ETL – projeto final

Especialização de Analytics

Índice

[1 Introdução 3](#_Toc123640318)

[2 Arquitetura 4](#_Toc123640319)

[2.1 Design escolhido 4](#_Toc123640320)

[2.1.1 Data Source 4](#_Toc123640321)

[2.1.2 Data Ingestion 4](#_Toc123640322)

[2.1.3 Data Engineering 5](#_Toc123640323)

[2.1.4 Data Visualization 9](#_Toc123640324)

[2.1.5 Componentes da seleção e decisão da Arquitetura 12](#_Toc123640325)

[2.2 Data Flow Design 12](#_Toc123640326)

Figura 1 - Modelo inicial 3

Figura 2 - Arquitetura do projeto de Analytics 4

Figura 3 - Importação das bibliotecas no Databricks 5

Figura 4 - Modelo Dimensional BikeCapStore 5

Figura 5 - Dim\_Staffs final 6

Figura 6 - Dim\_Customers final 6

Figura 7 - Dim\_Products final 6

Figura 8 - Dim\_Stores final 7

Figura 9 - Criação da Dimensão Data 7

Figura 10 - Dim\_Date final 8

Figura 11 - Função da transformação das datas para chaves 8

Figura 12 - Fact\_Orders final 8

Figura 13 - Limpeza do DBFS e exportação dos dataframes para tabelas 9

Figura 14 - Ligação direta do Databricks para o PowerBI 10

Figura 15 - Pedido de especificações do cluster pelo PowerBI 10

Figura 16 - Especificações do cluster 11

Figura 17 - Modelo Dimensional importado no PowerBI 11

Figura 18 - Data Flow do projeto 12

Tabela 1 – Inputs, caminhos e endpoints do ETL....................................................................................................................13

# Introdução

Este relatório pretende fornecer toda a informação técnica utilizada durante o desenvolvimento do projeto de ETL para a BikeCapStore, de forma a possibilitar a sua reprodução por outros membros da equipa.

Para a elaboração do projeto final da Especialização de Analytics, os dados fornecidos foram trabalhados no Dbeaver, e posteriormente exportados para o repositório do GitHub. De seguida foi feita a ingestão dos ficheiros para o Databricks, onde foi construído o modelo dimensional, através de várias operações de transformação de dados. No final desta tarefa, as novas tabelas foram exportadas para o Power BI.

Apresenta-se o input original na Figura 1.

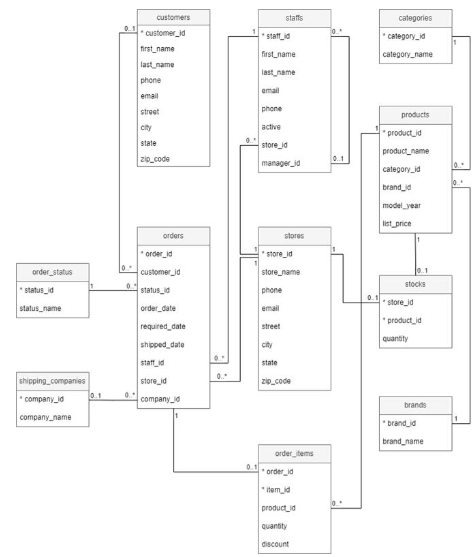


Figura 1 - Modelo inicial

# Arquitetura

## Design escolhido

A Figura 2 representa a arquitetura do projeto de Analytics.

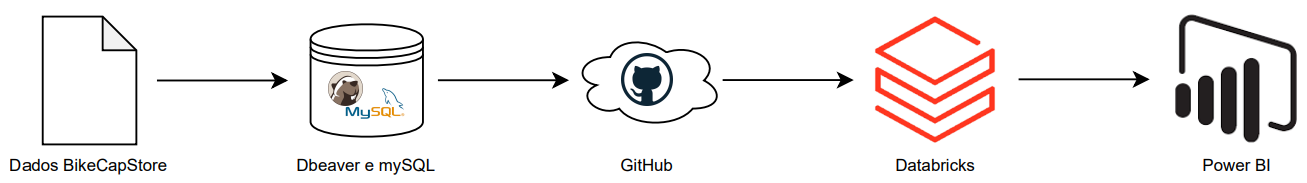


Figura 2 - Arquitetura do projeto de Analytics

### Data Source

O GitHub é responsável por manter os ficheiros em Cloud para diferentes equipas terem acesso aos dados.

