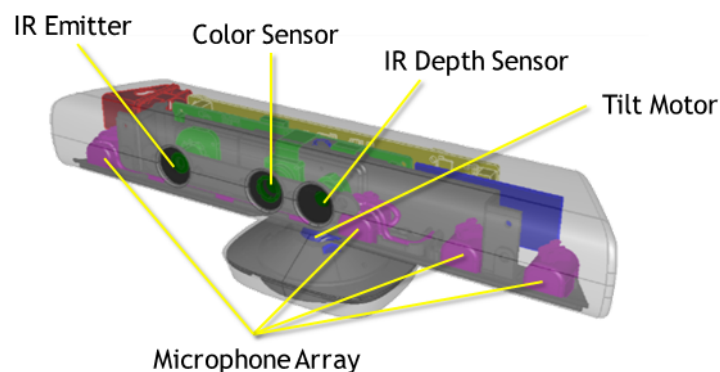


Figura 1.2: Sensor Kinect <sup>2</sup>

### 1.1.2 Camera vídeo com iluminação artificial

Apesar dos artigos apresentados nesta secção utilizarem também câmeras, é interessante separá-los dos anteriores uma vez que a deteção dos buracos é feita de forma diferente, consoante a passagem pelo próprio buraco. Em Yu e Salari 2011 é utilizado um laser vermelho que projeta uma reta no asfalto, perpendicular ao sentido de deslocação do automóvel. Esta reta é então capturada por uma câmara que analisa a imagem e a filtra, através de um filtro de mediana a quatro máscaras, de modo a remover ruído, mantendo detalhe suficiente na imagem. Utilizando depois o método de Otsu, a imagem é binarizada para uma fácil deteção da linha laser. Essa imagem binarizada é então colocada numa matrix X Y, juntamente com as imagens sucessivas para que seja possível mapear um buraco e a sua área de superfície. Utilizando cada imagem e calculando a distância do desvio da linha é também possível calcular a profundidade do buraco, sendo esse um dos parâmetros utilizados nas classificações de todas as deteções. Apesar de mostrar bons resultados, são salientadas nas conclusões que existem algumas perturbações da linha

<sup>2</sup><https://developer.microsoft.com/en-us/windows/kinect>

aquando a passagem por zonas mais danificadas, diminuindo assim a avaliação de uma detecção e é também referida a necessidade de um algoritmo de avaliação robusto que leva a que seja necessário um nível de processamento elevado.

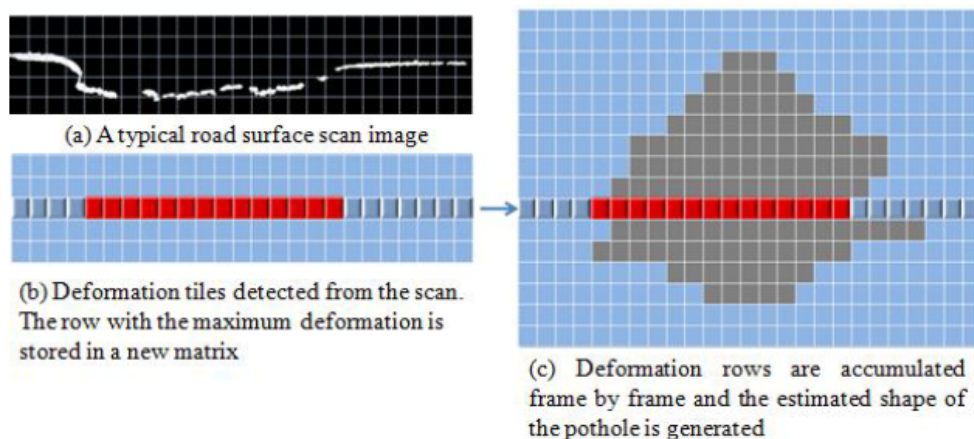


Figura 1.3: Matriz de detecção de um buraco <sup>3</sup>

Já em He et al. 2011 as linhas vermelhas projetadas no solo são provenientes de uma luz *led* de baixa potência, também perpendiculares à estrada. Este sistema utiliza duas câmeras para fazer a detecção da linha, aumentando a precisão no que toca à classificação do buraco quanto à sua profundidade. De modo a diminuir ruído existente na imagem, é utilizado o método de Otsu para detecção da linha, bem como uma avaliação dos *pixels* da vizinhança para que possam ser eliminados falsas detecções, uma vez que a linha é composta por vários *pixels* de cor semelhante. Por fim a coloração da imagem é invertida, e é feita uma dilatação e uma contração para uma melhor definição da linha detetada. Este sistema consegue apresentar erros inferiores a 2mm em termos de profundidade quando comparado com instrumentação de alta precisão para medição de irregularidades.



Figure 2. LED light band on smooth-riding surface.



Figure 3. LED light band in pothole.

Figura 1.4: Comparação da linha em superfície lisa e num buraco <sup>4</sup>

<sup>3</sup>Yu e Salari 2011

<sup>4</sup>He et al. 2011

### 1.1.3 Sensor ultrassônico

Outra abordagem completamente diferente é a utilização de sensores ultrassom. Um sensor ultrassônico é um dispositivo utilizado para medir a distância a um objeto que utiliza ondas sonoras. Este dispositivo mede a distância enviando uma onda sonora com uma frequência específica e aguarda que a onda sonora retorne. Ao gravar o tempo decorrido entre emissão da onda sonora e a onda sonora de retorno, é possível calcular a distância entre o sensor e o objeto. Uma vez que se sabe que o som viaja através do ar a cerca de 344m/s, é possível multiplicar o tempo de viagem da onda sonora por 344 metros para encontrar a distância total da viagem da onda sonora, sendo que este valor representa a distância de ida e volta da onda pelo que para encontrar a distância até o objeto, basta dividir esta distância por dois. Esta é uma opção semelhante à que foi tomada neste projeto na medida em que é quase obrigatório passar por um buraco para fazer a sua detecção, ao contrário da análise de imagens em que é possível evitá-los. Em Hegde et al. 2015 é mostrado um protótipo que permite a detecção de buracos através de um sensor ultrassônico testado num protótipo autónomo que comunica com outro veículo de modo a ambos ajustarem as suas velocidades dependendo da qualidade da estrada, diminuindo-a sempre que são detetados buracos. Para a comunicação foi utilizado um módulo ZigBee que permite estender a comunicação a mais veículos em trabalhos futuros. O sistema apresentou bons resultados em estradas simuladas mas foi necessário fazer uma calibração específica para este protótipo, sendo que é referido nas conclusões a necessidade de um ajuste aos valores de detecção, consoante o veículo onde o sistema se encontra montado. Este fator é um grande inconveniente pois depende de cada veículo e da qualidade das suspensões pois caso a suspensão seja muito sensível poderão ser detetados falsos positivos. Madli et al. 2015 é um projeto mais elaborado, em que uma montagem semelhante é utilizada num veículo e o conceito é testado em estradas reais. Este sistema tem também a vantagem de detetar elevações no asfalto, através de uma diminuição da distância entre o sensor e o asfalto, ao contrário do que acontece num buraco em que a distância aumenta. O sistema é também composto por um módulo GPS e outro GSM para que seja possível determinar as coordenadas de uma detecção e enviá-las para um telemóvel, através de uma mensagem de texto, onde é possível consultar e armazenar todas as detecções. Foi também desenvolvida uma aplicação Android que determina a localização do utilizador e, caso a distância até uma detecção anterior seja inferior a 100m, um alerta é gerado para que o utilizador possa ajustar a sua velocidade em relação às condições do asfalto. Apesar dos bons resultados na detecção, o sistema tem algumas falhas ao nível da comunicação com o utilizador pois não é possível remover detecções no caso delas serem reparadas, sendo enviados avisos mesmo que uma estrada se encontre em boas condições. Também existe o mesmo problema que em Hegde et al. 2015 quando avaliada a instalação do sistema. Uma vez que este não se encontra no quadro do veículo, o sistema pode detetar falsos positivos devido a oscilações no terreno que não sejam causadas por buracos nem lombas mas apenas pelo asfalto defeituoso.



Figura 1.5: Sensor Ultrassom

### 1.1.4 Acelerómetro

Esta será a abordagem a ser utilizada e aquela em que os artigos analisados se mostram mais relevantes, devido à semelhança com o projeto que se pretende desenvolver. Em Mednis et al. 2011 é utilizado o acelerómetro do telefone e são apresentados vários algoritmos para a determinação do que é ou não um buraco na estrada. São também feitas algumas comparações sobre os resultados que diferentes telefones apresentam, dependendo do seu acelerómetro, tendo em conta a frequência de amostragem e o desvio padrão do eixo Z. Neste trabalho foi concluído que diferentes telemóveis recolhem diferentes informações no mesmo percurso, sendo crítico que o acelerómetro não esteja incluído no telemóvel para que exista consistência nos dados recolhidos por diferentes utilizadores. Foram também avaliados quatro métodos de análise de aceleração para determinar deteções de buracos, sendo que o método "*Z-DIFF*" foi o que apresentou melhores resultados. Graças a esta análise, foi desenvolvido código que pudesse fazer uma análise semelhante à proposta neste trabalho pois este método apresentou resultados positivos em 92% dos casos. No que diz respeito a Fouad et al. 2014, é um documento semelhante ao anterior mas adiciona um giroscópio para melhor deteção de buracos. Embora os resultados apresentados sejam bons, terá que ser tido em conta o processamento extra necessário para a utilização dos dados do giroscópio que não parece adicionar muito mais sensibilidade ao sistema. Neste trabalho, foi também desenvolvida uma aplicação em Android para que os dados recolhidos possam ser armazenados e consultados pelos utilizadores, apesar de não existirem comunicações com outros elementos, como uma base de dados para cruzamento de dados com outros utilizadores. Tal como mencionado nas conclusões, este trabalho não apresentou resultados promissores uma vez que teve uma taxa de sucesso de apenas 75% graças à utilização do acelerómetro do telemóvel. Em Chen et al. 2011 foi criado um elemento que contém GPS e acelerómetro e ainda um microprocessador para processar os dados adquiridos. É um sistema muito bem construído e que apresenta vários resultados em estradas de diferentes condições mas poderá ser mais caro do que o

pretendido desenvolver neste projeto. No artigo apresentado em Jang et al. 2015 é apresentada a comunicação entre vários veículos quanto à sua detecção de buracos. Esta ideia é bastante interessante uma vez que possibilita a comunicação em tempo real e caminha na direção das *Smart Cities* e da IoT (Internet of Things - Internet das Coisas em Português). O sistema é composto por um par de acelerómetros instalado no *tablier* do automóvel e uma antena GPS no seu exterior para a obtenção de coordenadas. O sistema apresenta resultados bastante precisos mas tem um elevado nível de processamento, pelo que foi necessário um processador mais potente para a gestão dos dados detetados, levando a que o preço total da montagem fosse mais elevado do que o desejado. Por fim, é ainda de salientar o trabalho em Kattan e Aboalmaaly 2014 que apresenta uma metodologia semelhante à que foi tomada neste trabalho no que diz respeito à comunicação com uma base de dados mas com pouco desenvolvimento e uma curta fase de testes, sendo que as detecções de buracos tiveram uma taxa de sucesso de apenas 61% devido à colocação do telemóvel no *tablier* do veículo e à baixa qualidade do algoritmo de detecção. Foi também utilizado o GPS do telemóvel que apresentou algumas falhas devido à sua baixa precisão e potência, sendo que em alguns casos não era possível obter a localização do utilizador. Ainda assim é mencionado que para trabalho futuro é proposto um melhoramento deste algoritmo bem como uma nova solução para a detecção de coordenadas.

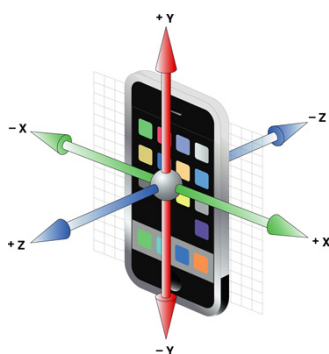


Figura 1.6: Direção dos eixos de um acelerómetro

## 1.2 Comparação de resultados

Comparando todos os artigos apresentados, é de notar que cada um tem os seus pontos fortes e fracos, como seria de esperar. Para o trabalho que será desenvolvido as qualidades mais importantes a ter em conta são o preço do material utilizado e também o tempo de processamento que o método consome. A tabela 1.1 mostra as características que serão tidas em conta bem como quais as metodologias que as apresentam.

Desta forma, a utilização de um acelerómetro é a mais indicada. Dentro desta metodologia, ainda é possível fazer uma comparação dos vários artigos analisados e tirar algumas conclusões, apresentadas na tabela 1.2.

Tabela 1.1: Comparação de resultados de artigos

	Qualidade de resultados	Preço de material	Tempo de processamento	Processador já incluído
Camera vídeo	✓	✗	✗	✗
Camera + iluminação	✓	✗	✗	✗
Ultrassom	✓	✓	✓	✗
Acelerómetro	✓	✓	✓	✓

Tabela 1.2: Comparação de resultados com utilização de acelerómetro

	Acelerómetro do telefone	GPS do telefone	Giroscópio do telefone	Processador do telefone	Custos além do telefone
Mednis et al. 2011	✓	✓	✗	✓	✗
Fouad et al. 2014	✓	✓	✓	✓	✗
Chen et al. 2011	✗	✗	✗	✗	✓

Embora os resultados dos trabalhos em que todos os componentes fazem parte do telefone sejam mais promissores em termos do preço dos materiais, a sua viabilidade é mais baixa, uma vez que um sistema que seja para o público em geral necessita de apresentar resultados consistentes, independentemente da situação e se o acelerómetro não estiver sempre no mesmo sítio (neste caso, o telefone) as leituras de cada irregularidade detetada são alteradas a cada passagem, dependendo do local em que o telefone se encontra, seja no bolso do casaco, no banco do veículo ou no seu *tablier*. A utilização de GPS é obrigatória para que os buracos detetados possam ser localizados, sendo que um GPS externo pode apresentar mais precisão do que o integrado no telemóvel e evita falhas de receção tal como é referido em Kattan e Aboalmaaly 2014. Desta forma, a solução apresentada neste projeto conta com um smartphone do utilizador, bem como um acelerómetro e GPS externos que estão fixados no veículo. O acelerómetro está colocado diretamente no quadro do veículo de testes de modo a que a qualidade das suspensões não interfira com as leituras. O GPS foi colocado no interior do veículo pois a sua antena é bastante potente, fazendo com que não seja necessário um local muito específico para a sua colocação.



## TRABALHO DESENVOLVIDO

A solução proposta para a resolução do problema apresentado em XXX é um sistema separado em três fases, cada uma delas ligada a um elemento físico: o Arduino, o telemóvel e a base de dados (alocada num computador). Assim surgem as seguintes secções, referentes às três fases do sistema.

### 2.1 Funcionamendo do Arduino

Tal como descrito na secção XXX do capítulo XXX, o código Arduino está separado em três grupos chave específicos, sendo descrita a sua utilização específica na presente secção:

- Na zona de declarações são incluídas as bibliotecas referentes ao acelerómetro, ao cartão de memória e ao GPS. São também definidos os modos de escrita no cartão de memória, os portos de ligação de comunicação do GPS e criadas variáveis globais referentes ao GPS e ao acelerómetro. Além disso, são criadas variáveis globais de controlo do código, variáveis de passagem de parâmetros entre vários ciclos da função *loop* e declaração de constantes que serão utilizadas diversas vezes ao longo do código, facilitando a alteração do seu valor em todo o código, caso tal se mostre necessário.
- Na função *setup* são declaradas as bandas de comunicação entre o Arduino e o telemóvel e entre o GPS e o Arduino. São também inicializados dois *leds* como elementos de saída de informação e um botão como entrada de informação e é atribuído o estado de “desligado” a ambos os *leds*. Por fim, são executadas as funções de inicialização do leitor do cartão de memória e do acelerómetro, *SD\_init* e *AccelInit* respectivamente, ambas explicadas mais á frente.

