

Busca_em_Largura_2

March 30, 2022

1 Estudando BDGD

Notebook referente a primeira versão das análises exploratórias da BDGD.

Vamos ler as tabelas referentes a rede de Média Tensão (MT) da Energisa-Paraíba (EPB) a partir da base de dados de 31/12/2020.

Atenção:

Estamos lendo os dados com um primeiro filtro:

Tabelas que possuem o campo **SIT_ATIV: Tipo de Situação de Ativação (TSITATI)** já foram filtradas para ler apenas situação **AT Ativada**.

```
[1]: import pandas as pd
import numpy as np
```

1.1 Leitura dos DataFrames de Interesse - MT

Todos os dataframes lidos serão armazenados no dicionário **base**, para facilitar o acesso.

```
[2]: base = {}
```

```
[3]: def ler_dataframe(filename, index_col, filter = ''):
    '''
    Lê o dataframe a partir do arquivo e imprime as 3 primeiras linhas e o
    ↪shape.
    '''

    df = pd.read_csv(filename, index_col=index_col)
    return df.query(filter) if len(filter)>0 else df
```

1.1.1 Leitura: SUB

Subestação

- Designação da Entidade: Subestação
- Designação da Modelagem: SUB
- Tipo: Polígono

```
[4]: base['SUB'] = ler_dataframe(filename = '../20201231_Exportação/20201231_sub.
→csv', index_col = 1)
display(base['SUB'].head(3))
```

		WKT	COD_ID	DIST	\
OBJECTID					
1	MULTIPOLYGON (((-35.036465324615 -6.9844584661...	125389094	6600		
2	MULTIPOLYGON (((-35.0381456700866 -7.493050442...	125389096	6600		
3	MULTIPOLYGON (((-38.169453141749 -6.8835603674...	125389097	6600		

	POS	NOM	DESCR	Shape_Length	Shape_Area
OBJECTID					
1	PD	DERIVAÃ	JAPUNGU	0.002362	3.481311e-07
2	PD	DERIVAÃ	POTY	0.001089	7.373553e-08
3	PD	DERIVAÃ	VÃ RZEA DE SOUZA	0.001076	7.226439e-08

1.1.2 Leitura: CTMT

Circuito de Média Tensão

- Designação da Entidade: Circuito de Média Tensão
- Designação da Modelagem: CTMT
- Tipo: Entidades Não Geográficas

```
[5]: base['CTMT'] = ler_dataframe(filename = '../20201231_Exportação/20201231_ctmt.
→csv', index_col = 0)
display(base['CTMT'].head(3))
```

	COD_ID	NOM	BARR	SUB	PAC	TEN_NOM	\
OBJECTID							
1	17067303	IBR-L1	404181344	12873436	56560487917146437MT	49	
2	17067304	IBR-L2	404181344	12873436	56560312917147011MT	49	
3	17067308	IBR-L4	273725354	12873436	56560193917147954MT	49	

	TEN_OPE	ATIP	RECONFIG	DIST	...	PNTBT_06	PNTBT_07	\
OBJECTID					...			
1	1.029	0	0	6600	...	25489.904739	28300.107793	
2	1.029	0	0	6600	...	29168.047490	34798.767123	
3	1.029	0	0	6600	...	47019.126530	54452.304372	

	PNTBT_08	PNTBT_09	PNTBT_10	PNTBT_11	\
OBJECTID					
1	36189.774149	34932.604990	53273.719264	28495.406676	
2	45886.252346	43400.600957	60956.705062	33167.035131	
3	71262.602440	68976.054974	98080.409456	53571.931293	

PNTBT_12	DESCR	PERD_A3aA4	PERD_A4A3a
----------	-------	------------	------------

OBJECTID					
1	40634.251546	SE	IBIARA	0	0
2	47549.473862	SE	IBIARA	0	0
3	78323.682392	SE	IBIARA	0	0

[3 rows x 58 columns]

1.1.3 Leitura: UCMT

Unidade Consumidora de Média Tensão

- Designação da Entidade: Unidade Consumidora de Média Tensão
- Designação da Modelagem: UCMT
- Tipo: Ponto

```
[6]: base['UCMT'] = ler_dataframe(
    filename = '../20201231_Exportação/20201231_ucmt.csv',
    index_col = 0,
    filter = 'SIT_ATIV == "AT"',
)
display(base['UCMT'].head(3))
```

	PN_CON	DIST		PAC		CEG	CTMT	\
OBJECTID								
1	215318339	6600	95071600919893254MT				51696835	
2	212200903	6600	68443548922522502MT	GD.PB.000.288.174			17067396	
3	208830151	6600	94574235920713836MT				51696938	

	UNI_TR_S	SUB	CONJ	MUN	BRR	...	ENE_08	\
OBJECTID						...		
1	211896113	12873482	13694	2504603	GRAMAME	...	48944.0	
2	12910486	12873439	13728	2510808	AREA RURAL	...	-1428.0	
3	12911647	12873417	13738	2513703	STA RITA	...	1154.0	

	ENE_09	ENE_10	ENE_11	ENE_12	DIC	FIC	SEMRED	DESCR	ARE_LOC
OBJECTID									
1	53146.0	56375.0	59707.0	58374	4.38	4	0		UB
2	-2236.0	-2395.0	-1845.0	-836	23.93	6	0		NU
3	1178.0	1053.0	1100.0	1079	0.53	1	0		UB

[3 rows x 51 columns]

1.1.4 Leitura: UGMT

Unidade Geradora de Média Tensão

- Designação da Entidade: Unidade Geradora de Média Tensão
- Designação da Modelagem: UGMT

- Tipo: Ponto

```
[7]: base['UGMT'] = ler_dataframe(
      filename = '../20201231_Exportação/20201231_ugmt.csv',
      index_col = 0,
      filter = 'SIT_ATIV == "AT"',
    )
display(base['UGMT'].head(3))
```

	PN_CON	DIST	PAC	CEG	\
OBJECTID					
1	48609303	6600	94080102927187766MT	EOL.CV.PB.029075-0.01	
2	307167171	6600	92786260924008931MT	UTE.AI.PB.034891-0.01	
3	225272837	6600	94369950921027026BT	UTE.RU.PB.043199-0.02	

	CTMT	UNI_TR_S	SUB	CONJ	MUN	BRR	...	\
OBJECTID							...	
1	51696884	12875414	12912337	15806	2509305	AREA RURAL	...	
2	210952743	12910813	12873487	13730	2508901	MAMANGUAPE	...	
3	51696942	34820744	12873417	13738	2513703	CENTRO	...	

	ENE_06	ENE_07	ENE_08	ENE_09	ENE_10	\
OBJECTID						
1	690638.994	884255.136	1282595.512	1216664.058	870275.106	
2	0.000	0.000	694666.803	1382124.756	2080004.825	
3	751298.974	1639810.853	1958572.721	2367510.458	2326935.806	

	ENE_11	ENE_12	DIC	FIC	DESCR
OBJECTID					
1	1057531.366	1034874.153	2.01	1	Vitoria Eolica
2	1937204.994	2026783.968	9.75	6	UTE Monte Alegre Biomassa
3	2451976.676	2478586.494	0.00	0	ASJA Bioenergia

[3 rows x 46 columns]

1.1.5 Leitura: SSDMT

Segmento do Sistema de Distribuição de Média Tensão

- Designação da Entidade: Segmento de Rede Média Tensão
- Designação da Modelagem: SSDMT
- Tipo: Linha

```
[8]: base['SSDMT'] = ler_dataframe(filename = '../20201231_Exportação/20201231_ssdtmt.
      ↪csv', index_col = 1)
display(base['SSDMT'].head(3))
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:

DtypeWarning: Columns (4) have mixed types.Specify dtype option on import or set low_memory=False.

"""Entry point for launching an IPython kernel.

	WKT	COD_ID	\
OBJECTID			
1	MULTILINESTRING ((-38.2559744372704 -6.5957049...	15223672	
2	MULTILINESTRING ((-38.2590492067554 -6.5834943...	15223673	
3	MULTILINESTRING ((-38.2571946131408 -6.5848633...	15223674	

	PN_CON_1	PN_CON_2	CTMT	UNI_TR_S	SUB	CONJ	DIST	\
OBJECTID								
1	20084633	20084577	17067438	12911922	12873440	13739	6600	
2	20083770	20084206	17067438	12911922	12873440	13739	6600	
3	20326698	20084492	17067438	12911922	12873440	13739	6600	

	PAC_1	...	ODI_FAS	TI_FAS	ODI_NEU	TI_NEU	COMP	\
OBJECTID		...						
1	58224935927088283MT	...	1	41		0	180.579	
2	58191143927223327MT	...	1	41		0	178.969	
3	58211624927208161MT	...	1	41		0	60.545	

	DESCR	SITCONFAS	SITCONTNEU	ARE_LOC	Shape_Length
OBJECTID					
1		AT1	0	NU	0.001633
2		AT1	0	NU	0.001619
3		AT1	0	NU	0.000548

[3 rows x 24 columns]

1.1.6 Leitura: UNCRMT

Unidade Compensadora de Reativo de Média Tensão

- Designação da Entidade: Unidade Compensadora de Reativo Média Tensão
- Designação da Modelagem: UNCRMT
- Tipo: Ponto

```
[9]: base['UNCRMT'] = ler_dataframe(
    filename = '../20201231_Exportação/20201231_uncrmt.csv',
    index_col = 1,
    filter = 'SIT_ATIV == "AT"',
)
display(base['UNCRMT'].head(3))
```

	WKT	COD_ID	DIST	FAS_CON	\
OBJECTID					
1	POINT (-35.6626347765736 -7.24688882935595)	12910620	6600	ABC	

2	POINT (-35.1010751879695 -6.81570055113946)	12910817	6600	ABC
3	POINT (-35.1010751879695 -6.81570055113946)	12910818	6600	ABC

	SIT_ATIV	TIP_UNID	POT_NOM	PAC_1	PAC_2	CTMT	UNI_TR_S	\
OBJECTID								
1	AT	10	14	RIC12910630		26260976	281690694	
2	AT	10	16	RTT12910871		51696925	12910813	
3	AT	10	16	RTT12910875		51696927	12910814	

	SUB	CONJ	MUN	DAT_CON	BANC	POS	DESCR	ARE_LOC
OBJECTID								
1	12873447	13729	2512754	01/01/1920	0	PD		UB
2	12873487	13730	2512903	01/01/1920	0	PD		UB
3	12873487	13730	2512903	01/01/1920	0	PD		UB

1.1.7 Leitura: UNREMT

Unidade Reguladora de Média Tensão

- Designação da Entidade: Unidade Reguladora de Média Tensão
- Designação da Modelagem: UNREMT
- Tipo: Ponto

```
[10]: base['UNREMT'] = ler_dataframe(
    filename = '../20201231_Exportação/20201231_unremt.csv',
    index_col = 1,
    filter = 'SIT_ATIV == "AT"',
)
display(base['UNREMT'].head(3))
```

		WKT	COD_ID	DIST	\
OBJECTID					
1	POINT (-38.0577686211249 -7.4431955120632)	320380142	6600		
2	POINT (-38.4886518070352 -7.55012825511483)	335443000	6600		
3	POINT (-38.1822249806983 -6.81963307330761)	339110082	6600		

	FAS_CON	SIT_ATIV	TIP_UNID	PAC_1	PAC_2	\
OBJECTID						
1	ABC	AT	13	60397344917714660MT	60397394917714574MT	
2	ABC	AT	13	55641141916540268MT	55641044916540243MT	
3	ABC	AT	13	59036095924611374MT	59036170924611309MT	

	CTMT	UNI_TR_S	SUB	CONJ	MUN	DAT_CON	BANC	POS	\
OBJECTID									
1	17067317	12907810	12873433	13707	2511004	01/01/1920	1	PD	
2	387937142	12877171	12873436	13704	2504405	01/01/1920	1	PD	
3	17067450	12911923	12873440	13739	2516201	01/01/1920	1	PD	

	DESCR
OBJECTID	
1	
2	
3	

1.1.8 Leitura: UNSEMT

Unidade Seccionadora de Média Tensão

- Designação da Entidade: Unidade Seccionadora de Média Tensão
- Designação da Modelagem: UNSEMT
- Tipo: Ponto

```
[11]: base['UNSEMT'] = ler_dataframe(
    filename = '../20201231_Exportação/20201231_unsemt.csv',
    index_col = 1,
    filter = 'SIT_ATIV == "AT"',
)
display(base['UNSEMT'].head(3))
```

		WKT	COD_ID	DIST	\
OBJECTID					
1	POINT (-35.7050706318042 -7.61082473027705)	24312960	6600		
2	POINT (-35.5900458995985 -7.18070067739433)	24312961	6600		
3	POINT (-36.8918768843107 -7.34120679735577)	24312963	6600		

	PAC_1	PAC_2	FAS_CON	SIT_ATIV	TIP_UNID	\
OBJECTID						
1	86362930915733974MT	86363082915733979MT	B	AT	22	
2	87670313920487277MT	87670162920487260MT	ABC	AT	22	
3	73272227918798438MT	73272101918798353MT	A	AT	22	

	P_N_OPE	CAP_ELO	...	TLCD	DAT_CON	POS	CTMT	UNI_TR_S	\
OBJECTID			...						
1	F	2H	...	0	01/01/1920	PD	26260949	397864865	
2	F	2H	...	0	01/01/1920	PD	26260976	281690694	
3	F	1H	...	0	01/01/1920	PD	139962991	12908668	

	SUB	CONJ	MUN	DESCR	ARE_LOC
OBJECTID					
1	12873450	13686	2501302		NU
2	12873447	13729	2507606		UB
3	12873453	16188	2508505		NU

[3 rows x 21 columns]

1.1.9 Resultado

```
[12]: for key in base.keys():  
       print(f'[{key}]: {base[key].shape}')
```

```
[SUB]: (91, 8)  
[CTMT]: (360, 58)  
[UCMT]: (1891, 51)  
[UGMT]: (5, 46)  
[SSDMT]: (434463, 24)  
[UNCRMT]: (145, 19)  
[UNREMT]: (265, 17)  
[UNSEMT]: (102984, 21)
```

1.2 Consistência da base em relação às SUBs

Vou guardar em `lista_SUBs` os códigos das subestações declaradas em **SUB**.

Todos os dataframes filtrados serão armazenados nos dicionários `base_ok` e `base_not_ok`, para facilitar o acesso.

```
[13]: base_ok = {}  
       base_not_ok = {}  
  
       lista_SUBs = list(base['SUB']['COD_ID'].unique())  
       lista_SUBs[:3]
```

```
[13]: [125389094, 125389096, 125389097]
```

```
[14]: def filtrar_por_SUB(key, base, lista_SUBs):  
       '''  
       Separa o dataframe base[key] olhando para a coluna 'SUB'.  
       Registros com 'SUB' pertencentes a 'lista_SUBs' vão para 'df'.  
       Os demais vão para 'df_not'.  
       '''  
  
       def converte_para_int(valor):  
           try:  
               return int(valor)  
           except:  
               return -1  
  
       flag = [(converte_para_int(item) in lista_SUBs) for item in_  
→base[key]['SUB']]  
       flag_not = [not item for item in flag]  
  
       df = base[key][flag].copy()  
       display(df.head(3))  
       print(f'df.shape = {df.shape}')
```



```

df_not = base[key][flag_not].copy()
display(df_not.head(3))
print(f'df_not.shape = {df_not.shape}')

return df, df_not

```

1.2.1 Filtro CTMT

```
[15]: base_ok['CTMT'], base_not_ok['CTMT'] = filtrar_por_SUB('CTMT', base, lista_SUBs)
```

	COD_ID	NOM	BARR	SUB	PAC	TEN_NOM	\
OBJECTID							
1	17067303	IBR-L1	404181344	12873436	56560487917146437MT	49	
2	17067304	IBR-L2	404181344	12873436	56560312917147011MT	49	
3	17067308	IBR-L4	273725354	12873436	56560193917147954MT	49	

	TEN_OPE	ATIP	RECONFIG	DIST	...	PNTBT_06	PNTBT_07	\
OBJECTID					...			
1	1.029	0	0	6600	...	25489.904739	28300.107793	
2	1.029	0	0	6600	...	29168.047490	34798.767123	
3	1.029	0	0	6600	...	47019.126530	54452.304372	

	PNTBT_08	PNTBT_09	PNTBT_10	PNTBT_11	\
OBJECTID					
1	36189.774149	34932.604990	53273.719264	28495.406676	
2	45886.252346	43400.600957	60956.705062	33167.035131	
3	71262.602440	68976.054974	98080.409456	53571.931293	

	PNTBT_12	DESCR	PERD_A3aA4	PERD_A4A3a
OBJECTID				
1	40634.251546	SE IBIARA	0	0
2	47549.473862	SE IBIARA	0	0
3	78323.682392	SE IBIARA	0	0

[3 rows x 58 columns]

```
df.shape = (346, 58)
```

	COD_ID	NOM	BARR	SUB	PAC	\
OBJECTID						
65	137884650	CGD-01Y4	259791375	12873455	84147243919589116MT	
69	17067266	CJZ-L2	12875486	164370026	17067266	
110	53149444	CGD-01Y1	259791375	12873455	84147527919589942MT	

	TEN_NOM	TEN_OPE	ATIP	RECONFIG	DIST	...	PNTBT_06	\
OBJECTID						...		

65	49	1.029	0	0	6600	...	0.000000
69	49	1.029	0	0	6600	...	206896.072758
110	49	1.029	0	0	6600	...	0.000000

	PNTBT_07	PNTBT_08	PNTBT_09	PNTBT_10	\
OBJECTID					
65	0.000000	0.000000	0.000000	0.000000	
69	243548.239114	312777.615982	300575.281758	5803.728748	
110	0.000000	0.000000	0.000000	0.000000	

	PNTBT_11	PNTBT_12	\
OBJECTID			
65	0.000000	0.000000	
69	1818.800136	1474.978926	
110	0.000000	0.000000	

	DESCR	PERD_A3aA4	PERD_A4A3a
OBJECTID			
65	A linha deriva de SE de atendimento precário	0	0
69	Atendimento Precário	0	0
110	A linha deriva de SE de atendimento precário	0	0

[3 rows x 58 columns]

df_not.shape = (14, 58)

1.2.2 Filtro UCMT

```
[16]: base_ok['UCMT'], base_not_ok['UCMT'] = filtrar_por_SUB('UCMT', base, lista_SUBs)
```

	PN_CON	DIST	PAC	CEG	CTMT	\
OBJECTID						
1	215318339	6600	95071600919893254MT		51696835	
2	212200903	6600	68443548922522502MT	GD.PB.000.288.174	17067396	
3	208830151	6600	94574235920713836MT		51696938	

	UNI_TR_S	SUB	CONJ	MUN	BRR	...	ENE_08	\
OBJECTID						...		
1	211896113	12873482	13694	2504603	GRAMAME	...	48944.0	
2	12910486	12873439	13728	2510808	AREA RURAL	...	-1428.0	
3	12911647	12873417	13738	2513703	STA RITA	...	1154.0	

	ENE_09	ENE_10	ENE_11	ENE_12	DIC	FIC	SEMRED	DESCR	ARE_LOC
OBJECTID									
1	53146.0	56375.0	59707.0	58374	4.38	4	0		UB
2	-2236.0	-2395.0	-1845.0	-836	23.93	6	0		NU
3	1178.0	1053.0	1100.0	1079	0.53	1	0		UB

```
[3 rows x 51 columns]
```

```
df.shape = (1890, 51)
```

	PN_CON	DIST		PAC	CEG		CTMT	UNI_TR_S	SUB	\
OBJECTID										
717		6600	80280396920211948MT				245932938	BVA41T1	BVA	

	CONJ	MUN	BRR	...	ENE_08	ENE_09	\
OBJECTID				...			
717	13722	2516102	AREA RURAL	...	1472569.85	1431428.256	

	ENE_10	ENE_11	ENE_12	DIC	FIC	SEMRED	DESCR	\
OBJECTID								
717	1501620.696	1403008.808	1453582	0.0	0	0	PBBVA-01P1-01	

	ARE_LOC
OBJECTID	
717	NU

```
[1 rows x 51 columns]
```

```
df_not.shape = (1, 51)
```

No campo SUB aparece 'BVA'.

1.2.3 Filtro UGMT

```
[17]: base_ok['UGMT'], base_not_ok['UGMT'] = filtrar_por_SUB('UGMT', base, lista_SUBs)
```

	PN_CON	DIST		PAC		CEG	\
OBJECTID							
1	48609303	6600	94080102927187766MT	EOL.CV.PB.029075-0.01			
2	307167171	6600	92786260924008931MT	UTE.AI.PB.034891-0.01			
3	225272837	6600	94369950921027026BT	UTE.RU.PB.043199-0.02			

	CTMT	UNI_TR_S	SUB	CONJ	MUN	BRR	...	\
OBJECTID							...	
1	51696884	12875414	12912337	15806	2509305	AREA RURAL	...	
2	210952743	12910813	12873487	13730	2508901	MAMANGUAPE	...	
3	51696942	34820744	12873417	13738	2513703	CENTRO	...	

	ENE_06	ENE_07	ENE_08	ENE_09	ENE_10	\
OBJECTID						
1	690638.994	884255.136	1282595.512	1216664.058	870275.106	
2	0.000	0.000	694666.803	1382124.756	2080004.825	
3	751298.974	1639810.853	1958572.721	2367510.458	2326935.806	

	ENE_11	ENE_12	DIC	FIC		DESCR
OBJECTID						
1	1057531.366	1034874.153	2.01	1		Vitoria Eolica
2	1937204.994	2026783.968	9.75	6	UTE Monte Alegre Biomassa	
3	2451976.676	2478586.494	0.00	0	ASJA Bioenergia	

[3 rows x 46 columns]

df.shape = (5, 46)

Empty DataFrame

Columns: [PN_CON, DIST, PAC, CEG, CTMT, UNI_TR_S, SUB, CONJ, MUN, BRR, CEP, CNAE, FAS_CON, GRU]

Index: []

[0 rows x 46 columns]

df_not.shape = (0, 46)

1.2.4 Filtro SSDMT

```
[18]: base_ok['SSDMT'], base_not_ok['SSDMT'] = filtrar_por_SUB('SSDMT', base,
↳ lista_SUBs)
```

		WKT	COD_ID	\
OBJECTID				
1	MULTILINESTRING ((-38.2559744372704 -6.5957049...		15223672	
2	MULTILINESTRING ((-38.2590492067554 -6.5834943...		15223673	
3	MULTILINESTRING ((-38.2571946131408 -6.5848633...		15223674	

	PN_CON_1	PN_CON_2	CTMT	UNI_TR_S	SUB	CONJ	DIST	\
OBJECTID								
1	20084633	20084577	17067438	12911922	12873440	13739	6600	
2	20083770	20084206	17067438	12911922	12873440	13739	6600	
3	20326698	20084492	17067438	12911922	12873440	13739	6600	

	PAC_1	...	ODI_FAS	TI_FAS	ODI_NEU	TI_NEU	COMP	\
OBJECTID		...						
1	58224935927088283MT	...	1	41		0	180.579	
2	58191143927223327MT	...	1	41		0	178.969	
3	58211624927208161MT	...	1	41		0	60.545	

	DESCR	SITCONFAS	SITCONTNEU	ARE_LOC	Shape_Length
OBJECTID					
1		AT1	0	NU	0.001633
2		AT1	0	NU	0.001619
3		AT1	0	NU	0.000548

```
[3 rows x 24 columns]
```

```
df.shape = (433966, 24)
```

		WKT	COD_ID	\
OBJECTID				
5897	MULTILINESTRING	((-35.9136537529183 -7.2581441...	157508226	
5952	MULTILINESTRING	((-35.9144243720871 -7.2580269...	157508253	
5958	MULTILINESTRING	((-35.9145847400936 -7.2576449...	157508262	

	PN_CON_1	PN_CON_2	CTMT	UNI_TR_S	SUB	CONJ	DIST	\
OBJECTID								
5897	157508214	157508227	53149444		12873455	13722	6600	
5952	157508227	203707362	53149444		12873455	13722	6600	
5958	203707362	53162729	53149444		12873455	13722	6600	

	PAC_1	...	ODI_FAS	TI_FAS	ODI_NEU	TI_NEU	COMP	\
OBJECTID		...						
5897	84086003919655332MT	...	1	40		0	86.174	
5952	84077493919656688MT	...	1	40		0	45.856	
5958	84075749919660929MT	...	1	40		0	43.064	

	DESCR	SITCONFAS	SITCONTNEU	ARE_LOC	Shape_Length
OBJECTID					
5897		AT1	0	UB	0.000779
5952		AT1	0	UB	0.000414
5958		AT1	0	UB	0.000389

```
[3 rows x 24 columns]
```

```
df_not.shape = (497, 24)
```

```
[19]: 12873455 in lista_SUBs
```

```
[19]: False
```

1.2.5 Filtro UNCRMT

```
[20]: base_ok['UNCRMT'], base_not_ok['UNCRMT'] = filtrar_por_SUB('UNCRMT', base, lista_SUBs)
```

		WKT	COD_ID	DIST	FAS_CON	\
OBJECTID						
1	POINT	(-35.6626347765736 -7.24688882935595)	12910620	6600	ABC	
2	POINT	(-35.1010751879695 -6.81570055113946)	12910817	6600	ABC	
3	POINT	(-35.1010751879695 -6.81570055113946)	12910818	6600	ABC	

	SIT_ATIV	TIP_UNID	POT_NOM	PAC_1	PAC_2	CTMT	UNI_TR_S	\
OBJECTID								
1	AT	10	14	RIC12910630		26260976	281690694	
2	AT	10	16	RTT12910871		51696925	12910813	
3	AT	10	16	RTT12910875		51696927	12910814	

	SUB	CONJ	MUN	DAT_CON	BANC	POS	DESCR	ARE_LOC
OBJECTID								
1	12873447	13729	2512754	01/01/1920	0	PD		UB
2	12873487	13730	2512903	01/01/1920	0	PD		UB
3	12873487	13730	2512903	01/01/1920	0	PD		UB

```
df.shape = (145, 19)
```

```
Empty DataFrame
```

```
Columns: [WKT, COD_ID, DIST, FAS_CON, SIT_ATIV, TIP_UNID, POT_NOM, PAC_1, PAC_2, CTMT, UNI_TR_S]
```

```
Index: []
```

```
df_not.shape = (0, 19)
```

1.2.6 Filtro UNREMT

```
[21]: base_ok['UNREMT'], base_not_ok['UNREMT'] = filtrar_por_SUB('UNREMT', base,
↳ lista_SUBs)
```

	WKT	COD_ID	DIST	\
OBJECTID				
1	POINT (-38.0577686211249 -7.4431955120632)	320380142	6600	
2	POINT (-38.4886518070352 -7.55012825511483)	335443000	6600	
3	POINT (-38.1822249806983 -6.81963307330761)	339110082	6600	

	FAS_CON	SIT_ATIV	TIP_UNID	PAC_1	PAC_2	\
OBJECTID						
1	ABC	AT	13	60397344917714660MT	60397394917714574MT	
2	ABC	AT	13	55641141916540268MT	55641044916540243MT	
3	ABC	AT	13	59036095924611374MT	59036170924611309MT	

	CTMT	UNI_TR_S	SUB	CONJ	MUN	DAT_CON	BANC	POS	\
OBJECTID									
1	17067317	12907810	12873433	13707	2511004	01/01/1920	1	PD	
2	387937142	12877171	12873436	13704	2504405	01/01/1920	1	PD	
3	17067450	12911923	12873440	13739	2516201	01/01/1920	1	PD	

```
DESCR
```

```
OBJECTID
```

```
1
```

2
3

```
df.shape = (263, 17)
```

	WKT	COD_ID	DIST	FAS_CON	\
OBJECTID					
77	POINT (-35.9343140041879 -7.22037880293936)	53090327	6600	ABC	
253	POINT (-38.5222371552205 -7.31690242390766)	32797585	6600	ABC	

	SIT_ATIV	TIP_UNID	PAC_1	PAC_2	\
OBJECTID					
77	AT	14	83860474920074919MT	83860399920074986MT	
253	AT	13	55273310919119032MT	55273408919119041MT	

	CTMT	UNI_TR_S	SUB	CONJ	MUN	DAT_CON	BANC	POS	\
OBJECTID									
77	53149444		12873455	13722	2504009	01/01/1920	1	PD	
253	51696888		188425020	13691	2502409	25/08/2015	1	PD	

	DESCR
OBJECTID	
77	
253	

```
df_not.shape = (2, 17)
```

```
[22]: 12873455 in lista_SUBs
```

```
[22]: False
```

1.2.7 Filtro UNSEMT

```
[23]: base_ok['UNSEMT'], base_not_ok['UNSEMT'] = filtrar_por_SUB('UNSEMT', base, ↵  
↪ lista_SUBs)
```

	WKT	COD_ID	DIST	\
OBJECTID				
1	POINT (-35.7050706318042 -7.61082473027705)	24312960	6600	
2	POINT (-35.5900458995985 -7.18070067739433)	24312961	6600	
3	POINT (-36.8918768843107 -7.34120679735577)	24312963	6600	

	PAC_1	PAC_2	FAS_CON	SIT_ATIV	TIP_UNID	\
OBJECTID						
1	86362930915733974MT	86363082915733979MT	B	AT	22	
2	87670313920487277MT	87670162920487260MT	ABC	AT	22	
3	73272227918798438MT	73272101918798353MT	A	AT	22	

	P_N_OPE	CAP_ELO	...	TLCD	DAT_CON	POS	CTMT	UNI_TR_S	\
OBJECTID			...						
1	F	2H	...	0	01/01/1920	PD	26260949	397864865	
2	F	2H	...	0	01/01/1920	PD	26260976	281690694	
3	F	1H	...	0	01/01/1920	PD	139962991	12908668	

	SUB	CONJ	MUN	DESCR	ARE_LOC
OBJECTID					
1	12873450	13686	2501302		NU
2	12873447	13729	2507606		UB
3	12873453	16188	2508505		NU

[3 rows x 21 columns]

df.shape = (102873, 21)

	WKT	COD_ID	DIST	\
OBJECTID				
1779	POINT (-35.9182381810472 -7.78355066193348)	38336904	6600	
2489	POINT (-35.508735524225 -7.49956097276066)	46715444CT	6600	
2491	POINT (-35.5052773350809 -7.5023135412369)	46715446CT	6600	

	PAC_1	PAC_2	FAS_CON	SIT_ATIV	TIP_UNID	\
OBJECTID						
1779	83994336913839179MT	83994336913839331MT	A	AT	22	
2489	88542109916949174MT	88542109916949174ET	ABC	AT	22	
2491	88580089916918385MT	88580089916918385ET	C	AT	22	

	P_N_OPE	CAP_ELO	...	TLCD	DAT_CON	POS	CTMT	UNI_TR_S	\
OBJECTID			...						
1779	F	6K	...	0	09/08/2014	PD	26260908		
2489	F	0	...	0	01/01/1920	0	51696889		
2491	F	0	...	0	01/01/1920	0	51696889		

	SUB	CONJ	MUN	DESCR	\
OBJECTID					
1779	12873465	13743	2513158	Medição de Fronteira / SE particular	
2489	12873421	13686	2609006	Medição de Fronteira / SE particular	
2491	12873421	13686	2613800	Medição de Fronteira / SE particular	

	ARE_LOC
OBJECTID	
1779	NU
2489	NU
2491	NU

[3 rows x 21 columns]


```
df_not.shape = (111, 21)
```

1.2.8 Resultado

```
[24]: for key in base_ok.keys():  
       print(f'[{key}]: {base_ok[key].shape}')
```

```
[CTMT]: (346, 58)  
[UCMT]: (1890, 51)  
[UGMT]: (5, 46)  
[SSDMT]: (433966, 24)  
[UNCRMT]: (145, 19)  
[UNREMT]: (263, 17)  
[UNSEMT]: (102873, 21)
```

```
[25]: for key in base_not_ok.keys():  
       print(f'[{key}]: {base_not_ok[key].shape}')
```

```
[CTMT]: (14, 58)  
[UCMT]: (1, 51)  
[UGMT]: (0, 46)  
[SSDMT]: (497, 24)  
[UNCRMT]: (0, 19)  
[UNREMT]: (2, 17)  
[UNSEMT]: (111, 21)
```

1.3 Criação de Dataframe único com equipamentos de 2 portas

A idéia aqui é deixar a base de dados de entrada genérica:

- **base** significa usar os dados completos;
- **base_ok** significa usar apenas os registros que tem **SUB** na lista de subestações.

A escolha ocorre aqui:

```
[26]: work = base
```

```
[27]: df_2portas = pd.DataFrame(columns=['COD_ID', 'PAC_1', 'PAC_2', 'SUB', 'CTMT', 'KEY'])  
for key in work.keys():  
    if key in ['SUB', 'CTMT', 'UCMT', 'UGMT']:  
        continue  
  
    if key == 'UNSEMT':  
        dff = work[key].query('P_N_OPE == "F"')  
        dff = dff[['COD_ID', 'PAC_1', 'PAC_2', 'SUB', 'CTMT']]  
    else:  
        dff = work[key][['COD_ID', 'PAC_1', 'PAC_2', 'SUB', 'CTMT']].copy()
```

```

dff['KEY'] = key
df_2portas = pd.concat([df_2portas, dff], ignore_index=True)

display(df_2portas.head(3))
display(df_2portas.tail(3))
df_2portas.shape

```

	COD_ID	PAC_1	PAC_2	SUB	CTMT \
0	15223672	58224935927088283MT	58221888927070484MT	12873440	17067438
1	15223673	58191143927223327MT	58205538927212693MT	12873440	17067438
2	15223674	58211624927208161MT	58216190927204185MT	12873440	17067438

	KEY
0	SSDMT
1	SSDMT
2	SSDMT

	COD_ID	PAC_1	PAC_2	SUB \
536043	53089327CT	82244469921547437MT	82244469921547437ET	58561573
536044	53242965	82494551921692917MT	82494702921692936MT	58561573
536045	53242853	82403794921823971MT	82403696921823854MT	58561573

	CTMT	KEY
536043	184385076	UNSEMT
536044	184385076	UNSEMT
536045	184385076	UNSEMT

[27]: (536046, 6)

Na próxima etapa, uma busca em largura será feita para descobrir os trechos que formam cada um dos circuitos de MT.

Essa busca será feita de modo que novas colunas serão geradas em `df_2portas` visando validar os resultados obtidos:

- `circ_SUB`
- `circ_CTMT`
- `circ_PAC_de`
- `circ_PAC_pr`
- `circ_contador`

```

[28]: df_2portas['circ_SUB'] = -1
df_2portas['circ_CTMT'] = -1
df_2portas['circ_PAC_de'] = -1
df_2portas['circ_PAC_pr'] = -1
df_2portas['circ_cont'] = -1
df_2portas.head(3)

```

```
[28]:
```

	COD_ID	PAC_1	PAC_2	SUB	CTMT \
0	15223672	58224935927088283MT	58221888927070484MT	12873440	17067438
1	15223673	58191143927223327MT	58205538927212693MT	12873440	17067438
2	15223674	58211624927208161MT	58216190927204185MT	12873440	17067438

	KEY	circ_SUB	circ_CTMT	circ_PAC_de	circ_PAC_pr	circ_cont
0	SSDMT	-1	-1	-1	-1	-1
1	SSDMT	-1	-1	-1	-1	-1
2	SSDMT	-1	-1	-1	-1	-1

1.4 Criação dos circuitos por Busca em Largura

Existe um PAC em cada circuito de MT armazenado em CTMT que identifica sua conexão na SUB.

Esse PAC será usado como ponto de partida para a busca em largura.

De forma didática, faremos primeiro a busca referente ao primeiro circuito:

```
[29]: row = work['CTMT'].iloc[0]
sub = row['SUB']
pac = row['PAC']

print(f'sub = {sub}, pac = {pac}')
```

sub = 12873436, pac = 56560487917146437MT

```
[30]: def busca_em_largura(df, sub, pac_circ):
    flag = (df['circ_SUB'] == -1)
    dff = df[flag]

    C = [pac_circ,]
    D = set(list(dff['PAC_1']) + list(dff['PAC_2'])) - set(C)
    D = [item for item in D if len(item.strip()) > 0]
    cont = 0

    while(len(C) > 0):

        atual = C[0]
        # print('\n\n')
        # print(f'atual = {atual}\n')
        # print('====')
        # for item in C:
        #     print(f'C.item = {item}')
        #     print('====')

        flag1 = (dff['PAC_1'] == atual)
        # print(f'sum(flag1) = {sum(flag1)}\n')
```

```

    filhos = dff[flag1]
#    display(filhos)

    for idx, row in filhos.iterrows():
        if (row['PAC_2'] in D) or (len(row['PAC_2'].strip()) == 0):
#            print('Achou trecho!')
            df.loc[idx, 'circ_SUB'] = sub
            df.loc[idx, 'circ_CTMT'] = pac_circ
            df.loc[idx, 'circ_PAC_de'] = row['PAC_1']
            df.loc[idx, 'circ_PAC_pr'] = row['PAC_2']
            df.loc[idx, 'circ_cont'] = cont
            cont += 1

            if len(row['PAC_2'].strip()) > 0:
                D.remove(row['PAC_2'])
                C.append(row['PAC_2'])
            else:
#                print('Achou malha!')
                df.loc[idx, 'circ_SUB'] = 'Malha'
                df.loc[idx, 'circ_CTMT'] = 'Malha'
                df.loc[idx, 'circ_PAC_de'] = 'Malha'
                df.loc[idx, 'circ_PAC_pr'] = 'Malha'
#                breakpoint()

        flag2 = (dff['PAC_2'] == atual)
#        print(f'sum(flag2) = {sum(flag2)}\n')

    filhos = dff[flag2]
#    display(filhos)

    for idx, row in filhos.iterrows():
        if (row['PAC_1'] in D) or (len(row['PAC_1'].strip()) == 0):
#            print('Achou trecho!')
            df.loc[idx, 'circ_SUB'] = sub
            df.loc[idx, 'circ_CTMT'] = pac_circ
            df.loc[idx, 'circ_PAC_de'] = row['PAC_2']
            df.loc[idx, 'circ_PAC_pr'] = row['PAC_1']
            df.loc[idx, 'circ_cont'] = cont
            cont += 1

            if len(row['PAC_2'].strip()) > 0:
                D.remove(row['PAC_1'])
                C.append(row['PAC_1'])
            else:
#                print('Achou malha!')
                df.loc[idx, 'circ_SUB'] = 'Malha'
                df.loc[idx, 'circ_CTMT'] = 'Malha'

```

```

        df.loc[idx, 'circ_PAC_de'] = 'Malha'
        df.loc[idx, 'circ_PAC_pr'] = 'Malha'

    flag = (df['circ_SUB'] == -1)
    dff = df[flag]
#     print(f'dff.shape = {dff.shape}')
    if (cont % 500) == 0:
        print('+500 iterações.')
#     breakpoint()
    C.remove(atual)

```

```

[31]: df = df_2portas.copy()
      busca_em_largura(df, sub, pac)

```

```

+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.
+500 iterações.

```

Quantos trechos foram classificados como sendo da SUB?

```

[50]: flag = (df['circ_SUB'] == sub)
      sum(flag)

```

```

[50]: 10196

```

Quantos trechos classificados como sendo da SUB eram esperados?

```

[51]: flag = (df['SUB'] == sub)
      sum(flag)

```

[51]: 10207

Mostre os trechos em que houve diferença.

```
[53]: flag = (df['circ_SUB'] != -1) & (df['SUB'] != df['circ_SUB'])
df[flag]
```

```
[53]:
```

	COD_ID	PAC_1	PAC_2	SUB \
344743	15315511	56559101917147769MT	56559563917148551MT	12873436
440272	4755863160	57906902917786703MT	57906930917786713MT	12873436
440274	475586316INT	57906809917786666MT	57906902917786703MT	12873436
444105	169329330	55520569916508998MT	55520539916508997MT	12873436
444109	16932933INT	55520668916509009MT	55520569916508998MT	12873436
444502	4048082810	56190368916654074MT	56190355916654047MT	12873436
444505	404808281INT	56190411916654165MT	56190368916654074MT	12873436
457611	1132800240	56360583916917849MT	56360563916917826MT	12873436
457614	113280024INT	56360648916917924MT	56360583916917849MT	12873436
457630	3354430000	55641044916540243MT	55641015916540235MT	12873436
457844	335443000INT	55641141916540268MT	55641044916540243MT	12873436

	CTMT	KEY	circ_SUB	circ_CTMT	circ_PAC_de	circ_PAC_pr \
344743	17067308	SSDMT	Malha	Malha	Malha	Malha
440272	17067308	UNSEMT	Malha	Malha	Malha	Malha
440274	17067308	UNSEMT	Malha	Malha	Malha	Malha
444105	17067306	UNSEMT	Malha	Malha	Malha	Malha
444109	17067306	UNSEMT	Malha	Malha	Malha	Malha
444502	387937142	UNSEMT	Malha	Malha	Malha	Malha
444505	387937142	UNSEMT	Malha	Malha	Malha	Malha
457611	17067306	UNSEMT	Malha	Malha	Malha	Malha
457614	17067306	UNSEMT	Malha	Malha	Malha	Malha
457630	387937142	UNSEMT	Malha	Malha	Malha	Malha
457844	387937142	UNSEMT	Malha	Malha	Malha	Malha

	circ_cont
344743	-1
440272	-1
440274	-1
444105	-1
444109	-1
444502	-1
444505	-1
457611	-1
457614	-1
457630	-1
457844	-1

Observe que nos 11 trechos a busca em largura identificou o fechamento de malha e por isso não registrou como parte da subestação sub.

Há apenas um segmento de rede e o restante é chave.

O segmento é mostrado a seguir:

```
[49]: work['SSDMT'].loc[344743,:]
```

```
[49]: WKT                MULTILINESTRING ((-36.3920526712937 -6.6450393...
      COD_ID                23406177
      PN_CON_1              26619564
      PN_CON_2              26619714
      CTMT                  26260925
      UNI_TR_S              12913257
      SUB                   12912336
      CONJ                   13726
      DIST                   6600
      PAC_1                  78836021926473070MT
      PAC_2                  78869913926480504MT
      FAS_CON                C
      TIP_CND                16512332H-1
      POS                    PD
      ODI_FAS                1
      TI_FAS                 41
      ODI_NEU                0
      TI_NEU                 0
      COMP                   346.977
      DESCR
      SITCONFAS              AT1
      SITCONTNEU             0
      ARE_LOC                NU
      Shape_Length           0.003136
      Name: 344743, dtype: object
```

Por curiosidade, quantos trechos foram classificados como malha?

```
[54]: flag = (df['circ_SUB'] == 'Malha')
      sum(flag)
```

```
[54]: 11
```

Exatamente os 11 trechos. Era lógico... Os trechos diferentes de -1 são sub ou Malha.