

# Minimum Viable Product (MVP)

Sprint: Engenharia de Dados (40530010057\_20240\_01)

## 1. Introdução

### 1.1 Objetivo

No contexto atual, a eficiência logística é um fator crucial para o sucesso de empresas que dependem de entregas de produtos aos seus clientes. A capacidade de gerenciar e otimizar a logística de distribuição não apenas melhora a satisfação do cliente, mas também reduz custos operacionais e aumenta a competitividade no mercado. No entanto, essa tarefa pode ser desafiadora, especialmente para empresas que operam em grande escala e em diversas regiões geográficas.

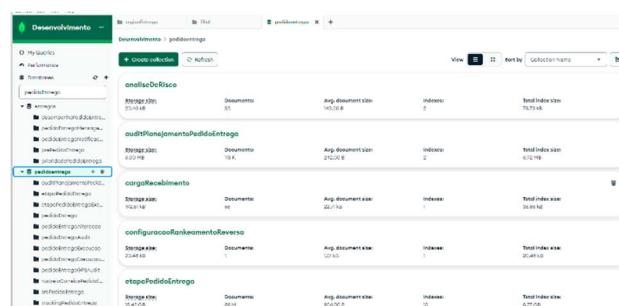
Com a evolução das tecnologias de big data e cloud computing, tornou-se possível coletar, armazenar e analisar grandes volumes de dados de maneira eficiente. Estas tecnologias permitem a criação de pipelines de dados robustos que podem integrar, processar e analisar dados de várias fontes para fornecer insights valiosos e facilitar a tomada de decisões estratégicas.

O objetivo deste MVP (Minimum Viable Product) é desenvolver um pipeline de dados utilizando tecnologias de nuvem para responder a três questões críticas relacionadas à logística de distribuição:

1. Quantos pedidos saem diariamente para entrega?
2. Quantos pedidos foram entregues por Estado até a presente data?
3. Qual a distância total percorrida por mês?

Para alcançar esse objetivo, serão utilizados quatro conjuntos de dados principais:

- Origem do legado
  - Banco de dados – Atlas Mongo
    - Databases:
      - pedidoentrega:
        - documento: pedidoEntrega
      - entregas:
        - documentos: roteiro, filial
      - endereço



Collection	Storage Size	Documents	Avg. document size	Indexes	Total Index Size
pedidoentrega	15.43 MB	85	181.0 B	2	79.73 KB
entregas	1.02 MB	184	5.52 KB	2	4.72 MB
endereco	19.61 MB	98	201.1 B	1	39.89 KB
configuracaofilial	23.48 KB	1	23.48 KB	1	23.48 KB
etapaPedidoentrega	8.47 MB	28	302.0 B	0	0.77 MB

➢ regioaoEntrega

- **Ingestão de Dados – camada raw**

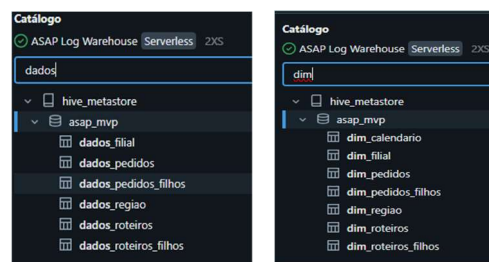
- **dados\_pedidos:** Contém informações detalhadas sobre todos os pedidos diários desde janeiro de 2024.
- **dados\_rotatorios:** Inclui dados sobre as rotas e os respectivos veículos de entrega.
- **dados\_regioes:** Cadastro das regiões para onde os pedidos são enviados.
- **dados\_filial:** Cadastro de todos os centros de distribuição envolvidos no processo.
- **dados\_pedidos\_filho:** Contém todos os códigos de roteiros que estão no legado estruturado como array.
- **dados\_rotatorio\_filho:** Contém todos os códigos de pedidos que estão no legado estruturado como array.

