

TP2

IMPORTACION DE LIBRERIAS

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

FUNCIONES

```
In [ ]: ##explorar los datos de los df y los muestra
def explorar_df(df) -> None:
    print('Muestra de datos')
    df.head()
    print('\nFormato del dataframe')
    df.shape
    print('\nBúsqueda de valores nulos por columna')
    df.isnull().sum()
    print('\nFormato de los datos por columna')
    df.dtypes

##Chequeo de valores null en el df
def check_null_values(df):
    if df.isnull().sum().values.sum() > 0:

        df = limpiar_nan(df) #quita los NaN
    else:
        pass
    return df

##buscar columnas identicas (duplicadas)

def search_duplicates(df):

    columns = df.columns
    # Inicializar una lista para almacenar las columnas que son exactamente iguales
    duplicates = []
    # Comparar todas las columnas entre sí
    for i in range(len(columns)):
        for j in range(i + 1, len(columns)):
            if df[columns[i]].equals(df[columns[j]]):
                duplicates.append((columns[i], columns[j]))

    print("Pares de columnas que son exactamente iguales:", len(duplicates))
    return duplicates

##buscar filas identicas (duplicadas)
def search_duplicates_row(df):

    if df.duplicated().sum() != 0 :
        print( f'Pares de filas que son exactamente iguales: {df.duplicated().sum()}' )

        df = df.drop_duplicates(subset=df.columns, keep='first', inplace=True)
    else:
        print( f'Pares de filas que son exactamente iguales: 0' )
```

```

        return
    return df

##transforma Los valores de la col distributors para que se presenten como int
def transformar_valor(valor):
    if valor < 2000 and valor > 999:
        return valor
    return valor/10

def transformar_valor2(valor):
    if valor < 2 and valor > 1 :
        return valor*1000
    return valor

##rellena Los NaN con 0
def limpiar_nan(df):
    return df.fillna(0)

##castea una columna a int
def cast_int(df):
    return df.astype(str).str.replace('.', '').astype(int).apply(transformar_valor)

##limpiar simbolos y reemplazar otros
def limpiar_simbolos(df):
    return df.replace('$', '').replace('.', '').replace(',', '.', ',')

##elimina columnas innecesarias o vacias
def del_columnas(df, lista_col):
    for col in lista_col:
        if col in df.columns:
            df.drop(col, axis=1, inplace=True)
        else:
            pass
    return df

##Crea un configuracion inicial de como se deben mostrar Los datos del df
def setup_df():

    pd.set_option('display.max_rows', None)
    pd.set_option('display.float_format', '{:.2f}'.format)

```

- Configuración de df

```

In [ ]: # Configuración de como mostrar Los datos del DataFrame
        #setup_df() #aplicar config

```

IMPORTACION Y OBTENCIÓN DE DATOS

```

In [ ]: #!-----
        #!IMPORTACION DE DATOS
        #!-----

        ##Lista con Los nombres de Las hojas de La base de datos
        HOJAS = ['sales_in_paraguay', 'distributors_profiles', 'exports_to_paraguay', 'locatio

        ##id de La planilla que contiene todas Las hojas

```

```
id_sheet = '1_1lWfVuc66VgauNo1EzNEpAg55nXHgkp4CNTxpbDETK'

##Lista que guarda todas las urls correspondientes a la base de datos para poder se
URLs = [f'https://docs.google.com/spreadsheets/d/{id_sheet}/gviz/tq?tqx=out:csv&she

##creacion de Los dataframes
dataframes = [pd.read_csv(url) for url in URLs]

##desempaquetado de los df
df_sales_in_paraguay, df_distributors_profiles, df_exports_to_paraguay, df_location
```

INSPECCION PRELIMINAR

1.Ventas

- Primeros 5 elementos del *df*

```
In [ ]: #primero 5 elementos del df
df_sales_in_paraguay.head()
```

```
Out[ ]:
```

	distributor	Rubber	Brass	Vinyl	Granite	Stone	Brick
0	583.00	\$54.510.203,61	\$45.268.636,86	\$51.579.748,25	\$21.780.180,58	\$26.576.776,52	\$0,00
1	1104.00	\$32.438.788,20	\$25.837.100,49	\$36.603.264,50	\$21.883.374,92	\$1.473.437,08	\$0,00
2	1384.00	\$21.780.180,58	\$78.927.599,01	\$25.837.100,49	\$33.102.840,61	\$51.579.748,25	\$0,00
3	379.00	\$79.358.855,35	\$90.185.311,22	\$45.268.636,86	\$54.510.203,61	\$59.358.855,35	\$0,00
4	1599.00	\$11.758.005,07	\$21.780.180,58	\$57.187.306,41	\$9.945.371,16	\$32.067.534,68	\$0,00

- Tamaño del *df*

```
In [ ]: #tamaño del df
df_sales_in_paraguay.shape
```

```
Out[ ]: (45, 13)
```

- Valores nulos en el *df*

```
In [ ]: #valores Nulos en el df
df_sales_in_paraguay.isnull().sum()
```

```
Out[ ]: distributor    1
        Rubber      0
        Brass       0
        Vinyl       0
        Granite     0
        Stone       0
        Brick       0
        Aluminum    0
        Glass       0
        Plexiglass  0
        Steel       0
        Wood        0
        Plastic     0
        dtype: int64
```

- tipos de datos en el df

```
In [ ]: #tipos de datos en el df
        df_sales_in_paraguay.dtypes
```

```
Out[ ]: distributor    float64
        Rubber      object
        Brass       object
        Vinyl       object
        Granite     object
        Stone       object
        Brick       object
        Aluminum    object
        Glass       object
        Plexiglass  object
        Steel       object
        Wood        object
        Plastic     object
        dtype: object
```

2.Exportaciones

- Primeros 5 elementos del df

```
In [ ]: #primero 5 elementos del df
        df_exports_to_paraguay.head()
```

```
Out[ ]:
```

	distributor	Rubber	Brass	Vinyl	Granite	Stone	
0	1.53	\$22.431.099,00	\$36.031.577,00	\$31.118.167,00	\$21.322.223,00	\$35.382.848,00	\$35.280.000,00
1	1.55	\$27.566.922,00	\$21.996.538,00	\$39.412.316,00	\$25.681.987,00	\$41.861.783,00	\$22.408.000,00
2	1.67	\$37.577.095,00	\$41.457.655,00	\$31.467.967,00	\$37.577.926,00	\$35.845.106,00	\$42.953.000,00
3	364.00	\$36.012.730,00	\$41.667.692,00	\$22.837.073,00	\$29.288.200,00	\$39.553.494,00	\$33.513.000,00
4	920.00	\$43.416.417,00	\$36.290.780,00	\$23.679.738,00	\$21.183.706,00	\$25.210.622,00	\$30.864.000,00

- Tamaño del df

```
In [ ]: #tamaño del df
df_exports_to_paraguay.shape
```

```
Out[ ]: (46, 16)
```

*Valores nulos en el df

```
In [ ]: #valores Nulos en el df
df_exports_to_paraguay.isnull().sum()
```

```
Out[ ]: distributor      0
Rubber             0
Brass              0
Vinyl              0
Granite            0
Stone              0
Brick              0
Aluminum           0
Glass              0
Plexiglass         0
Steel              0
Wood               0
Plastic            0
Unnamed: 13        46
Unnamed: 14        45
Columnas           45
dtype: int64
```

- Tipos de datos en el df

```
In [ ]: #tipos de datos en el df
df_exports_to_paraguay.dtypes
```

```
Out[ ]: distributor    float64
Rubber              object
Brass               object
Vinyl              object
Granite            object
Stone              object
Brick              object
Aluminum           object
Glass              object
Plexiglass         object
Steel              object
Wood               object
Plastic            object
Unnamed: 13        float64
Unnamed: 14        object
Columnas           object
dtype: object
```

3.Distribuidores

- Primeros 5 elementos del df

```
In [ ]: #primero 5 elementos del df
df_distributors_profiles.head()
```

Out[]:

	id	distributor	distributor activities	years in the construction market
0	565.00	Abernathy-Hayes	construction materials import/distribution, ir...	15
1	1.38	Balistreri LLC	construction materials import/distribution, si...	22
2	1.18	Brekke-Stiedemann	construction materials import/distribution	13
3	1.53	Collins LLC	construction materials import/distribution	12
4	29.00	Cummings-Ward	construction materials import/distribution, ir...	23

- Tamaño del df

In []:

```
#tamaño del df
df_distributors_profiles.shape
```

Out[]:

(46, 4)

- Valores nulos en el df

In []:

```
#valores Nulos en el df
df_distributors_profiles.isnull().sum()
```

Out[]:

```
id                2
distributor       2
distributor activities  2
years in the construction market  0
dtype: int64
```

- Tipos de datos en el df

In []:

```
#tipos de datos en el df
df_distributors_profiles.dtypes
```

Out[]:

```
id                float64
distributor       object
distributor activities  object
years in the construction market  int64
dtype: object
```

LIMPIEZA Y ACONDICIONAMIENTO DE DATOS

1.Limpieza ventas

In []:

```
#!/-----
#! Limpieza df_sales_in_paraguay

df_sales_in_paraguay = check_null_values(df_sales_in_paraguay)#chekear valores NaN
```

```
df_sales_in_paraguay = df_sales_in_paraguay.drop(df_sales_in_paraguay.index[-1]) #eliminar la ultima fila

a = search_duplicates(df_sales_in_paraguay)
if len(a) != 0:
    df_sales_in_paraguay = del_columnas(df=df_sales_in_paraguay, lista_col=a) #eliminar columnas con duplicados

search_duplicates_row(df_sales_in_paraguay) # buscar filas duplicadas y eliminarlas

for columna in df_sales_in_paraguay.columns:
    if columna == 'distributor':
        df_sales_in_paraguay[columna] = df_sales_in_paraguay[columna].apply(transformar_distributor)
    else:
        df_sales_in_paraguay[columna] = df_sales_in_paraguay[columna].apply(limpiar_valor)
df_sales_in_paraguay[columna] = df_sales_in_paraguay[columna].astype(float)
```

Pares de columnas que son exactamente iguales: 0

Pares de filas que son exactamente iguales: 0

- Primeros 5 elementos del df

```
In [ ]: #inspeccion de los datos obtenidos luego de la limpieza
df_sales_in_paraguay.head()
```

```
Out[ ]:      distributor  Rubber  Brass  Vinyl  Granite  Stone  Brick  Aluminum
0          583  54510203.61  45268636.86  51579748.25  21780180.58  26576776.52  0.00  55872547.71
1          1104  32438788.20  25837100.49  36603264.50  21883374.92  1473437.08  0.00  46239695.30
2          1384  21780180.58  78927599.01  25837100.49  33102840.61  51579748.25  0.00  21883374.92
3           379  79358855.35  90185311.22  45268636.86  54510203.61  59358855.35  0.00  79358855.35
4          1599  11758005.07  21780180.58  57187306.41  9945371.16  32067534.68  0.00  53172624.14
```

- Tamaño del df

```
In [ ]: #tamaño del df
df_sales_in_paraguay.shape
```

```
Out[ ]: (44, 13)
```

- Valores nulos en el df

```
In [ ]: #valores Nulos en el df
df_sales_in_paraguay.isnull().sum()
```

```
Out[ ]: distributor    0
Rubber            0
Brass             0
Vinyl             0
Granite           0
Stone             0
Brick             0
Aluminum          0
Glass            0
Plexiglass        0
Steel            0
Wood              0
Plastic           0
dtype: int64
```

- Tipos de datos en el df

```
In [ ]: #tipos de datos en el df
df_sales_in_paraguay.dtypes
```

```
Out[ ]: distributor    int64
Rubber            float64
Brass             float64
Vinyl             float64
Granite           float64
Stone             float64
Brick             float64
Aluminum          float64
Glass            float64
Plexiglass        float64
Steel            float64
Wood              float64
Plastic           float64
dtype: object
```

2.Limpieza exportaciones

```
In [ ]: #!-----
##!Limpieza df_exports_to_paraguay

a = search_duplicates(df_sales_in_paraguay)

if len(a) != 0:
    a = a + ['Columnas', 'Unnamed: 13', 'Unnamed: 14']
    df_sales_in_paraguay = del_columnas(df=df_sales_in_paraguay, lista_col=a)
else:
    df_exports_to_paraguay = del_columnas(df_exports_to_paraguay, ['Columnas', 'Unna

df_exports_to_paraguay = check_null_values(df_exports_to_paraguay) #chekear valores
search_duplicates_row(df_exports_to_paraguay) # buscar filas duplicadas y eliminarl
for columna in df_exports_to_paraguay.columns:
    if columna == 'distributor':
        df_exports_to_paraguay[columna] = df_exports_to_paraguay[columna].apply(tr

    else:
        df_exports_to_paraguay[columna] = df_exports_to_paraguay[columna].apply
        df_exports_to_paraguay[columna] = df_exports_to_paraguay[columna].astyp
```

Pares de columnas que son exactamente iguales: 0

Pares de filas que son exactamente iguales: 2

- Primeros 5 elementos en el df

```
In [ ]: #inspeccion de los datos obtenidos luego de la limpieza
df_exports_to_paraguay.head()
```

```
Out[ ]:
```

	distributor	Rubber	Brass	Vinyl	Granite	Stone	Brick	Alu
0	1526	22431099.00	36031577.00	31118167.00	21322223.00	35382848.00	35280292.00	3236
1	1553	27566922.00	21996538.00	39412316.00	25681987.00	41861783.00	22408742.00	4069
2	1666	37577095.00	41457655.00	31467967.00	37577926.00	35845106.00	42953168.00	3381
3	364	36012730.00	41667692.00	22837073.00	29288200.00	39553494.00	33513588.00	3682
4	920	43416417.00	36290780.00	23679738.00	21183706.00	25210622.00	30864041.00	4117

- Tamaño del df

```
In [ ]: #tamaño del df
df_exports_to_paraguay.shape
```

```
Out[ ]: (44, 13)
```

- Valores nulos en el df

```
In [ ]: #valores Nulos en el df
df_exports_to_paraguay.isnull().sum()
```

```
Out[ ]:
```

distributor	0
Rubber	0
Brass	0
Vinyl	0
Granite	0
Stone	0
Brick	0
Aluminum	0
Glass	0
Plexiglass	0
Steel	0
Wood	0
Plastic	0

dtype: int64

- Tipos de datos en el df

```
In [ ]: #tipos de datos en el df
df_exports_to_paraguay.dtypes
```

```
Out[ ]: distributor      int64
Rubber              float64
Brass               float64
Vinyl              float64
Granite            float64
Stone              float64
Brick              float64
Aluminum           float64
Glass             float64
Plexiglass         float64
Steel             float64
Wood              float64
Plastic            float64
dtype: object
```

3.Limpieza distribuidores

```
In [ ]: #!/-----
#!/Limpieza df_distributors_profiles

a = search_duplicates(df_distributors_profiles)
if len(a) != 0:
    df_distributors_profiles = del_columnas(df=df_distributors_profiles, lista_col=
df_distributors_profiles = check_null_values(df_distributors_profiles) #chekear val

search_duplicates_row(df_distributors_profiles) # buscar filas duplicadas y elimin

df_distributors_profiles['id'] = df_distributors_profiles['id'].apply(transformar_v
df_distributors_profiles.loc[36, 'id'] = 1017
for columna in df_distributors_profiles.columns:
    df_distributors_profiles = df_distributors_profiles.drop(df_distributors_profil

Pares de columnas que son exactamente iguales: 0
Pares de filas que son exactamente iguales: 1

In [ ]: display(df_distributors_profiles['id'])
```

0 565
1 1384
2 1183
3 1526
4 29
5 1055
6 1093
7 920
8 1062
10 379
11 1489
12 818
13 707
14 504
15 715
16 619
17 846
18 860
19 241
20 234
21 1463
22 583
23 1104
24 175
25 1710
26 161
27 325
28 1666
29 1553
30 1302
31 364
33 1599
34 308
35 808
36 1017
37 815
38 378
39 1723
40 723
41 1418
42 1235
43 1560
44 1169
45 1679
Name: id, dtype: int64

- Primeros 5 elementos en el df

```
In [ ]: #inspeccion de los datos obtenidos luego de la limpieza  
display(df_distributors_profiles)
```

	id	distributor	distributor activities	years in the construction market
0	565	Abernathy-Hayes	construction materials import/distribution, ir...	15
1	1384	Balistreri LLC	construction materials import/distribution, si...	22
2	1183	Brekke-Stiedemann	construction materials import/distribution	13
3	1526	Collins LLC	construction materials import/distribution	12
4	29	Cummings-Ward	construction materials import/distribution, ir...	23
5	1055	Cummings, Kemmer and Walker	construction materials import/distribution, co...	23
6	1093	Davis, Kiehn and Heller	construction materials import/distribution, ne...	12
7	920	Denesik, Schmidt and Wyman	construction materials import/distribution, zi...	18
8	1062	Feil, MacGyver and Schamberger	construction materials import/distribution, \n...	20
10	379	Gerlach Group	construction materials import/distribution, el...	2
11	1489	Goldner Inc	construction materials import/distribution, in...	18
12	818	Grady, Kessler and Stokes	construction materials import/distribution	16
13	707	Haley-Larson	construction materials import/distribution	14
14	504	Herman, Turcotte and Osinski	construction materials import/distribution, ce...	16
15	715	Keebler, Deckow and Watsica	construction materials import/distribution	8
16	619	Kerluke, Barrows and Murazik	construction materials import/distribution, as...	19
17	846	Kertzmann-Nolan	construction materials import/distribution, zi...	15
18	860	Kling Group	construction materials import/distribution, in...	20
19	241	Koch, Lang and Mertz	construction materials import/distribution, po...	12
20	234	Koepp, Hessel and Runolfsson	construction materials import/distribution	15
21	1463	Kreiger, Leffler and Dibbert	construction materials import/distribution	15
22	583	Kulas-Terry	construction materials import/distribution, co...	19

	id	distributor	distributor activities	years in the construction market
23	1104	Langworth Group	construction materials import/distribution, ne...	22
24	175	Legros Group	construction materials import/distribution, st...	23
25	1710	Lesch, Reinger and Russel	construction materials import/distribution, in...	13
26	161	Maggio-Prohaska	construction materials import/distribution, ne...	20
27	325	Marks Inc	construction materials import/distribution	19
28	1666	Marks, Walker and Streich	construction materials import/distribution, ce...	25
29	1553	Mraz Group	construction materials import/distribution, br...	15
30	1302	O'Reilly, Weber and Larkin	construction materials import/distribution	13
31	364	Okuneva, Simonis and Hintz	construction materials import/distribution	24
33	1599	Orn-Bogan	construction materials import/distribution, co...	17
34	308	Reichel and Sons	construction materials import/distribution	16
35	808	Reilly, Hickle and Thiel	construction materials import/distribution, ne...	18
36	1017	Rosenbaum LLC	construction materials import/distribution	17
37	815	Schamberger and Sons	construction materials import/distribution, co...	8
38	378	Schiller-Bartoletti	construction materials import/distribution	17
39	1723	Sipes and Sons	construction materials import/distribution, co...	23
40	723	Stehr-Mitchell	construction materials import/distribution	9
41	1418	Tremblay Group	construction materials import/distribution, br...	17
42	1235	Tromp-Abshire	construction materials import/distribution, as...	19
43	1560	Volkman-Goyette	construction materials import/distribution, in...	18
44	1169	Williamson, Bailey and McLaughlin	construction materials import/distribution	12
45	1679	Windler-Baumbach	construction materials import/distribution, st...	15

- Tamaño del df

```
In [ ]: #tamaño del df
df_distributors_profiles.shape
```

```
Out[ ]: (44, 4)
```

- Valores nulos en el df

```
In [ ]: #valores Nulos en el df
df_distributors_profiles.isnull().sum()
```

```
Out[ ]: id                                0
distributor                             0
distributor activities                   0
years in the construction market        0
dtype: int64
```

- Tipos de datos en el df

```
In [ ]: #tipos de datos en el df
df_distributors_profiles.dtypes
```

```
Out[ ]: id                                int64
distributor                             object
distributor activities                   object
years in the construction market        int64
dtype: object
```

ANALISIS

Bajas ventas de acero y ladrillos

- Realizamos una comparacion entre Exportación y Ventas.

```
In [ ]: #ordeno el df por distribuidor y Los ordeno de forma asc
df_sales_in_paraguay = df_sales_in_paraguay.sort_values(by='distributor', ascending=True)

#sumo todos los valores despues de 'distributor' y Los ordeno de forma asc
total_sales = df_sales_in_paraguay.iloc[:,1:].sum().sort_values(ascending=False)

#creo un nuevo df para el total de ventas correspondiente a cada rubro
total_sales = pd.DataFrame(total_sales/1000000, columns=['Ventas Totales (Millones)'])

display(total_sales)
```

Ventas Totales (Millones de \$)

Plexiglass	25974.04
Glass	2173.28
Aluminum	2163.83
Plastic	2047.34
Wood	1930.57
Vinyl	1912.59
Granite	1845.29
Brass	1840.66
Rubber	1680.45
Stone	1651.52
Brick	0.15
Steel	0.06

```
In [ ]: #ordeno el df por distribuidor y los ordeno de forma asc
df_exports_to_paraguay = df_exports_to_paraguay.sort_values(by='distribuidor', ascending=True)

#sumo todos los valores despues de 'distribuidor' y los ordeno de forma desc
total_exports = df_exports_to_paraguay.iloc[:,1:].sum().sort_values(ascending=False)

#creo un nuevo df para el total de ventas correspondiente a cada rubro
total_exports = pd.DataFrame(total_exports/1000000, columns=['Exportaciones Totales'])

display(total_exports)
```

Exportaciones Totales (Millones de \$)

Plexiglass	20779.23
Steel	13652.93
Brick	5260.55
Aluminum	1513.17
Stone	1474.57
Rubber	1465.04
Plastic	1456.14
Glass	1429.79
Wood	1409.17
Vinyl	1396.05
Brass	1394.44
Granite	1337.17

```
In [ ]: #combinacion de los df

total_combinados = pd.merge(total_exports.reset_index(), total_sales.reset_index(),
```

```
total_combinados.set_index('index', inplace=True)

display(total_combinados)
```

	Exportaciones Totales (Millones de \$)	Ventas Totales (Millones de \$)
index		
Plexiglass	20779.23	25974.04
Steel	13652.93	0.06
Brick	5260.55	0.15
Aluminum	1513.17	2163.83
Stone	1474.57	1651.52
Rubber	1465.04	1680.45
Plastic	1456.14	2047.34
Glass	1429.79	2173.28
Wood	1409.17	1930.57
Vinyl	1396.05	1912.59
Brass	1394.44	1840.66
Granite	1337.17	1845.29

```
In [ ]: # add ratio/ventas sobre las exportaciones(proporcion)

total_combinados['ratio Ventas/Exportaciones'] = total_combinados['Ventas Totales (Millones de $)'] / total_combinados['Exportaciones Totales (Millones de $)']
display(total_combinados)
```

