# Proyecto 1 IA

November 5, 2020

### 1 Lectura de la base de datos

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

Se modificaron las variables: BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, FireplaceQu, GarageTyp, GarageFinish, GarageQual, GarageCond, PoolQC, Fence, MiscFeature. Estas variables tenían una categoría llamada "NA", sin embargo, Python al momento de leer esa cadena la interpreta como dato faltante, por lo tanto se reemplazó el nombre de esa categoría por "NoA"

```
[2]: train = pd.read_csv("train.csv", index_col="Id")
  test = pd.read_csv("test.csv", index_col="Id")
  sample_submission = pd.read_csv("sample_submission.csv", index_col="Id")
```

En cuanto a las variables numéricas, si no se registróa valor no pasa nada si se reemplaza con 0 pues las variables numéricas representan longitudes no afecta.

```
[3]: train.fillna(0,inplace=True) test.fillna(0,inplace=True)
```

```
[4]: train.head()
```

[4]:		MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
	Id								
	1	60	RL	65.0	8450	Pave	NoA	Reg	
	2	20	RL	80.0	9600	Pave	NoA	Reg	
	3	60	RL	68.0	11250	Pave	NoA	IR1	
	4	70	RL	60.0	9550	Pave	NoA	IR1	
	5	60	RL	84.0	14260	Pave	NoA	IR1	

```
LandContour Utilities LotConfig
                                     ... PoolArea PoolQC Fence MiscFeature \
Id
                   AllPub
1
           Lvl
                              Inside
                                                  0
                                                       NoA
                                                              NoA
                                                                           NoA
                                      . . .
2
           Lvl
                   AllPub
                                 FR2
                                                  0
                                                       NoA
                                                              NoA
                                                                           NoA
3
           Lvl
                   AllPub
                              Inside
                                                  0
                                                       NoA
                                                              NoA
                                                                           NoA
4
                   AllPub
                             Corner
                                                  0
           Lvl
                                                       NoA
                                                              NoA
                                                                           NoA
5
           Lvl
                   AllPub
                                 FR2
                                                       NoA
                                                  0
                                                              NoA
                                                                           NoA
```

	${ t MiscVal}$	MoSold	YrSold	SaleType	SaleCondition	SalePrice
Ιd						
1	0	2	2008	WD	Normal	208500
2	0	5	2007	WD	Normal	181500
3	0	9	2008	WD	Normal	223500
4	0	2	2006	WD	Abnorml	140000
5	0	12	2008	WD	Normal	250000

[5 rows x 80 columns]

## 2 Revisión de la lectura de los datos

Es importante revisar que el tipo de dato de cada variable sea el adecuado y si no, definirlo de forma correcta. También debe verificarse que las celdas no contengan NaN

```
[5]: print(train.info())
```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1460 entries, 1 to 1460
Data columns (total 80 columns):

columns (cotal	oo corumns).				
Column	Non-Null Count	Dtype			
MSSubClass	1460 non-null	 int64			
MSZoning	1460 non-null	object			
LotFrontage	1460 non-null	float64			
LotArea	1460 non-null	int64			
Street	1460 non-null	object			
Alley	1460 non-null	object			
LotShape	1460 non-null	object			
LandContour	1460 non-null	object			
Utilities	1460 non-null	object			
LotConfig	1460 non-null	object			
LandSlope	1460 non-null	object			
Neighborhood	1460 non-null	object			
Condition1	1460 non-null	object			
Condition2	1460 non-null	object			
BldgType	1460 non-null	object			
HouseStyle	1460 non-null	object			
OverallQual	1460 non-null	int64			
OverallCond	1460 non-null	int64			
YearBuilt	1460 non-null	int64			
YearRemodAdd	1460 non-null	int64			
RoofStyle	1460 non-null	object			
RoofMatl	1460 non-null	object			
Exterior1st	1460 non-null	object			
Exterior2nd	1460 non-null	object			
	Column MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl Exterior1st	MSSubClass 1460 non-null MSZoning 1460 non-null LotFrontage 1460 non-null LotArea 1460 non-null Street 1460 non-null Alley 1460 non-null LotShape 1460 non-null LandContour 1460 non-null Utilities 1460 non-null LotConfig 1460 non-null LandSlope 1460 non-null Neighborhood 1460 non-null Condition1 1460 non-null Condition2 1460 non-null HouseStyle 1460 non-null OverallQual 1460 non-null OverallCond 1460 non-null YearBuilt 1460 non-null RoofStyle 1460 non-null RoofStyle 1460 non-null RoofMatl 1460 non-null Exterior1st 1460 non-null			

24	${\tt MasVnrType}$	1460	non-null	object
25	MasVnrArea	1460	non-null	float64
26	ExterQual	1460	non-null	object
27	ExterCond	1460	non-null	object
28	Foundation	1460	non-null	object
29	BsmtQual	1460	non-null	object
30	BsmtCond	1460	non-null	object
31	BsmtExposure	1460	non-null	object
32	BsmtFinType1	1460	non-null	object
33	BsmtFinSF1	1460	non-null	int64
34	BsmtFinType2	1460	non-null	object
35	BsmtFinSF2	1460	non-null	int64
36	BsmtUnfSF	1460	non-null	int64
37	TotalBsmtSF	1460	non-null	int64
38	Heating	1460	non-null	object
39	HeatingQC	1460	non-null	object
40	CentralAir	1460	non-null	object
41	Electrical	1460	non-null	object
42	1stFlrSF	1460	non-null	int64
43	2ndFlrSF	1460	non-null	int64
44	LowQualFinSF	1460	non-null	int64
45	GrLivArea	1460	non-null	int64
46	BsmtFullBath	1460	non-null	int64
47	BsmtHalfBath	1460	non-null	int64
48	FullBath	1460	non-null	int64
49	HalfBath	1460	non-null	int64
50	BedroomAbvGr	1460	non-null	int64
51	KitchenAbvGr	1460	non-null	int64
52	KitchenQual	1460	non-null	object
53	TotRmsAbvGrd	1460	non-null	int64
54	Functional	1460	non-null	object
55	Fireplaces	1460	non-null	int64
56	FireplaceQu	1460	non-null	object
57	GarageType	1460	non-null	object
58	GarageYrBlt	1460	non-null	float64
59	GarageFinish	1460		object
60	GarageCars	1460	non-null	int64
61	GarageArea	1460	non-null	int64
62	GarageQual	1460	non-null	object
63	GarageCond	1460	non-null	object
64	PavedDrive	1460	non-null	object
65	WoodDeckSF	1460	non-null	int64
66	OpenPorchSF	1460	non-null	int64
67	EnclosedPorch	1460		int64
68	3SsnPorch	1460	non-null	int64
69	ScreenPorch	1460	non-null	int64
70	PoolArea	1460	non-null	int64
71	PoolQC	1460	non-null	object
			<b></b>	

```
72 Fence
                   1460 non-null
                                   object
 73 MiscFeature
                   1460 non-null
                                   object
                   1460 non-null
 74 MiscVal
                                   int64
75 MoSold
                   1460 non-null
                                   int64
                  1460 non-null
 76 YrSold
                                   int64
                   1460 non-null
    SaleType
                                   object
    SaleCondition 1460 non-null
                                   object
 79 SalePrice
                   1460 non-null
                                   int64
dtypes: float64(3), int64(34), object(43)
memory usage: 923.9+ KB
None
```

#### 2.1 Var 00 MSSubClass

Por definición de los datos la variable MSSubClass aunque toma valores enteros realmente es una variable categórica, pues indica el tipo de vivienda involucrada en la venta, por lo tanto, debe trabajarse como variable catégorica y no numérica.

```
[6]: train['MSSubClass'] = train['MSSubClass'].astype('category')
    print(train['MSSubClass'].dtype)
    print(train['MSSubClass'].isnull().sum())
    ## este cambio tambien lo hacemos en test
    test['MSSubClass'] = test['MSSubClass'].astype('category')
category
```

### 2.2 Var 01 MSZoning

Por definición esta variable es categórica, así que está bien definida.

```
[7]: print(train['MSZoning'].dtype)
  print(train['MSZoning'].isnull().sum())

object
0
```

#### 2.3 Var 02 LotFrontage

Esta variable representa una medida en pies de la calle conectada a la propiedad, por lo tanto, como flotante **está bien definida**.

```
[8]: print(train['LotFrontage'].dtype)
  print(train['LotFrontage'].isnull().sum())

float64
0
```

### 2.4 Var 03 LotArea

Representa la medida del lote en pies cuadrados, si se identificó como entero significa que ningún registro tiene una medida decimal, lo que es raro pero ya ni modo, el tipo de dato está bien definido

```
[9]: print(train['LotArea'].dtype)
  print(train['LotArea'].isnull().sum())

int64
0
```

#### 2.5 Var 04 Street

Igualmente **está bien definida**, representa si calle está pavimentada o es grava.

```
[10]: print(train['Street'].dtype)
    print(train['Street'].isnull().sum())

    object
    0
```

### 2.6 Var 05 Alley

Igualmente **está bien definida**, representa si el acceso a la calle está pavimentada o es grava.

```
[11]: print(train['Alley'].dtype)
    print(train['Alley'].isnull().sum())

    object
    0
```

### 2.7 Var 06 LotShape

Representa la forma del lote, está bien definida.

```
[12]: print(train['LotShape'].dtype)
print(train['LotShape'].isnull().sum())

object
0
```

### 2.8 Var 07 LandContour

Representa la inflación de la propiedad, está bien definida.

```
[13]: print(train['LandContour'].dtype)
print(train['LandContour'].isnull().sum())
```

```
object
0
```

#### 2.9 Var 08 Utilities

Variable categórica que indica los serivios que tiene la propiedad (agua, luz, gas, etc). Está bien definida.

```
[14]: print(train['Utilities'].dtype)
print(train['Utilities'].isnull().sum())

object
0
```

### 2.10 Var 09 LotConfig

Representa si la propiedad está en una cerrada, sobre la avenida, en esquina o es un predio dentro de otro, etc. **Está bien definida**.

```
[15]: print(train['LotConfig'].dtype)
  print(train['LotConfig'].isnull().sum())

  object
  0
```

### 2.11 Var 10 LandSlope

Indica si la propiedad está sobre terreno plano o si tiene cierta inclinación. Está bien definido.

```
[16]: print(train['LandSlope'].dtype)
print(train['LandSlope'].isnull().sum())

object
0
```

### 2.12 Var 11 Neighborhood

Localización física dentro de los límites de Ames City. Está bien definida.

```
[17]: print(train['Neighborhood'].dtype)
print(train['Neighborhood'].isnull().sum())

object
0
```

#### 2.13 Var 12 Condition1

Proximidad a varias condiciones (cerca a una avenida principal, etc.) Está bien definida.

```
[18]: print(train['Condition1'].dtype)
print(train['Condition1'].isnull().sum())

object
0
```

#### 2.14 Var 13 Condition2

Igualmente a la anterior es categórica, está bien definida.

```
[19]: print(train['Condition2'].dtype)
print(train['Condition2'].isnull().sum())

object
0
```

### 2.15 Var 14 BldgType

Tipo de vivienda (1 familia, originalmente contruida para 1 familia y adaptada para 2, duplex, etc.). Está bien definida.

```
[20]: print(train['BldgType'].dtype)
print(train['BldgType'].isnull().sum())

object
0
```

### 2.16 Var 15 HouseStyle

Estilo de vivienda. Está bien definida.

```
[21]: print(train['HouseStyle'].dtype)
print(train['HouseStyle'].isnull().sum())

object
0
```

### 2.17 Var 16 OverallQual

Esta variable representa una calificación en la calidad de los materiales y acabados de la clase, aunque es categórica es una variable ordinal, entonces como entero está bien definida pues la calificación, el valor, sí proporcionan información, el 10 es Muy Excelente y el 1 en Muy Pobre. está bien.

```
[22]: print(train['OverallQual'].dtype)
print(train['OverallQual'].isnull().sum())

int64
0
```

#### 2.18 Var 17 Overall Cond

Esta calificación representa una calificación en la condición de la casa, al igual que la anterior, **está** bien definida.

```
[23]: print(train['OverallCond'].dtype)
print(train['OverallCond'].isnull().sum())

int64
0
```

#### 2.19 Var 18 YearBuilt

Año de construcción, está bien definida.

```
[24]: print(train['YearBuilt'].dtype)
print(train['YearBuilt'].isnull().sum())

int64
0
```

#### 2.20 Var 19 YearRemodAdd

Añe de remodelación, mismo año que construcción sino ha sido remodelada, está bien definida.

```
[25]: print(train['YearRemodAdd'].dtype)
print(train['YearRemodAdd'].isnull().sum())

int64
0
```

### 2.21 Var 20 RoofStyle

Tipo de techo, variable categ´rotica, está bien definida.

```
[26]: print(train['RoofStyle'].dtype)
print(train['RoofStyle'].isnull().sum())

object
0
```

#### 2.22 Var 21 RoofMatl

Material del techo, está bien definida.

```
[27]: print(train['RoofMatl'].dtype)
print(train['RoofMatl'].isnull().sum())

object
0
```

#### 2.23 Var 22 Exterior1st

Cuvierta del exterior de la casa, variable categórica, está bien definida.

```
[28]: print(train['Exterior1st'].dtype)
print(train['Exterior1st'].isnull().sum())

object
0
```

#### 2.24 Var 23 Exterior2nd

Si es que tiene otro material la fachada de la casa, variable categórica, está bien definida.

```
[29]: print(train['Exterior2nd'].dtype)
print(train['Exterior2nd'].isnull().sum())

object
0
```

### 2.25 Var 24 MasVnrType

Tipo de revestimiento de mamposería, está bien definida.

```
[30]: print(train['MasVnrType'].dtype)
print(train['MasVnrType'].isnull().sum())

object
0
```

#### 2.26 Var 25 MasVnrArea

Área, en pies cuadrados, de recubriminetos de mampostería. Está bien definida.

```
[31]: print(train['MasVnrArea'].dtype)
print(train['MasVnrArea'].isnull().sum())

float64
0
```

### 2.27 Var 26 ExterQual

Variable categorica de la calidad del material exterior. está bien definida.

```
[32]: print(train['ExterQual'].dtype)
print(train['ExterQual'].isnull().sum())

object
0
```

#### 2.28 Var 27 ExterCond

Evalúa la condición de lo materiales del exterior, variable categótica. Está bien definida.

```
[33]: print(train['ExterCond'].dtype)
print(train['ExterCond'].isnull().sum())

object
output
```

### 2.29 Var 28 Foundation

Tipo de fundamento, está bien definida.

```
[34]: print(train['Foundation'].dtype)
print(train['Foundation'].isnull().sum())

object
0
```

### 2.30 Var 29 BsmtQual

Evalúa el grosor de los fundamentos, variable categórica, está bien definida.

```
[35]: print(train['BsmtQual'].dtype)
print(train['BsmtQual'].isnull().sum())

object
0
```

#### 2.31 Var 30 BsmtCond

Condición general de los simientos, está bien definida.

```
[36]: print(train['BsmtCond'].dtype)
print(train['BsmtCond'].isnull().sum())

object
0
```

### 2.32 Var 31 BsmtExposure

Se refiere a los miros de la entrada o jardín, **está bien definida**.

```
[37]: print(train['BsmtExposure'].dtype)
print(train['BsmtExposure'].isnull().sum())

object
0
```

### 2.33 Var 32 BsmtFinType1

Calificación de los simientos terminados, está bien definida.

```
[38]: print(train['BsmtFinType1'].dtype)
print(train['BsmtFinType1'].isnull().sum())

object
o
```

#### 2.34 Var 33 BsmtFinSF1

Metros cuadrados terminado, está bien definida.

```
[39]: print(train['BsmtFinSF1'].dtype)
print(train['BsmtFinSF1'].isnull().sum())

int64
0
```

### 2.35 Var 34 BsmtFinType2

rango de los simientos del área terminada, está bien definida.

```
[40]: print(train['BsmtFinType2'].dtype)
print(train['BsmtFinType2'].isnull().sum())

object
0
```

#### 2.36 Var 35 BsmtFinSF2

pies cuadrados terminados, está bien definida.

```
[41]: print(train['BsmtFinSF2'].dtype)
print(train['BsmtFinSF2'].isnull().sum())

int64
0
```

#### 2.37 Var 36 BsmtUnfSF

pues cuadrados in terminar de área de simientos, está bien definida.

```
[42]: print(train['BsmtUnfSF'].dtype)
print(train['BsmtUnfSF'].isnull().sum())

int64
```

#### 2.38 Var 37 TotalBsmtSF

pies cuadrados totales de área de simientos, está bien definida.

```
[43]: print(train['TotalBsmtSF'].dtype)
print(train['TotalBsmtSF'].isnull().sum())

int64
0
```

### 2.39 Var 38 Heating

tipo de califección, está bien definida.

```
[44]: print(train['Heating'].dtype)
print(train['Heating'].isnull().sum())

object
0
```

### 2.40 Var 39 HeatingQC

Calidad de la caleffación, está bien definida.

```
[45]: print(train['HeatingQC'].dtype)
print(train['HeatingQC'].isnull().sum())

object
0
```

### 2.41 Var 40 CentralAir

si/no tiene aire acondicionado centra, está bien definida.

```
[46]: print(train['CentralAir'].dtype)
print(train['CentralAir'].isnull().sum())

object
```

### 2.42 Var 41 Electrical

Tipo de sistema eléctrico, está bien definida.

```
[47]: print(train['Electrical'].dtype)
print(train['Electrical'].isnull().sum())

object
0
```

#### 2.43 Var 42 1stFlrSF

Pies cuadrados del primer piso, está bien definida.

```
[48]: print(train['1stFlrSF'].dtype)
print(train['1stFlrSF'].isnull().sum())

int64
0
```

#### 2.44 Var 43 2ndFlrSF

Pies cuadrados del segundo piso, está bien definida.

```
[49]: print(train['2ndFlrSF'].dtype)
print(train['2ndFlrSF'].isnull().sum())

int64
0
```

### 2.45 Var 44 LowQualFinSF

pies cuadrados de baja calidad, está bien definida.

```
[50]: print(train['LowQualFinSF'].dtype)
print(train['LowQualFinSF'].isnull().sum())

int64
0
```

#### 2.46 Var 45 GrLivArea

pies cuadrados de superficie habitable, está bien definida.

```
[51]: print(train['GrLivArea'].dtype)
print(train['GrLivArea'].isnull().sum())

int64
0
```

#### 2.47 Var 46 BsmtFullBath

Baños completos en el sótano, está bien definida.

```
[52]: print(train['BsmtFullBath'].dtype)
print(train['BsmtFullBath'].isnull().sum())

int64
0
```

### 2.48 Var 47 BsmtHalfBath

Medios baños en el sótano, está bien definida.

```
[53]: print(train['BsmtHalfBath'].dtype)
print(train['BsmtHalfBath'].isnull().sum())

int64
0
```

#### 2.49 Var 48 FullBath

Baños completos, está bien definida.

```
[54]: print(train['FullBath'].dtype)
print(train['FullBath'].isnull().sum())

int64
0
```

#### 2.50 Var 49 HalfBath

Medios baños, está bien definida.

```
[55]: print(train['HalfBath'].dtype)
print(train['HalfBath'].isnull().sum())

int64
0
```

#### 2.51 Var 50 BedroomAbvGr

Recamaras sin incluir las de sótano, está bien definida.

```
[56]: print(train['BedroomAbvGr'].dtype)
print(train['BedroomAbvGr'].isnull().sum())

int64
0
```

### 2.52 Var 51 Kitchen Abv Gr

Numero de cocinas en la casa, está bien definida.

```
[57]: print(train['KitchenAbvGr'].dtype)
print(train['KitchenAbvGr'].isnull().sum())

int64
```

### 2.53 Var 52 KitchenQual

Calidad de la cocina, está bien definida.

```
[58]: print(train['KitchenQual'].dtype)
print(train['KitchenQual'].isnull().sum())

object
0
```

### 2.54 Var 53 TotRmsAbvGrd

Habitaciones totales sin incluir baños, está bien definida.

```
[59]: print(train['TotRmsAbvGrd'].dtype)
print(train['TotRmsAbvGrd'].isnull().sum())

int64
0
```

#### 2.55 Var 54 Functional

Funcionalidad de la casa, está bien definida.

```
[60]: print(train['Functional'].dtype)
print(train['Functional'].isnull().sum())

object
0
```

### 2.56 Var 55 Fireplaces

Número de chimeneas, está bien definida.

```
[61]: print(train['Fireplaces'].dtype)
print(train['Fireplaces'].isnull().sum())

int64
0
```

### 2.57 Var 56 FireplaceQu

Calidad de las chimeneas, está bien definida.

```
[62]: print(train['FireplaceQu'].dtype)
print(train['FireplaceQu'].isnull().sum())

object
```

### 2.58 Var 57 GarageType

ubicación del gargae, está bien definida.

```
[63]: print(train['GarageType'].dtype)
print(train['GarageType'].isnull().sum())

object
0
```

### 2.59 Var 58 GarageYrBlt

año en que el garage se construyó, está bien definida.

```
[64]: print(train['GarageYrBlt'].dtype)
print(train['GarageYrBlt'].isnull().sum())

float64
0
```

### 2.60 Var 59 GarageFinish

estatus del garage, está bien definida.

```
[65]: print(train['GarageFinish'].dtype)
print(train['GarageFinish'].isnull().sum())

object
0
```

### 2.61 Var 60 GarageCars

Capacidad de carros en el garage, está bien definida.

```
[66]: print(train['GarageCars'].dtype)
print(train['GarageCars'].isnull().sum())

int64
0
```

### 2.62 Var 61 GarageArea

pies cuadrados del garage, está bien definida.

```
[67]: print(train['GarageArea'].dtype)
print(train['GarageArea'].isnull().sum())

int64
0
```

### 2.63 Var 62 GarageQual

Calidad del garage, está bien definida.

```
[68]: print(train['GarageQual'].dtype)
print(train['GarageQual'].isnull().sum())

object
output
```

### 2.64 Var 63 GarageCond

condición del garage, está bien definida.

```
[69]: print(train['GarageCond'].dtype)
print(train['GarageCond'].isnull().sum())

object
0
```

### 2.65 Var 64 PavedDrive

Pavimentado, variable categórica, estábien definida.

```
[70]: print(train['PavedDrive'].dtype)
print(train['PavedDrive'].isnull().sum())

object
```

#### 2.66 Var 65 WoodDeckSF

pies cuadrados de área decorada con madera, está bien definida.

```
[71]: print(train['WoodDeckSF'].dtype)
print(train['WoodDeckSF'].isnull().sum())

int64
0
```

### 2.67 Var 66 OpenPorchSF

pies cuadrados de porch abierto, está bien definida.

```
[72]: print(train['OpenPorchSF'].dtype)
print(train['OpenPorchSF'].isnull().sum())

int64
0
```

### 2.68 Var 67 EnclosedPorch

pies cuadrados de porch cerrado, está bien definida.

```
[73]: print(train['EnclosedPorch'].dtype)
print(train['EnclosedPorch'].isnull().sum())

int64
0
```

#### 2.69 Var 68 3SsnPorch

pies cuadrados de proch de tres estaciones Q\_Q, ,está bien definida.

```
[74]: print(train['3SsnPorch'].dtype)
print(train['3SsnPorch'].isnull().sum())

int64
0
```

#### 2.70 Var 69 ScreenPorch

pies cuadrados de fachada del proch, está bien definida.

```
[75]: print(train['ScreenPorch'].dtype)
print(train['ScreenPorch'].isnull().sum())

int64
0
```

#### 2.71 Var 70 PoolArea

pies cuadrados de la superficie de la alberca, está bien definida.

```
[76]: print(train['PoolArea'].dtype)
print(train['PoolArea'].isnull().sum())

int64
0
```

### 2.72 Var 71 PoolQC

Calidad de la alberca, está bien definida.

```
[77]: print(train['PoolQC'].dtype)
print(train['PoolQC'].isnull().sum())

object
0
```

#### 2.73 Var 72 Fence

calidad de la cerca, está bien definida.

```
[78]: print(train['Fence'].dtype)
print(train['Fence'].isnull().sum())

object
```

#### 2.74 Var 73 MiscFeature

articulos miscelaneos, es 'ta bien definida.

```
[79]: print(train['MiscFeature'].dtype)
print(train['MiscFeature'].isnull().sum())

object
```

#### 2.75 Var 74 MiscVal

valor de los articulos miscelaneos, está bien definida.

```
[80]: print(train['MiscVal'].dtype)
print(train['MiscVal'].isnull().sum())

int64
0
```

### 2.76 Var 75 MoSold

Mes de venta, está bien definida.

```
[81]: print(train['MoSold'].dtype)
print(train['MoSold'].isnull().sum())

int64
```

#### 2.77 Var 76 YrSold

0

año de venta, está bien definida.

```
[82]: print(train['YrSold'].dtype)
print(train['YrSold'].isnull().sum())

int64
0
```

### 2.78 Var 77 SaleType

tipo de venta, está bien definida

```
[83]: print(train['SaleType'].dtype)
print(train['SaleType'].isnull().sum())

object
0
```

### 2.79 Var 78 SaleCondition

condiciones en que se dio la venta, está bien definida

```
[84]: print(train['SaleCondition'].dtype)
print(train['SaleCondition'].isnull().sum())

object
0
```

#### 2.80 Var 79 SalePrice

Precio de venta, está bien definida.

```
[85]: print(train['SalePrice'].dtype)
print(train['SalePrice'].isnull().sum())

int64
0
```

### 2.81 Resumen y comentarios

La variable 0 "MSSubClass", originalmente identificada como entero, es en realidad categórica.

Existe inconsistencia en cuanto a la categorización pues hay variables que evalúan calidad como "ExterQual" y "ExterCond" cuyas categórias son:

```
Ex Excellent
Gd Good
TA Average/Typical
Fa Fair
Po Poor
```

Sin embargo, "OverallCond" es:

- 10 Very Excellent9 Excellent
- 8 Very Good
- 7 Good
- 6 Above Average

- 5 Average
- 4 Below Average
- 3 Fair
- 2 Poor
- 1 Very Poor

Aunque ambas "miden" o "califican" unas decidieron hacerlo catgóricamente y otras numéricamente.

Se decidió dejar "OverallCond" como numérica.

# 3 Breve análisis descriptivo de los datos

### 3.1 Variables categóricas

Primeramente, si se realiza un histograma se puede inferir acerca de la distribución de los datos, por ejemplo, a continuación se realiza para aquellas variables que son categóricas, esto porque las variables numéricas sería mejor analizarlo con una matriz de correlación.

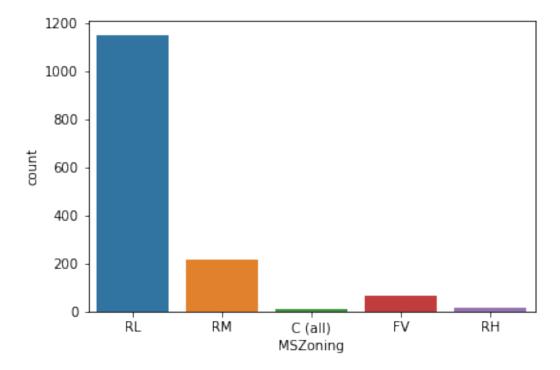
```
[86]: #Las variables que son object
      import numpy as np
      import copy
      trainObj = train.select_dtypes(include=['object' or 'category']).copy()
      #también lo hacemos para test
      testObj = test.select_dtypes(include=['object'or 'category']).copy()
      trainObj.head()
[86]:
         MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope \
      Ιd
      1
                RL
                                                                                    Gtl
                     Pave
                             NoA
                                                   Lvl
                                                           AllPub
                                                                      Inside
                                      Reg
      2
                RL
                     Pave
                             NoA
                                      Reg
                                                   Lvl
                                                           AllPub
                                                                         FR2
                                                                                    Gtl
      3
                RL
                             NoA
                                      IR1
                                                           AllPub
                                                                      Inside
                                                                                    Gtl
                     Pave
                                                   Lvl
      4
                RL
                     Pave
                             NoA
                                      IR1
                                                   Lvl
                                                           AllPub
                                                                      Corner
                                                                                   Gtl
      5
                RL
                     Pave
                            NoA
                                      IR1
                                                   Lvl
                                                           AllPub
                                                                         FR2
                                                                                    Gtl
                                    ... GarageType GarageFinish GarageQual GarageCond \
         Neighborhood Condition1
      Ιd
      1
               CollgCr
                                             Attchd
                                                              RFn
                                                                           TA
                                                                                       TA
                              Norm
                                    . . .
      2
                                                                           TΑ
                                                                                       TΑ
               Veenker
                                             Attchd
                                                              RFn
                             Feedr
      3
                                                                           TΑ
                                                                                       TA
               CollgCr
                              Norm
                                             Attchd
                                                              RFn
                                    . . .
      4
               Crawfor
                              Norm
                                    . . .
                                             Detchd
                                                              Unf
                                                                           TA
                                                                                       TA
      5
               NoRidge
                              Norm
                                   . . .
                                             Attchd
                                                              RFn
                                                                           TA
                                                                                       TA
         PavedDrive PoolQC Fence MiscFeature SaleType SaleCondition
      Id
      1
                   Y
                        NoA
                               NoA
                                            NoA
                                                      WD
                                                                 Normal
      2
                   Y
                                            NoA
                                                                 Normal
                        NoA
                               NoA
                                                      WD
```

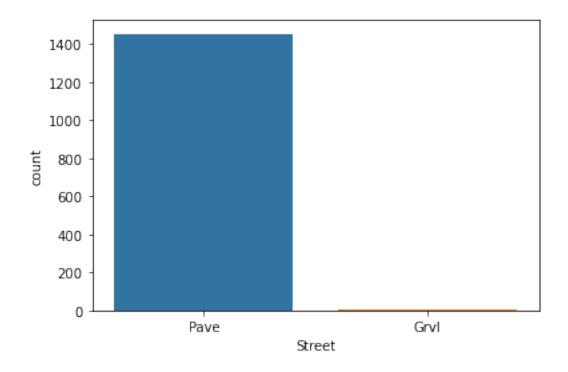
3	Y	NoA	NoA	NoA	WD	Normal
4	Y	NoA	NoA	NoA	WD	Abnorml
5	Y	NoA	NoA	NoA	WD	Normal

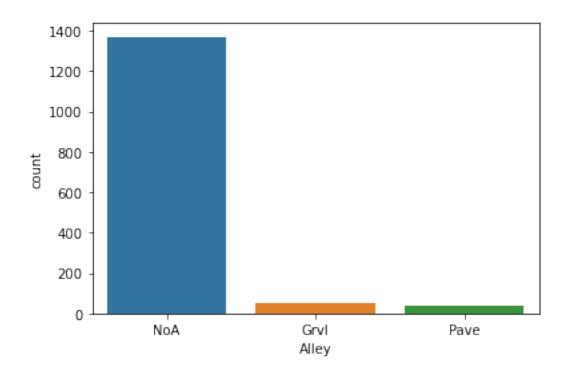
[5 rows x 43 columns]

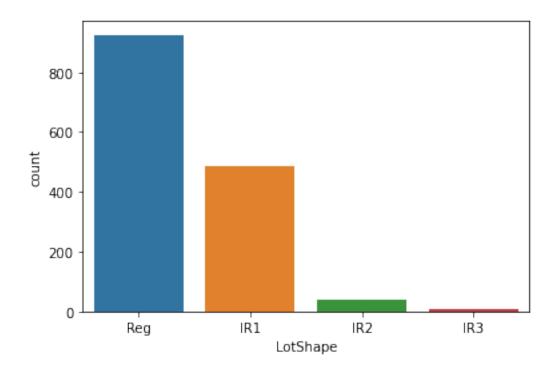
```
[87]: %matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
for i, col in enumerate(trainObj.columns):
    plt.figure(i)
    sns.countplot(x=col, data=trainObj)
```

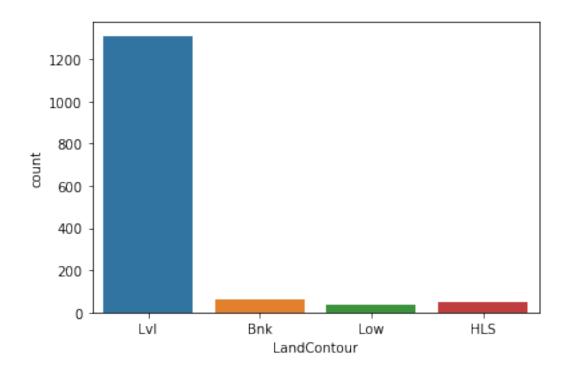
<ipython-input-87-d937b5dcecfe>:5: RuntimeWarning: More than 20 figures have
been opened. Figures created through the pyplot interface
(`matplotlib.pyplot.figure`) are retained until explicitly closed and may
consume too much memory. (To control this warning, see the rcParam
`figure.max\_open\_warning`).
 plt.figure(i)

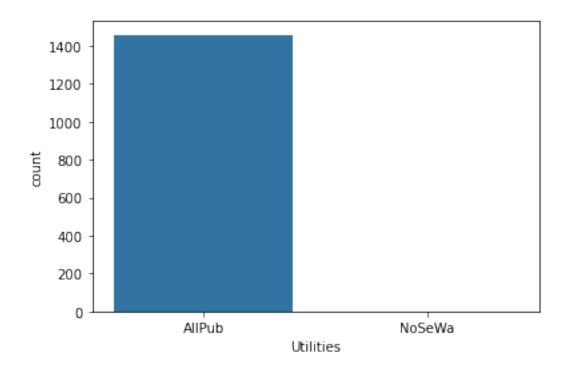


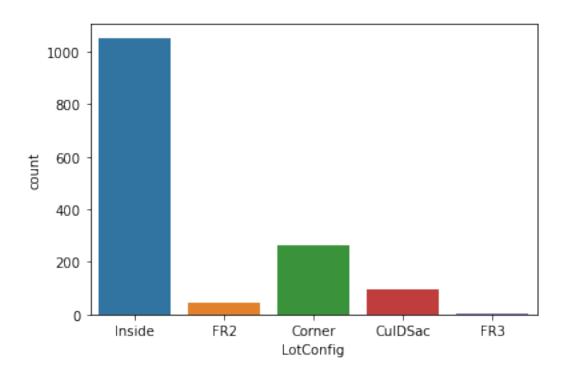


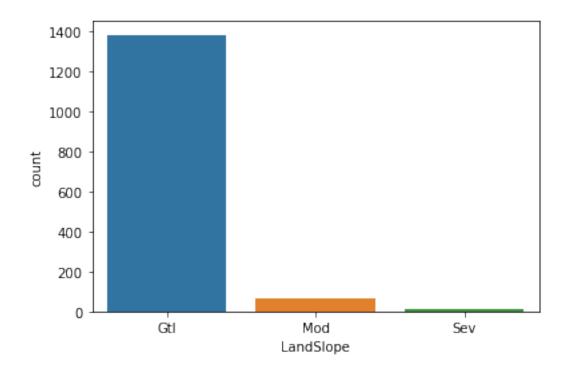


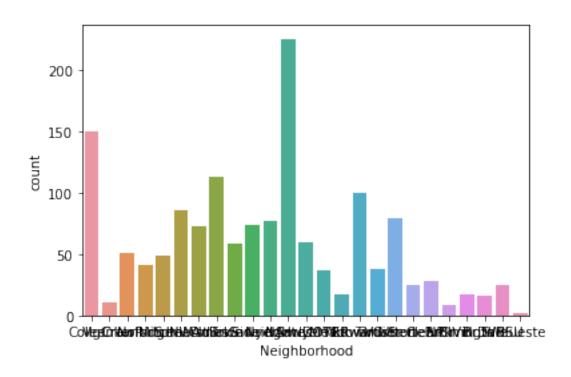


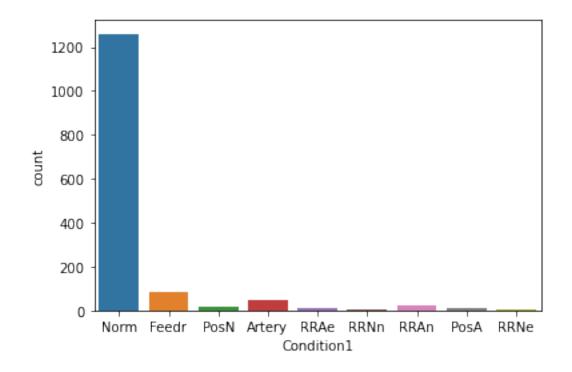


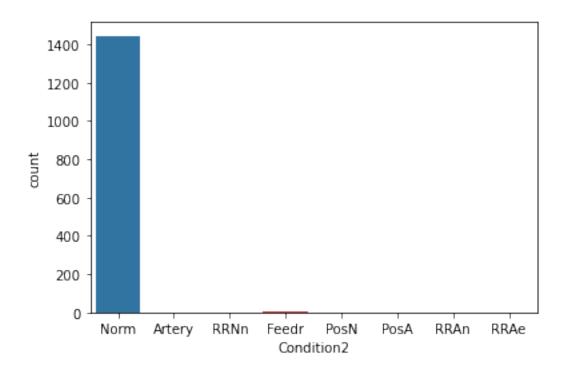


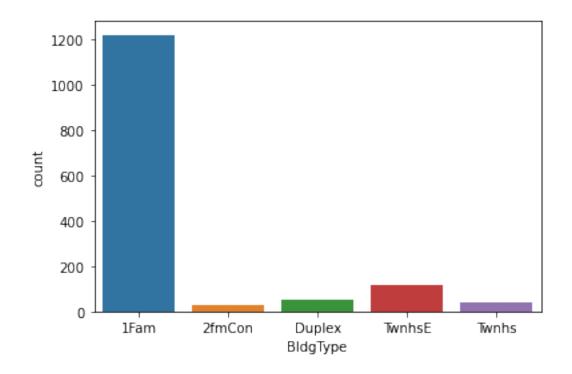


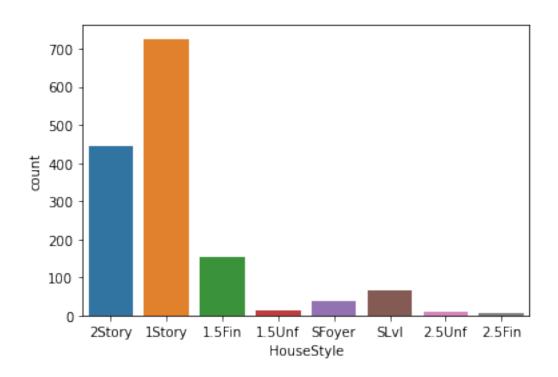


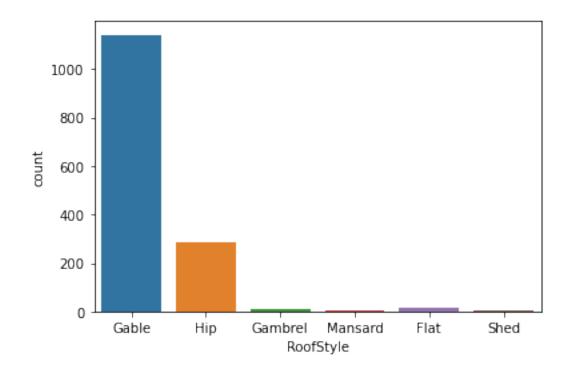


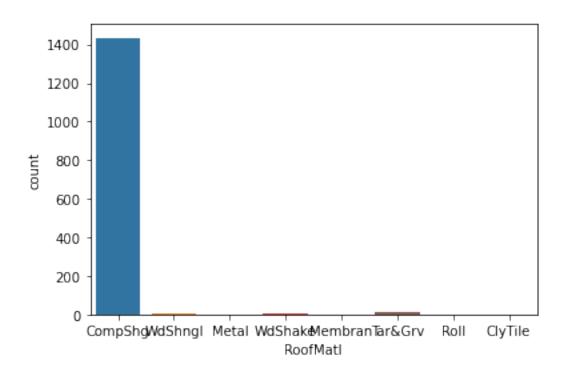


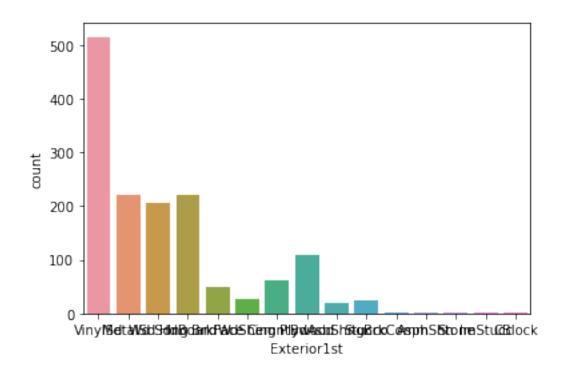


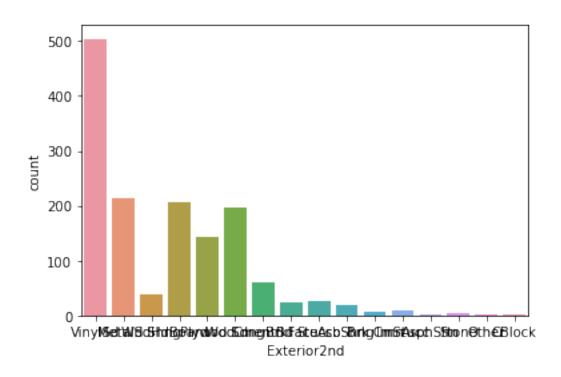


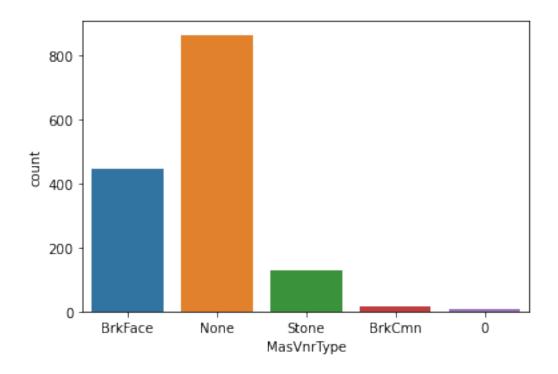


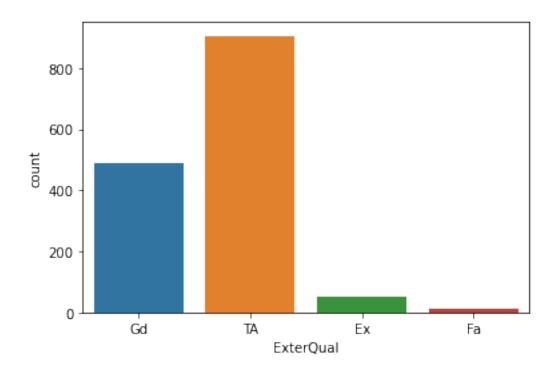


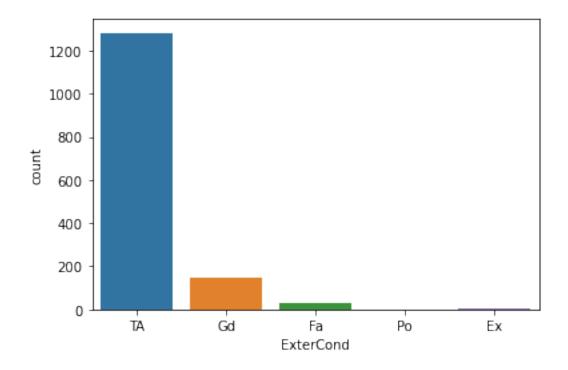


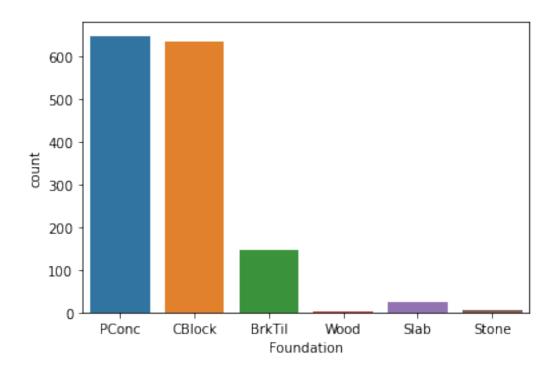


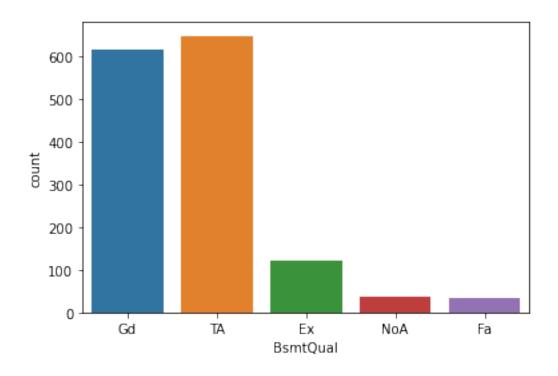


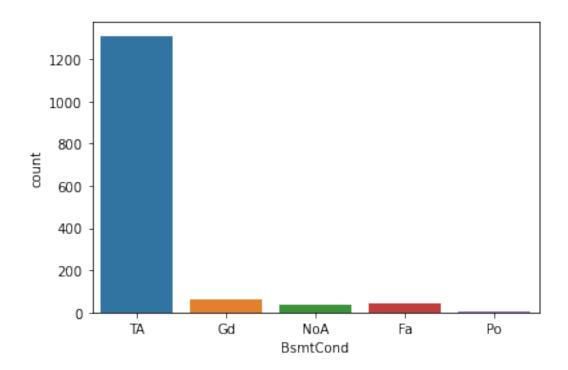


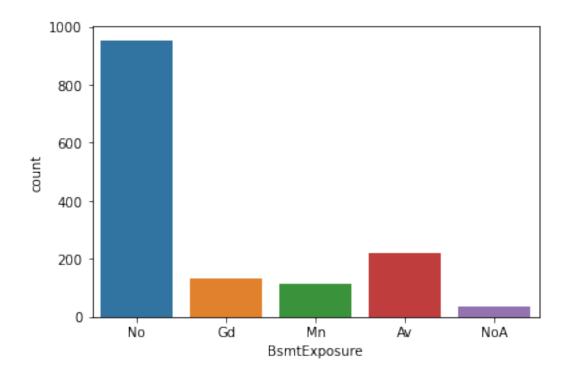


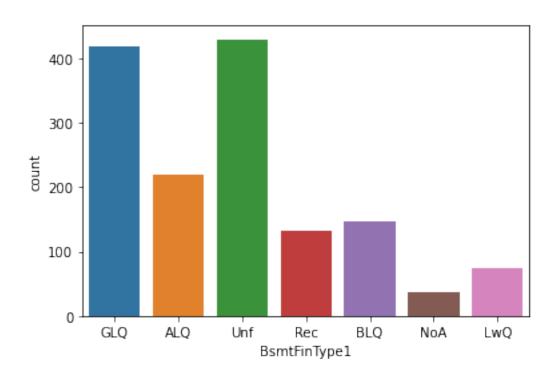


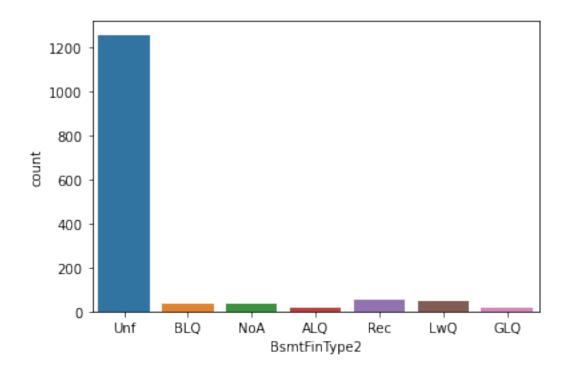


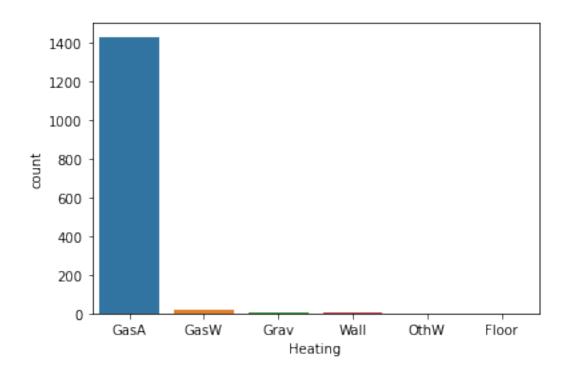


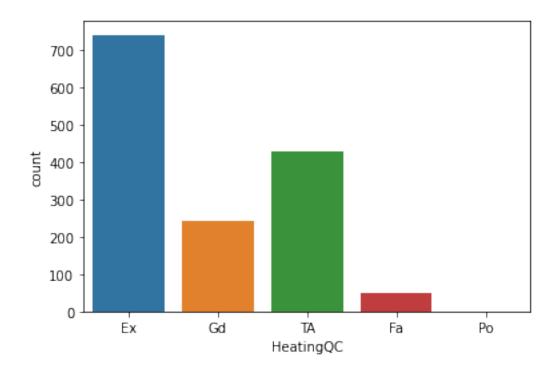


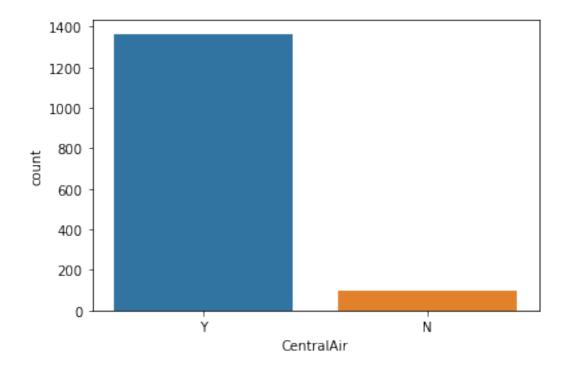


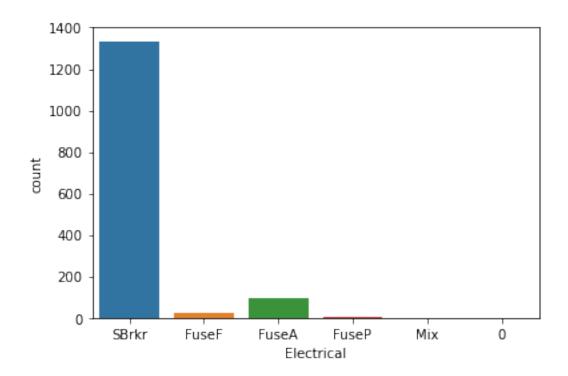


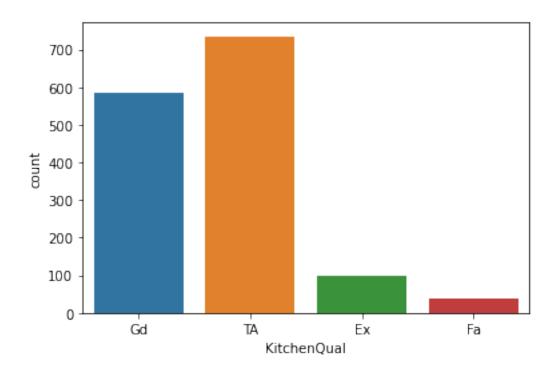


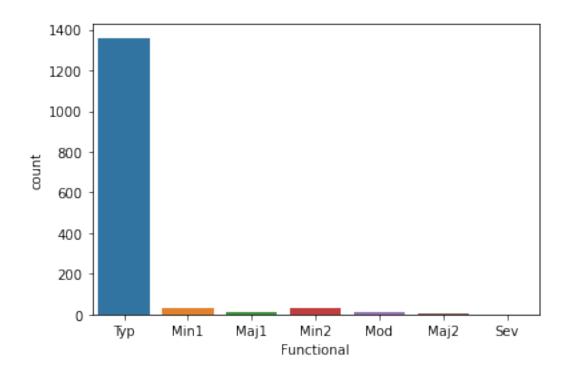


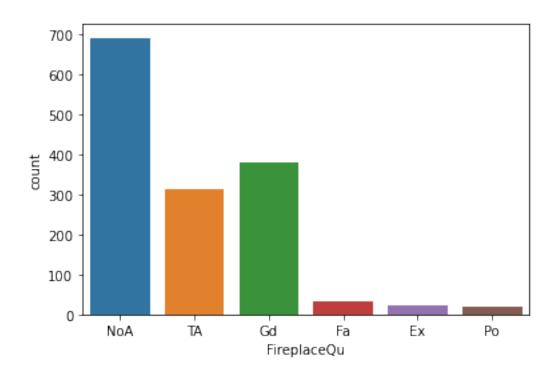


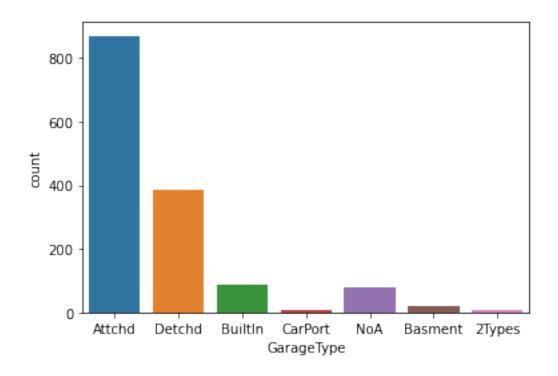


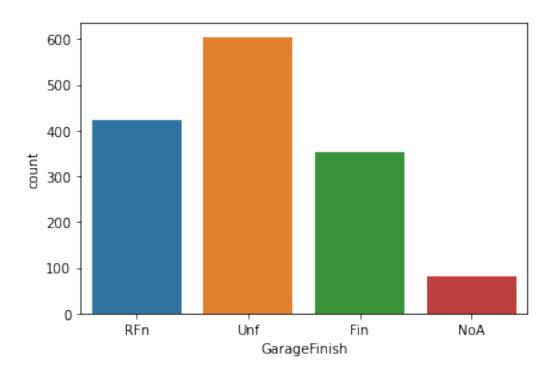


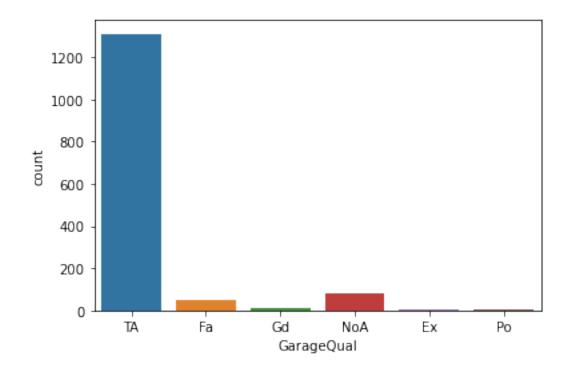


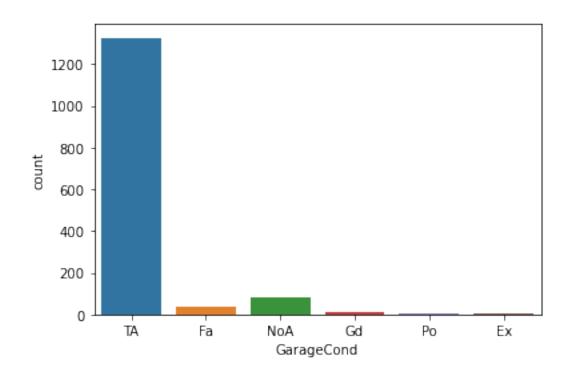


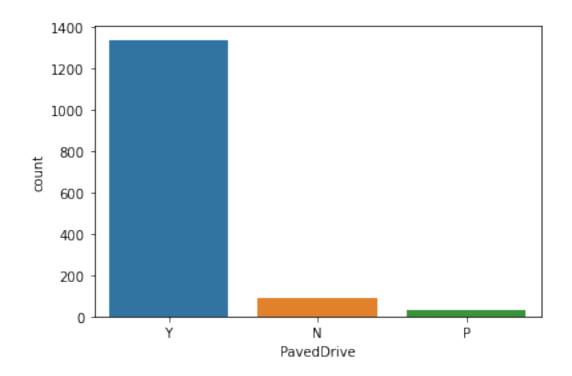


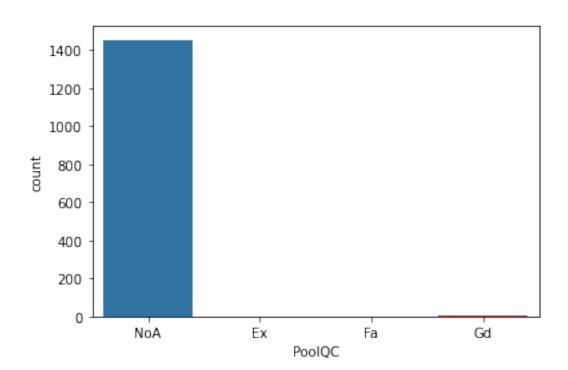


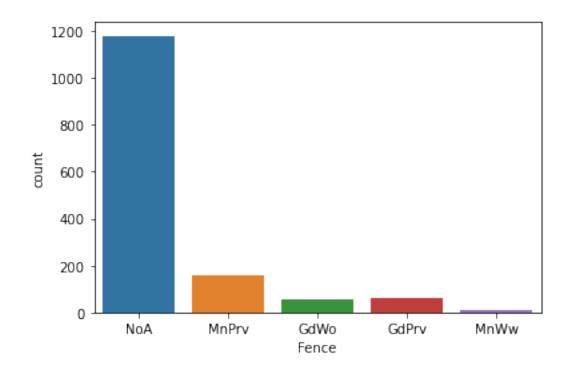


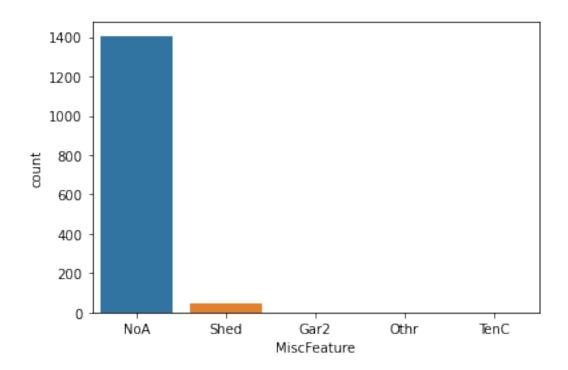


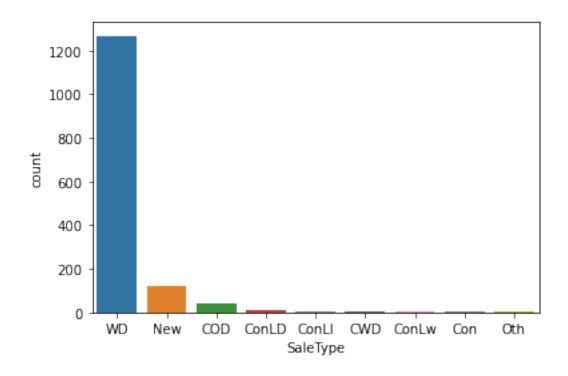


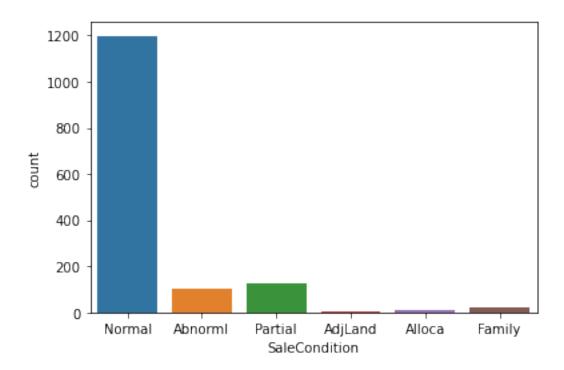












#### 3.1.1 Conclusión

[88]: #Variables explicativas categóricas

Existen varias variables que no tienen mucho sentido, por ejemplo "street"en la que casi el total de las casas pertencen a la categoría "Pave", o la variable "Utilities" sucede lo mismo, todas las casas son "AllPub", y también con "Condition2", "RoofMatl", "BsmtFinType2", "Heating", "Functional" y "PoolQC". Seguramente el método LASSO hará que estas variables tengan coeficiente cero, pues realmente no describen/modelan el precio de venta.

```
XtrainCat = trainObj.copy()
       #también lo hacemos para test
      XtestCat = test[XtrainCat.columns]
      XtrainCat
[88]:
            MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope \
      Τd
                                 NoA
                                                                 AllPub
                                                                            Inside
                                                                                           Gtl
      1
                   RL
                         Pave
                                           Reg
                                                         Lvl
      2
                   RL
                         Pave
                                 NoA
                                           Reg
                                                         Lvl
                                                                 AllPub
                                                                                FR2
                                                                                           Gtl
      3
                                                                 AllPub
                   RL
                         Pave
                                 NoA
                                           IR1
                                                         Lvl
                                                                            Inside
                                                                                           Gtl
      4
                   RL
                                           IR1
                                                         Lvl
                                                                 AllPub
                                                                            Corner
                                                                                           Gtl
                         Pave
                                 NoA
      5
                   RL
                                           IR1
                                                         Lvl
                                                                 AllPub
                                                                                FR2
                                                                                           Gtl
                         Pave
                                 NoA
       . . .
                  . . .
                                 . . .
                                           . . .
                                                         . . .
                                                                                           . . .
      1456
                   RL
                         Pave
                                 NoA
                                           Reg
                                                         Lvl
                                                                 AllPub
                                                                            Inside
                                                                                           Gtl
      1457
                   RL
                                 NoA
                                                         Lvl
                                                                 AllPub
                                                                            Inside
                                                                                           Gtl
                         Pave
                                           Reg
      1458
                   RL
                        Pave
                                 NoA
                                           Reg
                                                         Lvl
                                                                 AllPub
                                                                            Inside
                                                                                           Gtl
      1459
                   RL
                         Pave
                                                                 AllPub
                                                                            Inside
                                                                                           Gtl
                                 NoA
                                           Reg
                                                         Lvl
                   RL
                                                                 AllPub
      1460
                                                         Lvl
                                                                            Inside
                                                                                           Gtl
                        Pave
                                 NoA
                                           Reg
            Neighborhood Condition1
                                         ... GarageType GarageFinish GarageQual
      Ιd
                                         . . .
      1
                  CollgCr
                                  Norm
                                         . . .
                                                  Attchd
                                                                    RFn
                                                                                  TA
      2
                                                                    RFn
                  Veenker
                                 Feedr
                                                  Attchd
                                                                                  TA
                                         . . .
      3
                  CollgCr
                                                  Attchd
                                                                    RFn
                                                                                  TA
                                  Norm
      4
                  Crawfor
                                                  Detchd
                                                                    Unf
                                                                                  TΑ
                                  Norm
      5
                                                                    RFn
                                                                                  TΑ
                  NoRidge
                                                  Attchd
                                  Norm
                                         . . .
                                                                    . . .
                                                                                 . . .
       . . .
                                   . . .
      1456
                  Gilbert
                                  Norm
                                                  Attchd
                                                                    RFn
                                                                                  TΑ
                                         . . .
      1457
                   NWAmes
                                  Norm
                                                  Attchd
                                                                    Unf
                                                                                  TΑ
      1458
                  Crawfor
                                                  Attchd
                                                                    RFn
                                                                                  TΑ
                                  Norm
                                         . . .
      1459
                    NAmes
                                  Norm
                                                  Attchd
                                                                    Unf
                                                                                  TΑ
      1460
                  Edwards
                                  Norm
                                                  Attchd
                                                                    Fin
                                                                                  TΑ
            GarageCond PavedDrive PoolQC Fence MiscFeature SaleType SaleCondition
      Ιd
      1
                                   Y
                                                                          WD
                     TA
                                         NoA
                                                 NoA
                                                              NoA
                                                                                     Normal
      2
                     TA
                                   Y
                                         NoA
                                                 NoA
                                                              NoA
                                                                          WD
                                                                                     Normal
      3
                     TA
                                   Υ
                                         NoA
                                                 NoA
                                                              NoA
                                                                          WD
                                                                                     Normal
      4
                     TΑ
                                   Y
                                         NoA
                                                                          WD
                                                                                    Abnorml
                                                 NoA
                                                              NoA
```

5	TA	Y	NoA	NoA	NoA	WD	Normal
1456	TA	Y	NoA	NoA	NoA	WD	Normal
1457	TA	Y	NoA	${ t MnPrv}$	NoA	WD	Normal
1458	TA	Y	NoA	${\tt GdPrv}$	Shed	WD	Normal
1459	TA	Y	NoA	NoA	NoA	WD	Normal
1460	TA	Y	NoA	NoA	NoA	WD	Normal

[1460 rows x 43 columns]

# 3.2 Variables numéricas

Para ver la matríz de correlaciones primero seleccionamos las variables numéricas de la base de datos Train

[89]:	<pre>trainNum = train.select_dtypes(include=['int64' or 'float64']).copy()</pre>
	trainNum

[89]:		LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	BsmtFinSF1	\
	Id							
	1	8450	7	5	2003	2003	706	
	2	9600	6	8	1976	1976	978	
	3	11250	7	5	2001	2002	486	
	4	9550	7	5	1915	1970	216	
	5	14260	8	5	2000	2000	655	
	1456	7917	6	5	1999	2000	0	
	1457	13175	6	6	1978	1988	790	
	1458	9042	7	9	1941	2006	275	
	1459	9717	5	6	1950	1996	49	
	1460	9937	5	6	1965	1965	830	
		BsmtFinS	F2 BsmtUnfSF	TotalBsmtSF	1stFlrSF	WoodDeck	SF \	
	Id					• • •		
	1		0 150	856	856	• • •	0	
	2		0 284	1262	1262		298	
	3		0 434	920	920		0	
	4		0 540	756	961		0	
	5		0 490	1145	1145		192	
						• • •	•••	
	1456		0 953	953	953		0	
	1457	1	63 589	1542	2073	3	349	
	1458		0 877	1152	1188		0	
	1459	10	29 0	1078	1078	3	366	
	1460	2	90 136	1256	1256		736	

 ${\tt OpenPorchSF} \quad {\tt EnclosedPorch} \quad {\tt 3SsnPorch} \quad {\tt ScreenPorch} \quad {\tt PoolArea} \quad {\tt MiscVal} \quad \backslash \\$ 

Id						
1	61	0	0	0	0	0
2	0	0	0	0	0	0
3	42	0	0	0	0	0
4	35	272	0	0	0	0
5	84	0	0	0	0	0
1456	40	0	0	0	0	0
1457	0	0	0	0	0	0
1458	60	0	0	0	0	2500
1459	0	112	0	0	0	0
1460	68	0	0	0	0	0

MoSold	YrSold	SalePrice
2	2008	208500
5	2007	181500
9	2008	223500
2	2006	140000
12	2008	250000
8	2007	175000
2	2010	210000
5	2010	266500
4	2010	142125
6	2008	147500
	2 5 9 2 12  8 2 5 4	2 2008 5 2007 9 2008 2 2006 12 2008  8 2007 2 2010 5 2010 4 2010

[1460 rows x 33 columns]

Ahora busquemos variables que no estén correlacionadas con la variable respuesta.

```
[90]: MC=trainNum.corr()
    SaleCorr=abs(MC.loc[:,"SalePrice"])
    aux=SaleCorr.sort_values(ascending=False)
    to_drop=aux.index[aux<0.1]
    to_drop</pre>
```

Las variables anteriores tienen una correlación menor a 0.1, no tiene caso que estén en el modelo, por lo tanto las quitamos de la matriz.

```
[91]: #Generamos la matriz de X de variables numéricas quitando la var resp

XtrainNum = trainNum.iloc[:,0:32].copy()

XtrainNum=XtrainNum.drop(XtrainNum[to_drop], axis=1).copy()

XtrainNum
```

[91]:		LotArea	Over	allQual	YearB	uilt	YearRe	modA	dd I	BsmtF	inSF1	Bsmt	tUnfSF	\
	Id													
	1	8450		7		2003		200			706		150	
	2	9600		6		1976		19			978		284	
	3	11250		7		2001		200			486		434	
	4	9550		7		1915		19			216		540	
	5	14260		8		2000		200	00		655		490	
	• • •	• • •		• • •										
	1456	7917		6		1999		200			0		953	
	1457	13175		6		1978		198			790		589	
	1458	9042		7		1941		200			275		877	
	1459	9717		5		1950		199			49		0	
	1460	9937		5		1965		196	65		830		136	
		TotalBsm	tSF	1stFlrSF	2ndF	lrSF	GrLivA	rea		Bed	roomAbv	/Gr	\	
	Id													
	1		856	856		854	1	710				3		
	2	1	262	1262		0	1	262				3		
	3		920	920		866	1	786				3		
	4		756	961		756	1	717				3		
	5	1	145	1145		1053	2	198				4		
	1456		953	953		694	1	.647				3		
	1457	1	542	2073		0	2	2073				3		
	1458	1	152	1188		1152	2	2340				4		
	1459	1	078	1078		0	1	.078				2		
	1460		256	1256		0		.256				3		
		KitchenA	bvGr	TotRmsAb	vGrd	Fire	places	Gara	ageCa	ars	Garage <i>l</i>	Area	\	
	Id						•		0		O			
	1		1		8		0			2		548		
	2		1		6		1			2		460		
	3		1		6		1			2		608		
	4		1		7		1			3		642		
	5		1		9		1			3		836		
	1456		1		7		1			2		460		
	1457		1		7		2			2		500		
	1458		1		9		2			1		252		
	1459		1		5		0			1		240		
	1460		1		6		0			1		276		
		WoodDeck	SF O	penPorchS	SF En	close	dPorch	Scr	eenPo	orch				
	Id			-										
	1		0	6	31		0			0				
	2	2	98		0		0			0				
	3		0		12		0			0				

4	0	35	272	0
5	192	84	0	0
1456	0	40	0	0
1457	349	0	0	0
1458	0	60	0	0
1459	366	0	112	0
1460	736	68	0	0

[1460 rows x 23 columns]

## [92]: print(XtrainNum.info())

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1460 entries, 1 to 1460
Data columns (total 23 columns):

#	Column	Non-Null Count	Dtype
0	LotArea	1460 non-null	int64
1	OverallQual	1460 non-null	int64
2	YearBuilt	1460 non-null	int64
3	${\tt YearRemodAdd}$	1460 non-null	int64
4	BsmtFinSF1	1460 non-null	int64
5	${\tt BsmtUnfSF}$	1460 non-null	int64
6	${\tt TotalBsmtSF}$	1460 non-null	int64
7	1stFlrSF	1460 non-null	int64
8	2ndFlrSF	1460 non-null	int64
9	GrLivArea	1460 non-null	int64
10	${\tt BsmtFullBath}$	1460 non-null	int64
11	FullBath	1460 non-null	int64
12	HalfBath	1460 non-null	int64
13	${\tt BedroomAbvGr}$	1460 non-null	int64
14	KitchenAbvGr	1460 non-null	int64
15	${\tt TotRmsAbvGrd}$	1460 non-null	int64
16	Fireplaces	1460 non-null	int64
17	${\tt GarageCars}$	1460 non-null	int64
18	${\tt GarageArea}$	1460 non-null	int64
19	WoodDeckSF	1460 non-null	int64
20	OpenPorchSF	1460 non-null	int64
21	${\tt EnclosedPorch}$	1460 non-null	int64
22	ScreenPorch	1460 non-null	int64

dtypes: int64(23) memory usage: 273.8 KB

None

A continuación, vamos a explorar la correlación entre las variables explicativas y filtramos aquellas que tienen una correlación (en valor absoluto) mayor que 0.6

```
[93]: import numpy as np
    corr_matrix = XtrainNum.corr().abs()
    #La mitad de arriba de la matriz de correlaciones en valores absolutos
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.
    →bool))
    #Lista de variables con correlación mayor a 0.6
    to_drop2 = [column for column in upper.columns if any(upper[column] > 0.6)]
    to_drop2
    #

[93]: ['1stFlrSF',
    'GrLivArea',
```

Estas variables tienen una correlación mayor a 0.6 con alguna otra de las variables explicativas, por ejemplo, '1stFlrSF' tiene una correlación de 0.819530 con 'TotalBsmtSF', basta con dejar solo una de estas dos variables y el algoritmo anterior seleccionó a '1stFlrSF', 'GrLivArea' tiene una correlación de 0.687501 con '2ndFlrSF', basta con dejar solo una de estas dos variables y así sucesivamente.

```
[94]: corr_matrix.iloc[:,[7,9]]
```

```
[94]:
                      1stFlrSF
                                GrLivArea
      LotArea
                     0.299475
                                 0.263116
      OverallQual
                                 0.593007
                     0.476224
      YearBuilt
                     0.281986
                                 0.199010
      YearRemodAdd
                     0.240379
                                 0.287389
      BsmtFinSF1
                                 0.208171
                     0.445863
      BsmtUnfSF
                     0.317987
                                 0.240257
      TotalBsmtSF
                                 0.454868
                     0.819530
      1stFlrSF
                      1.000000
                                 0.566024
      2ndFlrSF
                     0.202646
                                 0.687501
      GrLivArea
                     0.566024
                                 1.000000
      BsmtFullBath
                     0.244671
                                 0.034836
      FullBath
                     0.380637
                                 0.630012
      HalfBath
                     0.119916
                                 0.415772
      BedroomAbvGr
                     0.127401
                                 0.521270
      KitchenAbvGr
                     0.068101
                                 0.100063
      TotRmsAbvGrd
                     0.409516
                                 0.825489
      Fireplaces
                     0.410531
                                 0.461679
      GarageCars
                     0.439317
                                 0.467247
      GarageArea
                     0.489782
                                 0.468997
      WoodDeckSF
                     0.235459
                                 0.247433
```

 OpenPorchSF
 0.211671
 0.330224

 EnclosedPorch
 0.065292
 0.009113

 ScreenPorch
 0.088758
 0.101510

# [95]: XtrainNum=XtrainNum.drop(XtrainNum[to\_drop2], axis=1).copy() XtrainNum

[95]:		LotArea O	verallQual	YearBuilt	YearRe	modAdd 1	BsmtFinSF1	BsmtUnf	SF	\
	Id	0.450	_	0000		0000	<b>500</b>			
	1	8450	7	2003		2003	706		150	
	2	9600	6	1976		1976	978		284	
	3	11250	7	2001		2002	486		134	
	4	9550	7	1915		1970	216		540	
	5	14260	8	2000		2000	655		190	
							• • •			
	1456	7917	6	1999		2000	0		953	
	1457	13175	6	1978		1988	790		589	
	1458	9042	7	1941		2006	275	8	377	
	1459	9717	5	1950		1996	49		0	
	1460	9937	5	1965		1965	830	1	L36	
		TotalBsmtS	F 2ndFlrSF	BedroomAb	vGr Ki	tchenAbv	Gr Firepla	.ces \		
	Id									
	1	85	6 854		3		1	0		
	2	126	2 0		3		1	1		
	3	92	0 866		3		1	1		
	4	75	6 756		3		1	1		
	5	114	5 1053		4		1	1		
	1456	95	3 694		3		1	1		
	1457	154	2 0		3		1	2		
	1458	115	2 1152		4		1	2		
	1459	107	8 0		2		1	0		
	1460	125	6 0		3		1	0		
		WoodDeckSF	OpenPorchS	SF Enclose	dPorch	ScreenP	orch			
	Id		1							
	1	0	6	31	0		0			
	2	298		0	0		0			
	3	0		12	0		0			
	4	0	3	35	272		0			
	5	192		34	0		0			
	1456	0		10	0		0			
	1457	349		0	0		0			
	1458	0	$\epsilon$	30	0		0			
	1459	366		0	112		0			

1460 736 68 0 0

[1460 rows x 15 columns]

Y únicamente nos quedamos con 15 variables numéricas para inlcuir en el modelo.

```
[96]: #Preparamos la base de Test
XtestNum = test[XtrainNum.columns]
```

#### 3.3 Conclusión

La matriz de variables explicativas queda de la siguietne forma con 58 variables:

```
[97]: Xtrain = pd.merge(XtrainNum, XtrainCat, on='Id')
      #lo mismo para test
      Xtest = pd.merge(XtestNum, XtestCat, on='Id')
      Xtrain.head()
          LotArea OverallQual YearBuilt YearRemodAdd BsmtFinSF1 BsmtUnfSF \
[97]:
      Ιd
                               7
      1
              8450
                                        2003
                                                       2003
                                                                     706
                                                                                 150
      2
              9600
                               6
                                        1976
                                                       1976
                                                                     978
                                                                                 284
      3
             11250
                               7
                                        2001
                                                       2002
                                                                     486
                                                                                 434
                               7
      4
              9550
                                        1915
                                                       1970
                                                                     216
                                                                                 540
                                                                     655
      5
             14260
                               8
                                        2000
                                                       2000
                                                                                 490
          TotalBsmtSF
                        2ndFlrSF
                                   BedroomAbvGr KitchenAbvGr
                                                                       GarageType \
                                                                  . . .
      Id
                                                                  . . .
                              854
      1
                   856
                                               3
                                                               1
                                                                  . . .
                                                                            Attchd
      2
                  1262
                                0
                                               3
                                                                            Attchd
                                                               1
                                                                  . . .
      3
                   920
                              866
                                               3
                                                               1
                                                                  . . .
                                                                            Attchd
      4
                   756
                              756
                                               3
                                                                           Detchd
                                                               1
                                                                  . . .
      5
                  1145
                             1053
                                               4
                                                               1
                                                                            Attchd
```

	${\tt GarageFinish}$	GarageQual	${\tt GarageCond}$	PavedDrive	${\tt PoolQC}$	Fence	MiscFeature	\
Id								
1	RFn	TA	TA	Y	NoA	NoA	NoA	
2	RFn	TA	TA	Y	NoA	NoA	NoA	
3	RFn	TA	TA	Y	NoA	NoA	NoA	
4	Unf	TA	TA	Y	NoA	NoA	NoA	
5	RFn	TA	TA	Y	NoA	NoA	NoA	

SaleType SaleCondition

Id		
1	WD	Normal
2	WD	Normal
3	WD	Normal

```
4 WD Abnorml
5 WD Normal
```

[5 rows x 58 columns]

# 4 Modelo LASSO

Al momento de trabajar las variables explicativas con la que se va a modelar la variable "PriceSale" conviene trabajar juntas las bases de datos de train y de test, para generar las mismas variables doomies.

```
[98]: Xjoin = pd.concat([Xtrain, Xtest])
Xjoin.info()
```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2919 entries, 1 to 2919
Data columns (total 58 columns):

#	Column	Non-Null Count	Dtype
0	 LotArea	2919 non-null	 int64
1	OverallQual	2919 non-null	int64
2	YearBuilt	2919 non-null	int64
3	YearRemodAdd	2919 non-null	int64
4	BsmtFinSF1	2919 non-null	float64
5	BsmtUnfSF	2919 non-null	float64
6	TotalBsmtSF	2919 non-null	float64
7	2ndFlrSF	2919 non-null	int64
8	${\tt BedroomAbvGr}$	2919 non-null	int64
9	KitchenAbvGr	2919 non-null	int64
10	Fireplaces	2919 non-null	int64
11	WoodDeckSF	2919 non-null	int64
12	OpenPorchSF	2919 non-null	int64
13	${\tt EnclosedPorch}$	2919 non-null	int64
14	ScreenPorch	2919 non-null	int64
15	MSZoning	2919 non-null	object
16	Street	2919 non-null	object
17	Alley	2919 non-null	object
18	LotShape	2919 non-null	object
19	LandContour	2919 non-null	object
20	Utilities	2919 non-null	object
21	LotConfig	2919 non-null	object
22	LandSlope	2919 non-null	object
23	Neighborhood	2919 non-null	object
24	Condition1	2919 non-null	object
25	Condition2	2919 non-null	object
26	BldgType	2919 non-null	object
27	HouseStyle	2919 non-null	object

```
29
          RoofMatl
                                          object
                          2919 non-null
      30
          Exterior1st
                          2919 non-null
                                          object
      31
          Exterior2nd
                          2919 non-null
                                          object
          MasVnrType
                                          object
      32
                          2919 non-null
      33
          ExterQual
                          2919 non-null
                                          object
      34
          ExterCond
                          2919 non-null
                                          object
          Foundation
                          2919 non-null
                                          object
          BsmtQual
                          2919 non-null
                                          object
          BsmtCond
      37
                          2919 non-null
                                          object
      38
          BsmtExposure
                          2919 non-null
                                          object
      39
          BsmtFinType1
                          2919 non-null
                                          object
          BsmtFinType2
      40
                          2919 non-null
                                          object
      41
          Heating
                          2919 non-null
                                          object
      42
          HeatingQC
                          2919 non-null
                                          object
          CentralAir
                          2919 non-null
                                          object
          Electrical
                          2919 non-null
                                          object
      45
         KitchenQual
                          2919 non-null
                                          object
          Functional
                          2919 non-null
                                          object
      47
          FireplaceQu
                          2919 non-null
                                          object
          GarageType
                                          object
      48
                          2919 non-null
      49
          GarageFinish
                          2919 non-null
                                          object
          GarageQual
                          2919 non-null
                                          object
      51
          GarageCond
                          2919 non-null
                                          object
      52
          PavedDrive
                          2919 non-null
                                          object
      53
          PoolQC
                          2919 non-null
                                          object
      54
          Fence
                          2919 non-null
                                          object
      55
          MiscFeature
                          2919 non-null
                                          object
                          2919 non-null
                                          object
          SaleType
          SaleCondition 2919 non-null
                                          object
     dtypes: float64(3), int64(12), object(43)
     memory usage: 1.3+ MB
[99]: dummies = []
      for i in Xjoin.columns:
          if (Xjoin[i].dtype=='object'):
              dummies.append(i)
      dummies
[99]: ['MSZoning',
       'Street',
       'Alley',
       'LotShape',
       'LandContour',
       'Utilities',
       'LotConfig',
       'LandSlope',
```

object

28 RoofStyle

2919 non-null

```
'Condition1',
        'Condition2',
        'BldgType',
        'HouseStyle',
        'RoofStyle',
        'RoofMatl',
        'Exterior1st',
        'Exterior2nd',
        'MasVnrType',
        'ExterQual',
        'ExterCond',
        'Foundation',
        'BsmtQual',
        'BsmtCond',
        'BsmtExposure',
        'BsmtFinType1',
        'BsmtFinType2',
        'Heating',
        'HeatingQC',
        'CentralAir',
        'Electrical',
        'KitchenQual',
        'Functional',
        'FireplaceQu',
        'GarageType',
        'GarageFinish',
        'GarageQual',
        'GarageCond',
        'PavedDrive',
        'PoolQC',
        'Fence',
        'MiscFeature',
        'SaleType',
        'SaleCondition']
[100]: status = pd.get_dummies(Xjoin[dummies],drop_first=True) ## one hot encoding on_
       →all variables
       Xjoin = pd.concat([Xjoin,status],axis=1)
       Xjoin.drop(dummies,axis=1,inplace=True)
       Xjoin.head()
[100]:
           LotArea OverallQual YearBuilt YearRemodAdd BsmtFinSF1 BsmtUnfSF \
       Ιd
       1
                               7
                                       2003
                                                                 706.0
              8450
                                                      2003
                                                                             150.0
       2
              9600
                                       1976
                                                      1976
                                                                 978.0
                                                                             284.0
                               6
                               7
       3
             11250
                                       2001
                                                      2002
                                                                 486.0
                                                                             434.0
```

'Neighborhood',

```
4
       9550
                         7
                                  1915
                                                 1970
                                                             216.0
                                                                          540.0
5
      14260
                         8
                                  2000
                                                 2000
                                                             655.0
                                                                          490.0
    TotalBsmtSF
                  2ndFlrSF
                             BedroomAbvGr KitchenAbvGr
                                                                  SaleType_ConLI
                                                            . . .
Ιd
1
          856.0
                        854
                                         3
                                                                                0
                                                         1
                                                            . . .
2
          1262.0
                          0
                                                                                0
                                         3
                                                            . . .
                                                         1
3
          920.0
                        866
                                         3
                                                                                0
                                                         1
4
          756.0
                        756
                                         3
                                                                                0
5
          1145.0
                       1053
                                         4
                                                                                0
    SaleType_ConLw SaleType_New SaleType_Oth SaleType_WD
Ιd
1
                  0
                                  0
                                                 0
                                                                1
2
                  0
                                  0
                                                 0
                                                                1
3
                  0
                                  0
                                                 0
                                                                1
4
                  0
                                  0
                                                 0
                                                                1
                  0
5
                                                 0
                                                                1
    SaleCondition_AdjLand SaleCondition_Alloca SaleCondition_Family \
Ιd
1
                          0
                                                  0
                                                                           0
2
                          0
                                                  0
                                                                           0
3
                          0
                                                  0
                                                                           0
4
                          0
                                                  0
                                                                           0
5
                          0
                                                  0
                                                                           0
    SaleCondition_Normal SaleCondition_Partial
Ιd
                                                  0
1
                         1
2
                         1
                                                  0
3
                                                  0
                         1
4
                                                  0
                         0
                                                  0
```

[5 rows x 247 columns]

# Ahora separamos la matriz Xjoin nuevamente:

```
[101]: Xtrain = Xjoin.iloc[0:1460,]
Xtrain
```

[101]:		LotArea	OverallQual	YearBuilt	YearRemodAdd	BsmtFinSF1	${\tt BsmtUnfSF}$	\
I	d							
1	L	8450	7	2003	2003	706.0	150.0	
2	2	9600	6	1976	1976	978.0	284.0	
3	3	11250	7	2001	2002	486.0	434.0	

4	9550	7	1915	1970	216.0	540.0
5	14260	8	2000	2000	655.0	490.0
1456	7917	6	1999	2000	0.0	953.0
1457	13175	6	1978	1988	790.0	589.0
1458	9042	7	1941	2006	275.0	877.0
1459	9717	5	1950	1996	49.0	0.0
1460	9937	5	1965	1965	830.0	136.0
	${\tt TotalBsmtSF}$	2ndFlrSF	${\tt BedroomAbvGr}$	KitchenAbv	Gr Sa	leType_ConLI \
Id						
1	856.0	854	3		1	0
2	1262.0	0	3		1	0
3	920.0	866	3		1	0
4	756.0	756	3		1	0
5	1145.0	1053	4		1	0
1456	953.0	694	3		1	0
1457	1542.0	0	3		1	0
1458	1152.0	1152	4		1	0
1459	1078.0	0	2		1	0
1460	1256.0	0	3		1	0
	SaleType_Con	Lw SaleTy	pe_New SaleTy	pe_Oth Sal	_eType_WD \	
Id						
1		0	0	0	1	
2		0	0	0	1	
3		0	0	0	1	
4		0	0	0	1	
5		0	0	0	1	
1456		0	0	0	1	
1457		0	0	0	1	
1458		0	0	0	1	
1459		0	0	0	1	
1460		0	0	0	1	
	SaleConditio	n_AdjLand	SaleCondition	_Alloca Sa	aleCondition	_Family $\setminus$
Id						
1		0		0		0
2		0		0		0
3		0		0		0
4		0		0		0
5		0		0		0
1456		0		0		0
1457		0		0		0

1458		0		0		0	
1459		0		0		0	
1460		0		0		0	
	SaleCondition_Normal SaleCondition_Partial						
Id							
1		1		0			
2		1		0			
3		1		0			
4		0		0			
5		1		0			
1456		1		0			
1457		1		0			
1458		1		0			
1459		1		0			
1459		1		0			
1460		1		U			
Γ1.46	O rows x 247	columnal					
[1400	J TOWS X 247	COLUMNIS					
[100] . <b>V</b> + ag	t = Xjoin.ilo	ο <b>Γ</b> 1460 • 2016	n 1				
	•	C[1460:291	۶, ا				
Xtes	T						
[400]	T - + A O	1101	VD11+ V	/D1/4-1-1	D+E GE4	D+II £QE \	
[102]:	LotArea Ov	erallyual	rearbuilt r	YearRemodAdd	BSMTFINSFI	BsmtUnfSF \	
Id	44000	_	1001	1001	400.0	070 0	
1461	11622	5	1961	1961	468.0	270.0	
1462	14267	6	1958	1958	923.0	406.0	
1463	13830	5	1997	1998	791.0	137.0	
1464	9978	6	1998	1998	602.0	324.0	
1465	5005	8	1992	1992	263.0	1017.0	
2915	1936	4	1970	1970	0.0	546.0	
2916	1894	4	1970	1970	252.0	294.0	
2917	20000	5	1960	1996	1224.0	0.0	
2918		5	1992	1992	337.0	575.0	
2919	9627	7	1993	1994	758.0	238.0	
		•			, , , , ,		
	TotalBsmtSF	2ndFlrSF	BedroomAbw0	r KitchenAby	Gr Sa	leType_ConLI \	
Id	TOUGLEDEMUEL	Ziidi II bi	Doar comme v	ii iii oononno v		101ypo_001111 (	
1461	882.0	0		2	1	0	
1461				3	1	0	
1463	928.0	701		3	1	0	
1464		678		3	1	0	
1465	1280.0	0		2	1	0	
• • •					• • • • • •	• • •	
2915	546.0	546		3	1	0	
2916	546.0	546		3	1	0	

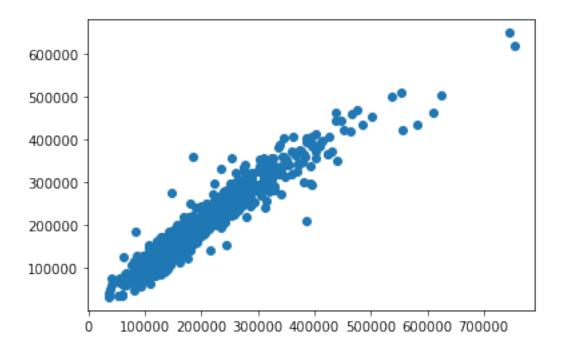
2917	1224.0	0	4	1	0
2918	912.0	0	3	1	0
2919	996.0	1004	3	1	0
	SaleType_ConLw	SaleType_New	SaleType_Oth	SaleType_WD	\
Id					
1461	0	0	0	1	
1462	0	0	0	1	
1463	0	0	0	1	
1464	0	0	0	1	
1465	0	0	0	1	
	• • •				
2915	0	0	0	1	
2916	0	0	0	1	
2917	0	0	0	1	
2918	0	0	0	1	
2919	0	0	0	1	
	SaleCondition_A	djLand SaleCo	ondition_Alloca	SaleCondition	on_Family \
Id					·
1461		0	0		0
1462		0	0		0
1463		0	0		0
1464		0	0		0
1465		0	0		0
2915		0	0		0
2916		0	0		0
2917		0	0		0
2918		0	0		0
2919		0	0		0
	SaleCondition_N	Jormal SaleCor	dition Partial		
Id	241000H41010H_H	Januar Darooon			
1461		1	0		
1462		1	0		
1463		1	0		
1464		1	0		
1465		1	0		
2915		1	0		
2916		0	0		
2917		0	0		
2918		1	0		
2919		1	0		

[1459 rows x 247 columns]

```
[103]: | ytrain=train.iloc[:,79]
       ytrain.head()
[103]: Id
       1
            208500
       2
            181500
       3
            223500
            140000
       4
       5
            250000
       Name: SalePrice, dtype: int64
[104]: from sklearn.linear_model import Lasso
       alpha = 0.0002
       lasso = Lasso(alpha=alpha)
       lasso.fit(Xtrain, ytrain)
       #lasso.coef_
      C:\Users\Brendis\anaconda3\lib\site-
      packages\sklearn\linear_model\_coordinate_descent.py:529: ConvergenceWarning:
      Objective did not converge. You might want to increase the number of iterations.
      Duality gap: 367468093716.1706, tolerance: 920791133.4609977
        model = cd_fast.enet_coordinate_descent(
[104]: Lasso(alpha=0.0002)
[105]: lasso_df=pd.DataFrame()
       lasso_df['Features'] = Xtrain.columns
       lasso_df['Coefficients']=lasso.coef_
       lasso_df['ABS Coefficients'] = abs(lasso.coef_)
       lasso_df.sort_values(by=['ABS Coefficients'],ascending=False,inplace = True)
       lasso_df
[105]:
                      Features Coefficients ABS Coefficients
       93
              RoofMatl_Membran 7.775861e+05
                                                   7.775861e+05
              RoofMatl_WdShngl 7.482665e+05
                                                  7.482665e+05
       98
                RoofMatl_Metal 7.465950e+05
       94
                                                  7.465950e+05
              RoofMatl_WdShake 7.023657e+05
                                                   7.023657e+05
              RoofMatl_Tar&Grv 6.945362e+05
                                                  6.945362e+05
       96
       . .
       12
                   OpenPorchSF 1.357898e+00
                                                   1.357898e+00
       0
                                                  8.951859e-01
                       LotArea 8.951859e-01
       207
              GarageFinish_NoA 1.418699e-08
                                                   1.418699e-08
           Exterior2nd_CBlock -0.000000e+00
                                                  0.000000e+00
       118
       30
              Utilities_NoSeWa -0.000000e+00
                                                   0.00000e+00
```

[247 rows x 3 columns]

[107]: <matplotlib.collections.PathCollection at 0x20d642f8610>



# 5 Probando con la base test

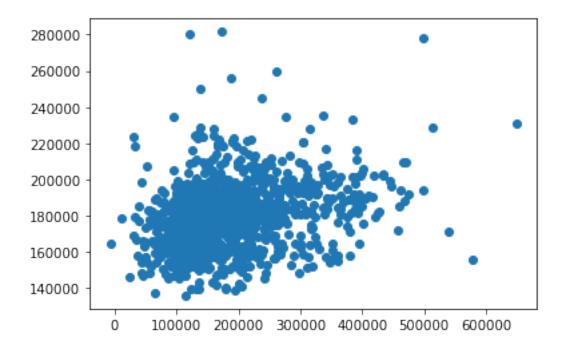
```
[108]: ygorrotest = lasso.predict(Xtest)
ygorrotest
[108]: array([111925.76107905, 168608.11494094, 186806.10640742, ...,
```

173887.34545318, 110457.10675563, 222513.35407205])

Acontinuación se grafica la estimación del precio de venta de la base de datos test usando el modelo lasso contra el precio estimado de la base de datos

```
[109]: plt.scatter(ygorrotest,sample_submission['SalePrice'])
```

[109]: <matplotlib.collections.PathCollection at 0x20d641e9400>



# 5.1 Conclusión

Existe demasiada dispersión entre los precios estimados del modelo lasso obtenido en el presente análisis comparado con la estimación proporcionada en la base de datos ample\_submission.