

```
In [1]: pip install pandas==2.2.2 # Instalação da biblioteca pandas.
```

```
Collecting pandas==2.2.2
  Using cached pandas-2.2.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.23.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2.1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas==2.2.2)
  Using cached pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas==2.2.2)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas==2.2.2) (1.16.0)
Using cached pandas-2.2.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.0 MB)
Using cached pytz-2024.2-py2.py3-none-any.whl (508 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.2 pytz-2024.2 tzdata-2024.1
Note: you may need to restart the kernel to use updated packages.
```

```
In [3]: import pandas as pd # Importa da biblioteca pandas.
```

```
In [4]: # Cria um dataframe com os dados do arquivo
"DadosZazzleSemTratamento.csv".
df = pd.read_csv("DadosZazzleSemTratamento.csv")
```

```
In [7]: # Visualiza os dados do dataframe.
display(df)
```

	Date	Product ID	Product Title	Product Type	Custom
0	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	
1	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	
2	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
3	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
4	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
...	
32389	6/21/2012 9:51:00 AM	240-36297848-8052705	Cartão Elegante	zazzle_profilecard	
32390	3/23/2012 5:13:30 PM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	
32391	1/25/2012 1:32:32 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	
32392	12/14/2011 4:51:41 AM	149-99118322-0835239	Nossa Senhora da Conceição Aparecida	zazzle_bag	
32393	12/10/2011 4:48:28 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	

32394 rows × 18 columns

```
In [8]: # Vamos descartar a última coluna ("Original Product Type")  
        pois seus dados não são necessários.  
        df = df.drop(columns=["Original Product Type"])
```

```
In [9]: # Vamos visualizar como ficou o topo da tabela após a exclusão.  
        df.head()
```

Out[9]:

	Date	Product ID	Product Title	Product Type	Is Customized	Ca
0	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
1	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
2	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
3	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
4	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	

In [10]:

```
# Vamos visualizar o fim da tabela após a exclusão.  
df.tail()
```

Out[10]:

	Date	Product ID	Product Title	Product Type	Is Customized	Canceled
32389	6/21/2012 9:51:00 AM	240-36297848-8052705	Cartão Elegante	zazzle_profilecard	Yes	
32390	3/23/2012 5:13:30 PM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	Yes	
32391	1/25/2012 1:32:32 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	
32392	12/14/2011 4:51:41 AM	149-99118322-0835239	Nossa Senhora da Conceição Aparecida	zazzle_bag	Yes	
32393	12/10/2011 4:48:28 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	

In []:

```
# Buscamos obter a informação do total de vendas por país. A  
coluna "Shipped To"  
# nos fornece essa informação. Mas ela necessita de  
tratamentos.  
# Recebi do proprietário a informação de que provavelmente  
haverá um grande número de pedidos  
# dos EUA pois a filial da Zazzle é na Califórnia. (Zazzle é o  
serviço que a RicardoArtes utiliza  
# na confecção dos produtos ). Vou focar inicialmente no  
tratamento dos pedidos dos EUA.  
  
# Vamos analisar a linha 32389  
# Erdogan - Ridgefield, NJ  
# Há um padrão que parece repetir (nome_usuario - Distrito -  
Sigla Estado )  
# Ridgefield é um Distrito em New Jersey.  
# Se observarmos outras linhas esse padrão repete logo:
```

```
In [11]: import re # Importa a biblioteca de expressões regulares
          (regex).

          # Tratamento dados EUA.
          estados_validos = [
              'AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA',
              'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD',
              'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ',
              'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC',
              'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY',
              'DC'
          ]

          # Percorre o dataframe verificando o padrão na coluna "Shipped
          To".
          # Se os dois últimos caracteres do padrão representarem um
          estado válido
          # atualiza a coluna com o nome do país.

          for i in range(len(df)):
              enviado_para = df.at[i, 'Shipped To']
              match = re.match(r"^[^-]+ - [^,]+, ([A-Z]{2})",
              enviado_para)
              if match:
                  estado = match.group(1)
                  if estado in estados_validos:
                      df.at[i, 'Shipped To'] = 'United States'
```

```
In [13]: # Vamos ver como ficou o dataframe.
          display(df)
```

	Date	Product ID	Product Title	Product Type	Custom
0	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	
1	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	
2	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
3	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
4	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	
...	
32389	6/21/2012 9:51:00 AM	240-36297848-8052705	Cartão Elegante	zazzle_profilecard	
32390	3/23/2012 5:13:30 PM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	
32391	1/25/2012 1:32:32 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	
32392	12/14/2011 4:51:41 AM	149-99118322-0835239	Nossa Senhora da Conceição Aparecida	zazzle_bag	
32393	12/10/2011 4:48:28 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	

32394 rows × 17 columns

```
In [15]: # Vamos ver as 15 primeiras linhas.  
df.head(15)
```

Out[15]:

		Date	Product ID	Product Title	Product Type	Is Customized	C
	0	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
	1	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
	2	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	3	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	4	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	5	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	6	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	7	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	8	9/10/2024 10:44:52 AM	256-10893432-1682747	Professional Chic Elegant Plain and Monogram	visualpromotions_folder	Yes	
	9	9/10/2024 1:29:07 AM	240-71784709-9740850	Elegant Minimal Plain Professional Black	zazzle_businesscard	Yes	
	10	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
	11	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	

	Date	Product ID	Product Title	Product Type	Is Customized	C
12	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
13	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
14	9/9/2024 6:35:20 PM	240-71198261-7218569	Minimalist Modern Elegant Loyalty Discount	zazzle_flatloyaltycard	Yes	

```
In [16]: # Agora as 15 linhas finais.  
df.tail(15)
```


Out[16]:

	Date	Product ID	Product Title	Product Type	Is Customized	Canc
32379	10/5/2012 3:11:06 AM	240-99921669-9837741	Elegante 2	zazzle_profilecard	Yes	
32380	9/29/2012 7:23:14 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32381	9/29/2012 7:23:14 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32382	9/28/2012 9:30:49 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32383	9/27/2012 11:33:40 PM	240-96436692-6877736	Verde Nouveau	zazzle_profilecard	Yes	
32384	9/12/2012 6:00:45 PM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32385	9/7/2012 8:39:04 PM	240-99921669-9837741	Elegante 2	zazzle_profilecard	Yes	
32386	8/21/2012 3:38:54 PM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32387	7/20/2012 8:29:12 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32388	6/21/2012 9:51:00 AM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	Yes	
32389	6/21/2012 9:51:00 AM	240-36297848-8052705	Cartão Elegante	zazzle_profilecard	Yes	
32390	3/23/2012 5:13:30 PM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	Yes	
32391	1/25/2012 1:32:32 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	
32392	12/14/2011 4:51:41 AM	149-99118322-0835239	Nossa Senhora da Conceição Aparecida	zazzle_bag	Yes	
32393	12/10/2011 4:48:28 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	

In []:

```
# Vamos tratar os dados dos demais países.  
# Aqui mais um padrão ocorre:  
# Adriana - Sao Paulo, Brazil  
# Em alguns casos o nome do usuário não é informado:  
# - Japan  
# Há sempre uma constante: O nome do país é informado.  
# Podemos comparar o texto da célula e verificar se ele informa  
um país válido em algum ponto.  
# Se for localizado podemos substituir o valor da célula com o  
nome do país.
```

In [17]:

Lista de países

```
países_validos = [
    "Afghanistan", "Albania", "Algeria", "Andorra", "Angola",
    "Antigua and Barbuda", "Argentina", "Armenia",
    "Australia", "Austria", "Azerbaijan", "Bahamas", "Bahrain",
    "Bangladesh", "Barbados", "Belarus", "Belgium",
    "Belize", "Benin", "Bhutan", "Bolivia", "Bosnia and
    Herzegovina", "Botswana", "Brazil", "Brunei", "Bulgaria",
    "Burkina Faso", "Burundi", "Cabo Verde", "Cambodia",
    "Cameroon", "Canada", "Central African Republic", "Chad",
    "Chile", "China", "Colombia", "Comoros", "Congo, Democratic
    Republic of the", "Congo, Republic of the", "Costa Rica",
    "Croatia", "Cuba", "Cyprus", "Czech Republic", "Denmark",
    "Djibouti", "Dominica", "Dominican Republic",
    "East Timor", "Ecuador", "Egypt", "El Salvador",
    "Equatorial Guinea", "Eritrea", "Estonia", "Eswatini",
    "Ethiopia",
    "Fiji", "Finland", "France", "Gabon", "Gambia", "Georgia",
    "Germany", "Ghana", "Greece", "Grenada", "Guatemala",
    "Guinea", "Guinea-Bissau", "Guyana", "Haiti", "Honduras",
    "Hungary", "Iceland", "India", "Indonesia", "Iran", "Iraq",
    "Ireland", "Israel", "Italy", "Jamaica", "Japan", "Jordan",
    "Kazakhstan", "Kenya", "Kiribati", "Korea, North",
    "Korea, South", "Kosovo", "Kuwait", "Kyrgyzstan", "Laos",
    "Latvia", "Lebanon", "Lesotho", "Liberia", "Libya",
    "Liechtenstein", "Lithuania", "Luxembourg", "Madagascar",
    "Malawi", "Malaysia", "Maldives", "Mali", "Malta",
    "Marshall Islands", "Mauritania", "Mauritius", "Mexico",
    "Micronesia", "Moldova", "Monaco", "Mongolia",
    "Montenegro", "Morocco", "Mozambique", "Myanmar",
    "Namibia", "Nauru", "Nepal", "Netherlands", "New Zealand",
    "Nicaragua", "Niger", "Nigeria", "North Macedonia",
    "Norway", "Oman", "Pakistan", "Palau", "Panama",
    "Papua New Guinea", "Paraguay", "Peru", "Philippines",
    "Poland", "Portugal", "Qatar", "Romania", "Russia",
    "Rwanda", "Saint Kitts and Nevis", "Saint Lucia", "Saint
    Vincent and the Grenadines", "Samoa", "San Marino",
    "Sao Tome and Principe", "Saudi Arabia", "Senegal",
    "Serbia", "Seychelles", "Sierra Leone", "Singapore",
    "Slovakia", "Slovenia", "Solomon Islands", "Somalia".
```

```
----- , ----- , ----- , ----- ,
"South Africa", "South Sudan", "Spain", "Sri Lanka",
    "Sudan", "Suriname", "Sweden", "Switzerland", "Syria",
"Taiwan", "Tajikistan", "Tanzania", "Thailand",
    "Timor-Leste", "Togo", "Tonga", "Trinidad and Tobago",
"Tunisia", "Turkey", "Turkmenistan", "Tuvalu", "Uganda",
    "Ukraine", "United Arab Emirates", "United Kingdom",
"United States", "Uruguay", "Uzbekistan", "Vanuatu",
    "Vatican City", "Venezuela", "Vietnam", "Yemen", "Zambia",
"Zimbabwe"
]

dicionario_paises = {pais: pais for pais in paises_validos}

for index, row in df.iterrows():
    enviado_para = row['Shipped To']
    for pais in dicionario_paises:
        if pais in enviado_para:
            df.at[index, 'Shipped To'] = pais
            break
```

```
In [18]: # Vamos analisar o topo do dataframe após o processamento.
df.head(15)
```

Out[18]:

		Date	Product ID	Product Title	Product Type	Is Customized	C
	0	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
	1	9/10/2024 2:13:06 PM	217-02854107-2295764	Modern Minimalist Real Estate Black	zazzle_sticker	Yes	
	2	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	3	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	4	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	5	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	6	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	7	9/10/2024 2:01:50 PM	256-50771486-3204370	Personalized Teacher Name	hamptontech_selfinkingstamp	Yes	
	8	9/10/2024 10:44:52 AM	256-10893432-1682747	Professional Chic Elegant Plain and Monogram	visualpromotions_folder	Yes	
	9	9/10/2024 1:29:07 AM	240-71784709-9740850	Elegant Minimal Plain Professional Black	zazzle_businesscard	Yes	
	10	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
	11	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	

	Date	Product ID	Product Title	Product Type	Is Customized	C
12	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
13	9/9/2024 11:53:50 PM	217-86366175-7539770	Round Custom Your Company Logo	zazzle_sticker	Yes	
14	9/9/2024 6:35:20 PM	240-71198261-7218569	Minimalist Modern Elegant Loyalty Discount	zazzle_flatloyaltycard	Yes	

In [19]:

```
# Vamos analisar o final do dataframe.  
df.tail(15)
```

Out[19]:

	Date	Product ID	Product Title	Product Type	Is Customized	Canc
32379	10/5/2012 3:11:06 AM	240-99921669-9837741	Elegante 2	zazzle_profilecard	Yes	
32380	9/29/2012 7:23:14 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32381	9/29/2012 7:23:14 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32382	9/28/2012 9:30:49 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32383	9/27/2012 11:33:40 PM	240-96436692-6877736	Verde Nouveau	zazzle_profilecard	Yes	
32384	9/12/2012 6:00:45 PM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32385	9/7/2012 8:39:04 PM	240-99921669-9837741	Elegante 2	zazzle_profilecard	Yes	
32386	8/21/2012 3:38:54 PM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32387	7/20/2012 8:29:12 AM	240-27050935-7725549	Nouveau graphite	zazzle_profilecard	Yes	
32388	6/21/2012 9:51:00 AM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	Yes	
32389	6/21/2012 9:51:00 AM	240-36297848-8052705	Cartão Elegante	zazzle_profilecard	Yes	
32390	3/23/2012 5:13:30 PM	240-48013704-5084466	Grécia Contemporânea	zazzle_profilecard	Yes	
32391	1/25/2012 1:32:32 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	
32392	12/14/2011 4:51:41 AM	149-99118322-0835239	Nossa Senhora da Conceição Aparecida	zazzle_bag	Yes	
32393	12/10/2011 4:48:28 PM	239-89217540-7970094	Aparecida do Norte	zazzle_postcard	Yes	

```
In [ ]: # Vamos fazer o tratamento do restante dos dados. Com
        # informações que tenho até aqui os dados
        # em sua grande maioria foram tratados.
        # Vamos automatizar a verificação do restante dos dados para
        # constatar o estado atual:
        # Eu desejo preliminarmente obter uma lista de países que
        # ocorrem nos meus dados
        # e a quantidade de linhas com erros.
```

```
In [20]: # Verificar o dataframe para checar quais linhas ainda
        # apresentam erro.
        # E verificar de vale a pena tratá-las.

        # Armazena países válidos localizados.
        paises_encontrados = set()
        # Armazena as linhas que apresentam erros.
        linhas_erro = []

        for index, row in df.iterrows():
            pais = row['Shipped To']
            if pais in paises_validos:
                paises_encontrados.add(pais)
            else:
                linhas_erro.append(index)

        # Numero de países válidos encontrados.
        quantidade_paises_validos = len(paises_encontrados)
        quantidade_linhas_erro = len(linhas_erro)

        # Exibir resultados
        print("Quantidade de países diferentes e válidos encontrados:",
              quantidade_paises_validos)
        print("Quantidade de linhas com erros:",
              quantidade_linhas_erro)
        print("Linhas que apresentam erro:", linhas_erro)

        # Exibir lista com países válidos encontrados no dataframe.
        print(" Países válidos encontrados no DataFrame:")
        print(sorted(paises_encontrados))
```


Quantidade de países diferentes e válidos encontrados: 47

Quantidade de linhas com erros: 155

Linhas que apresentam erro: [69, 108, 439, 773, 1033, 1322, 1591, 1760, 3119, 4088, 5207, 5208, 5209, 5210, 7763, 7802, 8515, 8729, 9219, 9567, 9568, 9569, 9570, 9571, 9572, 9573, 9579, 10444, 10573, 10698, 10785, 10904, 10905, 11080, 11285, 12089, 12182, 12848, 13084, 14275, 14446, 14447, 14449, 14693, 14958, 15140, 15169, 15324, 15496, 15566, 15568, 15569, 15597, 15769, 15770, 16418, 16958, 17132, 17327, 17328, 17553, 18573, 18885, 20154, 20262, 21467, 21923, 22264, 23018, 23100, 23259, 23260, 23261, 23262, 23263, 23264, 23265, 23266, 23267, 23856, 24022, 24069, 24593, 24623, 24670, 24824, 24825, 24826, 24827, 24828, 24829, 24830, 24831, 24832, 24833, 24834, 24835, 24836, 24837, 24838, 24839, 24840, 24841, 24842, 25518, 25743, 26021, 26022, 26023, 26024, 26216, 26217, 26959, 27584, 27585, 27586, 27587, 27588, 27589, 27590, 27591, 27592, 27593, 27594, 27595, 27596, 27597, 28025, 28118, 28127, 29714, 29911, 29912, 30272, 30377, 30467, 30488, 30611, 30640, 30684, 30726, 30745, 30746, 30858, 30958, 30961, 30987, 31094, 31614, 31615, 31634, 31709, 31970, 32225, 32261]

Países válidos encontrados no DataFrame:

['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada', 'Chile', 'Colombia', 'Croatia', 'Denmark', 'Egypt', 'Finland', 'France', 'Germany', 'Greece', 'Hungary', 'Ireland', 'Israel', 'Italy', 'Japan', 'Jordan', 'Kenya', 'Kuwait', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malaysia', 'Mali', 'Mexico', 'Monaco', 'Morocco', 'Netherlands', 'New Zealand', 'Norway', 'Peru', 'Philippines', 'Portugal', 'Qatar', 'Singapore', 'Slovenia', 'Spain', 'Sweden', 'Switzerland', 'Taiwan', 'United Arab Emirates', 'United Kingdom', 'United States']

In [21]:

```
# Constatamos acima que possuímos uma lista de 47 países diferentes nos nossos dados.  
# Constatamos também que há um total de 155 linhas com erros.  
# Vamos analisar se o custo de tratamento destas linhas restantes é viável.  
  
# Informações sobre o dataframe.  
df.shape
```

Out[21]: (32394, 17)

In []:

```
# Nossos dados apresentam um total de 32394 registros  
# Porcentagem=(155/32394)×100  
# Logo :  
# 0.478  
# Então, 155 é aproximadamente 0.478% de 32.394 o que não representa  
# um valor significativo para o total de registros.  
# Logo, com o consentimento do gestor tomei a iniciativa de descartar esses dados da análise.
```

```
In [22]: # Processo de eliminação das linhas com erro:
# Vamos aproveitar aquela lista de linhas que apresentaram
# defeito.

indices_remocao = [69, 108, 439, 773, 1033, 1322, 1591, 1760,
3119, 4088, 5207, 5208, 5209, 5210, 7763, 7802, 8515, 8729,
9219, 9567, 9568, 9569, 9570, 9571, 9572, 9573, 9579, 10444,
10573, 10698, 10785, 10904, 10905, 11080, 11285, 12089, 12182,
12848, 13084, 14275, 14446, 14447, 14449, 14693, 14958, 15140,
15169, 15324, 15496, 15566, 15568, 15569, 15597, 15769, 15770,
16418, 16958, 17132, 17327, 17328, 17553, 18573, 18885, 20154,
20262, 21467, 21923, 22264, 23018, 23100, 23259, 23260, 23261,
23262, 23263, 23264, 23265, 23266, 23267, 23856, 24022, 24069,
24593, 24623, 24670, 24824, 24825, 24826, 24827, 24828, 24829,
24830, 24831, 24832, 24833, 24834, 24835, 24836, 24837, 24838,
24839, 24840, 24841, 24842, 25518, 25743, 26021, 26022, 26023,
26024, 26216, 26217, 26959, 27584, 27585, 27586, 27587, 27588,
27589, 27590, 27591, 27592, 27593, 27594, 27595, 27596, 27597,
28025, 28118, 28127, 29714, 29911, 29912, 30272, 30377, 30467,
30488, 30611, 30640, 30684, 30726, 30745, 30746, 30858, 30958,
30961, 30987, 31094, 31614, 31615, 31634, 31709, 31970, 32225,
32261]

# Remover as linhas indesejadas do dataframe e armazenar em um
# novo.
df_limpo = df.drop(indices_remocao, errors='ignore')
```

```
In [ ]: # Vamos comparar os dados antes e após a remoção:
```

```
In [23]: # Antes da remoção:
df.shape
```