Neste projeto foram utilizados dados de uma única fonte, o repositório GitHub, onde estão ficheiros do tipo csv, na seguinte localização: <https://github.com/XLucas27X/bikecapstore>

### Data Ingestion

O Databricks foi utilizada como ferramenta de ETL para ingerir os ficheiros do GitHub, através do comando “wget” do Power Shell.

Para a ingestão é utilizada o raw link de cada ficheiro.

%sh  
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/brands.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/categories.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/customers.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/order_items.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/orders.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/products.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/staffs.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/stocks.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/stores.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/shipping_companies.csv>   
wget <https://raw.githubusercontent.com/XLucas27X/bikecapstore/main/order_status.csv>

O notebook pode ser consultado no seguinte link (é necessário manter sempre o cluster ativo):

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/3904951712713107/315028503103403/245661924386905/latest.html>

### Data Engineering

O processo inicia-se com a importação das bibliotecas necessárias de modo a facilitar a transformação dos dados. Neste projeto utilizaram-se as linguagens Python, SQL e Linux.

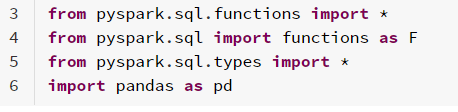


Figura 3 - Importação das bibliotecas no Databricks

Foi desenvolvida uma função que automatiza a criação de dataframes a partir dos csv importados. Nos parâmetros utilizados assume-se a primeira linha como “TRUE”, inferimos o schema de cada ficheiro e utilizou-se a vírgula como separador. O formato de leitura é csv e os ficheiros foram carregados a partir do filesystem do databricks: file:/databricks/driver/

Após a criação dos dataframes, procedemos ao design do modelo:

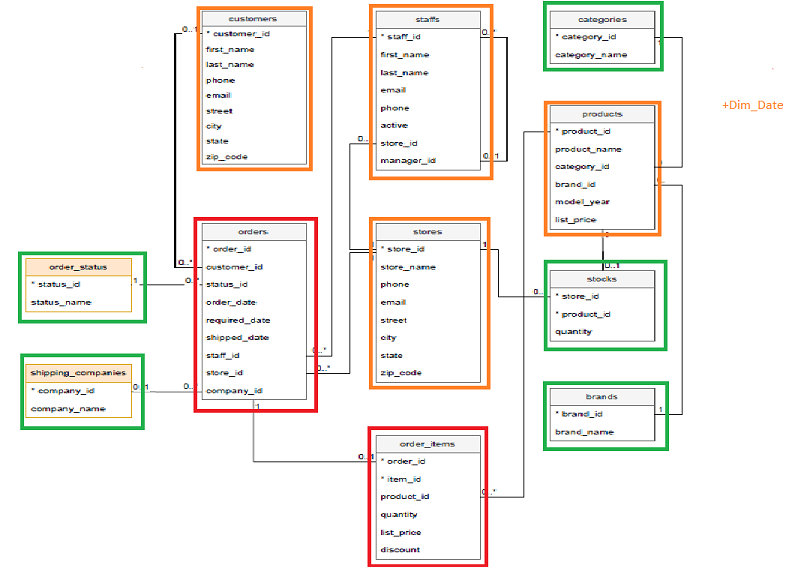


Figura 4 - Modelo Dimensional BikeCapStore

As tabelas foram primeiramente classificadas consoante o grau de granularidade, e foram criadas as diferentes dimensões do modelo e a facts table, de modo a formar o star schema.

**Vermelho** 🡪 Facts Table

**Laranja** 🡪 Dimensões

**Verde** 🡪Tabelas que vão ser integradas conforme abaixo

**Dim\_Staffs:** Eliminação da coluna com o id desncessário (store\_id) para o modelo final.

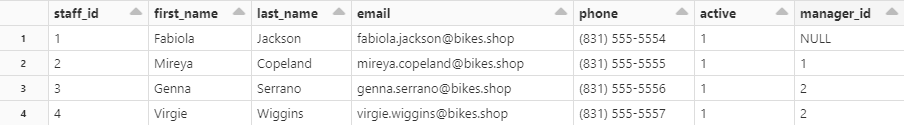


Figura 5 - Dim\_Staffs final

**Dim\_Customers:** Foi mantida a tabela original.



Figura 6 - Dim\_Customers final

**Dim\_Products:**

* Join das categories e brands com os products para criar a dimensao products
* Eliminação dos id’s repetidos e desnecessários (brands\_id e category\_id), resultantes do join das tabelas.



Figura 7 - Dim\_Products final

**Dim\_Stores:** Foi mantida a tabela stores original, descartando a tabela stocks, uma vez que a mesma não continha informação relevante para responder aos KPI’s solicitados.

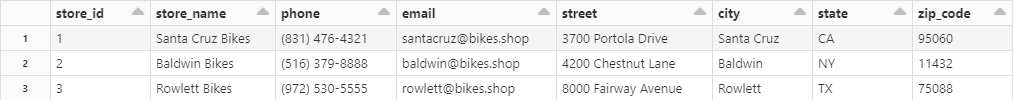


Figura 8 - Dim\_Stores final

**Dim\_Date:**

* Criação manual da dimensão data, recorrendo a uma view desenvolvida em Spark SQL, com as seguintes colunas:

1. calendarDate (data completa)
2. Day\_Name (nome do dia da semana)
3. DayOfMonth (número do dia do mês)
4. DayOfWeek (número do dia da semana)
5. DayOfYear (número do dia do ano)
6. IsWeekDay (se é dia útil ou não)
7. Month (número do mês)
8. Month\_Name (nome do mês)
9. QuarterOfYear (número do trimestre)
10. Year (ano)
11. SK\_date (chave primária)



Figura 9 - Criação da Dimensão Data

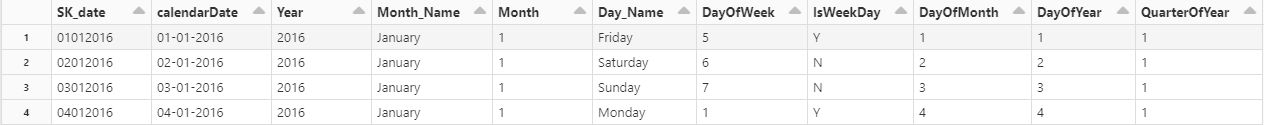


Figura 10 - Dim\_Date final

**Fact\_Orders:**

* Join das tabelas orders, orders\_status e shipping\_companies para criar a facts table;
* Eliminação dos id’s repetidos e desnecessários (status\_id e company\_id), que resultaram do join das tabelas;
* Aplicação de uma função que foi desenvolvida para alterar o formato default do Databricks para o do PBI (yyyyMMdd para ddMMyyyy);
* Transformação das datas em chaves (da facts table), de forma a ligar com a dimensão data através da SK\_date (i.e., foram retirados os “-“ e zeros (00:00:00) das datas).

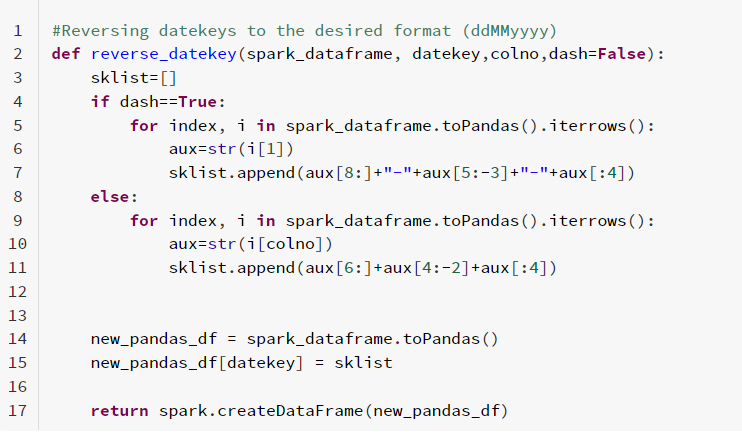


Figura 11 - Função da transformação das datas para chaves

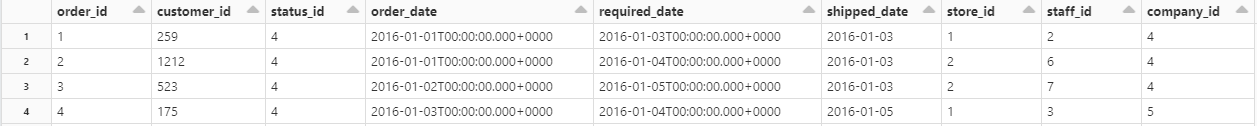


Figura 12 - Fact\_Orders final

Por fim, efetou-se a limpeza do DBFS e a exportação dos dataframes para tabelas no Databricks.

