Análisis y Modelado con Aprendizaje No Supervisado

May 3, 2025

1 Actividad: Análisis y Modelado con Aprendizaje No Supervisado

1.1 Jaime Reina

1.2 Diccionario de Datos

[3]:
```
import pandas as pd
from tabulate import tabulate

# Diccionario de las variables
data_variables = {
    "Id": "Identificador único de la propiedad",
    "MSSubClass": "Clase de construcción",
    "MSZoning": "Clasificación de la zona (residencial, comercial, etc.)",
    "LotFrontage": "Longitud de frente de calle (en pies)",
    "LotArea": "Área del lote (en pies cuadrados)",
    "Street": "Tipo de calle (Pavimentada o Grava)",
    "Alley": "Tipo de callejón (si existe)",
    "LotShape": "Forma del lote",
    "LandContour": "Contorno del terreno",
    "Utilities": "Servicios públicos disponibles",
    "LotConfig": "Configuración del lote",
    "LandSlope": "Inclinación del terreno",
    "Neighborhood": "Barrio",
    "Condition1": "Proximidad a carreteras principales o vías férreas",
    "Condition2": "Otra proximidad a carreteras principales o vías férreas (si␣
 ↪aplica)",
    "BldgType": "Tipo de edificio",
    "HouseStyle": "Estilo de la casa",
    "OverallQual": "Calidad general del material y acabado",
    "OverallCond": "Condición general de la casa",
    "YearBuilt": "Año de construcción",
    "YearRemodAdd": "Año de remodelación o adición",
    "RoofStyle": "Estilo del techo",
    "RoofMatl": "Material del techo",
    "Exterior1st": "Material exterior de recubrimiento",
    "Exterior2nd": "Otro material exterior de recubrimiento (si aplica)",
```

```
    "MasVnrType": "Tipo de recubrimiento de mampostería",
    "MasVnrArea": "Área de recubrimiento de mampostería (en pies cuadrados)",
    "ExterQual": "Calidad del material exterior",
    "ExterCond": "Condición del material exterior",
    "Foundation": "Tipo de cimientos",
    "BsmtQual": "Altura de los cimientos",
    "BsmtCond": "Condición de los cimientos",
    "BsmtExposure": "Exposición del sótano",
    "BsmtFinType1": "Calidad del área terminada del sótano",
    "BsmtFinSF1": "Área terminada del sótano (en pies cuadrados)",
    "BsmtFinType2": "Calidad adicional del área terminada del sótano",
    "BsmtFinSF2": "Área adicional terminada del sótano (en pies cuadrados)",
    "BsmtUnfSF": "Área sin terminar del sótano (en pies cuadrados)",
    "TotalBsmtSF": "Área total del sótano (en pies cuadrados)",
    "Heating": "Tipo de sistema de calefacción",
    "HeatingQC": "Calidad del sistema de calefacción",
    "CentralAir": "Aire acondicionado central (Sí o No)",
    "Electrical": "Sistema eléctrico",
    "1stFlrSF": "Área del primer piso (en pies cuadrados)",
    "2ndFlrSF": "Área del segundo piso (en pies cuadrados)",
    "LowQualFinSF": "Área de baja calidad terminada (en pies cuadrados)",
    "GrLivArea": "Área de vida sobre el nivel del suelo (en pies cuadrados)",
    "BsmtFullBath": "Número de baños completos en el sótano",
    "BsmtHalfBath": "Número de medios baños en el sótano",
    "FullBath": "Número de baños completos sobre el nivel del suelo",
    "HalfBath": "Número de medios baños sobre el nivel del suelo",
    "BedroomAbvGr": "Número de habitaciones sobre el nivel del suelo",
    "KitchenAbvGr": "Número de cocinas sobre el nivel del suelo",
    "KitchenQual": "Calidad de las cocinas",
    "TotRmsAbvGrd": "Número total de habitaciones sobre el nivel del suelo␣
↪(excluyendo baños)",
    "Functional": "Funcionalidad del hogar",
    "Fireplaces": "Número de chimeneas",
    "FireplaceQu": "Calidad de las chimeneas",
    "GarageType": "Ubicación del garaje",
    "GarageYrBlt": "Año de construcción del garaje",
    "GarageFinish": "Acabado interior del garaje",
    "GarageCars": "Capacidad del garaje en términos de autos",
    "GarageArea": "Área del garaje (en pies cuadrados)",
    "GarageQual": "Calidad del garaje",
    "GarageCond": "Condición del garaje",
    "PavedDrive": "Calle pavimentada (Sí o No)",
    "WoodDeckSF": "Área de la terraza de madera (en pies cuadrados)",
    "OpenPorchSF": "Área del porche abierto (en pies cuadrados)",
    "EnclosedPorch": "Área del porche cerrado (en pies cuadrados)",
    "3SsnPorch": "Área del porche de tres estaciones (en pies cuadrados)",
    "ScreenPorch": "Área del porche con mosquitero (en pies cuadrados)",
```

```
    "PoolArea": "Área de la piscina (en pies cuadrados)",
    "PoolQC": "Calidad de la piscina",
    "Fence": "Calidad de la cerca",
    "MiscFeature": "Características misceláneas no cubiertas en otras␣
 ↪categorías",
    "MiscVal": "Valor de las características misceláneas",
    "MoSold": "Mes de venta",
    "YrSold": "Año de venta",
    "SaleType": "Tipo de venta",
    "SaleCondition": "Condición de la venta"
}

# Convertir a un DataFrame
variables_df = pd.DataFrame(list(data_variables.items()), columns=["Variable",␣
 ↪"Descripción"])

# Mostrar la tabla formateada
print(tabulate(variables_df, headers="keys", tablefmt="pretty"))
```

```
+----+---------------+-------------------------------------------------------------
---------------+
|    |   Variable    |                              Descripción
|
+----+---------------+-------------------------------------------------------------
---------------+
| 0  |      Id       |                   Identificador único de la propiedad
|
| 1  |  MSSubClass   |                         Clase de construcción
|
| 2  |   MSZoning    |        Clasificación de la zona (residencial, comercial,
etc.)          |
| 3  |  LotFrontage  |                 Longitud de frente de calle (en pies)
|
| 4  |    LotArea    |                   Área del lote (en pies cuadrados)
|
| 5  |    Street     |                  Tipo de calle (Pavimentada o Grava)
|
| 6  |     Alley     |                      Tipo de callejón (si existe)
|
| 7  |   LotShape    |                           Forma del lote
|
| 8  |  LandContour  |                         Contorno del terreno
|
| 9  |   Utilities   |                     Servicios públicos disponibles
|
| 10 |   LotConfig   |                         Configuración del lote
|
```

11	LandSlope	Inclinación del terreno
12	Neighborhood	Barrio
13	Condition1	Proximidad a carreteras principales o vías férreas
14	Condition2	Otra proximidad a carreteras principales o vías férreas (si aplica)
15	BldgType	Tipo de edificio
16	HouseStyle	Estilo de la casa
17	OverallQual	Calidad general del material y acabado
18	OverallCond	Condición general de la casa
19	YearBuilt	Año de construcción
20	YearRemodAdd	Año de remodelación o adición
21	RoofStyle	Estilo del techo
22	RoofMatl	Material del techo
23	Exterior1st	Material exterior de recubrimiento
24	Exterior2nd	Otro material exterior de recubrimiento (si aplica)
25	MasVnrType	Tipo de recubrimiento de mampostería
26	MasVnrArea	Área de recubrimiento de mampostería (en pies cuadrados)
27	ExterQual	Calidad del material exterior
28	ExterCond	Condición del material exterior
29	Foundation	Tipo de cimientos
30	BsmtQual	Altura de los cimientos
31	BsmtCond	Condición de los cimientos
32	BsmtExposure	Exposición del sótano
33	BsmtFinType1	Calidad del área terminada del sótano
34	BsmtFinSF1	Área terminada del sótano (en pies cuadrados)

35	BsmtFinType2	Calidad adicional del área terminada del sótano
36	BsmtFinSF2	Área adicional terminada del sótano (en pies cuadrados)
37	BsmtUnfSF	Área sin terminar del sótano (en pies cuadrados)
38	TotalBsmtSF	Área total del sótano (en pies cuadrados)
39	Heating	Tipo de sistema de calefacción
40	HeatingQC	Calidad del sistema de calefacción
41	CentralAir	Aire acondicionado central (Sí o No)
42	Electrical	Sistema eléctrico
43	1stFlrSF	Área del primer piso (en pies cuadrados)
44	2ndFlrSF	Área del segundo piso (en pies cuadrados)
45	LowQualFinSF	Área de baja calidad terminada (en pies cuadrados)
46	GrLivArea	Área de vida sobre el nivel del suelo (en pies cuadrados)
47	BsmtFullBath	Número de baños completos en el sótano
48	BsmtHalfBath	Número de medios baños en el sótano
49	FullBath	Número de baños completos sobre el nivel del suelo
50	HalfBath	Número de medios baños sobre el nivel del suelo
51	BedroomAbvGr	Número de habitaciones sobre el nivel del suelo
52	KitchenAbvGr	Número de cocinas sobre el nivel del suelo
53	KitchenQual	Calidad de las cocinas
54	TotRmsAbvGrd	Número total de habitaciones sobre el nivel del suelo (excluyendo baños)
55	Functional	Funcionalidad del hogar
56	Fireplaces	Número de chimeneas
57	FireplaceQu	Calidad de las chimeneas
58	GarageType	Ubicación del garaje

59	GarageYrBlt	Año de construcción del garaje
60	GarageFinish	Acabado interior del garaje
61	GarageCars	Capacidad del garaje en términos de autos
62	GarageArea	Área del garaje (en pies cuadrados)
63	GarageQual	Calidad del garaje
64	GarageCond	Condición del garaje
65	PavedDrive	Calle pavimentada (Sí o No)
66	WoodDeckSF	Área de la terraza de madera (en pies cuadrados)
67	OpenPorchSF	Área del porche abierto (en pies cuadrados)
68	EnclosedPorch	Área del porche cerrado (en pies cuadrados)
69	3SsnPorch	Área del porche de tres estaciones (en pies cuadrados)
70	ScreenPorch	Área del porche con mosquitero (en pies cuadrados)
71	PoolArea	Área de la piscina (en pies cuadrados)
72	PoolQC	Calidad de la piscina
73	Fence	Calidad de la cerca
74	MiscFeature	Características misceláneas no cubiertas en otras categorías
75	MiscVal	Valor de las características misceláneas
76	MoSold	Mes de venta
77	YrSold	Año de venta
78	SaleType	Tipo de venta
79	SaleCondition	Condición de la venta

1.3 1. EDA Basico

1.3.1 1.1 Cargar el dataset y visualizar las primeras filas (df.head()).

[28]:
```
import pandas as pd

# Cargar el archivo CSV
df = pd.read_csv("train.csv")

# Visualizar las primeras 5 filas
print("Primeras 5 filas del conjunto de datos:")
filas = df.head()
print(tabulate(filas, headers="keys", tablefmt="pretty"))
```

```
Primeras 5 filas del conjunto de datos:
+---+----+------------+----------+-------------+---------+--------+-------+-----
-----+-------------+-----------+-----------+-----------+--------------+---------
---+------------+----------+------------+-------------+-------------+-----------
+-------------+-----------+----------+-------------+-------------+------------
+------------+-----------+-----------+------------+----------+----------+-------
-------+--------------+------------+-------------+------------+-----------+----
---------+---------+-----------+------------+------------+----------+----------
+-------------+-----------+--------------+--------------+----------+----------
+-------------+--------------+-------------+--------------+------------+-------
-----+-------------+------------+-------------+--------------+------------+-----
-------+------------+------------+------------+------------+-------------+------
---------+-----------+-------------+----------+--------+-------+-------------+--
-------+--------+--------+----------+---------------+-----------+
|   | Id | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley |
LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighborhood |
Condition1 | Condition2 | BldgType | HouseStyle | OverallQual | OverallCond |
YearBuilt | YearRemodAdd | RoofStyle | RoofMatl | Exterior1st | Exterior2nd |
MasVnrType | MasVnrArea | ExterQual | ExterCond | Foundation | BsmtQual |
BsmtCond | BsmtExposure | BsmtFinType1 | BsmtFinSF1 | BsmtFinType2 | BsmtFinSF2
| BsmtUnfSF | TotalBsmtSF | Heating | HeatingQC | CentralAir | Electrical |
1stFlrSF | 2ndFlrSF | LowQualFinSF | GrLivArea | BsmtFullBath | BsmtHalfBath |
FullBath | HalfBath | BedroomAbvGr | KitchenAbvGr | KitchenQual | TotRmsAbvGrd |
Functional | Fireplaces | FireplaceQu | GarageType | GarageYrBlt | GarageFinish
| GarageCars | GarageArea | GarageQual | GarageCond | PavedDrive | WoodDeckSF |
OpenPorchSF | EnclosedPorch | 3SsnPorch | ScreenPorch | PoolArea | PoolQC |
Fence | MiscFeature | MiscVal | MoSold | YrSold | SaleType | SaleCondition |
SalePrice |
+---+----+------------+----------+-------------+---------+--------+-------+-----
-----+-------------+-----------+-----------+-----------+--------------+---------
---+------------+----------+------------+-------------+-------------+-----------
+-------------+-----------+----------+-------------+-------------+------------
+------------+-----------+-----------+------------+----------+----------+-------
-------+--------------+------------+-------------+------------+-----------+----
```

```
---------+---------+----------+----------+----------+---------+---------
+-------------+----------+-------------+-------------+---------+---------
+-------------+-------------+------------+-------------+-----------+------
-----+------------+-----------+------------+-------------+-----------+-----
------+------------+------------+-----------+-----------+------------+------
---------+----------+------------+---------+-------+-------+------------+--
-------+--------+--------+----------+---------------+-----------+
| 0 | 1  |     60     |    RL    |    65.0     |  8450   |  Pave  |  nan  |
Reg    |     Lvl     |  AllPub   |  Inside   |    Gtl    |   CollgCr    |
Norm    |    Norm    |   1Fam   |   2Story   |      7      |      5      |
2003    |     2003     |   Gable   | CompShg  |   VinylSd   |   VinylSd   |
BrkFace   |   196.0    |    Gd     |    TA     |   PConc    |    Gd    |    TA
|      No      |     GLQ      |    706     |     Unf      |     0      |    150
|     856     |  GasA   |    Ex     |     Y      |   SBrkr    |   856    |   854
|      0       |   1710    |      1       |      0       |    2     |    1     |
3       |      1       |     Gd      |      8       |    Typ     |     0      |
nan     |   Attchd   |   2003.0    |     RFn      |     2      |    548     |
TA     |     TA     |     Y      |     0      |     61      |       0       |
0     |      0      |    0     |  nan   |  nan  |     nan     |    0    |   2
|  2008  |    WD    |    Normal     |  208500   |
| 1 | 2  |     20     |    RL    |    80.0     |  9600   |  Pave  |  nan  |
Reg    |     Lvl     |  AllPub   |    FR2    |    Gtl    |   Veenker    |
Feedr    |    Norm    |   1Fam   |   1Story   |      6      |      8      |
1976    |     1976     |   Gable   | CompShg  |   MetalSd   |   MetalSd   |
nan     |    0.0     |    TA     |    TA     |   CBlock   |    Gd    |    TA
|      Gd      |     ALQ      |    978     |     Unf      |     0      |    284
|    1262     |  GasA   |    Ex     |     Y      |   SBrkr    |   1262   |    0
|      0       |   1262    |      0       |      1       |    2     |    0     |
3       |      1       |     TA      |      6       |    Typ     |     1      |
TA      |   Attchd   |   1976.0    |     RFn      |     2      |    460     |
TA     |     TA     |     Y      |    298     |      0      |       0       |
0     |      0      |    0     |  nan   |  nan  |     nan     |    0    |   5
|  2007  |    WD    |    Normal     |  181500   |
| 2 | 3  |     60     |    RL    |    68.0     |  11250  |  Pave  |  nan  |
IR1    |     Lvl     |  AllPub   |  Inside   |    Gtl    |   CollgCr    |
Norm    |    Norm    |   1Fam   |   2Story   |      7      |      5      |
2001    |     2002     |   Gable   | CompShg  |   VinylSd   |   VinylSd   |
BrkFace   |   162.0    |    Gd     |    TA     |   PConc    |    Gd    |    TA
|      Mn      |     GLQ      |    486     |     Unf      |     0      |    434
|     920     |  GasA   |    Ex     |     Y      |   SBrkr    |   920    |   866
|      0       |   1786    |      1       |      0       |    2     |    1     |
3       |      1       |     Gd      |      6       |    Typ     |     1      |
TA      |   Attchd   |   2001.0    |     RFn      |     2      |    608     |
TA     |     TA     |     Y      |     0      |     42      |       0       |
0     |      0      |    0     |  nan   |  nan  |     nan     |    0    |   9
|  2008  |    WD    |    Normal     |  223500   |
| 3 | 4  |     70     |    RL    |    60.0     |  9550   |  Pave  |  nan  |
IR1    |     Lvl     |  AllPub   |  Corner   |    Gtl    |   Crawfor    |
```

```
Norm    |    Norm    |   1Fam   |   2Story   |      7      |      5      |
1915    |     1970     |   Gable   | CompShg  |   Wd Sdng   |   Wd Shng   |
nan     |    0.0     |    TA     |    TA     |   BrkTil   |    TA    |    Gd
|      No      |     ALQ      |    216     |     Unf      |     0      |    540
|     756     |  GasA   |    Gd     |     Y      |   SBrkr    |   961    |   756
|      0       |   1717    |      1       |      0       |    1     |    0     |
3       |      1       |     Gd      |      7       |    Typ     |     1      |
Gd      |   Detchd   |   1998.0    |     Unf      |     3      |    642     |
TA     |     TA     |     Y      |     0      |     35      |      272      |
0     |      0      |    0     |  nan   |  nan  |     nan     |    0    |   2
|  2006  |    WD    |    Abnorml    |  140000   |
| 4 | 5  |     60     |    RL    |    84.0     |  14260  |  Pave  |  nan  |
IR1    |     Lvl     |  AllPub   |    FR2    |    Gtl    |   NoRidge    |
Norm    |    Norm    |   1Fam   |   2Story   |      8      |      5      |
2000    |     2000     |   Gable   | CompShg  |   VinylSd   |   VinylSd   |
BrkFace   |   350.0    |    Gd     |    TA     |   PConc    |    Gd    |    TA
|      Av      |     GLQ      |    655     |     Unf      |     0      |    490
|    1145     |  GasA   |    Ex     |     Y      |   SBrkr    |   1145   |
1053   |      0       |   2198    |      1       |      0       |    2     |
1     |      4       |      1       |     Gd      |      9       |    Typ     |
1      |     TA      |   Attchd   |   2000.0    |     RFn      |     3      |
836     |     TA     |     TA     |     Y      |    192     |     84      |
0       |     0     |      0      |    0     |  nan   |  nan  |     nan     |
0    |   12   |  2008  |    WD    |    Normal     |  250000   |
+---+----+-----------+---------+------------+--------+-------+-------+-----
-----+-------------+-----------+-----------+-----------+--------------+---------
---+-----------+---------+-----------+------------+------------+----------
+-------------+----------+---------+------------+------------+-----------
+-----------+----------+----------+-----------+---------+---------+-------
------+-------------+-----------+-------------+------------+----------+----
--------+---------+----------+-----------+-----------+---------+----------
+-------------+----------+-------------+-------------+---------+----------
+-------------+-------------+------------+-------------+------------+-------
-----+-------------+-----------+------------+-------------+------------+-----
------+-----------+-----------+-----------+-----------+-------------+------
--------+----------+------------+---------+--------+------+-------------+--
------+--------+--------+---------+---------------+----------+
```

1.3.2 1.2 Identificar tipos de variables y valores nulos (df.info(), df.isnull().sum()).

[8]:
```
from tabulate import tabulate

# Verificar tipos de datos
tipos_datos = pd.DataFrame(df.dtypes, columns=["Tipo de Dato"])

# Verificar valores nulos y ordenarlos de mayor a menor
valores_nulos = pd.DataFrame(df.isnull().sum(), columns=["Valores Nulos"])
```

```
valores_nulos = valores_nulos[valores_nulos["Valores Nulos"] > 0]  # Opcional: Mostrar solo columnas con valores nulos
valores_nulos = valores_nulos.sort_values(by="Valores Nulos", ascending=False)

print("\nTipos de datos:")
print(tabulate(tipos_datos, headers="keys", tablefmt="pretty"))

print("\nValores nulos por columna (de mayor a menor):")
print(tabulate(valores_nulos, headers="keys", tablefmt="pretty"))
```

```
Tipos de datos:
+---------------+--------------+
|               | Tipo de Dato |
+---------------+--------------+
|      Id       |    int64     |
|  MSSubClass   |    int64     |
|   MSZoning    |    object    |
|  LotFrontage  |   float64    |
|    LotArea    |    int64     |
|    Street     |    object    |
|     Alley     |    object    |
|   LotShape    |    object    |
|  LandContour  |    object    |
|   Utilities   |    object    |
|   LotConfig   |    object    |
|   LandSlope   |    object    |
| Neighborhood  |    object    |
|  Condition1   |    object    |
|  Condition2   |    object    |
|   BldgType    |    object    |
|  HouseStyle   |    object    |
|  OverallQual  |    int64     |
|  OverallCond  |    int64     |
|   YearBuilt   |    int64     |
| YearRemodAdd  |    int64     |
|   RoofStyle   |    object    |
|   RoofMatl    |    object    |
|  Exterior1st  |    object    |
|  Exterior2nd  |    object    |
|  MasVnrType   |    object    |
|  MasVnrArea   |   float64    |
|   ExterQual   |    object    |
|   ExterCond   |    object    |
|  Foundation   |    object    |
|   BsmtQual    |    object    |
|   BsmtCond    |    object    |
| BsmtExposure  |    object    |
```

BsmtFinType1	object
BsmtFinSF1	int64
BsmtFinType2	object
BsmtFinSF2	int64
BsmtUnfSF	int64
TotalBsmtSF	int64
Heating	object
HeatingQC	object
CentralAir	object
Electrical	object
1stFlrSF	int64
2ndFlrSF	int64
LowQualFinSF	int64
GrLivArea	int64
BsmtFullBath	int64
BsmtHalfBath	int64
FullBath	int64
HalfBath	int64
BedroomAbvGr	int64
KitchenAbvGr	int64
KitchenQual	object
TotRmsAbvGrd	int64
Functional	object
Fireplaces	int64
FireplaceQu	object
GarageType	object
GarageYrBlt	float64
GarageFinish	object
GarageCars	int64
GarageArea	int64
GarageQual	object
GarageCond	object
PavedDrive	object
WoodDeckSF	int64
OpenPorchSF	int64
EnclosedPorch	int64
3SsnPorch	int64
ScreenPorch	int64
PoolArea	int64
PoolQC	object
Fence	object
MiscFeature	object
MiscVal	int64
MoSold	int64
YrSold	int64
SaleType	object
SaleCondition	object
SalePrice	int64

```
+--------------+--------------+

Valores nulos por columna (de mayor a menor):
+--------------+---------------+
|              | Valores Nulos |
+--------------+---------------+
|    PoolQC    |     1453      |
| MiscFeature  |     1406      |
|    Alley     |     1369      |
|    Fence     |     1179      |
|  MasVnrType  |      872      |
| FireplaceQu  |      690      |
| LotFrontage  |      259      |
|  GarageType  |      81       |
| GarageYrBlt  |      81       |
| GarageFinish |      81       |
|  GarageQual  |      81       |
|  GarageCond  |      81       |
| BsmtFinType2 |      38       |
| BsmtExposure |      38       |
| BsmtFinType1 |      37       |
|   BsmtCond   |      37       |
|   BsmtQual   |      37       |
|  MasVnrArea  |       8       |
|  Electrical  |       1       |
+--------------+---------------+
```

1.3.3 1.3 Eliminar o imputar nulos (justificando la estrategia).

[31]:
```
# Analizar valores nulos
missing_values = df.isnull().sum()
total_rows = len(df)
missing_percentage = (missing_values / total_rows) * 100

# Mostrar porcentajes con dos decimales y en formato de porcentaje
missing_percentage = missing_percentage[missing_percentage > 0].
 ↪sort_values(ascending=False)
missing_percentage_formatted = missing_percentage.apply(lambda x: f"{x:.2f}%")

print("Porcentaje de valores nulos por columna:")
print(missing_percentage_formatted)

# --- Eliminar columnas con más del 80% de valores nulos ---
columns_to_drop = missing_percentage[missing_percentage > 80].index
df = df.drop(columns=columns_to_drop)
print(f"\nColumnas eliminadas (más del 80% de valores nulos):␣
 ↪{list(columns_to_drop)}")
```

```python
# --- Imputación de valores nulos ---
# Para columnas numéricas, imputar con la mediana
num_cols = df.select_dtypes(include=["float64", "int64"]).columns
for col in num_cols:
    if df[col].isnull().sum() > 0:
        median_value = df[col].median()
        df[col] = df[col].fillna(median_value)  # Actualizar directamente sin␣
 ↪inplace
        print(f"Columna '{col}' (numérica): imputada con la mediana␣
 ↪({median_value}).")

# Para columnas categóricas, imputar con la moda
cat_cols = df.select_dtypes(include=["object"]).columns
for col in cat_cols:
    if df[col].isnull().sum() > 0:
        mode_value = df[col].mode()[0]
        df[col] = df[col].fillna(mode_value)  # Actualizar directamente sin␣
 ↪inplace
        print(f"Columna '{col}' (categórica): imputada con la moda␣
 ↪('{mode_value}').")

# Verificar si quedan valores nulos
print("\n¿Quedan valores nulos?")
print(df.isnull().sum().sum())
```

```
Porcentaje de valores nulos por columna:
PoolQC          99.52%
MiscFeature     96.30%
Alley           93.77%
Fence           80.75%
MasVnrType      59.73%
FireplaceQu     47.26%
LotFrontage     17.74%
GarageType       5.55%
GarageYrBlt      5.55%
GarageFinish     5.55%
GarageQual       5.55%
GarageCond       5.55%
BsmtFinType2     2.60%
BsmtExposure     2.60%
BsmtFinType1     2.53%
BsmtCond         2.53%
BsmtQual         2.53%
MasVnrArea       0.55%
Electrical       0.07%
dtype: object
```

```
Columnas eliminadas (más del 80% de valores nulos): ['PoolQC', 'MiscFeature',
'Alley', 'Fence']
Columna 'LotFrontage' (numérica): imputada con la mediana (69.0).
Columna 'MasVnrArea' (numérica): imputada con la mediana (0.0).
Columna 'GarageYrBlt' (numérica): imputada con la mediana (1980.0).
Columna 'MasVnrType' (categórica): imputada con la moda ('BrkFace').
Columna 'BsmtQual' (categórica): imputada con la moda ('TA').
Columna 'BsmtCond' (categórica): imputada con la moda ('TA').
Columna 'BsmtExposure' (categórica): imputada con la moda ('No').
Columna 'BsmtFinType1' (categórica): imputada con la moda ('Unf').
Columna 'BsmtFinType2' (categórica): imputada con la moda ('Unf').
Columna 'Electrical' (categórica): imputada con la moda ('SBrkr').
Columna 'FireplaceQu' (categórica): imputada con la moda ('Gd').
Columna 'GarageType' (categórica): imputada con la moda ('Attchd').
Columna 'GarageFinish' (categórica): imputada con la moda ('Unf').
Columna 'GarageQual' (categórica): imputada con la moda ('TA').
Columna 'GarageCond' (categórica): imputada con la moda ('TA').

¿Quedan valores nulos?
0
```

Justificacion

1. Eliminación de Columnas con un Alto Porcentaje de Valores Nulos

Justificación: Las columnas con más del 80% de valores nulos contienen información incompleta que puede no ser representativa o útil para el análisis. Mantener estas columnas podría introducir ruido en los resultados y complicar la interpretación de los datos.

2. Imputación de Valores Nulos para Columnas Numéricas

Justificación: La mediana es robusta frente a valores atípicos, ya que no se ve afectada por datos extremos como lo haría el promedio. Usar la mediana asegura que los valores imputados representen mejor la tendencia central de la distribución.

3. Imputación de Valores Nulos para Columnas Categóricas

Justificación: La moda es adecuada para columnas categóricas porque representa la categoría más común en los datos. Esto asegura que los valores imputados sean consistentes con las tendencias observadas en los datos.

1.3.4 1.4 Seleccionar solo variables numéricas o categóricas codificadas (ideal para clustering).

[33]:
```python
from tabulate import tabulate

# Seleccionar solo variables numéricas (int64 y float64)
numeric_df = df.select_dtypes(include=["int64", "float64"])

# Convertir el DataFrame en formato tabulado
```

```
tabla = tabulate(numeric_df.head(), headers='keys', tablefmt='grid')

# Imprimir la tabla tabulada
print("Variables numéricas seleccionadas (tabuladas):")
print(tabla)
```

```
Variables numéricas seleccionadas (tabuladas):
+----+------+-------------+--------------+-----------+--------------+--------
-------+-------------+---------------+-------------+-------------+-----------
---+-------------+--------------+-----------+-----------+----------------+---
----------+---------------+---------------+-----------+-----------+---------
-------+---------------+---------------+-------------+--------------+-------
-------+--------------+--------------+---------------+----------------+--------
-----+---------------+-----------+----------+----------+----------+-----------
--+
|    |   Id |   MSSubClass |   LotFrontage |   LotArea |   OverallQual |
OverallCond |   YearBuilt |   YearRemodAdd |   MasVnrArea |   BsmtFinSF1 |
BsmtFinSF2 |   BsmtUnfSF |   TotalBsmtSF |   1stFlrSF |   2ndFlrSF |
LowQualFinSF |   GrLivArea |   BsmtFullBath |   BsmtHalfBath |   FullBath |
HalfBath |   BedroomAbvGr |   KitchenAbvGr |   TotRmsAbvGrd |   Fireplaces |
GarageYrBlt |   GarageCars |   GarageArea |   WoodDeckSF |   OpenPorchSF |
EnclosedPorch |   3SsnPorch |   ScreenPorch |   PoolArea |   MiscVal |   MoSold
|   YrSold |   SalePrice |
+====+======+=============+==============+===========+==============+========
=======+=============+===============+=============+=============+===========
===+=============+===============+============+============+================+===
==========+===============+===============+============+============+=========
=======+===============+===============+=============+==============+=======
=======+==============+==============+===============+================+========
=====+===============+============+===========+==========+==========+==========
==+
|  0 |    1 |          60 |           65 |      8450 |            7 |
5 |        2003 |           2003 |          196 |          706 |            0 |
150 |          856 |        856 |        854 |             0 |        1710 |
1 |             0 |          2 |          1 |             3 |             1 |
8 |           0 |          2003 |           2 |          548 |            0 |
61 |              0 |           0 |            0 |          0 |         0 |
2 |     2008 |      208500 |
+----+------+-------------+--------------+-----------+--------------+--------
-------+-------------+---------------+-------------+-------------+-----------
---+-------------+--------------+-----------+-----------+----------------+---
----------+---------------+---------------+-----------+-----------+---------
-------+---------------+---------------+-------------+--------------+-------
-------+--------------+--------------+---------------+----------------+--------
-----+---------------+-----------+----------+----------+----------+-----------
--+
|  1 |    2 |          20 |           80 |      9600 |            6 |
8 |        1976 |           1976 |            0 |          978 |            0 |
```

```
284 |          1262 |       1262 |          0 |              0 |        1262 |
0 |             1 |          2 |          0 |              3 |             1 |
6 |           1 |          1976 |            2 |          460 |          298 |
0 |              0 |           0 |             0 |          0 |         0 |
5 |     2007 |      181500 |
+----+------+-------------+---------------+----------+--------------+--------
-------+-------------+---------------+--------------+--------------+----------
---+-------------+---------------+-----------+-----------+----------------+---
----------+----------------+----------------+------------+------------+---------
-------+----------------+----------------+--------------+--------------+-------
-------+--------------+-------------+---------------+----------------+--------
-----+---------------+-----------+----------+----------+----------+----------
--+
|  2 |    3 |           60 |            68 |     11250 |             7 |
5 |        2001 |           2002 |          162 |          486 |            0 |
434 |           920 |        920 |        866 |              0 |        1786 |
1 |             0 |          2 |          1 |              3 |             1 |
6 |           1 |          2001 |            2 |          608 |            0 |
42 |              0 |           0 |             0 |          0 |         0 |
9 |     2008 |      223500 |
+----+------+-------------+---------------+----------+--------------+--------
-------+-------------+---------------+--------------+--------------+----------
---+-------------+---------------+-----------+-----------+----------------+---
----------+----------------+----------------+------------+------------+---------
-------+----------------+----------------+--------------+--------------+-------
-------+--------------+-------------+---------------+----------------+--------
-----+---------------+-----------+----------+----------+----------+----------
--+
|  3 |    4 |           70 |            60 |      9550 |             7 |
5 |        1915 |           1970 |            0 |          216 |            0 |
540 |           756 |        961 |        756 |              0 |        1717 |
1 |             0 |          1 |          0 |              3 |             1 |
7 |           1 |          1998 |            3 |          642 |            0 |
35 |            272 |           0 |             0 |          0 |         0 |
2 |     2006 |      140000 |
+----+------+-------------+---------------+----------+--------------+--------
-------+-------------+---------------+--------------+--------------+----------
---+-------------+---------------+-----------+-----------+----------------+---
----------+----------------+----------------+------------+------------+---------
-------+----------------+----------------+--------------+--------------+-------
-------+--------------+-------------+---------------+----------------+--------
-----+---------------+-----------+----------+----------+----------+----------
--+
|  4 |    5 |           60 |            84 |     14260 |             8 |
5 |        2000 |           2000 |          350 |          655 |            0 |
490 |          1145 |       1145 |       1053 |              0 |        2198 |
1 |             0 |          2 |          1 |              4 |             1 |
9 |           1 |          2000 |            3 |          836 |          192 |
```

```
84 |               0 |           0 |              0 |          0 |         0 |
12 |     2008 |      250000 |
+----+------+-------------+--------------+-----------+--------------+--------
-------+-------------+---------------+--------------+--------------+-----------
---+-------------+---------------+------------+------------+----------------+---
----------+---------------+---------------+------------+------------+---------
-------+----------------+----------------+--------------+---------------+-------
-------+--------------+-------------+---------------+----------------+--------
-----+---------------+------------+-----------+----------+----------+-----------
--+
```

1.3.5 1.5 Escalar los datos con StandardScaler o MinMaxScaler

[35]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from tabulate import tabulate

# Excluir columnas con alto sesgo o identificadores directos
columns_to_exclude = ["Id", "SalePrice"]  # Columnas a excluir
df_filtered = df.drop(columns=columns_to_exclude, errors="ignore")

# Seleccionar solo variables numéricas
numeric_df = df_filtered.select_dtypes(include=["int64", "float64"])

# Escalar las variables numéricas con StandardScaler
standard_scaler = StandardScaler()
scaled_standard = standard_scaler.fit_transform(numeric_df)
scaled_standard_df = pd.DataFrame(scaled_standard, columns=numeric_df.columns)

# Escalar las variables numéricas con MinMaxScaler
minmax_scaler = MinMaxScaler()
scaled_minmax = minmax_scaler.fit_transform(numeric_df)
scaled_minmax_df = pd.DataFrame(scaled_minmax, columns=numeric_df.columns)

# Mostrar las primeras filas de los datos escalados en formato tabulado
print("Datos escalados con StandardScaler:")
print(tabulate(scaled_standard_df.head(), headers='keys', tablefmt='grid'))

print("\nDatos escalados con MinMaxScaler:")
print(tabulate(scaled_minmax_df.head(), headers='keys', tablefmt='grid'))
```

```
Datos escalados con StandardScaler:
+----+-------------+--------------+-----------+--------------+-------------
-+------------+---------------+-------------+-------------+-------------+--
----------+--------------+-----------+-----------+---------------+---------
----+---------------+--------------+-----------+-----------+--------------
-+----------------+---------------+-------------+--------------+-------------
-+-------------+-------------+--------------+----------------+------------
+--------------+-----------+-----------+----------+----------+
```

```
|    |   MSSubClass |   LotFrontage |    LotArea |   OverallQual |   OverallCond
|   YearBuilt |   YearRemodAdd |   MasVnrArea |   BsmtFinSF1 |   BsmtFinSF2 |
BsmtUnfSF |   TotalBsmtSF |   1stFlrSF |   2ndFlrSF |   LowQualFinSF |
GrLivArea |   BsmtFullBath |   BsmtHalfBath |   FullBath |   HalfBath |
BedroomAbvGr |   KitchenAbvGr |   TotRmsAbvGrd |   Fireplaces |   GarageYrBlt |
GarageCars |   GarageArea |   WoodDeckSF |   OpenPorchSF |   EnclosedPorch |
3SsnPorch |   ScreenPorch |   PoolArea |    MiscVal |    MoSold |    YrSold |
+====+==============+==============+============+==============+==============
=+=============+===============+==============+==============+==============+==
==========+==============+============+============+===============+=========
====+===============+===============+============+============+==============
=+================+===============+==============+==============+=============
=+==============+==============+==============+================+=============+
==============+============+============+===========+===========+
|  0 |     0.073375 |    -0.220875  | -0.207142  |     0.651479  |      -0.5172
|    1.05099  |       0.878668 |     0.514104 |    0.575425  |    -0.288653 |
-0.944591  |     -0.459303 | -0.793434  |   1.16185  |      -0.120242 |
0.370333 |       1.10781  |      -0.241061 |   0.789741 |   1.22759  |
0.163779 |      -0.211454 |       0.91221  |    -0.951226 |      1.0176   |
0.311725 |     0.351    |    -0.752176 |     0.216503  |       -0.359325 |
-0.116339 |     -0.270208 | -0.0686917 | -0.0876878 | -1.59911  |  0.138777 |
+----+--------------+--------------+------------+--------------+--------------
-+-------------+----------------+-------------+--------------+--------------+--
-----------+--------------+------------+------------+---------------+---------
----+----------------+---------------+------------+------------+--------------
-+----------------+---------------+--------------+---------------+-------------
-+--------------+--------------+--------------+-----------------+-------------
+---------------+------------+-----------+----------+-----------+
|  1 |    -0.872563 |     0.46032   | -0.0918864 |    -0.0718361 |       2.17963
|    0.156734 |      -0.429577 |    -0.57075  |    1.17199   |    -0.288653 |
-0.641228  |      0.466465 |  0.25714   |  -0.795163 |      -0.120242 |
-0.482512 |      -0.819964 |       3.94881  |   0.789741 |  -0.761621 |
0.163779 |      -0.211454 |      -0.318683 |     0.600495 |     -0.107927 |
0.311725 |    -0.060731 |     1.62619  |    -0.704483  |       -0.359325 |
-0.116339 |     -0.270208 | -0.0686917 | -0.0876878 | -0.48911  | -0.614439 |
+----+--------------+--------------+------------+--------------+--------------
-+-------------+----------------+-------------+--------------+--------------+--
-----------+--------------+------------+------------+---------------+---------
----+----------------+---------------+------------+------------+--------------
-+----------------+---------------+--------------+---------------+-------------
-+--------------+--------------+--------------+-----------------+-------------
+---------------+------------+-----------+----------+-----------+
|  2 |     0.073375 |    -0.0846361 |  0.07348   |     0.651479  |      -0.5172
|    0.984752 |       0.830215 |     0.325915 |    0.0929072 |    -0.288653 |
-0.301643  |     -0.313369 | -0.627826  |   1.18935  |      -0.120242 |
0.515013 |       1.10781  |      -0.241061 |   0.789741 |   1.22759  |
0.163779 |      -0.211454 |      -0.318683 |     0.600495 |      0.934226 |
0.311725 |     0.631726 |    -0.752176 |    -0.0703615 |       -0.359325 |
```

```
-0.116339 |     -0.270208 | -0.0686917 | -0.0876878 |  0.990891 |  0.138777 |
+----+--------------+--------------+------------+---------------+--------------
-+-------------+----------------+-------------+--------------+--------------+--
----------+--------------+------------+------------+----------------+---------
----+----------------+----------------+------------+------------+--------------
-+----------------+---------------+-------------+--------------+-------------
-+--------------+--------------+--------------+-----------------+-------------
+--------------+------------+------------+-----------+-----------+
|  3 |     0.309859 |    -0.44794   | -0.0968975 |     0.651479  |      -0.5172
|   -1.86363  |      -0.720298 |    -0.57075  |   -0.499274  |    -0.288653 |
-0.0616696 |     -0.687324 | -0.521734  |   0.937276 |      -0.120242 |
0.383659 |       1.10781  |      -0.241061 |  -1.02604  |  -0.761621 |
0.163779 |      -0.211454 |       0.296763 |     0.600495 |      0.809167 |
1.65031  |     0.790804 |    -0.752176 |    -0.176048  |        4.09252  |
-0.116339 |     -0.270208 | -0.0686917 | -0.0876878 | -1.59911  | -1.36765  |
+----+--------------+--------------+------------+---------------+--------------
-+-------------+----------------+-------------+--------------+--------------+--
----------+--------------+------------+------------+----------------+---------
----+----------------+----------------+------------+------------+--------------
-+----------------+---------------+-------------+--------------+-------------
-+--------------+--------------+--------------+-----------------+-------------
+--------------+------------+------------+-----------+-----------+
|  4 |     0.073375 |     0.641972  |  0.375148  |     1.37479   |      -0.5172
|    0.951632 |       0.733308 |     1.36649  |    0.463568  |    -0.288653 |
-0.174865  |      0.19968  | -0.0456113 |   1.61788  |      -0.120242 |
1.29933  |       1.10781  |      -0.241061 |   0.789741 |   1.22759  |
1.39002  |      -0.211454 |       1.52766  |     0.600495 |      0.89254  |
1.65031  |     1.69848  |     0.780197 |     0.56376   |       -0.359325 |
-0.116339 |     -0.270208 | -0.0686917 | -0.0876878 |  2.10089  |  0.138777 |
+----+--------------+--------------+------------+---------------+--------------
-+-------------+----------------+-------------+--------------+--------------+--
----------+--------------+------------+------------+----------------+---------
----+----------------+----------------+------------+------------+--------------
-+----------------+---------------+-------------+--------------+-------------
-+--------------+--------------+--------------+-----------------+-------------
+--------------+------------+------------+-----------+-----------+

Datos escalados con MinMaxScaler:
+----+--------------+--------------+-----------+---------------+--------------
+-------------+----------------+-------------+--------------+--------------+---
----------+--------------+------------+------------+----------------+----------
---+----------------+----------------+------------+------------+---------------
+----------------+---------------+-------------+--------------+--------------
+--------------+--------------+--------------+-----------------+-------------+-
--------------+------------+-----------+-----------+-----------+
|    |   MSSubClass |   LotFrontage |   LotArea |   OverallQual |   OverallCond
|   YearBuilt |   YearRemodAdd |   MasVnrArea |   BsmtFinSF1 |   BsmtFinSF2 |
BsmtUnfSF |   TotalBsmtSF |   1stFlrSF |   2ndFlrSF |   LowQualFinSF |
```

```
GrLivArea |   BsmtFullBath |   BsmtHalfBath |   FullBath |   HalfBath |
BedroomAbvGr |   KitchenAbvGr |   TotRmsAbvGrd |   Fireplaces |   GarageYrBlt |
GarageCars |   GarageArea |   WoodDeckSF |   OpenPorchSF |   EnclosedPorch |
3SsnPorch |   ScreenPorch |   PoolArea |   MiscVal |    MoSold |   YrSold |
+====+==============+==============+===========+==============+==============
+============+===============+=============+=============+=============+===
==========+==============+===========+===========+===============+==========
===+===============+===============+===========+===========+================
+================+================+=============+==============+=============
+=============+=============+=============+================+=============+=
=============+===========+==========+==========+==========+
|  0 |     0.235294 |      0.150685 | 0.0334198 |      0.666667 |         0.5
|    0.949275 |       0.883333 |      0.1225  |    0.125089  |           0 |
0.0642123 |      0.140098 |   0.11978  |   0.413559 |              0 |
0.259231 |       0.333333 |            0   |   0.666667 |        0.5 |
0.375 |       0.333333 |       0.5      |     0        |      0.936364 |
0.5  |     0.38646  |     0        |     0.111517  |        0        |
0 |             0 |          0 |         0 | 0.0909091 |     0.5  |
+----+--------------+--------------+-----------+---------------+--------------
+------------+----------------+-------------+-------------+-------------+---
---------+--------------+-----------+-----------+---------------+----------
---+----------------+---------------+-----------+-----------+---------------
+---------------+---------------+-------------+--------------+-------------
+-------------+-------------+-------------+----------------+-------------+-
-------------+-----------+----------+----------+----------+
|  1 |     0        |      0.202055 | 0.038795  |      0.555556 |         0.875
|    0.753623 |       0.433333 |      0       |    0.173281  |           0 |
0.121575  |      0.206547 |   0.212942 |   0        |              0 |
0.17483  |       0        |            0.5 |   0.666667 |        0   |
0.375 |       0.333333 |       0.333333 |     0.333333 |      0.690909 |
0.5  |     0.324401 |     0.347725 |     0         |        0        |
0 |             0 |          0 |         0 | 0.363636  |     0.25 |
+----+--------------+--------------+-----------+---------------+--------------
+------------+----------------+-------------+-------------+-------------+---
---------+--------------+-----------+-----------+---------------+----------
---+----------------+---------------+-----------+-----------+---------------
+---------------+---------------+-------------+--------------+-------------
+-------------+-------------+-------------+----------------+-------------+-
-------------+-----------+----------+----------+----------+
|  2 |     0.235294 |      0.160959 | 0.0465073 |      0.666667 |         0.5
|    0.934783 |       0.866667 |      0.10125 |    0.0861091 |           0 |
0.185788  |      0.150573 |   0.134465 |   0.41937  |              0 |
0.273549 |       0.333333 |            0   |   0.666667 |        0.5 |
0.375 |       0.333333 |       0.333333 |     0.333333 |      0.918182 |
0.5  |     0.428773 |     0        |     0.0767824 |        0        |
0 |             0 |          0 |         0 | 0.727273  |     0.5  |
+----+--------------+--------------+-----------+---------------+--------------
+------------+----------------+-------------+-------------+-------------+---
```

```
---------+--------------+-----------+-----------+---------------+---------
---+---------------+---------------+-----------+-----------+---------------
+---------------+---------------+-------------+--------------+-------------
+-------------+--------------+--------------+-----------------+------------+-
-------------+-----------+----------+----------+----------+
|  3 |     0.294118 |      0.133562 | 0.0385613 |      0.666667 |         0.5
|    0.311594 |       0.333333 |      0       |    0.0382707 |           0 |
0.231164  |      0.123732 |   0.143873 |   0.366102 |              0 |
0.26055  |       0.333333 |            0   |   0.333333 |        0   |
0.375 |       0.333333 |       0.416667 |     0.333333 |      0.890909 |
0.75 |     0.45275  |     0        |     0.0639854 |        0.492754 |
0 |             0 |          0 |         0 | 0.0909091 |     0    |
+----+-------------+--------------+----------+---------------+--------------
+-------------+----------------+-------------+--------------+--------------+---
---------+--------------+-----------+-----------+---------------+---------
---+---------------+---------------+-----------+-----------+---------------
+---------------+---------------+-------------+--------------+-------------
+-------------+--------------+--------------+-----------------+------------+-
-------------+-----------+----------+----------+----------+
|  4 |     0.235294 |      0.215753 | 0.0605763 |      0.777778 |         0.5
|    0.927536 |       0.833333 |      0.21875 |    0.116052  |           0 |
0.20976   |      0.187398 |   0.186095 |   0.509927 |              0 |
0.351168 |       0.333333 |            0   |   0.666667 |        0.5 |
0.5   |       0.333333 |       0.583333 |     0.333333 |      0.909091 |
0.75 |     0.589563 |     0.224037 |     0.153565  |        0        |
0 |             0 |          0 |         0 | 1         |     0.5  |
+----+-------------+--------------+----------+---------------+--------------
+-------------+----------------+-------------+--------------+--------------+---
---------+--------------+-----------+-----------+---------------+---------
---+---------------+---------------+-----------+-----------+---------------
+---------------+---------------+-------------+--------------+-------------
+-------------+--------------+--------------+-----------------+------------+-
-------------+-----------+----------+----------+----------+
```

Interpretacion:

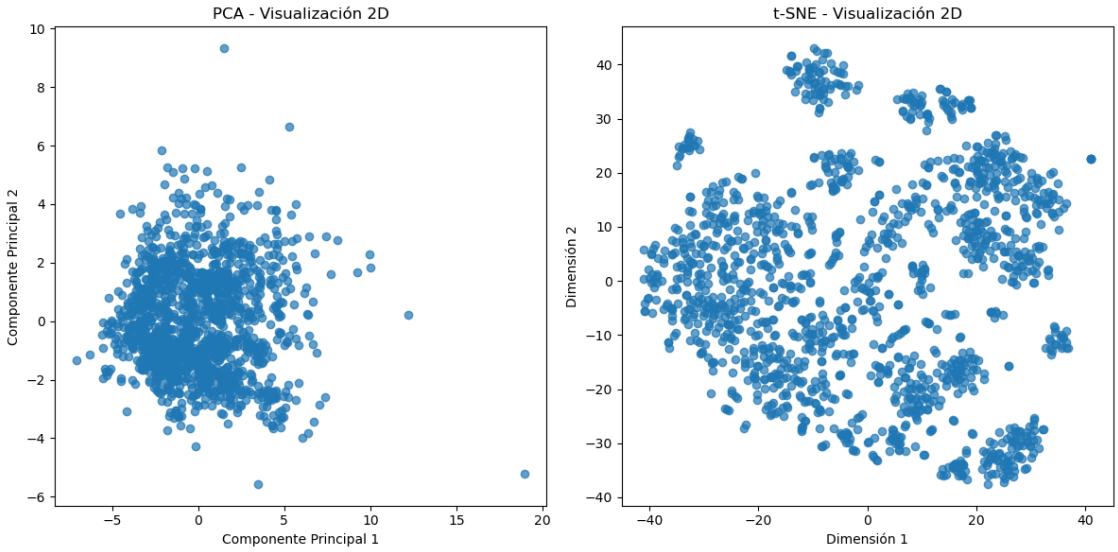
- El propósito del escalado con **StandardScaler** es centrar los datos en una media de 0 y una desviación estándar de 1. Esto es útil para algoritmos que son sensibles a la escala de las variables, como PCA, regresión logística o SVM.
- El propósito del escalado con **MinMaxScaler** es transformar los datos para que estén en un rango entre 0 y 1. Esto es útil para algoritmos que no son robustos ante magnitudes de variables muy diferentes, como redes neuronales o K-Means.

1.4 2. Reducción de dimensionalidad (opcional pero recomendable)

1.4.1 2.1 Aplicar PCA o t-SNE para visualizar datos en 2D.

[40]:
```
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

# 1. Aplicar PCA
pca = PCA(n_components=2)  # Reducir a 2 dimensiones
pca_result = pca.fit_transform(scaled_standard_df)

# 2. Aplicar t-SNE
tsne = TSNE(n_components=2, random_state=42, perplexity=30, n_iter=1000)
tsne_result = tsne.fit_transform(scaled_standard_df)

# Visualización de resultados
plt.figure(figsize=(12, 6))

# Gráfico PCA
plt.subplot(1, 2, 1)
plt.scatter(pca_result[:, 0], pca_result[:, 1], alpha=0.7, cmap='viridis')
plt.title('PCA - Visualización 2D')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')

# Gráfico t-SNE
plt.subplot(1, 2, 2)
plt.scatter(tsne_result[:, 0], tsne_result[:, 1], alpha=0.7, cmap='viridis')
plt.title('t-SNE - Visualización 2D')
plt.xlabel('Dimensión 1')
plt.ylabel('Dimensión 2')

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

```
C:\Users\aleja\AppData\Local\Temp\ipykernel_22520\1715214050.py:18: UserWarning:
No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  plt.scatter(pca_result[:, 0], pca_result[:, 1], alpha=0.7, cmap='viridis')
C:\Users\aleja\AppData\Local\Temp\ipykernel_22520\1715214050.py:25: UserWarning:
No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  plt.scatter(tsne_result[:, 0], tsne_result[:, 1], alpha=0.7, cmap='viridis')
```



Interpretacion:

- Aca mantenemos la escala de datos con StandardScaler, ya que PCA funciona bien con este escalado.
- Gráfico de la izquierda: PCA - Visualización 2D
  - Este gráfico utiliza PCA, una técnica lineal que proyecta los datos en dos componentes principales que capturan la mayor cantidad de varianza posible en el conjunto de datos.
  - Los ejes representan las dos componentes principales:
    - Componente Principal 1: Captura la mayor varianza del dataset.
    - Componente Principal 2: Captura la segunda mayor varianza, ortogonal a la primera.
  - Análisis:
    - Los puntos están densamente agrupados alrededor del centro del gráfico, lo que indica que la mayor parte de las observaciones tienen valores similares en términos de varianza.
    - No se observan agrupaciones claras o patrones definidos. Esto sugiere que las relaciones entre las variables pueden no ser completamente lineales.
    - Los puntos que están más alejados del centro podrían ser outliers o datos atípicos.
- Gráfico de la derecha: t-SNE - Visualización 2D
  - Este gráfico utiliza t-SNE, una técnica no lineal que intenta preservar las relaciones locales entre los puntos en el espacio de alta dimensionalidad, mostrando agrupamientos naturales en el espacio reducido.
  - Los ejes (Dimensión 1 y Dimensión 2) no tienen un significado explícito; son abstracciones generadas por el algoritmo.
  - Análisis:
    - Se observan agrupamientos claros o clusters en el gráfico, lo que indica que el algoritmo t-SNE ha identificado relaciones locales entre los datos.

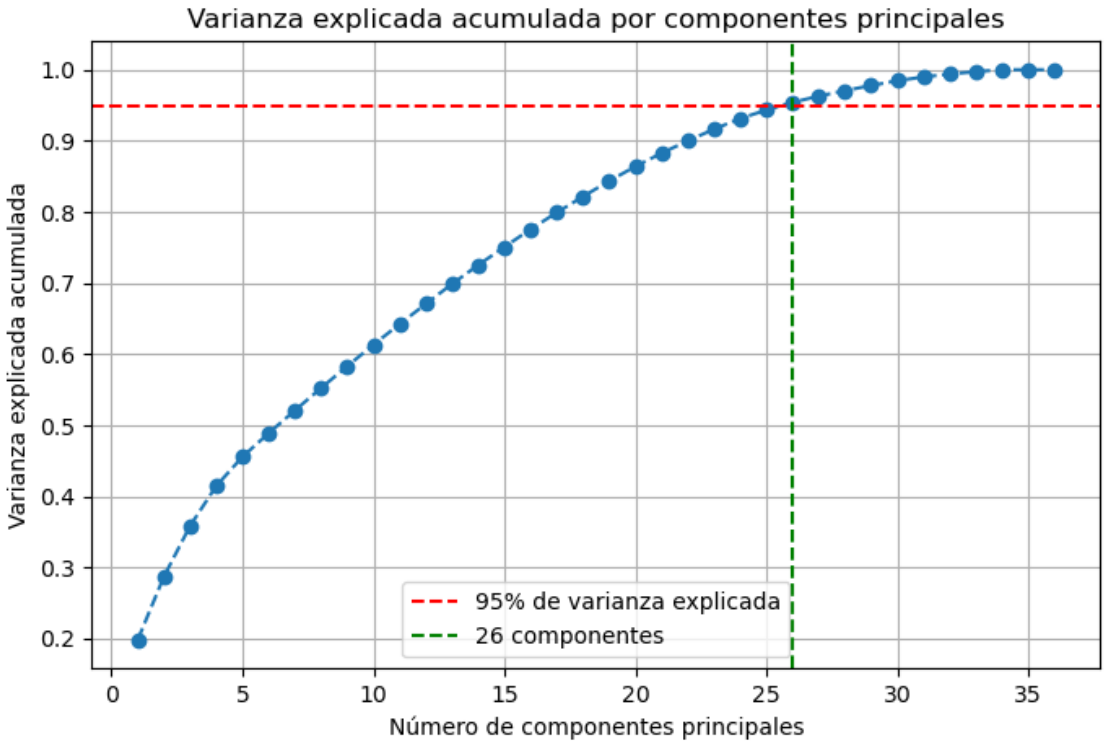
* Los puntos dentro de un cluster están muy juntos, mientras que los clusters están bien separados entre sí.
* Esto sugiere que los datos tienen estructuras internas que t-SNE logra resaltar, como subgrupos o clases.

1.4.2 2.2 Graficar los componentes principales para observar distribución y estructura.

[42]:
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np

# Excluir columnas con alto sesgo o identificadores directos
columns_to_exclude = ["Id", "SalePrice"]  # Columnas a excluir
df_filtered = df.drop(columns=columns_to_exclude, errors="ignore")

# Seleccionar solo variables numéricas
numeric_df = df_filtered.select_dtypes(include=["int64", "float64"])

# Escalar los datos con StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_df)

# Aplicar PCA
pca = PCA()
pca_result = pca.fit_transform(scaled_data)

# Varianza explicada por cada componente
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance_ratio = np.cumsum(explained_variance_ratio)

# Decidir cuántos componentes conservar (mínimo 95% de varianza explicada)
n_components = np.argmax(cumulative_variance_ratio >= 0.95) + 1
print(f"Se conservarán {n_components} componentes principales para explicar al␣
 ↪menos el 95% de la varianza.")

# Graficar la varianza explicada acumulada
plt.figure(figsize=(8, 5))
plt.plot(range(1, len(cumulative_variance_ratio) + 1),␣
 ↪cumulative_variance_ratio, marker='o', linestyle='--')
plt.axhline(y=0.95, color='r', linestyle='--', label='95% de varianza␣
 ↪explicada')
plt.axvline(x=n_components, color='g', linestyle='--', label=f'{n_components}␣
 ↪componentes')
plt.title('Varianza explicada acumulada por componentes principales')
plt.xlabel('Número de componentes principales')
```

```
plt.ylabel('Varianza explicada acumulada')
plt.legend()
plt.grid()
plt.show()
```

```
Se conservarán 26 componentes principales para explicar al menos el 95% de la
varianza.
```



Interpretacion:

El primer grafico muestra la varianza explicada acumulada por los componentes principales de un analisis PCA.

Descripción del Gráfico - Ejes del gráfico: - Eje X (horizontal): Número de componentes principales. Representa los componentes principales en orden, desde el primero (que explica más varianza) hasta el último. - Eje Y (vertical): Varianza explicada acumulada. Indica el porcentaje de varianza total explicada por los componentes principales seleccionados de manera acumulativa. - Curva azul (puntos conectados): - Representa cómo se acumula la varianza explicada a medida que se incluyen más componentes principales. - Línea roja discontinua: - Marca el umbral del 95% de varianza explicada. Este es un criterio común para determinar cuántos componentes principales son necesarios para capturar la mayor parte de la información del dataset. - Línea verde discontinua: - Indica el número de componentes principales (en este caso, 26) necesarios para alcanzar o superar el 95% de varianza explicada.

El gráfico indica que 26 componentes principales son suficientes para explicar al menos el 95% de

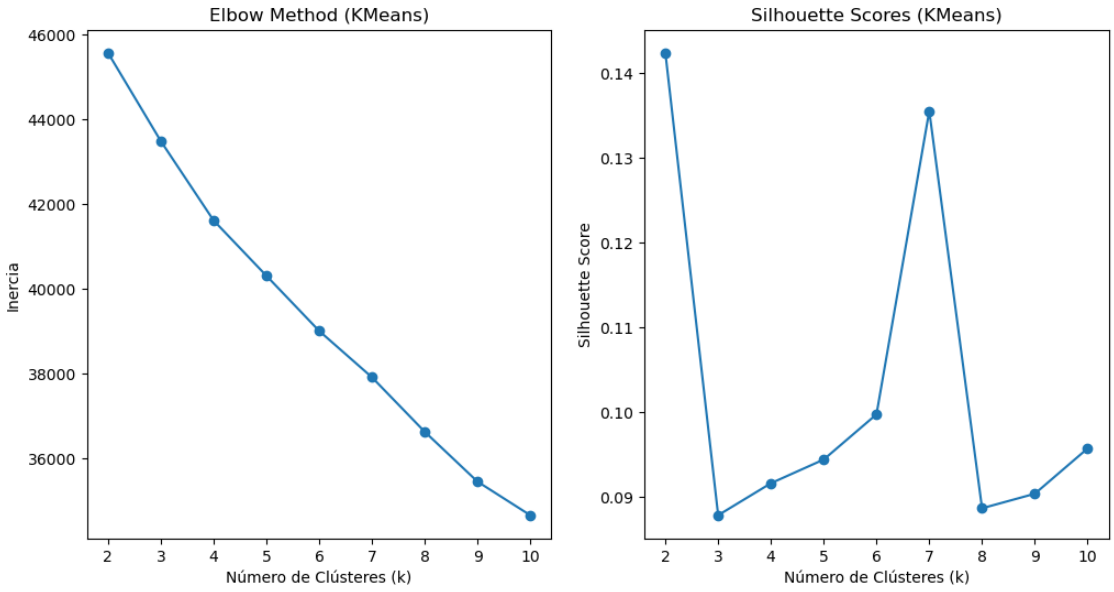
la varianza total en los datos, lo que es una práctica estándar en reducción de dimensionalidad. Esto permite trabajar con un conjunto de datos más compacto sin perder la mayor parte de la información relevante.

1.5 3. Clustering

1.5.1 3.1 Aplicar al menos dos técnicas distintas: KMeans (revisar inercia, Elbow method) y DBSCAN o Agglomerative Clustering como alternativa.

3.1.1 Metodo KMeans (Elbow Method y Silhouette Score)

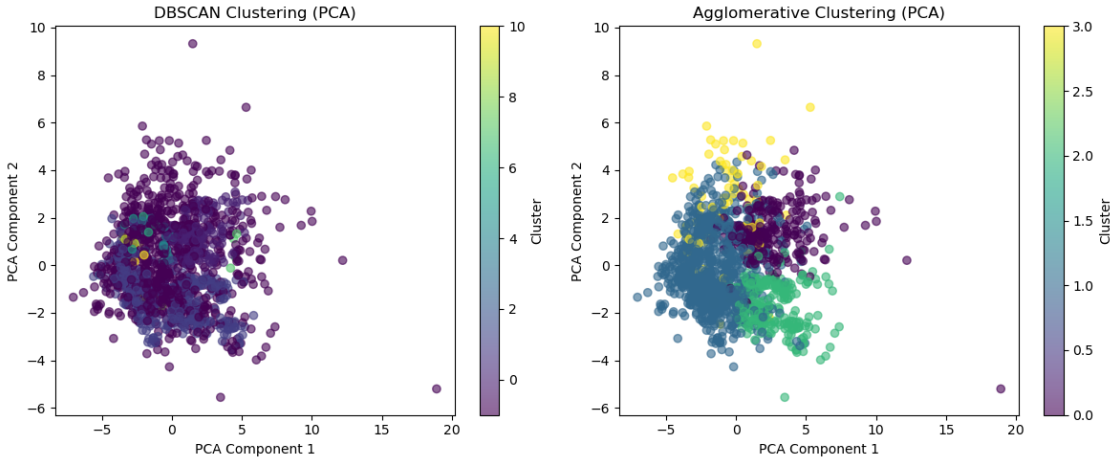
[44]:
```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Aplicar KMeans
def apply_kmeans(data, max_k=10):
    inertia = []
    silhouette_scores = []
    for k in range(2, max_k + 1):
        kmeans = KMeans(n_clusters=k, random_state=42)
        labels = kmeans.fit_predict(data)
        inertia.append(kmeans.inertia_)
        silhouette_scores.append(silhouette_score(data, labels))

    # Graficar Elbow Method
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.plot(range(2, max_k + 1), inertia, marker='o')
    plt.title('Elbow Method (KMeans)')
    plt.xlabel('Número de Clústeres (k)')
    plt.ylabel('Inercia')

    # Graficar Silhouette Score
    plt.subplot(1, 2, 2)
    plt.plot(range(2, max_k + 1), silhouette_scores, marker='o')
    plt.title('Silhouette Scores (KMeans)')
    plt.xlabel('Número de Clústeres (k)')
    plt.ylabel('Silhouette Score')
    plt.show()

apply_kmeans(scaled_data)
```



3.1.2 Metodo DBSCAN y Agglomerative Clustering

[46]:
```
from sklearn.cluster import DBSCAN, AgglomerativeClustering
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Función para DBSCAN
def apply_dbscan(data, eps=3, min_samples=5):
    dbscan = DBSCAN(eps=eps, min_samples=min_samples)
    labels = dbscan.fit_predict(data)
    return labels

# Función para Agglomerative Clustering
def apply_agglomerative(data, n_clusters=4):
    agglo = AgglomerativeClustering(n_clusters=n_clusters, linkage='ward')
    labels = agglo.fit_predict(data)
    return labels

# Visualización combinada con PCA
def visualize_clustering(data, dbscan_labels, agglo_labels):
    # Reducir dimensiones con PCA
    pca = PCA(n_components=2)
    pca_result = pca.fit_transform(data)

    # Configurar gráficos
    plt.figure(figsize=(12, 5))

    # Gráfico de DBSCAN
```

```
    plt.subplot(1, 2, 1)
    plt.scatter(pca_result[:, 0], pca_result[:, 1], c=dbscan_labels,␣
 ↪cmap='viridis', alpha=0.6)
    plt.title('DBSCAN Clustering (PCA)')
    plt.xlabel('PCA Component 1')
    plt.ylabel('PCA Component 2')
    plt.colorbar(label='Cluster')

    # Gráfico de Agglomerative Clustering
    plt.subplot(1, 2, 2)
    plt.scatter(pca_result[:, 0], pca_result[:, 1], c=agglo_labels,␣
 ↪cmap='viridis', alpha=0.6)
    plt.title('Agglomerative Clustering (PCA)')
    plt.xlabel('PCA Component 1')
    plt.ylabel('PCA Component 2')
    plt.colorbar(label='Cluster')

    plt.tight_layout()
    plt.show()

# Aplicar DBSCAN
dbscan_labels = apply_dbscan(scaled_data, eps=3, min_samples=5)

# Aplicar Agglomerative Clustering
agglo_labels = apply_agglomerative(scaled_data, n_clusters=4)

# Visualizar resultados
visualize_clustering(scaled_data, dbscan_labels, agglo_labels)
```



Interpretacion:

1. K-Means Clustering (revisar inercia, Elbow method):

- Elbow Method: (k=4) o (k=7) son buenos candidatos, con una ligera preferencia por (k=7) debido a su posición más marcada en el codo.
- Silhouette Score:(k=7) es claramente el mejor valor, ya que maximiza la cohesión y separación de los clústeres.
- Basado en ambos criterios, (k=7) es el valor óptimo para (k) en este dataset.

2. DBSCAN Clustering
   - Es más adecuado si el dataset tiene clústeres con densidad variable o formas irregulares.
   - En este caso, parece que los parámetros ((eps=3), (min_samples=5)) no capturan perfectamente la estructura de los datos. Ajustar estos valores podría mejorar los resultados.
3. Agglomerative Clustering
   - Funciona bien cuando se conoce de antemano el número de clústeres ((n=4)).
   - Los clústeres parecen estar más balanceados, pero puede ser menos efectivo si los datos contienen ruido significativo.

1.6 4. Evaluación de los clústeres

1.6.1 4.1 Calcular métricas internas como: Silhouette Score y Davies-Bouldin Score

[48]:
```
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
from sklearn.metrics import silhouette_score, davies_bouldin_score

# Función para calcular las métricas internas de clustering
def evaluate_clustering_metrics(data, labels, method_name):
    if len(set(labels)) > 1:  # Asegurarse de que haya más de un clúster
        silhouette = silhouette_score(data, labels)
        davies_bouldin = davies_bouldin_score(data, labels)
        print(f"{method_name} - Silhouette Score: {silhouette:.2f}")
        print(f"{method_name} - Davies-Bouldin Score: {davies_bouldin:.2f}")
    else:
        print(f"{method_name} no generó clústeres suficientes para calcular␣
 ↪métricas.")

# Aplicar algoritmos de clustering y calcular métricas
def run_clustering_analysis(data):
    # KMeans con k=7
    kmeans = KMeans(n_clusters=7, random_state=42)
    kmeans_labels = kmeans.fit_predict(data)
    evaluate_clustering_metrics(data, kmeans_labels, "KMeans (k=7)")

    # DBSCAN
    dbscan = DBSCAN(eps=1.5, min_samples=5)  # Ajusta estos parámetros según␣
 ↪sea necesario
    dbscan_labels = dbscan.fit_predict(data)
    evaluate_clustering_metrics(data, dbscan_labels, "DBSCAN")

    # Agglomerative Clustering con n_clusters=7
    agglo = AgglomerativeClustering(n_clusters=7)
    agglo_labels = agglo.fit_predict(data)
```

```
    evaluate_clustering_metrics(data, agglo_labels, "Agglomerative Clustering␣
 ↪(n=7)")

# Llamar a la función principal
run_clustering_analysis(scaled_data)
```

```
KMeans (k=7) - Silhouette Score: 0.14
KMeans (k=7) - Davies-Bouldin Score: 2.11
DBSCAN - Silhouette Score: -0.19
DBSCAN - Davies-Bouldin Score: 1.88
Agglomerative Clustering (n=7) - Silhouette Score: 0.11
Agglomerative Clustering (n=7) - Davies-Bouldin Score: 2.28
```

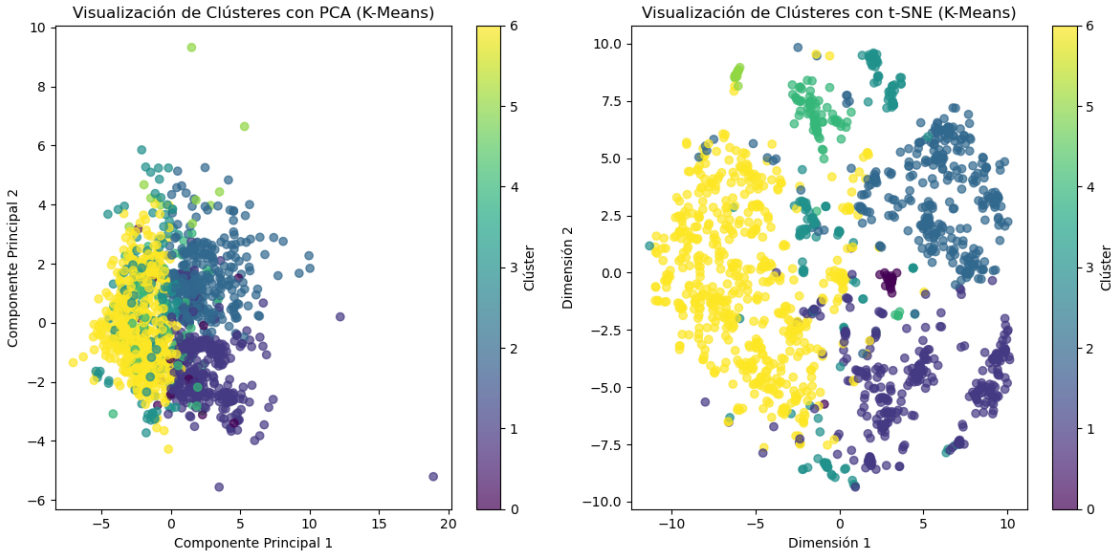
Interpretacion:

1. K-Means (7 clusters):
   - Silhouette Score: 0.14
     - Este valor indica una cohesión y separación moderadamente baja entre los clústeres.
     - Aunque el valor es positivo (indica que los puntos están más cerca de sus clústeres que de otros), es relativamente bajo, lo que sugiere que los clústeres tienen cierta superposición. -Davies-Bouldin Score: 2.11
     - Este valor evalúa la relación entre la dispersión de los puntos dentro de los clústeres y la separación entre ellos.
     - Un valor de 2.11 indica que los clústeres están moderadamente separados, pero no son ideales. Valores más cercanos a 1.0 serían mejores.
   - KMeans con (k=7) ofrece una partición aceptable, siendo el modelo más equilibrado en este análisis. Sin embargo, los clústeres no son perfectamente definidos.
2. DBSCAN:
   - Silhouette Score: -0.19
     - Este valor negativo indica que muchos puntos están mal clasificados, es decir, están más cerca de clústeres vecinos que del propio clúster asignado.
   - Davies-Bouldin Score: 1.88
     - Este valor es más bajo que el de KMeans y Agglomerative Clustering, lo que indica que, a pesar de la mala cohesión, los clústeres identificados por DBSCAN están relativamente bien separados.
   - DBSCAN tiene un desempeño deficiente en este caso debido a su Silhouette Score negativo. Esto sugiere que los parámetros actuales (por ejemplo, eps=1.5 y min_samples=5) no son los ideales para este conjunto de datos. Ajustar estos valores podría mejorar los resultados.
3. Agglomerative Clustering (n=7):
   - Silhouette Score: 0.11
     - Este puntaje es el más bajo entre los modelos con (n=7), indicando que los clústeres tienen una cohesión y separación pobre.
     - Esto podría deberse a que el algoritmo fuerza la creación de (n=7) clústeres, incluso si los datos no tienen una estructura clara que respalde dicha partición.
   - Davies-Bouldin Score: 2.28
     - Un valor más alto que los otros modelos, lo que sugiere que los clústeres tienen mayor dispersión interna y están menos separados entre sí.
   - Agglomerative Clustering con (n=7) tiene el peor desempeño en general, con clústeres

mal definidos, baja cohesión y mala separación.

1.6.2 4.2 Visualizar los clústeres en 2D (usando PCA/t-SNE).

[50]:
```
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

# Aplicar un algoritmo de clustering K-Means
optimal_k = 7
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
kmeans_labels = kmeans.fit_predict(scaled_data)

# Reducción de dimensionalidad con PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)

# Reducción de dimensionalidad con t-SNE
tsne = TSNE(n_components=2, perplexity=30, random_state=42, n_iter=300)  #␣
 ↪Ajustar parámetros según sea necesario
tsne_result = tsne.fit_transform(scaled_data)

# Visualización de los clusters en PCA
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
scatter_pca = plt.scatter(pca_result[:, 0], pca_result[:, 1], c=kmeans_labels,␣
 ↪cmap='viridis', alpha=0.7)
plt.title('Visualización de Clústeres con PCA (K-Means)')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.colorbar(scatter_pca, label='Clúster')

# Visualización de los clusters en t-SNE
plt.subplot(1, 2, 2)
scatter_tsne = plt.scatter(tsne_result[:, 0], tsne_result[:, 1],␣
 ↪c=kmeans_labels, cmap='viridis', alpha=0.7)
plt.title('Visualización de Clústeres con t-SNE (K-Means)')
plt.xlabel('Dimensión 1')
plt.ylabel('Dimensión 2')
plt.colorbar(scatter_tsne, label='Clúster')

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



Interpretacion:

- t-SNE ofrece una visualización más clara y útil para interpretar los clústeres generados por K-Means. Los clústeres están mejor definidos y separados, lo que confirma que (k=7) es un valor razonable.
- PCA es útil para entender la variabilidad lineal de los datos, pero su capacidad para visualizar clústeres es limitada debido a la superposición.

1.6.3 4.3 Analizar las características promedio por clúster (df.groupby('cluster').mean()).

[52]:
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from tabulate import tabulate
import pandas as pd

# --- Preprocesamiento de datos ---
# Escalar los datos para que KMeans funcione correctamente
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df.select_dtypes(include=['number']))  #␣
 ↪Solo columnas numéricas

# --- Aplicar KMeans ---
optimal_k = 7  # Ajusta el número de clústeres según sea necesario
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
labels = kmeans.fit_predict(scaled_data)

# --- Asignar etiquetas al DataFrame original ---
df['cluster'] = labels  # Agregar los clústeres como una nueva columna
```

```python
# --- Calcular características promedio por clúster ---
cluster_means = df.groupby('cluster').mean(numeric_only=True)

# --- Renombrar clústeres como Grupo A, B, ... ---
cluster_names = {idx: f"Grupo {chr(65 + idx)}" for idx in sorted(df['cluster'].
 ↪unique())}
cluster_means['cluster_nombre'] = cluster_means.index.map(cluster_names)
cluster_means = cluster_means.set_index('cluster_nombre')

# --- Mostrar resultados en formato tabulado ---
print("Características promedio por clúster:")
print(tabulate(cluster_means, headers='keys', tablefmt='fancy_grid', floatfmt=".
 ↪2f"))
```

```
Características promedio por clúster:
```

```
  cluster_nombre         Id     MSSubClass     LotFrontage     LotArea
OverallQual     OverallCond     YearBuilt     YearRemodAdd     MasVnrArea
BsmtFinSF1     BsmtFinSF2     BsmtUnfSF     TotalBsmtSF     1stFlrSF
2ndFlrSF     LowQualFinSF     GrLivArea     BsmtFullBath     BsmtHalfBath
FullBath     HalfBath     BedroomAbvGr     KitchenAbvGr     TotRmsAbvGrd
Fireplaces     GarageYrBlt     GarageCars     GarageArea     WoodDeckSF
OpenPorchSF     EnclosedPorch     3SsnPorch     ScreenPorch     PoolArea
MiscVal     MoSold     YrSold     SalePrice
```

```
  Grupo A            616.50          46.36           79.86    12150.09
6.45            5.91       1979.68          1993.14         136.73
611.23           4.36        560.50         1176.09      1337.91       261.82
0.00       1599.73             0.50             0.18         1.68         0.36
2.68             1.00             6.36           0.77         1984.55
2.00         542.32          67.18           48.36              3.18
218.86           0.00         0.00       31.82       7.05    2007.91
214341.68
```

```
  Grupo B            741.83          46.60           69.15     9870.34
5.22            6.00       1966.62          1979.91          55.92
604.35          99.64        287.87           991.86      1065.71        14.72
1.17       1081.59             0.69             0.11         1.07         0.18
2.52             1.01             5.30            0.42         1972.67
1.53         434.20           94.98           26.75             12.57
0.00            3.61         3.66       75.12       6.00    2007.85
141221.19
```

```
  Grupo C            722.44          54.24           74.24    12017.76
6.25            5.86       1960.87          1980.05         122.86
521.06         116.14        513.35          1150.56      1265.76       408.78
7.24       1681.78             0.44             0.09         1.48         0.56
3.03             1.01             6.80            1.15         1966.57
1.82         478.42           56.16           56.81               0.53
0.00          212.20         6.57       82.91       6.67    2007.85
194602.32
```

```
  Grupo D            734.61          48.92           70.40    10120.82
6.91            5.08       1995.50          1997.32         131.65
467.01          33.99        973.53          1474.53      1522.56         3.76
0.00       1526.31             0.45             0.03         1.97         0.06
2.70             1.06             6.44            0.65         1996.24
2.20         580.71          114.07           58.18               9.28
0.09            3.64         0.00        9.80       6.36    2007.82
208495.06
```

```
  Grupo E            728.96          61.66           60.95     8084.53
5.05            5.70       1937.16          1967.12          29.98
157.92          12.68        564.23          734.83       918.52       348.85
13.33       1280.70             0.14             0.03         1.23         0.22
            2.84             1.11             6.12           0.32
1956.03           1.06         278.41          36.76           23.08
57.73          0.40            3.20         0.00       37.00       6.36
2007.84     115479.26
```

```
  Grupo F            738.98          75.82           69.57     9965.37
6.66            5.39       1988.38          1994.71         100.18
304.42          33.73        526.61          864.76       981.60       869.90
8.16       1859.67             0.28             0.05         1.97         0.92
3.28             1.04             7.50           0.73         1991.68
2.06         522.97         108.27           66.35             11.04
0.00            1.10         2.42       47.64       6.31    2007.75
201919.64
```

```
  Grupo G            700.49          43.78           92.37    20133.17
8.33            5.25       1995.98          2001.49         372.33
1043.87          18.98        661.75          1724.59      1758.74       606.53
3.19       2368.46             0.81             0.02         2.10         0.60
3.12             1.00             8.52           1.25         1998.24
2.77         780.65         211.79           89.63             14.46
0.00           13.54        12.58        9.76       6.68    2007.76
354247.91
```

[ ]: