

O sistema público de transporte de São Paulo é utilizado por milhões de pessoas diariamente, e a eficiência no gerenciamento da frota de ônibus é essencial para garantir a qualidade do serviço. Para otimizar a operação do transporte público, é necessário implementar uma solução que permita o **monitoramento em near real-time** dos ônibus em circulação, além de fornecer **métricas e KPIs** relevantes para a tomada de decisão.

Objetivo: Criar um produto de dados em near real time ou batch que possibilite métricas, KPIs, monitoramento e acompanhamento dos ônibus em circulação no sistema público de transporte da cidade de São Paulo.

Entregáveis:

1. Desenho arquitetural da solução indicando as ferramentas utilizadas e fluxos propostos.
2. Explicação do por que de cada escolha realizada na arquitetura proposta.
3. Catálogo de metadados e documentação da aplicação.
4. Apresentação da solução proposta em perfeito funcionamento, com ingestão, processamento e entrega de dados, bem como o código fonte.
5. Apresentação dos entregáveis.

Especificação:

- Utilizar a API OLHO VIVO da SPTRANS para coletar os dados em near real time (a cada 2 minutos) da posição de todos os ônibus em circulação.
API: <https://www.sptrans.com.br/desenvolvedores>
ENDPOINTS: <https://www.sptrans.com.br/desenvolvedores/api-do-olho-vivo-guia-de-referencia/documentacao-api/>
- Utilizar o GTFS da SPTRANS para dados complementares (Dados Estáticos/Cadastrais).
- Enriquecer os dados de paradas com o endereço da localidade, através de latitude e longitude.
- Dados do GTFS: <https://gtfs.org/documentation/schedule/reference/>
 - trips: Uma viagem é uma sequência de duas ou mais paradas que ocorrem durante um período de tempo específico.
 - stops: Paradas onde os veículos pegam ou deixam passageiros.
 - stop_times: Horários em que um veículo chega e parte das paradas para cada viagem.
 - shapes: Todos os trajetos no mapa. Regras para mapear caminhos de viagem de veículos, às vezes chamadas de alinhamentos de rotas.
 - routes: Uma rota é um grupo de viagens que são exibidas aos passageiros como um único serviço.
- Criar um data warehouse/data lake/data lakehouse com no mínimo 2 camadas de dados, onde os dados de entrada sejam armazenados da raw data e os dados tratados na camada trusted, etc...

- Deve ser aplicado regra de negócio para apresentação no produto de dados.
- Realizar discovery nas bases de dados para entender os possíveis cruzamentos de dados (API e GTFS) e as visualizações ou api que serão entregues como produto final.

OBS: Pode ser utilizado qualquer biblioteca open source ou base pública externa para enriquecimento e cruzamento de dados.