PROTÓTIPO DE SISTEMA DE RASTREAMENTO PARA VEÍCULOS DE TRANSPORTE DE CARGAS: uma abordagem com ferramentas e protocolos de código aberto.

José Henrique Lemos Xavier[1]
Graduando do Curso de Engenharia de Computação
Universidade Federal do Pará – Campus Castanhal
henriquecastanhal@gmail.com

Resumo: Rastreamento de veículos não é algo novo, porém, com o advento de novas tecnologias no mundo e a ascensão dos softwares de código aberto tornaram esse mundo, que antes era uma "caixa-preta" apenas decifrável pelos próprios fabricantes em algo possível de ser utilizado e operado por usuários sem grande capacidade financeira, fazendo assim com que esses sistemas possam ser adotados por empresas de pequeno e médio porte de maneira satisfatória. Utilizamos a plataforma *Thingsboard* como base para a criação de um sistema de rastreamento completamente independente de um dispositivo específico atrelado ao seu funcionamento, permitindo ao usuário a escolha do dispositivo que melhor atender suas necessidades. Através de um código escrito em Python, responsável por simular o comportamento de um dispositivo de rastreamento, foi possível atestar o funcionamento do sistema, atendendo a todos os requisitos elencados neste trabalho.

Abstract: Vehicle tracking isn't something new, whatever, with the outcome of news technologies in the world and the ascension of open-source software community turned these solutions, before a "black-box" only know by each manufacturer into something reachable to be used and operated by small and medium business operations. We utilize *Thingsboard* as foundation for the creating a tracking system completely independent of a specific hardware, allowing the user to choose the one which will fill better their needs. To prove the system functionalities, a tracking device simulator was written in Python, the result was satisfactory and all the requirements for the system has been met.

Palavras-chave: Rastreamento, veículos, GPS, Thingsboard, open-source, código aberto;

1. Introdução

Alguns dados do IPEA (2011) [1] mostram que as cidades sofreram transformações profundas nos seus sistemas de mobilidade urbana a partir de 1950, tendo como principais motivos o intenso processo de urbanização e aumento da quantidade de veículos circulando, sejam eles ônibus ou automóveis.

Este mesmo estudo ainda aponta que nos anos de 1977 a 2015, houve uma diminuição da quantidade de pessoas que utilizam transporte público (68% para 51%), em contrapartida, as pessoas que utilizam carros para locomoção diária, subiram de 32% para 49%.

Um dos setores chave da economia brasileira que vem aplicando TI em seu negócio é o transporte rodoviário de cargas, visando uma melhoria no abastecimento dos estados brasileiros e de todo o Mercosul, que representa importantes parceiros comerciais brasileiros. As

Classificado como Público

ferramentas de TI permitem que tenhamos prazos reduzidos de entrega, economia de combustível e segurança no transporte.

Um sistema de rastreamento traz consigo uma maior confiança para o usuário do serviço de transporte, eliminando dúvidas que ele possa ter no processo, permitindo-o a visualização em tempo real da localização de seus produtos que estão sendo entregues. O *Global Positioning System* (GPS) pode ser utilizado para este fim, trazendo com precisão a localização atual de um objeto.

Este trabalho propõe a criação de um sistema de rastreamento com toda a parte de software utilizando-se de ferramentas e protocolos de código aberto utilizando uma plataforma gratuita de Internet das Coisas associado ao protocolo HTTP, altamente difundido, tornando a implementação deste sistema o mais barato possível e permitindo a sua adoção por empresas de pequeno, médio e grande portes garantindo a elas a entrada no mundo tecnológico e uma melhor qualidade nos serviços fornecidos para seus clientes.

2. Trabalhos Relacionados

Nesta seção, alguns trabalhos relacionados ao rastreamento de veículos e cargas serão apresentados e terão seus resultados discutidos.

Assis(2010)[2] propôs a criação de um Sistema Web integrado a um módulo específico de rastreamento. O trabalho consiste no envio das informações de geoposicionamento recebidas no módulo para um servidor WEB utilizando-se de rede GSM, um script fica responsável por fazer o tratamento dos dados recebidos, eles são então armazenados no banco de dados do sistema principal, uma vez armazenado, o sistema principal, aqui denominado iGeoRastreador, interpreta os dados recebidos e os exibe para o usuário. Abaixo temos uma figura mostrando a arquitetura proposta no trabalho.

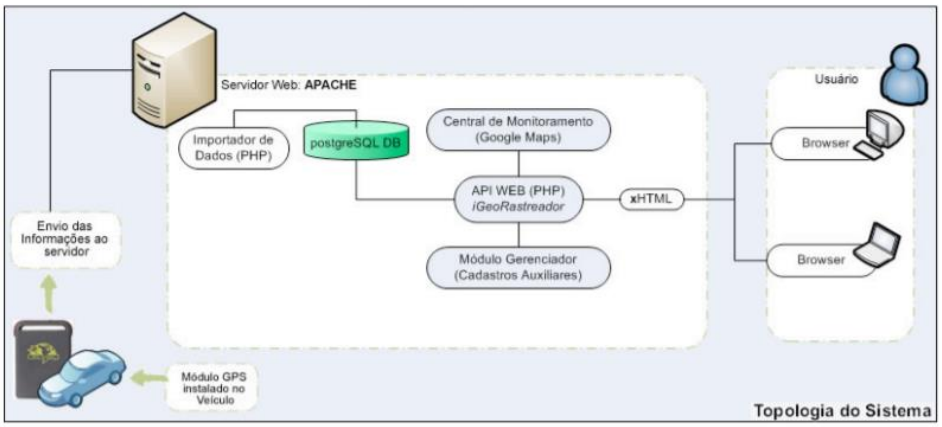

Figura 1: Arquitetura iGeoRastreador, fonte: Assis (2010)

Ele conclui com sucesso sua proposta. A principal desvantagem deste modelo de abordagem se deve ao fato de o sistema ficar preso em um único hardware (Módulo GPS Xexun TK 102), permitindo que o sistema fique obsoleto rapidamente, visto todos os avanços que temos diariamente nos meios tecnológicos e a redução dos preços de produtos eletrônicos, a

Classificado como Público

possibilidade de atualização de hardware também deve ser considerada para a criação do sistema.

Já Galon (2014) [3] foi além de Assis (2010) e propôs a criação de um sistema que agora utiliza mensagens SMS tanto para notificações enviadas do veículo, quanto para ações que podem ser tomadas pelo proprietário diretamente ao veículo. Isso tudo é feito utilizando-se de hardware complexo embarcado no veículo, composto de um módulo GSM para a comunicação via rede celular, um módulo GPS para obtenção das informações de geolocalização, um microcontrolador para fazer a gestão das mensagens enviadas pelos módulos e ativar as funções do veículo, como alarme, vidros e travas elétricas.

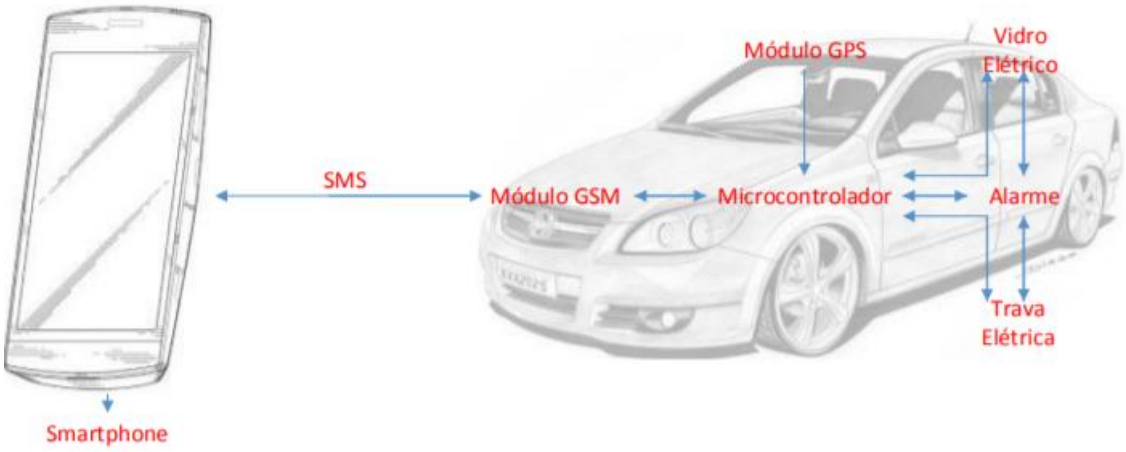

Figura 2: Arquitetura proposta pelo autor no trabalho, fonte: Galon (2014)

Diferente do trabalho anterior, a presença do item centralizador de informações e atuador(microcontrolador) muitas vezes pode permitir a integração de outros módulos, um módulo de comunicação LTE, por exemplo, pode ser utilizado no lugar do módulo GSM, em caso de não haver cobertura para redes 2G(GSM) na localidade, ou até um módulo GNSS que utilize também outros satélites.

Por outro lado, a não visualização das informações de localização de certa forma acabam tornando o sistema mais complexo para o usuário, uma vez que as coordenadas de localização são recebidas via mensagem de texto (SMS), não temos um trabalho de visualização dessa informação para o usuário final, tornando-as difíceis de serem interpretadas pelo usuário final.

3. Desenvolvimento

Trabalharemos neste capítulo as etapas necessárias para o desenvolvimento do sistema proposto, explorando seus requisitos de desenvolvimento, arquitetura do sistema e método de rastreamento utilizado para rastrear os veículos.

I. Requisitos para o desenvolvimento do sistema

O processo de desenvolvimento de uma aplicação completa, responsiva e adaptável para diferentes resoluções de telas pode ser bastante complicado, contudo, existem ferramentas que

Classificado como Público

nos auxiliaram no desenvolvimento, tornando as etapas complicadas, simples de serem resolvidas, vamos a eles:

a. Requisitos de software: Os requisitos primários para desenvolver este sistema são:
    i. Thingsboard [6] Open-source IoT Platform: Será o principal componente na criação do sistema, será responsável por armazenar os dados, gerar ferramentas de visualização dos dados (mapas, gráficos, etc.) e integração com diferentes tipos de protocolos de comunicação, pois possui uma biblioteca de REST APIs para integrações HTTP e conectores para outros protocolos, como MQTT e AMQP, por exemplo.
    ii. Linguagens de programação: Para a integração dos dispositivos de rastreamento com a plataforma este precisa ter uma linguagem de programação embarcada, permitindo a criação de scripts que formatem as mensagens e comunique-as a plataforma, para este trabalho utilizaremos um simulador de hardware escrito em Python [7] para envio das posições, simulando o comportamento de um veículo rastreado na plataforma
    iii. JSON: As mensagens recebidas pela plataforma seguem o padrão JSON, definido na ECMA-404 The JSON Data Interchange Standard [4], é um protocolo que define o formato para troca de arquivos entre diferentes sistemas, permitindo a integração de diferentes sistemas, utilizaremos o JSON para permitir a intercomunicação entre o rastreador e a plataforma Thingsboard.
    iv. Internet das coisas: Quando falamos de rastreamento de objetos e visualização dessas informações de maneira humana compreensível, estamos falando de internet das coisas. Graças aos grandes avanços tecnológicos que tivemos nos últimos anos, a internet das cosias está abrindo as portas para novas aplicações e sistemas que prometem melhorar a qualidade de vida dos seus usuários, mudando a forma de interação na internet, que passa de um sistema apenas human-human, para thing-human, human-human e thing-thing [5] (Tan et. All, 2010).

    Neste contexto, temos os serviços baseados em localização, os quais oferecem serviços para seus usuários como a localização de um

Classificado como Público

dispositivo, o clima em uma determinada região, vendas de produtos regionais, serviços de transporte, como o Uber por exemplo, são alguns entre uma diversificada gama de serviços que estas tecnologias podem prover.

b. Requisitos de Hardware: Este trabalho não determina um hardware específico para utilização, define apenas capacidades mínimas para que um dispositivo possa ser adotado para essa solução, são elas:

   i. Capacidade de formatação da mensagem: A mensagem enviada precisa seguir o padrão JSON e conter, no mínimo, a estrutura abaixo:

```
{
    "latitude": DOUBLE,
    "longitude": DOUBLE,
    "velocidade": FLOAT
}
```

   ii. Protocolo HTTP: O hardware escolhido precisará ser capaz de utilizar o protocolo HTTP para envio das informações de telemetria geradas pelos objetos rastreados.

II. Arquitetura do Sistema

Nesta etapa faremos um aparato sobre as possibilidades de integração da plataforma. Primeiramente precisamos de um módulo rastreador capaz de realizar comunicação HTTP para poder enviar seus dados de posicionamento para a plataforma.

Na figura 3, observamos a arquitetura do sistema utilizando o módulo rastreador. Este, por sua vez, precisará ser capaz de formatar sua mensagem, de maneira a comportar o formato requerido pelo o sistema aqui proposto no tópico de requerimentos de hardware, abaixo temos como ficaria arquitetura:

Classificado como Público

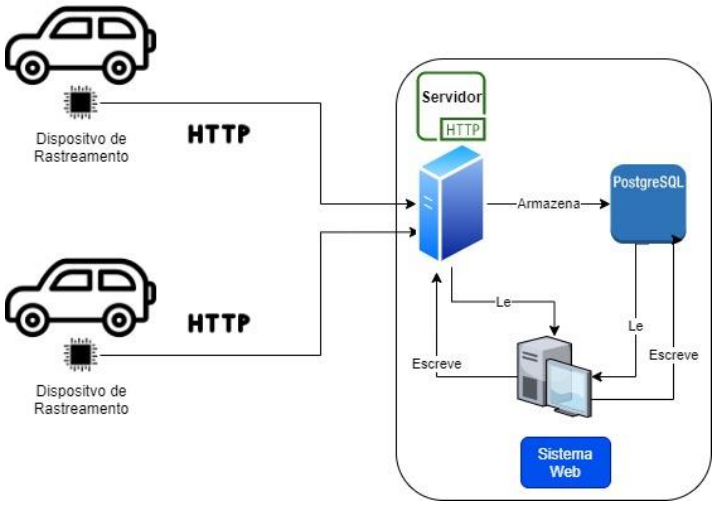

Figura 3: Arquitetura do sistema proposto, fonte: O Autor(2021)

III. Requisitos de segurança

Nos últimos anos, a segurança de sistemas de software vem trazendo muita atenção com a intenção de cessar vulnerabilidades que existem e podem ser exploradas se não tivermos cuidado no processo de desenvolvimento de um software. Para garantir a segurança deste trabalho, seguiremos alguns dos *Princípios de Saltzer & Schroeder's* (1975), sendo eles:

- Economia de mecanismos: Mantendo o design mais simples possível para utilização;
- Métodos padrão anti-falhas: Negar todo acesso a não ser aqueles aprovados;
- Mediação completa: Acesso a todos objetos deve ter sua autoridade checada;
- Design aberto: Ser o mais claro possível, não depender de sub-entendimentos;
- Separação de privilégio: Duas chaves são melhores que uma;
- Privilégios mínimos: Usuários e Software operam com o privilégio mínimo necessário para seu funcionamento;
- Minima de mecanismos comuns: Usuários de diferentes níveis de acesso devem ter o mínimo possível de mecanismos comuns entre eles, tornando-os específicos para aquele nicho de usuários;

IV. Método utilizado para o rastreamento dos veículos

O método de rastreamento depende primariamente do GPS, que por sua vez é um sistema de navegação baseado em satélites que prove um sistema gratuito de posicionamento em qualquer lugar no mundo.

Classificado como Público

```
processo Envia posição;
    enquanto executando faça;
        recupera posição dos satélites;
        se posição é retornada então;
            Formata a mensagem em objeto JSON;
            Cria uma mensagem HTTP POST;
            Envia mensagem HTTP POST;
        se não
            Espera intervalo de tempo e continua;
        Fim se
    Fim enquanto
Fim processo
```

Figura: Processo de envio de posições pelo módulo rastreador, fonte: o Autor(2021).

As informações geradas pelos módulos de rastreamento devem ser enviadas para a plataforma e esta se encarregará do armazenamento e visualização desses dados. O painel de controle criado na plataforma Thingsboard proverá a localização em tempo real do veículo em um mapa, utilizando-se das APIs do Open Street Map, um serviço gratuito que provê funcionalidades de mapas, similar ao Google Maps, mas totalmente gratuíto e de código aberto. Podemos observar o fluxo completo na figura abaixo:

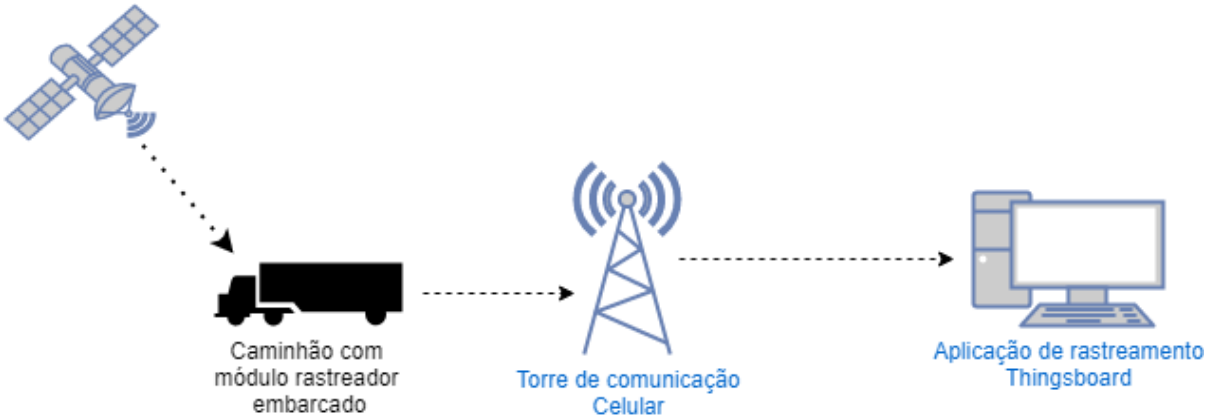

Figura 4: Fluxo completo de funcionamento do sistema, fonte: o autor(2021)

V. Levantamento de Requisitos

Segundo a norma IEEE-90, os requisitos de software são definidos como:

- Uma capacidade que um usuário necessita para resolver um problema ou atingir um objetivo;

- Uma capacidade que deve ser atendida ou possuída por um sistema ou componente de um sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto;
- O conjunto de todos os requisitos que formam a base para o desenvolvimento subseqüente de um software ou componentes de um software;

Seguindo estas premissas, os requisitos foram levantados com base em observação obtidas durante minha experiencia profissional e em discussões sobre sistemas de rastreamentos com diversos fornecedores do Brasil e do exterior. Seguindo o método de brainstorm[8], foram definindo os seguintes requisitos para o sistema proposto:

RF01 – O software precisa ter capacidade validar campo de e-mail no login;

RF02 – O software precisa ter capacidade validar senha do usuário indicado no e-mail;

RF03 – O software precisa ter capacidade exibir mapa com posição atual dos veículos;

RF04 – O software precisa ter capacidade exibir velocidade atual dos veículos registrados;

RF05 – O software precisa ter capacidade exibir histórico de posições dos dispositivos rastreados em um determinado espaço de tempo;

RF06 – O software precisa ter capacidade validar as permissões dos usuários em cada operação feita no sistema;

RF07 – O software precisa ter capacidade para cadastro de novos dispositivos;

RF08 – O software precisa ter capacidade para remoção de dispositivos;

RF09 – O sistema precisa possuir um painel de controle para gerenciamento dos veículos/dispositivos rastreados;

RNF01 – O sistema deverá rodar em qualquer navegador;

RNF02 – Deverá ser utilizado a plataforma Thingsboard como provedor de Gateway para recebimento das informações geradas;

RNF03 – O sistema deverá receber apenas mensagens no formato JSON;

RNF04 – O sistema deverá armazenar os dados recebidos em um banco de dados;

RNF05 – O sistema deverá ser capaz de disponibilizar os dados armazenados para consumo por aplicações terceiras através de interfaces de programação de aplicações(APIs)

VI. Construção do software

a. A organização do banco de dados

A plataforma utilizada para o desenvolvimento do sistema provê duas maneiras de armazenar os dados já integradas nos gateways de recebimento de dados da plataforma, são eles o

Classificado como Público

Postgresql e o *Apache Cassandra*. Qualquer um dos dois podem ser utilizados, a recomendação do fabricante da plataforma é que utilizemos o *Apache Cassandra* apenas quando o sistema construido possuir um nível de mensagens maior que 5000 por segundo e o *Postgresql* para sistemas abaixo desse valor. Adotaremos aqui o *Postgresql* como o banco utilizado, as tabelas e views e os elementos de cada uma delas podem ser visualizados na imagem abaixo.

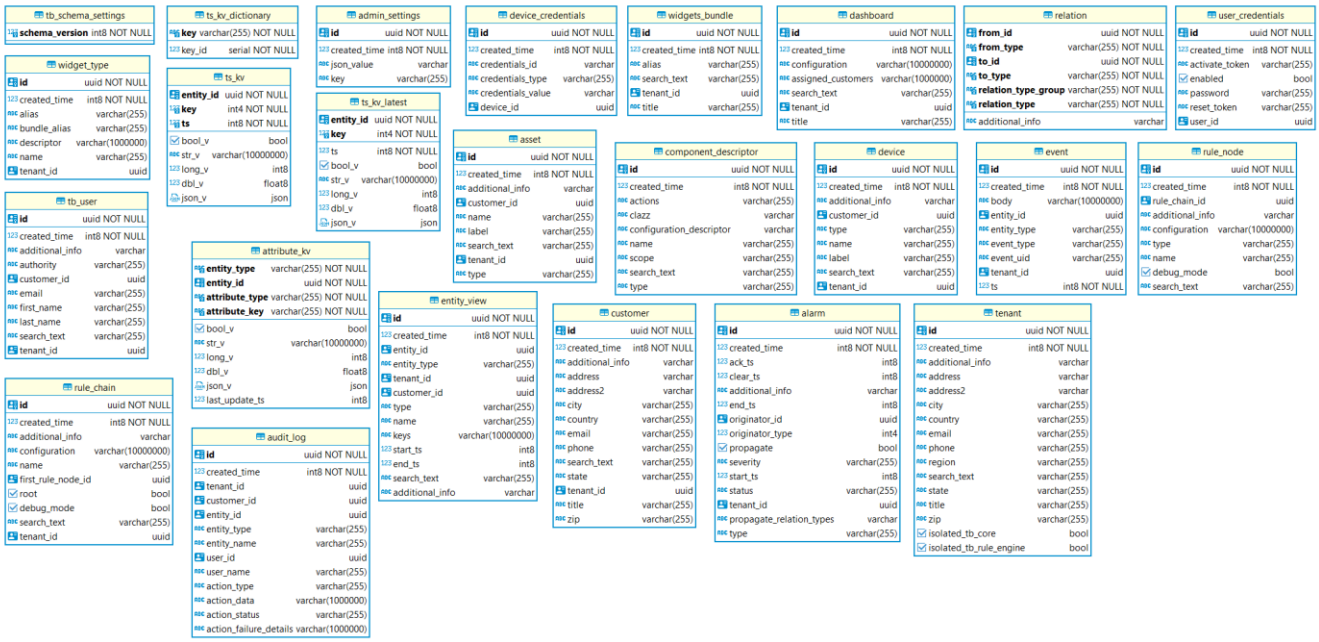

Figura 5: Diagrama ER utilizado pela plataforma para armazenar os dados, fonte: o Autor (2021).

