



# Projeto Coleta, Armazenamento e Visualização de Dados

Pipeline de Coleta de Dados de Processos do TJSP

Prof. Fábio Lopes

Aluno: Marcos A. Specá Junior – TIA: 72256826

# Contexto

- O Brasil possui mais de 77 milhões de processos judiciais, e a grande maioria corre em meio eletrônico.
- Empresas de grande porte possuem alto volume de processos cíveis/consumidor, e gastam muito dinheiro com estes processos.
- Como os dados dos processos e suas decisões são públicas, existe o desejo de comparar o resultado de demandas judiciais de empresas com seus concorrentes.
- O objetivo principal desta comparação é entender possíveis tendências do judiciário e fazer ajuste em estratégias de defesa e economizar dinheiro.

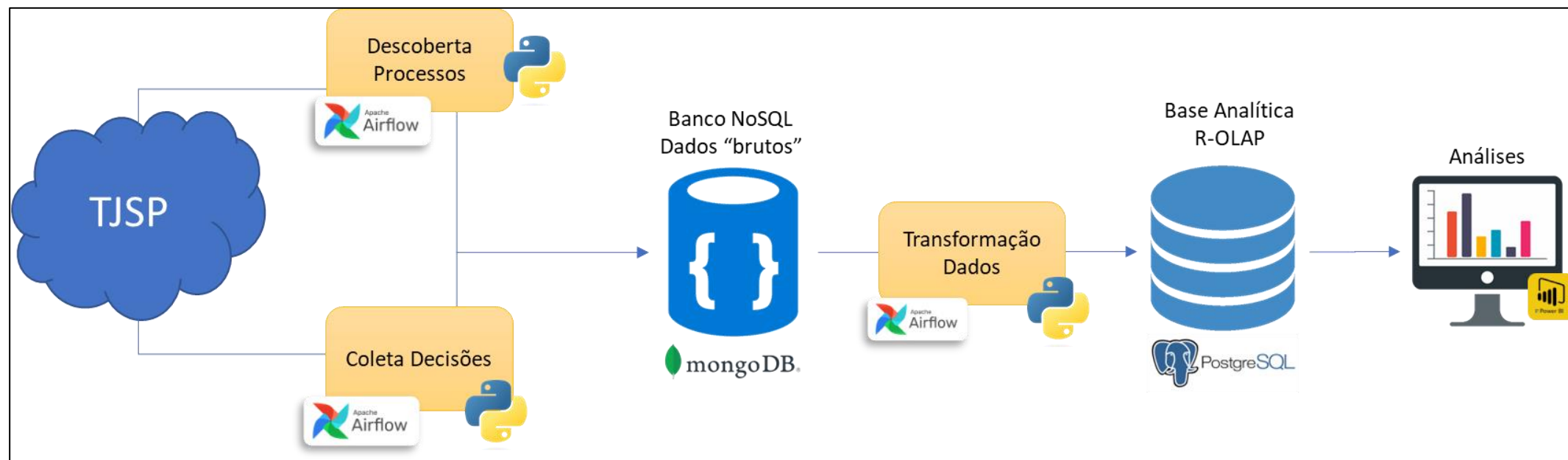
# Problema

- Apesar de a grande maioria dos processos estarem em tramitação eletrônica, existe uma diversidade de sistemas e origens das informações, praticamente um sistema para cada tribunal.
- Além disso os tribunais não possuem uma API, ou seja, a coleta automatizada pode ocorrer apenas com webscrapping.
- Outro desafio é armazenar as informações de processos de diversas naturezas e diversos tribunais, pois o numero e tipo de atributos variam bastante.
- Também não há nas decisões dos processos um atributo que indique o resultado de forma sintetizada, tendo que ser garimpado do texto das decisões.

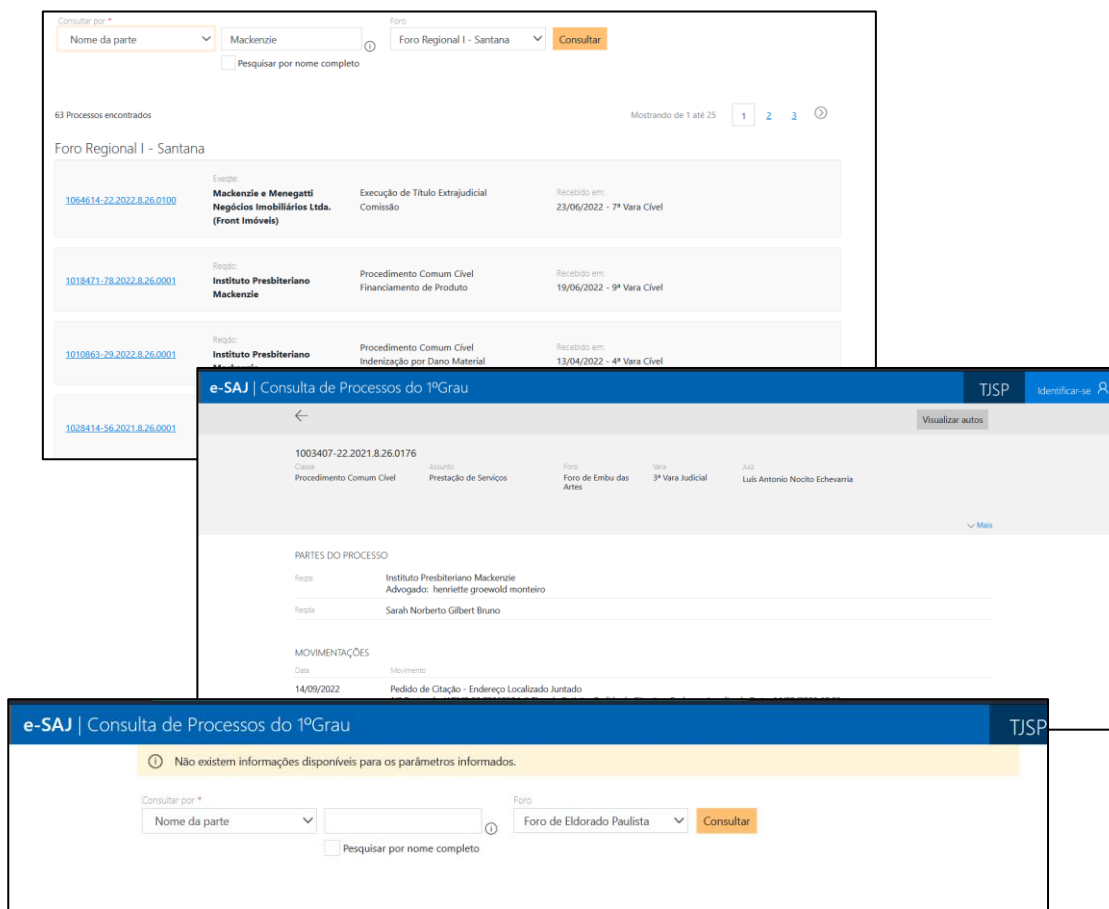
# Solução Proposta

- Pipeline de dados que realize a coleta no site do tribunal (neste caso considerando apenas o TJSP por enquanto), o armazenamento das informações, tanto brutas quanto analíticas e posteriormente uma maneira de análise destes dados.
- Scripts de Webscrapping para coletar os processos e suas respectivas decisões de 1ª Instância
- Armazenamento dos dados de forma a facilitar a tratativa e processamento textual (regular expressions, nlp e etc.).
- Enriquecimento dos dados através de modelos de regex e/ou machine learning para sintetizar os resultados e extrair valores de condenação e danos morais;
- Criar uma base analítica onde seja possível realizar comparações de indicadores como volume de processos por assunto, taxa de improcedência (êxito total) e outras análises futuras.

# Pipeline



# Detalhes Origem



The image displays two screenshots of the TJSP e-SAJ system interface.

**Top Screenshot: Search Results**

Search criteria: Nome da parte: Mackenzie, Foro: Foro Regional I - Santana. 63 processos encontrados. Mostrando de 1 até 25.

Processo	Evento	Assunto	Recebido em
<a href="#">1064614-22.2022.8.26.0100</a>	Mackenzie e Menegatti Negócios Imobiliários Ltda. (Front Imóveis)	Execução de Título Extrajudicial Comissão	23/06/2022 - 7ª Vara Cível
<a href="#">1018471-78.2022.8.26.0001</a>	Instituto Presbiteriano Mackenzie	Procedimento Comum Cível Financiamento de Produto	19/06/2022 - 9ª Vara Cível
<a href="#">1010863-29.2022.8.26.0001</a>	Instituto Presbiteriano Mackenzie	Procedimento Comum Cível Indenização por Dano Material	13/04/2022 - 4ª Vara Cível
<a href="#">1028414-56.2021.8.26.0001</a>			

**Bottom Screenshot: Process Details**

Processo: 1003407-22.2021.8.26.0176

Assunto: Prestação de Serviços

Foro: Foro de Embu das Artes

Vara: 3ª Vara Judicial

Ass: Luis Antonio Nocito Echevarria

**PARTES DO PROCESSO**

Rege: Instituto Presbiteriano Mackenzie

Advogado: henriette groewold monteiro

Rege: Sarah Norberto Gilbert Bruno

**MOVIMENTAÇÕES**

Data: 14/09/2022

Movimento: Pedido de Citação - Endereço Localizado Juntado

**Bottom Screenshot: Search Error**

Não existem informações disponíveis para os parâmetros informados.

Search criteria: Nome da parte: (empty), Foro: Foro de Eldorado Paulista. 0 resultados encontrados.

- O site do TJSP busca de processos do 1º Grau.
- Não há API, portanto é necessário fazer o webscrapping.
- A busca pode ser feita pelo número do processo, nome da parte, cnpj, advogados e outros.
- O site retorna algumas mensagens caso não existam processos para a busca.
- Caso haja um processo apenas para a busca a página de detalhes já é aberta.
- Caso vários processos sejam encontrados uma lista com 25 processos é exibida, criando uma paginação.

# Detalhes Coleta

```
# Importando as Bibliotecas
import pandas as pd
import requests
from bs4 import BeautifulSoup
import json
import pymongo
import time
import yaml

# Parâmetros Iniciais TJSP Esaj
url_base = "https://esaj.tjsp.jus.br/cpopg/search.do?"

# Parâmetros de Busca
busca_nome = ""
busca_cnpj = "60967551000150"
cod_empresa = "MACKENZIE"
dados = []

if busca_cnpj:
    url_pesquisa = url_base + "cbPesquisa=DOCPARTE&"
else:
    url_pesquisa = url_base + "cbPesquisa=IMPORTE&"

url_pesquisa

# Busca Foro por Foro
for foro in lista_foros:
    scrapping_foro(foro)

15.9s

Scrapped do foro: 1 processos: 57
Scrapped do foro: 2 processos: 54
Scrapped do foro: 3 processos: 19
Scrapped do foro: 4 processos: 27
Scrapped do foro: 5 processos: 15
Scrapped do foro: 6 processos: 21

df_processos = pd.DataFrame(dados, columns=["nro_proc", "nome_empresa", "valor", "data"])
df_processos

nome_arquivo = str(time.strftime("%Y_%m_%d")) + '_tjsp_processos_' + cod_empresa + '.csv'

# Salva o arquivo
df_processos.to_csv('./dados/' + nome_arquivo, sep=";", index=False)
```

- Foram criados scripts para a descoberta (pesquisa pelo CNPJ) e para a coleta das decisões.
- Utilizamos algumas bibliotecas como pandas, pymongo (para conexão com mongo), BeautifulSoup e request para o acesso a página.
- Transformamos os dados em json e carregamos no banco de dados Mongo de acordo com sua coleção.
- Também gravamos alguns CVS no início para garantir que os dados fossem armazenados em caso de quebra do script.
- Em algumas situações o site do tribunal falhava devido ao numero de requests.

# Detalhes Processamento

- O processamento das informações foi realizado para classificar as decisões de acordo com seu resultado.
- Consideramos os resultados possíveis: Procedente, Procedente em Parte, Improcedente, Extinto ou Acordo.
- Utilizamos neste momento apenas expressões regulares para classificar as decisões, e no futuro algoritmos mais sofisticados podem ser utilizados.
- Após o processamento, geramos scripts que realizaram a carga no banco de dados analíticos em duas tabelas: processos e decisões, alguns dados textuais mais longos não foram carregados (como o texto das decisões).

```
Tags "Parcialmente Procedente"

col_decisoos.update_many(
  {'$and': [
    {'decisao.texto_decisao': {'$regex': "( PARCIALMENTE PROCEDENTE)"},
    {"decisao.resultado": {"$exists": False}}
  ]},
  {'$set': {'decisao.resultado': 'Parcialmente Procedente'}}
)

col_decisoos.update_many(
  {'$and': [
    {'decisao.texto_decisao': {'$regex': "( JULGO EXTINTO)"},
    {"decisao.resultado": {"$exists": False}}
  ]},
  {'$set': {'decisao.resultado': 'Extinto'}}
)

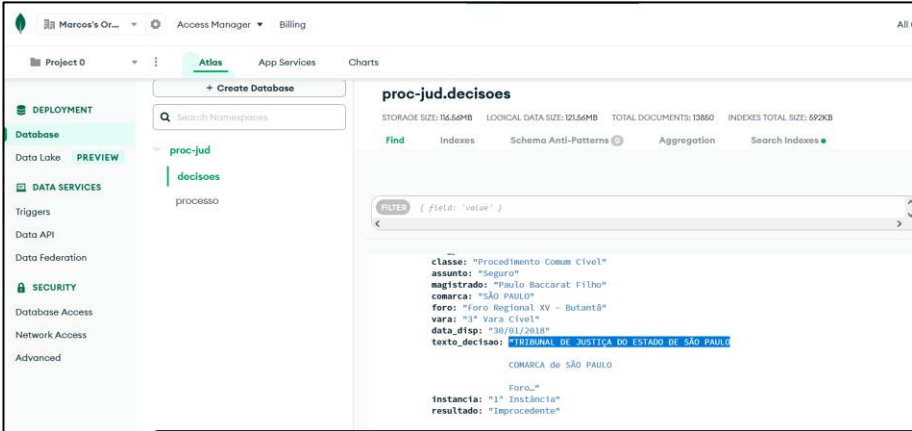
[31] ... <pymongo.results.UpdateResult at 0x258bef27880>

col_decisoos.update_many(
  {'$and': [
    {'decisao.texto_decisao': {'$regex': "( EXTINTA a presente ação)"},
    {"decisao.resultado": {"$exists": False}}
  ]},
  {'$set': {'decisao.resultado': 'Extinto'}}
)

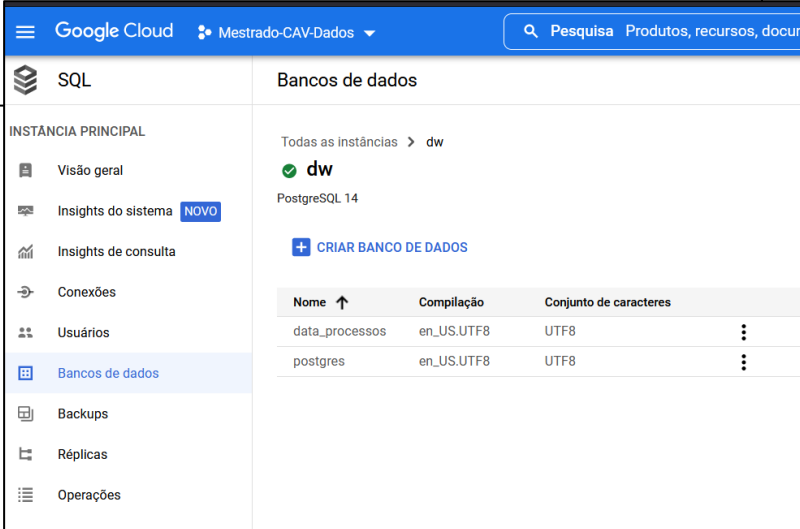
[91] ... <pymongo.results.UpdateResult at 0x258bef25e70>
```



# Detalhes Armazenamento



The screenshot shows the MongoDB Atlas interface. The left sidebar contains navigation options: DEPLOYMENT, Database, Data Lake, DATA SERVICES, Triggers, Data API, Data Federation, SECURITY, Database Access, Network Access, and Advanced. The main panel displays details for the 'proc-jud.decisoes' database, including storage size (14.6MB), logical data size (12.6MB), total documents (13660), and index size (692KB). A sample document is shown with fields like 'classe', 'assunto', 'registrado', 'comarca', 'foro', 'vara', 'data\_disp', and 'texto\_decisao'.



The screenshot shows the Google Cloud SQL console. The left sidebar lists navigation options: INSTÂNCIA PRINCIPAL, Visão geral, Insights do sistema, Insights de consulta, Conexões, Usuários, Bancos de dados, Backups, Réplicas, and Operações. The main panel displays a table of database instances.

Nome	Compilação	Conjunto de caracteres
data_processos	en_US.UTF8	UTF8
postgres	en_US.UTF8	UTF8

- O armazenamento dos dados brutos foi feito em um banco de dados NoSQL MongoDB;
- Banco orientado a documentos com facilidade de tratativas de dados textuais;
- Já para o banco de dados analítico utilizamos o PostgreSQL;
- A base analítica facilitaria a consulta e a modelagem dimensional, favorecendo as análises.

# Detalhes Visualização



- Para visualização utilizamos a ferramenta PowerBI Desktop para facilitar a conexão e a exploração dos dados.
- Criamos algumas visões comparativas como a taxa de improcedência por foro e geral.
- Visão da taxa de improcedência por empresa e assunto e evolução no tempo.
- Também analisamos o perfil do magistrado/juiz que julgou as ações.

# Limitações do Trabalho

- A coleta considerou apenas o tribunal de justiça de São Paulo, e para as empresas que possuem um volume grande de processos faz sentido expandir a análise para os demais tribunais de justiça do Brasil.
- Na prova de conceito não foi possível implantar a ferramenta Apache Airflow, portanto mesmo entendendo que a ferramenta é adequada às necessidades expostas análises futuras de sua viabilidade devem ser realizadas.
- No processo de transformação de dados apenas algumas transformações simples foram aplicadas podendo ser melhoradas através de modelos classificadores para identificar melhor o resultado das decisões.
- Coletamos dados apenas de algumas empresas do segmento de Seguros, e ainda sim as comparações realizadas devem ser feitas com cuidado pois mesmo se tratando de empresas do mesmo segmento os processos podem ser diferentes entre si.

# Conclusão

- O objetivo de estruturar a arquitetura de pipeline de dados para a coleta, processamento e visualização dos dados de processos judiciais do Tribunal de Justiça de São Paulo foi concluído com sucesso.
- Foi possível validar a utilidade de scripts de webscrapping, o armazenamento de dados brutos, as etapas de tratativas de dados, o armazenamento de dados analíticos a sua visualização em ferramentas de dashboards.
- Como etapas futuras para dar continuidade neste desenvolvimento precisaríamos construir um servidor Apache Airflow, adaptar os scripts ao framework Airflow e publicar os dashboards para que o processo todo seja automatizado.

# Dúvidas?



**OBRIGADO!**