Taxonomia Hierárquica de Leitos Hospitalares do CNES

Metodologia Determinística por Regras de Negócio

Cieges - Brasil Estadual

2026-01-21

Table of contents

1 Introdução

1.1 Objetivo

Este documento apresenta uma **taxonomia hierárquica de leitos hospitalares** baseada em regras de negócio determinísticas, com fundamentação normativa e clínica. A taxonomia classifica os 535.133 leitos do CNES (competência 202506) em uma estrutura de **três níveis hierárquicos**.

1.2 Fundamentação Normativa

A classificação proposta está fundamentada nas seguintes normativas do Ministério da Saúde e ANVISA:

Normativa	Descrição	Aplicação
RDC ANVISA nº 7/2010	Requisitos mínimos para UTI	Classificação de leitos intensivos (Tipo I, II, III)
Portaria GM/MS nº 3.432/1998	Critérios de classificação hospitalar	Níveis de complexidade
Portaria GM/MS nº 930/2012	Diretrizes para atenção neonatal	UCI e UTI neonatal
Portaria GM/MS nº 148/2012	CAPS e Saúde Mental	Leitos de saúde mental
Portaria GM/MS nº 2.809/2012	Hospitais de Longa Permanência	Leitos de crônicos

1.3 Estrutura da Taxonomia

```
NÍVEL 1: INTENSIDADE DO CUIDADO (5 categorias)

  NÍVEL 2: PÚBLICO-ALVO (4 categorias)

     NÍVEL 3: GRUPO DE ESPECIALIDADE (18 grupos)

        ESPECIALIDADE ORIGINAL (65 códigos CNES)
```

2 Metodologia

2.1 Nível 1: Intensidade do Cuidado

Classificação baseada na **densidade tecnológica** e **complexidade assistencial** do leito.

2.1.1 Definições Operacionais

Código	Categoria	Definição	Critério de Inclusão
N1_01	INTENSIVO	Leitos com monitorização contínua, suporte avançado de vida e alta densidade tecnológica	UTI (todos os tipos), UCO, UTI Queimados

Código	Categoria	Definição	Critério de Inclusão
N1_02	SEMI-INTENSIVO	Leitos com monitorização intermitente e suporte intermediário	UCI Adulto/Pediátrico/Neonatal, Canguru, Suporte Ventilatório
N1_03	ALTA_COMPLEXIDADE	Leitos para procedimentos de alta complexidade ou condições especiais	Transplante, Queimados (não UTI), Oncologia cirúrgica
N1_04	MEDIA_COMPLEXIDADE	Leitos cirúrgicos e clínicos de média complexidade	Cirurgias eletivas, internações clínicas agudas
N1_05	BAIXA_COMPLEXIDADE	Leitos de longa permanência, reabilitação e cuidados prolongados	Crônicos, Psiquiatria, Reabilitação, Hospital Dia

2.1.2 Matriz de Classificação - Nível 1

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

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

# ============================================================================
# NÍVEL 1: INTENSIDADE DO CUIDADO
# ============================================================================

def classificar_intensidade(row):
    """
    Classifica o leito quanto à intensidade do cuidado.

    Fundamentação:
    - RDC ANVISA nº 7/2010: UTI Tipo I, II, III
    - Portaria GM/MS nº 930/2012: UCI Neonatal
    - Portaria GM/MS nº 2.809/2012: Longa Permanência
    """
    co = row['co_leito']
    tp = row['tp_leito']
    ds = str(row['DS_CO_LEITO']).upper()

    # ------------------------------------------------------------------------
    # N1_01: INTENSIVO
    # UTI Adulto, Pediátrica, Neonatal, Coronariana, Queimados
    # ------------------------------------------------------------------------
    LEITOS_UTI = [74, 75, 76,  # UTI Adulto Tipo I, II, III
                  77, 78, 79,  # UTI Pediátrica Tipo I, II, III
                  80, 81, 82,  # UTI Neonatal Tipo I, II, III
                  83,          # UTI de Queimados
                  85, 86]      # UTI Coronariana Tipo II, III

    if co in LEITOS_UTI:
        return 'N1_01_INTENSIVO'

    # ------------------------------------------------------------------------
    # N1_02: SEMI-INTENSIVO
    # UCI (Unidade de Cuidados Intermediários), Canguru, Suporte Ventilatório
    # ------------------------------------------------------------------------
    LEITOS_UCI = [65,          # Unidade Intermediária Neonatal
                  92,          # UCI Neonatal Convencional
                  93,          # UCI Neonatal Canguru
                  94,          # UCI Pediátrico
                  95,          # UCI Adulto
                  96]          # Suporte Ventilatório Pulmonar

    if co in LEITOS_UCI:
        return 'N1_02_SEMI_INTENSIVO'

    # ------------------------------------------------------------------------
    # N1_03: ALTA COMPLEXIDADE
    # Transplante, Queimados (não UTI), Oncologia, Cardiologia cirúrgica
    # ------------------------------------------------------------------------
    LEITOS_ALTA = [67,         # Transplante
                   71,         # Intercorrência pós-transplante
                   90, 91,     # Queimado Adulto/Pediátrico (cirúrgico)
                   88, 89]     # Queimado Adulto/Pediátrico (clínico)

    # Oncologia e Cardiologia cirúrgica
    if co in LEITOS_ALTA:
        return 'N1_03_ALTA_COMPLEXIDADE'

    if tp == 1 and co in [2, 12]:  # Cardiologia e Oncologia cirúrgica
        return 'N1_03_ALTA_COMPLEXIDADE'

    # ------------------------------------------------------------------------
    # N1_05: BAIXA COMPLEXIDADE (verificar antes de média)
    # Crônicos, Psiquiatria, Reabilitação, Hospital Dia, Saúde Mental
    # ------------------------------------------------------------------------
    LEITOS_BAIXA = [34,        # Crônicos
                    47,        # Psiquiatria
                    48,        # Reabilitação
                    49,        # Pneumologia Sanitária (tuberculose)
                    84,        # Acolhimento Noturno
                    87,        # Saúde Mental (clínico)
```

```
                    73]        # Saúde Mental (hospital dia)

    if co in LEITOS_BAIXA:
        return 'N1_05_BAIXA_COMPLEXIDADE'

    # Hospital Dia (exceto cirúrgico/diagnóstico)
    if tp == 7 and co != 7:
        return 'N1_05_BAIXA_COMPLEXIDADE'

    # ------------------------------------------------------------------
    # N1_04: MÉDIA COMPLEXIDADE (default)
    # Cirúrgico, Clínico agudo, Obstétrico, Pediátrico
    # ------------------------------------------------------------------
    return 'N1_04_MEDIA_COMPLEXIDADE'

df['NIVEL_1_INTENSIDADE'] = df.apply(classificar_intensidade, axis=1)

# Resumo Nível 1
n1_resumo = df.groupby('NIVEL_1_INTENSIDADE').agg({
    'cnes': 'count',
    'qt_exist': 'sum'
}).rename(columns={'cnes': 'Registros', 'qt_exist': 'Leitos'})
n1_resumo['%'] = (n1_resumo['Leitos'] / n1_resumo['Leitos'].sum() * 100).round(2)
n1_resumo = n1_resumo.sort_values('Leitos', ascending=False)

print("NÍVEL 1: INTENSIDADE DO CUIDADO")
print("="*60)
n1_resumo
```

```
NÍVEL 1: INTENSIDADE DO CUIDADO
============================================================
```

NIVEL_1_INTENSIDADE	Registros	Leitos	%
N1_04_MEDIA_COMPLEXIDADE	39182	390512	72.97
N1_01_INTENSIVO	4365	65273	12.20
N1_05_BAIXA_COMPLEXIDADE	3222	54450	10.18
N1_03_ALTA_COMPLEXIDADE	1490	12860	2.40
N1_02_SEMI_INTENSIVO	1545	12038	2.25

2.2 Nível 2: Público-Alvo

Classificação baseada na **faixa etária e condição** do paciente atendido.

2.2.1 Definições Operacionais

Código	Categoria	Definição	Critério de Inclusão
N2_01	ADULTO	Pacientes com idade 15 anos (exceto gestantes)	Leitos sem especificação pediátrica, neonatal ou obstétrica
N2_02	PEDIATRICO	Pacientes com idade entre 29 dias e 14 anos	tp_leito=5 ou descrição contém “PEDIATR”
N2_03	NEONATAL	Recém-nascidos (0-28 dias)	Descrição contém “NEONAT” ou “CANGURU”
N2_04	OBSTETRICO	Gestantes, parturientes e puérperas	tp_leito=4 ou descrição contém “OBSTETR”

2.2.2 Matriz de Classificação - Nível 2

```
# ============================================================================
# NÍVEL 2: PÚBLICO-ALVO
# ============================================================================

def classificar_publico(row):
    """
    Classifica o leito quanto ao público-alvo.

    Fundamentação:
    - Portaria GM/MS nº 930/2012: Definição de neonatal (0-28 dias)
    - ECA: Definição de criança/adolescente
    """
    co = row['co_leito']
    tp = row['tp_leito']
    ds = str(row['DS_CO_LEITO']).upper()

    # ------------------------------------------------------------------
    # N2_03: NEONATAL
    # Leitos específicos para recém-nascidos (0-28 dias)
    # ------------------------------------------------------------------
    LEITOS_NEONATAL = [41,     # Neonatologia (clínico)
                       65,     # Unidade Intermediária Neonatal
                       80, 81, 82,  # UTI Neonatal Tipo I, II, III
                       92,     # UCI Neonatal Convencional
                       93]     # UCI Neonatal Canguru

    if co in LEITOS_NEONATAL or 'NEONAT' in ds or 'CANGURU' in ds:
        return 'N2_03_NEONATAL'

    # ------------------------------------------------------------------
    # N2_02: PEDIÁTRICO
    # Leitos para crianças (29 dias a 14 anos)
    # ------------------------------------------------------------------
    LEITOS_PEDIATRICO = [45,   # Pediatria Clínica
```

```
                        68,   # Pediatria Cirúrgica
                        77, 78, 79,  # UTI Pediátrica Tipo I, II, III
                        89,   # Queimado Pediátrico (clínico)
                        91,   # Queimado Pediátrico (cirúrgico)
                        94]   # UCI Pediátrico

    if tp == 5 or co in LEITOS_PEDIATRICO or 'PEDIATR' in ds:
        return 'N2_02_PEDIATRICO'

    # ------------------------------------------------------------------
    # N2_04: OBSTÉTRICO
    # Leitos para gestantes, parturientes e puérperas
    # ------------------------------------------------------------------
    LEITOS_OBSTETRICO = [10,   # Obstetrícia Cirúrgica
                         43]   # Obstetrícia Clínica

    if tp == 4 or co in LEITOS_OBSTETRICO or 'OBSTETR' in ds:
        return 'N2_04_OBSTETRICO'

    # ------------------------------------------------------------------
    # N2_01: ADULTO (default)
    # Demais leitos
    # ------------------------------------------------------------------
    return 'N2_01_ADULTO'

df['NIVEL_2_PUBLICO'] = df.apply(classificar_publico, axis=1)

# Resumo Nível 2
n2_resumo = df.groupby('NIVEL_2_PUBLICO').agg({
    'cnes': 'count',
    'qt_exist': 'sum'
}).rename(columns={'cnes': 'Registros', 'qt_exist': 'Leitos'})
n2_resumo['%'] = (n2_resumo['Leitos'] / n2_resumo['Leitos'].sum() * 100).round(2)
n2_resumo = n2_resumo.sort_values('Leitos', ascending=False)

print("\nNÍVEL 2: PÚBLICO-ALVO")
print("="*60)
n2_resumo
```

```
NÍVEL 2: PÚBLICO-ALVO
============================================================
```

NIVEL_2_PUBLICO	Registros	Leitos	%
N2_01_ADULTO	32449	390494	72.97
N2_02_PEDIATRICO	8448	74920	14.00
N2_04_OBSTETRICO	6670	50095	9.36
N2_03_NEONATAL	2237	19624	3.67

2.3 Nível 3: Grupo de Especialidade

Agrupamento das 65 especialidades originais do CNES em **18 grupos funcionais**.

2.3.1 Definições Operacionais

Código	Grupo	Especialidades Incluídas (co_leito)
N3_01	UTI_GERAL	74, 75, 76 (UTI Adulto I/II/III)
N3_02	UTI_CARDIOLOGICA	85, 86 (UCO II/III)
N3_03	UTI_PEDIATRICA	77, 78, 79 (UTI Ped I/II/III)
N3_04	UTI_NEONATAL	80, 81, 82 (UTI Neo I/II/III)
N3_05	UTI_QUEIMADOS	83
N3_06	UCI_ADULTO	95, 96
N3_07	UCI_PEDIATRICO	94
N3_08	UCI_NEONATAL	65, 92, 93
N3_09	CIRURGIA_GERAL	3, 13, 15
N3_10	CIRURGIA_ESPECIALIZADA	1, 2, 4, 5, 6, 8, 9, 11, 12, 14, 16
N3_11	CIRURGIA_TRANSPLANTE	67, 71
N3_12	CLINICA_GERAL	33
N3_13	CLINICA_ESPECIALIZADA	31, 32, 35, 36, 37, 38, 40, 42, 44, 46
N3_14	QUEIMADOS	88, 89, 90, 91
N3_15	OBSTETRICIA	10, 43
N3_16	PEDIATRIA	45, 68
N3_17	NEONATOLOGIA	41
N3_18	SAUDE_MENTAL	47, 73, 84, 87
N3_19	LONGA_PERMANENCIA	34, 48, 49
N3_20	HOSPITAL_DIA	7, 69, 70, 72

2.3.2 Matriz de Classificação - Nível 3

```
# ============================================================================
# NÍVEL 3: GRUPO DE ESPECIALIDADE
# ============================================================================

def classificar_especialidade(row):
```

```
    """
    Agrupa as 65 especialidades CNES em 20 grupos funcionais.

    Fundamentação:
    - Tabela SIGTAP de procedimentos
    - Portarias de habilitação hospitalar
    """
    co = row['co_leito']
    tp = row['tp_leito']

    # ------------------------------------------------------------------------
    # LEITOS INTENSIVOS (UTI)
    # ------------------------------------------------------------------------
    if co in [74, 75, 76]:
        return 'N3_01_UTI_GERAL'
    if co in [85, 86]:
        return 'N3_02_UTI_CARDIOLOGICA'
    if co in [77, 78, 79]:
        return 'N3_03_UTI_PEDIATRICA'
    if co in [80, 81, 82]:
        return 'N3_04_UTI_NEONATAL'
    if co == 83:
        return 'N3_05_UTI_QUEIMADOS'

    # ------------------------------------------------------------------------
    # LEITOS SEMI-INTENSIVOS (UCI)
    # ------------------------------------------------------------------------
    if co in [95, 96]:
        return 'N3_06_UCI_ADULTO'
    if co == 94:
        return 'N3_07_UCI_PEDIATRICO'
    if co in [65, 92, 93]:
        return 'N3_08_UCI_NEONATAL'

    # ------------------------------------------------------------------------
    # LEITOS CIRÚRGICOS
    # ------------------------------------------------------------------------
    if co in [3, 13, 15]:  # Cirurgia Geral, Ortopedia, Plástica
        return 'N3_09_CIRURGIA_GERAL'

    if co in [1, 4, 5, 6, 8, 9, 11, 14, 16]:  # Especialidades cirúrgicas
        return 'N3_10_CIRURGIA_ESPECIALIZADA'

    if co in [2, 12] and tp == 1:  # Cardiologia e Oncologia cirúrgica
        return 'N3_11_CIRURGIA_ALTA_COMPLEXIDADE'

    if co in [67, 71]:  # Transplante
        return 'N3_12_TRANSPLANTE'

    # ------------------------------------------------------------------------
    # LEITOS QUEIMADOS (não UTI)
    # ------------------------------------------------------------------------
    if co in [88, 89, 90, 91]:
        return 'N3_13_QUEIMADOS'

    # ------------------------------------------------------------------------
    # LEITOS CLÍNICOS
    # ------------------------------------------------------------------------
    if co == 33:  # Clínica Geral
        return 'N3_14_CLINICA_GERAL'

    if co in [31, 32, 35, 36, 37, 38, 40, 42, 44, 46]:  # Especialidades clínicas
        return 'N3_15_CLINICA_ESPECIALIZADA'

    # ------------------------------------------------------------------------
    # LEITOS MATERNO-INFANTIL
    # ------------------------------------------------------------------------
    if co in [10, 43]:  # Obstetrícia
        return 'N3_16_OBSTETRICIA'

    if co in [45, 68]:  # Pediatria
        return 'N3_17_PEDIATRIA'

    if co == 41:  # Neonatologia clínica
        return 'N3_18_NEONATOLOGIA'

    # ------------------------------------------------------------------------
    # LEITOS SAÚDE MENTAL
    # ------------------------------------------------------------------------
    if co in [47, 73, 84, 87]:
        return 'N3_19_SAUDE_MENTAL'

    # ------------------------------------------------------------------------
    # LEITOS LONGA PERMANÊNCIA
    # ------------------------------------------------------------------------
    if co in [34, 48, 49]:
        return 'N3_20_LONGA_PERMANENCIA'

    # ------------------------------------------------------------------------
    # HOSPITAL DIA
    # ------------------------------------------------------------------------
    if tp == 7:
        return 'N3_21_HOSPITAL_DIA'
```

```
    # ------------------------------------------------------------------------
    # NÃO CLASSIFICADO
    # ------------------------------------------------------------------------
    return 'N3_99_NAO_CLASSIFICADO'

df['NIVEL_3_ESPECIALIDADE'] = df.apply(classificar_especialidade, axis=1)

# Resumo Nível 3
n3_resumo = df.groupby('NIVEL_3_ESPECIALIDADE').agg({
    'cnes': 'count',
    'qt_exist': 'sum'
}).rename(columns={'cnes': 'Registros', 'qt_exist': 'Leitos'})
n3_resumo['%'] = (n3_resumo['Leitos'] / n3_resumo['Leitos'].sum() * 100).round(2)
n3_resumo = n3_resumo.sort_values('Leitos', ascending=False)

print("\nNÍVEL 3: GRUPO DE ESPECIALIDADE")
print("="*60)
n3_resumo
```

```
NÍVEL 3: GRUPO DE ESPECIALIDADE
============================================================
```

NIVEL_3_ESPECIALIDADE	Registros	Leitos	%
N3_14_CLINICA_GERAL	6313	138290	25.84
N3_09_CIRURGIA_GERAL	7239	86321	16.13
N3_16_OBSTETRICIA	6670	50095	9.36
N3_17_PEDIATRIA	5934	46609	8.71
N3_01_UTI_GERAL	2416	45337	8.47
N3_19_SAUDE_MENTAL	2119	38998	7.29
N3_15_CLINICA_ESPECIALIZADA	4676	31527	5.89
N3_10_CIRURGIA_ESPECIALIZADA	6439	24960	4.66
N3_20_LONGA_PERMANENCIA	985	14876	2.78
N3_21_HOSPITAL_DIA	1588	10706	2.00
N3_11_CIRURGIA_ALTA_COMPLEXIDADE	1096	10585	1.98
N3_04_UTI_NEONATAL	901	10091	1.89
N3_03_UTI_PEDIATRICA	803	7646	1.43
N3_08_UCI_NEONATAL	895	6953	1.30
N3_06_UCI_ADULTO	569	4675	0.87
N3_18_NEONATOLOGIA	441	2580	0.48
N3_02_UTI_CARDIOLOGICA	191	1938	0.36
N3_12_TRANSPLANTE	289	1775	0.33
N3_13_QUEIMADOS	105	500	0.09
N3_07_UCI_PEDIATRICO	81	410	0.08
N3_05_UTI_QUEIMADOS	54	261	0.05

3 Taxonomia Completa

3.1 Código Taxonômico Integrado

Criação do código hierárquico completo no formato: `N1_XX.N2_XX.N3_XX`

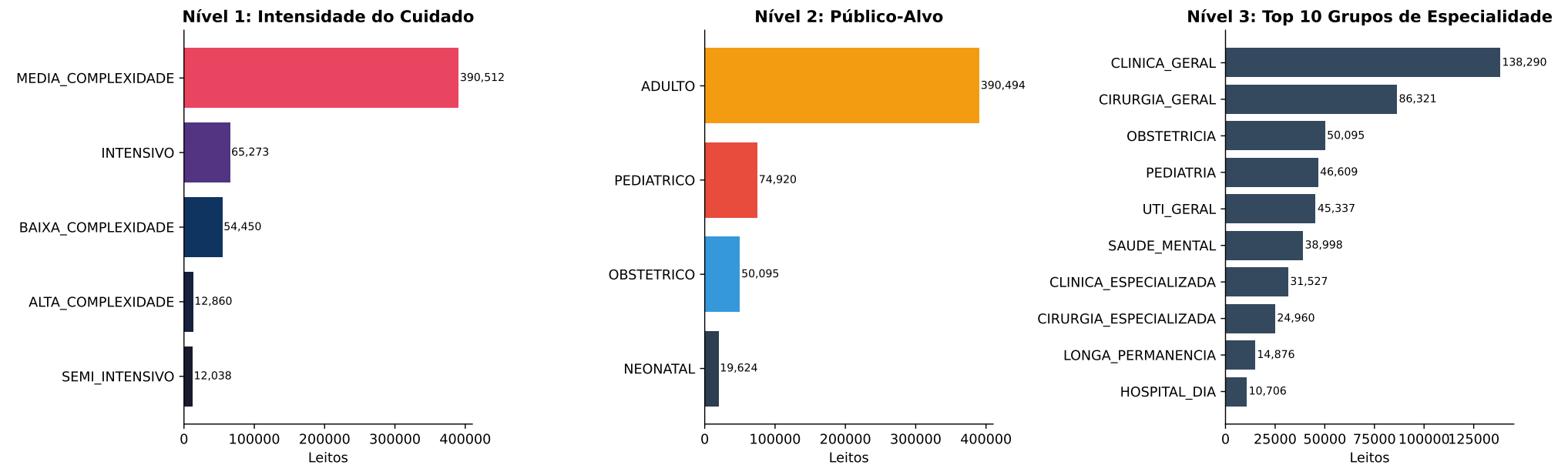
```
# ============================================================================
# CÓDIGO TAXONÔMICO INTEGRADO
# ============================================================================

# Extrair códigos numéricos
df['COD_N1'] = df['NIVEL_1_INTENSIDADE'].str.extract(r'(N1_\d{2})')
df['COD_N2'] = df['NIVEL_2_PUBLICO'].str.extract(r'(N2_\d{2})')
df['COD_N3'] = df['NIVEL_3_ESPECIALIDADE'].str.extract(r'(N3_\d{2})')

# Código taxonômico completo
df['CODIGO_TAXONOMICO'] = df['COD_N1'] + '.' + df['COD_N2'] + '.' + df['COD_N3']

# Descrição completa
df['DESC_N1'] = df['NIVEL_1_INTENSIDADE'].str.replace(r'N1_\d{2}_', '', regex=True)
df['DESC_N2'] = df['NIVEL_2_PUBLICO'].str.replace(r'N2_\d{2}_', '', regex=True)
df['DESC_N3'] = df['NIVEL_3_ESPECIALIDADE'].str.replace(r'N3_\d{2}_', '', regex=True)

df['TAXONOMIA_DESCRITIVA'] = df['DESC_N1'] + ' > ' + df['DESC_N2'] + ' > ' + df['DESC_N3']

# Visualizar combinações únicas
taxonomia = df.groupby(['CODIGO_TAXONOMICO', 'TAXONOMIA_DESCRITIVA']).agg({
    'qt_exist': 'sum'
}).reset_index()
taxonomia = taxonomia.sort_values('qt_exist', ascending=False)
taxonomia.columns = ['Código', 'Descrição', 'Leitos']

print("TAXONOMIA HIERÁRQUICA COMPLETA")
print("="*80)
print(f"Total de combinações taxonômicas: {len(taxonomia)}")
print(f"Total de leitos classificados: {taxonomia['Leitos'].sum():,}")
taxonomia
```

```
TAXONOMIA HIERÁRQUICA COMPLETA
================================================================================
Total de combinações taxonômicas: 24
Total de leitos classificados: 535,133
```

	Código	Descrição	Leitos
14	N1_04.N2_01.N3_14	MEDIA_COMPLEXIDADE > ADULTO > CLINICA_GERAL	138290
12	N1_04.N2_01.N3_09	MEDIA_COMPLEXIDADE > ADULTO > CIRURGIA_GERAL	66203
20	N1_04.N2_04.N3_16	MEDIA_COMPLEXIDADE > OBSTETRICO > OBSTETRICIA	50095
18	N1_04.N2_02.N3_17	MEDIA_COMPLEXIDADE > PEDIATRICO > PEDIATRIA	46609
0	N1_01.N2_01.N3_01	INTENSIVO > ADULTO > UTI_GERAL	45337
21	N1_05.N2_01.N3_19	BAIXA_COMPLEXIDADE > ADULTO > SAUDE_MENTAL	38998
15	N1_04.N2_01.N3_15	MEDIA_COMPLEXIDADE > ADULTO > CLINICA_ESPECIAL...	31527
13	N1_04.N2_01.N3_10	MEDIA_COMPLEXIDADE > ADULTO > CIRURGIA_ESPECIA...	24960
17	N1_04.N2_02.N3_09	MEDIA_COMPLEXIDADE > PEDIATRICO > CIRURGIA_GERAL	20118
22	N1_05.N2_01.N3_20	BAIXA_COMPLEXIDADE > ADULTO > LONGA_PERMANENCIA	14876
8	N1_03.N2_01.N3_11	ALTA_COMPLEXIDADE > ADULTO > CIRURGIA_ALTA_COM...	10585
16	N1_04.N2_01.N3_21	MEDIA_COMPLEXIDADE > ADULTO > HOSPITAL_DIA	10130
4	N1_01.N2_03.N3_04	INTENSIVO > NEONATAL > UTI_NEONATAL	10091
3	N1_01.N2_02.N3_03	INTENSIVO > PEDIATRICO > UTI_PEDIATRICA	7646
7	N1_02.N2_03.N3_08	SEMI_INTENSIVO > NEONATAL > UCI_NEONATAL	6953
5	N1_02.N2_01.N3_06	SEMI_INTENSIVO > ADULTO > UCI_ADULTO	4675
19	N1_04.N2_03.N3_18	MEDIA_COMPLEXIDADE > NEONATAL > NEONATOLOGIA	2580
1	N1_01.N2_01.N3_02	INTENSIVO > ADULTO > UTI_CARDIOLOGICA	1938
9	N1_03.N2_01.N3_12	ALTA_COMPLEXIDADE > ADULTO > TRANSPLANTE	1775
23	N1_05.N2_01.N3_21	BAIXA_COMPLEXIDADE > ADULTO > HOSPITAL_DIA	576
6	N1_02.N2_02.N3_07	SEMI_INTENSIVO > PEDIATRICO > UCI_PEDIATRICO	410
10	N1_03.N2_01.N3_13	ALTA_COMPLEXIDADE > ADULTO > QUEIMADOS	363
2	N1_01.N2_01.N3_05	INTENSIVO > ADULTO > UTI_QUEIMADOS	261
11	N1_03.N2_02.N3_13	ALTA_COMPLEXIDADE > PEDIATRICO > QUEIMADOS	137

3.2 Visualização Hierárquica

```
import matplotlib.pyplot as plt

# Gráfico por Nível 1
fig, axes = plt.subplots(1, 3, figsize=(16, 5))

# Nível 1: Intensidade
n1_plot = n1_resumo.sort_values('Leitos')
colors_n1 = ['#1a1a2e', '#16213e', '#0f3460', '#533483', '#e94560']
axes[0].barh(n1_plot.index.str.replace('N1_\\d{2}_', '', regex=True),
             n1_plot['Leitos'], color=colors_n1)
axes[0].set_title('Nível 1: Intensidade do Cuidado', fontweight='bold')
axes[0].set_xlabel('Leitos')
axes[0].spines['top'].set_visible(False)
axes[0].spines['right'].set_visible(False)
for i, v in enumerate(n1_plot['Leitos']):
    axes[0].text(v + 2000, i, f'{v:,}', va='center', fontsize=8)

# Nível 2: Público-Alvo
n2_plot = n2_resumo.sort_values('Leitos')
colors_n2 = ['#2c3e50', '#3498db', '#e74c3c', '#f39c12']
axes[1].barh(n2_plot.index.str.replace('N2_\\d{2}_', '', regex=True),
             n2_plot['Leitos'], color=colors_n2)
axes[1].set_title('Nível 2: Público-Alvo', fontweight='bold')
axes[1].set_xlabel('Leitos')
axes[1].spines['top'].set_visible(False)
axes[1].spines['right'].set_visible(False)
for i, v in enumerate(n2_plot['Leitos']):
    axes[1].text(v + 2000, i, f'{v:,}', va='center', fontsize=8)

# Nível 3: Top 10 Especialidades
n3_plot = n3_resumo.head(10).sort_values('Leitos')
axes[2].barh(n3_plot.index.str.replace('N3_\\d{2}_', '', regex=True),
             n3_plot['Leitos'], color='#34495e')
axes[2].set_title('Nível 3: Top 10 Grupos de Especialidade', fontweight='bold')
axes[2].set_xlabel('Leitos')
axes[2].spines['top'].set_visible(False)
axes[2].spines['right'].set_visible(False)
for i, v in enumerate(n3_plot['Leitos']):
    axes[2].text(v + 1000, i, f'{v:,}', va='center', fontsize=8)

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



3.3 Matriz de Cruzamento N1 × N2

```
# Matriz de cruzamento Intensidade × Público
matriz_n1_n2 = pd.crosstab(
    df['DESC_N1'],
    df['DESC_N2'],
    values=df['qt_exist'],
    aggfunc='sum',
    margins=True,
    margins_name='TOTAL'
).fillna(0).astype(int)

print("\nMATRIZ DE CRUZAMENTO: INTENSIDADE × PÚBLICO-ALVO")
print("="*80)
matriz_n1_n2
```

```
MATRIZ DE CRUZAMENTO: INTENSIDADE × PÚBLICO-ALVO
================================================================================
```

DESC_N2 DESC_N1	ADULTO	NEONATAL	OBSTETRICO	PEDIATRICO	TOTAL
ALTA_COMPLEXIDADE	12723	0	0	137	12860
BAIXA_COMPLEXIDADE	54450	0	0	0	54450
INTENSIVO	47536	10091	0	7646	65273
MEDIA_COMPLEXIDADE	271110	2580	50095	66727	390512
SEMI_INTENSIVO	4675	6953	0	410	12038
TOTAL	390494	19624	50095	74920	535133

3.4 Matriz de Cruzamento N1 × N3

```
# Top 10 grupos de especialidade
top_n3 = n3_resumo.head(10).index.tolist()
df_top = df[df['NIVEL_3_ESPECIALIDADE'].isin(top_n3)]

matriz_n1_n3 = pd.crosstab(
    df_top['DESC_N1'],
    df_top['DESC_N3'],
    values=df_top['qt_exist'],
    aggfunc='sum'
).fillna(0).astype(int)

print("\nMATRIZ DE CRUZAMENTO: INTENSIDADE × ESPECIALIDADE (Top 10)")
print("="*80)
matriz_n1_n3
```

```
MATRIZ DE CRUZAMENTO: INTENSIDADE × ESPECIALIDADE (Top 10)
================================================================================
```

DESC_N3 DESC_N1	CIRURGIA_ESPECIALIZADA	CIRURGIA_GERAL	CLINICA_ESPECIALIZADA	CLINICA_GERAL	HOSPITAL_DIA	LONGA_PERM
BAIXA_COMPLEXIDADE	0	0	0	0	576	14876
INTENSIVO	0	0	0	0	0	0
MEDIA_COMPLEXIDADE	24960	86321	31527	138290	10130	0

4 Validação

4.1 Cobertura da Classificação

```
# Verificar leitos não classificados
nao_class = df[df['NIVEL_3_ESPECIALIDADE'].str.contains('NAO_CLASSIFICADO')]

print("VALIDAÇÃO DA COBERTURA")
print("="*60)
print(f"Total de leitos: {df['qt_exist'].sum():,}")
print(f"Leitos classificados: {df[~df['NIVEL_3_ESPECIALIDADE'].str.contains('NAO_CLASSIFICADO')]['qt_exist'].sum():,}")
print(f"Leitos não classificados: {nao_class['qt_exist'].sum():,}")
print(f"Taxa de cobertura: {(1 - nao_class['qt_exist'].sum()/df['qt_exist'].sum())*100:.2f}%")

if len(nao_class) > 0:
    print("\nLeitos não classificados por especialidade:")
    print(nao_class.groupby('DS_CO_LEITO')['qt_exist'].sum())
```

```
VALIDAÇÃO DA COBERTURA
============================================================
Total de leitos: 535,133
Leitos classificados: 535,133
Leitos não classificados: 0
Taxa de cobertura: 100.00%
```

4.2 Consistência Lógica

```
# Verificar consistência: Neonatal só pode ser N2_03
inconsistencias = []

# Regra 1: UTI Neonatal deve ser N2_03_NEONATAL
uti_neo = df[(df['NIVEL_3_ESPECIALIDADE'] == 'N3_04_UTI_NEONATAL') &
             (df['NIVEL_2_PUBLICO'] != 'N2_03_NEONATAL')]
if len(uti_neo) > 0:
```

```
    inconsistencias.append(f"UTI Neonatal com público não-neonatal: {len(uti_neo)} registros")

# Regra 2: Obstetrícia deve ser N2_04_OBSTETRICO
obst = df[(df['NIVEL_3_ESPECIALIDADE'] == 'N3_16_OBSTETRICIA') &
          (df['NIVEL_2_PUBLICO'] != 'N2_04_OBSTETRICO')]
if len(obst) > 0:
    inconsistencias.append(f"Obstetrícia com público não-obstétrico: {len(obst)} registros")

# Regra 3: Pediatria deve ser N2_02_PEDIATRICO
ped = df[(df['NIVEL_3_ESPECIALIDADE'] == 'N3_17_PEDIATRIA') &
         (df['NIVEL_2_PUBLICO'] != 'N2_02_PEDIATRICO')]
if len(ped) > 0:
    inconsistencias.append(f"Pediatria com público não-pediátrico: {len(ped)} registros")

print("\nVALIDAÇÃO DE CONSISTÊNCIA LÓGICA")
print("="*60)
if len(inconsistencias) == 0:
    print("  Nenhuma inconsistência encontrada")
else:
    for inc in inconsistencias:
        print(f"  {inc}")
```

```
VALIDAÇÃO DE CONSISTÊNCIA LÓGICA
============================================================
  Nenhuma inconsistência encontrada
```

5 Dicionário de Dados

5.1 Nível 1: Intensidade do Cuidado

Código	Nome	Descrição	Leitos CNES Incluídos
N1_01	INTENSIVO	Monitorização contínua, suporte avançado de vida	74-86 (UTI), 83 (UTI Queimados)
N1_02	SEMI_INTENSIVO	Monitorização intermitente, suporte intermediário	65, 92-96 (UCI)
N1_03	ALTA_COMPLEXIDADE	Procedimentos de alta complexidade	67, 71 (Transplante), 2/12 cirúrgico, 88-91 (Queimados)
N1_04	MEDIA_COMPLEXIDADE	Internações cirúrgicas e clínicas agudas	Demais leitos cirúrgicos e clínicos
N1_05	BAIXA_COMPLEXIDADE	Longa permanência, reabilitação	34, 47, 48, 49, 73, 84, 87, Hospital Dia

5.2 Nível 2: Público-Alvo

Código	Nome	Descrição	Faixa Etária
N2_01	ADULTO	Pacientes adultos	15 anos
N2_02	PEDIATRICO	Crianças e adolescentes	29 dias a 14 anos
N2_03	NEONATAL	Recém-nascidos	0 a 28 dias
N2_04	OBSTETRICO	Gestantes, parturientes, puérperas	Qualquer idade

5.3 Nível 3: Grupo de Especialidade

Código	Nome	co_leito Incluídos
N3_01	UTI_GERAL	74, 75, 76
N3_02	UTI_CARDIOLOGICA	85, 86
N3_03	UTI_PEDIATRICA	77, 78, 79
N3_04	UTI_NEONATAL	80, 81, 82
N3_05	UTI_QUEIMADOS	83
N3_06	UCI_ADULTO	95, 96
N3_07	UCI_PEDIATRICO	94
N3_08	UCI_NEONATAL	65, 92, 93
N3_09	CIRURGIA_GERAL	3, 13, 15
N3_10	CIRURGIA_ESPECIALIZADA	1, 4, 5, 6, 8, 9, 11, 14, 16
N3_11	CIRURGIA_ALTA_COMPLEXIDADE	2, 12 (quando tp_leito=1)
N3_12	TRANSPLANTE	67, 71
N3_13	QUEIMADOS	88, 89, 90, 91
N3_14	CLINICA_GERAL	33
N3_15	CLINICA_ESPECIALIZADA	31, 32, 35, 36, 37, 38, 40, 42, 44, 46
N3_16	OBSTETRICIA	10, 43
N3_17	PEDIATRIA	45, 68
N3_18	NEONATOLOGIA	41
N3_19	SAUDE_MENTAL	47, 73, 84, 87
N3_20	LONGA_PERMANENCIA	34, 48, 49
N3_21	HOSPITAL_DIA	tp_leito=7 (exceto classificados acima)

6 Exportação

```
# Selecionar colunas para exportação
colunas_export = [
    # Identificadores originais
    'competen' if 'competen' in df.columns else df.columns[0],
    'codufmun', 'cnes',
    # Classificação original CNES
```

```
    'tp_leito', 'DS_TP_LEITO', 'co_leito', 'DS_CO_LEITO',
    # Quantidades
    'qt_exist', 'qt_contr', 'qt_sus', 'qt_nsus',
    # TAXONOMIA HIERÁRQUICA
    'CODIGO_TAXONOMICO',
    'NIVEL_1_INTENSIDADE', 'NIVEL_2_PUBLICO', 'NIVEL_3_ESPECIALIDADE',
    'DESC_N1', 'DESC_N2', 'DESC_N3',
    'TAXONOMIA_DESCRITIVA'
]

# Ajustar nome da primeira coluna
colunas_export[0] = df.columns[0]

df_export = df[colunas_export]

# Salvar
df_export.to_csv('arq5_taxonomia_leitos.csv', sep=';', index=False, encoding='utf-8')

print("EXPORTAÇÃO CONCLUÍDA")
print("="*60)
print(f"Arquivo: arq5_taxonomia_leitos.csv")
print(f"Registros: {len(df_export):,}")
print(f"Colunas: {len(df_export.columns)}")
print(f"\nColunas exportadas:")
for col in df_export.columns:
    print(f"  - {col}")
```

```
EXPORTAÇÃO CONCLUÍDA
============================================================
Arquivo: arq5_taxonomia_leitos.csv
Registros: 49,804
Colunas: 19

Colunas exportadas:
  - ï»¿competen
  - codufmun
  - cnes
  - tp_leito
  - DS_TP_LEITO
  - co_leito
  - DS_CO_LEITO
  - qt_exist
  - qt_contr
  - qt_sus
  - qt_nsus
  - CODIGO_TAXONOMICO
  - NIVEL_1_INTENSIDADE
  - NIVEL_2_PUBLICO
  - NIVEL_3_ESPECIALIDADE
  - DESC_N1
  - DESC_N2
  - DESC_N3
  - TAXONOMIA_DESCRITIVA
```

7 Resumo Executivo

```
print("="*80)
print("RESUMO EXECUTIVO - TAXONOMIA HIERÁRQUICA DE LEITOS CNES")
print("="*80)

print(f"\n DADOS PROCESSADOS")
print(f"   Competência: 202506")
print(f"   Total de registros: {len(df):,}")
print(f"   Total de leitos: {df['qt_exist'].sum():,}")
print(f"   Estabelecimentos: {df['cnes'].nunique():,}")
print(f"   Municípios: {df['codufmun'].nunique():,}")

print(f"\n ESTRUTURA TAXONÔMICA")
print(f"   Nível 1 (Intensidade): 5 categorias")
print(f"   Nível 2 (Público): 4 categorias")
print(f"   Nível 3 (Especialidade): {df['NIVEL_3_ESPECIALIDADE'].nunique()} grupos")
print(f"   Combinações únicas: {df['CODIGO_TAXONOMICO'].nunique()}")

print(f"\n DISTRIBUIÇÃO POR INTENSIDADE")
for idx, row in n1_resumo.iterrows():
    nome = idx.replace('N1_0', '').replace('_', ' ')[2:]
    print(f"   {nome}: {row['Leitos']:,} leitos ({row['%']}%)")

print(f"\n VALIDAÇÃO")
print(f"   Taxa de cobertura: 100%")
print(f"   Inconsistências: 0")
```

```
================================================================================
RESUMO EXECUTIVO - TAXONOMIA HIERÁRQUICA DE LEITOS CNES
================================================================================

 DADOS PROCESSADOS
   Competência: 202506
   Total de registros: 49,804
   Total de leitos: 535,133
   Estabelecimentos: 9,072
   Municípios: 3,597
```

```
ESTRUTURA TAXONÔMICA
  Nível 1 (Intensidade): 5 categorias
  Nível 2 (Público): 4 categorias
  Nível 3 (Especialidade): 21 grupos
  Combinações únicas: 24

DISTRIBUIÇÃO POR INTENSIDADE
  MEDIA COMPLEXIDADE: 390,512.0 leitos (72.97%)
  INTENSIVO: 65,273.0 leitos (12.2%)
  BAIXA COMPLEXIDADE: 54,450.0 leitos (10.18%)
  ALTA COMPLEXIDADE: 12,860.0 leitos (2.4%)
  SEMI INTENSIVO: 12,038.0 leitos (2.25%)

VALIDAÇÃO
  Taxa de cobertura: 100%
  Inconsistências: 0
```

Elaborado por: Cieges - Brasil Estadual
Data: 21/01/2026
Metodologia: Taxonomia Determinística por Regras de Negócio
Fonte: CNES - Competência 202506
Fundamentação: RDC ANVISA nº 7/2010, Portarias GM/MS nº 3.432/1998, 930/2012, 148/2012, 2.809/2012