Data Engineering – Applied Project

Gonçalo Nogueira

1 Dashboard Carris em tempo real

O objetivo deste projeto será criar uma pipeline de dados que poderá suportar um dashboard real-time que mostra as posições dos autocarros da Carris, bem como outras informações relevantes.

A construção do dashboard está fora do âmbito do projeto. No entanto, é importante clarificar a estrutura do mesmo, fazendo diagramas e/ou mockups.

2 Fontes de dados

A principal fonte de dados para o projeto será a API da Carris.

A informação do endpoint `/vehicles` deve ser consumida e processada em streaming para permitir a atualização em tempoo real do dashboard. Outros endpoints relevantes devem ser consumidos em batch com uma certa periodicidade. Estes dados devem ser usados para enriquecer o dataset.

Além dos dados disponibilizados pela Carris, encorajamos o uso de outros dados disponíveis gratuitamente para enriquecer o dataset. Algumas ideias:

- Dados meteorológicos
- Condicionamentos de trânsito na cidade de Lisboa
- Eventos de massas, como eventos desportivos, concertos, etc.
- Pontos de interesse da cidade

Um bom lugar para começar a explorar estes dados é o portal de dados abertos da cidade de Lisboa.

Mas não se têm de limitar a estas ideias nem a esta fonte. Esta é a oportunidade para darem um toque criativo ao vosso projeto.

3 Requisitos do dashboard

O dashboard terá várias secções com diferentes propósitos. Alguns dos widgets devem ser atualizados em real-time, enquanto outros têm como principal objetivo mostrar informação histórica.

3.1 Mapa de posições

Esta secção é a que ocupa mais espaço e mostra as posições em real-time dos vários veículos ao serviço neste exato momento.

3.2 Veículo em foco

Esta secção mostra métricas real-time e atributos de um determinado veículo selecionado pelo utilizador.

Métricas:

- Velocidade média nos últimos 2 minutos
- Distância percorrida nos últimos 2 minutos
- Tempo estimado até à próxima paragem

O desafio é não recorrer à informação de velocidade fornecida pelo endpoint `/vehicles`. Em vez disso, devem calcular estas métricas (exceto o tempo até próxima paragem) através da agregação de uma janela de dois minutos aplicada às coordenadas do veículo.

Atributos:

- Linha
- Rota
- Direção
- Próxima paragem

3.3 Métricas históricas

Esta secção tem como objetivo mostrar métricas agregadas para um determinado período selecionado pelo utilizador. Sendo informação histórica, a expectativa é que estas métricas sejam atualizadas uma vez por dia.

- Velocidade média
- Número de viagens
- Quilómetros percorridos
- Tempo de viagem total

3.4 Filtros gerais

O dashboard permite ao utilizador definir um conjunto de filtros que se aplicam a todos os widgets não real-time. Estes são:

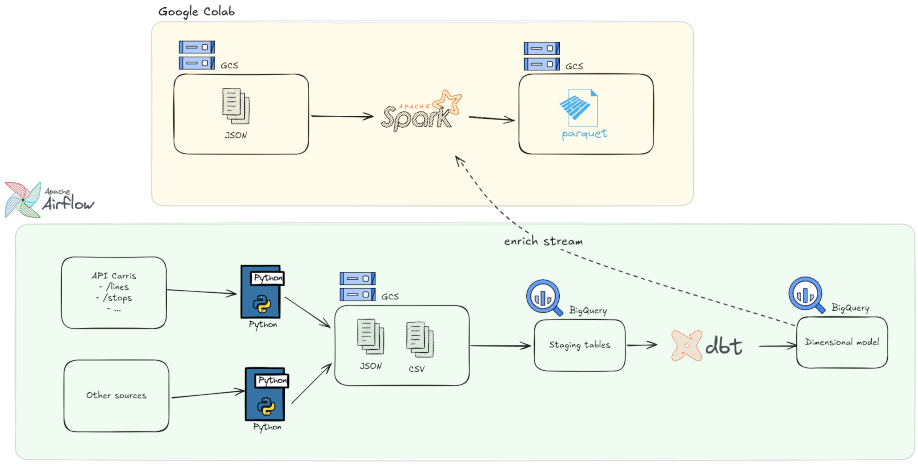
- Data
- Linha
- Rota
- Direção
- Rotas que param em estação X
- Rotas que servem munícipio X

3.5 Data em foco

Esta secção do dashboard deve mostrar a data corrente, bem como qualquer informação relevante sobre essa mesma data.

4 Arquitetura geral

Vão haver essencialmente duas pipelines de dados: uma para dados em streaming e outra para dados em batch. A primeira deve ser desenvolvida recorrendo ao Spark Structured Streaming em Google Colab. A segunda deve fazer uso do Airflow para orquestrar as várias tarefas de movimentação e transformação de dados. A figura abaixo apresenta a arquitetura geral do projeto.



Para a pipeline de streaming, existe um bucket de Google Cloud Storage `edit-de-project-streaming-data` onde irá ser colocado um ficheiro JSON a cada 30 segundos. Este ficheiro irá conter a última resposta do endpoint `/vehicles` da API da Carris. Os ficheiros neste bucket têm uma vida de apenas 24 horas. A vossa pipeline deve tratar cada um destes ficheiros como um novo evento a ser processado. O resultado do processamento deve ser escrito num outro bucket de GCS.

Para a pipeline de batch, todo o processo deve ser orquestrado pelo Airflow. Devem escrever o código Python necessário para extrair os dados da fonte e colocá-los num bucket de GCS. Nesta fase, os dados devem ser mantidos o mais crus possível, sem qualquer processamento. Numa outra tarefa da pipeline, os dados crus deverão ser modelados num formato tabular e carregados no BigQuery. Mais uma vez, queremos manter as transformações ao mínimo. O objetivo final da pipeline de batch será construir um modelo dimensional que irá dar resposta aos requisitos do dashboard. As transformações necessárias para obter este modelo devem ser definidas no dbt.

O modelo dimensional pode também ser usado para enriquecer a stream de

dados no Spark, fazendo join dos dados de veículos com dados dimensionais, como atributos da linha, rota, etc.

5 Entregáveis

Os principais entregáveis do projeto serão todos os ativos usados para produzir o dataset final. Isto inclui:

- DAGs Airflow
- Projeto dbt
- Jobs de Spark

Todos estes ativos devem estar devidamente organizados num repositório de GitHub.

Além disto, devem preparar um relatório escrito e uma apresentação final de 20 minutos. Devem pensar no relatório como a documentação que deveriam escrever sobre o vosso trabalho numa empresa. Este deve conter:

- Uma introdução ao propósito do trabalho
- Uma vista geral sobre a arquitetura usada
- Detalhe sobre o modelo de dados final
- Detalhes sobre os vários passos de extração e transformação dos dados
- Razões para as várias decisões tomadas em relação à solução final

A apresentação será feita para toda a turma na última aula de Applied Project. Deve seguir a mesma estrutura geral do relatório, tendo obviamente que descartar muito detalhe. A apresentação deverám também incluir um pequeno live demo das pipelines a funcionarem.