	Exportaciones Totales (Millones de \$)	Ventas Totales (Millones de \$)	ratio Ventas/Exportaciones
index			
Plexiglass	20779.23	25974.04	1.25
Steel	13652.93	0.06	0.00
Brick	5260.55	0.15	0.00
Aluminum	1513.17	2163.83	1.43
Stone	1474.57	1651.52	1.12
Rubber	1465.04	1680.45	1.15
Plastic	1456.14	2047.34	1.41
Glass	1429.79	2173.28	1.52
Wood	1409.17	1930.57	1.37
Vinyl	1396.05	1912.59	1.37
Brass	1394.44	1840.66	1.32
Granite	1337.17	1845.29	1.38

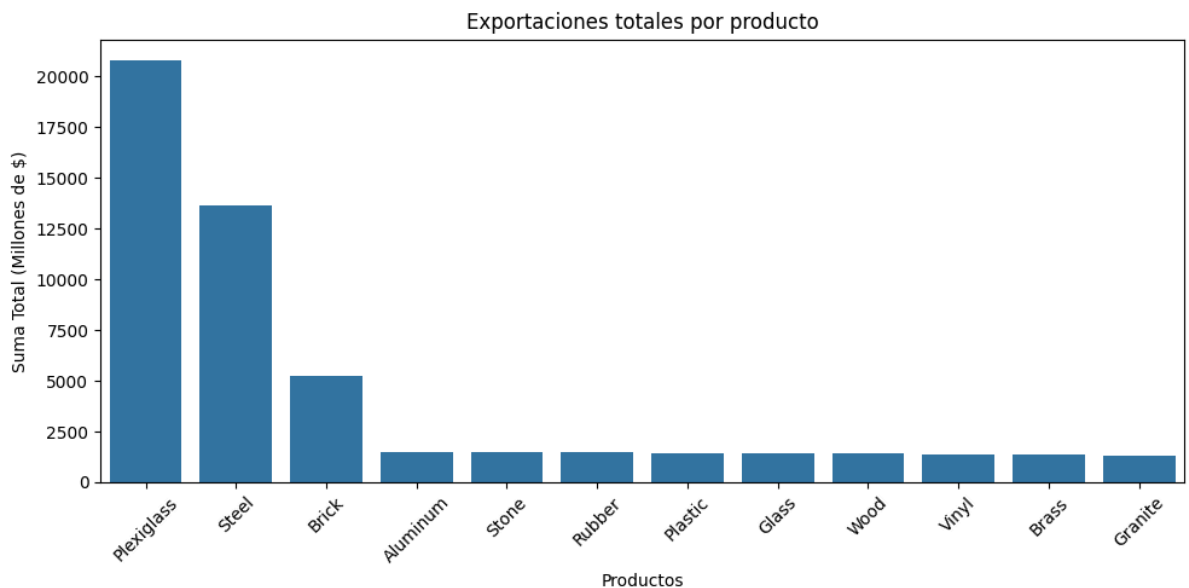
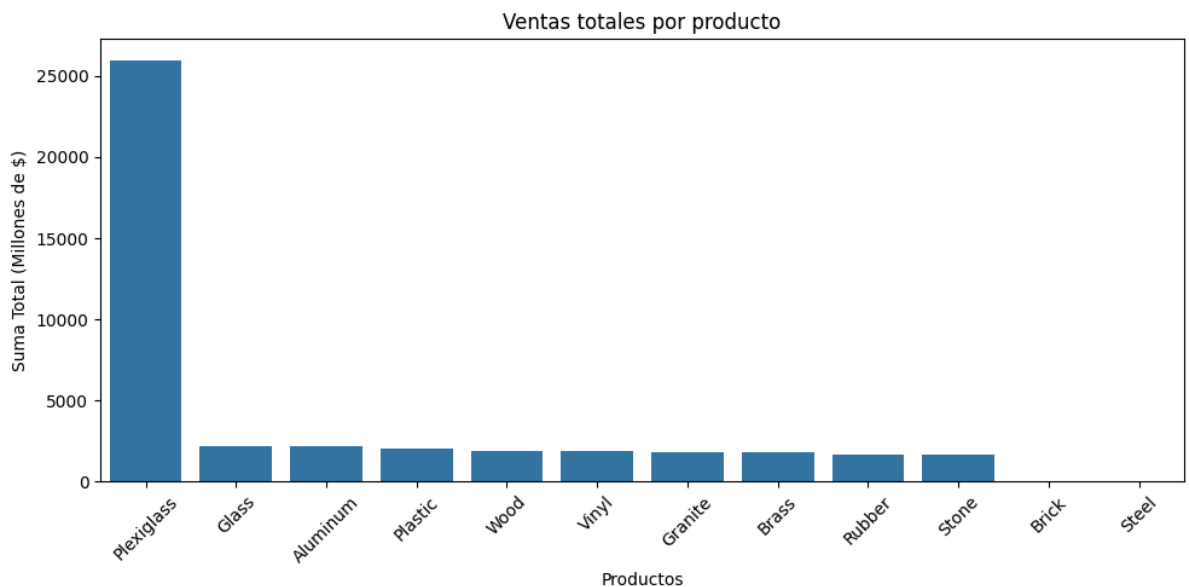
ANALISIS VISUAL

```
In [ ]: # Grafico de barras
fig,ax = plt.subplots(nrows=2, ncols=1 ,figsize=(10, 10))

#Grafico de barras para ventas
sns.barplot(x=total_sales.index, y= total_sales['Ventas Totales (Millones de $)'],
ax[0].set_title('Ventas totales por producto') # titulo del grafico
ax[0].set_xlabel('Productos') # nombre del eje x
ax[0].set_ylabel('Suma Total (Millones de $)') # nombre del eje Y
ax[0].tick_params(axis='x', rotation=45) # ver la etiquetas en eje x

#Grafico de barras para exportaciones
sns.barplot(x=total_exports.index, y= total_exports['Exportaciones Totales (Millones de $)'],
ax[1].set_title('Exportaciones totales por producto') # titulo del grafico
ax[1].set_xlabel('Productos') # nombre del eje x
ax[1].set_ylabel('Suma Total (Millones de $)') # nombre del eje Y
ax[1].tick_params(axis='x', rotation=45) # ver la etiquetas en eje x

plt.tight_layout()
plt.show()
```

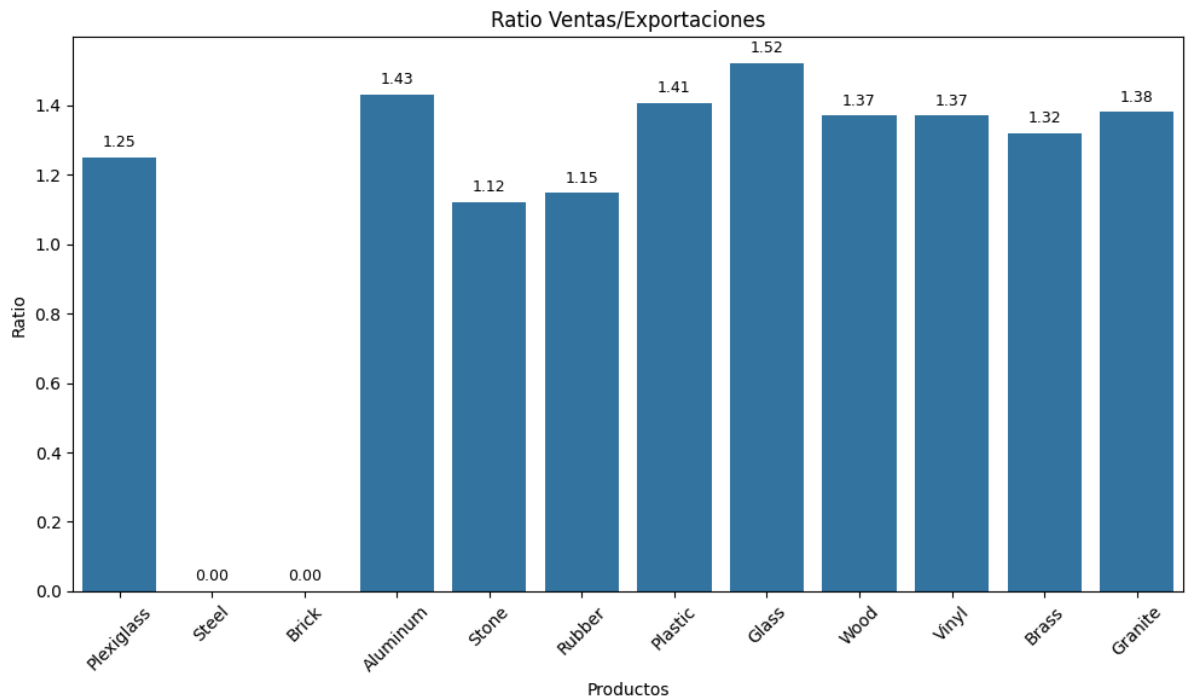


```
In [ ]: #GRAFICO PRORCION DE VENTAS/EXPORTACION

plt.figure(figsize=(10,6))
sns.barplot(x=total_combinados.index, y= 'ratio Ventas/Exportaciones', data=total_c
plt.title('Ratio Ventas/Exportaciones')
plt.xlabel('Productos')
plt.ylabel('Ratio')
plt.xticks(rotation=45)

for index, value in enumerate(total_combinados['ratio Ventas/Exportaciones']):
    plt.text(index, value + 0.02, f'{value:.2f}', ha='center', va='bottom', fontsize=12)

plt.tight_layout()
plt.show()
```

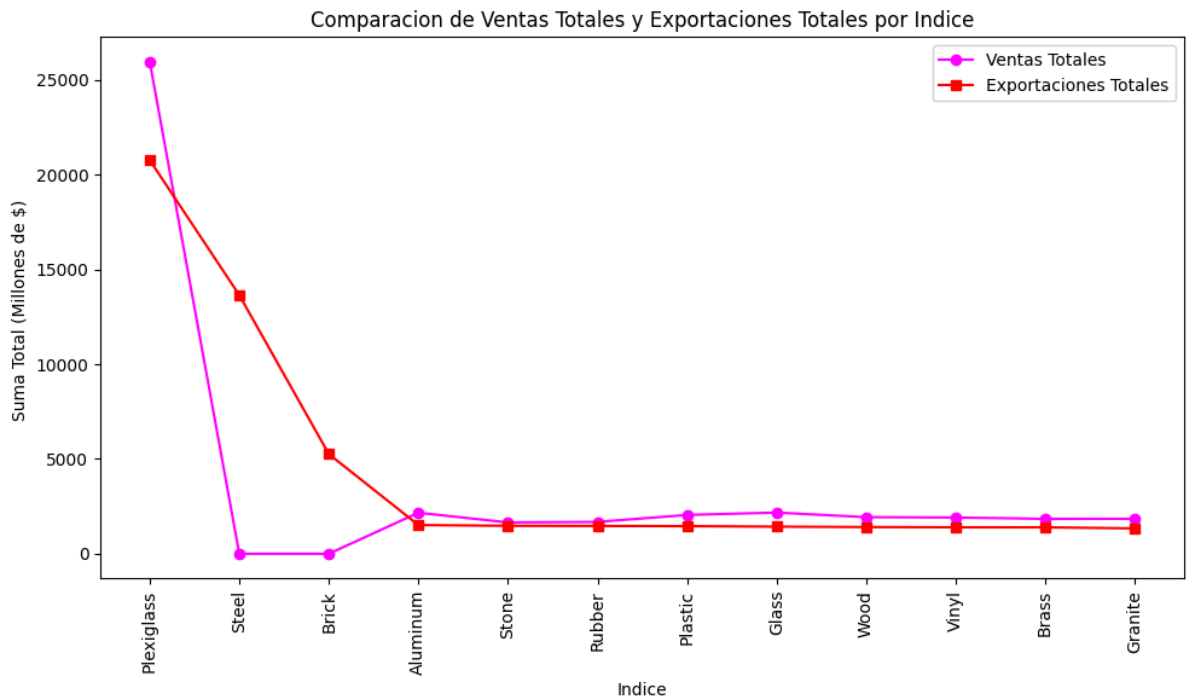


```
In [ ]: #GRAFICO DE BARRAS COMPARACION VENTAS TOTALES Y EXPORTACIONES TOTALES

plt.figure(figsize=(10,6))
plt.plot(total_combinados.index, total_combinados['Ventas Totales (Millones de $)'])
plt.plot(total_combinados.index, total_combinados['Exportaciones Totales (Millones de $)'])

plt.title('Comparacion de Ventas Totales y Exportaciones Totales por Indice')
plt.xlabel('Indice')
plt.ylabel('Suma Total (Millones de $) ')
plt.xticks(rotation=90)
plt.legend()

plt.tight_layout()
plt.show()
```



```
In [ ]: #Grafico de barras acumu

plt.figure(figsize=(10,6))

bar_width = 0.35
indices = range(len(total_combinados))

plt.bar(indices, total_combinados['Ventas Totales (Millones de $)'], width=bar_width)
plt.bar([i + bar_width for i in indices], total_combinados['Exportaciones Totales (Millones de $)'], width=bar_width)

plt.xlabel('Productos')
plt.ylabel('Suma Total (Millones de $)')
plt.title('COMPARACION DE VENTAS TOTALES Y EXPORTACIONES TOTALES POR PRODUCTO')
plt.xticks([i + bar_width / 2 for i in indices], total_combinados.index, rotation=90)
plt.legend()
plt.tight_layout()
plt.show()
```

