História das Olimpíadas - Parte 2

Você recentemente trabalhou (ou está trabalhando) em uma análise de dados históricos dos jogos olímpicos utilizando o Pandas para auxiliá-lo.

Desde que você iniciou seus trabalhos nesse projeto, novas ferramentas bastante poderosas foram ensinadas! O seu papel agora será utilizar essas novas ferramentas para gerar algumas visualizações que tornarão certas informações muito mais claras.

Utilize qualquer uma das bibliotecas estudadas (`matplotlib` , `seaborn` e `plotly`) para realizar as atividades propostas. Não há problema em usar apenas uma para realizar todas as atividades, nem em utilizar cada uma delas em uma atividade diferente - siga suas preferências pessoais!

Utilize os (muitos) parâmetros permitidos por cada função e/ou atributos dos objetos fornecidos pelas bibliotecas para criar uma identidade visual coesa para ser utilizada em todo o projeto. Use títulos, legendas e rótulos nos eixos para deixar os gráficos verdadeiramente informativos. E não se esqueça que a simples escolha das cores a serem utilizadas pode tornar os gráficos ainda mais interessantes!

Você utilizará o mesmo dataset fornecido no projeto anterior. Não há problemas em reaproveitar códigos do projeto anterior para economizar tempo e focar seus esforços na geração dos gráficos.

Para começar, importe o Pandas e carregue o arquivo `athlete_events.csv` fornecido no projeto anterior.

In [1]:
```
import pandas as pd

# Leitura do arquivo e pequena amostragem para identificar as colunas de dados
df_atletas = pd.read_csv('athlete_events.csv', encoding='UTF-8', index_col='ID')
df_atletas.head(3)
```

Out[1]:

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN

In [2]:
```
df_atletas = df_atletas.rename(axis=1,
                               mapper={'Name':'Nome','Sex':'Gênero','Age':'Idade', \
                               'Height':'Altura','Weight':'Peso','Team':'Time','NOC':'NOC','Games':'Edição', \
                                       'Year':'Ano','Season':'Temporada','City':'Cidade','Sport':'Esporte', \
                                       'Event':'Evento','Medal':'Medalha'})

# Adição de colunas úteis, como 0 ou 1 para cada tipo de medalha e uma coluna Qtd que será 1 se alguma medalha
# foi ganha pelo atleta, independente de qual.
# Finalmente, uma coluna "Ordem" para facilitar a apresenção Ouro -> Prata -> Bronze -> Nenhuma medalha
```

```
df_atletas['Ouros'], df_atletas['Pratas'], df_atletas['Bronzes'], df_atletas['Qtd'], df_atletas['Ordem'] = \
[df_atletas.Medalha.apply(lambda x : 1 if x == 'Gold' else 0), \
 df_atletas.Medalha.apply(lambda x : 1 if x == 'Silver' else 0), \
 df_atletas.Medalha.apply(lambda x : 1 if x == 'Bronze' else 0), \
 df_atletas.Medalha.apply(lambda x : 1 if x == 'Gold' or x == 'Silver' or x == 'Bronze' else 0), \
 df_atletas.Medalha.apply(lambda x : 1 if x == 'Gold' else 2 if x == 'Silver' else 3 if x == 'Bronze' else 4)]

df_atletas.head(3)
```

Out[2]:

ID	Nome	Gênero	Idade	Altura	Peso	Time	NOC	Edição	Ano	Temporada	Cidade	Esporte	Evento	Medalha	Ouros	Pratas	Bronzes	Qtd	Ordem
1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	0	0	0	0	4
2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	0	0	0	0	4
3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	0	0	0	0	4

1. O Brasil nas Olimpíadas

Vamos começar filtrando novamente os dados que iremos trabalhar. Crie um DataFrame contendo apenas informações sobre atletas **medalhistas** brasileiros.

In [3]:
```
df_meds_br = df_atletas[(df_atletas.NOC == 'BRA') & (df_atletas.Qtd > 0)]
df_meds_br.head(3)
```

Out[3]:

ID	Nome	Gênero	Idade	Altura	Peso	Time	NOC	Edição	Ano	Temporada	Cidade	Esporte	Evento	Medalha	Ouros	Pratas	Bronzes	Qtd	Ordem
918	Ademir Roque Kaefer	M	24.0	179.0	74.0	Brazil	BRA	1984 Summer	1984	Summer	Los Angeles	Football	Football Men's Football	Silver	0	1	0	1	2
918	Ademir Roque Kaefer	M	28.0	179.0	74.0	Brazil	BRA	1988 Summer	1988	Summer	Seoul	Football	Football Men's Football	Silver	0	1	0	1	2
925	Adenzia Aparecida Ferreira da Silva	F	25.0	187.0	65.0	Brazil	BRA	2012 Summer	2012	Summer	London	Volleyball	Volleyball Women's Volleyball	Gold	1	0	0	1	1

Vamos caracterizar fisicamente nossos medalhistas, verificando se há alguma correlação entre o desempenho em certos esportes e o tipo físico dos atletas.

Gere um gráfico de barras contendo os diferentes esportes no eixo X e a altura dos atletas no eixo Y. Utilize barras lado-a-lado para separar atletas do sexo masculino e feminino.

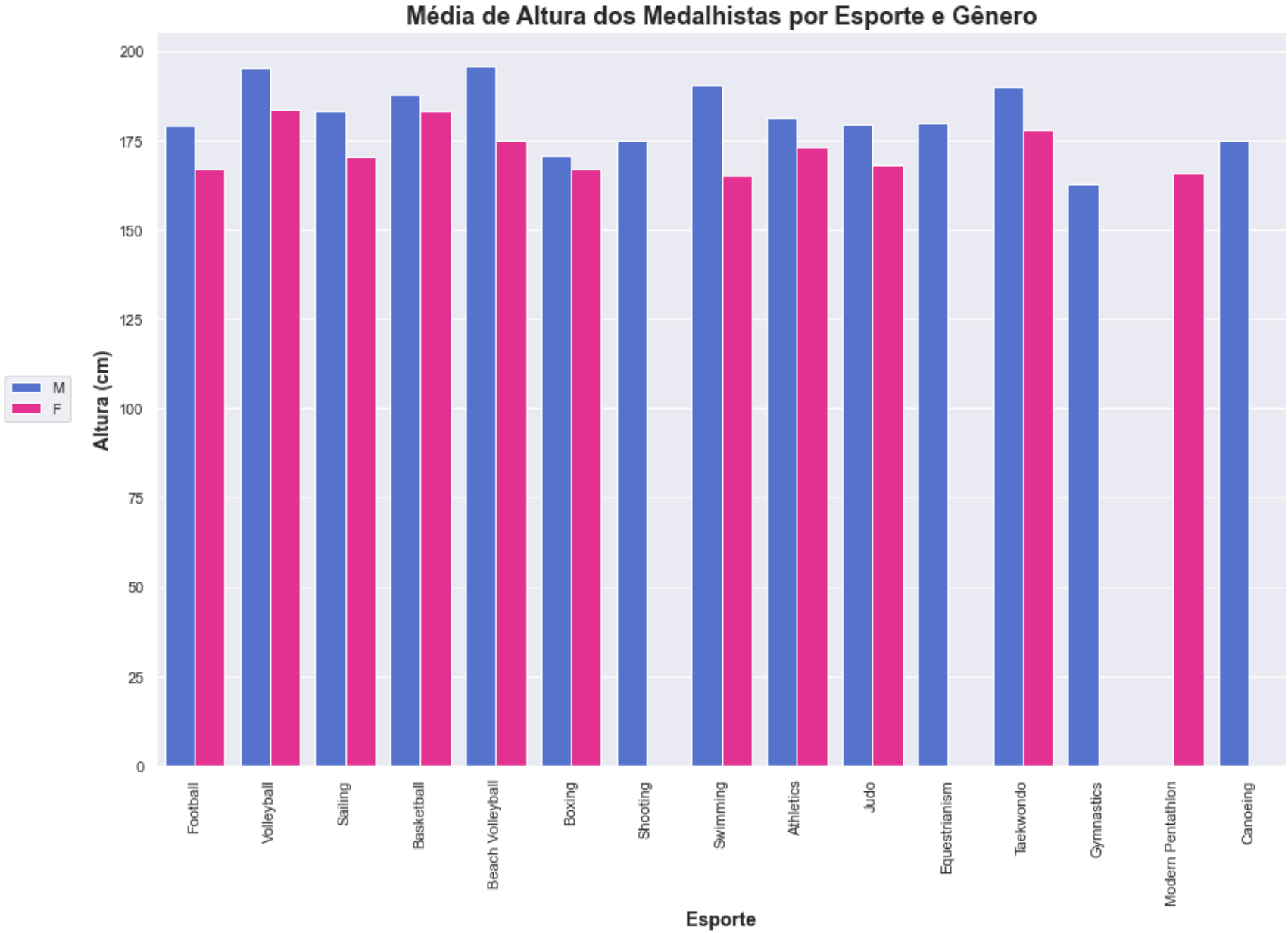
In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
```

```
import seaborn as sns
%matplotlib inline

# Configura o tema para os gráficos em linhas gerais
sns.set_theme(style='darkgrid', palette='bright')

# Essa lista de cores é utilizada para padronizar os gráficos de pizza
cores_pizza = ['gold','silver','goldenrod','darkorange','darkgray','peru','khaki','gainsboro','chocolate',
               'orangered','coral','limegreen','salmon'];

# Funções que serão chamadas antes e após desenho dos gráficos para melhorar a apresentação
'''
Configura dimensões para um novo gráfico
'''
def pre_g(largura=15, altura=10):
    plt.figure(figsize=(largura, altura))

'''
Pós-composição para gráficos
Customiza título do gráfico, títulos dos eixos X e Y, legenda
'''
def config_g(grafico, tit, tit_y=None, tit_x=None, rot_x=0, legenda=None, ancora_leg=(0,0)):

    # Titulo do gráfico
    plt.title(tit, fontsize=18, fontweight='bold')
    # Rotação do eixo X
    plt.xticks(rotation=rot_x)
    # Títulos dos eixos X e Y
    if tit_x is None:
        try:
            tit_x = grafico.get_xlabel()
        except:
            pass

    if tit_x is not None:
        grafico.set_xlabel(tit_x)
        grafico.xaxis.get_label().set_fontsize(14)
        grafico.xaxis.get_label().set_fontweight('bold')

    if tit_y is None:
        try:
            tit_y = grafico.get_ylabel()
        except:
            pass

    if tit_y is not None:
        grafico.set_ylabel(tit_y)
        grafico.yaxis.get_label().set_fontsize(14)
        grafico.yaxis.get_label().set_fontweight('bold')

