

Sistema de Rastreamento Veicular com IoT, Node.js e React: Uma Abordagem Open-Source

Everton Estevão Ferreira da Silva

everton.efdsilva@gmail.com

Abstract. *This article presents the development of a complete vehicle tracking system using Internet of Things (IoT) devices, the open-source platform Traccar, Node.js backend, and React/Next.js frontend. The system demonstrates how physical devices communicate with a digital platform, providing real-time tracking, status monitoring, and data visualization. By leveraging open-source tools, the project achieves flexibility, transparency, and scalability, allowing integration with municipal services, corporate fleets, or custom IoT applications. Key challenges addressed include device connectivity, real-time updates, and data organization, illustrating practical solutions for IoT system development.*

Resumo. *Este artigo descreve o desenvolvimento de um sistema completo de rastreamento veicular, integrando dispositivos de Internet das Coisas (IoT), a plataforma open-source Traccar, um backend em Node.js e um frontend em React/Next.js. O sistema permite o monitoramento em tempo real de veículos, exibição de status e visualização de dados de forma intuitiva. O uso de ferramentas open-source proporciona flexibilidade, transparência e escalabilidade, possibilitando integrações com serviços municipais, frotas corporativas ou aplicações IoT personalizadas. Durante o desenvolvimento, desafios como conectividade dos dispositivos, atualização em tempo real e organização dos dados foram superados, oferecendo insights valiosos para projetos IoT práticos.*

1. Introdução

O rastreamento veicular é uma das aplicações mais claras da Internet das Coisas (IoT), conectando dispositivos físicos a sistemas digitais capazes de processar e exibir informações em tempo real.

Neste projeto pessoal, eu explorei como integrar rastreadores GPS com a plataforma Traccar, processar os dados em um backend Node.js e apresentá-los de forma visual no frontend React/Next.js. A ideia não era construir um sistema completo, mas sim entender o fluxo de comunicação entre IoT e software, testando a integração, a confiabilidade dos dados e a visualização em tempo real.

2. Traccar: o coração do rastreamento

O Traccar é uma plataforma open-source que recebe os sinais enviados pelos rastreadores e os organiza de forma compreensível. Ele oferece duas formas de interação: uma interface web para monitoramento direto e APIs/Webhooks para integração com sistemas próprios.

Escolhi trabalhar com o Traccar porque ele é flexível e gratuito, permitindo que eu teste diferentes cenários e integre com meu backend sem limitações de licenciamento. Ele

foi a ponte que transformou os sinais brutos do rastreador em dados úteis, que pude manipular e exibir no frontend.

3. O rastreador: conectando o mundo físico ao digital

O rastreador é o dispositivo que envia informações do mundo físico para o sistema digital. Ele capta dados como posição geográfica, velocidade e, dependendo do modelo, outras informações como bateria ou estado do veículo.

- Para funcionar corretamente, ele precisa de:
- **Chip de dados (SIM card):** o rastreador usa a rede celular para enviar os dados ao servidor Traccar. Sem chip, ele não consegue se comunicar.
- **Configuração inicial:** cada rastreador deve ser configurado com o endereço do servidor Traccar e, em alguns casos, porta e protocolo corretos para envio de dados.
- **Testes de conectividade:** é importante validar que o rastreador está online e enviando dados corretamente, porque qualquer falha aqui impacta toda a integração IoT.
- No meu projeto, utilizei o **modelo GT06**, um rastreador GPS bastante comum em projetos de IoT. Ele se conecta à internet via chip e envia dados diretamente para o Traccar. Essa etapa é crítica, porque **o backend e o frontend só recebem informações que foram corretamente transmitidas pelo rastreador.**

3.1. Configuração do Traccar Demo

Para testar o projeto rapidamente, usei o **Traccar Demo**, que é uma versão online da plataforma. Durante a configuração, executei alguns comandos e ajustes básicos para conectar meu rastreador GT06 ao servidor:

- **Definir o endereço do servidor:** configurei o rastreador com a URL do Traccar Demo.
- **Escolher porta e protocolo corretos:** cada modelo de rastreador exige protocolo específico para enviar dados.
- **Verificar conexão:** usei os comandos internos do Traccar para confirmar que o dispositivo estava online e enviando informações.

Esses passos me permitiram validar rapidamente que **o rastreador estava enviando dados corretamente**, antes de integrar com meu backend Node.js e frontend React.

4. Integração IoT: do rastreador ao sistema

A essência da IoT neste projeto é a comunicação entre o mundo físico e digital. O fluxo que experimentei foi:

1. O rastreador GPS envia posições e status.

2. O Traccar recebe essas informações e mantém um registro estruturado.
3. O Node.js processa os dados, organiza e fornece ao frontend.
4. O frontend React exibe os veículos em tempo real, mostrando posição e status.

Mesmo não sendo um sistema completo, pude validar o funcionamento de toda essa cadeia, o que é o ponto central da IoT: coletar dados do mundo físico e transformá-los em informação útil.

5. Backend Node.js: conectando os pontos

No backend, minha função foi:

- Receber dados do Traccar via API ou webhook;
- Filtrar e organizar informações relevantes, como posição, status e horários;
- Enviar os dados para o frontend de forma instantânea, utilizando WebSocket para garantir que o mapa e os veículos fossem atualizados em tempo real;
- Disponibilizar também os dados de forma estruturada para consultas históricas ou filtragens futuras.

Trabalhar com Node.js me permitiu experimentar como processar dados de IoT de forma confiável e em tempo real, criando uma base sólida para futuras expansões sem depender de serviços pagos.

6. Frontend React: visualizando o rastreamento

No frontend, meu objetivo foi transformar dados em informação visual. Desenvolvi:

- Um mapa interativo mostrando veículos e seu status;
- Badges e cores para diferenciar veículos ativos, inativos ou offline;
- Filtros simples para mostrar apenas veículos relevantes em cada contexto.

Essa camada é onde a IoT se torna tangível: o que antes era um sinal de rádio ou pacote de dados se transforma em informação que pode ser interpretada rapidamente.

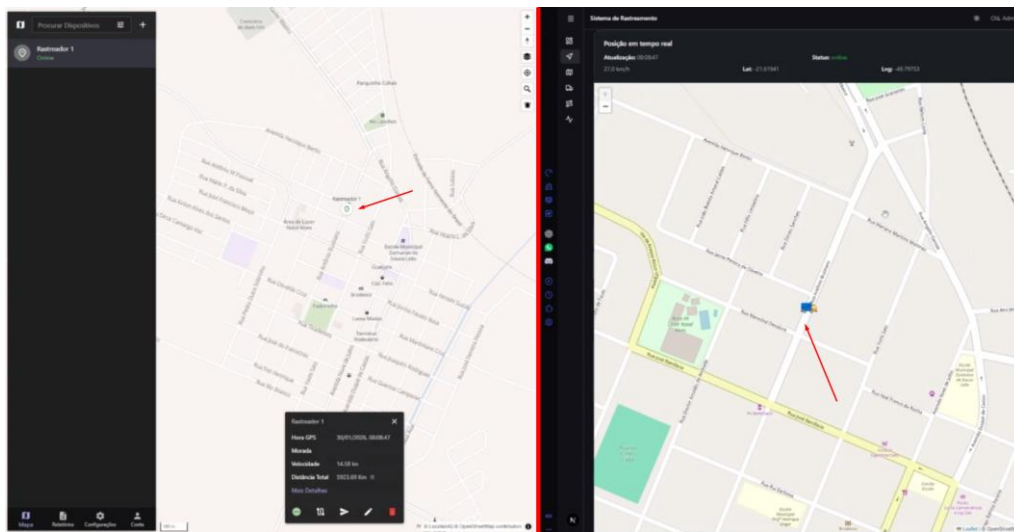


Figura 1 Comparação entre o rastreador GT06 no Traccar Demo (à esquerda), mostrando o dispositivo ativo e os dados recebidos, e a visualização no frontend com Leaflet (à direita), exibindo o veículo em tempo real via WebSocket.

7. Desafios enfrentados

Mesmo trabalhando com um único rastreador, encontrei desafios típicos de projetos IoT e integração:

- O rastreador nem sempre aparecia online, exigindo ajustes na configuração do Traccar e atenção à conexão de rede;
- Diferenças de horário entre o servidor e o dispositivo precisaram ser corrigidas para que os dados fossem exibidos corretamente;
- Garantir atualização em tempo real no frontend exigiu decidir entre consultas periódicas (polling) ou comunicação via WebSocket, optando por esta última para maior fluidez;
- Ajustar a visualização dos dados e do mapa para que as informações ficassem claras e intuitivas, mesmo com dados em constante atualização.

Cada desafio foi um aprendizado sobre como conectar hardware e software de forma prática, mesmo em um projeto experimental, mostrando como IoT pode ser explorada mesmo em protótipos.

8. Conclusão

Neste projeto, eu explorei a integração entre dispositivos IoT, Traccar open-source, Node.js e React, construindo um sistema funcional de rastreamento veicular em tempo real.

Mesmo sendo um protótipo, ele permitiu compreender como dados físicos são transformados em informações digitais, reforçando o valor do open-source e mostrando que é possível criar soluções práticas de IoT com ferramentas acessíveis e flexíveis.

9. Referências

Traccar Open-Source GPS Tracking Platform

- <https://www.traccar.org>
Documentação oficial e guia do Traccar, incluindo APIs, web interface e configuração de rastreadores.

Node.js – JavaScript Runtime

- <https://nodejs.org>
Site oficial do Node.js, com guias de instalação e documentação sobre servidores e manipulação de dados.