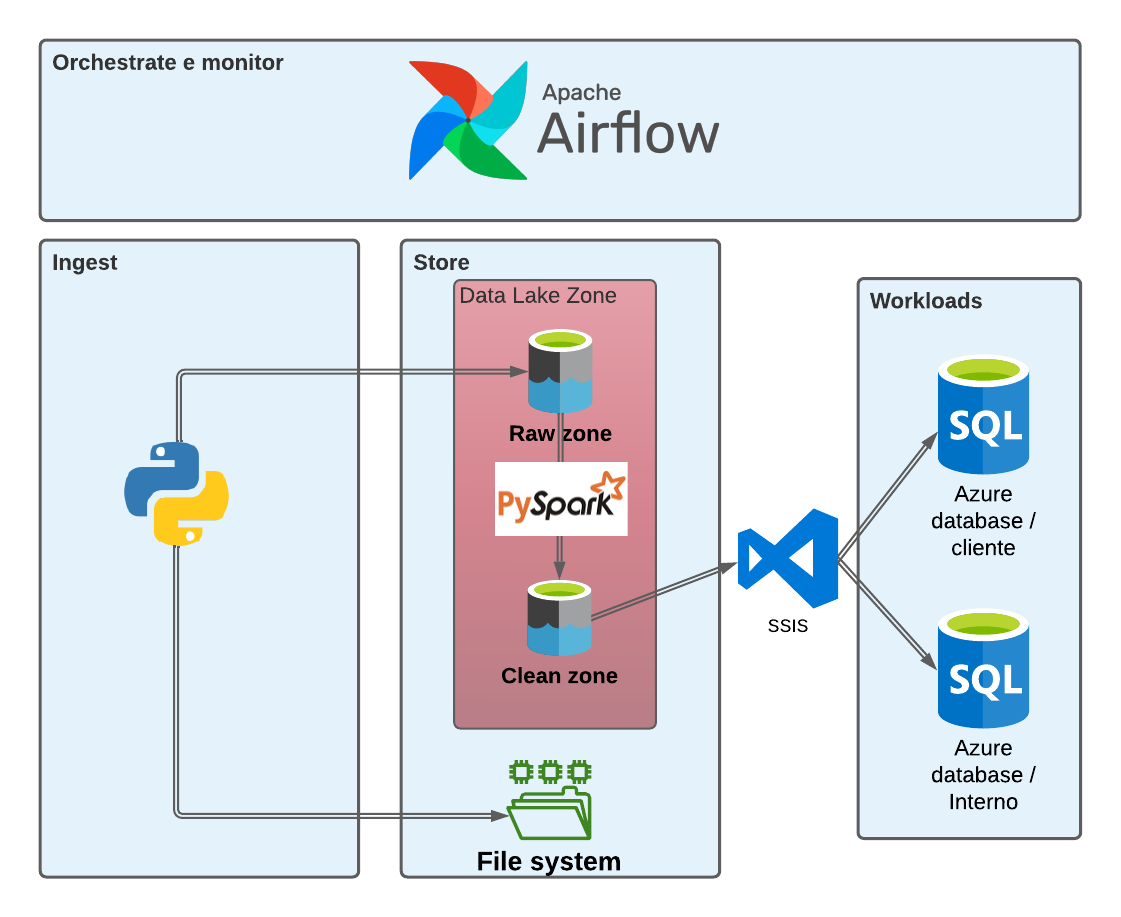
**A – Resposta:**

Estrutura SQL, pois, os dados dos arquivos são estruturados e claramente definidos por seus relacionamentos. As tabelas se aplica o conceito de schema.

**B – resposta**

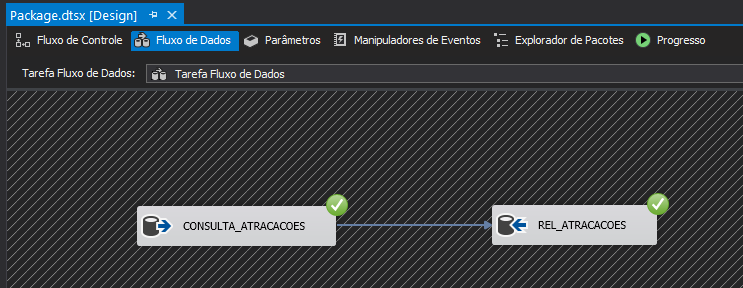
Desenho do projeto:



* Para ingestão dos dados inicialmente, foi desenvolvido um código em Python para estração dos dados dos arquivos e importado dentro do Daka Lake.
  + Arquivo **“main\_ingest.py”**
* O Data Lake é composto basicamente por duas zonas, RAW e CLEAN:
  + RAW – Estágio onde o dado será armazenada no seu formado bruto, conforme vindo da origem. Não receberá nenhum tipo de tratamento.
    - Script das tabelas: **“LAKE\_RAW.TBL\_ATRACACAO.SQL”** e **“LAKE\_RAW.TBL\_CARGA.SQL”**
  + CLEAN – Nesse estágio, os dados são armazenados após receberem os tramentos necessários para limpeza e estruturação dos dados.
    - Script das tabels: **“LAKE\_CLEAN.ATRACACAO\_FATO.SQL”** e **“LAKE\_CLEAN.CARGA\_FATO.SQL”**
* Para realizar a transformação dos dados da RAW ZONE para CLEAN ZONE, foi usado o PySpark. Após o processamento, os dados serão disponibilizados para serem consumido pelo time de cientistas de dados.
  + Arquivo **“main\_transformation.py”**
* Para atender ao time de analistas do cliente e internos da empresa, foi criado ETL’s(Data Flow) no SQL Server Integration Services. Após aplicado as regras de negócio necessárias, os dados serão enviados para um banco de dados na nuvem, para serer consumido pelos analistas.
  + Obs: Foi usado o SQL Server Integration Services, mas poderia ser usado qualquer outra ferramenta para esse fim.

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

Como exemplo, foi gerado a carga da consulta solicitado pelos economistas e disponibilizado no banco de dados com as informações tratadas.

* Para orquestração do Work Flow, foi usado o Apache Airflow.

**C – Resposta:**

Script da consulta **“Query-RES\_C.sql”.**

Uma imagem com mesa

Descrição gerada automaticamente

**Questão Bônus!**

Foi criado a DAG no Apache Airflow.   
Script DAG: **“carga\_antaq.py”**

Uma imagem com texto

Descrição gerada automaticamente

