UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PALOMA GUENES COSTA
GIOVANNI LUIZ ALVES PEREIRA

ELK Saúde - Ambiente de análise e visualização de dados do SUS utilizando ElasticSearch e Kibana

RIO DE JANEIRO
2021

PALOMA GUENES COSTA
GIOVANNI LUIZ ALVES PEREIRA

ELK Saúde - Ambiente de análise e visualização de dados do SUS utilizando ElasticSearch e Kibana

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Valeria Menezes Bastos

RIO DE JANEIRO
2021

Costa, Paloma Guenes
C837e ELK Saúde: ambiente de análise e visualização de dados do SUS utilizando ElasticSearch e Kibana / Paloma Guenes Costa, Giovanni Luiz Alves Pereira. – 2021.

63 f.

Orientadora: Valéria Menezes Bastos.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2021.

1. ElasticSearch. 2. Kibana. 3. Datasus. 4. SUS. 5. Análise de dados. I. Pereira, Giovanni Luiz Alves. II. Bastos, Valéria Menezes (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática. IV. Título.

PALOMA GUENES COSTA
GIOVANNI LUIZ ALVES PEREIRA

ELK Saúde - Ambiente de análise e visualização de dados do SUS utilizando ElasticSearch e Kibana

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 08 de JANEIRO de 2021

BANCA EXAMINADORA:

Valeria Menezes Bastos

Valeria Menezes Bastos, D. Sc. (UFRJ)

Participação por vídeo-conferência

Adriana Santarosa Vivacqua, D. Sc.
(UFRJ)

Participação por vídeo-conferência

Claudia Medina Coeli, D. Sc. (UFRJ)

Participação por vídeo-conferência

Rejane Sobriño Pinheiro, D. Sc. (UFRJ)

Dedicamos este trabalho aos profissionais e pesquisadores da área da saúde e à toda população que utiliza o SUS.

AGRADECIMENTOS

Por Paloma Guenes Costa:

Agradeço à nossa orientadora que nos acompanhou e guiou com seriedade e paciência durante o trabalho todo, que nos corrigiu quando precisou e vibrou com cada conquista nossa neste projeto.

Agradeço à universidade, professores e funcionários que fazem desta instituição de ensino uma das melhores do país.

Agradeço aos professores dedicados a ensinar e que ao tocar em uma alma humana, eram apenas outra alma humana.

Agradeço à minha família que sempre acreditou no meu potencial.

Agradeço imensamente ao meu mentor favorito que esclareceu todas as minhas dúvidas e nos guiou com maestria por todos os percalços que enfrentamos.

Agradeço à minha psicanalista que tratou das minhas feridas emocionais e me mostrou o caminho para curá-las e fazer possível concluir este trabalho.

Agradeço a parceria deste trabalho que foi significativa para que chegássemos neste resultado.

Agradeço aos meus amigos dentro e fora da universidade que me deram todo apoio e compreenderam minha ausência dedicada aos estudos para que a conclusão deste trabalho pudesse acontecer.

E agradeço muito ao amigo que segurou na minha mão em momentos de ansiedade e não me deixou desistir.

Por Giovanni Luiz Alves Pereira:

Dedico minha gratidão por esse trabalho à nossa orientadora que sempre empenhada no sucesso, tanto desse trabalho como nosso como alunos e profissionais, nos acolheu e acompanhou ainda que desabitual fosse a situação para realizarmos esse projeto.

À parceira de trabalho por todo esforço conjunto e perseverança para realização do grande feito que é esse projeto.

Aos professores que me proporcionaram um rico caminhar de aprendizado durante toda minha jornada acadêmica.

À minha família que fundamentou minha vida e deu todo incentivo para que eu pudesse alcançar êxito em tudo que me fosse possível.

À minha companheira que me foi suporte, sustento e estímulo em tudo para a realização desse trabalho e de tudo que eu aspirasse.

Aos amigos que acompanharam de perto e de longe o empenho que tive na minha formação e sempre me deram a palavra de ânimo que precisava.

Ao grande e sábio amigo de quem aprendo sempre e com quem contei com toda ajuda para chegar até aqui.

"Medicine is going to become an information science. In 10 years or so, we may have billions of data points on each individual, and the real challenge will be to develop information technology that can reduce that to real hypotheses about that individual."

Leroy Hood

RESUMO

Incontáveis informações e *insights* podem ser extraídos do grande conjunto de dados do Sistema Único de Saúde (SUS). O departamento de informática do SUS, o DATASUS, faz custódia e disponibiliza informações que servem de insumo para análises e pesquisas de saúde pública. Como toda empresa com um grande volume de dados, existe um valor muito grande de conhecimento que pode ser extraído ao tratar e analisar esse conjunto de informações. E, se tratando de *big data*, esse conjunto de dados é grande demais para ser analisado por sistemas tradicionais. Alguns tabuladores disponibilizados pelo DATASUS fornecem apenas visualizações simples que não são adequadas a cargas analíticas e não permitem o cruzamento de informações de diferentes bases de dados. É importante que exista um ambiente de ETL e pesquisa amigável e que forneça visualizações robustas que possam facilitar a tomada de decisão sobre as instituições de saúde no Brasil. Este trabalho objetiva disponibilizar um ambiente de extração, transformação, carga, pesquisa e visualização, que pode ser utilizado para análises estatísticas, mineração de dados, aprendizado de máquina, *deep learning* com dados baixados do DATASUS.

Palavras-chave: ElasticSearch. Kibana. Datasus. SUS. Análise de dados.

ABSTRACT

Countless information and insights can be extracted from the large data set of the *Sistema Único de Saúde* (SUS). DATASUS, the informatics departament of SUS, custodies information that serves as input for public health analysis and research. Like any company with a large volume of data, there is a very large amount of knowledge that can be extracted by treating and analyzing this set of information. And, in the case of big data, this dataset is very large to be analyzed by traditional systems. Some tabulators made available by DATASUS provide only simple visualizations that are not appropriate to analytical load and do not allow crossing of information of different databases. It is important that there is a friendly ETL and research environment and that it provides robust visualizations that enable decision-making about health institutions in Brazil. This work aims to provide an extraction, transformation, loading, research and visualization environment that can be used for statistical analysis, data mining, machine learning, deep learning with data downloaded from DATASUS.

Keywords: ElasticSearch. Kibana. Datasus. SUS. Data Analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE CÓDIGOS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CID	Classificação Internacional de Doenças
ETL	Extract, Transform, Load
ELK	ElasticSearch, Logstash e Kibana
Fig.	Area of the i^{th} component
FTP	File Transfer Protocol
RAM	Random Access Memory
SIH	Sistema de Informações Hospitalares
SIM	Sistema de Informações sobre Mortalidade
SINASC	Sistema de Informações sobre Nascidos Vivos
SUS	Sistema Único de Saúde
SSD	Solid-State Drive

SUMÁRIO

1 INTRODUÇÃO

"O Sistema Único de Saúde (SUS) é um dos maiores e mais complexos sistemas de saúde pública do mundo (Ministério da Saúde, 2020)", permitindo o direito de acesso integral, universal e gratuito à saúde da população brasileira. Há uma quantidade muito grande de dados coletados anualmente referentes a atendimentos de saúde ocorridos em todo o país. Esses dados são disponibilizados através do portal DATASUS, que faz custódia de, pelo menos, dez bases diferentes, algumas com informações desde 1979, e para todos os vinte e seis estados brasileiros e o Distrito Federal. São exemplos de bases de dados disponíveis: Sistema de informação sobre Nascidos Vivos (SINASC), Sistema de informações sobre Mortalidade (SIM) e a base do Sistema de Informações Hospitalares do SUS (SIH). Essas bases são pertencentes ao Ministério da Saúde e a secretarias de saúde.

Por possuir dados do nascimento até a morte da população brasileira, a análise de dados de saúde pode gerar percepções importantes e ajudar a identificar padrões sobre a saúde da população. O grande volume de informações presente no DATASUS pode ser utilizado para encontrar respostas sobre algumas questões cruciais para profissionais e pesquisadores da saúde, como por exemplo: "Quais fatores contribuíram para o reaparecimento de uma Classificação Internacional de Doenças (CID), num determinado estado da federação, no ano de 2002?". Para que tais perguntas sejam respondidas, uma investigação nos dados disponíveis se faz necessária.

Trabalhar com esse grande volume de dados requer técnicas de bigdata, que se refere ao alto volume de dados com potencial para análise. Especialistas consideram importante que dados de saúde sejam abordados com ferramentas de *bigdata* (MURDOCH; DETSKY, 2013), e usar essa técnica onde a população é heterogênea, como no Brasil, pode impulsionar pesquisas relacionadas à saúde em cenários reais. (ANDREU-PEREZ, 2015)

Uma questão importante a ser considerada é o armazenamento e manutenção dos dados. Estudos concordam que um sistema de gerenciamento de dados NoSQL, ou seja, estrutura não relacional, são úteis quando se trabalha com uma grande quantidade de dados. (ANDREU-PEREZ, 2015)

Sistemas distribuídos NoSQL são designados para armazenar massivamente dados processados paralelamente. Bancos de dados desse tipo conseguem suportar múltiplas atividades, podendo ser do tipo *Extract, Load e Transform* (ETL), permitindo análises preditivas, prescritivas e exploratórias.

O Elasticsearch é um banco de dados NoSQL distribuído de texto completo. Em outras palavras, ele usa documentos em vez de esquemas ou tabelas. É uma ferramenta gratuita e de código aberto que permite pesquisar e analisar dados em tempo real. Uma aplicação de visualização de dados sugerida é o Kibana, que é *open source*, e cujo *front-end* trabalha

com o Elastic Stack, proporcionando a utilização de recursos de busca extremamente rápidos e visualização de dados indexados.

1.1 MOTIVAÇÃO

O DATASUS é o departamento de informática do Sistema Único de Saúde (SUS) do Brasil. Eles custodiam e disponibilizam informações que servem de insumo para análises e pesquisas de saúde pública. Como toda empresa com um grande volume de dados, existe um valor muito grande de conhecimento que pode ser extraído ao tratar e analisar esse conjunto de informações. E, em se tratando de *big data*, esse conjunto de dados é grande demais para ser analisado por sistemas tradicionais.

Em um mundo que está evoluindo rapidamente em inteligência computacional, *deep learning*, *machine learning* e mineração de dados, diversas soluções podem auxiliar na leitura e interpretação dos dados, melhorando os *insights* sobre as evidências obtidas através da representação gráfica desses dados. Além disso, para que informações mais consistentes sejam colhidas e exploradas por essas tecnologias modernas, mais de uma base de dados deve ser usada, permitindo o cruzamento de informações.

No contexto DATASUS, existem dois *softwares* disponibilizados para a área da saúde e a população em geral: TABNET e TABWIN. O TABNET "foi elaborado com a finalidade de permitir às equipes técnicas do Ministério da Saúde, das Secretarias Estaduais de Saúde e das Secretarias Municipais de Saúde a realização de tabulações rápidas sobre os arquivos que constituem os componentes básicos dos Sistemas de Informações do Sistema Único de Saúde, dentro de suas Intranets ou em seus sites Internet."(DATASUS, 2020b). Esse tabulador de dados pode ser acessado em <https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>. Por ele é possível obter uma análise dos dados em forma de mapa, gráfico e exportar em csv[1]. Infelizmente, só é possível analisar os dados com temas definidos previamente e não existe a possibilidade de cruzar mais de um tipo de base. Além disso, a visualização gráfica é bem básica se for comparado com o poder computacional e a quantidade de ferramentas de visualização de dados que existem atualmente no mercado.

O TABWIN "permite aos usuários realizar o cruzamento de dados dos diversos sistemas de informações em saúde, localmente, utilizando base de dados próprias para construção de indicadores, mapas, gráficos etc."(DATASUS, 2020b).No TABWIN, o pesquisador pode analisar dados obtidos com o TABNET. Com interface para resultados em duas dimensões e apenas alguns tipos básicos de análises disponíveis, a estrutura de armazenamento do TABWIN não é adequada para cargas analíticas.

Para ampliar as possibilidades de análise e gerar informação útil à tomada de decisões no setor saúde, é de interesse um ambiente de pesquisa a existência de um ambiente

[1] Um arquivo csv (comma-separeted values) tem os valores separados por vírgulas.

de pesquisas com mais potencial de exploração chamado ELK Saúde. O ELK Saúde foi projetado para atender a essas necessidades de maneira muito mais rápida, disponibilizando um banco de dados de busca acelerada. O Docker <https://www.docker.com/> é o contêiner que possui todo o ambiente necessário para análise de dados. O pacote ELK permite que os dados sejam armazenados, encontrados e analisados com o Elasticsearch, e toda a parte de visualização dos dados pode ser feita no Kibana. Dentre as vantagens de utilizar o ELK Saúde estão:

- facilidade em montar o ambiente
- pesquisas extremamente rápidas
- elaboração de dashboards iterativos
- visualização de dados em tempo real
- suporte para aprendizado de máquina e outras tecnologias de inteligência artificial

1.2 OBJETIVO

Este trabalho tem como objetivo disponibilizar um ambiente de extração, transformação e carga de três bases de dados do DATASUS e servir de ponto de partida para que outras bases sejam adicionadas posteriormente. As bases abordadas neste projeto são SIH, SINASC e SIM. Este ambiente pode ser utilizado para análises estatísticas, mineração de dados, aprendizado de máquina, *deep learning*, etc.

O DATASUS contém inúmeras bases de dados que podem ser baixadas ou analisadas por meio de ferramentas no próprio site, porém, não existe ainda um sistema adequado e capaz de fazer o cruzamento de informações de mais de uma delas para um grande volume de dados. O resultado da extração de dados do ELK Saúde serve para acelerar pesquisas, melhorar análises e, através das evidências coletadas, elaborar iniciativas que visam melhorar as políticas de saúde pública. Além disso, o projeto tem como objetivo facilitar tomadas de decisão disponibilizando um ambiente de visualização simples e funcional, para que seja possível extrair *insights* úteis a toda comunidade da área de saúde.

1.3 TRABALHOS RELACIONADOS

Existem dois trabalhos relacionados à obtenção dos dados da saúde pública, que nos últimos anos que serviram de base para este projeto. O primeiro fala sobre a utilização de um banco NoSql para fornecer um ambiente de busca e define que este banco pode ser

o MongoDB (CARVALHO; ALMEIDA, 2017). O segundo utiliza efetivamente este banco para fazer o ETL de uma das bases do SUS (SAMPAIO, 2019).

A proposta dos dois trabalhos anteriores é sobre criar um banco de dados e não são ferramentas completas como a que estamos propondo neste trabalho, em que as pesquisas são feitas de forma performática e é uma ferramenta única com armazenamento e visualização de dados.

1.3.1 Ambiente de exploração de dados de saúde usando um Banco de Dados NoSQL

O foco principal desse trabalho foi a criação de um ambiente de armazenamento dos dados saúde, obtidos por meio de bases públicas que permitissem consultas rápidas. Ele é dividido em duas partes.

O trabalho apresenta informações a respeito das bases de dados do SUS (SINASC e SIM) utilizadas no projeto, suas características e formas de obtenção. O objetivo foi apresentar a metodologia utilizada na construção do ambiente de pesquisa, sua preparação, modelagem e importação. Outro objetivo foi fazer uma comparação de tempos de acesso em consultas entre um banco de dados relacional versus um banco de dados NoSQL. Foram usados os bancos de dados PostGres (relacional) e MongoDB, que é NoSQL.

1.3.2 Ambiente de Dados do SIHSUS com MongoDB

O Ambiente de Dados do SIHSUS com o MongoDB foi feito para extrair, transformar e carregar os dados da base SIHSUS, fornecidos pelo DATASUS. O resultado da pesquisa foi uma base de dados analítica utilizando o MongoDB, visando também oferecer dados de pesquisa para a saúde pública.

Nesse projeto foi desenvolvido o Mongo Saúde, cujo projeto foi construir rotinas para fazer o ETL (Extração, Tratamento e Carga) dos dados do ambiente DATASUS para o banco de dados NoSQL, MongoDB. Todo o processo de ETL foi escrito em Python e está disponível no gitlab (https://gitlab.com/gitsaude/mongosaude) e foi desenvolvido para facilitar os usuários na tarefa de *download* e carga dos dados. O programa está dividido em 9 passos.

1. Criação de diretórios para arquivos temporários
2. *Download* da tabela escolhida e das tabelas auxiliares
3. Conversão da tabela escolhida de DBC para DBF
4. Conversão da tabela escolhida de DBF para CSV
5. Importação da tabela escolhida para o MongoDB

6. Importação das tabelas auxiliares para o MongoDB
7. Execução do pipeline do MongoDB com o ETL
8. Limpeza das tabelas auxiliares do MongoDB
9. Limpeza da pasta temporária

O Mongo Saúde possui um arquivo README.md que explica as dependências e como usar. Na pasta doc se encontra uma breve documentação com maiores detalhes sobre as dependências e como usar, inclusive com a explicação sobre os campos do arquivo de entrada. Ao instalar todas as dependências e executar o comando de ETL, o programa baixa as bases do SUS por FTP em uma pasta temporária e executa diversos passos do ETL. O *script* pode ser executado direto no terminal, sem compilação.

O arquivo de entrada é de extensão JSON e é utilizado no passo 2. Ele especifica detalhes para fazer o *download* das bases e possui todas as especificações sobre o ETL a ser feito. Existem dois arquivos de entrada prontos, o sih.json e o sinasc.json para estas duas bases, já o arquivo de entrada para a base SIM, contemplado nesse projeto, não foi produzido para o Mongo Saúde.

Na pasta do projeto o arquivo input.json possui um exemplo de arquivo de entrada, já os arquivos sih.json e sinasc.json possuem o modelo das especificações para o ETL de SIH e SINASC. Um arquivo de entrada pode conter os seguintes campos:

- **base**: Pode assumir SIM, SINASC ou SIH, bastando especificar qual base.
- **ano**: Especifica o ano ao qual a base deve ser obtida.
- **database**: Nome da database no MongoDB em que a *collection* final do ETL será colocada.
- **collection**: Nome da *collection* final do ETL no MongoDB.
- **campos**: Especifica cada campo inserido na *collection* final. É um array onde cada campo contém o nome do atributo e pode conter tabelas_auxiliares e campo_auxiliar_juncao caso o campo seja de junção.
- **campo**: Nome do campo da base a ser inserida no MongoDB.
- **tabelas_auxiliares**: Array com uma ou mais tabelas auxiliares para a execução da junção no ETL.
- **campo_auxiliar_juncao**: Campo da tabela auxiliar usado na junção.

Após fazer o *download* das bases definidas pelo arquivo de entrada descrito acima, o programa converte o tipo do arquivo, que originalmente é de extensão .dbc. Os passos 3 e 4 do programa foram destinados às conversões. No passo 3 o sistema converte os arquivos .dbc para .dbf e no passo 4 converte de .dbf para .csv.

O passo 5, 6 e 7 são referentes à etapa *Transform* e *Load* do ETL. Os passos finais são destinados a limpeza das pastas e arquivos temporários no computador, que foram criados no passo 1.

1.3.3 Asper TI

Recentemente, a Asper TI[2] desenvolveu uma solução para fornecer uma alternativa à questões relacionadas ao grande volume de dados fragmentados em diversos sistemas não integrados do DATASUS. Eles também tiveram a ideia de utilizar o que há de mais novo no mercado para buscas e visualização de dados, o pacote ELK. A Asper TI utiliza o Elastic Stack no Elastic Cloud Enterprise para extrair as informações das bases do Ministério da Saúde.

Um dos cases divulgados pela empresa no *site* deles, <https://www.asperti.com.br/leitos> foi a disponibilização para o Ministério da Saúde de *insights* e evidências de dados sobre a ocupação dos leitos do Estado de forma simples e rápida. Esse case mostra que é o ELK Saúde é capaz de agregar valor para pesquisadores da saúde.

1.4 ESTRUTURA DO DOCUMENTO

Este documento começa apresentando a introdução, motivação e objetivo do projeto para que o leitor entenda o contexto principal. Também são mencionados trabalhos relacionados e ferramentas disponíveis no site do DATASUS para análise dos dados. No capítulo 2, estado da arte, algumas ferramentas e tecnologias utilizadas no projeto são descritas. O capítulo seguinte explica como a solução foi projetada e mostra a utilização do processo de ETL para o ambiente Elastic Stack. Apresenta todo o processo a partir do início do ETL: busca dos dados, extração, transformação e carga. Logo após, alguns exemplos de visualização do cruzamento dos dados das bases do DATASUS são apresentados. No capítulo 5 é descrita a conclusão do projeto é algumas sugestões de trabalhos futuros que irão agregar um valor maior ao sistema de saúde pública.

[2] Empresa privada parceira estratégica da Elastic no Brasil.

2 ESTADO DA ARTE

Para que seja possível construir toda a estrutura deste ambiente de pesquisa, foi necessário criar um processo de ETL: converter arquivos, formatar dados das bases e importar essas informações no Elasticsearch. Aqui descrevemos o estado da arte das técnicas e tecnologias utilizadas neste processo.

2.1 ETL

ETL é o acrônimo para Extract, Transform e Load e é a sistematização do processo de extração de dados, a transformação e tratamento, conforme regras previstas pelo negócio, e o carregamento de dados para um sistema de organização destes.

A extração compreende o segmento em que pacotes de dados são extraídos de um ambiente de origem desses dados e prepara esses para iniciar a transformação. No projeto desenvolvido nesse trabalho, compreende a seção de Data Loading. A transformação inclui diversos procedimentos menores de tratamento de dados: desde seleção desses, junção entre dados de fontes diferentes, enriquecimento e até indexação. Nesse trabalho, a transformação compõe a seção de Data Transform. Por fim, é em Data Ingestion que é realizada o carregamento dos dados. Nesse segmento, após o tratamento prévio dos dados, esses são persistidos em uma base, podendo manter histórico e versionamento desses.

2.2 PYTHON

A automatização de ETL se faz necessária tendo em vista que se tornam repetitivos os procedimentos menores que compõe essa processo de integraçao de dados. Conversão de arquivos, adequação de valores, enriquecimento dos dados são apenas algumas das atividades realizadas durante o ETL que podem ser desempenhadas de maneira iterada. Essa automatização é feita de maneira programada em Python. A linguagem de programação escolhida é de alto nível, de fácil leitura e está entre as mais populares no mundo (REDMONK, 2020), além de permitir diversas integrações e disponibilizar acesso às mais diferentes bibliotecas, permitindo versatilidade no desenvolvimento de aplicações. No projeto, essa linguagem foi aplicada ao longo de todo o processo de Data Loading, Data Transform e Data Ingestion, atendendo ao que se compreende pelo ETL proposto.

2.3 PANDAS

Pandas é uma biblioteca muito utilizada e poderosa para análise de dados desenvolvida em Python. Ela é flexível, prática e fácil de usar, e foi feita para atuar em data visuali-

zation e data analysis. Existe uma documentação bem completa sobre essa biblioteca no site <https://pandas.pydata.org/docs/user_guide/index.html>, ensinando com detalhes desde como instalar até utilizar as funções mais complexas. Além disso, este site contém *links* para tirar dúvidas nos fóruns da comunidade.

Neste projeto usamos essa biblioteca para manipular os dados quando tranformados em csv. Utilizamos dataframes para tratar os dados estruturados em duas dimensões (linhas e colunas). Para ler um arquivo neste formato, basta informar o nome dele para uma instância da biblioteca:

```
pandas.read_csv('filename.csv')
```

2.4 FTP

Toda o conteúdo de dados disponibilizados através do Datasus é feito por meio de FTP. Sendo esse um protocolo da camada de aplicação utilizado para transferência de arquivos (IETF, 2020), ao ser usado por um cliente, estabelece uma conexão que permite o acesso aos arquivos disponívis no servidor. O servidor FTP do Datasus se encontra no endereço ftp.datasus.gov.br e disponibiliza os dados anônimos de diversas bases públicas, que nesse trabalho compreende a camada de Data Source da arquitetura desenvolvida e o acesso à esse servidor FTP é realizado na camada de Data Loading.

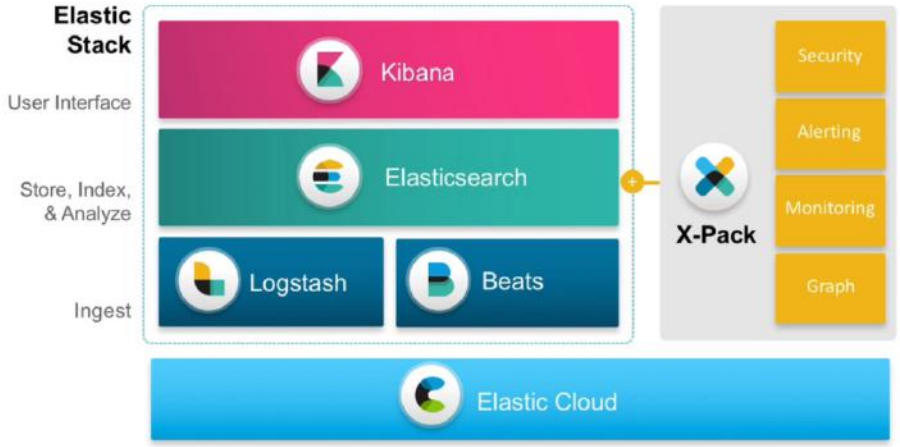
2.5 ELK STACK

ELK Stack ou Elastic Stack é o conjunto das ferramentas Elasticsearch, Logstash e Kibana e demais agentes de dados que fornecidos pela Elastic como mostrado na figura 1. Essas ferramentas podem ser integradas para gestão de grande volumes de dados: desde seu processamento e ingestão, assim como permitir busca e análise desses dados e também sua devida visualização.

2.5.1 Elastic Search

O Elastic Search é um mecanismo de código aberto para análise e busca de grande quantidade de dados e tem como características notáveis velocidade e escalabilidade distribuídas. É baseado em Lucene, uma biblioteca de mecanismo de pesquisa escrita em Java. É parte principal do Elastic Stack e opera através de indexação de dados para possibilitar execução ágil de consultas complexas propiciando que análises e visualizações desses dados possam ser implementadas.

Figura 1 – Representação do conjunto de ferramentas do Elastic Stack



Fonte: https://www.margo-group.com/en/news/establishment-of-a-centralised-log-management-platform-with-the-elastic-suite/

2.5.2 Kibana

"Data visualization é a representação gráfica da informação e dos dados" (TABLEAU, 2020) . Para que esse grande volume de dados faça sentido para os usuários finais, é importante que a representação visual dos dados fosse feita no Kibana. O Kibana é uma aplicacação *open-source* integrada com o Elastic Search que viabiliza visualização dos dados indexados no cluster do Elastic. Comumente usado como interface gráfica para prover dashboards que permitam a busca dos dados ele proporciona, através de elementos visuais numa interface web, análises, monitoramentos e logs que podem ser acessados pelo usuário e, assim, permitindo consultas a diversas informações que os dados podem revelar.

2.6 DOCKER

Docker é uma tecnologia open source baseada em contêiners, lançada em 2013, que "concentra tudo que é necessário para executar uma aplicação: código, *runtime*, ferramentas do sistema, bibliotecas e configurações."(DOCKER, 2020) Cada contêiner docker pode ser considerado um pacote, ou uma máquina virtual excepcionalmente leve, que contém o ambiente que o sistema/projeto precisa para ser executado com segurança e garantia em qualquer computador. Docker contém uma interface simples e, segundo Thanh Bui[1] é seguro mesmo com suas configurações padrão.

A utilização do Docker no mercado é bastante popular, tendo sido parte de sistemas amplamente conhecidos como Uber, Spotify e PayPal. Muito comparado com máquinas virtuais porque ambos incluem a aplicação, as bibliotecas e arquivos necessários, o Docker é destinado a resolver problemas diferentes de uma VM. Um contêiner Docker é utilizado

[1] Thanh Bui, Analysis of Docker Security, 2015.

para isolar aplicações individuais, enquanto máquinas virtuais isolam sistemas inteiros, já que necessitam de um sistema operacional. Além disso, utilizar um contêiner é mais eficiente e podem reduzir substancialmente o tempo de implantação de um sistema. Outro ponto importante é que não é necessário possuir um grande conhecimento de implantação de ambientes para utilizar um docker.

A aplicação de um contêiner docker neste projeto é de grande importância para facilitar a instalação, execução e visualização de todo conteúdo baixado no site do DATASUS, o que significa que, ao instalar o docker, nenhuma configuração prévia da máquina utilizada deverá conflitar com o que está dentro do contêiner e não é necessário um grande conhecimento técnico para fazer uso do sistema.

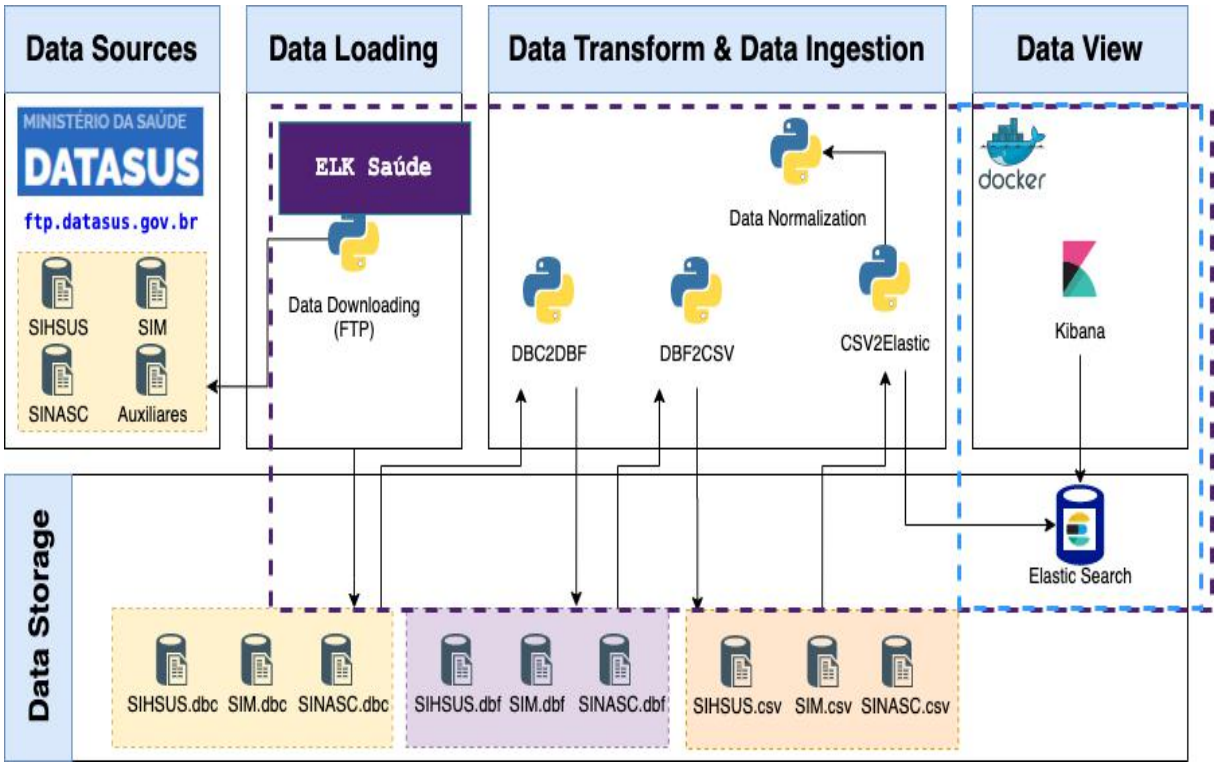
3 A APLICAÇÃO - CRIAÇÃO DO AMBIENTE NO ELASTIC SEARCH

Aqui descrevemos o que desenvolvemos para implementar o ambiente, começando pela arquitetura proposta.

3.1 ARQUITETURA DA SOLUÇÃO

A arquitetura desenvolvida para o ELK Saúde segue o desenho da Figura 2, e se divide em Data Sources, Data Loading, Data Storage, Data Transform e Data Ingestion e Data Visualization.

Figura 2 – Arquitetura da Solução



Fonte: Os autores (2020)

3.1.1 Data Source

O Data Source é o DATASUS do Ministério da Saúde, e todas as bases estudadas neste documento são obtidas acessando o endereço <ftp.datasus.gov.br>. Para este projeto, as bases obtidas foram:

- **Sistema de Informações Hospitalares do SUS** (SIHSUS): "registra todos os dados provenientes das internações provenientes de internações hospitalares financi-

adas pelo SUS, e a partir deste processamento, são gerados os relatórios para que os gestores possam fazer os pagamentos dos estabelecimentos de saúde." (DATASUS, 2020a)

- **Sistema de Informações de Mortalidade** (SIM): "obtenção de cada óbito ocorrido no país." (DATASUS, 2020a)
- **Sistema de Informações de Nascidos Vivos** (SINASC): "reúne informações epidemiológicas referentes aos nascimentos informados em todo território nacional." (DATASUS, 2020a)
- **Tabelas auxiliares**: contém dados que auxiliam no entendimento de outros dados. Todos os conjuntos de dados citados acima possuem uma relação com tabelas auxiliares, sendo que algumas complementam os dados das tabelas do SUS. Pelo *site* do datasus, essas tabelas são encontradas ao selecionar "Arquivos Auxiliares de Tabulação.", como em <http://www2.datasus.gov.br/DATASUS/index.php?area=0901&item=1&acao=25>.

3.1.2 Data Loading

A extração dos dados do DATASUS é feita por um endereço utilizando o protocolo ftp que " basicamente estabelece uma conexão que permite a troca de arquivos entre dois computadores conectados à internet." (HOSTINGER, 2020). Esse processo foi feito em Python e é parte do projeto ELK Saúde.

3.1.3 Data Storage

Data Storage significa armazenamento de dados, e se refere à persistência das informações. O armazenamento dos dados é feito nesse ponto. O ELK Saúde utiliza duas abordagens de armazenamento: temporária e permanente. Durante a execução, alguns arquivos são salvos numa pasta temporária, criada logo no início, para auxiliar o processo de conversão da extensão dos dados. No último passo do ETL, o armazenamento dos dados já normalizados é feito no ELK Saúde.

3.1.4 Data Transform e Data Ingestion

Data Transform ou transformação dos dados, é a etapa onde eles são modificados e preparados para a inserção no banco de dados (Data Ingestion). Nesse ponto são feitas as conversões dos arquivos e a normalização dos dados. Esse processo também foi feito em Python.

3.1.5 Data Visualization

A visualização dos dados é feita utilizando o Kibana, que é parte do projeto ELK. Essa ferramenta permite ao usuário verificar o comportamento dos dados em tempo real, através de mapas, histogramas, *dashboards*, gráficos de linhas e de pizza.

3.1.6 Container Docker

"Docker é uma ferramenta projetada para facilitar a criação, *deploy*, e execução de aplicaçoes utilizando conteiners."(OPENSOURCE, 2020). Cada contêiner de Docker pode ser comparado a um pacote que contém os recursos necessários instalados para que o sistema funcione. Nessa arquitetura, todo o processamento de dados e armazenamento final é feito dentro de um contêiner docker para facilitar a instalação e execução do projeto. Dentro de um docker foram concentradas todas as bibliotecas e dependências que o código fonte precisa, assim como toda a estrutura do ELK.

3.2 ARQUIVOS DE ENTRADA

A execução do ETL é parametrizada, através de um arquivo *json*, que especifica as condições das atividades desempenhadas pela aplicação. Os parâmetros definem quais arquivos das bases serão buscados e quais conversões serão feitas dentro desse escopo, além de definir quais enriquecimentos serão feitos nos dados, através da inclusão das tabelas auxiliares disponibilizadas pelo DATASUS. Tendo como modelo inicial a parametrização feita no trabalho anterior, o arquivo indica a base a ser trabalhada, o período de referência dos dados, assim como os campos selecionados e os critérios para associação desses com as tabelas auxiliares, da seguinte forma:

- base - Indica a base trabalhada pelo ETL, podendo assumir os valores "SIH", "SINASC"ou "SIM".
- ano - Indica o ano de referência dos dados disponibilizados pelo DATASUS que serão extraídos e carregados dos Data Sources.
- elastic_server - Indica o *publish id port* que é o endereço do ElasticSearch na máquina local. O padrão é "http://localhost:[número da porta do elk]".
- kibana_server - Indica o *publish id port* que é o endereço do Kibana na máquina local. O padrão é "http://localhost:[número da porta do kibana]".
- campos - Indica a lista de campos selecionados na base escolhida e que devem ser persistidos pelo ETL, bem como informações de suas tabelas auxiliares correspondentes, seguindo os parâmetros:

- campo - Indica o nome do campo na entidade da base.
- tipo_campo - Indica o tipo do dado nesse campo quando presente nos arquivos dos Data Sources, previamente informado pelo DATASUS, através da documentação no portal.
- formato_campo - Indica o formato do campo no caso de possuir valor previamente formatado.
- tipo_campo_convertido - Indica o tipo a ser convertido do valor desse campo, respeitando-se o formato se houver.
- tabelas_auxiliares - Lista o nome dos arquivos nos quais estão presentes os dados das tabelas auxiliares associadas a esse campo.
- campo_auxiliar_juncao - Indica o nome do campo na tabela auxiliar que servirá de chave primária para a associação da base com a tabela auxiliar.
- campo_valor_auxiliar - Indica o nome do campo na tabela auxiliar que servirá de valor correspondente do campo na tabela auxiliar.
- tipo_valor_auxiliar - Indica o tipo do campo de valor correspondente do campo na tabela auxiliar.

No código A.1 no anexo A desse trabalho encontra-se um arquivo json de exemplo, apresentando todos esses parâmetros sendo utilizados.

