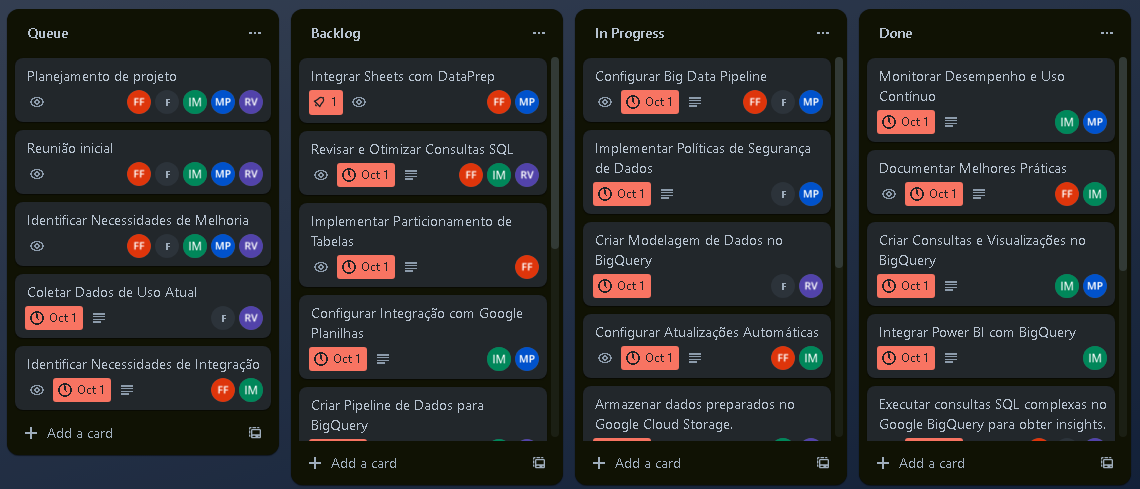
**Problema escolhido:** 6) Aumento do número de reclamações no SAC.

**Trello**

****

O desenvolvimento do projeto, embora tenha sido rápido, seguiu os critérios pré-estabelecidos conforme o planejamento no Kanban (Trello).

o Backlog consiste em lista de pendências para começarmos, In progress, como trabalhos que já estavamos realizando e fomos concluindo. Backlog, um intermediário entre os dois anteriores: definida como prioridades e movidas para In progress assim que possível. Done, como etapas concluidas.

**Fonte de Dados**

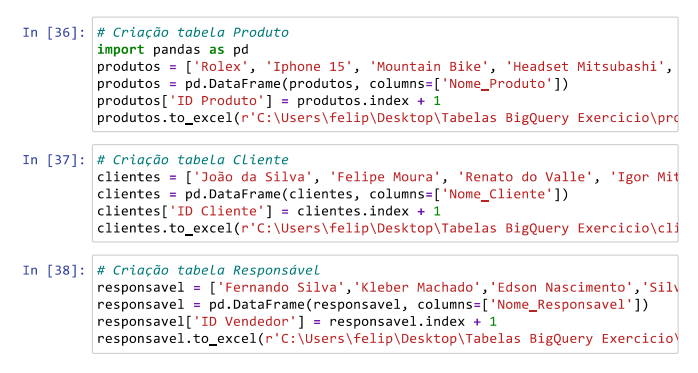
Tabelas utilizadas:

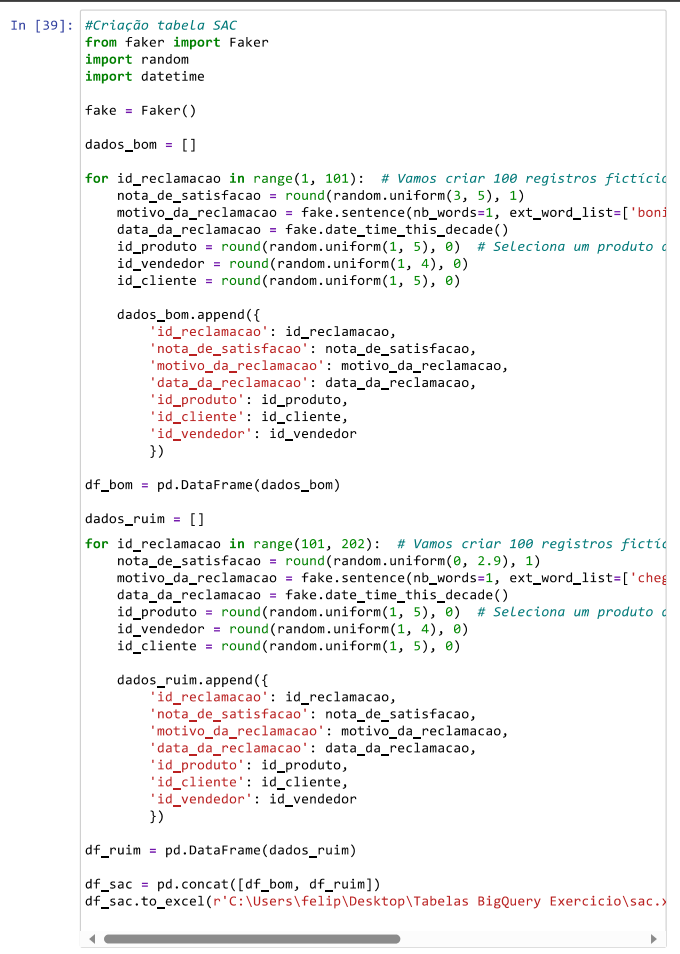
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Fonte de Dados** | **Formato** | **Velocidade** | **Volume** | **Horário de Coleta** | **Localização** | **Proprietário** |
| Produtos.csv | CSV | Diária | Menos de 10 registros únicos de Produtos | 05:00 AM (horário de Brasília) | Servidor interno da empresa | Departamento de Produtos |
| Responsável.csv | CSV | Diária | Menos de 10 registros únicos de Produtos | 05:00 AM (horário de Brasília) | Servidor interno da empresa | Departamento de Recursos Humanos |
| SAC.csv | CSV | Diária | Variável, mas em média 200 reclamações por dia | 05:00 AM (horário de Brasília) | Servidor interno da empresa | Departamento de Atendimento ao Cliente |
| Cliente.csv | CSV | Diária | Menos de 10 registros únicos de Produtos | 05:00 AM (horário de Brasília) | Servidor interno da empresa | Departamento de Engenharia |

Carga Inicial das tabelas:

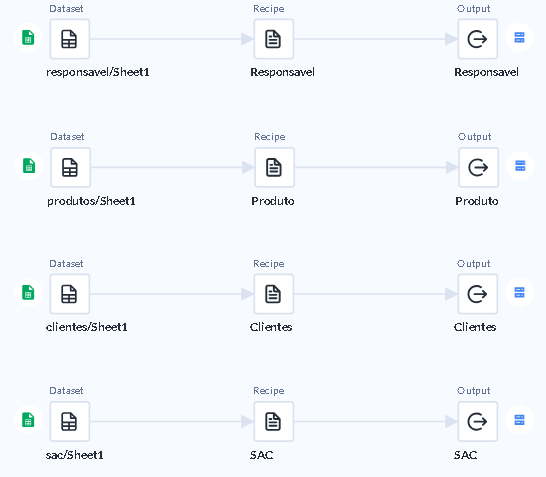
Foi feita uma geração randômica de dados via python com bibliotecas pandas, faker, random e datetime.

Criamos cada uma das tabelas dimensão e, por fim, a tabela fato embasada nessas:

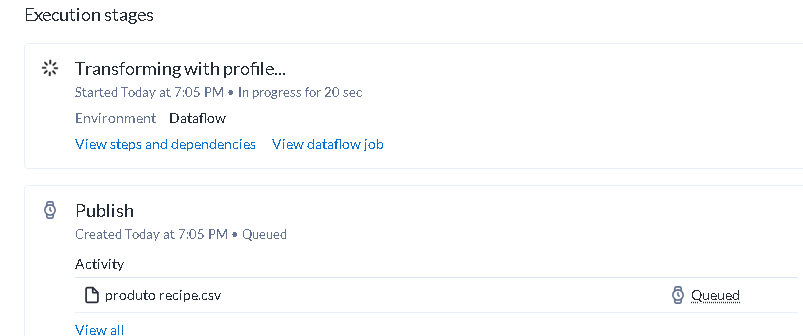
tabela fato (junção de de dois Dataframes, feedback positivos e negativos):

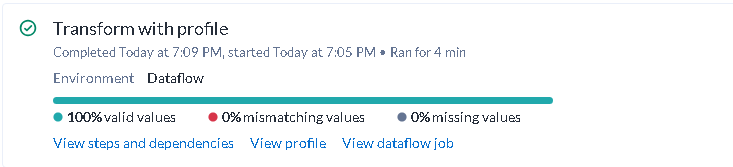


Depois da geração, foram exportadas para o Google Sheets, onde alimenta o Google Data Prep, com alguns pequenos ajustes para remover o index,

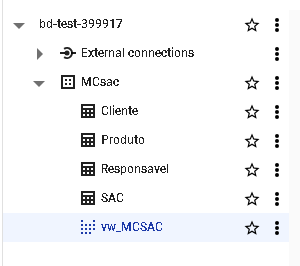


e chega ao Google Data Flow.



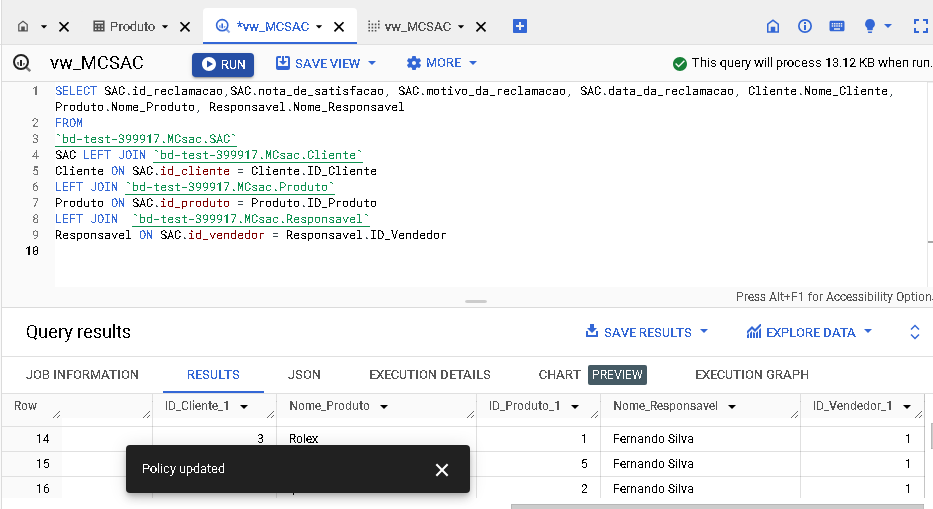


Após esse job criado e agendado para cada tabela, rodamos a atividade para que chegue ao nosso bucket, Google Cloud Storage, que por sua vez alimenta o nosso Data Warehouse, Google BigQuery.

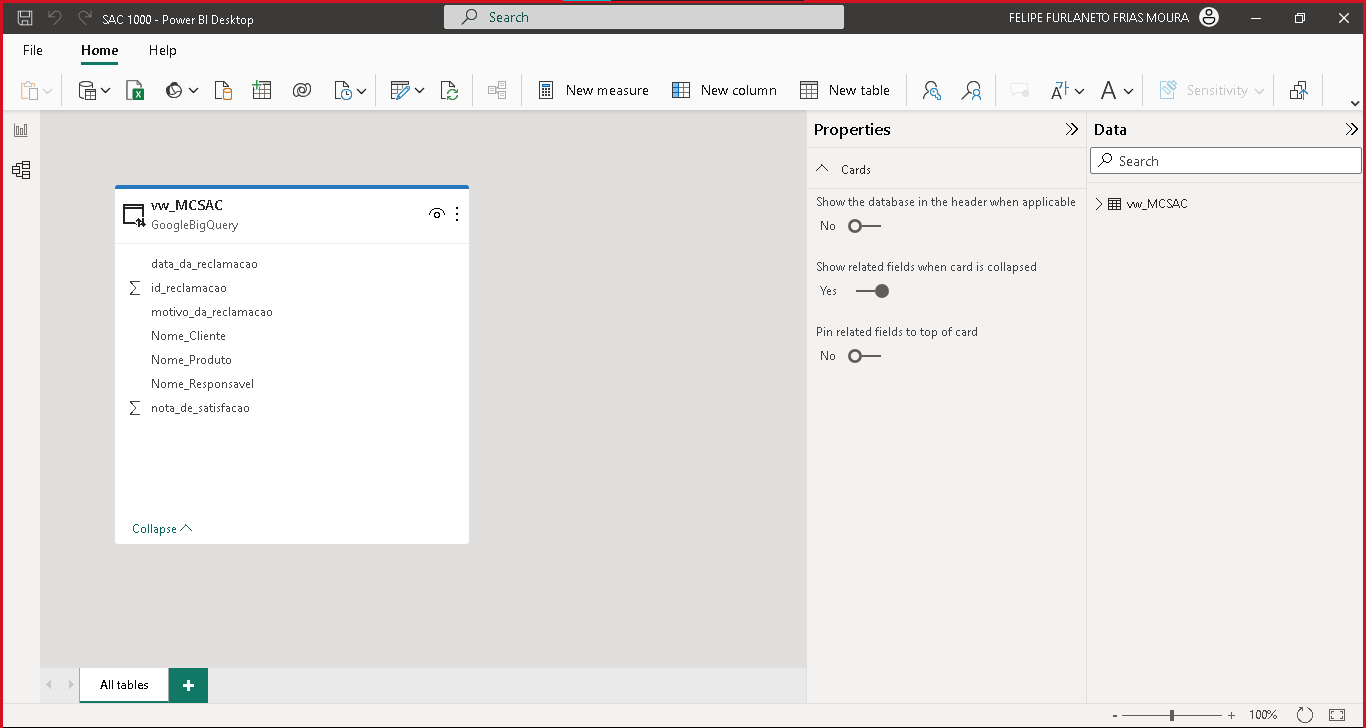


Assim se encerra o ETL da fonte para o nosso Star Schema. Há uma view automática que junta as tabelas e facilita a utilização em ferramentas de Business Intelligence, como o Power BI da Microsoft:

Criação da View

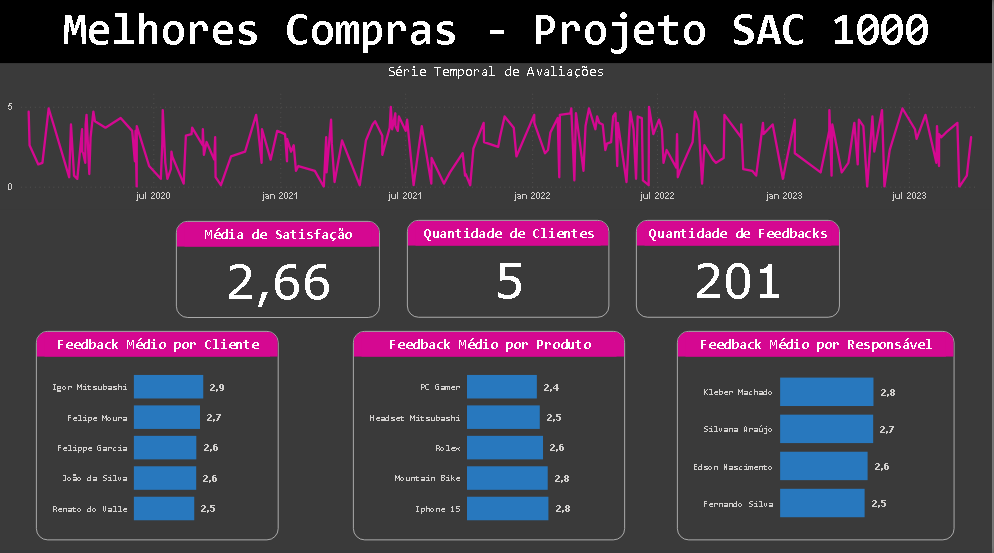


Importação para o Power BI:

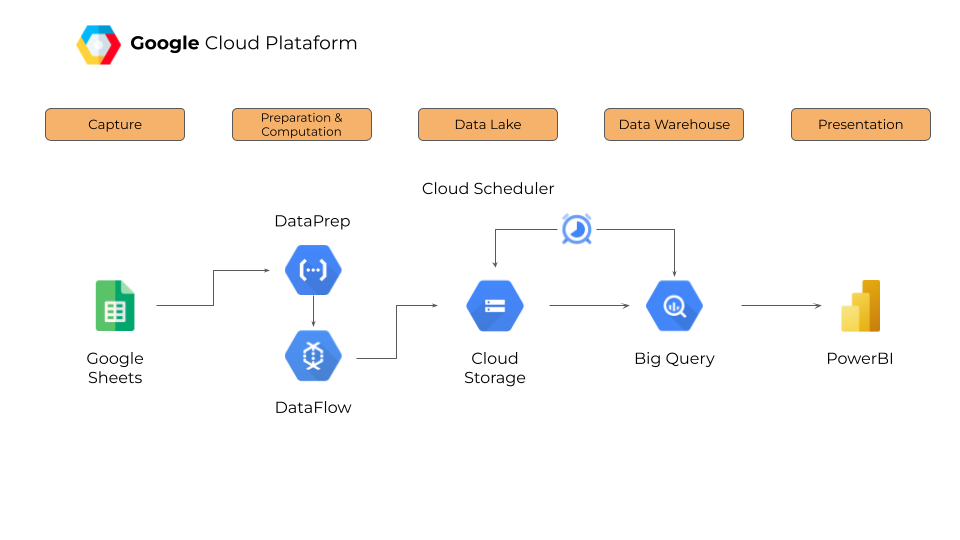


\*o Google BigQuery atrelado já indica atualização automática, basta agendar a atualização no site do Power BI pro e esse processo se automatiza de ponta a ponta.

Chega-se à solução: através dos dados, podemos transformá-los em informação e, consequentemente, decisões. Assim, somos capazes de solucionar os problemas de reclamações no SAC elevadas, do ponto de vista de produto, cliente e responsável:



Big Data GCP - Resumo da Arquitetura e funções individuais



Google Sheets: Inicialmente, usamos o Planilhas Google para coletar dados brutos ou realizar análises exploratórias iniciais de dados. Isto pode incluir a inserção manual de informações ou a importação de dados de fontes externas.

**Google Cloud DataPrep:** Em seguida, após coletar os dados no Planilhas Google, exportamos os dados para o Google Cloud DataPrep. Nesta fase, realizamos tarefas de limpeza, transformação e enriquecimento de dados para garantir que os dados estejam prontos para processamento e análise adicionais.

**Google Cloud Storage:** Os dados preparados no DataPrep são armazenados no Google Cloud Storage. Este é um armazenamento seguro e escalável de dados brutos ou preparados que podemos acessar quando necessário. Os dados no Cloud Storage servirão como fonte de dados para as próximas etapas.

**Google Cloud Dataflow:** Usamos o Google Cloud Dataflow para criar pipelines de processamento de dados. Isso inclui extrair dados do Cloud Storage, fazer transformações adicionais (se necessário) e depois carregar os dados processados ​​em outro lugar, como o BigQuery.

**BigQuery:** Os dados processados ​​pelo Dataflow serão carregados no Google BigQuery, que funciona como um data warehouse altamente escalável. Aqui, executamos consultas SQL complexas em tempo real para obter insights e analisar dados em profundidade. O BigQuery é onde ocorrem nossas consultas e análises de Big Data.

**Power BI:** Por fim, usamos o Power BI para criar painéis interativos e relatórios de visualização de dados. O Power BI pode se conectar diretamente ao BigQuery e levar os resultados de nossas análises para a equipe de negócios de uma forma fácil de entender e envolvente, permitindo-lhes tomar decisões informadas com base nos dados processados ​​e analisados. Este fluxo sequencial nos permite coletar, preparar, armazenar, processar e visualizar nossos dados de maneira eficiente e escalável, permitindo que nossa equipe tome decisões informadas com base em análises avançadas de dados.

**Conclusão:**

Tiramos desse desafio, maior conhecimento sobre a stack do GCP. Foi o nosso segundo ETL e dessa vez, em vez de fazer ingestão direta no lake (Google Cloud Storage) via python, utilizamos o Data Flow com DataPrep, para trazer de fontes como o google sheets. Além de mais contato com o python para gerar as bases randômicas que serviriam de início da nossa solução.

Essa abordagem de Big Data Serverless, que simplifica sem comprometer a potência, foi essencial para a conclusão prática do desafio e a implementação bem-sucedida da arquitetura inicial. Essa experiência fortaleceu nossa confiança, proporcionando uma base sólida para enfrentar desafios reais em Big Data.