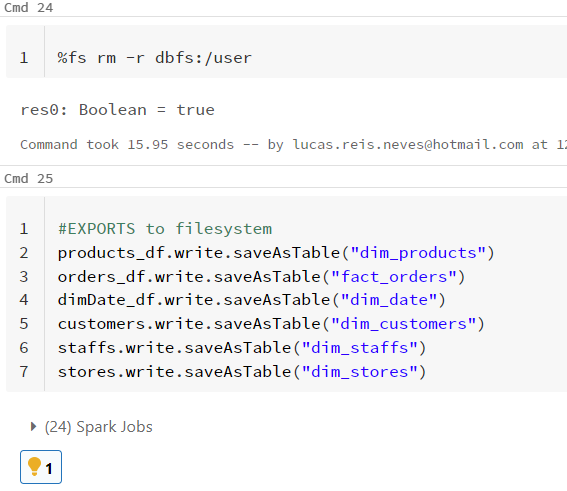


Figura 13 - Limpeza do DBFS e exportação dos dataframes para tabelas

No que concerne ao processamento dos dados, foi utilizado o motor Spark através de um cluster do Databricks. O cluster tem um runtime 10.4, active memory 15GB e active cores 2.

Spark config:

spark.databricks.rocksDB.fileManager.useCommitService - false

spark.databricks.delta.preview.enabled - true

Environment variables:

PYSPARK\_PYTHON=/databricks/python3/bin/python3

### Data Visualization

O PowerBI é o último componente da arquitetura, utilizado como ferramenta para a visualização. O PowerBI permite uma ligação directa com o Databricks.

Os dataframes criados encontram-se no Databricks filesystem, no seguinte diretório:

dbfs:/user/hive/warehouse

Para processar esta ligação é necessário abrir o PBI, fazer a importação dos dados pelo Get Data, escolher o Azure Databricks e conectar.

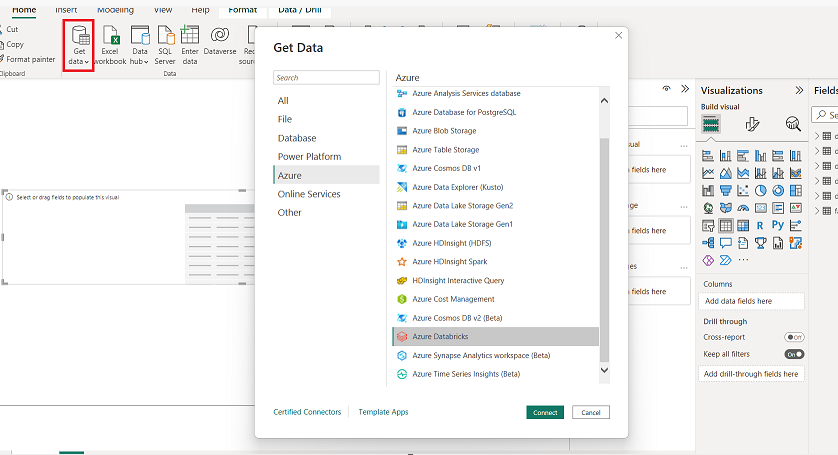


Figura 14 - Ligação direta do Databricks para o PowerBI

De seguida colocar as especificações solicitadas -Server Hostname e e HTTP- (No Databricks, Ir a Compute, entrar no cluster utilizado e escolher a opção JDBC/ODBC). Posteriormente, utilizar as credenciais de acesso ou token de autenticação do Databricks. Selecionar as tabelas a carregar para o modelo.

