Análise dos Dados das Despesas no Portal da Transparência da EMURB

Eduardo Carlos Santos Pereira Junior, Adolfo Pinto Guimarães (Orientador)

Universidade Tiradentes

Resumo— Este trabalho tem por objetivo fornecer uma lógica para a análise dos dados públicos disponíveis no portal da transparência da Empresa Municipal de Obras e Urbanização (EMURB) de acordo com a Lei de Acesso à Informação, n° 12.527, de 18 de novembro de 2011 e a Lei da Transparência, n° 131, de 27 de maio de 2009. As instruções contemplaram as etapas de engenharia de dados necessárias, ou seja, modelagem de um banco de dados *Data Warehouse* utilizando o Esquema Estrela, o auxílio da linguagem de programação *Python* para implementar uma estrutura completa de Extração, Transformação e Carregamento dos dados, processo popularmente chamado de ETL, bem como mostrará o uso da ferramenta de *business intelligence Power BI* para a elaboração de um *Dashboard* que pode ser usado para obter uma melhor visualização dos dados analisados. Os resultados do trabalho englobam a metodologia utilizada, um dicionário para os dados das despesas da EMURB, um modelo de armazém de dados, uma estrutura de banco de dados e ETL montada com auxílio de conteinerização e um *dashboard* que permite análises de forma simplificada.

Palavras chave— Dados Públicos, Engenharia de Dados, Data Warehouse, Visualização dos dados.

Abstract— This work aims to provide a logic for the analysis of public data available on the transparency portal of the Municipal Company of Works and Urbanization (EMURB) in accordance with the Access to Information Law, no. 12,527, of November 18, 2011 and the Transparency Law, No. 131, of May 27, 2009. The instructions included the permitted data engineering steps, that is, modeling a Data Warehouse database using the Estrela Schema, the aid of the Python programming language to Implement a complete data Extraction, Transformation and Loading structure, a process popularly called ETL, as well as showing the use of the Power BI business intelligence tool to create a Dashboard that can be used to obtain a better visualization of the analyzed data . The results of the work include the methodology used, a dictionary for EMURB expenditure data, a data storage model, a database structure and ETL assembled with the aid of containerization and a dashboard that allows simplified analysis.

Index Terms— Public Data, Data Engineering, Data Warehouse, Data Visualization.

1 Introdução

O modelo de gestão governamental do Brasil tem passado por mudanças, principalmente a partir do século XXI. No entanto, algumas mudanças específicas e que são provenientes de leis mudaram a cultura do segredo na administração pública. Esta proposta foi abandonada e outra surgiu em seu lugar, pregando a clareza e facilidade no acesso à informações das entidades públicas (SISGOV, s.d.). Essas leis determinam, dentre outras coisas, que receitas e despesas devem ser disponibilizadas em endereço eletrônico (acessível pela internet).

Essas informações não possuem um padrão em comum e muitas vezes se tornam difíceis de serem analisadas, afastando-se da ideia inicial da clareza e facilidade. Nesse contexto surgiu este estudo, que tem o objetivo de fornecer uma metodologia à qual o cidadão pode interagir e encurtar o caminho para a realização das suas próprias análises, focando nos dados que estão no Portal da Transparência da EMURB [29], que contemplam o período de janeiro de 2018 até a data de realização deste trabalho. O intuito de fornecer este trabalho é tentar aproximar o cidadão para que ele possa realizar uma fiscalização maior da gestão pública.

As etapas de desenvolvimento do trabalho e que são apresentadas neste documento são: extração dos dados, dicionarização dos dados, modelagem do data warehouse, programação do processo de ETL com *Python* e biblioteca *Pandas*, provisionamento da estrutura com Docker, execução dos códigos *Python* e elaboração de um dashboard final com *Power BI* para exemplificar o que foi feito.

Toda a estrutura de código e o arquivo final .pbix

estão disponíveis no repositório utilizado [26]. Lá também estão salvos os dados exatamente como foram extraídos do portal da transparência, ou seja, em formato de planilhas eletrônicas .xslx e sem nenhum tratamento adicional. O processo de tratamento e armazenamento desses dados foi feito com a montagem prévia de uma estrutura que utiliza a tecnologia *Docker*, conforme tópicos 2.3.5 e 3.4 deste trabalho.

O objetivo do uso dos contêineres é simplificar ao máximo as etapas comuns à análise de dados, poupando o cidadão inexperiente em programação e banco de dados de ter que montar toda essa estrutura em sua própria máquina e da necessidade de executar diversos comandos.

A seção 1 apresenta uma introdução de como o trabalho foi desenvolvido. No tópico 2 são apresentadas fontes externas que foram utilizadas como fundamentação para este estudo, contemplando sites de documentação, blogs e trabalhos acadêmicos relacionados. Estas fontes estão devidamente referenciadas no final do documento.

No capítulo 3 é fornecida a metodologia do trabalho, desenvolvendo o raciocínio utilizado em todas as etapas citadas no tópico anterior. A seção 4 traz os resultados finais do trabalho, considerando tudo que pode ser gerado a partir da reprodução do mesmo. Por fim, o 5° tópico apresenta uma conclusão para o trabalho e apresenta os próximos passos para trabalhos futuros.

2 Trabalhos Relacionados e Fundamentação Teórica

2.1 Metodologia Para a Seleção de Trabalhos Relacionados

A seção de pesquisa por trabalhos relacionados foi auxiliada pelos mecanismos de pesquisa do Google, que possui uma ferramenta própria para esse tipo de busca, o Google Acadêmico. Trata-se de um motor de busca para encontrar tudo que pode se relacionar à área acadêmica, o que inclui teses e dissertações, livros e qualquer publicação referente a um estudo. As pesquisas podem ser feitas inclusive em outros idiomas, favorecendo uma amplitude maior ao se tratar de fundamentação teórica para a realização deste trabalho.

O objetivo da pesquisa foi encontrar trabalhos que estivessem relacionados aos temas pertinentes à realização deste estudo, como Análise de Dados, Banco de Dados, Lei de Acesso à Informação, Lei da Transparência, Portal da Transparência etc. Uma gama de trabalhos foram encontrados, mas nenhum que fosse direcionado a EMURB ou a Prefeitura de Aracaju como um todo.

2.2 Trabalhos Relacionados

O trabalho de Scherer [21] buscou utilizar ferramentas de *Business Intelligence - BI* para explorar dados públicos referentes à concessão de bolsas do ProUni. Nele foram usados conceitos referentes a banco de dados, modelagem de bancos de dados, a ferramenta de visualização de dados *Power BI* etc., etapas que constituem o chamado ETL. Por final, algumas visualizações de dados foram criadas usando gráficos. O resultado foi um *dashboard* interativo que resume o que os dados disponíveis estudados representam, tornando o entendimento mais simples.

O trabalho de Filho [22] também teve como objetivo o estudo dos dados públicos referentes ao ProUni, a principal diferença para o trabalho de Scherer [21] foi a ferramenta de visualização de dados utilizada, neste caso sendo o *Tableau Public*. O resultado foi um *dashboard* que resume o que os dados disponíveis representam.

O trabalho de Benedito [23] consistiu em uma análise dos microdados e referenciais disponibilizados pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). O tratamento desses dados foi feito com o *Power BI* e o *Excel*. O resultado final foi similar ao de Scherer [21] e Filho [22], um *dashboard* que permite visualizações mais fáceis, apesar da metodologia ser um pouco diferente

O trabalho de Magalhães e Cardoso [24] consistiu em uma análise dos dados abertos do ensino superior no Brasil, o trabalho deles envolveu as etapas de criação de um *Data Warehouse*, utilizou dados fornecidos pelo INEP e gerou *dashboards* para a visualização dos dados.