As principais relações são entre os objetos chamados de “Entity”, “Device”, “Asset”, “Customer” e “Tenant”. Um Entity é composta por um ou mais *devices*, e pode ser utilizado para identificar, por exemplo, o modelo de dispositivo usado e o formato de mensagem esperado por esses dispositivos. Um Asset nada mais é que uma localização para o dispositivo ou entidade, por exemplo, podemos ter dispositivos agrupados em diferentes entidades localizados na mesma cidade, a cidade poderia ser considerada como um Asset para esses dispositivos, permitindo agrupa-los desta forma para facilitar a criação de visualizações para esses dados. O Customer e o Tenant são os detentores de todos os outros objetos da base.
O Tenant é o administrador do sistema, capaz de criar novas contas de usuário para operação da plataforma. Os dispositivos cadastrados podem estar em dois escopos de permissão de visualização, o primeiro é o escopo Público, ou seja, qualquer usuário com acesso ao sistema poderá acessar os dados deste dispositivo, o outro escopo é quando o dispositivo é associado a

Classificado como Público

um Customer específico, desta forma os dados gerados pelo dispositivo só estarão disponíveis para o Customer detentor do mesmo. Os dispositivos não são a única coisa que podem ser associadas com os Customers, qualquer objeto de banco pode ser registrado como pertencente a um Customer e um Customer só, ou seja, uma entidade de dispositivos pode pertencer a um Customer só e só ele será capaz de visualizar os dados dos dispositivos registrados naquela entidade.

A figura abaixo mostra a relação entre as principais entidades citadas acima:

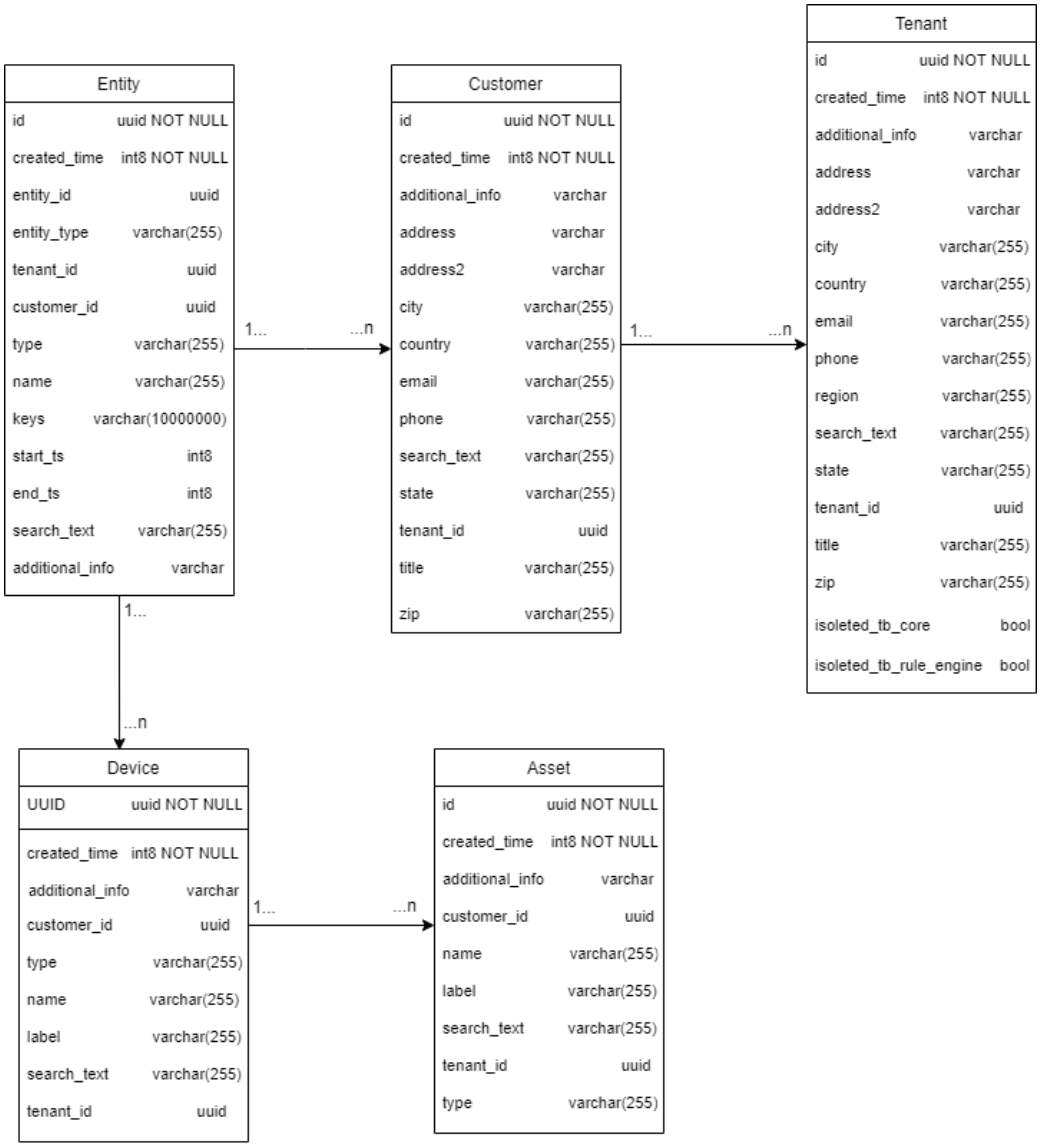

Figura 6: Diagrama de classes das principais entidades do banco, fonte: o Autor(2021).

b. Aplicação Web

O sistema que propomos aqui deverá ser utilizado única e exclusivamente para gerenciamento por parte da empresa e não é um sistema pensado para ser disponibilizado para terceiros acessarem, apenas pessoas habilitadas na empresa devem ter acesso. ~~O diagrama de atividades abaixo exemplifica o processo de acesso a plataforma e aos recursos propostos nos requisitos~~ A plataforma utilizada aqui, *Thingsboard*, funciona através da arquitetura de microserviços utilizando o modelo REST para exposição dos serviços, estes, por sua vez, responsáveis pelas por atender todas as requisições e suas demandas, os diagramas de processos abaixo exemplificam algumas funcionalidades elencadas como requisitos.

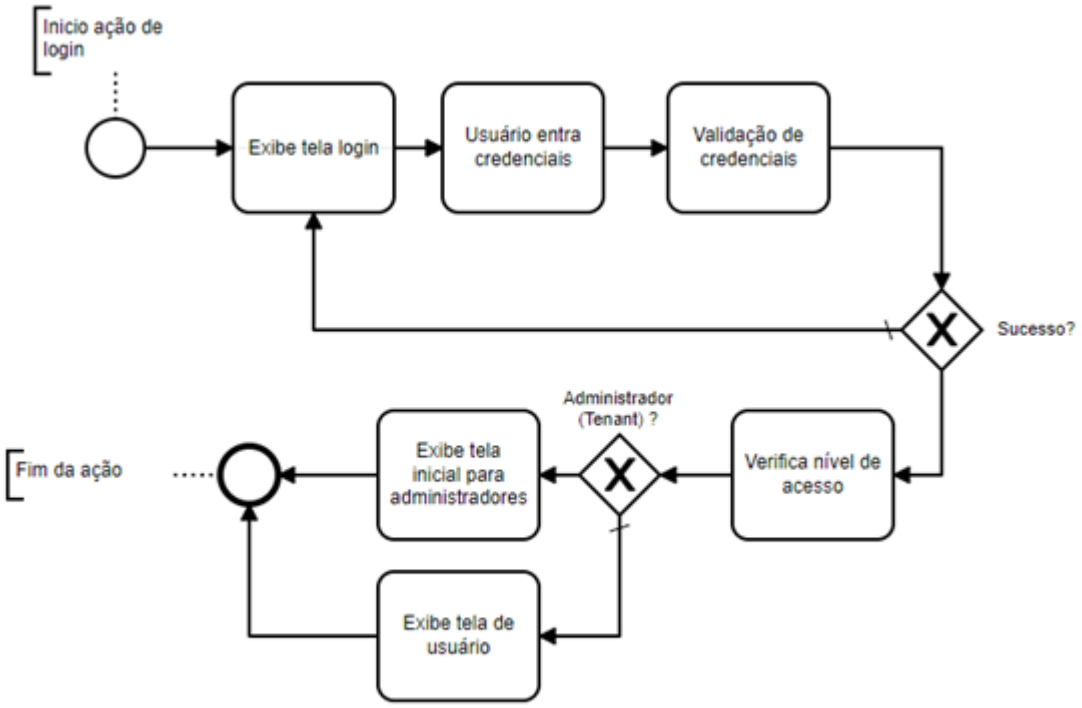

Figura 7: Diagrama de processos de login do sistema proposto, fonte: o Autor(2021).

Para a adição e/ou remoção de novos dispositivos para rastreio, a plataforma Thingsboard oferece um menu onde podemos realizar as operações. Durante o cadastro de um novo dispositivo precisa-se definir quem detém este dispositivo ou se o dispositivo é de propriedade pública, mantendo a integridade e segurança dos dados armazenados na plataforma.

Classificado como Público

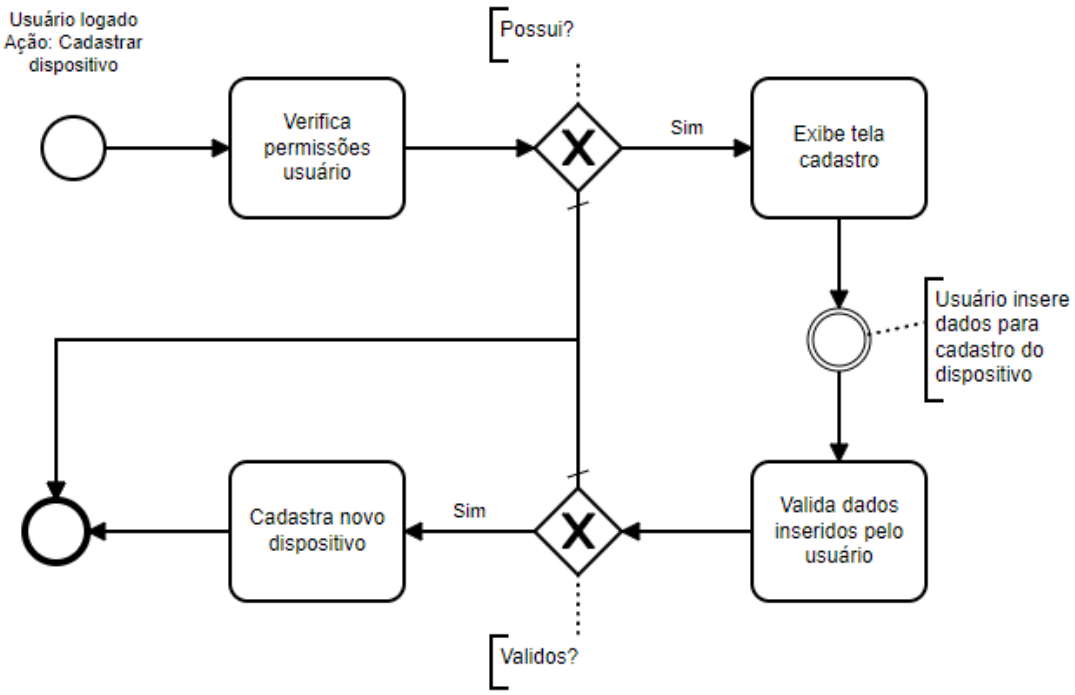

Figura 8: Diagrama de processos para cadastro de novos dispositivos no sistema proposto, fonte: o Autor(2021).

O menu para cadastro e remoção dos dispositivos na plataforma pode ser visto abaixo:

