

Uma Solução Baseada na Internet dos Veículos Inteligentes para a Vigilância Metrológica de Bombas de Combustível

Pedro Andrade* Lucas Marques* João Dias* Marianne Diniz*
Gabriel Signoretti* Ivanovitch Silva* Wilson de Souza Melo**
Carlos Galhardo**

* Universidade Federal do Rio Grande do Norte, Natal-RN
** Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro),
Petrópolis-RJ

E-mails: pedro.meira.055@ufrn.edu.br, lucas.marques.016@ufrn.edu.br,
joao.xavier.704@ufrn.edu.br, marianne.silva.086@ufrn.edu.br,
gabriel.sig@ufrn.edu.br, ivanovitch.silva@ufrn.br,
wsjunior@inmetro.gov.br, cegalhardo@inmetro.gov.br.

Abstract: The vehicle fuel market has a very high demand and involves a significant financial volume, being a great target for fraudsters. Criminals alter fuel pumps in order to reduce the amount delivered to the customer. In this context, this work proposes a solution for metrological surveillance of fuel supply companies based on built-in vehicle sensors. A Kotlin API was developed to collect vehicle data from an OBD-II reader (On-Board Diagnostics) that establishes communication between the vehicle and a computational cloud established by a mobile application prototype (Android operating system). The application was developed to detect supply events and inform the user if the volume filled corresponds to the amount paid.

Resumo:

O mercado de combustível para veículos possui uma demanda muito grande e envolve um volume financeiro significativo, sendo um grande alvo para fraudadores. Criminosos alteram as bombas de combustível com o objetivo de reduzir a quantidade entregue ao cliente. Nesse contexto, este trabalho propõe uma solução para a vigilância metrológica de empresas de abastecimento de combustível com base em sensores embutidos do veículo. Uma API Kotlin foi desenvolvida para coletar dados veiculares a partir de um leitor OBD-II (On-Board Diagnostics) que estabelece a comunicação entre o veículo e uma nuvem computacional estabelecida por um protótipo de aplicativo móvel (sistema operacional Android). A aplicação foi desenvolvida para detectar eventos de reabastecimento e informar ao usuário se o volume abastecido corresponde ao valor pago.

Keywords: Vehicle telemetry; Intelligent Vehicles; Metrology; Mobile Application, Fuel Supply.

Palavras-chaves: Telemetria veicular; Veículos Inteligentes; Metrologia; Aplicativo, Abastecimento de Combustível.

1. INTRODUÇÃO

A Internet das Coisas (*Internet of Things* – IoT) é um paradigma de comunicação recente no qual os objetos da vida cotidiana serão equipados com uma infraestrutura em rede que os tornarão capazes de se comunicarem entre si e com os usuários, tornando-se parte integrante da Internet. Tal funcionalidade abre aplicações nas mais diversas áreas como serviços públicos, domótica, eficiência energética, tomada de decisões e transporte (Oliveira et al., 2021; Zanella et al., 2014).

A incorporação dos dispositivos IoT (*Internet of Things*) pelo setor automotivo permitiu um alto grau de instrumentação com centenas de sensores para o monitoramento de variáveis internas como do veículo como a velocidade, as rotações por minuto (RPM), pressão, temperatura no mo-

tor e nível de combustível no tanque, através do protocolo OBD-II (*On Board Diagnostic*) (Silva et al., 2019). Esta massiva instrumentação permite inúmeras aplicações para o setor automotivo como diagnóstico, monitoramento, desempenho e manutenção; focando em áreas como a privacidade e a segurança dos usuários (Silva et al., 2018), em que a adoção de técnicas IoT voltadas para os veículos é definida como Internet dos Veículos Inteligentes (IoIV) (Wu and Liaw, 2018; Dandala et al., 2017).

Com o monitoramento do nível de combustível no tanque, diversas possibilidades se abrem como o controle e a supervisão metrológica realizada pelo Inmetro, em que um importante exemplo de instrumento de medição controlado e supervisionado é a bomba medidora de combustível líquido (BMC) (Melo Junior et al., 2021).

Para que uma transação comercial envolvendo um instrumento de medição seja justa, o instrumento precisa ser exato (sua medição deve concordar com os padrões nacionais), confiável (estável em relação às perturbações ambientais) e seguro (protegido contra modificações acidentais e não autorizadas).

As bombas medidoras de combustível líquido (BMCs) são instrumentos de medição que desempenham um papel crucial na sociedade brasileira, pois intermediam toda transação comercial de combustível líquido destinado ao consumidor final. Pode-se dizer que a maioria das BMCs em funcionamento no Brasil pertencem à classe dos chamados “instrumentos inteligentes” (*smart meters*), pois esses instrumentos utilizam em suas arquiteturas sistemas de controle e processamento dos dados metrológicos baseados em eletrônica microprocessada controladas por *software* embarcado (Dai et al., 2019).

Infelizmente, as arquiteturas dos instrumentos instalados no parque brasileiro apresentam vulnerabilidades para a segurança da informação metrológica. Dentre elas, destacam-se: (i) a falta de proteção dos dados de medição que trafegam no interior do instrumento, desde o transdutor da bomba, passando pelos seus dispositivos de controle e processamento, até a apresentação final do valor de medição no dispositivo indicador (*display*); e (ii) a inexistência de mecanismos eficientes de proteção do *software* embarcado no instrumento.

A existência de tais vulnerabilidades e o grande volume financeiro envolvido nas transações comerciais de combustíveis tornaram-se um grande atrativo para os fraudadores. Criminosos alteram intencionalmente as características originais da BMC, adulterando o valor das medições e impondo prejuízo ao consumidor final.

A evolução tecnológica das fraudes eletrônicas, no que diz respeito às suas formas de implementação, métodos de acionamento, efeitos causados e localização no interior do instrumento, o que tem representado um grande obstáculo para as operações de fiscalização em campo. A cada dia continuam surgindo novas e diferentes modalidades de fraudes, sempre em busca de ludibriar as equipes de fiscalização.

Torna-se então, absolutamente necessário o aprimoramento das tarefas de regulamentação e supervisão metrológica com o objetivo de combater as fraudes disseminadas em solo brasileiro. Diante deste contexto, o presente trabalho desenvolve uma solução que detecta eventos de reabastecimento através da leitura do sensor veicular de medição de volume de combustível, usando o protocolo OBD-II.

Considerando as contribuições do referido trabalho, podemos citar:

- Uma solução para vigilância metrológica de bombas de combustível baseada em sensores embarcados em um veículo típico.
- API Kotlin para coletar dados veiculares.
- Arquitetura de comunicação em tempo real entre veículos e nuvem computacional.
- Algoritmo para detecção de eventos de abastecimento.

- Um protótipo de aplicativo baseado no sistema operacional Android como prova de conceito para validação da proposta.
- Uma proposição de uma arquitetura emulada para experimentação.

Por fim, o restante desse artigo encontra-se organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados enquanto que na Seção 3 são elucidados detalhes da solução desenvolvida; a Seção 4 discute os principais experimentos e resultados obtidos enquanto que a Seção 5 enumera as ameaças à validade do respectivo estudo; por fim, a Seção 6 apresenta as considerações finais e indica caminhos promissores para trabalhos futuros.

2. TRABALHOS RELACIONADOS

2.1 Importância da Metrologia Legal e o desafio das fraudes em instrumentos de medição

O correto funcionamento de instrumentos de medição é requisito primordial para se garantir a idoneidade de processos que dependem desses dispositivos. Nesse contexto, a Metrologia Legal é a área de conhecimento responsável por estabelecer requisitos e atividades de controle de instrumentos de medição utilizados em relações de consumo (*i.e.*, negociações comerciais de produtos com base em medições como por exemplo tamanho, peso e volume) e procedimentos envolvendo saúde, segurança pública e proteção ao meio ambiente (Rodrigues Filho and Gonçalves, 2015). Em praticamente todo mundo, a Metrologia Legal atua por meio de organismos de controle metrológico que desempenham um conjunto de atividades específicas, entre as quais se destacam:

- A aprovação de modelo (*type approval*), em que um instrumento de medição é avaliado objetivamente com base em um regulamento técnico metrológico, de modo a se garantir que esse instrumento atende requisitos funcionais específicos, incluindo aqueles relacionados à confiabilidade das medições (precisão, acurácia, autenticidade e integridade das informações legalmente relevantes);
- A supervisão metrológica, que garante que os instrumentos de medição são fabricados e utilizados em conformidade com o modelo aprovado. Neste grupo, merecem ênfase as atividades envolvendo a vigilância de mercado (*market surveillance*), que verifica os instrumentos imediatamente após sua fabricação; e a vigilância de campo (*field surveillance*), que verifica o correto uso do instrumento em sua designação final).

No Brasil, as atividades de controle associadas a instrumentos sob controle legal são coordenadas pelo Inmetro¹ e executadas pela Rede Brasileira de Metrologia Legal e Qualidade - Inmetro (RBMLQ-I). A RBMLQ-I é constituída por 26 órgãos delegados que efetivamente executam as atividades no âmbito da Metrologia Legal em todo território nacional. Um dos grandes desafios enfrentados pela RBMLQ-I é a disseminação de fraudes de medição em medidores sob controle legal. Essas fraudes usualmente consistem em uma estratégia para a obtenção de vantagens financeiras indevidas, por meio da adulteração dos

¹ Instituto Nacional de Metrologia, Qualidade e Tecnologia

valores de medição providos por um instrumento. Em alguns segmentos, os montantes financeiros envolvidos são significativos, o que motiva a elaboração de fraudes cada vez mais sofisticadas, de difícil detecção e comprovação por meio de técnicas tradicionais de supervisão metrológica. Este é o caso do setor de distribuição de combustíveis para veículos automotores.

2.2 Fraudes em bombas medidoras de combustível

Bombas medidoras de combustível (BMCs) modernas são instrumentos de medição bastante sofisticados que realizam medições extremamente precisas quando utilizados de forma correta. Entretanto, os mesmos recursos tecnológicos que caracterizam uma BMC como um *smart meter* servem como plataforma para a implementação de fraudes igualmente sofisticadas, que exploram principalmente os componentes eletrônicos e o software embarcado desses instrumentos. No caso das BMCs, a fraude mostra-se altamente rentável para agentes maliciosos envolvidos em sua concepção, disseminação e execução. Um estudo de caso desenvolvido por (Rodrigues Filho and Gonçalves, 2016) reporta que, apenas no Brasil, as perdas com fraudes relacionadas a BMCs representam um montante na ordem de U\$ 300 milhões de dólares/ano. Essa ideia de alta disseminação de fraudes em BMCs é reforçada por outros trabalhos (Leitão et al., 2014; Beteto et al., 2016). Leitão et al. (2014) reportam um estudo de campo onde foram catalogados em todo o Brasil mais de 20 diferentes métodos de fraude em BMCs. Por sua vez, Beteto et al. (2016) analisam o cenário do Estado de São Paulo, demonstrando que a negociação de combustível adulterado e a implementação de fraudes metrológicas é um problema preocupante. É importante ressaltar que esse cenário não é exclusivo do Brasil, uma vez que fraudes na medição de combustível são reportadas por pesquisadores de diferentes países, em especial aqueles ainda em desenvolvimento (Luchsinger et al., 2008; Melo Junior et al., 2021).

Dada a seriedade e complexidade associadas ao problema das fraudes em BMCs, diversos esforços foram empreendidos nos últimos anos na tentativa de mitigar fraudes, identificando comportamentos suspeitos e dificultando sua implementação.

3. ARQUITETURA PARA VIGILÂNCIA METROLÓGICA DE BOMBAS DE COMBUSTÍVEL

Nesta seção são apresentados detalhes dos módulos da solução desenvolvida. A Figura 1 demonstra, de forma simplificada, a composição do fluxo de informação por meio da aplicação, sendo constituída por três grandes camadas: **Camada de Extração de Dados**, **Camada de Serviços** e **Camada de Análise dos Dados**. Esses três conceitos serão explicados minuciosamente nas subseções a seguir.

3.1 Camada de Extração de Dados

Nesta subseção serão descritas as principais características e funcionamento da **Camada de Extração de Dados**, conforme ilustra o primeiro grupo da Figura 1.

A finalidade desta camada é realizar as medições dos níveis do tanque do veículo do usuário e enviar esses dados utilizando o *Hypertext Transfer Protocol* (HTTP). Os componentes mais importantes para a comunicação Android-Veículo são explicados a seguir.

O primeiro dispositivo de suma importância para a aplicação é o *On-Board Diagnostic* (OBD-II), podendo ser qualquer modelo com conexão Bluetooth. Este *Hardware* é utilizado para realizar a comunicação entre a central do veículo, que por sua vez possui todos os dados medidos pelos diversos sensores espalhados no veículo, e o celular, onde a responsabilidade é executar o aplicativo. Os protocolos utilizados nessa comunicação são: a rede CAN para a comunicação com o veículo; e o Bluetooth - para a comunicação com o celular.

A principal funcionalidade desta camada está na aplicação que é executada pelo celular. Esta possui um total de sete telas desenhadas a partir de arquivos XML, onde cada uma está associada a um arquivo **Kotlin** responsável pela alteração dos componentes contidos em cada uma delas, seguindo a arquitetura *Model-View-ViewModel* (MVVM) (Developers, 2021), que consiste em manter as classes *Activity* e *Fragments* responsáveis pela Interface de Usuário (IU) do aplicativo, enquanto outras classes fazem o papel de criação e gerenciamento da conexão entre a IU e a API Rest - comentada nas próximas subseções -, sendo essas, respectivamente, classes de *Model* e *ViewModel*.

Ao iniciar o aplicativo, caso o celular do usuário disponha do protocolo Bluetooth, o usuário será solicitado que ligue o seu adaptador e, em seguida, estabeleça a conexão com o dispositivo OBD-II, que deve ser pareado previamente pelo usuário - essa seleção ocorre através de uma lista de dispositivos pareados. Após essa etapa de comunicação, o usuário deve informar o intervalo de medição em milissegundos, sendo o mínimo de 500ms, bem como a capacidade total do tanque de combustível do respectivo veículo. Em seguida, é realizada a leitura dos dados coletados pelo sensor de nível do tanque de combustível do carro, até que as últimas sejam iguais, parando a leitura automaticamente.

Para fazer requisições e coletar os dados do OBD-II conectado ao veículo através do Bluetooth, foi desenvolvida uma biblioteca de acesso aberto, disponibilizada no repositório do GitHub².

Como a requisição dos dados do sensor de nível do tanque do veículo são realizadas através de um Socket Bluetooth, é necessário que o aplicativo não seja fechado ou minimizado. Para solucionar esse problema, foi criada um *ForegroundService* cuja função é manter o Socket Bluetooth ativo e funcionando para as requisições ocorrerem normalmente.

Como foi citado anteriormente, visando a maior comodidade para o usuário, foi desenvolvido um sistema de detecção automático de fim do abastecimento, baseado nos valores das amostras coletados pelo OBD-II. Tal sistema consiste na checagem de n -variáveis, onde n é o número de amostras realizadas em um segundo mais uma amostra. O pseudo-código do Algoritmo 1 demonstra o funcionamento da função.

² Repositório do GitHub: <https://github.com/eltonvs/kotlin-obd-api>

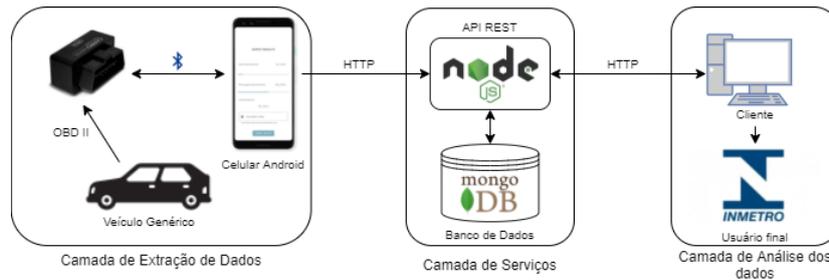


Figura 1. Arquitetura simplificada da plataforma proposta.

Algoritmo 1: Reconhecimento automático de fim do abastecimento

Entrada: *nvl_tanque*- Vetor com as medições dos níveis do tanque em cada instante;
intervalo- intervalo de medição definido pelo

usuário;

início

```

repita
  num ←  $\frac{5 * 1000}{intervalo} + 1$ 
  tam_vet ← Tamanho(nvl_tanque)
  se tam_vet > num então
    para cont ← 0 até cont == num faça
      ult_amst[cont] ← nvl_tanque[tam_vet - cont]
      se ult_amst[Tamanho(ult_amst)] == ult_amst[0] então
        ParaLeitura()
        controle ← 1
    senão
      controle ← 0
até controle == 1
    
```

Em suma, o que estamos fazendo é checando se a quantidade de leituras realizadas até o momento é suficiente para checarmos as amostras. Caso essa constatação seja verdadeira, sabendo que todas as medições do nível do tanque de combustível do veículo ficam armazenados em uma lista de tamanho m , caso m seja superior a n , o programa verificará se o último elemento desse vetor e o elemento de índice $m - n$ são iguais, isto é $lista[m] == lista[m - n]$. Se o resultado dessa operação for verdadeiro também, estará indicando que não há aumento no nível do tanque e, portanto, o abastecimento parou, conseqüentemente o aplicativo não fará mais requisições para o dispositivo OBD-II. Caso o resultado seja falso na primeira constatação, indica que o número de medições é menor que a quantidade de amostras necessárias para constatar o fim do abastecimento. Caso a segunda condicional seja falsa, estará indicando que o nível do tanque continua a aumentar e, portanto, o aplicativo continua com as requisições ao dispositivo OBD-II, ou seja, o programa continua o seu fluxo normalmente, até que este resultado seja verdadeiro.

Ao finalizar as leituras, o usuário deve informar a quantidade de litros abastecidos informados pela bomba de combustível do posto em questão e clicar no botão “Enviar

Relatório”, onde será aberta uma caixa de confirmação, para só então passar para uma página de confirmação de envio do relatório, onde, caso necessite, o usuário poderá reiniciar o processo, voltando para a página de leitura de dados.

3.2 Camada de Serviços

Nesta subseção, serão discutidas algumas técnicas e ferramentas que foram utilizadas para o funcionamento do sistema.

Para garantir a persistência dos dados coletados, foi desenvolvida uma *Application Programming Interface* (API) que utiliza uma arquitetura *Representational State Transfer* (REST) para receber as requisições enviadas pelo aplicativo. Tal API persiste os dados recebidos em um banco de dados NoSQL, o MongoDB, e também no *Amazon Web Services* (AWS). Além disso, foi utilizado o *Google Cloud Platform* (GCP) para hospedar o sistema.

A API foi desenvolvida em linguagem de programação Javascript, mais especificamente, a *runtime Node*, que é voltada para aplicações *web*.

Quando o aplicativo envia o relatório ao servidor, é feita uma requisição HTTP a um *endpoint* específico da API, contendo no corpo da requisição os dados obtidos no experimento. Assim, a API salva esses dados como um novo documento no MongoDB, que podem ser acessados posteriormente.

3.3 Camada de Análise de Dados

Nesta subseção será descrita a camada de análise dos dados, sua principal função e deveres, bem como o usuário final esperado.

Através de uma Aplicação Web simples, podendo esta ser ou não criada pela equipe, é possível comunicar a API REST com um cliente que pode ser acessado pelo usuário final da solução, que nesse cenário seria o Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO), cujo objetivo principal seria facilitar a vigilância metrológica de bombas de combustível.

O método de avaliação dos dados assim como a sua utilização são de total responsabilidade do usuário final, sendo a aplicação *web* que rodará no *desktop* ou *notebook* do usuário apenas uma interface de comunicação e visualização para com os dados armazenados no banco de dados da solução, explicado melhor na Subseção 3.2.

4. RESULTADOS E DISCUSSÕES

Para a utilização da API proposta, foram realizados alguns experimentos iniciais. O objetivo desses experimentos é avaliar sua viabilidade prática com relação a captura do sensor de nível de combustível e seus resultados finais, a partir dos dados disponibilizados.

Para isso, utilizou-se uma simulação por meio dos dados reais de capacidade do tanque de combustível de um veículo Volkswagen Up, 2017 com tanque de combustível com capacidade de 50 litros. Além disso, sua instrumentação é composta por:

- OBD-II:** dispositivo de prateleira de marca Multilaser para coletar o sensor nível de combustível.
- Smartphone:** dispositivo Android para comunicação (Bluetooth) com o OBD-II.
- Freematics OBD-II Emulator MK2:** Tal emulador dispõe uma porta OBD-II fêmea de 16 pinos idêntica às reais encontradas nos veículos e responde à requisições PID OBD-II, incluindo DTC (*Diagnostic Trouble Codes*) e VIN (*Vehicle Identification Number*) de dispositivos plugados. Dessa forma, o emulador se torna muito prático e útil para aplicações que necessitam de testes em bancada, sobretudo relacionado ao desenvolvimento de novas soluções. O dispositivo pode ser controlado por um *software GUI*³ via cabo USB conectado a um computador *Desktop* ou *Notebook*, ou até mesmo por conexão sem fio por meio de dispositivos iOS, como iPad ou iPhone, através do *Freematics Emulator App*.

Assim, apresentam-se como resultados da operação os dados de captura do sensor e seu nível de combustível final após abastecimento. Dessa forma, conectou-se o OBD-II no *Freematics OBD-II Emulator MK2*, em seguida a API ao OBD-II - para realizar a captura de dados do sensor e posteriormente, após definição do valor a ser abastecido (Figura 2) e nível inicial do tanque, obtem-se a quantidade de combustível final (Figura 3).

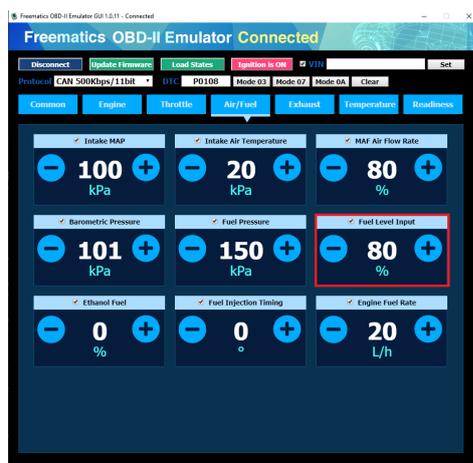


Figura 2. Captura do Software Emulador - Abastecimento.

A Tabela 1, disponibiliza os resultados obtidos após as simulações em 5 cenários, variando a quantidade de litros

³ Site contendo informações sobre o software do *Freematics OBD-II Emulator GUI*: <https://freematics.com/pages/software/freematics-obd-emulator-gui/>

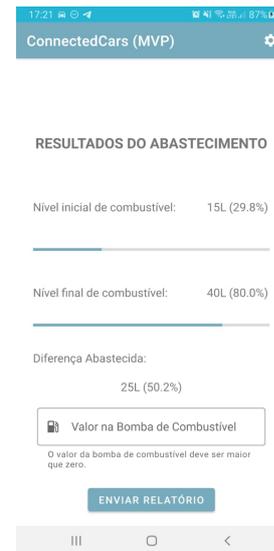


Figura 3. Captura da API - Cenário 3.

no abastecimento e a capacidade inicial do tanque de combustível.

Tabela 1. Cenários de Avaliação.

Cenário	Abastecimento	Nível Inicial	Nível Final	Total
1	10 litros	50%	69,8%	35 litros
2	15 litros	40%	69,8%	35 litros
3	25 litros	30%	80,0%	40 litros
4	35 litros	20%	89,8%	45 litros
5	45 litros	10%	99,6%	50 litros

Percebe-se na Tabela 1 que apenas o Cenário 3 ficou com o valor fechado, visto que os dados são aproximados, conforme definido na Seção 3.

Para complementar os resultados, apresenta-se a captura dos dados armazenados na base de dados (Figura 4).

```

_id: ObjectId("609dca8c79c40d000bc12d2b")
autor: "SAMSUNG SM-N9600"
horarioInicio: "1620953666"
horarioFim: "1620953691"
volumeInicio: 20
volumeFim: 89.8
volumeTotalTanque: 50
latitude: -1
longitude: -1
nomeEstabelecimento: "None"
numeroBomba: -1
__v: 0
    
```

Figura 4. Captura de uma das amostras no banco de dados.

Sendo importante a notificação de que as variáveis cujos valores são -1 e None devem ser ignoradas para essa primeira versão da aplicação.

5. AMEAÇAS À VALIDADE

5.1 Validade de Construção

Esse tipo de ameaça existe pelo fato dos experimentos realizados não terem abordado todos os cenários.

5.2 Validade de Conclusão

A principal ameaça é a limitação no gerenciamento dos riscos e, portanto, pode ter influenciado os resultados do experimento.

5.3 Validade Externa

As conclusões para o cenário real não podem ser generalizadas, pois foi realizada uma experimentação simulada.

6. CONCLUSÃO

Este artigo apresentou uma solução para Veículos Inteligentes para vigilância metrológica de bombas de combustíveis. Foram realizados experimentos em cenários emulados com o objetivo de investigar a viabilidade da solução, na captura do sensor de nível de combustível, bem como os cálculos utilizados para verificação do resultado final da quantidade de combustível abastecido. Os resultados indicaram a viabilidade da proposta. Com relação as estimativas de quantidade final, os resultados são aproximados e apenas no Cenário 3 foi apresentado o valor real.

Trabalhos futuros incluem, mas não se limitam a: realização de experimentos em cenários reais, aplicação de técnicas de Aprendizado de Máquina para identificar padrões, inclusão das funcionalidades de localização, número da bomba e nome do estabelecimento. Ademais, neste artigo, o usuário colocou o valor abastecido como entrada numérica. Em futuras versões do aplicativo, pretendemos utilizar a fotografia do visor da bomba de combustível para evitar erros acidentais ou voluntários pelo usuário.

AGRADECIMENTOS

O presente artigo foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo nº 435683/2018-7.

REFERÊNCIAS

- Beteto, A., Melo, V., and Dias, E. (2016). Fuel Reselling: Electronic Documents and Tax Surveillance. *International Journal of Economics and Management Systems*, 1, 163–168.
- Dai, H.N., Zheng, Z., and Zhang, Y. (2019). Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5), 8076–8094. doi:10.1109/JIOT.2019.2920987.
- Dandala, T.T., Krishnamurthy, V., and Alwan, R. (2017). Internet of vehicles (ioV) for traffic management. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 1–4. doi:10.1109/ICCCSP.2017.7944096.
- Developers, G. (2021). Guide to app architecture. URL <https://developer.android.com/jetpack/guide>.
- Leitão, F.O., Vasconcelos, M.T., and Brandão, P.C.R. (2014). Hardware and Software Countermeasures on High Technology Fraud at Fuel Dispensers under the Scope of Legal Metrology. In *IX Simposio Internacional 'Metrologia 2014'*, 1–10. Havana.
- Luchsinger, H., Cajica, C., Maldonado, M., and Castelazo, I. (2008). Are Gas Pumps Measuring Up? The Mexican Experience. *NCSLI Measure*, 3(2), 62–68.
- Melo Junior, W., Tarelho, L., Rodrigues Filho, B., Bessani, A., and Carmo, L. (2021). Field surveillance of fuel dispensers using iot-based metering and blockchains. *Journal of Network and Computer Applications*, 175, 102914. doi:10.1016/j.jnca.2020.102914.
- Oliveira, F., Nery, D., Costa, D.G., Silva, I., and Lima, L. (2021). A survey of technologies and recent developments for sustainable smart cycling. *Sustainability*, 13(6). doi:10.3390/su13063422. URL <https://www.mdpi.com/2071-1050/13/6/3422>.
- Rodrigues Filho, B.A. and Gonçalves, R.F. (2015). Legal metrology, the economy and society: A systematic literature review. *Measurement*, 69, 155–163. doi:10.1016/j.measurement.2015.03.028.
- Rodrigues Filho, B.A. and Gonçalves, R.F. (2016). Measuring the economic impact of metrological frauds in trade metrology using an Input-Output Model. *IFIP Advances in Information and Communication Technology*, 488. doi:10.1007/978-3-319-51133-7.
- Silva, M., Signoretti, G., Oliveira, J., Silva, I., and Costa, D.G. (2019). A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective. *Future Internet*, 11(1). doi:10.3390/fi11010013.
- Silva, M., Vieira, E., Signoretti, G., Silva, I., Silva, D., and Ferrari, P. (2018). A customer feedback platform for vehicle manufacturing compliant with industry 4.0 vision. *Sensors (Basel)*, 18, 13. doi:10.3390/s18103298.
- Wu, W.C. and Liaw, H.T. (2018). The next generation of internet of things: Internet of vehicles. In J.C. Hung, N.Y. Yen, and L. Hui (eds.), *Frontier Computing*, 278–282. Springer Singapore, Singapore.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22–32. doi:10.1109/JIOT.2014.2306328.