2.3 Fundamentação Teórica

2.3.1 Banco de Dados

Desde o princípio humano, os indivíduos sempre enxergaram motivos para registrar acontecimentos e conhecimentos que julgavam importantes, e dependendo da época em que estiveram inseridos, usaram diversas técnicas para isso, exemplos são as pinturas pré-históricas, os hieróglifos inscritos pelos antigos egípcios, o papel etc. (Alves, 2014) [1]. Isso denota que naturalmente existe uma evolução nas maneiras pelas quais nossa espécie procura guardar informações.

Ainda de acordo com Alves [1], o papel teve indiscutível utilidade, porém sua utilização também era caracterizada por alguns pontos negativos. Em seu livro, ele exemplificou como uma loja que guardava informações dos funcionários, clientes e fornecedores em papel teria dificuldade nas tarefas administrativas diariamente, demandando alto prazo para cumprimento.

Posteriormente veio a se materializar umas das primeiras formas de guardar informações com o auxílio das tecnologias advindas da era computacional, a fita de papel perfurada, posteriormente sucedida pelo cartão perfurado. Com a chegada desse artefato, apesar de também ter alguns pontos negativos, os problemas do papel convencional foram resolvidos, uma vez que ele garantia praticidade, eficiência, rapidez na consulta e confiabilidade das informações, levando ao crescimento dos bancos de dados computadorizados (Alves, 2014) [1]. De acordo com Elmasri [2], um banco de dados em sua definição é uma coleção de dados relacionados, onde os dados são fatos que podem ser gravados e que possuem um significado implícito.

2.3.2 Banco de Dados Relacional

Segundo a *Oracle* [5], nos primeiros bancos de dados, os aplicativos armazenavam as informações em estruturas próprias, ou seja, quando fosse necessário que

desenvolvedores criassem aplicativos, eles precisavam conhecer muito daquela estrutura em específico. O modelo relacional forneceu uma maneira padrão para manipular esses dados através de tabelas, sendo uma maneira intuitiva, eficiente e flexível para armazenar informações estruturadas.

Os bancos de dados relacionais são aqueles em que os dados disponíveis são armazenados em tabelas, onde cada tabela terá atributos e linhas (Silberschatz, 2020) [7]. Cada linha em uma tabela representa algo que corresponde a uma entidade ou relacionamento do mundo real (Elmasri, 2005) [2]. Segundo o *Google Cloud* [4], relacionamentos são uma relação lógica entre diferentes tabelas, estabelecidas com base na interação entre elas. Esses relacionamentos são permitidos a partir de atributos chamados chaves primárias, que em tabelas diferentes da sua de origem, podem ser usados como chaves estrangeiras.

2.3.3 *Data Warehouse* - DW

Armazém de dados, em tradução direta, é um repositório que contém dados de diferentes momentos e que pode ser de possível interesse por parte das gerências das organizações (Turban, 2009) [6]. Segundo Oracle [5], trata-se de um repositório central de dados, ou seja, um banco de dados projetado especificamente para consultas e análises rápidas.

De acordo Silberschatz (2020) [7], normalmente suas relações podem ser classificadas como tabelas de fatos e tabelas de dimensão. As tabelas fatos registram acontecimentos individuais, seus atributos podem ser classificados como atributos de medição, que são informações quantitativas que podem ser agregadas, ou podem ser atributos de dimensão, que significam dimensões sobre as quais os atributos de medição podem ser agrupados e visualizados.

Segundo Nery [8], para que a abordagem do DW cumpra suas finalidades, existem dois modelos de dados principais: esquema estrela e floco de neve. O modelo estrela, mais usado, é representado por uma estrutura de uma única tabela fato centralizada e que recebe atributos das suas tabelas dimensões, dimensões estas que são dependentes unicamente de si, não possuindo chaves estrangeiras de outras dimensões (Kimball, 2002) [9]. Já o modelo floco de neve se diferencia apenas pelo fato de que uma tabela dimensão pode depender de outras, ou seja, permite uma relação entre dimensões por meio de atributos chave (Nery, 2013) [8].

Silberschatz (2020) [7] aponta que os dados que serão depositados nessa estrutura devem ser preparados anteriormente na etapa chamada extração, transformação e carregamento (*extract, transform, load - ETL*). O *Google Cloud* [13] define que o ETL descreve um processo completo de coleta de dados estruturados e não estruturados, e os processa de forma que eles se tornem realmente úteis para fins comerciais.

2.3.4 *Python* e Biblioteca *Pandas*

De acordo com a Wikipédia [10], *Python* é uma linguagem de programação de alto nível, conta com orientação a objetos e tipagem dinâmica e forte. Além disso, é um projeto *open-source*, com ampla contribuição da comunidade na produção de conteúdo, nesse contexto surge a biblioteca *Pandas*. Segundo Mulinari escreveu no site Harve [11], trata-se de uma ferramenta de uso gratuito (sob licença BSD) que fornece ferramentas para análise e manipulação de dados, permitindo que as informações sejam extraídas de uma fonte e transformadas num *DataFrame*, onde serão manipulados da maneira desejada - muitas vezes sendo suficiente para suprir as necessidades de extração e manipulação do ETL. Marcus Almeida, em seu artigo publicado no Alura [12], define o *DataFrame* como uma estrutura em formato de tabela, com dados organizados em linhas e colunas.

2.3.5 *Docker*

Docker é uma plataforma aberta para desenvolvimento, envio e execução de aplicativos. Ele permite que um software seja replicado rapidamente, sua proposta principal é que seus recursos sejam funcionais independentemente da infraestrutura que esteja instalada no sistema, dependendo apenas dele mesmo para ter êxito na execução de alguma estrutura de software (*Docker Docs*) [14].

2.3.6 *Power BI*

O *Power BI* é uma plataforma unificada e escalonável para inteligência de negócios (*business inteligence - BI*). Ela permite uma conexão com uma fonte de dados para facilitar a criação de relatórios, além de contar com algumas ferramentas prontas de inteligência artificial (*Power BI*) [15]. Para Sharda, Delen e Turban [17], BI é um termo que combina arquiteturas, ferramentas, base de dados, ferramentas analíticas, aplicativos e metodologias. Quanto à arquitetura, eles defendem que um sistema de BI possui 4 componentes principais: um *data warehouse*, análise de negócios, uma coleção de ferramentas para manipular e analisar os dados e um processo para analisar e monitorar desempenhos.

Dentre as utilidades dessa ferramenta, existe a chamada Expressão de Análise de Dados (*Data Analysis Expressions - DAX*), trata-se de uma coleção de funções, operadores e constantes que podem ser usados como fórmulas para realizar cálculos, alterar linhas ou colunas etc. Essas expressões também são utilizadas em *background* pelo motor do *Power BI* quando aplicado um filtro em alguma visualização de dados (Learn Microsoft) [15].

2.3.7 Dados Abertos

"Dados e conteúdos abertos podem ser livremente usados, modificados e compartilhados por qualquer pessoa para qualquer finalidade." (Open Definition) [18]. No Brasil, a popularização da publicação de dados abertos vindos da gestão pública se deu principalmente por conta da Lei de Acesso à Informação, n° 12.527, de 18 de novembro de 2011, a qual dispõe sobre procedimentos a serem observados pela União, Estados, Distrito Federal e Municípios, com o fim de garantir o acesso à informação (Brasil, 2011) [19]. Segundo o blog SISGOV [28], ainda existe a causa vinda da Lei Complementar n° 131, de 27 de maio de 2009, também chamada de Lei da Transparência. O blog ressalta a obrigatoriedade de que os governos divulguem as despesas e receitas das entidades públicas, enquadrando o objeto de estudo deste

trabalho.