- **Tabelas Staging – camada bronze**

- **dim\_pedidos:** Contém informações detalhadas sobre todos os pedidos diários com tratamento inicial das informações.
- **dim\_rotatorios:** Inclui dados sobre as rotas e os respectivos veículos de entrega com tratamento inicial das informações.
- **dim\_regioes:** Cadastro das regiões para onde os pedidos são enviados com tratamento inicial das informações.
- **dim\_filial:** Cadastro de todos os centros de distribuição envolvidos no processo com tratamento inicial das informações.
- **dim\_pedidos\_filhos:** Contém todos os códigos de roteiros que estão no legado.
- **dim\_rotatorio\_filho:** Contém todos os códigos de pedidos que estão no legado.

Este documento descreverá a construção de um pipeline de dados que envolve a busca, coleta, modelagem, carga e análise desses dados. A infraestrutura será desenvolvida utilizando tecnologias de nuvem para garantir escalabilidade, flexibilidade e eficiência no processamento dos dados. Ao final, espera-se obter uma análise detalhada que permita identificar padrões e insights valiosos sobre a logística de distribuição da empresa.

Todas as tabelas estão no hive\_metastore dentro do schema asap\_mvp.



## 1.2 Visão Geral

Este MVP (Minimum Viable Product) tem como objetivo a construção de um pipeline de dados na nuvem para responder a quatro questões críticas relacionadas à logística de distribuição. A seguir, detalharemos a abordagem e as etapas principais envolvidas no desenvolvimento do pipeline, bem como as tecnologias que serão utilizadas.

### 1.2.1 Abordagem

A abordagem para o desenvolvimento do pipeline de dados será dividida em cinco etapas principais:

1. **Coleta de Dados:**
  - Leitura do legado com origem no Atlas Mongo.
  - Carga dos dados contendo os dados de pedidos, roteiros, regiões e centros de distribuição.
  - Armazenamento dos dados brutos em um serviço de armazenamento na nuvem no Amazon S3 em formato Delta Table Parquet.
2. **Modelagem de Dados:**
  - Limpeza e pré-processamento dos dados utilizando ferramentas como Pyspark, Pandas e SQL, todos dentro de notebooks.
  - Estruturação dos dados em um formato adequado para análise, garantindo a integridade e consistência dos mesmos.
3. **Carga de Dados:**
  - Implementação de processos de ETL (Extract, Transform, Load) para mover os dados dos arquivos brutos para um data warehouse na nuvem, como Delta Table, usando notebooks.
  - Utilização de ferramentas como Apache Pyspark para orquestrar e automatizar o processo de ETL.
4. **Análise de Dados:**
  - Desenvolvimento de consultas SQL para extrair insights dos dados armazenados no data warehouse.
  - Utilização de ferramentas de análise e visualização de dados, como Databricks Painel, para criar dashboards e relatórios interativos.
5. **Resposta às Questões Críticas:**
  - Análise dos dados processados para responder às três questões principais:
    - Quantos pedidos saem diariamente para entrega?
    - Quantos pedidos foram entregues por Estado até a presente data?
    - Qual a distância total percorrida por mês?

## 1.2.2 Tecnologias Utilizadas

Para garantir a eficiência, escalabilidade e flexibilidade do pipeline de dados, serão utilizadas as seguintes tecnologias:

- **Plataforma:**
  - Databricks
- **Armazenamento de Dados:**
  - Amazon S3 em formato parquet – Delta Tables
- **Processamento e Modelagem de Dados:**
  - Pyspark
  - SQL
- **ETL (Extract, Transform, Load):**
  - Pyspark
- **Data Warehouse:**
  - Delta Tables – catálogo Hive Metastore com o schema asap\_mvp
- **Análise e Visualização de Dados:**
  - Painel Databricks

## 1.2.3 Fluxo de Trabalho do Pipeline

O fluxo de trabalho do pipeline de dados pode ser resumido nos seguintes passos:

1. **Importação e leitura:**
  - Ler e carregar os dados do Mongo Atlas.
2. **Transformação e Limpeza:**
  - Limpar e transformar os dados utilizando Pyspark e SQL.
3. **Carga no Data Warehouse:**
  - Executar processos de ETL para mover os dados transformados para o data warehouse.
4. **Análise e Visualização:**
  - Criar consultas SQL e dashboards para analisar os dados e responder às perguntas críticas.

## 1.2.4 Benefícios Esperados

Com a implementação deste pipeline de dados, espera-se alcançar os seguintes benefícios:

- **Eficiência Operacional:** Automação do processo de coleta, transformação e carga de dados.
- **Insights Valiosos:** Capacidade de identificar padrões e tendências nas operações logísticas.
- **Tomada de Decisão Informada:** Dados precisos e atualizados para suportar decisões estratégicas.
- **Escalabilidade:** Infraestrutura de dados escalável para acomodar o crescimento dos dados e das operações.

## 2. Descrição do Produto

### 2.1 Funcionalidades Principais

Conectar ao mongo usando script de conexão;

Ler os dados em ambiente Analytics do Mongo Atlas;

Carregar os dados em um dataframe;

Filtrar as colunas que serão base de uso para projeto proposto;

Carregar em tabelas delta na camada raw.

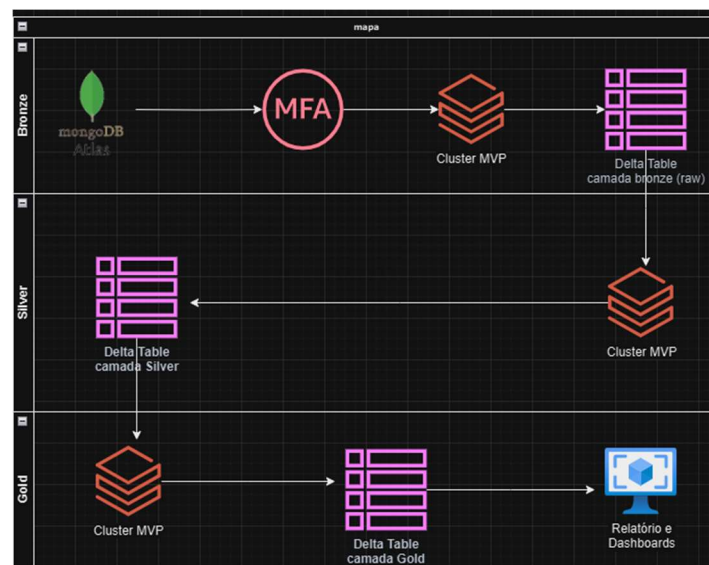
Tratar os dados na camada silver;

Carregar na camada Gold;

Aplicar com o SQL nos painéis.

### 2.2 Fluxo de Usuário

O usuário apenas acessa o painel e visualiza os resultados.



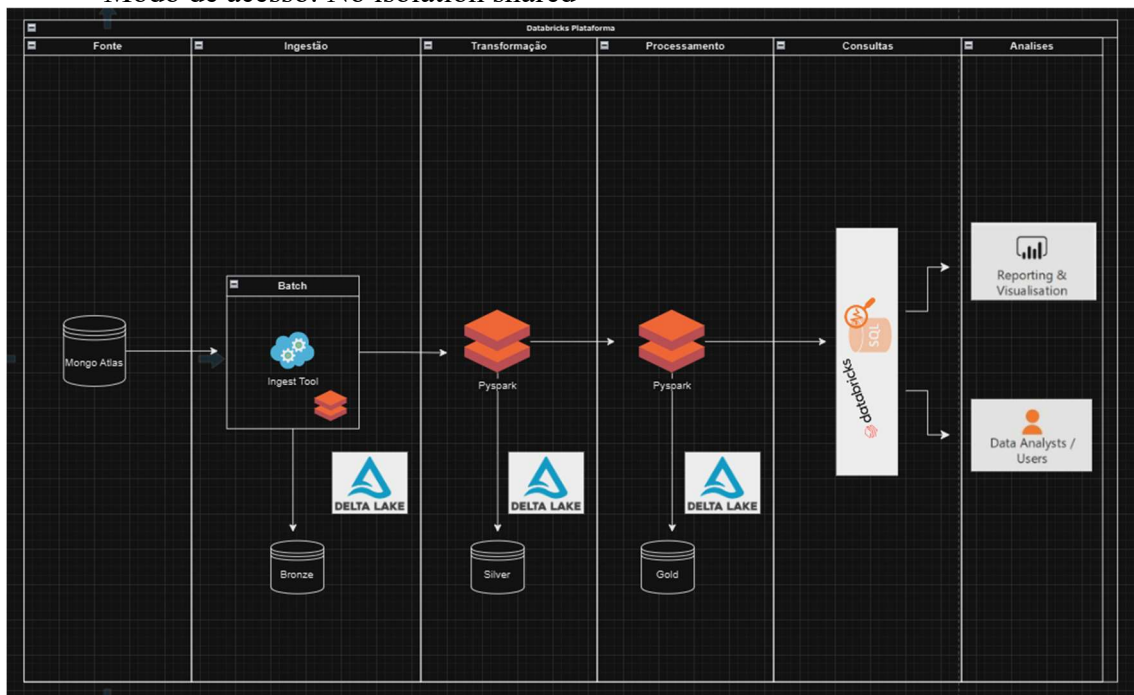
## 3. Requisitos Técnicos

### 3.1 Arquitetura do Sistema

Utilizei a plataforma Databricks, onde fazemos uso em 100% do Pyspark nas fases de ingestão, transformação e processamento com cargas em Delta Tables com formato parquet salvos no S3, em seguida uso o SQL Warehouse como serverless para utilização de consultas em Sql e publicação e integração com painéis.

Para tudo isso uso um cluster com as seguintes configurações:

- Apache Spark na versão 3.4.1, Scala 2.12
- Work Type: m4.2xlarge com 32 Gb de memória e 8 núcleos
- Mínimo de Workers 2 e máximo de Workers 8
- Com política sem restrições multi node
- Modo de acesso: No isolation shared



**cluster\_mvp**

Configuração | Blocos de notas (0) | Bibliotecas | Registro de eventos | IU do Spark | Logs do driver

Política ⓘ

Sem restrições

☒ Multi node ☐ Nó único

Modo de acesso ⓘ

No isolation shared

**Desempenho**

Versão do Databricks Runtime

13.3 LTS (includes Apache Spark 3.4.1, Scala 2.12)

☐ Usar a aceleração do Photon ⓘ

Worker Type ⓘ

Min. workers Máx. workers

m4.2xlarge 32 GB de memória, 8 núcleos 2 8

Tipo de driver

m4.2xlarge 32 GB de memória, 8 núcleos

### **3.2 Tecnologias Utilizadas**

- Databricks
- Delta Tables
- Sql Warehouse
- Sql Ansi

### **3.3 Requisitos de Desempenho**

de 2 a 8 nós

com capacidade de 64 a 256 GB de memória

de 16 a 64 núcleos com dimensionamento automático

de 4 a 14 DBU/h

## **4. Plano de Desenvolvimento**

### **4.1 Cronograma**

O cronograma foi dispensado em virtude da implantação ser imediata ao desenvolvimento e validação.

### **4.2 Recursos Necessários**

Notebook

Acesso ao databricks

Ambiente AWS com acesso ao S3

## **5. Testes e Validação**

### **5.1 Estratégia de Testes**

Os testes são realizados em tempo de desenvolvimento validando através de visualizações com instruções em Sql.

### **5.2 Critérios de Aceitação**

A disponibilidade dos dados, a garantia dos dados com informações comparadas com o legado, painel com informações básicas ajudarão a garantir a qualidade.

## **6. Lançamento e Feedback**

### **6.1 Plano de Lançamento**

Sem definição para este MVP

### **6.2 Coleta de Feedback**

Através da reunião de apresentação vamos buscar entender se o produto final satisfaz o cliente.

## **7. Conclusão**

### **7.1 Resumo**

Este MVP serviu para testar e ajudar a implementar uma forma de processo de pipeline com ingestão, transformação e carga para consumo com Slowly Changing Dimension (SCD) Tipo 1, onde as tabelas sofrem atualizações de Update para registros já existentes e Insert para os novos registros encontrados.

Autor: **Antonio Lacerda de Castro Junior**



Anexos:

Anexo 1 – imagens do databricks rodando o pipeline

Anexo 2 – imagens do catálogo asap\_mvp

Anexo 3 – imagens do databricks Sql do painel

Git: [mvplacerdadecastro/mvp\\_engenharia\\_de\\_dados \(github.com\)](https://github.com/mvplacerdadecastro/mvp_engenharia_de_dados)

[Acesso databricks:](#)

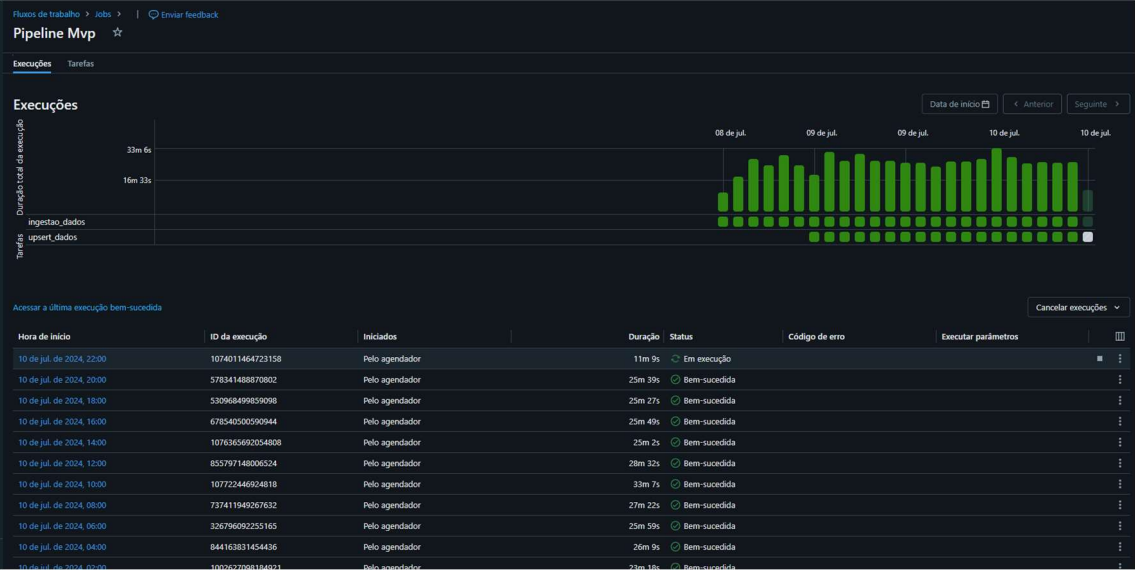
[Login - Databricks](#)

Usuário: [mvp.lacerdadecastro@gmail.com](mailto:mvp.lacerdadecastro@gmail.com)

Senha: Mvp#2024

Anexo 1

Pipeline sendo executado:



Fluxos de trabalho > Jobs >

Pipeline Mvp ☆

Execuções Tarefas

ingestao\_dados

upsert\_dados

+ Adicionar tarefa

Nome da tarefa\* upsert\_dados

Tipo\* Notebook

Origem\* Espaço de trabalho

Caminho\* /Workspace/Users/mvp.lacerdadecastro@gmail.com/upsert dados

Compute\* cluster\_mvp 224 GB - 56 núcleos - DBR 13.3 LTS - Spark 3.4.1 - Scala 2.12

A execução de jobs em clusters de uso geral é considerada computação multiusos (All-purpose). Saiba mais

