In [15]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
import matplotlib.gridspec as gridspec
import seaborn as sns
from scipy import stats
import os
os.chdir('/Users/Lenovo/Desktop/EBAC')
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df = pd.read_csv('House Pricing.csv')
df.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fenc
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	Nal
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	Nal
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	Nal
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	Nal
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	Nal

5 rows × 81 columns

In [5]:
```
#Estadistica Descriptiva
df.describe()
```

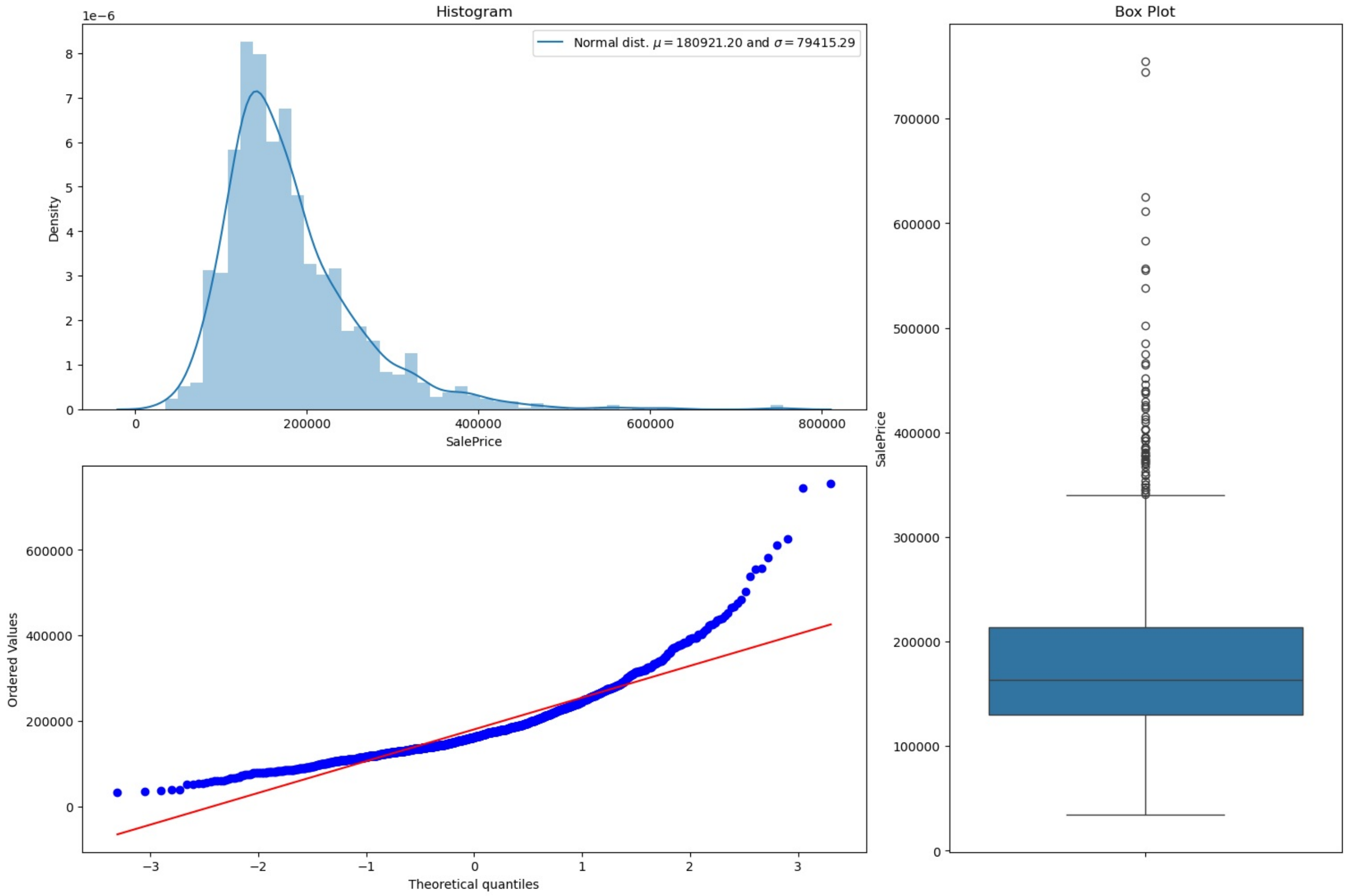
Out[5]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrAre
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.00000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.68526
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.06620
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.00000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.00000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.00000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.00000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.00000

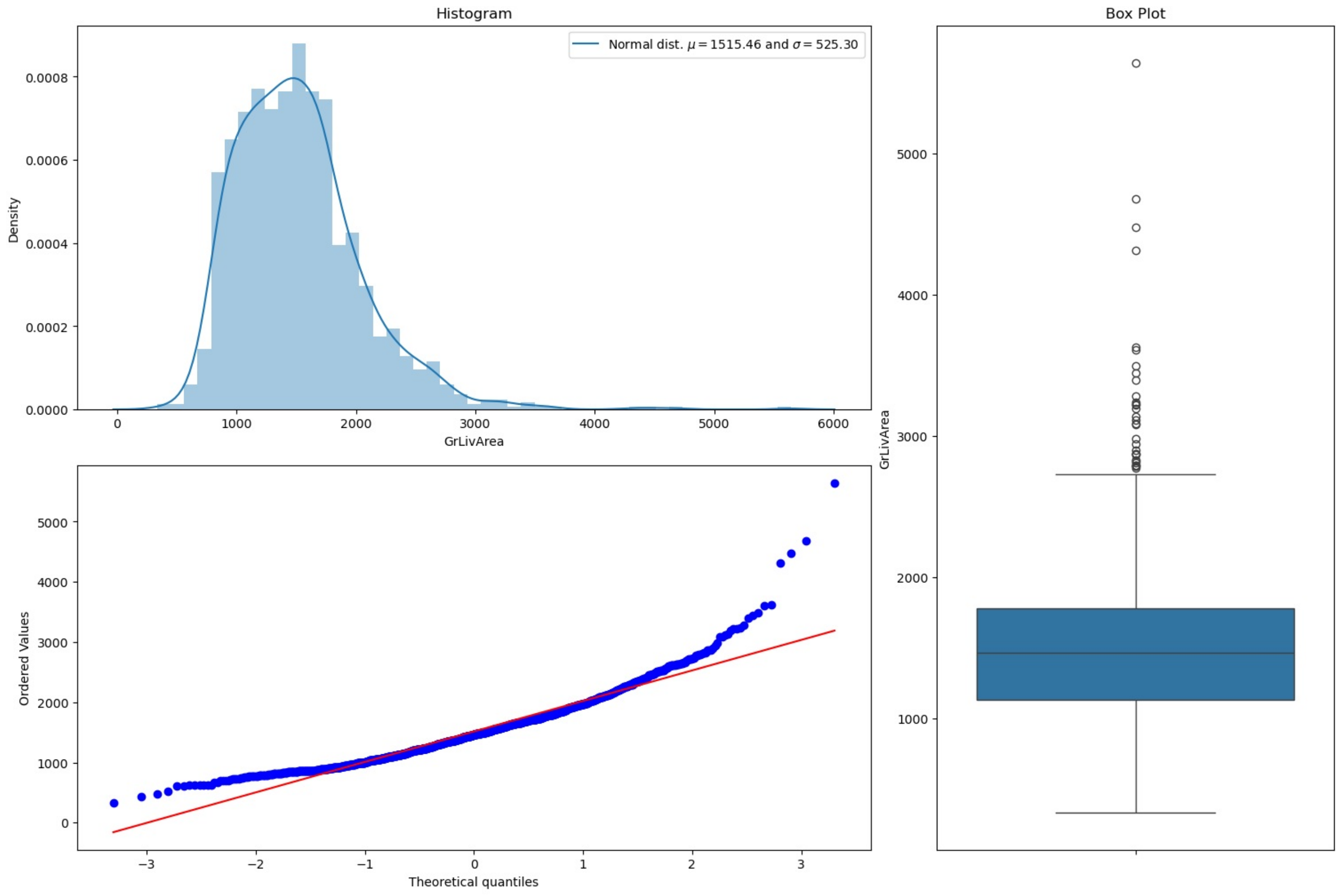
8 rows × 38 columns

In [17]:
```
def plot_dist_char(df, feature):
    # Figura
    fig = plt.figure(constrained_layout=True, figsize=(15,10))
    grid = gridspec.GridSpec(ncols=3, nrows=2, figure=fig)

    # Media y Desviacion Estandar
    mu = np.mean(df[feature])
    sigma = np.std(df[feature])

    # Histograma
    ax1 = fig.add_subplot(grid[0, :2])
    ax1.set_title('Histogram')
    sns.distplot(df.loc[:,feature], norm_hist=True, ax=ax1)
    plt.legend(['Normal dist. $\mu={:.2f}$ and $\sigma={:.2f}$'.format(mu, sigma)])

    # QQ Plot
    ax2 = fig.add_subplot(grid[1, :2])
    stats.probplot(df.loc[:,feature], plot=ax2)
    ax2.set_title('')

    # Box Plot
    ax3 = fig.add_subplot(grid[:, 2])
    ax3.set_title('Box Plot')
    sns.boxplot(y=df.loc[:,feature], ax=ax3)
```

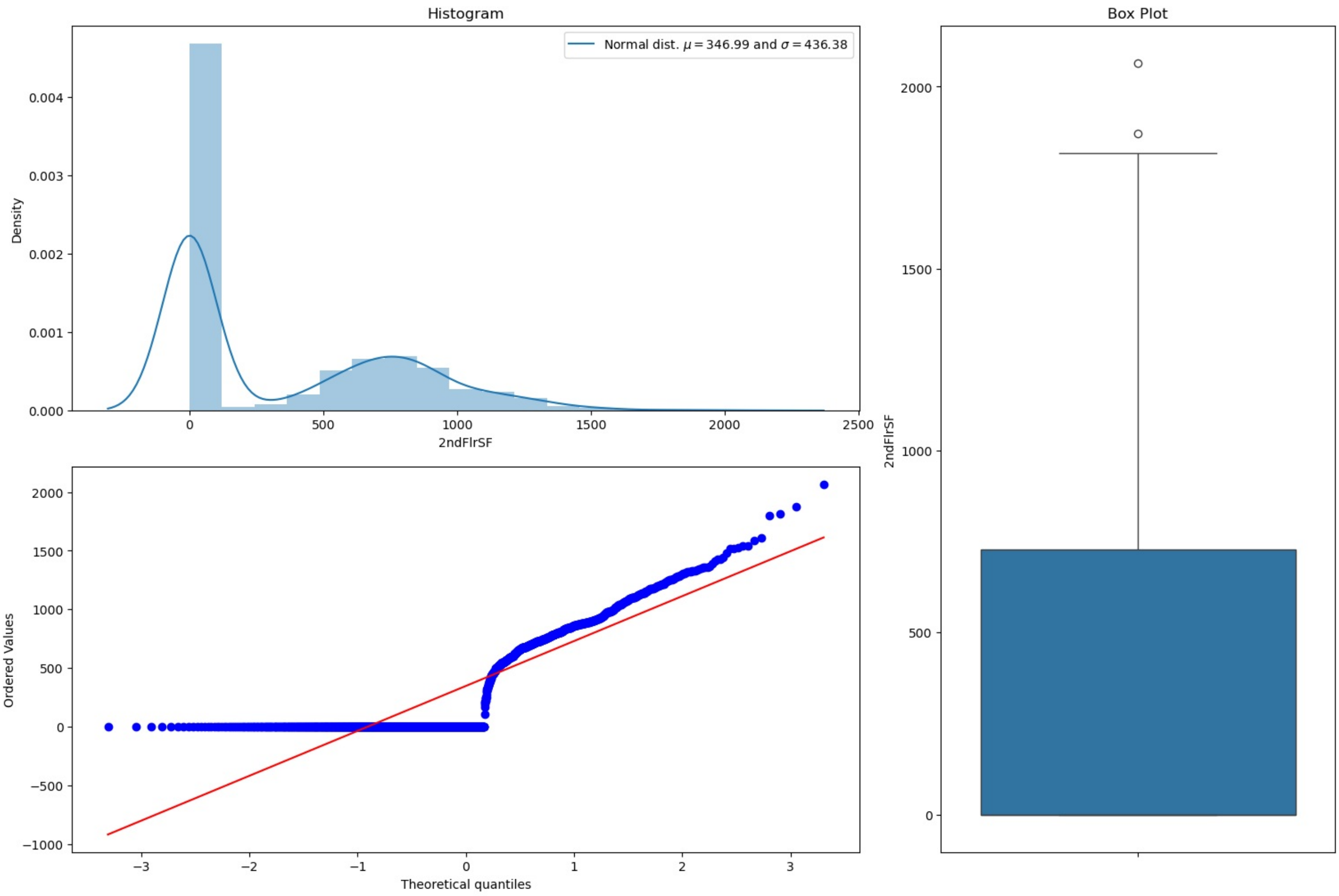
In [19]:
```
# Visualizacion estadistica (SalePrice)
plot_dist_char(df, 'SalePrice')
```



In [29]:
```python
# Visualizacion estadistica (GrLivArea)
plot_dist_char(df, 'GrLivArea')
```



In [31]:
```python
# Visualizacion estadistica (2ndFlrSF)
plot_dist_char(df, '2ndFlrSF')
```



Conclusion

En este caso, para estas 3 variables, podemos observar que no tienen una distribucion normal.

In [33]:
```
# Seleccion de variables numericas
df_nuevo = df.select_dtypes(include = 'number')
df_nuevo.corr()
```

Out[33]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea
Id	1.000000	0.011156	-0.010601	-0.033226	-0.028365	0.012609	-0.012713	-0.021998	-0.050298
MSSubClass	0.011156	1.000000	-0.386347	-0.139781	0.032628	-0.059316	0.027850	0.040581	0.022936
LotFrontage	-0.010601	-0.386347	1.000000	0.426095	0.251646	-0.059213	0.123349	0.088866	0.193458
LotArea	-0.033226	-0.139781	0.426095	1.000000	0.105806	-0.005636	0.014228	0.013788	0.104160
OverallQual	-0.028365	0.032628	0.251646	0.105806	1.000000	-0.091932	0.572323	0.550684	0.411876
OverallCond	0.012609	-0.059316	-0.059213	-0.005636	-0.091932	1.000000	-0.375983	0.073741	-0.128101
YearBuilt	-0.012713	0.027850	0.123349	0.014228	0.572323	-0.375983	1.000000	0.592855	0.315707
YearRemodAdd	-0.021998	0.040581	0.088866	0.013788	0.550684	0.073741	0.592855	1.000000	0.179618
MasVnrArea	-0.050298	0.022936	0.193458	0.104160	0.411876	-0.128101	0.315707	0.179618	1.000000
BsmtFinSF1	-0.005024	-0.069836	0.233633	0.214103	0.239666	-0.046231	0.249503	0.128451	0.264736
BsmtFinSF2	-0.005968	-0.065649	0.049900	0.111170	-0.059119	0.040229	-0.049107	-0.067759	-0.072319
BsmtUnfSF	-0.007940	-0.140759	0.132644	-0.002618	0.308159	-0.136841	0.149040	0.181133	0.114442
TotalBsmtSF	-0.015415	-0.238518	0.392075	0.260833	0.537808	-0.171098	0.391452	0.291066	0.363936
1stFlrSF	0.010496	-0.251758	0.457181	0.299475	0.476224	-0.144203	0.281986	0.240379	0.344501
2ndFlrSF	0.005590	0.307886	0.080177	0.050986	0.295493	0.028942	0.010308	0.140024	0.174561
LowQualFinSF	-0.044230	0.046474	0.038469	0.004779	-0.030429	0.025494	-0.183784	-0.062419	-0.069071
GrLivArea	0.008273	0.074853	0.402797	0.263116	0.593007	-0.079686	0.199010	0.287389	0.390857
BsmtFullBath	0.002289	0.003491	0.100949	0.158155	0.111098	-0.054942	0.187599	0.119470	0.085310
BsmtHalfBath	-0.020155	-0.002333	-0.007234	0.048046	-0.040150	0.117821	-0.038162	-0.012337	0.026673
FullBath	0.005587	0.131608	0.198769	0.126031	0.550600	-0.194149	0.468271	0.439046	0.276833
HalfBath	0.006784	0.177354	0.053532	0.014259	0.273458	-0.060769	0.242656	0.183331	0.201444
BedroomAbvGr	0.037719	-0.023438	0.263170	0.119690	0.101676	0.012980	-0.070651	-0.040581	0.102821
KitchenAbvGr	0.002951	0.281721	-0.006069	-0.017784	-0.183882	-0.087001	-0.174800	-0.149598	-0.037610
TotRmsAbvGrd	0.027239	0.040380	0.352096	0.190015	0.427452	-0.057583	0.095589	0.191740	0.280682
Fireplaces	-0.019772	-0.045569	0.266639	0.271364	0.396765	-0.023820	0.147716	0.112581	0.249070
GarageYrBlt	0.000072	0.085072	0.070250	-0.024947	0.547766	-0.324297	0.825667	0.642277	0.252691
GarageCars	0.016570	-0.040110	0.285691	0.154871	0.600671	-0.185758	0.537850	0.420622	0.364204
GarageArea	0.017634	-0.098672	0.344997	0.180403	0.562022	-0.151521	0.478954	0.371600	0.373066
WoodDeckSF	-0.029643	-0.012579	0.088521	0.171698	0.238923	-0.003334	0.224880	0.205726	0.159718
OpenPorchSF	-0.000477	-0.006100	0.151972	0.084774	0.308819	-0.032589	0.188686	0.226298	0.125703
EnclosedPorch	0.002889	-0.012037	0.010700	-0.018340	-0.113937	0.070356	-0.387268	-0.193919	-0.110204
3SsnPorch	-0.046635	-0.043825	0.070029	0.020423	0.030371	0.025504	0.031355	0.045286	0.018796
ScreenPorch	0.001330	-0.026030	0.041383	0.043160	0.064886	0.054811	-0.050364	-0.038740	0.061466
PoolArea	0.057044	0.008283	0.206167	0.077672	0.065166	-0.001985	0.004950	0.005829	0.011723
MiscVal	-0.006242	-0.007683	0.003368	0.038068	-0.031406	0.068777	-0.034383	-0.010286	-0.029815
MoSold	0.021172	-0.013585	0.011200	0.001205	0.070815	-0.003511	0.012398	0.021490	-0.005965
YrSold	0.000712	-0.021407	0.007450	-0.014261	-0.027347	0.043950	-0.013618	0.035743	-0.008201
SalePrice	-0.021917	-0.084284	0.351799	0.263843	0.790982	-0.077856	0.522897	0.507101	0.477493

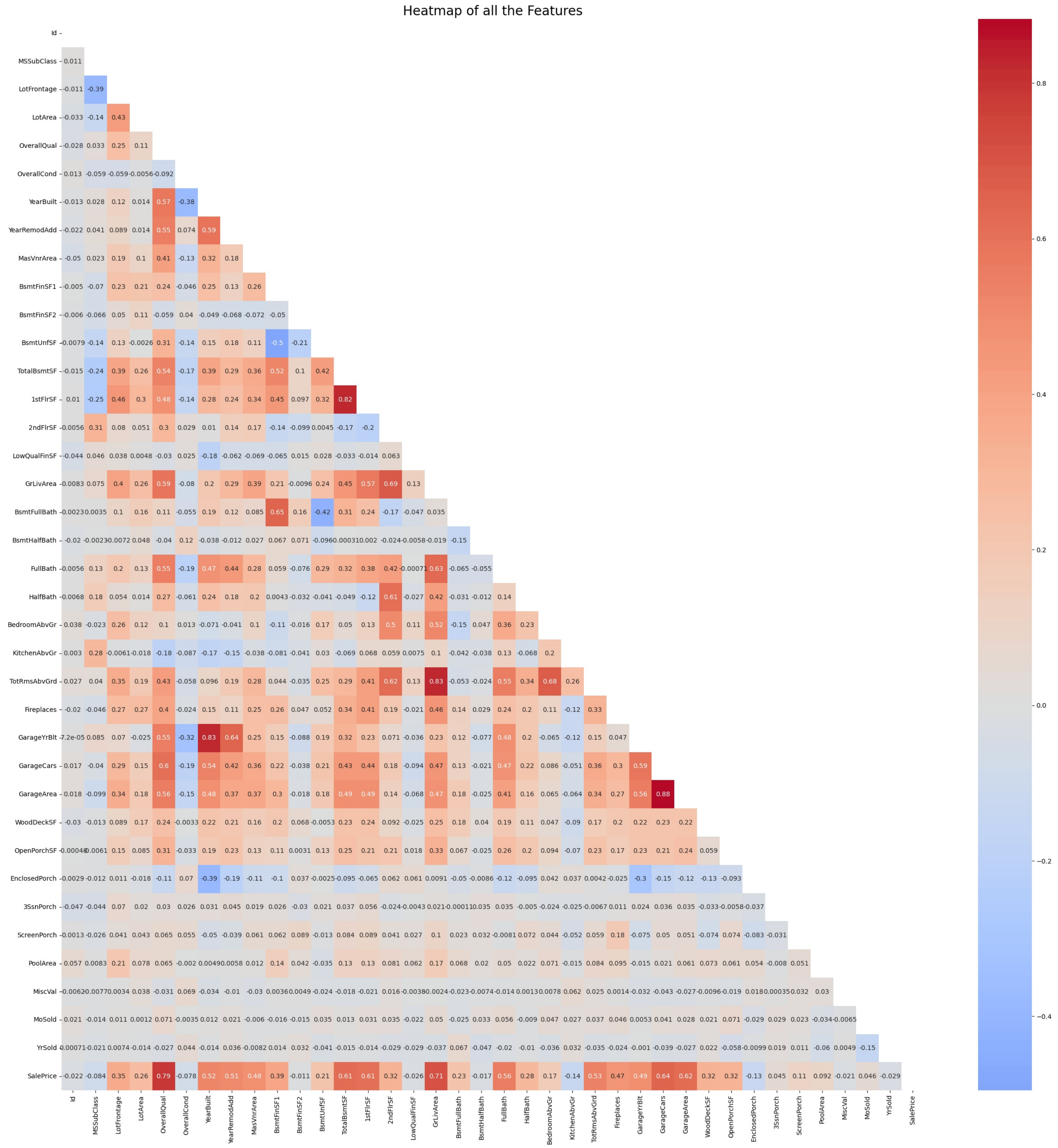
38 rows × 38 columns

In [37]:

```
#Correlacion Heatmap
# Definimos un tamaño para la gráfica
f, ax = plt.subplots(figsize=(30, 30))

# Con esto quitamos los valores de la matriz de identidad (Nos ayudará a tener una mejor visualización)
mask = np.triu(np.ones_like(df_nuevo.corr(), dtype=bool))

# Grafiquemos el Heatmap
sns.heatmap(df_nuevo.corr(),
            cmap=sns.color_palette('coolwarm', 200),
            mask=mask,
            annot=True,
            center=0)
```

```
# Agrega un título a la gráfica
plt.title("Heatmap of all the Features", fontsize=20)
```

Out[37]: Text(0.5, 1.0, 'Heatmap of all the Features')



Conclusion

Las variables que mas se relacionan a SalePrice son OverallQual, GrLivArea y Garagecars

In [45]:
```
import statsmodels.api as sm

df_nuevo = df_nuevo.dropna()
y = df_nuevo['SalePrice']
X = df_nuevo.drop(columns='SalePrice')

#Reporte de regresion
X = sm.add_constant(X)
modelo = sm.OLS(y, X).fit()

# 5. Muestra el resumen del modelo
print(modelo.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.810
Model:                            OLS   Adj. R-squared:                  0.803
Method:                 Least Squares   F-statistic:                     131.8
Date:                Wed, 23 Jul 2025   Prob (F-statistic):               0.00
Time:                        19:54:27   Log-Likelihood:                -13358.
No. Observations:                1121   AIC:                         2.679e+04
Df Residuals:                    1085   BIC:                         2.697e+04
Df Model:                          35
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
const         -3.351e+05    1.7e+06     -0.197      0.844   -3.67e+06       3e+06
Id               -1.2053      2.658     -0.453      0.650      -6.421       4.011
MSSubClass     -200.0623     34.511     -5.797      0.000    -267.779    -132.346
LotFrontage    -116.0282     61.264     -1.894      0.059    -236.237       4.181
LotArea           0.5422      0.158      3.442      0.001       0.233       0.851
OverallQual    1.866e+04   1481.619     12.592      0.000    1.57e+04    2.16e+04
OverallCond    5239.4864   1367.853      3.830      0.000    2555.550    7923.422
YearBuilt       316.4201     87.663      3.610      0.000     144.412     488.428
YearRemodAdd    119.4141     86.682      1.378      0.169     -50.669     289.497
MasVnrArea       31.4076      7.022      4.473      0.000      17.629      45.186
BsmtFinSF1        9.6803      3.129      3.094      0.002       3.541      15.820
BsmtFinSF2        0.6662      5.587      0.119      0.905     -10.295      11.628
BsmtUnfSF        -2.6710      2.937     -0.910      0.363      -8.433       3.091
TotalBsmtSF       7.6755      4.223      1.818      0.069      -0.610      15.961
1stFlrSF         14.4718      8.483      1.706      0.088      -2.173      31.117
2ndFlrSF         15.1237      7.713      1.961      0.050      -0.010      30.258
LowQualFinSF      1.9062     20.947      0.091      0.928     -39.194      43.006
GrLivArea        31.5017      7.767      4.056      0.000      16.262      46.741
BsmtFullBath   9042.8022   3198.072      2.828      0.005    2767.697    1.53e+04
BsmtHalfBath   2465.0370   5073.115      0.486      0.627   -7489.190    1.24e+04
FullBath       5433.1446   3531.117      1.539      0.124   -1495.447    1.24e+04
HalfBath      -1098.3395   3321.384     -0.331      0.741   -7615.402    5418.723
BedroomAbvGr  -1.022e+04   2155.038     -4.742      0.000   -1.44e+04   -5990.397
KitchenAbvGr  -2.202e+04   6709.938     -3.282      0.001   -3.52e+04   -8857.560
TotRmsAbvGrd   5464.1204   1487.289      3.674      0.000    2545.833    8382.408
Fireplaces     4371.8698   2188.667      1.998      0.046      77.370    8666.369
GarageYrBlt     -47.2763     91.060     -0.519      0.604    -225.949     131.397
GarageCars     1.685e+04   3490.579      4.827      0.000       1e+04    2.37e+04
GarageArea        6.2744     12.127      0.517      0.605     -17.521      30.070
WoodDeckSF       21.4407     10.024      2.139      0.033       1.772      41.109
OpenPorchSF      -2.2524     19.486     -0.116      0.908     -40.486      35.982
EnclosedPorch     7.2949     20.621      0.354      0.724     -33.167      47.757
3SsnPorch        33.4852     37.584      0.891      0.373     -40.261     107.232
ScreenPorch      58.0465     20.407      2.844      0.005      18.005      98.088
PoolArea        -60.5171     29.898     -2.024      0.043    -119.182      -1.852
MiscVal          -3.7615      6.960     -0.540      0.589     -17.419       9.896
MoSold         -221.6980    422.859     -0.524      0.600   -1051.411     608.015
YrSold         -247.4485    845.813     -0.293      0.770   -1907.064    1412.167
==============================================================================
Omnibus:                      433.915   Durbin-Watson:                   1.941
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            64998.981
Skew:                          -0.670   Prob(JB):                         0.00
Kurtosis:                      40.280   Cond. No.                     1.21e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.39e-21. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

In [47]:

```python
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Calculamos el VIF para cada variable en X
vif = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

# Creamos un DataFrame para mostrar los resultados
pd.DataFrame({'VIF': vif}, index=X.columns)
```

Out[47]:

	VIF
const	2.396251e+06
Id	1.034686e+00
MSSubClass	1.718790e+00
LotFrontage	1.827885e+00
LotArea	1.356288e+00
OverallQual	3.461493e+00
OverallCond	1.765745e+00
YearBuilt	6.094850e+00
YearRemodAdd	2.747160e+00
MasVnrArea	1.464423e+00
BsmtFinSF1	inf
BsmtFinSF2	inf
BsmtUnfSF	inf
TotalBsmtSF	inf
1stFlrSF	inf
2ndFlrSF	inf
LowQualFinSF	inf
GrLivArea	inf
BsmtFullBath	2.219913e+00
BsmtHalfBath	1.151092e+00
FullBath	3.120675e+00
HalfBath	2.269333e+00
BedroomAbvGr	2.288685e+00
KitchenAbvGr	1.593948e+00
TotRmsAbvGrd	4.631822e+00
Fireplaces	1.585157e+00
GarageYrBlt	4.572723e+00
GarageCars	4.314005e+00
GarageArea	4.448564e+00
WoodDeckSF	1.234169e+00
OpenPorchSF	1.301930e+00
EnclosedPorch	1.320733e+00
3SsnPorch	1.035532e+00
ScreenPorch	1.150664e+00
PoolArea	1.196011e+00
MiscVal	1.100833e+00
MoSold	1.068357e+00
YrSold	1.054518e+00

En este caso las variables que presentan un 'VIF' infinito (BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea), asi como tambien YearBuilt.

In [52]:

```
# Importar las librerías necesarias
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Definir las variables: independiente (fertility) y dependiente (life)
X = df["GarageCars"].values
y = df["GarageArea"].values

# Redimensionar X para que tenga el formato correcto (n_samples, n_features)
X = X.reshape(-1, 1)

# Crear el modelo de regresión lineal
modelo = LinearRegression()
```

```python
# Entrenar el modelo con los datos
modelo.fit(X, y)

# Realizar predicciones sobre los datos de entrada
predicciones = modelo.predict(X)

# Calcular el error cuadrático medio (RMSE)
rmse = np.sqrt(mean_squared_error(y, predicciones))

# Calcular el coeficiente de determinación R^2
r2 = modelo.score(X, y)

# Imprimir los resultados
print("R²: {:.2f}".format(r2))
print("RMSE: {:.2f}".format(rmse))
```

```
R²: 0.78
RMSE: 100.53
```

Conclusion

Para el modelo de regresion lineal, tome las variables de GarageCars y GarageArea, porque eran las que mayor correlacion tenian, al correr el modelo, la R2 fue muy positiva, teniendo 0.78 como resultado.

In [ ]: