

# analise\_exploratoria\_IA2A

November 17, 2025

```
[ ]: import os
import pandas as pd
import glob

try:
    from google.colab import drive # type: ignore
except ModuleNotFoundError:
    drive = None

if drive is not None:
    drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: # Caminho do arquivo
caminho = '/content/drive/MyDrive/silver/inmet/estacao=A201/ano=2025/
↪2274a1e5d0bf416ea52743b044afdb9c-0.parquet'

# Lendo o arquivo parquet
df = pd.read_parquet(caminho)

# Exibindo as 5 primeiras linhas
df.head(20)
```

```
[ ]:
```

	timestamp_utc	mes	chuva_mm	temp_c	umid_rel_pct	vento_vel_ms	\
0	2025-01-01 00:00:00	1	0.0	24.7	96.0	0.0	
1	2025-01-01 01:00:00	1	0.0	24.6	96.0	0.8	
2	2025-01-01 02:00:00	1	0.0	24.3	96.0	0.3	
3	2025-01-01 03:00:00	1	0.0	24.3	96.0	0.0	
4	2025-01-01 04:00:00	1	0.0	24.3	97.0	0.0	
5	2025-01-01 05:00:00	1	0.0	24.2	97.0	0.0	
6	2025-01-01 06:00:00	1	0.0	24.1	98.0	0.0	
7	2025-01-01 07:00:00	1	0.0	24.2	98.0	0.0	
8	2025-01-01 08:00:00	1	0.0	24.2	98.0	0.1	
9	2025-01-01 09:00:00	1	0.2	24.0	98.0	0.0	
10	2025-01-01 10:00:00	1	0.0	24.5	98.0	0.2	
11	2025-01-01 11:00:00	1	0.0	27.5	90.0	1.0	
12	2025-01-01 12:00:00	1	0.0	28.9	81.0	1.3	

13	2025-01-01	13:00:00	1	0.0	31.6	73.0	2.8
14	2025-01-01	14:00:00	1	0.0	32.2	68.0	1.9
15	2025-01-01	15:00:00	1	0.0	32.9	63.0	2.6
16	2025-01-01	16:00:00	1	0.0	33.4	59.0	0.8
17	2025-01-01	17:00:00	1	1.2	28.6	82.0	2.3
18	2025-01-01	18:00:00	1	9.2	26.3	93.0	0.7
19	2025-01-01	19:00:00	1	0.0	27.8	91.0	0.9

	vento_dir_graus	vento_rajada_ms	pressao_atm_mb	radiacao_global_kj_m2
0	NaN	2.3	1008.0	NaN
1	NaN	2.2	1008.3	NaN
2	NaN	1.8	1008.4	NaN
3	NaN	1.5	1008.0	NaN
4	NaN	0.0	1007.3	NaN
5	NaN	0.7	1007.2	NaN
6	NaN	0.7	1006.9	NaN
7	NaN	0.0	1006.8	NaN
8	NaN	1.0	1006.7	NaN
9	NaN	0.8	1007.2	NaN
10	NaN	1.0	1007.9	NaN
11	NaN	2.3	1008.3	NaN
12	NaN	3.8	1008.8	NaN
13	NaN	6.2	1008.8	NaN
14	NaN	6.7	1008.7	NaN
15	NaN	7.8	1008.0	NaN
16	NaN	6.8	1007.0	NaN
17	NaN	8.3	1006.8	NaN
18	NaN	10.2	1005.6	NaN
19	NaN	3.3	1004.9	NaN

```
[ ]: import pandas as pd

# Caminho do arquivo
caminho = '/content/drive/MyDrive/silver/ibge_populacao/populacao.parquet'

# Lendo o parquet
pop = pd.read_parquet(caminho)

# Municípios únicos (baseado no código)
municipios_unicos = pop[['cod_mun']].drop_duplicates().sort_values('cod_mun')

# Mostra total e lista
print(f"Total de municípios: {municipios_unicos.shape[0]}")
display(municipios_unicos)
```

Total de municípios: 8

cod\_mun

```

1 1500800
7 1501303
0 1501402
3 1501501
6 1502400
2 1504422
4 1506351
5 1506500

```

```

[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
import warnings
warnings.filterwarnings('ignore')

# =====
# CONFIGURAÇÕES INICIAIS
# =====
BASE_PATH = '/content/drive/MyDrive/silver'
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12, 6)

# =====
# FASE 1: EXPLORAÇÃO INICIAL DA ESTRUTURA DOS DADOS
# =====

def explorar_estrutura(caminho, nome_dataset, sample_size=5):
    """Explora a estrutura de um dataset antes de processar"""
    print(f'\n{"="*80}')
    print(f'EXPLORANDO ESTRUTURA: {nome_dataset}')
    print(f'{"="*80}\n')

    try:
        df = pd.read_parquet(caminho)

        print(f' Arquivo carregado com sucesso!')
        print(f' Shape: {df.shape}')
        print(f' Memória: {df.memory_usage(deep=True).sum() / 1024**2:.2f} MB')

        print(f'\n--- Colunas Disponíveis ---')
        for i, col in enumerate(df.columns, 1):
            print(f' {i}. {col} ({df[col].dtype})')

        print(f'\n--- Primeiras {sample_size} Linhas ---')
        print(df.head(sample_size))

```

```

    print(f'\n--- Informações Gerais ---')
    print(df.info())

    return df

except FileNotFoundError:
    print(f' Arquivo não encontrado: {caminho}')
    return None
except Exception as e:
    print(f' Erro ao carregar: {e}')
    return None

# =====
# FASE 2: EXPLORAÇÃO DE ESTRUTURAS COM SUBPASTAS
# =====

def listar_arquivos_disponiveis(base_path, subpasta):
    """Lista todos os arquivos disponíveis em uma estrutura de pastas"""
    print(f'\n{"="*80}')
    print(f'ARQUIVOS DISPONÍVEIS: {subpasta}')
    print(f'{"="*80}\n')

    caminho_completo = f'{base_path}/{subpasta}'
    path_obj = Path(caminho_completo)

    if not path_obj.exists():
        print(f' Pasta não encontrada: {caminho_completo}')
        return []

    arquivos = []
    for item in sorted(path_obj.rglob('*.parquet')):
        arquivos.append(str(item))
        print(f'    {item.relative_to(path_obj)}')

    print(f'\nTotal de arquivos .parquet encontrados: {len(arquivos)}')
    return arquivos

def explorar_amostra_temporal(base_path, subpasta, anos=[2018], meses=[1]):
    """Explora uma amostra de arquivos temporais"""
    print(f'\n{"="*80}')
    print(f'EXPLORANDO AMOSTRA: {subpasta}')
    print(f'{"="*80}\n')

    for ano in anos:
        for mes in meses:
            # Tentar diferentes padrões de caminho

```

```

padroes = [
    f'{base_path}/{subpasta}/ano={ano}/mes={mes}/data.parquet',
    f'{base_path}/{subpasta}/ano-{ano}/mes-{mes}/data.parquet',
    f'{base_path}/{subpasta}/{ano}/{mes}/data.parquet',
]

for caminho in padroes:
    if Path(caminho).exists():
        print(f'\n--- {ano}/{mes:02d} ---')
        df = pd.read_parquet(caminho)
        print(f'Shape: {df.shape}')
        print(f'Colunas: {list(df.columns)}')
        print(f'Preview:\n{df.head(2)}')
        return df

print(f' Nenhum arquivo encontrado nos padrões testados')
return None

# =====
# FASE 3: CONSOLIDAÇÃO INTELIGENTE
# =====

def consolidar_dataset_temporal(base_path, subpasta, anos=range(2018, 2026),
    ↪meses=range(1, 13)):
    """Consolida datasets com estrutura temporal de forma robusta"""
    print(f'\n{"="*80}')
    print(f'CONSOLIDANDO: {subpasta}')
    print(f'{"="*80}\n')

    dfs = []
    arquivos_encontrados = 0
    arquivos_faltantes = 0

    # Tentar diferentes padrões de caminho
    padroes = [
        lambda a, m: f'{base_path}/{subpasta}/ano={a}/mes={m}/data.parquet',
        lambda a, m: f'{base_path}/{subpasta}/ano-{a}/mes-{m}/data.parquet',
        lambda a, m: f'{base_path}/{subpasta}/{a}/{m}/data.parquet',
    ]

    for ano in anos:
        for mes in meses:
            arquivo_encontrado = False

            for padrao in padroes:
                caminho = padrao(ano, mes)

```

```

        if Path(caminho).exists():
            try:
                df = pd.read_parquet(caminho)
                df['ano'] = ano
                df['mes'] = mes
                dfs.append(df)
                arquivos_encontrados += 1
                arquivo_encontrado = True
                print(f' {ano}/{mes:02d} - {df.shape[0]:,} registros')
                break
            except Exception as e:
                print(f' Erro ao ler {ano}/{mes:02d}: {e}')

    if not arquivo_encontrado:
        arquivos_faltantes += 1

if not dfs:
    print(f'\n Nenhum arquivo foi carregado!')
    return None

df_consolidado = pd.concat(dfs, ignore_index=True)

print(f'\n--- RESUMO DA CONSOLIDAÇÃO ---')
print(f' Arquivos encontrados: {arquivos_encontrados}')
print(f' Arquivos faltantes: {arquivos_faltantes}')
print(f' Total de registros: {df_consolidado.shape[0]:,}')
print(f' Período: {df_consolidado["ano"].min()}-{df_consolidado["ano"].
↪max()}')
print(f' Memória: {df_consolidado.memory_usage(deep=True).sum() / 1024**2:.
↪2f} MB')

return df_consolidado

# =====
# FASE 4: ANÁLISE EXPLORATÓRIA DETALHADA
# =====

def analise_exploratoria_completa(df, nome_dataset):
    """Realiza análise exploratória completa de um dataset"""
    print(f'\n{"="*80}')
    print(f'ANÁLISE EXPLORATÓRIA: {nome_dataset}')
    print(f'{"="*80}\n')

    # 1. INFORMAÇÕES GERAIS
    print('--- 1. INFORMAÇÕES GERAIS ---')
    print(f'Shape: {df.shape}')
    print(f'Memória: {df.memory_usage(deep=True).sum() / 1024**2:.2f} MB')

```

```

print(f'\nTipos de dados:')
print(df.dtypes.value_counts())

# 2. VALORES FALTANTES
print('\n--- 2. VALORES FALTANTES ---')
missing = df.isnull().sum()
missing_pct = 100 * missing / len(df)
missing_df = pd.DataFrame({
    'Missing': missing,
    'Percent': missing_pct
}).sort_values('Percent', ascending=False)

if missing_df['Missing'].sum() > 0:
    print(missing_df[missing_df['Missing'] > 0].head(10))
else:
    print(' Nenhum valor faltante!')

# 3. ESTATÍSTICAS NUMÉRICAS
print('\n--- 3. ESTATÍSTICAS DESCRITIVAS (Colunas Numéricas) ---')
numeric_cols = df.select_dtypes(include=[np.number]).columns
if len(numeric_cols) > 0:
    print(df[numeric_cols].describe())
else:
    print('Nenhuma coluna numérica encontrada')

# 4. COLUNAS CATEGÓRICAS
print('\n--- 4. COLUNAS CATEGÓRICAS (Top 5 valores) ---')
categorical_cols = df.select_dtypes(include=['object', 'category']).columns
for col in categorical_cols[:5]: # Limitar a 5 colunas
    print(f'\n{n{col}:}')
    print(df[col].value_counts().head())

# 5. COBERTURA TEMPORAL
if 'ano' in df.columns and 'mes' in df.columns:
    print('\n--- 5. COBERTURA TEMPORAL ---')
    cobertura = df.groupby(['ano', 'mes']).size().
↪reset_index(name='registros')
    pivot = cobertura.pivot(index='mes', columns='ano', values='registros')
    print(pivot.fillna(0).astype(int))

# 6. DUPLICATAS
print('\n--- 6. VERIFICAÇÃO DE DUPLICATAS ---')
duplicates = df.duplicated().sum()
print(f'Registros duplicados: {duplicates:,} ({100*duplicates/len(df):.
↪2f}%)')

return {

```

```

        'shape': df.shape,
        'missing': missing_df,
        'numeric_summary': df[numeric_cols].describe() if len(numeric_cols) > 0
    else None,
        'duplicates': duplicates
    }

# =====
# EXECUÇÃO PRINCIPAL
# =====

print('='*80)
print('ANÁLISE EXPLORATÓRIA - TCC SANEAMENTO RMB')
print('='*80)

# ETAPA 1: Explorar estrutura da população
print('\n\n### ETAPA 1: POPULAÇÃO ###')
pop = explorar_estrutura(
    f'{BASE_PATH}/ibge_populacao/populacao.parquet',
    'POPULAÇÃO RMB'
)

# ETAPA 2: Listar arquivos disponíveis
print('\n\n### ETAPA 2: INVENTÁRIO DE ARQUIVOS ###')
listar_arquivos_disponiveis(BASE_PATH, 'inmet')
listar_arquivos_disponiveis(BASE_PATH, 'sih')
listar_arquivos_disponiveis(BASE_PATH, 'siops')
listar_arquivos_disponiveis(BASE_PATH, 'sisagua')

# ETAPA 3: Explorar amostras
print('\n\n### ETAPA 3: EXPLORANDO AMOSTRAS ###')
print('\n--- INMET ---')
amostra_inmet = explorar_amostra_temporal(BASE_PATH, 'inmet', anos=[2018],
    meses=[1])

print('\n--- SIH ---')
amostra_sih = explorar_amostra_temporal(BASE_PATH, 'sih', anos=[2018],
    meses=[1])

print('\n--- SISAGUA ---')
amostra_sisagua = explorar_amostra_temporal(BASE_PATH, 'sisagua', anos=[2018],
    meses=[1])

print('\n--- SIOPS ---')
siops = explorar_estrutura(
    f'{BASE_PATH}/siops/indicadores.parquet',
    'SIOPS'
)

```



```
)

# ETAPA 4: Consolidação (descomentar após verificar as estruturas)
# print('\n\n### ETAPA 4: CONSOLIDAÇÃO ###')
# inmet_full = consolidar_dataset_temporal(BASE_PATH, 'inmet')
# sih_full = consolidar_dataset_temporal(BASE_PATH, 'sih')
# sisagua_full = consolidar_dataset_temporal(BASE_PATH, 'sisagua')

# ETAPA 5: Análise exploratória completa (após consolidação)
# print('\n\n### ETAPA 5: ANÁLISE EXPLORATÓRIA COMPLETA ###')
# if inmet_full is not None:
#     analise_exploratoria_completa(inmet_full, 'INMET')
# if sih_full is not None:
#     analise_exploratoria_completa(sih_full, 'SIH')
# if sisagua_full is not None:
#     analise_exploratoria_completa(sisagua_full, 'SISAGUA')
# if siops is not None:
#     analise_exploratoria_completa(siops, 'SIOPS')

print('\n\n' + '='*80)
print('EXPLORAÇÃO INICIAL CONCLUÍDA!')
print('='*80)
print('\nPróximos passos:')
print('1. Verifique a estrutura dos dados acima')
print('2. Descomente a ETAPA 4 para consolidar')
print('3. Descomente a ETAPA 5 para análise completa')
```

```
=====
ANÁLISE EXPLORATÓRIA - TCC SANEAMENTO RMB
=====
```

```
### ETAPA 1: POPULAÇÃO ###
```

```
=====
EXPLORANDO ESTRUTURA: POPULAÇÃO RMB
=====
```

Arquivo carregado com sucesso!

Shape: (48, 3)

Memória: 0.00 MB

--- Colunas Disponíveis ---

1. cod\_mun (int32)

2. ano (int32)

3. populacao (int32)

--- Primeiras 5 Linhas ---

	cod_mun	ano	populacao
0	1501402	2018	1485732
1	1500800	2018	525566
2	1504422	2018	129321
3	1501501	2018	61689
4	1506351	2018	20704

--- Informações Gerais ---

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 48 entries, 0 to 47

Data columns (total 3 columns):

#	Column	Non-Null Count	Dtype
0	cod_mun	48 non-null	int32
1	ano	48 non-null	int32
2	populacao	48 non-null	int32

dtypes: int32(3)

memory usage: 708.0 bytes

None

### ETAPA 2: INVENTÁRIO DE ARQUIVOS ###

=====

ARQUIVOS DISPONÍVEIS: inmet

=====

estacao=A201/ano=2018/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2019/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2020/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2021/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2022/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2023/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2024/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A201/ano=2025/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2018/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2019/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2020/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2021/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2022/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2023/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2024/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A202/ano=2025/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2018/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2019/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2020/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2021/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2022/2274a1e5d0bf416ea52743b044afdb9c-0.parquet

estacao=A227/ano=2023/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2024/2274a1e5d0bf416ea52743b044afdb9c-0.parquet  
estacao=A227/ano=2025/2274a1e5d0bf416ea52743b044afdb9c-0.parquet

Total de arquivos .parquet encontrados: 24

=====

ARQUIVOS DISPONÍVEIS: sih

=====

ano=2018/mes=1/data.parquet  
ano=2018/mes=10/data.parquet  
ano=2018/mes=11/data.parquet  
ano=2018/mes=12/data.parquet  
ano=2018/mes=2/data.parquet  
ano=2018/mes=3/data.parquet  
ano=2018/mes=4/data.parquet  
ano=2018/mes=5/data.parquet  
ano=2018/mes=6/data.parquet  
ano=2018/mes=7/data.parquet  
ano=2018/mes=8/data.parquet  
ano=2018/mes=9/data.parquet  
ano=2019/mes=1/data.parquet  
ano=2019/mes=10/data.parquet  
ano=2019/mes=11/data.parquet  
ano=2019/mes=12/data.parquet  
ano=2019/mes=2/data.parquet  
ano=2019/mes=3/data.parquet  
ano=2019/mes=4/data.parquet  
ano=2019/mes=5/data.parquet  
ano=2019/mes=6/data.parquet  
ano=2019/mes=7/data.parquet  
ano=2019/mes=8/data.parquet  
ano=2019/mes=9/data.parquet  
ano=2020/mes=1/data.parquet  
ano=2020/mes=10/data.parquet  
ano=2020/mes=11/data.parquet  
ano=2020/mes=12/data.parquet  
ano=2020/mes=2/data.parquet  
ano=2020/mes=3/data.parquet  
ano=2020/mes=4/data.parquet  
ano=2020/mes=5/data.parquet  
ano=2020/mes=6/data.parquet  
ano=2020/mes=7/data.parquet  
ano=2020/mes=8/data.parquet  
ano=2020/mes=9/data.parquet  
ano=2021/mes=1/data.parquet  
ano=2021/mes=10/data.parquet

ano=2021/mes=11/data.parquet  
ano=2021/mes=12/data.parquet  
ano=2021/mes=2/data.parquet  
ano=2021/mes=3/data.parquet  
ano=2021/mes=4/data.parquet  
ano=2021/mes=5/data.parquet  
ano=2021/mes=6/data.parquet  
ano=2021/mes=7/data.parquet  
ano=2021/mes=8/data.parquet  
ano=2021/mes=9/data.parquet  
ano=2022/mes=1/data.parquet  
ano=2022/mes=10/data.parquet  
ano=2022/mes=11/data.parquet  
ano=2022/mes=12/data.parquet  
ano=2022/mes=2/data.parquet  
ano=2022/mes=3/data.parquet  
ano=2022/mes=4/data.parquet  
ano=2022/mes=5/data.parquet  
ano=2022/mes=6/data.parquet  
ano=2022/mes=7/data.parquet  
ano=2022/mes=8/data.parquet  
ano=2022/mes=9/data.parquet  
ano=2023/mes=1/data.parquet  
ano=2023/mes=10/data.parquet  
ano=2023/mes=11/data.parquet  
ano=2023/mes=12/data.parquet  
ano=2023/mes=2/data.parquet  
ano=2023/mes=3/data.parquet  
ano=2023/mes=4/data.parquet  
ano=2023/mes=5/data.parquet  
ano=2023/mes=6/data.parquet  
ano=2023/mes=7/data.parquet  
ano=2023/mes=8/data.parquet  
ano=2023/mes=9/data.parquet  
ano=2024/mes=1/data.parquet  
ano=2024/mes=10/data.parquet  
ano=2024/mes=11/data.parquet  
ano=2024/mes=12/data.parquet  
ano=2024/mes=2/data.parquet  
ano=2024/mes=3/data.parquet  
ano=2024/mes=4/data.parquet  
ano=2024/mes=5/data.parquet  
ano=2024/mes=6/data.parquet  
ano=2024/mes=7/data.parquet  
ano=2024/mes=8/data.parquet  
ano=2024/mes=9/data.parquet  
ano=2025/mes=1/data.parquet  
ano=2025/mes=2/data.parquet

ano=2025/mes=3/data.parquet  
ano=2025/mes=4/data.parquet  
ano=2025/mes=5/data.parquet  
ano=2025/mes=6/data.parquet  
ano=2025/mes=7/data.parquet  
ano=2025/mes=8/data.parquet

Total de arquivos .parquet encontrados: 92

=====

ARQUIVOS DISPONÍVEIS: siops

=====

indicadores.parquet

Total de arquivos .parquet encontrados: 1

=====

ARQUIVOS DISPONÍVEIS: sisagua

=====

ano=2018/mes=1/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=10/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=11/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=12/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=2/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=3/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=4/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=5/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=6/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=7/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=8/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2018/mes=9/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=1/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=10/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=11/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=12/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=2/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=3/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=4/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=5/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=6/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=7/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=8/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2019/mes=9/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2020/mes=1/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2020/mes=10/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
ano=2020/mes=11/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet



ano=2024/mes=12/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=2/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=3/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=4/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=5/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=6/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=7/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=8/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2024/mes=9/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=1/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=2/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=3/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=4/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=5/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=6/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet  
 ano=2025/mes=7/034a8f2a6ebd43bf8a886ae0175eef2d-0.parquet

Total de arquivos .parquet encontrados: 91

### ### ETAPA 3: EXPLORANDO AMOSTRAS ###

--- INMET ---

=====

EXPLORANDO AMOSTRA: inmet

=====

Nenhum arquivo encontrado nos padrões testados

--- SIH ---

=====

EXPLORANDO AMOSTRA: sih

=====

--- 2018/01 ---

Shape: (8, 10)

Colunas: ['cod\_mun', 'municipio', 'ano', 'mes', 'internacoes\_total',  
 'internacoes\_hidricas', 'dias\_perm\_total', 'dias\_perm\_hidricas', 'valor\_total',  
 'valor\_hidricas']

Preview:

	cod_mun	municipio	ano	mes	internacoes_total	internacoes_hidricas	\
0	1500800	ANANINDEUA	2018	1	2575	177	
1	1501303	BARCARENA	2018	1	630	12	

	dias_perm_total	dias_perm_hidricas	valor_total	valor_hidricas
--	-----------------	--------------------	-------------	----------------

0	11508.0	664.0	2466386.90	63250.29
1	2695.0	52.0	500537.44	3819.15

--- SISAGUA ---

```
=====
EXPLORANDO AMOSTRA: sisagua
=====
```

Nenhum arquivo encontrado nos padrões testados

--- SIOPS ---

```
=====
EXPLORANDO ESTRUTURA: SIOPS
=====
```

Arquivo carregado com sucesso!

Shape: (56, 13)

Memória: 0.01 MB

--- Colunas Disponíveis ---

1. cod\_mun (object)
2. municipio (object)
3. ano (int64)
4. pct\_transferencias\_sus\_recursos (float64)
5. pct\_receita\_impostos\_transf (float64)
6. despesa\_saude\_pc (float64)
7. pct\_despesa\_pessoal\_saude (float64)
8. pct\_despesa\_medicamentos\_saude (float64)
9. pct\_despesa\_terceiros\_pj\_saude (float64)
10. pct\_despesa\_investimentos\_saude (float64)
11. pct\_despesa\_privado\_sem\_fins (float64)
12. pct\_transferencias\_sobre\_despesa (float64)
13. pct\_receita\_propria\_asps (float64)

--- Primeiras 5 Linhas ---

	cod_mun	municipio	ano	pct_transferencias_sus_recursos \
0	1500800	ANANINDEUA	2018	29.50
1	1500800	ANANINDEUA	2019	29.83
2	1500800	ANANINDEUA	2020	34.44
3	1500800	ANANINDEUA	2021	30.67
4	1500800	ANANINDEUA	2022	27.65

	pct_receita_impostos_transf	despesa_saude_pc	pct_despesa_pessoal_saude \
0	36.48	442.94	50.56
1	37.78	479.90	46.20
2	31.46	626.14	37.80



3	36.25	511.44	37.06
4	31.99	593.56	36.72

	pct_despesa_medicamentos_saude	pct_despesa_terceiros_pj_saude \
0	3.52	30.09
1	1.77	28.83
2	2.38	29.52
3	4.15	28.35
4	4.38	31.25

	pct_despesa_investimentos_saude	pct_despesa_privado_sem_fins \
0	0.81	0.81
1	1.01	6.48
2	1.59	9.42
3	5.20	11.11
4	2.53	1.24

	pct_transferencias_sobre_despesa	pct_receita_propria_asps
0	71.98	15.73
1	67.15	16.00
2	75.60	18.66
3	82.22	19.11
4	75.37	16.08

--- Informações Gerais ---

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 56 entries, 0 to 55

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	cod_mun	56 non-null	object
1	municipio	56 non-null	object
2	ano	56 non-null	int64
3	pct_transferencias_sus_recursos	56 non-null	float64
4	pct_receita_impostos_transf	56 non-null	float64
5	despesa_saude_pc	56 non-null	float64
6	pct_despesa_pessoal_saude	56 non-null	float64
7	pct_despesa_medicamentos_saude	56 non-null	float64
8	pct_despesa_terceiros_pj_saude	56 non-null	float64
9	pct_despesa_investimentos_saude	56 non-null	float64
10	pct_despesa_privado_sem_fins	56 non-null	float64
11	pct_transferencias_sobre_despesa	56 non-null	float64
12	pct_receita_propria_asps	56 non-null	float64

dtypes: float64(10), int64(1), object(2)

memory usage: 5.8+ KB

None

=====

EXPLORAÇÃO INICIAL CONCLUÍDA!

=====

Próximos passos:

1. Verifique a estrutura dos dados acima
2. Descomente a ETAPA 4 para consolidar
3. Descomente a ETAPA 5 para análise completa

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
import warnings
warnings.filterwarnings('ignore')

# =====
# CONFIGURAÇÕES INICIAIS
# =====
BASE_PATH = '/content/drive/MyDrive/silver'
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (14, 6)

print('='*80)
print('ANÁLISE EXPLORATÓRIA COMPLETA - TCC SANEAMENTO RMB')
print('='*80)

# =====
# FASE 1: CARREGAR POPULAÇÃO
# =====
print('\n\n### FASE 1: CARREGANDO POPULAÇÃO ###\n')
pop = pd.read_parquet(f'{BASE_PATH}/ibge_populacao/populacao.parquet')
print(f' População: {pop.shape[0]} registros')
print(f' Municípios únicos: {pop["cod_mun"].nunique()}')
print(f' Anos: {pop["ano"].min()} - {pop["ano"].max()}')
print(f'\nMunicípios da RMB:')
municipios_rmb = pop['cod_mun'].unique()
print(municipios_rmb)

# =====
# FASE 2: CONSOLIDAR INMET
# =====
print('\n\n### FASE 2: CONSOLIDANDO INMET ###\n')

def consolidar_inmet_por_estacao():
    """Consolida INMET que está organizado por estação"""
```

```

dfs = []
estacoes = ['A201', 'A202', 'A227']

for estacao in estacoes:
    for ano in range(2018, 2026):
        caminho = f'{BASE_PATH}/inmet/estacao={estacao}/ano={ano}/'
        path_obj = Path(caminho)

        if path_obj.exists():
            arquivos = list(path_obj.glob('*.parquet'))
            if arquivos:
                try:
                    df = pd.read_parquet(arquivos[0])
                    df['estacao'] = estacao
                    df['ano'] = ano
                    dfs.append(df)
                    print(f' {estacao}/{ano} - {df.shape[0]:,} registros')
                except Exception as e:
                    print(f' Erro {estacao}/{ano}: {e}')

if not dfs:
    return None

inmet_full = pd.concat(dfs, ignore_index=True)
print(f'\n--- RESUMO INMET ---')
print(f'Total de registros: {inmet_full.shape[0]:,}')
print(f'Período: {inmet_full["ano"].min()} - {inmet_full["ano"].max()}')
print(f'Estações: {inmet_full["estacao"].unique()}')
print(f'Memória: {inmet_full.memory_usage(deep=True).sum() / 1024**2:.2f} MB')

return inmet_full

inmet = consolidar_inmet_por_estacao()

# =====
# FASE 3: CONSOLIDAR SIH
# =====
print('\n\n### FASE 3: CONSOLIDANDO SIH ###\n\n')

def consolidar_sih():
    """Consolida dados SIH de 2018-2025"""
    dfs = []

    for ano in range(2018, 2026):
        for mes in range(1, 13):
            caminho = f'{BASE_PATH}/sih/ano={ano}/mes={mes}/data.parquet'

```

```

        if Path(caminho).exists():
            try:
                df = pd.read_parquet(caminho)
                dfs.append(df)
                print(f' {ano}/{mes:02d} - {df.shape[0]} municípios')
            except Exception as e:
                print(f' Erro {ano}/{mes:02d}: {e}')

    if not dfs:
        return None

    sih_full = pd.concat(dfs, ignore_index=True)
    print(f'\n--- RESUMO SIH ---')
    print(f'Total de registros: {sih_full.shape[0]:,}')
    print(f'Período: {sih_full["ano"].min()}-{sih_full["mes"].min()} a_
↳ {sih_full["ano"].max()}-{sih_full["mes"].max()}')
    print(f'Municípios únicos: {sih_full["cod_mun"].nunique()}')
    print(f'Memória: {sih_full.memory_usage(deep=True).sum() / 1024**2:.2f} MB')

    return sih_full

sih = consolidar_sih()

# =====
# FASE 4: CONSOLIDAR SISAGUA
# =====
print('\n\n### FASE 4: CONSOLIDANDO SISAGUA ###\n')

def consolidar_sisagua():
    """Consolida dados SISAGUA de 2018-2025"""
    dfs = []

    for ano in range(2018, 2026):
        for mes in range(1, 13):
            caminho_pattern = f'{BASE_PATH}/sisagua/ano={ano}/mes={mes}/'
            path_obj = Path(caminho_pattern)

            if path_obj.exists():
                arquivos = list(path_obj.glob('*.parquet'))
                if arquivos:
                    try:
                        df = pd.read_parquet(arquivos[0])
                        df['ano'] = ano
                        df['mes'] = mes
                        dfs.append(df)
                        print(f' {ano}/{mes:02d} - {df.shape[0]:,} registros')

```

```

        except Exception as e:
            print(f' Erro {ano}/{mes:02d}: {e}')

    if not dfs:
        return None

    sisagua_full = pd.concat(dfs, ignore_index=True)
    print(f'\n--- RESUMO SISAGUA ---')
    print(f'Total de registros: {sisagua_full.shape[0],}')
    print(f'Período: {sisagua_full["ano"].min()}-{sisagua_full["mes"].min()} a ↵
↵{sisagua_full["ano"].max()}-{sisagua_full["mes"].max()}'')
    print(f'Memória: {sisagua_full.memory_usage(deep=True).sum() / 1024**2:.2f} ↵
↵MB')

    return sisagua_full

sisagua = consolidar_sisagua()

# =====
# FASE 5: CARREGAR SIOPS
# =====
print('\n\n### FASE 5: CARREGANDO SIOPS ###\n')
siops = pd.read_parquet(f'{BASE_PATH}/siops/indicadores.parquet')
print(f' SIOPS: {siops.shape[0]} registros')
print(f' Municípios únicos: {siops["cod_mun"].nunique()}')
print(f' Anos: {siops["ano"].min()} - {siops["ano"].max()}')

# =====
# FASE 6: SALVAR DATASETS CONSOLIDADOS
# =====
print('\n\n### FASE 6: SALVANDO DATASETS CONSOLIDADOS ###\n')

if inmet is not None:
    inmet.to_parquet(f'{BASE_PATH}/inmet_consolidado.parquet', index=False)
    print(f' INMET consolidado salvo')

if sih is not None:
    sih.to_parquet(f'{BASE_PATH}/sih_consolidado.parquet', index=False)
    print(f' SIH consolidado salvo')

if sisagua is not None:
    sisagua.to_parquet(f'{BASE_PATH}/sisagua_consolidado.parquet', index=False)
    print(f' SISAGUA consolidado salvo')

# =====
# FASE 7: ANÁLISE EXPLORATÓRIA DETALHADA
# =====

```

```

def analise_exploratoria(df, nome):
    """Análise exploratória completa"""
    print(f'\n{"="*80}')
    print(f'ANÁLISE EXPLORATÓRIA: {nome}')
    print(f'{"="*80}\n')

    # 1. INFO GERAL
    print('--- 1. INFORMAÇÕES GERAIS ---')
    print(f'Shape: {df.shape}')
    print(f'Colunas: {list(df.columns)}')
    print(f'Memória: {df.memory_usage(deep=True).sum() / 1024**2:.2f} MB\n')

    # 2. MISSING VALUES
    print('--- 2. VALORES FALTANTES ---')
    missing = df.isnull().sum()
    missing_pct = 100 * missing / len(df)
    missing_df = pd.DataFrame({
        'Missing': missing,
        'Percent': missing_pct
    }).sort_values('Percent', ascending=False)

    if missing_df['Missing'].sum() > 0:
        print(missing_df[missing_df['Missing'] > 0].head(10))
    else:
        print(' Nenhum valor faltante!\n')

    # 3. ESTATÍSTICAS NUMÉRICAS
    print('\n--- 3. ESTATÍSTICAS DESCRITIVAS ---')
    numeric_cols = df.select_dtypes(include=[np.number]).columns
    if len(numeric_cols) > 0:
        print(df[numeric_cols].describe().round(2))

    # 4. VARIÁVEIS CATEGÓRICAS
    print('\n--- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---')
    cat_cols = df.select_dtypes(include=['object', 'category']).columns
    for col in cat_cols[:3]: # Mostrar apenas 3
        n_unique = df[col].nunique()
        print(f'\n{n}{col}: {n_unique} valores únicos')
        if n_unique <= 10:
            print(df[col].value_counts())
        else:
            print(df[col].value_counts().head())

    # 5. COBERTURA TEMPORAL
    if 'ano' in df.columns:
        print('\n--- 5. DISTRIBUIÇÃO TEMPORAL ---')

```

```

        if 'mes' in df.columns:
            cobertura = df.groupby(['ano', 'mes']).size().
↪reset_index(name='registros')
            pivot = cobertura.pivot(index='mes', columns='ano',
↪values='registros')
            print('\nRegistros por mês/ano:')
            print(pivot.fillna(0).astype(int))
        else:
            print(df['ano'].value_counts().sort_index())

# 6. DUPLICATAS
print('\n--- 6. DUPLICATAS ---')
dup = df.duplicated().sum()
print(f'Registros duplicados: {dup:,} ({100*dup/len(df):.2f}%)')

return missing_df

# =====
# EXECUTAR ANÁLISES
# =====

print('\n\n' + '='*80)
print('FASE 7: ANÁLISES EXPLORATÓRIAS DETALHADAS')
print('='*80)

# População
print('\n\n### POPULAÇÃO ###')
analise_exploratoria(pop, 'POPULAÇÃO RMB')

# INMET
if inmet is not None:
    print('\n\n### INMET ###')
    analise_exploratoria(inmet, 'INMET - Dados Meteorológicos')

# SIH
if sih is not None:
    print('\n\n### SIH ###')
    analise_exploratoria(sih, 'SIH - Sistema de Informações Hospitalares')

# SISAGUA
if sisagua is not None:
    print('\n\n### SISAGUA ###')
    analise_exploratoria(sisagua, 'SISAGUA - Sistema de Informação de
↪Vigilância da Qualidade da Água')

# SIOPS
print('\n\n### SIOPS ###')

```

```

analise_exploratoria(siops, 'SIOPS - Sistema de Informações sobre Orçamentos_
↳ Públicos em Saúde')

# =====
# FASE 8: ANÁLISES CRUZADAS PRELIMINARES
# =====

print('\n\n' + '='*80)
print('FASE 8: ANÁLISES CRUZADAS PRELIMINARES')
print('='*80)

# 8.1 VERIFICAR ALINHAMENTO DE MUNICÍPIOS
print('\n--- 8.1 ALINHAMENTO DE MUNICÍPIOS ---')
pop_munic = set(pop['cod_mun'].unique())
sih_munic = set(sih['cod_mun'].unique()) if sih is not None else set()
siops_munic = set(siops['cod_mun'].astype(str).unique())

print(f'\nMunicípios na base de População: {len(pop_munic)}')
print(f'Municípios na base de SIH: {len(sih_munic)}')
print(f'Municípios na base de SIOPS: {len(siops_munic)}')

if sih is not None:
    intersecao = pop_munic & sih_munic & siops_munic
    print(f'\nMunicípios presentes nas 3 bases: {len(intersecao)}')
    print(f'Códigos: {sorted(intersecao)}')

# 8.2 COBERTURA TEMPORAL
print('\n--- 8.2 COBERTURA TEMPORAL ---')
print(f'\nPopulação: {pop["ano"].min()} - {pop["ano"].max()}')
if sih is not None:
    print(f'SIH: {sih["ano"].min()} - {sih["ano"].max()}')
print(f'SIOPS: {siops["ano"].min()} - {siops["ano"].max()}')
if inmet is not None:
    print(f'INMET: {inmet["ano"].min()} - {inmet["ano"].max()}')
if sisagua is not None:
    print(f'SISAGUA: {sisagua["ano"].min()} - {sisagua["ano"].max()}')

# 8.3 PRIMEIRAS CORRELAÇÕES (SIH)
if sih is not None:
    print('\n--- 8.3 CORRELAÇÕES PRELIMINARES (SIH) ---')
    sih_corr = sih.select_dtypes(include=[np.number]).corr()
    print('\nCorrelação entre internações totais e hídricas:')
    print(sih_corr.loc['internacoes_total', 'internacoes_hidricas'].round(3))

    print('\nCorrelação entre dias de permanência:')
    print(sih_corr.loc['dias_perm_total', 'dias_perm_hidricas'].round(3))

```



ANÁLISE EXPLORATÓRIA COMPLETA - TCC SANEAMENTO RMB

===

### FASE 1: CARREGANDO POPULAÇÃO ###

População: 48 registros  
Municípios únicos: 8  
Anos: 2018 - 2025

Municípios da RMB:

[1501402 1500800 1504422 1501501 1506351 1506500 1502400 1501303]

### FASE 2: CONSOLIDANDO INMET ###

A201/2018 - 8,760 registros  
A201/2019 - 8,760 registros  
A201/2020 - 8,784 registros  
A201/2021 - 8,760 registros  
A201/2022 - 8,760 registros  
A201/2023 - 8,760 registros  
A201/2024 - 8,784 registros  
A201/2025 - 6,552 registros  
A202/2018 - 8,760 registros  
A202/2019 - 8,760 registros  
A202/2020 - 8,784 registros  
A202/2021 - 8,760 registros  
A202/2022 - 8,760 registros  
A202/2023 - 8,760 registros  
A202/2024 - 8,784 registros  
A202/2025 - 6,552 registros  
A227/2018 - 8,760 registros  
A227/2019 - 8,760 registros  
A227/2020 - 8,784 registros  
A227/2021 - 8,760 registros  
A227/2022 - 8,760 registros  
A227/2023 - 8,760 registros  
A227/2024 - 8,784 registros  
A227/2025 - 6,552 registros

--- RESUMO INMET ---

Total de registros: 203,760

Período: 2018 - 2025

Estações: ['A201' 'A202' 'A227']

Memória: 26.62 MB

### FASE 3: CONSOLIDANDO SIH ###

2018/01 - 8 municípios  
2018/02 - 8 municípios  
2018/03 - 8 municípios  
2018/04 - 8 municípios  
2018/05 - 8 municípios  
2018/06 - 8 municípios  
2018/07 - 8 municípios  
2018/08 - 8 municípios  
2018/09 - 8 municípios  
2018/10 - 8 municípios  
2018/11 - 8 municípios  
2018/12 - 8 municípios  
2019/01 - 8 municípios  
2019/02 - 8 municípios  
2019/03 - 8 municípios  
2019/04 - 8 municípios  
2019/05 - 8 municípios  
2019/06 - 8 municípios  
2019/07 - 8 municípios  
2019/08 - 8 municípios  
2019/09 - 8 municípios  
2019/10 - 8 municípios  
2019/11 - 8 municípios  
2019/12 - 8 municípios  
2020/01 - 8 municípios  
2020/02 - 8 municípios  
2020/03 - 8 municípios  
2020/04 - 8 municípios  
2020/05 - 8 municípios  
2020/06 - 8 municípios  
2020/07 - 8 municípios  
2020/08 - 8 municípios  
2020/09 - 8 municípios  
2020/10 - 8 municípios  
2020/11 - 8 municípios  
2020/12 - 8 municípios  
2021/01 - 8 municípios  
2021/02 - 8 municípios  
2021/03 - 8 municípios  
2021/04 - 8 municípios  
2021/05 - 8 municípios  
2021/06 - 8 municípios  
2021/07 - 8 municípios  
2021/08 - 8 municípios  
2021/09 - 8 municípios  
2021/10 - 8 municípios

2021/11 - 8 municípios  
2021/12 - 8 municípios  
2022/01 - 8 municípios  
2022/02 - 8 municípios  
2022/03 - 8 municípios  
2022/04 - 8 municípios  
2022/05 - 8 municípios  
2022/06 - 8 municípios  
2022/07 - 8 municípios  
2022/08 - 8 municípios  
2022/09 - 8 municípios  
2022/10 - 8 municípios  
2022/11 - 8 municípios  
2022/12 - 8 municípios  
2023/01 - 8 municípios  
2023/02 - 8 municípios  
2023/03 - 8 municípios  
2023/04 - 8 municípios  
2023/05 - 8 municípios  
2023/06 - 8 municípios  
2023/07 - 8 municípios  
2023/08 - 8 municípios  
2023/09 - 8 municípios  
2023/10 - 8 municípios  
2023/11 - 8 municípios  
2023/12 - 8 municípios  
2024/01 - 8 municípios  
2024/02 - 8 municípios  
2024/03 - 8 municípios  
2024/04 - 8 municípios  
2024/05 - 8 municípios  
2024/06 - 8 municípios  
2024/07 - 8 municípios  
2024/08 - 8 municípios  
2024/09 - 8 municípios  
2024/10 - 8 municípios  
2024/11 - 8 municípios  
2024/12 - 8 municípios  
2025/01 - 8 municípios  
2025/02 - 8 municípios  
2025/03 - 8 municípios  
2025/04 - 8 municípios  
2025/05 - 8 municípios  
2025/06 - 8 municípios  
2025/07 - 8 municípios  
2025/08 - 8 municípios

--- RESUMO SIH ---

Total de registros: 736  
Período: 2018-1 a 2025-12  
Municípios únicos: 8  
Memória: 0.12 MB

### FASE 4: CONSOLIDANDO SISAGUA ###

2018/01 - 56 registros  
2018/02 - 49 registros  
2018/03 - 45 registros  
2018/04 - 48 registros  
2018/05 - 48 registros  
2018/06 - 48 registros  
2018/07 - 55 registros  
2018/08 - 55 registros  
2018/09 - 55 registros  
2018/10 - 48 registros  
2018/11 - 48 registros  
2018/12 - 47 registros  
2019/01 - 93 registros  
2019/02 - 98 registros  
2019/03 - 91 registros  
2019/04 - 161 registros  
2019/05 - 163 registros  
2019/06 - 140 registros  
2019/07 - 143 registros  
2019/08 - 116 registros  
2019/09 - 116 registros  
2019/10 - 148 registros  
2019/11 - 146 registros  
2019/12 - 116 registros  
2020/01 - 120 registros  
2020/02 - 93 registros  
2020/03 - 96 registros  
2020/04 - 108 registros  
2020/05 - 106 registros  
2020/06 - 107 registros  
2020/07 - 121 registros  
2020/08 - 108 registros  
2020/09 - 106 registros  
2020/10 - 108 registros  
2020/11 - 107 registros  
2020/12 - 108 registros  
2021/01 - 94 registros  
2021/02 - 99 registros  
2021/03 - 96 registros  
2021/04 - 95 registros

2021/05 - 113 registros  
2021/06 - 107 registros  
2021/07 - 96 registros  
2021/08 - 100 registros  
2021/09 - 104 registros  
2021/10 - 103 registros  
2021/11 - 94 registros  
2021/12 - 13 registros  
2022/01 - 115 registros  
2022/02 - 117 registros  
2022/03 - 119 registros  
2022/04 - 104 registros  
2022/05 - 14 registros  
2022/06 - 21 registros  
2022/07 - 6 registros  
2022/08 - 13 registros  
2022/09 - 12 registros  
2022/10 - 90 registros  
2022/11 - 4 registros  
2022/12 - 7 registros  
2023/01 - 26 registros  
2023/02 - 24 registros  
2023/03 - 23 registros  
2023/04 - 18 registros  
2023/05 - 40 registros  
2023/06 - 36 registros  
2023/07 - 13 registros  
2023/08 - 19 registros  
2023/09 - 13 registros  
2023/10 - 8 registros  
2023/11 - 8 registros  
2023/12 - 8 registros  
2024/01 - 165 registros  
2024/02 - 184 registros  
2024/03 - 169 registros  
2024/04 - 115 registros  
2024/05 - 102 registros  
2024/06 - 169 registros  
2024/07 - 586 registros  
2024/08 - 559 registros  
2024/09 - 165 registros  
2024/10 - 175 registros  
2024/11 - 169 registros  
2024/12 - 62 registros  
2025/01 - 92 registros  
2025/02 - 95 registros  
2025/03 - 98 registros  
2025/04 - 95 registros

2025/05 - 450 registros  
2025/06 - 130 registros  
2025/07 - 6 registros

--- RESUMO SISAGUA ---

Total de registros: 8,979  
Período: 2018-1 a 2025-12  
Memória: 8.73 MB

### FASE 5: CARREGANDO SIOPS ###

SIOPS: 56 registros  
Municípios únicos: 8  
Anos: 2018 - 2024

### FASE 6: SALVANDO DATASETS CONSOLIDADOS ###

INMET consolidado salvo  
SIH consolidado salvo  
SISAGUA consolidado salvo

=====  
FASE 7: ANÁLISES EXPLORATÓRIAS DETALHADAS  
=====

### POPULAÇÃO ###

=====  
ANÁLISE EXPLORATÓRIA: POPULAÇÃO RMB  
=====

--- 1. INFORMAÇÕES GERAIS ---

Shape: (48, 3)  
Colunas: ['cod\_mun', 'ano', 'populacao']  
Memória: 0.00 MB

--- 2. VALORES FALTANTES ---

Nenhum valor faltante!

--- 3. ESTATÍSTICAS DESCRITIVAS ---

	cod_mun	ano	populacao
count	48.00	48.00	48.00
mean	1503084.88	2021.17	326365.81

std	2211.88	2.57	459296.20
min	1500800.00	2018.00	20704.00
25%	1501377.25	2019.00	69550.00
50%	1501950.50	2020.50	129327.00
75%	1504904.25	2024.00	283804.00
max	1506500.00	2025.00	1506420.00

--- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---

--- 5. DISTRIBUIÇÃO TEMPORAL ---

```
ano
2018    8
2019    8
2020    8
2021    8
2024    8
2025    8
Name: count, dtype: int64
```

--- 6. DUPLICATAS ---

Registros duplicados: 0 (0.00%)

### INMET ###

=====

ANÁLISE EXPLORATÓRIA: INMET - Dados Meteorológicos

=====

--- 1. INFORMAÇÕES GERAIS ---

```
Shape: (203760, 12)
Colunas: ['timestamp_utc', 'mes', 'chuva_mm', 'temp_c', 'umid_rel_pct',
'vento_vel_ms', 'vento_dir_graus', 'vento_rajada_ms', 'pressao_atm_mb',
'radiacao_global_kj_m2', 'estacao', 'ano']
Memória: 26.62 MB
```

--- 2. VALORES FALTANTES ---

	Missing	Percent
radiacao_global_kj_m2	203760	100.000000
vento_dir_graus	203760	100.000000
chuva_mm	51902	25.472124
vento_rajada_ms	49650	24.366902
vento_vel_ms	49438	24.262858
umid_rel_pct	47876	23.496270
temp_c	47654	23.387318
pressao_atm_mb	47649	23.384865

--- 3. ESTATÍSTICAS DESCRITIVAS ---

	mes	chuva_mm	temp_c	umid_rel_pct	vento_vel_ms \
count	203760.00	151858.00	156106.00	155884.00	154322.00
mean	6.38	0.36	26.91	80.96	0.98
std	3.40	2.20	2.82	12.78	1.22
min	1.00	0.00	-5.00	30.00	0.00
25%	3.00	0.00	24.70	72.00	0.00
50%	6.00	0.00	26.20	84.00	0.40
75%	9.00	0.00	29.00	92.00	1.70
max	12.00	81.20	37.10	100.00	8.40

	vento_dir_graus	vento_rajada_ms	pressao_atm_mb \
count	0.0	154110.00	156111.00
mean	NaN	4.00	1008.27
std	NaN	3.06	2.47
min	NaN	0.00	998.90
25%	NaN	1.30	1006.60
50%	NaN	4.00	1008.30
75%	NaN	6.20	1010.00
max	NaN	19.60	1016.70

	radiacao_global_kj_m2	ano
count	0.0	203760.00
mean	NaN	2021.39
std	NaN	2.24
min	NaN	2018.00
25%	NaN	2019.00
50%	NaN	2021.00
75%	NaN	2023.00
max	NaN	2025.00

#### --- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---

estacao: 3 valores únicos

estacao

A201 67920

A202 67920

A227 67920

Name: count, dtype: int64

#### --- 5. DISTRIBUIÇÃO TEMPORAL ---

Registros por mês/ano:

ano	2018	2019	2020	2021	2022	2023	2024	2025
mes								
1	2232	2232	2232	2232	2232	2232	2232	2232
2	2016	2016	2088	2016	2016	2016	2088	2016
3	2232	2232	2232	2232	2232	2232	2232	2232
4	2160	2160	2160	2160	2160	2160	2160	2160



5	2232	2232	2232	2232	2232	2232	2232	2232
6	2160	2160	2160	2160	2160	2160	2160	2160
7	2232	2232	2232	2232	2232	2232	2232	2232
8	2232	2232	2232	2232	2232	2232	2232	2232
9	2160	2160	2160	2160	2160	2160	2160	2160
10	2232	2232	2232	2232	2232	2232	2232	0
11	2160	2160	2160	2160	2160	2160	2160	0
12	2232	2232	2232	2232	2232	2232	2232	0

--- 6. DUPLICATAS ---

Registros duplicados: 0 (0.00%)

### SIH ###

=====

ANÁLISE EXPLORATÓRIA: SIH - Sistema de Informações Hospitalares

=====

--- 1. INFORMAÇÕES GERAIS ---

Shape: (736, 10)

Colunas: ['cod\_mun', 'municipio', 'ano', 'mes', 'internacoes\_total',  
'internacoes\_hidricas', 'dias\_perm\_total', 'dias\_perm\_hidricas', 'valor\_total',  
'valor\_hidricas']

Memória: 0.12 MB

--- 2. VALORES FALTANTES ---

Nenhum valor faltante!

--- 3. ESTATÍSTICAS DESCRITIVAS ---

	ano	mes	internacoes_total	internacoes_hidricas	\
count	736.00	736.00	736.00	736.00	
mean	2021.35	6.33	1426.73	31.74	
std	2.22	3.42	1916.44	62.84	
min	2018.00	1.00	41.00	0.00	
25%	2019.00	3.00	258.00	2.00	
50%	2021.00	6.00	579.50	7.00	
75%	2023.00	9.00	1649.00	17.00	
max	2025.00	12.00	7886.00	397.00	

	dias_perm_total	dias_perm_hidricas	valor_total	valor_hidricas
count	736.00	736.00	736.00	736.00
mean	6995.91	119.87	2167868.33	13179.61
std	9982.83	227.17	3420228.25	24451.72
min	142.00	0.00	34900.92	0.00
25%	1291.50	8.00	329020.36	721.29
50%	2707.00	29.00	676899.70	2637.92

75%	7227.50	81.00	2205750.81	8174.74
max	41293.00	1372.00	17059951.52	147787.41

#### --- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---

cod\_mun: 8 valores únicos

cod\_mun

1500800	92
1501303	92
1501402	92
1501501	92
1502400	92
1504422	92
1506351	92
1506500	92

Name: count, dtype: int64

municipio: 8 valores únicos

municipio

ANANINDEUA	92
BARCARENA	92
BELEM	92
BENEVIDES	92
CASTANHAL	92
MARITUBA	92
SANTA BARBARA DO PARA	92
SANTA IZABEL DO PARA	92

Name: count, dtype: int64

#### --- 5. DISTRIBUIÇÃO TEMPORAL ---

Registros por mês/ano:

ano	2018	2019	2020	2021	2022	2023	2024	2025
mes								
1	8	8	8	8	8	8	8	8
2	8	8	8	8	8	8	8	8
3	8	8	8	8	8	8	8	8
4	8	8	8	8	8	8	8	8
5	8	8	8	8	8	8	8	8
6	8	8	8	8	8	8	8	8
7	8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8	8
9	8	8	8	8	8	8	8	0
10	8	8	8	8	8	8	8	0
11	8	8	8	8	8	8	8	0
12	8	8	8	8	8	8	8	0

#### --- 6. DUPLICATAS ---

Registros duplicados: 0 (0.00%)

### SISAGUA ###

=====

ANÁLISE EXPLORATÓRIA: SISAGUA - Sistema de Informação de Vigilância da Qualidade da Água

=====

--- 1. INFORMAÇÕES GERAIS ---

Shape: (8979, 23)

Colunas: ['cod\_mun', 'municipio\_alvo', 'municipio\_sisagua', 'uf', 'parametro', 'parametro\_original', 'dataset', 'ponto\_monitoramento', 'forma\_abastecimento\_tipo', 'forma\_abastecimento\_nome', 'forma\_abastecimento\_codigo', 'eta\_uta\_nome', 'instituicao\_sigla', 'instituicao\_nome', 'fonte\_arquivo', 'amostras\_conformes', 'amostras\_sem\_classificacao', 'amostras\_nao\_conformes', 'amostras\_total', 'percentil\_95', 'pct\_conformes', 'ano', 'mes']

Memória: 8.98 MB

--- 2. VALORES FALTANTES ---

	Missing	Percent
classificacao		
pct_conformes	1938	21.583695

--- 3. ESTATÍSTICAS DESCRITIVAS ---

	amostras_conformes	amostras_sem_classificacao \
count	8979.00	8979.00
mean	11.12	2.00
std	18.24	16.72
min	0.00	0.00
25%	1.00	0.00
50%	9.00	0.00
75%	13.00	1.00
max	233.00	1480.00

	amostras_nao_conformes	amostras_total	percentil_95 \
count	8979.00	8979.00	8979.0
mean	0.45	14.36	0.0
std	3.28	20.44	0.0
min	0.00	0.00	0.0
25%	0.00	8.00	0.0
50%	0.00	11.00	0.0
75%	0.00	13.00	0.0
max	86.00	234.00	0.0

classificacao	pct_conformes	ano	mes
---------------	---------------	-----	-----

count	7041.00	8979.00	8979.00
mean	94.90	2021.74	6.10
std	20.19	2.32	3.17
min	0.00	2018.00	1.00
25%	100.00	2020.00	3.00
50%	100.00	2021.00	6.00
75%	100.00	2024.00	8.00
max	100.00	2025.00	12.00

#### --- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---

cod\_mun: 7 valores únicos

cod\_mun

150130	7124
150140	1373
150442	201
150080	190
150240	75
150150	8
150635	8

Name: count, dtype: int64

municipio\_alvo: 7 valores únicos

municipio\_alvo

Barcarena	7124
Belém	1373
Marituba	201
Ananindeua	190
Castanhal	75
Benevides	8
Santa Bárbara do Pará	8

Name: count, dtype: int64

municipio\_sisagua: 7 valores únicos

municipio\_sisagua

BARCARENA	7124
BELEM	1373
MARITUBA	201
ANANINDEUA	190
CASTANHAL	75
BENEVIDES	8
SANTA BARBARA DO PARA	8

Name: count, dtype: int64

#### --- 5. DISTRIBUIÇÃO TEMPORAL ---

Registros por mês/ano:

ano	2018	2019	2020	2021	2022	2023	2024	2025
-----	------	------	------	------	------	------	------	------

mes								
1	56	93	120	94	115	26	165	92
2	49	98	93	99	117	24	184	95
3	45	91	96	96	119	23	169	98
4	48	161	108	95	104	18	115	95
5	48	163	106	113	14	40	102	450
6	48	140	107	107	21	36	169	130
7	55	143	121	96	6	13	586	6
8	55	116	108	100	13	19	559	0
9	55	116	106	104	12	13	165	0
10	48	148	108	103	90	8	175	0
11	48	146	107	94	4	8	169	0
12	47	116	108	13	7	8	62	0

--- 6. DUPLICATAS ---

Registros duplicados: 0 (0.00%)

### SIOPS ###

```
=====
ANÁLISE EXPLORATÓRIA: SIOPS - Sistema de Informações sobre Orçamentos Públicos
em Saúde
=====
```

--- 1. INFORMAÇÕES GERAIS ---

Shape: (56, 13)

Colunas: ['cod\_mun', 'municipio', 'ano', 'pct\_transferencias\_sus\_recursos',  
'pct\_receita\_impostos\_transf', 'despesa\_saude\_pc', 'pct\_despesa\_pessoal\_saude',  
'pct\_despesa\_medicamentos\_saude', 'pct\_despesa\_terceiros\_pj\_saude',  
'pct\_despesa\_investimentos\_saude', 'pct\_despesa\_privado\_sem\_fins',  
'pct\_transferencias\_sobre\_despesa', 'pct\_receita\_propria\_asps']

Memória: 0.01 MB

--- 2. VALORES FALTANTES ---

Nenhum valor faltante!

--- 3. ESTATÍSTICAS DESCRITIVAS ---

	ano	pct_transferencias_sus_recursos	pct_receita_impostos_transf	\
count	56.00	56.00	56.00	
mean	2021.00	17.79	45.56	
std	2.02	7.25	9.62	
min	2018.00	7.42	30.93	
25%	2019.00	12.40	38.41	
50%	2021.00	15.21	44.30	
75%	2023.00	24.25	50.43	
max	2024.00	34.44	69.65	

	despesa_saude_pc	pct_despesa_pessoal_saude \
count	56.00	56.00
mean	710.85	48.81
std	239.75	7.77
min	245.61	35.80
25%	533.25	42.54
50%	680.64	48.71
75%	865.48	54.40
max	1482.52	63.90

	pct_despesa_medicamentos_saude	pct_despesa_terceiros_pj_saude \
count	56.00	56.00
mean	4.97	19.30
std	5.05	10.69
min	0.00	4.80
25%	0.77	8.78
50%	3.36	20.14
75%	8.29	29.58
max	18.50	37.89

	pct_despesa_investimentos_saude	pct_despesa_privado_sem_fins \
count	56.00	56.00
mean	2.96	0.89
std	2.04	2.29
min	0.22	0.00
25%	1.54	0.00
50%	2.51	0.00
75%	4.18	0.00
max	8.96	11.11

	pct_transferencias_sobre_despesa	pct_receita_propria_asps
count	56.00	56.00
mean	57.15	19.14
std	14.30	3.85
min	25.62	15.02
25%	49.53	16.06
50%	57.91	17.80
75%	65.90	21.54
max	87.67	27.15

#### --- 4. VARIÁVEIS CATEGÓRICAS (Amostra) ---

cod\_mun: 8 valores únicos

cod_mun	
1500800	7
1501303	7
1501402	7

```
1501501      7
1502400      7
1504422      7
1506351      7
1506500      7
Name: count, dtype: int64
```

```
municipio: 8 valores únicos
municipio
ANANINDEUA      7
BARCARENA        7
BELÉM            7
BENEVIDES        7
CASTANHAL        7
MARITUBA         7
SANTA BÁRBARA DO PARÁ  7
SANTA IZABEL DO PARÁ  7
Name: count, dtype: int64
```

#### --- 5. DISTRIBUIÇÃO TEMPORAL ---

```
ano
2018      8
2019      8
2020      8
2021      8
2022      8
2023      8
2024      8
Name: count, dtype: int64
```

#### --- 6. DUPLICATAS ---

Registros duplicados: 0 (0.00%)

### =====

## FASE 8: ANÁLISES CRUZADAS PRELIMINARES

### =====

#### --- 8.1 ALINHAMENTO DE MUNICÍPIOS ---

Municípios na base de População: 8  
Municípios na base de SIH: 8  
Municípios na base de SIOPS: 8

Municípios presentes nas 3 bases: 0  
Códigos: []

#### --- 8.2 COBERTURA TEMPORAL ---

População: 2018 - 2025  
SIH: 2018 - 2025  
SIOPS: 2018 - 2024  
INMET: 2018 - 2025  
SISAGUA: 2018 - 2025

### --- 8.3 CORRELAÇÕES PRELIMINARES (SIH) ---

Correlação entre internações totais e hídricas:  
0.774

Correlação entre dias de permanência:  
0.756

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

# =====
# CONFIGURAÇÕES
# =====
BASE_PATH = '/content/drive/MyDrive/silver'
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (14, 8)

print('='*80)
print('ANÁLISE EXPLORATÓRIA APROFUNDADA - TCC SANEAMENTO RMB')
print('='*80)

# =====
# CARREGAR DADOS CONSOLIDADOS
# =====
print('\n### CARREGANDO DADOS ###\n')

pop = pd.read_parquet(f'{BASE_PATH}/ibge_populacao/populacao.parquet')
inmet = pd.read_parquet(f'{BASE_PATH}/inmet_consolidado.parquet')
sih = pd.read_parquet(f'{BASE_PATH}/sih_consolidado.parquet')
sisagua = pd.read_parquet(f'{BASE_PATH}/sisagua_consolidado.parquet')
siops = pd.read_parquet(f'{BASE_PATH}/siops/indicadores.parquet')

# Converter cod_mun do SIOPS para int
siops['cod_mun'] = siops['cod_mun'].astype(int)
```



```

# Criar dicionário de nomes dos municípios
municipios_dict = dict(zip(sih['cod_mun'], sih['municipio']))

print(f' População: {pop.shape}')
print(f' INMET: {inmet.shape}')
print(f' SIH: {sih.shape}')
print(f' SISAGUA: {sisagua.shape}')
print(f' SIOPS: {siops.shape}')

# =====
# 1. ANÁLISE DE DISTRIBUIÇÕES E OUTLIERS
# =====

print('\n\n' + '='*80)
print('1. ANÁLISE DE DISTRIBUIÇÕES E OUTLIERS')
print('='*80)

# 1.1 SIH - Internações
print('\n--- 1.1 INTERNAÇÕES (SIH) ---\n')

fig, axes = plt.subplots(2, 2, figsize=(14, 10))

# Histograma internações totais
axes[0, 0].hist(sih['internacoes_total'], bins=30, edgecolor='black', alpha=0.7)
axes[0, 0].set_title('Distribuição de Internações Totais', fontsize=12,
    ↪weight='bold')
axes[0, 0].set_xlabel('Internações Totais')
axes[0, 0].set_ylabel('Frequência')
axes[0, 0].axvline(sih['internacoes_total'].median(), color='red',
    ↪linestyle='--', label=f'Mediana: {sih["internacoes_total"].
    ↪median():.0f}')
axes[0, 0].legend()

# Boxplot internações totais
axes[0, 1].boxplot(sih['internacoes_total'], vert=True)
axes[0, 1].set_title('Boxplot Internações Totais', fontsize=12, weight='bold')
axes[0, 1].set_ylabel('Internações')

# Histograma internações hídricas
axes[1, 0].hist(sih['internacoes_hidricas'], bins=30, edgecolor='black',
    ↪alpha=0.7, color='coral')
axes[1, 0].set_title('Distribuição de Internações Hídricas', fontsize=12,
    ↪weight='bold')
axes[1, 0].set_xlabel('Internações Hídricas')
axes[1, 0].set_ylabel('Frequência')
axes[1, 0].axvline(sih['internacoes_hidricas'].median(), color='red',
    ↪linestyle='--', label=f'Mediana:
    ↪{sih["internacoes_hidricas"].median():.0f}')

```

```

axes[1, 0].legend()

# Boxplot interações hídricas
axes[1, 1].boxplot(sih['internacoes_hidricas'], vert=True)
axes[1, 1].set_title('Boxplot Interações Hídricas', fontsize=12, weight='bold')
axes[1, 1].set_ylabel('Interações')

plt.tight_layout()
plt.savefig('1_distribuicoes_internacoes.png', dpi=300, bbox_inches='tight')
plt.show()

# Identificar outliers (usando IQR)
def identificar_outliers(df, coluna):
    Q1 = df[coluna].quantile(0.25)
    Q3 = df[coluna].quantile(0.75)
    IQR = Q3 - Q1
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR
    outliers = df[(df[coluna] < limite_inferior) | (df[coluna] >
↳ limite_superior)]
    return outliers, limite_inferior, limite_superior

outliers_total, li_total, ls_total = identificar_outliers(sih,
↳ 'internacoes_total')
outliers_hidricas, li_hidricas, ls_hidricas = identificar_outliers(sih,
↳ 'internacoes_hidricas')

print(f'Outliers em interações totais: {len(outliers_total)} registros')
print(f' Limites: [{li_total:.0f}, {ls_total:.0f}]')
print(f'\nOutliers em interações hídricas: {len(outliers_hidricas)} registros')
print(f' Limites: [{li_hidricas:.0f}, {ls_hidricas:.0f}]')

if len(outliers_total) > 0:
    print(f'\nTop 5 outliers (interações totais):')
    print(outliers_total.nlargest(5, 'internacoes_total')[['municipio', 'ano',
↳ 'mes', 'internacoes_total']])

# 1.2 SIOPS - Gastos em saúde
print('\n\n--- 1.2 GASTOS EM SAÚDE (SIOPS) ---\n')

fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Despesa per capita
axes[0].hist(siops['despesa_saude_pc'], bins=20, edgecolor='black', alpha=0.7,
↳ color='green')
axes[0].set_title('Distribuição de Despesa em Saúde per Capita', fontsize=12,
↳ weight='bold')

```

```

axes[0].set_xlabel('Despesa per Capita (R$)')
axes[0].set_ylabel('Frequência')
axes[0].axvline(siops['despesa_saude_pc'].median(), color='red',
                linestyle='--', label=f'Mediana: R$ {siops["despesa_saude_pc"]}.
                ↪median():.2f}')
axes[0].legend()

# % Investimentos em saúde
axes[1].hist(siops['pct_despesa_investimentos_saude'], bins=20,
            ↪edgecolor='black', alpha=0.7, color='purple')
axes[1].set_title('Distribuição de % Investimentos em Saúde', fontsize=12,
            ↪weight='bold')
axes[1].set_xlabel('% Investimentos')
axes[1].set_ylabel('Frequência')
axes[1].axvline(siops['pct_despesa_investimentos_saude'].median(), color='red',
                linestyle='--', label=f'Mediana:
                ↪{siops["pct_despesa_investimentos_saude"].median():.2f}%',
                ↪weight='bold')
axes[1].legend()

plt.tight_layout()
plt.savefig('2_distribuicoes_gastos.png', dpi=300, bbox_inches='tight')
plt.show()

# =====
# 2. MATRIZ DE CORRELAÇÃO MULTIVARIADA
# =====

print('\n\n' + '='*80)
print('2. ANÁLISE DE CORRELAÇÕES')
print('='*80)

# 2.1 Correlações SIH
print('\n--- 2.1 CORRELAÇÕES SIH ---\n')

sih_numeric = sih.select_dtypes(include=[np.number]).drop(['ano', 'mes'],
            ↪axis=1, errors='ignore')
corr_sih = sih_numeric.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_sih, annot=True, fmt='.2f', cmap='coolwarm', center=0,
            square=True, linewidths=1, cbar_kws={"shrink": 0.8})
plt.title('Matriz de Correlação - SIH (Internações)', fontsize=14,
            ↪weight='bold', pad=20)
plt.tight_layout()
plt.savefig('3_correlacao_sih.png', dpi=300, bbox_inches='tight')
plt.show()

```

```

print('Correlações mais fortes com interações hídricas:')
corr_hidricas = corr_sih['internacoes_hidricas'].sort_values(ascending=False)
print(corr_hidricas[corr_hidricas.index != 'internacoes_hidricas'].head())

# 2.2 Correlações SIOPS
print('\n\n--- 2.2 CORRELAÇÕES SIOPS ---\n')

siops_numeric = siops.select_dtypes(include=[np.number]).drop(['ano'], axis=1,
    ↪errors='ignore')
corr_siops = siops_numeric.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(corr_siops, annot=True, fmt='.2f', cmap='RdYlGn', center=0,
    square=True, linewidths=1, cbar_kws={"shrink": 0.8})
plt.title('Matriz de Correlação - SIOPS (Gastos em Saúde)', fontsize=14,
    ↪weight='bold', pad=20)
plt.xticks(rotation=45, ha='right')
plt.yticks(rotation=0)
plt.tight_layout()
plt.savefig('4_correlacao_siops.png', dpi=300, bbox_inches='tight')
plt.show()

print('Correlações mais fortes com despesa per capita:')
corr_despesa = corr_siops['despesa_saude_pc'].sort_values(ascending=False)
print(corr_despesa[corr_despesa.index != 'despesa_saude_pc'].head())

# =====
# 3. COMPARAÇÃO ENTRE MUNICÍPIOS
# =====

print('\n\n' + '='*80)
print('3. COMPARAÇÃO ENTRE MUNICÍPIOS')
print('='*80)

# 3.1 Interações por município
print('\n--- 3.1 INTERNAÇÕES POR MUNICÍPIO ---\n')

sih_by_municipio = sih.groupby('municipio').agg({
    'internacoes_total': ['sum', 'mean', 'std'],
    'internacoes_hidricas': ['sum', 'mean', 'std'],
    'valor_total': 'sum',
    'valor_hidricas': 'sum'
}).round(2)

sih_by_municipio.columns = ['_'.join(col) for col in sih_by_municipio.columns]
sih_by_municipio = sih_by_municipio.sort_values('internacoes_total_sum',
    ↪ascending=False)

```

```

print(sih_by_municipio)

# Gráfico de barras - Internações totais e hídricas por município
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

municipios = sih_by_municipio.index
x = np.arange(len(municipios))

# Internações totais
axes[0].bar(x, sih_by_municipio['internacoes_total_sum'], color='steelblue',
            ↪alpha=0.8)
axes[0].set_xticks(x)
axes[0].set_xticklabels(municipios, rotation=45, ha='right')
axes[0].set_title('Total de Internações por Município (2018-2025)',
            ↪fontsize=12, weight='bold')
axes[0].set_ylabel('Total de Internações')
axes[0].grid(axis='y', alpha=0.3)

# Internações hídricas
axes[1].bar(x, sih_by_municipio['internacoes_hidricas_sum'], color='coral',
            ↪alpha=0.8)
axes[1].set_xticks(x)
axes[1].set_xticklabels(municipios, rotation=45, ha='right')
axes[1].set_title('Total de Internações Hídricas por Município (2018-2025)',
            ↪fontsize=12, weight='bold')
axes[1].set_ylabel('Total de Internações Hídricas')
axes[1].grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('5_comparacao_municipios.png', dpi=300, bbox_inches='tight')
plt.show()

# 3.2 Gastos por município
print('\n\n--- 3.2 GASTOS EM SAÚDE POR MUNICÍPIO ---\n')

siops_by_municipio = siops.groupby('municipio').agg({
    'despesa_saude_pc': 'mean',
    'pct_despesa_investimentos_saude': 'mean',
    'pct_despesa_pessoal_saude': 'mean',
    'pct_despesa_medicamentos_saude': 'mean'
}).round(2).sort_values('despesa_saude_pc', ascending=False)

print(siops_by_municipio)

# Gráfico
fig, ax = plt.subplots(figsize=(12, 6))
x = np.arange(len(siops_by_municipio))

```

```

ax.bar(x, siops_by_municipio['despesa_saude_pc'], color='green', alpha=0.7)
ax.set_xticks(x)
ax.set_xticklabels(siops_by_municipio.index, rotation=45, ha='right')
ax.set_title('Despesa Média em Saúde per Capita por Município (2018-2024)',
             ↪fontsize=12, weight='bold')
ax.set_ylabel('Despesa per Capita (R$)')
ax.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.savefig('6_gastos_municipios.png', dpi=300, bbox_inches='tight')
plt.show()

# =====
# 4. ANÁLISE SAZONAL (SEM SÉRIES TEMPORAIS)
# =====

print('\n\n' + '='*80)
print('4. PADRÕES SAZONAIS')
print('='*80)

# 4.1 Média de internações por mês (todos os anos)
print('\n--- 4.1 SAZONALIDADE DAS INTERNAÇÕES ---\n')

sih_by_mes = sih.groupby('mes').agg({
    'internacoes_total': ['mean', 'std'],
    'internacoes_hidricas': ['mean', 'std']
}).round(2)

print('Média de internações por mês:')
print(sih_by_mes)

# Gráfico
fig, axes = plt.subplots(1, 2, figsize=(16, 5))

meses = ['Jan', 'Fev', 'Mar', 'Abr', 'Mai', 'Jun', 'Jul', 'Ago', 'Set', 'Out',
        ↪'Nov', 'Dez']

# Internações totais
axes[0].bar(range(1, 13), sih_by_mes['internacoes_total']['mean'],
            yerr=sih_by_mes['internacoes_total']['std'], capsize=5,
            color='steelblue', alpha=0.7)
axes[0].set_xticks(range(1, 13))
axes[0].set_xticklabels(meses, rotation=45)
axes[0].set_title('Média Mensal de Internações Totais', fontsize=12,
                 ↪weight='bold')
axes[0].set_ylabel('Média de Internações')
axes[0].grid(axis='y', alpha=0.3)

# Internações hídricas

```

```

axes[1].bar(range(1, 13), sih_by_mes['internacoes_hidricas']['mean'],
            yerr=sih_by_mes['internacoes_hidricas']['std'], capsize=5,
            color='coral', alpha=0.7)
axes[1].set_xticks(range(1, 13))
axes[1].set_xticklabels(meses, rotation=45)
axes[1].set_title('Média Mensal de Internações Hídricas', fontsize=12,
                 weight='bold')
axes[1].set_ylabel('Média de Internações')
axes[1].grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('7_sazonalidade.png', dpi=300, bbox_inches='tight')
plt.show()

# Identificar meses com mais internações hídricas
print('\nTop 3 meses com mais internações hídricas (média):')
top_meses = sih_by_mes['internacoes_hidricas']['mean'].nlargest(3)
for i, (mes, valor) in enumerate(top_meses.items(), 1):
    print(f'{i}. {meses[mes-1]}: {valor:.2f} internações')

# 4.2 Clima e sazonalidade
print('\n\n--- 4.2 PADRÕES CLIMÁTICOS ---\n')

inmet_by_mes = inmet.groupby('mes').agg({
    'chuva_mm': 'mean',
    'temp_c': 'mean',
    'umid_rel_pct': 'mean'
}).round(2)

print('Média mensal de variáveis climáticas:')
print(inmet_by_mes)

# Gráfico
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

# Chuva
axes[0].bar(range(1, 13), inmet_by_mes['chuva_mm'], color='dodgerblue', alpha=0.7)
axes[0].set_xticks(range(1, 13))
axes[0].set_xticklabels(meses, rotation=45)
axes[0].set_title('Média Mensal de Chuva', fontsize=12, weight='bold')
axes[0].set_ylabel('Chuva (mm)')
axes[0].grid(axis='y', alpha=0.3)

# Temperatura
axes[1].bar(range(1, 13), inmet_by_mes['temp_c'], color='orangered', alpha=0.7)
axes[1].set_xticks(range(1, 13))

```

```

axes[1].set_xticklabels(meses, rotation=45)
axes[1].set_title('Média Mensal de Temperatura', fontsize=12, weight='bold')
axes[1].set_ylabel('Temperatura (°C)')
axes[1].grid(axis='y', alpha=0.3)

# Umidade
axes[2].bar(range(1, 13), inmet_by_mes['umid_rel_pct'], color='teal', alpha=0.7)
axes[2].set_xticks(range(1, 13))
axes[2].set_xticklabels(meses, rotation=45)
axes[2].set_title('Média Mensal de Umidade Relativa', fontsize=12,
↳weight='bold')
axes[2].set_ylabel('Umidade (%)')
axes[2].grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('8_clima_mensal.png', dpi=300, bbox_inches='tight')
plt.show()

# =====
# 5. ANÁLISES CRUZADAS
# =====
print('\n\n' + '='*80)
print('5. ANÁLISES CRUZADAS')
print('='*80)

# 5.1 Criar dataset integrado anual
print('\n--- 5.1 DATASET INTEGRADO (ANUAL) ---\n')

# Agregar SIH por município e ano
sih_anual = sih.groupby(['cod_mun', 'municipio', 'ano']).agg({
    'internacoes_total': 'sum',
    'internacoes_hidricas': 'sum',
    'valor_total': 'sum',
    'valor_hidricas': 'sum'
}).reset_index()

# Converter cod_mun para int em todas as bases
sih_anual['cod_mun'] = sih_anual['cod_mun'].astype(int)
pop['cod_mun'] = pop['cod_mun'].astype(int)
siops['cod_mun'] = siops['cod_mun'].astype(int)

# Merge com população
df_integrado = sih_anual.merge(pop, on=['cod_mun', 'ano'], how='left')

# Merge com SIOPS
df_integrado = df_integrado.merge(siops, on=['cod_mun', 'ano'], how='left',
↳suffixes=('', '_siops'))

```



```

# Calcular taxas por 100k habitantes
df_integrado['taxa_internacoes_100k'] = (df_integrado['internacoes_total'] /
    ↳df_integrado['populacao']) * 100000
df_integrado['taxa_hidricas_100k'] = (df_integrado['internacoes_hidricas'] /
    ↳df_integrado['populacao']) * 100000

print(f' Dataset integrado criado: {df_integrado.shape}')
print(f'\nPrimeiras linhas:')
print(df_integrado.head())

# 5.2 Correlação: Investimento x Internações
print('\n\n--- 5.2 INVESTIMENTO vs INTERNAÇÕES ---\n')

# Remover NaNs
df_analise = df_integrado.dropna(subset=['pct_despesa_investimentos_saude',
    ↳'taxa_hidricas_100k'])

if len(df_analise) > 0:
    corr_invest_hidricas = df_analise['pct_despesa_investimentos_saude'].
    ↳corr(df_analise['taxa_hidricas_100k'])
    print(f'Correlação entre % investimentos e taxa de internações hídricas:
    ↳{corr_invest_hidricas:.3f}')

    # Scatter plot
    plt.figure(figsize=(10, 6))
    plt.scatter(df_analise['pct_despesa_investimentos_saude'],
        df_analise['taxa_hidricas_100k'],
        alpha=0.6, s=100)
    plt.xlabel('% Despesa em Investimentos na Saúde', fontsize=11)
    plt.ylabel('Taxa de Internações Hídricas (por 100k hab.)', fontsize=11)
    plt.title('Relação: Investimentos em Saúde vs Internações Hídricas',
    ↳fontsize=13, weight='bold')
    plt.grid(alpha=0.3)

    # Linha de tendência
    z = np.polyfit(df_analise['pct_despesa_investimentos_saude'],
        df_analise['taxa_hidricas_100k'], 1)
    p = np.poly1d(z)
    plt.plot(df_analise['pct_despesa_investimentos_saude'],
        p(df_analise['pct_despesa_investimentos_saude']),
        "r--", alpha=0.8, linewidth=2)

    plt.tight_layout()
    plt.savefig('9_investimento_vs_internacoes.png', dpi=300,
    ↳bbox_inches='tight')

```

```

plt.show()

# 5.3 Ranking de municípios
print('\n\n--- 5.3 RANKING DE MUNICÍPIOS ---\n')

ranking = df_integrado.groupby('municipio').agg({
    'taxa_hidricas_100k': 'mean',
    'despesa_saude_pc': 'mean',
    'pct_despesa_investimentos_saude': 'mean',
    'populacao': 'mean'
}).round(2).sort_values('taxa_hidricas_100k', ascending=False)

print('Ranking por taxa de internações hídricas (média 2018-2024):')
print(ranking[['taxa_hidricas_100k', 'despesa_saude_pc', 'populacao']])

print('\n\n' + '='*80)
print('ANÁLISE EXPLORATÓRIA APROFUNDADA CONCLUÍDA!')
print('='*80)
print('\nGRÁFICOS SALVOS:')
print(' 1. distribuicoes_internacoes.png')
print(' 2. distribuicoes_gastos.png')
print(' 3. correlacao_sih.png')
print(' 4. correlacao_siops.png')
print(' 5. comparacao_municipios.png')
print(' 6. gastos_municipios.png')
print(' 7. sazonalidade.png')
print(' 8. clima_mensal.png')
print(' 9. investimento_vs_internacoes.png')
print('\nDATASET INTEGRADO:')
print(' - df_integrado: pronto para análises avançadas e modelagem')

```

```

=====
ANÁLISE EXPLORATÓRIA APROFUNDADA - TCC SANEAMENTO RMB
=====

```

```

### CARREGANDO DADOS ###

```

```

População: (48, 3)
INMET: (203760, 12)
SIH: (736, 10)
SISAGUA: (8979, 23)
SIOPS: (56, 13)

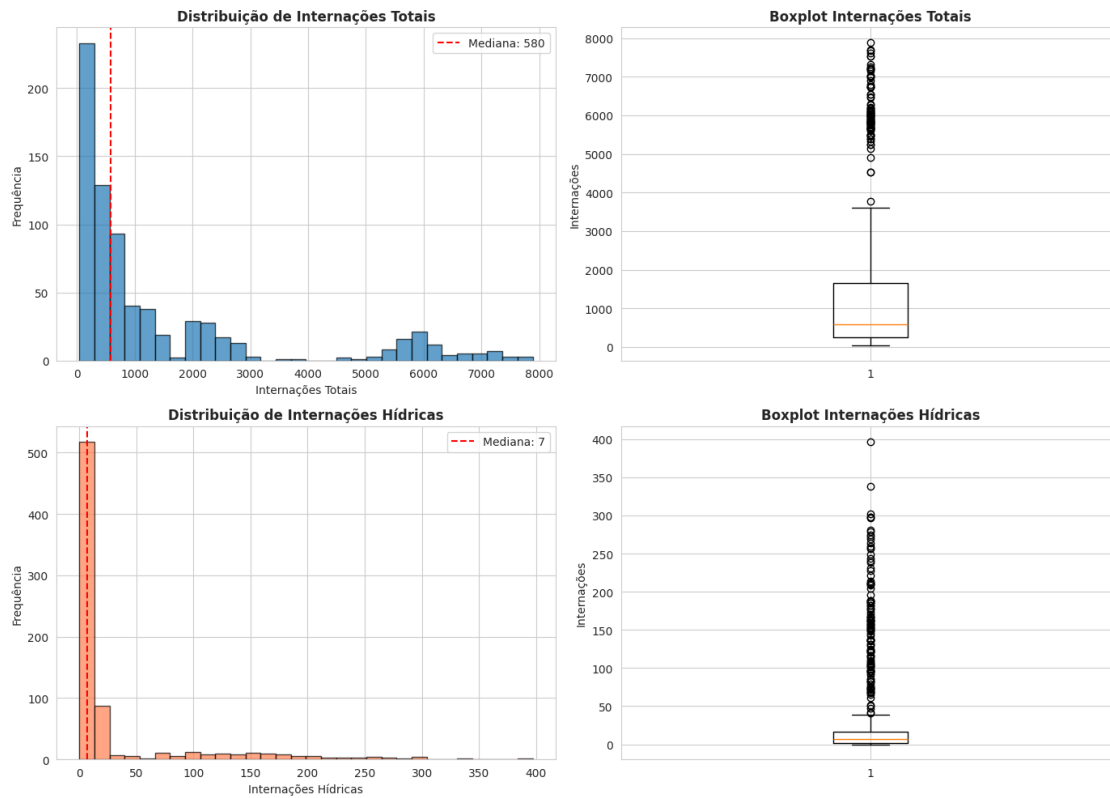
```

```

=====
1. ANÁLISE DE DISTRIBUIÇÕES E OUTLIERS
=====

```

## --- 1.1 INTERNAÇÕES (SIH) ---



Outliers em internações totais: 91 registros

Limites: [-1828, 3736]

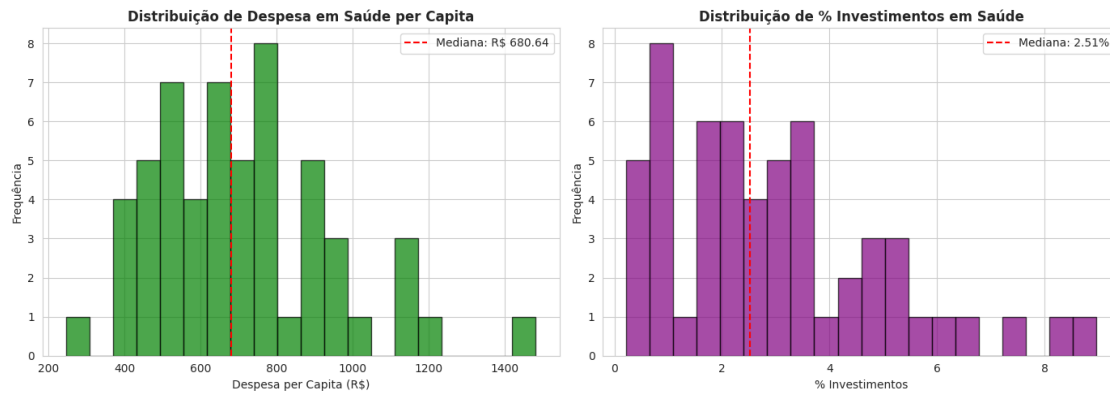
Outliers em internações hídricas: 125 registros

Limites: [-20, 40]

Top 5 outliers (internações totais):

	municipio	ano	mes	internacoes_total
714	BELEM	2025	6	7886
706	BELEM	2025	5	7696
730	BELEM	2025	8	7670
658	BELEM	2024	11	7616
722	BELEM	2025	7	7524

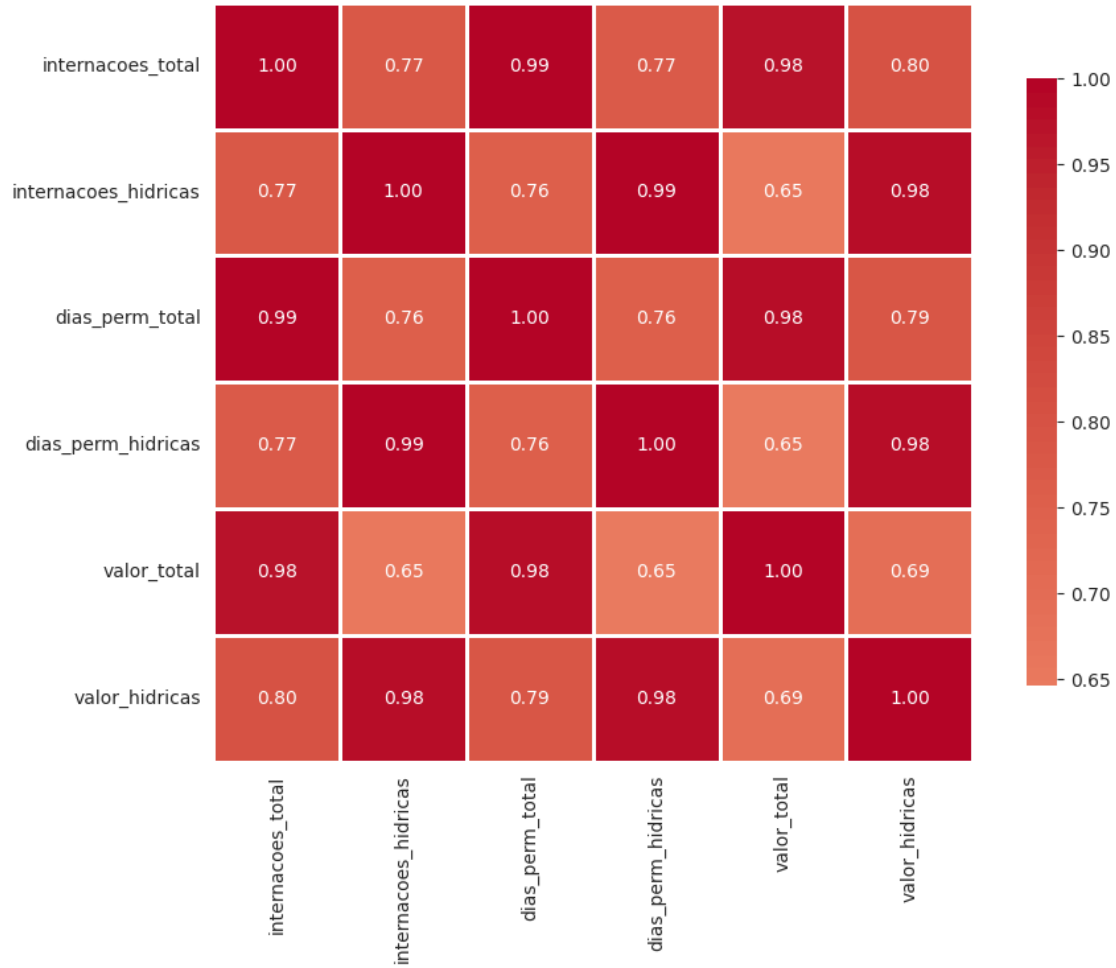
## --- 1.2 GASTOS EM SAÚDE (SIOPS) ---



## 2. ANÁLISE DE CORRELAÇÕES

### --- 2.1 CORRELAÇÕES SIH ---

**Matriz de Correlação - SIH (Internações)**



Correlações mais fortes com internações hídricas:

dias\_perm\_hidricas 0.993204

valor\_hidricas 0.979574

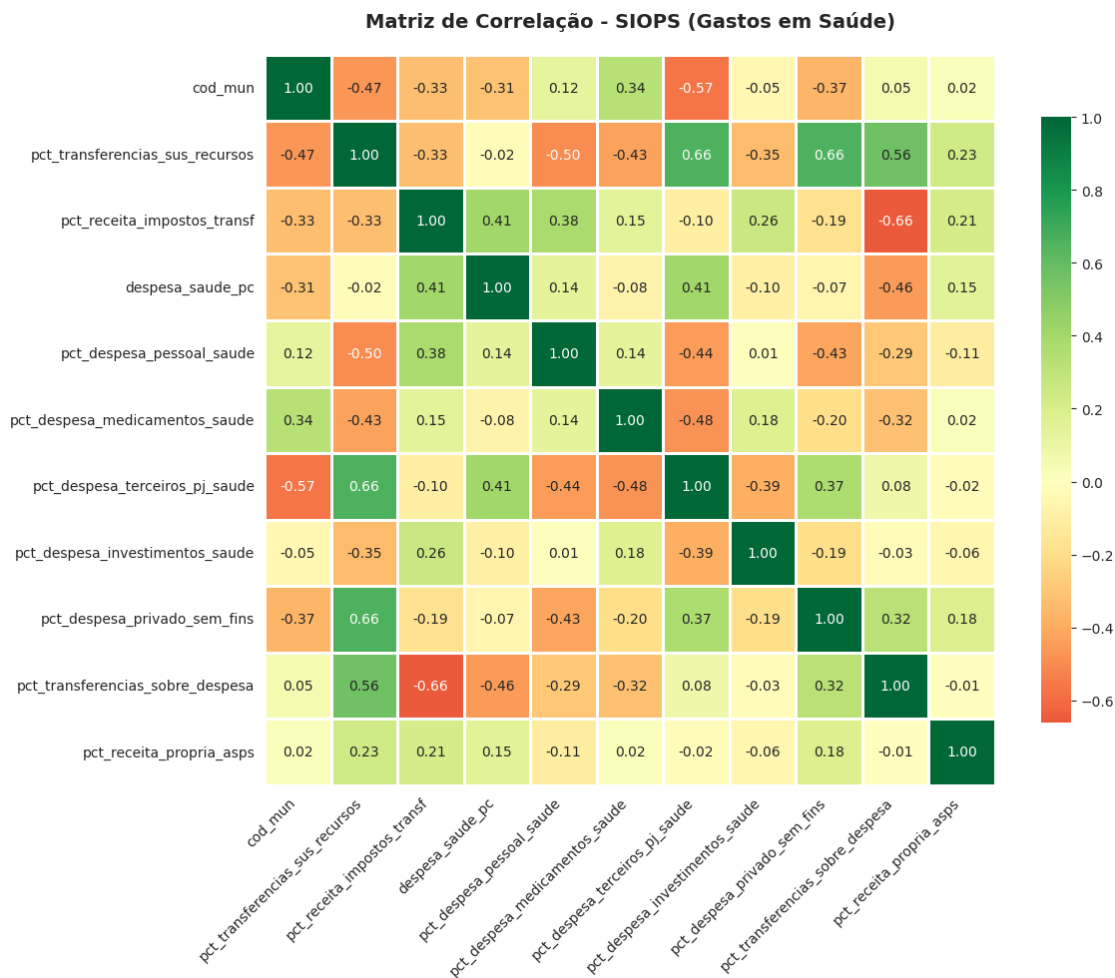
internacoes\_total 0.773714

dias\_perm\_total 0.757096

valor\_total 0.649747

Name: internacoes\_hidricas, dtype: float64

--- 2.2 CORRELAÇÕES SIOPS ---



Correlações mais fortes com despesa per capita:

```
pct_despesa_terceiros_pj_saude      0.409805
pct_receita_impostos_transf         0.409777
pct_receita_propria_asps             0.153904
pct_despesa_pessoal_saude            0.139382
pct_transferencias_sus_recursos      -0.022103
Name: despesa_saude_pc, dtype: float64
```

### 3. COMPARAÇÃO ENTRE MUNICÍPIOS

#### --- 3.1 INTERNAÇÕES POR MUNICÍPIO ---

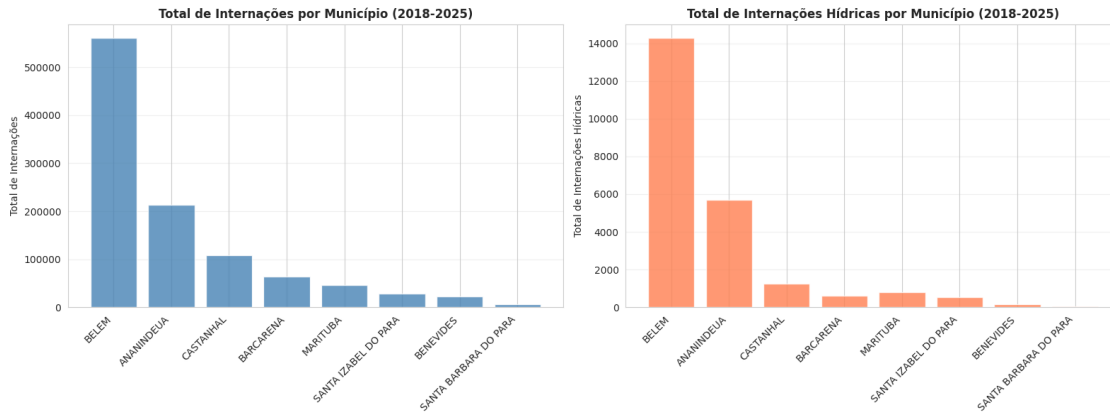
```
internacoes_total_sum  internacoes_total_mean \
municipio
```

BELEM	561264	6100.70
ANANINDEUA	213166	2317.02
CASTANHAL	107800	1171.74
BARCARENA	63375	688.86
MARITUBA	46791	508.60
SANTA IZABEL DO PARA	28547	310.29
BENEVIDES	22333	242.75
SANTA BARBARA DO PARA	6797	73.88

	internacoes_total_std	internacoes_hidricas_sum \
municipio		
BELEM	793.66	14291
ANANINDEUA	309.03	5687
CASTANHAL	178.99	1263
BARCARENA	102.62	599
MARITUBA	56.35	804
SANTA IZABEL DO PARA	91.86	542
BENEVIDES	38.02	148
SANTA BARBARA DO PARA	11.80	29

	internacoes_hidricas_mean	internacoes_hidricas_std \
municipio		
BELEM	155.34	83.22
ANANINDEUA	61.82	66.09
CASTANHAL	13.73	5.92
BARCARENA	6.51	3.29
MARITUBA	8.74	7.80
SANTA IZABEL DO PARA	5.89	5.68
BENEVIDES	1.61	1.79
SANTA BARBARA DO PARA	0.32	0.57

	valor_total_sum	valor_hidricas_sum
municipio		
BELEM	9.600674e+08	5698035.61
ANANINDEUA	3.080774e+08	2664280.76
CASTANHAL	1.231247e+08	532848.79
BARCARENA	6.961955e+07	226261.15
MARITUBA	6.001125e+07	309940.21
SANTA IZABEL DO PARA	3.161577e+07	160225.62
BENEVIDES	3.278443e+07	84593.62
SANTA BARBARA DO PARA	1.025060e+07	24003.92



### --- 3.2 GASTOS EM SAÚDE POR MUNICÍPIO ---

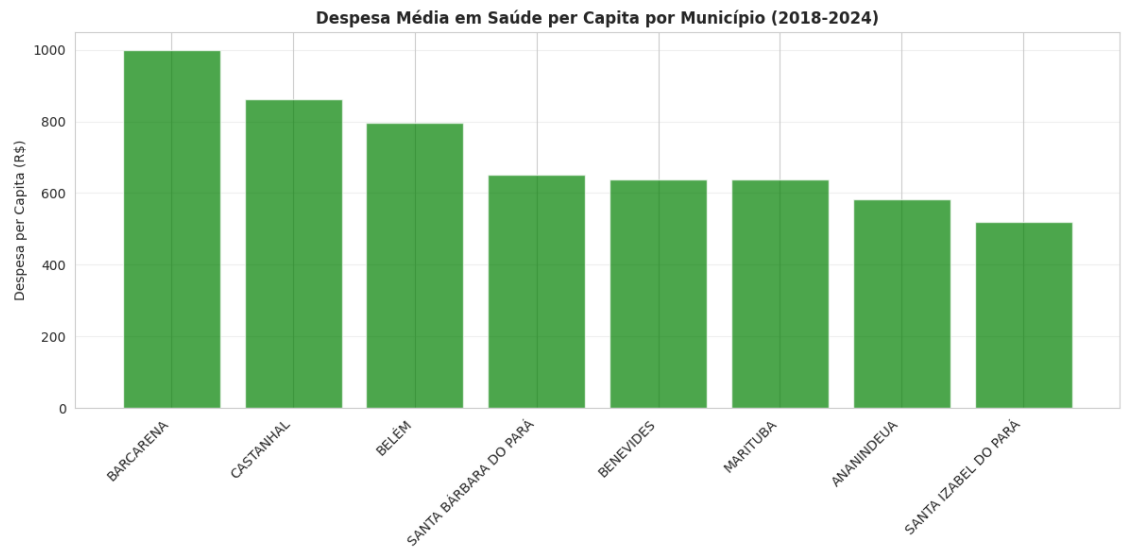
	despesa_saude_pc	pct_despesa_investimentos_saude \
municipio		
BARCARENA	999.63	4.30
CASTANHAL	861.51	2.36
BELEM	796.92	1.04
SANTA BÁRBARA DO PARÁ	651.91	3.29
BENEVIDES	637.34	5.39
MARITUBA	636.95	1.68
ANANINDEUA	582.57	2.51
SANTA IZABEL DO PARÁ	519.97	3.11

	pct_despesa_pessoal_saude \
municipio	
BARCARENA	56.97
CASTANHAL	50.01
BELEM	41.15
SANTA BÁRBARA DO PARÁ	54.24
BENEVIDES	49.07
MARITUBA	55.66
ANANINDEUA	41.59
SANTA IZABEL DO PARÁ	41.82

	pct_despesa_medicamentos_saude
municipio	
BARCARENA	9.62
CASTANHAL	1.92
BELEM	1.26
SANTA BÁRBARA DO PARÁ	9.52
BENEVIDES	2.10



MARITUBA	5.63
ANANINDEUA	3.20
SANTA IZABEL DO PARÁ	6.49

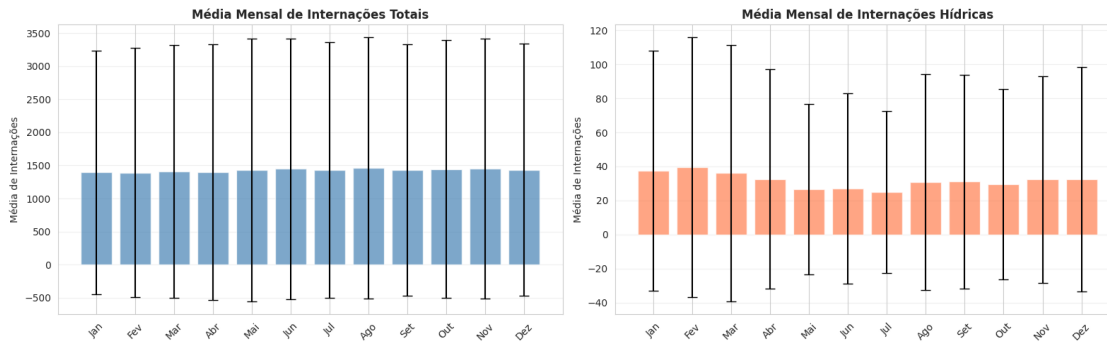


=====  
4. PADRÕES SAZONAIS  
=====

--- 4.1 SAZONALIDADE DAS INTERNAÇÕES ---

Média de internações por mês:

mes	internacoes_total		internacoes_hidricas	
	mean	std	mean	std
1	1393.97	1842.90	37.50	70.51
2	1391.34	1889.14	39.41	76.43
3	1413.19	1912.73	36.17	75.39
4	1398.20	1934.28	32.62	64.54
5	1432.28	1986.12	26.48	50.07
6	1447.39	1972.02	27.11	56.05
7	1428.66	1932.44	24.97	47.77
8	1460.72	1974.41	30.84	63.55
9	1428.07	1900.64	31.04	62.88
10	1443.68	1949.97	29.66	55.98
11	1454.86	1966.13	32.34	60.87
12	1435.27	1903.58	32.61	65.93



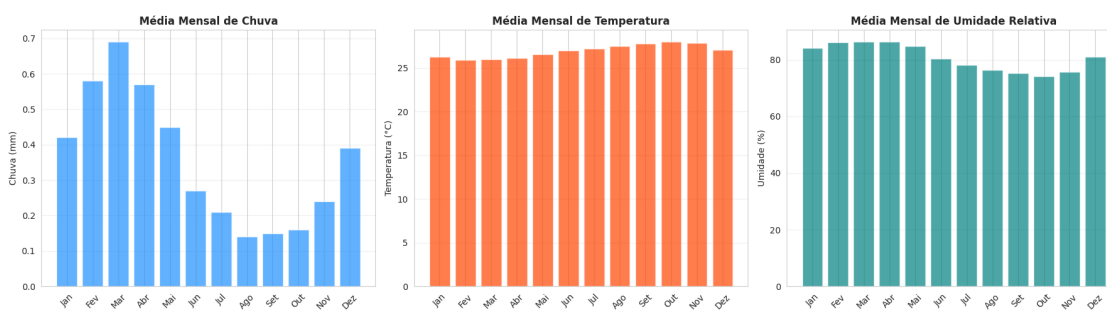
Top 3 meses com mais interações hídricas (média):

1. Fev: 39.41 interações
2. Jan: 37.50 interações
3. Mar: 36.17 interações

#### --- 4.2 PADRÕES CLIMÁTICOS ---

Média mensal de variáveis climáticas:

	chuva_mm	temp_c	umid_rel_pct
mes			
1	0.42	26.26	84.12
2	0.58	25.93	86.21
3	0.69	25.98	86.32
4	0.57	26.17	86.32
5	0.45	26.56	84.74
6	0.27	27.02	80.40
7	0.21	27.22	78.19
8	0.14	27.50	76.46
9	0.15	27.80	75.30
10	0.16	27.99	74.27
11	0.24	27.90	75.63
12	0.39	27.10	81.02



## 5. ANÁLISES CRUZADAS

### --- 5.1 DATASET INTEGRADO (ANUAL) ---

Dataset integrado criado: (64, 21)

Primeiras linhas:

	cod_mun	municipio	ano	internacoes_total	internacoes_hidricas	\
0	1500800	ANANINDEUA	2018	33625	1907	
1	1500800	ANANINDEUA	2019	31742	1869	
2	1500800	ANANINDEUA	2020	28648	1219	
3	1500800	ANANINDEUA	2021	24784	213	
4	1500800	ANANINDEUA	2022	23719	88	

	valor_total	valor_hidricas	populacao	municipio_siops	\
0	33358325.69	687622.26	525566.0	ANANINDEUA	
1	30968523.41	672328.37	530598.0	ANANINDEUA	
2	35739684.11	470035.90	535547.0	ANANINDEUA	
3	39538225.46	252334.72	540410.0	ANANINDEUA	
4	39168060.98	128402.19	NaN	ANANINDEUA	

	pct_transferencias_sus_recursos	...	despesa_saude_pc	\
0	29.50	...	442.94	
1	29.83	...	479.90	
2	34.44	...	626.14	
3	30.67	...	511.44	
4	27.65	...	593.56	

	pct_despesa_pessoal_saude	pct_despesa_medicamentos_saude	\
0	50.56	3.52	
1	46.20	1.77	
2	37.80	2.38	
3	37.06	4.15	
4	36.72	4.38	

	pct_despesa_terceiros_pj_saude	pct_despesa_investimentos_saude	\
0	30.09	0.81	
1	28.83	1.01	
2	29.52	1.59	
3	28.35	5.20	
4	31.25	2.53	

	pct_despesa_privado_sem_fins	pct_transferencias_sobre_despesa	\
0	0.81	71.98	
1	6.48	67.15	
2	9.42	75.60	
3	11.11	82.22	
4	1.24	75.37	

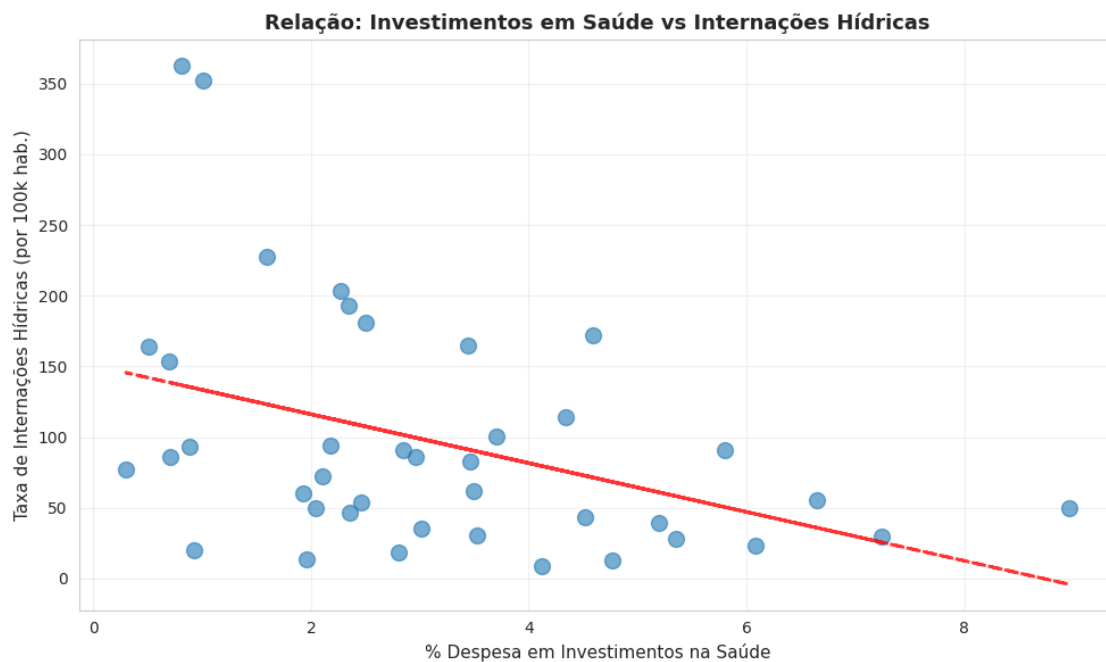
  

	pct_receita_propria_asps	taxa_internacoes_100k	taxa_hidricas_100k
0	15.73	6397.864398	362.846912
1	16.00	5982.306756	352.244072
2	18.66	5349.297074	227.617744
3	19.11	4586.147555	39.414519
4	16.08	NaN	NaN

[5 rows x 21 columns]

### --- 5.2 INVESTIMENTO vs INTERNAÇÕES ---

Correlação entre % investimentos e taxa de internações hídricas: -0.410



### --- 5.3 RANKING DE MUNICÍPIOS ---

Ranking por taxa de internações hídricas (média 2018-2024):

municipio	taxa_hidricas_100k	despesa_saude_pc	populacao
ANANINDEUA	172.47	582.57	524864.33
BELEM	136.95	796.92	1463397.33
SANTA IZABEL DO PARA	98.95	519.97	73792.00
MARITUBA	89.15	636.95	128129.00
CASTANHAL	75.24	861.51	204122.33
BARCARENA	56.41	999.63	129956.83
BENEVIDES	35.31	637.34	65021.17
SANTA BARBARA DO PARA	20.35	651.91	21643.50

=====

ANÁLISE EXPLORATÓRIA APROFUNDADA CONCLUÍDA!

=====

#### GRÁFICOS SALVOS:

1. distribuicoes\_internacoes.png
2. distribuicoes\_gastos.png
3. correlacao\_sih.png
4. correlacao\_siops.png
5. comparacao\_municipios.png
6. gastos\_municipios.png
7. sazonalidade.png
8. clima\_mensal.png
9. investimento\_vs\_internacoes.png

#### DATASET INTEGRADO:

- df\_integrado: pronto para análises avançadas e modelagem