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

In [2]:
```
confirmed = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
confirmed.head()
```

Out[2]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0

5 rows × 486 columns

In [3]:
```
confirmed.shape
```

Out[3]:
```
(275, 486)
```

In [4]:
```
last_date = "5/13/21"
confirmed[last_date]
```

Out[4]:
```
0       63045
1      131890
2      124889
3       13470
4       29695
        ...
270      3740
271    303270
272      6507
273     92262
274     38491
Name: 5/13/21, Length: 275, dtype: int64
```

In [5]:
```
# valor falso pois o erro é muito grande
# não vou considerar essa análise pois tem muito campo vazio (mui
ta disparidade no padrão)

confirmed['Country/Region'].value_counts()
```

Out[5]:
```
China                  34
Canada                 16
United Kingdom         12
France                 12
Australia               8
                       ..
Mauritania              1
Marshall Islands        1
Dominican Republic      1
Saint Kitts and Nevis   1
Eswatini                1
Name: Country/Region, Length: 192, dtype: int64
```

In [6]:
```
confirmed_by_country = confirmed.groupby('Country/Region').sum()
confirmed_by_country.head(10)
```

Out[6]:

Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/2(
Afghanistan	33.93911	67.709953	0	0	0	0	0	(
Albania	41.15330	20.168300	0	0	0	0	0	(
Algeria	28.03390	1.659600	0	0	0	0	0	(
Andorra	42.50630	1.521800	0	0	0	0	0	(
Angola	-11.20270	17.873900	0	0	0	0	0	(
Antigua and Barbuda	17.06080	-61.796400	0	0	0	0	0	(
Argentina	-38.41610	-63.616700	0	0	0	0	0	(
Armenia	40.06910	45.038200	0	0	0	0	0	(
Australia	-256.85020	1130.843900	0	0	0	0	4	5
Austria	47.51620	14.550100	0	0	0	0	0	(

10 rows × 484 columns

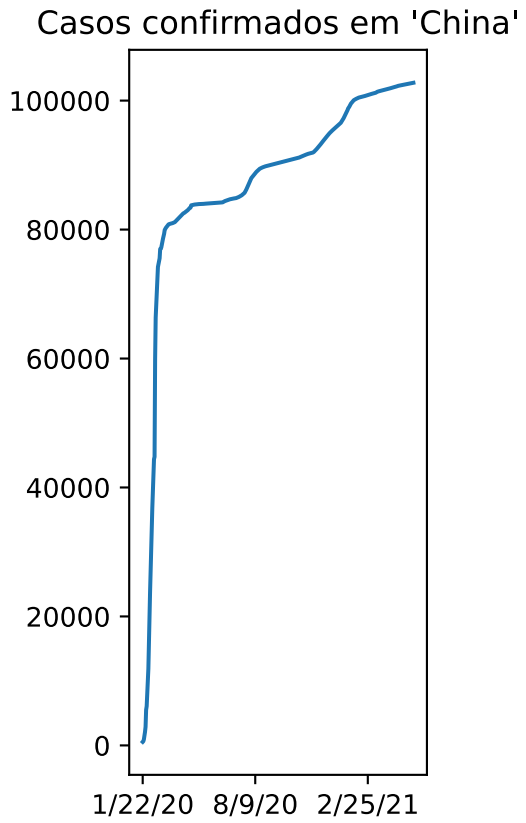
In [7]:
```
country = 'China'
```

In [8]:
```
confirmed_by_country.loc[country][2:]
```

Out[8]:
```
1/22/20       548.0
1/23/20       643.0
1/24/20       920.0
1/25/20      1406.0
1/26/20      2075.0
             ...
5/13/21    102681.0
5/14/21    102696.0
5/15/21    102717.0
5/16/21    102746.0
5/17/21    102769.0
Name: China, Length: 482, dtype: float64
```

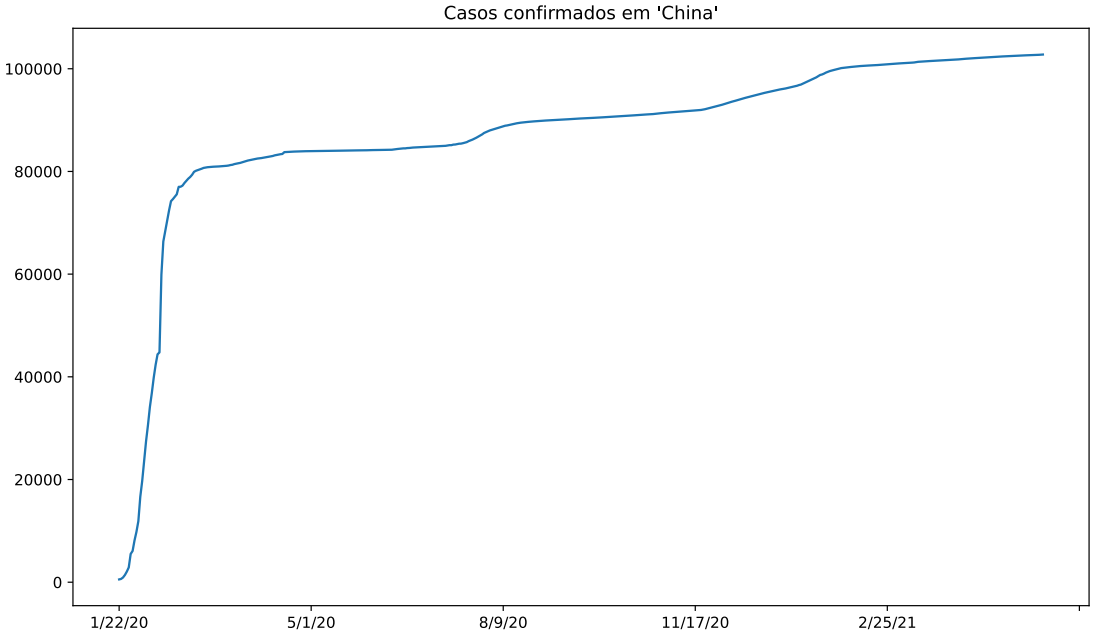
In [9]:
```
import matplotlib.pyplot as plt
```

In [10]:
```
# grafico extremamente apertado, estamos sendo longe do adequado
em passar uma mensagem errada

plt.figure(figsize=(2,5))
confirmed_by_country.loc[country][2:].plot()
plt.title("Casos confirmados em 'China'")
plt.show()
```



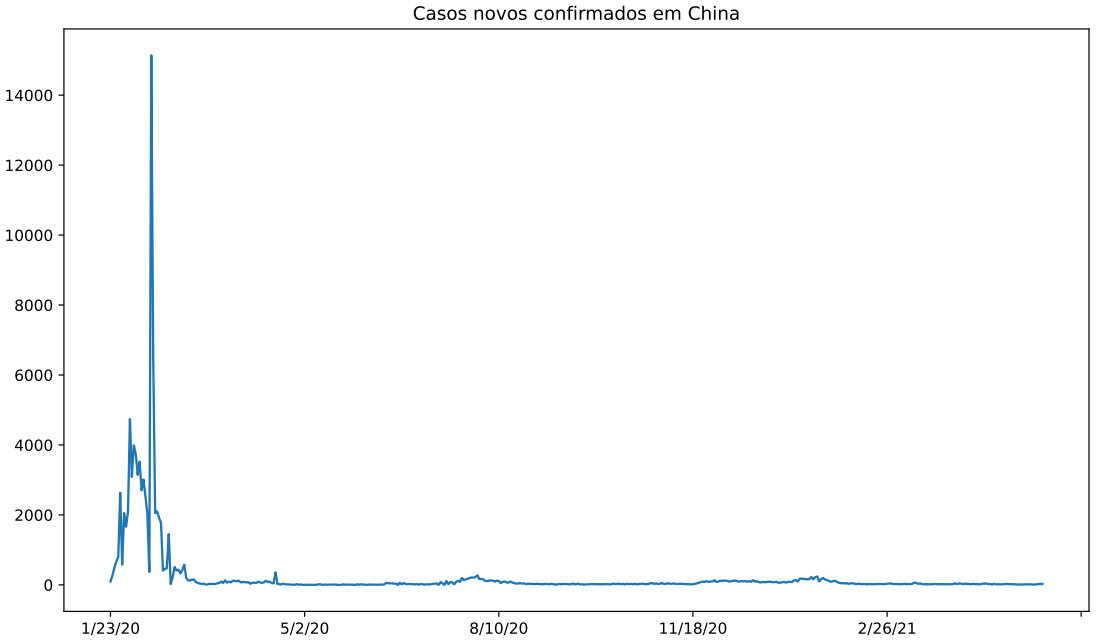
In [11]:
```
# comparando o momento inicial com 1.386 bi (população da China 2017)

plt.figure(figsize=(12,7))
confirmed_by_country.loc[country][2:].plot()
plt.title("Gráfico errado. Passa uma mensagem errada com comparações injustas. Casos confirmados em 'China'")
plt.ylim(0, 1386000000)
plt.show()
```

In [12]:
```
# tentando comparar o crescimento do momento inicial ao momento atual

plt.figure(figsize=(12,7))
confirmed_by_country.loc[country][2:].plot()
plt.title("Casos confirmados em 'China'")
plt.show()
```

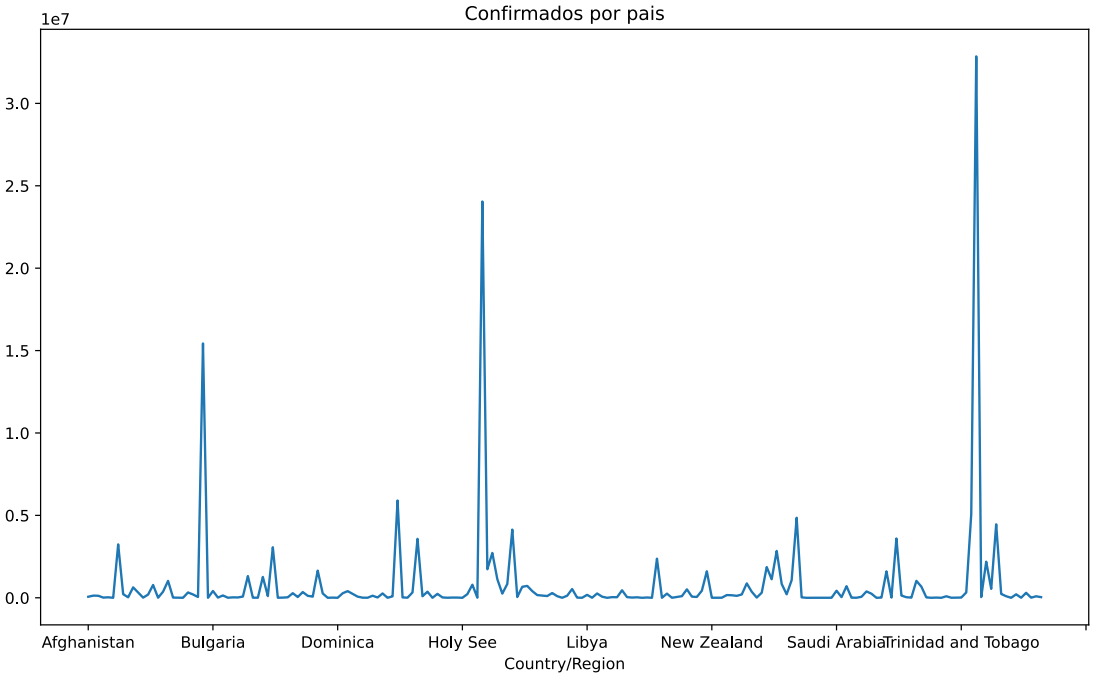


In [13]:
```
new_cases_country = confirmed_by_country.loc[country][2:].diff().
dropna()

plt.figure(figsize=(12,7))
plt.title(f'Casos novos confirmados em {country}')
new_cases_country.plot()
```

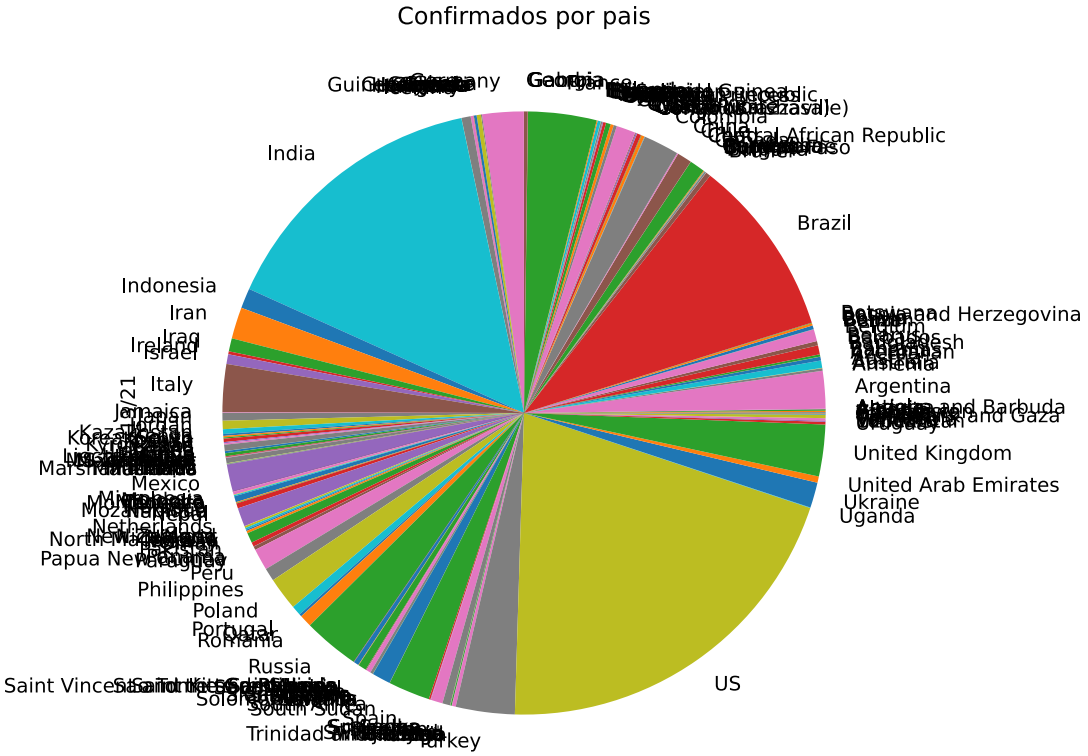
Out[13]: <AxesSubplot:title={'center':'Casos novos confirmados em China'}>



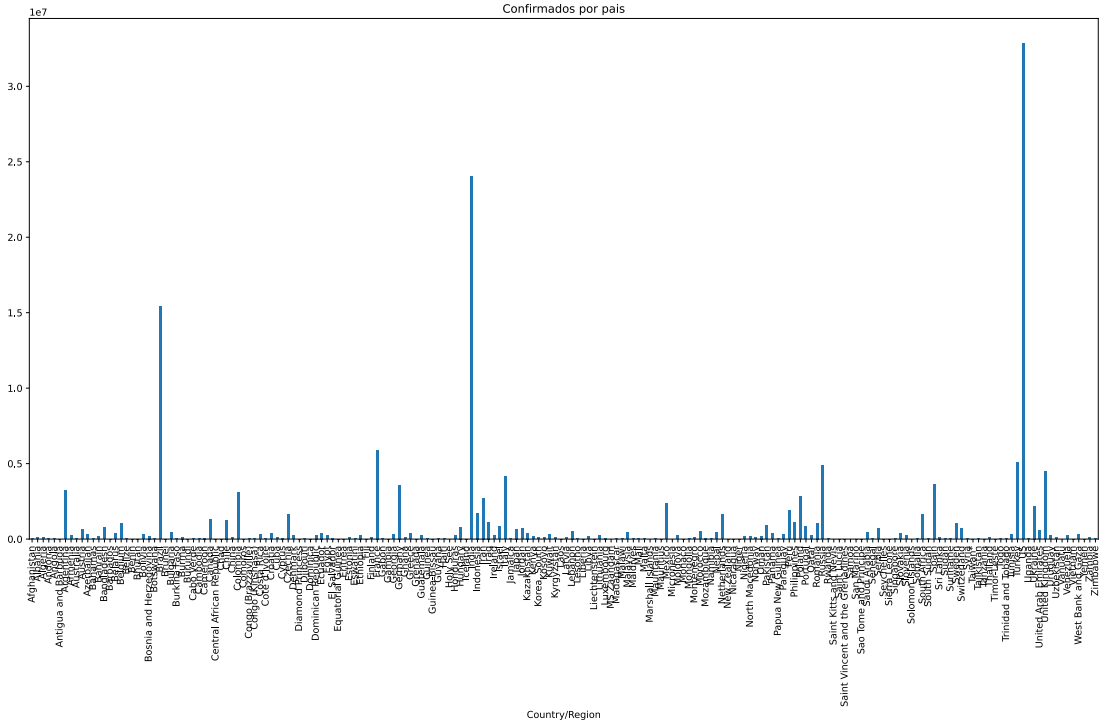
In [14]:
```
plt.figure(figsize=(12,7))
confirmed_by_country[last_date].plot()
plt.title('Confirmados por pais')
plt.show()
```



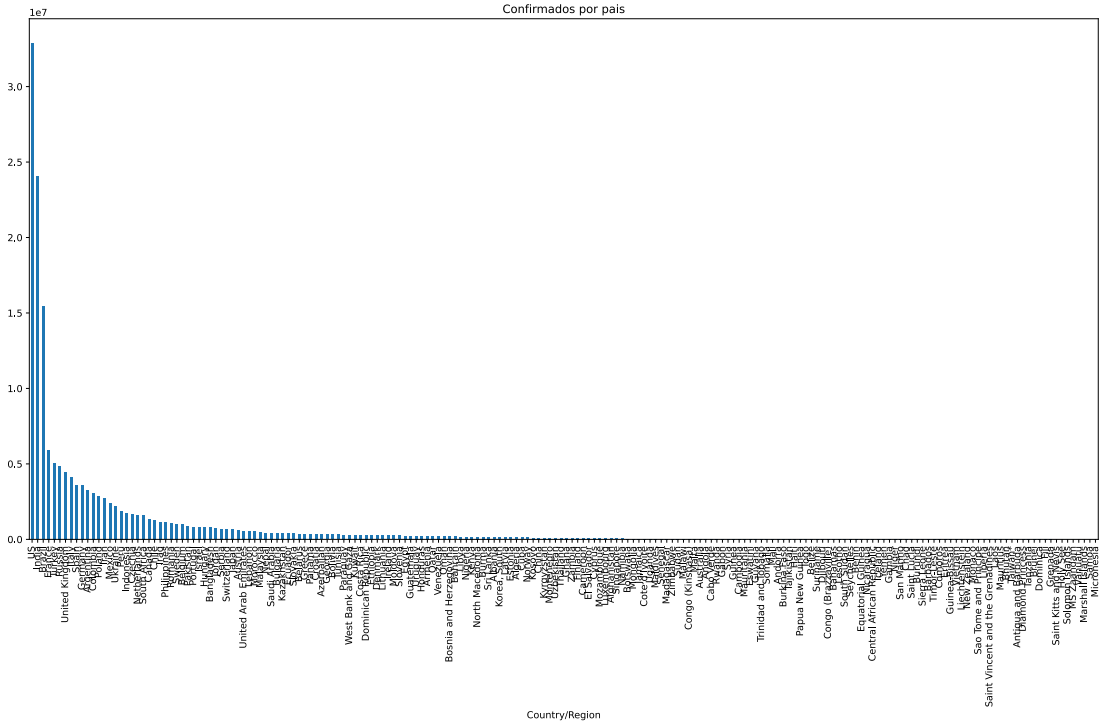
In [15]:
```
# 99,99% das vezes não devemos usar gráfico de pizza

# 1- Muitas categorias, inviável
# 2- Poucas categorias, a comparação é de área!

plt.figure(figsize=(12,7))
confirmed_by_country[last_date].plot(kind='pie')
plt.title('Confirmados por pais')
plt.show()
```



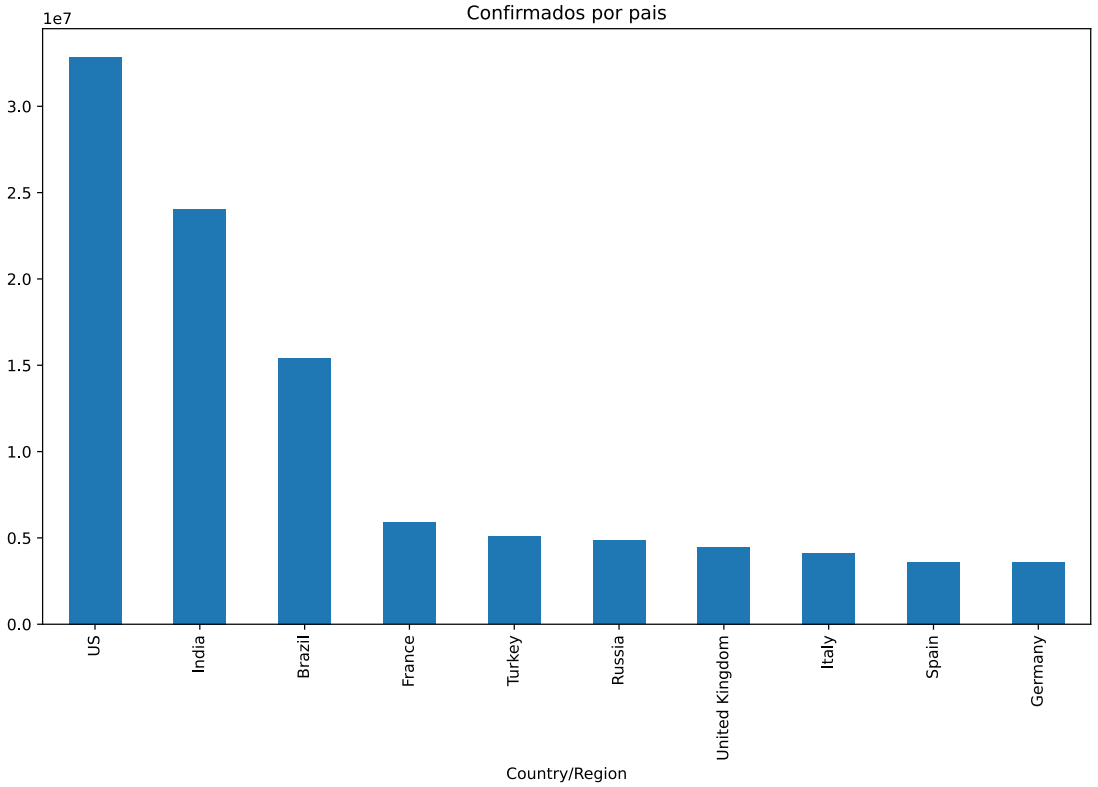
In [16]:
```
plt.figure(figsize=(20,10))
confirmed_by_country[last_date].plot(kind='bar')
plt.title('Confirmados por pais')
plt.show()
```



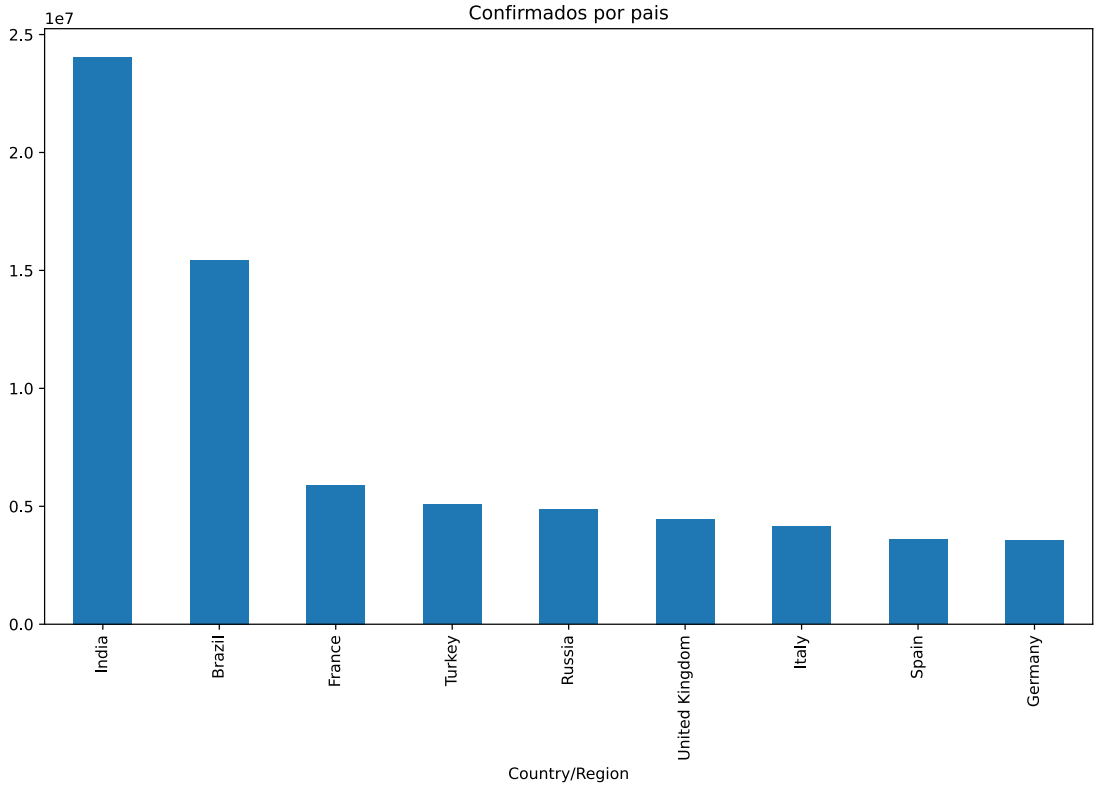
In [17]:
```
plt.figure(figsize=(20,10))
confirmed_by_country[last_date].sort_values(ascending=False).plot
(kind='bar')
plt.title('Confirmados por pais')
plt.show()
```



In [18]:
```
plt.figure(figsize=(12,7))
confirmed_by_country[last_date].sort_values(ascending=False)[0:10].plot(kind='bar')
plt.title('Confirmados por pais')
plt.show()
```



In [19]:
```
plt.figure(figsize=(12,7))
confirmed_by_country[last_date].sort_values(ascending=False)[1:1
0].plot(kind='bar')
plt.title('Confirmados por pais')
plt.show()
```



In [20]:
```
deaths = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv")

recovered = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv")

display(deaths.head())
display(recovered.head())
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0

5 rows × 486 columns

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0

5 rows × 486 columns

In [21]:
```
def latest_by_country(data):
    return data.groupby('Country/Region').sum().iloc[:,-1]

def latest_by_country_at(data, date):
    return data.groupby('Country/Region').sum()[date]
```

In [22]:
```
informations = [latest_by_country(confirmed), latest_by_country(deaths), latest_by_country(recovered)]
combined = pd.concat(informations, axis=1)
combined.columns = ['Confirmados', 'Mortos', 'Recuperados']
combined
```

Out[22]:

Country/Region	Confirmados	Mortos	Recuperados
Afghanistan	63598	2745	55010
Albania	132032	2435	125419
Algeria	125485	3381	87476
Andorra	13555	127	13211
Angola	30787	677	25995
...	...	...	...
Vietnam	4359	37	2668
West Bank and Gaza	304074	3437	293808
Yemen	6568	1294	3042
Zambia	92460	1261	90862
Zimbabwe	38572	1582	36349

192 rows × 3 columns

In [23]:
```
informations = [latest_by_country_at(confirmed, '2/20/20'), latest_by_country_at(deaths, '2/20/20'), latest_by_country_at(recovered, '2/20/20')]
combined_2_20_20 = pd.concat(informations, axis=1)
combined_2_20_20.columns = ['Confirmados', 'Mortos', 'Recuperados']
combined_2_20_20
```

Out[23]:

Country/Region	Confirmados	Mortos	Recuperados
Afghanistan	0	0	0
Albania	0	0	0
Algeria	0	0	0
Andorra	0	0	0
Angola	0	0	0
...	...	...	...
Vietnam	16	0	7
West Bank and Gaza	0	0	0
Yemen	0	0	0
Zambia	0	0	0
Zimbabwe	0	0	0

192 rows × 3 columns

In [24]:
```
# taxa de letalidade expressa em %
# número de casos letais dentro dos casos que foram letais e não letais

sum_up = combined.sum()
taxa_letalidade_1 = sum_up['Mortos'] / sum_up['Confirmados']
taxa_letalidade_2 = sum_up['Mortos'] / (sum_up['Recuperados'] + sum_up['Mortos'])
print(f"No conjunto de dados que estamos utilizando a taxa de letalidade 1 é {(taxa_letalidade_1 * 100):.2f}%")
print(f"No conjunto de dados que estamos utilizando a taxa de letalidade 2 é {(taxa_letalidade_2 * 100):.2f}%")
```

```
No conjunto de dados que estamos utilizando a taxa de letalidade 1 é 2.07%
No conjunto de dados que estamos utilizando a taxa de letalidade 2 é 3.28%
```

In [25]:
```
sum_up = combined_2_20_20.sum()
taxa_letalidade_1 = sum_up['Mortos'] / sum_up['Confirmados']
taxa_letalidade_2 = sum_up['Mortos'] / (sum_up['Recuperados'] + s
um_up['Mortos'])
print(f"No conjunto de dados que estamos utilizando a taxa de let
alidade 1 é {(taxa_letalidade_1 * 100):.2f}%")
print(f"No conjunto de dados que estamos utilizando a taxa de let
alidade 2 é {(taxa_letalidade_2 * 100):.2f}%")
```

```
No conjunto de dados que estamos utilizando a taxa de letalidade
1 é 2.95%
No conjunto de dados que estamos utilizando a taxa de letalidade
2 é 11.01%
```

In [26]:
```
informations = [latest_by_country_at(confirmed, '2/8/20'), latest
_by_country_at(deaths, '2/20/20'), latest_by_country_at(recovere
d, '2/20/20')]
combined_12 = pd.concat(informations, axis=1)
combined_12.columns = ['Confirmados', 'Mortos', 'Recuperados']

sum_up = combined_12.sum()
taxa_letalidade_1 = sum_up['Mortos'] / sum_up['Confirmados']
print(f"No conjunto de dados que estamos utilizando a taxa de let
alidade 1 é {(taxa_letalidade_1 * 100):.2f}%")
```

```
No conjunto de dados que estamos utilizando a taxa de letalidade
1 é 6.05%
```

In [27]:
```
taxa_letalidade_1 = (combined['Mortos'] / combined['Confirmados
']) * 100
taxa_letalidade_2 = (combined['Mortos'] / (combined['Recuperados
'] + combined['Mortos'])) * 100
combined['taxa_letalidade_1'] = taxa_letalidade_1
combined['taxa_letalidade_2'] = taxa_letalidade_2
combined.head(10)
```

Out[27]:

Country/Region	Confirmados	Mortos	Recuperados	taxa_letalidade_1	taxa_letalidade_2
Afghanistan	63598	2745	55010	4.316173	4.752835
Albania	132032	2435	125419	1.844250	1.904516
Algeria	125485	3381	87476	2.694346	3.721232
Andorra	13555	127	13211	0.936924	0.952167
Angola	30787	677	25995	2.198980	2.538242
Antigua and Barbuda	1251	42	1182	3.357314	3.431373
Argentina	3335965	71027	2973267	2.129129	2.333119
Armenia	220927	4333	208899	1.961281	2.032059
Australia	29983	910	23499	3.035053	3.728133
Austria	637573	10480	617307	1.643733	1.669356

In [28]:
```
combined.sort_values('taxa_letalidade_1', ascending=False).head(10)
```

Out[28]:

Country/Region	Confirmados	Mortos	Recuperados	taxa_letalidade_1	taxa_letalidade_2
Vanuatu	4	1	3	25.000000	25.000000
MS Zaandam	9	2	7	22.222222	22.222222
Yemen	6568	1294	3042	19.701583	29.843173
Mexico	2382745	220493	1903494	9.253739	10.381090
Syria	23788	1705	21073	7.167479	7.485293
Sudan	34889	2446	27949	7.010806	8.047376
Egypt	246909	14388	182024	5.827248	7.325418
Somalia	14486	753	6325	5.198122	10.638598
Ecuador	410870	19786	354499	4.815635	5.286346
China	102769	4846	97543	4.715430	4.732930

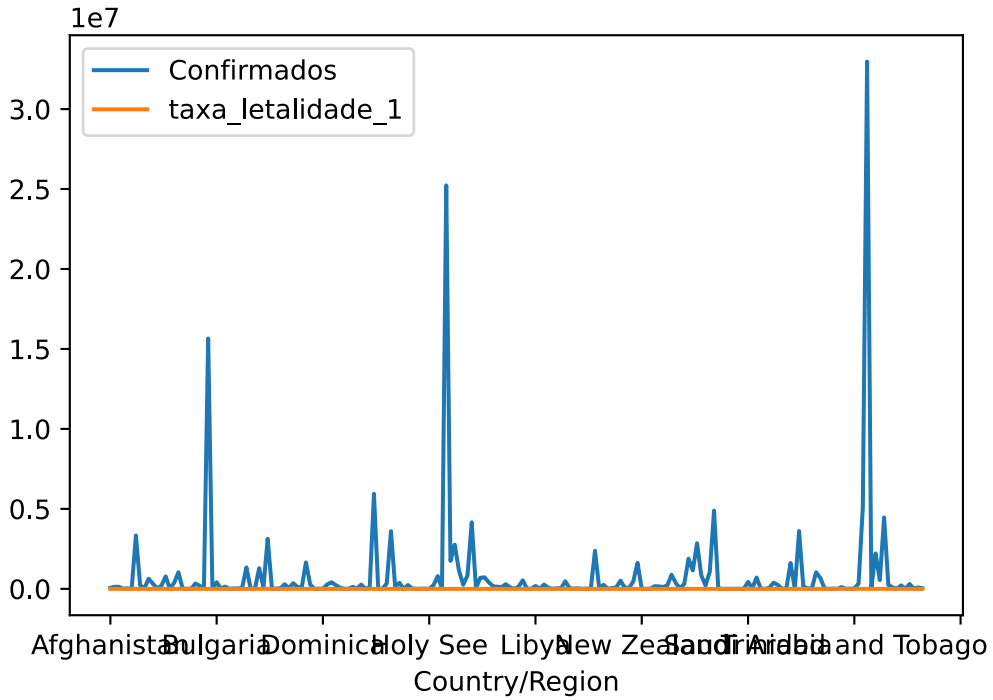
In [29]:
```
combined.sort_values('taxa_letalidade_2', ascending=False).head(10)
```

Out[29]:

Country/Region	Confirmados	Mortos	Recuperados	taxa_letalidade_1	taxa_letalidade_2
Sweden	1037126	14275	0	1.376400	100.000000
Belgium	1032895	24723	0	2.393564	100.000000
Serbia	707033	6696	0	0.947056	100.000000
US	32969480	586362	0	1.778499	100.000000
United Kingdom	4468582	127946	15292	2.863235	89.324062
Netherlands	1627997	17725	26534	1.088761	40.048352
Spain	3615860	79432	150376	2.196766	34.564506
Yemen	6568	1294	3042	19.701583	29.843173
Vanuatu	4	1	3	25.000000	25.000000
MS Zaandam	9	2	7	22.222222	22.222222

In [30]:
```
# Visualização horenda

combined[['Confirmados', 'taxa_letalidade_1']].plot()
```

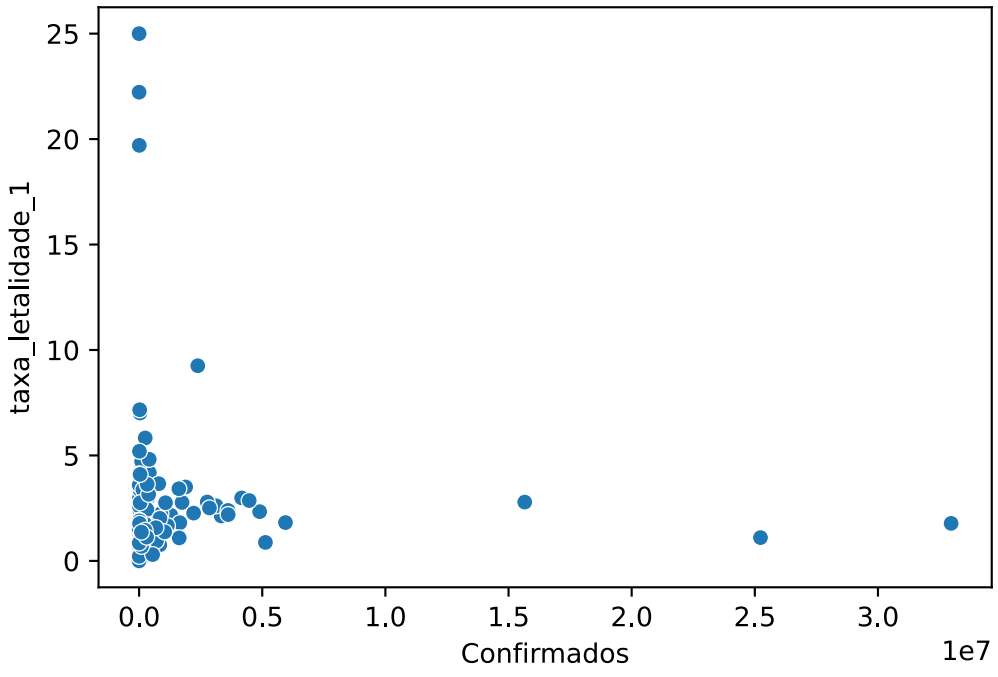
Out[30]: <AxesSubplot:xlabel='Country/Region'>



In [31]:
```
import seaborn as sns
```

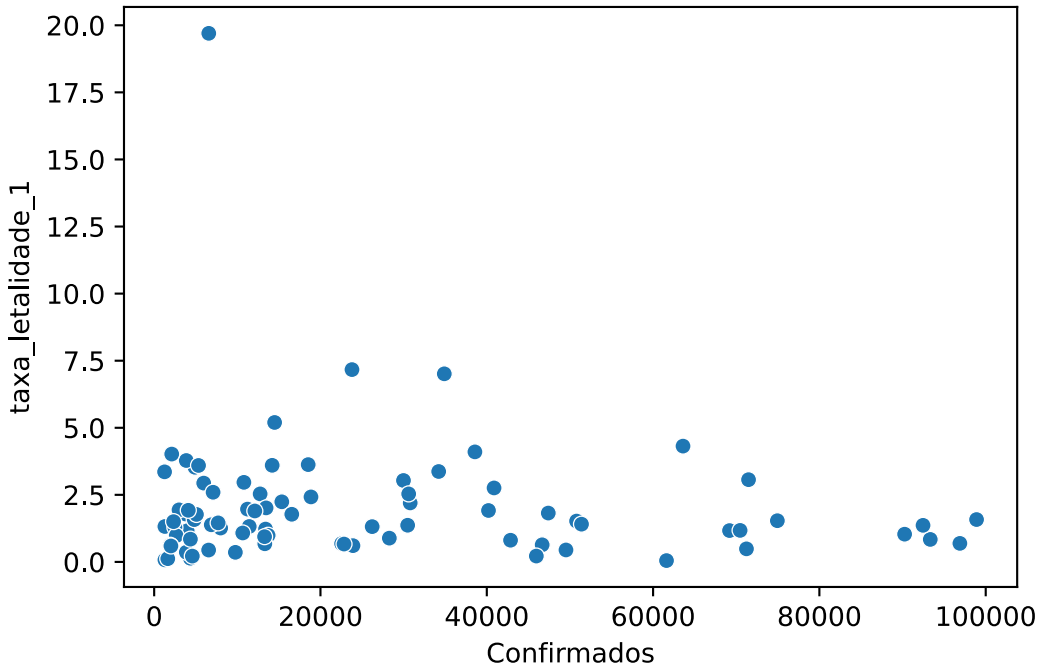
In [32]:
```
sns.scatterplot(data = combined, x='Confirmados', y='taxa_letalidade_1')
```

Out[32]: <AxesSubplot:xlabel='Confirmados', ylabel='taxa_letalidade_1'>



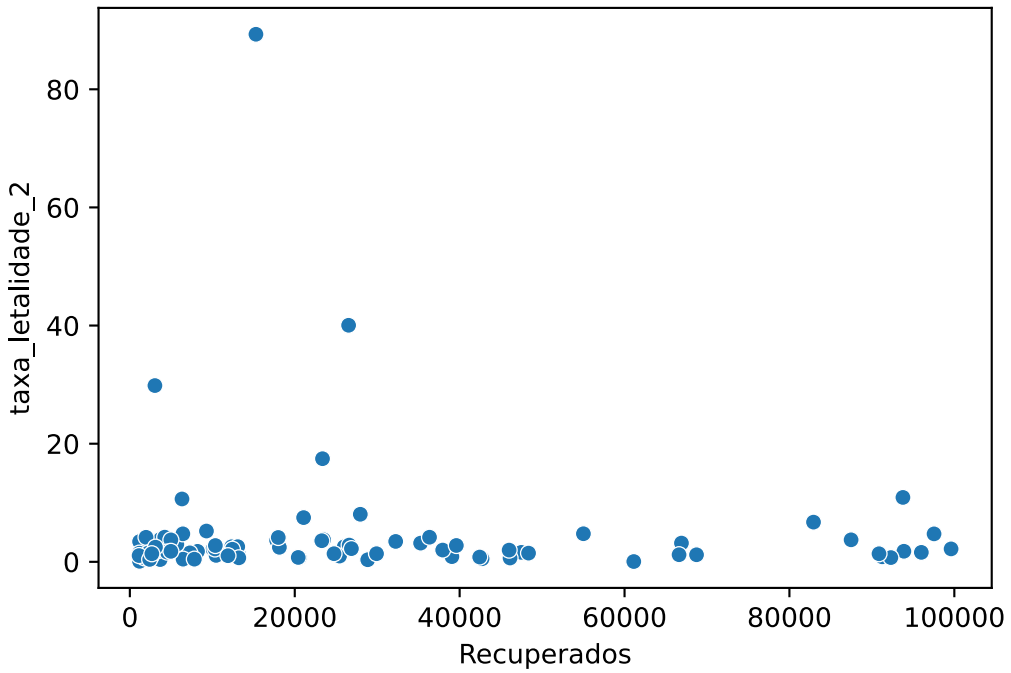
In [33]:
```
sns.scatterplot(data = combined.query('Confirmados > 1000 and Con
firmados < 100000'), x='Confirmados', y='taxa_letalidade_1')
```

Out[33]: <AxesSubplot:xlabel='Confirmados', ylabel='taxa_letalidade_1'>



In [34]:
```
sns.scatterplot(data = combined.query('Recuperados > 1000 and Rec
uperados < 100000'), x='Recuperados', y='taxa_letalidade_2')
```

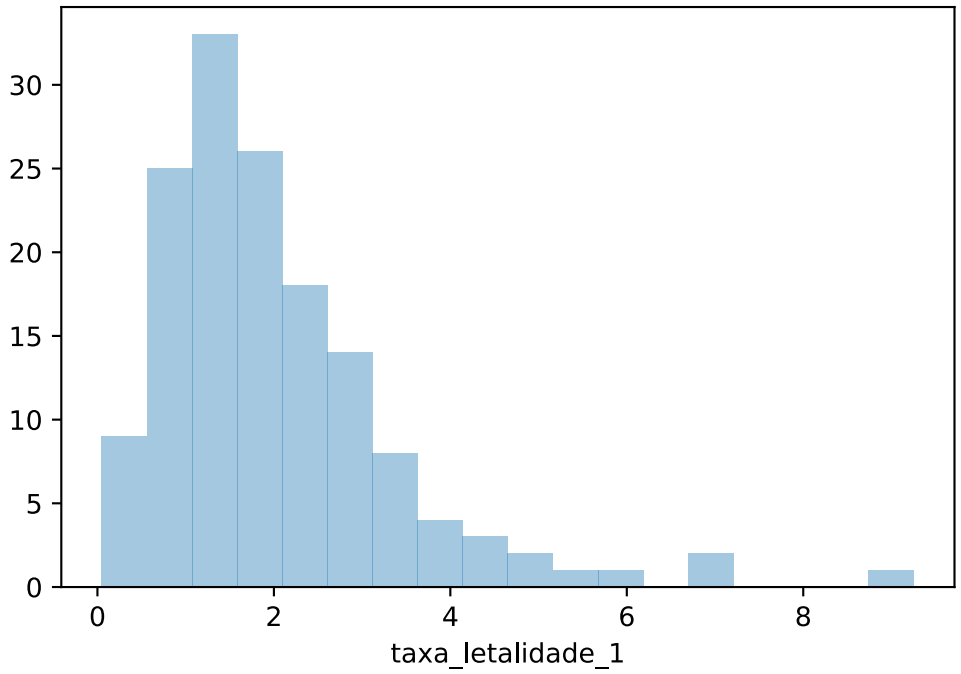
Out[34]: <AxesSubplot:xlabel='Recuperados', ylabel='taxa_letalidade_2'>



In [35]:
```
sns.distplot(combined.query('Confirmados > 10000')['taxa_letalidade_1'], kde=False)
```

```
/home/edcarlos/anaconda3/envs/data_science/lib/python3.6/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

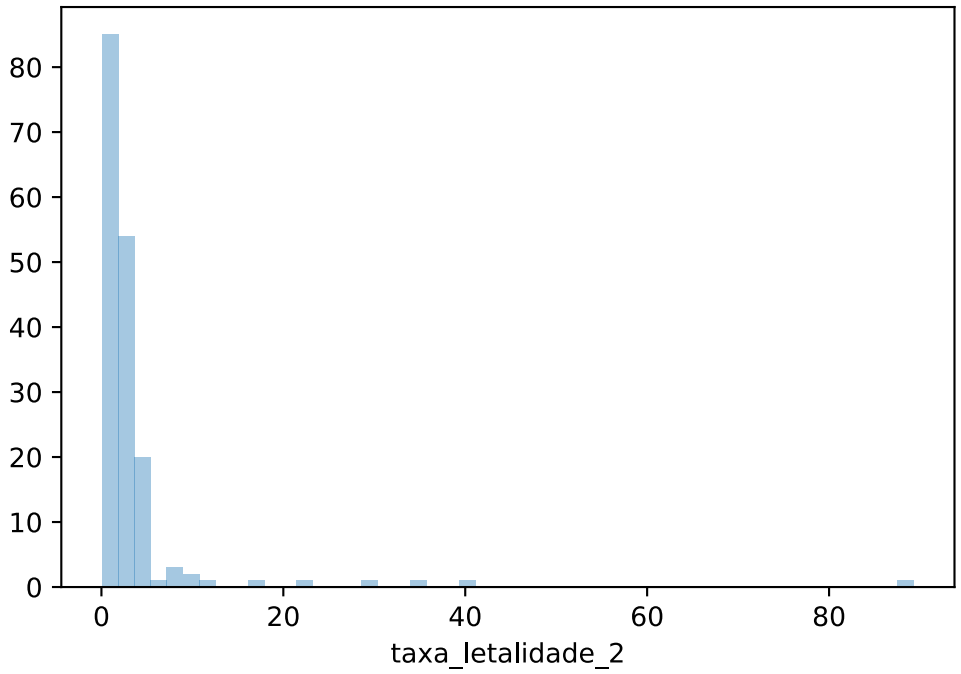
Out[35]: <AxesSubplot:xlabel='taxa_letalidade_1'>



In [36]:
```
sns.distplot(combined.query('Recuperados > 1000')['taxa_letalidade_2'], kde=False)
```

```
/home/edcarlos/anaconda3/envs/data_science/lib/python3.6/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[36]: <AxesSubplot:xlabel='taxa_letalidade_2'>



In [37]:
```
combined.sort_values('taxa_letalidade_1', ascending=False).query('Confirmados > 10000').head(10)
```

Out[37]:

Country/Region	Confirmados	Mortos	Recuperados	taxa_letalidade_1	taxa_letalidade_2
Mexico	2382745	220493	1903494	9.253739	10.381090
Syria	23788	1705	21073	7.167479	7.485293
Sudan	34889	2446	27949	7.010806	8.047376
Egypt	246909	14388	182024	5.827248	7.325418
Somalia	14486	753	6325	5.198122	10.638598
Ecuador	410870	19786	354499	4.815635	5.286346
China	102769	4846	97543	4.715430	4.732930
Bosnia and Herzegovina	202490	9051	169455	4.469850	5.070418
Afghanistan	63598	2745	55010	4.316173	4.752835
Bulgaria	414869	17343	364682	4.180356	4.539755

In [38]:
```
combined.sort_values('taxa_letalidade_2', ascending=False).query
('Recuperados > 1000').head(10)
```

Out[38]:

Country/Region	Confirmados	Mortos	Recuperados	taxa_letalidade_1	taxa_letalidade_2
United Kingdom	4468582	127946	15292	2.863235	89.324062
Netherlands	1627997	17725	26534	1.088761	40.048352
Spain	3615860	79432	150376	2.196766	34.564506
Yemen	6568	1294	3042	19.701583	29.843173
France	5942370	107973	379812	1.817002	22.135367
Ireland	254870	4941	23364	1.938635	17.456280
Greece	378485	11471	93764	3.030767	10.900366
Somalia	14486	753	6325	5.198122	10.638598
Mexico	2382745	220493	1903494	9.253739	10.381090
Sudan	34889	2446	27949	7.010806	8.047376

In [ ]: