Projeto de Formatura – Turmas 2019

PCS - Departamento de Engenharia de Computação e Sistemas Digitais

Engenharia de Computação

Tema:

Smart Traffic - Plataforma para Tracking de Veículos em Cidades Inteligentes

Proposta

O projeto consiste na implementação de uma API comercializável, com suporte a vídeo e a dados estruturados, responsável por analisar, persistir e disponibilizar a geolocalização mais recente, bem como o percurso diário de um grupo de veículos, o que permite o monitoramento da infraestrutura urbana, conectando os cidadãos, o estado e a iniciativa privada em um único lugar. Ao conectar distintas fontes de dados e interconectar as informações geradas pelos diversos agentes do sistema, introduz-se as características chave presentes no cenário da indústria 4.0: extensibilidade, interoperabilidade e interconexão dos dispositivos e de seus agentes, a fim de se automatizar decisões estratégicas pelas organizações públicas. Este projeto, portanto, focou em duas grandes áreas da computação no que tange cidades inteligentes, big data e IoT. Para a implementação, utilizou-se PostgreSQL, Hadoop, Apache Hive e serverless computing. Adicionalmente, utilizou-se como provedor de infraestrutura a AWS, que proveu o suporte necessário à implementação de uma solução robusta de IoT encima da infraestrutura de câmeras LPR (License Plate Recogntion) já existentes no mercado.

Arquitetura

A especificação da arquitetura do projeto foi feita com base no princípio dos 5V's (volume, velocidade, variedade, variabilidade e valor). Cada um desses fatores é crucial na implantação da arquitetura, o que permitiu a construção de uma arquitetura mínima viável que é extensível e cujo processamento é NRT (Near Real Time).

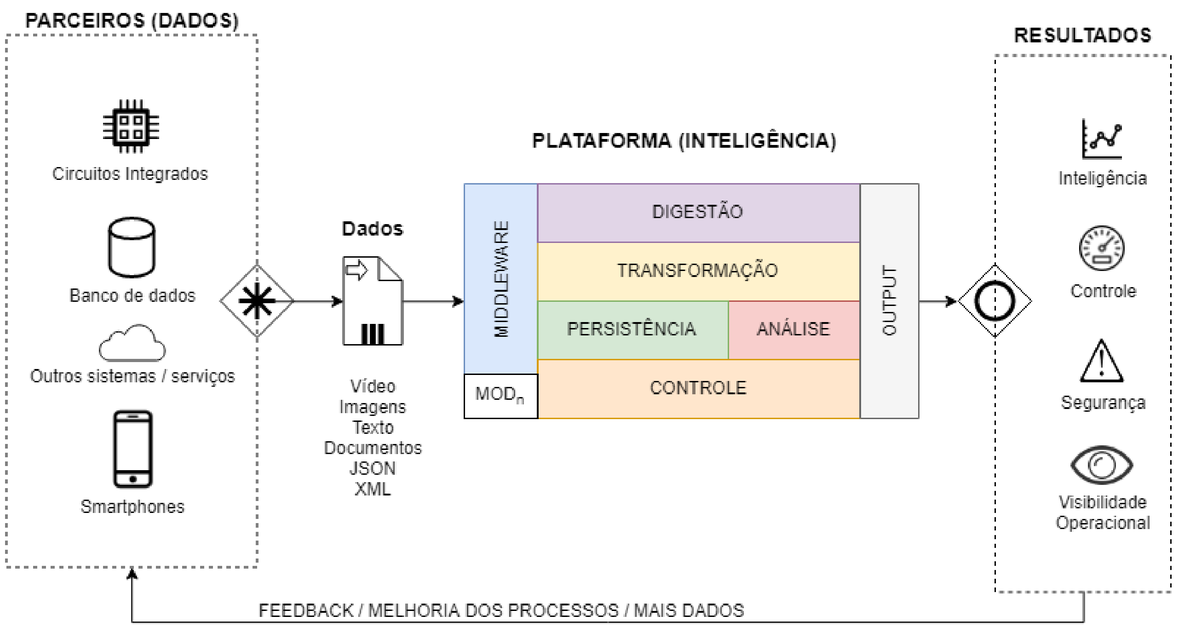

Figura 1. Arquitetura Macroscópica da Plataforma sob a ótica dos 5V's

Integrantes:

Pedro Paulo Azevedo Gryzinsky Grillo
Rafael Trostli Costella

Professor Orientador:

Prof. Dr. Jorge Luis Risco Becerra

Implementação e Resultados

1. Extração (Extract)

Para realizar a extração de dados criou-se uma função pura em NodeJs que executa um dump do banco de dados relacional que contém informações de veículos e de geolocalização do dia anterior e passa a informação via stream para a próxima etapa.

2. Transformação (Transform)

Para realizar a transformação de dados criou-se uma função pura em NodeJs que manipula o stream recebido para traduzir em dois futuros streams, um para salvar dentro de uma estrutura de veículos no Data Warehouse e outro para salvar dentro de uma estrutura de geolocalizações.

3. Carregamento (Load)

Com ambos os streams prontos, criou-se uma outra função pura em NodeJs que recebe ambos os streams, salva em localidades conhecidas para veículos e para geolocalizações e dispara uma etapa do cluster do EMR\cite{EMR}, que utiliza-se do hadoop e um script Hive para reorganizar os dados em tabelas particionadas.

4. IoT

Foram extraídas amostras de câmeras LPR comerciais, que compõe a componente de vídeo, e simulou-se tanto para interfaces USB, quanto para câmeras via IP, através do protocolo RTSP, um stream de vídeo, utilizando-se o framework GStreamer. Para os dispositivos, utilizou-se o protocolo TLS, com autenticação do lado do cliente e certificados X.509 emitidos pela CA, no caso o serviço AWS IoT Core, para a autenticação apenas dos dispositivos autorizados por nós. Adicionalmente, a cada quadro de vídeo adicionou-se a informação de latitude e longitude da fonte.

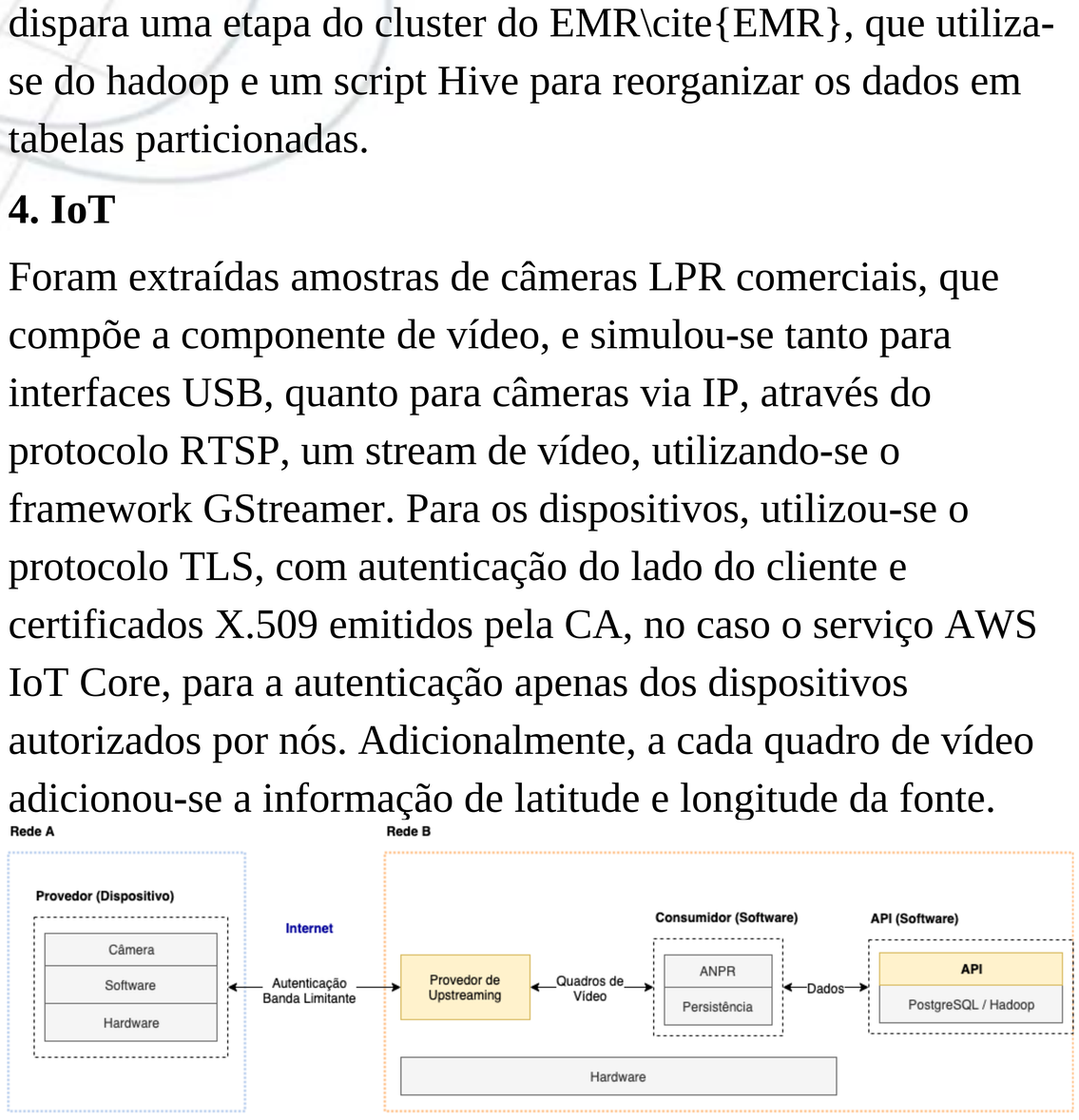

Figura 2. Conexão entre dispositivo IoT local e infraestrutura remota

5. API

Por fim foi desenvolvida uma API que permite a ingestão e o acesso aos dados já estruturados. Para o consumo dessa API, elaborou-se um software web responsável pelo cadastro dos clientes e visualização dos dados em um mapa. Para o cálculo das rotas, utilizou-se a API do Google Maps. Para a inserção dos dados, o seguinte formato foi utilizado.