

Etapa 1 – Exploración Visual de los Datos

El objetivo de esta etapa es explorar visualmente los datos del proyecto eMarket sin modificar los valores originales. Se busca detectar patrones generales, problemas de calidad de datos y relaciones entre columnas que servirán como base para la siguiente etapa de limpieza y transformación.

Carga de datos

Se utilizaron los siguientes archivos CSV como base de análisis:

- `clientes.csv`
- `facturas.csv`
- `productos.csv`
- `facturadetalle.csv`
- `empleados.csv`

Todos fueron cargados con pandas. A continuación, se presenta una exploración visual y estadística de cada uno.

Etapa 1 – Exploración Visual de los Datos

El objetivo de esta etapa es explorar visualmente los datos del proyecto eMarket sin modificar los valores originales. Se busca detectar patrones generales, problemas de calidad de datos y relaciones entre columnas que servirán como base para la siguiente etapa de limpieza y transformación.

Análisis de Clientes

Se analiza la distribución geográfica de los clientes, así como variables relevantes como el teléfono, región y país.

Carga de datos

Se utilizaron los siguientes archivos CSV como base de análisis:

- `clientes.csv`
- `facturas.csv`
- `productos.csv`

- facturadetalle.csv
- empleados.csv

Todos fueron cargados con pandas. A continuación, se presenta una exploración visual y estadística de cada uno.

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Aquí vamos a descargar todas los csv necesarios

Análisis de Productos

Se explora la distribución de productos, precios unitarios y relación con los descuentos aplicados.

```
In [2]: df=pd.read_csv("clientes.csv")
df2=pd.read_csv("facturas.csv")
df3=pd.read_csv("productos.csv")
df4=pd.read_csv("facturadetalle.csv")
df5=pd.read_csv("empleados.csv")
```

EXPLORACION DE DATOS

Análisis de Facturas y Detalles

Se analiza el comportamiento de las ventas por cliente, país y tiempo.

A partir de aquí, iremos explorar los detalles de cada planilla, descripción, información, contaje, entre otros.

CLIENTES

```
In [3]: df.head()
```

Out[3]:

	ClienteID	Compania_Limpia	Contacto	Titulo	Direccion	Ciudad	Regiones	Co
0	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	
1	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constituci? n 2222	M?xico D.F.	NaN	
2	ANTON	Antonio Moreno Taquerea	Antonio Moreno	Owner	Mataderos 2312	M?xico D.F.	NaN	
3	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NaN	
4	BERGS	Berglunds snabbkep	Christina Berglund	Order Administrator	Berguvsv? gen 8	Lule?	NaN	

In [4]: df.shape

Out[4]: (91, 11)

La planilla tiene 11 columnas con 91 lineas.

In [5]: df.columns

Out[5]: Index(['ClienteID', 'Compania_Limpia', 'Contacto', 'Titulo', 'Direccion', 'Ciudad', 'Regiones', 'CodigoPostal', 'Pais', 'Telefono', 'Fax'], dtype='object')

In [6]: df.dtypes

Out[6]: ClienteID object
Compania_Limpia object
Contacto object
Titulo object
Direccion object
Ciudad object
Regiones object
CodigoPostal object
Pais object
Telefono object
Fax object
dtype: object

In [7]: df.isna().sum()

```
Out[7]: ClienteID      0
        Compania_Limpia  0
        Contacto       0
        Titulo         0
        Direccion      0
        Ciudad         0
        Regiones       60
        CodigoPostal    1
        Pais           0
        Telefono       0
        Fax            22
        dtype: int64
```

```
In [8]: df.describe
```

```
Out[8]: <bound method NDFrame.describe of      ClienteID      Compania_Limpia
Contacto \
0      ALFKI      Alfreds Futterkiste      Maria Anders
1      ANATR      Ana Trujillo Emparedados y helados      Ana Trujillo
2      ANTON      Antonio Moreno Taquerea      Antonio Moreno
3      AROUT      Around the Horn      Thomas Hardy
4      BERGS      Berglunds snabbkep      Christina Berglund
..      ...      ...      ...
86      WARTH      Wartian Herkku      Pirkko Koskitalo
87      WELLI      Wellington Importadora      Paula Parente
88      WHITC      White Clover Markets      Karl Jablonski
89      WILMK      Wilman Kala      Matti Karttunen
90      WOLZA      Wolski Zajazd      Zbyszek Piestrzeniewicz

      Titulo      Direccion      Ciudad \
0      Sales Representative      Obere Str. 57      Berlin
1      Owner      Avda. de la Constituci?n 2222      M?xico D.F.
2      Owner      Mataderos  2312      M?xico D.F.
3      Sales Representative      120 Hanover Sq.      London
4      Order Administrator      Berguvsv?gen  8      Lule?
..      ...      ...      ...
86      Accounting Manager      Torikatu 38      Oulu
87      Sales Manager      Rua do Mercado, 12      Resende
88      Owner      305 - 14th Ave. S. Suite 3B      Seattle
89      Owner/Marketing Assistant      Keskuskatu 45      Helsinki
90      Owner      ul. Filtrowa 68      Warszawa

      Regiones  CodigoPostal      Pais      Telefono      Fax
0      NaN      12209      Germany      030-0074321      030-0076545
1      NaN      05021      Mexico      (5) 555-4729      (5) 555-3745
2      NaN      05023      Mexico      (5) 555-3932      NaN
3      NaN      WA1 1DP      UK      (171) 555-7788      (171) 555-6750
4      NaN      S-958 22      Sweden      0921-12 34 65      0921-12 34 67
..      ...      ...      ...      ...      ...
86      NaN      90110      Finland      981-443655      981-443655
87      SP      08737-363      Brazil      (14) 555-8122      NaN
88      WA      98128      USA      (206) 555-4112      (206) 555-4115
89      NaN      21240      Finland      90-224 8858      90-224 8858
90      NaN      01-012      Poland      (26) 642-7012      (26) 642-7012
```

```
[91 rows x 11 columns]>
```

Las columnas "Regiones" y "Fax" presentan valores faltantes (NaN) y tipo de dato object. Tras revisar su contenido, concluí que no aportan información relevante para los objetivos de este análisis. Por lo tanto, no serán utilizadas ni transformadas.

```
In [9]: df2.head()
```

Out[9]:

	FacturaID	CienteID	EmpleadoID	FechaFactura	FechaRegistro	FechaEnvio	EnvioVia	1
0	10248	VINET	5	7/4/1996	8/1/1996	7/16/1996	3	
1	10249	TOMSP	6	7/5/1996	8/16/1996	7/10/1996	1	
2	10250	HANAR	4	7/8/1996	8/5/1996	7/12/1996	2	
3	10251	VICTE	3	7/8/1996	8/5/1996	7/15/1996	1	
4	10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996	2	

Fue necesario unir las dos tablas: "Clientes" y ""Facturas"

In [10]:

```
df6=df.join(df2.set_index(["ClienteID"]),how='cross')
df6.head()
```

Out[10]:

	ClienteID	Compania_Limpia	Contacto	Titulo	Direccion	Ciudad	Regiones	Coc
0	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	
2	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	
3	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	
4	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NaN	

5 rows × 24 columns

In [11]:

```
df6.columns
```

Out[11]:

```
Index(['ClienteID', 'Compania_Limpia', 'Contacto', 'Titulo', 'Direccion',
      'Ciudad', 'Regiones', 'CodigoPostal', 'Pais', 'Telefono', 'Fax',
      'FacturaID', 'EmpleadoID', 'FechaFactura', 'FechaRegistro',
      'FechaEnvio', 'EnvioVia', 'Transporte', 'NombreEnvio', 'DireccionEnvio',
      'CiudadEnvio', 'RegionEnvio', 'CodigoPostalEnvio', 'PaisEnvio'],
      dtype='object')
```

In [12]:

```
df['ClienteID'].value_counts()
```

```
Out[12]: ClienteID
ALFKI      1
ANATR      1
ANTON      1
AROUT      1
BERGS      1
..
WARTH      1
WELLI      1
WHITC      1
WILMK      1
WOLZA      1
Name: count, Length: 91, dtype: int64
```

```
In [13]: df['Pais'].value_counts()
```

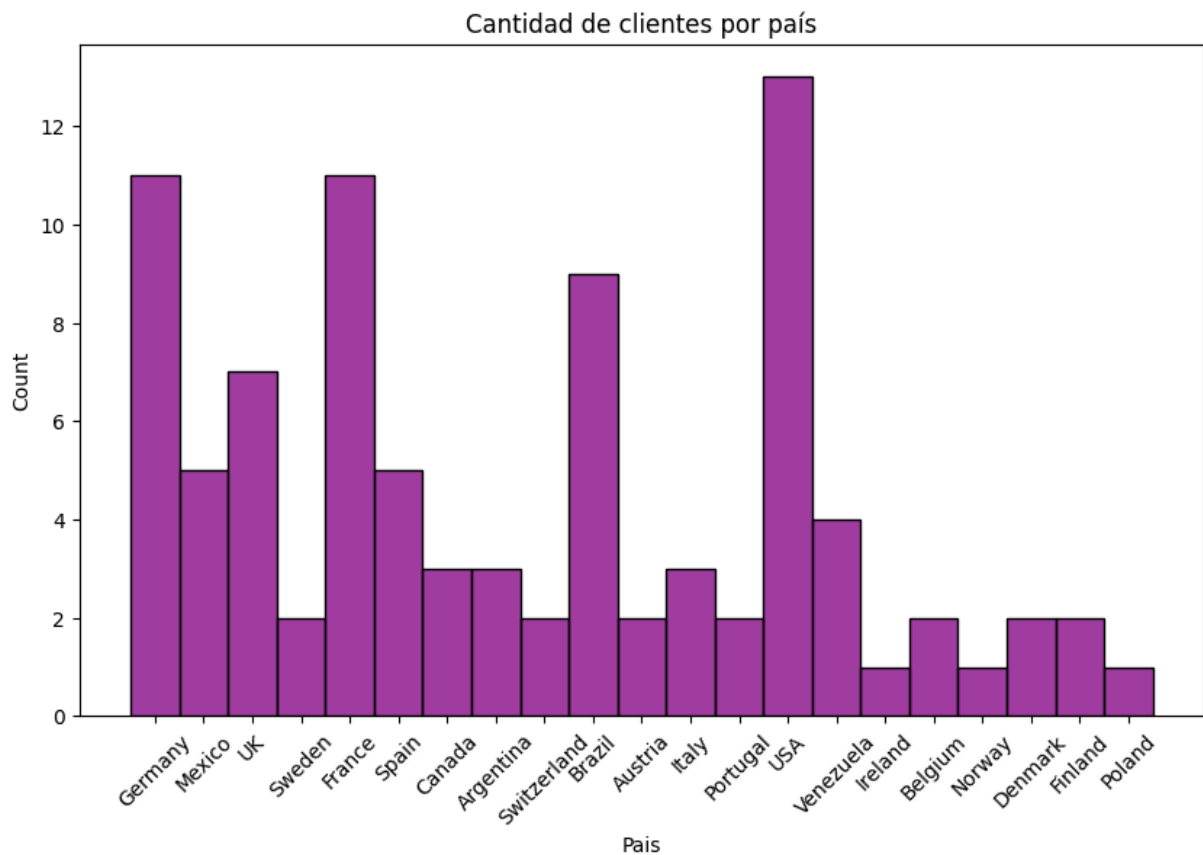
```
Out[13]: Pais
USA          13
France       11
Germany      11
Brazil        9
UK            7
Spain         5
Mexico        5
Venezuela     4
Italy         3
Argentina     3
Canada        3
Sweden        2
Switzerland   2
Portugal      2
Austria       2
Belgium       2
Denmark       2
Finland       2
Ireland       1
Norway        1
Poland        1
Name: count, dtype: int64
```

```
In [14]: df.describe()
```

Out[14]:

	ClienteID	Compania_Limpia	Contacto	Titulo	Direccion	Ciudad	Regiones
count	91	91	91	91	91	91	31
unique	91	91	91	12	91	69	18
top	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	London	SF
freq	1	1	1	17	1	6	6

```
In [58]: plt.figure(figsize=(10,6))
sns.histplot(df["Pais"], kde=False, color='purple')
plt.xticks(rotation=45)
plt.title("Cantidad de clientes por país")
plt.show()
```



```
In [16]: df2["ClienteID"].value_counts()
```

```
Out[16]: ClienteID
SAVEA    31
ERNSH    30
QUICK    28
HUNGO    19
FOLKO    19
..
NORTS     3
FRANR     3
GROSR     2
LAZYK     2
CENTC     1
Name: count, Length: 89, dtype: int64
```

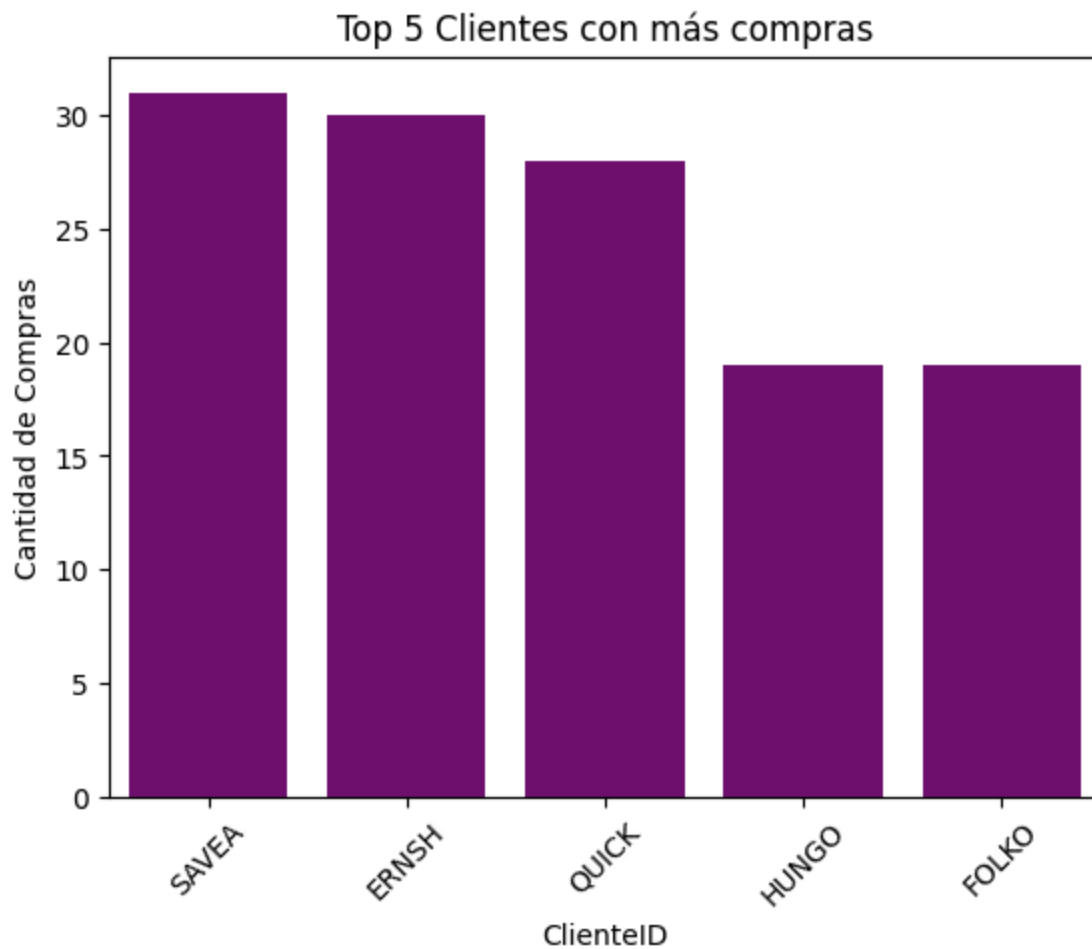
En el grafico abajo, seleccionaré los top 5 compradores y la cantidad de sus compras.

```
In [17]: top5_compradores = df2["ClienteID"].value_counts().head(5)
```

```
In [18]: sns.barplot(x=top5_compradores.index, y=top5_compradores.values, color='purple')
```



```
plt.title("Top 5 Clientes con más compras")  
plt.xlabel("ClienteID")  
plt.ylabel("Cantidad de Compras")  
plt.xticks(rotation=45)  
plt.show()
```



In []:

In []:

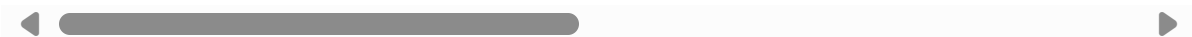
In []:

PRODUCTOS

In [19]: `df2.head()`

Out[19]:

	FacturaID	ClienteID	EmpleadoID	FechaFactura	FechaRegistro	FechaEnvio	EnvioVia	1
0	10248	VINET	5	7/4/1996	8/1/1996	7/16/1996	3	
1	10249	TOMSP	6	7/5/1996	8/16/1996	7/10/1996	1	
2	10250	HANAR	4	7/8/1996	8/5/1996	7/12/1996	2	
3	10251	VICTE	3	7/8/1996	8/5/1996	7/15/1996	1	
4	10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996	2	



In [20]: `df2.shape`

Out[20]: (830, 14)

In [21]: `df2.columns`

Out[21]: Index(['FacturaID', 'ClienteID', 'EmpleadoID', 'FechaFactura', 'FechaRegistro', 'FechaEnvio', 'EnvioVia', 'Transporte', 'NombreEnvio', 'DireccionEnvio', 'CiudadEnvio', 'RegionEnvio', 'CodigoPostalEnvio', 'PaisEnvio'], dtype='object')

In [22]: `df2.dtypes`

Out[22]: FacturaID int64
 ClienteID object
 EmpleadoID int64
 FechaFactura object
 FechaRegistro object
 FechaEnvio object
 EnvioVia int64
 Transporte float64
 NombreEnvio object
 DireccionEnvio object
 CiudadEnvio object
 RegionEnvio object
 CodigoPostalEnvio object
 PaisEnvio object
 dtype: object

In [23]: `df2.isna().sum()`

```
Out[23]: FacturaID      0
         ClienteID     0
         EmpleadoID    0
         FechaFactura   0
         FechaRegistro  0
         FechaEnvio     0
         EnvioVia       0
         Transporte     0
         NombreEnvio    0
         DireccionEnvio 0
         CiudadEnvio    0
         RegionEnvio    507
        CodigoPostalEnvio 19
         PaisEnvio      0
         dtype: int64
```

```
In [24]: df2.describe
```

```
Out[24]: <bound method NDFrame.describe of
a FechaRegistro \
0      10248      VINET      5      7/4/1996      8/1/1996
1      10249      TOMSP      6      7/5/1996      8/16/1996
2      10250      HANAR      4      7/8/1996      8/5/1996
3      10251      VICTE      3      7/8/1996      8/5/1996
4      10252      SUPRD      4      7/9/1996      8/6/1996
..      ...      ...      ...      ...      ...
825     11073      PERIC      2      5/5/1998      6/2/1998
826     11074      SIMOB      7      5/6/1998      6/3/1998
827     11075      RICSU      8      5/6/1998      6/3/1998
828     11076      BONAP      4      5/6/1998      6/3/1998
829     11077      RATTC      1      5/6/1998      6/3/1998
```

```

      FechaEnvio  EnvioVia  Transporte      NombreEnvio \
0      7/16/1996      3      32.38      Vins et alcools Chevalier
1      7/10/1996      1      11.61      Toms Spezialiteten
2      7/12/1996      2      65.83      Hanari Carnes
3      7/15/1996      1      41.34      Victuailles en stock
4      7/11/1996      2      51.30      Supremes delices
..      ...      ...      ...      ...
825  0000-00-00 00:00:00      2      24.95      Pericles Comidas clesicas
826  0000-00-00 00:00:00      2      18.44      Simons bistro
827  0000-00-00 00:00:00      2      6.19      Richter Supermarkt
828  0000-00-00 00:00:00      2      38.28      Bon app'
829  0000-00-00 00:00:00      2      8.53      Rattlesnake Canyon Grocery
```

```

      DireccionEnvio      CiudadEnvio  RegionEnvio CodigoPostalEnvio \
0      59 rue de l'Abbaye      Reims      NaN      51100
1      Luisenstr. 48      Menster      NaN      44087
2      Rua do Paeo, 67      Rio de Janeiro      RJ      05454-876
3      2, rue du Commerce      Lyon      NaN      69004
4      Boulevard Tirou, 255      Charleroi      NaN      B-6000
..      ...      ...      ...      ...
825  Calle Dr. Jorge Cash 321      Mexico D.F.      NaN      5033
826      Vinbeltet 34      Kobenhavn      NaN      1734
827      Starenweg 5      Geneve      NaN      1204
828      12, rue des Bouchers      Marseille      NaN      13008
829      2817 Milton Dr.      Albuquerque      NM      87110
```

```

      PaisEnvio
0      France
1      Germany
2      Brazil
3      France
4      Belgium
..      ...
825      Mexico
826      Denmark
827      Switzerland
828      France
829      USA
```

```
[830 rows x 14 columns]>
```

```
In [25]: df4.head()
```

```
Out[25]:
```

	FacturaID	ProductoID	PrecioUnitario	Cantidad	Descuento
0	10248	11	14.0	12	0.0
1	10248	42	9.8	10	0.0
2	10248	72	34.8	5	0.0
3	10249	14	18.6	9	0.0
4	10249	51	42.4	40	0.0

```
In [26]: df4.shape
```

```
Out[26]: (2155, 5)
```

```
In [27]: df4.columns
```

```
Out[27]: Index(['FacturaID', 'ProductoID', 'PrecioUnitario', 'Cantidad', 'Descuento'], dtype='object')
```

```
In [28]: df4.dtypes
```

```
Out[28]: FacturaID      int64
ProductoID    int64
PrecioUnitario float64
Cantidad      int64
Descuento     float64
dtype: object
```

```
In [29]: df4.isna().sum()
```

```
Out[29]: FacturaID      0
ProductoID    0
PrecioUnitario 0
Cantidad      0
Descuento     0
dtype: int64
```

```
In [30]: df4.describe
```

```
Out[30]: <bound method NDFrame.describe of
          tidad Descuento
0          10248          11          14.00          12          0.00
1          10248          42           9.80          10          0.00
2          10248          72          34.80           5          0.00
3          10249          14          18.60           9          0.00
4          10249          51          42.40          40          0.00
...          ...          ...          ...          ...          ...
2150         11077           64          33.25           2          0.03
2151         11077           66          17.00           1          0.00
2152         11077           73          15.00           2          0.01
2153         11077           75           7.75           4          0.00
2154         11077           77          13.00           2          0.00

[2155 rows x 5 columns]>
```

```
In [31]: df3.head()
```

Out[31]:

	ProductoID	ProductoNombre	ProveedorID	CategorialID	CantidadPorUnidad	PrecioUnitario
0	1	Chai	1	1	10 boxes x 20 bags	
1	2	Chang	1	1	24 - 12 oz bottles	
2	3	Aniseed Syrup	1	2	12 - 550 ml bottles	
3	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	
4	5	Chef Anton's Gumbo Mix	2	2	36 boxes	

```
In [32]: df3.columns
```

```
Out[32]: Index(['ProductoID', 'ProductoNombre', 'ProveedorID', 'CategorialID',
               'CantidadPorUnidad', 'PrecioUnitario', 'UnidadesStock',
               'UnidadesPedidas', 'NivelReorden', 'Discontinuado'],
              dtype='object')
```

```
In [33]: df3.dtypes
```

```
Out[33]: ProductoID          int64
ProductoNombre        object
ProveedorID           int64
CategorialID          int64
CantidadPorUnidad     object
PrecioUnitario        float64
UnidadesStock         int64
UnidadesPedidas       int64
NivelReorden          int64
Discontinuado         int64
dtype: object
```

```
In [34]: df3.shape
```

Out[34]: (77, 10)

In [35]: `df3.isna().sum()`

Out[35]:

ProductoID	0
ProductoNombre	0
ProveedorID	0
CategoriaID	0
CantidadPorUnidad	0
PrecioUnitario	0
UnidadesStock	0
UnidadesPedidas	0
NivelReorden	0
Discontinuado	0
dtype:	int64

In [36]: `df3.describe`

```
Out[36]: <bound method NDFrame.describe of
```

	ProveedorID	CategoriaID	ProductoID	ProductoNombre
0	1		Chai	1
1	2		Chang	1
2	3		Aniseed Syrup	2
3	4		Chef Anton's Cajun Seasoning	2
4	5		Chef Anton's Gumbo Mix	2
..
72	73		Red Kaviar	8
73	74		Longlife Tofu	7
74	75		Rhenbreu Klosterbier	1
75	76		Lakkalikeeri	1
76	77		Original Frankfurter grene Soee	2

	CantidadPorUnidad	PrecioUnitario	UnidadesStock	UnidadesPedidas
0	10 boxes x 20 bags	18.00	39	0
1	24 - 12 oz bottles	19.00	17	40
2	12 - 550 ml bottles	10.00	13	70
3	48 - 6 oz jars	22.00	53	0
4	36 boxes	21.35	0	0
..
72	24 - 150 g jars	15.00	101	0
73	5 kg pkg.	10.00	4	20
74	24 - 0.5 l bottles	7.75	125	0
75	500 ml	18.00	57	0
76	12 boxes	13.00	32	0

	NivelReorden	Discontinuado
0	10	0
1	25	0
2	25	0
3	0	0
4	0	0
..
72	5	0
73	5	0
74	25	0
75	20	0
76	15	0

```
[77 rows x 10 columns]>
```

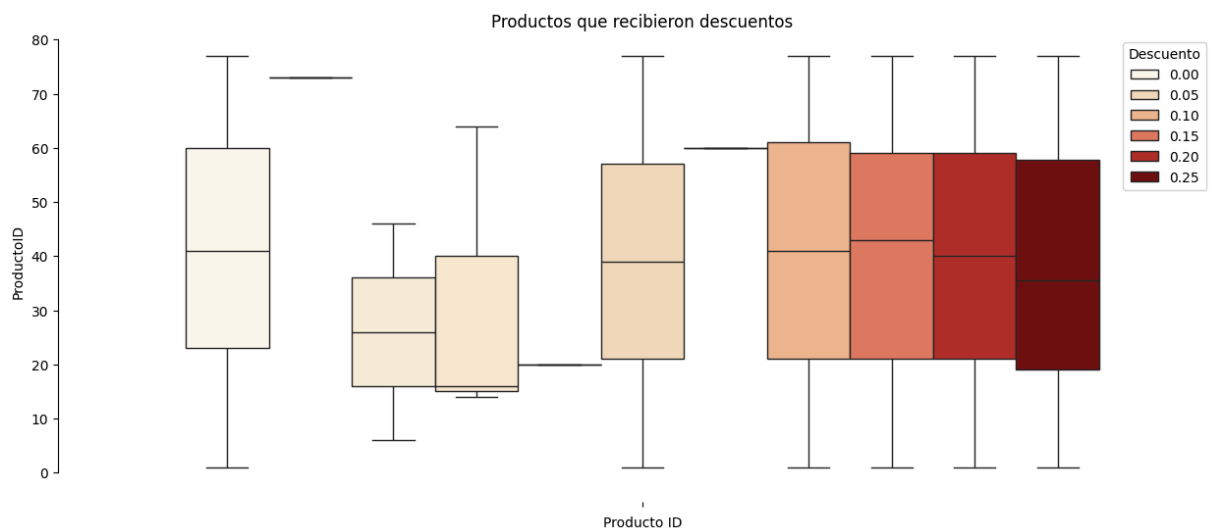
```
In [37]: df4["Descuento"].value_counts()
```



```
Out[37]: Descuento
0.00    1317
0.05     185
0.10     173
0.20     161
0.15     157
0.25     154
0.03        3
0.02        2
0.04        1
0.06        1
0.01        1
Name: count, dtype: int64
```

```
In [59]: plt.figure(figsize=(15,6))
sns.boxplot(data=df4, hue="Descuento", y="ProductoID", palette="OrRd")
sns.despine(offset=10, trim=True)
plt.xlabel("Producto ID")
plt.title("Productos que recibieron descuentos")
plt.xticks(rotation=45)
```

```
Out[59]: (array([0]), [Text(0, 0, '')])
```

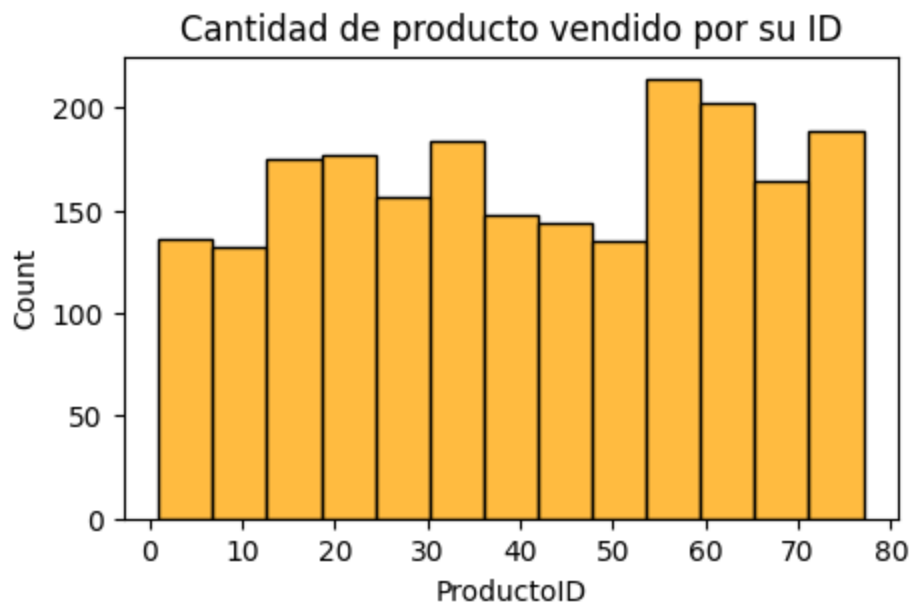


Un boxplot que para que veamos cuales productos recibieron descuentos con sus outliers

```
In [39]: df4["ProductoID"].value_counts()
```

```
Out[39]: ProductoID
59      54
31      51
60      51
24      51
56      50
..
66       8
37       6
15       6
48       6
9        5
Name: count, Length: 77, dtype: int64
```

```
In [60]: plt.figure(figsize=(5,3))
sns.histplot(df4["ProductoID"], kde=False, color='orange')
plt.title("Cantidad de producto vendido por su ID")
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

Análisis de Empleados

Se evalúan variables como fecha de nacimiento, fecha de contratación, edad y antigüedad del personal.

EMPLEADOS

```
In [41]: df5.head()
```

Out[41]:

	EmpleadoID	Apellido	Nombre	Titulo	TituloCortesia	FechaNacimiento	Fecha
0	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08	
1	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19	
2	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30	
3	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19	
4	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04	

In [42]: df5.shape

Out[42]: (9, 18)

In [43]: df5.columns

Out[43]: Index(['EmpleadoID', 'Apellido', 'Nombre', 'Titulo', 'TituloCortesia', 'FechaNacimiento', 'FechaContratacion', 'Direccion', 'Ciudad', 'Regiones', 'CodigoPostal', 'Pais', 'Telefono', 'Extension', 'Foto', 'Notas', 'Jefe', 'RutaFoto'], dtype='object')

In [44]: df5.dtypes

```
Out[44]: EmpleadoID      int64
         Apellido       object
         Nombre         object
         Titulo          object
         TituloCortesia  object
         FechaNacimiento object
         FechaContratacion object
         Direccion      object
         Ciudad         object
         Regiones       object
        CodigoPostal    object
         Pais           object
         Telefono       object
         Extension      int64
         Foto           float64
         Notas          object
         Jefe           float64
         RutaFoto       object
         dtype: object
```

```
In [45]: df5.isna().sum()
```

```
Out[45]: EmpleadoID      0
         Apellido       0
         Nombre         0
         Titulo          0
         TituloCortesia  0
         FechaNacimiento 0
         FechaContratacion 0
         Direccion      0
         Ciudad         0
         Regiones       4
        CodigoPostal    0
         Pais           0
         Telefono       0
         Extension      0
         Foto           9
         Notas          0
         Jefe           1
         RutaFoto       0
         dtype: int64
```

```
In [46]: df4.describe
```

```
Out[46]: <bound method NDFrame.describe of
          FacturaID ProductoID PrecioUnitario Can
          tidad Descuento
0          10248          11          14.00          12          0.00
1          10248          42           9.80          10          0.00
2          10248          72          34.80           5          0.00
3          10249          14          18.60           9          0.00
4          10249          51          42.40          40          0.00
...          ...          ...          ...          ...          ...
2150         11077          64          33.25           2          0.03
2151         11077          66          17.00           1          0.00
2152         11077          73          15.00           2          0.01
2153         11077          75           7.75           4          0.00
2154         11077          77          13.00           2          0.00
```

[2155 rows x 5 columns]>

Aqui quiero tener el tiempo de antigüedad de cada empleado

```
In [47]: from datetime import datetime
```

```
In [48]: df5["FechaNacimiento"] = pd.to_datetime(df5["FechaNacimiento"])
df5["FechaContratacion"] = pd.to_datetime(df5["FechaContratacion"])

hoy = datetime.now()

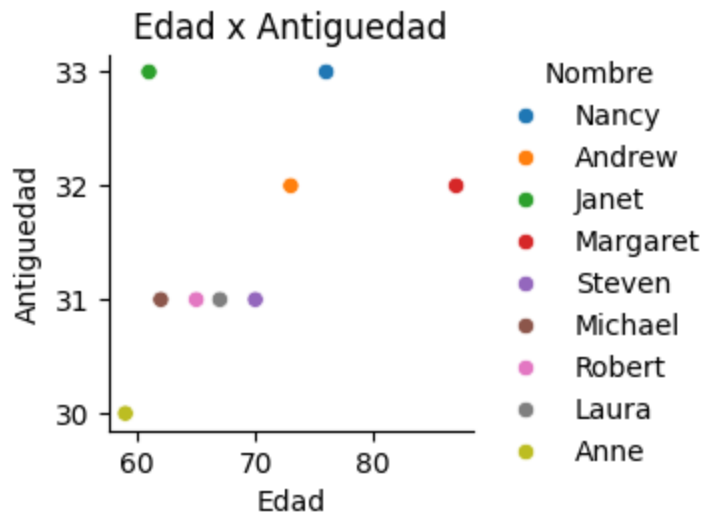
df5["Edad"] = hoy.year - df5["FechaNacimiento"].dt.year
df5["Edad"] -= ( (hoy.month < df5["FechaNacimiento"].dt.month) |
                ((hoy.month == df5["FechaNacimiento"].dt.month) & (hoy.day < df5["
FechaNacimiento"].dt.day))

df5["Antigüedad"] = hoy.year - df5["FechaContratacion"].dt.year
df5["Antigüedad"] -= ( (hoy.month < df5["FechaContratacion"].dt.month) |
                      ((hoy.month == df5["FechaContratacion"].dt.month) & (hoy.day < df5["
FechaContratacion"].dt.day))

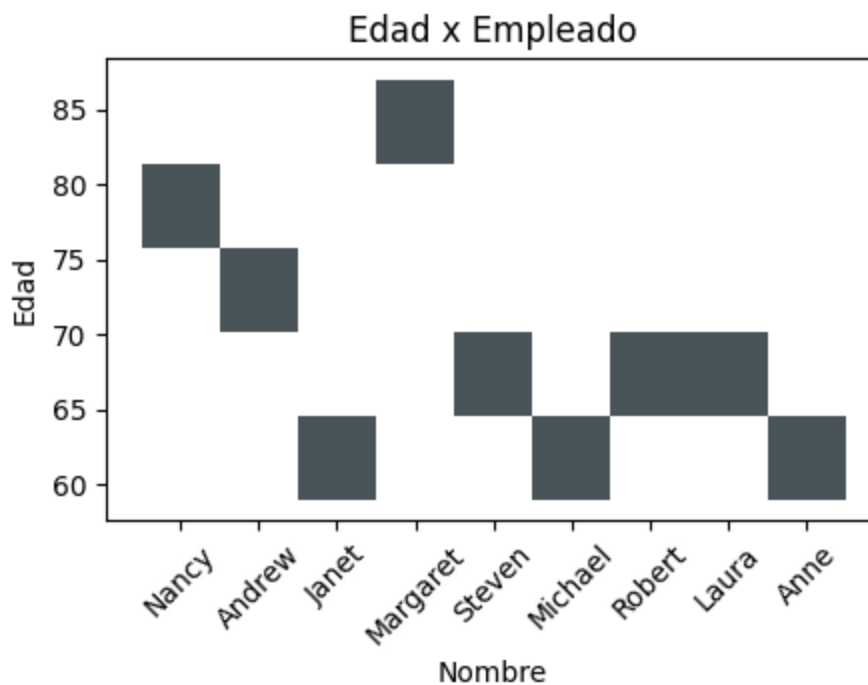
print(df5[["Nombre", "FechaNacimiento", "Edad", "FechaContratacion", "Antigüedad"]])
```

	Nombre	FechaNacimiento	Edad	FechaContratacion	Antigüedad
0	Nancy	1948-12-08	76	1992-05-01	33
1	Andrew	1952-02-19	73	1992-08-14	32
2	Janet	1963-08-30	61	1992-04-01	33
3	Margaret	1937-09-19	87	1993-05-03	32
4	Steven	1955-03-04	70	1993-10-17	31
5	Michael	1963-07-02	62	1993-10-17	31
6	Robert	1960-05-29	65	1994-01-02	31
7	Laura	1958-01-09	67	1994-03-05	31
8	Anne	1966-01-27	59	1994-11-15	30

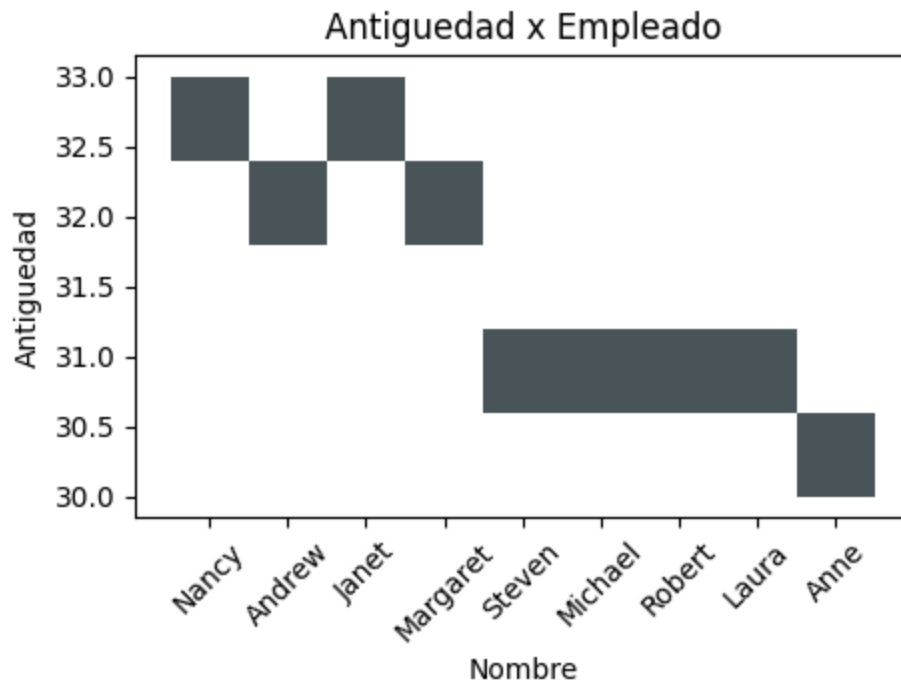
```
In [61]: sns.pairplot(df5, x_vars="Edad", y_vars="Antigüedad", hue="Nombre")
plt.title("Edad x Antigüedad")
plt.show()
```



```
In [62]: plt.figure(figsize=(5,3))
sns.histplot(df5, x="Nombre", y="Edad", color="lightblue")
plt.xticks(rotation=45)
plt.title("Edad x Empleado")
plt.show()
```



```
In [63]: plt.figure(figsize=(5,3))
sns.histplot(df5, x="Nombre", y="Antigüedad", color="lightblue")
plt.xticks(rotation=45)
plt.title("Antigüedad x Empleado")
plt.show()
```

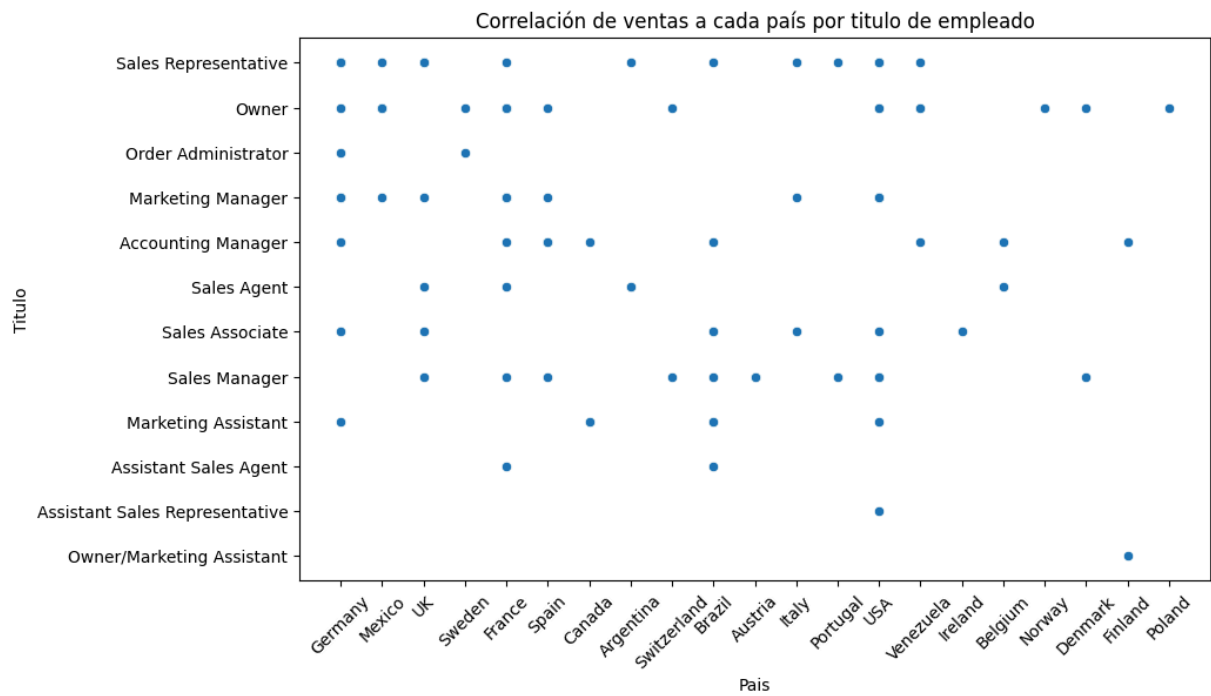


```
In [52]: df5.value_counts("EmpleadoID").sum()
```

```
Out[52]: np.int64(9)
```

En el grafico de abajo, nos muestra la ventas por titulo de cada empleado y a cual país se vendió algo.

```
In [64]: plt.figure(figsize=(10,6))
sns.scatterplot(x=df.Pais,y=df.Titulo)
plt.title("Correlación de ventas a cada país por titulo de empleado")
plt.xticks(rotation=45)
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

ENTREGAS

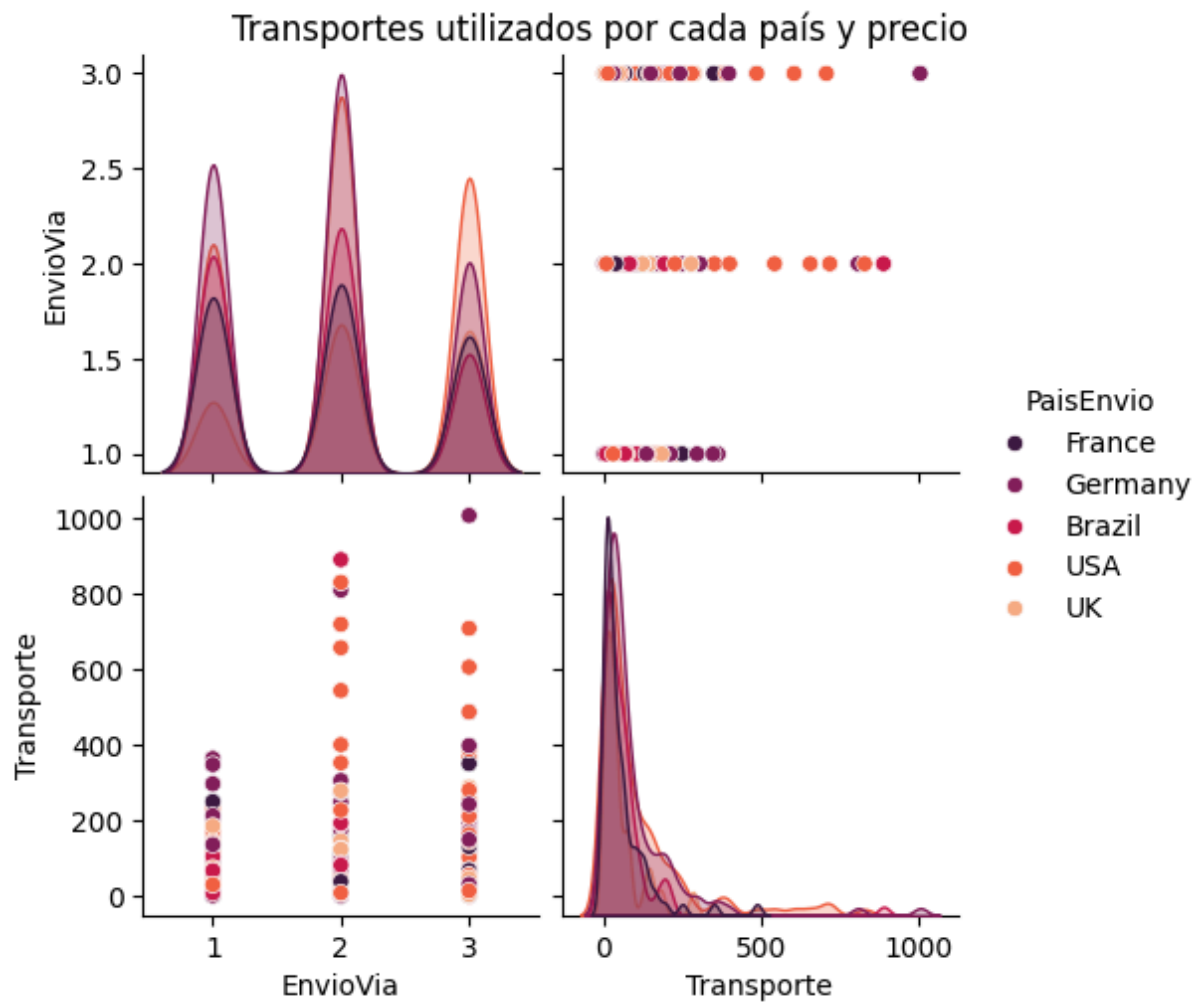
Análisis de Entregas

Se revisan los modos de envío, tiempos y valores de transporte, así como su relación con el país de destino.

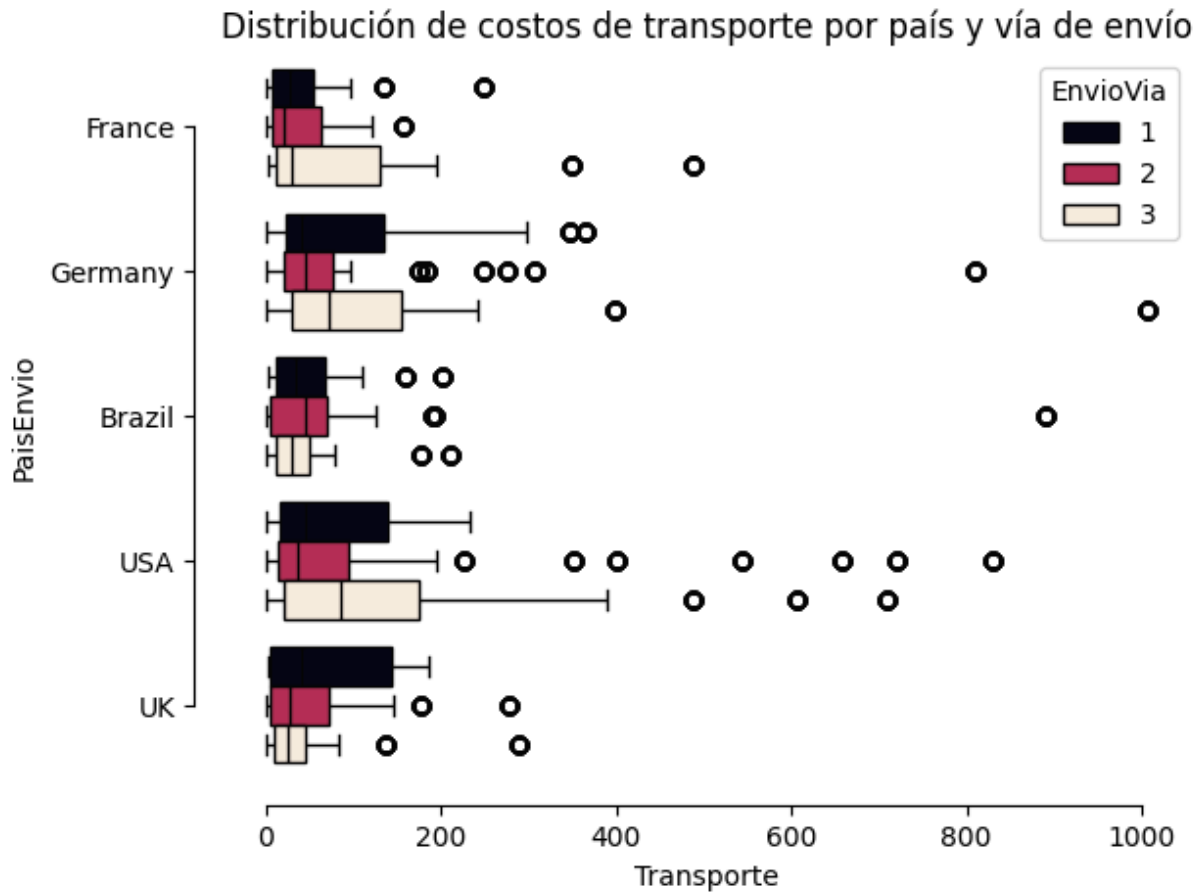
En ese grafico de abajo, quise analizar los tipos de transportes de envio a cada país y el monto pagado para enviar a cada país.

```
In [65]: top5_paises = df6["PaisEnvio"].value_counts().head(5).index
df_top5 = df6[df6["PaisEnvio"].isin(top5_paises)]

pair = sns.pairplot(df_top5, vars=['EnvioVia', 'Transporte'], hue='PaisEnvio', pale
pair.fig.suptitle("Transportes utilizados por cada país y precio", y=1.02)
plt.show()
```

```
In [66]: sns.boxplot(data=df_top5, hue="EnvioVia", x="Transporte", y="PaisEnvio", palette="r
plt.title("Distribución de costos de transporte por país y vía de envío")
sns.despine(offset=10, trim=True)
```



Aquí podemos verificar el tipo de transporte más utilizado en los pedidos para cada país (de los top 5)

Análisis de Facturas y Detalle

```
In [ ]: total_facturas = len(df2)
facturas_sin_fecha = df2["FechaFactura"].isna().sum()
porcentaje_sin_fecha = round(facturas_sin_fecha / total_facturas * 100, 2)
print(f"Facturas totales: {total_facturas}")
print(f"Facturas sin fecha registrada: {facturas_sin_fecha} ({porcentaje_sin_fecha})")
```

Facturas totales: 830

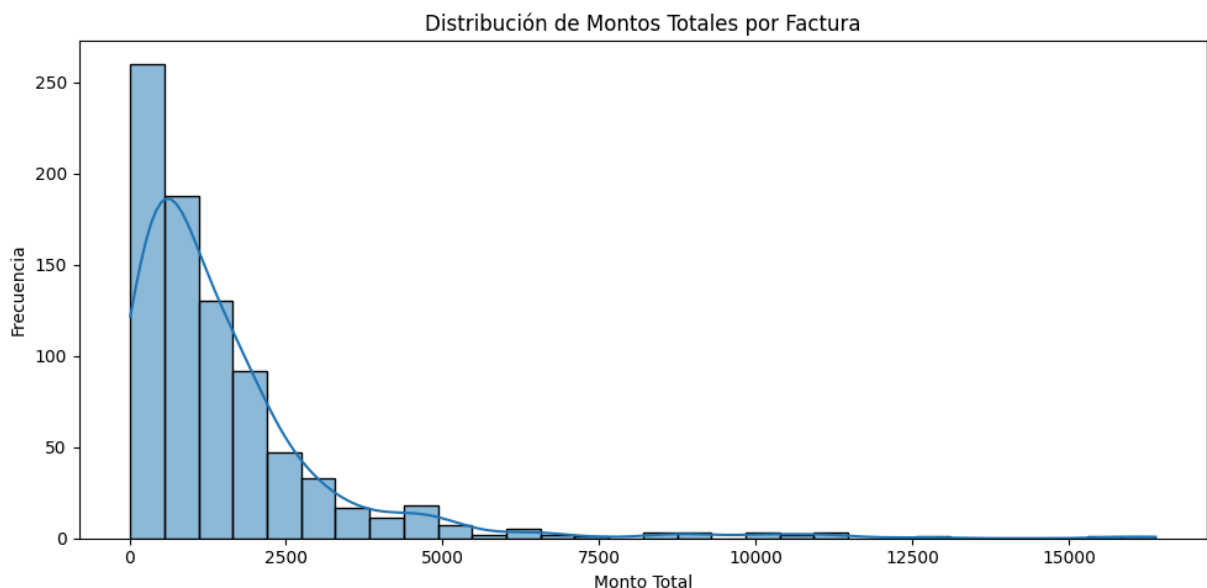
Facturas sin fecha registrada: 0 (0.0%)

```
In [69]: df4["MontoTotal"] = df4["Cantidad"] * df4["PrecioUnitario"] * (1 - df4["Descuento"])
df_total = df4.groupby("FacturaID")["MontoTotal"].sum().reset_index()
df_total.head()
```

Out[69]:

	FacturaID	MontoTotal
0	10248	440.000000
1	10249	1863.400000
2	10250	1552.599983
3	10251	654.060000
4	10252	3597.900000

```
In [70]: plt.figure(figsize=(10, 5))
sns.histplot(df_total["MontoTotal"], bins=30, kde=True)
plt.title("Distribución de Montos Totales por Factura")
plt.xlabel("Monto Total")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.show()
```



In []:

PRINCIPALES HALLAZGOS ACTUALIZADOS

CLIENTES

- United States representa **18.1%** del total de clientes.
- País con mayor número de clientes: **United States** (13 clientes).

PRODUCTOS

- Producto más vendido: **Camembert Pierrot** (1577 unidades).
- Segundo más vendido: **Raclette Courdavault** (1496 unidades).

- Tercero más vendido: **Gorgonzola Telino** (1397 unidades).
- Cuarto más vendido: **Gnocchi di nonna Alice** (1263 unidades).
- Quinto más vendido: **Pavlova** (1158 unidades).

FACTURAS Y DETALLE

- Total de facturas: 830
- 0% de facturas sin fecha registrada
- Monto promedio por factura \approx 1450 unidades
- No hay facturas sin productos asociados
- Se identifican descuentos aplicados en ventas