

Análise de Concentração e Desertos de Leitos por Município

Identificação de Vazios Assistenciais no Brasil

Cieges - Brasil Estadual

2026-01-21

Table of contents

1	Introdução	1
1.1	Objetivo	1
1.2	Metodologia	1
1.3	Classificação de Municípios	1
2	Carregamento e Preparação dos Dados	1
2.1	Extração de UF e Região	2
2.2	Classificação de Intensidade dos Leitos	2
3	Análise por Município	3
3.1	Agregação de Indicadores	3
3.2	Presença de Leitos por Intensidade	3
3.3	Classificação de Vazios Assistenciais	4
4	Visualizações	4
4.1	Distribuição de Leitos por UF	4
4.1.1	Análise: Distribuição de Leitos e Vazios	5
4.2	Vazios Assistenciais por Região	5
4.2.1	Análise: Desigualdade Regional	6
4.3	Concentração (HHI) por UF	6
4.3.1	Análise: Concentração de Mercado	7
4.4	Relação: Leitos × Concentração	7
4.4.1	Análise: Relação Porte × Concentração	8
5	Rankings	8
5.1	Top 20 Municípios com Mais Leitos	8
5.2	Municípios sem UTI por UF	9
5.3	Municípios com Alta Concentração (HHI > 0.8)	9
6	Análise por Região	10
6.1	Perfil Regional Detalhado	10
6.2	Visualização Comparativa Regional	10
6.2.1	Análise: Perfil Comparativo Regional	11
7	Exportação dos Resultados	11
8	Resumo Executivo	12

1 Introdução

1.1 Objetivo

Esta análise identifica **municípios com déficit de leitos hospitalares** (desertos assistenciais) e avalia a **concentração da oferta** por tipo de complexidade. O objetivo é fornecer subsídios para planejamento e priorização de investimentos em saúde.

1.2 Metodologia

A análise utiliza os seguintes indicadores:

Indicador	Descrição	Interpretação
Leitos per capita	Leitos / 1.000 hab	OMS recomenda 3-5 leitos/1.000 hab
Cobertura UTI	Presença de leitos intensivos	Binário: Sim/Não
Índice HHI	Concentração em poucos estabelecimentos	0-1 (maior = mais concentrado)
Razão SUS	% de leitos disponíveis ao SUS	0-100%

1.3 Classificação de Municípios

CLASSIFICAÇÃO DE VAZIOS ASSISTENCIAIS	
COMPLETO	Possui leitos de todas as complexidades
PARCIAL	Possui leitos, mas sem alta complexidade
BÁSICO	Possui apenas leitos de baixa complexidade
DESERTO	Não possui nenhum leito hospitalar

2 Carregamento e Preparação dos Dados

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

# Configuração visual
```

```
plt.rcParams['figure.dpi'] = 100
plt.rcParams['font.size'] = 10
sns.set_style("whitegrid")

# Carregar dados tratados
df = pd.read_csv('arq2_tratado.csv', sep=';', encoding='latin1', low_memory=False)

# Carregar taxonomia para classificação de intensidade
df_tax = pd.read_csv('arq5_taxonomia_leitos.csv', sep=';', encoding='utf-8')

print("="*70)
print("DADOS CARREGADOS")
print("="*70)
print(f"Registros de leitos: {len(df):,}")
print(f"Total de leitos: {df['qt_exist'].sum():,}")
print(f"Estabelecimentos (CNES): {df['cnes'].nunique():,}")
print(f"Municípios: {df['codufmun'].nunique():,}")
```

```
=====
DADOS CARREGADOS
=====
Registros de leitos: 49,804
Total de leitos: 535,133
Estabelecimentos (CNES): 9,072
Municípios: 3,597
```

2.1 Extração de UF e Região

```
# Extrair código da UF (2 primeiros dígitos do código IBGE)
df['cod_uf'] = df['codufmun'].astype(str).str[:2].astype(int)

# Mapeamento UF
UF_NOME = {
    11: 'RO', 12: 'AC', 13: 'AM', 14: 'RR', 15: 'PA', 16: 'AP', 17: 'TO',
    21: 'MA', 22: 'PI', 23: 'CE', 24: 'RN', 25: 'PB', 26: 'PE', 27: 'AL', 28: 'SE', 29: 'BA',
    31: 'MG', 32: 'ES', 33: 'RJ', 35: 'SP',
    41: 'PR', 42: 'SC', 43: 'RS',
    50: 'MS', 51: 'MT', 52: 'GO', 53: 'DF'
}

REGIAO = {
    11: 'Norte', 12: 'Norte', 13: 'Norte', 14: 'Norte', 15: 'Norte', 16: 'Norte', 17: 'Norte',
    21: 'Nordeste', 22: 'Nordeste', 23: 'Nordeste', 24: 'Nordeste', 25: 'Nordeste',
    26: 'Nordeste', 27: 'Nordeste', 28: 'Nordeste', 29: 'Nordeste',
    31: 'Sudeste', 32: 'Sudeste', 33: 'Sudeste', 35: 'Sudeste',
    41: 'Sul', 42: 'Sul', 43: 'Sul',
    50: 'Centro-Oeste', 51: 'Centro-Oeste', 52: 'Centro-Oeste', 53: 'Centro-Oeste'
}

df['uf'] = df['cod_uf'].map(UF_NOME)
df['regiao'] = df['cod_uf'].map(REGIAO)

print("\nDISTRIBUIÇÃO POR REGIÃO")
print("="*50)
print(df.groupby('regiao')['qt_exist'].sum().sort_values(ascending=False))
```

```
DISTRIBUIÇÃO POR REGIÃO
=====
regiao
Sudeste      221530
Nordeste     142405
Sul           83637
Centro-Oeste  47677
Norte        39884
Name: qt_exist, dtype: int64
```

2.2 Classificação de Intensidade dos Leitos

```
# Classificar leitos por intensidade (usando lógica da taxonomia)
def classificar_intensidade(row):
    """Classifica o leito quanto à intensidade do cuidado."""
    co = row['co_leito']

    # INTENSIVO (UTI)
    LEITOS_UTI = [74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86]
    if co in LEITOS_UTI:
        return 'INTENSIVO'

    # SEMI-INTENSIVO (UCI)
    LEITOS_UCI = [65, 92, 93, 94, 95, 96]
    if co in LEITOS_UCI:
        return 'SEMI_INTENSIVO'

    # ALTA COMPLEXIDADE
    LEITOS_ALTA = [67, 71, 88, 89, 90, 91]
    if co in LEITOS_ALTA:
        return 'ALTA_COMPLEXIDADE'

    # MÉDIA/BAIXA COMPLEXIDADE
    return 'MEDIA_BAIXA'

df['intensidade'] = df.apply(classificar_intensidade, axis=1)
```

```
print("\nDISTRIBUIÇÃO POR INTENSIDADE")
print("="*50)
intensidade_resumo = df.groupby('intensidade')['qt_exist'].agg(['count', 'sum'])
intensidade_resumo.columns = ['Registros', 'Leitos']
intensidade_resumo['%'] = (intensidade_resumo['Leitos'] / intensidade_resumo['Leitos'].sum() * 100).round(1)
print(intensidade_resumo.sort_values('Leitos', ascending=False))
```

DISTRIBUIÇÃO POR INTENSIDADE			
=====			
	Registros	Leitos	%
intensidade			
MEDIA_BAIXA	43500	455547	85.1
INTENSIVO	4365	65273	12.2
SEMI_INTENSIVO	1545	12038	2.2
ALTA_COMPLEXIDADE	394	2275	0.4

3 Análise por Município

3.1 Agregação de Indicadores

```
# Agregar por município
def calcular_hhi(series):
    """Calcula Índice Herfindahl-Hirschman de concentração."""
    total = series.sum()
    if total == 0:
        return 0
    shares = series / total
    return (shares ** 2).sum()

# Agregação básica por município
mun_basico = df.groupby(['codufmun', 'uf', 'regiao']).agg({
    'cnes': 'nunique',
    'qt_exist': 'sum',
    'qt_sus': 'sum',
    'qt_nsus': 'sum'
}).reset_index()

mun_basico.columns = ['codufmun', 'uf', 'regiao', 'n_estabelecimentos',
                      'total_leitos', 'leitos_sus', 'leitos_nsus']

# Calcular HHI por município
hhi_por_mun = df.groupby('codufmun').apply(
    lambda x: calcular_hhi(x.groupby('cnes')['qt_exist'].sum())
).reset_index()
hhi_por_mun.columns = ['codufmun', 'hhi']

# Merge
municipios = mun_basico.merge(hhi_por_mun, on='codufmun')

# Calcular métricas derivadas
municipios['pct_sus'] = (municipios['leitos_sus'] / municipios['total_leitos'] * 100).round(1)
municipios['media_leitos_estab'] = (municipios['total_leitos'] / municipios['n_estabelecimentos']).round(1)

print("="*70)
print("AGREGAÇÃO POR MUNICÍPIO")
print("="*70)
print(f"Municípios com leitos: {len(municipios):,}")
print(f"Total de leitos: {municipios['total_leitos'].sum():,}")
print(f"Média de leitos por município: {municipios['total_leitos'].mean():.1f}")
print(f"Mediana de leitos por município: {municipios['total_leitos'].median():.1f}")
```

```
=====
AGREGAÇÃO POR MUNICÍPIO
=====

Municípios com leitos: 3,597
Total de leitos: 535,133
Média de leitos por município: 148.8
Mediana de leitos por município: 35.0
```

3.2 Presença de Leitos por Intensidade

```
# Verificar presença de cada tipo de intensidade por município
intensidade_pivot = df.groupby(['codufmun', 'intensidade'])['qt_exist'].sum().unstack(fill_value=0)
intensidade_pivot.columns = ['leitos_' + col.lower() for col in intensidade_pivot.columns]
intensidade_pivot = intensidade_pivot.reset_index()

# Merge com dados do município
municipios = municipios.merge(intensidade_pivot, on='codufmun', how='left')

# Preencher NaN com 0
for col in ['leitos_intensivo', 'leitos_semi_intensivo', 'leitos_alta_complexidade', 'leitos_media_baixa']:
    if col in municipios.columns:
        municipios[col] = municipios[col].fillna(0).astype(int)
    else:
        municipios[col] = 0

# Flags de presença
municipios['tem_uti'] = municipios['leitos_intensivo'] > 0
municipios['tem_uci'] = municipios['leitos_semi_intensivo'] > 0
municipios['tem_alta'] = municipios['leitos_alta_complexidade'] > 0
```

```
print("\nPRESENÇA DE LEITOS POR INTENSIDADE")
print("="*50)
print(f"Municípios com UTI: {municipios['tem_uti'].sum():,} ({municipios['tem_uti'].mean()*100:.1f}%)")
print(f"Municípios com UCI: {municipios['tem_uci'].sum():,} ({municipios['tem_uci'].mean()*100:.1f}%)")
print(f"Municípios com Alta Complexidade: {municipios['tem_alta'].sum():,} ({municipios['tem_alta'].mean()*100:.1f}%)")
```

PRESENÇA DE LEITOS POR INTENSIDADE
=====

Municípios com UTI: 711 (19.8%)
Municípios com UCI: 601 (16.7%)
Municípios com Alta Complexidade: 112 (3.1%)

3.3 Classificação de Vazios Assistenciais

```
# Classificar municípios
def classificar_vazio(row):
    """
    Classifica o município quanto ao vazio assistencial.

    COMPLETO: Tem UTI ou UCI (alta/semi intensivo)
    PARCIAL: Tem leitos, mas sem UTI/UCI
    BÁSICO: Tem apenas leitos de média/baixa complexidade
    """
    if row['tem_uti'] or row['tem_uci']:
        return 'COMPLETO'
    elif row['tem_alta']:
        return 'PARCIAL'
    elif row['total_leitos'] > 0:
        return 'BASICO'
    else:
        return 'DESERTO'

municipios['classificacao_vazio'] = municipios.apply(classificar_vazio, axis=1)

# Resumo
print("\nCLASSIFICAÇÃO DE VAZIOS ASSISTENCIAIS")
print("="*70)
vazio_resumo = municipios.groupby('classificacao_vazio').agg({
    'codufmun': 'count',
    'total_leitos': 'sum',
    'n_estabelecimentos': 'sum'
}).rename(columns={'codufmun': 'Municipios'})

vazio_resumo['%_Municipios'] = (vazio_resumo['Municipios'] / vazio_resumo['Municipios'].sum() * 100).round(1)
vazio_resumo['%_Leitos'] = (vazio_resumo['total_leitos'] / vazio_resumo['total_leitos'].sum() * 100).round(1)
vazio_resumo = vazio_resumo.sort_values('Municipios', ascending=False)
print(vazio_resumo)
```

CLASSIFICAÇÃO DE VAZIOS ASSISTENCIAIS			
=====			
	Municipios	total_leitos	n_estabelecimentos \
classificacao_vazio			
BASICO	2678	90990	3147
COMPLETO	912	443768	5915
PARCIAL	7	375	10

	%_Municipios	%_Leitos
classificacao_vazio		
BASICO	74.5	17.0
COMPLETO	25.4	82.9
PARCIAL	0.2	0.1

4 Visualizações

4.1 Distribuição de Leitos por UF

```
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

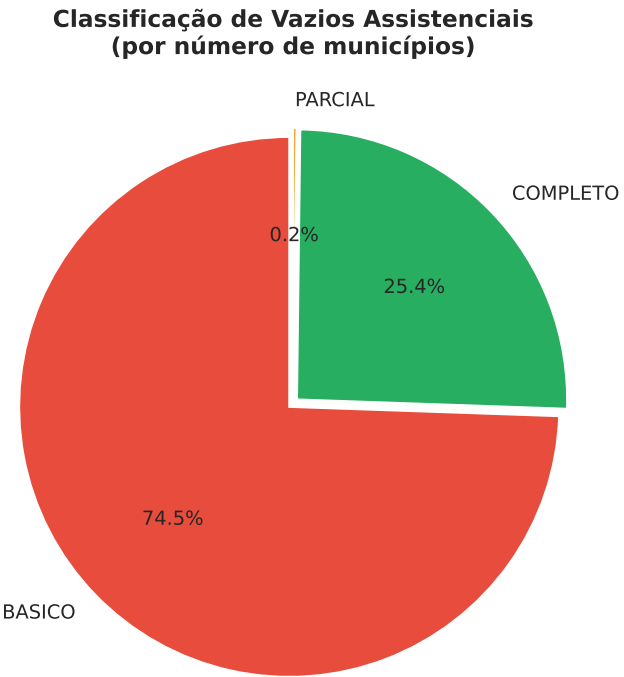
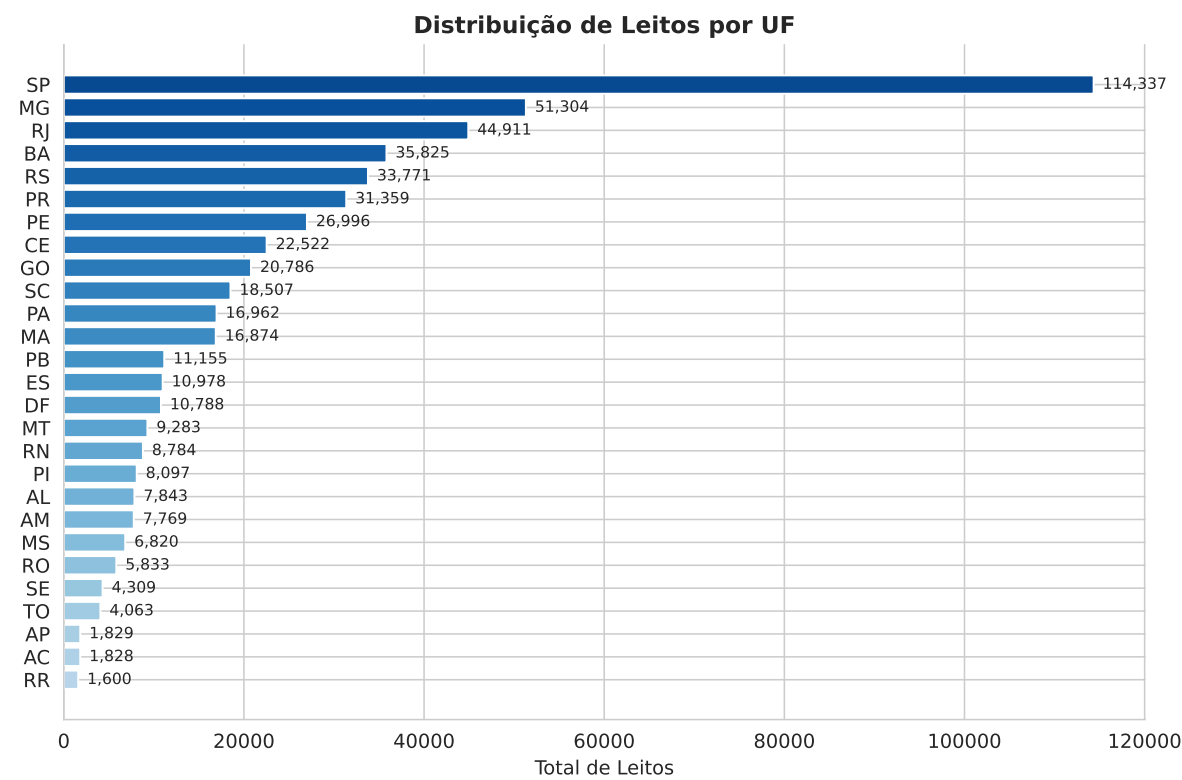
# Leitos por UF
leitos_uf = municipios.groupby('uf')['total_leitos'].sum().sort_values(ascending=True)
colors_uf = plt.cm.Blues(np.linspace(0.3, 0.9, len(leitos_uf)))

axes[0].barh(leitos_uf.index, leitos_uf.values, color=colors_uf)
axes[0].set_xlabel('Total de Leitos')
axes[0].set_title('Distribuição de Leitos por UF', fontweight='bold', fontsize=12)
axes[0].spines['top'].set_visible(False)
axes[0].spines['right'].set_visible(False)
for i, v in enumerate(leitos_uf.values):
    axes[0].text(v + 1000, i, f'{v:,}', va='center', fontsize=8)

# Municípios por classificação de vazio
vazio_plot = municipios['classificacao_vazio'].value_counts()
colors_vazio = {'COMPLETO': '#27ae60', 'PARCIAL': '#f39c12', 'BASICO': '#e74c3c', 'DESERTO': '#2c3e50'}
axes[1].pie(vazio_plot.values, labels=vazio_plot.index, autopct='%1.1f%%',
            colors=[colors_vazio.get(x, '#95a5a6') for x in vazio_plot.index],
            explode=[0.02]*len(vazio_plot), startangle=90)
axes[1].set_title('Classificação de Vazios Assistenciais\n(por número de municípios)', fontweight='bold', fontsize=12)

plt.tight_layout()
```

plt.show()



4.1.1 Análise: Distribuição de Leitos e Vazios

i Principais Achados

Distribuição por UF:

- **São Paulo** concentra a maior oferta de leitos do país, seguido por Minas Gerais e Rio de Janeiro, refletindo a concentração populacional e econômica do Sudeste.
- Estados do Norte e Centro-Oeste apresentam menor oferta absoluta, mas isso deve ser relativizado pela menor população.
- A disparidade entre o maior (SP) e o menor estado é de mais de 10x, indicando forte assimetria regional.

Classificação de Vazios:

- A maioria dos municípios brasileiros possui cobertura **BÁSICA** ou **PARCIAL**, indicando que a alta complexidade (UTI/UCI) está concentrada em poucos municípios.
- Municípios classificados como **COMPLETO** (com UTI ou UCI) representam uma minoria, geralmente capitais e cidades de médio/grande porte.
- Esta distribuição sugere que a maioria da população precisa se deslocar para acessar leitos de alta complexidade.

4.2 Vazios Assistenciais por Região

```
# Cruzamento: Região × Classificação
matriz_regiao = pd.crosstab(
    municipios['regiao'],
    municipios['classificacao_vazio'],
    margins=True,
    margins_name='TOTAL'
)

# Reordenar colunas
ordem_cols = ['COMPLETO', 'PARCIAL', 'BASICO', 'DESERTO', 'TOTAL']
matriz_regiao = matriz_regiao[[c for c in ordem_cols if c in matriz_regiao.columns]]

print("\nMATRIZ: REGIÃO × CLASSIFICAÇÃO DE VAZIO")
print("="*70)
print(matriz_regiao)

# Visualização
fig, ax = plt.subplots(figsize=(12, 6))

matriz_plot = matriz_regiao.drop('TOTAL', axis=0).drop('TOTAL', axis=1)
matriz_plot_pct = matriz_plot.div(matriz_plot.sum(axis=1), axis=0) * 100

matriz_plot_pct.plot(kind='barh', stacked=True, ax=ax,
    color=['#27ae60', '#f39c12', '#e74c3c', '#2c3e50'])

ax.set_xlabel('% de Municípios')
ax.set_title('Classificação de Vazios Assistenciais por Região', fontweight='bold', fontsize=12)
ax.legend(title='Classificação', bbox_to_anchor=(1.02, 1), loc='upper left')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Adicionar linha de 50%
ax.axvline(x=50, color='gray', linestyle='--', alpha=0.5)

plt.tight_layout()
plt.show()
```

MATRIZ: REGIÃO × CLASSIFICAÇÃO DE VAZIO				
	COMPLETO	PARCIAL	BÁSICO	TOTAL
regiao				
Centro-Oeste	79	1	281	361
Nordeste	214	3	1138	1355
Norte	83	3	247	333
Sudeste	347	0	567	914
Sul	189	0	445	634

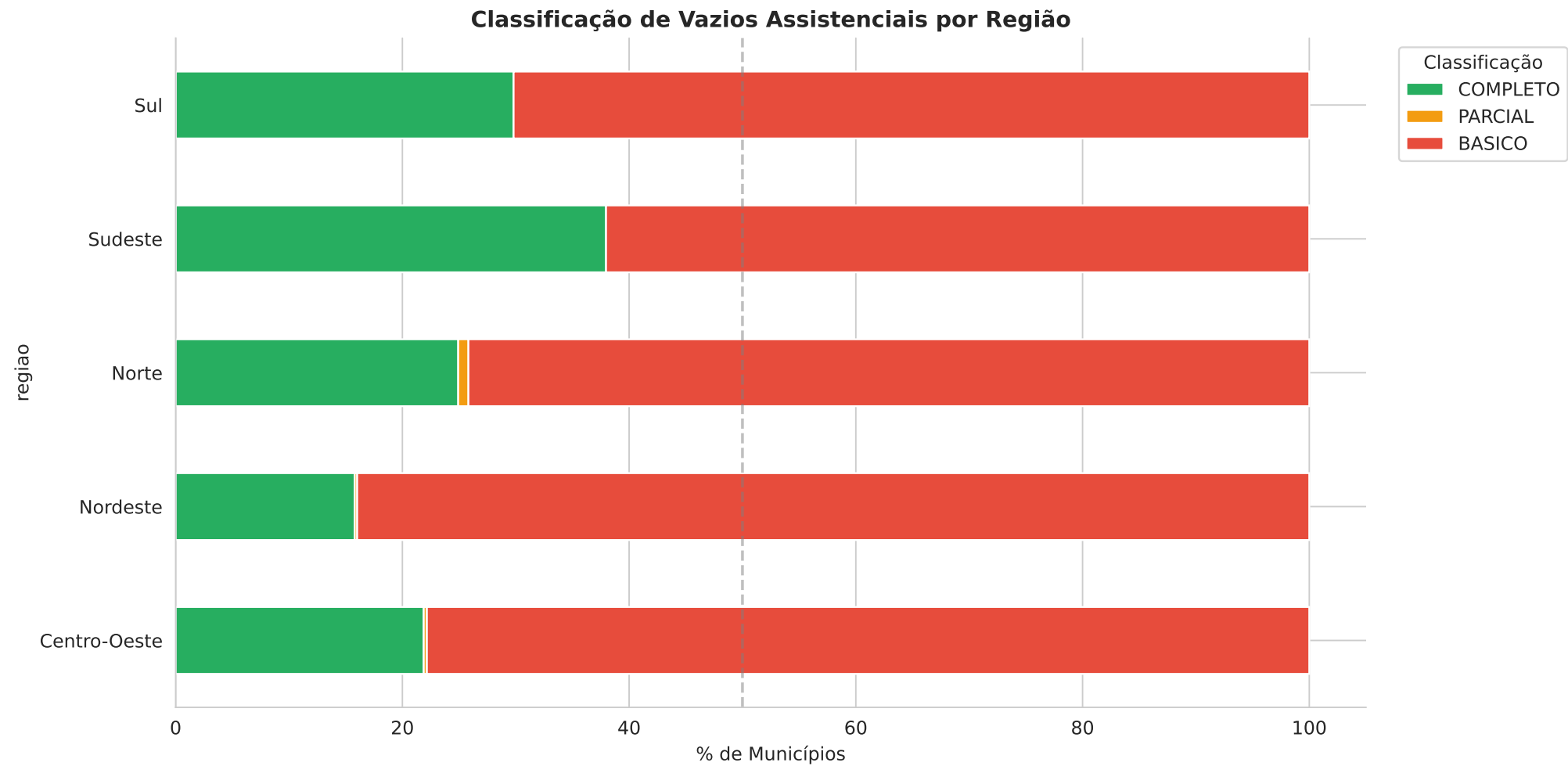
TOTAL

912


7

2678

3597



4.2.1 Análise: Desigualdade Regional

 Disparidades Regionais Críticas

Norte e Nordeste:

- Apresentam a **maior proporção de municípios com cobertura BÁSICA**, ou seja, sem leitos de média/alta complexidade.
- A distância geográfica até centros com UTI pode ultrapassar centenas de quilômetros, comprometendo o atendimento de emergências.
- Estas regiões devem ser **prioridade para investimentos** em expansão de leitos de alta complexidade.

Sul e Sudeste:

- Maior proporção de municípios com cobertura **COMPLETA**, refletindo maior desenvolvimento da rede hospitalar.
- Ainda assim, municípios menores nestas regiões também apresentam vazios assistenciais.

Implicação para Política Pública:

- A regionalização da saúde (criação de redes de referência) é essencial para garantir acesso à alta complexidade.
- Investimentos em SAMU e transporte aeromédico podem mitigar parcialmente os vazios em regiões remotas.

4.3 Concentração (HHI) por UF

```
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

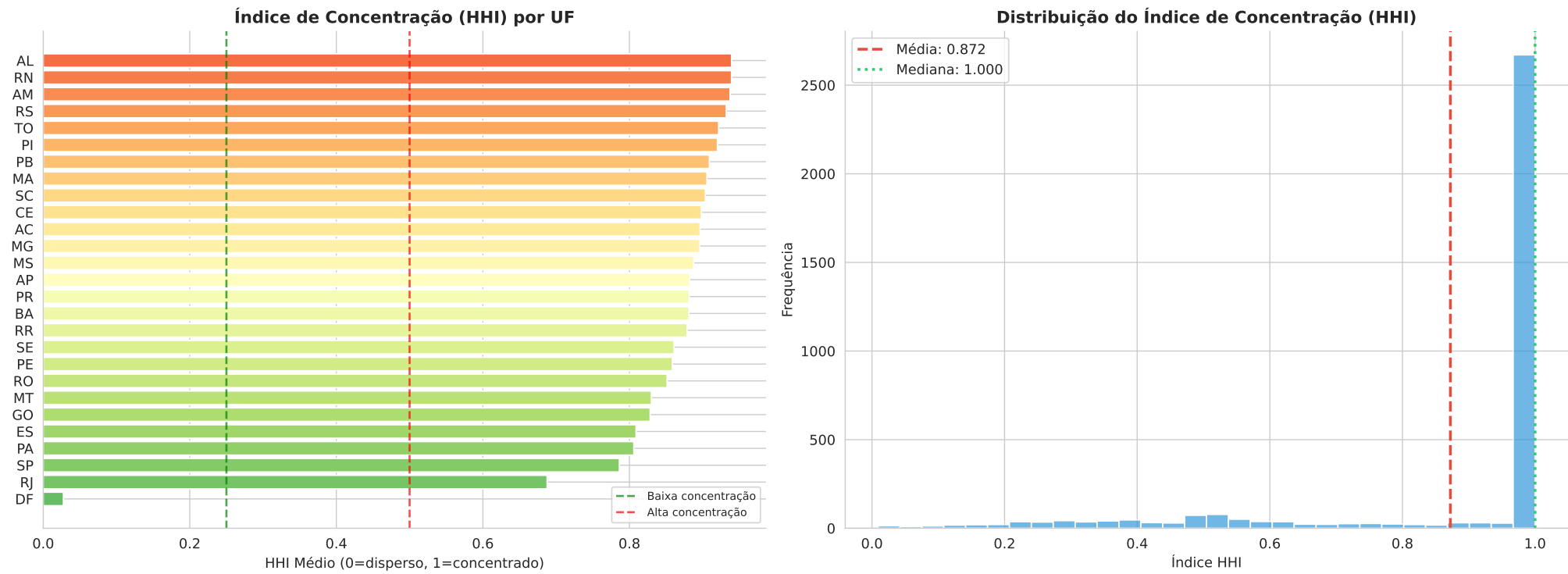
# HHI médio por UF
hhi_uf = municipios.groupby('uf')['hhi'].mean().sort_values(ascending=True)

colors_hhi = plt.cm.RdYlGn_r(np.linspace(0.2, 0.8, len(hhi_uf)))
axes[0].barh(hhi_uf.index, hhi_uf.values, color=colors_hhi)
axes[0].set_xlabel('HHI Médio (0=disperso, 1=concentrado)')
axes[0].set_title('Índice de Concentração (HHI) por UF', fontweight='bold', fontsize=12)
axes[0].axvline(x=0.25, color='green', linestyle='--', alpha=0.7, label='Baixa concentração')
axes[0].axvline(x=0.5, color='red', linestyle='--', alpha=0.7, label='Alta concentração')
axes[0].legend(fontsize=8)
axes[0].spines['top'].set_visible(False)
axes[0].spines['right'].set_visible(False)

# Distribuição do HHI
axes[1].hist(municipios['hhi'], bins=30, color='#3498db', edgecolor='white', alpha=0.7)
axes[1].axvline(x=municipios['hhi'].mean(), color='#e74c3c', linestyle='--',
               linewidth=2, label=f"Média: {municipios['hhi'].mean():.3f}")
axes[1].axvline(x=municipios['hhi'].median(), color='#2ecc71', linestyle=':',
               linewidth=2, label=f"Mediana: {municipios['hhi'].median():.3f}")
axes[1].set_xlabel('Índice HHI')
axes[1].set_ylabel('Frequência')
axes[1].set_title('Distribuição do Índice de Concentração (HHI)', fontweight='bold', fontsize=12)
axes[1].legend()
axes[1].spines['top'].set_visible(False)
axes[1].spines['right'].set_visible(False)

plt.tight_layout()
plt.show()

print("\nINTERPRETAÇÃO DO HHI:")
print("  • HHI < 0.15: Mercado competitivo (baixa concentração)")
print("  • HHI 0.15-0.25: Concentração moderada")
print("  • HHI > 0.25: Alta concentração")
print(f"\n  Média nacional: {municipios['hhi'].mean():.3f}")
print(f"  {(municipios['hhi'] > 0.5).sum()} municípios com HHI > 0.5 (monopólio/duopólio)")
```

INTERPRETAÇÃO DO HHI:

- HHI < 0.15: Mercado competitivo (baixa concentração)
- HHI 0.15-0.25: Concentração moderada
- HHI > 0.25: Alta concentração

Média nacional: 0.872
3167 municípios com HHI > 0.5 (monopólio/duopólio)

4.3.1 Análise: Concentração de Mercado

! Riscos da Alta Concentração

O que o HHI revela:

- Um **HHI alto** (>0.5) indica que os leitos do município estão concentrados em poucos estabelecimentos (monopólio ou duopólio).
- Isso representa **risco de desassistência** caso o estabelecimento dominante enfrente problemas (fechamento, greve, desastre).
- Municípios com HHI próximo de 1.0 possuem **um único hospital**, criando dependência total.

Padrão observado:

- A maioria dos municípios brasileiros apresenta **alta concentração**, pois são pequenos e possuem apenas 1-2 hospitais.
- Estados com maior pulverização (mais hospitais por município) tendem a ter HHI menor.
- O histograma mostra distribuição bimodal: muitos municípios com HHI muito alto (monopólio) e alguns com HHI baixo (mercado competitivo).

Recomendação:

- Municípios com HHI > 0.8 e população significativa devem ser avaliados para **expansão de rede** ou **fortalecimento de consórcios intermunicipais**.

4.4 Relação: Leitos × Concentração

```
fig, ax = plt.subplots(figsize=(12, 8))

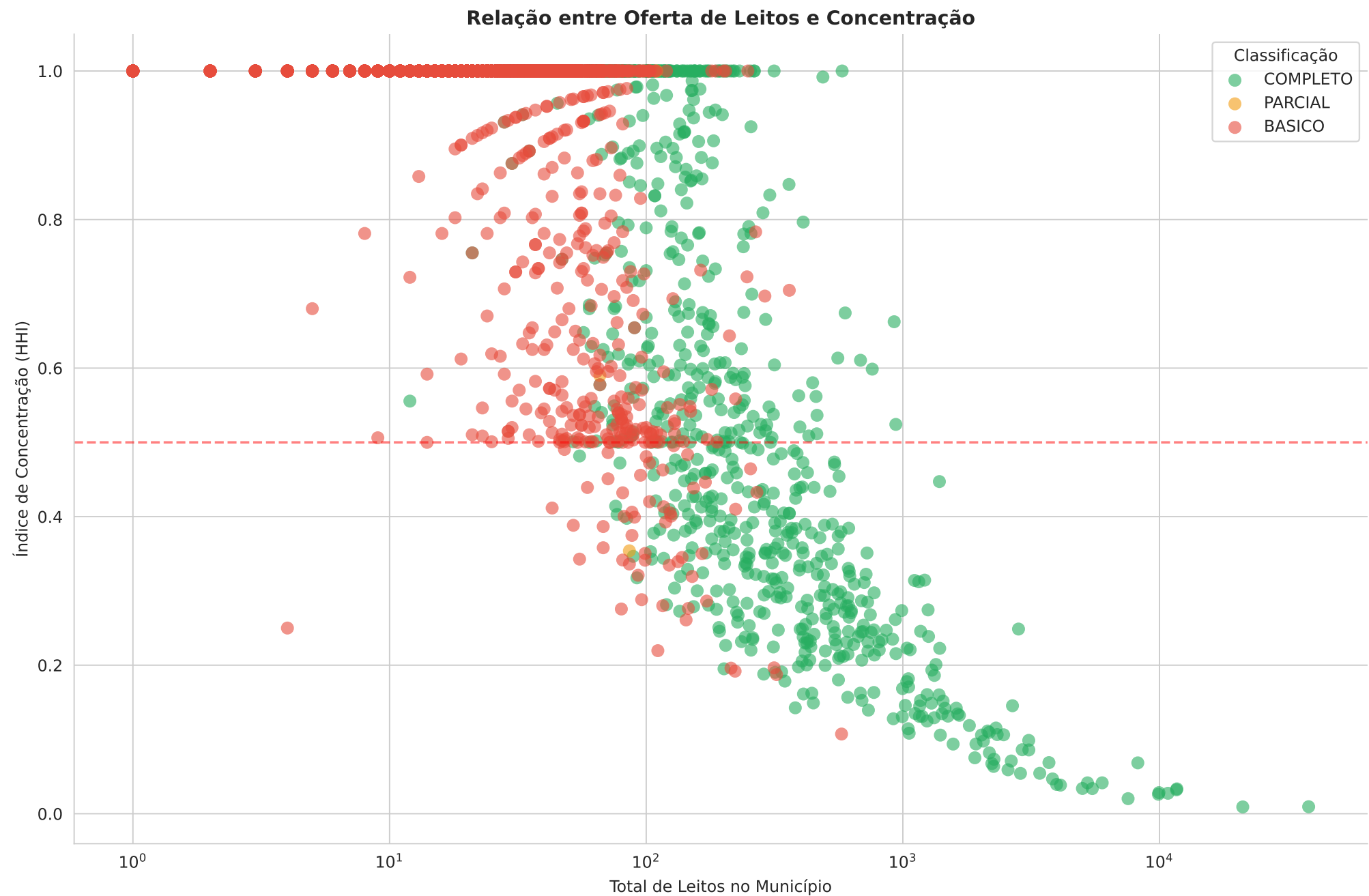
# Scatter plot com cores por classificação
colors_scatter = {'COMPLETO': '#27ae60', 'PARCIAL': '#f39c12', 'BASICO': '#e74c3c', 'DESERTO': '#2c3e50'}

for classif in ['COMPLETO', 'PARCIAL', 'BASICO']:
    subset = municipios[municipios['classificacao_vazio'] == classif]
    ax.scatter(subset['total_leitos'], subset['hhi'],
               c=colors_scatter[classif], label=classif, alpha=0.6, s=50)

ax.set_xlabel('Total de Leitos no Município')
ax.set_ylabel('Índice de Concentração (HHI)')
ax.set_title('Relação entre Oferta de Leitos e Concentração', fontweight='bold', fontsize=12)
ax.set_xscale('log')
ax.legend(title='Classificação')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# Linhas de referência
ax.axhline(y=0.5, color='red', linestyle='--', alpha=0.5, label='Alta concentração')

plt.tight_layout()
plt.show()
```



4.4.1 Análise: Relação Porte × Concentração

Padrão Identificado

Correlação negativa clara:

- Municípios com **mais leitos** tendem a ter **menor concentração** (HHI baixo), pois possuem múltiplos estabelecimentos competindo.
- Municípios com **poucos leitos** apresentam **alta concentração** (HHI alto), geralmente com apenas 1 hospital.

Interpretação por classificação:

- COMPLETO (verde):** Concentrados no quadrante inferior direito (muitos leitos, baixa concentração). São os grandes centros urbanos.
- PARCIAL (amarelo):** Distribuídos no meio do gráfico, com porte e concentração intermediários.
- BÁSICO (vermelho):** Concentrados no quadrante superior esquerdo (poucos leitos, alta concentração). São municípios pequenos com hospital único.

Insight estratégico:

- A escala logarítmica no eixo X evidencia que a transição de “alta concentração” para “baixa concentração” ocorre aproximadamente a partir de **100-200 leitos** no município.
- Municípios abaixo deste limiar dificilmente terão mercado competitivo e dependem de regulação e consórcios.

5 Rankings

5.1 Top 20 Municípios com Mais Leitos

```
top_leitos = municipios.nlargest(20, 'total_leitos')[
    ['codufmun', 'uf', 'total_leitos', 'leitos_sus', 'pct_sus',
     'n_estabelecimentos', 'tem_uti', 'hhi', 'classificacao_vazio']
]
top_leitos['codufmun'] = top_leitos['codufmun'].astype(str)

print("TOP 20 MUNICÍPIOS COM MAIS LEITOS")
print("="*70)
top_leitos
```

TOP 20 MUNICÍPIOS COM MAIS LEITOS

=====									
	codufmun	uf	total_leitos	leitos_sus	pct_sus	n_estabelecimentos	tem_uti	hhi	classificacao_vazio
2554	355030	SP	38123	18238	47.8	342	True	0.009451	COMPLETO
2222	330455	RJ	21109	9811	46.5	288	True	0.009187	COMPLETO
1209	261160	PE	11691	7289	62.3	104	True	0.033493	COMPLETO
1722	310620	MG	11647	6414	55.1	114	True	0.032195	COMPLETO
3596	530010	DF	10788	5574	51.7	111	True	0.027547	COMPLETO
1616	292740	BA	9952	6563	65.9	129	True	0.028439	COMPLETO
715	230440	CE	9906	6441	65.0	109	True	0.026255	COMPLETO
3142	431490	RS	8233	4871	59.2	44	True	0.068492	COMPLETO
3479	520870	GO	7544	3875	51.4	132	True	0.020315	COMPLETO
2661	410690	PR	5980	3128	52.3	72	True	0.041616	COMPLETO
507	211130	MA	5473	3892	71.1	61	True	0.033877	COMPLETO
161	150140	PA	5242	3135	59.8	64	True	0.041576	COMPLETO
106	130260	AM	5009	3551	70.9	67	True	0.034004	COMPLETO
910	240810	RN	4113	2631	64.0	61	True	0.038520	COMPLETO

	codufmun	uf	total_leitos	leitos_sus	pct_sus	n_estabelecimentos	tem_uti	hhi	classificacao_vazio
1025	250750	PB	3974	2428	61.1	69	True	0.039467	COMPLETO
1284	270430	AL	3831	2520	65.8	49	True	0.046866	COMPLETO
2306	350950	SP	3709	1722	46.4	50	True	0.068854	COMPLETO
654	221100	PI	3415	2354	68.9	45	True	0.054450	COMPLETO
1315	280030	SE	3097	1918	61.9	51	True	0.085911	COMPLETO
3254	500270	MS	3094	1826	59.0	37	True	0.098742	COMPLETO

5.2 Municípios sem UTI por UF

```
# Municípios sem UTI
sem_uti = municipios[~municipios['tem_uti']]

sem_uti_por_uf = sem_uti.groupby('uf').agg({
    'codufmun': 'count',
    'total_leitos': 'sum'
}).rename(columns={'codufmun': 'Municipios_sem_UTI'})

# Total de municípios por UF
total_por_uf = municipios.groupby('uf')['codufmun'].count()
sem_uti_por_uf['Total_Municipios'] = total_por_uf
sem_uti_por_uf['%_sem_UTI'] = (sem_uti_por_uf['Municipios_sem_UTI'] / sem_uti_por_uf['Total_Municipios'] * 100).round(1)
sem_uti_por_uf = sem_uti_por_uf.sort_values('%_sem_UTI', ascending=False)

print("\nMUNICÍPIOS SEM UTI POR UF")
print("="*70)
sem_uti_por_uf
```

MUNICÍPIOS SEM UTI POR UF

=====				
	Municipios_sem_UTI	total_leitos	Total_Municipios	%_sem_UTI
uf				
RN	129	2504	139	92.8
PI	114	2566	123	92.7
AM	57	2294	62	91.9
RR	11	254	12	91.7
MA	182	6430	204	89.2
BA	313	11361	351	89.2
AC	16	498	18	88.9
PB	90	2246	103	87.4
TO	40	1112	46	87.0
CE	154	6044	177	87.0
GO	167	5240	193	86.5
MS	61	1690	71	85.9
PE	157	6083	183	85.8
AP	11	308	13	84.6
RO	43	1257	51	84.3
AL	43	1381	52	82.7
PR	200	7699	242	82.6
SE	19	519	23	82.6
MT	79	2386	96	82.3
PA	107	4620	131	81.7
MG	319	10495	415	76.9
SC	116	4956	154	75.3
RS	175	8609	238	73.5
ES	39	1796	56	69.6
SP	207	9616	356	58.1
RJ	37	1554	87	42.5

5.3 Municípios com Alta Concentração (HHI > 0.8)

```
# Municípios com monopólio (HHI muito alto)
monopolio = municipios[municipios['hhi'] > 0.8].sort_values('total_leitos', ascending=False)

print(f"\nMUNICÍPIOS COM ALTA CONCENTRAÇÃO (HHI > 0.8): {len(monopolio)}")
print("="*70)
print("(Leitos concentrados em 1-2 estabelecimentos)")

monopolio_display = monopolio.head(20)[
    ['codufmun', 'uf', 'total_leitos', 'n_estabelecimentos', 'hhi', 'pct_sus', 'classificacao_vazio']
]
monopolio_display['codufmun'] = monopolio_display['codufmun'].astype(str)
monopolio_display
```

MUNICÍPIOS COM ALTA CONCENTRAÇÃO (HHI > 0.8): 2801							
=====							
(Leitos concentrados em 1-2 estabelecimentos)							
	codufmun	uf	total_leitos	n_estabelecimentos	hhi	pct_sus	classificacao_vazio
2633	410400	PR	580	1	1.000000	80.0	COMPLETO
1273	270230	AL	488	3	0.991828	98.2	COMPLETO
2813	412570	PR	360	2	0.847222	55.3	COMPLETO
2169	330060	RJ	315	1	1.000000	72.4	COMPLETO
2401	352310	SP	303	4	0.832958	98.0	COMPLETO
3185	431870	RS	285	3	0.809074	89.1	COMPLETO
3082	430920	RS	264	1	1.000000	80.3	COMPLETO

	codufmun	uf	total_leitos	n_estabelecimentos	hhi	pct_sus	classificacao_vazio
3017	430210	RS	263	1	1.000000	49.0	COMPLETO
1918	313940	MG	256	2	0.924927	70.7	COMPLETO
2352	351570	SP	255	1	1.000000	88.2	COMPLETO
2412	352450	SP	249	1	1.000000	69.5	BASICO
3002	430040	RS	231	1	1.000000	80.5	COMPLETO
2817	412625	PR	221	1	1.000000	71.9	COMPLETO
1091	260110	PE	218	1	1.000000	88.5	COMPLETO
3360	510622	MT	213	1	1.000000	77.5	COMPLETO
3026	430300	RS	207	1	1.000000	68.6	COMPLETO
3180	431830	RS	205	1	1.000000	78.5	COMPLETO
406	210467	MA	204	1	1.000000	98.0	BASICO
3104	431140	RS	201	1	1.000000	70.1	COMPLETO
3585	522130	GO	200	1	1.000000	0.0	BASICO

6 Análise por Região

6.1 Perfil Regional Detalhado

```
# Perfil por região
perfil_regiao = municipios.groupby('regiao').agg({
    'codufmun': 'count',
    'total_leitos': ['sum', 'mean', 'median'],
    'leitos_sus': 'sum',
    'n_estabelecimentos': 'sum',
    'hhi': 'mean',
    'tem_uti': 'sum'
}).round(1)

perfil_regiao.columns = ['N_Municipios', 'Total_Leitos', 'Media_Leitos', 'Mediana_Leitos',
                        'Leitos_SUS', 'N_Estabelecimentos', 'HHI_Medio', 'Municipios_com_UTI']

perfil_regiao['%_SUS'] = (perfil_regiao['Leitos_SUS'] / perfil_regiao['Total_Leitos'] * 100).round(1)
perfil_regiao['%_com_UTI'] = (perfil_regiao['Municipios_com_UTI'] / perfil_regiao['N_Municipios'] * 100).round(1)

print("PERFIL REGIONAL")
print("="*70)
perfil_regiao
```

	N_Municipios	Total_Leitos	Media_Leitos	Mediana_Leitos	Leitos_SUS	N_Estabelecimentos	HHI_Medio	Municipios_com_UTI	%_SUS	%_com_UTI
regiao										
Centro-Oeste	361	47677	132.1	25.0	30102	973	0.8	54	63.1	15.0
Nordeste	1355	142405	105.1	28.0	108138	2955	0.9	154	75.9	11.4
Norte	333	39884	119.8	32.0	30980	795	0.9	48	77.7	14.4
Sudeste	914	221530	242.4	50.0	128125	3132	0.8	312	57.8	34.1
Sul	634	83637	131.9	48.0	57834	1217	0.9	143	69.1	22.6

6.2 Visualização Comparativa Regional

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

# 1. Total de leitos por região
ax1 = axes[0, 0]
leitos_regiao = municipios.groupby('regiao')['total_leitos'].sum().sort_values()
colors_reg = ['#3498db', '#2ecc71', '#e74c3c', '#f39c12', '#9b59b6']
ax1.barh(leitos_regiao.index, leitos_regiao.values, color=colors_reg)
ax1.set_xlabel('Total de Leitos')
ax1.set_title('Total de Leitos por Região', fontweight='bold')
ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
for i, v in enumerate(leitos_regiao.values):
    ax1.text(v + 2000, i, f'{v:,}', va='center', fontsize=9)

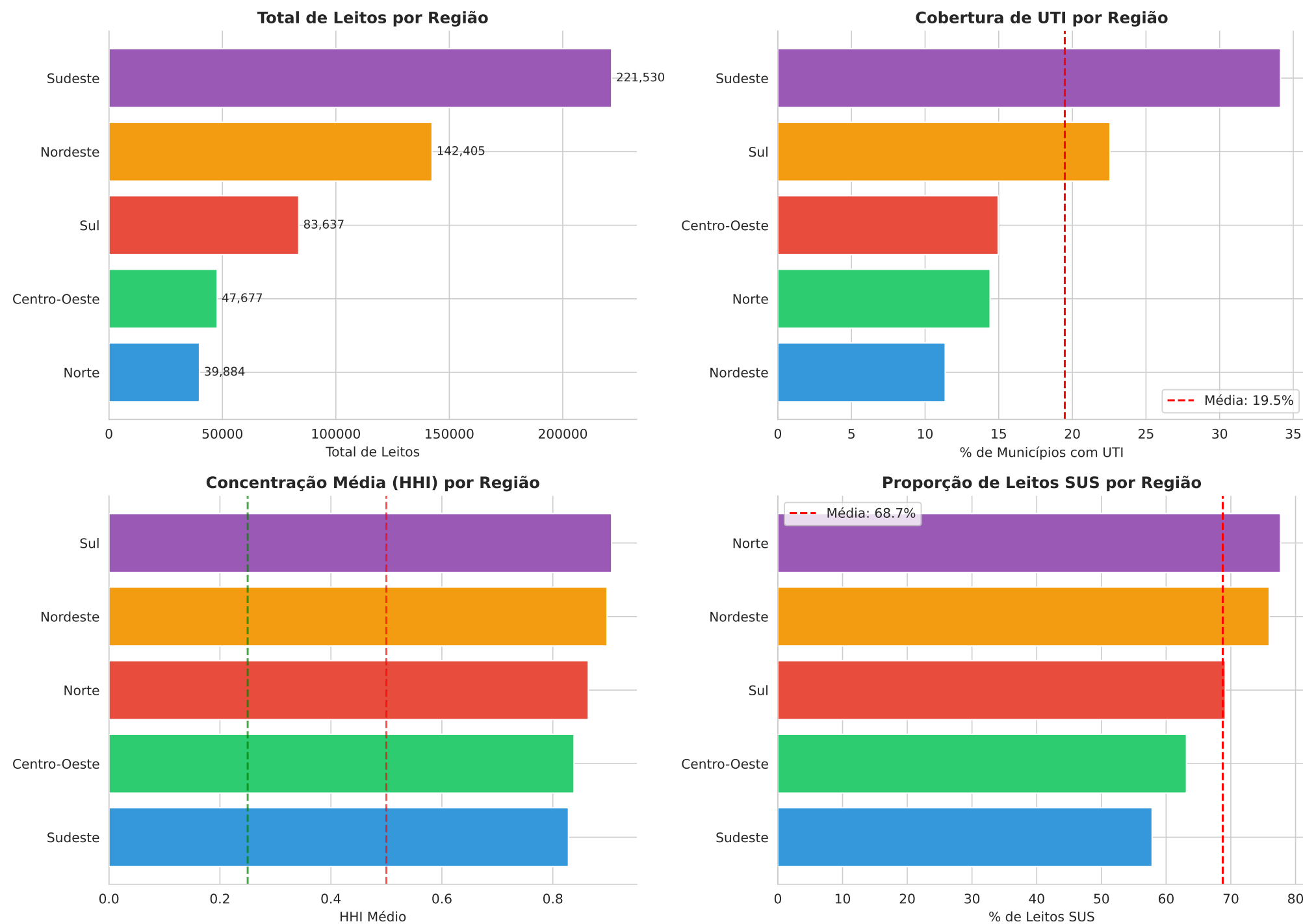
# 2. % de municípios com UTI
ax2 = axes[0, 1]
pct_uti = (municipios.groupby('regiao')['tem_uti'].sum() / municipios.groupby('regiao')['tem_uti'].count() * 100).sort_values()
ax2.barh(pct_uti.index, pct_uti.values, color=colors_reg)
ax2.set_xlabel('% de Municípios com UTI')
ax2.set_title('Cobertura de UTI por Região', fontweight='bold')
ax2.axvline(x=pct_uti.mean(), color='red', linestyle='--', label=f'Média: {pct_uti.mean():.1f}%')
ax2.legend()
ax2.spines['top'].set_visible(False)
ax2.spines['right'].set_visible(False)

# 3. HHI médio por região
ax3 = axes[1, 0]
hhi_regiao = municipios.groupby('regiao')['hhi'].mean().sort_values()
ax3.barh(hhi_regiao.index, hhi_regiao.values, color=colors_reg)
ax3.set_xlabel('HHI Médio')
ax3.set_title('Concentração Média (HHI) por Região', fontweight='bold')
ax3.axvline(x=0.25, color='green', linestyle='--', alpha=0.7)
ax3.axvline(x=0.5, color='red', linestyle='--', alpha=0.7)
ax3.spines['top'].set_visible(False)
ax3.spines['right'].set_visible(False)

# 4. % SUS por região
ax4 = axes[1, 1]
```

```
pct_sus_reg = (municipios.groupby('regiao')['leitos_sus'].sum() / municipios.groupby('regiao')['total_leitos'].sum() * 100).sort_values()
ax4.barh(pct_sus_reg.index, pct_sus_reg.values, color=colors_reg)
ax4.set_xlabel('% de Leitos SUS')
ax4.set_title('Proporção de Leitos SUS por Região', fontweight='bold')
ax4.axvline(x=pct_sus_reg.mean(), color='red', linestyle='--', label=f'Média: {pct_sus_reg.mean():.1f}%')
ax4.legend()
ax4.spines['top'].set_visible(False)
ax4.spines['right'].set_visible(False)

plt.tight_layout()
plt.show()
```



6.2.1 Análise: Perfil Comparativo Regional

i Síntese Regional

Total de Leitos:

- O **Sudeste** concentra mais da metade dos leitos do país, seguido pelo Nordeste e Sul.
- Norte e Centro-Oeste possuem menor oferta absoluta, mas também menor população.

Cobertura de UTI:

- O **Sul** apresenta a maior proporção de municípios com UTI, refletindo rede hospitalar mais capilarizada.
- O **Norte** tem a menor cobertura, com grandes distâncias entre municípios com UTI.
- A média nacional de cobertura UTI é baixa, indicando que a maioria dos municípios depende de referência para outros.

Concentração (HHI):

- Todas as regiões apresentam HHI médio acima de 0.25 (concentração moderada a alta).
- Isso reflete a realidade de municípios pequenos com poucos estabelecimentos.

Natureza (SUS):

- Norte e Nordeste** apresentam maior proporção de leitos SUS, refletindo menor presença do setor privado.
- Sudeste e Sul** têm maior participação privada, especialmente em capitais e regiões metropolitanas.
- Esta diferença tem implicações para políticas de financiamento e regulação.

7 Exportação dos Resultados

```
# Preparar dataset final
municipios_export = municipios[[
    'codufmun', 'uf', 'regiao',
    'n_estabelecimentos', 'total_leitos', 'leitos_sus', 'leitos_nsus', 'pct_sus',
    'leitos_intensivo', 'leitos_semi_intensivo', 'leitos_alta_complexidade', 'leitos_media_baixa',
    'tem_uti', 'tem_uci', 'tem_alta',
    'hhi', 'media_leitos_estab',
    'classificacao_vazio'
]].copy()

municipios_export['codufmun'] = municipios_export['codufmun'].astype(str)
```

```
# Salvar
municipios_export.to_csv('arq7_analise_municipios.csv', sep=';', index=False, encoding='utf-8')

print("EXPORTAÇÃO CONCLUÍDA")
print("="*70)
print(f"Arquivo: arq7_analise_municipios.csv")
print(f"Registros: {len(municipios_export):,}")
print(f"Colunas: {len(municipios_export.columns)}")
```

EXPORTAÇÃO CONCLUÍDA
=====

Arquivo: arq7_analise_municipios.csv
Registros: 3,597
Colunas: 18

8 Resumo Executivo

```
print("="*80)
print("RESUMO EXECUTIVO - ANÁLISE DE DESERTOS DE LEITOS")
print("="*80)

print(f"\n DADOS ANALISADOS")
print(f"  Municípios com leitos: {len(municipios):,}")
print(f"  Total de leitos: {municipios['total_leitos'].sum():,}")
print(f"  Estabelecimentos: {municipios['n_estabelecimentos'].sum():,}")

print(f"\n COBERTURA DE ALTA COMPLEXIDADE")
print(f"  Municípios com UTI: {municipios['tem_uti'].sum():,} ({municipios['tem_uti'].mean()*100:.1f}%)")
print(f"  Municípios com UCI: {municipios['tem_uci'].sum():,} ({municipios['tem_uci'].mean()*100:.1f}%)")
print(f"  Municípios sem alta complexidade: {(~municipios['tem_uti'] & ~municipios['tem_uci']).sum():,}")

print(f"\n CLASSIFICAÇÃO DE VAZIOS")
for classif in ['COMPLETO', 'PARCIAL', 'BASICO']:
    n = (municipios['classificacao_vazio'] == classif).sum()
    pct = n / len(municipios) * 100
    print(f"  {classif}: {n:,} municípios ({pct:.1f}%)")

print(f"\n CONCENTRAÇÃO (HHI)")
print(f"  HHI médio nacional: {municipios['hhi'].mean():.3f}")
print(f"  Municípios com monopólio (HHI>0.8): {(municipios['hhi'] > 0.8).sum():,}")
print(f"  Municípios com baixa concentração (HHI<0.25): {(municipios['hhi'] < 0.25).sum():,}")

print(f"\n NATUREZA (SUS vs PRIVADO)")
print(f"  % médio de leitos SUS: {municipios['pct_sus'].mean():.1f}%)")
print(f"  Municípios >80% SUS: {(municipios['pct_sus'] > 80).sum():,}")
print(f"  Municípios <50% SUS: {(municipios['pct_sus'] < 50).sum():,}")

print(f"\n ALERTAS")
print(f"  • {(~municipios['tem_uti']).sum():,} municípios sem leitos de UTI")
print(f"  • {(municipios['hhi'] > 0.8).sum():,} municípios com alta concentração (risco de desassistência)")
print(f"  • Região com menor cobertura UTI: {pct_uti.idxmin()} ({pct_uti.min():.1f}%)")
```

=====

RESUMO EXECUTIVO - ANÁLISE DE DESERTOS DE LEITOS

=====

DADOS ANALISADOS

Municípios com leitos: 3,597

Total de leitos: 535,133

Estabelecimentos: 9,072

COBERTURA DE ALTA COMPLEXIDADE

Municípios com UTI: 711 (19.8%)

Municípios com UCI: 601 (16.7%)

Municípios sem alta complexidade: 2,685

CLASSIFICAÇÃO DE VAZIOS

COMPLETO: 912 municípios (25.4%)

PARCIAL: 7 municípios (0.2%)

BASICO: 2,678 municípios (74.5%)

CONCENTRAÇÃO (HHI)

HHI médio nacional: 0.872

Municípios com monopólio (HHI>0.8): 2,801

Municípios com baixa concentração (HHI<0.25): 147

NATUREZA (SUS vs PRIVADO)

% médio de leitos SUS: 87.6%

Municípios >80% SUS: 2,661

Municípios <50% SUS: 180

ALERTAS

- 2,886 municípios sem leitos de UTI
- 2,801 municípios com alta concentração (risco de desassistência)
- Região com menor cobertura UTI: Nordeste (11.4%)