

# Prototipagem de uma Placa Veicular para Ecossistema de Cidades Inteligentes

Benjamin Silva <sup>1</sup>, Ivan Saraiva <sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Federal do Piauí (UFPI)  
Teresina – PI – Brasil

benjamin.santos@ufpi.edu.br, ivan@ufpi.edu.br

**Abstract.** *Dysfunctional urban mobility is an issue related to the horizontal growth of cities. Among the main problems in this context are traffic congestion, traffic accidents, lack of coordination in vehicle flow, vehicle thefts and robberies, waiting times for payments, traffic violations, among other issues. The Intelligent Vehicle Plate is a component of a transportation system that integrates into the smart cities ecosystem, aiming to mitigate these problems through the monitoring and management of urban traffic. This work presents the functionality and prototyping of congestion detection of the plate, developing the software layer and integration of discrete devices.*

**Resumo.** *A mobilidade urbana disfuncional é um problema relacionado ao crescimento horizontal das cidades. Entre os principais problemas nesse contexto, estão o congestionamento de tráfego, acidentes de trânsito, ausência de coordenação no fluxo de veículos, furtos e roubos de veículos, tempo de espera para pagamentos, infrações de trânsito, entre outros problemas. A Placa Veicular Inteligente é um componente de um sistema de transporte que integra o ecossistema de cidades inteligentes que visa mitigar esses problemas por meio do monitoramento e gerenciamento do tráfego urbano. Este trabalho apresenta a funcionalidade e prototipagem da detecção de congestionamento da placa, desenvolvendo a camada de software e integração dos dispositivos discretos.*

## 1. Introdução

O crescimento urbano horizontal é a expansão das áreas urbanas para regiões periféricas. Quando esse processo acontece sem planejamento e consequente aumento da infraestrutura urbana, esse crescimento é categorizado como disfuncional. Segundo os especialistas, o crescimento disfuncional traz diversos problemas para o bem-estar social e ambiental, além de desafios para gestão pública das cidades [Sun et al. 2020]. O crescimento disfuncional é particularmente impactante nos problemas relacionados a mobilidade urbana, pois aumentam as distâncias entre pontos de interesse dos cidadãos, aumentando também a necessidade e demanda de transporte para o deslocamento entre esses pontos [Alvim et al. 2024].

Associado a isso, estão o congestionamento de tráfego, pois as vias urbanas crescem em um ritmo inferior que do fluxo de veículos que trafega por elas diariamente, resultando em uma sobrecarga das vias e aumentando o tempo de descolamento nas cidades [Jilani et al. 2023]. Além da limitação das vias, pontos de pedágios e fila de espera em estacionamento são gargalos no deslocamento pelas vias urbanas. Outro problema é

dificuldade de coordenar o fluxo de veículos cada vez maior, agravando o problema de congestionamento e proporcionando um ambiente mais inseguro e favorável a acidentes [Retallack and Ostendorf 2019]. Junto a isso, existe o desafio de fiscalização das vias pelos órgãos públicos para evitar infrações de trânsito [Portal do Trânsito 2023]. Por fim, um problema adjacente é o furto e roubo de veículos no contexto urbana, resultado de vários fatores complexos, que afeta a toda a sociedade [IBGE 2021].

A Placa Veicular Inteligente é um dispositivo que compõe um sistema de transporte que integra um ecossistema de cidades inteligentes, cujo objetivo é mitigar os efeitos dos problemas relacionados à mobilidade urbana mencionados anteriormente.

## 2. Trabalhos relacionados

As cidades inteligentes podem ser definida como uma forma de desenvolvimento urbano que tem por objetivo solucionar problemas desse ambiente, utilizando computação, recursos humano e social, e o poder público para o crescimento econômico, social, ambiental e cultural, que integra várias dimensões para melhorar a qualidade de vida de seus habitantes [Albino et al. 2015].

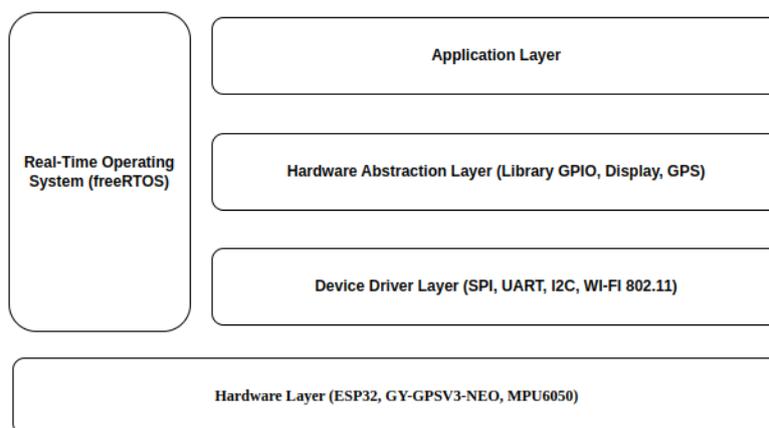
Entre essas dimensões, está a mobilidade inteligente, que é um dos principais componentes das cidades inteligentes. A mobilidade inteligente reduz os congestionamentos, tempos de deslocamento e acidentes, gerando um maior qualidade de vida para seus cidadãos, e sua definição exata varia conforme a visão e as condições de cada país e/ou cidade [So et al. 2020]. Apesar disso, segundo Biyik, a aplicação da mobilidade inteligente está restrita a quatro segmento, são eles: sistema de transporte inteligente, dados abertos e programas de código aberto, ciência de dados no contexto de *big data* e participação ativa dos cidadãos, que trabalham de forma inter-dependente [Biyik et al. 2021].

Os *Intelligent Transport Systems* (ITS) são soluções que utilizam tecnologias modernas de estatísticas e telecomunicações para integrar humanos, vias urbanas e veículos, permitindo comunicação com a infraestrutura (V2I) e comunicação entre veículos (V2V) [Ravi and Mamdakar 2022]. Nesse contexto, surge o conceito de borda inteligente aplicada a ITS, permitindo aplicações com menor latência, maior segurança, menos pressão na rede principal e melhor uso de grandes volumes de dados na borda [Gong et al. 2023].

Além do processamento de informações pela borda, outro aspecto central dentro de ITS é a conectividade, um dentro outros conceitos fundamentais para conectividade são as Redes Ad Hoc Veiculares (VANETs), que são um tipo particular de Redes Ad Hoc Móveis. Essas redes têm características únicas, como alta mobilidade e topologia dinâmica, exigindo novos algoritmos e protocolos específicos [Sakiz and Sen 2017].

## 3. Metodologia

Para o desenvolvimento do protótipo, as atividades foram divididas em três partes, são elas a arquitetura de software e aplicações, o projeto e montagem do circuito físico, e a integração entre os sistemas de software e hardware. Como o objetivo de construir uma prova de conceito para as funcionalidades da placa e a integração entre elas. Nesse trabalho, será apresentado a funcionalidade de detecção de congestionamento.



**Figura 1. Diagrama da arquitetura de software**

### 3.1. Arquitetura de software

A arquitetura de software do protótipo é organizada em camadas (Figura 1), com cada camada sendo responsável por um conjunto específico de funcionalidades e interagindo com as camadas adjacentes (Figura 1). As camadas de software são divididas da seguinte forma: *Device Driver Layer*, *Hardware Abstraction Layer* (HAL), *Real-Time Operating System* (RTOS), e *Application Layer*. Para a prototipagem da placa, as três primeiras camadas de software foram implementadas utilizando o *Espressif IoT Development Framework* (ESP-IDF) [Espressif™].

A *Device Driver Layer* é composta por interfaces de baixo nível para interação entre os dispositivos de hardware e as camadas do RTOS e HAL. Sua principal função é abstrair as operações de controle complexas de cada dispositivo, fornecendo uma interface simplificada para o restante do sistema. Essa camada gerencia a comunicação direta com os componentes de hardware, como sensores, interfaces de comunicação e periféricos de entrada/saída, traduzindo comandos de alto nível em instruções específicas do dispositivo. No protótipo do detector de congestionamento, foram utilizados drivers UART para comunicação serial como módulo GPS, I2C para gerenciamento do acelerômetro, e ESP-WiFi para conectividade sem fio, todos presentes no ESP-IDF.

A *Hardware Abstraction Layer* fornece uma interface uniforme para as camadas superiores do sistema, abstraindo os detalhes específicos do hardware subjacente. Sua principal função é oferecer uma camada de abstração que permite ao software interagir com o hardware de maneira independente do dispositivo. A HAL gerencia a comunicação entre os drivers de dispositivo e as camadas de RTOS e da aplicação, permitindo a integração do código. Essa camada fornece chamada de funções padronizadas para acessar e controlar periféricos e temporizadores. No protótipo, a HAL do ESP-IDF foi utilizada para garantir que as operações nos periféricos fossem realizadas de forma independente das variações de hardware.

A *Real-Time Operating System* gerencia a execução de tarefas em tempo real, fornecendo serviços essenciais como escalonamento, sincronização e comunicação inter-tarefas. Sua principal função é garantir que as tarefas sejam executadas dentro de prazos específicos, mantendo a estabilidade e a previsibilidade do sistema. No protótipo, foi

utilizado o FreeRTOS, presente no ESP-IDF, que permitiu a criação e o gerenciamento de múltiplas tarefas, temporizadores e filas de comunicação. Isso assegurou que as operações fossem realizadas de maneira coordenada, atendendo aos requisitos de tempo real do sistema embarcado [AWS™].

### 3.2. Projeto de hardware

O desenvolvimento do projeto de hardware para detecção de congestionamento baseou-se na utilização de módulos. Um dos módulos de desenvolvimento selecionada foi o *ESP32-DevKitV1*, como o núcleo da placa veicular devido ser um microcontrolador *dual-core* e ter conectividade Wi-Fi, e pelo *framework* de desenvolvimento que permite a simplificação da prototipagem de funcionalidades. Para a aquisição de dados de movimento, foi utilizado o sensor MPU6050, que combina um acelerômetro e um giroscópio, fornecendo informações sobre a aceleração e a orientação do veículo, utilizados para detectar mudanças bruscas de velocidade e comportamento de direção que podem indicar condições de tráfego congestionado.

Também, foi empregado o módulo GY-GPSV3-NEO, que integra um receptor GPS NEO-6M, para fornecer dados de localização em tempo real. Este módulo GPS permite o monitoramento da posição do veículo, fornecendo informações para a detecção de congestionamentos baseadas na análise do descolamento do veículo. A combinação desses módulos no protótipo permitiu a coleta e o processamento de dados de movimento e localização, a integração dos componentes foi realizada com base em interfaces como I2C para o MPU6050 e UART para o GY-GPSV3-NEO.

## 4. Resultados

---

### Algorithm 1 Detecção de Congestionamento na Borda

---

**Require:**  $v_{limite}$ ,  $limiar\_variancia$

**Ensure:** Alerta de congestionamento local

```
1: while verdadeiro do
2:   Coletar  $dados\_gps$  da fila de comunicação GPS
3:   Coletar  $dados\_acelerometro$  da fila de comunicação do acelerômetro
4:   Filtrar ruídos dos  $dados\_acelerometro$ 
5:   Calcular  $magnitude\_aceleracao$ 
6:   Calcular  $velocidade\_media$  dos  $dados\_gps$ 
7:   Calcular  $variancia\_aceleracao$ 
8:   if  $velocidade\_media < v_{limite}$  and  $variancia\_aceleracao < limiar\_variancia$  then
9:     Marcar como possível congestionamento
10:  else
11:    Marcar como tráfego normal
12:  end if
13:  Compartilhar estado de congestionamento e vetor  $velocidade\_media$ 
14: end while
```

---

A *Application Layer* no protótipo tem a função de executar as lógicas dos serviços prestados, incluindo a detecção de congestionamento. O algoritmo de detecção de congestionamento baseado em GPS e acelerômetro está integrado nesta camada para analisar os dados recebidos e fornece informações sobre as condições de tráfego.

---

**Algorithm 2** Detecção de Congestionamento Distribuído

---

**Ensure:** Alerta de congestionamento global

```
1: while verdadeiro do
2:   Coletar estados de congestionamento dos nós vizinhos da fila de comunicação
3:   Realizar agregação dos estados recebidos
4:   if maioria dos vizinhos reportar congestionamento then
5:     Confirmar estado de congestionamento
6:     Gerar alerta de congestionamento
7:   else
8:     Manter estado como tráfego normal
9:   end if
10: end while
```

---

A implementação do algoritmo é dividido em duas etapas, *edge computing* (Algorithm 1) e *distributed computing* (Algorithm 2). Na primeira etapa, o processamento inicial dos dados de sensores é realizado localmente. Isso inclui a coleta de dados, pré-processamento, e detecção de congestionamento local. Essa abordagem reduz a latência de comunicação e a permite uma economia de largura de banda. Na segunda etapa, os dados processados localmente (estado de congestionamento) são compartilhados entre veículos na rede descentralizada. Cada veículo coleta informações de seus vizinhos, realiza agregação de dados e chega a um consenso sobre o estado do tráfego.

Os algoritmos adotada tem potencial para melhorar a detecção de congestionamento, com benefícios esperados na redução de latência e na economia de largura de banda. Embora os resultados ainda sejam preliminares, a combinação de detecção local e distribuída parece promissora para uma identificação mais precisa das condições de tráfego em ambientes descentralizados.

## 5. Conclusão

Este trabalho apresentou uma visão geral da placa veicular inteligente e da funcionalidade de detecção de congestionamento, como uma prova de conceito para validar a arquitetura de software e o projeto de hardware. O protótipo tem o objetivo de integrar dispositivos discretos e camadas de software para monitorar e gerenciar o tráfego urbano. Com a coleta e análise de dados em tempo real, o sistema conseguirá identificar padrões de congestionamento. Além disso, a Placa Veicular Inteligente tem o potencial de melhorar a coordenação do fluxo de veículos e aumentar a segurança nas vias, reduzindo acidentes. Em trabalhos futuros, o aprimoramento dos algoritmos para obter maior precisão e a exploração de novas funcionalidades, a fim de atender as diversas demandas dos meios urbanos. Os resultados do protótipo indicam que esta tecnologia tem pode ser amplamente adotada, contribuindo para cidades mais inteligentes e uma mobilidade urbana mais saudável.

## Referências

Albino, V., Berardi, U., and Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22:21 – 3.

- Alvim, A. T. B., Izaga, F. G. d., and Claps, R. F. (2024). Mobilidade urbana em perspectiva: novos olhares sobre as dinâmicas da cidade contemporânea. *Cadernos Metrópole*, 26(60):413–421.
- AWS™. FreeRTOS docs. <https://www.freertos.org/features.html>. Accessed: 2024-07-15.
- Biyik, C., Abareshi, A., Paz, A., Ruiz, R. A., Battarra, R., Rogers, C. D. F., and Lizárraga, C. (2021). Smart mobility adoption: A review of the literature. *Journal of Open Innovation: Technology, Market, and Complexity*.
- Espressif™. Esp-idf programming guide. <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>. Accessed: 2024-07-15.
- Gong, T., Zhu, L., Yu, F. R., and Tang, T. (2023). Edge intelligence in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):8919–8944.
- IBGE (2021). Pnad contínua: Vitimização: Furtos e roubos 2021. Accessed: 2024-06-15.
- Jilani, U., Asif, M., Zia, M. Y. I., Rashid, M., Shams, S., and Otero, P. (2023). A systematic review on urban road traffic congestion. *Wireless Personal Communications*.
- Portal do Trânsito (2023). Falta fiscalização de trânsito em três de cada quatro municípios. Accessed: 2024-06-15.
- Ravi, S. and Mamdakar, M. R. (2022). A review on its (intelligent transportation systems) technology. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 155–159.
- Retallack, A. E. and Ostendorf, B. (2019). Current understanding of the effects of congestion on traffic accidents. *International Journal of Environmental Research and Public Health*, 16(18).
- Sakiz, F. and Sen, S. (2017). A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks*, 61:33–50.
- So, J. J., An, H., and Lee, C.-J. (2020). Defining smart mobility service levels via text mining. *Sustainability*, 12:9293.
- Sun, L., Chen, J., Li, Q., et al. (2020). Dramatic uneven urbanization of large cities throughout the world in recent decades. *Nat Commun*, 11:5366.