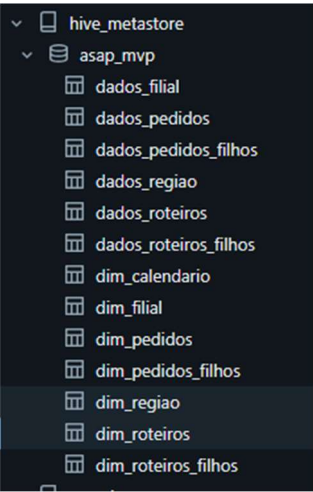
Depende de ingestao\_dados X

Executar as dependências If Tudo foi bem-sucedido

Bibliotecas dependentes + Adicionar

Anexo 2

Imagens do catálogo asap\_mvp



Todas as tabelas da asap\_mvp



Tabela stage dados\_filial

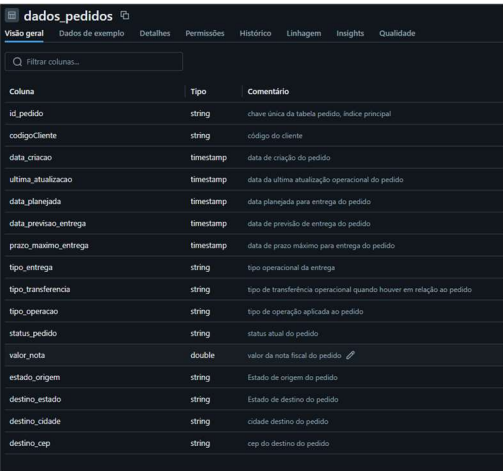


Tabela stage dados\_pedidos

**dados\_regiao**

Visão geral

Dados de exemplo

Detalhes

Permissões

Histórico

Linhagem

Insights

Qualidade

Filtrar colunas...

Coluna	Tipo	Comentário
id_regiao	string	chave única da tabela de região, índice principal
dateCreated	timestamp	data de criação da região cadastrada
lastUpdated	timestamp	data de última atualização sofrida no cadastro
codigoRegiao	string	código da região
filialId	string	id da filial usada nos relacionamentos entre filiais
nome	string	nome da filial
uf	string	uf a qual a filial pertence

Tabela stage dados\_regiao

**dados\_rotatorios**

Visão geral

Dados de exemplo

Detalhes

Permissões

Histórico

Linhagem

Insights

Qualidade

Filtrar colunas...

Coluna	Tipo	Comentário
codigoRotatorio	string	chave única da tabela de rotatorios, índice principal
date	timestamp	data de saída em rota
dateCreated	timestamp	data de criação do rotatorio
lastUpdated	timestamp	data de última atualização do rotatorio
placaVeiculo	string	placa do veículo utilizado na rota
qtdPedidos	int	quantidade de pedidos que saíram em rota
qtdVolumes	int	quantidade de volumes transportados
distanciaKm	double	distância percorrida em km na rota
regiaoEntregaGuid	string	identificador da região de entrega

Tabela stage dados\_rotatorios

## Tabelas Bronze

Possuem a mesma estrutura de metadados, mas tratamentos nos tipos e situações de nulos.

Explorador de Catálogos > hive\_metastore > asap\_mvp >

dim\_filial

Visão geralDados de exemploDetalhesPermissõesHistóricoLinhagemInsightsQualidade

Q Filtrar colunas...

Coluna	Tipo
id_filial	string
nome	string
codigoFilial	int
nomeFantasia	string

Tabela bronze dim\_filial

Explorador de Catálogos > hive\_metastore > asap\_mvp >

dim\_pedidos

Visão geralDados de exemploDetalhesPermissõesHistóricoLinhagemInsightsQualidade

Q Filtrar colunas...

Coluna	Tipo
id_pedido	string
codigoCliente	string
data_criacao	timestamp
ultima_atualizacao	timestamp
data_planejada	timestamp
data_previsao_entrega	timestamp
prazo_maximo_entrega	timestamp
tipo_entrega	string
tipo_transferencia	string
tipo_operacao	string
status_pedido	string
valor_nota	double
estado_origem	string
destino_estado	string
destino_cidade	string
destino_cep	string

Tabela bronze dim\_pedidos

Explorador de Catálogos > hive\_metastore > asap\_mvp >

dim\_regiao

Visão geralDados de exemploDetalhesPermissõesHistóricoLinhagemInsightsQualidade

Q Filtrar colunas...

Coluna	Tipo
id_regiao	string
dateCreated	timestamp
lastUpdated	timestamp
codigoRegiao	string
filialId	string
nome	string
uf	string

Tabela bronze dim\_regiao

Explorador de Catálogos > hive\_metastore > asap\_mvp >

**dim\_roteiros**

Visão geral Dados de exemplo Detalhes Permissões Histórico Linhagem Insights Qualidade

Q Filtrar colunas...

Coluna	Tipo
codigoRoteiro	string
date	timestamp
dateCreated	timestamp
lastUpdated	timestamp
placaVeiculo	string
qtdPedidos	int
qtdVolumes	int
distanciaKm	double
regiaoEntregaGuid	string

Tabela bronze dim\_roteiros

Anexo 3

Tela do Painel publicado



O painel é interativo e possui filtro de datas com opções de selecionar range de datas e períodos específicos.