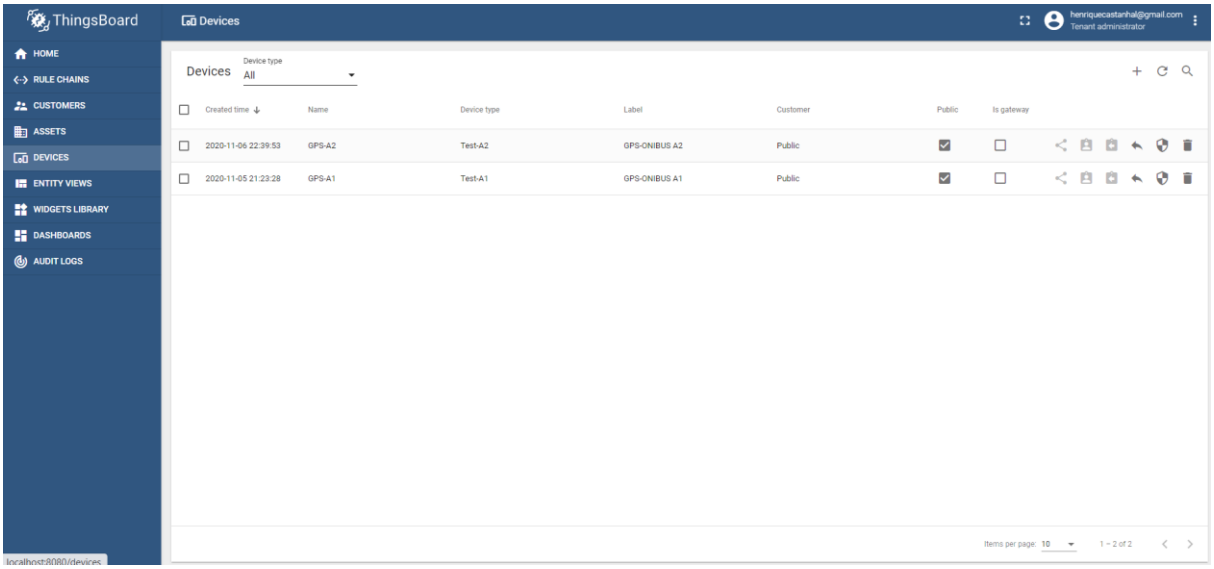

Figura 9: Tela de adição e remoção de dispositivos da plataforma. Fonte: o Autor(2021).

Um novo veículos para ser rastreado nada mais é que um novo dispositivo cadastrado na plataforma, para enviar os dados, é necessário a utilização do protocolo HTTP através da interface de programação de aplicações (API) da plataforma thingsboard. Cada dispositivo tem um identificador único que pode tanto ser colocado pelo usuário ou um aletório gerado pela plataforma. A plataforma também permite a importação e exportação dos seus paines de controle(em inglês, *Dashboards*) e utilizaremos tal recurso para poder difundir o painel de controle criado para a aplicação aqui prosposta.

A plataforma provê toda a documentação necessária para a configuração de novos dispositivos e guias para auxiliar no envio dos dados gerados pelos dispositivos para a plataforma, todas informações sobre a plataforma podem ser verificadas em http://www.thingsboard.io .

Uma vez que tenhamos todos os dispositivos cadastrados na plataforma, precisamos criar a visualização dos dados enviados por eles em um mapa. Toda a configuração para esta etapa também está disponiível no site da Thingsboard, o qual pode ser acessado no endereço informado acima.

4. Análise dos resultados:

Este capítulo descreve a implementação do protótipo, a obtenção e tratamento dos dados e parametros iniciais para alimentar o sistema proposto.

A fim de validar a solução proposta no tópico 3, foi desenvolvido um simulador de dispositivos de rastreamento através da linguagem Python utilizando da IDE PyCharm ® para codificação. Existem dois arquivos de textos contendo posições de GPS no formato a seguir:

$GPGGA,210230,3855.4487,N,09446.0071,W,1,07,1.1,370.5,M,-29.5,M,,*7ª

Para geração do arquivo com coordenadas, foram utilizados dois softwares, o Google Maps® para definição de rota e criação de arquivo do tipo KML(Keyhole Markup Language, uma linguagem baseada em XML para expressar notações geográficas) e o GPS Babel® para conversão do arquivo KML gerado no Google Maps® em um arquivo com coordendas GPS, tal formato é o mesmo gerado por dispotivos GPS que seguem o padrão NMEA-0183, cada campo da mensagem representa uma informação e são descritos na documentação do protocolo NMEA-0183.

O software então interpreta cada uma das linhas do arquivo como uma nova coordanada de GPS recebida do sistema de navegação, ele então converte a o valor das coordenadas

Classificado como Público

recebidas (latitude e longitude) para decimal e envia para o sistema proposto, simulando então, com as mudanças de posição, uma rota de transporte/trafego de um veículo rastreado. Além disso, valores de nível de combustível e velocidade atual também são simulados. O resultado da simulação pode ser observado no mapa abaixo.

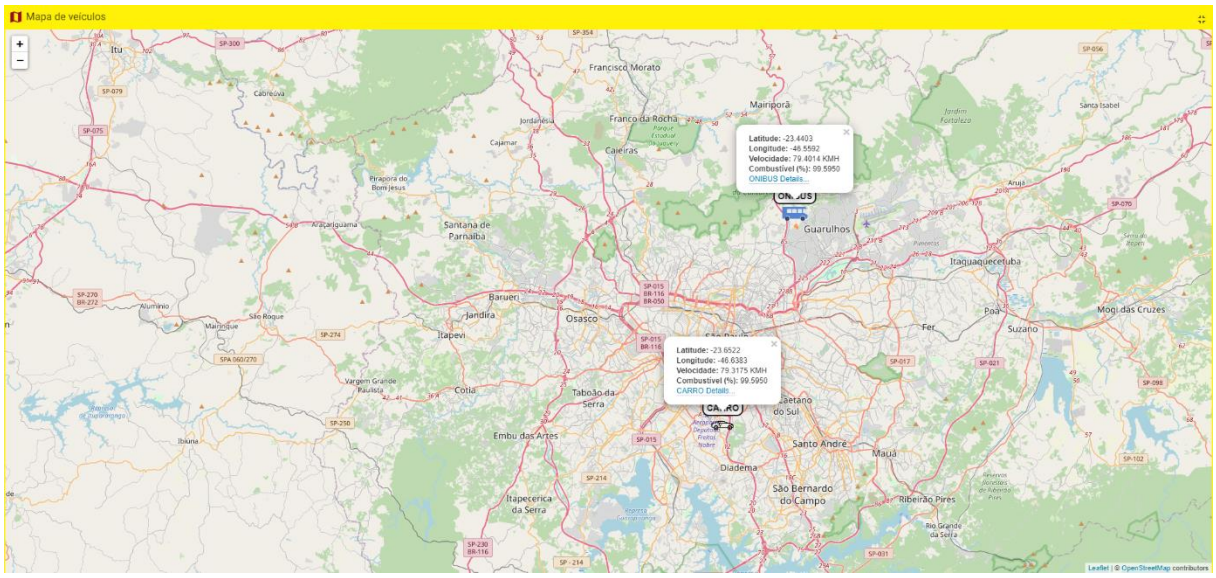

Figura 10: Mapa contendo veículos rastreados, fonte: o Autor(2021).

5. **Conclusões e trabalhos futuros**:

O conceito de rastreamento de veículos já existe a bastante tempo, porém com os avanços recentes das tecnologias de transmissão de dados, o barateamento no custo de produção de dispositivos com capacidade de acessar os satélites de navegação GPS e os avanços na área de código aberto permitem hoje uma adoção por empresas que não possuem tantos recursos para investir em sistemas de alta complexidade.

Além da capacidade de rastreamento, sistemas desse tipo também servem como ferramentas de manutenção preditiva nos veículos rastreados, onde, por exemplo, podemos manter o controle da distancia percorrida por um veículo em um espaço de tempo e identificar se está na hora de fazer a manutenção. Também há a possibilidade de criação de alarmes para situações inesperadas de freagem ou condução perigosa do motorista, como acesso a outra faixa de forma continua ou velocidade acima da permitida no trecho, podendo assim criar sistemas de recompensa para motoristas com melhor desempenho.

O principal diferencial entre o sistema proposto por este trabalho e os aqui analisados no tópico 2 é o fato de que, o sistema que propomos aqui, tem sua implementação mais simples, visto que utiliza-se de uma plataforma gratuita para gestão dos dispositivos conectados a ela. Outro fato é a não necessidade de um hardware específico para o funcionamento do sistema,

Classificado como Público

como citado antes, há outros requisitos para o dispotivo, como conectividade e capacidade de utilizar protocolos específicos e formatar mensagens para o formato *JSON*. Abaixo temos um quadro comparativo entre os trabalhos analisados e o trabalho aqui realizado.

Trabalhos relacionados	**Protocolos**	**Formato de Mensagem**	**Hardware**	**Rastreamento Visual em tempo real**
SISTEMA DE RASTREAMENTO DE VEÍCULOS PARA EMPRESAS DE TRANSPORTE UTILIZANDO NAVEGAÇÃO POR SATÉLITE, Assis (2010).	Transmissão de dados: HTTP/1.1 Recebimento de cordenadas: GPS Conectividade: GPRS, dado que o hardare aqui exige este padrão.	Modelo seguindo padrão próprio de formatação	Estático, necessário hardware específico	Sistema possui interface para visualização da posição dos veículos rastreados. Contudo, devido a arquitetura do sistema, o mesmo tem um tempo longo entre atualização e exibição de posição.
SISTEMA DE RASTREAMENTO E CONTROLE DE RECURSOS DE UM VEÍCULO UTILIZANDO UM SMARTPHONE ANDROID, Galon (2014)	Transmissão de dados: HTTP/1.1 Recebimento de cordenadas: GPS Conectividade: GPRS, uma vez que o hardware utilizado não possui outro tipo de conectividade	Modelo seguindo padrão próprio de formatação	Estático, necessário hardware específico	Sistema não possui interface para visualização em tempo
PROTÓTIPO DE SISTEMA DE RASTREAMENTO PARA VEÍCULOS DE TRANSPORTE DE CARGAS: uma abordagem com ferramentas e protocolos de código aberto.	Transmissão de dados: HTTP/1.1 Recebimento de cordenadas: GPS Conectividade: Agnostico, depende do dispositivo escolhido	Padrão JSON	Flexível, pode ser utilizado qualquer um que atenda os requisitos minimos.	Possui interface de visualização em *near-real-time* dos dados recebidos dos dispositivos pelo sistema

Classificado como Público

Quadro 1: Comparação entre os trabalhos analisados e o proposto, fonte: o autor(2021)

O sistema proposto utiliza o máximo possível de código aberto, a plataforma utilizada provê o armazenamento dos dados através de um banco de dados relacional e, graças as suas bibliotecas de API, possui grande capacidade de integração com diferentes sistemas, ou seja, os dados armazenados e visualizados através da plataforma também podem ser recuperados e exibidos em outras aplicações e sistemas.

Com isso em mente, seria possível criar uma versão para os clientes utilizarem, atráves de um aplicativo móvel disponibilizado para os principais sistemas operacionais atuais, onde seriam capazes de rastrear suas encomendas em tempo real, podendo inclusive ter opções para agendar retirada de objetos a serem transportados ou agendar horário de entrega de cargas já em trânsito.

6. Biliografia

[1] IPEA. Ipea divulgou pesquisa sobre mobilidade urbana. IPEA, 2011. Disponível em: https://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=8588. Acessado em 01/03/2021.

[2] ASSIS, Paulo Ueiner Moreria de. Sistema de rastreamento de veículos para empresas de transporte utilizando navegação por satélite. 2010. 121 páginas. Centro Universitário de Brasília – UniCEUB, Brasília - DF, 2010.

[3] GALON, Handre Emanuel. Sistema de rastreamento e controle de recursos de um veículo utilizando um smartphone Android. 2014. 79 páginas. Universidade Tecnológica do Paraná, Pato Branco - PR, 2014.

[4] DEVMEDIA. O que é JSON?. DEVMEDIA. 2011. Disponível em: https://www.devmedia.com.br/o-que-e-json/23166 . Acessado em: 28/02/2021.

[5] TAN et al. Future Internet: The Internet of Things. 2010. International Conference on Advanced Computer Theory and Engineering (ICACTE). Chengdu, Terceira Edição, páginas 385 a 390, Outubro de 2010.

[6] Thingsboard – Open-source IoT Platform. Introduction to Thingsboard. 2020. Disponível em: https://thingsboard.io/docs/getting-started-guides/helloworld/#introduction . Acessado em 25/02/2021

Classificado como Público

[7] Welcome to Python.org. Getting started with Python for Beginners. 2021. Disponível em: https://wiki.python.org/moin/BeginnersGuide/Programmers. Acessado em 05/03/2021

[8] Siang, Teo Yu. What is Brainstorming. Interaction Design Foundation. 2018. Disponível em: https://www.interaction-design.org/literature/topics/brainstorming#:~:text=Brainstorming%20is%20a%20method%20design,them%20to%20find%20potential%20solutions. Acessado em 05/01/2021