Arquitetura de Dados POC

A arquitetura foi planejada para prover escalabilidade, performance e flexibilidade. Para tal foi considerado a utilização do Google Cloud Plataform (GCP) porém os conceitos utilizados podem ser aplicados a Clouds similares como por exemplo AWS e Azure.

# 1. Serviços Utilizados

Para a concepção da plataforma foram utilizados os seguintes serviços:

* Google Kubernates - Serviço gerenciado do Google para gerenciamento dos contêineres;
* Google Cloud Storage - Serviço gerenciado do google para armazenamento de arquivos em blobs;
* Google BigQuery - Serviço gerenciado do google big data para armazenamento de dados estruturados e não estruturados;
* Google Cloud Storage mySQL - Serviço gerenciado do google para dados transacionais;
* Google DataPrep - Serviço gerenciado do google para limpeza e manipulação de dados;
* Google DataFlow - Serviço gerenciado do google para transformação de dados em massa utilizando linguagem de programação;
* Google Apigee - Serviço gerenciado do google para desenvolvimento de APIs customizadas;
* Google Composer - Serviço Gerenciado do google para airflow para orquestração de fluxo de dados;
* Google Machine Learning Engine - Serviço gerenciado do google para desenvolvimento Machine Learning;
* Talend Data Integration - Ferramenta de desenvolvimento baseado em ETL da Talend;
* Google Cloud IAM - Serviço do Google para gerenciamento de acesso a plataforma.

# 2. Funcionamento

## 2.1. Ingestão de Dados

Será criado um Job do Talend ETL para cada formato de arquivo e irá rodar dentro de um contêiner gerenciado pelo Kubernates, dessa forma será possível realizar o upload de vários arquivos de uma única vez de forma paralela.

2.2. Data Lake

Os dados capturados (dados brutos) serão enviados para o Google Storage a fim de armazenar todos os arquivos originais recebidos de diversas fontes e depois replicados em tabelas do Google BigQuery, isso irá prover a possibilidade de estudos avançados em questão de qualidade de dados recebidos e auditoria.

Os arquivos originais serão compactados para menor consumo de espaço no Cloud Storage e serão acessados apenas por perfis de acesso elevado.

Na etapa de replicação dos dados no BigQuery algumas medidas serão adotadas para segurança dos dados, dentre elas:

* Anonimização de Nome do Paciente, Pai, Mãe e Dependentes;
* Anonimização de CPF;

## 2.3. Dados Limpos

Os dados brutos irão passar por uma limpeza e consistência de dados utilizando o Google Cloud Data Prep e depois enviados a tabela do Google BigQuery novamente. Nesta etapa será realizada a separação dos dados mestres e aplicado os domínios utilizados pela plataforma. Esses dados serão armazenados no Google Cloud SQL.

Para o Cloud SQL será utilizado uma instância do mySQL.

2.4. Transformação

Uma vez com todos os dados tratados, serão utilizados Jobs do Talend e ou algumas rotinas criadas no Google DataFlow para trabalhar os dados com base nas regras de negócio e realizar as transformações dos dados para indicadores a serem salvos dentro do DataWarehouse no Google BigQuery.

O que definirá se a rotina será escrita no DataFlow ou Talend será a complexidade da regra de negócio. Caso seja escrita utilizando o Talend, seus jobs serão armazenados em containers.

Exemplo de alguns Indicadores a serem gerados pelo DataWarehouse:

* Sinistro por mês;
* Sinistro por prestador;
* Sinistro de internação;
* Sinistro de ambulatório;
* Quantidade de passagens de internação;
* Quantidade de passagens de ambulatório;
* VCMH (Variação do custo médico hospitalar);

2.5. Dado Refinado

Os dados refinados serão armazenados dentro do BigQuery pois é o banco do Google pensado para DataWarehouse. Esse banco possui APIs REST que possibilita a consulta e ingestão de dados por códigos customizados.

2.6. SelfService Data

Com o DataWarehouse pronto, os dados serão disponibilizados através de uma API customizada criada no Google Apigee API, consultados pelo Google BigQuery, acessados através de dashboards em ferramentas analiticas como PowerBI e Tableau ou utilizados nas APIs de Machine Learning.

A plataforma foi pensada para possibilitar que os dados sejam acessados de várias formas criando assim um selfservice de dados.

## 2.7. Orquestração de Dados

Para orquestrar todas as fases de dados será utilizado o apache airflow pelo Google Composer. Com essa ferramenta será determinado quando cada container deve ser chamado iniciado cada fase do processamento de dados.

## 2.8. Segurança

Para comunicação dos Jobs e serviços da plataforma será criado um usuário de serviço no Google IAM que é responsável por controlar e gerenciar os acessos a Cloud do Google.

Os demais usuários que precisam acessar a plataforma serão criados no Google IAM e vinculados a grupos com diferentes níveis de permissão compatíveis a suas atividades.

A princípios são pensados os seguintes grupos:

* Desenvolvedores - Acesso aos componentes de desenvolvimento da plataforma sendo:
  + Google Storage;
  + Google DataFlow;
  + Google DataPrep;
  + Google Composes;
* Analista de Dados - Acesso de leitura e escrita aos componentes de análise de dados da plataforma:
  + Google Cloud SQL;
  + Google BigQuery;
  + Google Storage;
* Consumo de Dados - Acesso de leitura destinado a Cientista de dados e analistas de BI:
  + Google DataLab - Ferramenta para testes de algoritmos de machine learning;
  + Google BigQuery;
* Administradores - Acesso total a plataforma do Google Cloud;

# 3. Pontos Positivos

A plataforma foi pensada com o foco em escalabilidade e por isso está sendo desenvolvida para rodar em containers, dessa forma os containers podem ser criados com uma configuração básica e quando necessário o Kubernates irá subir novos para aumentar a performance do processamento de dados.

A utilização do Talend Data Integration possibilita a baixa utilização de código nos ETLs sendo necessário um conhecimento de negócio e parametrização da ferramenta. Também é possível de forma simples realizar versionamento dos Jobs e documentação através da ferramenta e facilita a manutenção uma vez que é uma ferramenta que apresenta o fluxo de dados de forma visual.

Com o BigQuery armazenando todos os dados brutos, dados limpos e o DataWarehouse não ocorrerá preocupação com Locks em tabelas, performance de escrita e leitura uma vez que este banco foi projetado para uma quantidade de dados gigantesca. Outro ponto positivo se deve ao BigQuery ser um banco de dados colunar e por este motivo as querys terão uma performance maior quando estiverem filtrando as informações para análise.

O Apigee da Google vai permitir a customização de APIs para a plataforma com baixa utilização de código uma vez que possui de vários templates e possui integração com os serviços do google.

# 4. Possíveis Gargalos

O BigQuery é cobrado por armazenamento e transação de dados sendo que seu primeiro 1 terabyte de dados por mês é isento. A falta de conhecimento dos utilizadores da ferramenta podem elevar o custo mensal tornando a plataforma muito cara.

Como o BigQuery não possui chaves primárias e estrangeiras toda a consistência de dados precisa estar nos códigos ETL, dessa forma assumimos que todos dado salvo no BigQuery possui integridade verificada.