3.2.1 BASES DE DADOS DO DATASUS

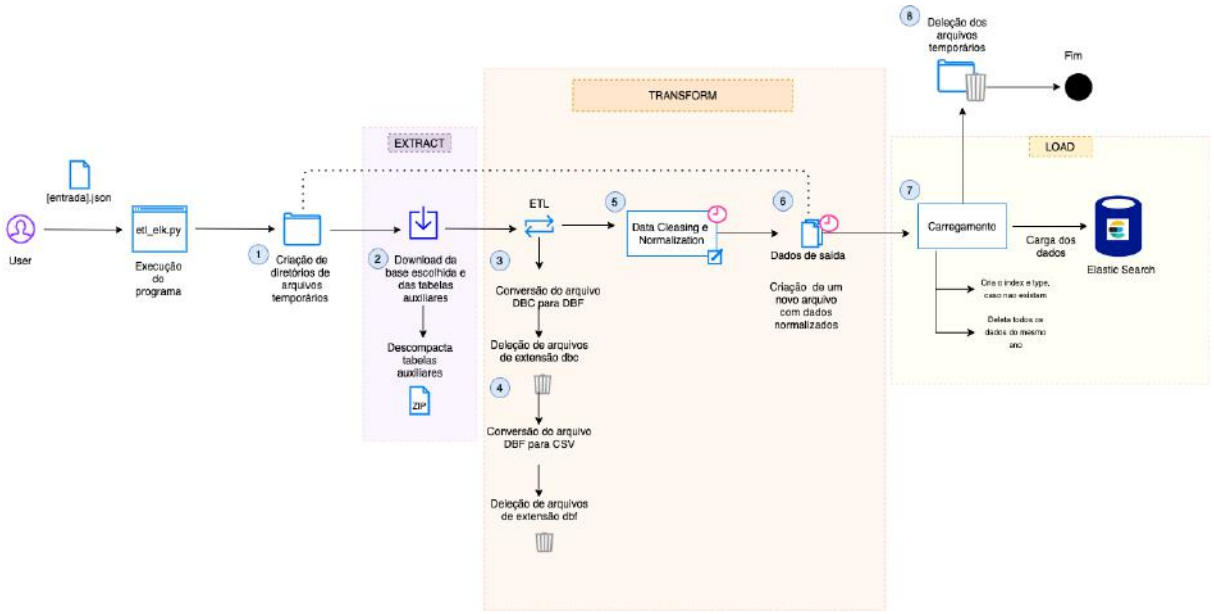
O DATASUS é o departamento do Sistema Único de Saúde que provê suporte tecnológico e também disponibiliza através de seu portal dados da situação sanitária da população do Brasil. Entre as bases de dados disponibilizadas estão: o Sistema de Informação sobre Nascidos Vivos (SINASC), o Sistema de Informações Hospitalares do SUS (SIH) e o Sistema de Informações sobre Mortalidade (SIM), que serão utilizadas nesse trabalho.

Dessas bases, foi feita uma seleção de quais dados seriam persistidos, tendo em vista os trabalhos anteriores, (CARVALHO; ALMEIDA, 2017) e (SAMPAIO, 2019) , onde houve um levantamento feito por especialistas da área da saúde coletiva. Dessa maneira, pode-se estruturar através dos modelos de dados descritos nos diagramas de entidade e relacionamento anexados neste trabalho.

3.3 PIPELINE DE EXECUÇÃO

O código desenvolvido para o ELK Saúde segue o pipeline de execução descrito na figura 3.

Figura 3 – Pipeline de Execução



Fonte: Os autores (2020)

Para iniciar a primeira etapa do fluxo de execução, o usuário precisa executar o código disponível no repositório desse projeto[1], passando um arquivo de entrada como parâmetro para que o sistema tenha as informações mínimas para buscar os dados no site do DATASUS. O comando **python etl_elk.py sinasc.json** (python, seguido do nome do programa etl_elk.py, e do nome do arquivo de entrada json) permite que os passos a seguir sejam executados sequencialmente.

1. Criação de diretórios e arquivos temporários. As pastas e arquivos criados nesse passo servem para incluir os dados baixados do site do DATASUS por FTP. Também é feito o download das tabelas auxiliares. Esses diretórios são utilizados durante toda execução do programa, efetuando as conversões dos arquivos e a geração do arquivo final.

2. O passo 2 é onde começa o ETL, mais precisamente o *"extract"*. Neste passo ocorre o download dos arquivos de acordo com a base escolhida no arquivo de entrada. Todos os parâmetros necessários, como ano e cidade, são considerados no processo. Os arquivos são baixados como .zip, e descompactados em seguida.

3. No passo 3 começa a etapa *"transform"*, de conversão dos arquivos. Primeiro, os arquivos são convertidos de .dbc para .dbf e em seguida os arquivos .dbc são deletados.

[1] https://bitbucket.org/pguenes/elksaude/src/master/

4. No passo 4 os arquivos são convertidos de .dbf para .csv, extensão em que os arquivos serão manipulados no programa.

5. O passo 5 é responsável pela limpeza e normalização dos dados. Os dados das tabelas auxiliares são padronizados para que possam ser comparados e substituídos no passo 6.

6. Neste passo, os dados da base são lidos e guardados em uma variável para que sejam manipulados e substituídos por dados das tabelas auxiliares. Neste passo, é aplicado um filtro em todas as colunas para que sejam utilizadas apenas as que estão no json de entrada. Os códigos de data são formatados para que sejam entendidos como datas e, finalmente, o arquivo de saída é gerado.

7. Na etapa de carregamento *"loading"* dos dados no Elasticsearch, o *index* e o *type* daquela base são criados e, caso existam, são deletados primeiro. Em seguida, os dados são inseridos no banco. O index "é como um '*database*' em um banco relacional. Ele possui um mapeamento que define múltiplos *types*."(ELASTICSEARCH, 2020d). *Types* "são uma maneira conveniente de armazenar vários tipos de dados no mesmo índice, a fim de manter baixo o número total de índices"(ELASTICSEARCH, 2020c).

8. No passo 8 todos os arquivos e diretórios temporários criados no computador local no passo 1 são apagados.

3.3.1 Dados técnicos da execução

O sistema permite o carregamento de um arquivo de entrada por vez, ou seja, uma base é escolhida para ser baixada para cada execução do ELK Saúde. O log apresentado em 3.1 mostra a execução do ELK Saúde para a base SINASC, nos anos de 2010 a 2015. A duração total da execução foi de 17 minutos e 24 segundos em uma máquina com sistema operacional *MacOs*, processador core i7, 16 GB de RAM[2] e 512 GB SSD[3]. Todos os passos foram executados, com exceção do download dos arquivos. Este log serve para mostrar a duração das etapas do sistema e a velocidade de carga dos dados.

[2] Significa random access memory, armazenamento volátil.
[3] Significa solid-state drive, armazenamento em disco.

Código 3.1 – Log gerado na carga de uma base do DATASUS

```
\UseRawInputEncoding

1 de 9:Criando diretorios para arquivos temporarios
Duracao de descompactar Zip: 0:00:00.339754

3 de 9:Convers o da tabela SINASC DBC->DBF
Duracao de Conversao de DBC->DBF: 0:00:06.961839

4 de 9:Convers o da tabela SINASC DBF->CSV
Duracao de Conversao de DBF->CSV: 0:05:58.682499

Passo  5: Limpeza  e normalizacao dos dados das tabelas auxiliares -
    este processamento leva tempo
Duracao do tratamento das tabelas auxiliares e escrita de campos:
    0:00:02.056533

Passo  6: Limpeza  e normalizacao dos arquivos da base original - este
    processamento leva tempo
Duracao de tratamento do arquivo: 0:01:31.291360 >>> DNRJ2012.csv
Duracao de tratamento do arquivo: 0:01:40.662219 >>> DNRJ2013.csv
Duracao de tratamento do arquivo: 0:01:35.206872 >>> DNRJ2011.csv
Duracao de tratamento do arquivo: 0:01:28.568457 >>> DNRJ2010.csv
Duracao de tratamento do arquivo: 0:01:50.180017 >>> DNRJ2014.csv
Duracao de tratamento do arquivo: 0:01:07.207311 >>> DNRJ2015.csv

7 de 9:Importacao da tabela DNRJ2012.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:22.117755 >>> arquivo: DNRJ2012.csv

7 de 9:Importacao da tabela DNRJ2013.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:24.031890 >>> arquivo: DNRJ2013.csv

7 de 9:Importacao da tabela DNRJ2011.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:20.526802 >>> arquivo: DNRJ2011.csv

7 de 9:Importacao da tabela DNRJ2010.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:18.670484 >>> arquivo: DNRJ2010.csv

7 de 9:Importacao da tabela DNRJ2014.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:25.064323 >>> arquivo: DNRJ2014.csv

7 de 9:Importacao da tabela DNRJ2015.csv para o ElasticSearch
Duracao de carregamento do arquivo para o elasticsearch e gera  o do
    arquivo de saida: 0:00:13.131026 >>> arquivo: DNRJ2015.csv
```

A figura 4 a seguir, apresenta uma comparação do tamanho dos arquivos baixados e dos arquivos de saída enriquecidos para a base SINASC. Um total de 273,5 MB de dados foi baixado do DATASUS e processado no ELK Saúde. Os dados de saída correspondem a 421,1 MB, o que representa, em média, por base, 24,5 MB a mais de dados complementares incorporados na base final.

Figura 4 – Dados de entrada e saída SINASC

Fonte: Os autores (2020)

3.3.2 Carregamento dos dados no ELK

Neste estudo, utilizou-se um *bulk helper*, que serve para carregar arquivos em massa para o Elastic Search. *Bulk helpers* são "funções auxiliares simples que abstraem algumas especificidades da API *raw*"(ELASTICSEARCH, 2020b). A figura 5 mostra a quantidade de dados inicial de 1.3 milhões de registros para serem analisados e visualizados no dashboard. São dados da SINASC de 2010 à 2015.

Figura 5 – Dados no Discover - Kibana

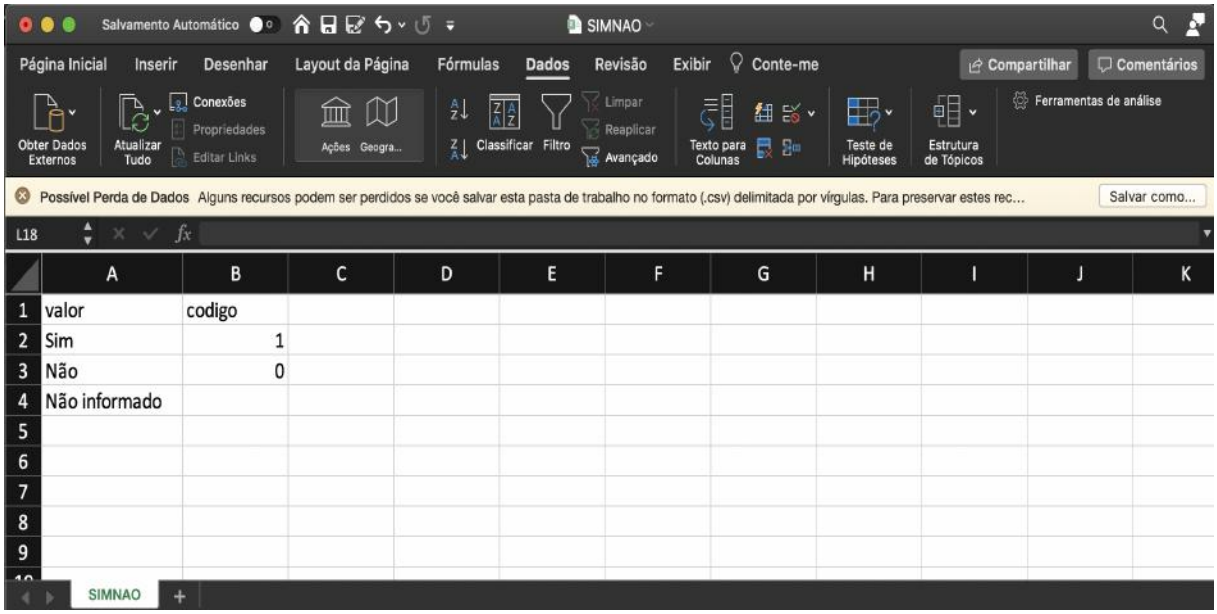
Fonte: Os autores (2020)

3.3.3 Limpeza e Normalização das Tabelas Auxiliares

A tabela base, que foi baixada do DATASUS, contém diversas colunas com códigos cujo significado está nas tabelas auxiliares, que os complementam com uma descrição. Ou seja, os arquivos auxiliares possuem pelo menos duas colunas: código e valor, como mostra a figura 6, já a base possui colunas que contém somente os códigos, como destacado na figura 7.

No passo 6, ao fazer a leitura e conversão das tabelas auxiliares, algumas delas precisaram de tratamento especial para que os dados pudessem ser convertidos corretamente.

Figura 6 – Tabela auxiliar SIMNAO



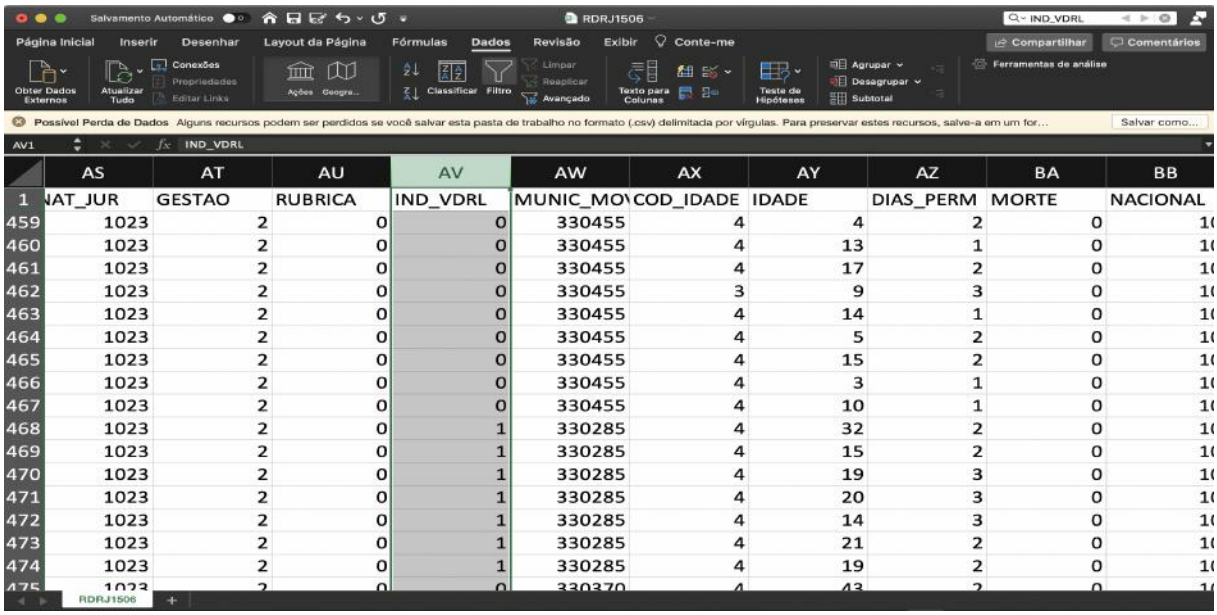
Fonte: Os autores (2020)

Ficou decidido durante e execução que seria mais enriquecedor para o usuário final se ele pudesse ver tanto o dado *raw* (puro, como vem da tabela que foi baixada do DATASUS), quanto o dado convertido. Por isso, a necessidade de criar colunas a mais com "_desc"para que ficasse claro que esses dados foram convertidos pelo sistema. O resultado da conversão é uma coluna a mais na tabela de saída, IND_VDLR_desc, contendo apenas os valores: Sim, Não e vazio.

Para que os dados de saída sejam compreensíveis, é importante que as informações contidas nas tabelas auxiliares tenham um único código (ou chave) associado a uma descrição. Não é compreensível para o sistema, e nem para quem faz essa conversão visualmente, quando uma chave possui dois valores associados a ela.

Durante o experimento, várias tabelas auxiliares possuíam chaves duplicadas e cada uma delas precisou de um tratamento especial, como mostrado a seguir.

Figura 7 – SIH 2015 - Tabela base



	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB
1	NAT_JUR	GESTAO	RUBRICA	IND_VDRL	MUNIC_MO	COD_IDADE	IDADE	DIAS_PERM	MORTE	NACIONAL
459	1023	2	0	0	330455	4	4	2	0	1
460	1023	2	0	0	330455	4	13	1	0	1
461	1023	2	0	0	330455	4	17	2	0	1
462	1023	2	0	0	330455	3	9	3	0	1
463	1023	2	0	0	330455	4	14	1	0	1
464	1023	2	0	0	330455	4	5	2	0	1
465	1023	2	0	0	330455	4	15	2	0	1
466	1023	2	0	0	330455	4	3	1	0	1
467	1023	2	0	0	330455	4	10	1	0	1
468	1023	2	0	1	330285	4	32	2	0	1
469	1023	2	0	1	330285	4	15	2	0	1
470	1023	2	0	1	330285	4	19	3	0	1
471	1023	2	0	1	330285	4	20	3	0	1
472	1023	2	0	1	330285	4	14	3	0	1
473	1023	2	0	1	330285	4	21	2	0	1
474	1023	2	0	1	330285	4	19	2	0	1
475	1023	2	0	0	330370	4	43	2	0	1

Fonte: Os autores (2020)

3.3.3.1 Chaves duplicadas para valores ignorados

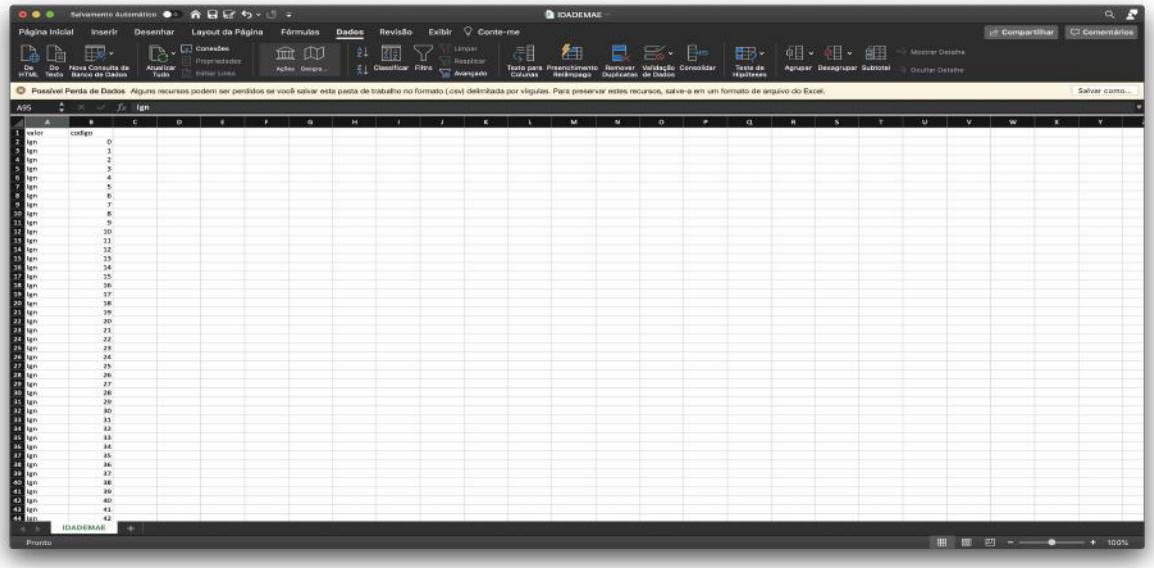
Algumas tabelas possuem códigos relacionados a valores como "Ignorado", "ign", "Não Informado", "N inf"que possuem, além dessa descrição, um significado substancial. Um exemplo é a tabela auxiliar IDADEMAE, encontrada no json de entrada da SINASC. Essa tabela objetiva classificar as idades das mães, que são números inteiros, em faixas etárias. Essa tabela possui duas colunas, chave e valor, e a chave deve ser um identificador único. As figuras 8 e 9 mostram a tabela IDADEMAE com chaves duplicadas e associadas a valores Ign.

Essas linhas com ign foram apagadas para que as chaves estejam associadas a apenas um valor que faça sentido na faixa etária correspondente. Outras tabelas precisaram de ajustes similares a esses.

3.3.3.2 Chaves duplicadas para valores importantes

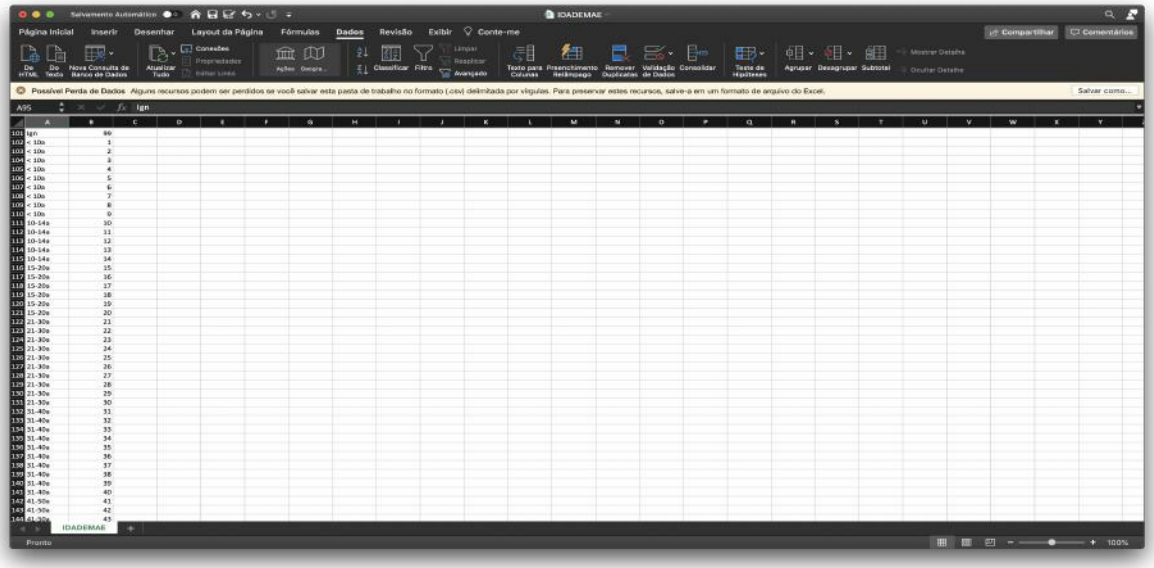
A tabela auxiliar NATUR, encontrada nos dados da SIH, possui chaves duplicadas para valores que aparentemente são necessários para o usuário final. A figura 10 mostra que o código 73 está relacionado à descrição "EIRE"e "IRLANDA". A solução utilizada para este tipo de problema foi juntar estes valores, como se fossem uma única descrição, como mostra a figura 11.

Figura 8 – Tabela Auxiliar Idade Mãe - Chaves duplicadas associadas a Ign



Fonte: Os autores (2020)

Figura 9 – Tabela Auxiliar Idade Mãe - Chaves Corretas

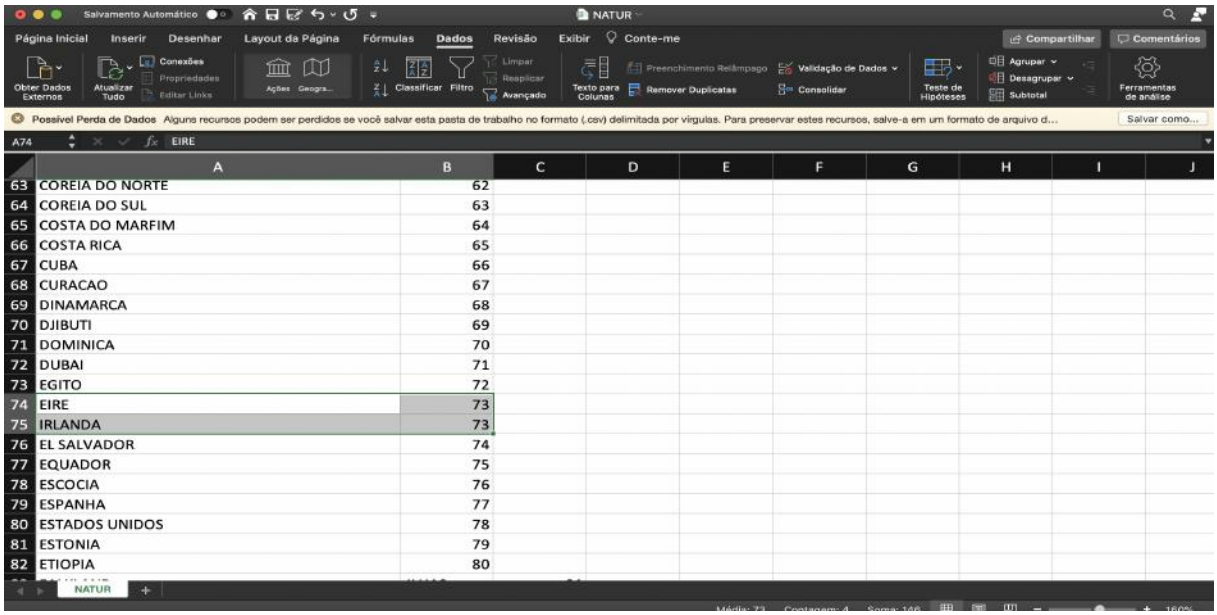


Fonte: Os autores (2020)

3.3.3.3 Linhas com vírgula

Os arquivos utilizados para carga no ELK Saúde são conhecidos como .csv - *comma separated values*, ou seja, em cada linha o código é separado do valor por uma vírgula. Algumas tabelas auxiliares possuem mais de uma vírgula, o que dificulta a análise da coluna "codigo"corretamente. A figura 12 mostra um exemplo que ocorreu na tabela

Figura 10 – Tabela Auxiliar Natur - Chaves duplicadas



	A	B
63	COREIA DO NORTE	62
64	COREIA DO SUL	63
65	COSTA DO MARFIM	64
66	COSTA RICA	65
67	CUBA	66
68	CURACAO	67
69	DINAMARCA	68
70	DJIBUTI	69
71	DOMINICA	70
72	DUBAI	71
73	EGITO	72
74	EIRE	73
75	IRLANDA	73
76	EL SALVADOR	74
77	EQUADOR	75
78	ESCOCIA	76
79	ESPANHA	77
80	ESTADOS UNIDOS	78
81	ESTONIA	79
82	ETIOPIA	80

Fonte: Os autores (2020)

Figura 11 – Tabela Auxiliar Natur - Solução

	valor	codigo
277	PERNAMBUCO	826
278	ALAGOAS	827
279	SERGIPE	828
280	BAHIA	829
281	MINAS GERAIS	831
282	ESPIRITO SANTO	832
283	RIO DE JANEIRO	833
284	SAO PAULO	835
285	SANTA CATARINA	842
286	RIO GRANDE DO SUL	843
287	MATO GROSSO DO SUL	850
288	MATO GROSSO	851
289	GOIAS	852
290	PARANA	841
291	DISTRITO FEDERAL	853
292	N Inf	nan
0	CAMBOJA | LAOS	044
1	EIRE | IRLANDA	073
2	FALKLAND; ILHAS | MALVINAS; ILHAS	081

Fonte: Os autores (2020)

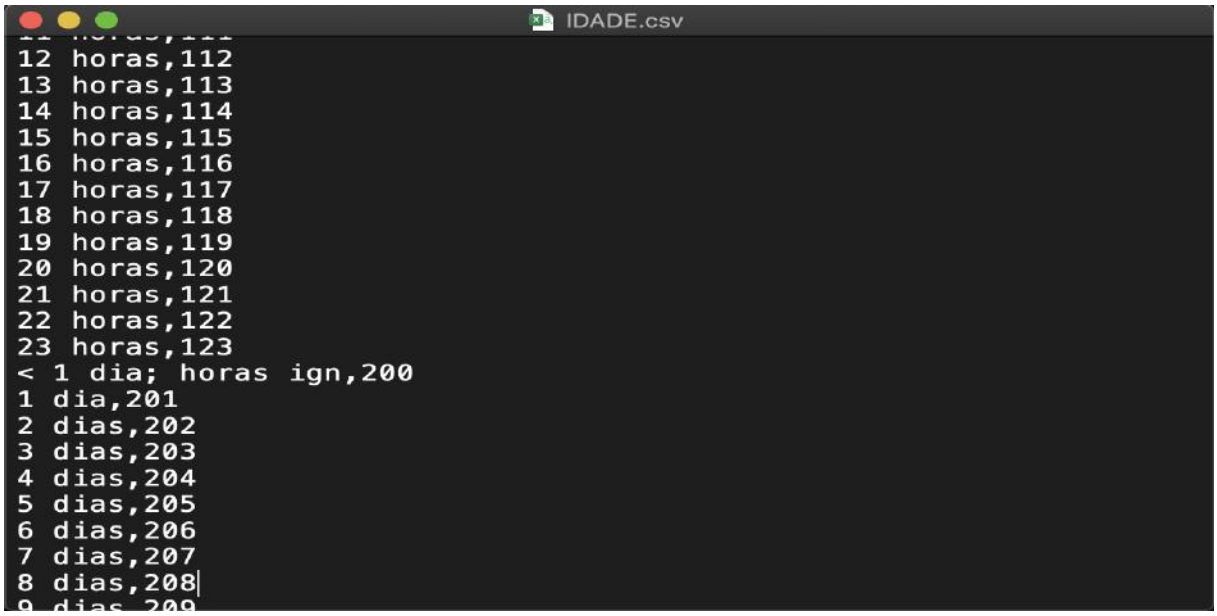
auxiliar IDADE na base SIM. A solução foi criar uma função que substituísse essa primeira vírgula por um outro caractere (o escolhido foi o ";") para que a chave correta estivesse associada à descrição, como mostra o *dataframe*[4] da figura 13.

[4] Um dataframe é uma estrutura da biblioteca pandas que armazena os dados das tabelas lidas em csv. Todas as tabelas auxiliares são transformadas em dataframes em tempo de execução.

Figura 12 – Tabela Auxiliar IDADE

Fonte: Os autores (2020)

Figura 13 – Dataframe da Tabela Auxiliar IDADE - Solução



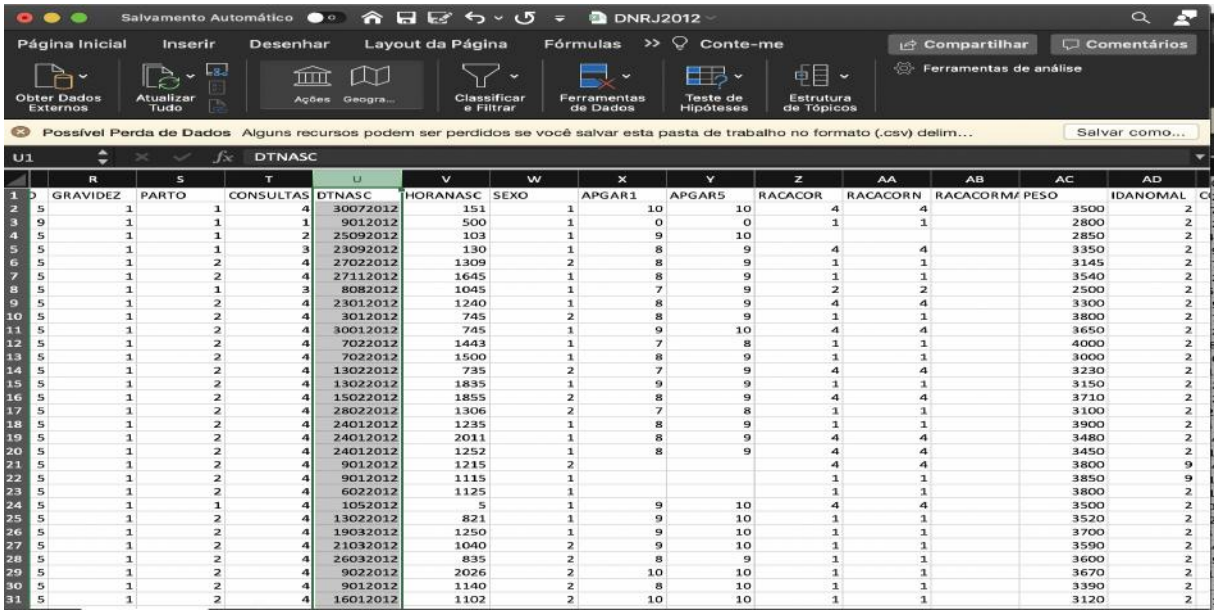
Fonte: Os autores (2020)

3.3.3.4 Formatação das datas

A tabela base possui colunas com datas que não estão claramente decifráveis e são apenas números do tipo "1042015". Por isso, foi necessário criar uma função para formatá-las e deixá-las no padrão dia/mês/ano. Ao analisar a coluna DTNASC da base SINASC, foi identificado que o padrão de uma data completa contém oito dígitos: dois para o dia,

dois para o mês e quatro para o ano. Como pode ser visto na figura 14, algumas datas possuem apenas sete números. Observando a linha 10, "3012012", não se sabe ao certo se a data se trata de 30 de janeiro de 2012 ou 3 de janeiro de 2012. Porém, na linha 8, "8082012", as opções de data seriam 80 de fevereiro de 2012 ou 8 de agosto de 2012, sendo que a primeira opção não é correta. Desta forma, foi identificado que, mesmo com 7 dígitos, as datas possuem um padrão: o mês sempre tem dois números. A função que formata as datas, em 3.2, foi construída considerando este ponto, e, antes de transformar em data, todas foram formatadas para possuírem 8 dígitos, apenas colocando um zero no começo.

Figura 14 – Coluna DTNASC da sinasc



	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	GRAVIDEZ	PARTO	CONSULTAS	DTNASC	HORANASC	SEXO	APGAR1	APGAR5	RACACOR	RACACORN	RACACORMA	PESO	IDANOMAL
2	1	1	4	30072012	151	1	10	10	4	4		3500	2
3	1	1	1	9012012	500	1	0	0	1	1		2800	2
4	1	1	2	25092012	103	1	9	10				2850	2
5	1	1	3	23092012	130	1	8	9	4	4		3350	2
6	1	2	4	27022012	1309	2	8	9	1	1		3145	2
7	1	2	4	27112012	1645	1	8	9	1	1		3540	2
8	1	1	3	8082012	1045	1	7	9	2	2		2500	2
9	1	2	4	23012012	1240	1	8	9	4	4		3300	2
10	1	2	4	3012012	745	2	8	9	1	1		3800	2
11	1	2	4	30012012	745	1	9	10	4	4		3650	2
12	1	2	4	7022012	1443	1	7	8	1	1		4000	2
13	1	2	4	7022012	1500	1	8	9	1	1		3000	2
14	1	2	4	13022012	735	2	7	9	4	4		3230	2
15	1	2	4	13022012	1835	1	9	9	1	1		3150	2
16	1	2	4	15022012	1855	2	8	9	4	4		3710	2
17	1	2	4	28022012	1306	2	7	8	1	1		3100	2
18	1	2	4	24012012	1235	1	8	9	1	1		3900	2
19	1	2	4	24012012	2011	1	8	9	4	4		3480	2
20	1	2	4	24012012	1252	1	8	9	4	4		3450	2
21	1	2	4	9012012	1215	2			4	4		3800	9
22	1	2	4	9012012	1115	1			1	1		3850	9
23	1	2	4	6022012	1125	1			1	1		3800	2
24	1	1	4	1052012	5	1	9	10	4	4		3500	2
25	1	2	4	13022012	821	1	9	10	1	1		3520	2
26	1	2	4	19032012	1250	1	9	10	1	1		3700	2
27	1	2	4	21032012	1040	2	9	10	1	1		3590	2
28	1	2	4	26032012	835	2	8	9	1	1		3600	2
29	1	2	4	9022012	2026	2	10	10	1	1		3670	2
30	1	2	4	9012012	1140	2	8	10	1	1		3390	2
31	1	2	4	16012012	1102	2	10	10	1	1		3120	2

Fonte: Os autores (2020)

Código 3.2 – Exemplo em Linguagem Python

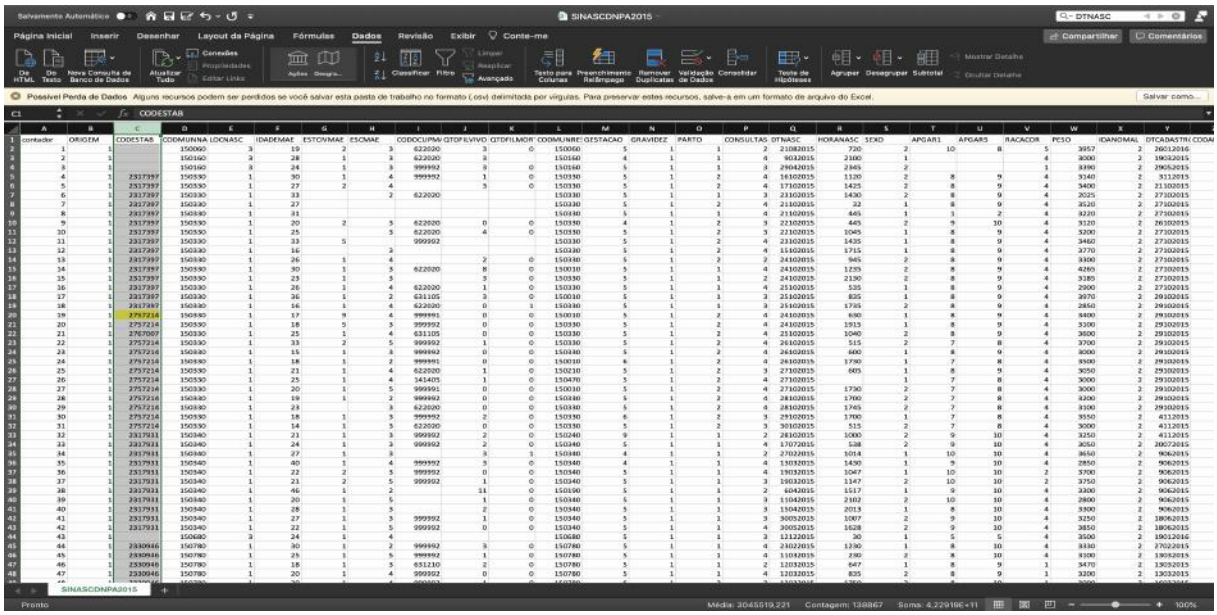
```
def adiciona_zero_nos_dias(data):
    data = str(data)
    if len(data) == 7:
        data = '0' + data
    return data

def formata_datas(coluna_da_base):
    coluna_da_base = coluna_da_base.apply(adiciona_zero_nos_dias)
    return pd.to_datetime(
        coluna_da_base.astype(str),
        format='%d %m %Y', errors='coerce')
```

3.3.3.5 Repetição de valores sem descrição

Ao realizar o experimento de conversão dos dados para o valor correspondente na tabela auxiliar, o sistema nem sempre encontrou um código com descrição para fazer a ligação dos dados da base. Um exemplo é a tabela auxiliar CNESDN18, que se relaciona à coluna CODESTAB da SINASC. O código 2757214 não existe e nem possui descrição na tabela auxiliar, gerando uma saída com valores vazios. Esse problema pode ser visualizado na figura 15, a tabela auxiliar na figura 16 e o resultado na figura 17.

Figura 15 – Coluna CODESTAB da base antes da conversão



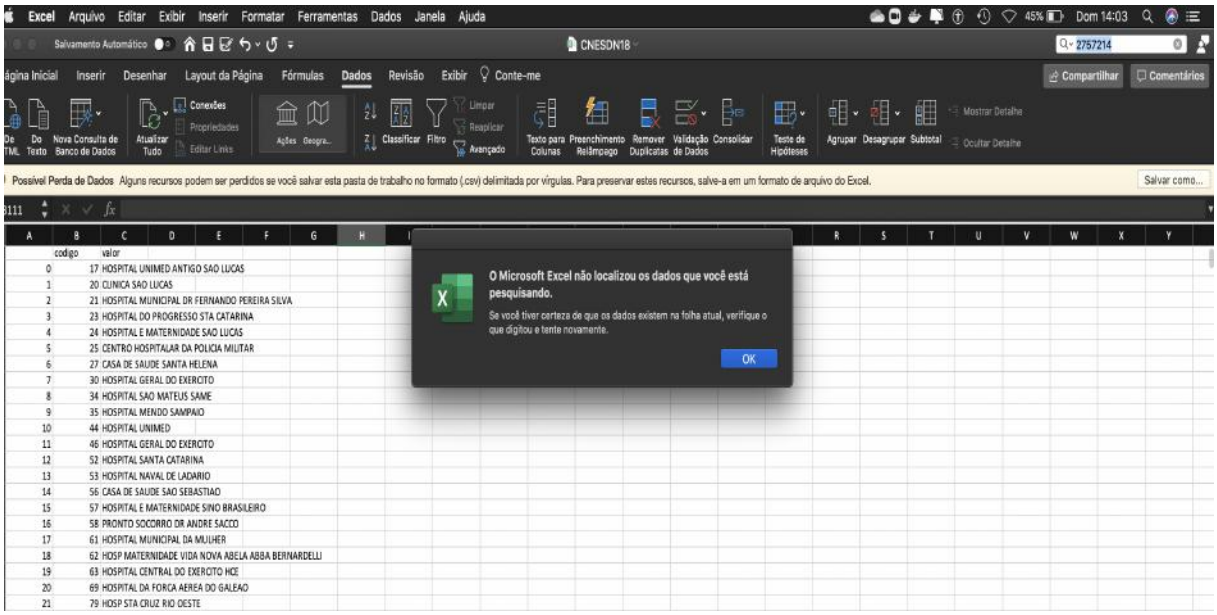
Fonte: Os autores (2020)

Foi considerada uma outra solução: ao invés de apagar os códigos sem valor correspondente nas tabelas auxiliares, o sistema os repete para que o usuário faça as considerações que achar relevante, caso tenha mais informações de outras fontes. O resultado com a repetição desses valores está na figura 18.

3.3.3.6 Remoção de linhas

Algumas tabelas auxiliares precisaram de um tratamento específico de remoção de linhas porque possuem dados que não puderam ser consideradas pelo sistema por estarem incompletos. Um exemplo é a tabela FILTIDO da SINASC que tem uma linha com uma chave vazia que tem a descrição "Br". A figura 19 mostra o problema na linha 2.

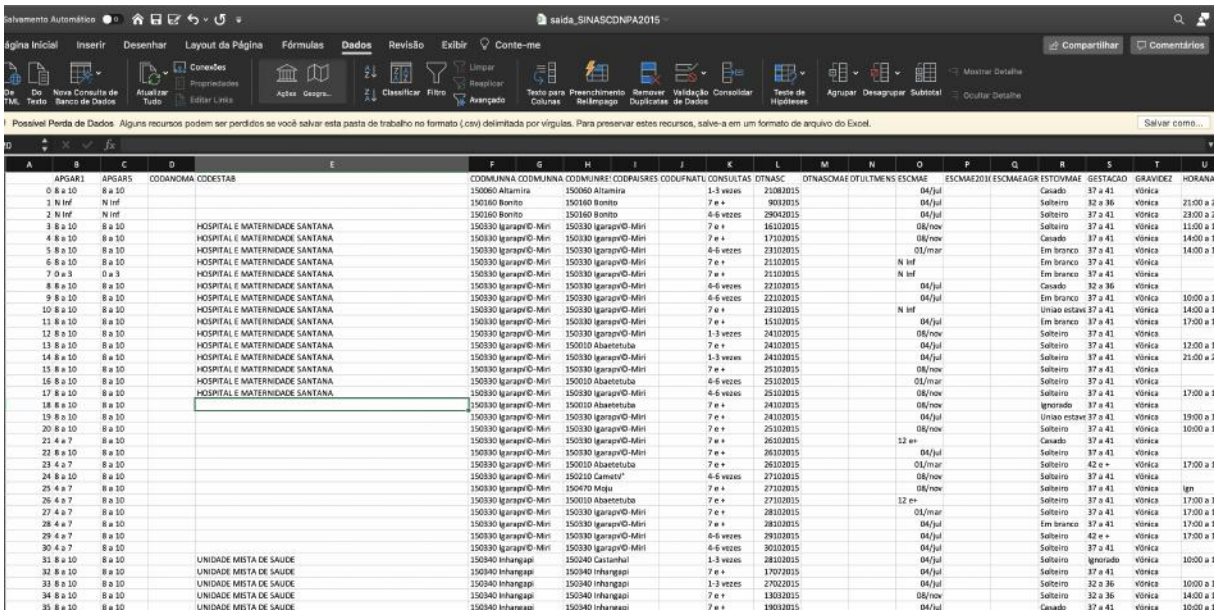
Figura 16 – Tabela auxiliar CNESDN18 sem o código 2757214



	codigo	valor
0	17	HOSPITAL UNIMED ANTIGO SAO LUCAS
1	20	CLINICA SAO LUCAS
2	21	HOSPITAL MUNICIPAL DR FERNANDO PEREIRA SILVA
3	23	HOSPITAL DO PROGRESSO STA CATARINA
4	24	HOSPITAL E MATERNIDADE SAO LUCAS
5	25	CENTRO HOSPITALAR DA POLICIA MILITAR
6	27	CASA DE SAUDE SANTA HELENA
7	30	HOSPITAL GERAL DO EXERCITO
8	34	HOSPITAL SAO MATEUS SAME
9	35	HOSPITAL MENDO SAMPAIO
10	44	HOSPITAL UNIMED
11	46	HOSPITAL GERAL DO EXERCITO
12	52	HOSPITAL SANTA CATARINA
13	53	HOSPITAL NAVAL DE LADARIO
14	56	CASA DE SAUDE SAO SEBASTIAO
15	57	HOSPITAL E MATERNIDADE SINO BRASILEIRO
16	58	PRONTO SOCORRO DR ANDRE SACCO
17	61	HOSPITAL MUNICIPAL DA MULHER
18	62	HOSP MATERNIDADE VIDA NOVA ABELA ABBA BERNARDELLI
19	63	HOSPITAL CENTRAL DO EXERCITO HCE
20	69	HOSPITAL DA FORCA AEREA DO GALEAO
21	79	HOSP STA CRUZ RIO OESTE

Fonte: Os autores (2020)

Figura 17 – Resultado da coluna CODESTAB da base após a conversão



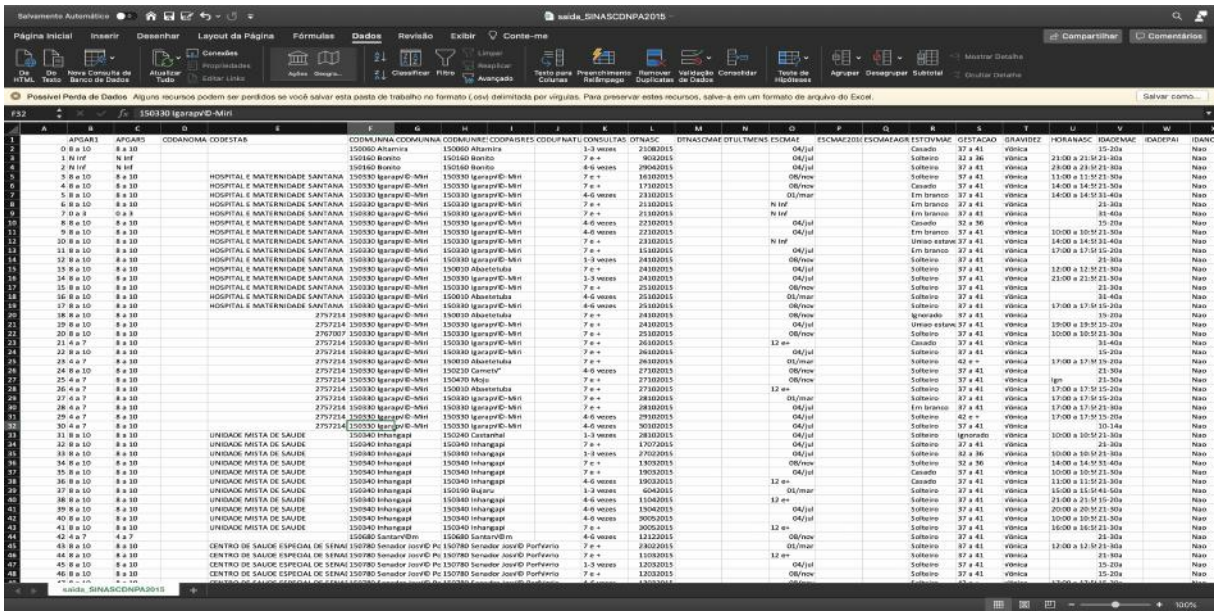
	APGAR1	APGAR5	CODANOMA	CODESTAB
0	8 a 10	8 a 10		
1	N Inf	N Inf		
2	N Inf	N Inf		
3	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
4	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
5	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
6	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
7	0 a 3	0 a 3		HOSPITAL E MATERNIDADE SANTANA
8	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
9	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
10	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
11	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
12	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
13	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
14	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
15	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
16	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
17	8 a 10	8 a 10		HOSPITAL E MATERNIDADE SANTANA
18	8 a 10	8 a 10		
19	8 a 10	8 a 10		
20	8 a 10	8 a 10		
21	4 a 7	8 a 10		
22	8 a 10	8 a 10		
23	4 a 7	8 a 10		
24	8 a 10	8 a 10		
25	4 a 7	8 a 10		
26	4 a 7	8 a 10		
27	4 a 7	8 a 10		
28	4 a 7	8 a 10		
29	4 a 7	8 a 10		
30	4 a 7	8 a 10		
31	8 a 10	8 a 10		UNIDADE MISTA DE SAUDE
32	8 a 10	8 a 10		UNIDADE MISTA DE SAUDE
33	8 a 10	8 a 10		UNIDADE MISTA DE SAUDE
34	8 a 10	8 a 10		UNIDADE MISTA DE SAUDE
35	8 a 10	8 a 10		UNIDADE MISTA DE SAUDE

Fonte: Os autores (2020)

3.3.3.7 Padronização de colunas

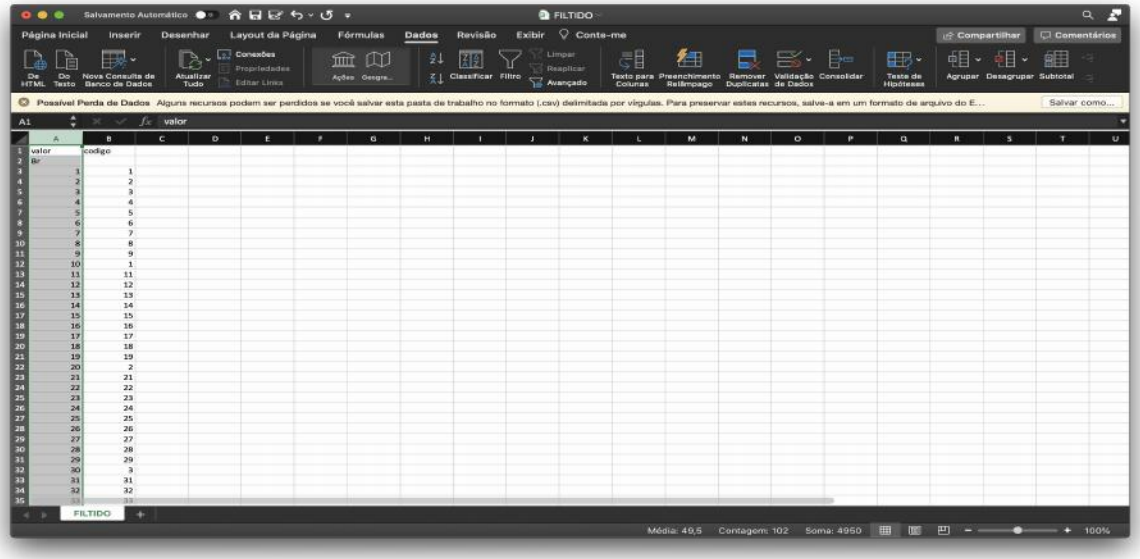
As tabelas de entrada, baixadas no DATASUS, nem sempre possuem todas as colunas listadas no arquivo json de entrada. Para padronizar, foi criada uma função que verifica as colunas de cada base adicionando as que faltam. No exemplo apresentado na figura 20 é possível ver que as colunas listadas não possuem dados. Isso simplifica o trabalho de

Figura 18 – Resultado da coluna CODESTAB da base após a conversão com repetição



Fonte: Os autores (2020)

Figura 19 – Tabela auxiliar FILTIDO



Fonte: Os autores (2020)

identificação de colunas com valores inexistentes.

3.4 DADOS DE SAÍDA

Nesta seção, são apresentadas obervações importantes sobre os dados de saída.

Figura 20 – Log de teste da tabela de 2011 da sinasc

```
coluna que não estava na base SINASC:CODBAINASC
coluna que não estava na base SINASC:CODBAIRES
coluna que não estava na base SINASC:CODUFNATU
coluna que não estava na base SINASC:ESCMAE2010
coluna que não estava na base SINASC:ESCMAEAGR1
coluna que não estava na base SINASC:NUMERODN
coluna que não estava na base SINASC:TPNASCASSI
```

Fonte: Os autores (2020)

Como dito anteriormente, acredita-se que seja mais enriquecedor para o usuário final se ele puder ver tanto o dado *raw*, quanto o dado convertido. Por conta disso, o ELK Saúde mantém o dado raw e cria colunas adicionais, enriquecidas com as conversões das tabelas auxiliares. Essas colunas são identificadas pelo "[nome da coluna]_desc"para ficar claro que esses dados foram convertidos pelo sistema.

A tabela de saída foi criada em formato .csv apenas para teste e antes de ter o carregamento no Elastic Search pronto. Algumas colunas dessa saída estão na figura 21

Analisando o resultado, sabemos que as colunas RACACOR, SEXO e TIPOBITO possuem tabelas auxiliares relacionadas, cada uma com uma coluna análoga, contendo o prefixo "_desc", que possui o significado que descreve o código da coluna original.

As colunas SEMANAGESTAC, SERIESFAL, SERIESCMAE, STDOEPIDEM, STDONOVA não possuem tabelas auxiliares associadas.Nestes casos, apesar da conversão dos dados para os valores auxiliares ser de suma importância para a compreensão do dado, o valor original foi mantido.

Figura 21 – Saída SIH

	AU	AV	AW	AX	AY	AZ	BA	BB	CL	CM	CN
1	RACACOR	SEMAGESTAC	SERIESCFAL	SERIESCMAE	SEXO	STDOEPIDEM	STDONOVA	TIPOBITO	RACACOR_desc	SEXO_desc	TIPOBITO_desc
2	1				2	0	0	2	Branca	Fem	Não Fetal
3	1				2	0	0	2	Branca	Fem	Não Fetal
4	1				1	0	0	2	Branca	Mas	Não Fetal
5	1				2	0	0	2	Branca	Fem	Não Fetal
6	1				1	0	0	2	Branca	Mas	Não Fetal
7	1				2	0	0	2	Branca	Fem	Não Fetal
8	1				1	0	0	2	Branca	Mas	Não Fetal
9	2				2	0	0	2	Preta	Fem	Não Fetal
10	1				2	0	0	2	Branca	Fem	Não Fetal
11	4				2	0	0	2	Parda	Fem	Não Fetal
12	1				2	0	0	2	Branca	Fem	Não Fetal
13	1				1	0	0	2	Branca	Mas	Não Fetal
14	1				1	0	0	2	Branca	Mas	Não Fetal
15	1				2	0	0	2	Branca	Fem	Não Fetal
16	1				1	0	0	2	Branca	Mas	Não Fetal
17	1				2	0	0	2	Branca	Fem	Não Fetal
18	1				2	0	0	2	Branca	Fem	Não Fetal
19	1				2	0	0	2	Branca	Fem	Não Fetal
20	1				1	0	0	2	Branca	Mas	Não Fetal
21	4				1	0	0	2	Parda	Mas	Não Fetal
22	1				2	0	0	2	Branca	Fem	Não Fetal
23	1				1	0	0	2	Branca	Mas	Não Fetal
24	1				2	0	0	2	Branca	Fem	Não Fetal

Fonte: Os autores (2020)

4 VISUALIZAÇÃO DOS DADOS

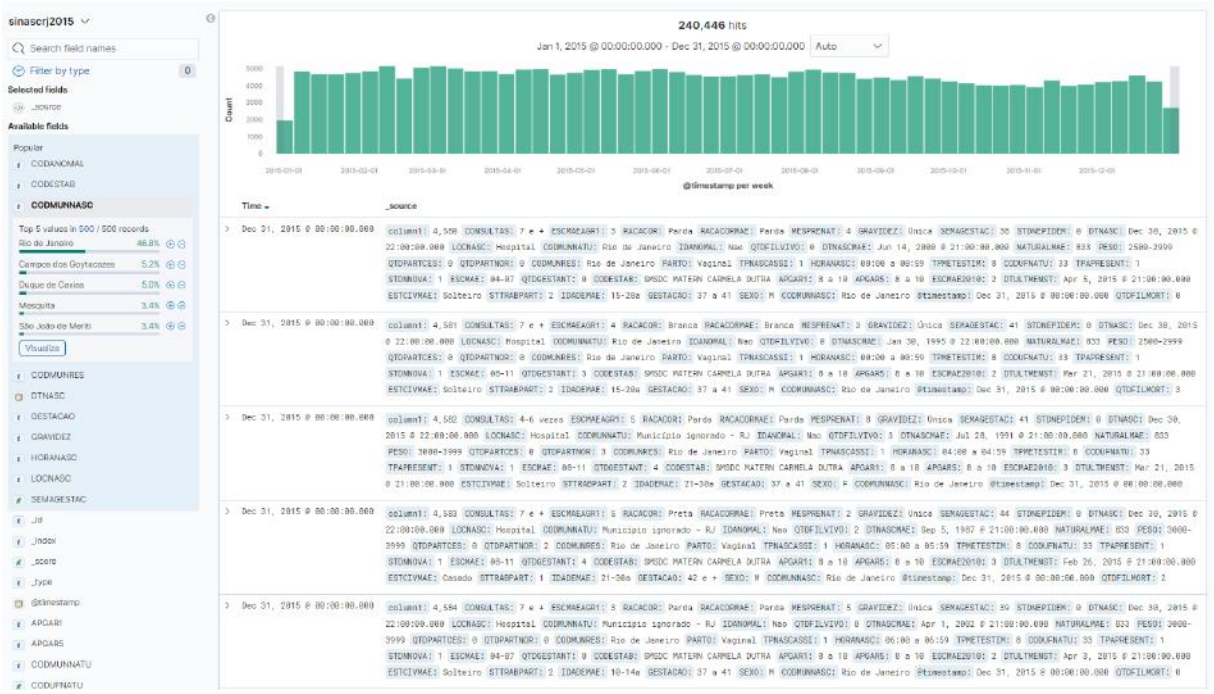
Após a execução do processo de ETL dos dados disponíveis no site do Datasus, conforme apresentado no capítulo 3, cujos campos foram identificados pelos especialistas, das tabelas auxiliares e das conversões devidas, a etapa de Data Ingestion, com o carregamento dos dados no ELK Saúde foi concluída. Os dados carregados em massa para o ELK Saúde tem tipos compreendidos entre *numbers* (tipo numérico inteiro e com ponto flutuante), *text* (cadeia de caracteres não estruturada), *date* (tipo de valor temporal) e geo_point (tipo de dado contendo latitude e longitude). O tipo de cada dado no ELK Saúde foi definido durante o processo de carga na base (conforme descrito em 3.3.2).

Para facilitar a indexação dos dados no ELK Saúde, campos do tipo *date* quando selecionados são nomeados como *@timestamp*. Dessa forma, os dados podem ser mais facilmente acessados numa perspectiva temporal, frequentemente usada nessa ferramenta para análise de dados periódicos. No ELK Saúde, para cada base do Datasus trabalhada, é delegado o campo *@timestamp* a um campo de dado do tipo *date* que tenha um significado específico para a base. Por exemplo, para a base SINASC, o campo DT_NASC, que contém a data de nascimento dos registros de nascidos. Para a base SIM, o campo DT_OBITO contém a data do óbito dos registros de mortalidade. Por fim, para a base SIH, o campo DT_INTER contém a data de início da hospitalização dos registros de internação hospitalar. Além disso, foi criado no ELK Saúde um campo índice que assume o comportamento de chave primária na base. Dessa forma, os dados estão disponíveis para que sejam feitas buscas em grandes volumes de dados.

Assim, como mencionado em 2.5.2, para visualização dos dados carregados, o Kibana se mostrou uma ferramenta de fácil utilização e de rápida exibição de resultados, devido a sua integração com o ELK Saúde. Com os dados devidamente carregados, o Kibana fornece para o usuário, na seção *Discover*, uma prévia dos dados da base indexada, como é possível ver na figura 22. Nessa página é exibido um histograma baseado no campo considerado *@timestamp* da base. Em seguida, são discriminadas todas as colunas com seus respectivos tipos, com a possibilidade de análise dos valores assumidos nos primeiros registros na base, na lista de campos a esquerda na figura 22. Ainda é possível encontrar uma distribuição temporal dos registros dispostos em função do campo *@timestamp*. Por fim, exibe uma lista com os primeiros registros armazenados nessa base. Toda essa prévia, pode ser um forma do usuário interagir com o ambiente, dando a possibilidade de aplicar diversos filtros necessários para efetuar as análises.

Para esse trabalho foram desenvolvidos, através do Kibana, alguns *dashboards*, que fornecem uma visão rápida dos dados e permitem uma análise detalhada dos mesmos. Esses painéis estão disponíveis do diretório *visualization* presente no repositório desse

Figura 22 – Visualização dos dados SINASC no ELK Saúde (Discover - Kibana)



Fonte: Os autores (2020)

projeto[1]. (ELASTICSEARCH, 2020a). Os *dashboards* são constituídos de componentes menores nomeados de *visualization*. Uma *visualization* é um componente do Kibana que permite uma observação visual sobre os dados. No caso do ELK Saúde, a observação visual é baseada em consultas que retornam uma análise sobre os dados indexados. Com os *dashboards* construídos, foi possível aplicar filtros sobre os dados indexados pelo ELK Saúde, além de personalizar os intervalos de tempo desses dados.

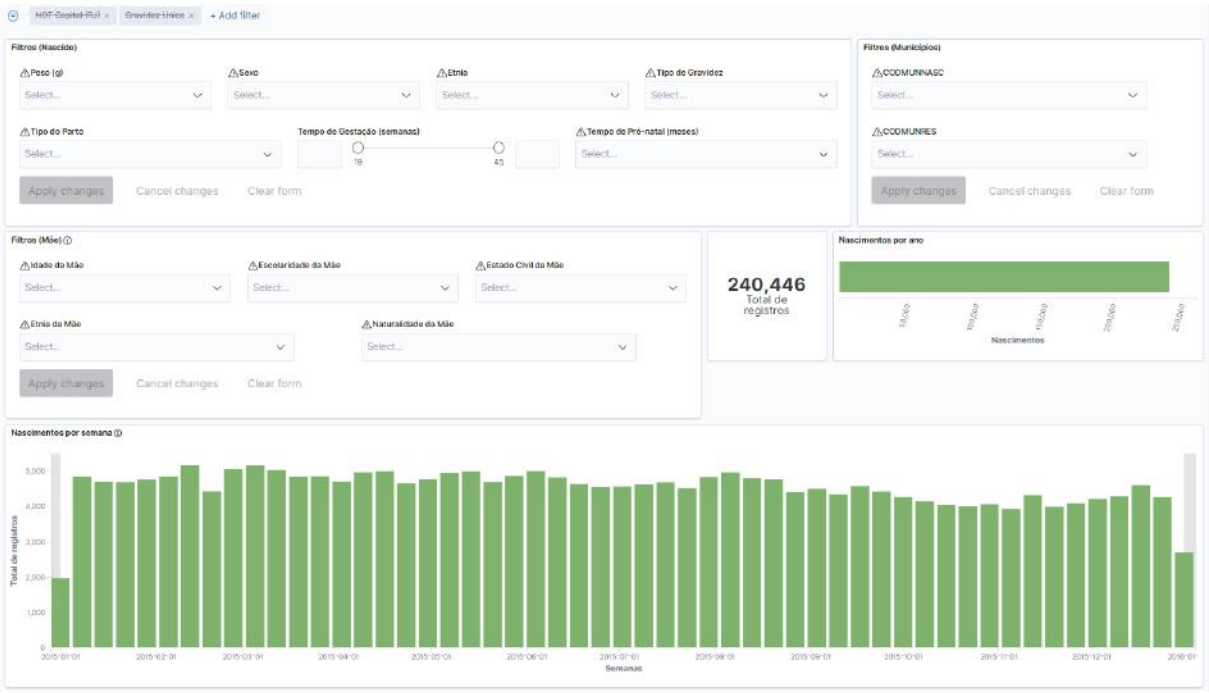
É importante salientar que o Kibana permite fácil edição dos seus componentes presentes nos *dashboards*. Essa condição oferece ao usuário final a possibilidade de adequar as configurações de cada componente de acordo com as necessidades do uso sem qualquer conhecimento com relação ao processo de ETL realizado anteriormente não obstante um entendimento razoável dos dados presentes nas bases.

Para todos os *dashboards*, além da opção padrão do Kibana de inserir critérios para a consulta feita, foram incluídos componentes experimentais do tipo "Controls", que permitem rápida aplicação de filtros ao dados apresentados. Os filtros foram pensados para facilitar a experiência do usuário com relação a buscas mais específicas, aplicando critérios a campos significativos da base. Com a finalidade de favorecer a experiência do usuário, em cada *dashboard* é apresentado também um resumo dos dados resultantes da busca, que inclui: total de registros retornados pela consulta, total de registros distribuídos pelo ano do *@timestamp* e total de registros distribuídos em um espaço de tempo menor, que pode

[1] https://bitbucket.org/pguenes/elksaude/src/master/

ser meses ou semanas. Assim, é possível ter uma noção do quantitativo e comportamento dos dados de cada base, como visto na figura 23.

Figura 23 – Componentes de Visualização Padrão e Filtros da SINASC



Fonte: Os autores (2020)

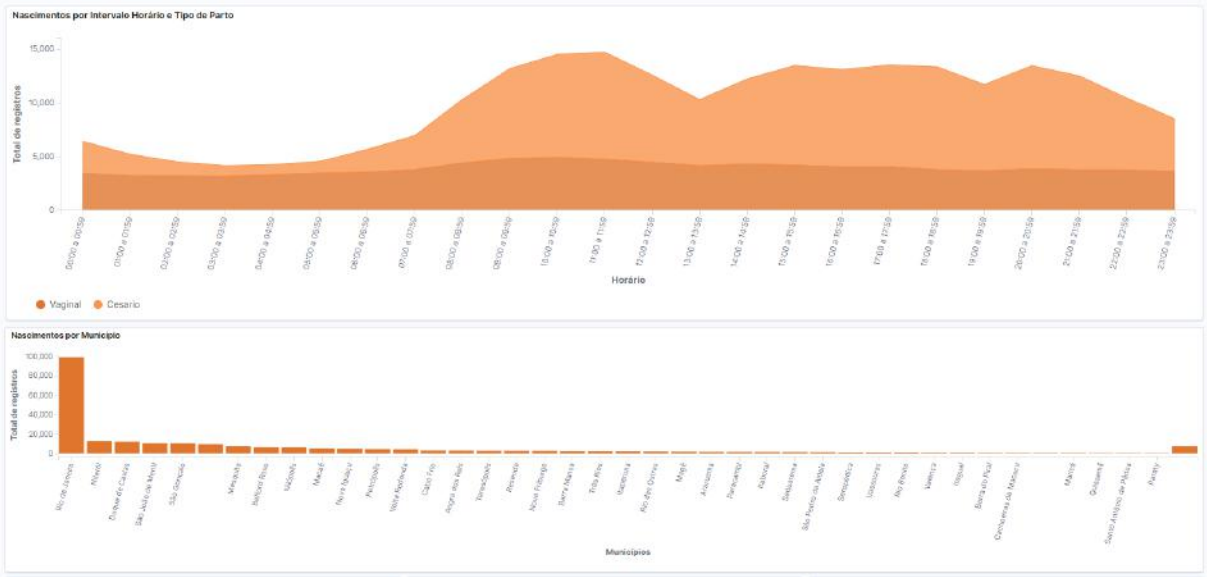
Todas as visualizações apresentadas nos *dashboards* desenvolvidos nesse trabalho, bem como os próprios *dashboards* do Kibana, podem ser editados e adequados às necessidades do usuário pela própria interface do Kibana. Esse é mais um dos benefícios da ferramenta, onde cada componente em tela pode ser configurado de acordo com os requisitos de cada consulta.

4.1 SINASC

Para a base SINASC foram implementados diversos conjuntos de filtros, separados por domínio da informação usada como critério, sendo esses domínios divididos entre dados do nascido (como sexo e etnia) e informações da gestação e nascimento, dados da mãe do nascido (como idade e escolaridade), e dados do local do nascimento e de residência. Essa organização visa permitir ao usuário a definição dos filtros que deseja aplicar sobre os dados em tela. As primeiras visualizações feitas para o *dashboard* incluem uma exibição dos nascimentos separados temporalmente e também por localidade, que podem ser observados na figura 24. Além do gráfico de barras, que ordena os registros de nascimento para cada semana do ano da ocorrência, está presente em todos os *dashboards* (figura 23) o total de registros por faixas de horário, que é apresentado em um gráfico de área, com os horários exibidos em intervalos de uma hora. Este gráfico tem sua área

secionada exibindo o quantitativo de nascimentos de cada horário dependendo do tipo de parto. As quantidades de registros discriminados por município são apresentados através de gráfico de barras.

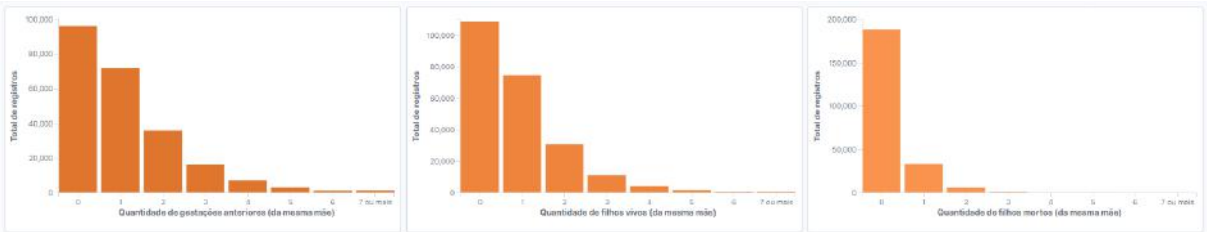
Figura 24 – Visualizações de registros da SINASC por tempo e localidade



Fonte: Os autores (2020)

As próximas visualizações, feitas para esse *dashboard*, incluem dados sobre as gestações anteriores da mãe do nascido, mostrando os resultados em gráficos de barra, que podem ser vistos na figura 25. Os gráficos apresentam as quantidades de registros na base, ordenados respectivamente pela quantidade de gestações anteriores, quantidade de filhos vivos e quantidade de filhos não vivos da mãe do nascido. Tais dados estão armazenados no ELK Saúde como valores numéricos e agrupam, no último termo do gráfico, os valores numéricos superiores a 7 ou que tiveram seu preenchimento ignorado.

Figura 25 – Visualizações sobre gestações anteriores

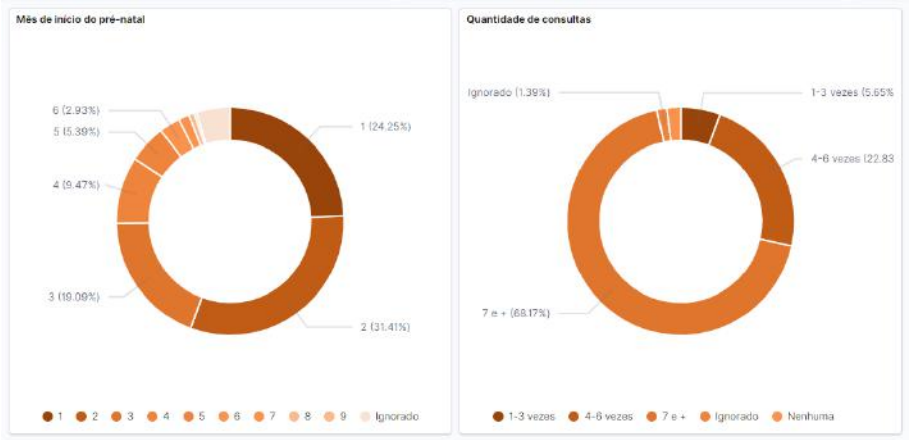


Fonte: Os autores (2020)

Esse agrupamento de valores superiores a deteminado critério pode ser também aplicado nos gráficos de setores, como pode ser visto nas visualizações da figura 26. Os gráficos apresentam o total de nascimentos, ordenados pela quantidade de meses antes do nascimento em que o pré-natal foi iniciado e também pela quantidade de consultas ocorridas

durante a gestação. É possível observar que os valores numéricos apresentam imprecisão no preenchimento dos registros, como por exemplo, a quantidade de meses de pré-natal superior a 9, e, nesse caso, os valores foram definidos como "Ignorado"e agrupados.

Figura 26 – Visualizações sobre tempo e consultas no pré-natal



Fonte: Os autores (2020)

Os campos usados como critério nessas duas visualizações anteriores puderam ser relacionados na visualização seguinte. A opção de secionar um gráfico de área (como apresentado na figura 24), foi também aplicada nessa visualização, que apresenta os totais de registros de nascimento, ordenados pelo mês de gestação que a mãe do nascido deu início ao pré-natal. Esse quantitativo foi subdividido entre o critério do número de consultas que a mãe teve durante o pré-natal, como pode ser visto na figura 27

Por fim, foram incluídas duas visualizações que pudessem mostrar mais funcionalidades do Kibana. Na figura 28, o primeiro gráfico de setor é estratificado em dois níveis: no nível mais interno o total de registros é dividido entre os possíveis valores de APGAR avaliado no primeiro minuto (APGAR1), e no nível mais externo cada seção de valores de APGAR1 é subdividido entre os possíveis valores de APGAR avaliado no quinto minuto (APGAR5). Essa possibilidade de subdividir os valores entre dois critérios permite que se exiba mais de um estatística de uma única vez. O segundo gráfico apresenta, assim como nos demais gráficos de setor, a quantidade de registros secionada entre valores, no caso do tipo do parto. No entanto, para essa visualização foi aplicado um filtro onde foram selecionados apenas os registros de nascimentos prematuros. Por isso, ao lado do gráfico, se exibe o valor numérico que corresponde ao total de registros encontrados nessa consulta.

Figura 27 – Visualizações relacionando tempo e consultas no pré-natal

Fonte: Os autores (2020)

4.2 SIM

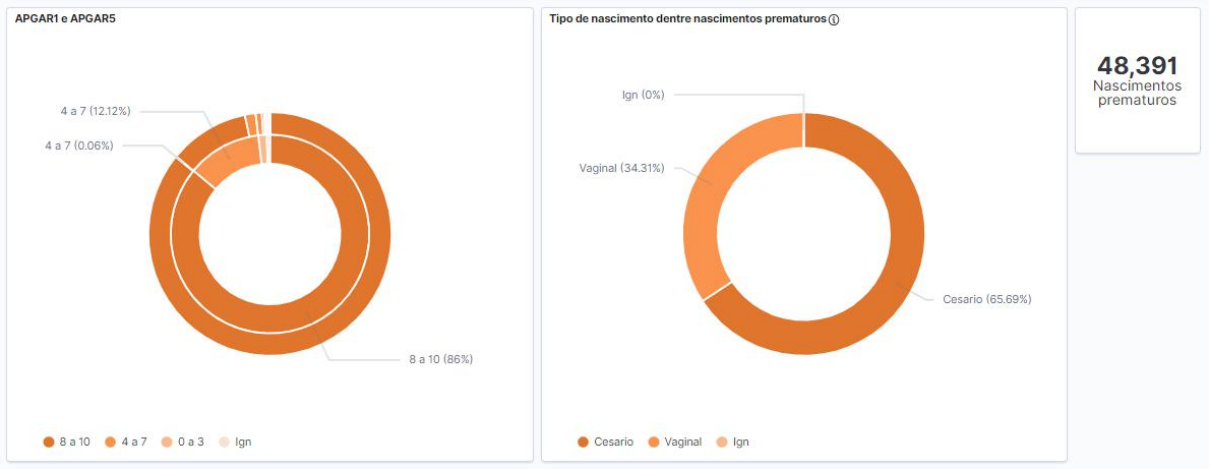
O conjunto de dados presentes na base SIM contém informações do falecido (como identificação e residência), assim como da ocorrência do óbito (causas e condições) e informações mais específicas que envolvem óbito na gestação ou de recém nascidos. Dado esse contexto, conjuntos de filtros para domínios específicos foram implementados no *dashboard* para a base da SIM, assim como feito para a SINASC.

