Design do Pipeline

1. **Extração**:
    - **Serviços Utilizados**:
        - **Cloud Functions**: Para acionar extrações automáticas de fontes externas via APIs ou bancos de dados, ele funciona executando requisições com Python, SQL e conexões nativas com Google Sheets por tanto é facilmente ajustável e incremental. É facilmente escalável e trabalha com paralelismo e redundância.
        - **Cloud Scheduler:** Vai cuidar das rotinas de extração em intervalos definidos ou em tempo real. Esse módulo a parte permite a criação de rotinas distintas para cada fonte de dado conforme a demanda.
        - **Cloud Monitoring:** Essa ferramenta vai acompanhar o fluxo de extração e controlar tratativas de erro ou anomalias informando o responsável, repetindo o fluxo ou até pulando se necessário.
2. **Carregamento**:
    - **Serviços Utilizados**:
        - **Google Cloud Storage (GCS)**: Os dados extraídos virão em formato Bruto, XML, CSV, Json, quando entrarem no GCS serão catalogados em um bucket exclusivo chamado Dados_bruto e marcados com fonte e data da extração/última atualização.
3. **Transformação:**
    - **Serviços Utilizados**:
        - **Dataflow**: Usaremos pipelines de processamento baseados em Apache Beam ideal para lotes e grandes volumes de dados com ele vamos limpar, alterar, agregar e juntar dados de diferentes fontes. Do Dataflow os dados voltam para o Storage e são guardados em outro Bucket chamado Dados_processados, igualmente catalogados.
4. **Armazenamento e Análise**:
    - **Serviços Utilizados**:
        - **BigQuery**: Os dados processados podem cair diretamente no BigQuery mas também é possível ler do Storage, e abrir em formato de BD para leitura com SQL, ou através de CTEs pré-definidas e com níveis diferentes de hierarquia.
5. **Visualização e Relatórios**:
    - **Serviços Utilizados**:
        - **Looker Studio:** Por fim vamos apresentar os dados com Google Data Studio, por ser mais eficiente no ambiente Google.

Resumo

- **Extração (Cloud Functions/Scheduler) → Carregamento (GCS) → Transformação (Dataflow) → Armazenamento (BigQuery) → Análise (Looker Studio)**

Este design fornece um pipeline escalável, eficiente e flexível, que pode crescer com a demanda por novas análises e relacionando os dados de todas as fontes. Com a modularidade toda dentro do ambiente Google, não será necessário recorrer outras soluções. A principal atenção é quanto ao custo, porque tanto Dataflow quanto o GCS terão valores de processamento, a principal ferramenta para controlar esse crescimento é o Monitoring. Essa arquitetura também permite inserções e transformações usando Python e ferramentas opensource para mitigar custos mas dessa forma perde-se aderência do ambiente Google Cloud.