    # Legenda
```

```
    if legenda is not None:
        g.legend(loc=legenda, bbox_to_anchor=ancora_leg);
```

In [5]:

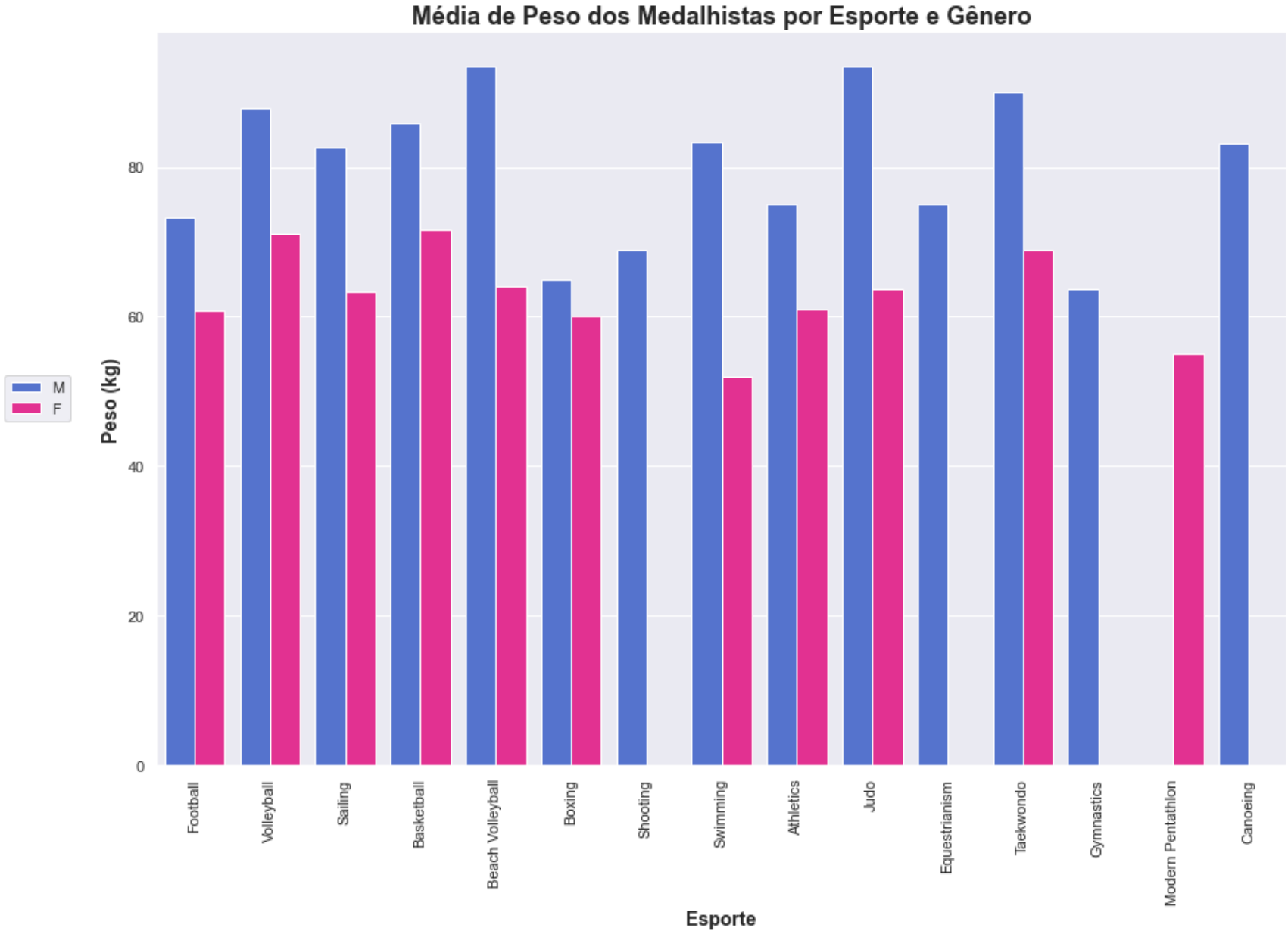
```
pre_g()

g = sns.barplot(data=df_meds_br, x='Esporte', y='Altura', hue='Gênero', ci=None, palette=['royalblue', 'deeppink']);

# Azul para H, Rosa para M e títulos do eixo X rotacionados para melhor apresentação
# Fontes dos títulos destacadas
# Legenda em posição customizada
config_g(g, tit='Média de Altura dos Medalhistas por Esporte e Gênero', \
         tit_y='Altura (cm)', rot_x=90, legenda='center right', ancora_leg=(-0.07, 0.5))
```



Agora gere um gráfico semelhante ilustrando o peso dos atletas.

In [6]:
```
pre_g()

g = sns.barplot(data=df_meds_br, x='Esporte', y='Peso', hue='Gênero', ci=None, palette=['royalblue', 'deeppink']);
```

```
config_g(g, tit='Média de Peso dos Medalhistas por Esporte e Gênero', \
         tit_y='Peso (kg)', rot_x=90, legenda='center right', ancora_leg=(-0.07, 0.5))
```

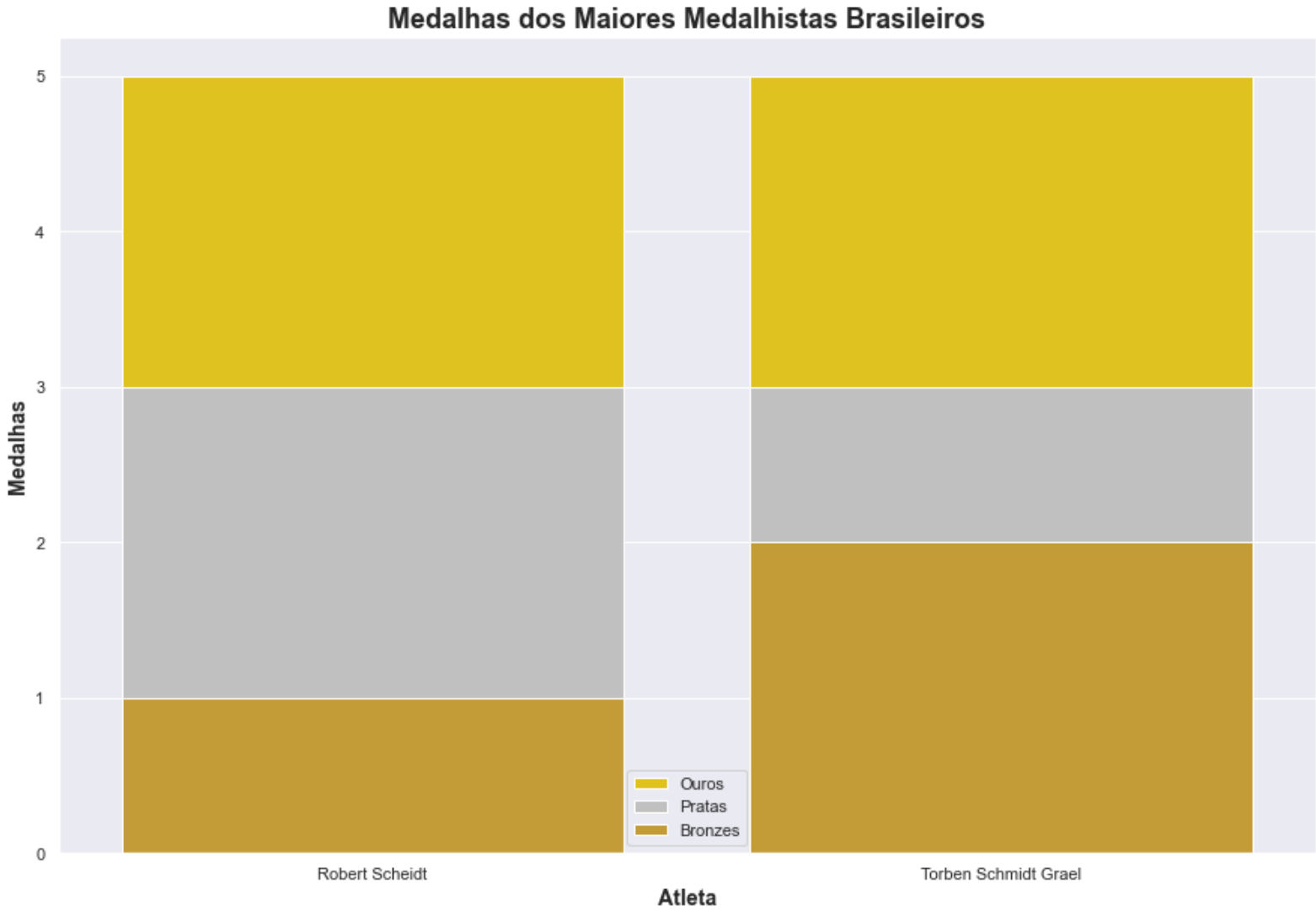


Vamos analisar agora as medalhas que nossos atletas trouxeram para casa.

Encontre os maiores medalhistas brasileiros em **total de medalhas**. Em seguida, faça um gráfico de barras empilhadas. No eixo X coloque o nome dos atletas, e no eixo Y coloque o número de medalhas. Utilize as barras empilhadas para mostrar, respectivamente, as medalhas de bronze, prata e ouro de cada atleta.

In [7]:
```
#import matplotlib.patches as mpatches
df_meds_atleta_br = df_meds_br.groupby(by='Nome').sum()
df_mai_atleta_br = df_meds_atleta_br[df_meds_atleta_br.Qtd == df_meds_atleta_br.Qtd.max()]
df_mai_atleta_br.reset_index(inplace=True)
```

In [8]:
```
pre_g()

# Para empilhar as barras, somei os valores das medalhas inferiores (truque)
# Sinceramente, não encontrei uma forma mais simples de alcançar esse resultado

# Ouros
sns.barplot(data=df_mai_atleta_br, x=df_mai_atleta_br.Nome,
            y=df_mai_atleta_br.Ouros+df_mai_atleta_br.Pratas+df_mai_atleta_br.Bronzes, color='gold',
            label='Ouros')
# Pratas
sns.barplot(data=df_mai_atleta_br, x=df_mai_atleta_br.Nome,
            y=df_mai_atleta_br.Pratas+df_mai_atleta_br.Bronzes, color='silver',
            label='Pratas')
# Bronzes
g = sns.barplot(data=df_mai_atleta_br, x=df_mai_atleta_br.Nome,
            y=df_mai_atleta_br.Bronzes, color='goldenrod',
            label='Bronzes')

# Legenda customizada
plt.legend(loc=8)

# Títulos
config_g(g, tit='Medalhas dos Maiores Medalhistas Brasileiros', tit_x='Atleta', tit_y='Medalhas')
```



Agora gere o mesmo gráfico de barras empilhadas substituindo os nomes dos atletas pelo nome de todos os esportes onde o Brasil já ganhou medalhas.

DICA: tome muito cuidado nessa análise: cada **evento esportivo** rende 1 medalha. Por exemplo, quando a equipe de futebol vence, isso é considerado 1 medalha, mesmo tendo cerca de 20 atletas medalhistas na equipe.

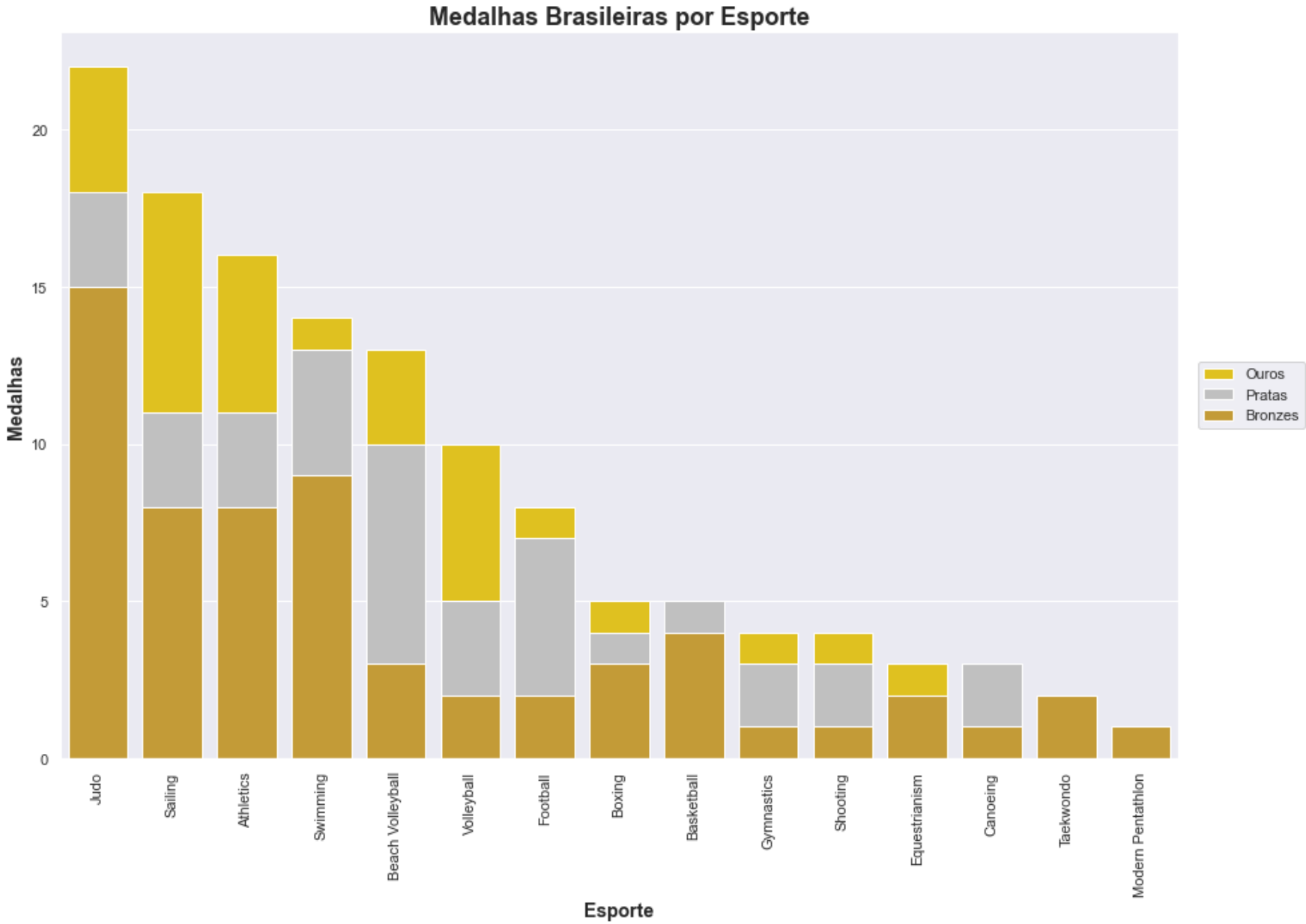
In [9]:
```
df_por_evento_br = df_meds_br[['Temporada','Ano','Esporte','Evento','Ouros','Pratas','Bronzes','Qtd','Ordem']].drop_duplicates()
df_por_esporte_br = df_por_evento_br.groupby('Esporte').sum()

# Por ordem decrescente, para rápida identificação dos esportes mais vencedores
df_por_esporte_br = df_por_esporte_br.sort_values(['Qtd','Ouros','Pratas','Bronzes'],ascending=False)
df_por_esporte_br.reset_index(inplace=True)
df_por_esporte_br.head(3)
```

Out[9]:

	Esporte	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	Judo	44024	4	3	15	22	55
1	Sailing	35904	7	3	8	18	37
2	Athletics	31748	5	3	8	16	35

In [10]:

```
pre_g()

sns.barplot(data=df_por_esporte_br, x=df_por_esporte_br.Esporte,
            y=df_por_esporte_br.Ouros+df_por_esporte_br.Pratas+df_por_esporte_br.Bronzes, color='gold', label='Ouros', ci=None)

sns.barplot(data=df_por_esporte_br, x=df_por_esporte_br.Esporte,
            y=df_por_esporte_br.Pratas+df_por_esporte_br.Bronzes, color='silver', label='Pratas', ci=None)

g = sns.barplot(data=df_por_esporte_br, x=df_por_esporte_br.Esporte,
            y=df_por_esporte_br.Bronzes, color='goldenrod', label='Bronzes', ci=None)

config_g(g, tit='Medalhas Brasileiras por Esporte', tit_x='Esporte', rot_x=90, tit_y='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```



Mais um gráfico de barras empilhadas: agora mostre os **eventos esportivos** que renderam medalhas para o Brasil.

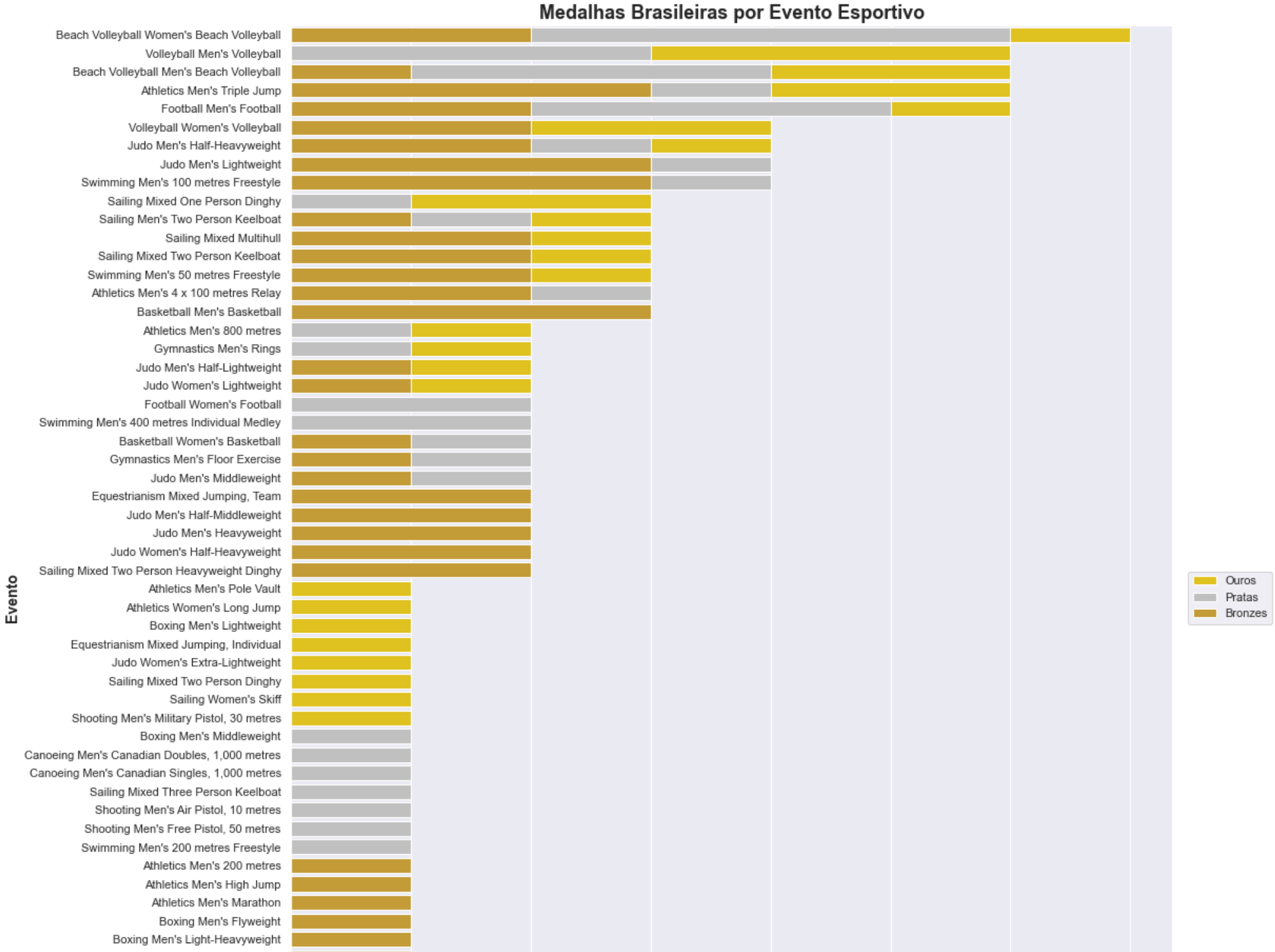
Lembrando: cada "categoria" dentro de um esporte é considerado um evento. Por exemplo, dentro de "atletismo", temos uma competição de 100m masculina, uma de 100m feminino, um revezamento 4 x 100m masculino, um revezamento 4 x 100m feminino, uma competição de 400m masculino, uma de 400m feminino, uma maratona masculina, uma maratona feminina, e assim sucessivamente.

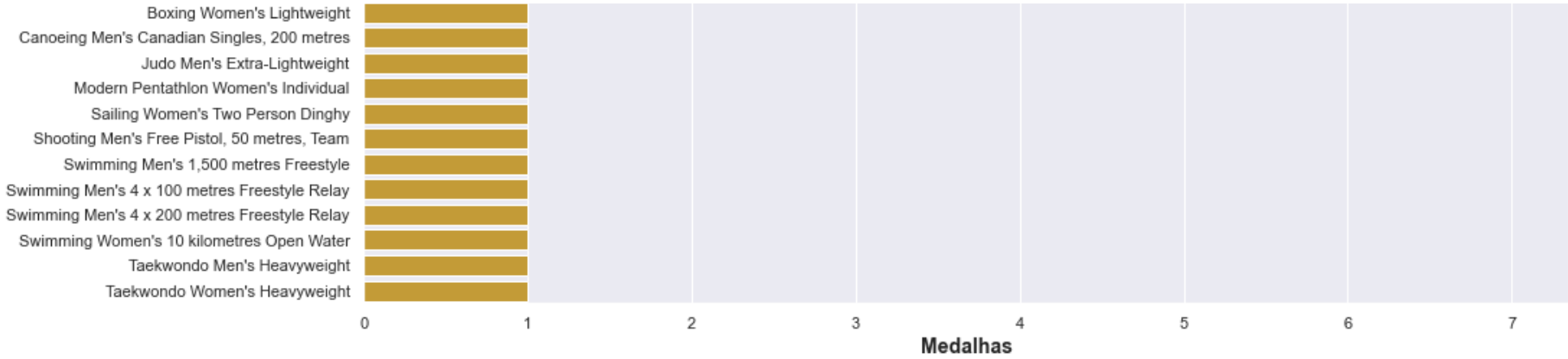
In [11]:
```
df_evento_br = df_meds_br[['Temporada','Ano','Esporte','Evento','Ouros','Pratas','Bronzes','Qtd','Ordem']].drop_duplicates()
df_evento_br = df_evento_br.groupby('Evento').sum()
df_evento_br = df_evento_br.sort_values(['Qtd','Ouros','Pratas','Bronzes'],ascending=False)
df_evento_br.reset_index(inplace=True)
df_evento_br.head(3)
```

Out[11]:

	Evento	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	Beach Volleyball Women's Beach Volleyball	14024	1	4	2	7	15
1	Volleyball Men's Volleyball	12016	3	3	0	6	9
2	Beach Volleyball Men's Beach Volleyball	12048	2	3	1	6	11

In [12]:
```
# Para esse gráfico utilizei barras horizontais para uma melhor visualização dos dados

pre_g(altura=20)

sns.barplot(data=df_evento_br, orient='h', y=df_evento_br.Evento,
            x=df_evento_br.Ouros+df_evento_br.Pratas+df_evento_br.Bronzes, color='gold',
            label='Ouros')

sns.barplot(data=df_evento_br, orient='h', y=df_evento_br.Evento,
            x=df_evento_br.Pratas+df_evento_br.Bronzes, color='silver',
            label='Pratas')

g = sns.barplot(data=df_evento_br, orient='h', y=df_evento_br.Evento,
            x=df_evento_br.Bronzes, color='goldenrod',
            label='Bronzes')

config_g(g, tit='Medalhas Brasileiras por Evento Esportivo', tit_y='Evento', tit_x='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```


Medalhas Brasileiras por Evento Esportivo
Evento
Beach Volleyball Women's Beach Volleyball
Volleyball Men's Volleyball
Beach Volleyball Men's Beach Volleyball
Athletics Men's Triple Jump
Football Men's Football
Volleyball Women's Volleyball
Judo Men's Half-Heavyweight
Judo Men's Lightweight
Swimming Men's 100 metres Freestyle
Sailing Mixed One Person Dinghy
Sailing Men's Two Person Keelboat
Sailing Mixed Multihull
Sailing Mixed Two Person Keelboat
Swimming Men's 50 metres Freestyle
Athletics Men's 4 x 100 metres Relay
Basketball Men's Basketball
Athletics Men's 800 metres
Gymnastics Men's Rings
Judo Men's Half-Lightweight
Judo Women's Lightweight
Football Women's Football
Swimming Men's 400 metres Individual Medley
Basketball Women's Basketball
Gymnastics Men's Floor Exercise
Judo Men's Middleweight
Equestrianism Mixed Jumping, Team
Judo Men's Half-Middleweight
Judo Men's Heavyweight
Judo Women's Half-Heavyweight
Sailing Mixed Two Person Heavyweight Dinghy
Athletics Men's Pole Vault
Athletics Women's Long Jump
Boxing Men's Lightweight
Equestrianism Mixed Jumping, Individual
Judo Women's Extra-Lightweight
Sailing Mixed Two Person Dinghy
Sailing Women's Skiff
Shooting Men's Military Pistol, 30 metres
Boxing Men's Middleweight
Canoeing Men's Canadian Doubles, 1,000 metres
Canoeing Men's Canadian Singles, 1,000 metres
Sailing Mixed Three Person Keelboat
Shooting Men's Air Pistol, 10 metres
Shooting Men's Free Pistol, 50 metres
Swimming Men's 200 metres Freestyle
Athletics Men's 200 metres
Athletics Men's High Jump
Athletics Men's Marathon
Boxing Men's Flyweight
Boxing Men's Light-Heavyweight
Ouros
Pratas
Bronzes



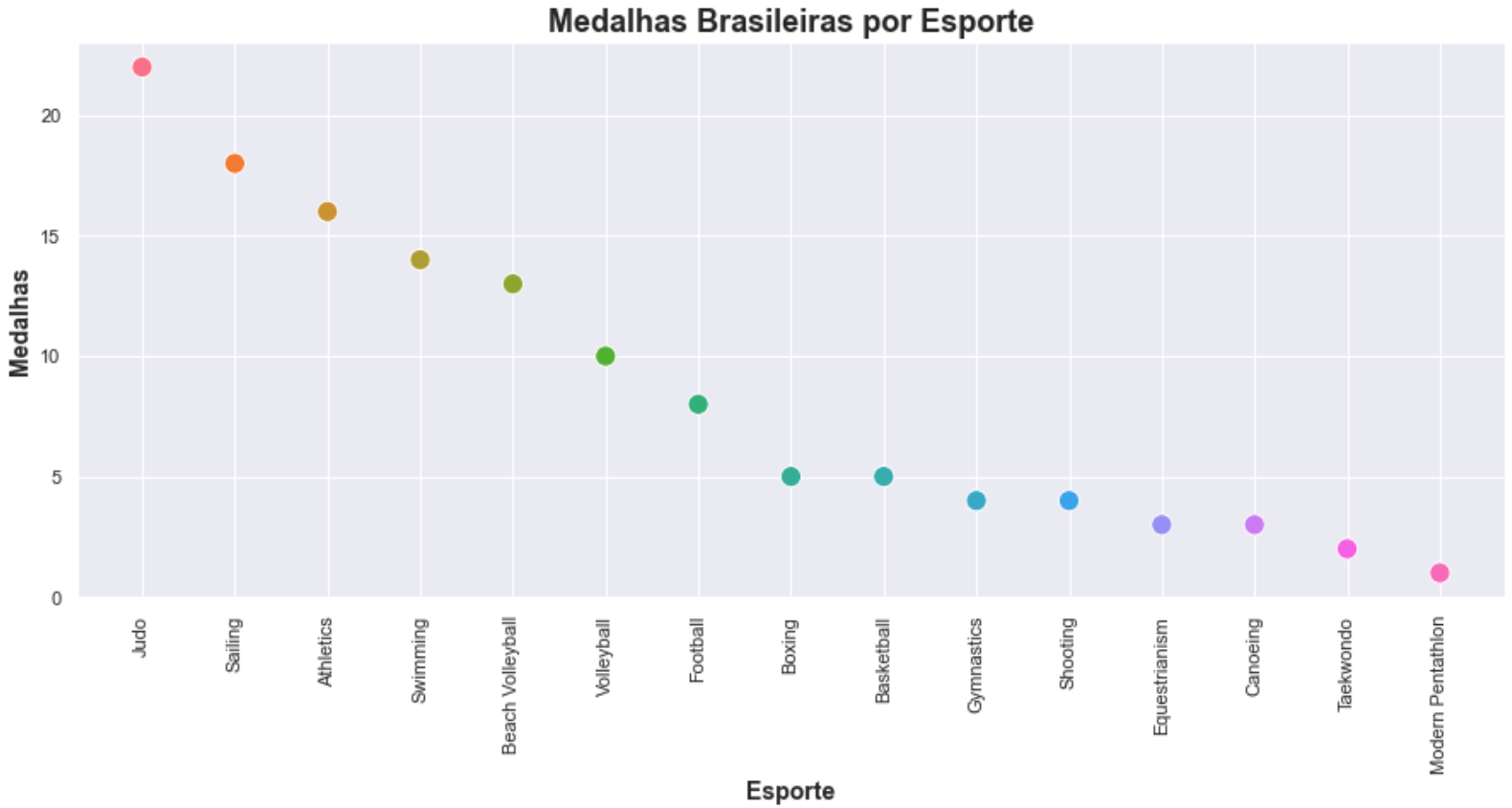
Utilize um gráfico de distribuição (como um histograma, por exemplo) ilustrando a quantidade total de medalhas do Brasil por esporte.

In [13]:
```
df_esporte_br = df_por_evento_br.groupby('Esporte').sum()
df_esporte_br.reset_index(inplace=True)
df_esporte_br.head(3)
```

Out[13]:

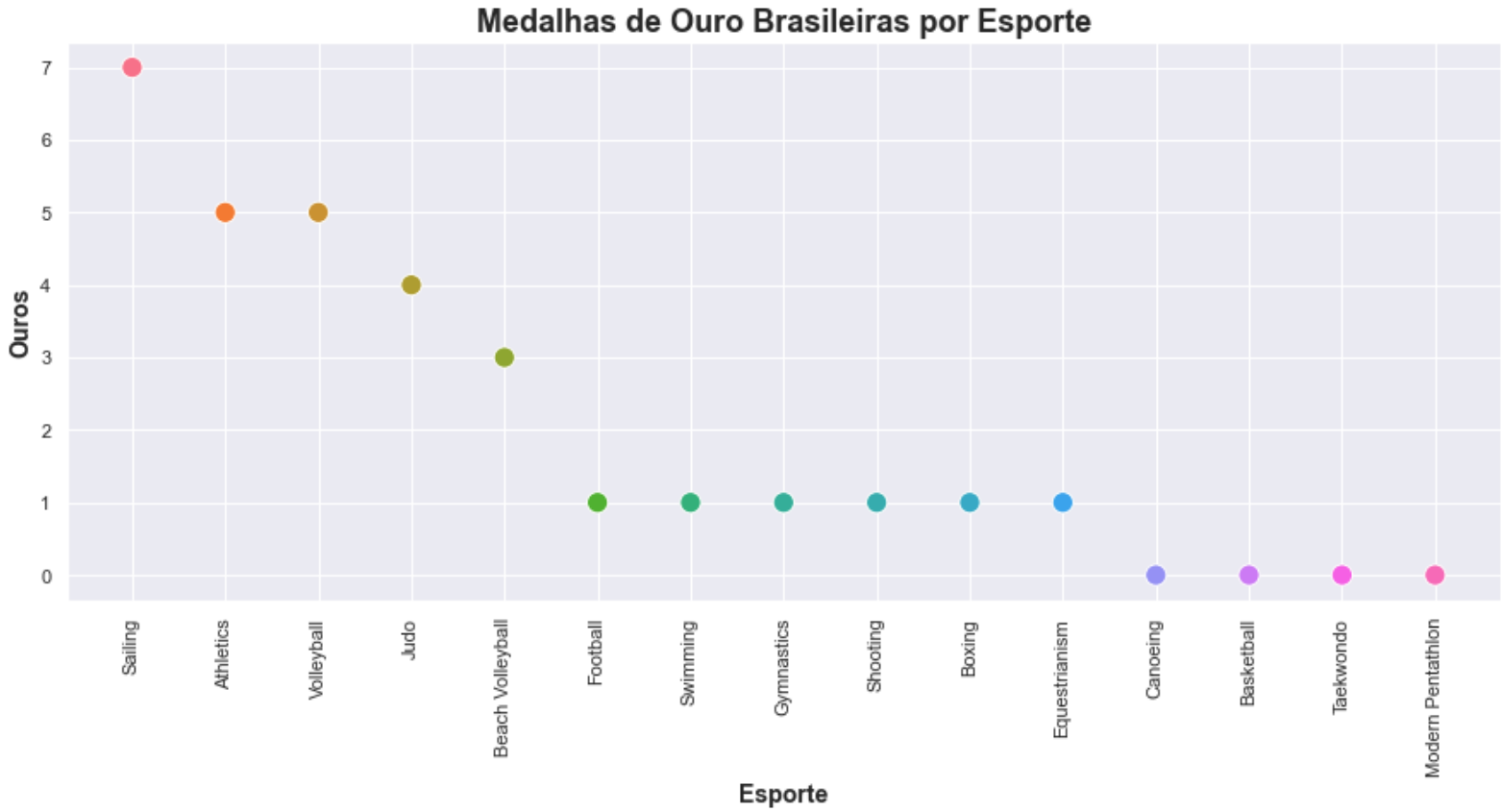
	Esporte	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	Athletics	31748	5	3	8	16	35
1	Basketball	9868	0	1	4	5	14
2	Beach Volleyball	26072	3	7	3	13	26

In [14]:
```
pre_g(altura=6)

# Hue utilizado para maior sofisticação visual. Tamanho do marcador customizado. Legenda automática omitida já que os esportes
# já estão no eixo X; Dados ordenados de forma decrescente, que permite identificar rapidamente os esportes mais vencedores
g = sns.scatterplot(data=df_por_esporte_br, x='Esporte', y='Qtd', hue='Esporte', s=150, legend=False)

config_g(g, tit='Medalhas Brasileiras por Esporte', tit_y='Medalhas', tit_x='Esporte', rot_x=90)
```



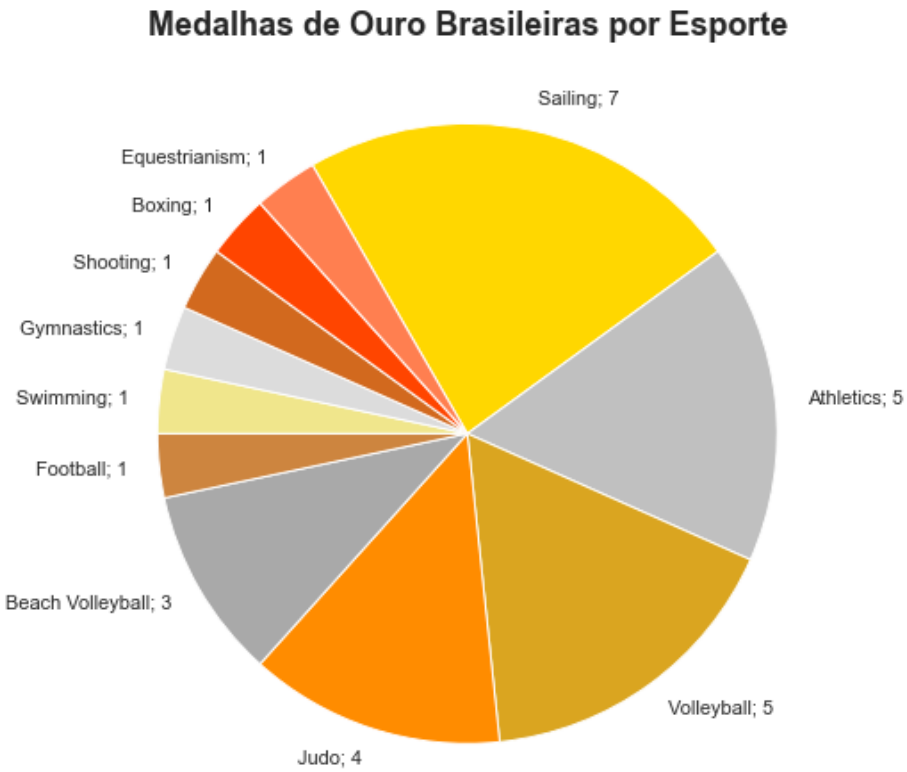
Repita o procedimento acima, mas com medalhas de ouro.

In [25]:
```
pre_g(altura=6)

df_ouros_esporte_br = df_por_esporte_br.sort_values(['Ouros','Pratas','Bronzes'], ascending=False)

# Similar ao gráfico anterior
g = sns.scatterplot(data=df_ouros_esporte_br, x='Esporte', y='Ouros', hue='Esporte', s=150, legend=False)

config_g(g, tit='Medalhas de Ouro Brasileiras por Esporte', tit_y='Ouros', tit_x='Esporte', rot_x=90)
```



Agora faça um gráfico de setores (pizza) mostrando a distribuição de medalhas de ouro do Brasil por esporte.

In [26]:
```
# Seaborn não possui gráficos de pizza, logo será utilizada a versão da Matplotlib

df_pizza = df_ouros_esporte_br[df_ouros_esporte_br.Ouros > 0][['Esporte','Ouros']]

# Para tornar a visualização mais direta, acrescentamos o total de medalha de cada esporte à sua direita, como título da fatia
df_pizza['Rotulos'] = df_pizza['Esporte']+'; '+df_pizza['Ouros'].astype(str)

pre_g(altura=8)

# O ângulo customizado visa impedir a sobreposição dos rótulos dos esportes com menos medalhas
g = plt.pie(df_pizza.Ouros, labels=df_pizza.Rotulos, counterclock=False, startangle=120, colors=cores_pizza);

config_g(g, tit='Medalhas de Ouro Brasileiras por Esporte')
```



Para finalizar a história do Brasil, vamos ver a série temporal de medalhas brasileiras. Crie um gráfico de linhas contendo 3 linhas: ouro, prata e bronze. Coloque no eixo X a edição da olimpíada (em ordem cronológica) e no eixo Y o total de medalhas de cada tipo.

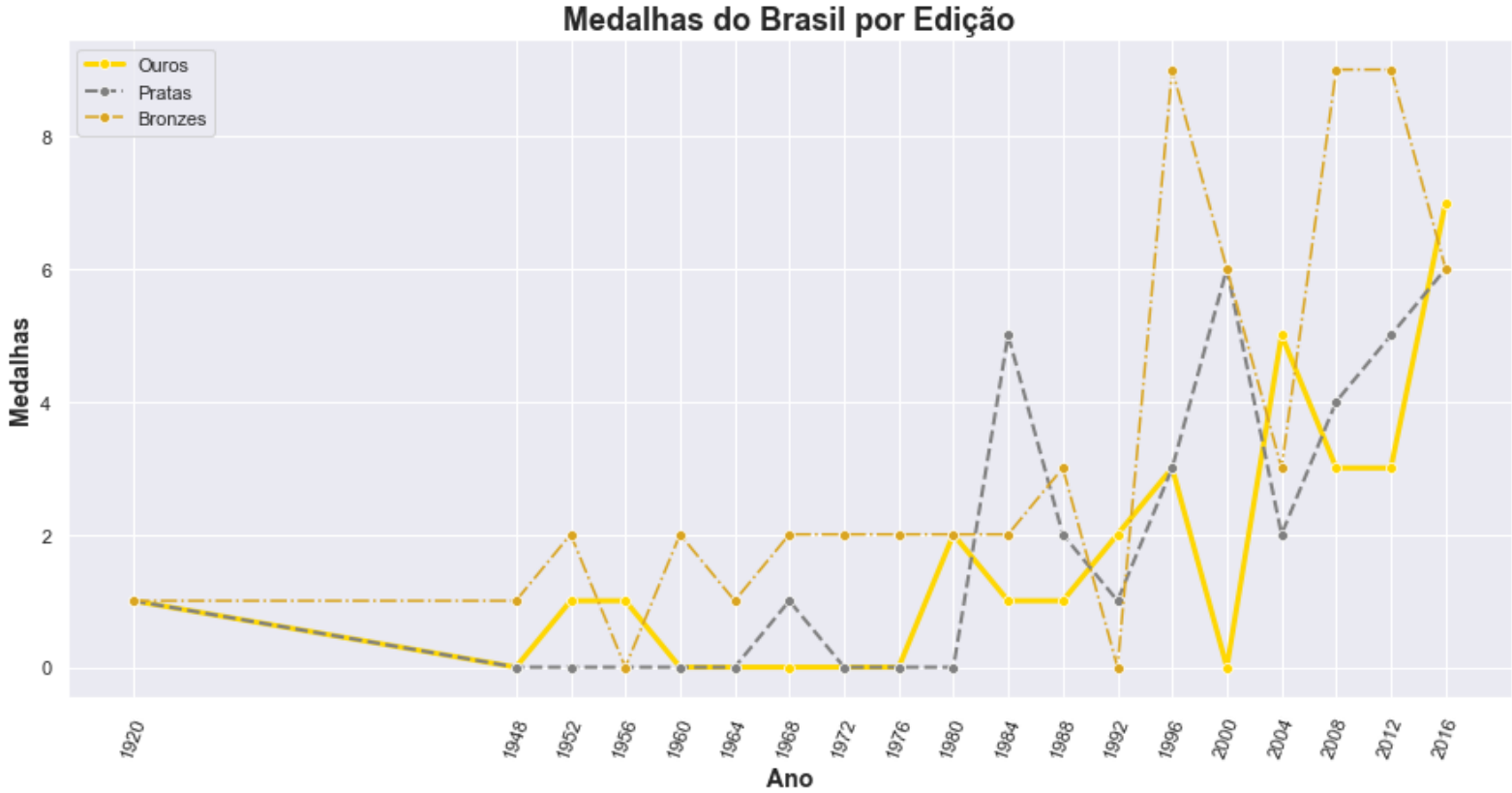
In [27]:
```
df_hist_br = df_meds_br[['Temporada','Ano','Esporte','Evento','Ouros','Pratas','Bronzes','Qtd','Ordem']].drop_duplicates()
df_hist_br = df_hist_br.groupby('Ano').sum()
df_hist_br.reset_index(inplace=True)
df_hist_br.head(5)
```

Out[27]:

	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	1920	1	1	1	3	6
1	1948	0	0	1	1	3
2	1952	1	0	2	3	7
3	1956	1	0	0	1	1
4	1960	0	0	2	2	6

In [28]:
```
pre_g(altura=7)
```

```
# Para cada linha vamos utlizar uma espessura e padrão de pontilhado para facilitar a distinção
g1 = sns.lineplot(data=df_hist_br, x='Ano', y='Ouros', color='gold', label='Ouros',
                  marker='o', linewidth=3)

g2 = sns.lineplot(data=df_hist_br, x='Ano', y='Pratas', color='gray', label='Pratas',
                  marker='o', linestyle='--', linewidth=2)

g3 = sns.lineplot(data=df_hist_br, x='Ano', y='Bronzes', color='goldenrod', label='Bronzes',
                  marker='o', linestyle='-.')

config_g(g1, tit='Medalhas do Brasil por Edição', tit_y='Medalhas', tit_x='Ano', rot_x=70)

# Customizada a escala do eixo X para apresentar todos os anos
plt.xticks(df_hist_br.Ano);
plt.legend();
```



2. O mundo nos jogos de verão

Filtre o DataFrame original para conter apenas informações sobre os **medalhistas** de todos os países **nos jogos de verão**.

In [29]:
```
df_med_mundo = df_atletas[(df_atletas.Qtd > 0) & (df_atletas.Temporada == 'Summer')]
df_med_mundo.head(3)
```

Out[29]:

ID	Nome	Gênero	Idade	Altura	Peso	Time	NOC	Edição	Ano	Temporada	Cidade	Esporte	Evento	Medalha	Ouros	Pratas	Bronzes	Qtd	Ordem
4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	1	0	0	1	1
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 200 metres Breaststroke	Bronze	0	0	1	1	3
15	Arvo Ossian Aaltonen	M	30.0	NaN	NaN	Finland	FIN	1920 Summer	1920	Summer	Antwerpen	Swimming	Swimming Men's 400 metres Breaststroke	Bronze	0	0	1	1	3

Utilizando subplots, crie 2 boxplots ilustrando a quantidade de medalhas por atleta. Em um deles, considere todos os atletas. No segundo, experimente remover os *outliers*.

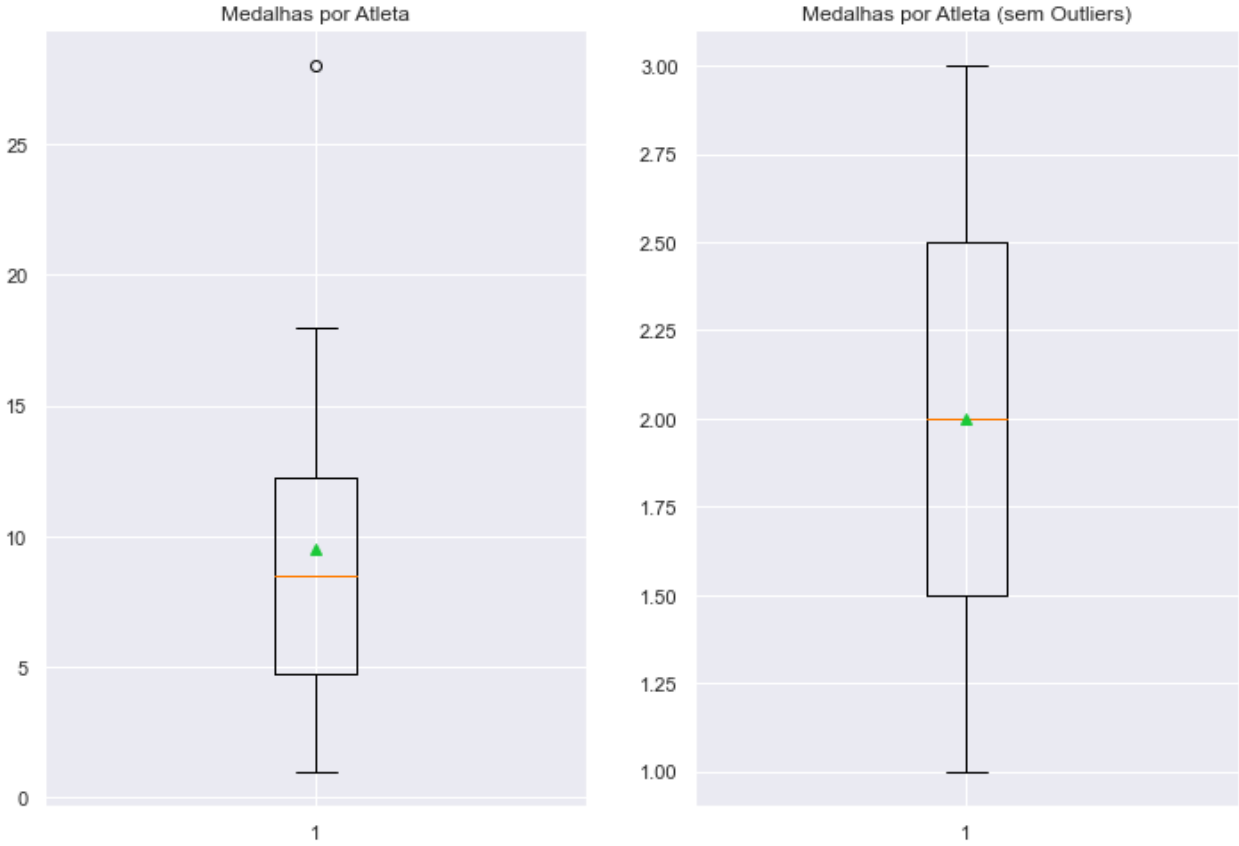
In [30]:
```
# Gráfico da linha 1 coluna 1 (gráfico 1)
df_qtd_atleta = df_med_mundo[['Nome','Qtd']].groupby('Nome').sum()
df_qtd_atleta.describe()
```

Out[30]:

	Qtd
count	24545.000000
mean	1.388796
std	0.939906
min	1.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	28.000000

In [31]:
```
plt.figure(figsize=(12,8))

df_qtd_1 = df_qtd_atleta.groupby('Qtd').count()
df_qtd_1.reset_index(inplace=True)

plt.subplot(121)
plt.boxplot(df_qtd_1, showmeans=True);
```

```
plt.title('Medalhas por Atleta');

# Removendo os outliers
#df_qtd_atleta_nout = df_qtd_atleta[df_qtd_atleta.Qtd <= df_qtd_atleta.Qtd.quantile(0.95)]
df_qtd_2 = df_qtd_atleta[df_qtd_atleta.Qtd <= df_qtd_atleta.Qtd.quantile(0.95)]
df_qtd_2 = df_qtd_2.groupby('Qtd').count()
df_qtd_2.reset_index(inplace=True)

# Gráfico da linha 1 coluna 2 (gráfico 2)
plt.subplot(122)
plt.boxplot(df_qtd_2, showmeans=True);
#plt.boxplot(x=df_qtd_2, showmeans=True);
plt.title('Medalhas por Atleta (sem Outliers)');
```



Descubra o total de medalhas de ouro de cada país (lembrando-se da restrição dos eventos esportivos, para não contabilizar múltiplas medalhas em esportes de equipe!).

Agora pegue os 10 países com mais medalhas e crie uma categoria "Outros" para o restante dos países. Exiba um gráfico de pizza mostrando a distribuição de medalhas de ouro entre essas 11 "equipes".

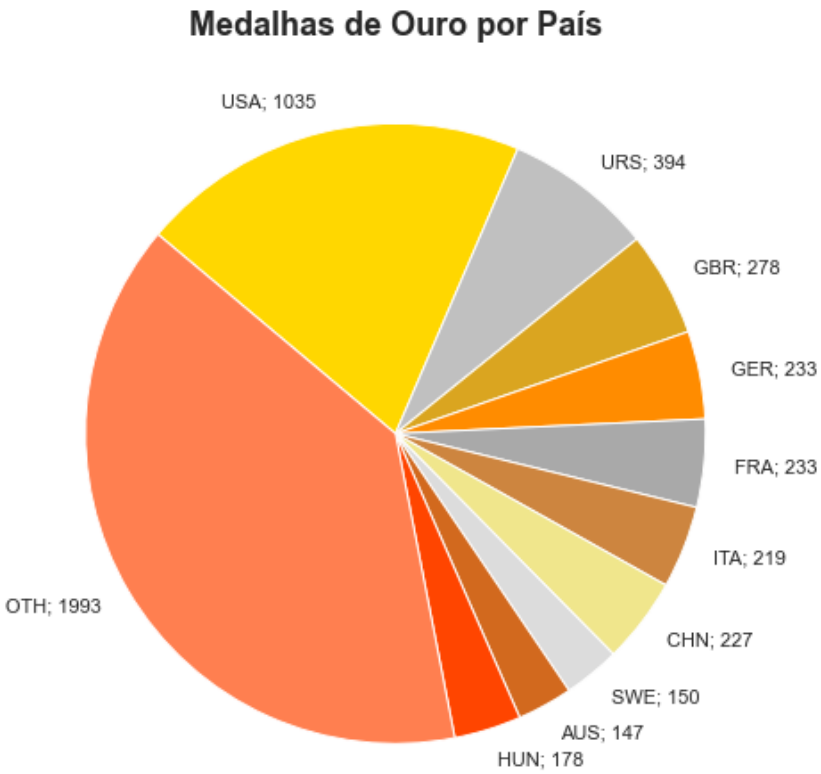
In [32]:
```
# Obtemos o total de medalhas por modalidade, removendo o nome do atleta. Então removemos as duplicidades
```

```
df_med_pais = df_med_mundo[['NOC','Ano','Esporte','Evento','Ouros','Pratas','Bronzes','Qtd','Ordem']].drop_duplicates()

# Agrupamos o resultado por país e ordenamos de forma decrescente do total de medalhas e tipo
df_tot_pais = df_med_pais.groupby('NOC').sum()
df_tot_pais.sort_values(['Qtd','Ouros','Pratas','Bronzes'], ascending=False, inplace=True)
df_tot_pais.reset_index(inplace=True)

# Separamos em dois dataframes: 10 maiores e "demais"
df_10_maiores = df_tot_pais[:10].copy()
df_outros = df_tot_pais[11:].sum()
df_outros['NOC'] = 'OTH'      # Ajustamos o NOC do grupo "demais" países

# Juntamos os dois dataframes
df_11_maiores = df_10_maiores.append(df_outros, ignore_index=True)
df_11_maiores
```

Out[32]:

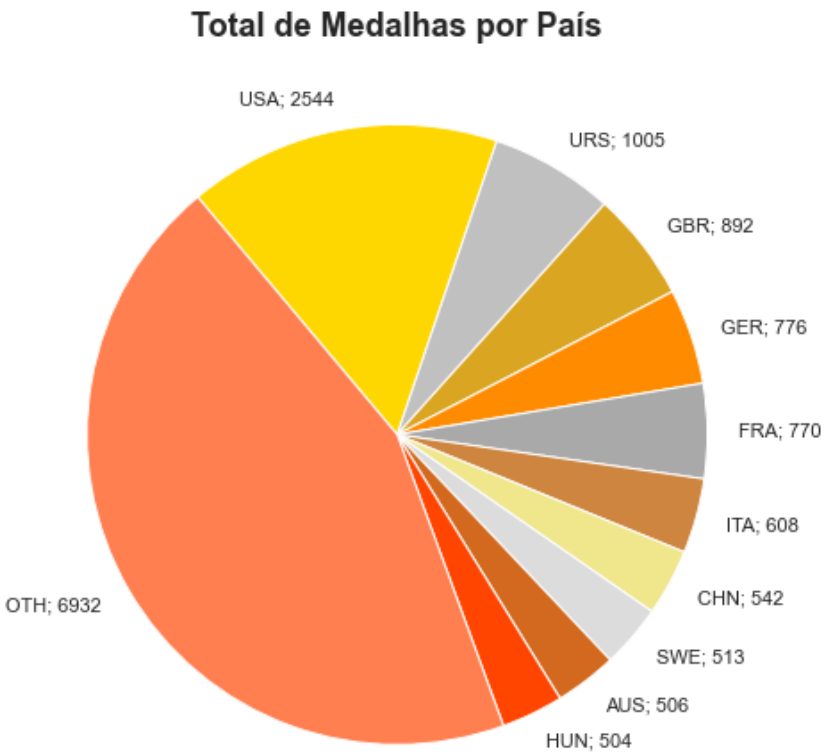
	NOC	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	USA	4989950	1035	802	707	2544	4760
1	URS	1980704	394	317	294	1005	1910
2	GBR	1744990	278	316	298	892	1804
3	GER	1527536	233	261	282	776	1601
4	FRA	1506074	233	255	282	770	1589
5	ITA	1196488	219	191	198	608	1195
6	CHN	1085756	227	162	153	542	1010
7	SWE	997768	150	175	188	513	1064
8	AUS	1003962	147	167	192	506	1057
9	HUN	991064	178	154	172	504	1002
10	OTH	13714004	1993	2269	2670	6932	14541

In [33]:
```
# A direita da sigla do país será apresentado o total de medalhas
df_pizza_11 = df_11_maiores[['NOC','Ouros']].copy()
df_pizza_11['Rotulos'] = df_pizza_11['NOC']+'; '+df_pizza_11['Ouros'].astype(str)

pre_g(altura=8)

g = plt.pie(df_pizza_11.Ouros, labels=df_pizza_11.Rotulos, counterclock=False, startangle=140, colors=cores_pizza);

config_g(g, tit='Medalhas de Ouro por País')
```



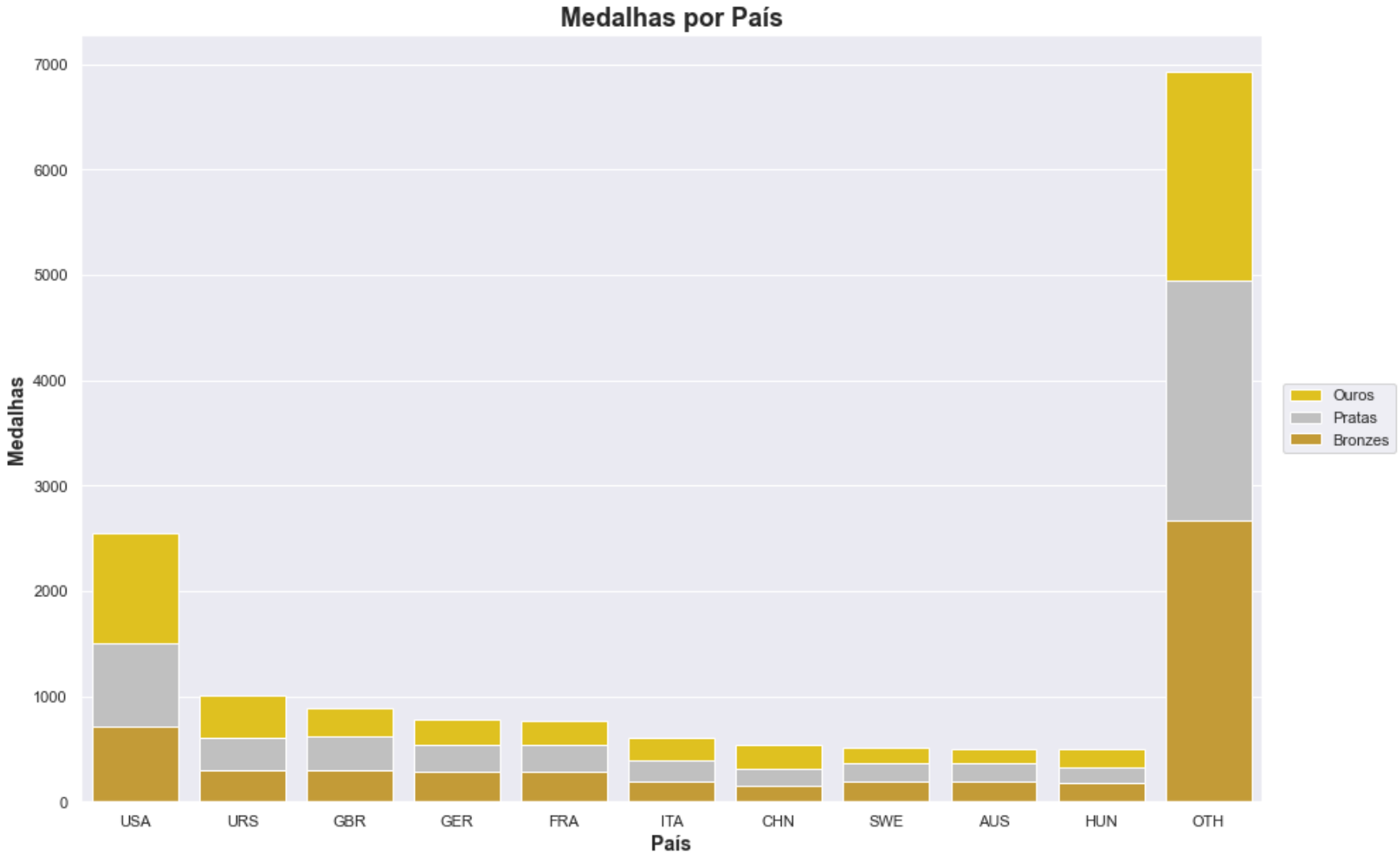
Repita o procedimento acima, mas mostrando o total de medalhas ao invés de apenas medalhas de ouro.

In [34]:
```
df_pizza_11T = df_11_maiores[['NOC','Qtd']].copy()
df_pizza_11T['Rotulos'] = df_pizza_11T['NOC']+'; '+df_pizza_11T['Qtd'].astype(str)

pre_g(altura=8)

g = plt.pie(df_pizza_11T.Qtd, labels=df_pizza_11T.Rotulos, colors=cores_pizza, counterclock=False,
            startangle=130);

config_g(g, tit='Total de Medalhas por País')
```



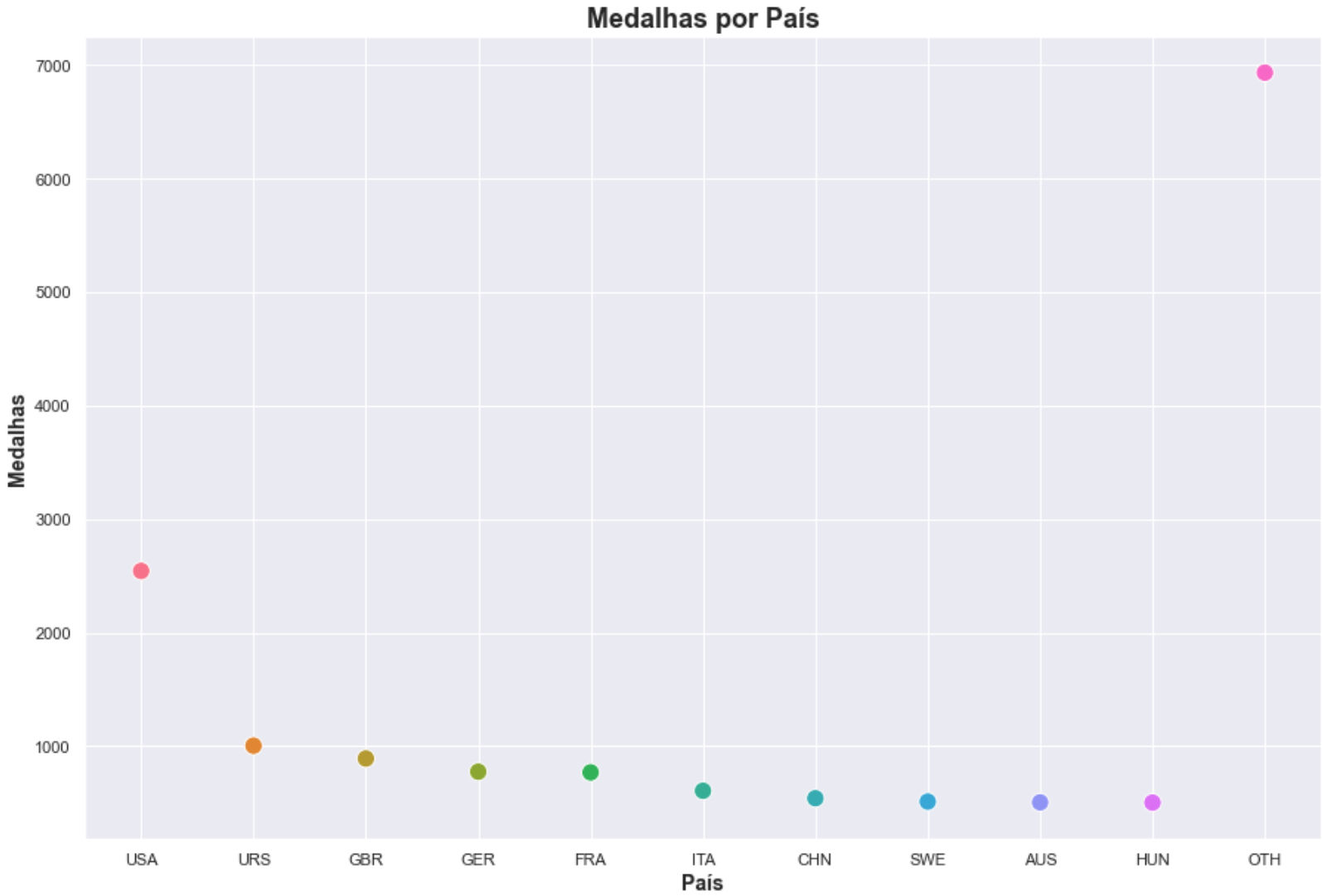
Crie um gráfico de barras empilhadas, com cada país das categorias acima no eixo X, total de medalhas no eixo Y, e barras empilhadas representando as medalhas de ouro, prata e bronze de cada país.

In [35]:
```
pre_g()

sns.barplot(data=df_11_maiores, x=df_11_maiores.NOC,
            y=df_11_maiores.Ouros+df_11_maiores.Pratas+df_11_maiores.Bronzes, color='gold', label='Ouros', ci=None)

sns.barplot(data=df_11_maiores, x=df_11_maiores.NOC,
            y=df_11_maiores.Pratas+df_11_maiores.Bronzes, color='silver', label='Pratas', ci=None)

g = sns.barplot(data=df_11_maiores, x=df_11_maiores.NOC,
            y=df_11_maiores.Bronzes, color='goldenrod', label='Bronzes', ci=None)

config_g(g, tit='Medalhas por País', tit_x='País', tit_y='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```



Crie um gráfico de distribuição (por exemplo, um histograma) mostrando a quantidade total de medalhas por país.

In [36]:
```
pre_g()

g = sns.scatterplot(data=df_11_maiores, x=df_11_maiores.NOC, y=df_11_maiores.Qtd, hue=df_11_maiores.NOC.values, s=150, legend=False)

config_g(g, tit='Medalhas por País', tit_x='País', tit_y='Medalhas')
```



Desafio bônus: Crie uma visualização da quantidade de medalhas de ouro e outra para quantidade total de medalhas por país, ambas utilizando mapas. Utilize o tipo de mapa que achar mais adequado.

In [37]:
```
# É necessário utilizar ISO-3 para os códigos de país, pois é a configuração que mais se
# aproxima dos códigos utilizados na coluna NOC do DataFrame
df_11_geo = df_10_maiores.copy()
df_11_geo.loc[df_11_geo['NOC'] == 'URS','NOC'] = 'RUS'   # Russia não é URS mas sim RUS em ISO-3
df_11_geo.loc[df_11_geo['NOC'] == 'GER','NOC'] = 'DEU'   # Alemanha não é GER mas sim DEU em ISO-3

# Vamos utilizar Plotly e suas características de gráficos geográficos

import plotly.express as px
```

```
import plotly.graph_objects as go

# Desenhamos o gráfico
fig = px.scatter_geo(df_11_geo, locations='NOC', color='NOC', size='Ouros',
                     locationmode='ISO-3', projection="equirectangular",
                     height=500, width=900)

# Customizamos as cores de fundo
fig.update_geos(resolution=50,showcoastlines=True, coastlinecolor="DarkGreen",
    showland=True, landcolor="LightGreen", showocean=True, oceancolor="LightBlue",
    showlakes=False, showrivers=False)

# Apresentamos
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()

# Agora para medalhas de prata
fig = px.scatter_geo(df_11_geo, locations='NOC', color='NOC', size='Pratas',
                     locationmode='ISO-3', projection="equirectangular",
                     height=500, width=900)

fig.update_geos(resolution=50,showcoastlines=True, coastlinecolor="DarkGreen",
    showland=True, landcolor="LightGreen", showocean=True, oceancolor="LightBlue",
    showlakes=False, showrivers=False)

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

3. Brasil vs Mundo

Faça um gráfico de barras comparando os maiores medalhistas brasileiros com os maiores medalhistas do mundo em suas respectivas categorias.

Represente o esporte no eixo X, a quantidade de medalhas no eixo Y, coloque barras lado-a-lado representando os diferentes atletas de uma mesma modalidade e empilhe as medalhas de ouro, prata e bronze de cada atleta.

In [28]:
```
# Indexa DataFrame de atletas por nome, reduzindo para as colunas desejadas apenas
df_atletas_nome = df_atletas[['Nome','Esporte','NOC']].set_index('Nome')

# Adiciona o esporte ao DataFrame de maiores medalhistas do Brasil para poder buscar os atletas
# dos outros paises por essa chave
df_comp_br = df_mai_atleta_br.join(df_atletas_nome, on='Nome', how='left').drop_duplicates()
df_comp_br.reset_index(inplace=True, drop=True)

# Busca os atletas de todo mundo para o esporte dos maiores medalhistas brasileiros
df_mai_mundo = df_med_mundo[df_med_mundo.Esporte.isin(df_comp_br.Esporte.unique())].groupby('Nome').sum()
```

```
# ...filtrando para aqueles que são os maiores medalhistas na modalidade
df_mai_mundo = df_mai_mundo[(df_mai_mundo.Qtd == df_mai_mundo.Qtd.max())]
df_mai_mundo = df_mai_mundo.join(df_atletas_nome, on='Nome', how='left').drop_duplicates()
df_mai_mundo.reset_index(inplace=True)

# Agora juntamos os maiores medalhistas do mundo da(s) modalidade(s) apuradas, e eliminamos as duplicidades,
# que podem ocorrer
# se os brasileiros tiverem o mesmo total de medalhas dos demais atletas do mundo
df_mai_mundo.append(df_comp_br).drop_duplicates()

# Para uma melhor visualização, acrescenta o país ao nome do atleta que vamos apresentar no eixo X do gráfico
df_mai_mundo['Nome'] = df_mai_mundo['Nome'] + ' ('+ df_mai_mundo['NOC']+ ')'
```

In [29]:

```
pre_g()

sns.barplot(data=df_mai_mundo, x=df_mai_mundo.Nome,
            y=df_mai_mundo.Ouros+df_mai_mundo.Pratas+df_mai_mundo.Bronzes, color='gold', label='Ouros')

sns.barplot(data=df_mai_mundo, x=df_mai_mundo.Nome,
            y=df_mai_mundo.Pratas+df_mai_mundo.Bronzes, color='silver', label='Pratas')

g = sns.barplot(data=df_mai_mundo, x=df_mai_mundo.Nome,
            y=df_mai_mundo.Bronzes, color='goldenrod', label='Bronzes')

config_g(g, tit='Maiores Medalhistas BR x Mundo', tit_x='Atleta', tit_y='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```

Repita o procedimento acima, mas ao invés de atletas, considere o(s) esporte(s) onde o Brasil mais possui medalha comparando-os com o país com maior quantidade de medalhas naquele esporte.

In [30]:
```
# DataFrame contendo apenas o(s) esporte(s) brasieiros com mais medalhas no total
df_mai_esporte_br = df_esporte_br[df_esporte_br.Qtd == df_esporte_br.Qtd.max()]

# Seleciona o total de medalhas de todos os países na(s) modalidade(s) identificadas no passo anterior
df_mais_pais_esp = df_med_pais[df_med_pais.Esporte.isin(df_mai_esporte_br.Esporte.unique())] \
    .groupby(['NOC','Esporte']).sum()

# Filtra o(s) país(es) com o máximo de medalhas na modalidade
df_mais_pais_esp = df_mais_pais_esp[df_mais_pais_esp.Qtd == df_mais_pais_esp.Qtd.max()]
```

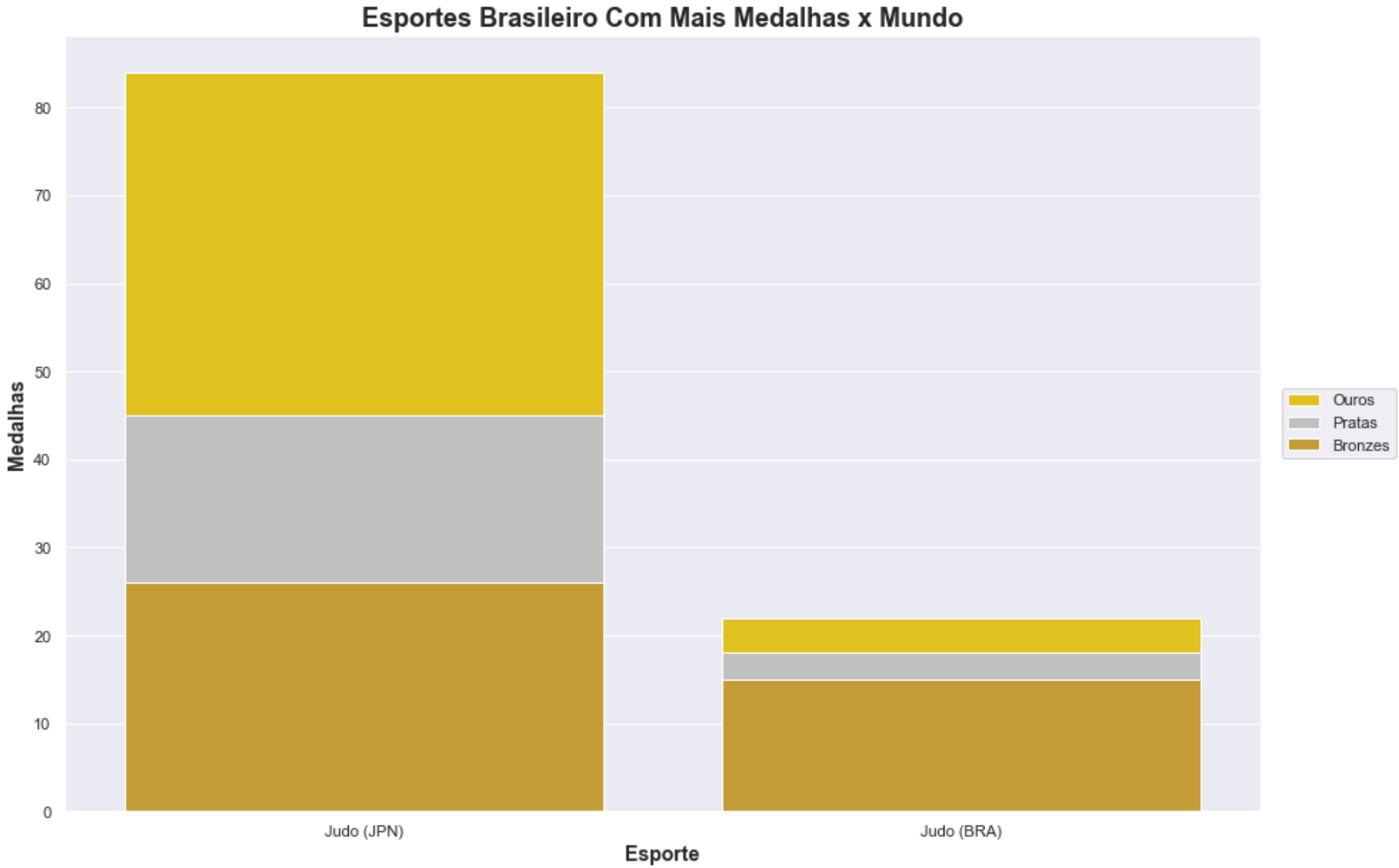
```
# Prepara para concatenar o Brasil
df_mais_pais_esp.reset_index(inplace=True)

# Concatena e adiciona o código NOC para o Brasil, pois o DataFrame não contem esse dado
df_mais_pais_esp = df_mais_pais_esp.append(df_mai_esporte_br, ignore_index=True).drop_duplicates()
df_mais_pais_esp.fillna('BRA', inplace=True)
df_mais_pais_esp.Esporte = df_mais_pais_esp.Esporte + ' (' + df_mais_pais_esp.NOC + ')'
df_mais_pais_esp
```

Out[30]:

	NOC	Esporte	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	JPN	Judo (JPN)	167756	39	19	26	84	155
1	BRA	Judo (BRA)	44024	4	3	15	22	55

In [31]:

```
pre_g()

sns.barplot(data=df_mais_pais_esp, x=df_mais_pais_esp.Esporte,
            y=df_mais_pais_esp.Ouros+df_mais_pais_esp.Pratas+df_mais_pais_esp.Bronzes, color='gold', label='Ouros')

sns.barplot(data=df_mais_pais_esp, x=df_mais_pais_esp.Esporte,
            y=df_mais_pais_esp.Pratas+df_mais_pais_esp.Bronzes, color='silver', label='Pratas')

g = sns.barplot(data=df_mais_pais_esp, x=df_mais_pais_esp.Esporte,
            y=df_mais_pais_esp.Bronzes, color='goldenrod', label='Bronzes')

config_g(g, tit='Esportes Brasileiro Com Mais Medalhas x Mundo', tit_x='Esporte', tit_y='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```



Para finalizar, repita os gráficos que você gerou com os 10 países com mais medalhas, mas remova o Brasil da categoria "Outros" e mostre-o também no gráfico.

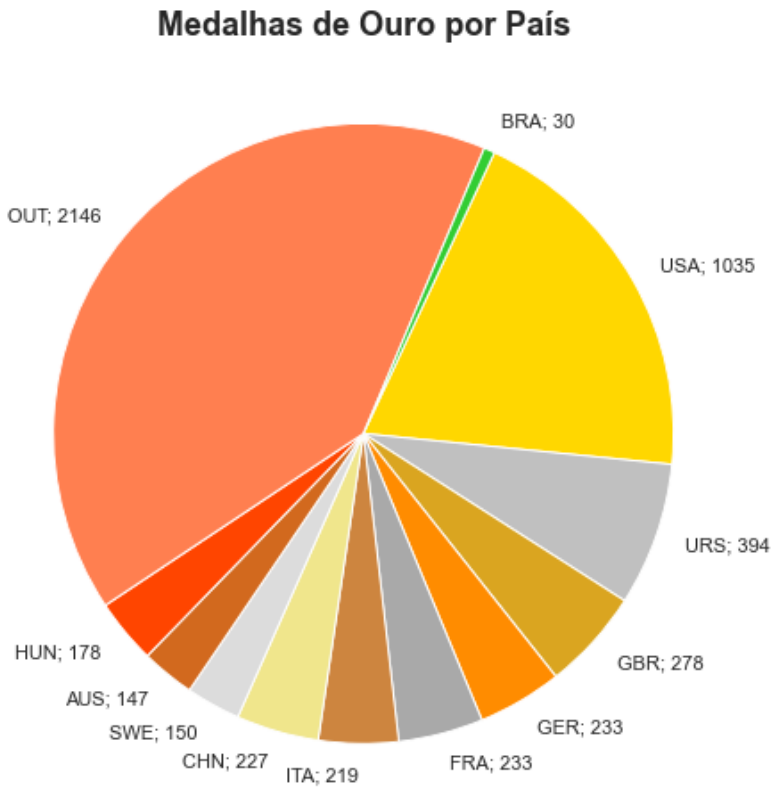
In [54]:
```
# Adicionamos o Brasil ao DataFrame com os 10 países com mais medalhas
df_10_BRA = df_10_maiores.copy()

# Removemos o Brasil do grupo "outros"
df_outros = df_tot_pais[~df_tot_pais.NOC.isin(df_10_BRA.NOC.unique())].sum()
df_outros['NOC'] = 'OUT'      # Ajusta o NOC do grupo "outros"

# Une os dois DataFrames
df_10_BRA = df_10_BRA.append(df_outros, ignore_index=True)
```

```
# Um pequeno truque: Brasil adicionado por último para ter uma cor específica nos gráficos de pizza
df_10_BRA = df_10_BRA.append(df_tot_pais[df_tot_pais.NOC == 'BRA'], ignore_index=True)
df_10_BRA
```

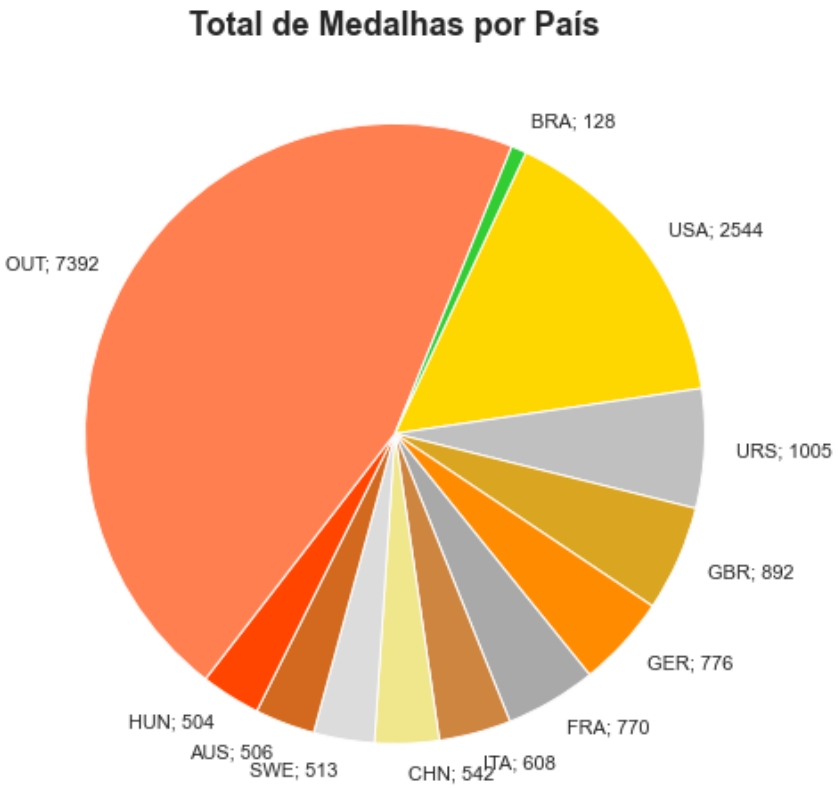
Out[54]:

	NOC	Ano	Ouros	Pratas	Bronzes	Qtd	Ordem
0	USA	4989950	1035	802	707	2544	4760
1	URS	1980704	394	317	294	1005	1910
2	GBR	1744990	278	316	298	892	1804
3	GER	1527536	233	261	282	776	1601
4	FRA	1506074	233	255	282	770	1589
5	ITA	1196488	219	191	198	608	1195
6	CHN	1085756	227	162	153	542	1010
7	SWE	997768	150	175	188	513	1064
8	AUS	1003962	147	167	192	506	1057
9	HUN	991064	178	154	172	504	1002
10	OUT	14635852	2146	2412	2834	7392	15472
11	BRA	255516	30	36	62	128	288

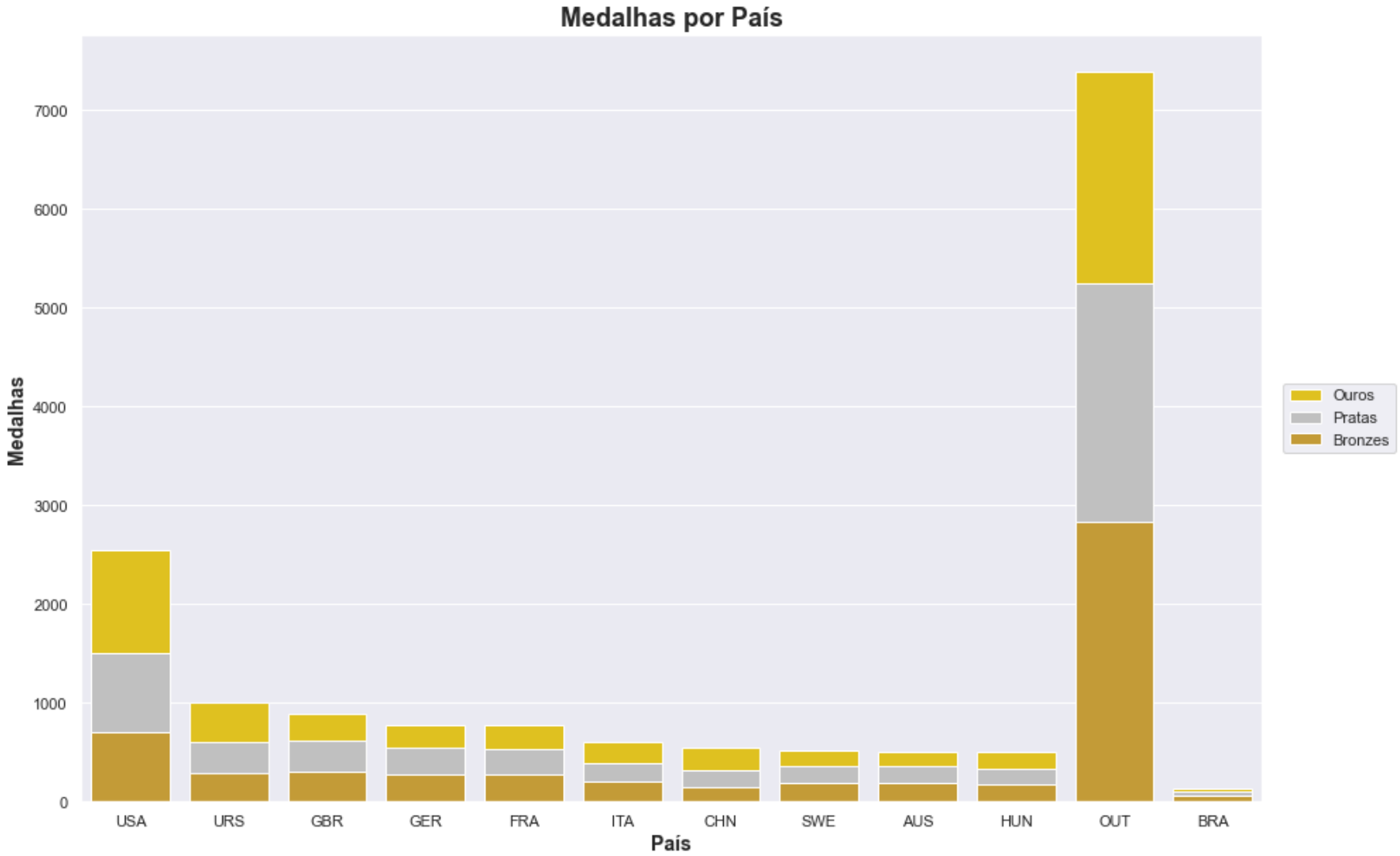
In [69]:
```
# a) Gráfico de pizza do total de ouros
df_pizza_11 = df_10_BRA[['NOC','Ouros']].copy()
df_pizza_11['Rotulos'] = df_pizza_11['NOC']+'; '+df_pizza_11['Ouros'].astype(str)

pre_g(altura=8)

g = plt.pie(df_pizza_11.Ouros, labels=df_pizza_11.Rotulos, colors=cores_pizza, counterclock=False,
            startangle=65);

config_g(g, tit='Medalhas de Ouro por País')
```



In [68]:
```
# b) Gráfico de pizza do total de medalhas
df_pizza_11T = df_10_BRA[['NOC','Qtd']].copy()
df_pizza_11T['Rotulos'] = df_pizza_11T['NOC']+'; '+df_pizza_11T['Qtd'].astype(str)

pre_g(altura=8)

g = plt.pie(df_pizza_11T.Qtd, labels=df_pizza_11T.Rotulos, colors=cores_pizza, counterclock=False,
            startangle=65);

config_g(g, tit='Total de Medalhas por País')
```



In [70]:
```
# c) Gráfico de barras empilhadas por tipo de medalha
pre_g()
sns.barplot(data=df_10_BRA, x=df_10_BRA.NOC,
            y=df_10_BRA.Ouros+df_10_BRA.Pratas+df_10_BRA.Bronzes, color='gold', label='Ouros', ci=None)

sns.barplot(data=df_10_BRA, x=df_10_BRA.NOC,
            y=df_10_BRA.Pratas+df_10_BRA.Bronzes, color='silver', label='Pratas', ci=None)

g = sns.barplot(data=df_10_BRA, x=df_10_BRA.NOC,
            y=df_10_BRA.Bronzes, color='goldenrod', label='Bronzes', ci=None)

config_g(g, tit='Medalhas por País', tit_x='País', tit_y='Medalhas',
         legenda='center right', ancora_leg=(1.12, 0.5))
```


Medalhas por País
Medalhas
0
1000
2000
3000
4000
5000
6000
7000
USA
URS
GBR
GER
FRA
ITA
CHN
SWE
AUS
HUN
OUT
BRA
País
Ouros
Pratas
Bronzes