Como pode ser observado na figura 29, os filtros foram organizados em 3 conjuntos: o primeiro conjunto, com dados do falecido, contém campos como idade, sexo e informações de localidade do falecido. O segundo, com filtros condizentes com a ocorrência do óbito, possui campos sobre as circunstâncias do óbito, bem como a localidade da ocorrência. Por fim, o terceiro conjunto de filtros, com dados específicos para casos de óbito na gestação e próximos ao nascimento, com dados da mãe, da gestação e do parto.

Na figura 29, podem ser observadas algumas visualizações padrões a todos os *dashboards* que informam a quantidade total de registros na base, bem como a distribuição dos dados em um período de tempo. Além disso, pode-se observar na 30 que, analogamente ao trabalho feito para o *dashboard* da SINASC, foi desenvolvida uma visualização para apresentar os registros da base distribuídos pelo município de ocorrência do óbito.

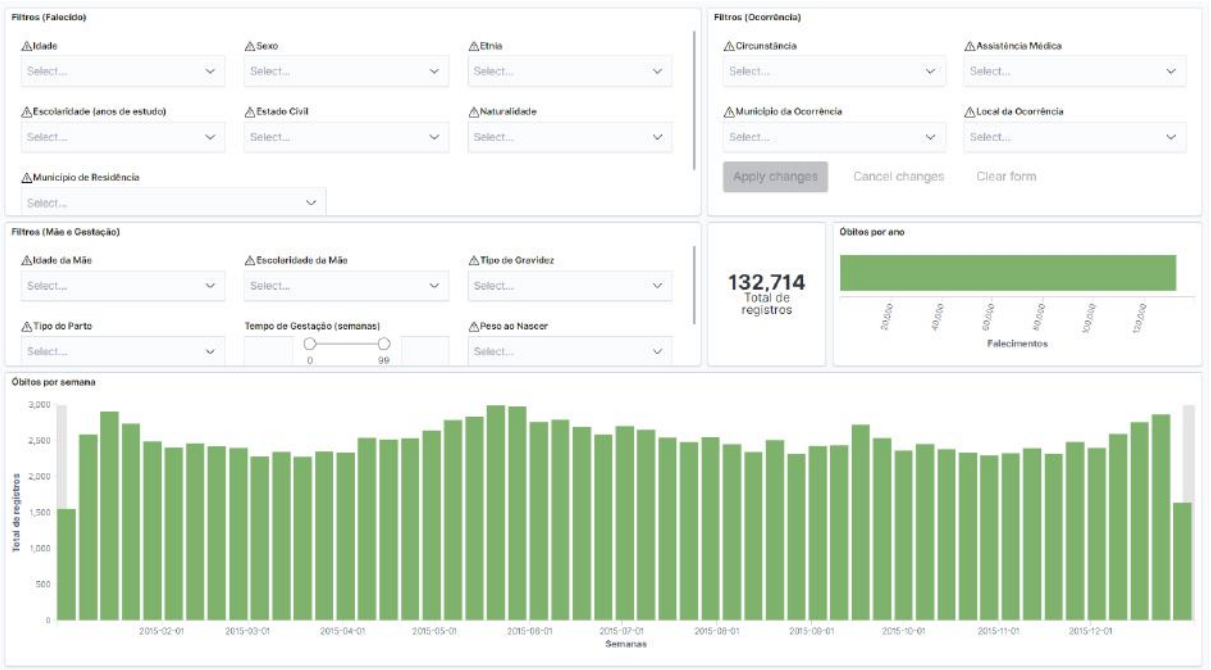
Ainda no *dashboard* da SIM, foram feitas algumas visualizações que usaram como

Figura 28 – Visualizações sobre teste APGAR e tipo de parto



Fonte: Os autores (2020)

Figura 29 – Filtros da SIM e Componentes de Visualização Padrão

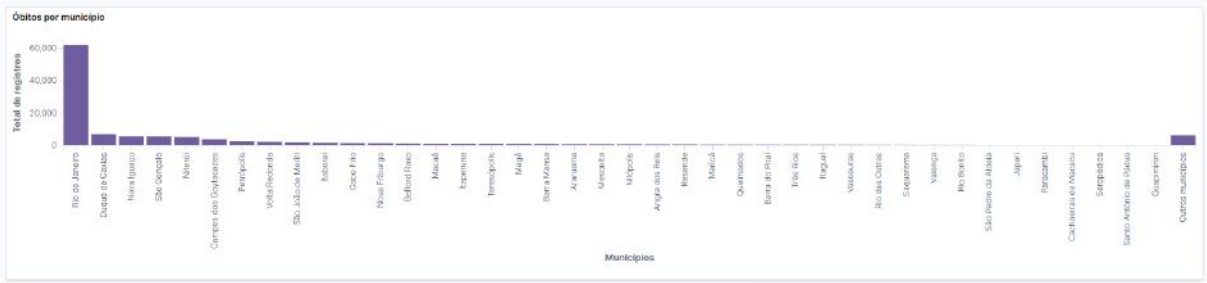


Fonte: Os autores (2020)

critérios principais as informações de ocorrência do óbito. Na figura 31, foram gerados dois gráficos de setores. O primeiro organiza os dados de acordo com a circunstância do óbito (homicídio, acidente, etc) e o seguinte de acordo com a localização da ocorrência do óbito (hospital, domicílio, etc).

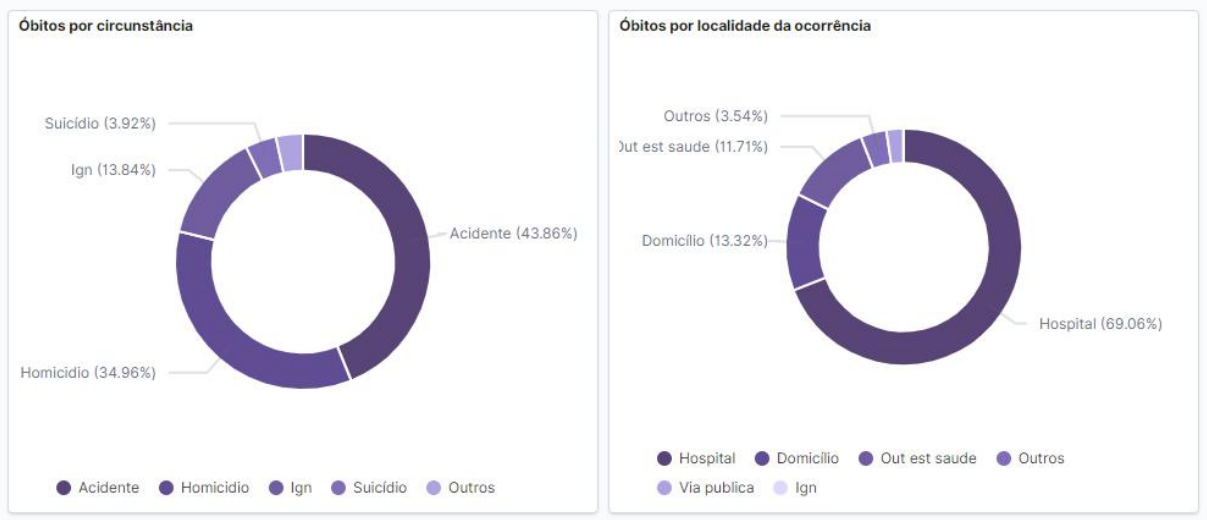
Na figura 32 uma visualização mais refinada é apresentada. Nela, um gráfico de barras verticais distribui o total de registros que tenham "Suicídio"como circunstância entre a faixa etária dos falecidos, além de dividir os indivíduos entre sexo masculino ("Mas") e

Figura 30 – Visualização de registros da SIM por localidade



Fonte: Os autores (2020)

Figura 31 – Visualizações sobre circunstância e localidade dos óbitos



Fonte: Os autores (2020)

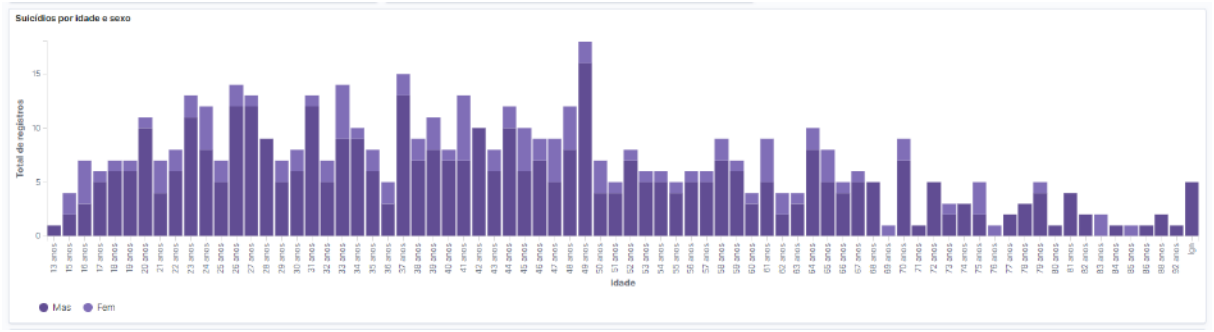
feminino ("Fem"). Essa possibilidade de secionar as barras de um gráfico, permite uma análise dos dados com diversos critérios aplicados simultaneamente.

Uma última visualização foi implementada na figura 33 onde é possível observar um gráfico de barras verticais contendo a distribuição do total de registros que caracterizam óbitos de recém nascidos, através da quantidade de semanas de gestação do recém nascido até o parto. Além disso, os dados foram secionados sob o critério de tipo de parto, sendo possível fazer uma análise entre esses dois diferentes critérios.

4.3 SIH

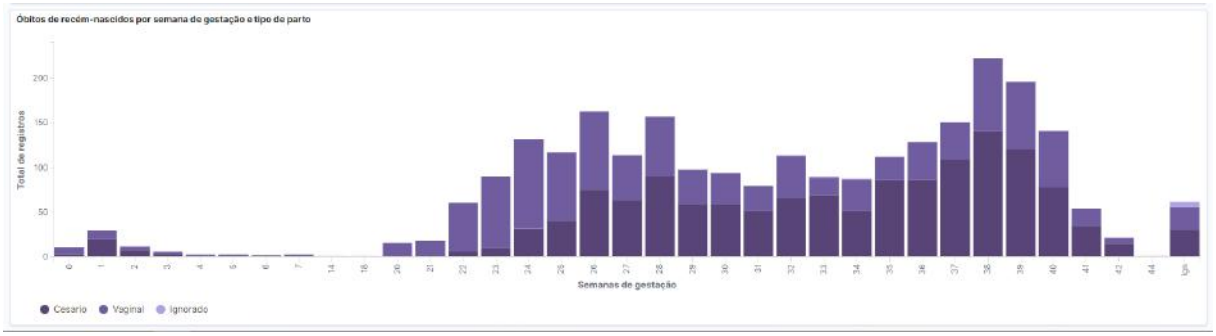
A base SIH contém os dados de internações hospitalares do SUS e possui uma diversidade maior de dados que as duas bases anteriores. Os campos selecionados para esse trabalho, no entanto, apresentam uma completude baixa para mostrar o potencial de desempenho do Kibana, havendo diversas lacunas de preenchimento de registros das in-

Figura 32 – Visualização sobre suicídio relacionado a idade e sexo



Fonte: Os autores (2020)

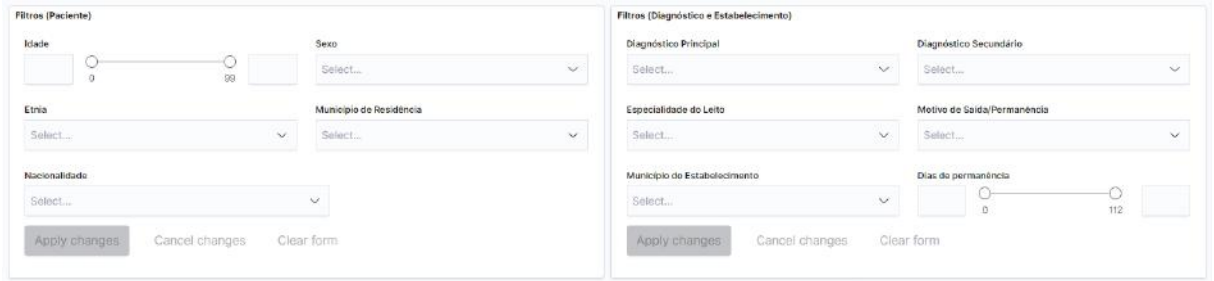
Figura 33 – Visualização sobre óbito relacionado a gestação e parto



Fonte: Os autores (2020)

ternações. Devido a essa baixa completude, alguns campos específicos foram selecionados para integrar as visualizações do *dashboard* desenvolvido para essa base, assim como para os filtros existentes. Dentre os campos selecionados para filtros, alguns foram agrupados e correspondem aos dados do paciente (idade, sexo, etc) e outros correspondem aos dados da internação e do estabelecimento (diagnóstico, permanência, localidade, etc), como observado na figura 34.

Figura 34 – Filtros da SIH

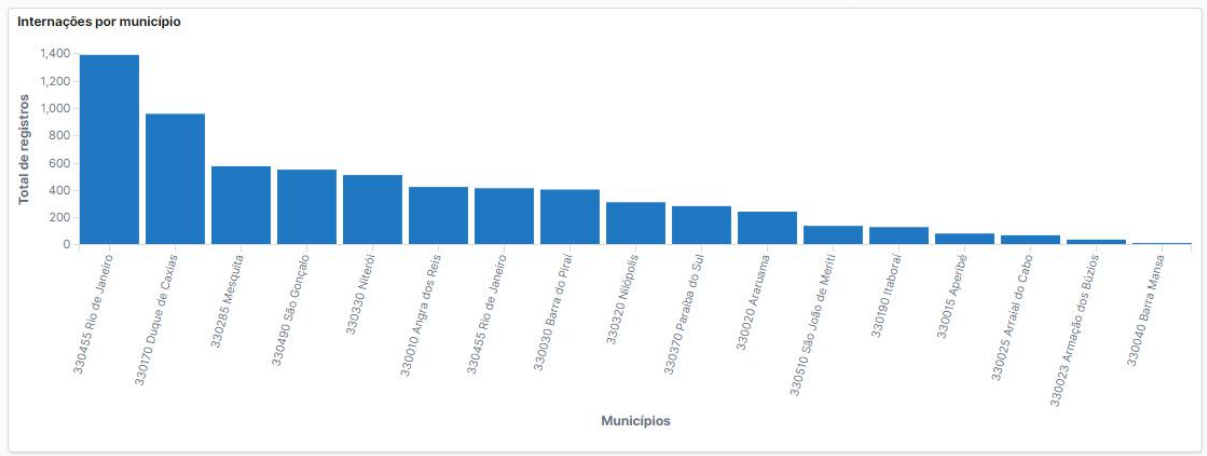


Fonte: Os autores (2020)

Assim como nos *dashboards* anteriores, a próxima visualização proposta distribui o

total de registros geograficamente. Na figura 35 é apresentado o gráfico de barras verticais com a quantidade de registros de internações pelo município do estabelecimento onde foi realizada a internação.

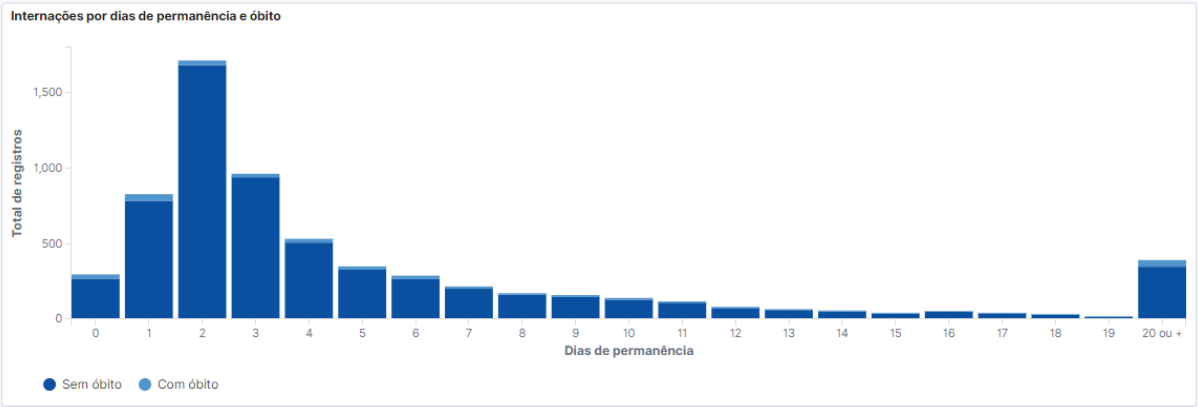
Figura 35 – Visualização de registros da SIH por localidade



Fonte: Os autores (2020)

Um outro gráfico de barras (figura 36) foi elaborado para apresentar o comportamento do quantitativo de internações, dependendo da quantidade de dias de permanência. Além disso, o gráfico foi secionado entre as internações concluídas, com e sem óbito, permitindo análise paralela dos dois dados apresentados. Por fim, uma visualização com gráfico de setores foi implementada para apresentar as internações hospitalares de acordo com o tipo de leito. O gráfico, exibido na figura 37, é apresentado secionando o quantitativo pela especialidade do leito da internação, com um agrupamento dentre os valores menores no último termo, como apresentado anteriomente na figura 25.

Figura 36 – Visualização sobre a duração da internação



Fonte: Os autores (2020)

Figura 37 – Visualização sobre a especialidade da internação

Fonte: Os autores (2020)

5 CONCLUSÃO

Este trabalho confirma que a computação é aplicável dentro do campo de saúde coletiva e podemos auxiliá-los disponibilizando um ambiente de ETL das três bases do DATASUS: SIH, SIM e SINASC. Para além disso, o ELK Saúde possui uma estrutura que facilita a extração e carregamento dos dados de outras bases. O ELK Saúde dispensa a necessidade de mudar várias vezes de ambientes para efetivamente visualizar os dados e entender as informações que se encontram nele, ou seja, ao utilizá-lo, não é precisar baixar os dados e mudar de sistema, por exemplo: utilizar o navegador para baixar os dados, exportar com o tabnet em csv, abrir no excel ou em outra ferramenta para visualizar os dados.

O ELK Saúde é composto pelo pacote ELK e pela ferramenta da ETL, disponíveis no repositório do projeto [1] e imagem do docker disponível em no dockerhub [2]. Neste trabalho, exemplificamos o uso da ferramenta visual e de pesquisa, o Kibana, e da ferramenta que extrai os dados por ftp do site, faz as transformações necessárias para que esses dados sejam compreendidos e carrega os dados no Kibana para que as visualizações e análises sejam feitas. O capítulo 4 mostra que é viável fazer uma análise dos dados que possibilita e contribui para tomadas de decisão importantes para a comunidade da área da saúde.

Houve interesse da junção das três bases utilizadas neste trabalho, de modo que se pudesse cruzar os dados e traçar relações que enriqueceriam ainda mais as análises possíveis desses dados. No entanto, essa tarefa se apresentou inviável devido a ausência de uma propriedade em comum entre os registros das diferentes bases que oportunizaria tal iniciativa. Essa ausência se dá em virtude da anonimidade dos registros utilizados. Todavia essa impossibilidade não desagrega a qualidade das análises que são possíveis com o trabalho feito.

Durante o processo de programação, a ferramenta de extração do ELK Saúde precisou ser otimizada para que fizesse o processamento e transformação dos dados mais rápido, trazendo mais eficiência. Inúmeros ajustes nos dados foram necessários para que eles fossem compreensíveis para o usuário final na fase de visualização, como apresentado no capítulo 3. Algumas melhorias no sistema foram pensadas e estão listadas na próxima seção, em trabalhos futuros.

5.1 TRABALHOS FUTUROS

Este projeto possui potencial para trabalhos futuros na área de dados de saúde, permitindo a utilização de técnicas de *machine learning* sobre diversos dados disponíveis no site do Datasus. No quesito dados, uma aplicação futura seria a construção de um *Data*

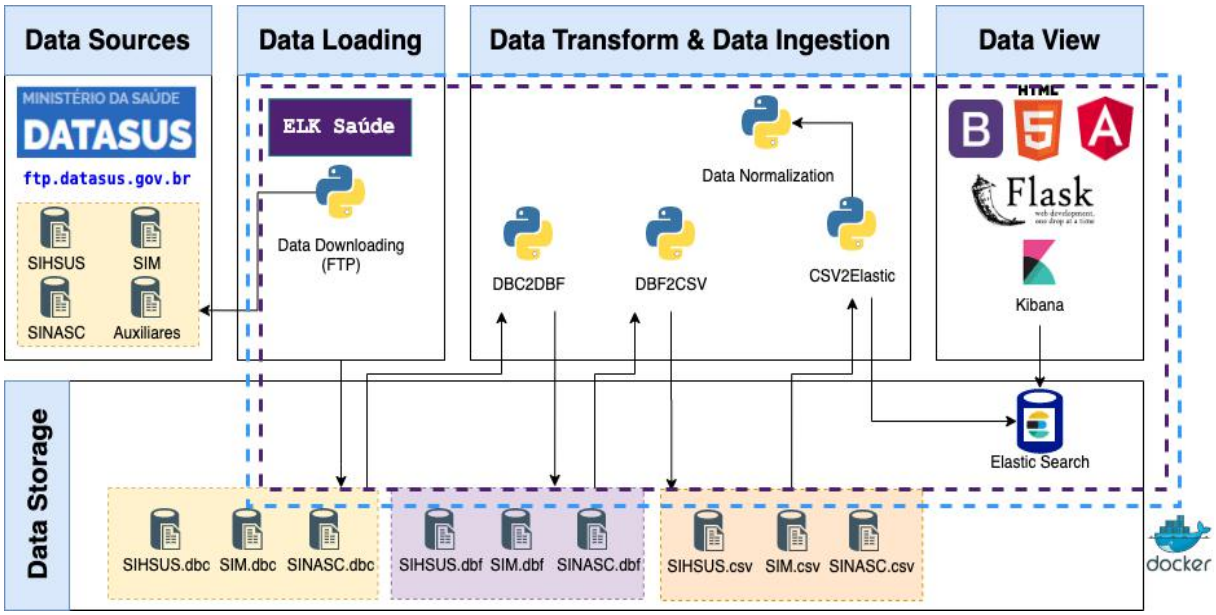
[1] https://bitbucket.org/pguenes/elksaude/src/master/
[2] https://hub.docker.com/r/sebp/elk/

lake[3] para processamento de *Big Data*. Para isso, será necessário expandir o download para outros estados e outros anos das três bases tratadas neste projeto, e preparar o ambiente para fazer download de outras bases disponíveis no site do datasus, tais como bases epidemiológicas, indicadores de saúde, dentre outras.

Considerando *Machine* e *Deep Learning*, será possível utilizar a robustez do ELK Saúde para análise de dados, assim como prover recursos de predição com base nos dados, após uma análise das bases.

Além disso, como o ELK Saúde precisa de um arquivo json de entrada, será interessante, para o usuário final, criar uma interface amigável, onde o usuário possa construir suas próprias consultas, identificando as variáveis desejadas. Uma outra interessante possibilidade de trabalho que se mostra necessária é a composição de metadados para as bases exploradas, para tornar mais completo o ambiente e mais satisfatório a experiência do usuário final. Outra ideia é que os dados baixados pela aplicação sejam disponibilizados por uma API (Application Programming Interface) para geração de microserviços, utilizando o framework Flask[4]. Uma imagem do Docker com todos os elementos, pacote ELK e ELK Saúde poderá ser desenvolvida, e essas alterações na arquitetura da solução estão apresentadas na figura 38.

Figura 38 – Arquitetura futura



Fonte: Os autores (2020)

Outra alteração interessante para o ELK Saúde será utilizar a interface proposta para selecionar um período de tempo para o qual as bases serão atualizadas automaticamente.

[3] *Data lake* é um tipo de repositório, como um lago, que armazena um conjunto de dados em seu estado natural, como um corpo d'água que não foi filtrado ou contido.

[4] *Flask* é um *framework web* escrito em *Python*, que é flexível e fornece um modelo simples para diversos desenvolvimentos *web*.

Neste caso, será necessário desenvolver um *script* em *Python* para efetuar esta atualização.

Um estudo que fosse feito sobre o comportamento dos dados de cada base poderia viabilizar um entendimento maior sobre a duplicidade de informação que pode vir a acontecer nas bases do DATASUS devido a diversos fatores que envolvem bases dessa tamanho e que tem dados inseridos manualmente. Esse tipo de tratamento traria maior precisão para as análises feitas e poderia até promover um estudo crítico próprio sobre essa duplicidade e suas causas.

Para ter visualizações em formato de mapas, o Kibana pode utilizar a latitude e a longitude correspondentes às localizações geográficas das instituções de saúde. Para projetar mapas a partir dessas localizações, uma opção será utilizar o *OpenStreetMap*[5]. apresentando as informações solicitadas pelo usuário.

[5] É uma ferramenta que provê dados de mapa para milhares de sites, aplicativos móveis e dispositivos de hardware.

REFERÊNCIAS

ANDREU-PEREZ, J. Big data for health. 2015. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7154395>. Acesso em: 15 setembro 2020.

CARVALHO, R.; ALMEIDA, I. Ambiente de exploração de dados da saúde usando um banco de dados nosql. 2017.

DATASUS. Sistemas datasus. 2020. Disponível em: <http://datasus1.saude.gov.br/sistemas-e-aplicativos>. Acesso em: 18 agosto.2020.

DATASUS, C. Catálogo de produtos datasus. 2020. Disponível em: <https://datasus.saude.gov.br/wp-content/uploads/2019/08/Catalogo-de-Produtos-DATASUS.pdf>. Acesso em: 15 setembro 2020.

DOCKER. What is a container. 2020. Disponível em: <https://www.docker.com/resources/what-container>. Acesso em: 12 setembro 2020.

ELASTICSEARCH. Dashboard. 2020. Disponível em: <https://www.elastic.co/guide/en/kibana/current/dashboard.html>. Acesso em: 08 de novembro de 2020.

ELASTICSEARCH. Helpers. 2020. Disponível em: <https://elasticsearch-py.readthedocs.io/en/7.9.1/helpers.html>. Acesso em: 06 de novembro de 2020.

ELASTICSEARCH. Index vs type. 2020. Disponível em: <https://www.elastic.co/pt/blog/index-vs-type>. Acesso em: 23 de novembro de 2020.

ELASTICSEARCH. What is an elasticsearch index? 2020. Disponível em: <https://www.elastic.co/pt/blog/what-is-an-elasticsearch-index>. Acesso em: 08 de novembro de 2020.

HOSTINGER. Ftp: o que é, como funciona e qual o melhor tipo para gerenciar arquivos na internet. 2020. Disponível em: <https://www.hostinger.com.br/tutoriais/ftp-o-que-e-como-funciona>. Acesso em: 18 agosto.2020.

IETF. File transfer protocol (ftp). 2020. Disponível em: <https://www.ietf.org/rfc/rfc0959.txt>. Acesso em: 20 de agosto de 2020.

Ministério da Saúde. Sus. 2020. Disponível em: <http://www.saude.gov.br/sistema-unico-de-saude>. Acesso em: 15 setembro 2020.

MURDOCH, T. B.; DETSKY, A. S. The inevitable application of big data to health care. 2013. Disponível em: <https://jamanetwork.com/journals/jama/article-abstract/1674245>. Acesso em: 15 setembro 2020.

OPENSOURCE. What is docker. 2020. Disponível em: <https://opensource.com/resources/what-docker>. Acesso em: 10 setembro.2020.

REDMONK. The redmonk programming language rankings: June 2020. 2020. Disponível em: <https://redmonk.com/sogrady/2020/07/27/language-rankings-6-20/>. Acesso em: 20 de agosto de 2020.

SAMPAIO, R. S. Ambiente de dados do sihsus com mongodb. 2019. Disponível em: <https://pantheon.ufrj.br/handle/11422/11197>. Acesso em: 16 setembro 2020.

TABLEAU. Data visualization beginner's guide: a definition, examples, and learning resources. 2020. Disponível em: <https://www.tableau.com/learn/articles/data-visualization>. Acesso em: 22 agosto.2020.

ANEXOS

ANEXO A – ARQUIVO JSON DE ENTRADA PARA A BASE SINASC

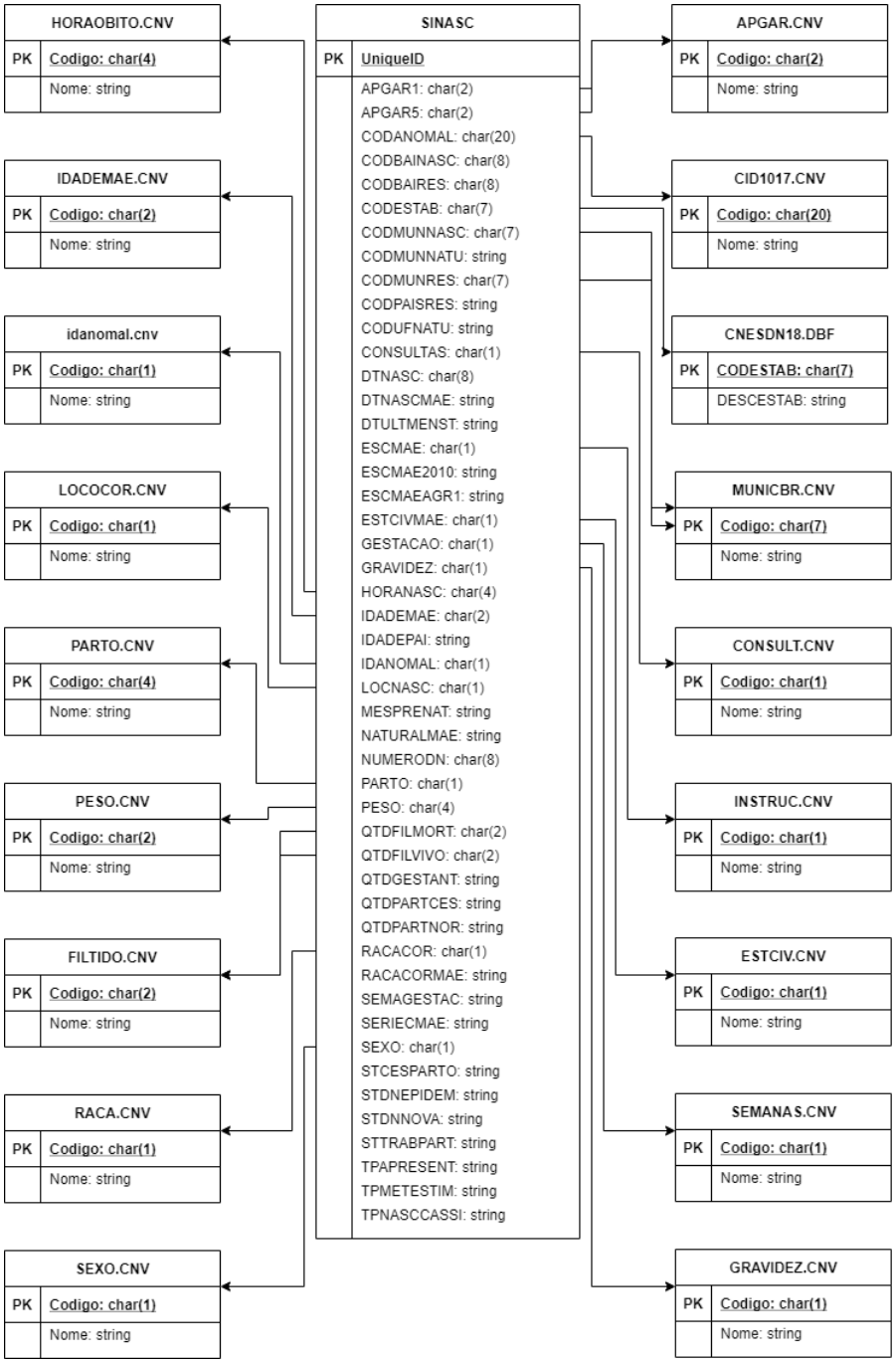
Código A.1 – Exemplo de json de entrada para a base SINASC

```
{
    "base":"SINASC",
    "ano":2015,
    "database":"DATASUS",
    "collection":"SINASC15",

    "ES_CLUSTER": "['http://localhost:32773']",
    "ES_INDEX": "datasus",
    "ES_TYPE": "SINASC",
    "campos":[
        {
            "campo":"CODESTAB",
            "campo_auxiliar_juncao":"CODESTAB",
            "tabelas_auxiliares":[
                "CNESDN18.DBF"
            ]
        },
        {
            "campo":"CODANOMAL",
            "campo_auxiliar_juncao":"codigo",
            "tabelas_auxiliares":[
                "CID1017.cnv"
            ]
        },
        {
            "campo":"APGAR1",
            "campo_auxiliar_juncao":"codigo",
            "tabelas_auxiliares":[
                "APGAR.CNV"
            ]
        },
        {
            "campo":"APGAR5",
            "campo_auxiliar_juncao":"codigo",
            "tabelas_auxiliares":[
                "APGAR.CNV"
            ]
        },
        {
            "campo":"CODBAINASC"
        },
        {
            "campo":"CODBAIRES"
        },
        {
            "campo":"CODMUNNASC",
```

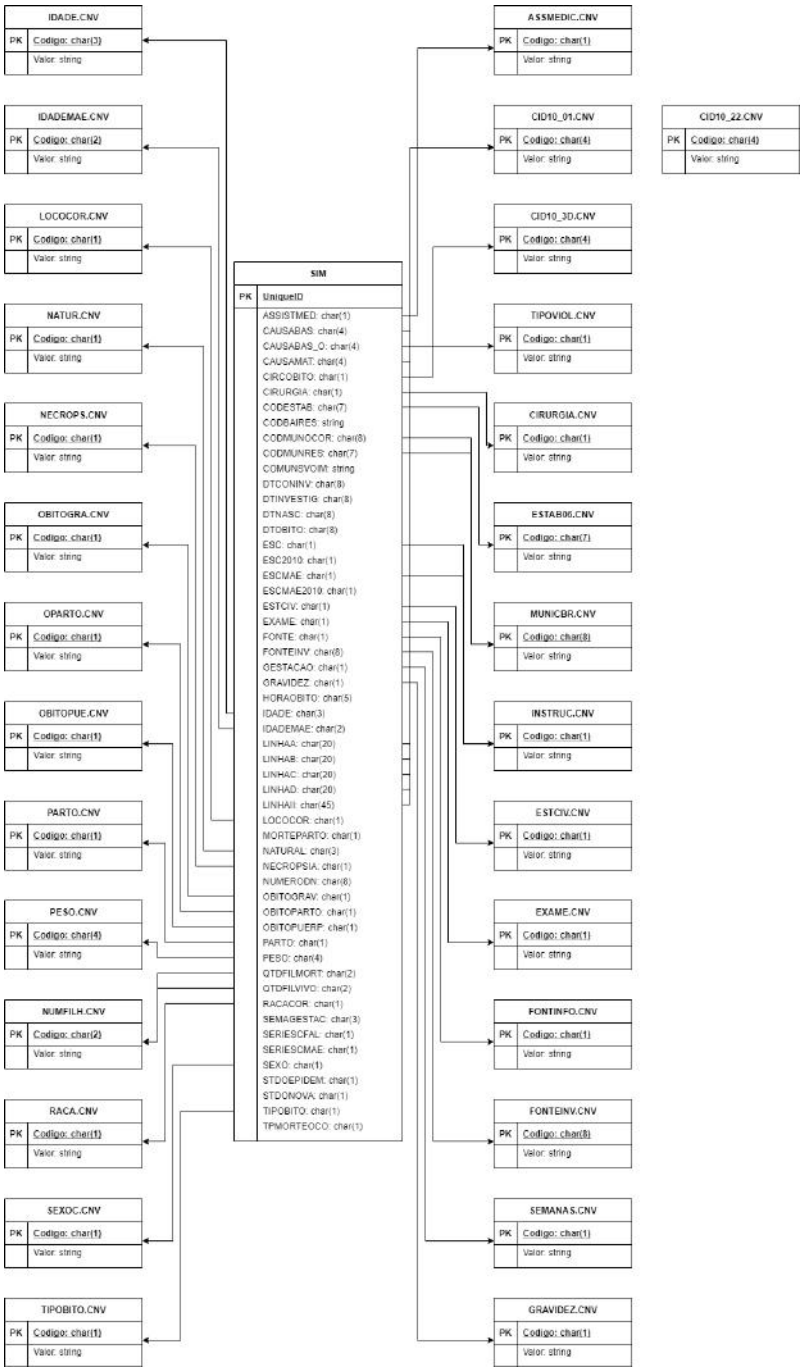
– DIAGRAMA DE ENTIDADES E RELACIONAMENTOS PARA A BASE SINASC

Figura 39 – DER - SINASC



ANEXO C – DIAGRAMA DE ENTIDADES E RELACIONAMENTOS PARA A BASE SIM

Figura 40 – DER - SIM



ANEXO D – DIAGRAMA DE ENTIDADES E RELACIONAMENTOS PARA A BASE SIH

Figura 41 – DER - SIH