Documentação Profissional do Projeto de Análise Jurídica

Projeto: Processos Judiciais

Autor: José Augusto Palermo de Farias - Data: 31/07/2025

1. Documentação de Requisitos

Objetivo: Entender profundamente as necessidades do negócio jurídico e definir os objetivos da análise.

1.1 Levantamento de Requisitos

- Quantidade de processos por juiz, advogado e parte envolvida;- Distribuição de resultados (ganho/perda);- Impacto da conciliação nos resultados;- Valor das causas ao longo do tempo;

2. Documentação Técnica

Objetivo: Descrever como os dados jurídicos são coletados, tratados e armazenados.

2.1 Modelo de Dados (Tabelas e Relacionamentos)

Dimensões: dim_pessoa, dim_advogado, dim_juiz, d_calendario; Fato: fato_processos.

dim_juiz {

id uuid [pk]

juiz varchar(128)

Vara varchar(128)

tipo_processo varchar(128)

dim_pessoa {

id [pk]

cliente varchar

cpf varchar

endereco varchar

cidade varchar

estado varchar

faixa etaria varchar

status_pessoa [reu ou autor] varchar

numero_do_processo varchar

}

Table dim_advogado

id[pk]

advogado varchar

oab varchar

especialidade varchar

tipo_processo varchar

}

Table fato_processo {

id [pk]

id_pessoa INT

id_juiz INT

id_advogado INT

status [Em andamento, Suspenso, Arquivado e Encerrado] varchar

conciliacao [false ou true] boolean

valor_causa nuemeric(15,2)

resultado do processo [ganho ou perda] varchar

data_inicio_processo date

data_fim_processo date

numero_do_processo varchar

}

Table d_calendario {

data [date]

}Principais relacionamentos:- dim_pessoa.id → fato_processos.id_pessoa- dim_advogado.id → fato_processos.id_advogado- dim_juiz.id → fato_processos.id_juiz- d_calendario.data → fato_processos.data_inicio_processo

2.2 Pipeline de Dados

O pipeline em Python realiza extração dos dados do PostgreSQL, validação, limpeza e exportação para CSV. Além disso, registra logs detalhados em tabela dedicada.Principais etapas do script:1. Conexão dinâmica com PostgreSQL usando db_config.json;2. Funções para conversão de tipos numpy para tipos nativos Python;3. Criação e manutenção da tabela de logs (log_extractions);4. Validação de duplicatas e nulos com tratamento específico por coluna;5. Extração validada de tabelas (dim_pessoa, dim_advogado, dim_juiz, fato_processos) para CSV;6. Geração de relatório consolidado de logs;7. Estatísticas finais no terminal (quantidade de registros, duplicatas removidas, nulos tratados).

3. Documentação de Processos

Execução do script: python pipeline_juridico.py

Orquestração sugerida: Agendador de Tarefas do Windows ou Pentaho.

4. Documentação Analítica

4.1 Hipóteses

- Advogados com mais casos tendem a ter maior taxa de ganhos;- Processos conciliados têm maior probabilidade de ganho;- Faixas etárias específicas concentram maior número de autores ou réus.

4.2 Medidas DAX principais

• Qtd Processos por Advogado – conta processos vinculados a cada advogado;• Valor Ganha por Advogado – soma do valor de causas ganhas por advogado;• Valor Perdido por Advogado – soma do valor de causas perdidas;• % Ganha com Conciliação – proporção de ganhos conciliados sobre ganhos totais;• Qtd Autores e Réus – contagem de partes no papel de autor e réu;• Rank Faixa Etária por Processos – classificação das faixas etárias por volume de processos.

🔷 **Documentação das Medidas DAX**

🔹 **Classificação do Saldo**

Classificação do Saldo =

```
VAR Saldo = [Saldo por Advogado]

RETURN

  IF(

    Saldo > 0,

    "Saldo Positivo",

    IF(Saldo < 0, "Saldo Negativo", "Saldo Zerado")

  )
```

Descrição: Classifica o saldo financeiro de cada advogado como positivo, negativo ou zerado, com base na diferença entre causas ganhas e perdidas.

🔹 **Qtd Processos por Advogado**

Qtd Processos por Advogado =

```
CALCULATE(
    COUNT(fato_processos[numero_do_processo]),
    ALLEXCEPT(dim_advogado, dim_advogado[advogado])
)
```

Descrição: Conta o número total de processos atribuídos a cada advogado, preservando o contexto individual de cada um.

🔹 **Saldo**

Saldo = [Total causa ganha] - [Total causa perdida]

Descrição: Calcula o saldo geral do sistema, subtraindo o valor total das causas perdidas do total das causas ganhas.

🔹 **Saldo por Advogado**

Saldo por Advogado = [Valor Ganha por Advogado] - [Valor Perdido por Advogado]

Descrição: Calcula o saldo individual de cada advogado com base nos valores das causas ganhas e perdidas.

🔹 **Valor Ganha por Advogado**

```
CALCULATE(
    SUM(fato_processos[valor_causa]),
    fato_processos[resultado_processo] = "ganho",
    ALLEXCEPT(dim_advogado, dim_advogado[advogado])
)
```

Descrição: Soma o valor das causas ganhas por cada advogado, mantendo o contexto individual.

🔹 **Valor Perdido por Advogado**

```
CALCULATE(
    SUM(fato_processos[valor_causa]),
    fato_processos[resultado_processo] = "Perda",
    ALLEXCEPT(dim_advogado, dim_advogado[advogado])
)
```

Descrição: Soma o valor das causas perdidas por cada advogado, mantendo o contexto individual.

🔹 **%GanhaComConciliação**

```
DIVIDE([Ganho c/ conciliação], [Total causa ganha], 0)
```

Descrição: Calcula o percentual de causas ganhas que foram conciliadas.

🔹 **%GanhaSEMComConciliação**

```
DIVIDE([Ganho s/ conciliação], [Total causa ganha], 0)
```

Descrição: Calcula o percentual de causas ganhas sem conciliação.

🔹 **%PerdaComConciliação**

```
DIVIDE([Valor Causa - perda conciliado], [Total causa perdida], 0)
```

Descrição: Percentual do valor perdido em causas conciliadas.

🔹 **%PerdaSEMComConciliação**

```
DIVIDE([Valor Causa - perda sem conciliação], [Total causa perdida], 0)
```

Descrição: Percentual do valor perdido em causas sem conciliação.

🔹 **Ganho c/ conciliação**

```
COALESCE(
  CALCULATE(
    SUM(fato_processos[valor_causa]),
    fato_processos[resultado_processo] = "ganho",
    fato_processos[conciliacao] = TRUE()
  ),
  0
)
```

Descrição: Soma dos valores de causas ganhas que foram conciliadas.

🔹 **Ganho s/ conciliação**

```
COALESCE(
  CALCULATE(
    SUM(fato_processos[valor_causa]),
    fato_processos[resultado_processo] = "ganho",
    fato_processos[conciliacao] = FALSE()
  ),
  0
)
```

Descrição: Soma dos valores de causas ganhas sem conciliação.

🔹 **Qtd Autores**

```
CALCULATE(
  COUNTROWS(dim_pessoa),
  dim_pessoa[status_pessoa] = "autor"
)
```

Descrição: Conta o número de pessoas com status de autor.

🔹 **Qtd ganho conciliado**

```
CALCULATE(
  COUNTROWS(fato_processos),
  fato_processos[resultado_processo] = "ganho",
  fato_processos[conciliacao] = TRUE()
)
```

Descrição: Conta os processos ganhos que foram conciliados.

🔹 **Qtd Perda conciliado**

```
CALCULATE(
  COUNTROWS(fato_processos),
  fato_processos[resultado_processo] = "perda",
  fato_processos[conciliacao] = TRUE()
```

)

Descrição: Conta os processos perdidos que foram conciliados.

🔹 **Qtd Processos**

```
DISTINCTCOUNT(fato_processos[numero_do_processo])
```

Descrição: Conta o número total de processos distintos.

🔹 **Qtd Processos por Faixa**

```
CALCULATE(
  DISTINCTCOUNT(fato_processos[numero_do_processo]),
  FILTER(
    ALL(dim_pessoa),
    dim_pessoa[faixa_etaria] = SELECTEDVALUE(dim_pessoa[faixa_etaria])
  )
)
```

Descrição: Conta os processos por faixa etária, respeitando o contexto de filtro.

🔹 **Qtd Réus**

```
CALCULATE(
  COUNTROWS(dim_pessoa),
  dim_pessoa[status_pessoa] = "reu"
)
```

Descrição: Conta o número de pessoas com status de réu.

🔹 **QtdClientes**

```
DISTINCTCOUNT(dim_pessoa[cpf])
```

Descrição: Conta o número total de clientes únicos com base no CPF.

🔹 **Rank Faixa Etária por Processos**

```
RANKX(
```

```
    ALLSELECTED(dim_pessoa[faixa_etaria]),
    [Qtd Processos por Faixa],
    ,
    DESC,
    DENSE
)
```

Descrição: Classifica as faixas etárias com base na quantidade de processos, respeitando os filtros aplicados.

🔹 **Total causa ganha**

```
[Ganho c/ conciliação] + [Ganho s/ conciliação]
```

Descrição: Soma total dos valores de causas ganhas, conciliadas e não conciliadas.

🔹 **Total causa perdida**

```
[Valor Causa - perda conciliado] + [Valor Causa - perda sem conciliação]
```

Descrição: Soma total dos valores de causas perdidas, conciliadas e não conciliadas.

🔹 **Valor Causa - perda conciliado**

```
COALESCE(
    CALCULATE(
        SUM(fato_processos[valor_causa]),
        fato_processos[resultado_processo] = "perda",
        fato_processos[conciliacao] = TRUE()
    ),
    0
)
```

Descrição: Soma dos valores de causas perdidas que foram conciliadas.

🔹 **Valor Causa - perda sem conciliação**

```
COALESCE(
    CALCULATE(
```

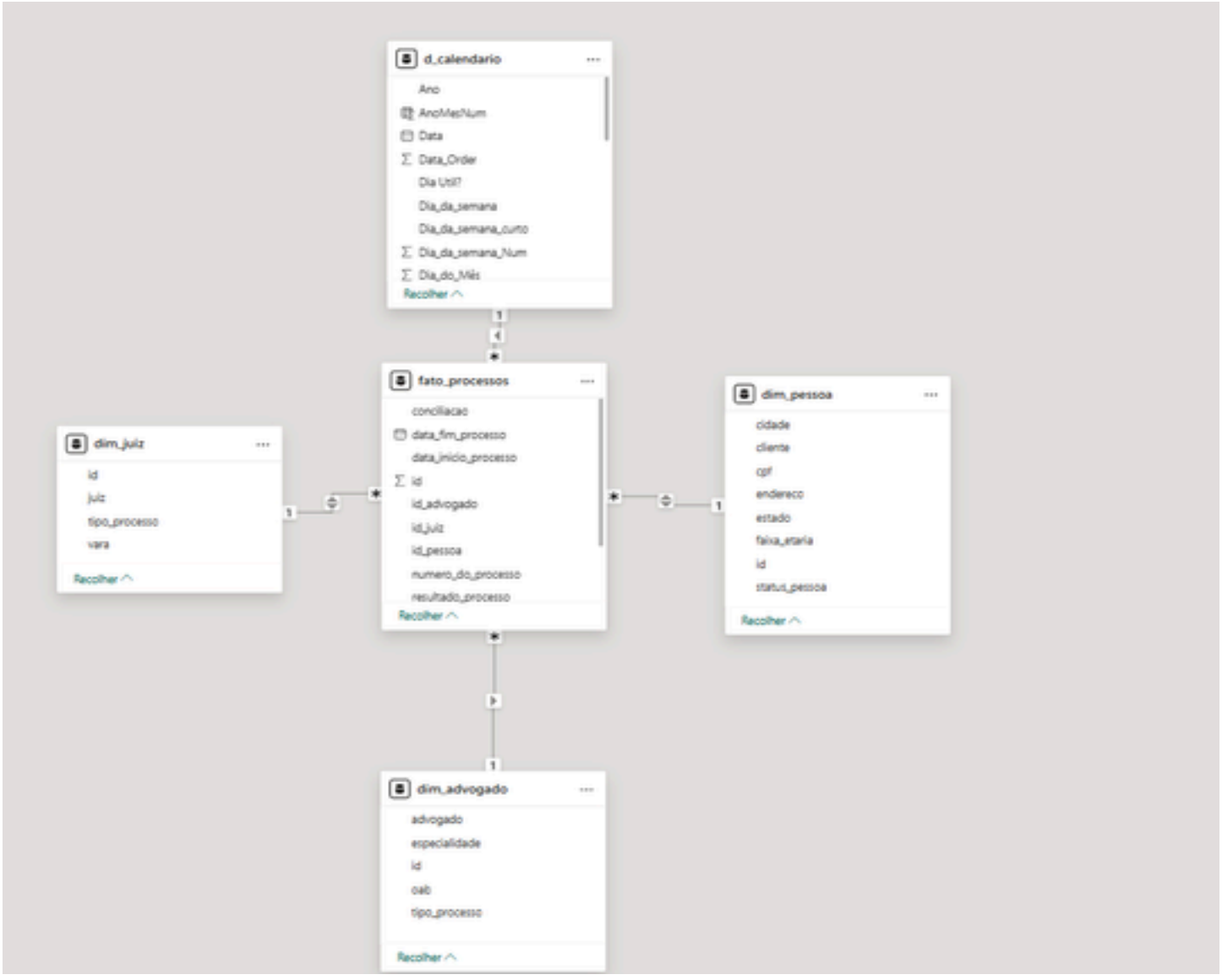
```
    SUM(fato_processos[valor_causa]),
    fato_processos[resultado_processo] = "perda",
    fato_processos[conciliacao] = FALSE()
  ),
  0
)
```

Descrição: Soma dos valores de causas perdidas sem conciliação.

4.3 Paleta de Cores

- Texto: #d9d9d9- Fundo: #080016- Autores: #5B6B00- Réus: #AB222C- Cartão negativo: #9A0814- Fatia pizza verde claro: #BFC5A0

4.4 Relacionamento de Tabelas:



5. Documentação de Qualidade e Governança

Logs detalhados são gravados na tabela log_extractions. Dados sensíveis como CPF são padronizados e tratados.SLA: atualização semanal com resposta a falhas em até 24h.

6. Documentação de Projetos

Ferramentas utilizadas:- Python (pipeline de dados e ETL)- PostgreSQL (armazenamento)- Power BI (visualização e medidas DAX)

Cronograma inclui etapas de requisitos, desenvolvimento do pipeline, testes e publicação.