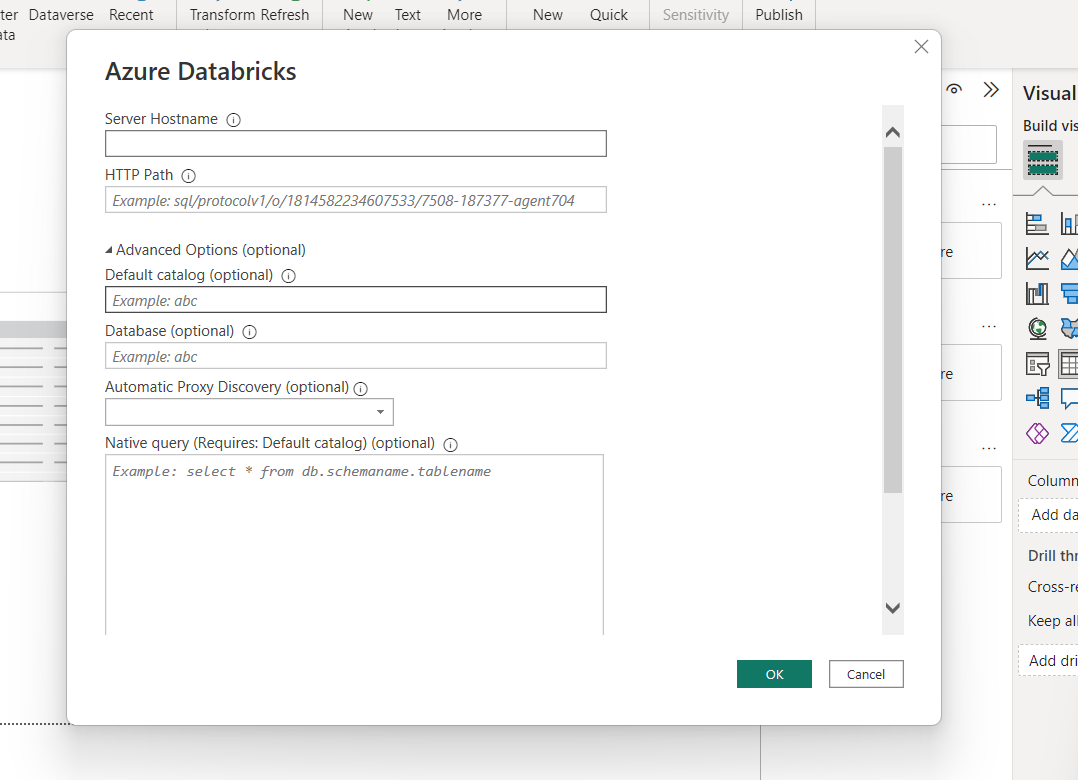


Figura 15 - Pedido de especificações do cluster pelo PowerBI

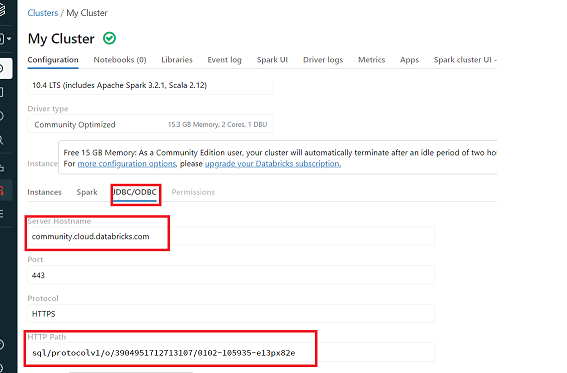


Figura 16 - Especificações do cluster

Da importação resulta o seguinte modelo:

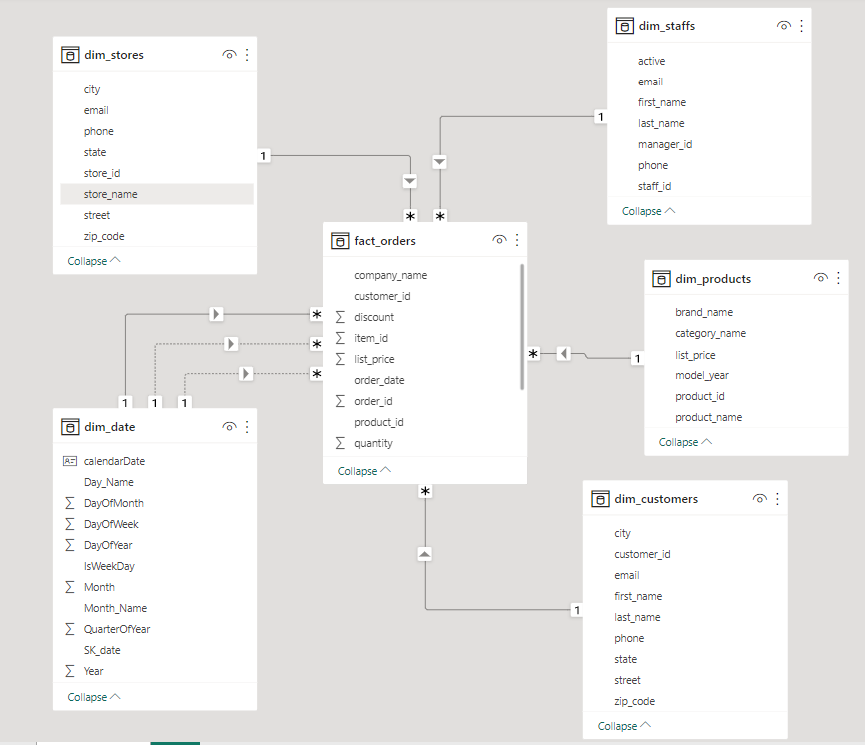


Figura 17 - Modelo Dimensional importado no PowerBI

As ligações entre as dimensões e facts table surgem automaticamente com a importação, com exceção da dim\_date, onde é necessário ligar a SK\_date com as 3 chaves de datas da facts table.

### Componentes da seleção e decisão da Arquitetura

* GitHub: permite uma fácil colaboração de equipas de desenvolvimento, o tracking do histórico das alterações e o armazenamento dos dados em Cloud, de forma gratuita.

Documentação oficial: <https://docs.github.com/en/get-started>

* Databricks: fornece uma ingestão simplificada de diversas e múltiplas fontes, possui uma abordagem declarativa para construir pipelines e automizar o ETL. Possibilita ainda uma ligação direta com o Power BI, permite a utilização de diferentes linguagens (e.g. Pyspark e SQL) e tem escalabilidade para Big Data.

Documentação oficial: <https://docs.databricks.com/introduction/index.html>

* Power BI: tecnologia líder de mercado como ferramenta de visualização, que permite a transformação de dados em informação coerente, visualmente envolventes e interativos. Acrescenta-se a facilidade de integração com as outras tecnologias escolhidas e o tipo de ficheiros (csv) utilizados.

Documentação oficial: <https://learn.microsoft.com/en-us/power-bi/>

<https://powerbi.microsoft.com/en-us/why-power-bi/>

## Data Flow Design

Na figura 12 apresenta-se o Data Flow esperado após toda a integração.

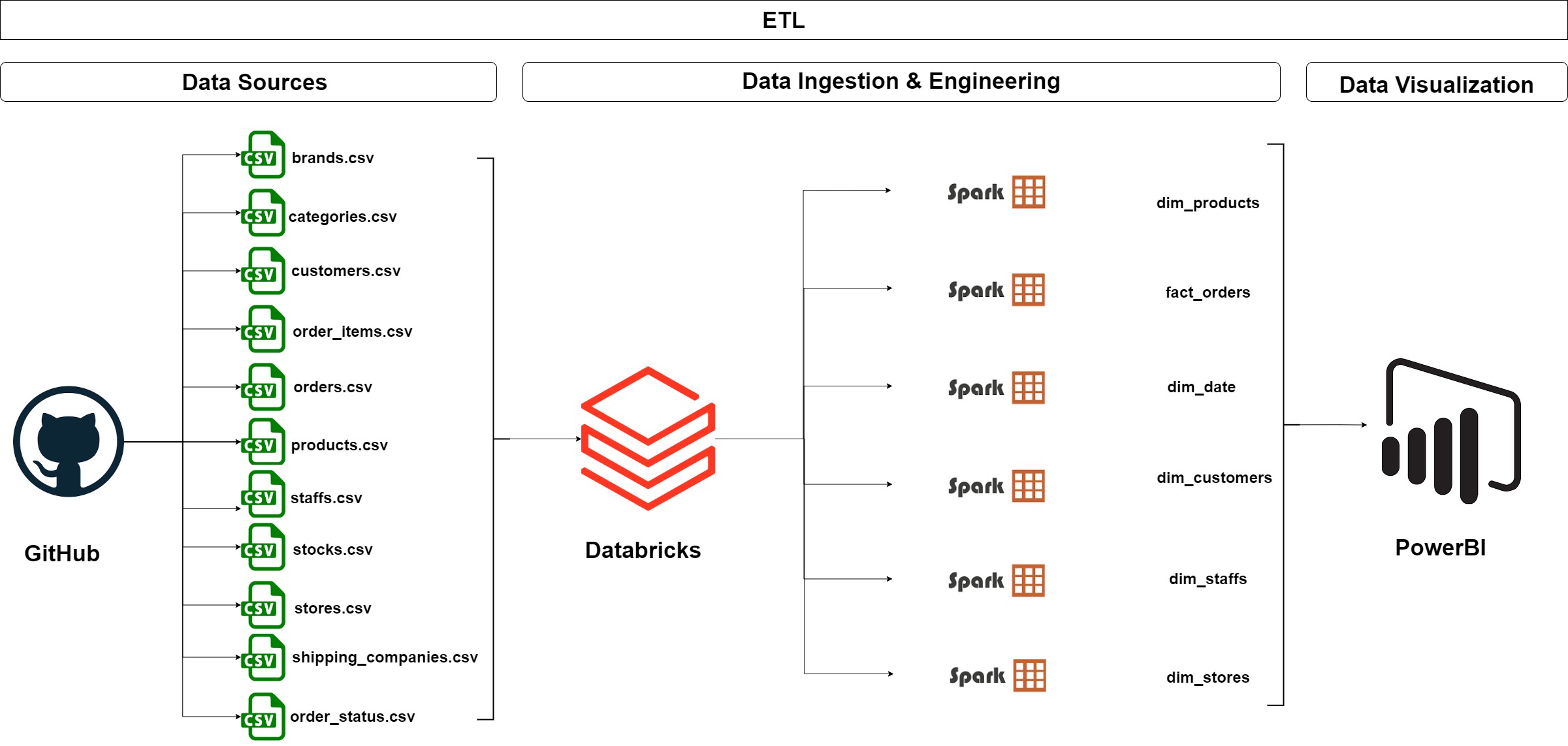


Figura 18 - Data Flow do projeto

Na Tabela 1 apresentam-se os inputs, caminhos e endpoints do ETL.

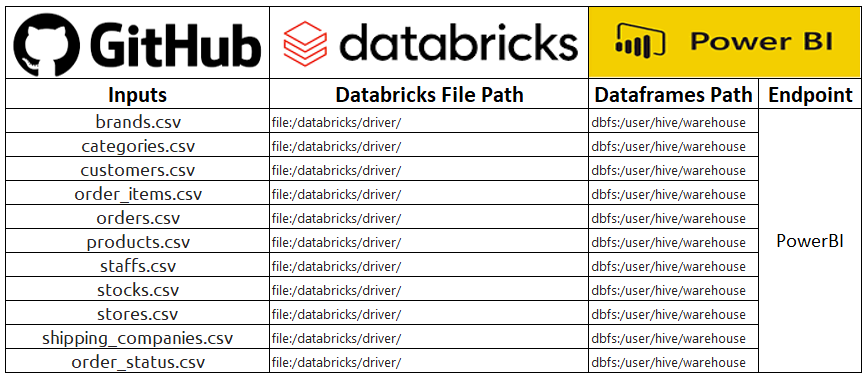


Tabela 1 - Inputs, caminhos e endpoints do ETL

Sobre Capgemini

A Capgemini é uma líder global em parcerias com empresas para transformar e gerir os seus negócios, aproveitando o poder da tecnologia. O Grupo é guiado diariamente pelo objetivo de libertar a energia humana através da tecnologia para um futuro inclusivo e sustentável. É uma organização responsável e diversificada com mais de 270.000 profissionais em quase 50 países. Com uma forte herança de 50 anos e profunda experiência no setor, a Capgemini tem a confiança dos seus clientes para responder a toda a amplitude das suas necessidades de negócio, desde estratégia e desenho até operações, alimentada pelo mundo inovador e em rápida evolução de *cloud*, dados, IA, conectividade, software, engenharia digital e plataformas. O Grupo reportou em 2020 uma receita global de € 16 mil milhões.

Get the Future You Want | [www.capgemini.com](https://www.capgemini.com/)



Este documento contém informações que podem ser privilegiadas ou   
confidenciais e a propriedade é do Grupo Capgemini.

Choose an item. Copyright © 2021 Capgemini. Todos os direitos reservados.