Os principais objetivos destas leis são promover a transparência e a participação social, salvo em situações de sigilo em razão de sua imprescindibilidade para a segurança da sociedade e do Estado.

Para agrupar esses tipos de dados, surgiram os chamados portais da transparência. A prefeitura de Aracaju [20] define que o portal da transparência tem o objetivo de fornecer de forma clara e de fácil compreensão informações sobre a execução orçamentária e financeira do Município, desde receitas e despesas, até a remuneração de servidores ativos, aposentados e pensionistas. Defendendo ainda que com esse portal, a população pode verificar o trabalho realizado pela gestão, garantindo ao cidadão o direito de acompanhar a execução dos projetos e ações da prefeitura.

3 Metodologia

A metodologia utilizada no trabalho está representada pelo diagrama de fluxo abaixo (Figura 1). As esferas numeradas representam o tópico deste documento em que a atividade está detalhada.

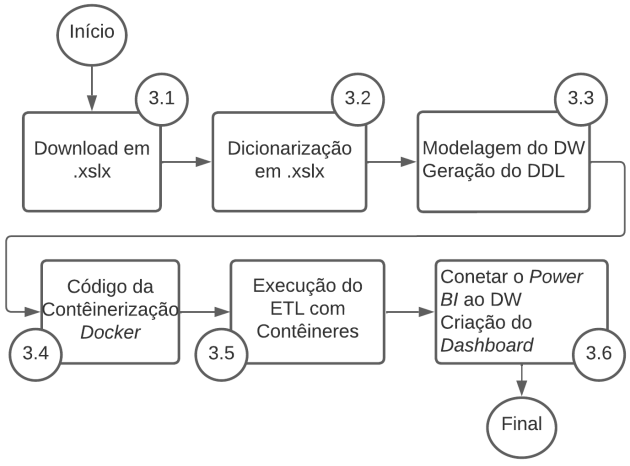


Figura 1: Fluxograma da metodologia.
Fonte: Autor.

3.1 Da Extração dos Dados

O processo prático para realização deste trabalho foi desenvolvido em algumas etapas. A primeira delas foi a obtenção dos dados, para isso houve uma pesquisa para encontrar o local em que eles estariam disponíveis. A pesquisa foi direcionada ao site da EMURB [20]. O endereço eletrônico conta com um layout claro e que facilmente dá acesso ao espaço destinado aos dados de transparência. No Portal da Transparência [25], algumas opções estão listadas, incluindo a de despesa, objeto deste trabalho. Dentre as subcategorias de despesas, foram escolhidas para este trabalho as de empenhos, liquidações e pagamentos. A visualização online é feita através de uma planilha anexada ao próprio site, que permite a utilização de filtros por mês/ano, vale salientar que apenas estão disponíveis os dados a partir do ano 2018. Essa página também permite a extração desses dados através de downloads, tendo que ser feito no máximo de ano em ano (não é possível realizar o download num único arquivo com período maior que 12 meses ou que contenha meses pertencentes a anos diferentes) e por no máximo um tipo de despesa (empenho, pagamento ou liquidação). Nos formatos de download estão habilitados .xlsx, .csv, .xml e .pdf.

Para facilitar a etapa posterior de dicionarização dos dados, a opção escolhida foi o formato de planilhas eletrônicas, que de maneira mais fácil permitiu a visualização das colunas de cabeçalho e o tipo de conteúdo contido nas linhas células. O período extraído foi referente a janeiro de 2018 até setembro de 2023 (última data disponível na realização do trabalho).

3.2 Da Dicionarização dos Dados

A dicionarização dos dados foi feita com o intuito de entender quais colunas seriam aproveitadas para este trabalho, podendo considerar que essa foi a parte inicial da construção do *Data Warehouse*. Para mapear, foi criada uma tabela auxiliar chamada Dicionário de Dados.xlsx, onde foram comparados os nomes das colunas vindas da extração inicial de dados, e como elas vieram a ficar na modelagem, além de abranger a tipagem dos conteúdos, o tamanho necessário e uma breve descrição de o que elas significam.

Nesta etapa alguns problemas foram encontrados, principalmente no que se refere aos cabeçalhos das colunas. Quanto ao número identificador atrelado a despesa, quando analisado nas planilhas empenhos, é nomeado por “SqEmpenho”, no entanto, uma informação similar quando vinda das planilhas de pagamentos, foi nomeada como “Nota de Pagamento”, já nas planilhas de liquidações, foi chamada de “DsEmpenho”. Por representar apenas o identificador daquela despesa e para deixar o armazém de dados enxuto, ambas as colunas foram mapeadas como “id_despesa”.

Outro problema encontrado foi que, nas planilhas de empenhos e pagamentos, existe o cabeçalho “DsEmpenho” para descrever o que foi feito naquelas despesas. Já nas planilhas de liquidações, a mesma informação foi chamada de “DsItemDespesa”. Similarmente ao que foi feito em “id_despesa”, essas colunas foram mapeadas como “desc_despesa”.

Além disso, nas planilhas de empenhos e pagamentos, a coluna chamada de “DsItemDespesa” informava um identificador do item da despesa e uma descrição do que é aquele item, informações essas separadas por um hífen, um exemplo que não representa os valores reais: 1 - Obras e Instalações; 2 - Deslocação de pessoal. Apesar de também existir esse tipo de informação nas planilhas de liquidações, o cabeçalho da coluna vem sem nome. A solução adotada foi mapear o conteúdo dessas colunas em duas novas colunas - “id_item_despesa” e “desc_item_despesa” respectivamente, usando o hífen como separador do conteúdo.

Quanto à coluna responsável por informar valores referentes a anulação, outra divergência foi encontrada entre os dados. Nas planilhas empenhos, a coluna foi definida como “Anulado”, já nas planilhas pagamentos e liquidações foi definida como “Anulação”.

A solução adotada foi mapear para que todas passassem a ser chamadas de “valor_anulado”.

Terminados os problemas citados, o dicionário dos dados seguiu com a separação de algumas colunas que ainda não foram citadas acima e que receberam nomes novos:

- “Órgão”: “id_orgao” e “desc_orgao”;
- “Unidade”: “id_unidade” e “desc_unidade”;
- “Data”: “id_data”, “ano”, “mes”, “dia”;
- “Credor”: “id_credor”, “codigo_nacional”, “desc_credor”;
- “Empenhado”: “valor_empenhado”;
- “Pago”: “valor_pago”;
- “Liquidado”: “valor_liquidado”;
- “Reforçado”: ”valor_reforcado”;
- “Retido”: “valor_retido”.

Além disso, ao dicionário foram adicionadas duas novas colunas que não existem nas tabelas do portal da transparência. As colunas são “id_tipo_despesa” e “desc_tipo_despesa”, elas servem respectivamente para atribuir um identificador para diferenciar quando o dado vem de uma planilha de empenho, liquidação ou pagamento, enquanto a outra serve para escrever como texto, exatamente se a planilha é de empenho, liquidação ou pagamento. A tabela abaixo ilustra o prescrito:

id_tipo_despesa	desc_tipo_despesa
1	Empenho
2	Liquidação
3	Pagamento

O dicionário de dados completo está disponível no repositório do projeto [26].

3.3 Da Modelagem dos Dados

A modelagem adotada foi voltada aos conceitos de construção de *Data Warehouse*, sendo completamente alinhada com o modelo estrela, isso porque a construção de somente uma tabela fato e de tabelas dimensões que não possuem necessidade de interação com outras dimensões, foi suficiente para que houvesse um armazém de dados consistente.

A etapa de modelagem foi realizada utilizando os mapeamentos feitos anteriormente com a dicionarização dos dados enquanto concretizava as tabelas e suas respectivas colunas com o auxílio do programa *Oracle Data Modeler*, que é uma interface gráfica para modelagem e que ao final de tudo é capaz de emitir um DDL. Andrade [27] define que DDL ou *Data Definition Language* (Linguagem de Definição de dados) permite ao usuário definir as novas tabelas e os elementos que serão associados a elas. Também define que ela é responsável pelos comandos SQL de criação e alteração no banco de dados, sendo composto por três comandos: *CREATE*, *ALTER* e *DROP*.

Ainda nos conceitos de Andrade [27], SQL ou *Structured Query Language* (Linguagem de Consulta Estruturada), é uma linguagem padrão de gerenciamento de dados que interage com os principais bancos de dados baseados no modelo relacional. Alguns dos principais sistemas que utilizam SQL são: *Oracle*, *PostgreSQL*, *Firebird*, *MySQL*, entre outros.

3.4 Do Ambiente de Execução do Projeto

O projeto foi desenvolvido para ser executado através de uma camada de virtualização, para isso foi escolhido o *Docker* por sua simplicidade, também foi usado o *Docker Compose*, que nada mais é do que um orquestrador para os serviços *Docker*, sendo algo parecido com um plugin. O ambiente de execução foi montado com dois contêineres, um do banco de dados *PostgreSQL* onde o *Data Warehouse* foi hospedado e outro com um serviço *Python* que executa os programas de ETL (Figura 23).

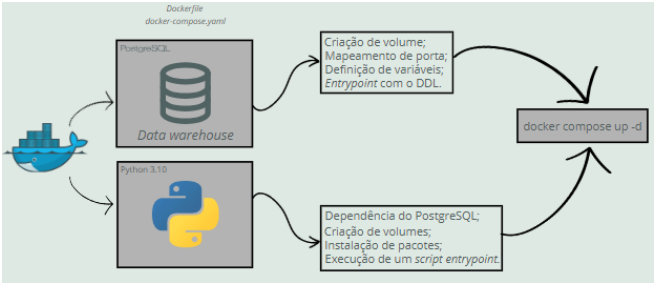


Figura 22: Representação da conteinerização.
Fonte: Autor.

Essa escolha de implementar a arquitetura com *Docker* surgiu a partir da premissa de que os portais da transparência servem para popularizar a transparência e a participação da sociedade nos gastos públicos. No entanto, não são todos os cidadãos que possuem os conhecimentos necessários para executar todas as etapas que são comuns a um trabalho como o realizado neste estudo, seriam diversos comandos e alguns dias de estudo para entender como fazer. Para sanar isso, o *Docker* entrou em cena, permitindo que após a configuração dos arquivos *Dockerfile* e *docker-compose.yaml*, a tarefa para o usuário final fique mais simples, sendo apenas necessário a instalação do *Docker Desktop* e do orquestrador *Docker Compose*, para que então toda a estrutura seja montada e executada com apenas um único comando.

O provisionamento do banco de dados se deu a partir da montagem da estrutura onde o comando ddl.sql obtido anteriormente, foi usado de maneira que ao criar o contêiner, tudo fosse executado e em poucos instantes o banco de dados estivesse disponível. Para chegar a esse resultado, no arquivo de orquestração *docker-compose.yaml* foi definido um serviço ‘postgres’ (que utiliza a imagem do *PostgreSQL*), dentro do bloco de código desse serviço foi fornecido o nome que seria dado ao contêiner (postgres) e a instrução DDL através de um *volume* montado. Ainda foram definidas variáveis de ambiente através do atributo *enviroment* para que fosse definido o nome do usuário *root*, a senha do mesmo e o nome do banco de dados a ser criado. Essas variáveis foram preenchidas respectivamente com ‘postgres’, ‘postgres’, ‘*datawarehouse*’. Também foi mapeada uma porta de

execução do contêiner para a máquina local que está sendo utilizada através do atributo *ports*, por convenção, a porta definida para o *PostgreSQL* é a 5432.

O provisionamento do serviço *Python* foi feito através do mesmo arquivo de orquestração *docker-compose.yaml* e do arquivo *Dockerfile*. No primeiro, um novo serviço foi definido como 'etl', nele, o nome do contêiner foi definido como 'python' e foram passados através de *volumes* os arquivos com extensão .py que seriam executados mais tarde e os arquivos de planilha eletrônica onde os dados estavam até então armazenados. Após, foi definido que o contêiner deve ser criado com base nos atributos definidos em *Dockerfile*, que foi criado para especificar o arquivo *requirements.txt*, responsável por baixar as dependências necessárias para o uso das bibliotecas utilizadas no projeto. O atributo 'CMD' foi definido para apontar para um arquivo 'execute.sh' que aponta para todos os arquivos .py criados no ETL e os executa.

O processo feito anteriormente pode ser colocado em prática através do comando 'docker-compose up'. Este comando automaticamente irá baixar as imagens através da internet, construir os contêineres e configurá-los. Após criados, o contêiner 'python' irá executar 'execute.sh', que por sua vez irá executar os arquivos .py desejados. Após a finalização da execução, o banco de dados estará devidamente povoado com os dados fornecidos.

3.5 Da Extração, Transformação e Carga (ETL)

Como denotado no parágrafo anterior, somente a linguagem de programação *Python* foi utilizada. Também houve a instalação de duas bibliotecas extras para o projeto: *pandas, openpyxl* e *psycopg2-binary*.

A biblioteca *openpyxl* é responsável por permitir uma integração maior entre a linguagem *Python* e as planilhas *Excel*. Ela serviu de "suporte" para a biblioteca *pandas* na etapa de extração dos dados das planilhas.

A biblioteca *pandas* foi protagonista no processo. Para isso, foram feitos códigos .py para cada tabela do *data warehouse*, todos eles contaram com um loop para ler as planilhas de todos os anos e dos três tipos de despesas analisados. A cada leitura de tabela, os dados eram adicionados a um único *dataframe pandas*. Após o término da leitura, começou a transformação dos dados conforme os passos descritos no capítulo 3.3, ou seja, basicamente foi uma tradução do que estava no dicionário para a linguagem *Python*, onde tudo foi colocado em prática.

A biblioteca *psycopg2-binary* foi responsável por realizar a conexão com o banco de dados. Seus métodos foram utilizados conforme um loop iterou sobre o *dataframe* único criado, a cada linha era feito o comando SQL de inserção na tabela devida, após a finalização do loop, ou seja, após todas as linhas terem sido inseridas, o código executa a função de *commit*, responsável por de fato realizar a inserção e salvar para visualização no banco.

3.6 Da Visualização com *Power BI Desktop*

Após a finalização dos scripts do capítulo 3.5, o *Power BI Desktop* foi utilizado para realização de gráficos que permitissem um melhor entendimento dos dados analisados. O modo de usar foi através de uma conexão direta com o banco de dados.

Para tal, dentro do *Power BI* Desktop foi utilizada a opção 'Obter Dados', onde foram fornecidas as informações referentes ao banco de dados criado com Docker. Caso a conexão seja feita com sucesso, as tabelas serão listadas e será possível selecionar todas para efetivamente importar para o programa.

Dentro do *Power BI*, foram feitas mais duas camadas de tratamento de dados, denotando a capacidade que a linguagem DAX tem para realizar tarefas de transformação de dados. Na primeira, na tabela FT_DESPESAS, os valores das colunas: 'valor_empenhado', 'valor_pago', 'valor_liquidado', 'valor_reforcado' e 'valor_retido' possuíam alguns valores do tipo *Not a Number* (NaN), o tratamento foi encontrar e substituir em todas as ocorrências pelo valor "R$0,00". O intuito desse tratamento foi padronizar para que as colunas acima pudessem ser transformadas para o tipo *Number* com formato fiduciário, ou seja, "R$" deixou de ser um texto e passou a ser entendido como moeda. A segunda camada foi criar uma nova coluna chamada 'desc_tipo_credor' em DM_CREDOR, usando novamente DAX, o script foi para que o valor de 'codigo_nacional_credor' fosse capturado em cada linha e a coluna nova fosse preenchida respectivamente com o valor devido. A diferenciação entre ambos se dá através dos dígitos "*" na coluna 'codigo_nacional_credor', onde os que são pessoa física devem ter seu Cadastro de Pessoa Física (CPF) protegido.

Ao término desses pequenos tratamentos, as informações estavam prontas para utilização nas visualizações. O modelo de *dashboard* adotado foi realizado em duas páginas para cada tipo de despesa. Em todas as páginas foram utilizados os visuais de segmentação de dados, que nada mais são do que filtros que funcionam a partir da coluna que é passada como valor ao mesmo, no caso, as páginas foram feitas com filtro de tempo (Figura 4), onde o valor do filtro foi uma hierarquia feita na ordem ano/mês que trabalha com seleção única (só pode ser escolhido um ano por vez, vários meses de um mesmo ano são permitidos). A construção da hierarquia se dá através da guia de dados, onde a coluna topo da hierarquia deve receber um clique com o botão direito e deverá ser selecionada a opção criar hierarquia (Figura 2), após, a coluna a ser o próximo nível da hierarquia também deve receber o mesmo clique, e a opção adicionar a hierarquia deve ser utilizada (Figura 3).

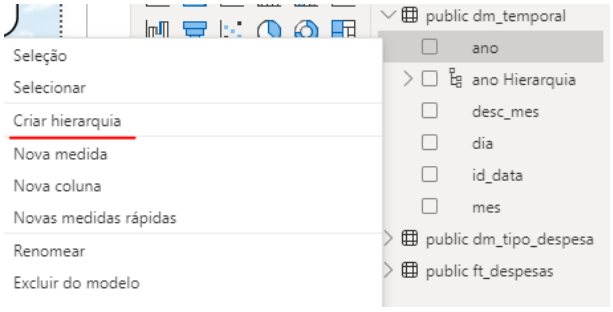


Figura 2: Criação de hierarquia no *Power BI*.
Fonte: Autor.

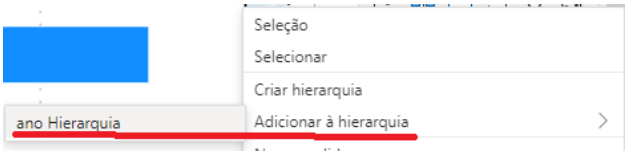


Figura 3: Adição de coluna à hierarquia existente.
Fonte: Autor.

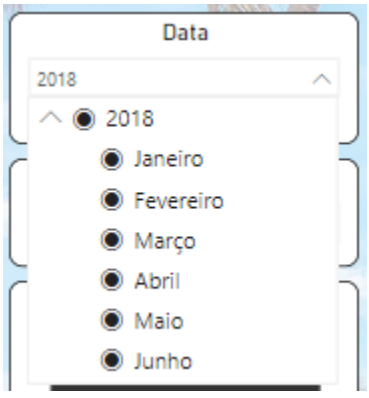


Figura 4: Segmentação de dados de seleção única e com hierarquia.
Fonte: Autor.

Outro filtro utilizado foi para diferenciar o tipo de credor, distinguindo entre pessoa física e pessoa jurídica, de acordo com o parágrafo anterior. Nesse caso, a segmentação foi feita com o estilo de lista vertical e permite seleção múltipla, conforme figura 5.

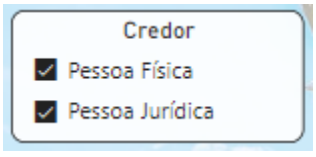


Figura 5: Segmentação de dados em lista com escolha múltipla.
Fonte: Autor.

O último filtro interativo utilizado foi com o valor da coluna 'desc_item_despesa', permitindo ao usuário escolher o(s) item(ns) de despesa desejados. O estilo utilizado foi o tipo bloco e permite seleção múltipla, incluindo também a opção "Selecionar tudo" para maior comodidade. O filtro está representado na figura 6.

Figura 6: Segmentação de dados do tipo bloco e com seleção múltipla.
Fonte: Autor.

De forma não interativa porém funcional, um filtro foi utilizado e travado para dizer se a página é referente ao tipo de despesa empenho, pagamento ou liquidação. Ele foi utilizado para situar o usuário sobre qual informação ele está visualizando. A figura 7 representa o filtro numa página de empenhos. A "trava" citada é feita na aba "Filtros" com a segmentação de dados selecionada, bastando um clique ao cadeado conforme figura 8.

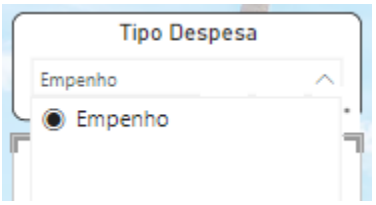


Figura 7: Segmentação de dados do tipo de despesa.
Fonte: Autor.

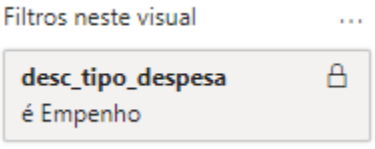


Figura 8: Trava na interação com a segmentação.
Fonte: Autor.

Para a exibição dos dados foram utilizados os cartões, que são caixas flutuantes que podem receber valores. No caso, eles foram utilizados para demonstrar de acordo com a filtragem de cada página, os valores que o portal da transparência trás sobre cada tipo de despesa. Exemplo na figura 9:

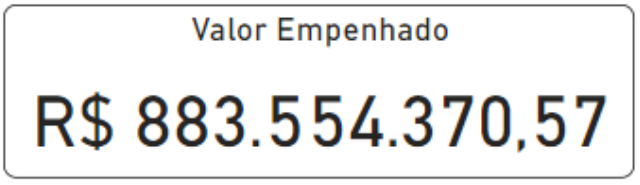
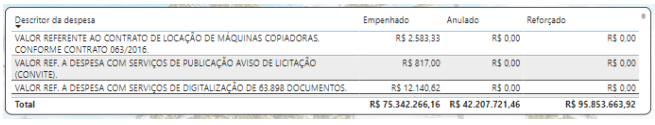


Figura 9: Visual cartão no *Power BI*
Fonte: Autor.

Nas páginas ímpares, visuais de tabela foram utilizados para detalhar o que foi feito nas despesas (desc_despesa) e seus respectivos valores de acordo com os tipos de despesas. Essa visualização permite ainda, para as colunas que possuem valores numéricos, uma visualização do total contido na tabela, ou seja, somando todas as linhas. Exemplo da página Empenhos 1 na figura 10.



Descritor da despesa	Empenhado	Anulado	Reforçado	
VALOR REFERENTE AO CONTRATO DE LOCAÇÃO DE MÁQUINAS COPIADORAS, CONFORME CONTRATO 063/2016.	R$ 2.583,33	R$ 0,00	R$ 0,00	
VALOR REF. A DESPESA COM SERVIÇOS DE PUBLICAÇÃO AVISO DE LICITAÇÃO (CONVITE).	R$ 817,00	R$ 0,00	R$ 0,00	
VALOR REF. A DESPESA COM SERVIÇOS DE DIGITALIZAÇÃO DE 63.898 DOCUMENTOS.	R$ 12.140,62	R$ 0,00	R$ 0,00	
Total	R$ 75.342.266,16	R$ 42.207.721,46	R$ 95.853.663,92	

Figura 10: Representação do visual tabela no *Power BI*.
Fonte: Autor.

Nas páginas pares, os visuais utilizados tiveram

um foco além da tabela dita anteriormente. Nesse caso, foram utilizados gráficos para poder facilitar a visualização dos dados através de agrupamentos e recursos visuais que facilitam a compreensão. Para dar destaque a informações que possivelmente sejam de maior interesse dos usuários, essas visualizações resumem os 5 maiores valores de cada tipo de despesa, separados por: 1 - credores; 2 - item despesa. Esse resumo dos cinco maiores é feito na aba de filtros com a visualização selecionada, lá deve ser definido o tipo de filtro como "N superior", a opção seguinte como "superior" e a caixa a direita deve ser o número de registros que o usuário deseja, no caso foram 5. O processo está representado na figura 11.

Ainda nas páginas pares, visuais de gráficos de colunas foram utilizados para comparar os valores empregados ao longo dos anos, permitindo que facilmente o usuário saiba qual o ano com mais ou menos gastos. Esse gráfico não obedece a segmentação de dados de tempo (Figura 4). Para cortar essa relação, com o gráfico selecionado, a aba de "Formato" foi aberta e a função "Editar interações utilizadas", acima de cada visual aparecerá um ícone para filtrar ou cortar interação, conforme a figura 12.

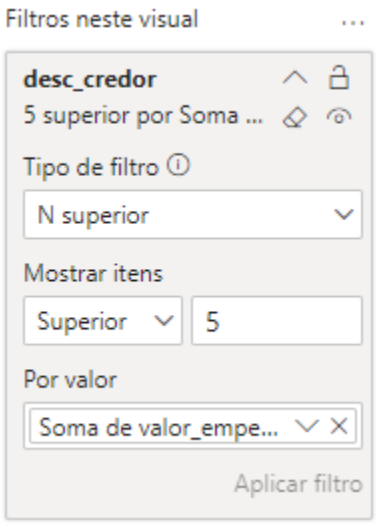


Figura 11: Filtragem N superior.
Fonte: Autor.

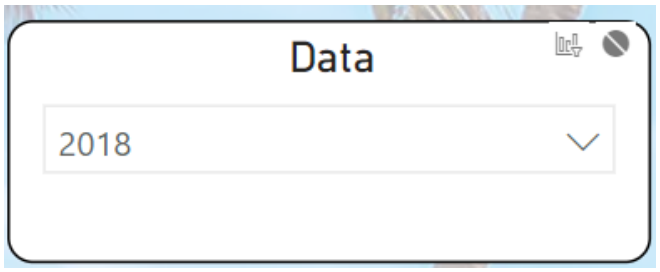


Figura 12: Remoção de interação entre visuais.
Fonte: Autor.

4 Resultados

O resultado de toda a metodologia citada no capítulo anterior é um conjunto de dados pronto para ser analisado ou reaproveitado, tendo um *dashboard* já pronto e que ainda pode ser implementado para ter novos visuais e permitir novos *insights*, um dicionário de dados para facilitar uma nova análise (visto que o Portal da Transparência não dispõe este recurso) e a infraestrutura disponibilizada com *Docker*, permitindo a criação de um banco de dados e de um serviço *Python* que executa scripts no momento da sua criação.

Em relação aos *dashboards*, nas páginas dedicadas ao tipo de despesa Empenho, foram mostrados cartões com valores empenhados, anulados e reforçados. Além disso, foi preparada uma tabela que descreve o que foi feito em cada empenho e detalha seus valores. Esses visuais podem ser segmentados por corte temporal, itens de despesa e tipos de credores, conforme figura 13.

Ainda falando dos empenhos, foram implementados visuais que demonstram, de acordo com as segmentações, os 5 credores que mais receberam valor empenhado, os 5 itens de despesa que tiveram maior valor empenhado e o valor empenhado, anulado e reforçado ao longo dos anos de estudo. Dentro desses visuais, é possível também segmentar dados entre eles, por exemplo, ao clicar no credor CAMEL EMPREENDIMENTOS no gráfico de credores, os demais ficam realçados apenas naquilo que é proporcional aos empenhos relativos a este credor. Além disso, é possível combinar dois ou mais visuais para segmentar dados. O dito está representado nas figuras 14 e 15.

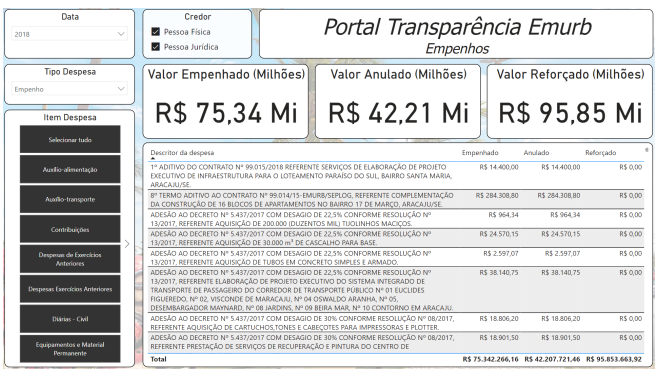


Figura 13: Página n° 1 do *dashboard*.
Fonte: Autor.

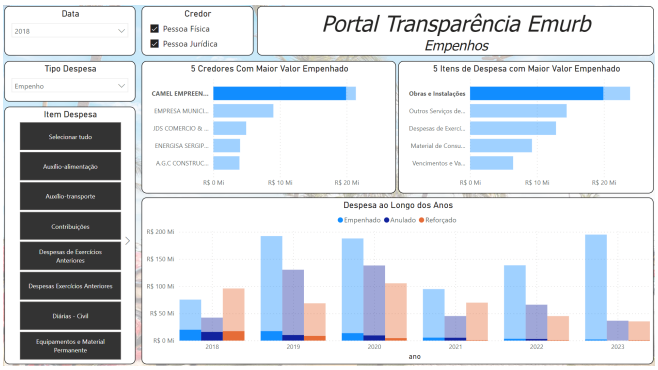


Figura 14: Representação do filtro entre visuais de gráficos.
Fonte: Autor.

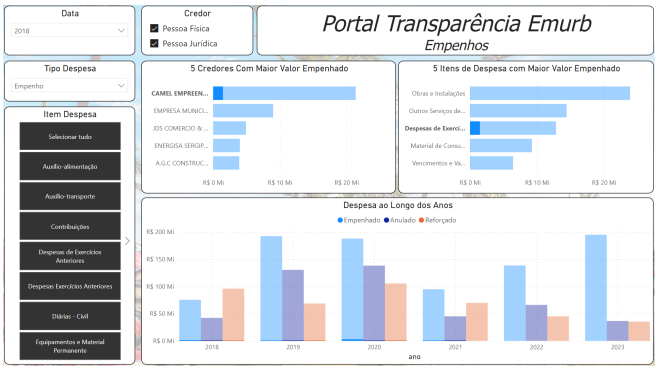


Figura 15: Representação do filtro entre visuais de gráficos.
Fonte: Autor.

As páginas de pagamentos foram bastante similares às de empenho, apenas com os tipos de valores de despesa devidamente trocados, já que em empenhos são fornecidos valores empenhados, anulados e reforçados, enquanto em pagamentos são valores pagos, anulados e retidos. As páginas estão representadas nas figuras 16 e 17.

As páginas referentes a liquidação também foram bastante similares às de empenhos e pagamentos, onde dessa vez os valores foram trocados para valor liquidado, anulado e retido. As páginas estão representadas nas figuras 18 e 19.

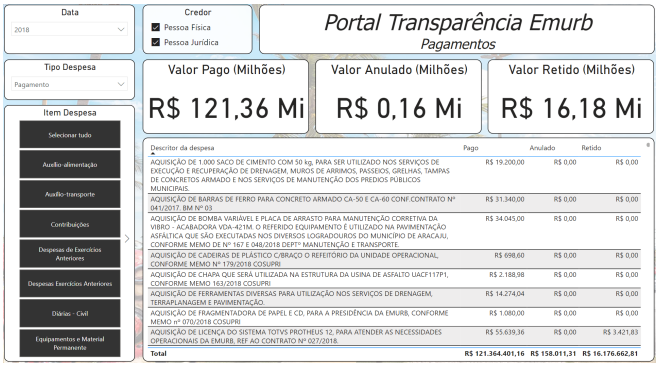


Figura 16: Página n° 3 do *dashboard*.
Fonte: Autor.

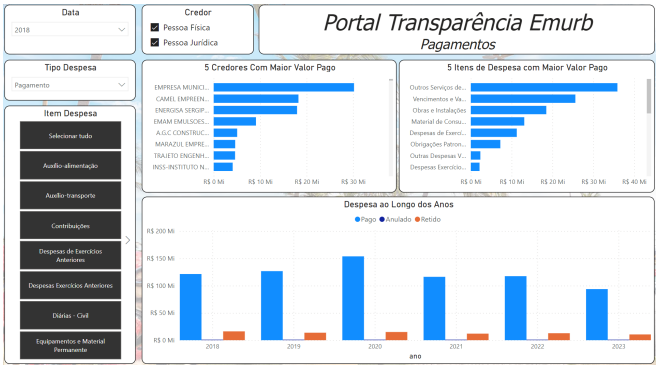


Figura 17: Página n° 4 do *dashboard*.
Fonte: Autor.

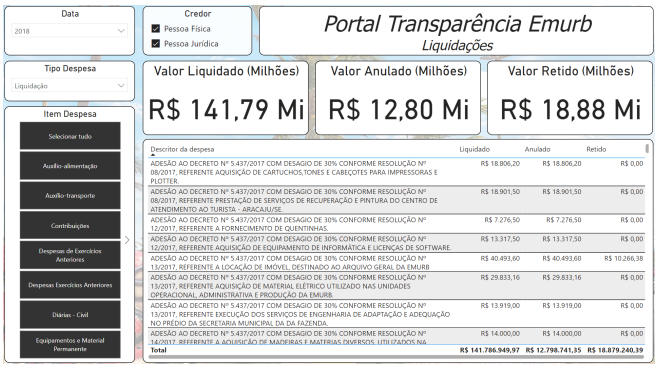


Figura 18: Página n° 5 do *dashboard*.
Fonte: Autor.

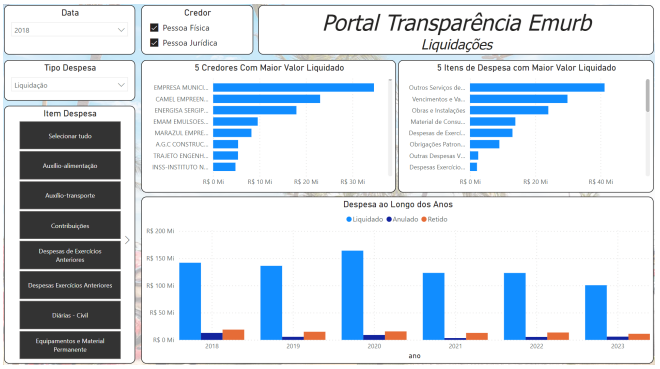


Figura 19: Página n° 6 do *dashboard*.
Fonte: Autor.

Diante do exposto, é possível notar que diversas análises podem ser feitas, adquirindo diversos resultados que dependem basicamente do que o usuário procura através dos dados.

O dicionário de dados ficou como resultado armazenado em uma planilha eletrônica disponível no repositório do projeto [26]. Nele estão mapeadas as colunas originais (da base de dados) e como elas ficaram na modelagem de dados, ainda possui a indicação das colunas criadas durante este estudo (id_tipo_despesa e desc_tipo_despesa), a tipagem dos dados das colunas, os tamanhos e as respectivas descrições. Este dicionário possui 3 guias, uma para cada tipo de despesa estudada. O dicionário é ilustrado pela figura 20.

COLUNA (Fonte)	COLUNA (Modelagem)	TIPO	TAMANHO	DESCRITOR
Órgão	id_orgao	Integer		Identificador do órgão responsável
	desc_orgao	String	45	Descritor do órgão responsável
Unidade	id_unidade	Integer		Identificador da unidade responsável
	desc_unidade	String	60	Descritor da unidade responsável
Data	id_data	Integer		Código da data
	ano	Integer		Número do ano
	mes	Integer		Número do mês
	dia	Integer		Número do dia
Credor	id_credor (Não existe)	Integer		Identificador para o credor (Número Aleatório)
	codigo_nacional_credor	String	18	CPF ou CNPJ do credor
	desc_credor	String	100	Nome do credor
Liquidado	valor_liquidado	Number	(11, 2)	Valor liquidado (Esclarecer)
Anulação	valor_anulado	Number	(11, 2)	Valor anulado (Esclarecer)
Retido	valor_retido	Number	(11, 2)	Valor retido (Esclarecer)
DsEmpenho	id_despesa	Integer		Identificador atrelado a despesa
DsitemDespesa	desc_despesa	String	500	Descritor do que foi feito
Coluna sem cabeçalho	id_item_despesa	Integer		Identificador do tipo da despesa
	desc_item_despesa	String	25	Descritor do tipo da despesa
-	id_tipo_despesa	Integer		Se empenho, liquidação ou pagamento (1, 2, 3 respectivamente)
	desc_tipo_despesa	String	25	Descritor se é empenho, liquidação ou pagamento

Figura 20: Página do dicionário de dados.
Fonte: Autor.

A infraestrutura *Docker* fica como um resultado que facilita a replicação do ambiente utilizado, bastando apenas a instalação das ferramentas e a utilização do comando "docker-compose up". A partir disso o usuário terá um banco de dados criado e um contêiner *Python* para execução de scripts dessa linguagem. A figura 21 mostra a estrutura no *Docker Desktop*.

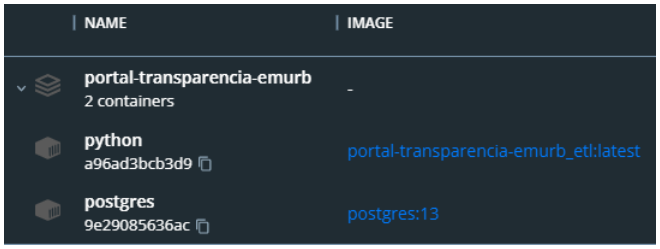


Figura 21: Contêineres estruturados.
Fonte: Autor.

Essa estrutura também pode ser modificada para alterar os recursos que serão criados no banco de dados, instalar novos pacotes no *Python*, executar outros scripts etc..

5 Conclusão e Trabalhos Futuros

5.1 Conclusão

Este trabalho é concluído demonstrando uma etapa completa para análise de dados do Portal da Transparência da EMURB, contemplando desde a extração de dados até a elaboração de um *dashboard* ao final. O trabalho trouxe ainda uma alternativa para tentar facilitar o acesso a esse tipo de estudo, ao ressaltar o *Docker* como algo que pode simplificar a replicação de tarefas, encurtando o caminho das etapas necessárias. Todos os resultados e arquivos utilizados neste projeto estão disponíveis no repositório utilizado [26], incluindo um link para acesso à visualização *web* do painel.

5.2 Trabalhos Futuros

Os trabalhos futuros incluirão a expansão do estudo para outras empresas e fundações da Prefeitura de Aracaju (Figura 22), estas que possuem algumas similaridades em seus portais da transparência. O objetivo será manter o processo de ETL válido para todos esses portais e permitir a união de tudo num único *data warehouse* que possa ser montado em qualquer sistema operacional com suporte a tecnologia *Docker*, continuando com a essência de que o cidadão sem conhecimentos amplos na área de dados possa ter acesso a esta metodologia aplicada, consiga adaptá-la às suas necessidades e possa executar tudo com apenas um comando.

Figura 22: Empresas e fundações da Prefeitura de Aracaju.
Fonte: Autor.

Referências

[1] ALVES, William P. Banco de Dados. Editora Saraiva, 2014. E-book. ISBN 9788536518961. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536518961/>. Acesso em: 20 nov. 2023.

[2] ELMASRI, R. NAVATHE, S. B. Sistemas de Banco de Dados. São Paulo : Pearson Education, sob o selo Addison Wesley, 2005.

[3] SOUZA, Ivan. Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa. Rockcontent, 2020. Disponível em: <https://rockcontent.com/br/blog/banco-de-dados/> Acesso em: 20 nov. 2023

[4] O que é um banco de dados relacional? Google Cloud, [s.d.]. Disponível em: <https://cloud.google.com/learn/what-is-a-relational-database?hl=pt-br>. Acesso em: 21 nov. de 2023.

[5] O que é um banco de dados relacional (RDBMS)?. Oracle, [s.d.]. Disponível em: <https://www.oracle.com/br/database/what-is-a-relational-database/>. Acesso em 21 nov. de 2023.

[6] TURBAN, E. SHARDA, R. ARONSON, J.E. KING, D. Business Intelligence: Um Enfoque Gerencial. Grupo A, 2009. 9788577804252. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577804252/> . Acesso em: 20 nov. 2020.

[7] SILBERSCHATZ, Abraham. Sistema de Banco de Dados. Grupo GEN, 2020. E-book. ISBN 9788595157552. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595157552/>. Acesso em: 21 nov. 2023.

[8] NERY, Felipe Rodrigues Machado. Tecnologia e Projeto de Data Warehouse. 6ª Edição, São Paulo, SP, 2013.

[9] KIMBALL, R. ROSS, M. The Data Warehouse Toolkit, Second Edition. Canadá: John Wiley & Sons, Inc. 2002.

[10] Python. Wikipédia, 2023. Disponível em: <https://pt.wikipedia.org/wiki/Python>. Acesso em: 22 nov. de 2023.

[11] MULINARI, Bruna. Pandas Python: vantagens e como começar, [s.d.]. Disponível em: <https://harve.com.br/blog/programacao-python-blog/pandas-python-vantagens-e-como-comecar/>. Acesso em 22 nov. de 2023.

[12] ALMEIDA, Marcus. Pandas Python: o que é, para que serve e como instalar. 2023. Disponível em: <https://www.alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar#:~:text=DataFrame,Series%20sob%20um%20mesmo%20index.> Acesso em: 21 nov. 2023.

[13] O que é ETL?. Google Cloud, [s.d.]. Disponível em: <https://cloud.google.com/learn/what-is-etl?hl=pt-br.>. Acesso em: 22 nov. de 2023.

[14] Visão geral do Docker. Docker Docs, [s.d.]. Disponível em:

<https://docs.docker.com/get-started/overview/>. Acesso em: 22 nov. de 2023.

[15] O que é o Power BI?. Power BI, [s.d.]. Disponível em: <https://powerbi.microsoft.com/pt-br/what-is-power-bi/>. Acesso em: 22 nov. de 2023.

[16] Aprenda noções básicas sobre o DAX no Power BI Desktop. Microsoft Learn, 2023. Disponível em: <https://learn.microsoft.com/pt-br/power-bi/transform-model/desktop-quickstart-learn-dax-basics>. Acesso em: 22 nov. de 2023.

[17] SHARDA, Ramesh; DELEN, Dursun; TURBAN, Efraim. Business intelligence e análise de dados para gestão do negócio. Grupo A, 2019. E-book. ISBN 9788582605202. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582605202/> . Acesso em: 22 nov. 2023.

[18] Open Definition. Definição aberta 2.1, [s.d.]. Disponível em: <https://opendefinition.org/>. Acesso em: 22 de nov. de 2023.

[19] Brasil. Lei n° 12.527, de 18 nov. de 2011. Lei do Acesso à Informação. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm> . Acesso em: 22 de nov. de 2023.

[20] EMURB. Portal da Transparência, [s.d.] Disponível em: <https://transparencia.aracaju.se.gov.br/emurb/portal-da-transparencia/> . Acesso em: 22 nov. 2023.

[21] SCHERER, Elis. Utilização de Recursos de BI para Análise de Dados Públicos. **Manancial - Repositório Digital da UFSM**, 2021, Santa Maria - RS. Disponível em: <https://repositorio.ufsm.br/handle/1/28506>. Acesso em: 19 nov. 2023.

[22] FILHO, Vanderlei; BRANDI, Letícia. Um estudo focado ao PROUNI através da análise de dados abertos: período de 2005 até 2016. **PRISMA.COM**, Porto - Portugal, n. 38, p. 37-53, 2019.

[23] OLIVEIRA, Benedito. Matemática e Suas Tecnologias: Um Levantamento Estatístico das Questões do Exame Nacional do Ensino Médio (ENEM) dos anos de 2017 a 2020, 2022. **Repositório Institucional UEA**. Manaus - AM. Disponível em: <http://repositorioinstitucional.uea.edu.br//handle/riuea/4266>. Acesso em: 22 nov. 2023.

[24] MAGALHÃES, Halley; CARDOSO, Abreu. Análise de Dados Abertos Sobre o Ensino Superior Brasileiro. **Repositório da UNB**, 2016, Brasília - DF. Disponível em <https://bdm.unb.br/handle/10483/17719>. Acesso em: 22 nov. de 2023.

[25] EMURB. Obras e Urbanização, [s.d.]. Disponível em: <https://www.aracaju.se.gov.br/obras_e_urbanizacao/>. Acesso em: 24 nov. 2023.

[26] PEREIRA, Eduardo. portal-transparencia-emurb. Disponível em: <https://github.com/eduardojnr/portal-transparencia-Emurb>. Acesso em: 24 nov. 2023.

[27] ANDRADE, Ana. Principais comandos SQL, agosto de 2019. Disponível em: <https://www.treinaweb.com.br/blog/principais-comandos-sql#>. Acesso em: 24 nov. de 2023.

[28] SISGOV. Transparência na gestão pública: o que é e como surgiu?. Disponível em: <https://www.sisgov.com/transparencia-na-gestao-publica-o-que-e-e-como-surgiu/>. Acesso em: 04 dez. de 2023.

[29] EMURB. Despesas, [s.d.]. Disponível em: <https://transparencia.aracaju.se.gov.br/emurb/despesas-2/>. Acesso em: 04 dez. de 2023.