



Desafio de Arquitetura Impulso.gov

Implementação de um Data Lake



Requisitos

- Requisições e ftp volume grande, job mensal
- Google DataStudio
- Banco de dados já existente servindo analistas e aplicação
- Gerenciado por 5 profissionais: 2 cientistas de dados, 2 desenvolvedores e 1 engenheiro de dados
- Não tem limitação financeira para contratar ferramentas a serem usadas
- Necessidade da solução atender de forma a não necessitar de muita manutenção e que seja simples para que diversos membros o possam modificar.

Quais são os esquemas de fluxo de dados e ferramentas que teria usado? Por que?

Qual a arquitetura ideal?



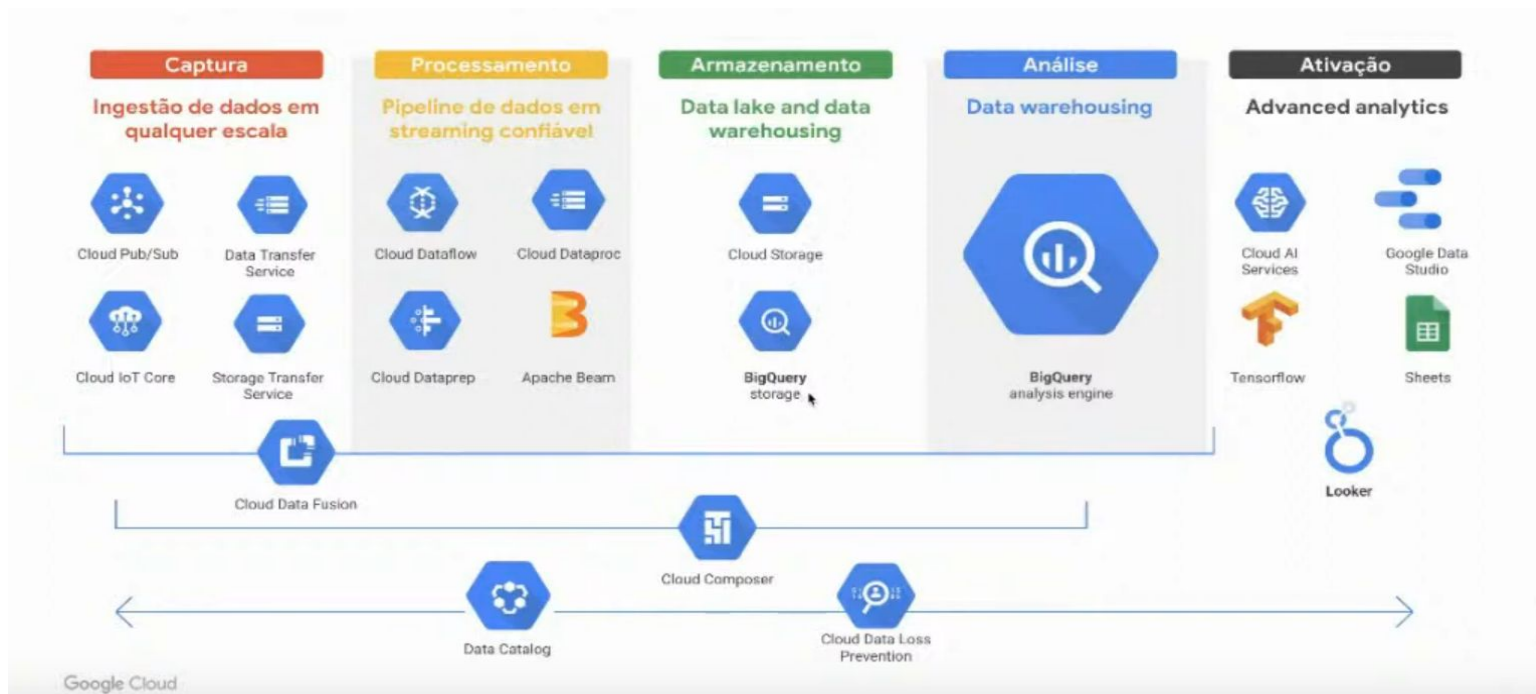
Dada a necessidade de simplificar a manutenção, equipe reduzida e liberdade orçamentária:

- Manutenção dos serviços em cloud (alta disponibilidade, redundância, sem estrutura física própria)
- Contratação de produtos serverless totalmente ou semi-gerenciados (infraestrutura)
- Boa experiência do usuário, interface agradável e fácil manuseio, controle de acesso (IAM)
- Suporte forte (comunidade, documentação, suporte técnico empresarial disponível)



Google Cloud Platform

Produtos GCP dados





A arquitetura para os dados

Dado o início do projeto, recomendo desde logo a otimização para a arquitetura Lakehouse. A mais moderna disponível com suporte a governança de dados maior organização, custos menores de processamento.

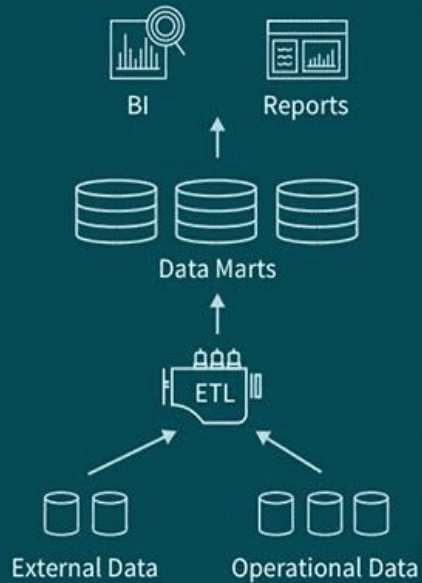
Um Data Lake mal otimizado pode virar um Data Swamp

“Data swamp ou pântano de dados, é um data lake mal projetado, sem governança, sem controle, falta de processos e padrões, dados difíceis de encontrar, de consumir etc...”

Evolução da plataforma de Análise de dados

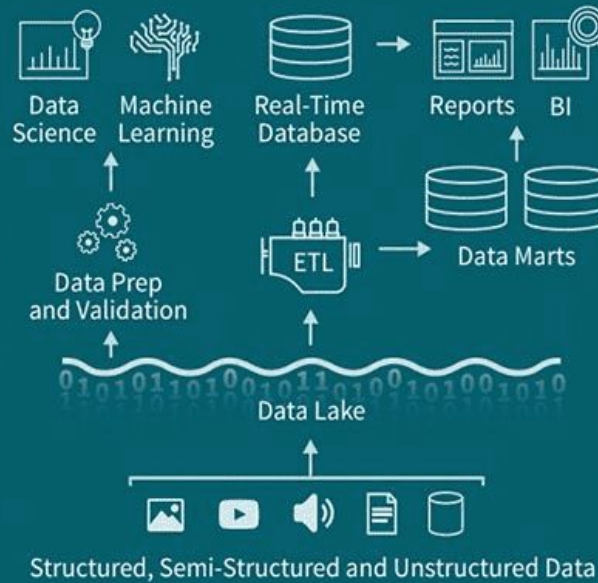
Late 1980's

Data Warehouse



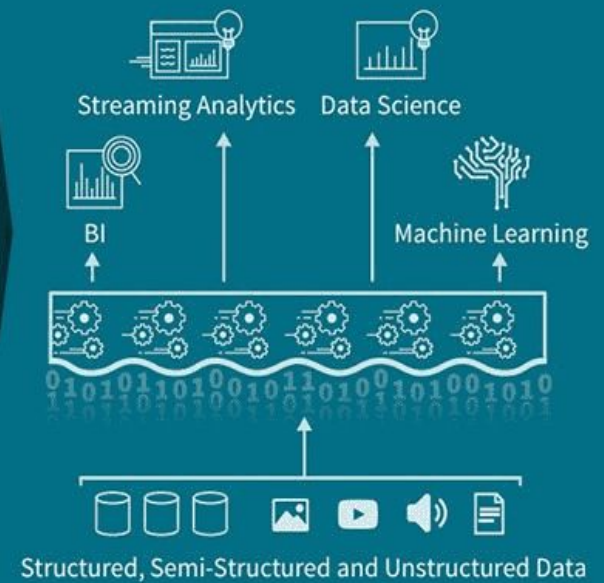
2011

Data Lake



2020

Lakehouse



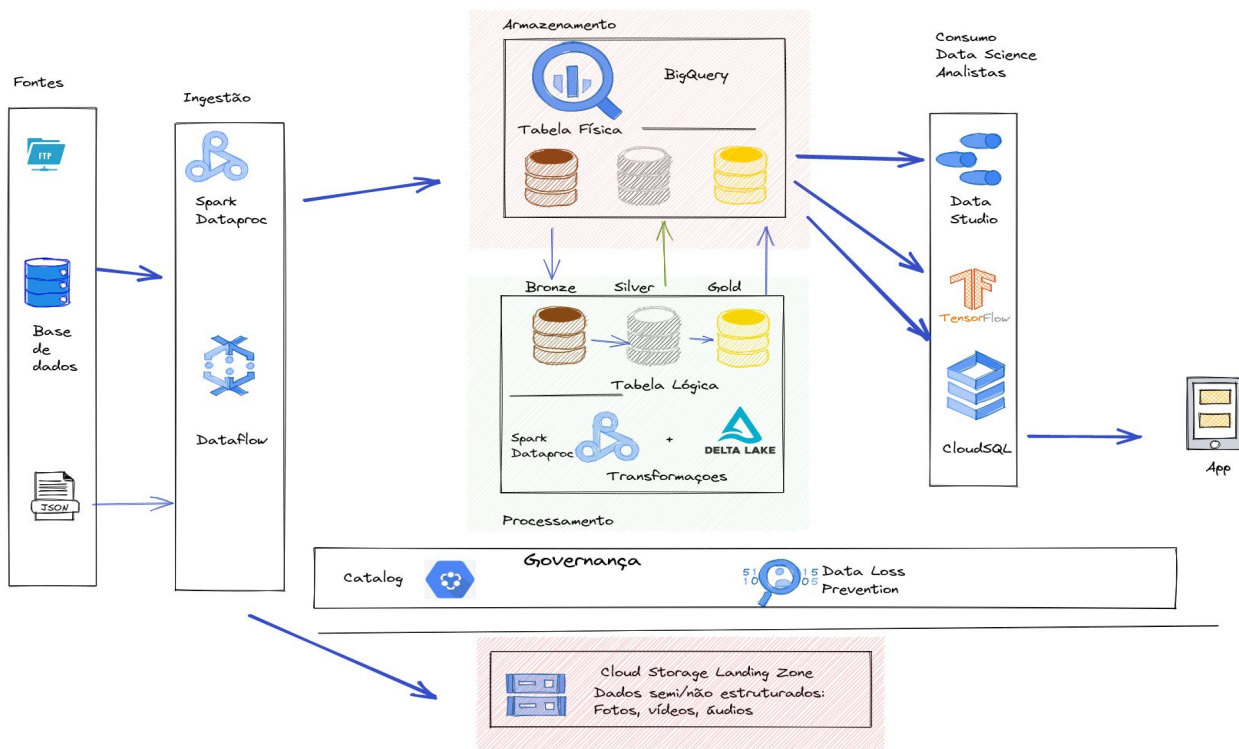


Vantagens Data Lakehouse

- Altamente escalável
- Pode ser implementado em um data lake já existente tornando-se a evolução do data lake
- Permite o monitoramento, modelagem de dados
- Governança de dados
- Implementação do ACID (DataWarehouse)
- Time Travel (Desfazer erros, recuperar dados que seriam perdidos)
- Metadados e registro de transações
- Ao contrário do Data Lake, Lakehouse evolui e permite a rastreabilidade transacional
- Leituras e escritas ocorrem simultaneamente sem inconsistências



Composer Orquestração



GCP lakehouse

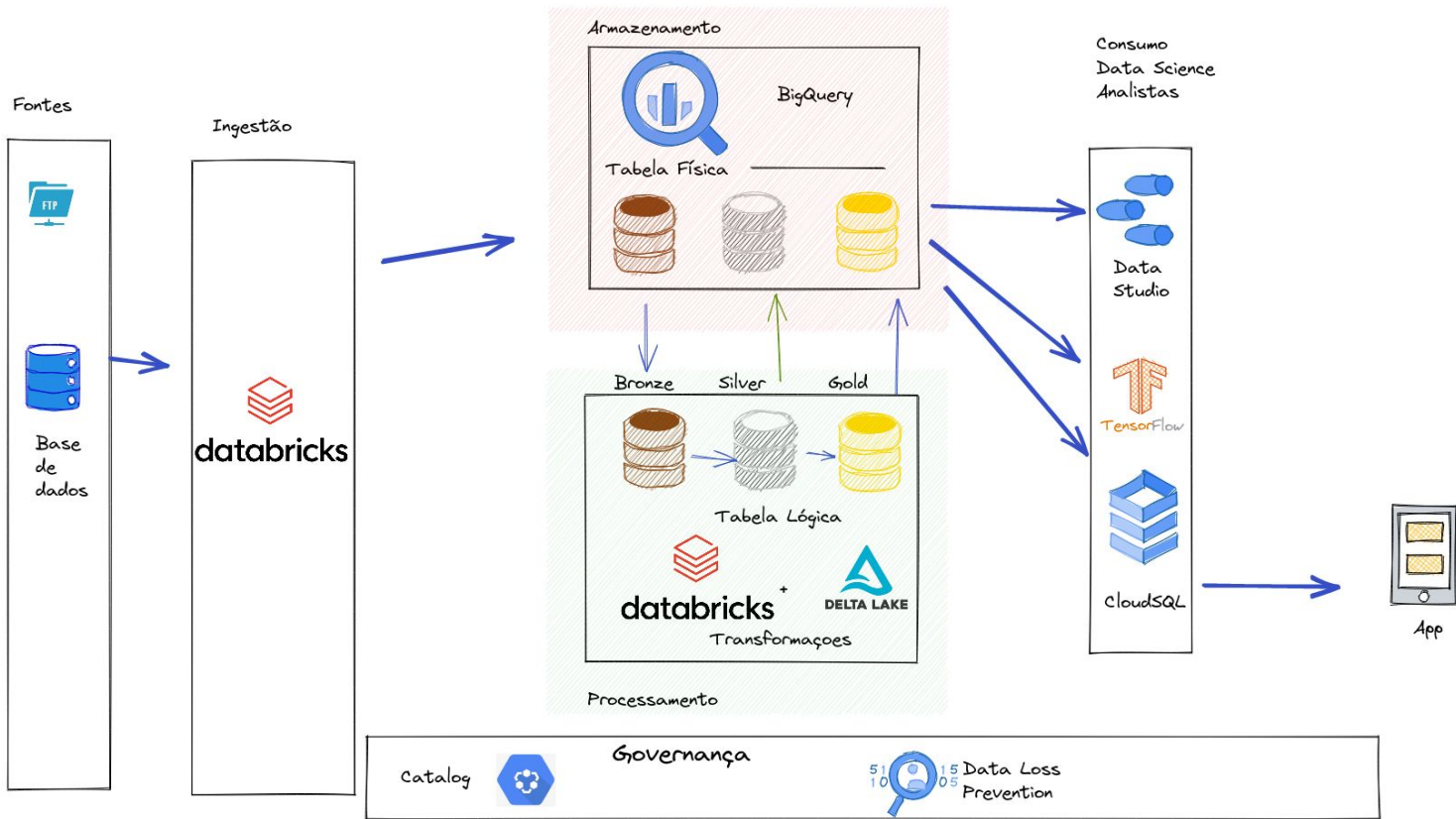


Databricks

- Apache Spark em minutos
- Dimensionamento automático
- Projetos compartilhados em um workspace interativo.
- Compatível com Python, Scala, R, Java e SQL, além de estruturas de ciência de dados e bibliotecas, incluindo TensorFlow, PyTorch e scikit-learn.



Composer Orquestração





Welcome to databricks



Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



Import & Explore Data





Quickly import data, preview its schema, create a table, and query it in a notebook.



Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

-  [New Notebook](#)
-  [Create Table](#)
-  [New Cluster](#)
-  [New Job](#)
-  [New MLflow Experiment](#)
-  [Import Library](#)
-  [Read Documentation](#)


Recents


-  [Quickstart](#)
-  [Feature Store Taxi example noteb...](#)


What's new in v3.54


- [Databricks Status](#)
- [View latest release notes](#)


Databricks Workspace


 **databricks**


 Machine Learning


 Create


 Workspace


 Recents


 Search


 Data


 Compute


 Jobs

 Experiments

 Help

 Settings

 mikael.ahonen@

 Menu options

test-notebook Python

test

Cmd 1

```
1 import pandas as pd
2 df = pd.DataFrame({'Column 1': [11, 21, 31], 'Column 2': [21, 22, 23]})
3 display(df)
```

▶ (3) Spark Jobs

	Column 1	Column 2
1	11	21
2	21	22
3	31	23

Showing all 3 rows.

Command took 0.56 seconds -- by mikael.ahonen@ at on test

Shift+Enter to run

Databricks notebook



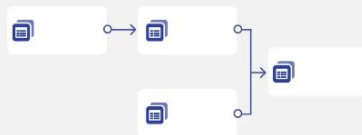
“Transforme seus dados brutos em conjuntos de dados confiáveis, documentados e atualizados.

Dataform é onde sua equipe de dados trabalha em conjunto para construir uma única fonte de verdade para os dados da sua empresa. Colabore em pipelines SQL no BigQuery sem escrever código ou gerenciar infraestrutura.”

```
select
*
from
${ref("table")}
```

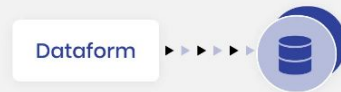
Develop in SQLX

STEP 1



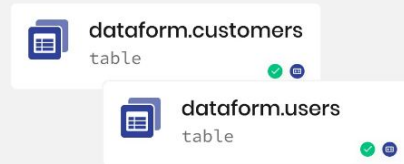
Your project is compiled into SQL
resolving dependencies

STEP 2



Dataform runs your project in your
data warehouse

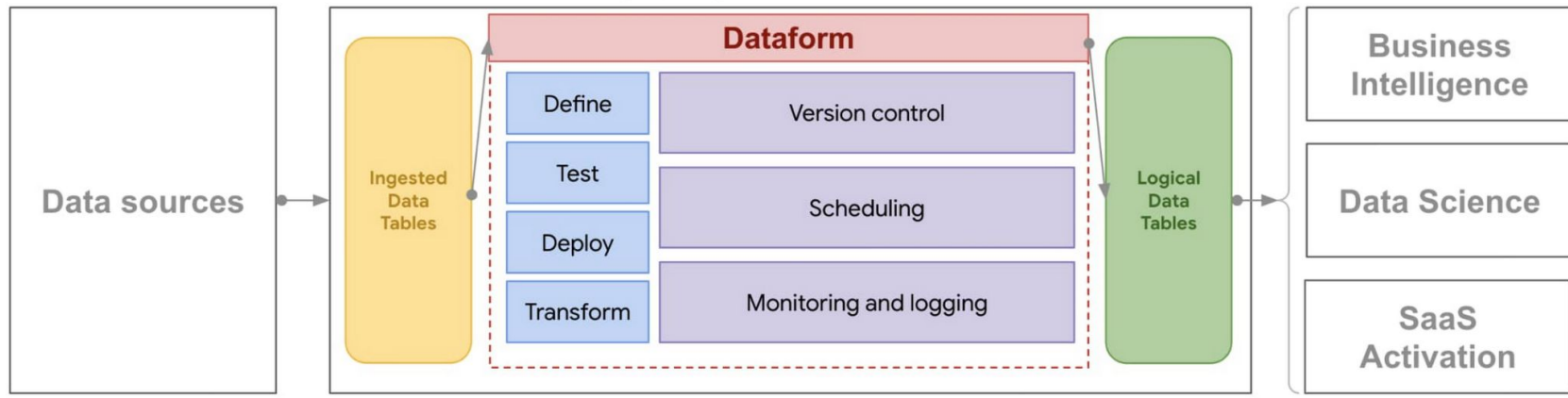
STEP 3

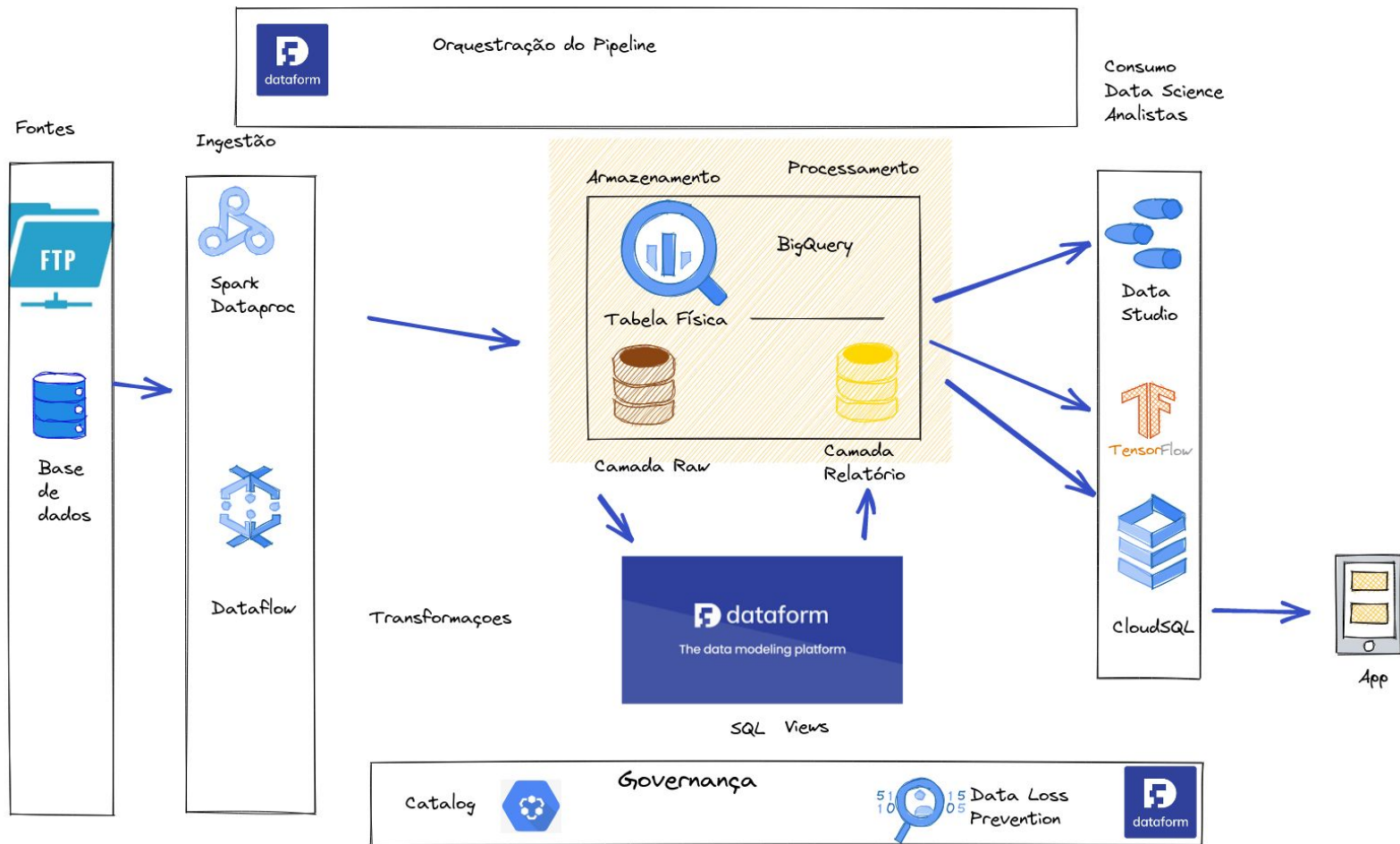


Use updated, tested and documented
tables in your analytics

STEP 4

BigQuery Data Warehouse





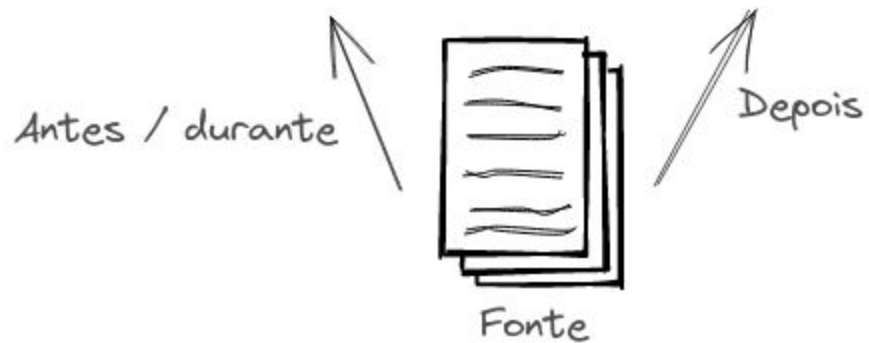
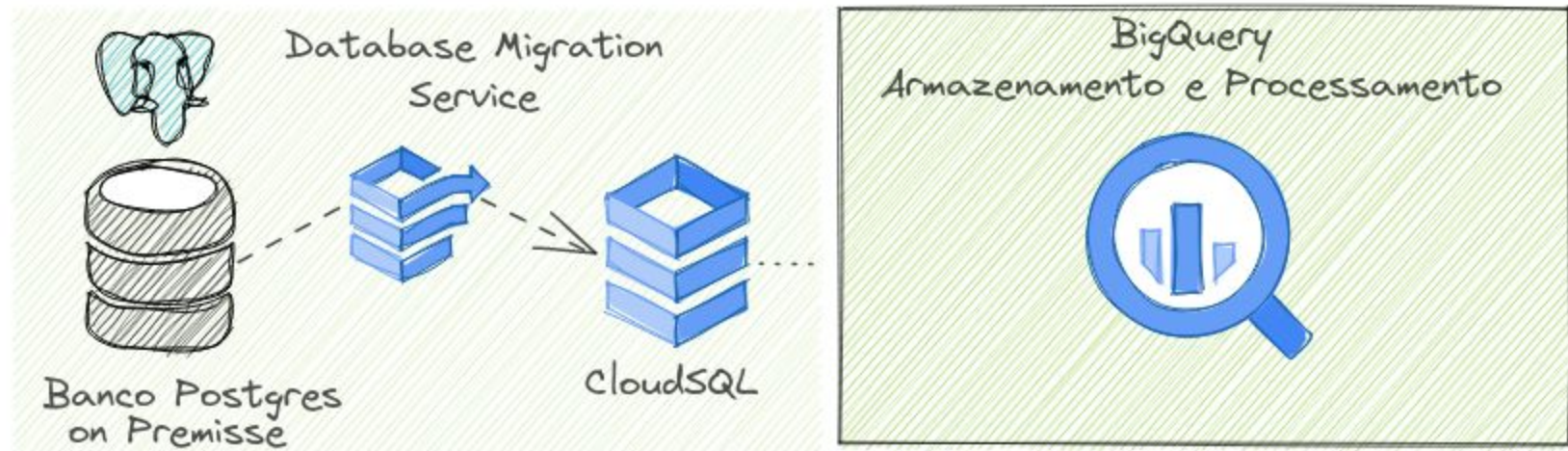


Conclusão:

Por experiência prévia e conhecimento da ferramenta, escolheria a primeira arquitetura. Bigquery + Spark. A documentação é vasta, a integração também, maleável para ajustes finos, comunidade forte, facilidade de encontrar profissionais. A interface do usuário fica a gosto do cliente. Se quiser trabalhar em uma única plataforma o Google Dataproc é ideal. Tem todas as features das tabelas delta e integração amigável com demais produtos google.

Já para um ambiente colaborativo e pipelines em notebooks, organização de pastas e estruturas de leitura de arquivos etc, a plataforma Databricks é a ideal. No entanto, fatalmente terá o trabalho de abrir as duas plataformas na configuração inicial e/ou conferência de algum job.

Sem tirar do radar, em um futuro próximo, quando terminar os estudos e análise da plataforma, migrar para a plataforma dataform. Sua utilização empodera analistas e pessoas não programadoras, centraliza a orquestração e governança e mantém o data lineage, dependências e documentação, democratizando os dados dentro do ambiente corporativo.



Primeiros passos