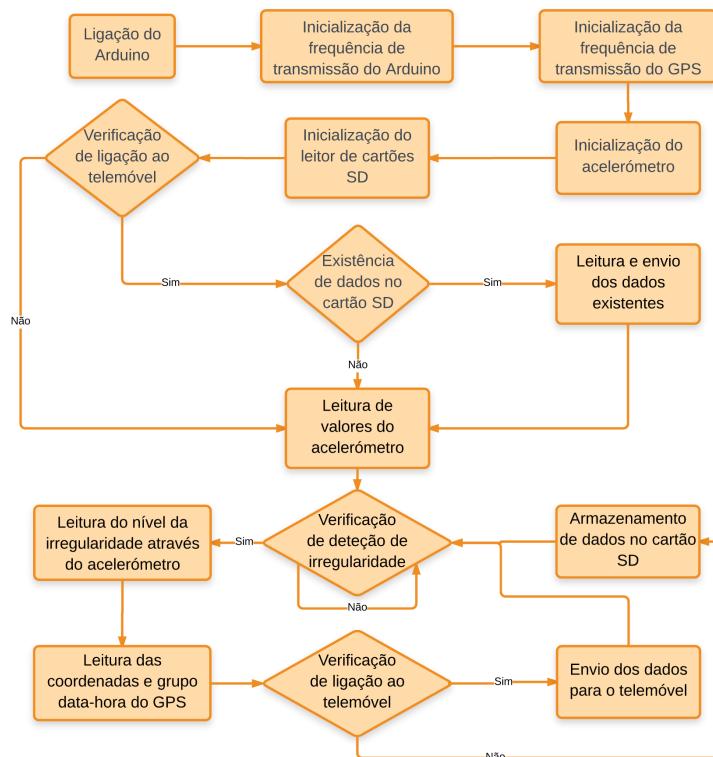


Figura 2.1: Fluxograma referente ao código Arduino

- A função *loop*, embora executada de uma forma constante, contém um código bastante simples mas importante. A função começa por verificar se existe informação disponível para leitura no módulo Bluetooth. Caso tal se verifique, essa informação é lida e comparada, para que seja possível determinar se a ligação ao módulo Bluetooth do Arduino sofreu alguma alteração, nomeadamente, quanto à ligação com a aplicação desenvolvida para esta dissertação. Na hipótese desta alteração se verificar, é avaliada se a ligação foi estabelecida ou cortada, sendo que, no caso de ser estabelecida, é executada a função *ReadSD* para leitura do cartão de memória e um *led* de controlo é aceso. Caso contrário, o mesmo *led* de controlo é desligado. De seguida são lidos os valores de aceleração dos eixos X, Y e Z, provenientes do acelerómetro, armazenados e comparados com a leitura anterior para verificação da existência de alguma irregularidade no asfalto. A maneira como esta comparação será explicada mais à frente. Caso seja detectada alguma irregularidade, um dos *leds* de controlo pisca e são armazenados, numa variável *S*, os valores do nível da irregularidade, bem como o grupo data-hora proveniente do GPS. Seguidamente é verificada a ligação à aplicação e caso esta exista, os valores de *S* são enviados directamente para o telemóvel. Se a ligação não existir, os valores de *S* são transferidos para o cartão de memória para que possam ser transmitidos para o telemóvel assim que exista uma ligação estável com o mesmo. Por fim, os valores actuais do acelerómetro são armazenados para que possam ser comparados na próxima execução da função *loop*.

## 2.2 Funções adicionais

### 2.2.1 SD\_init

Esta função serve para inicializar o leitor de cartões SD. Nela é chamada a função *begin* da biblioteca SD e verificada a resposta dessa mesma função. Caso a resposta seja o valor 10, um *led* de controlo pisca para que o utilizador saiba que a inicialização ocorreu com sucesso e é verificada a existência de um ficheiro com informação previamente armazenada sobre irregularidades detetadas anteriormente. Caso exista esse ficheiro, este é aberto de modo a que se possam juntar novos valores de irregularidades detetadas, caso contrário, o ficheiro é criado para futuros armazenamentos.

### 2.2.2 WriteSD

Para que o cartão de memória possa armazenar dados é apenas necessário abrir um ficheiro em modo escrita. A função *open* da biblioteca do SD está desenhada de forma a que, quando é pedido para abrir um ficheiro, este seja procurado no cartão de memória e caso não exista nenhum ficheiro com esse nome, então é criado um novo ficheiro vazio, com o nome inserido e posteriormente aberto. Para que possa ser mais fácil ao utilizador saber se a abertura e escrita do ficheiro foi feita com sucesso, um *led* pisca duas vezes. No final resta apenas fechar o ficheiro para que tudo fique guardado devidamente.

### 2.2.3 ReadSD

Semelhante à função de escrita, a função de leitura tenta abrir um ficheiro com o nome pretendido embora, neste caso, se o ficheiro desejado não existir, não é criado um novo pois se este não existe significa que não existem dados pendentes para envio. Quando um ficheiro é lido com sucesso, os seus valores são enviados pela porta série do Arduino para o telemóvel, o ficheiro é fechado e eliminado, de modo a que não sejam enviados dados duplicados. O código está protegido de modo a que os dados sejam apenas enviados para um telemóvel com a aplicação desenvolvida pois previamente foi recebida uma mensagem de controlo enviada pela própria aplicação, informando o estado da ligação, evitando que os dados sejam enviados para ligações desconhecidas.

### 2.2.4 AccelInit

Nesta função o acelerómetro é ligado e são determinados os valores de sensibilidade do mesmo. É também feita uma inicialização de valores para queda livre e batida que não são utilizados para esta dissertação. Esta funcionalidade foi mantida para possíveis aplicações futuras e uma deteção mais pormenorizada das irregularidades. Seguidamente, um *led* de controlo pisca para o utilizador saber que a inicialização foi bem sucedida.

### 2.2.5 Accel

Na função *Accel* é feito o processamento dos valores de aceleração dos três eixos do acelerómetro para determinar a classificação da irregularidade detetada. Embora o processo seja simples, é eficaz e semelhante ao método **Z-DIFF** apresentado em XXX, mas aplicado aos três eixos de aceleração. O método baseia-se em detetar a existência de desvios superiores a um limite previamente determinado, tendo sido considerados valores múltiplos de 100 nesta dissertação. Assim, é feita uma subtração de valores sucessivos de aceleração dos eixos e avaliado o valor absoluto dessa diferença. O valor atribuído à irregularidade é o mais baixo dos três desvios.

### 2.2.6 displayGPS

A função *displayGPS* é sem dúvida a mais elaborada nesta secção do trabalho devido aos dados enviados pelo GPS, muito extensos e, neste caso, não necessários, além de se encontrarem num formato não muito fácil de separar. A função é chamada dentro da função *loop* de modo a registar tanto o conjunto data-hora como as coordenadas em que a irregularidade foi detetada. Quando é chamada, faz uma leitura dos valores que o sensor GPS recebe dos satélites e percorre-os até encontrar o texto "GPRMC", a partir do qual vem toda a informação necessária e armazena todos esses valores num vetor de strings (sequências de caracteres). O passo seguinte consiste em percorrer este e retirar apenas os segmentos relevantes, pela ordem desejada, neste caso, começando pela latitude e longitude, seguindo-se a data e por fim a hora. Quando estes quatro parâmetros são armazenados, a função muda o valor de uma variável de controlo e o código prossegue.

### 2.2.7 FTOA - Float to ASCII

Dada a forma em que os valores das coordenadas do GPS são recebidos, é necessária fazer uma conversão para um formato mais simples de transmitir e como o processo de conversão é executado diversas vezes, foi criada uma função que evita repetição do código. Os dados enviados pelo GPS, relativos às coordenadas vêm no formato DDS – Degree Decimal Minutes (Graus e Minutos Decimais em Português) e apresentam o formato (1) da figura ?? . Tal como se pode verificar, existem seis parâmetros neste formato, nomeadamente graus, minutos e hemisfério, três para a latitude e outros três para a longitude. Um formato mais simples é o DD – Decimal Degree (Graus Decimais em Português), também apresentado na figura ?? , no ponto (2). Neste formato existem apenas quatro parâmetros (graus e hemisfério) que podem ser reduzidos a dois, se forem considerados graus negativos, representando o hemisfério a que a latitude ou longitude se referem, sendo assim necessário enviar menos dados para representar a mesma quantidade de informação, tornando o processo mais rápido. A conversão de DDS para DD é simples, sendo apenas necessário dividir a parte dos minutos decimais por 60 (convertendo minutos para graus) e somar esse valor aos graus já existentes. Como os valores que são recebidos do GPS

vêm em strings, é necessário convertê-los para formato numérico, através da função `atoi` (ASCII to Integer, ASCII para inteiro em português) que já existe nas bibliotecas do Arduino e posteriormente voltar a converter para formato de string. Apesar de ser possível escrever um valor numérico numa string, quando esta operação é feita, apenas duas casas decimais são utilizadas e para que as coordenadas possam apresentar um elevado grau de precisão são necessárias quatro casas decimais, sendo assim necessária a criação desta nova função de modo a reter tantas casas decimais quanto as desejadas.

**DDS - DD° MM.MMMM (1)**

**DD - DD.DDD° (2)**

Figura 2.2: Formatos de apresentação de coordenadas

## 2.3 Funcionamento do Android

Tal como explicado na secção XXX do capítulo XXX, o método escolhido para a programação Android foi a MIT App Inventor 2, no website <http://ai2.appinventor.mit.edu> devido à facilidade apresentada, graças a ser uma programação por blocos bem como ao facto de correr num web browser e fazendo o processamento e compilação de código no servidor em vez de ser no cliente, diminuindo bastante o processamento feito no computador local, tornando o processo de criação mais rápido.

A base do código da aplicação é um ciclo `while` que corre desde que a aplicação esteja aberta, semelhante à função `loop` do Arduino. Dentro deste ciclo existem duas condições: uma que verifica se o GPS do telemóvel está a detetar coordenadas e outra que verifica a ligação ao módulo *Bluetooth* do Arduino, bem como um procedimento que verifica a existência de dados armazenados no telemóvel para futuro envio para a base de dados. O GPS do telemóvel é utilizado para que seja possível reportar irregularidades existentes no asfalto sem que se esteja a conduzir, possibilitando todos os transeuntes adicionar informações sobre o local em que se encontram. Sempre que são detetadas coordenadas disponíveis, um botão com o texto "*Shock*" fica ativo e sempre que é pressionado são armazenadas as coordenadas atuais, bem como a data e hora presentes para um posterior envio para a base de dados. Para que estas ocorrências possam ser diferenciadas das detetadas pelo acelerómetro montado no Arduino, o código que lhe é atribuído tem um valor único, facilitando assim a diferenciação entre os dois tipos de irregularidades. No que toca à ligação a um módulo *Bluetooth*, sempre que esta existe, é enviado um código para o dispositivo a que foi feita a ligação. Este código serve para o Arduino saber que o dispositivo que se encontra ligado tem a aplicação a correr, evitando assim que os dados recolhidos por este sejam enviados para um dispositivo desconhecido. Depois de ser feita a ligação, o código fica a "escutar" o Arduino até que ele envie alguma informação para

o telemóvel e quando tal acontece, os vários conjuntos de dados recebidos são divididos para uma lista temporária. Após a divisão ser feita, a lista é percorrida e os dados são enviados para o cartão de memória do telemóvel ou para a base de dados, dependendo da existência de ligação Wi-Fi. Esta ligação é verificada a partir de um erro produzido automaticamente pelo sistema Android, sendo emitida uma mensagem de erro com um código específico. O envio de dados para a base de dados é feito através do método *POST*, sendo desta forma possível ocultar os dados enviados, sendo esta uma vantagem no caso de no futuro serem desenvolvidas contas de utilizador em que certas informações devem permanecer confidenciais. Se a ligação à *internet* não existir, os dados são armazenados no telemóvel para um futuro envio, sendo necessário pressionar um botão existente na aplicação. Este botão serve também para informar o utilizador que tem dados armazenados no dispositivo, uma vez que o botão nem sempre se encontra ativo. No que toca à seleção de dispositivos *Bluetooth*, é mostrada uma lista dos dispositivos com que o telemóvel está emparelhado. No caso de ser a primeira vez que um determinado telemóvel se liga ao Arduino, é necessário fazer o emparelhamento fora da aplicação, como para qualquer outro dispositivo. Quando a aplicação é terminada, é enviada uma mensagem para o Arduino de modo a que este saiba que não é possível enviar mais informação sobre irregularidades e armazene todas as deteções nesse cartão de memória.

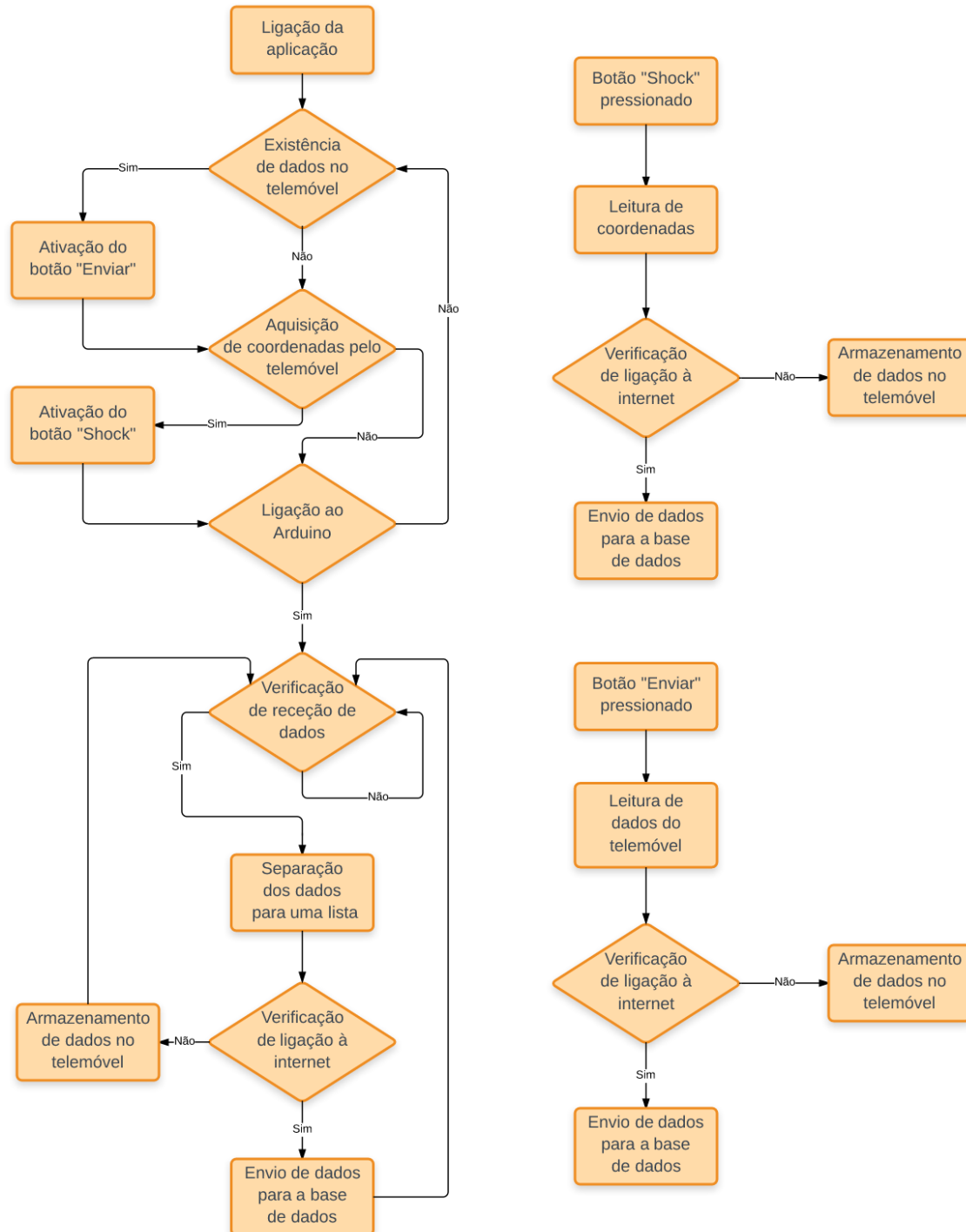


Figura 2.3: Fluxograma referente ao código Android

## 2.4 Base de dados

De modo a que seja possível consultar as irregularidades, foi necessário desenvolver uma base de dados onde ficassem armazenadas as coordenadas, data, hora e intensidade da irregularidade detectadas. Esta base de dados foi desenvolvida em MySQL e contém cinco campos, sendo eles o ID (campo de identificação de cada entrada), a intensidade da irregularidade, a latitude e a longitude da irregularidade e também o conjunto data-hora, utilizando apenas uma variável. Esta base de dados pode ainda ser consultada e alterada por um administrador da mesma, caso este ache necessário de modo a inserir novas funcionalidades no projeto.

## 2.5 WebSite

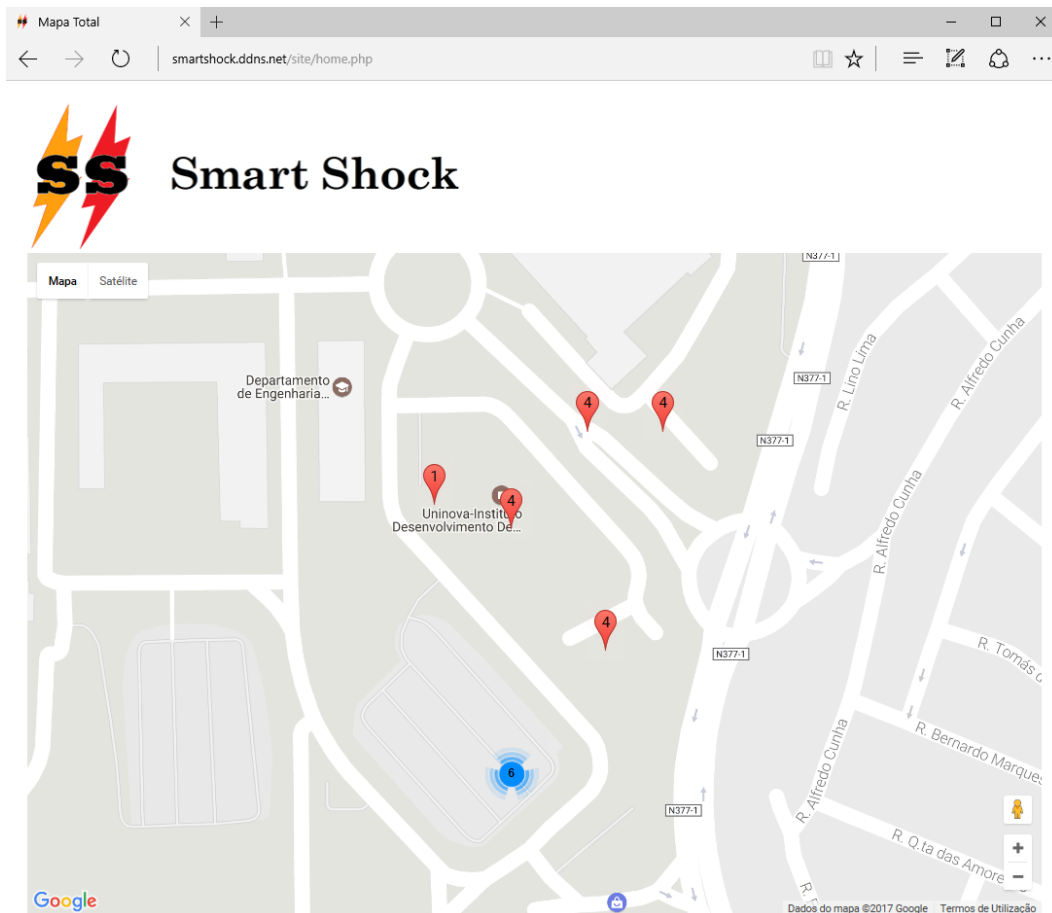
### 2.5.1 home.php

Esta é a página principal da parte *web* desta dissertação. Nela é possível ver-se o logótipo e o nome utilizados neste projeto e também um mapa que contém todas as irregularidades detetadas ou inseridas manualmente, referindo a intensidade das mesmas. Para que essas irregularidades possam ser mostradas, é percorrida toda a base de dados através de uma *query* (questão, em português) de MySQL em que são devolvidos os valores de latitude, longitude e intensidade. Depois, utilizando uma pequena interface desenvolvida pela Google, todos os pontos são marcados no mapa, sendo que no seu sinalizador está indicada a intensidade da irregularidade, como se pode ver na figura ?? . No caso de existirem várias irregularidades muito próximas fisicamente, a interface agrupa-as automaticamente, informando quantas se encontram nessa zona. A cor destes agrupamentos é alterada consoante o número de itens aglomerados, tornando assim mais fácil a deteção de locais com elevado número de ocorrências.

### 2.5.2 store.php

A página store não contém nenhuma componente gráfica para consulta de informação pois destina-se apenas ao armazenamento de dados na base de dados. Quando o telemóvel envia dados, é este o destino e como tal, tem que existir algum processamento dos dados recebidos. A informação recebida tem o aspecto da figura ?? e vem em quatro strings: força, latitude, longitude e data-hora, sendo portanto necessário fazer uma conversão para o formato correcto e consequente armazenamento em variáveis locais. De seguida é feita uma query MySQL para que seja possível fazer o armazenamento na base de dados, sendo emitida uma mensagem de controlo para o telemóvel de modo a dar informação sobre o sucesso ou falha deste armazenamento.



Figura 2.4: Página *web* home.php

### 2.5.3 listLocations.php

Esta página *web* é bastante semelhante à *home.php*, sendo que aqui apenas é mostrada a localização da irregularidade selecionada na página *listlocations.php* de modo a ser mais fácil descobrir a localização da irregularidade quando esta se encontra agrupada com outras irregularidades existentes nas proximidades, como acontece na figura ??.

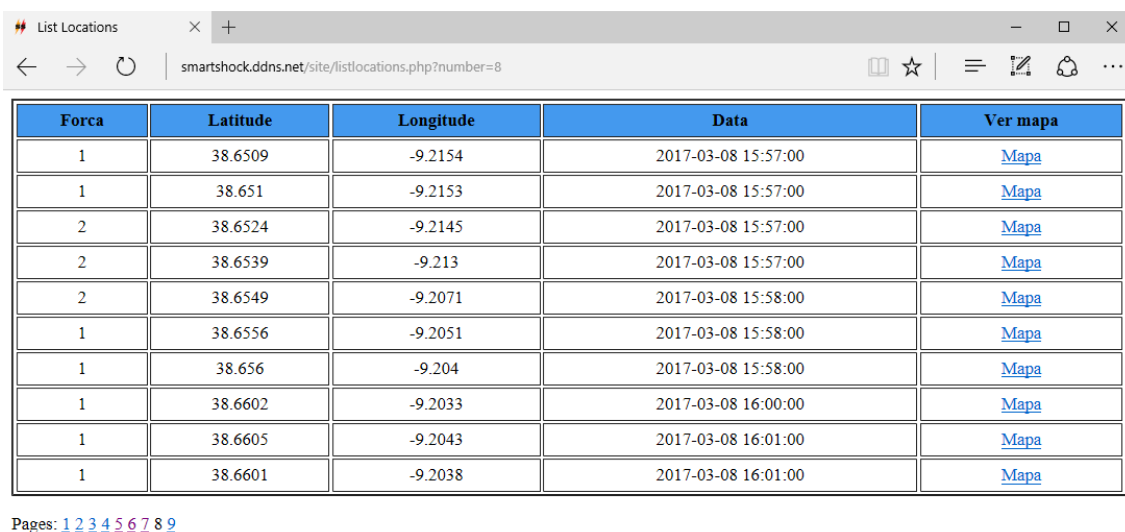
3 +38.6601 -09.2047 2017-02-15 14:37

Figura 2.5: informação tipo adicionada na base de dados

### 2.5.4 showLocation.php

Esta página tem a finalidade de consultar todas as irregularidades num formato numérico, ao invés da página Home.php, em que as irregularidades são apresentadas num mapa.

Tal como apresentado na figura ??, nesta página é possível consultar a toda a informação referente a uma irregularidade, incluindo a data e hora em que foi detetada, ao invés da página Home.php. Para que seja necessário visualizar a lista de irregularidades, é necessário colocar no endereço o número da página que se deseja visualizar, sendo o valor pré-definido “1”. Cada uma destas páginas mostra dez entradas da lista de irregularidades e no fundo da página é mostrado um navegador para as diferentes páginas. Alinhado com cada irregularidade, existe uma hiperligação para a página showLocation.php em que é mostrado no mapa a localização da irregularidade seleccionada para uma melhor compreensão dos valores de latitude e longitude.



Força	Latitude	Longitude	Data	Ver mapa
1	38.6509	-9.2154	2017-03-08 15:57:00	<a href="#">Mapa</a>
1	38.651	-9.2153	2017-03-08 15:57:00	<a href="#">Mapa</a>
2	38.6524	-9.2145	2017-03-08 15:57:00	<a href="#">Mapa</a>
2	38.6539	-9.213	2017-03-08 15:57:00	<a href="#">Mapa</a>
2	38.6549	-9.2071	2017-03-08 15:58:00	<a href="#">Mapa</a>
1	38.6556	-9.2051	2017-03-08 15:58:00	<a href="#">Mapa</a>
1	38.656	-9.204	2017-03-08 15:58:00	<a href="#">Mapa</a>
1	38.6602	-9.2033	2017-03-08 16:00:00	<a href="#">Mapa</a>
1	38.6605	-9.2043	2017-03-08 16:01:00	<a href="#">Mapa</a>
1	38.6601	-9.2038	2017-03-08 16:01:00	<a href="#">Mapa</a>

Pages: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Figura 2.6: Página *web* listlocations.php

## BIBLIOGRAFIA

- Chan, C. K., Y. Gao, Z. Zhang e N. Dahnoun (2014). "Implementation and evaluation of a pothole detection system on TI C6678 digital signal processor". Em: *EDERC 2014 - Proceedings of the 6th European Embedded Design in Education and Research Conference*.
- Chen, K., M. Lu, X. Fan, M. Wei e J. Wu (2011). "Road condition monitoring using on-board three-axis accelerometer and GPS sensor". Em: *Proceedings of the 2011 6th International ICST Conference on Communications and Networking in China, CHINACOM 2011*.
- Fouad, M. M., M. A. Mahmood, H. Mahmoud, A. Mohamed e A. E. Hassanien (2014). "Intelligent road surface quality evaluation using rough mereology". Em: *2014 14th International Conference on Hybrid Intelligent Systems*.
- He, Y., J. Wang, H. Qiu, W. Zhang e J. Xie (2011). "A research of pavement potholes detection based on three-dimensional projection transformation". Em: *Proceedings - 4th International Congress on Image and Signal Processing, CISP 2011*.
- Hegde, S., H. Mekali e G. Varaprasad (2015). "Pothole detection and inter vehicular communication". Em: *2014 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2014*.
- Jang, J., A. W. Smyth, Y. Yang e D. Cavalcanti (2015). "Road surface condition monitoring via multiple sensor-equipped vehicles". Em: *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*.
- Kattan, A. e M. F. Aboalmaaly (2014). "A smartphone-cloud application as an aid for street safety inventory". Em: *Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014*.
- Madli, R., S. Hebbar, P. Pattar e V. Golla (2015). "Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers". Em: *IEEE Sensors Journal*.
- Mednis, A., G. Strazdins, R. Zviedris, G. Kanonirs e L. Selavo (2011). "Real time pothole detection using Android smartphones with accelerometers". Em: *2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS'11*.
- Moazzam, I., K. Kamal, S. Mathavan, S. Usman e M. Rahman (2013). "Metrology and visualization of potholes using the microsoft kinect sensor". Em: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*.
- Yu, X. e E. Salari (2011). "Pavement pothole detection and severity measurement using laser imaging". Em: *IEEE International Conference on Electro Information Technology*.

## BIBLIOGRAFIA

---

Zhang, Z., X. Ai, C. K. Chan e N. Dahnoun (2014). “An efficient algorithm for pothole detection using stereo vision”. Em: *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*.