

# NexCar: Driven by Inovation

José Pedro Ribeiro Novais  
Eng de Telecom e Informática  
EEUM  
Guimarães, Portugal  
a105056@alunos.uminho.pt

Miguel Freixo Machado  
Eng de Telecom e Informática  
EEUM  
Barcelos, Portugal  
a103668@alunos.uminho.pt

Tiago Rigueira Soares Diogo  
Eng de Telecom e Informática  
EEUM  
Aveiro, Portugal  
a103665@alunos.uminho.pt

**Abstract**— Este artigo apresenta o design e a implementação do NEXCAR, um sistema de monitorização e controlo veicular em tempo real que integra sensores físicos, atuadores, uma API conectada à nuvem e interfaces de utilizador multiplataforma. O sistema tem como objetivo aumentar a segurança veicular e a consciencialização ambiental através do uso de protocolos de comunicação leves (MQTT), APIs REST e lógica de automação entre sensores e atuadores. A validação experimental demonstra a fiabilidade do sistema sob diversas condições físicas.

**Keywords**— IoT, monitorização veicular, MQTT, API REST, sensores, automação, aplicação móvel.

## I. INTRODUÇÃO

A mobilidade inteligente e a segurança rodoviária estão a ser cada vez mais potenciadas pela integração da Internet das Coisas (IoT) no setor automóvel. Por exemplo, sistemas de monitorização da velocidade veicular baseados em IoT demonstraram ser eficazes na recolha de dados em tempo-real para análises de desempenho e deteção de situações de risco [1]. Em paralelo, as redes veiculares (Internet of Vehicles, IoV) enfrentam desafios técnicos relativos à latência de rede e escalabilidade, os quais têm sido alvo de estudos aprofundados que propõem otimizações específicas para ambientes de tráfego dinâmicos [2]. Este artigo tem como tema o desenvolvimento do sistema NEXCAR no âmbito da Unidade Curricular de Projeto da Licenciatura em Engenharia de Telecomunicações e Informática da Universidade do Minho, Campus de Azurém, Guimarães. Com este projeto, tivemos de definir a arquitetura mais adequada para a monitorização de veículos inteligentes, pelo que a nossa pesquisa inicial se centrou na identificação de sensores, seleção de protocolos de comunicação e frameworks de backend. Em paralelo, explorámos novas linguagens de programação e ferramentas de base de dados, bem como a implementação de uma API REST segura e um frontend em HTML com Socket.IO para visualização em tempo real. A gestão de tarefas foi realizada através de metodologias ágeis, permitindo-nos coordenar eficazmente as fases de investigação, desenvolvimento e testes. As 3 fases deste projeto resumem-se em:

### A. Fase A- Sistema Sensor

Nesta fase foi desenvolvido um Sistema Sensor com o uso de duas ESP8266 e vários sensores com o envio dos dados recolhidos para o servidor online (Thingspeak).

### B. Fase B-Sistema Central

A Fase B garantiu a validação do pipeline de aquisição e armazenamento de dados, bem como o desenvolvimento de

uma interface de administração mínima, permitindo confirmar que os sensores (físicos ou simulados) conseguem publicar informações através do MQTT, que estas são captadas corretamente pelo sistema central e que podem ser visualizadas de modo simples por um utilizador administrador.

### C. Fase C- Base de Dados e Interface

Esta fase incluiu a definição completa do esquema relacional, o desenvolvimento do backend e das APIs, a criação da interface web de administração com todos os módulos de gestão e, por fim, a especificação e implementação das interfaces destinadas aos utilizadores finais.

## II. ARQUITETURA DO SISTEMA

A arquitetura do sistema NEXCAR foi concebida com foco na modularidade, escalabilidade e integração entre sensores físicos e aplicações front-end, tanto web como móveis. A Figura 1 apresenta uma visão geral da arquitetura.

O sistema é composto por três grandes blocos funcionais: aquisição de dados dos sensores, processamento e armazenamento, e interface com o utilizador.

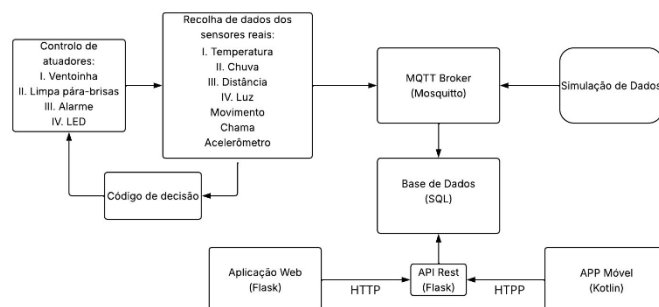


Figura 1- Arquitetura do Sistema

### A. Aquisição e Publicação de Dados:

A recolha de dados é realizada por um conjunto de sensores reais que monitorizam variáveis ambientais e de movimento do veículo, nomeadamente:

- Temperatura – SHT3x
- Chuva – YL-83
- Distância – HC-SR04
- Luz – LDR
- Movimento – PIR HC-SR501

- Chama – IR Flame Detector
- Acelerómetro – MPU6050

Os dados recolhidos são publicados num broker MQTT (Mosquitto), que atua como middleware leve e eficiente para comunicação assíncrona baseada em tópicos. Esta comunicação ocorre via tópicos estruturados, como /simulados/temperatura, /simulados/chuva, entre outros, ou /nexcar/sensor/\* em modo real. A arquitetura permite ainda a publicação de dados simulados para efeitos de testes, via módulo de simulação. Tópicos MQTT Publicados

## B. Comunicação por MQTT: Publicação e Subscrição

O protocolo MQTT é responsável por intermediar a comunicação entre os sensores e o backend do sistema. Cada sensor, físico ou simulado, publica os seus dados em tópicos individualizados. Os módulos subscritores (nomeadamente os scripts subscriber.py e subscriber\_real.py) estão configurados para escutar todos os tópicos relevantes através do padrão TOPIC\_BASE/#. À medida que as mensagens são recebidas, os dados são temporariamente armazenados, agregados e, uma vez completo o conjunto necessário, inseridos na base de dados relacional sensor\_data.db para posterior processamento.

Tópicos utilizados incluem:

- /simulados/temperatura, /simulados/humidade
- /simulados/fogo, /simulados/colisao
- /simulados/chuva, /simulados/distancia, /simulados/movimento
- /simulados/timestamp, /simulados/car\_id

Este mecanismo permite o processamento em tempo real com baixa latência e alta escalabilidade.

## C. Processamento, Armazenamento e Ação

Os dados armazenados na base de dados SQL são posteriormente analisados por um módulo de código de decisão, responsável por interpretar os valores sensoriais e tomar ações automáticas sempre que condições críticas são detetadas.

Dependendo das leituras, o Sistema pode ativar os seguintes atuadores:

- Ventoinha (quando a temperatura excede um limite)
- Limpa para-brisas (em caso de deteção de chuva)
- Alarme Sonoro (em caso de perigo de colisão)
- LED (ativado por baixa luminosidade)

Este modelo permite uma resposta automatizada e em tempo real.

## D. Interface de Utilizador

O sistema oferece duas formas principais de interação com o utilizador:

- Uma aplicação web desenvolvida com o framework Flask, acessível via HTTP;
- Uma aplicação móvel desenvolvida em Kotlin, que comunica também via API REST (Flask).

A API REST serve de ponte entre a base de dados e as interfaces de utilizador, garantindo o acesso controlado às informações do sistema e permitindo a monitorização remota do estado do veículo, histórico de eventos e estado dos sensores.



Figura 2- Base de Dados

## III. FUNCIONALIDADES DO SISTEMA

Sistema Web:

### A. Dashboard do Admin

A interface administrativa da WebApp permite o controlo e monitorização de diversas funcionalidades críticas do sistema:

1. Visualização de dados dos sensores em tempo real(ver Fig 3):

O admin tem acesso aos dados dos sensores de todos os clientes em tempo real e às suas estatísticas.

2. Visualização de eventos:

O admin tem acesso aos eventos em tempo real como deteção de colisão e deteção de chuva.

3. Gestao de Utilizadores:

O admin consegue filtrar a procura por clientes e consegue remover os carros que desejar de qualquer cliente.

4. Chat para apoio ao cliente:

O admin consegue responder a questões feitas pelos clientes através de um chat.

5. Comparação de dados de todos os veículos de todos os clientes:

O admin pode comparar os dados de todos veículos que estão no Sistema por motivos estatísticos.

6. Exportação de dados:

O admin consegue fazer download de um PDF ou de um ficheiro CSV com os dados de cada veículo de qualquer cliente.

## B. Dashboard do Cliente

A interface do Cliente da WebApp permite ao Cliente ter informações em tempo real sobre o seu veículo:

### 1. Mapa Personalizado(ver Fig4):

O Cliente tem acesso a um mapa com a sua localização exata que o permite saber quais os sítios com maior tráfego em tempo real e quais as bombas de combustíveis próximas da sua localização.

### 2. Chat com admin:

O cliente consegue fazer questões que tenha ao admin e obter a resposta em pouco tempo através de um chat.

### 3. Visualização de dados em tempo real:

O cliente tem acesso aos dados dos sensores e às suas estatísticas em tempo real de todos os veículos que estejam registados em seu nome.

### 4. Visualização de eventos:

*O cliente tem acesso aos eventos em tempo real como deteção de colisão e deteção de chuva.*

### 5. Gestão de Veículos:

O cliente consegue adicionar e remover veículos.

### 6. Exportação de dados:

O cliente consegue fazer download de um PDF ou de um ficheiro CSV com os dados de um veículo que esteja em seu nome.

Sistema móvel:

## C. Dashboard do Admin:

A interface administrativa da App Movel permite o controlo e monitorização de diversas funcionalidades críticas do Sistema:

### 1. Visualização de dados dos sensores em tempo real(ver Fig5):

O admin tem acesso aos dados dos sensores de todos os clientes em tempo real e às suas estatísticas.

### 2. Visualização de eventos:

O admin tem acesso aos eventos em tempo real como deteção de colisão e deteção de chuva.

### 3. Gestão de Utilizadores(Ver Fig6):

O admin consegue filtrar a procura por clientes e consegue remover os carros que desejar de qualquer cliente.

## D. Dashboard do Cliente

A interface do Cliente da App Movel permite ao Cliente ter informações em tempo real sobre o seu veículo:

### 1. Visualização de dados em tempo real:

O cliente tem acesso aos dados dos sensores e às suas estatísticas em tempo real de todos os veículos que estejam registados em seu nome.

### 2. Visualização de eventos:

*O cliente tem acesso aos eventos em tempo real como deteção de colisão e deteção de chuva.*

### 3. Gestão de Veículos:

O cliente consegue adicionar e remover veículos.

### 4. Alertas em tempo real:

Quando é detetado um evento como colisão ou fogo o utilizador recebe uma notificação.

## IV. TESTES REALIZADOS E RESULTADOS

Foi conduzida uma série de testes experimentais com o objetivo de validar o correto funcionamento dos sensores utilizados no sistema NEXCAR, bem como a sua integração com a dashboard e atuadores associados. Os testes foram realizados em ambiente controlado e, quando relevante, em condições reais para verificar o comportamento em cenários práticos.

- Sensor de Temperatura: A temperatura foi intencionalmente variada de forma abrupta, permitindo verificar a sensibilidade do sensor a alterações rápidas e assegurar a correta propagação dos dados para a dashboard.
- Sensor de Chuva: Simulações de chuva foram realizadas por meio da adição e remoção de água sobre o sensor. Verificou-se a precisão na deteção da presença de água e a subsequente ativação de um servo motor, utilizado como atuador para replicar, por exemplo, o funcionamento de um sistema de limpeza de para-brisas.
- Acelerómetro: Foram induzidas variações súbitas de aceleração, simulando colisões. O sensor respondeu conforme esperado, identificando os impactos e permitindo avaliar a eficácia do módulo de deteção de acidentes.
- Sensor de Luminosidade (LDR): O sensor foi testado em diferentes cenários de iluminação, desde ambientes fortemente iluminados até locais com baixa luminosidade. Observou-se o comportamento automático do sistema, que ativava um LED sempre que a intensidade luminosa caía abaixo de um valor pré-definido.
- Sensor de Movimento: Embora os detalhes não estejam completamente especificados, assumiu-se que o sensor foi utilizado para detetar presença em movimento. Testes preliminares indicaram uma resposta adequada ao movimento em áreas próximas, contribuindo para funcionalidades de segurança.
- Sensor de Distância: Utilizando tecnologia ultrassónica, este sensor foi testado ao aproximar e afastar obstáculos em frente ao emissor. As medições foram corretamente registadas e enviadas à aplicação, validando a sua utilização em contextos de manobras ou deteção de obstáculos.
- Sensor de Chama (fogo): Este sensor foi submetido a ensaios com uma fonte de chama em ambiente controlado e posteriormente em ambiente externo. Observou-se que, apesar de funcionar corretamente em condições controladas, o sensor apresentou falsos

positivos quando exposto à luz solar direta. Esta limitação deve-se à sua base de funcionamento por radiação infravermelha, o que sugere a necessidade de filtros ou calibração adicional para uso em espaços abertos.

Com o objetivo de validar a fiabilidade e a escalabilidade da comunicação entre sensores e o backend, foram realizados testes com o protocolo MQTT utilizando o broker Mosquitto. Os testes consistiram em simular diferentes quantidades de publicadores (sensores físicos e simulados) e subscritores (API e dashboards) em cenários de uso simultâneo.

- Latência média de entrega das mensagens (publicação → receção): cerca de 50 ms, mesmo com múltiplos sensores ativos.
- Taxa de mensagens perdidas: inferior a 1% para até 20 publicadores simultâneos com uma frequência de publicação de 1 mensagem por segundo.
- Estabilidade da API com mensagens recebidas em tempo real via subscrição, utilizando o módulo `subscriber.py` em execução contínua.

Foram realizados testes com múltiplos utilizadores autenticados a acederem simultaneamente à aplicação web e à API REST.

Os principais resultados observados foram:

- O sistema suportou até 30 clientes simultâneos sem degradação significativa no tempo de resposta (< 200 ms).
- O socket WebSocket (Socket.IO) mostrou bom desempenho em ambientes com múltiplas conexões, permitindo atualizações em tempo real da dashboard sem conflitos de sessão.

## V. CONCLUSÃO

O desenvolvimento do sistema NEXCAR permitiu consolidar um conjunto alargado de competências técnicas no domínio da Internet das Coisas, comunicação em tempo real e desenvolvimento de interfaces orientadas ao utilizador. A arquitetura modular e escalável demonstrou-se eficaz tanto em cenários simulados como em testes reais, com uma latência reduzida, elevada fiabilidade na comunicação MQTT e integração sólida entre sensores, atuadores e aplicações. As interfaces desenvolvidas — web e móvel — garantem uma experiência completa para administradores e utilizadores finais, reforçando a utilidade prática do sistema em contextos de monitorização veicular inteligente. Os resultados obtidos validam a robustez e aplicabilidade da solução, constituindo uma base sólida para futuras iterações e extensões do projeto em ambientes reais de mobilidade urbana.

## ACKNOWLEDGMENT

Os autores gostariam de agradecer aos docentes da Escola de Engenharia da Universidade do Minho pelo apoio e orientação prestados ao longo do desenvolvimento deste projeto. Um agradecimento especial é dirigido aos professores da unidade curricular pelo seu contributo, sugestões construtivas e acompanhamento contínuo durante todas as fases do trabalho.

## REFERENCES

- [1] S. U. Khan, N. Alam, S. U. Jan, and I. S. Koo, "IoT-Enabled Vehicle Speed Monitoring System," *\*Electronics\**, vol. 11, no. 4, Art. no. 614, 2022, doi: 10.3390/electronics11040614.
- [2] S. K. Panigrahy and H. Emany, "A Survey and Tutorial on Network Optimization for Intelligent Transport System Using the Internet of Vehicles," *\*Sensors\**, vol. 23, no. 1, Art. no. 555, Jan. 2023, doi: 10.3390/s23010555.

Anexos:



Figura 3- Dados Em Tempo Real

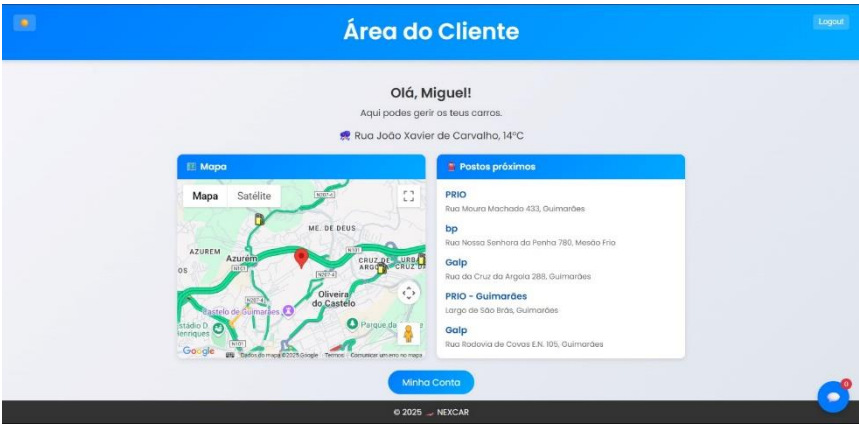


Figura 4- Mapa Personalizado

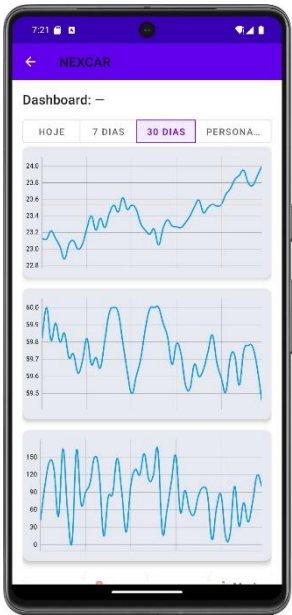


Figura 5-Dados em tempo real



Figura 6- Gestão de Utilizadores