# Removendo outliers Continuação

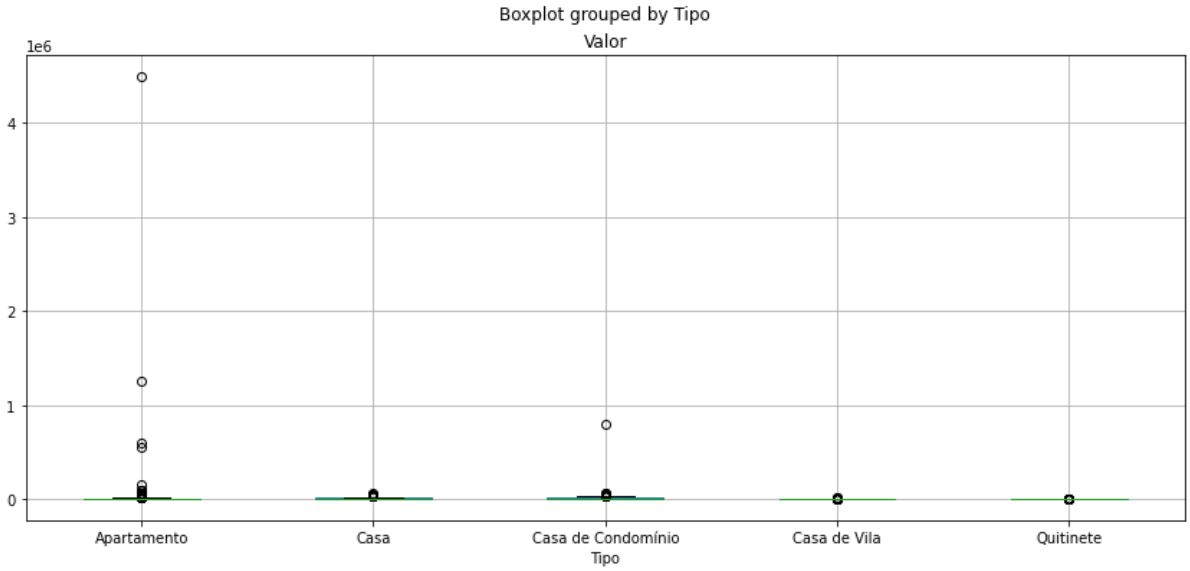
No notebook anterior, foi feita a remoção de outliers utilizando uma metodologia específica.
Neste notebook será usado a mesma metodologia, mas será feito o desagrupamento dos dados e uma análise modular.

In [1]:
```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('figure', figsize=(14, 6))
```

In [2]:
```
dados = pd.read_csv('../dados/aluguel_residencial_final.csv', sep=';')
```

In [3]:
```
# faz um boxplot individual pra cada tipo
dados.boxplot('Valor', by='Tipo')
```

Out[3]:
```
<AxesSubplot:title={'center':'Valor'}, xlabel='Tipo'>
```




Como esse agrupamento possui valores muito discrepantes, isso pode dificultar a visualização em conjunto