Out[23]: (32394, 17)

```
In [24]: # Após a remoção:
df_limpo.shape
```

Out[24]: (32239, 17)

```
In [ ]: # Logo
        # 32394 - 32239 = 155
        # As 155 linhas com erros foram descartadas.
```

```
In [25]: # Vamos armazenar o novo dataframe em um arquivo diferente para
        # trabalhar com ele mais a frente:
        df_limpo.to_csv("DadosZazzleComTratamento.csv")
```

```
In [26]: # Vamos capturar os dados do novo arquivo para um dataframe.
        df = pd.read_csv("DadosZazzleComTratamento.csv")
```

```
In [27]: # Vamos visualizar as linhas deste dataframe.
        df.shape
```

Out[27]: (32239, 18)

```
In [37]: pip install plotly==5.24.1 # Biblioteca para gerar gráficos.
```

```
Collecting plotly==5.24.1
  Using cached plotly-5.24.1-py3-none-any.whl.metadata (7.3 kB)
Collecting tenacity>=6.2.0 (from plotly==5.24.1)
  Using cached tenacity-9.0.0-py3-none-any.whl.metadata (1.2 kB)
Requirement already satisfied: packaging in /home/leonardo/Jupyter/ambien
tes/padrao/lib/python3.11/site-packages (from plotly==5.24.1) (24.1)
Using cached plotly-5.24.1-py3-none-any.whl (19.1 MB)
Using cached tenacity-9.0.0-py3-none-any.whl (28 kB)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.24.1 tenacity-9.0.0
Note: you may need to restart the kernel to use updated packages.
```

```
In [85]: # Geração do mapa global.
# Importa as bibliotecas que vamos usar.
import pandas as pd
import plotly.express as px

# Efetua a contagem do número de pedidos por país.
contagem_paises = df['Shipped To'].value_counts().reset_index()
# Troca os nomes das colunas do novo dataframe.
contagem_paises.columns = ['País', 'Pedidos']

# Gera o gráfico interativo.
fig = px.choropleth(contagem_paises,
                    locations='País',
                    locationmode='country names',
                    color='Pedidos',
                    title='Número de Pedidos por País',
                    color_continuous_scale='Rainbow')

# Exibe o gráfico.
fig.show()
```

In [49]:

```
# Importa as bibliotecas que preciso para o gráfico.
import pandas as pd
import plotly.express as px

# Efetua a contagem do pedido.
# Poderia utilizar tambem os dados da célula acima.
contagem_paises = df['Shipped To'].value_counts().reset_index()
contagem_paises.columns = ['País', 'Pedidos']

# Obtém os 10 países com mais pedidos.
top_10_paises = contagem_paises.head(10)

# Gera o gráfico de barras
fig = px.bar(top_10_paises,
              x='País',
              y='Pedidos',
              color='Pedidos',
              title='Top 10 Países com Mais Pedidos',
              color_continuous_scale='Rainbow') # Usando uma
escala de cores mais viva

# Ajustar o layout do gráfico
fig.update_layout(xaxis_title='País',
                  yaxis_title='Número de Pedidos',
                  xaxis_tickangle=-45) # Rotaciona os rótulos
do eixo x para melhor legibilidade

# Mostrar o gráfico
fig.show()
```



```
In [51]: # Geração do gráfico de pizza.

# Importa as bibliotecas que usaremos.
import pandas as pd
import plotly.express as px

# Contagem de número de pedidos por país.
contagem_países = df['Shipped To'].value_counts().reset_index()
contagem_países.columns = ['País', 'Pedidos']

# Selecionar os 10 países com mais pedidos
top_5_países = contagem_países.head(5)

# Gera o gráfico
fig = px.pie(top_5_países,
             names='País',
             values='Pedidos',
             title='Top 5 Países com Mais Pedidos',
             color='País', # Cada fatia representa um país com
uma cor.

color_discrete_sequence=px.colors.qualitative.Plotly) # Define
escala de cores qualitativa

# Exibe o gráfico.
fig.show()
```


In [58]:

```
# Importar as bibliotecas necessárias
import pandas as pd
import plotly.express as px

# Converter a coluna 'Date' para o tipo datetime especificando o formato
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y %I:%M:%S %p')

# Extrair o ano da coluna 'Date'
df['Ano'] = df['Date'].dt.year

# Contar o número de pedidos por ano
pedidos_por_ano =
df.groupby('Ano').size().reset_index(name='Pedidos')

# Criar o gráfico de linha com plotly
fig = px.line(pedidos_por_ano,
               x='Ano',
               y='Pedidos',
               title='Número de Pedidos por Ano',
               markers=True) # Adiciona marcadores aos pontos de dados

# Ajustar o layout do gráfico
fig.update_layout(xaxis_title='Ano',
                  yaxis_title='Número de Pedidos',
                  xaxis=dict(tickmode='linear')) # Define o modo de ticks do eixo x como linear

# Mostrar o gráfico
fig.show()
```

```
In [ ]: # Geração de relatório. Instale as bibliotecas abaixo para a  
        geração do arquivo pdf.
```

```
In [83]: pip install pandas==2.2.2 plotly==5.24.1 reportlab==4.2.2  
         kaleido==0.2.1 # Instalação da biblioteca.
```

Requirement already satisfied: pandas==2.2.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (2.2.2)
Requirement already satisfied: plotly==5.24.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (5.24.1)
Requirement already satisfied: matplotlib==3.9.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (3.9.2)
Requirement already satisfied: reportlab==4.2.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (4.2.2)
Requirement already satisfied: kaleido==0.2.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (0.2.1)
Requirement already satisfied: numpy>=1.23.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2.1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from pandas==2.2.2) (2024.1)
Requirement already satisfied: tenacity>=6.2.0 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from plotly==5.24.1) (9.0.0)
Requirement already satisfied: packaging in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from plotly==5.24.1) (24.1)
Requirement already satisfied: contourpy>=1.0.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (1.4.7)
Requirement already satisfied: pillow>=8 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from matplotlib==3.9.2) (3.1.4)
Requirement already satisfied: chardet in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from reportlab==4.2.2) (5.2.0)
Requirement already satisfied: six>=1.5 in /home/leonardo/Jupyter/ambientes/padrao/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas==2.2.2) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```
In [8]: import pandas as pd
df = pd.read_csv("DadosZazzleComTratamento.csv")
```

```
In [ ]: # Geração do relatório
```

```
In [37]: # Geração do mapa global.
# Importa as bibliotecas que vamos usar.
import pandas as pd
import plotly.express as px
from io import BytesIO
from reportlab.lib.pagesizes import A4
from reportlab.lib import colors
from reportlab.platypus import SimpleDocTemplate, Image, Table,
TableStyle, Paragraph, PageBreak
from reportlab.lib.styles import getSampleStyleSheet,
ParagraphStyle
from reportlab.lib.units import inch
from reportlab.lib.enums import TA_CENTER
from reportlab.pdfgen import canvas

# Efetua a contagem do número de pedidos por país.
contagem_paises = df['Shipped To'].value_counts().reset_index()
# Troca os nomes das colunas do novo dataframe.
contagem_paises.columns = ['País', 'Pedidos']

# Gera o gráfico interativo.
fig = px.choropleth(contagem_paises,
                    locations='País',
                    locationmode='country names',
                    color='Pedidos',
                    title='Número de Pedidos por País',
                    color_continuous_scale='Rainbow')

# Conversão do gráfico para imagem estática.
fig.write_image('mapa.png', engine='kaleido')

# Gráfico de barras.
# Efetua a contagem do pedido.
# Poderia utilizar também os dados da célula acima.
contagem_paises = df['Shipped To'].value_counts().reset_index()
contagem_paises.columns = ['País', 'Pedidos']

# Obtém os 10 países com mais pedidos.
top_10_paises = contagem_paises.head(10)

# Gera o gráfico de barras
```

```
fig = px.bar(top_10_paises,
             x='País',
             y='Pedidos',
             color='Pedidos',
             title='Top 10 Países com Mais Pedidos',
             color_continuous_scale='Rainbow') # Usando uma
escala de cores mais viva

# Ajustar o layout do gráfico
fig.update_layout(xaxis_title='País',
                  yaxis_title='Número de Pedidos',
                  xaxis_tickangle=-45) # Rotaciona os rótulos
do eixo x para melhor legibilidade

fig.write_image('barras.png', engine='kaleido')

# Geração do gráfico de pizza.

# Contagem de número de pedidos por país.
contagem_paises = df['Shipped To'].value_counts().reset_index()
contagem_paises.columns = ['País', 'Pedidos']

# Selecionar os 10 países com mais pedidos
top_5_paises = contagem_paises.head(5)

# Gera o gráfico
fig = px.pie(top_5_paises,
             names='País',
             values='Pedidos',
             title='Top 5 Países com Mais Pedidos',
             color='País', # Cada fatia representa um país com
uma cor.

color_discrete_sequence=px.colors.qualitative.Plotly) # Define
escala de cores qualitativa

fig.write_image('pizza.png', engine='kaleido')

# Gráfico de linha.

# Converter a coluna 'Date' para o tipo datetime especificando
```

```
o formato
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y
%i:%M:%S %p')

# Extrair o ano da coluna 'Date'
df['Ano'] = df['Date'].dt.year

# Contar o número de pedidos por ano
pedidos_por_ano =
df.groupby('Ano').size().reset_index(name='Pedidos')

# Criar o gráfico de linha com plotly
fig = px.line(pedidos_por_ano,
               x='Ano',
               y='Pedidos',
               title='Número de Pedidos por Ano',
               markers=True) # Adiciona marcadores aos pontos
de dados

# Ajustar o layout do gráfico
fig.update_layout(xaxis_title='Ano',
                  yaxis_title='Número de Pedidos',
                  xaxis=dict(tickmode='linear')) # Define o
modo de ticks do eixo x como linear

fig.write_image('linha.png', engine='kaleido')

# Bloco de geração do PDF.
arquivo_pdf = "Relatorio.pdf"
# Define o nome de arquivo e formato de página.
doc = SimpleDocTemplate(arquivo_pdf, pagesize=A4)
elements = []

# Adição da folha de rosto.
styles = getSampleStyleSheet()
title_style = ParagraphStyle(
    'TitleStyle',
    parent=styles['Title'],
    alignment=TA_CENTER,
    fontSize=24,
    spaceAfter=12
```

```
)

logo_style = ParagraphStyle(
    'LogoStyle',
    parent=styles['Normal'],
    alignment=TA_CENTER,
    spaceAfter=12
)

title = Paragraph("Relatório de Pedidos (2011 - 2024)",
title_style)
logo = Image("LogoLoja.jpeg", width=150, height=150)

# Adição da folha de rosto centralizada.
width, height = A4

# Espaçamento para centralizar verticalmente.
content_height = height - 2 * inch
title_height = 24
logo_height = 150

# Espaços verticais para centralização
space_before_title = (content_height - title_height -
logo_height) / 2
space_after_title = 0.5 * inch

elements.append(Paragraph("<br/>" * int(space_before_title
/12), styles['Normal']))
elements.append(title)
#elements.append(Paragraph("<br/>" * int(space_before_title
/12), styles['Normal']))
elements.append(logo)
elements.append(Paragraph("<br/><br/>", styles['Normal']))

# Adiciona uma quebra de página.
elements.append(PageBreak())

# Função para adição das imagens ao PDF.
def adiciona_imagem_pdf(caminho_imagem, width=6*inch,
height=4*inch):
    imagem = Image(caminho_imagem, width=width, height=height)
```

```
elements.append(imagem)
elements.append(Paragraph("<br/>", styles['Normal'])) #
Espaço após adição de imagem.

# Chama a função e adiciona todas as imagens ao PDF.
for arquivo_imagem in ['mapa.png', 'barras.png', 'pizza.png',
'linha.png']:adiciona_imagem_pdf(arquivo_imagem)

# Adiciona rodapé e numeração das páginas.
def footer(canvas, doc):
    canvas.saveState()
    canvas.setFont('Helvetica', 8)
    canvas.drawString(1*inch, 0.75*inch, "Gerado Por: Leonardo
Bruno / 202301011744 / 202301011744@alunos.estacio.br /
souzalb@proton.me / TADS")
    canvas.drawRightString(A4[0] - 1*inch, 0.75*inch,
str(doc.page))
    canvas.restoreState()

doc.build(elements, onFirstPage=footer, onLaterPages=footer)

print("Relatório Gerado.")
```

Relatório Gerado.

In []:

In []: