UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Gustavo Fukushima de Salles

Extração, transformação e carga de dados sobre compras da Prefeitura de Florianópolis usando Airflow

Florianópolis
2025

Gustavo Fukushima de Salles

Extração, transformação e carga de dados sobre compras da Prefeitura de Florianópolis usando Airflow

Trabalho de Conclusão de Curso do Curso de Graduação em Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de bacharel em Ciências da Computação.
Orientador: Prof. Renato Fileto, Dr.

Florianópolis
2025

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

http://portalbu.ufsc.br/ficha

Gustavo Fukushima de Salles

Extração, transformação e carga de dados sobre compras da Prefeitura de Florianópolis usando Airflow

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "bacharel em Ciências da Computação" e aprovado em sua forma final pelo Curso de Graduação em Ciências da Computação.

Florianópolis, [dia] de [mês] de [ano].

Prof. XXXXXX, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Renato Fileto, Dr.
Orientador

Prof.(a) xxxx, Dr(a).
Avaliador(a)
Instituição xxxx

Prof.(a) xxxx, Dr(a).
Avaliador(a)
Instituição xxxx

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Inserir os agradecimentos aos colaboradores à execução do trabalho.

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

"Texto da Epígrafe.
Citação relativa ao tema do trabalho.
É opcional. A epígrafe pode também aparecer
na abertura de cada seção ou capítulo.
Deve ser elaborada de acordo com a NBR 10520."
(Autor da epígrafe, ano)

RESUMO

Um fluxo de trabalho (*workflow*) especifica um processo envolvendo encadeamento de tarefas e fluxo de informações entre elas para alcançar um objetivo de negócio. A orquestração de um *workflow* automatiza e gerencia a execução de uma instância de processo de acordo com a especificação do *workflow*, usando um *software* de orquestração. O orquestrador invoca recursos humanos e/ou de TI apropriados para executar cada tarefa, coordenar a execução das mesmas de acordo com o encadeamento especificado e reportar eventuais problemas encontrados durante a execução. Este trabalho de conclusão de curso está no contexto do projeto de pesquisa CÉOS, executado pela Universidade Federal de Santa Catarina (UFSC), em parceria com o Ministério Público de Santa Catarina (MPSC). O projeto CÉOS tem o objetivo de desenvolver *workflows* para análise e processamento de dados em grande escala a fim de automatizar a extração de conhecimento e aumentar a eficiência das decisões tomadas em atividades do MPSC. Neste trabalho, será especificado e implementado um fluxo de trabalho para extração, transformação e carga (*extract, transform, load* - ETL) de dados referentes a licitações e compras feitas pela Prefeitura Municipal de Florianópolis. O fluxo será orquestrado pelo Airflow de acordo com grafos acíclicos direcionados (*directed acyclic graph* - DAG) especificados na plataforma. Os resultados serão avaliados usando métricas referentes à quantidade e qualidade dos dados, a frequência de erros e o tempo de execução dos processos de ETL.

Palavras-chave: Extração, transformação e carga de dados. Dados abertos. Licitações.

ABSTRACT

A workflow specifies a process involving the chaining of tasks and the flow of information between them in order to achieve a business objective. The orchestration of a workflow automates and manages the execution of a process instance according to the workflow specification, using orchestration software. The orchestrator invokes appropriate human and/or IT resources to execute each task, coordinates their execution according to the specified chaining, and reports any issues encountered during execution. This undergraduate thesis is part of the CÉOS research project, carried out by the Federal University of Santa Catarina (UFSC) in partnership with the Public Prosecutor's Office of Santa Catarina (MPSC). The CÉOS project aims to develop workflows for the analysis and processing of large-scale data in order to automate knowledge extraction and improve the efficiency of decision-making processes within MPSC activities. In this work, a workflow will be specified and implemented for extract, transform, and load (ETL) processes involving data related to bidding and procurement carried out by the Municipality of Florianópolis. The workflow will be orchestrated by Airflow according to directed acyclic graphs (DAGs) specified in the platform. The results will be evaluated using metrics related to data quantity and quality, error frequency, and ETL process execution time.

Keywords: Extraction, transformation and loading of data. Open data. Public procurement.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O uso de plataformas de orquestração de *workflows* é imprescindível para gerenciar aplicações com uso intenso de dados, como *data warehouses*. (HOLLINGSWORTH, 1995) define *workflow* como a automação informatizada total ou parcial de um processo de negócio e afirma que um sistema de gerenciamento de *workflows* deve fornecer suporte a três tipos de funções: (i) as funções *build-time*, referentes à definição e modelagem do processo de *workflow*; (ii) as funções *run-time control*, que envolvem o sequenciamento da execução das atividades e (iii) as interações de *run-time* entre os usuários e as aplicações de TI que processam os passos das atividades.

Workflows são frequentemente utilizados para implementar processos de extração, transformação e carga (em inglês *extraction-transformation-loading* - ETL) de dados. (VASSILIADIS; SIMITSIS; SKIADOPOULOS, 2002) fornecem uma descrição generalizada de processos ETL. Dados são obtidos de fontes como bancos de dados relacionais ou arquivos por rotinas de extração. Depois, os dados são propagados a uma ou mais *data staging areas*, onde são transformados a um formato homogêneo. Por fim, os dados formatados são inseridos em um banco de dados, muitas vezes no modelo multidimensional de um *data warehouse*.

O Apache Airflow é uma plataforma de orquestração de *workflows* de código aberto utilizada para implementar e manter *pipelines* de dados. No Airflow, *workflows* são definidos por grafos acíclicos direcionados, ou DAGs (do inglês *directed acyclic graph*). DAGs são *scripts* escritos na linguagem Python, utilizando construções básicas como operadores. DAGs definem as tarefas a serem executadas, as quais podem ser especificadas por operadores, e relações que indicam a ordem de execução e as dependências entre tarefas. Estes componentes formarão a *pipeline* do Airflow (FINNIGAN; TONER, 2021).

Este trabalho busca descrever, de forma detalhada, o desenvolvimento de um fluxo de ETL desde a obtenção de dados até o processamento e carga dos dados. Primeiramente, duas fontes serão acessadas: (i) um *web scraper* que acessa o Portal da Transparência da Controladoria-Geral da União e extrai dados referentes a contratos de compras feitas por órgãos do município de Florianópolis e (ii) um *script* que faz uma requisição do API de dados públicos do Portal de Transparência da Prefeitura Municipal de Florianópolis buscando registros de contratos e licitações, incluindo número, nome e CPF/CNPJ do fornecedor, período de vigência e valor monetário. Depois, a orquestração dos fluxos de trabalho será implementada por meio da execução de processos de acordo com especificações desses fluxos na forma de DAGs no Airflow. Esses processos visam limpar e padronizar os dados em um formato adequado para o armazenamento. A análise exploratória e a visualização dos dados serão feitas com a geração de tabelas, gráficos e eventualmente grafos, que permitirão analisar a estrutura e a distribuição dos dados, tais como os custos dos contratos para identificar irregularidades. O desempenho do fluxo de ETL será avaliado

com base nos tempos de execução das tarefas dos DAGs, na quantidade de dados tratados e na quantidade de erros encontrados durante e após a transformação.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é orquestrar um fluxo de extração, transformação e carregamento de dados relacionados a compras feitas por órgãos públicos do município de Florianópolis, com o intuito de suprir dados para analisar os gastos com as compras no projeto Céos. Os resultados serão avaliados através de uma análise exploratória de dados resultantes em tabelas e gráficos, que permitirão visualizar suas distribuições. Também serão medidos os tempos de execução das tarefas e as quantidades de erros encontrados nos dados, tratados ou não dentro nos processos de ETL implementados e avaliados.

1.1.2 Objetivos Específicos

Os objetivos específicos são:

1. Estudar conceitos de orquestração de fluxos de ETL.
2. Desenvolver códigos para obtenção de dados utilizando métodos de web scraping e requisições em uma API pública.
3. Construir DAGs (directed acyclic graphs) para implementar o fluxo de ETL, que padronizará os dados em um formato adequado.
4. Orquestrar processos de ETL usando as DAGs construídas, coletando métricas para avaliar sua capacidade, desempenho e custos computacionais.
5. Utilizar ferramentas de análise e visualização de dados para gerar tabelas e gráficos sobre as distribuições dos dados carregados e a análise dos custos computacionais dos processos de ETL.

1.2 METODOLOGIA

A metodologia adotada neste trabalho é estruturada em três etapas principais, contemplando desde o estudo preliminar até a implementação e análise dos dados extraídos.

A primeira etapa consiste em um levantamento teórico sobre os conceitos, ferramentas e práticas envolvidas na orquestração de *pipelines* de ETL (*extraction-transformation-loading*) utilizando o Apache Airflow. Foram reunidos artigos científicos, publicações técnicas e documentação especializada sobre a estrutura e implementação de DAGs (*directed acyclic graphs*) no Airflow, bem como sobre métodos de *web scraping*, a fim de possibilitar a extração de dados não estruturados diretamente de páginas da Web.

Na segunda etapa, foi realizada a implementação dos *scripts* responsáveis pela extração de dados por meio de *web scraping* e requisições a APIs REST. Os dados obtidos a partir dos *endpoints* públicos disponibilizados no Portal de Transparência da Prefeitura estavam incompletos, o que tornou necessário consultar os *endpoints* internos do sistema para extrair os dados faltantes. Após este processo, realizou-se uma análise exploratória dos dados, consistindo na geração de gráficos para visualização de possíveis *outliers* entre os registros e na construção de um dicionário de dados e, subsequentemente, a modelagem de um esquema para bancos de dados relacionais.

A terceira e última etapa envolve a orquestração completa do fluxo ETL no ambiente do Airflow, seguida da análise dos dados carregados. Foram realizados testes de execução e ajustes nos fluxos com base na avaliação de métricas, como tempo de execução e ocorrência de erros, e a partir dos dados transformados, foram geradas tabelas e gráficos mais detalhados, permitindo a análise mais complexa.

1.3 ORGANIZAÇÃO DO DOCUMENTO

O trabalho está organizado da seguinte maneira: o capítulo 2 apresenta os fundamentos de conceitos e tecnologias utilizados para desenvolver o trabalho. No capítulo 3, faz-se uma comparação de trabalhos que abordam temas semelhantes a este. O capítulo 4 descreve os processos de análise exploratória dos dados e implementação do fluxo ETL. No capítulo 5, são discutidos os resultados obtidos a partir da análise. Finalmente, o capítulo 6 apresenta conclusões e temas para trabalhos futuros.

2 FUNDAMENTOS

Neste capítulo são apresentados conceitos teóricos relacionados ao trabalho. Na seção 2.1, apresenta-se o processo de extração, transformação e carga de dados com as definições de cada etapa do processo. A seção 2.2 apresenta a plataforma de software Apache Airflow, utilizada para orquestrar fluxos de trabalho, além de seus componentes e sua interface gráfica. A seção 2.3 apresenta o conceito de dados abertos e iniciativas de transparência governamental em nível federal e municipal. Por fim, na seção 2.4, são definidos licitações e contratos administrativos de acordo com a Lei 14.133/2021.

2.1 EXTRAÇÃO, TRANSFORMAÇÃO E CARGA (ETL) DE DADOS

Processos de extração, transformação e carga de dados são responsáveis pela coleta, extraçao, limpeza, transformação integração e carga de dados de uma ou mais fontes para um sistema destino, tal como um *data warehouse* . De acordo com (VASSILIADIS; SIMITSIS, 2009), qualquer programa que filtra registros, calcula novos valores e alimenta uma fonte de dados diferente da original pode ser considerado um programa de ETL.

(VASSILIADIS; SIMITSIS, 2009) descrevem o fluxo de trabalho de um processo ETL como um grafo acíclico direcionado, no qual as tarefas executadas são análogas aos nodos do grafo e as relações entre entradas e saídas são análogas às arestas do grafo.

A primeira etapa do processo ETL é a extração. Nesta, dados são obtidos de fontes, que podem estar no formato de documentos de texto, planilhas, páginas da web ou streaming. Geralmente, busca-se extrair apenas os dados que foram inseridos ou atualizados após a última execução do processo.

A segunda etapa é a transformação. Esta etapa envolve a normalização, formatação, padronização e enriquecimento de dados visando resolver problemas em nível de esquema (conflitos de nomeação de objetos ou diferenças na estruturação de dados entre fontes e o *data warehouse*) e em nível de registro (registros duplicados, contraditórios, ou com referências temporais diferentes).

A terceira e última etapa é a carga. Os dados transformados são carregados nas tabelas apropriadas do *data warehouse*, onde servirão como base para processos de análise, visualização e apoio a tomadas de decisões.

2.2 APACHE AIRFLOW

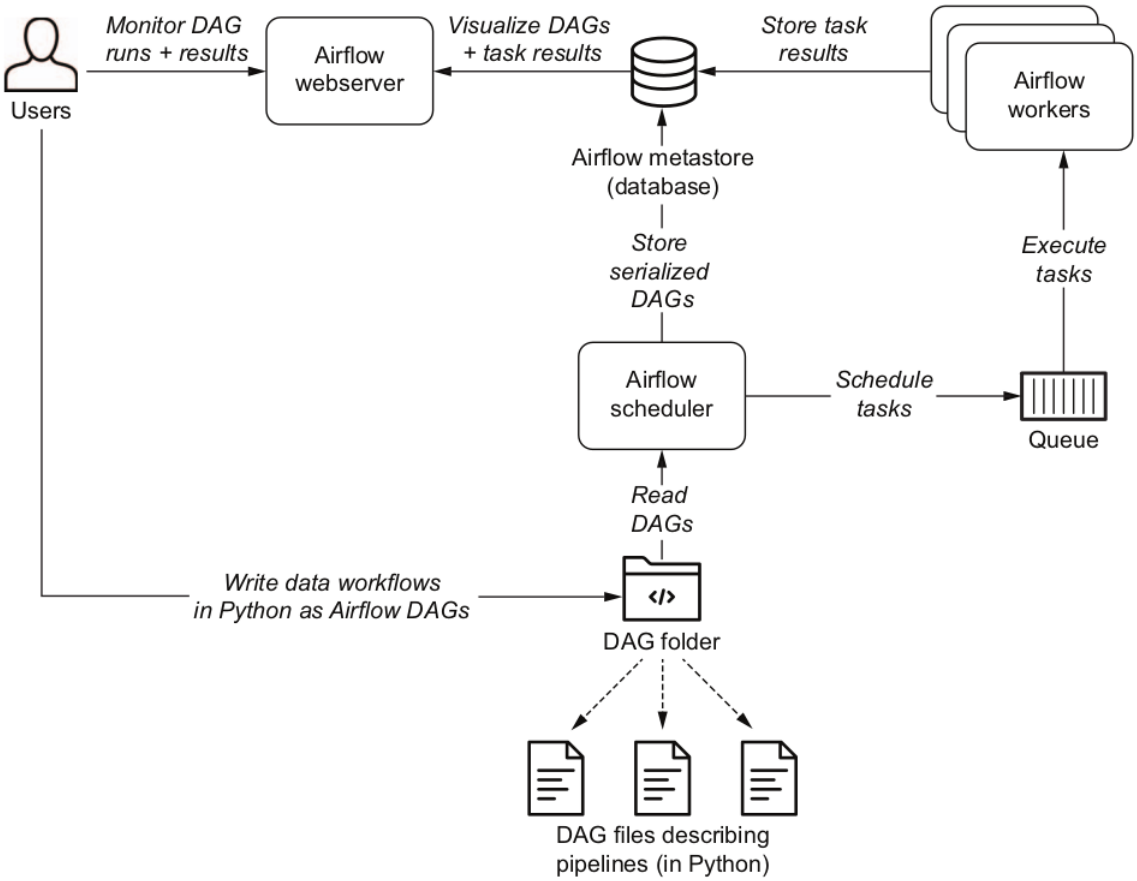
O Apache Airflow é uma plataforma de código aberto utilizada para orquestrar e monitorar fluxos de trabalho no formato de grafos acíclicos direcionados, ou DAGs. No Airflow, os DAGs são definidos por meio de arquivos, ou *scripts* na linguagem Python que descrevem as tarefas a serem executadas e as dependências entre as tarefas, além de metadados com configurações específicas. O Airflow interpreta estes *scripts* e modela a

estrutura das DAGs, executando as tarefas de acordo com o intervalo de tempo agendado nos metadados.

De acordo com (RUITER; HARENSLAK, 2021), o uso de código em Python para definir DAGs garante flexibilidade ao desenvolvedor, pois qualquer função implementada em Python pode ser executada pelo Airflow. Assim, é possível importar funções de bibliotecas externas e criar fluxos de trabalho altamente escaláveis que estabelecem comunicações entre diversos serviços.

(RUITER; HARENSLAK, 2021) afirmam que o Airflow pode ser dividido em três componentes: (i) o escalonador, que lê os *scripts* de DAGs desenvolvidos e obtém as tarefas, dependências e intervalos de execução, verifica se o intervalo de cada DAG passou e, se este for o caso, adiciona as respectivas tarefas à fila de execução, (ii) os trabalhadores, que obtêm tarefas escalonadas da fila e as executam em paralelo e (iii) o servidor *web*, que fornece uma interface com representações visuais das DAGs e permite que o usuário monitore os resultados das tarefas executadas.

Figura 1 – Diagrama mostrando os componentes do Airflow.



Fonte: (RUITER; HARENSLAK, 2021)

A interface *web* do Airflow oferece diferentes modos de visualização das DAGs. O modo *graph view* (visualização de grafo) mostra um esquema contendo as tarefas e as respectivas dependências, enquanto o modo *tree view* (visualização de árvore) mostra o histórico dos resultados das execuções das DAGs, também permitindo verificar detalhes de tarefas específicas, com indicadores de status e *logs* de textos com informações referentes a possíveis erros de execução.

2.3 TRANSPARÊNCIA E DADOS ABERTOS

(POSSAMAI; SOUZA, 2020) afirmam que dados abertos governamentais são dados públicos produzidos ou curados por órgãos estatais, padronizados em formatos abertos e legíveis por máquina, como JSON ou XML, e que podem ser livremente acessados e utilizados para quaisquer finalidades. A disponibilidade de dados abertos governamentais é importante para garantir a transparência com gastos públicos, incentivar a colaboração entre o cidadão e a Administração Pública e combater a corrupção.

Segundo (POSSAMAI; SOUZA, 2020), a Lei 12.527/2011 regula o acesso às informações contidas em documentos de órgãos públicos e estabelece duas formas de transparência a serem implementadas: (i) a transparência passiva, que envolve o atendimento a solicitações de informação por meio físico ou eletrônico, e (ii) a transferência ativa, referente à divulgação de um rol mínimo de informações por parte dos órgãos independentemente de requerimento. A Lei 12.527/2011 faz referência explícita aos princípios dos dados abertos em seu texto.

De acordo com (NAZÁRIO; SILVA; ROVER, 2012), um dos principais projetos de visualização de dados abertos governamentais é o Portal de Transparência do Governo Federal, inicialmente lançado em novembro de 2004. Em concordância com a Lei Complementar 131/09, o Portal disponibiliza informações referentes a receitas, transferências e convênios recebidos, assim como despesas envolvendo processos licitatórios, contratos e gastos com diárias.

Em relação à cidade de Florianópolis, (SANTOS, M. T. S. *et al.*, 2021) afirmam que a Prefeitura Municipal regulamentou o acesso à informação pública por meio do decreto 9988/12, posteriormente instituindo o Portal de Transparência de Florianópolis por meio da Lei Municipal 9.447/14. Analogamente ao Portal de Transparência do Governo Federal, o usuário possui acesso a receitas e despesas da Prefeitura e de seus órgãos.

2.3.1 Licitações e contratos administrativos

(TCU, 2023) define licitação como "o processo por meio do qual a Administração Pública convoca, sob condições estabelecidas em ato próprio (edital de licitação), interessados para apresentação de propostas relativas ao fornecimento de bens, prestação de serviços ou execução de obras". O processo licitatório é regulamentado pela Lei

14.133/2021, cujo artigo 17 o divide em sete fases: (i) fase preparatória, (ii) divulgação do edital, (iii) apresentação de propostas e lances, (iv) julgamento, (v) habilitação, (vi) recursal e (vii) homologação.

O artigo 11 da Lei 14.133/2021 aponta como objetivos (i) garantir que seja selecionada a proposta apta a gerar o resultado de contratação mais vantajoso à Administração Pública, (ii) assegurar o tratamento isonômico e a justa competição entre os licitantes, (iii) evitar sobrepreço nas contratações e superfaturamento na execução dos contratos e (iv) incentivar a inovação e a sustentabilidade no desenvolvimento nacional.

Segundo a Lei 14.133/2021, existem cinco modalidades de licitação:

1. Pregão: aquisição de bens e serviços comuns definidos pelo edital, seguindo critérios de julgamento de menor preço ou maior desconto.

2. Concorrência: aquisição de bens e serviços especiais e de obras e serviços comuns e especiais de engenharia, utilizando critérios de julgamento de menor preço, maior desconto, melhor técnica ou conteúdo artístico, técnica e preço ou maior retorno econômico.

3. Concurso: escolha de trabalho técnico, científico ou artístico, utilizando o critério de julgamento de melhor técnica ou conteúdo artístico e concedendo prêmio ou remuneração ao vencedor.

4. Leilão: alienação de bens imóveis ou de bens móveis inservíveis ou legalmente apreendidos, seguindo o critério de julgamento de maior lance.

5. Diálogo competitivo: contratação de obras, serviços e compras por meio de diálogos entre a Administração Pública e licitantes previamente selecionados, visando uma solução que atenda a uma necessidade definida.

De acordo com (TCU, 2023), contratos administrativos são “aqueles firmados entre órgãos ou entidades da Administração Pública e particulares, por meio do qual se estabelece acordo de vontades, para a formação de vínculo e a estipulação de obrigações recíprocas”, que têm como objetivo principal atender a um interesse coletivo.

2.4 DADOS ABERTOS DE COMPRAS PÚBLICAS DA PMF

Esta seção é dividida em duas subseções. A subseção 2.4.1 descreve a API de compras públicas disponível no *website* do Portal de Transparência da Prefeitura de Florianópolis, o método de consulta de dados e os parâmetros de pesquisa. A subseção 2.4.2 apresenta um dicionário de dados, no qual os conteúdos das respostas da API são esquematizados em tabelas.

2.4.1 API de compras públicas da PMF

O Portal de Transparência da Prefeitura Municipal de Florianópolis disponibiliza informações relacionadas a compras públicas efetuadas pela Prefeitura e as unidades gestoras que a compõem, em concordância com o princípio de dados abertos. O *website* do Portal, além de mostrar listas com registros de compras, oferece uma API REST que permite ao usuário fazer requisições destes registros utilizando o formato JSON.

A API possui diversos *endpoints*, dos quais são utilizados os de consulta de licitações[1] e de contratos administrativos[2]. O acesso a estes *endpoints* é livre, sem necessidade de chave para autenticação.

O único método fornecido para ambos os *endpoints* é o GET, que retorna dados de licitações ou contratos em formato JSON. Em ambos os *endpoints*, é possível filtrar os resultados pelas datas inicial e final do período de busca, o código da unidade gestora responsável e o limite máximo de registros a serem retornados. Apenas as datas inicial e final são parâmetros obrigatórios.

2.4.2 Descrição dos dados retornados

A estrutura dos registros retornados difere de acordo com os *endpoints*. A Figura 2 apresenta a estrutura dos registros retornados pelo *endpoint* Licitação. Estes registros contêm dados como o valor da licitação, datas de emissão e abertura, modalidade do processo licitatório, advogado responsável, lista de itens negociados e vencedores do processo, lista de *links* a documentos relacionados, lista de empenhos e lista de contratos.

A Figura 3 mostra a estrutura dos registros retornados pelo *endpoint* Contrato. Nestes registros, encontra-se dados como as datas de assinatura e início de vigência, valor total do contrato, nome do fornecedor, lista de itens e lista de empenhos.

O dicionário de dados a seguir foi desenvolvido a partir do conteúdo retornado pelo método GET da API. Além do cabeçalho do arquivo JSON, que contém metadados, são retornados elementos de dados JSON para as entidades `Licitação` e `Contrato administrativo`, bem como as entidades auxiliares, que representam objetos aninhados nos registros.

O *Website* do Portal de Transparência da Prefeitura de Florianópolis também disponibiliza *endpoints* para a consulta de receitas, despesas orçamentárias, execuções de despesa e dados da gestão de pessoal; estes não foram utilizados no trabalho, devido aos dados possuírem pouca relação ao contexto dos processos licitatórios.

Os arquivos JSON seguem padrões semelhantes; ambos os *endpoints* consultados no trabalho retornam arquivos com as seguintes propriedades:

[1] Disponível em: `https://transparencia.e-publica.net/epublica-portal/#/florianopolis/portal/dadosAbertos/licitacaoView`

[2] Disponível em: `https://transparencia.e-publica.net/epublica-portal/#/florianopolis/portal/dadosAbertos/contratoView`

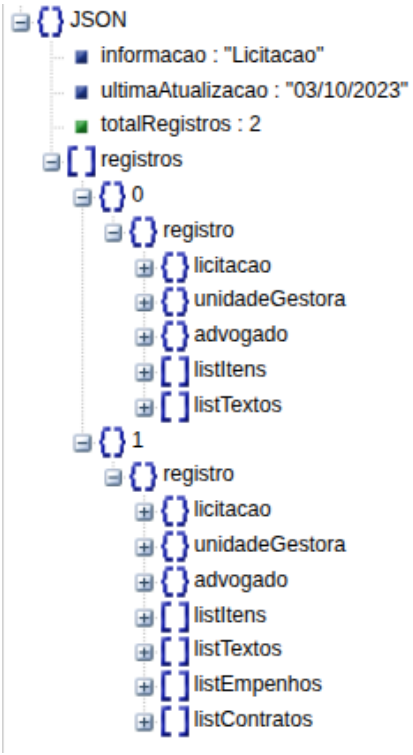


Figura 2 – Visão em árvore do JSON de licitações.

Cabeçalho do JSON

Nome do Campo	Tipo de Dado	Tamanho	Descrição
informacao	Texto	50	Tipo de informação retornada pela API (`'Licitacao'`, `'Contrato'`, `'Receita'`, `'Despesa Orçamentária'`, `'Execução de Despesa'`).
ultimaAtualizacao	Data	-	Data da última atualização dos dados no formato "dd/MM/yyyy".
totalRegistros	Inteiro	-	Número total de registros retornados na consulta.
registros	Lista	-	Lista de registros (licitações ou contratos).

Licitação

Nome do Campo	Tipo de Dado	Tamanho	Descrição
numero	Texto	15	Número identificador da licitação.

modalidade	Texto	50	Modalidade da licitação (exemplo: Pregão", Tomada de Preço").
valorEstimado	Decimal	10,2	Valor estimado da licitação.
objetoResumido	Texto	500	Descrição resumida do objeto da licitação.
dataEmissao	Data	-	Data de emissão da licitação no formato "yyyy-MM-dd".
aberturaData	Data	-	Data de abertura da licitação.
finalidade	Texto	100	Finalidade da licitação (exemplo: 'Compras e Outros Serviços', 'Obras e Serviços de Engenharia').
formaJulgamento	Texto	50	Forma de julgamento da licitação (exemplo: 'Por item').
unidadeGestora	Objeto	-	Órgão responsável pela licitação.
advogado	Objeto	-	Representante jurídico vinculado ao processo licitatório.
listItens	Objeto	-	Lista de itens licitados.
listTextos	Objeto	-	Lista de documentos e arquivos complementares.
listEmpenhos (opcional)	Objeto	-	Lista de empenhos associados à licitação.
listContratos (opcional)	Objeto	-	Lista de contratos associados à licitação.

Unidade gestora

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
codigo	Inteiro	-	Código identificador da unidade gestora.
denominacao	Texto	100	Nome da unidade gestora responsável pela licitação ou contrato.

Advogado

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
pessoa.nome	Texto	100	Nome do advogado responsável pelo processo licitatório.

Item da licitação

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
numero	Texto	20	Número do item.
denominacao	Texto	100	Descrição do item.
quantidade	Decimal	-	Quantidade de unidades do item.
unidadeMedida	Texto	20	Unidade de medida do item (exemplo: 'Serviço' ou 'Unidade').
valorUnitarioEstimado	Decimal	10,2	Valor monetário de uma unidade do item.
situacao	Texto	20	Situação do item (exemplo: 'Homologado').
listVencedores	Objeto	-	Lista de vencedores do processo licitatório.

Vencedor

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
fornecedor	Texto	100	Nome da pessoa jurídica fornecedora do item.
quantidade	Decimal	-	Quantidade de unidades do item fornecido.
valorUnitario	Decimal	10,2	Valor monetário de uma unidade do item.
situacao	Texto	20	Situação da pessoa jurídica (exemplo: 'Vencedor').

Texto

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
denominacao	Texto	200	Título do documento, contendo números de identificação.
link	Texto	100	Endereço URL da página do documento.

Empenho

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**

emissao	Data	-	Data de emissão do empenho.
numero	Texto	20	Número do empenho.
objetoResumido	Texto	255	Descrição resumida do empenho.
especie	Texto	20	Tipo do empenho (exemplo: 'Ordinário', 'Estimativo' ou 'Global').
categoria	Texto	20	Categoria do empenho (exemplo: 'Comum', 'Subvenção', 'Auxílio' etc.).
contrato	Texto	20	Número do contrato relacionado ao empenho.
licitacao	Texto	20	Número da licitação relacionada ao empenho.
recursoDiaria	Texto	20	Informação sobre recurso de diária (se aplicável).

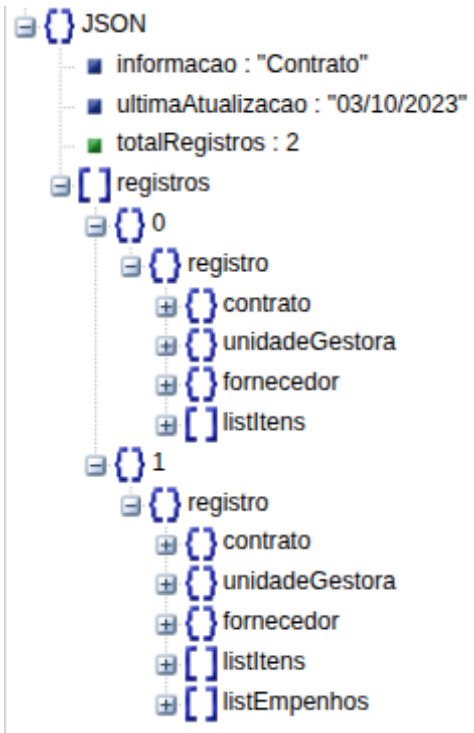


Figura 3 – Visão em árvore do JSON de contratos.

No arquivo dos contratos, como mostra a figura 3, o registro contém as seguintes propriedades:

Contrato

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
numero	Texto	20	Número do contrato.
assinatura	Data	-	Data da assinatura do contrato.
inicioVigencia	Data	-	Data de início da vigência do contrato.
vencimento	Data	-	Data de vencimento do contrato.
valorTotal	Decimal	10,2	Valor total do contrato.
objetoResumido	Texto	255	Descrição resumida do contrato.
unidadeGestora	Objeto	-	Órgão responsável pelo contrato.
fornecedor	Objeto	-	Fornecedor contratado.
listItens	Objeto	-	Lista de itens do contrato.
listEmpenhos (opcional)	Objeto	-	Lista de empenhos associados ao contrato.

Item do contrato

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
numero	Texto	20	Número do item.
denominacao	Texto	100	Descrição do item.
quantidade	Decimal	-	Quantidade de unidades do item.
unidadeMedida	Texto	20	Unidade de medida do item (exemplo: 'Serviço' ou 'Unidade').
valorUnitario	Texto	10	Valor monetário de uma unidade do item.
valorTotal	Texto	10	Valor monetário de todas as unidades do item.

Fornecedor

Nome do Campo	**Tipo de Dado**	**Tamanho**	**Descrição**
pessoa.nome	Texto	100	Nome do fornecedor contratado.

3 TRABALHOS RELACIONADOS

A análise de dados abertos governamentais é um tema cada vez mais pertinente na pesquisa acadêmica, especialmente considerando a disponibilidade de algoritmos de aprendizado de máquina e a capacidade de processar grandes quantidades de dados. Nesta seção, são apresentados trabalhos que envolvem a análise de dados de licitações e contratos administrativos e buscam facilitar o combate à corrupção nos processos licitatórios.

A busca bibliográfica foi feita pela plataforma Google Scholar utilizando as palavras-chave "ETL", "extração transformação e carga", "licitações", "portal de transparência"e "dados abertos". Também foram obtidos trabalhos a partir dos arquivos do Simpósio Brasileiro de Bancos de Dados de 2024 (SBBD).

(SCHMITZ *et al.*, 2024) propõem uma metodologia de aprendizado de máquina que utiliza modelos de mistura gaussiana (GMM) para identificar padrões de lances suspeitos de fraude em processos licitatórios, considerando que a escassez de casos confirmados de fraude torna o uso de métodos de aprendizado supervisionado inviável. O modelo avalia a similaridade entre casos de licitações possivelmente fraudulentas e casos conhecidos de fraude em diversos subespaços, criando um *ranking* eficaz de indicadores de risco.

(SANTOS, E. S. dos *et al.*, 2024) utilizam um conjunto de dados referentes a licitações investigadas pela Operação Lava Jato para avaliar as capacidades de diferentes modelos de aprendizado de máquina em detectar casos de conluio. O trabalho propõe uma metodologia com extração de dados para enriquecimento do conjunto original e técnicas como validação cruzada e otimização de hiperparâmetros para treinar os modelos, alcançando resultados mais precisos.

(SANTOS, M. T. S. *et al.*, 2021) apresentam uma ferramenta para visualizar dados obtidos do Portal de Transparência da Câmara Municipal de Florianópolis; o trabalho foca especificamente nos balancetes de vereadores. Os dados dos documentos em PDF são extraídos e coletados em arquivos CSV, que são lidos por uma aplicação escrita na linguagem PHP e adicionados a um banco de dados. Uma interface gráfica permite pesquisar o nome de um vereador e um ano, gerando gráficos dos gastos mensais durante este ano e comparando com gastos de outros anos.

(JESUS, 2021) apresenta um modelo preditivo que utiliza técnicas de mineração de dados e aprendizado de máquina para indicar riscos de irregularidades nas fases de divulgação do edital e apresentação de propostas do processo licitatório. O modelo analisa licitações do Estado de Goiás realizadas entre 2014 e 2019 e calcula o risco utilizando algoritmos de treinamento supervisionado com classificadores. Por fim, constrói-se um *ranking* de licitações potencialmente irregulares.

(MELLO *et al.*, 2024) propõem uma modelagem conceitual abrangente de dados para o domínio de licitações, também incluindo dados de fraudes associadas a licitações, denúncias e pessoas envolvidas no processo licitatório, com o objetivo de tornar o projeto

de um banco de dados mais robusto e fornecer apoio a sistemas de detecção de fraude. A modelagem inclui uma proposta de persistência poliglota, ou seja, o uso de diversos modelos de banco de dados, como relacional, orientado a grafos e orientado a documentos.

Tabela 12 – Comparação de trabalhos relacionados (parte 1)

Trabalho	**Metodologia**
(SCHMITZ *et al.*, 2024)	Aprendizado de máquina com modelos de mistura gaussiana
(SANTOS, E. S. dos *et al.*, 2024)	ETL com validação cruzada e otimização de hiperparâmetros
(SANTOS, M. T. S. *et al.*, 2021)	Extração de arquivos CSV e interface gráfica
(JESUS, 2021)	Mineração de dados e treinamento supervisionado
(MELLO *et al.*, 2024)	Modelagem conceitual com persistência poliglota

Fonte: Elaborada pelo autor.

Tabela 13 – Comparação de trabalhos relacionados (parte 2)

Trabalho	**Conjunto de dados**
(SCHMITZ *et al.*, 2024)	Licitações referentes à Operação Patrola
(SANTOS, E. S. dos *et al.*, 2024)	Licitações referentes à Operação Lava Jato
(SANTOS, M. T. S. *et al.*, 2021)	Balancetes de vereadores de Florianópolis
(JESUS, 2021)	Licitações do Estado de Goiás entre 2014 e 2019
(MELLO *et al.*, 2024)	-

Fonte: Elaborada pelo autor.

4 DESENVOLVIMENTO

Este capítulo descreve o que foi desenvolvido no trabalho de conclusão de curso, incluindo codificação do cliente da API, análise exploratória de dados coletados, definição do esquema do banco de dados destino implementação do processo de ETL. Todo o código, dados e resultados obtidos neste trabalho estão disponíveis via GitHub[1].

4.1 CLIENTE PARA A API DE COMPRAS PÚBLICAS DA PMF

Ao consultar a API da prefeitura por meio dos *endpoints* descritos na Seção 2, observou-se que os registros retornados apresentavam informações incompletas. Especificamente, as licitações não possuíam contratos vinculados, e tanto licitações quanto contratos não exibiam os empenhos associados. Este problema indicou que os *endpoints* públicos disponibilizados pela API forneciam apenas dados resumidos, insuficientes para a análise completa pretendida.

Para contornar essa limitação, foi necessário desenvolver um processo alternativo de extração de dados, baseado nos *endpoints* utilizados internamente pela aplicação Web do portal para alimentar as tabelas de listagem de licitações[2] e contratos[3]. Esses *endpoints*, embora não mencionados e descritos na documentação oficial da API, são responsáveis por fornecer as informações completas que populam dinamicamente as páginas detalhadas de cada licitação ou contrato.

O processo teve início com a execução de uma requisição HTTP do tipo POST para o *endpoint* que alimenta a tabela de listagem de licitações no portal. O objetivo dessa etapa foi obter os identificadores internos (*IDs*) utilizados pelo sistema para carregar as páginas individuais de cada compra pública. A ferramenta Postman foi empregada tanto para realizar testes nas requisições, como alterar parâmetros de intervalo de tempo e quantidade de registros a serem retornados em uma busca, quanto para capturar os dados e organizá-los em arquivos no formato JSON, servindo como base para as etapas subsequentes. O mesmo processo foi aplicado ao *endpoint* da tabela de listagem dos contratos.

Na sequência, foi desenvolvido um *script* na linguagem Python, responsável por automatizar o processo de extração. Esse *script* utiliza os *IDs* coletados como parâmetros em requisições POST direcionadas aos *endpoints* responsáveis por retornar os dados completos das licitações[4] e dos contratos[5]. Os dados extraídos incluem, além das informações

[1] `https://github.com/gustavodesalles/INE5454-trabalho`

[2] Disponível em: `https://transparencia.e-publica.net/epublica-portal/rest/florianopolis/compras/licitacao/listAll`

[3] Disponível em: `https://transparencia.e-publica.net/epublica-portal/rest/florianopolis/compras/contrato/listAll`

[4] Disponível em: `https://transparencia.e-publica.net/epublica-portal/rest/florianopolis/compras/licitacao/form`

[5] Disponível em: `https://transparencia.e-publica.net/epublica-portal/rest/florianopolis/`

da própria licitação, seus contratos vinculados e os respectivos empenhos. Todos os dados obtidos foram armazenados em arquivos JSON disponibilizados no GitHub[6].

4.2 ANÁLISE EXPLORATÓRIA DOS DADOS COLETADOS EM JSON

Esta etapa visa realizar uma análise exploratória do conjunto de dados obtidos anteriormente com o objetivo de compreender a distribuição dos dados, identificar padrões, avaliar sua consistência e detectar possíveis inconsistências ou lacunas. Para fazer isso, criou-se *scripts* na linguagem Python, utilizando as bibliotecas Pandas e Matplotlib para estruturar os dados e gerar gráficos com base nos atributos dos registros.

Foi analisada a distribuição temporal das datas de assinatura de licitações e contratos ao longo do período abrangido pelos dados coletados. Foram gerados gráficos de séries temporais que representam a quantidade de licitações e contratos registrados por dia. Essa análise permite observar padrões sazonais, concentrações em determinados períodos e eventuais flutuações no volume de registros.

Com o intuito de compreender a composição dos dados, foram geradas distribuições das unidades gestoras vinculadas às licitações presentes no conjunto. Além disso, foi realizada uma análise estatística dos valores estimados das licitações, utilizando gráficos do tipo *boxplot* para identificar a dispersão, a mediana e possíveis valores discrepantes (*outliers*). Esta análise é fundamental para entender a variabilidade dos valores envolvidos nas compras públicas, além de fornecer indícios sobre padrões ou anomalias financeiras.

Outra etapa da análise consistiu na verificação da consistência dos dados financeiros entre os valores das licitações, contratos e respectivos empenhos. Foram comparados os valores dos contratos com os somatórios dos empenhos vinculados, a fim de identificar eventuais divergências e avaliar se os dados refletem corretamente os compromissos financeiros firmados pela administração pública.

Também foi realizada uma análise de integridade e qualidade dos dados, com o levantamento de campos nulos, ausentes ou inconsistentes. Foram identificados registros com ausência de informações relevantes, como dados de fornecedores, valores estimados, datas de assinatura dos contratos ou unidades gestoras. A identificação dessas irregularidades nos dados é fundamental tanto para a correta interpretação dos resultados quanto para o desenvolvimento de eventuais etapas de pré-processamento e limpeza dos dados.

Com os resultados desta análise exploratória, obtém-se uma visão geral do comportamento dos dados e identifica-se limitações a serem consideradas nas etapas subsequentes do trabalho. Além disso, possíveis problemas de completude ou inconsistência nos dados podem indicar falhas nos sistemas de origem ou na própria disponibilização pública das informações.

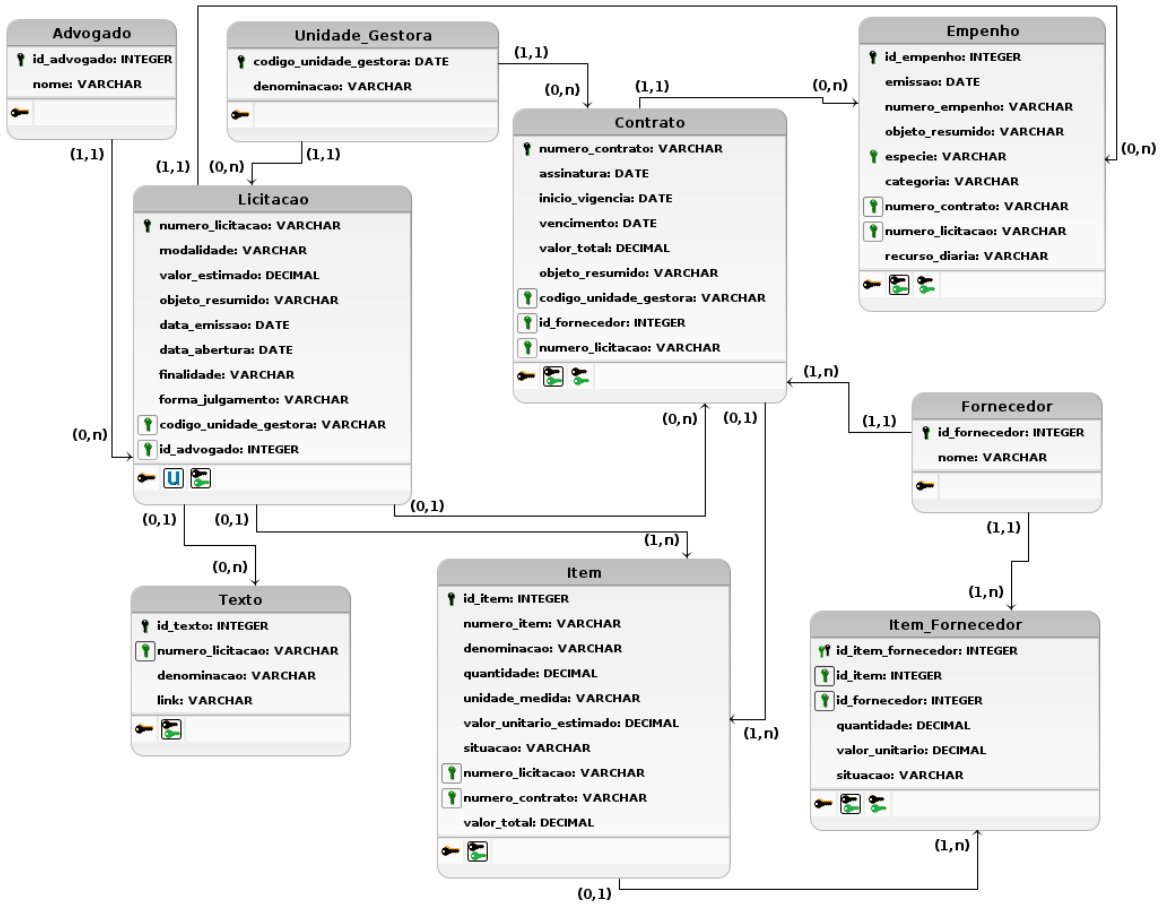
`compras/contrato/form`

[6] `https://github.com/gustavodesalles/licitacoes-contratos-fln`

4.3 ESQUEMA RELACIONAL PARA OS DADOS COLETADOS

Nesta seção, é apresentado um esquema de dados modelado com base na API de compras públicas da Prefeitura Municipal de Florianópolis e no dicionário de dados descritos na seção 2.4. O esquema visa representar a estrutura lógica dos dados de maneira clara e concisa para fins de análise. Um diagrama entidade-relacionamento é apresentado para mostrar a definição das tabelas em formato relacional e suas respectivas conexões.

Figura 4 – Diagrama entidade-relacionamento do esquema.



Fonte: Elaborada pelo autor.

Além das tabelas definidas no dicionário de dados, foi criada uma tabela `ItemFornecedor` para modelar o relacionamento entre as tabelas `Item` e `Fornecedor`, considerando que um item de licitação ou contrato pode possuir mais de um fornecedor.

4.4 PROCESSO DE ETL

Esta seção apresenta o planejamento e implementação do processo de extração, transformação e carga dos dados obtidos nas seções anteriores. As etapas do processo

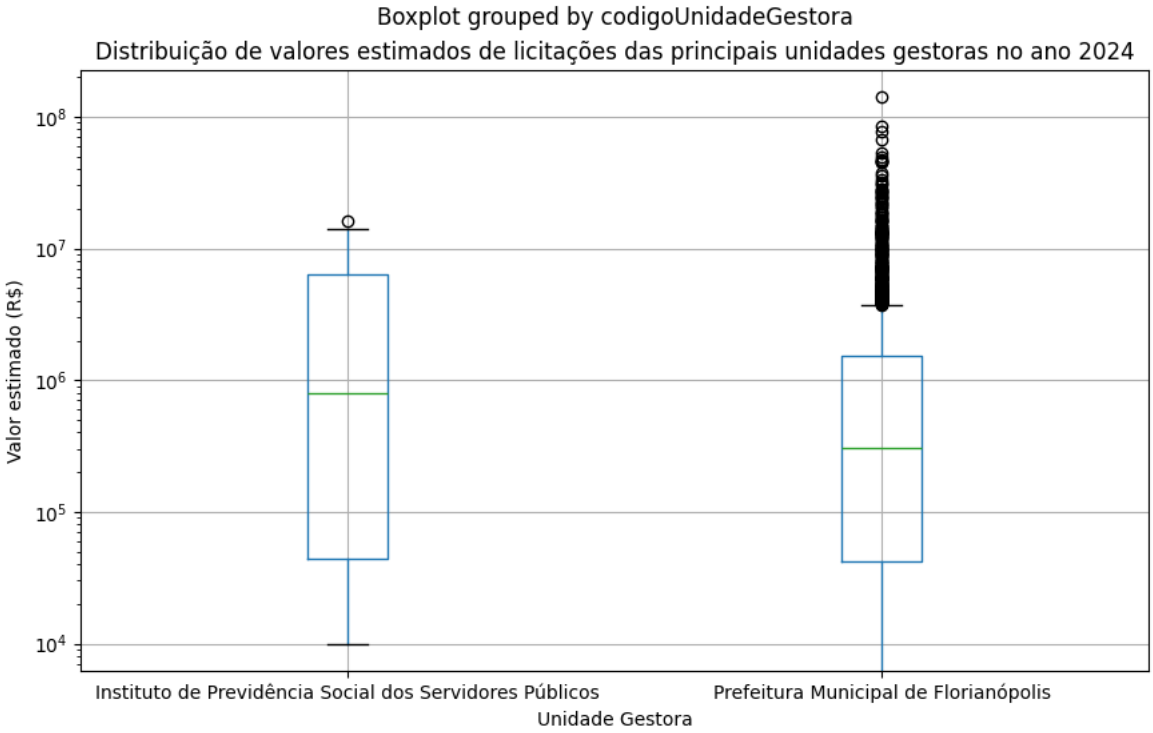
de ETL serão determinadas após a conclusão da modelagem do esquema relacional dos dados.

5 RESULTADOS

Nesta seção, apresenta-se os resultados obtidos a partir da análise exploratória. Foram utilizados os registros de licitações e contratos entre os anos de 2014 e 2024 para gerar gráficos e identificar irregularidades.

Em geral, nota-se uma grande quantidade de registros de licitações cuja unidade gestora é a Prefeitura Municipal de Florianópolis (código 1) ou o Instituto de Previdência Social dos Servidores Públicos (código 22) que não possuem valores estimados informados. Isto resulta em um *boxplot* com muitos *outliers*, como exemplificado pela Figura 5.

Figura 5 – Gráfico *boxplot* de licitações das unidades 1 e 22 no ano de 2024.



Fonte: Elaborada pelo autor.

Em relação aos contratos, verifica-se uma maior diversidade de unidades gestoras envolvidas, quando comparada às licitações. Essa diferença ocorre mesmo considerando que uma licitação pode estar associada a múltiplas unidades gestoras, enquanto um contrato está vinculado a apenas uma. Além disso, constatou-se uma quantidade significativamente menor de contratos sem o valor total informado, o que denota maior completude dos dados nessa categoria.

Entre todas as licitações analisadas, apenas quatro registros não apresentam unidade gestora informada, sendo todos referentes ao ano de 2024. No caso dos contratos, todos os registros contêm a informação da unidade gestora responsável.

6 CONCLUSÕES E TRABALHOS FUTUROS

As APIs da Prefeitura Municipal de Florianópolis oferecem dados incompletos entre si, sendo necessário realizar requisições em diversos *endpoints* para obter informações completas. Para trabalhos futuros, é sugerido o desenvolvimento de uma única API que reúna todos os conteúdos em um arquivo.

REFERÊNCIAS

CONTAS DA UNIÃO, Tribunal de. **Licitações & Contratos: Orientações e Jurisprudência do TCU / Tribunal de Contas da União**. 5. ed. Brasília: Secretaria-Geral da Presidência, 2023.

FINNIGAN, Leanne; TONER, Emily. Building and maintaining metadata aggregation workflows using apache airflow. **Temple University Libraries**, 2021.

HOLLINGSWORTH, David. Workflow management coalition: The workflow reference model, 1995.

JESUS, Mauricio Barros de. Modelo preditivo de risco de irregularidades em compras públicas no Estado de Goiás, 2021.

MELLO, Ronaldo *et al.* Uma Proposta de Modelagem de Dados no Domínio de Fraudes em Licitações Públicas. *In:* ANAIS Estendidos do XXXIX Simpósio Brasileiro de Bancos de Dados. Florianópolis/SC: SBC, 2024. P. 220–226. DOI: 10.5753/sbbd_estendido.2024.243665. Disponível em: https://sol.sbc.org.br/index.php/sbbd_estendido/article/view/30797.

NAZÁRIO, Débora Cabral; SILVA, Paulo Fernando da; ROVER, Aires José. Avaliação da qualidade da informação disponibilizada no Portal da Transparência do Governo Federal. **Revista Democracia Digital e Governo Eletrônico**, n. 6, 2012.

POSSAMAI, Ana Júlia; SOUZA, Vitoria Gonzatti de. Transparência e dados abertos governamentais: possibilidades e desafios a partir da lei de acesso à informação. **Administração Pública e Gestão Social**, 2020.

RUITER, Julian de; HARENSLAK, Bas. **Data Pipelines with Apache Airflow**. [*S.l.*]: Simon e Schuster, 2021.

SANTOS, Everton Schneider dos *et al.* Performance Variability of Machine Learning Models using Limited Data for Collusion Detection: A Case Study of the Brazilian Car Wash Operation. *In:* ANAIS do XXXIX Simpósio Brasileiro de Bancos de Dados. Florianópolis/SC: SBC, 2024. P. 431–443. DOI: 10.5753/sbbd.2024.240845. Disponível em: https://sol.sbc.org.br/index.php/sbbd/article/view/30711.

SANTOS, Maria Teresa Silva *et al.* Ferramenta de visualizaçao de dados abertos do portal de transparência da câmara municipal da cidade de florianópolis. *In:* SBC. WORKSHOP de Computação Aplicada em Governo Eletrônico (WCGE). [*S.l.*: *s.n.*], 2021. P. 71–82.

SCHMITZ, Fernando *et al.* Detecting Fraud in Public Procurement: A GMM-Based Approach to Analyzing Tender Data. *In:* ANAIS do XXXIX Simpósio Brasileiro de Bancos de Dados. Florianópolis/SC: SBC, 2024. P. 207–219. DOI:

10.5753/sbbd.2024.240649. Disponível em:
https://sol.sbc.org.br/index.php/sbbd/article/view/30694.

VASSILIADIS, Panos; SIMITSIS, Alkis. Extraction, Transformation, and Loading. **Encyclopedia of database systems**, Citeseer, v. 10, p. 14, 2009.

VASSILIADIS, Panos; SIMITSIS, Alkis; SKIADOPOULOS, Spiros. Conceptual modeling for ETL processes. *In:* PROCEEDINGS of the 5th ACM international workshop on Data Warehousing and OLAP. [*S.l.*: *s.n.*], 2002. P. 14–21.

APÊNDICE A – DESCRIÇÃO

Textos elaborados pelo autor, a fim de completar a sua argumentação. Deve ser precedido da palavra APÊNDICE, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.

Quadro 1 – Modelo A.

xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
xxxx	yyyyyyyyyyyyyyy
xxxx	yyyyyyyyyyyyyyy
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
xxxx	yyyyyyyyyyyyyyy
	ttttttttttttttttt
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
ttttttttttttt	
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
	gggggggggggggggggg
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee
rrrrrrrrrrrrrrrr	eeeeeeeeeeeeeeee

Fonte: Elaborada pelo autor (2016).

ANEXO A – DESCRIÇÃO

São documentos não elaborados pelo autor que servem como fundamentação (mapas, leis, estatutos). Deve ser precedido da palavra ANEXO, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.