

**ALBERT EINSTEIN INSTITUTO ISRAELITA DE ENSINO E PESQUISA**  
**Pós-graduação Lato Sensu, em Data Science e Informática para Área da Saúde**

**MODELO DE EXTRAÇÃO, TRATAMENTO E ANÁLISE DE DADOS PARA  
SISTEMAS HOSPITALARES. ESTUDO DE CASO COM DADOS DE  
RESULTANTES DE EXAMES SOROLÓGICOS REGISTRADOS EM SISTEMA DE  
PRONTUÁRIO ELETRÔNICO.**

Trabalho de Conclusão apresentado ao  
Curso de Data Science e Informática  
para Área da Saúde do Albert Einstein  
Instituto Israelita de Ensino e Pesquisa  
de São Paulo, como requisito parcial à  
obtenção do título de Pós-graduação  
Lato Sensu em Data Science e  
Informática para a Área da Saúde.

São Paulo  
2021

Beatriz Pimenta Mesquita  
Bernardete Cristina da Silva Sanchez  
Cássio Miranda Barbosa  
Denise dos Anjos L de S Campos  
Ernane Jesus Pereira Silva  
Pâmela Guimarães da Costa

**MODELO DE EXTRAÇÃO, TRATAMENTO E ANÁLISE DE DADOS PARA  
SISTEMAS HOSPITALARES. ESTUDO DE CASO COM DADOS DE  
RESULTANTES DE EXAMES SOROLÓGICOS REGISTRADOS EM SISTEMA DE  
PRONTUÁRIO ELETRÔNICO.**

Trabalho de Conclusão apresentado ao  
Curso de Data Science e Informática  
para Área da Saúde do Albert Einstein  
Instituto Israelita de Ensino e Pesquisa  
de São Paulo.

Orientador: Prof. Daniel Salvador

São Paulo  
2021

**ALBERT EINSTEIN INSTITUTO ISRAELITA DE ENSINO E PESQUISA**  
**Pós-graduação Lato Sensu, em Data Science e Informática para Área da Saúde**

Coordenador do Curso de Pós-Graduação em Data Science e Informática para Área da Saúde: Prof. Antonio Valadares Gomes Neto

## **AGRADECIMENTOS**

Primeiramente queremos agradecer a oportunidade da realização e concretização deste curso. Foi uma experiência preciosa e grande aprendizado em todos os temas abordados, as trocas de informações através dos debates em aula.

Vale ressaltar que este ano foi atípico para todos em inúmeros aspectos, consequente à pandemia que enfrentamos. Neste momento, os professores estão se desdobrando, estão mudando suas metodologias, estão se readaptando nesse ambiente virtual, que não está fácil para ninguém, mas estamos admirando muito a perseverança de cada um. E agradecemos a todos os professores, que estão nos motivando a continuar nessa caminhada, a continuar buscando um futuro. E esperamos que em um futuro próximo, possamos comemorar juntos novamente.

Por fim, um agradecimento especial a todos os autores e colegas de turma, que desprenderam esforços e apoio mútuo para a conclusão deste trabalho.

## SUMÁRIO

1.	INTRODUÇÃO .....	1
1.1.	OBJETIVOS .....	2
1.1.1	OBJETIVO GERAL .....	2
1.1.2	OBJETIVOS ESPECÍFICOS .....	2
1.2.	JUSTIFICATIVA .....	2
2.	REFERENCIAL TEÓRICO .....	3
2.1.	EXAMES SOROLÓGICOS LABORATORIAIS .....	3
2.2.	SISTEMAS DE PRONTUÁRIO ELETRÔNICO .....	4
2.3.	ETL (EXTRAÇÃO, TRANSFORMAÇÃO, CARREGAMENTO) .....	5
2.4.	AUTOMATIZAÇÕES DE PROCESSOS PARA TRANSIÇÃO DE DADOS .....	6
2.4.1	BANCOS RELACIONAIS .....	7
2.5.	VISUALIZAÇÕES DE DADOS ( <i>DASHBOARDS</i> ) .....	9
2.5.1.	TIPOS DE DASHOARDS .....	10
2.5.1.1.	DASHOARDS OPERACIONAIS .....	10
2.5.1.2.	TIPOS DE DASHOARDS TÁTICOS .....	10
2.5.1.3.	DASHOARDS ESTRATÉGICOS .....	10
2.6.1	DESING THINKING .....	11
2.6.2	TÉCNICAS PARA CONSTRUÇÃO DE PAINÉIS .....	12
2.6.2.1.	BOAS PRÁTICAS DE DESIGN .....	12
2.6.2.2.	ELEMENTOS VISUAIS .....	13
3.	PROCEDIMENTO METODOLÓGICO .....	15
3.1.	PIPELINE DE DESENVOLVIMENTO DO PROJETO .....	15
3.2.	ETAPA 1: KICKOFF COM O ESPECIALISTA .....	15
3.3.	ETAPA 2: TRATATIVA E AQUISIÇÃO DA BASE DE DADOS .....	16
3.4.	ETAPA 3: ARMAZENAMENTO DOS DADOS .....	28
3.5.	ETAPA 4: DOCUMENTAÇÃO TÉCNICA DO PROJETO .....	38
3.6.	ETAPA 5: CONEXÃO DO DATASET AO POWER BI .....	39
3.7.	ETAPA 6: ACOMODAÇÃO DOS DADOS EM POWER QUERY .....	40
3.8.	ETAPA 7: CONSTRUÇÃO DE DASHBOARD .....	42
4.	CONCLUSÃO .....	47
5.	REFERÊNCIAS BIBLIOGRAFICAS .....	48

## LISTA DE ILUSTRAÇÕES

Figura 1 Arquitetura simplificada de um processo de extração e armazenamento de dados. ....	8
Figura 2 Exemplos de serviços cloud para armazenamento de dados em um ambiente de análise. ....	9
Figura 3 Etapas de desenvolvimento da abordagem Design Thinking. Elaborado por PIMENTA 2021. Fonte: CÔRTEZ et al., 2020. ....	11
Figura 4 Aparelho de ressonância magnética temático. Disponível em < <a href="https://bityli.com/eSKG8N">https://bityli.com/eSKG8N</a> > Acesso em 31 out. 2021. ....	12
Figura 5 Pipeline de planejamento do estudo de caso. ....	15
Figura 6 Terminal Ubuntu instalado pela loja de aplicativos do Windows. ....	17
Figura 7 Ativação da Máquina Virtual pelo PowerShell. ....	17
Figura 8 Solicitação de um usuário e senha para acesso. ....	18
Figura 9 Resultado da atualização dos pacotes Ubuntu. ....	19
Figura 10 Resultado da atualização das variáveis de ambiente. ....	19
Figura 11 Formato final da estrutura de pastas. ....	22
Figura 12 Airflow está sendo executado na máquina local. ....	22
Figura 13 Pipeline do estudo de caso no Airflow. ....	24
Figura 14 Passos para criar uma conta de serviço e posteriormente gerar a chave de acesso. ....	25
Figura 15 Arquivos do fluxo do Airflow arquivados para versionamento. ....	28
Figura 16 Projeto, dataset e tabela raw no Google BigQuery. ....	29
Figura 17 Criando um projeto no BigQuery. ....	29
Figura 18 Criando um conjunto de dados no BigQuery. ....	30
Figura 19 Criando uma tabela de dados no BigQuery. ....	31
Figura 20 Tabela gerada com dados externos das UBS brasileiras. ....	31
Figura 21 Resultado obtido na consulta idealizada para ajuste da tabela UBS. ....	32
Figura 22 Verificação das opções disponíveis de cidade no banco de dados. ....	33
Figura 23 Resultado obtido na consulta idealizada para ajuste da tabela sorologia. ....	34
Figura 24 Detalhes da tabela dashboard criada. ....	37
Figura 25 Pré visualização da tabela dashboards. ....	37
Figura 26 Indicação de local para programação de consultas no BigQuery. ....	38
Figura 27 Fluxo de organização do processo de trabalho. ....	38
Figura 28 Ambiente criado para versionamento dos arquivos gerados no projeto. ....	39
Figura 29 Forma de conexão do BigQuery com o Power BI. ....	40
Figura 30 Demonstração do ambiente conectado (BigQuery x Power BI). ....	40
Figura 31 Selecionando uma tabela do BigQuery para utilização no Power Query. ....	41
Figura 32 Tabela preparada para transformações no Power Query. ....	41
Figura 33 Indicação de método para documentação das etapas de transformação no Power Query. ....	42
Figura 34 Visão geral do dashboard na ferramenta Power BI. ....	43
Figura 35 Visão geral do dashboard desenvolvido para o estudo de caso. ....	43
Figura 36 Visão do dashboard desenvolvida para mobile. ....	44
Figura 37 Gráfico para comportamento dos pacientes na procura pela realização de exames. ....	44
Figura 38 Gráfico do histograma da idade dos pacientes. ....	45
Figura 39 Quadro para cada método de verificação. ....	45
Figura 40 Arvore de ramificação com o caminho do paciente pelo diagnóstico. ....	46
Figura 41 Mapa do volume de exames realizados por cidade. ....	46
Figura 42 Filtros globais do painel. ....	47

## 1. INTRODUÇÃO

Idealizar sistemas de saúde integrados e inteligentes é um desafio que possuem muitos obstáculos, porém é possível de se alcançar e representa o futuro e o caminho da inovação em saúde. Para que essa ideia se torne realidade é preciso cada vez mais tornar natural termos como *big data*, *data driven* e *data analytics*. Para instituições onde o produto é, de fato, a tecnologia, como empresas de aplicativo, sites de compra, *streaming*, entre outras. A aplicação da tecnologia de dados faz parte do produto final a ser entregue e acontece naturalmente. O que é bem diferente em instituições de saúde, onde a atividade principal é o atendimento aos pacientes. Nesse caso, as instituições de saúde precisam passar por um processo de transformação digital, que envolve desde investimento em tecnologia, profissionais capacitados e mudança na cultura das equipes operacionais.

No processo de transformação digital é preciso ter, por exemplo, sistemas informatizados e substituir fichas de papel por prontuários eletrônicos — o que, além de evitar gastos e erros, economiza tempo e disponibiliza dados de fácil acesso. Esse seria um primeiro passo para ingressar no universo digital e conseguir adquirir e armazenar dados que possam ser utilizados e convertidos em valor e conhecimento. Porém, esta ainda é uma realidade muito distante da maioria das instituições de saúde do país.

*“Um retrato da situação atual é que 39% dos estabelecimentos de saúde brasileiros registram informações cadastrais e clínicas dos pacientes exclusivamente em prontuários manuscritos, enquanto apenas 21% das instituições com acesso à internet armazenam os dados em formato eletrônico.”* Medição da Saúde Digital (2019), publicado pela Organização Pan-Americana da Saúde e pelo Núcleo de Informação e Coordenação do Ponto BR (NIC.br).

Neste cenário, espera-se com este estudo, apresentar um modelo de extração, tratativa e análise de dados que exemplifique os passos de operação ao mesmo tempo que seja escalável e seguro, desmitificando a utilização de ambientes de análise e incentivando a utilização dos dados para a tomada de decisões estratégicas e apoio aos processos.

## **1.1. OBJETIVOS**

### **1.1.1 OBJETIVO GERAL**

Aplicação de um estudo de caso utilizando dados de exames laboratoriais devidamente anonimizados de um grande hospital no período de 2017 a 2021. A fim de desenvolver e demonstrar um método eficaz de extração, tratamento e análise de dados que sejam capazes de interagir com sistemas hospitalares e ajudar a implementação de fluxogramas diagnósticos que visam caracterizar com acurácia e precisão amostras biológicas submetidas a testes sorológicos.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

- Demonstrar uma prática automatizada e segura de extração de dados em sistemas;
- Apresentar tratamento dos dados (ETL-Python) a fim de prepará-los para um ambiente relacional;
- Armazenamento dos dados em formato de *datasets* e preparação, em SQL, de tabelas de consumo para painéis de visualização em Power BI;
- Demonstração de painel de visualização de dados, devidamente conectado e seguindo as melhores práticas de construção;
- Boas práticas de documentação e versionamento (GitHub) do processo e suas estruturas.

## **1.2. JUSTIFICATIVA**

Aplicar habilidades relacionadas a ciência e análise de dados, combinado a conhecimentos estatísticos e sistêmicos. Utilizando um estudo de caso relacionado a dados de sorologia, para demonstrar:

- Método de extração de dados - idealizando os dados em um sistema de prontuários eletrônico, de forma a demonstrar as ferramentas e construção do processo;
- Método de tratamento dos dados – demonstrando como os dados podem se tornar de analisáveis e armazenáveis via linguagem de programação;
- Método de análise dos dados – demonstrando o potencial da conversão de dados em conhecimento via ferramentas de construção de painéis;
- Documentação de processo – demonstrar a importância e as facilidades de utilizar ferramentas de versionamento e documentação;



- Pensamento *data driven* – desmitificar o universo de dados, mostrando um *pipeline* de aplicação prático.

## **2. REFERENCIAL TEÓRICO**

Este referencial teórico tem como propósito apresentar os conceitos de negócio necessários para o entendimento do estudo de caso proposto relacionado a exames sorológicos. Além disso, complementa com conhecimentos teóricos imprescindíveis para o desenvolvimento das técnicas propostas no processo metodológico.

### **2.1. EXAMES SOROLÓGICOS LABORATORIAIS**

Os testes sorológicos contribuem para o diagnóstico laboratorial de diversas doenças. Existem vários desafios associados a implementação destes testes em laboratórios clínicos, sejam estruturais ou operacionais. Alguns desafios permanecem constantes: a evolução tecnológica que introduz novas metodologias, e ainda sua aceitação para uso na rotina diária do diagnóstico em diferentes situações e instalações. Resultados indeterminados ou inconclusivos, falso-positivos ou falso negativos, podem ser obtidos com a utilização de qualquer teste ou metodologia, seja devido à limitação da própria metodologia e do que ela é capaz de detectar na amostra analisada, seja pela característica com que a doença pode progredir em cada indivíduo.

Como estratégia para o diagnóstico laboratorial podemos ter dois ou mais testes combinados, com o objetivo de aumentar o valor preditivo positivo de um resultado reagente no teste inicial. O fluxograma em série é lógico e custo-efetivo. O primeiro teste deve ser sempre o mais sensível, seguido por um segundo teste mais específico, a fim de eliminar resultados falso-positivos. É importante selecionar a correta combinação de testes para garantir o diagnóstico preciso.

Atualmente a saúde pública do país dispõem de sistema informatizado, onde todo o histórico relacionado à saúde do usuário é digitalizado – desde a ficha até os exames já realizados – e pode ser acessado por qualquer hospital do país, facilitando o acompanhamento do paciente (GUTIERREZ E CARVALHO, 2019). No Brasil, encontra-se em desenvolvimento, pelo DATASUS, o projeto Rede Nacional de Dados em Saúde (RNDS) com objetivo de criar um Prontuário Único que possibilite a troca de informações entre os diversos pontos da Rede de Atenção à Saúde, favorecendo a

continuidade do cuidado nos setores público e privado, através da interoperabilidade dos diferentes sistemas de informação em saúde utilizados no país (BRASIL, 2019). A fase piloto, em realização no estado de Alagoas, destaca como parte do escopo o uso do PEC na atenção primária e na atenção hospitalar, bem como aqueles estabelecimentos que utilizam o Aplicativo de Gestão para Hospitais Universitários (AGHU). Almeida et al (2016) afirmam que, apesar dos benefícios acarretados pelas inovações tecnológicas, é necessário que estas tragam melhorias que atentem para os diversos aspectos que cercam o indivíduo (éticos, sociais, econômicos e políticos) e que há a necessidade de pesquisas mais consistentes acerca do Prontuário Eletrônico do Paciente (PEP), como intuito de determinar, objetivamente, benefícios esperados, responsabilidades delegadas e riscos assumidos.

## **2.2. SISTEMAS DE PRONTUÁRIO ELETRÔNICO**

Segundo COSTA (2001), os sistemas de Prontuários Eletrônicos do Paciente (PEP), que surgiram na década 1970, nos Estados Unidos. Apresentando cinco diferentes níveis evolutivos, de acordo com Peter Waegemann, presidente do Medical Record Institute:

1. Registro Médico Automatizado;
2. Registro Médico Computadorizado;
3. Registro Médico Eletrônico;
4. Registro Eletrônico do Paciente;
5. Registro Eletrônico de Saúde.

As principais vantagens apontadas, segundo a literatura de base, para os prontuários eletrônicos são, dentre tantas, melhorar o acesso, ampliar a segurança e acrescentar novos recursos, de modo que sua implantação possa se justificar pela melhoria na qualidade da assistência à saúde do paciente. Isso acontece e é facilmente justificável observando as melhorias de processo desde o gerenciamento dos recursos até pela melhoria de processos administrativos e financeiros. Como desvantagens, podemos elencar principalmente o custo de implantação, tempo necessário para se avaliar os resultados (necessário time de especialistas) e sujeição a falhas operacionais (COSTA, 2001). O Registro Eletrônico de Saúde (EHR) definido como o registro computadorizado dos dados clínicos do paciente, foi implementado no Centro Hospitalar de Porto, Portugal. O EHR torna possível uma análise transversal dos dados de saúde em diferentes serviços de saúde que disponham de tecnologia e recursos

computacionais compatíveis. Além disso, pode agregar informações clínicas, administrativas e financeiras; e pode ser customizado conforme a necessidade ou perfil do usuário. O principal objetivo, é incentivo inicial, foi substituir um grande volume de documentos físicos por eletrônicos, melhorar o processamento de dados e reduzir custos; possibilitando prestar uma assistência mais efetiva, mais rápida e de melhor qualidade (PEREIRA et al, 2013).

Sendo assim, os sistemas de prontuário eletrônico (exemplos práticos: Tasy e Cerner) exercem papel fundamental na operação tecnológica de um hospital, sendo ele a principal fonte para coleta de dados e interação com os usuários.

### **2.3. ETL (EXTRAÇÃO, TRANSFORMAÇÃO, CARREGAMENTO)**

Do inglês *extract, transforms and load* (ETL) refere-se ao processo de extrair, transformar e carregar dados integrados provenientes de diferentes fontes de dados, reunindo-os em um repositório de dados a fim de oferecer suporte à descoberta, relatório, análise ou tomada de decisão (What is ETL?, 2021).

Esta prática surgiu como uma estratégia para simplificar a análise de dados armazenados em um banco de dados, sendo um processo altamente eficiente no quesito integração de dados, estabelecendo regras de otimização e manipulação padronizada dos mesmos a fim de facilitar sua inserção em ferramentas ou ambientes integrados (MJV TEAM, 2021).

O ETL permite que as empresas centralizem seus dados consolidados em um ambiente integrado, geralmente um *Data Warehouse* (DW) ou Data Mart (DM) podendo ser aplicado em banco de dados mais simples como SQL ou até mesmo em sistemas mais complexos, como nuvem de *Big Data* (MJV TEAM, 2021).

O processo de ETL está presente em qualquer trabalho de manuseio de dados e é considerado uma fase crítica da estratégia de utilização dos mesmos, visto que é o processo responsável por garantir a confiabilidade dos dados brutos que se transformarão em informações aplicáveis na área de negócio (IBM CLOUD EDUCATION, 2021).

O processo de extração, transformação e carga de dados é fundamental para qualquer trabalho de inteligência de dados, principalmente na integração de dados de origens distintas e no tratamento deles sob parâmetros qualitativos. O ETL é considerado o processo que confere a “inteligência” ao processo de *Business Intelligence* - BI, e é o processo que define as regras de exploração dos dados e condução dos

mesmos ao ambiente final.

Por meio deste cruzamento de dados, o ETL permite uma melhor compreensão dos mesmos, tendo em vista a geração de informações que agreguem valor ao negócio (MJV TEAM, 2021).

O processo de ETL segue a sequência de três etapas lineares que envolvem o tratamento de dados: extração, transformação e carregamento.

Extração: é a coleta de dados dos sistemas de origem (*Data Sources* ou sistemas operacionais), extraíndo-os e transferindo-os para um ambiente de transição temporária (*staging* área) onde serão organizados e convertidos para um formato único. Os times de gerenciamento de dados podem extraí-los de uma variedade distinta de fonte de dados, as quais podem ser estruturadas ou não.

Transformação: após a coleta, definição e formatação dos dados, a próxima etapa é a transformação, que pode ser entendida como a categorização e segmentação dos dados. Esta etapa deve atender a alguns critérios como limpeza, ajustes, (incluindo o tratamento das duplicidades), padronização, validação e a autenticação dos dados a fim de corrigir as imprecisões e inconsistências com o objetivo de consolidar as informações obtidas.

Carga de dados: consiste em estruturar os dados para que sejam lidos nos *staging* áreas e enviados para o ambiente de armazenamento escolhidos, seja um DW ou um *Datamart* (DM). Esse carregamento deve ser feito de forma que a informação seja mantida organizada, mapeada e acessível. O upload deve ser realizado conforme as necessidades da área de negócio, e deve ser discutida a periodicidade de reposição de dados a fim de evitar diminuição da performance dos sistemas da operação (MJV TEAM, 2021).

## **2.4.AUTOMATIZAÇÕES DE PROCESSOS PARA TRANSIÇÃO DE DADOS**

Para que um ambiente de dados seja escalável e vivo é necessário que dados sejam gerados, como em sistemas de prontuários eletrônicos por exemplo, e que sejam transferidos e armazenados em um ambiente onde possam ser analisados, como um banco de dados relacional. Imagine uma pessoa fazendo download dos dados em um sistema de prontuário eletrônico e posteriormente fazendo o upload desses dados em um banco de dados para análise, uma vez ou outra pode acontecer, mas e quando você desejar ter a visão de um indicador de 30 em 30 minutos, é inviável e impossível

ter uma pessoa fazendo esse processo manualmente. Nessa situação, será necessário recorrer a sistemas que façam esse trabalho automaticamente, ou seja, que orchestrem essas manobras de tempos em tempos. Para isso temos plataformas muito úteis que exercem esta função em larga escala. Neste projeto apresentaremos como solução a ferramenta Apache Airflow, extremamente difundida nas arquiteturas de dados das grandes empresas do mercado.

O Apache Airflow foi iniciado no Airbnb como código aberto desde o primeiro *commit*. A comunidade tem cerca de 500 membros ativos que se apoiam mutuamente na resolução de problemas. O objetivo da plataforma é criar, programar e monitorar fluxos de trabalho de maneira programática. Sendo extremamente escalável com uma arquitetura modular e usa uma fila de mensagens para orquestrar um número arbitrário de trabalhos. Além disso, os pipelines do Airflow são definidos em Python, permitindo a geração dinâmica do pipeline e facilidade de implementação, onde qualquer pessoa com conhecimento em Python pode implantar um fluxo de trabalho. O Apache Airflow não limita o escopo de seus pipelines, podendo ser utilizados para criar modelos de ML, transferir dados, gerenciar uma infraestrutura entre outras funções recorrentes. (AIRFLOW.APACHE, 2021).

Utilizando o Airflow é possível estabelecer um pipeline que movimenta os dados periodicamente do seu sistema para o seu banco de dados no ambiente de análise.

#### **2.4.1 BANCOS RELACIONAIS**

Um banco de dados relacional é um tipo de banco de dados que armazena e fornece acesso a pontos de observações relacionados entre si. São baseados no modelo relacional, um ramo da teoria dos conjuntos algébricos conhecido como álgebra relacional, uma maneira intuitiva e direta de representar dados em tabelas como linhas e colunas. Em um banco de dados relacional, cada linha na tabela é um registro com um ID (identificador) exclusivo. As colunas da tabela contêm atributos dos dados e cada registro geralmente tem um valor para cada atributo, facilitando o estabelecimento das relações entre os pontos de dados. A linguagem padrão dos Bancos de Dados Relacionais é a Structured Query Language, ou simplesmente SQL, como é mais conhecida. Isso os torna o tipo de banco muito versátil para consultas e análises, uma vez que possibilita os usuários utilizarem uma grande variedade de abordagens no tratamento das informações via consultas em SQL.

Estes bancos são a base dos ambientes de análises, elas compoem o ecossistema da arquitetura dos dados de forma essecial. Onde temos, conforme figura 1, desde a coleda dos dados em diversos *data sourses*, passando pela oceano de dados raiz *data lake*, onde todos os dados, estruturados ou não, são despejados. Posteriormente, temos a presença dos processos de ETL minerando os dados no *data lake*, estruturando e organizando para finalmente direcionar estes dados a um banco relacional de análise, com governância de dados e modelos *schemas* para otimização de consultas, podendo ser um *data warehouse* quando centralizado ou *data marts* quando sub categorias.

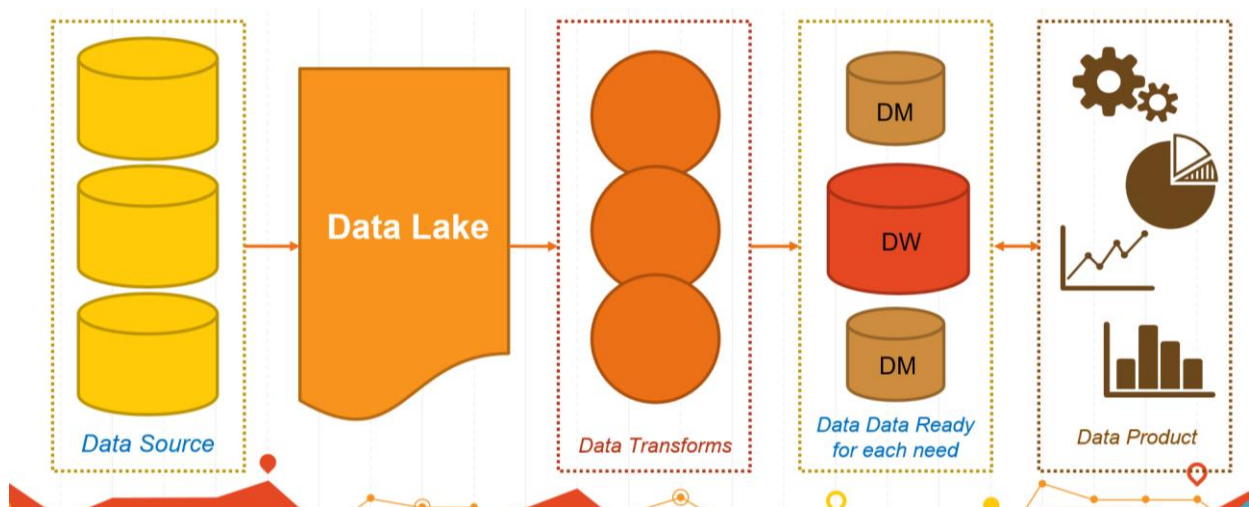
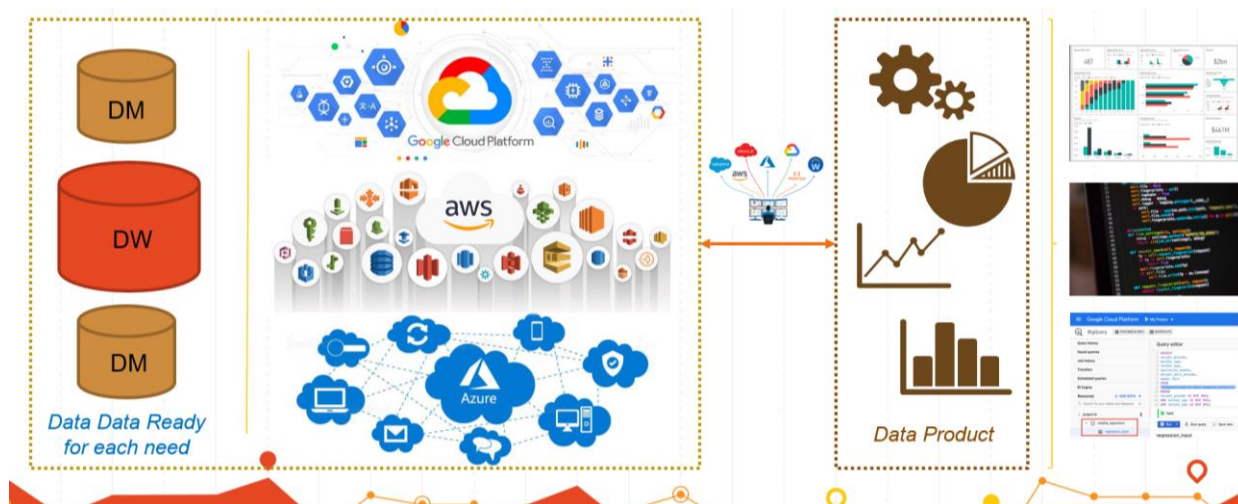


Figura 1 Arquitetura simplificada de um processo de extração e armazenamento de dados.

Na figura 2, temos como exemplo os tres grandes provedores de serviços comerciais de armazenamento em nuvem no formado de banco relacionais. Sendo o Google, com a sua plataforma GCP (Google Cloud Plataform) contendo o serviço Big Query. A Amazon AWS, com serviços como o Athena e o Redshift e a Microsoft, com o serviço Azure. Essas são as empresas que, hoje, dominam o mercado de ambientes de análises, pela sua praticidade, escalabilidade e integração com outras diversas ferramentas no mesmo ambiente que potencializam os processos.



*Figura 2 Exemplos de serviços cloud para armazenamento de dados em um ambiente de análise.*

## 2.5. VISUALIZAÇÕES DE DADOS (DASHBOARDS)

Um dashboard também pode ser chamado de Digital Cockpit (Few, 2013) ou Enterprise Digital Dashboard (Ganesh & Anand, 2005; Kerzner, 2013), embora o termo dashboard seja mais comum. Atualmente, ainda não existe um consenso absoluto na definição de dashboard, embora as definições mais aceites sejam semelhantes. Segundo Few (2013), um dashboard é um display digital visual da informação mais importante, necessária para alcançar um ou mais objetivos, consolidada num único ecrã para que a informação possa ser monitorizada num relance. Eckerson (2011), por sua vez, define dashboard como um sistema digital de entrega de informação por camadas que destaca individualmente a informação, conhecimento e alertas para os utilizadores conforme as suas necessidades, para que eles possam medir, monitorizar e gerir o desempenho da forma mais eficaz. Já Kerzner (2013) refere que os dashboards são ferramentas de software de comunicação para fornecer informação à audiência num relance e o seu objetivo é apresentar a informação certa, à pessoa certa, na hora certa, utilizando a aplicação correta, com uma boa relação custo benefício.

Segundo Few (2013), um dashboard deve estar contido num único ecrã, isto é, toda a informação deve estar disponível no campo de vista do utilizador para que esta possa ser consultada num relance, sem que este tenha que fazer scroll (rolagem) ou alternar entre vários ecrãs. Eckerson (2011) e Kerzner (2013), reconhecem que a possibilidade de colocar o dashboard num único ecrã é importante, mas não é determinante para a própria definição de dashboard.

### **2.5.1. TIPOS DE DASHOARDS**

Os dashboards podem ser utilizados em diversas áreas e setores de atividade e nos vários níveis hierárquicos de uma empresa servindo, tanto para auxiliar nas decisões estratégicas e monitorizar as operações diárias de uma equipa, como para gerir as tarefas individuais (Few, 2013). Apesar de todos os dashboards serem diferentes em termos de display e de funcionalidades, Eckerson (2009) refere que, com base nos requisitos dos utilizadores e na sua posição na hierarquia da empresa, existem três tipos de dashboards: operacionais, táticos e estratégicos.

#### **2.5.1.1. DASHOARDS OPERACIONAIS**

Os dashboards operacionais são concebidos para os colaboradores de primeira linha monitorizarem e controlarem os processos e operações, numa base diária. Estes dashboards têm informação detalhada, recolhida frequente e diretamente dos sistemas operacionais. São dashboards mais orientados para a ação do que os outros dois tipos. Têm um uso intensivo de alertas acerca de situações que excedem determinados limites e alguns até podem tomar decisões automatizadas para pequenos problemas com a solução já conhecida. A maioria dos indicadores presentes só tem significado em termos operacionais, mas alguns têm implicações diretas nos resultados de indicadores de níveis superiores (Eckerson, 2011).

#### **2.5.1.2. TIPOS DE DASHOARDS TÁTICOS**

Os dashboards táticos são concebidos para auxiliar os gestores de departamento ou de outra posição intermédia a monitorizar, gerir e otimizar o desempenho das pessoas e processos sob a sua supervisão. É o tipo de dashboard mais comum. A informação que contêm é uma combinação de dados atuais e históricos, resumidos e detalhados, recolhida numa base diária ou semanal de sistemas operacionais e bases de dados, para que os utilizadores possam identificar problemas e gerar soluções de forma a atingir os objetivos. Estes dashboards devem permitir explorar os dados através das diversas dimensões, atributos e hierarquias organizacionais para verificar as causas de determinadas ocorrências. Podem também incluir ferramentas de análise multidimensional, relatórios, análise what-if e modelação estatística (Eckerson, 2011).

#### **2.5.1.3. DASHOARDS ESTRATÉGICOS**



Os dashboards estratégicos são concebidos para permitir aos gestores de topo monitorizar a execução dos objetivos estratégicos, gerir e comunicar o desempenho e orientar novos comportamentos ou otimizá-los ao longo da empresa. O seu foco é na gestão da empresa, nos objetivos futuros e como os atingir. Normalmente a informação que suporta os indicadores de desempenho não existe nos sistemas informáticos e tem que ser colocada manualmente no dashboard, frequentemente concebidos em Microsoft Excel ou PowerPoint. Estes dashboards devem ainda permitir inserir comentários relativamente à informação apresentada e servir como um guia para as reuniões e discussões estratégicas (Eckerson, 2011).

### 2.6.1 DESING THINKING

De acordo com Brown (2020), o *Design Thinking* deve ser entendido como uma abordagem utilizada para encontrar soluções para problemas complexos, com enfoque nas necessidades humanas.

O termo *Design Thinking* ficou mais conhecido quando David Kelley e Tim Brown, designers e fundadores da IDEO começaram a aplicar a abordagem em diferentes empresas e projetos em 1991 (BONINI; SBRAGIA, 2011). IDEO é uma empresa transnacional de *design* e consultoria em inovação, fundada em Palo Alto, California, em 1991. De acordo com o website institucional da companhia:

*[...] “design thinking é uma abordagem centrada no ser humano para a inovação, que se baseia no modo de pensar do designer para integrar as necessidades das pessoas, como possibilidades da tecnologia e os requisitos para o sucesso do negócio”*

Embora existam diversos modelos de processo para o Design Thinking, David M. Kelley propõe três etapas esquematizadas na figura 3.



Figura 3 Etapas de desenvolvimento da abordagem Design Thinking. Elaborado por PIMENTA 2021. Fonte: CÔRTES et al., 2020.

A inspiração é uma etapa em que se analisa e observa o problema a ser trabalhado e que motivou a busca por soluções. Na etapa de ideação constrói-se hipóteses e protótipos. Por fim, há a etapa de implementação, quando a solução é validada para o problema proposto (CÔRTEZ et al., 2020).

O *Design Thinking* tem muito potencial na área da saúde, sendo utilizado para melhorar processos e serviços oferecidos pelas instituições de saúde, colocando o paciente no centro do processo. Como exemplo, é possível citar a atuação de Doug Dietz, designer da GE Healthcare. Dietz identificou que muitas crianças tinham medo de realizar exames de ressonância magnética por conta da aparência da máquina. Assim, a equipe do designer desenvolveu equipamentos coloridos e temáticos (figura 2) como solução (HES UNIVERSIDADE, 2021).



*Figura 4 Aparelho de ressonância magnética temático. Disponível em < <https://bityli.com/eSKG8N>> Acesso em 31 out. 2021.*

## **2.6.2 TÉCNICAS PARA CONSTRUÇÃO DE PAINÉIS**

### **2.6.2.1 BOAS PRÁTICAS DE DESIGN**

Para qualquer meio de comunicação existe um conjunto de boas práticas que permite tirar o máximo partido do mesmo. No caso dos dashboards, diversos autores referem que essas boas práticas foram definidas por Few (2013) (Gonzalez, 2008; Gemignani, 2009; Eckerson, 2011; Kerzner, 2013). As boas práticas de Few (2013) para o design de dashboards são:

1. Organizar a informação de forma a suportar o seu significado e utilização:

estruturar os itens do dashboard de forma a potenciar a sua utilidade, organizando-os de acordo com o processo a monitorizar ou como as situações são geridas, agrupando atividades relacionadas ou que necessitem de ser analisadas em conjunto, e não induzir à comparação itens que não devem ser comparados.

2. Manter consistência, permitindo uma interpretação rápida e precisa: se algo tem o mesmo significado ou função no dashboard, deve ter uma aparência semelhante. Não se deve variar só por variar. O layout do dashboard não se deve alterar, apenas a informação nele contida.

3. Colocar ao alcance a informação suplementar: um único ecrã pode não conter toda a informação que permite tomar decisões e agir. Nesses casos é necessário fazer a ligação de forma prática e subtil para a informação que o permite fazer.

4. Evitar alertas excessivos: se o dashboard criar demasiados alertas, alguns serão ignorados; como tal devem-se reservar os alertas só para situações que requeiram resposta e atenção imediatas.

#### **2.6.2.2 ELEMENTOS VISUAIS**

Cor: a cor é uma combinação de três propriedades: tonalidade, saturação e luminosidade. A cor não é vista isoladamente, mas de forma global considerando o contexto, e na sua utilização devem-se variar as três propriedades e não apenas a tonalidade. A cor deve ser utilizada criteriosamente, para realçar a informação mais importante do dashboard, porque é facilmente distinguida do resto. A sua utilização excessiva fará com que este efeito se perca, por isso a cor deve ser utilizada com moderação. Muitas vezes, uma marca adicionada ou a utilização de outro atributo pode ter o mesmo efeito, sem sobrecarregar o dashboard de cores.

Forma e tamanho: o cérebro humano consegue distinguir facilmente formas simples, como quadrados e círculos, e distinguir variações no comprimento e largura de linhas e retângulos. Isto é especialmente útil para gráficos de barras. O tamanho relativo dos objetos também pode ser utilizado para destacar elementos e normalmente é percecionado como proporcional à sua importância.

Zonas do dashboard: é importante posicionar a informação mais importante para onde as pessoas mais olham. Gemignani (2009) refere que as pessoas tendem a examinar um ecrã de forma semelhante, olhando primeiro para o canto superior esquerdo e dando também atenção considerável ao lado esquerdo e centro do ecrã.

Espaço em branco: na conceção do display o espaço em branco é

extremamente importante e é, muitas vezes, negligenciado. É importante criar espaços para os olhos descansarem, para que as zonas coloridas tenham mais impacto. Quando não há espaçamento suficiente, não conseguimos ver o que é mais importante. O espaço em branco pode ser utilizado para delinear secções. A sua utilização pode significar sacrificar um gráfico extra, mas faz uma diferença enorme para a compreensão por parte do utilizador (Gemignani, 2009).

Disposição dos elementos: segundo Eckerson (2011), a forma como os objetos estão posicionados no dashboard conta uma história e tem um significado, logo deve ser cuidadosamente planeada. Também se devem agrupar elementos relacionados ou que necessitem de ser comparados ou analisados em conjunto.

Os gráficos também devem ser flexíveis e facilmente alteráveis conforme as necessidades do utilizador (Few, 2013). Os elementos gráficos mais comuns de dashboards são:

Gráficos de barras: utilizados para apresentar simultaneamente várias instâncias de uma medida. São a melhor forma de apresentar medições associadas com itens discretos, ao longo de diversos tipos de escalas e intervalos. Permitem comparações fáceis entre os valores adjacentes (Few, 2013).

Bullet charts: variante do gráfico de barras, inventado por Few (2013), que representa uma única barra, horizontal ou vertical com uma escala, limites e targets associados. A reduzida dimensão e a sua capacidade de fornecer muita informação, permitem apresentar de forma eficaz uma grande quantidade de informação num pequeno espaço (Few, 2013).

Gráficos de linhas: gráficos extremamente úteis para evidenciar a forma e a tendência de séries de valores individuais, conectando-os entre si. Permitem uma leitura rápida dos valores e a sua evolução (Few, 2013).

Gráficos circulares: tipo de gráficos muito comuns em dashboards para apresentar quantitativamente diversas partes de um todo. Apesar de comuns, Few (2013) refere que não são muito eficazes.

Sparklines: variação simplificada dos gráficos de linhas, intencionalmente desprovida dos eixos e escala. Úteis para perceber apenas a tendência dos valores históricos e não o seu valor (Few, 2013).

Skeuomorphs ou widgets são versões digitais de instrumentos mecânicos de medição como medidores de pressão, velocímetros, termómetros, entre outros. (Johar, 2010). Ocupam muito espaço, apresentam pouca informação e nem sempre são

fáceis de entender (Gonzalez, 2005b).

### 3. PROCEDIMENTO METODOLÓGICO

#### 3.1. PIPELINE DE DESENVOLVIMENTO DO PROJETO

Para este estudo de caso, o objetivo é demonstrar todo o processo de coleta, tratamento, análise e transformação do dado em informação. Com isso, idealizamos o pipeline conforme figura 4, onde iniciamos por uma reunião de *kickoff* com especialistas da área, passamos pelo processo técnico de coleta e preaparação do dado e finalização com a apresentação de indicadores e métricas para solucionar as dores apresentadas pelos *stakeholders* de interesse.

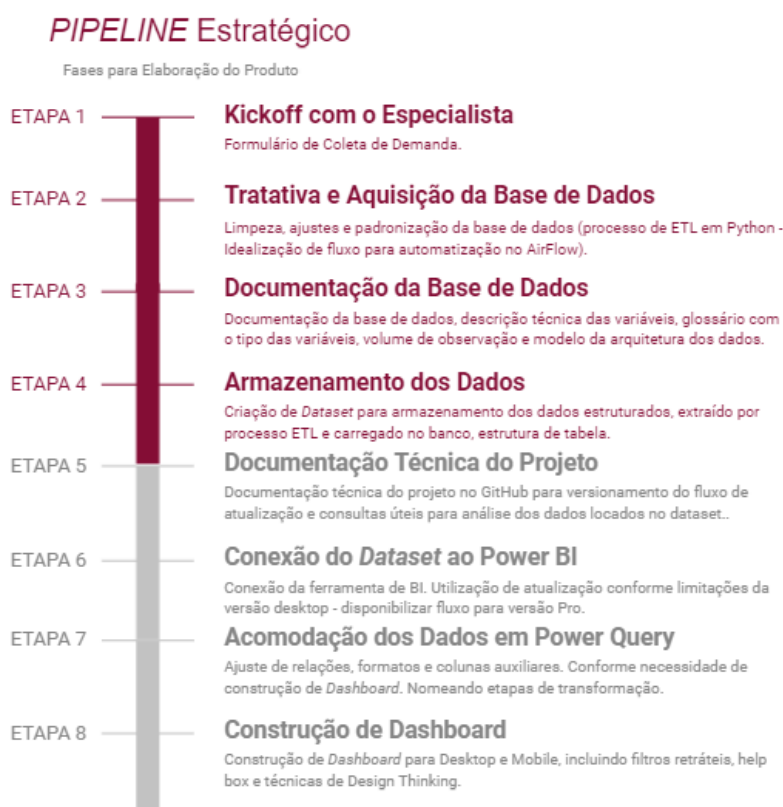


Figura 5 Pipeline de planejamento do estudo de caso.

#### 3.2. ETAPA 1: KICKOFF COM O ESPECIALISTA

Realizados entrevista com os stakeholders com o objetivo de entendermos melhor como é o dia a dia deles, problemas que enfrentam, como fazem para resolvê-los e principais melhorias do processo. Na entrevista levantamos os seguintes pontos:

1. Me conte um pouco sobre você... Há quanto tempo atua no Hospital?
2. Quais são as suas atividades principais?

3. Descreva a sua jornada de trabalho.
4. Como você faz para validar uma nova metodologia no laboratório clínico?
5. Quais são as dores/dificuldades da sua especialidade hoje?
6. Quais destas dores trazem maior impacto para a performance da sua especialidade?
7. Você tem problemas de sistemas de informações de dados hoje?
8. Para você qual o propósito de um painel de dados atualizados?
9. Quais métricas você tem hoje que espera mudar com este painel?
10. Qual funcionalidade seria espetacular para você?

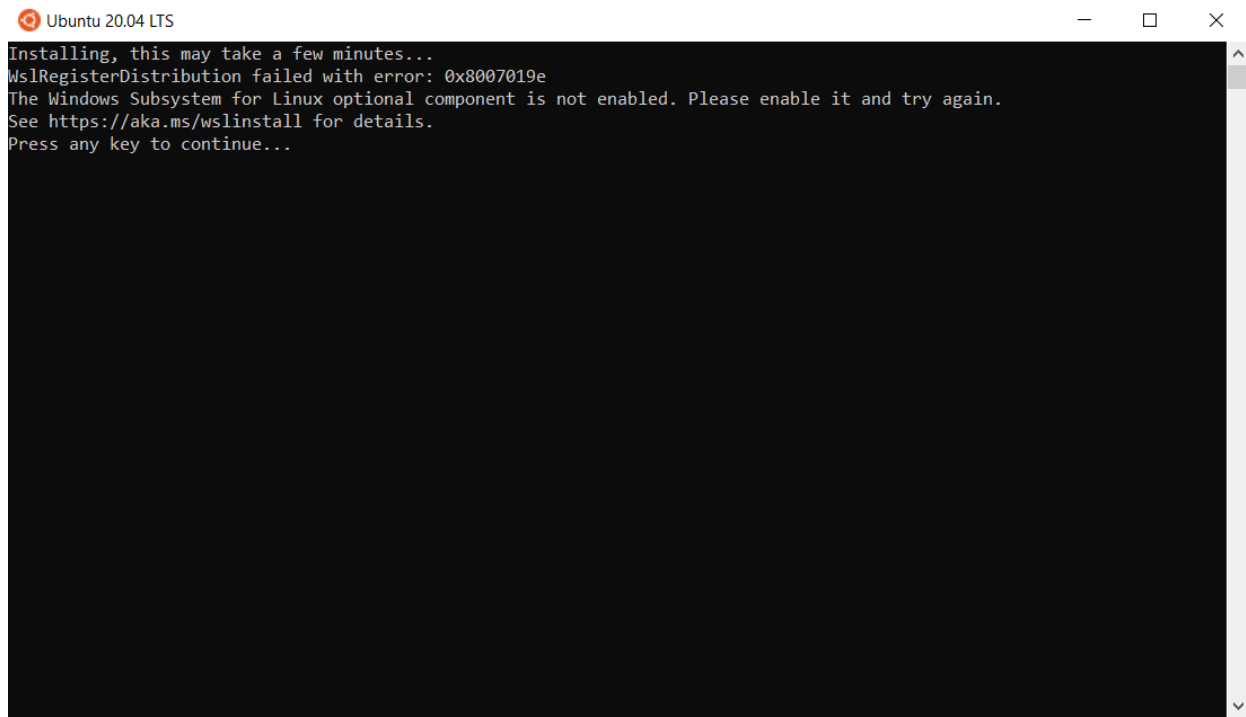
Com base nestas informações obtemos insumos e conhecimento necessário para o desenvolvimento das próximas etapas.

### **3.3. ETAPA 2: TRATATIVA E AQUISIÇÃO DA BASE DE DADOS**

Em um ambiente de produção o Airflow é hospedado em um serviço em nuvem como GCP (Google Cloud Platform) e AWS (Amazon Web Services), podendo ser ou não gerenciado. Para este estudo de caso, iremos simular o Airflow em uma máquina local, passando pelos passos de instalação, configuração e aplicação.

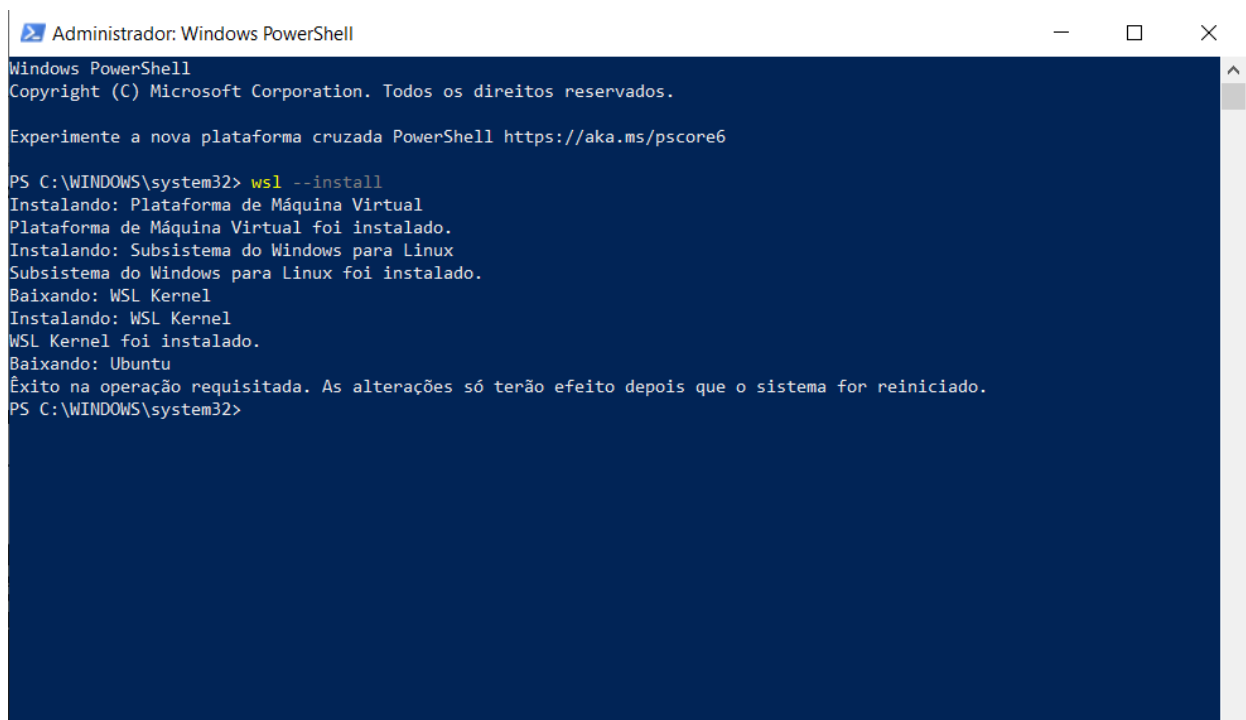
#### *1º Passo: Instalação e Configuração*

Utilizando uma máquina Windows, pode-se fazer as configurações utilizando um terminal Ubuntu. Primeiro, é necessário a instalação do terminal Ubuntu (figura 6), que pode ser feito pela própria loja de aplicativos do Windows.



*Figura 6 Terminal Ubuntu instalado pela loja de aplicativos do Windows.*

Na sequência, é necessário ativar uma máquina virtual via PowerShell (figura 7), executando como administrador, e reiniciar a máquina.



*Figura 7 Ativação da Máquina Virtual pelo PowerShell.*

Após reiniciado, o terminal Ubuntu será iniciado automaticamente e será solicitado um usuário e senha para acesso.

```
3@DESKTOP-7JVADVD: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: pamela.costa
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX[_SYSTEM] configuration variable. Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
Enter new UNIX username:
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat

System load:  0.91          Processes:            8
Usage of /:   0.4% of 250.98GB Users logged in:       0
Memory usage: 1%          IPv4 address for eth0:
Swap usage:   0%

0 updates can be installed immediately.
0 of these updates are security updates.
```

*Figura 8 Solicitação de um usuário e senha para acesso.*

Agora, é necessário atualizar os pacotes Ubuntu e na sequência atualizando algumas variáveis de ambiente, utilizando os comandos:

```
sudo su
cd ~
sudo apt-get update
```



```
root@DESKTOP-7JVADVD: ~
Get:19 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [4948 B]
Get:20 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [540 B]
Get:21 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:22 http://archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:23 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:24 http://archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:25 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:26 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1344 kB]
Get:27 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [276 kB]
Get:28 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.5 kB]
Get:29 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [569 kB]
Get:30 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [81.6 kB]
Get:31 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [528 B]
Get:32 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [876 kB]
Get:33 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [190 kB]
Get:34 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [19.5 kB]
Get:35 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.5 kB]
Get:36 http://archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [6856 B]
Get:37 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [616 B]
Get:38 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2568 B]
Get:39 http://archive.ubuntu.com/ubuntu focal-backports/main Translation-en [1120 B]
Get:40 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [400 B]
Get:41 http://archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:42 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6588 B]
Get:43 http://archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [3292 B]
Get:44 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [580 B]
Get:45 http://archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 22.4 MB in 6s (3478 kB/s)
Reading package lists... Done
root@DESKTOP-7JVADVD:~#
```

Figura 9 Resultado da atualização dos pacotes Ubuntu.

```
export SLUGIFY_USES_TEXT_UNIDECODING=yes
export LC_ALL="en_US.UTF-8"
export LC_CTYPE="en_US.UTF-8"
sudo dpkg-reconfigure locales
```

```
root@DESKTOP-7JVADVD: ~
Get:45 http://archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 22.4 MB in 6s (3478 kB/s)
Reading package lists... Done
root@DESKTOP-7JVADVD:~# export SLUGIFY_USER_TEXT_UNIDECODING=yes
root@DESKTOP-7JVADVD:~# export LC_ALL="en_US.UTF-8"
export: command not found
root@DESKTOP-7JVADVD:~# export LC_ALL="en_US.UTF-8"
root@DESKTOP-7JVADVD:~# export LC_CTYPE="en_US.UTF-8"
root@DESKTOP-7JVADVD:~# sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  en_US.UTF-8... done
Generation complete.
root@DESKTOP-7JVADVD:~#
```

Figura 10 Resultado da atualização das variáveis de ambiente.

