



Um app funcional para o MPSP com Streamlit

José Eduardo de Souza Pimentel

Conteúdo



Web Services da SEFAZ



Uso da aplicação




Questões técnicas envolvidas



Conclusão

Web Services da SEFAZ

SP + Digital

[fr](#)[in](#)[d](#)[v](#)[t](#)[ig](#)[f](#)[/governosp](#)

[A+](#)[A-](#)[🌙](#)[⚠](#)

[🏠 INÍCIO](#)[🏛 INSTITUCIONAL](#)[👤 CIDADÃO](#)[🏢 EMPRESA](#)[📄 TRANSPARÊNCIA](#)[📰 NOTÍCIAS](#)[☎ CANAIS DE COMUNICAÇÃO](#)

Você está em: [Transparência](#) > [Web Services](#)

[Receitas](#)

[Despesas](#)

[Convênios, Repasses e Transferências de Recursos](#)

[Balanco Geral do Estado](#)

[Quadros Demonstrativos da Lei de Responsabilidade Fiscal](#)

[Haveres e Dívida Pública](#)

[Tributos - Legislação, Benefícios e Indicadores](#)


Web Services

[f](#)[🗨](#)[t](#)[in](#)[✉](#)

Em atenção ao disposto na Lei Complementar nº 131/2009, o governo de São Paulo passa a disponibilizar, também por meio de Web Service, informações sobre a execução orçamentária e financeira para conhecimento e acompanhamento da sociedade.

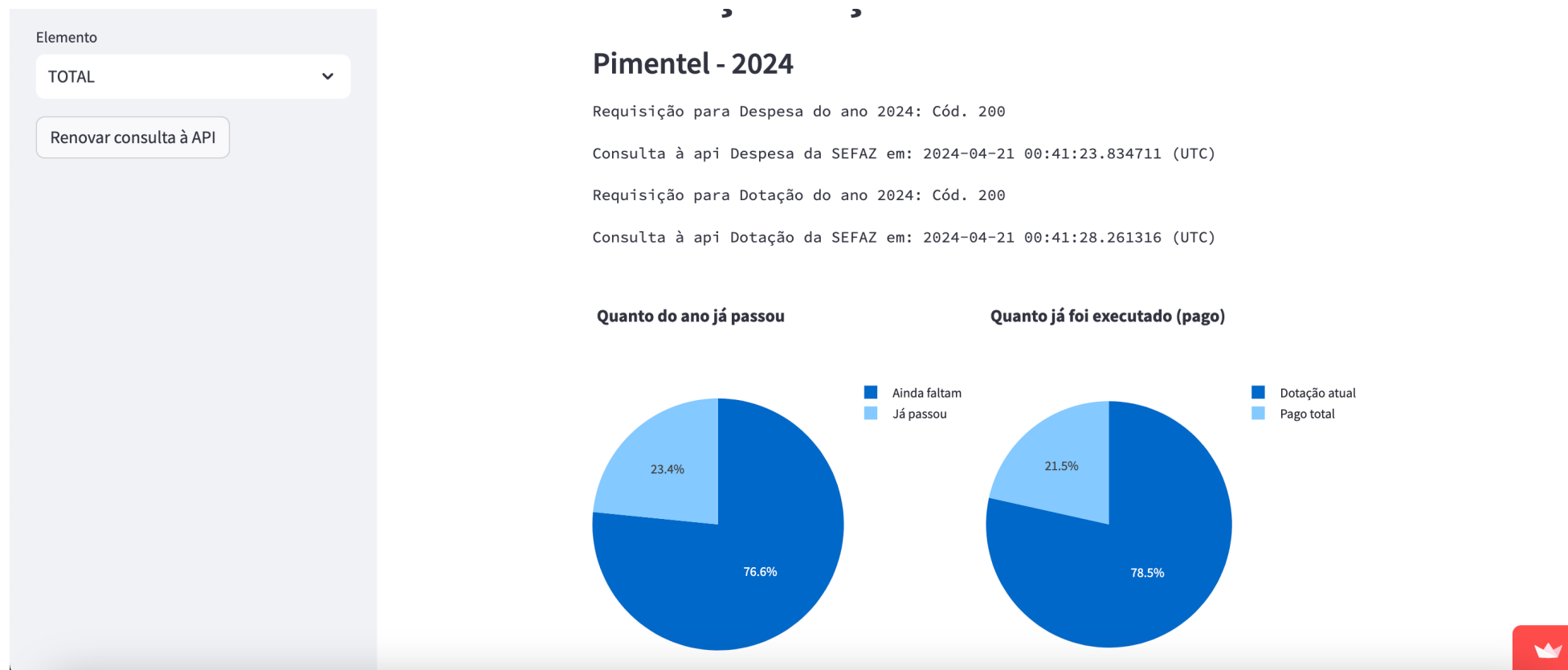
ACESSO AO WEB SERVICE

- <https://webservices.fazenda.sp.gov.br/WSTransparencia/TransparenciaServico.aspx>
- [Manual Técnico](#)



Pesquisa de Opir

Uso da aplicação



Questões técnicas

pip install streamlit plotly requests xmltodict pandas

```
1      # Importações e configuração inicial
2      import streamlit as st
3      import plotly.express as px
4      import requests
5      import xmltodict
6      import warnings
7      import pandas as pd
8      import datetime
```

Estratégias

```
112  ✓ def main():  
113      orgao = '27000'  
114      data_atual = datetime.datetime.now()  
115      ano = data_atual.year  
116      st.title("Execução orçamentária do MPSP")  
117      st.subheader('Pimentel - 2024')  
118
```

```
despesas = consulta_despesas(orgao, str(ano))  
dados_despesas = xmltodict.parse(despesas)  
despesas = dados_despesas['soap:Envelope']['soap:Body']['ConsultarDespesasResponse']
```

```
# Filtra o dataframe  
elemento = st.sidebar.selectbox("Elemento", df_dotacao['CodigoNomeElemento'].unique(), index=len(df_dotacao) - 1)  
df_dotacao_filtrado = df_dotacao[df_dotacao['CodigoNomeElemento'] == elemento]
```

Cache

```
13     # Funções
14     @st.cache_data
15     ✓ def consulta_despesas(orgao, ano):
16         url = "https://webservices.fazenda.sp.gov.br/WSTransparencia/TransparenciaServico.asmx?"
17         headers = {'Content-Type': 'text/xml'}
18
19
165         # Botão no sidebar
166         if st.sidebar.button("Renovar consulta à API"):
167             consulta_despesas.clear()
168             consulta_despesas_dotacao.clear()
169         ---
```

Conclusão

- Rápida prototipação
- Abstração da complexidade
- Solução para a interação com usuários





Referências

- SEFAZ (web services):
<https://portal.fazenda.sp.gov.br/acessoinformacao/Paginas/Web-Services.aspx>
- Streamlit: criando aplicações web (live do Eduardo “Dunossauro” Mendes):
https://www.youtube.com/live/le5ef_R_k6I?si=j_1R4BZJ3mVhZqcN
- Streamlit: <https://streamlit.io/>
- Plotly: <https://plotly.com/python/>