# Agrupando as variáveis

In [4]:
```
grupo_tipo = dados.groupby('Tipo')
type(grupo_tipo)
```

Out[4]:
```
pandas.core.groupby.generic.DataFrameGroupBy
```

## Agruparemos a variável tipo, mas dessa vez apenas para a variável valor, não para todo dataframe

In [5]:
```
grupo_tipo = dados.groupby('Tipo')['Valor']
type(grupo_tipo)
```

Out[5]:
```
pandas.core.groupby.generic.SeriesGroupBy
```

> Repare agora que esse agrupamento é do tipo SeriesGroupBy

## Visualizando o agrupamento

In [6]:
```
grupo_tipo.groups
```

Out[6]:
```
{'Apartamento': [2, 3, 4, 7, 8, 9, 11, 13, 14, 15, 17, 18, 19, 20, 21, 23, 2
4, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 43, 44, 4
5, 46, 47, 48, 49, 50, 51, 52, 53, 55, 56, 59, 60, 61, 62, 63, 64, 65, 66, 6
7, 68, 69, 70, 72, 73, 74, 75, 76, 77, 79, 80, 82, 83, 84, 85, 87, 88, 89, 9
0, 91, 92, 93, 94, 95, 97, 98, 99, 102, 103, 104, 105, 106, 107, 108, 109, 11
0, 111, 112, 113, 114, 115, 116, 117, 118, 119, ...], 'Casa': [1, 22, 54, 57,
96, 100, 144, 160, 180, 238, 250, 253, 264, 286, 310, 316, 334, 339, 357, 37
8, 385, 399, 424, 434, 474, 475, 481, 511, 542, 543, 567, 571, 572, 618, 621,
630, 635, 636, 660, 676, 700, 721, 724, 760, 771, 780, 856, 873, 937, 982, 10
29, 1036, 1118, 1123, 1125, 1157, 1178, 1249, 1256, 1316, 1335, 1350, 1371, 1
412, 1426, 1430, 1440, 1445, 1472, 1475, 1488, 1586, 1604, 1656, 1662, 1666,
1671, 1684, 1709, 1717, 1762, 1810, 1835, 1875, 1905, 1933, 1942, 1960, 2019,
2039, 2056, 2075, 2101, 2107, 2108, 2133, 2170, 2201, 2204, 2211, ...], 'Casa
de Condomínio': [5, 6, 12, 16, 42, 58, 166, 168, 183, 207, 222, 246, 259, 26
5, 279, 291, 308, 336, 391, 401, 440, 445, 449, 502, 556, 609, 622, 657, 663,
673, 707, 723, 781, 807, 809, 863, 883, 887, 934, 958, 961, 979, 986, 992, 10
04, 1008, 1028, 1082, 1095, 1112, 1129, 1148, 1158, 1182, 1220, 1227, 1229, 1
239, 1246, 1308, 1312, 1320, 1341, 1356, 1406, 1438, 1439, 1467, 1495, 1531,
1560, 1582, 1601, 1615, 1646, 1713, 1722, 1728, 1756, 1764, 1770, 1802, 1860,
1880, 1883, 1899, 1938, 2031, 2033, 2071, 2152, 2168, 2200, 2224, 2246, 2248,
2327, 2333, 2357, 2371, ...], 'Casa de Vila': [81, 212, 220, 303, 332, 697, 8
22, 844, 918, 1012, 1353, 1362, 1447, 1491, 1553, 1639, 1669, 1703, 1769, 208
7, 2249, 2267, 2446, 2533, 2547, 2605, 2641, 2727, 2840, 2872, 2977, 2984, 30
17, 3025, 3300, 3426, 3523, 3703, 3823, 3855, 3858, 3863, 4094, 4146, 4153, 4
165, 4340, 4444, 4826, 5151, 5170, 5175, 5198, 5294, 5410, 5535, 5597, 5724,
5751, 5911, 5950, 5995, 6008, 6031, 6049, 6201, 6236, 6300, 6348, 6402, 6429,
6754, 6795, 6939, 6957, 7033, 7091, 7146, 7296, 7697, 7712, 7778, 7837, 7843,
7968, 8004, 8136, 8427, 8452, 8578, 9229, 9234, 9319, 9476, 9619, 9624, 9716,
9739, 9784, 9867, ...], 'Quitinete': [0, 10, 28, 71, 78, 86, 101, 120, 146, 1
74, 191, 206, 223, 248, 301, 314, 327, 344, 355, 425, 426, 427, 460, 486, 53
2, 633, 650, 680, 808, 870, 917, 919, 924, 928, 939, 944, 970, 1001, 1016, 10
44, 1070, 1156, 1170, 1172, 1184, 1192, 1196, 1212, 1217, 1261, 1274, 1334, 1
351, 1360, 1393, 1404, 1407, 1483, 1496, 1510, 1543, 1595, 1611, 1613, 1633,
1696, 1697, 1706, 1733, 1753, 1772, 1824, 1839, 1853, 1910, 2013, 2085, 2098,
2125, 2142, 2149, 2156, 2160, 2227, 2237, 2239, 2258, 2272, 2326, 2362, 2382,
2383, 2384, 2394, 2445, 2457, 2462, 2493, 2507, 2630, ...]}
```

# Estatísticas da variável

In [19]:
```
# criamos novamente as estatísticas e os limites
# cada variável vira uma Series por tipo de imóvel
Q1 = grupo_tipo.quantile(.25)
Q3 = grupo_tipo.quantile(.75)
IIQ = Q3 - Q1
limite_inferior = Q1 - 1.5 * IIQ
limite_superior = Q3 + 1.5 * IIQ
```

**Agora cada variável é uma Series por Tipo**

In [8]:
```
Q1
```

Out[8]:
```
Tipo
Apartamento           1700.0
Casa                  1100.0
Casa de Condomínio    4000.0
Casa de Vila           750.0
Quitinete              900.0
Name: Valor, dtype: float64
```

In [9]:
```
Q3
```

Out[9]:
```
Tipo
Apartamento           5000.0
Casa                  9800.0
Casa de Condomínio   15250.0
Casa de Vila          1800.0
Quitinete             1500.0
Name: Valor, dtype: float64
```

In [10]:
```
IIQ
```

Out[10]:
```
Tipo
Apartamento           3300.0
Casa                  8700.0
Casa de Condomínio   11250.0
Casa de Vila          1050.0
Quitinete              600.0
Name: Valor, dtype: float64
```

In [11]:
```
limite_inferior
```

Out[11]:
```
Tipo
Apartamento          -3250.0
Casa                -11950.0
Casa de Condomínio  -12875.0
Casa de Vila          -825.0
Quitinete                0.0
Name: Valor, dtype: float64
```

In [12]:
```
limite_superior
```

Out[12]:
```
Tipo
Apartamento           9950.0
Casa                 22850.0
Casa de Condomínio   32125.0
Casa de Vila          3375.0
Quitinete             2400.0
Name: Valor, dtype: float64
```

## Acessando um tipo específico

In [20]:
```
limite_inferior['Apartamento']
```

Out[20]: -3250.0

In [13]:
```
limite_superior['Casa']
```

Out[13]: 22850.0

## Excluindo Outliers com Vários Grupos

In [14]:
```
for tipo in grupo_tipo.groups.keys():
    print(tipo)
```

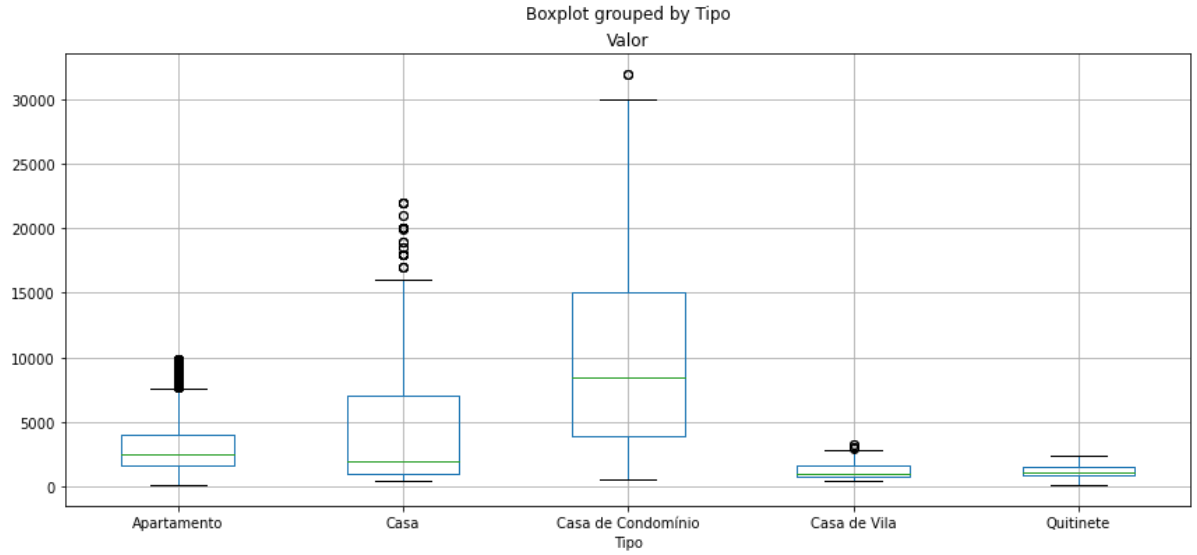
```
Apartamento
Casa
Casa de Condomínio
Casa de Vila
Quitinete
```

> Note que são acessadas as chaves do dicionário criado com o groupby.groups

In [22]:
```
dados_new = pd.DataFrame()
# gera um novo dataframe vazio para fazer a concatenação
for tipo in grupo_tipo.groups.keys():
    eh_tipo = dados['Tipo'] == tipo
    # seleciona o tipo de imóvel desejado nos dados
    eh_dentro_limite = (dados['Valor'] >= limite_inferior[tipo]) & (dados['Val
or'] <= limite_superior[tipo])
    # pega o tipo desejado apenas dentro da faixa estatistica desejada
    selecao = eh_tipo & eh_dentro_limite
    # junta as duas seleções
    dados_selecao = dados[selecao]
    # passa a seleção para um novo dataframe
    dados_new = pd.concat([dados_new, dados_selecao])
    # concatena dataframe gerado no laço em um outro novo dataframe
```

In [23]:
```
# agora é feita uma análise mais elaborada para cada grupo
dados_new.boxplot('Valor', by='Tipo')
```

Out[23]:
```
<AxesSubplot:title={'center':'Valor'}, xlabel='Tipo'>
```




Repare que os dados foram limpos e o boxplot foi aplicado para cada tipo de imóvel

## Exportando para utilizações futuras

In [17]:
```
dados_new.to_csv('../dados/aluguel_residencial_sem_outliers.csv', sep=';', ind
ex=False)
```