

Sistema para Coleta e Disponibilização de Dados de Potenciais Buracos em Rodovias

Jefferson Kamigashima, Leonardo Rauta

jeffersonkamigashima@gmail.com, leonardo.rauta@ifsc.edu.br

Instituto Federal de Santa Catarina (IFSC) – Câmpus Gaspar. Brasil.

DOI: [10.17013/risti.30.78-90](https://doi.org/10.17013/risti.30.78-90)

Resumo: A presença de buracos em rodovias é um problema existente nas estradas brasileiras e é agravado pela ausência de manutenção contínua e eficaz. Esse trabalho propõe um sistema capaz de realizar a coleta de dados de possíveis buracos e disponibilizá-los em uma aplicação web, utilizando um módulo embarcado em automóvel e outro hospedado na nuvem. Esse sistema é composto por placas, sensores e aplicações que coletam dados de possíveis buracos, armazenam os dados localmente e posteriormente enviam esses dados até um web service RESTful localizado em nuvem. Para a validação do sistema foram realizados testes simulando alguns cenários possíveis pelo qual o sistema pode ser submetido. Os resultados demonstraram a viabilidade de empregar o sistema no desenvolvimento de planos de manutenção de rodovias e na melhoria da qualidade das nossas estradas e do conforto para os seus usuários.

Palavras-chave: Sistemas embarcados; Buracos; Identificação; Web Service; RESTful.

System to collect and make available data from potential holes in highways

Abstract: Presence of holes in Brazilian highways is an existing problem and it is aggravated by the absence of continuous and inefficient maintenance. This work propose a system aimed to collect data from potential holes and making them available in a web application, using a embedded module in a car and another module hosted in cloud. This system makes use of boards, sensors and applications to collect data from potential holes, store data locally and then send it to a web service RESTful hosted in cloud. To validate the system tests were performed simulating some possible scenarios which the system can be submitted. The results demonstrated the feasibility of using the system in the road maintenance plans and in the improvement of the quality of our roads, making them comfortable for its users.

Keywords: Embedded Systems; Roles; Recognition; Web Service; RESTful.

1. Introdução

A malha rodoviária brasileira é responsável pelo escoamento da produção industrial e agrícola do país. Através dela é possível o tráfego de bens e pessoas pelo território

nacional e a sua qualidade reflete diretamente no custo e na segurança do transporte, tornando necessária a manutenção periódica das vias para que elas ofereçam segurança e conforto para seus usuários.

Dados do Sindicato Nacional Da Indústria De Componentes Para Veículos Automotores e da Associação Brasileira Da Indústria De Autopeças (2016) apresentam que, em 2016, no Brasil, a frota circulante de automóveis, caminhões e ônibus contava com aproximadamente 42,9 milhões de unidades, revelando um pequeno aumento de 0,7 % em relação ao ano anterior.

De acordo com a Confederação Nacional de Transporte (2016) nesse mesmo período houve, na malha rodoviária brasileira, um aumento de 26,6% no número de trechos com buracos grandes, quedas de barreiras, pontes caídas e erosões, passando de 327 para 414 o número de pontos críticos. Desses, 304 são trechos que possuem buracos com dimensões maiores que o de um pneu de um automóvel de passeio e são ocasionados geralmente pela sobrecarga dos veículos que trafegam e/ou pela espessura inadequada ou insuficiente empregadas na construção do pavimento.

No estado de Santa Catarina 59,3% (1.886 km) das rodovias apresentam algum tipo de deficiência. Dessa deficiência nas estradas estaduais, 43,3% apresentaram pavimento em condições regular, ruim ou péssimo, deficiência essa que eleva o custo da manutenção dos veículos e o consumo de combustível, além de reduzir a segurança no tráfego diário de automotores. Segundo a CNT (Confederação..., 2016), essas deficiências apresentadas no pavimento das rodovias fazem com que o custo operacional do transporte no Estado sofra um acréscimo estimado de 23,5%.

A manutenção das vias é de responsabilidade do governo ou das concessionárias proprietárias (Confederação..., 2016), sendo que as estatais são distribuídas nas esferas em que se encontram (municipal, estadual ou federal).

Nas vias estatais é necessário um processo para a autorização do reparo e/ou manutenção, visando a obtenção de recursos para realização do serviço. Para justificar a utilização desses recursos para manutenção, é importante que existam dados que comprovem a necessidade real do serviço com informações como localização e o tipo de dano encontrado.

Moreira et al. (2013) afirmam que a presença de buracos nas rodovias brasileiras e a influência que eles exercem na ocorrência de acidentes, fato ocasionado por uma manutenção deficiente e sem um critério de escolha dos locais a serem reparados.

Trindade (2015) defendem a importância do monitoramento de rodovias como forma de melhorar o planejamento da manutenção e adequação das vias, coletando dados que acusaram pontos que demandam manutenção.

No ano de 2015, uma análise realizada pela Confederação Nacional de Transporte (2016) apontou que 36,52% do montante de recursos autorizados para infraestrutura de transporte rodoviário não foram utilizados. Esses valores não utilizados poderiam ser convertidos em melhorias na malha viária que reduziriam o número de trechos deficientes e, consequentemente, aumentaria a segurança e satisfação de seus usuários.

Assim, torna-se essencial o uso de um sistema que apresente um mapeamento dos trechos com buracos existentes em rodovias e que ao mesmo tempo permite armazenar a sua localização. Isso possibilitaria aos usuários escolher qual rota utilizar para desviar de trechos esburacados e também seria um recurso útil para que os responsáveis pela manutenção pudesse justificar os orçamentos de manutenção da infraestrutura de transporte rodoviário ou para cobrar as empresas responsáveis, quando estas estiverem sob concessão de uma empresa de iniciativa privada.

Além disso, o sistema possibilitaria também a implantação de planos de monitoramento planejado das vias, favorecendo a qualidade da estrutura viária e um melhor aproveitamento dos recursos liberados para sua manutenção.

Esses dados mostram a colaboração que esse sistema oportunizaria aos usuários dessas estradas e também para a economia local, justificando a relevância de sua implementação

Com conhecimento desse problema e da importância de uma manutenção mais eficiente e melhor planejada de nossas rodovias, esse projeto visou a elaboração de um sistema que detecte a presença de buracos em rodovias, utilizando-se dos conceitos das Leis da Mecânica de Newton para identificar os possíveis buracos encontrados ao longo da estrada e realizando o seu mapeamento através do uso de GPS (Global Positioning System). Toda a informação coletada pelo sistema fica disponível em uma página Web, permitindo assim, que os dados sejam consultados por quem tiver interesse.

2. Trabalhos correlatos

Pensando no desenvolvimento do trabalho, foi necessário efetuar uma busca em bases de dados para analisar os trabalhos correlatos. Assim, foi realizada busca no portal de periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), no Google Acadêmico, e na base de dados da IEEE (IEEEExplore) onde foram encontrados apenas os trabalhos de Borges et al. (2011), Moreira et al. (2013) e Rauta et al. (2017).

Borges et al. (2011) apresentam a proposta de um sistema para identificar a localização geográfica dos buracos em estradas. No seu experimento foi utilizado um sistema com um microcontrolador, um módulo Bluetooth, um aparelho celular com GPS e um Wii remote adaptados a um carro de controle remoto. Com o projeto, Borges et al. (2011) concluíram que o sistema se mostrou viável para a detecção de buracos em estradas.

Moreira et al. (2013) desenvolveram um sistema semelhante. O objetivo do projeto era detectar buracos em estradas ou rodovias utilizando um acelerômetro. Para tanto foram utilizados um microcontrolador, um acelerômetro e um computador com um software para interpretação dos dados processados. No experimento foi utilizado uma bicicleta para realização dos testes. O resultado obtido pelos autores foi um sistema capaz de identificar todos os buracos pelo qual o protótipo passou, com a ressalva de que ao sair de um buraco o sistema acabava identificando outro buraco.

Já Rauta et al. (2017) se basearam nos dois outros trabalhos e apresentam uma proposta de um sistema para identificação de buracos através das leis da mecânica de Newton. Porém, trata-se apenas de uma proposta e não houve desenvolvimento no projeto.

Nos dois primeiros projetos apresentados a ideia de utilizar um sistema embarcado para identificar buracos obteve resultados satisfatórios, porém com suas aplicações em protótipos que não refletem na totalidade o cenário real. Esse fator acaba influenciando os resultados, pois a disposição dos eixos, o peso e as dimensões maiores que um carro possui modificaria a lógica necessária para a identificação de um buraco.

No trabalho de Borges et al. (2011) o protótipo utilizado possuía dimensões muito distantes de um carro em tamanho real, e a dimensão dos buracos pelos quais ele passou também eram igualmente menores. Esse detalhe foi explorado no trabalho de Moreira et al. (2013) onde o protótipo pôde passar por buracos no tamanho real pelo qual um carro passaria. Entretanto, o protótipo ainda estava longe da estrutura de um carro, uma vez que o carro possui quatro eixos e um peso substancialmente maior.

O intuito desse projeto foi desenvolver um sistema semelhante aos citados anteriormente em um carro, simulando o ambiente real pelo qual o sistema iria passar. Assim, esse projeto foi baseado na proposta de Rauta et al. (2017), o qual forneceu uma base teórica e uma metodologia a ser seguida no desenvolvimento do projeto.

A fim de comparar os trabalhos correlatos com o trabalho desenvolvido por esse projeto, foram elencados alguns critério de comparação entre os trabalhos. Os critérios utilizados foram: (i) se a localização era realizada por GPS ou por algum outro mecanismo; (ii) se o sistema era capaz de identificar buracos em tamanho real ou apenas grandes buracos; (iii) se foi desenvolvido um protótipo em escala real dentro de um carro, ou se foi montado um protótipo em uma outra escala; e (iv) se os dados lidos/obtidos pelo sistema seriam disponibilizados de alguma forma. Com base nesses critérios foi elaborada a Tabela 1, a qual apresenta o comparativo entre as principais diferenças entre os projetos citados em relação a este projeto.

Trabalho	Localização via GPS?	Buracos em tamanho real?	Protótipo em escala real?	Disponibilização dos dados?
Borges et al.	Sim	Não	Não	Não
Moreira et al.	Sim	Não	Não	Não
Rauta et al.	Sim	Sim	Não	Sim
Este trabalho	Sim	Sim	Sim	Sim

Tabela 1 – Comparativo entre os trabalhos correlatos

3. Solução desenvolvida

A solução proposta foi baseada na proposta de Rauta et al. (2017) com foco na coleta de informação dos dados de possíveis buracos encontrados em rodovias e a disponibilização desses dados na forma de um mapeamento dos buracos identificados. Para isso, o sistema desenvolvido possui um módulo embarcado que coleta informações de ocorrências de possíveis buracos e disponibiliza essa informação através de uma aplicação web instalada na nuvem. Assim, o sistema faz uso dos seguintes equipamentos de hardware: 1 Placa Arduino UNO; 1 Sensor acelerômetro MPU6050; 1 Raspberry PI3; 1 Sensor GPS GY-NEO6Mv2 + antena cerâmica; 1 servidor em cloud.

Em relação ao software, para obtermos as funcionalidades necessárias para o funcionamento do sistema, foram desenvolvidos os seguintes módulos: Módulo Servidor Web (MSW); Módulo Cliente (MC); Módulo Servidor Local (MSL), e Módulo Captação de Dados (MCD). Para o funcionamento do sistema completo, os módulos interagem entre si até os dados chegarem a um usuário conectado ao Módulo Cliente.

A Figura 1 apresenta o diagrama de distribuição do sistema, exemplificando as interações entre os módulos até as informações chegarem ao cliente. Esse projeto teve foco no desenvolvimento dos módulos destacados em amarelo. O Módulo Captação de Dados, em branco, não foi desenvolvido em sua totalidade pois não era o objetivo principal desse projeto.

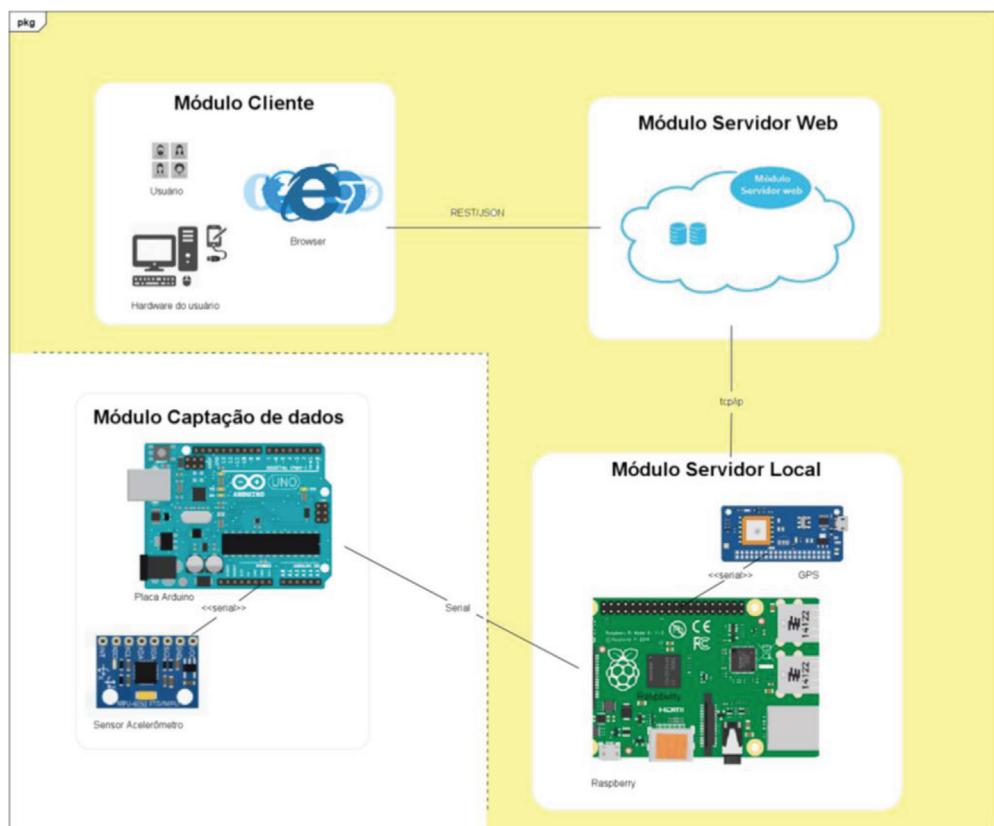


Figura 1 – Diagrama de distribuição

O Módulo Captação de Dados (MCD) faz uso de uma placa Arduino UNO instalada em cada um dos eixos do carro, próximo ao amortecedor. Nessa placa foi acoplado o sensor acelerômetro que faz a captação dos valores das forças gravitacionais, que serão recebidos pela aplicação instalada no Arduino e enviados ao Módulo Servidor Local (MSL). Nesse módulo não foi implementado o algoritmo completo para avaliação das

forças gravitacionais, apenas uma aplicação para teste e validação do sistema. Além disso, esse módulo deveria ser instalado nos eixos do carro, o que inviabilizou seu desenvolvimento completo.

O Módulo Servidor Local (MSL) faz a integração com o MC para receber os dados dos sensores. O MSL possui integração com o MC através do cabo USB utilizado para comunicação e para alimentação da placa Arduino. Isso possibilita o recebimento da leitura das variações identificadas pelo acelerômetro do MC.

Além de integrar com o Arduino, no módulo MSL são armazenados os dados dos eventos capturados pelos sensores e, no momento em que o usuário solicitar o envio, existindo a disponibilidade de uma conexão com a internet, ele deve integrar com o Módulo Servidor Web (MSW), permitindo o envio dos dados recolhidos pela aplicação instalada no Raspberry PI para armazenamento no banco de dados do MSW.

O Módulo Servidor Web (MSW) precisa estar disponível para o MSL enviar os dados e para o Módulo Cliente (MC) consumir esses dados. Para armazenar esses dados, foi utilizado um banco de dados não relacional MongoDB, que possui alta escalabilidade e facilidade na integração com outras tecnologias, hospedado na nuvem através de um serviço de gerenciamento de banco de dados online, o mLab. A inserção nesse banco de dados ocorre via socket, através de uma aplicação implementada utilizando TCP/IP, que recebe requisições da aplicação cliente existente no MSL.

Para disponibilizar esses dados para o MC, foi implementado um Web Service RESTful utilizando o framework Jersey e o servidor Apache Tomcat, que permite consultas ao banco de dados com a possibilidade de efetuar buscas por data e/ou carro.

Por fim, para disponibilizar e apresentar os dados para o usuário, foi implementado uma página web que permite buscar todos os buracos encontrados ou realizar uma busca filtrada por período, por nome do carro ou por carro em um determinado período. Esses dados são consumidos do Web Service e apresentados em um mapa utilizando a API do Google Maps e também em uma tabela. No mapa, cada buraco é representado por um marcador que, ao clicar sobre ele, apresenta um quadro com as informações individuais sobre cada buraco.

4. Validação

Para a validação do sistema foi realizada uma pesquisa experimental com os módulos desenvolvidos na qual foram realizados testes para validar a funcionalidade do MSL, MSW e MC. Foram realizados 9 testes, (I) teste de funcionamento no carro por 15 minutos; (II) teste de funcionamento no carro com interrupção; (III) teste de funcionamento no carro configurado para encontrar buracos em um intervalo menor de tempo; (IV) teste de funcionamento no carro configurado para encontrar efetivamente um buraco; (V) teste de envio dos dados; (VI) teste de inserção no banco de dados; (VII) teste de persistência e armazenamento de dados; (VIII) teste dos filtros de consulta e da apresentação dos dados; e (IX) teste do sistema funcionando por completo.

No Módulo Servidor Local (MSL), foram realizados testes para verificar a captura das leituras do acelerômetro e do GPS e o envio desses dados para o Módulo Servidor Web

(MSW). Para testar a captura das leituras dos sensores, foi instalado um programa configurado para que, a cada 15 segundos, simulasse que o sistema encontrou um buraco. Guardando assim as informações da posição do carro através do GPS e também a leitura atual, as 10 leituras anteriores e as 10 leituras posteriores à identificação do buraco, durante um período de 15 minutos. Nos testes realizados no automóvel, o sistema foi ligado à alimentação 12V.

Devido à alimentação das tomadas do carro serem energizadas no acionamento do motor, cada vez que o carro era desligado, essa alimentação era cortada automaticamente, interrompendo assim o funcionamento da aplicação. A Figura 2.a apresenta a disposição do sistema no interior do carro. Essa ligação e disposição permitiu que o sistema funcionasse por um tempo menor para testar a persistência de dados em condições adversas de funcionamento como, por exemplo, quando o motor do carro desliga sem a intenção do condutor. Já a Figura 2.b apresenta a ligação do sistema à alimentação 12V do carro. Nos primeiros testes, após configurada a aplicação no Raspberry PI, o carro percorreu dois trajetos diferentes: no teste I pelo tempo necessário para o sistema rodar o programa por completo (15 minutos), e no teste II o sistema funcionando por aproximadamente 7 minutos até ser interrompido pelo desligamento do motor do carro – corte na alimentação. Depois, no retorno desse trajeto, o sistema foi religado e pode funcionar pelos 15 minutos necessários para a aplicação rodar por completo, totalizando 22 minutos de captura de buracos no segundo teste.

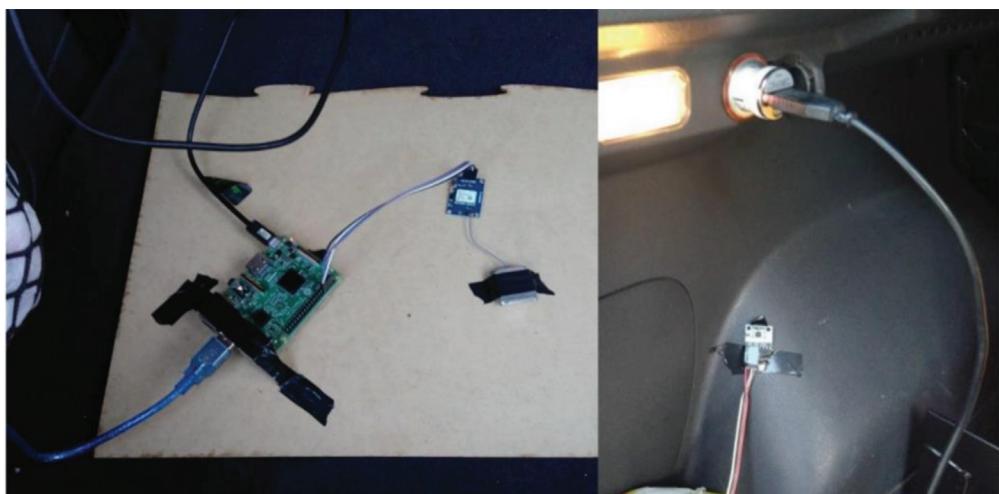


Figura 2 – Sistema instalado no porta-malas do automóvel. a) montagem do sistema contendo a placa Raspberry ligada ao módulo GPS; b) alimentação do sistema via 12V

O teste III foi realizado alterando a configuração do sistema para simular o encontro de um buraco a cada 7 segundos durante o período de 7,5 minutos. Após percorrer o trajeto por aproximadamente 3 minutos, o motor do carro foi desligado e logo em seguida religado, permitindo que a aplicação voltasse a funcionar, agora pelos 7,5 minutos necessários. O sistema foi submetido a esse teste para verificar a disponibilidade da

aplicação após uma interrupção repentina da alimentação, e para verificar a capacidade de buracos/segundo que ele poderia capturar.

Após finalizar os testes descritos acima, a configuração foi alterada para encontrar efetivamente um buraco, sendo o teste IV, realizado. Como o algoritmo para identificar um buraco não foi desenvolvido nesse projeto, foi convencionado que o sistema consideraria como buraco os eventos dos sensores onde a leitura “AngleX” do acelerômetro ultrapassasse o valor de 16.500. Valor escolhido arbitrariamente para que a aplicação não ficasse sensível a ponto de encontrar eventos a todo instante, nem rigoroso demais impedindo que fosse encontrado algum evento. O sistema então foi instalado no porta-malas do carro, onde o acelerômetro foi fixado sobre a caixa de ar do eixo traseiro esquerdo, com o restante do sistema fixado sobre uma prancha de madeira, alimentado pela saída 12V do automóvel (Figura 2.b).

Nesse quarto teste, o sistema foi programado para rodar durante um período de 10 minutos, no qual, durante o trajeto percorrido aleatoriamente, o carro fez várias paradas e, em cada uma, o carro foi desligado, fazendo com que o sistema não funcionasse pelo tempo total de 10 minutos, simulando a condição de uso na rotina diária de uma pessoa que precisa realizar vários pequenos trajetos, como fazer compras, abastecer e levar o filho para escola.

Com os dados dos buracos armazenados no servidor local, foi possível realizar o teste V: o envio de dados para o servidor web. Em cada vez que foi verificado o armazenamento dos dados coletados no Raspberry, a aplicação de envio era iniciada para integrar os dados coletados com o servidor web.

O teste VI foi realizado para confirmar se os dados foram realmente inseridos, possibilitando a sua consulta. Para isso, foi acessado o gerenciador do banco de dados e, após, foi acessado a página clicando para buscar todos os buracos.

Já o teste VII, de persistência e de armazenamento de dados, também foi realizado no servidor web: foram incluídos na pasta em que são armazenados os buracos no Raspberry, todos os arquivos JSON de buracos encontrados (244 no total) e então iniciou-se a aplicação de envio.

Após a inserção no banco de dados, o teste VIII foi realizado acessando a página web e efetuando consulta a todos os buracos (sem filtro), e depois foram realizadas consultas com o filtro de nome do carro, filtro por período e por último consulta com o filtro de nome do carro dentro de um período.

Como teste final IX, foi implementado um programa com as configurações do teste que encontrou efetivamente um buraco. Esse teste não foi realizado dentro do veículo, foi realizado próximo ao computador por um período de 75 segundos. O objetivo era simular o sistema funcionando integralmente, desde a identificação do buraco até a apresentação dos dados na página web.

Nesse teste IX o sistema foi posicionado com o sensor acelerômetro na extremidade de uma mesa para que, no momento oportuno, fosse possível incliná-lo para baixo, alterando assim o valor do “AngleX” do acelerômetro (Figura 3). Após essa movimentação no acelerômetro, um buraco seria identificado, dando início a captura dos dados.

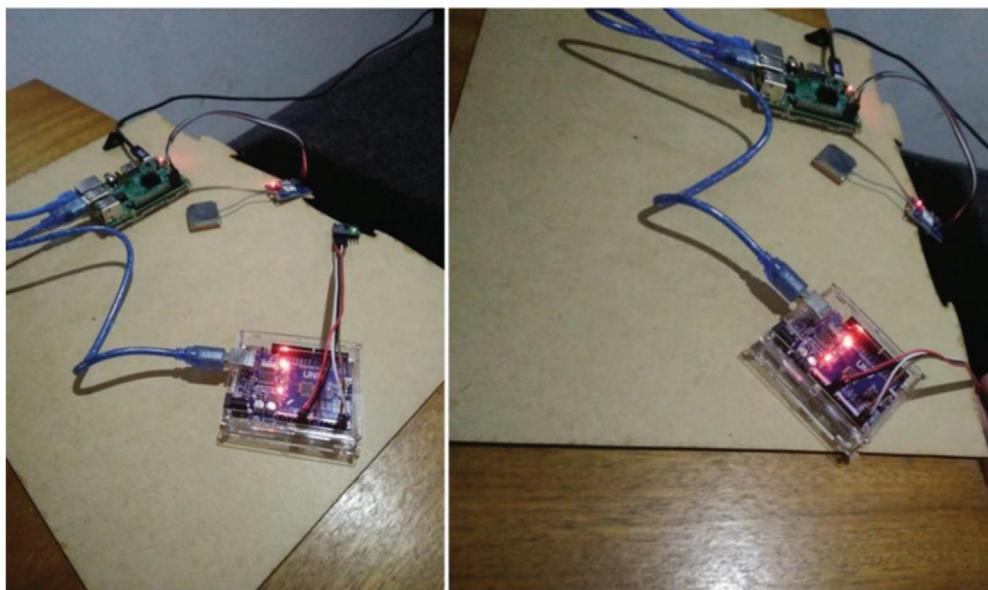


Figura 3 – Teste em bancada

5. Resultados

Com os testes realizados, foi possível verificar que no teste I, após percorrer os trajetos e acessado o sistema operacional do Raspberry PI, a aplicação foi capaz de encontrar 60 buracos e 95 buracos no teste I. No teste III, onde o sistema estava configurado para encontrar buracos em um intervalo menor de tempo, a aplicação foi capaz de encontrar 37 buracos.

Analizando o resultado dos três primeiros testes, verificou-se que há uma limitação do sistema para encontrar buracos em períodos menores que 10 segundos. A limitação ocorre pelo fato de que, após identificar um possível buraco, o sistema é programado para guardar as 10 próximas leituras do acelerômetro, que envia essa informação para o Raspberry a cada segundo. Como nesse momento ele está aguardando receber as 10 próximas leituras para gravar as informações em arquivo, foi colocada uma trava no sistema que só é liberada quando terminar a gravação para garantir a integridade dos dados coletados.

Essa funcionalidade de guardar as leituras seguintes foi inserida no sistema para que ela possa ser utilizada no desenvolvimento do algoritmo de identificação de buracos, sendo que, após esse algoritmo ser desenvolvido, existe a possibilidade de se retirar essa funcionalidade, permitindo reduzir o intervalo desse bloqueio. Além disso, em tese, com essas leituras o algoritmo para reconhecimento de buracos eliminaria o problema relatado por Moreira et al. (2013), de detectar um buraco quando o veículo está saindo do buraco que havia entrado.

O teste IV considerava um buraco quando os eventos captados pelos sensores tivessem a leitura do “AngleX” com valor acima de 16.500 e, com essa configuração, no trajeto percorrido, foram identificados 12 buracos. Demonstrando assim que é possível o

desenvolvimento de um algoritmo para identificação de buracos utilizando as variações de forças obtidas através do acelerômetro.

Nos testes V e VI, como resultado do envio dos dados coletados ao servidor web, no teste I o servidor retornou que foram inseridos 60 buracos, no envio dos buracos do teste II foram 95 buracos, e no trajeto aleatório do teste III, 12 registros foram inseridos no banco de dados.

Para confirmar se os dados foram realmente inseridos, foi acessado o gerenciador do banco de dados e confirmou-se, nas três ocasiões, que a inserção ocorreu corretamente e no mesmo instante que foram enviados.

O teste VII é semelhante ao anterior, e como resultado foram inseridos 244 novos registros no banco, e foi possível visualizar todos eles na página web. A Figura 4 mostra o envio através da aplicação no Raspberry PI com a quantidade de registros inseridos retornada pelo servidor web, e os registros atualizados no banco de dados. Com esse teste a persistência de dados foi colocada à prova. Nessa ocasião, o sistema demorou aproximadamente 1 minuto e 30 segundos para enviar todos os 244 registros armazenados, devido ao fato de ser inserido um registro por vez para garantir a atomicidade e a consistência dos dados, mas ao final, todos os dados foram inseridos. Ao realizar o teste VIII do consumo desses dados através do web service, ao clicar para buscar todos os buracos na página, foram apresentados todos os registros encontrados tanto na tabela quanto no mapa, com os marcadores apresentando as informações individuais de cada buraco, e ao utilizar os filtros, foram apresentados somente os dados filtrados.

A Figura 5 mostra a página web carregada após enviar os dados para o servidor, com o mapeamento e a tabela de buracos. No início da visualização do mapa os marcadores ficaram bem próximos uns dos outros, mas à medida que se aproximava a imagem utilizando o zoom, foi possível identificar os buracos através de cada marcador, melhor ilustrado pela Figura 6.

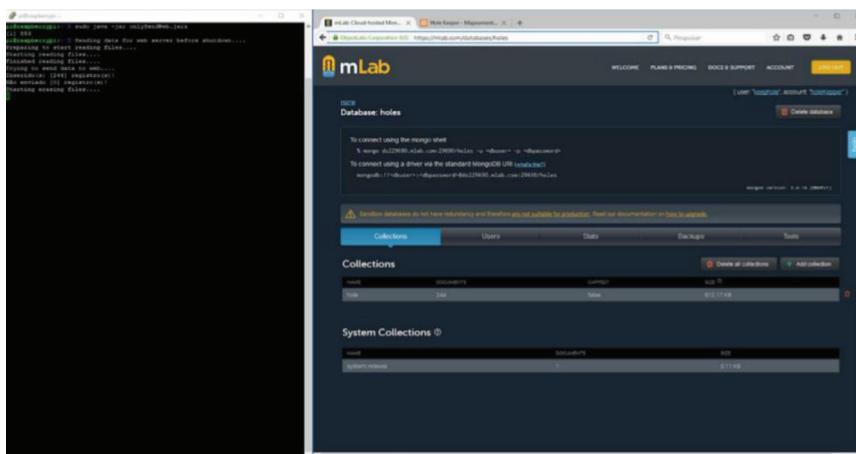


Figura 4 – Inserção no banco de dados. a) Logs da aplicação no Raspberry PI informando que foram enviados 244 registros b) Página do banco de dados demonstrando que foram armazenados 244 registros

The image shows two side-by-side windows of a web application. The left window is titled 'Hole Keeper' and displays a map of a road with several black markers indicating potholes. The right window is titled 'Hole Keeper - Mapeamento...' and shows a table of data with 30 rows. Each row contains a unique identifier (car100, car101, etc.), a timestamp (e.g., 2018-05-19T17:11:00), and coordinates (e.g., -26.82286, -49.05952). The table also includes a 'Status' column with values like '100% (2018-05-19T17:11:00)'.

car100	hole_20180519T171100	-26.82286	-49.05952
car100	hole_20180519T171100	-26.82222	-49.05933
car100	hole_20180519T164259	-26.82047	-49.05638
car100	hole_20180519T165708	-26.82053	-49.05643
car101	hole_20180519T165449	-26.83223	-49.05903
car101	hole_20180519T165449	-26.83607	-49.05603
car101	hole_20180519T165448	-26.82347	-49.05943
car101	hole_20180519T165444	-26.82578	-49.05219
car101	hole_20180519T165447	-26.81059	-49.05292
car101	hole_20180519T165447	-26.82578	-49.05219
car101	hole_20180519T165449	-26.82527	-49.05642
car100	hole_20180519T164249	-26.82062	-49.05639
car100	hole_20180519T164248	-26.82048	-49.05613
car101	hole_20180519T165226	-26.82073	-49.05208
car101	hole_20180519T165226	-26.82283	-49.05603
car101	hole_20180519T165226	-26.82057	-49.04222
car101	hole_20180519T165227	-26.82722	-49.05568
car101	hole_20180519T165227	-26.82193	-49.04208
car100	hole_20180519T161733	-26.82007	-49.05544
car100	hole_20180519T161733	-26.81713	-49.04503
car101	hole_20180519T160811	-26.83097	-49.04987
car101	hole_20180519T160811	-26.83097	-49.04987
car101	hole_20180519T160811	-26.83443	-49.07337
car101	hole_20180519T160812	-26.82742	-49.06938
car101	hole_20180519T160812	-26.81272	-49.05367

Figura 5 – Página web apresentando localização de possíveis buracos a) Site apresentando o mapa com os buracos identificados b) Informações detalhadas sobre os buracos identificados



Figura 6 – Zoom no mapa de buracos pra apresentá-los de forma mais visual

No teste final, teste IX, foram simulados 4 buracos. Com esse teste foi possível simular como o sistema se comportaria no ambiente real, desde a captura dos dados, a identificação do possível buraco até a apresentação dos dados ao usuário. O resultado foi positivo, pois foi possível visualizar os 4 buracos encontrados na página em tempo de execução, logo após o banco ter sido atualizado.

Os testes que foram realizados mostram o funcionamento da aplicação e demonstram a viabilidade do sistema para uso aplicado na detecção e mapeamento de buracos em rodovias, auxiliando na elaboração de programas para manutenção e melhorias de estradas.

6. Conclusões

O objetivo deste trabalho foi desenvolver uma ferramenta capaz de captar as informações e a localização de possíveis buracos encontrados em rodovias, disponibilizando e demonstrando esses dados na forma de um mapeamento de buracos nas rodovias.

O sistema necessita passar por várias etapas para chegar até o mapeamento apresentado ao usuário, e o desenvolvimento de módulos e a integração entre eles foi essencial para que essa funcionalidade pudesse ser oferecida. Como o objetivo inicial do projeto não era a identificação de buracos, mas sim o mapeamento dos mesmos, essa funcionalidade não foi desenvolvida.

Através dos Módulos de Captação de Dados (MCD) e Módulo Servidor Local (MSL) embarcados em um carro, foi possível identificar potenciais buracos e armazenar localmente esses dados. Com a integração dos Módulos Servidor Local (MSL) com o Módulo Servidor Web (MSW), foi possível enviar esses dados e armazená-los na nuvem, onde um Web Service disponibiliza a consulta a essas informações, consumidas através de uma página web (Módulo Cliente - MC) que apresenta um mapeamento dos buracos encontrados, buracos estes representados cada um por um marcador contendo as informações capturadas pelos sensores do MCD.

Os testes realizados atestaram a atomicidade e a consistência dos dados do sistema, além da sua disponibilidade no ambiente testado. Entretanto, para garantir a atomicidade dos dados, o desempenho da aplicação em identificar os possíveis buracos foi diminuído, limitando em 10 segundos o intervalo necessário para encontrar um próximo buraco, fator que poderá ser aperfeiçoado após o desenvolvimento do algoritmo para identificar efetivamente um buraco.

Ao visualizar o mapeamento dos dados na página, é possível identificar os pontos no trajeto percorrido através da identificação do carro e do horário de captura dos dados, permitindo assim a utilização da ferramenta por múltiplos usuários.

Com essas informações é possível concluir que o sistema é capaz de captar as informações dos possíveis buracos e apresentar um mapeamento de forma que o usuário pode localizar em que parte do trajeto há um possível buraco, gerando valor de utilidade ao sistema desenvolvido.

Como trabalhos futuros, sugere-se o desenvolvimento de um algoritmo capaz de identificar e classificar buracos usando os dados capturados pelo sensor acelerômetro;

fazer uso de câmeras para auxiliar na identificação de buracos; a instalação dos MCD nos quatro eixos de um carro, aumentando assim a performance do sistema em localizar buracos; e a implementação de uma solução para a inicialização do sistema e outro para enviar os dados para o servidor, de modo que o usuário não precise acessar o sistema operacional do Raspberry PI para efetuar o envio desses dados.

As melhorias sugeridas são para a página web do MC, na qual poderiam ser acrescentados outros tipos de filtros, otimizar a apresentação da tabela dos buracos, assim como implementar uma opção para buscar buracos por localidade e/ou região. Também seria interessante a implementação de um acesso à página como administrador, possibilitando a edição de dados, por exemplo a exclusão de buracos que foram recapeados ou registros muito próximos, que podem ser buracos duplicados.

Assim, o projeto desenvolvido pode ser utilizado como estrutura para um sistema completo de mapeamento de buracos em rodovias, no qual as melhorias realizadas pelos próximos trabalhos incrementariam a utilidade e usabilidade do sistema, fazendo com que a justificativa de se oferecer uma ferramenta útil tanto para quem necessita dar manutenção às rodovias quanto para quem é usuário sejam alcançadas.

Referências

- BORGES, P. de S. et al. (2011) Embedded system for detecting and georeferencing holes in roads. *IEEE Latin America Transactions*, 9(6), 921–925. DOI: 10.1109/TLA.2011.6096973
- CONFEDERAÇÃO NACIONAL DO TRANSPORTE (2016). *Pesquisa CNT de rodovias 2016: relatório gerencial*. 20. ed. Brasília: CNT: SEST: SENAT.
- MOREIRA, A. R. et al. (2013) *Sistema de detecção de buracos em estradas*. Trabalho de conclusão de curso (Graduação). Universidade Tecnológica Federal do Paraná, Departamento Acadêmico de Informática, Curso de engenharia de computação, Curitiba, Brasil.
- RAUTA, L. et al. (2017) Proposta de sistemas embarcados para identificação de buracos em rodovias utilizando as Leis da Mecânica de Newton. In: *Anais do Computer on the Beach 2017*. ISSN: 2358–0852.
- SINDICATO NACIONAL DA INDÚSTRIA DE COMPONENTES PARA VEÍCULOS AUTOMOTORES; ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE AUTOPEÇAS (2017). Relatório da frota circulante 2017. São Paulo: Sindicato Nacional da Indústria de Componentes para Veículos Automotores.
- TRINDADE, D. H. (2015) *Monitoramento de sistemas de transporte com Arduíno e Shield-GSM, GPS, GPRS*. Trabalho de conclusão de curso (Graduação). Universidade de Brasília, Faculdade UnB Gama, Curso de Engenharia Eletrônica, Brasília, Brasil.