Com o terminal preparado, seguimos com a instalação do Docker Engine para rodar uma imagem do Airflow, facilitando as configurações. Para tanto, é necessário configurar o repositório Docker. Depois disso, você pode instalar e atualizar o Docker a partir do repositório. Primeiro, instalaremos os pacotes para permitir o uso de um repositório por HTTPS.

```
sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

Adicionando a chave GPG oficial do Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Por fim, utilizamos o comando a seguir para configurar o repositório estável.

```
echo \
    "deb [arch=$(dpkg --print-architecture) signed by=/usr/share/keyrings/docker-archive-
    keyring.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Com o repositório preparado, seguimos com a instalação do Docker.

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Após instalado, basta iniciar o Docker para utilização:

```
sudo /etc/init.d/docker start
```

Por conta de problemas de acesso dentro de imagens docker pré-criadas, foi necessário criar uma imagem customizada, representada nos arquivos Dockerfile e docker-compose.yml. Isso é comum quando se deseja enviar dados e fazer conexões com personalizadas. Para isso, foi desenvolvido:

Dockerfile: Este é o arquivo que deve ser executado para criar a imagem e executar alguns comandos iniciais. Na primeira linha FROM é possível notar que estamos customizando uma imagem já existente (com airflow e postgresql já configurados), mas precisamos também de acesso root a esse sistema, assim como algumas bibliotecas do google (para conexão com o Google BigQuery) citadas no código abaixo.

```
FROM puckel/docker-airflow

USER root

RUN apt-get update -yqq \
    && pip install --upgrade google-api-python-client google-auth-http2 google-auth-oauthlib

RUN pip install --upgrade pandas-gbq 'google-cloud-bigquery[bqstorage,pandas]'

RUN pip install --upgrade ipython
```

Docker-compose: Este arquivo será encarregado de dizer qual arquivo Dockerfile executar, assim como definir qual é a pasta volume do nosso projeto.

```
services:
  web:
    build: .
    ports:
      - "8080:8080"
    volumes:
      - ./dags:/usr/local/airflow/dags
```

A pasta volume é extremamente importante para conectarmos uma pasta local à uma pasta dentro do container, uma vez que estamos utilizando, para este estudo de caso, um arquivo local. Isso é necessário porque a memória do container é volátil, então para não perdermos as mudanças feitas, precisamos que esse comando seja feito. Ele é composto pelo caminho da pasta dags (ainda vazia e pode ser criada manualmente) seguido do caminho padrão dessa imagem para guardar as dags. {caminho\_local\_dags}:{caminho\_padrao}.

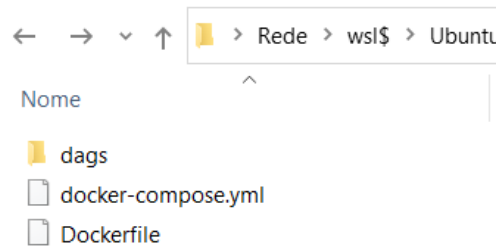


Figura 11 Formato final da estrutura de pastas.

Por fim, com estes últimos ajustes, o Airflow está sendo executado na máquina local e para visitar a IU basta acessar o endereço `http://localhost:8080/admin/`

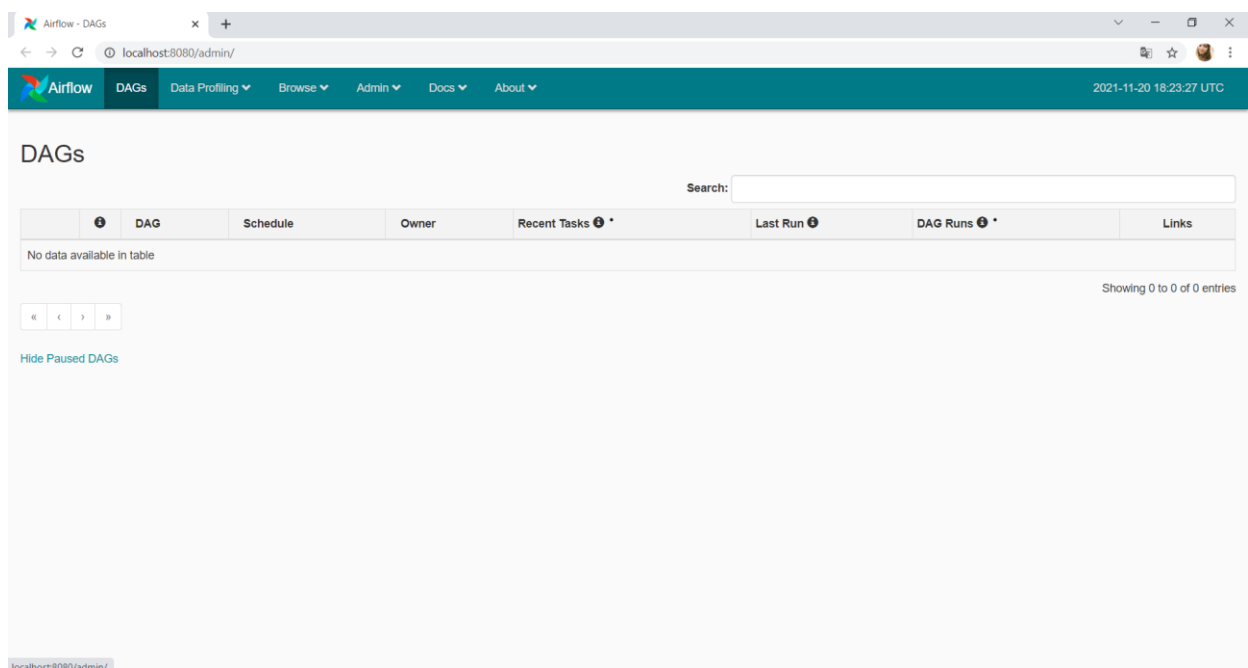


Figura 12 Airflow está sendo executado na máquina local.

## 2º Passo: Aplicação

Agora que já temos um Airflow funcional, precisamos adicionar o código que será executado na pasta DAGs, e também ligar o *scheduler* do Airflow, que será o encarregado de colocar todas as execuções de todos os códigos na fila.

As DAGs (*Directed Acyclic Graph*) representa um dos conceitos centrais do Airflow. Elas reúnem tarefas que são organizadas com dependências e relacionamentos para dizer como elas devem ser executadas. Na prática, podemos representar uma DAG como um código Python que tem operadores que irão executar em uma certa sequência definida. Já o *scheduler* monitora todas as tarefas e DAGs e, a seguir, aciona as instâncias de tarefa quando suas dependências são concluídas. Nos bastidores, o

planejador ativa um subprocesso, que monitora e permanece em sincronia com todos os DAGs no diretório DAG especificado. Uma vez por minuto, por padrão, o planejador coleta os resultados da análise do DAG e verifica se alguma tarefa ativa pode ser disparada. (AIRFLOW.DOC, 2021)

Para iniciar o *scheduler* precisamos primeiro descobrir qual o ID do container sendo executado:

```
docker ps
```

Replicando o nome do ID para entrar no container utilizando o comando:

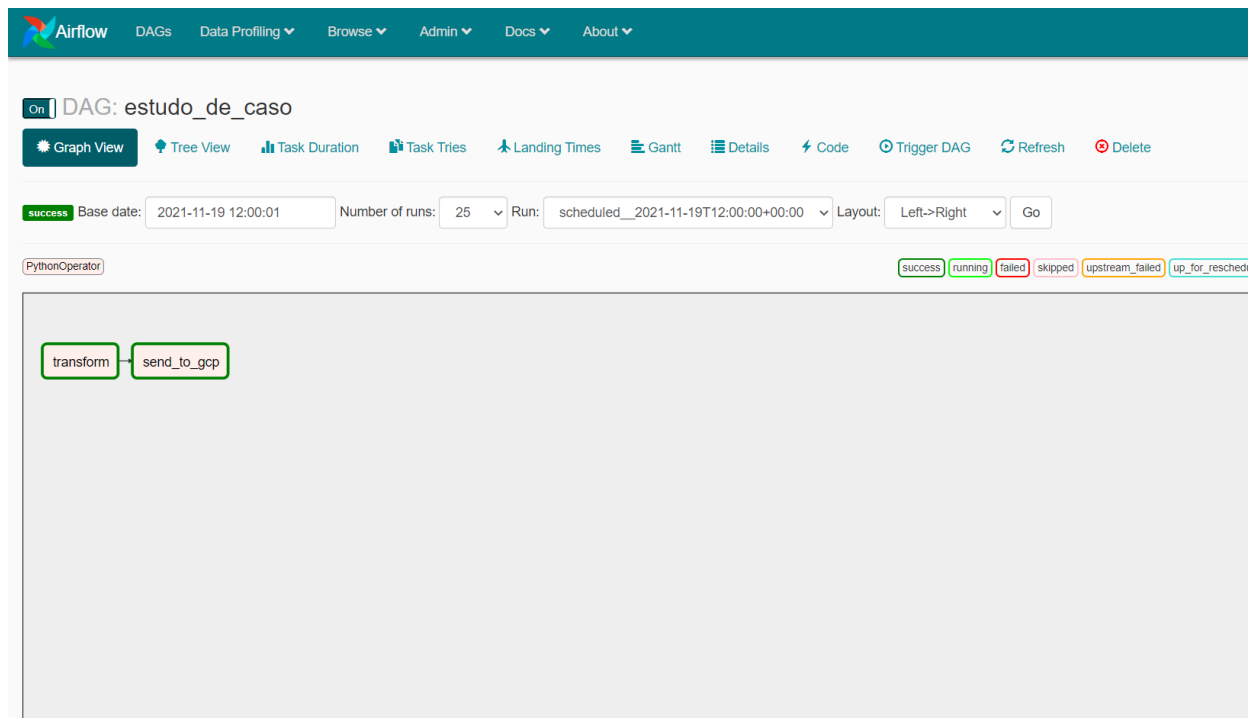
```
docker exec -ti ***** bash
```

Com isso, estaremos no espaço do container e já podemos iniciar o *scheduler* em segundo plano através do comando:

```
nohup airflow scheduler &
```

Agora, é necessário criar o pipeline com os códigos em Python nas pastas dags e configurar a recorrência.

No pipeline do estudo de caso temos dois steps, um com o código em Python do ETL desenvolvido. O qual já faz a leitura do arquivo de dados locais, o que pode ser feito de diversas formas, como via endpoint de uma aplicação, leitura de banco de dados de produção entre outras interfaces. Como é o caso de um sistema de prontuário eletrônico, onde é possível a coleta dos dados em seu banco de produção, geralmente Oracle. A utilização da linguagem Python possibilita esta variabilidade e versatilidade no acesso as informações.



*Figura 13 Pipeline do estudo de caso no Airflow.*

A etapa “*transform*” contém o código em Python desenvolvido na etapa de ETL. Onde é realizada uma limpeza prévia dos valores incorretos e ajuste no tipo de variável. Nesta etapa pode ser feito tanto o quanto necessário para tornar seus dados possíveis de serem carregados em um banco relacional.

Já a etapa “*send\_to\_gcp*” possui o código responsável pelo envio das informações coletadas e transformadas para o banco de dados de análise (*Google BigQuery*) desenvolvido no item 3.4.

Para estabelecer a conexão com o banco de dados e realizar o armazenamento dos dados é necessário configurar uma conta de serviço e gerar uma chave de segurança. No caso do ambiente GCP (Google Cloud Platform) basta acessar o canal “IAM e administrador” e na sequência “contas de serviço”, preencher as informações e gerar a chave de segurança.

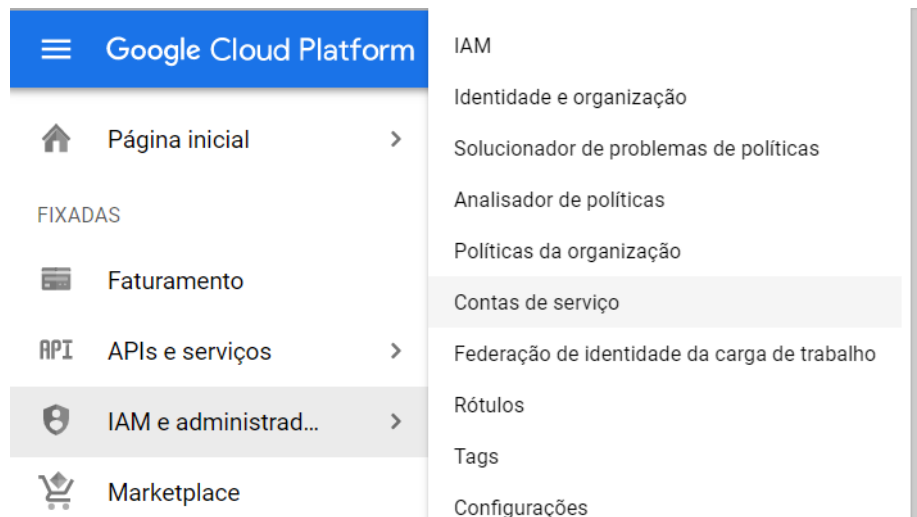


Figura 14 Passos para criar uma conta de serviço e posteriormente gerar a chave de acesso.

```
import os, json
import pandas as pd
import logging

from datetime import datetime

from airflow import DAG
from airflow.operators.dummy_operator import DummyOperator
from airflow.operators.python_operator import PythonOperator

# pip install google-api-python-client
from googleapiclient import discovery
# pip install google
from google.oauth2.service_account import Credentials

os.chdir('/usr/local/airflow/dags')

def get_credential():

    credentials = Credentials.from_service_account_file(
        'cred.json',
        scopes=[u'https://www.googleapis.com/auth/bigquery']
    )
```

```

    return credentials

def transform():
    logging.info(os.getcwd())
    df = pd.read_csv('data.csv')

    df = df[df['sample'].notna()]
    df = df[df['dt_birth'].notna()]
    df["age"] = df["age"].astype(int)

    df.to_csv('transformed_data.csv', index=False)

    return " "

def send_to_gcp():

    df = pd.read_csv('transformed_data.csv')

    df.to_gbq(
        destination_table="estudo_de_caso.sorologia",
        project_id="aerobic-datum-330818",
        if_exists='replace',
        credentials=get_credential(),
        table_schema=json.loads(open('data_schema.json').read())
    )

    return " "

dag = DAG('estudo_de_caso', description="
    schedule_interval='0 12 * * *',
    start_date=datetime(2021, 11, 1), catchup=False)

transform_op = PythonOperator(

```



```

    task_id='transform',
    python_callable=transform,
    dag=dag
)

send_to_gcp_op = PythonOperator(
    task_id='send_to_gcp',
    python_callable=send_to_gcp,
    dag=dag
)

dag >> transform_op >> send_to_gcp_op

```

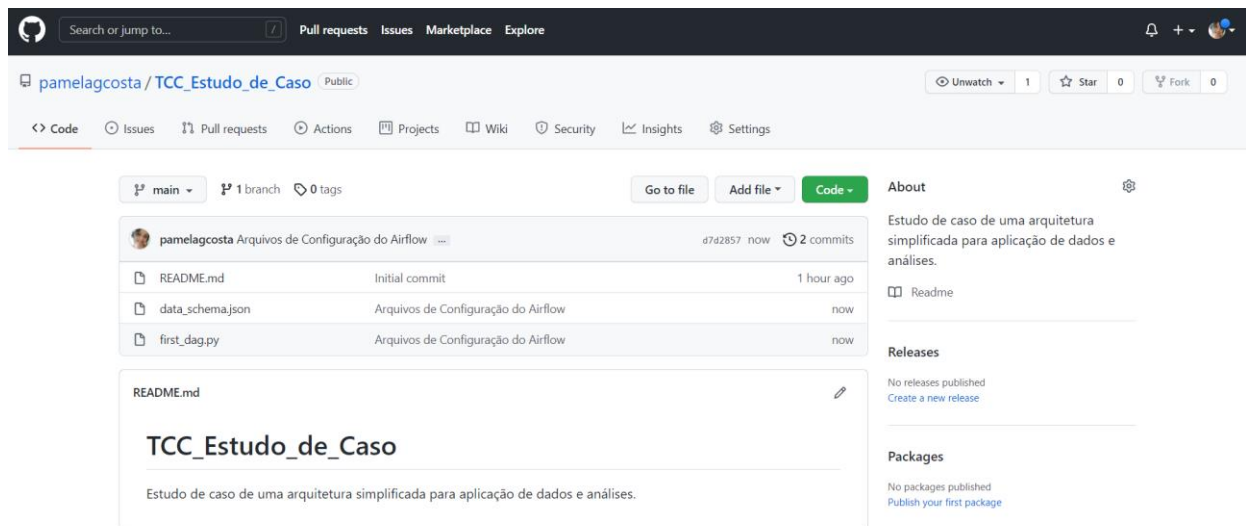
Modelo *schema* projetado para tabela de dados:

```

[
{"name": "sample", "type": "INTEGER", "mode": "NULLABLE"},
{"name": "dt_collection", "type": "STRING", "mode": "NULLABLE"},
{"name": "dt_birth", "type": "STRING", "mode": "NULLABLE"},
{"name": "age", "type": "INTEGER", "mode": "NULLABLE"},
{"name": "city", "type": "STRING", "mode": "NULLABLE"},
{"name": "sex", "type": "STRING", "mode": "NULLABLE"},
{"name": "method1_cmia_screening", "type": "STRING", "mode": "NULLABLE"},
{"name": "method1_elisa_screening", "type": "STRING", "mode": "NULLABLE"},
{"name": "method2_immunoblot_confirmatory", "type": "STRING", "mode": "NULLABLE"},
{"name": "method3_rtpcr_confirmatory", "type": "STRING", "mode": "NULLABLE"}
]

```

Esse processo está agendado para acontecer todos os dias às 12h. Com isso, temos exemplificado um processo de extração, transformação e armazenamento de dados automatizado. Uma observação importante deste processo é que estes códigos desenvolvidos podem ser versionados utilizando ferramentas como o GitHub, o que permite a conexão sincronizada com o Airflow ao GitHub de forma a sempre manter o código versionado, revisado e documentado.



*Figura 15 Arquivos do fluxo do Airflow arquivados para versionamento.*

### 3.4. ETAPA 3: ARMAZENAMENTO DOS DADOS

Para este projeto, visando um modelo de aplicação prática e escalável, optamos pela escolha de um dos três grandes fornecedores de serviços em nuvem, o Google Cloud Platform (GCP) para pilotar o nosso banco de dados de análise (DW). Sendo assim, criamos um projeto e um dataset no Google BigQuery de forma a receber nossa tabela “raw” do processo de automatização.

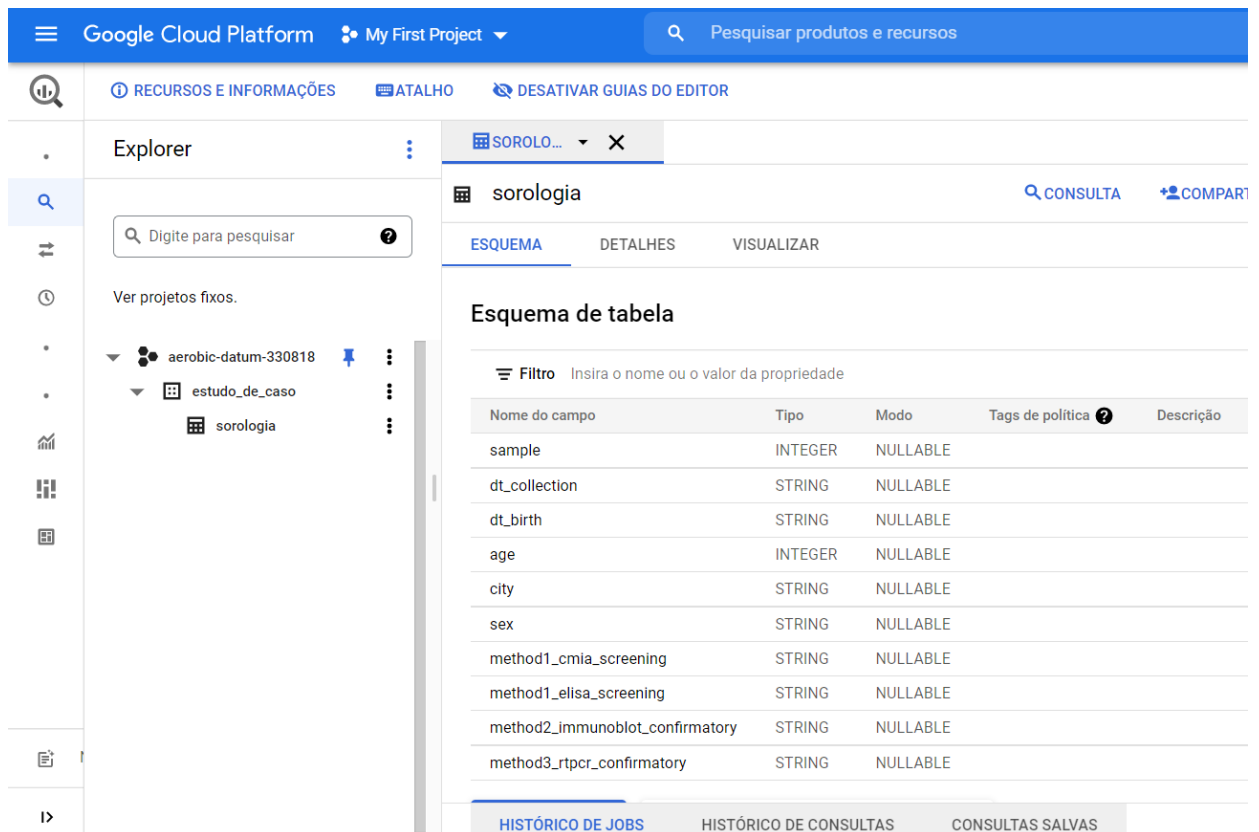


Figura 16 Projeto, dataset e tabela raw no Google BigQuery.

Para criar um projeto no Google BigQuery é necessário criar uma conta na plataforma GCP, o Google disponibiliza uma versão de conta gratuita com créditos para testes. Para criar o projeto é necessário acessar o Google BigQuery, a aba projetos e clicar em “novo projeto”.

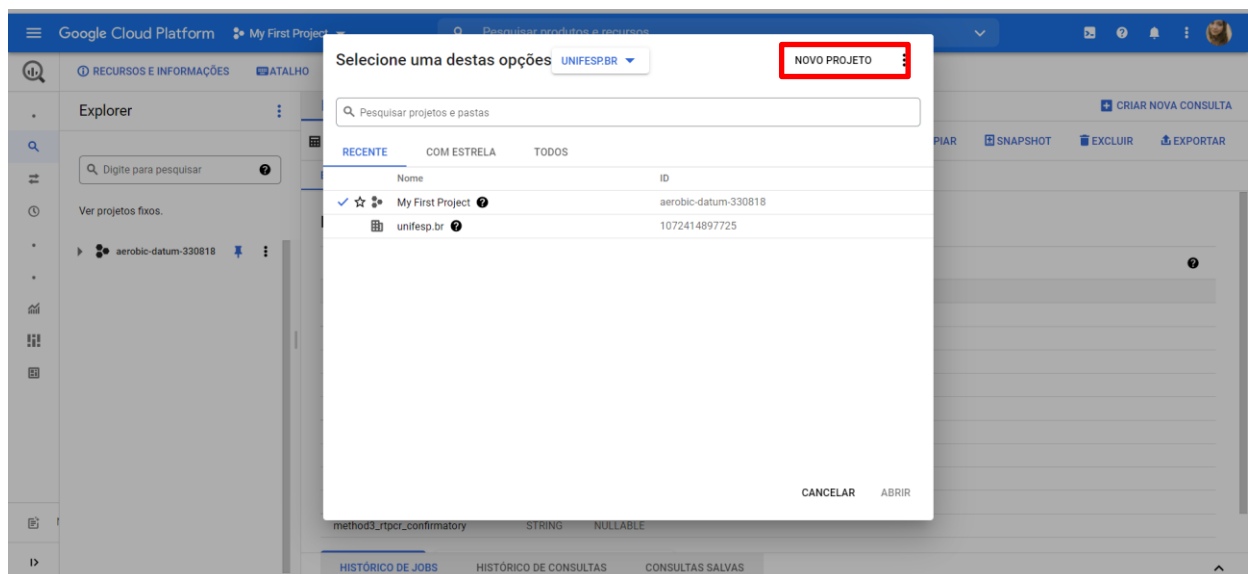


Figura 17 Criando um projeto no BigQuery.

Com o projeto pronto é possível criar um dataset e com isso o espaço para

as tabelas de dados.

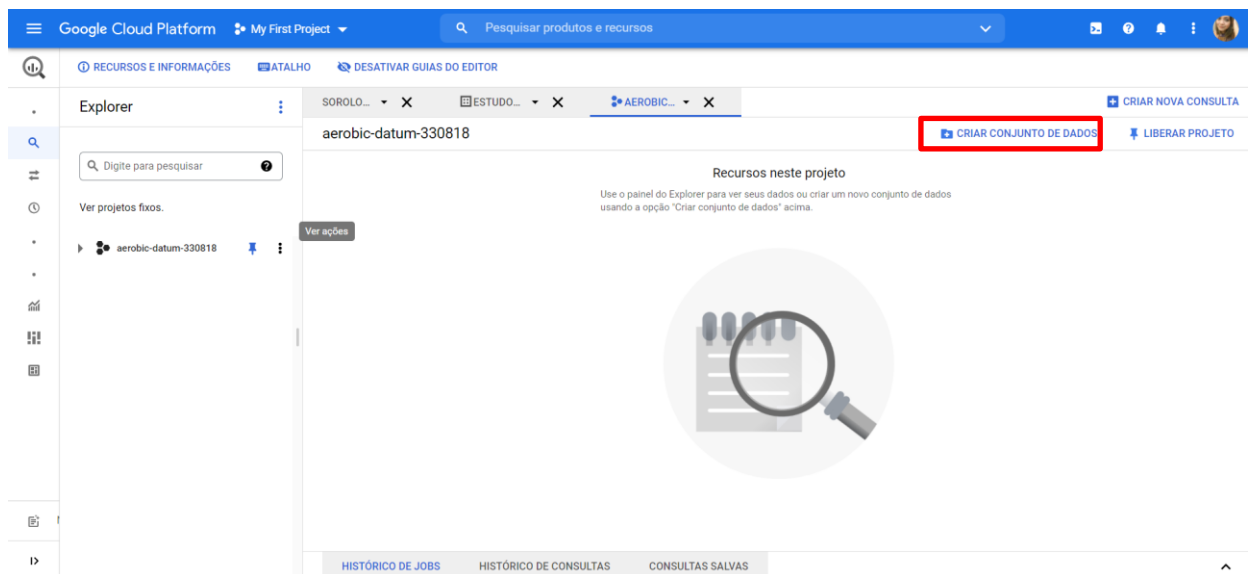


Figura 18 Criando um conjunto de dados no BigQuery.

Por fim, com a tabela de dados pronta podemos utilizar o ambiente para manipular os dados e gerar novas tabelas “*clean*” preparadas para a alimentação de dashboards.

Neste estudo de caso, utilizando código em linguagem Standard SQL, geramos a tabela *clean* “*dashboard*” a partir da tabela “*sorologia*” com dados vindos de sistema pelo *pipeline* automatizado e cruzamos com dados públicos referente a localidade de UBS’s no Brasil (<https://dados.gov.br/dataset/unidades-basicas-de-saude-ubs>), a fim de identificar a disponibilidade de serviços públicos nos estados onde o paciente solicitou seus exames. Existe formas de inserir, diretamente no Google BigQuery, tabela fixas de dados, como neste caso dos dados públicos. Para tanto, basta acessar o dataset desejado e clicar em “criar tabela”, uma página para configuração da tabela e upload do arquivo será aberta, permitindo diversos formatos.

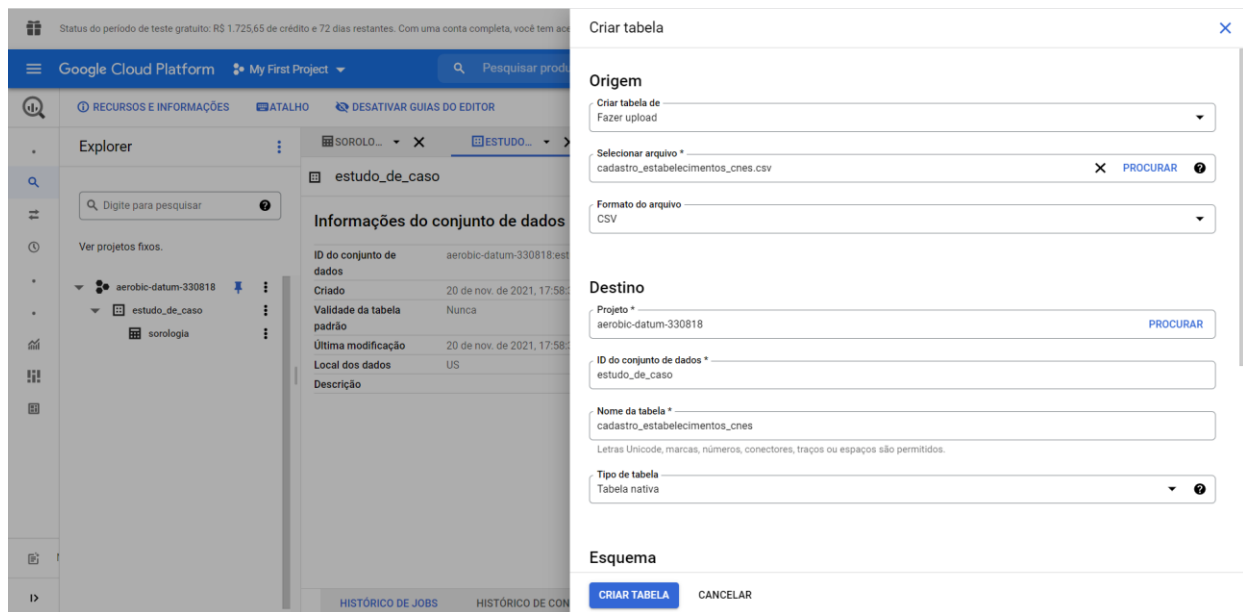


Figura 19 Criando uma tabela de dados no BigQuery.

Com ambas as tabelas criadas, iniciamos o processo de produção da tabela clean “dashboard”, foi utilizado um código em Standart SQL para ajustes e junção de ambas as tabelas “sorologia” e “cadastro\_UBS”. Onde seguimos os seguintes passos:

1º Passo: Como a tabela externa de dados públicos das UBS's foram carregados em formato de string de texto, foi necessário um ajuste para ajustar os dados em variáveis;

Linha	string_field_0
1	CNES;UF;IBGE;NOME;LOGRADOURO;BAIRRO;LATITUDE;LONGITUDE
2	0033820;52;520170;UNIDADE DE SAUDE DA FAMILIA PSF 307;RUA H;NOVO MUNDO;-15.90682;-52.22545
3	0000108;26;260290;USF ALTO DOS INDIOS;RUA 17;PONTE DOS CARVALHOS;-8.28389;-35.0321
4	0000116;26;260290;USF CHARNECA II;RUA 02;CHARNECA;-8.28353;-35.02819
5	0000124;26;260290;USF SAO FRANCISCO I;RUA MANOEL DOMINGOS BARROS;SAO FRANCISCO;-8.287;-35.035
6	0000132;26;260290;USF ROSARIO;RUA 01;ROSARIO;-8.28389;-35.0321
7	0000140;26;260290;USF JUSSARAL;ESTRADA DA VITORIA;JUSSARAL;-8.287;-35.035
8	0000167;26;260290;USF MERCES;RUA DO FERREIRO;MERCE;-8.287;-35.035
9	0000175;26;260290;USF SUAPE;RUA JOSE MIGUEL DE SANTANA;SUAPE;-8.287;-35.035
10	0000248;26;260290;USF MANOEL VIGIA;RUA PREFEITO DIOMEDES FERREIRA DE MELO;PONTE DOS CARVALHOS;-8.287;-35.035
11	0000256;26;260290;USF SANTA ROSA;RUA SANTA ROSA;PONTEZINHA;-8.2243;-34.96841
12	0000264;26;260290;USF MARUM;RUA DO PORTO;PONTE DOS CARVALHOS;-8.2313;-34.98027

Figura 20 Tabela gerada com dados externos das UBS brasileiras.

Consulta para ajuste:

```
WITH limp1 AS(
  SELECT SPLIT(string_field_0, ';')[OFFSET(0)] AS cnes
    , SPLIT(string_field_0, ';')[OFFSET(1)] AS uf
```

```

, SPLIT(string_field_0, ';')[OFFSET(2)] AS nome
, SPLIT(string_field_0, ';')[OFFSET(3)] AS logradouro
, SPLIT(string_field_0, ';')[OFFSET(4)] AS bairro
, SPLIT(string_field_0, ';')[OFFSET(5)] AS complemento
, SPLIT(string_field_0, ';')[OFFSET(6)] AS latitude
, SPLIT(string_field_0, ';')[OFFSET(7)] AS longitude
FROM `aerobic-datum-330818.estudo_de_caso.cadastro_ubs`
)
SELECT *
FROM limp1
WHERE CNES != 'CNES'


```


Resultados da consulta <a href="#">SALVAR RESULTADOS</a> <a href="#">EXPLORAR DADOS</a> ▼									
Consulta finalizada (tempo decorrido:1,5 s, bytes processados: 4 MB)									
Informações do job <a href="#">Resultados</a> <a href="#">JSON</a> <a href="#">Detalhes da execução</a>									
Linha	cnes	uf	nome	logradouro	bairro	complemento	latitude	longitude	
1	0010626	53	530010	UBS 05 TAGUATINGA	SETOR D SUL AREA ESPECIAL N 23	TAGUATINGA SUL	-15.85102	-48.0471	
2	0010634	53	530010	UBS 1 PARANOÁ	QD 21 AREA ESPECIAL CONJUNTO 15	PARANOÁ	-15.76913	-47.77991	
3	0010642	53	530010	UBS 01 SAMAMBAIA	QS 408 AREA ESPECIAL 01	SAMAMBAIA NORTE	-15.86213	-48.07976	
4	0010650	53	530010	UBS 2 PLANALTINA	ENTRE QUADRAS 1 10 AREA ESPECIAL	PLANALTINA	-15.61241	-47.6454	
5	0010669	53	530010	UBS 2 SANTA MARIA	EQ 217 317 LOTE E	SANTA MARIA	-16.00744	-47.98995	
6	0010677	53	530010	UBS 03 SAMAMBAIA	QN 429 CONJUNTO F LOTE	SAMAMBAIA NORTE	-15.78	-47.93	
7	0010685	53	530010	UBS 04 SAMAMBAIA	QN 512 CONJUNTO 2 LOTE	SAMAMBAIA SUL	-15.87891	-48.06898	
<div> Linhas por página: 100 ▼ 1 - 100 de 41822 Primeira página  &lt; &gt;  Última página </div>									
<div> HISTÓRICO DE JOBS HISTÓRICO DE CONSULTAS CONSULTAS SALVAS </div>									


*Figura 21 Resultado obtido na consulta idealizada para ajuste da tabela UBS.*

2º Passo: Ajuste da nomenclatura de estados, no dataset de sorologia possuíamos cidades de três estados diferentes, sendo São Paulo, Rio de Janeiro e Belo Horizonte;

▶ EXECUTAR

 SALVAR ▼

 PROGRAMAÇÃO ▼

 MAIS ▼


✔


Esta consulta processará 1,1 MIB quando executada.

1    SELECT DISTINCT city

2    FROM `aerobic-datum-330818.estudo\_de\_caso.sorologia`

Resultados da consulta

 SALVAR RESULTADOS

 EXPLORAR DADOS ▼

Consulta finalizada (tempo decorrido:0,4 s, bytes processados: 1,1 MB)

Informações do job

Resultados

JSON

Detalhes da execução

Linha	city
1	Rio de Janeiro
2	São Paulo
3	Santo Andre
4	São Caetano
5	Osasco
6	Guarulhos
7	Diadema
8	Belo Horizonte

Figura 22 Verificação das opções disponíveis de cidade no banco de dados.

SELECT sample

, dt\_collection

, dt\_birth

, age

, CASE city

WHEN "Rio de Janeiro"

THEN "33" -- Código RJ

WHEN "Belo Horizonte"

THEN "31" -- Código MG

ELSE "35" -- Código SP

END AS uf

, sex



, method1\_cmia\_screening

, method1\_elisa\_screening

, method2\_immunoblot\_confirmatory

, method3\_rtPCR\_confirmatory

FROM `aerobic-datum-330818.estudo\_de\_caso.sorologia`

Resultados da consulta										
<div>  SALVAR RESULTADOS            EXPLORAR DADOS         </div>										
Consulta finalizada (tempo decorrido:1,3 s, bytes processados: 10,7 MB)										
<div>           Informações do job           Resultados           JSON           Detalhes da execução         </div>										
Linha	sample	dt_collection	dt_birth	age	uf	sex	method1_cmia_screening	method1_elisa_screening	method2_immunoblot_confirmatory	method3_rtpcr_c
1	770	5/7/19 11:09	11/21/2018	0	33	MASCULINO	NÃO REAGENTE	0,04	NÃO REALIZADO	NÃO REALIZAD
2	4514	5/15/19 11:21	4/1/2019	0	35	FEMININO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
3	6695	5/20/20 20:50	5/14/2020	0	35	MASCULINO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
4	2458	5/7/18 11:44	12/19/2017	0	35	MASCULINO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
5	3356	1/4/19 11:34	7/2/2018	0	35	FEMININO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
6	7424	8/12/20 1:01	6/23/2020	0	35	MASCULINO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
7	1788	2/15/18 11:26	11/3/2017	0	35	FEMININO	NÃO REAGENTE	0,05	NÃO REALIZADO	NÃO REALIZAD
8	14142	6/5/19 7:07	4/1/2019	0	35	FEMININO	NÃO REAGENTE	0,06	NÃO REALIZADO	NÃO REALIZAD

Linhas por página: 100
 1 - 100 de 95901
 Primeira página
 <
 >
 > Última página

Figura 23 Resultado obtido na consulta idealizada para ajuste da tabela sorologia.

Por fim, unificando os ajustes e realizado a junção de ambas as tabelas disponíveis pela chave de “uf”, a tabela “dashboards” gerada a partir desta junção será a responsável por alimentar o painel de resultados final.

```
CREATE OR REPLACE TABLE `aerobic-datum-330818.estudo_de_caso.dashboard`
(
  sample INT64 OPTIONS (description = "ID da Amostra do Exame")
  , dt_collection STRING OPTIONS (description = "Data e Hora da coleta do Exame")
  , ubs_disp_uf INT64 OPTIONS (description = "Quantidade de UBS's disponíveis no e
stado do paciente")
  , dt_birth STRING OPTIONS (description = "Data de nascimento do Paciente")
  , age INT64 OPTIONS (description = "Idade do Paciente")
  , city STRING OPTIONS (description = "Cidade do Paciente")
  , uf STRING OPTIONS (description = "Estado (UF) do Paciente")
  , sex STRING OPTIONS (description = "Sexo do Paciente")
  , method1_cmia_screening STRING OPTIONS (description = "Método 1 CMIA")
  , method1_elisa_screening STRING OPTIONS (description = "Método 1 Elisa")
  , method2_immunoblot_confirmatory STRING OPTIONS (description = "Método 2 Im
munoBlot")
  , method3_rtpcr_confirmatorY STRING OPTIONS (description = "Método 3 RT-
PCR");

```



```
INSERT INTO `aerobic-datum-330818.estudo_de_caso.dashboard` (
```

```
WITH limp1 AS(
```

```
    SELECT SPLIT(string_field_0, ';')[OFFSET(0)] AS cnes
    , SPLIT(string_field_0, ';')[OFFSET(1)] AS uf
    , SPLIT(string_field_0, ';')[OFFSET(2)] AS nome
    , SPLIT(string_field_0, ';')[OFFSET(3)] AS logradouro
    , SPLIT(string_field_0, ';')[OFFSET(4)] AS bairro
    , SPLIT(string_field_0, ';')[OFFSET(5)] AS complemento
    , SPLIT(string_field_0, ';')[OFFSET(6)] AS latitude
    , SPLIT(string_field_0, ';')[OFFSET(7)] AS longitude
```

```
    FROM `aerobic-datum-330818.estudo_de_caso.cadastro_ubs`
```

```
)
```

```
, ajuste_cadastro_ubs as(
```

```
    SELECT uf
    , COUNT(DISTINCT cnes) AS ubs_disp_uf
    FROM limp1
    WHERE CNES != 'CNES'
    GROUP BY uf
```

```
)
```

```
, ajuste_sorologia as(
```

```
    SELECT sample
    , dt_collection
    , dt_birth
    , age
    , CASE city
        WHEN "Rio de Janeiro"
        THEN "33" -- Código RJ
        WHEN "Belo Horizonte"
        THEN "31" -- Código MG
        ELSE "35" -- Código SP
    END AS uf
```

```
    END AS uf
```

```
    , city
```

```

, sex
, method1_cmia_screening
, method1_elisa_screening
, method2_immunoblot_confirmatory
, method3_rtpcr_confirmatoryY
FROM `aerobic-datum-330818.estudo_de_caso.sorologia`
)
SELECT CAST(s.sample AS INT64) AS sample
, dt_collection
, CAST(u.ubs_disp_uf AS INT64) AS ubs_disp_uf
, s.dt_birth
, s.age
, s.city
, s.uf
, s.sex
, s.method1_cmia_screening
, s.method1_elisa_screening
, s.method2_immunoblot_confirmatory
, s.method3_rtpcr_confirmatoryY
FROM ajuste_sorologia AS s
LEFT JOIN ajuste_cadastro_ubs AS u
ON s.uf = u.uf
)

```

Nome do campo	Tipo	Modo	Tags de política	Descrição
sample	INTEGER	NULLABLE		ID da Amostra do Exame
dt_collection	STRING	NULLABLE		Data e Hora da coleta do Exame
ubs_disp_uf	INTEGER	NULLABLE		Quantidade de UBS's disponíveis no estado do paciente
dt_birth	STRING	NULLABLE		Data de nascimento do Paciente
age	INTEGER	NULLABLE		Idade do Paciente
city	STRING	NULLABLE		Cidade do Paciente
uf	STRING	NULLABLE		Estado (UF) do Paciente
sex	STRING	NULLABLE		Sexo do Paciente
method1_cmia_screening	STRING	NULLABLE		Método 1 CMIA
method1_elisa_screening	STRING	NULLABLE		Método 1 Elisa
method2_immunoblot_confirmatory	STRING	NULLABLE		Método 2 Immunoblot

Figura 24 Detalhes da tabela dashboard criada.

Linha	sample	dt_collection	ubs_disp_uf	dt_birth	age	city	uf	sex	method1_cmia_screening	method1_elisa_screening	method2_immunoblot_confirmatory	method3_uf
1	97646	8/11/20 7:05	5329	2/15/1993	27	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
2	97647	8/11/20 8:55	5329	3/12/1989	31	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
3	97684	9/11/20 17:23	5329	4/30/1980	40	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
4	97728	9/30/20 9:23	5329	3/28/1979	41	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
5	97800	10/27/20 12:12	5329	3/20/1998	22	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
6	97807	10/30/20 9:30	5329	12/26/1997	23	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
7	97893	6/16/21 12:43	5329	9/9/1982	38	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
8	97732	10/1/20 8:24	5329	12/11/1982	37	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
9	97864	12/8/20 8:55	5329	2/27/1981	39	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
10	97694	9/14/20 9:32	5329	10/24/1977	42	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
11	97702	9/17/20 9:33	5329	2/23/1988	32	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO
12	97723	9/29/20 8:47	5329	8/29/1972	47	São Paulo	35	FEMININO	NÃO REAGENTE	0,00	NÃO REALIZADO	NÃO REALIZADO

Figura 25 Pré visualização da tabela dashboards.

Para que ela esteja sempre atualizada o GCP fornece uma ferramenta muito interessante e útil chamada “Programação de Consultas”, com ela é possível programar a consulta desejada para que a mesma atualize uma tabela periodicamente.

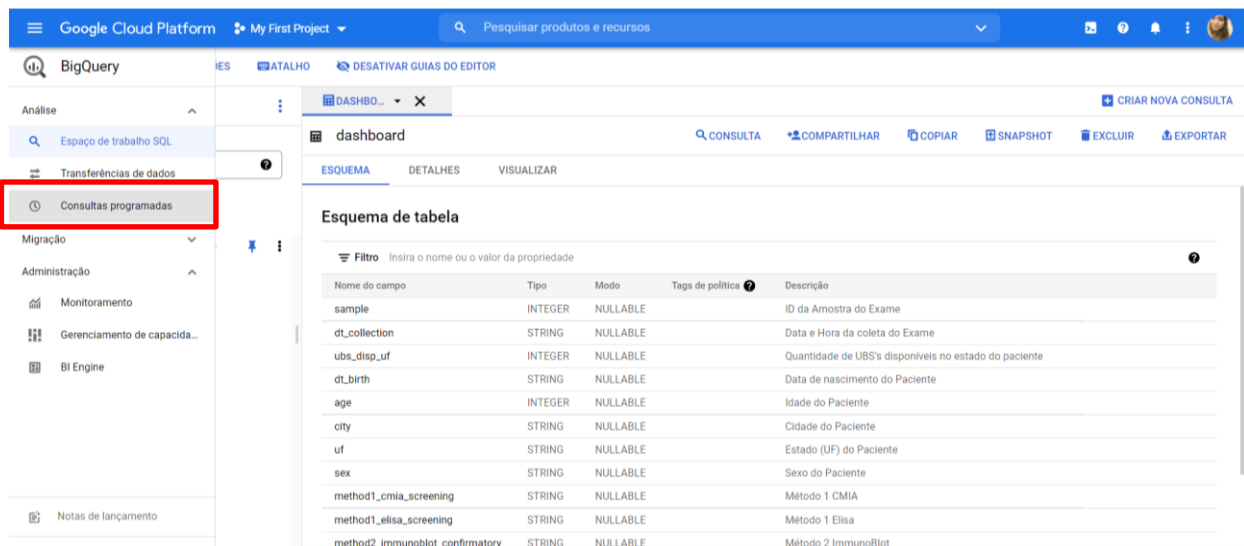


Figura 26 Indicação de local para programação de consultas no BigQuery.

### 3.5. ETAPA 4: DOCUMENTAÇÃO TÉCNICA DO PROJETO

Como uma boa prática de gestão de projetos e também de controle, utilizamos a plataforma GitHub para arquivar, documentar e versionar os arquivos técnicos deste estudo de caso. Muitas são as vantagens deste método, primeiramente em relação a etapa de automatização, uma vez conectado o Airflow ao código no GitHub podemos garantir sempre a versão mais atualizada em produção. Além disso, criamos um fluxo de segurança e revisão para o processo. Esse fluxo é muito similar e indispensável em ambientes de produção.

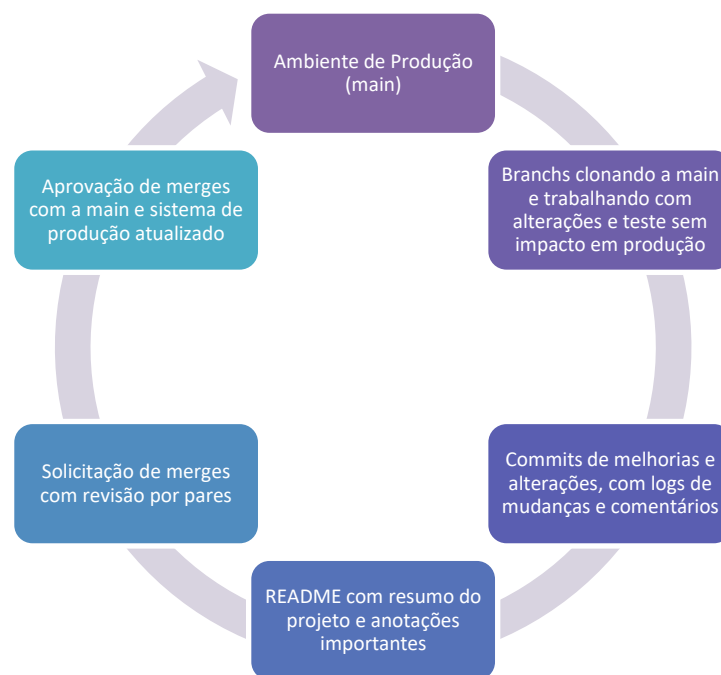


Figura 27 Fluxo de organização do processo de trabalho.

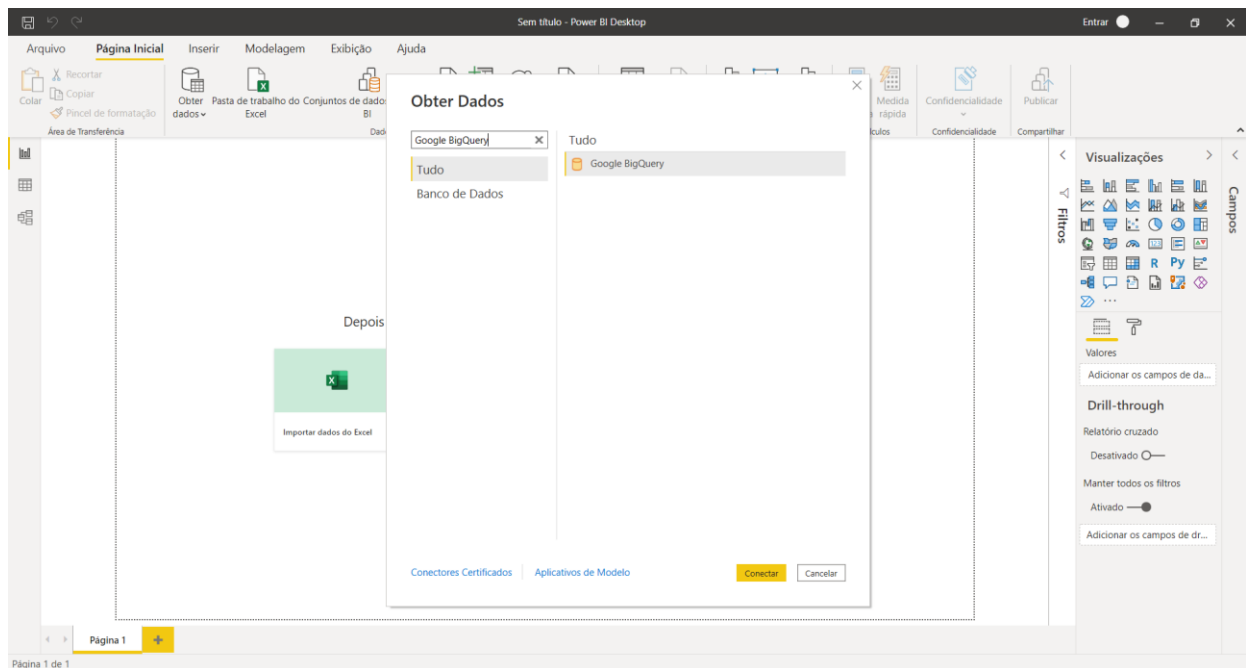


*Figura 28 Ambiente criado para versionamento dos arquivos gerados no projeto.*

Uma conta no GitHub é gratuita e feita pelo site da plataforma. É possível a criação de projetos fechados, para se trabalhar apenas com o seu time e também projetos abertos para compartilhamento de conhecimento com a comunidade. Opção escaláveis para empresas estão disponíveis em versões personalizadas, neste caso colocamos também como opção o GitLab, que possui o mesmo propósito que o GitHub, porém é uma opção para implantações em larga escala em empresas.

### **3.6. ETAPA 5: CONEXÃO DO DATASET AO POWER BI**

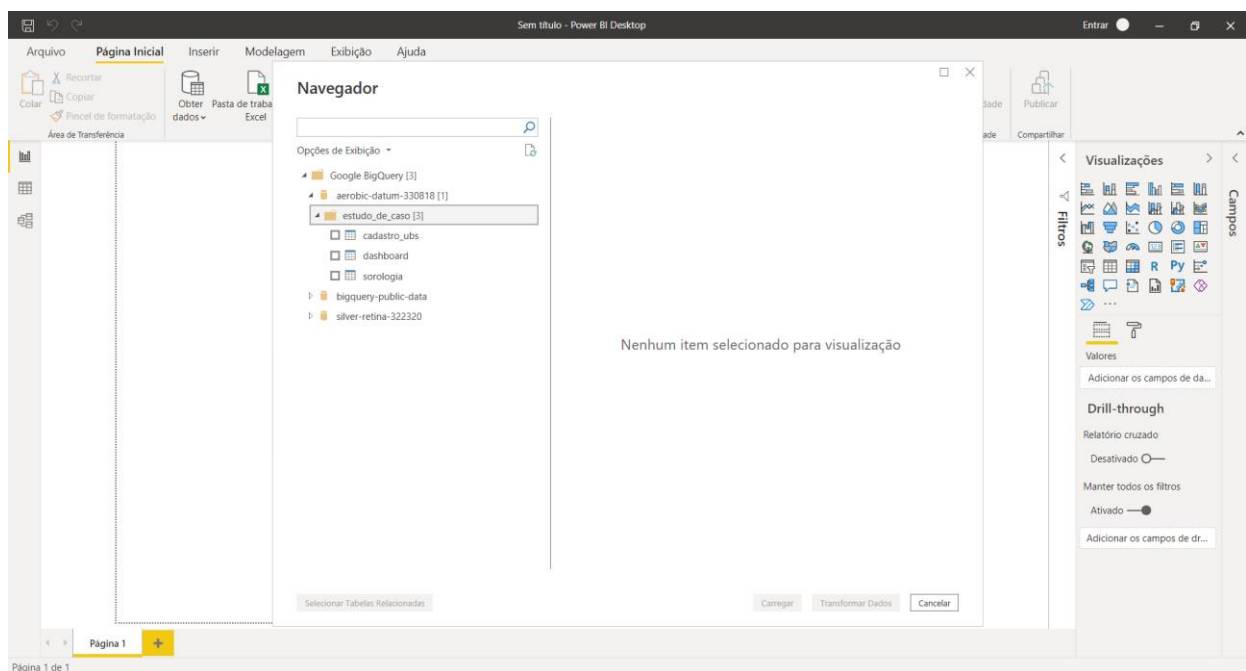
A plataforma GCP possui uma ferramenta para a construção de dashboards chamada Datastudio, quando se trabalha em um ambiente Google, essa ferramenta pode ser uma opção muito interessante para a construção de painéis, da mesma forma que a Amazon possui o QuickSight. Porém, para este estudo de caso, optamos pelo uso do Microsoft Power BI Desktop, por ser uma ferramenta bastante utilizada e difundida no mercado. Além disso, é possível se conectar a um banco de dados Google BigQuery e usar os dados subjacentes pela própria ferramenta. Para se conectar a um banco de dados Google BigQuery, selecione “Obter dados” na faixa de opções “Início” no Power BI Desktop. Selecione “Banco de dados” nas categorias à esquerda e você verá Google BigQuery.



*Figura 29 Forma de conexão do BigQuery com o Power BI.*

Na sequência basta colocar suas credenciais do bando de dados e toda a lista de datasets e tabelas estará disponível para uso no Power BI Desktop.

(<https://docs.microsoft.com/pt-br/power-bi/connect-data/desktop-connect-bigquery>)



*Figura 30 Demonstração do ambiente conectado (BigQuery x Power BI).*

### 3.7. ETAPA 6: ACOMODAÇÃO DOS DADOS EM POWER QUERY

Primeiramente, vamos conectar a nossa tabela clean “dashboard”. Após selecionada, acessar “Transformar dados” para abrir a ferramenta Power Query.

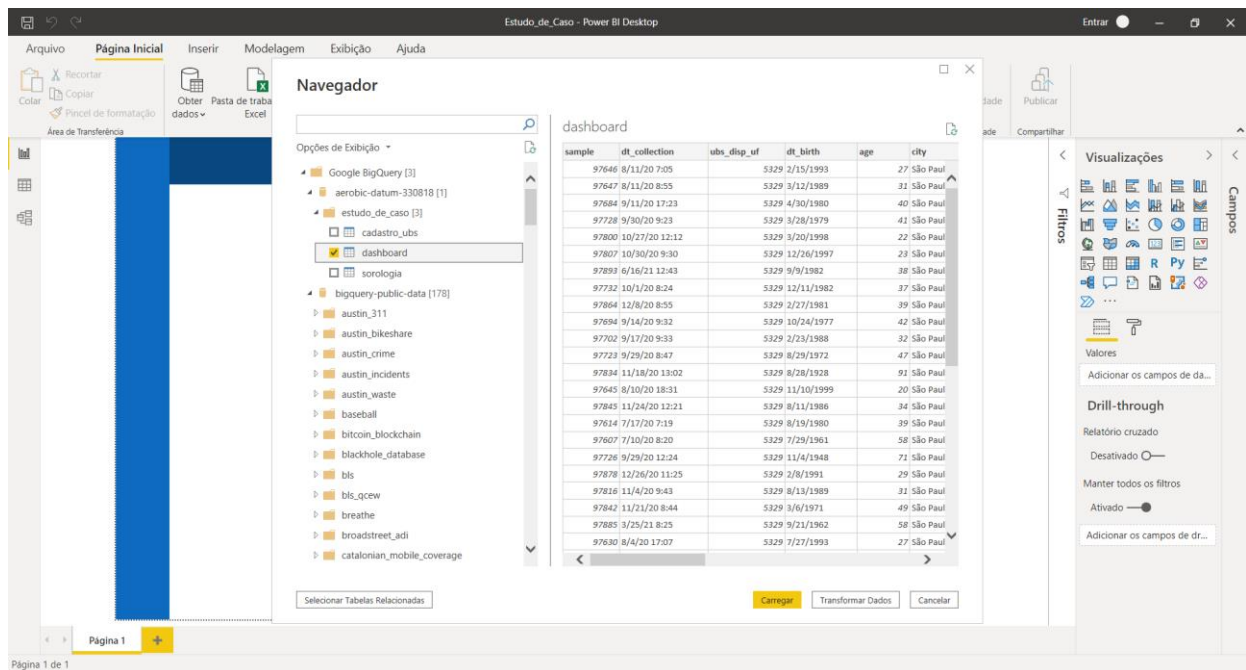


Figura 31 Selecionando uma tabela do BigQuery para utilização no Power Query.

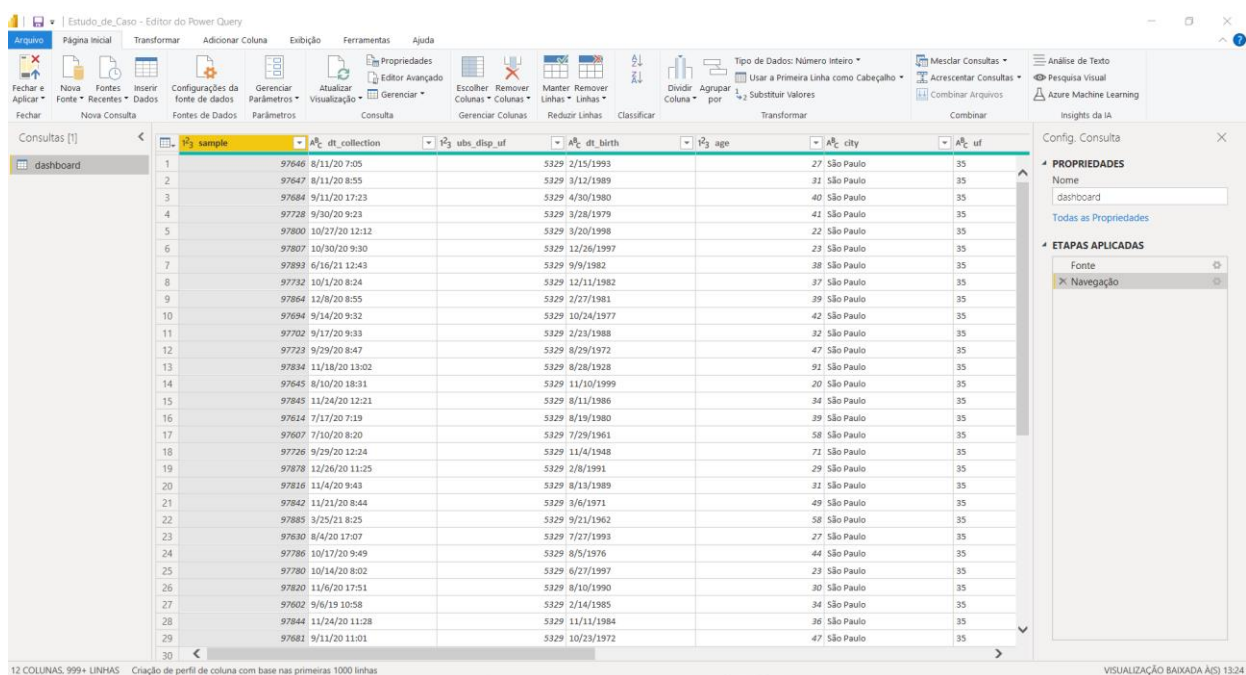
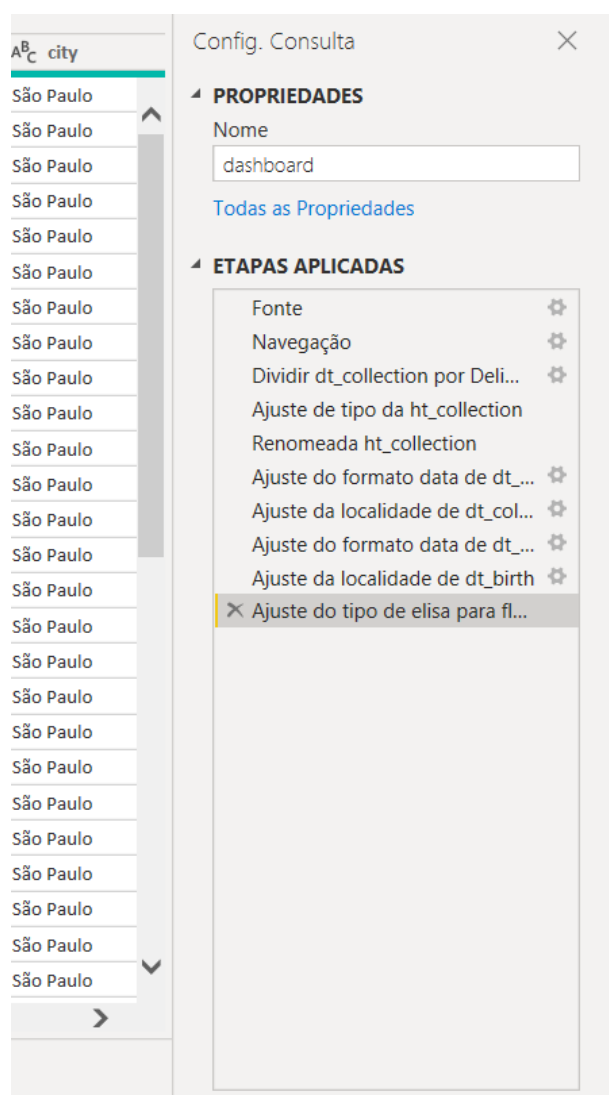


Figura 32 Tabela preparada para transformações no Power Query.

Utilizamos o Power Query para fazer ajustes na tabela que fazem sentido a aplicação visual do painel. Recomendamos que sempre seja renomeado as etapas aplicadas no Power Query. Pois, em caso de necessidade de alteração ou ajuste fica organizada a sequência das alterações. Neste estudo de caso ajustamos o formato das datas, separando a dt\_collection em data e hora e também ajustando valores string para

float no método elisa.



*Figura 33 Indicação de método para documentação das etapas de transformação no Power Query.*

Para salvar as alterações realizadas no Power Query, basta clicar em “fechar e aplicar” no canto superior esquerdo da página.

### **3.8. ETAPA 7: CONSTRUÇÃO DE DASHBOARD**

Conforme verificado com a área de negócio o painel de resultados (dashboard) precisaria ser simples, de fácil leitura e acompanhamento da área, para ficar à disposição dos stakeholders envolvidos no processo. Sendo assim, foi desenvolvido o seguinte painel utilizando a plataforma Power BI Desktop.



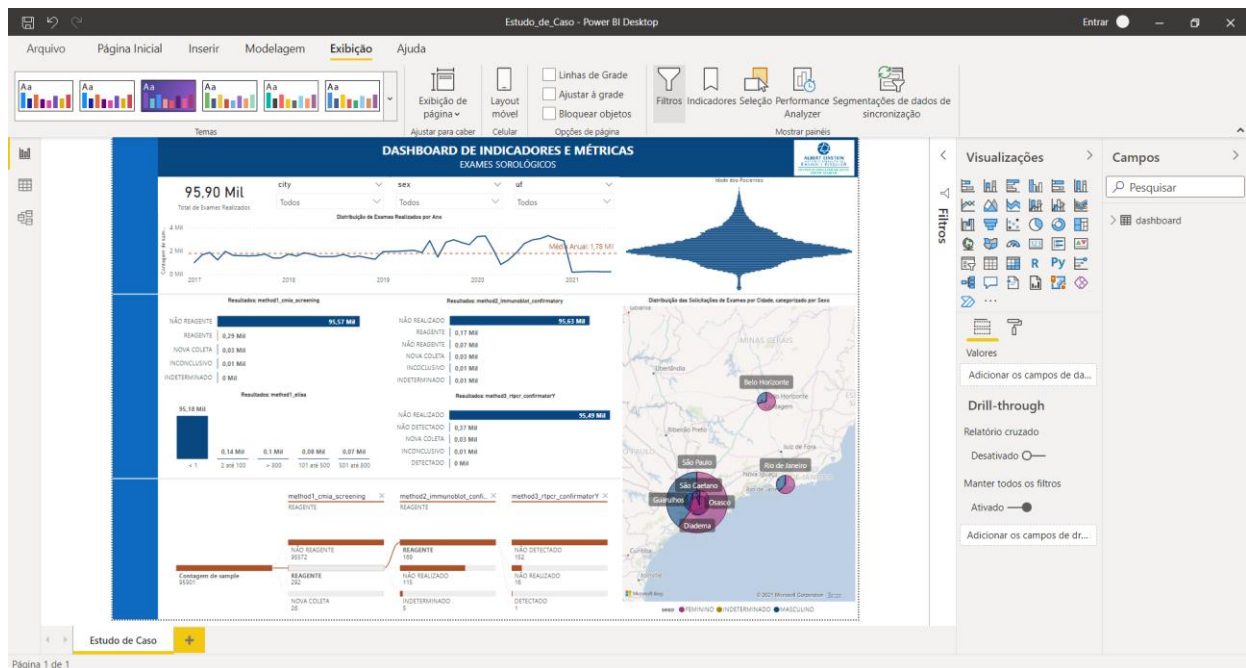
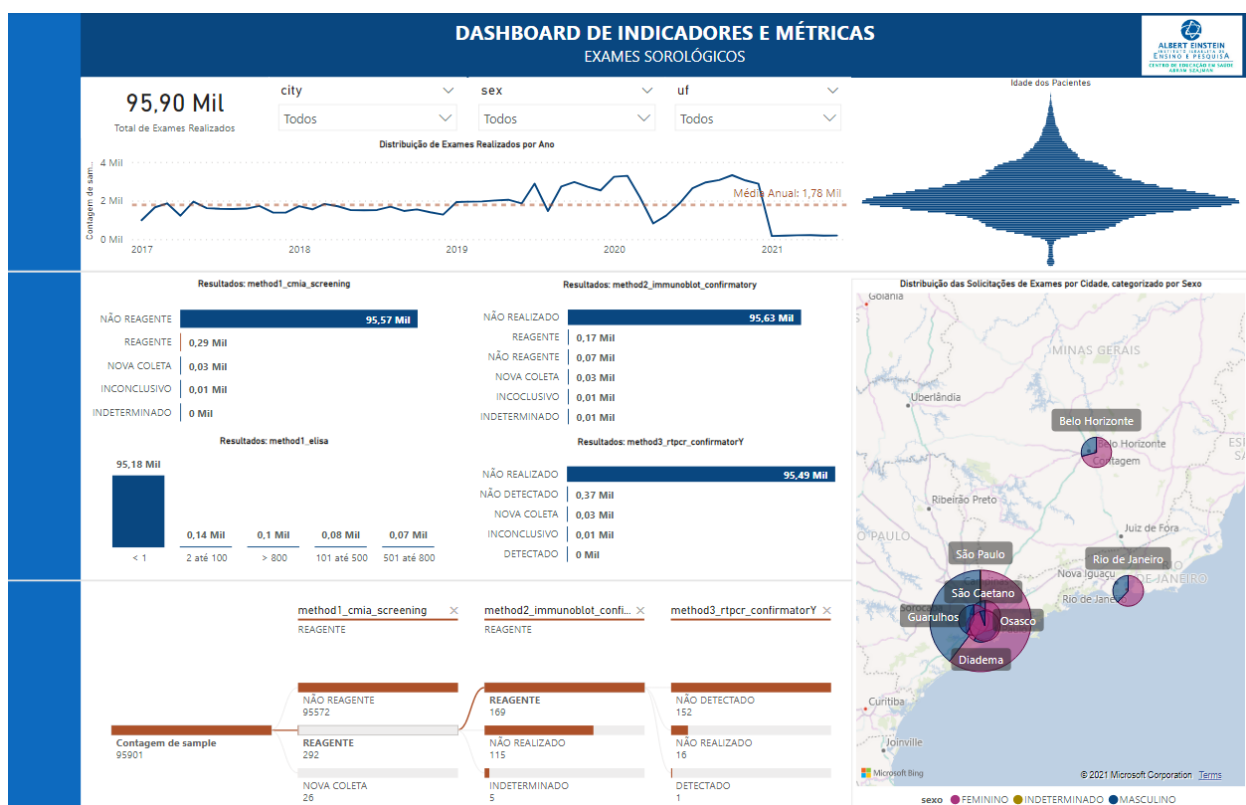


Figura 34 Visão geral do dashboard na ferramenta Power BI.

Sendo uma visão ampliada da versão de compartilhamento em desktop.



*Figura 35 Visão geral do dashboard desenvolvido para o estudo de caso.*

Como o intuito do painel é ser acessível, expandimos sua visualização também para mobile.

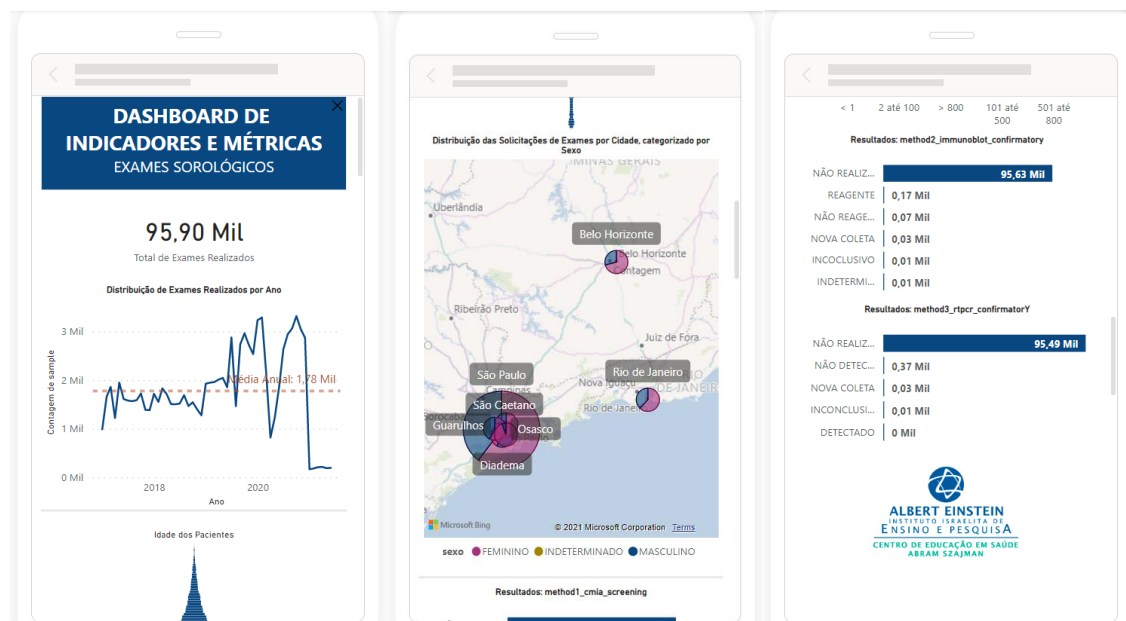


Figura 36 Visão do dashboard desenvolvida para mobile.

Foram estruturados cinco pontos focais visuais neste painel, sendo o primeiro a visão temporal do volume de exames realizados. Nele é possível ter uma visão do comportamento dos pacientes na procura pela realização de exames, implementamos como apoio a linha da média temporal.

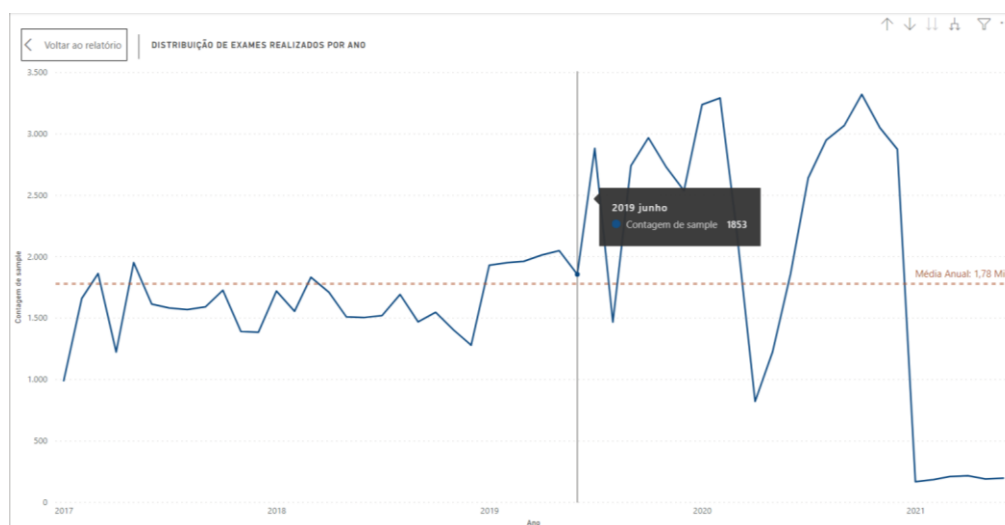


Figura 37 Gráfico para comportamento dos pacientes na procura pela realização de exames.

Como segundo ponto implementamos um histograma da idade dos pacientes para uma melhor visão do publico que vem sendo atendido. É possível, visualizar, por exemplo, que a maioria do publico atendido possui uma idade de 30 a 42 anos.

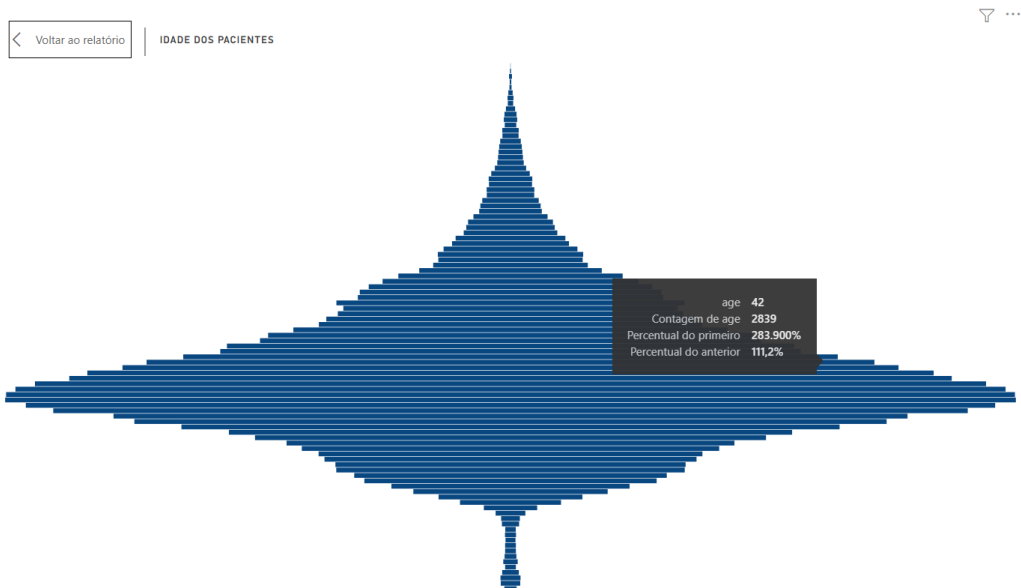


Figura 38 Gráfico do histograma da idade dos pacientes.

No terceiro bloco implementamos uma visão mais técnica, porém importante para o acompanhamento dos profissionais de controle da área. Nesta fase temos o resultado um quadro para cada um dos métodos, onde é possível verificar o volume absoluto de realização de exames em cada etapa. Além de identificar a proporção entre as classes de resultados.



Figura 39 Quadro para cada método de verificação.

O quarto loco é complementar ao terceiro e vem como um mapeamento visual do caminho do paciente pelo diagnóstico. Onde, é possível seleccionar os grupos de resultados e realizar um funil de valores.

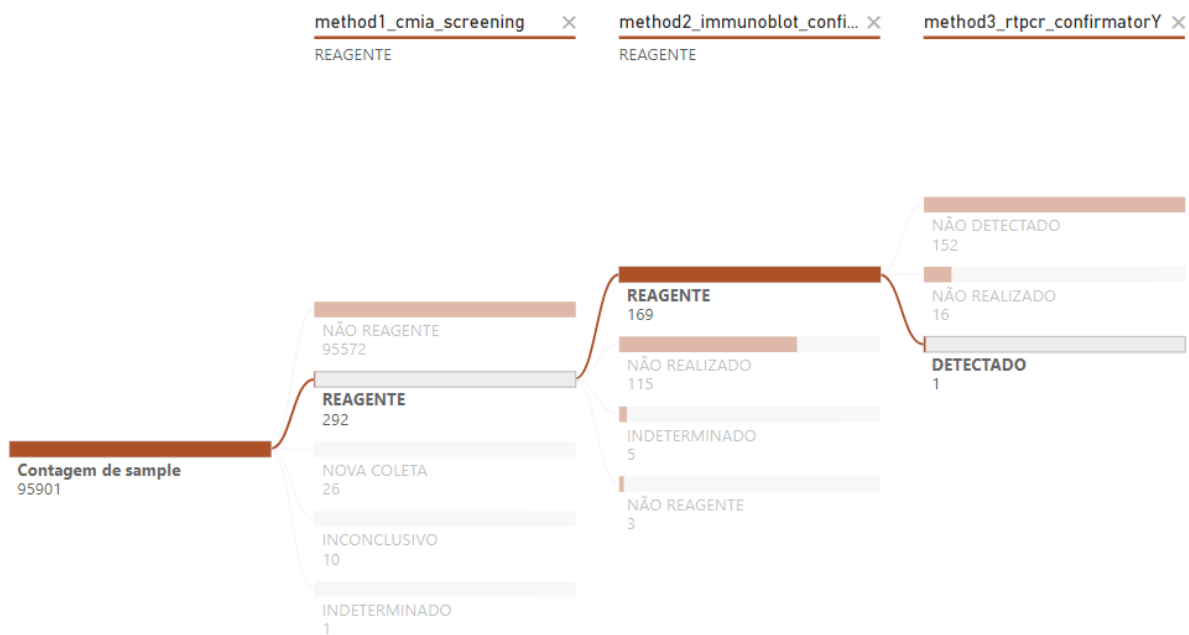


Figura 40 Arvore de ramificação com o caminho do paciente pelo diagnóstico.

Por fim, o quinto e último bloco representa uma visão em mapa do volume de exames realizados por cidade, classificados por sexo e com a informação complementar do volume de UBSs disponível.

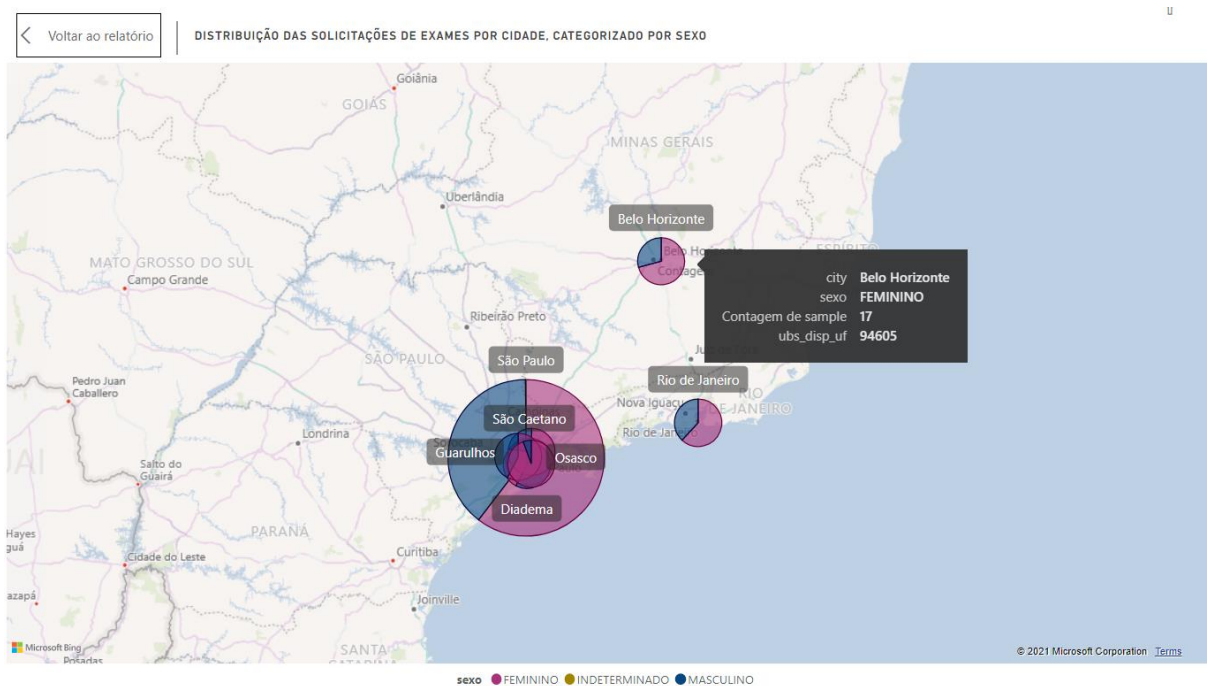
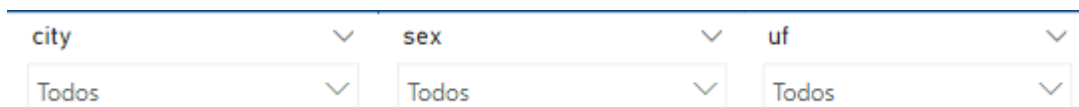


Figura 41 Mapa do volume de exames realizados por cidade.

Lembrando que todos os visuais são interativos e podem ser filtrados entre si e também pelos filtros disponíveis no painel.



The image shows three global filter dropdown menus arranged horizontally. Each menu has a label above it ('city', 'sex', 'uf') and a dropdown box below it. All three dropdown boxes currently display the word 'Todos' and have a small downward arrow icon on the right side of the box.

*Figura 42 Filtros globais do painel.*

#### **4. CONCLUSÃO**

O ambiente hospitalar possui diversos sistemas e fontes de dados ricos e dispersas. Entendemos existir uma oportunidade muito ampla de aplicação das tecnologias de análises de dados, de forma a agregar, organizar, aplicar governança e principalmente automatizar processos e dispor de dados com alta confiabilidade. A variabilidade de sistemas e fontes é, sem dúvida, o maior desafio para a aplicação de um processo de coleta, tratamento e análise de dados, porém também é de suma importância profissionais capacitados e tecnologia coerente com o objetivo final que é dispor de forma confiável e acessível as informações para que os stackholder envolvidos no processo possam usufruir do potencial estratégico que os dados dispõem. Lembrando que, para que todo esse ecossistema seja construído e se mantenha vivo e operante, é essencial o cultivo do pensamento data driven entre todos os times de trabalho.

Por fim, acreditamos que este estudo de caso trará um, de muitos, caminhos possíveis para a aplicação escalável e inteligente de conversão de dados em conhecimento.

## 5. REFERÊNCIAS BIBLIOGRAFICAS

- [1] IBM CLOUD EDUCATION. ETL (Extract, Transform, Load). IBM, 2021. Disponível em: <https://www.ibm.com/cloud/learn/etl#toc-what-is-et-xeCDpL69>. Acesso em: 10 nov. 2021.
- [2] MJV TEAM. O que é ETL e por que devemos integrar dados? MJV Technology and Innovation, 07 nov. 2021. Disponível em: <https://www.mjvinnovation.com/pt-br/blog/o-que-e-etl-como-funciona/>.
- [3] WHAT is ETL? ORACLE, 2021. Disponível em: <https://www.oracle.com/integration/what-is-etl/>. Acesso em: 07 nov. 2021.
- [4] BROWN, T. Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias. Brasil: Alta Books, 2020.
- [5] BONINI, L. A.; SBRAGIA, R. O modelo de design thinking como indutor da inovação nas empresas: um estudo empírico. Revista de Gestão e Projetos – GeP. São Paulo, v. 2, n. ja/ju 2011, p. 3-25, 2011. Disponível em <<http://www.revistagep.org/ojs/index.php/gep/article/view/36/136>> Acesso em 02 nov. 2021.
- [6] CÔRTEZ, J. C. S. et al. Design Thinking na Reestruturação do Sistema de Avaliação de Disciplina em um Curso de Medicina. Revista Brasileira de Educação Médica. 2020, v. 44, n. 04. Disponível em <<https://doi.org/10.1590/1981-5271v44.4-20200125>> Acesso em 01 nov. 2021.
- [7] IDEO. Design Thinking Defined. 2021. Disponível em <<https://designthinking.ideo.com/>> Acesso em 02 nov. 2021.
- [8] Universidade EHS. Cases de Sucesso em Design Thinking Aplicado à Saúde. 2021. Disponível em <<https://universidadeehs.com.br/6-cases-de-sucesso-em-design-thinking-aplicado-a-saude/>> Acesso em 31 out. 2021.
- [9] Sistemas de banco de dados / Ramez Elmasri e Shamkant B. Navathe; revisor técnico Luis Ricardo de Figueiredo. -- São Paulo : Pearson Addison Wesley, 2005.
- [10] AIRFLOW.APACHE. Página oficial da plataforma Apache Airflow. Disponível em < <https://airflow.apache.org/>> Acesso em 01 nov. 2021.
- [11] AIRFLOW.DOC. Página oficial da plataforma Apache Airflow. Disponível em < <https://airflow.apache.org/docs/apache-airflow/stable/concepts/scheduler.html>> Acesso em 01 nov. 2021.

- [12] Few, S. (2013). Information Dashboard Design: Displaying data for at-a-glance monitoring (2nd ed.). Burlingame, California: Analytics Press.
- [13] Eckerson, W. W. (2011). Performance Dashboards - Measuring, Monitoring and Managing your Business (2nd ed.). John Wiley & Sons, Inc, Hoboken, New Jersey
- [14] Ganesh, J., & Anand, S. (2005). Web services, enterprise digital dashboards and shared data services: A proposed framework. Proceedings - Third European Conference on Web Services, ECOWS2005, 2005, 130–137.  
<http://doi.org/10.1109/ECOWS.2005.29>
- [15] Gemignani, Z. (2009). A Guide to Creating Dashboards People Love to Use - Report. Retrieved from <http://www.juiceanalytics.com/white-papers-guides-and-more/>
- [16] Kerzner, H. (2013). Project Management: Metrics, KPIs and Dashboards – A Guide to Measuring and Monitoring Project Performance. John Wiley & Sons, Inc., Hoboken, New Jersey
- [17] Gonzalez, T. (2008). The Future Of BI - Report. Retrieved from <http://www.brightpointinc.com/data-visualization-articles/>
- [18] Johar, H. (2010). Data Visualization Basics For Dashboards. Retrieved June 14, 2015, from <http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/datavisualization-basics-for-dashboards-part-one.aspx#sthash.fQYhAalc.dpu>
- [19] <https://www.saude.gov.br/files//conecta-sus/produtos-tecnicos/II%20-%202020/Prontu%C3%A1rio%20Eletr%C3%B4nico%20do%20Paciente%20-%20PEP.pdf>
- [20] ALMEIDA, M.J.G.G. et al. Discussão Ética sobre o Prontuário Eletrônico do Paciente. Revista Brasileira de Educação Médica. V. 40 n. 3, 2016.
- [21] BRASIL. Rede Nacional de Dados em Saúde. 2019
- [22] COSTA, C.G.A. Desenvolvimento e avaliação tecnológica de um sistema de prontuário eletrônico do paciente, baseado nos paradigmas da World Wide Web e da engenharia de software. Campinas, SP. 2001.
- [23] GUTIERREZ, E., CARVALHO, C. Saúde Pública em Portugal: como funciona, preço e qualidade do serviço.
- [24] PEREIRA, R. et al. SWOT Analysis of a Portuguese Electronic Health Record. 12th Conference on e-Business, eServices, and e-Society (I3E). 2013.
- [25] UCHÔA, T. A diferença entre o PEC, do SUS, e o PEP. 2017