Exploración básica de Pandas — Diabetes

Nombre: Camille Yamilé Caltenco García

Matrícula: A00842805

In [63]:
```
#importa librerías
import pandas as pd
```

Descripción de Variables

Pclass Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd): Categórica Nominal survival Survival (0 = No; 1 = Yes)
name Name
sex Sex
age Age
sibsp Number of Siblings/Spouses Aboard
parch Number of Parents/Children Aboard
ticket Ticket Number
fare Passenger Fare (British pound)
cabin Cabin
embarked Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)
boat Lifeboat
body Body Identification Number
home.dest Home/Destination

Ejemplo: Crear un objeto DataFrame con base en un archivo .csv

In [64]:
```
#Lee archivo csv
df = pd.read_csv("diabetes.csv")
```

In [65]:
```
#Usa función shape para revisar el total de renglones y columnas
df.shape
```

Out[65]: (768, 9)

In [66]:
```
#Revisa los primeros 5 renglones del dataset usando la función head()
df.head()
```

Out[66]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [67]:
```
#Revisa los últimos 5 renglones del dataset usando la función tail()
df.tail() #default 5
```

Out[67]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

In [68]:
```
#Revisa la información mas completa del conjunto de datos usando la función info()
#Muestra el total de datos, las columnas y su tipo correspondiente, dice si contiene nulos o no
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

los atributos age y cabin tienen valores No nulos.

In [69]:
```
#revisa cuántos valores únicos tiene cada atributo del archivo usando la función nunique()
df.nunique()
```

Out[69]:
```
Pregnancies                 17
Glucose                    136
BloodPressure               47
SkinThickness               51
Insulin                    186
BMI                        248
DiabetesPedigreeFunction   517
Age                         52
Outcome                      2
dtype: int64
```

Exploración de Datos

In [9]:
```
#utiliza la función describe() para obtener estadística básica. se puede incluir -0
df.describe()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [70]:
```
df[['DiabetesPedigreeFunction', 'Age', 'Outcome']].describe()
```

Out[70]:

	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000
mean	0.471876	33.240885	0.348958
std	0.331329	11.760232	0.476951
min	0.078000	21.000000	0.000000
25%	0.243750	24.000000	0.000000
50%	0.372500	29.000000	0.000000
75%	0.626250	41.000000	1.000000
max	2.420000	81.000000	1.000000

Análisis e Interpretación

- **DiabetesPedigreeFunction:**
  La media (0.47) y mediana (0.37) indican que la mayoría de los pacientes tienen un riesgo familiar bajo.
  La desviación estándar (0.33) muestra una variabilidad moderada.

- **Age:**
  La media (33 años) sugiere que la mayoría de los pacientes son adultos jóvenes.
  La desviación estándar (11 años) indica que hay personas desde los 20 hasta más de 70 años.

- **Outcome:**
  El valor promedio (0.35) muestra que aproximadamente el 35% de los pacientes del conjunto tienen diabetes.
  Esto sugiere que la mayoría de los registros son de personas sin diabetes.

In [71]:
```
#Revisa Valores nulos con funcion isnull().sum()
df.isnull().sum()
```

Out[71]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [72]:
```
#Revisar valores únicos por columna usando función unique(): nombre-columna.unique()
df.Age.unique()
```

Out[72]:
```
array([50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 34, 57, 59, 51, 27, 41, 43,
       22, 38, 60, 28, 45, 35, 46, 56, 37, 48, 40, 25, 24, 58, 42, 44, 39,
       36, 23, 61, 69, 62, 55, 65, 47, 52, 66, 49, 63, 67, 72, 81, 64, 70,
       68])
```

In [73]:
```
df.DiabetesPedigreeFunction.unique()
```

Out[73]:
```
array([0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.248, 0.134, 0.158,
       0.232, 0.191, 0.537, 1.441, 0.398, 0.587, 0.484, 0.551, 0.254,
       0.183, 0.529, 0.704, 0.388, 0.451, 0.263, 0.205, 0.257, 0.487,
       0.245, 0.337, 0.546, 0.851, 0.267, 0.188, 0.512, 0.966, 0.42 ,
       0.665, 0.503, 1.39 , 0.271, 0.696, 0.235, 0.721, 0.294, 1.893,
       0.564, 0.586, 0.344, 0.305, 0.491, 0.526, 0.342, 0.467, 0.718,
       0.962, 1.781, 0.173, 0.304, 0.27 , 0.699, 0.258, 0.203, 0.855,
       0.845, 0.334, 0.189, 0.867, 0.411, 0.583, 0.231, 0.396, 0.14 ,
       0.391, 0.37 , 0.307, 0.102, 0.767, 0.237, 0.227, 0.698, 0.178,
       0.324, 0.153, 0.165, 0.443, 0.261, 0.277, 0.761, 0.255, 0.13 ,
       0.323, 0.356, 0.325, 1.222, 0.179, 0.262, 0.283, 0.93 , 0.801,
       0.207, 0.287, 0.336, 0.247, 0.199, 0.543, 0.192, 0.588, 0.539,
       0.22 , 0.654, 0.223, 0.759, 0.26 , 0.404, 0.186, 0.278, 0.496,
       0.452, 0.403, 0.741, 0.361, 1.114, 0.457, 0.647, 0.088, 0.597,
       0.532, 0.703, 0.159, 0.268, 0.286, 0.318, 0.272, 0.572, 0.096,
       1.4  , 0.218, 0.085, 0.399, 0.432, 1.189, 0.687, 0.137, 0.637,
       0.833, 0.229, 0.817, 0.204, 0.368, 0.743, 0.722, 0.256, 0.709,
       0.471, 0.495, 0.18 , 0.542, 0.773, 0.678, 0.719, 0.382, 0.319,
       0.19 , 0.956, 0.084, 0.725, 0.299, 0.244, 0.745, 0.615, 1.321,
       0.64 , 0.142, 0.374, 0.383, 0.578, 0.136, 0.395, 0.187, 0.905,
       0.15 , 0.874, 0.236, 0.787, 0.407, 0.605, 0.151, 0.289, 0.355,
       0.29 , 0.375, 0.164, 0.431, 0.742, 0.514, 0.464, 1.224, 1.072,
       0.805, 0.209, 0.666, 0.101, 0.198, 0.652, 2.329, 0.089, 0.645,
       0.238, 0.394, 0.293, 0.479, 0.686, 0.831, 0.582, 0.446, 0.402,
       1.318, 0.329, 1.213, 0.427, 0.282, 0.143, 0.38 , 0.284, 0.249,
       0.926, 0.557, 0.092, 0.655, 1.353, 0.612, 0.2  , 0.226, 0.997,
       0.933, 1.101, 0.078, 0.24 , 1.136, 0.128, 0.422, 0.251, 0.677,
       0.296, 0.454, 0.744, 0.881, 0.28 , 0.259, 0.619, 0.808, 0.34 ,
       0.434, 0.757, 0.613, 0.692, 0.52 , 0.412, 0.84 , 0.839, 0.156,
       0.215, 0.326, 1.391, 0.875, 0.313, 0.433, 0.626, 1.127, 0.315,
       0.345, 0.129, 0.527, 0.197, 0.731, 0.148, 0.123, 0.127, 0.122,
       1.476, 0.166, 0.932, 0.343, 0.893, 0.331, 0.472, 0.673, 0.389,
       0.485, 0.349, 0.279, 0.346, 0.252, 0.243, 0.58 , 0.559, 0.302,
       0.569, 0.378, 0.385, 0.499, 0.306, 0.234, 2.137, 1.731, 0.545,
       0.225, 0.816, 0.528, 0.509, 1.021, 0.821, 0.947, 1.268, 0.221,
       0.66 , 0.239, 0.949, 0.444, 0.463, 0.803, 1.6  , 0.944, 0.196,
       0.241, 0.161, 0.135, 0.376, 1.191, 0.702, 0.674, 1.076, 0.534,
       1.095, 0.554, 0.624, 0.219, 0.507, 0.561, 0.421, 0.516, 0.264,
       0.328, 0.233, 0.108, 1.138, 0.147, 0.727, 0.435, 0.497, 0.23 ,
       0.955, 2.42 , 0.658, 0.33 , 0.51 , 0.285, 0.415, 0.381, 0.832,
       0.498, 0.212, 0.364, 1.001, 0.46 , 0.733, 0.416, 0.705, 1.022,
       0.269, 0.6  , 0.571, 0.607, 0.17 , 0.21 , 0.126, 0.711, 0.466,
       0.162, 0.419, 0.63 , 0.365, 0.536, 1.159, 0.629, 0.292, 0.145,
       1.144, 0.174, 0.547, 0.163, 0.738, 0.314, 0.968, 0.409, 0.297,
       0.525, 0.154, 0.771, 0.107, 0.493, 0.717, 0.917, 0.501, 1.251,
       0.735, 0.804, 0.661, 0.549, 0.825, 0.423, 1.034, 0.16 , 0.341,
       0.68 , 0.591, 0.3  , 0.121, 0.502, 0.401, 0.601, 0.748, 0.338,
       0.43 , 0.892, 0.813, 0.693, 0.575, 0.371, 0.206, 0.417, 1.154,
       0.925, 0.175, 1.699, 0.682, 0.194, 0.4  , 0.1  , 1.258, 0.482,
       0.138, 0.593, 0.878, 0.157, 1.282, 0.141, 0.246, 1.698, 1.461,
       0.347, 0.362, 0.393, 0.144, 0.732, 0.115, 0.465, 0.649, 0.871,
       0.149, 0.695, 0.303, 0.61 , 0.73 , 0.447, 0.455, 0.133, 0.155,
       1.162, 1.292, 0.182, 1.394, 0.217, 0.631, 0.88 , 0.614, 0.332,
       0.366, 0.181, 0.828, 0.335, 0.856, 0.886, 0.439, 0.253, 0.598,
       0.904, 0.483, 0.565, 0.118, 0.177, 0.176, 0.295, 0.441, 0.352,
       0.826, 0.97 , 0.595, 0.317, 0.265, 0.646, 0.426, 0.56 , 0.515,
       0.453, 0.785, 0.734, 1.174, 0.488, 0.358, 1.096, 0.408, 1.182,
       0.222, 1.057, 0.766, 0.171])
```

In [74]:
```
df.Outcome.unique()
```

Out[74]:
```
array([1, 0])
```

Variables Cuantitativas

Medidas de tendencia central

In [18]:
```
#Edad
#Se puede obtener la media, mediana y moda para
mean_age = df['Age'].mean()
median_age = df['Age'].median()
mode_age = df['Age'].mode()
print("Mean_age:",mean_age)
print("Median_age:",median_age)
print("Mode_age:",mode_age)
```

```
Mean_age: 33.240885416666664
Median_age: 29.0
Mode_age: 0    22
Name: Age, dtype: int64
```

In [21]:
```
#Outcome
#Se puede obtener la media, mediana y moda para
mean_Outcome = df['Outcome'].mean()
median_Outcome = df['Outcome'].median()
mode_Outcome = df['Outcome'].mode()
print("Mean_Outcome:",mean_Outcome)
print("Median_Outcome:",median_Outcome)
print("Mode_Outcome:",mode_Outcome)
```

```
Mean_Outcome: 0.3489583333333333
Median_Outcome: 0.0
Mode_Outcome: 0    0
Name: Outcome, dtype: int64
```

In [22]:
```
#DiabetesPedigreeFunction
#Se puede obtener la media, mediana y moda para
mean_DiabetesPedigreeFunction = df['DiabetesPedigreeFunction'].mean()
median_DiabetesPedigreeFunction = df['DiabetesPedigreeFunction'].median()
mode_DiabetesPedigreeFunction = df['DiabetesPedigreeFunction'].mode()
print("Mean_DiabetesPedigreeFunction:",mean_DiabetesPedigreeFunction)
print("Median_DiabetesPedigreeFunction:",median_DiabetesPedigreeFunction)
print("Mode_DiabetesPedigreeFunction:",mode_DiabetesPedigreeFunction)
```

```
Mean_DiabetesPedigreeFunction: 0.47187630208333325
Median_DiabetesPedigreeFunction: 0.3725
Mode_DiabetesPedigreeFunction: 0    0.254
1    0.258
Name: DiabetesPedigreeFunction, dtype: float64
```

Descripción de Variables Analizadas

DiabetesPedigreeFunction:

Tipo: Cuantitativa continua

Representa una función que estima la probabilidad de tener diabetes según los antecedentes familiares.

Rango: de 0.078 a 2.42

Age:

Tipo: Cuantitativa discreta

Representa la edad de las personas en años.

Rango: de 21 a 81

Outcome:

Tipo: Categórica

Indica si el paciente tiene diabetes (1) o no (0).

Rango: 0 o 1

Variables Categóricas

In [75]:
```
#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
# nombreDataframe.columna.value_counts()
# nombreDataframe['columna'].value_counts()
df.Age.value_counts()
```

Out[75]:
```
Age
22    72
21    63
25    48
24    46
23    38
28    35
26    33
27    32
29    29
31    24
41    22
30    21
37    19
42    18
33    17
36    16
38    16
32    16
45    15
34    14
46    13
40    13
43    13
39    12
35    10
44     8
50     8
51     8
52     8
58     7
54     6
47     6
49     5
60     5
53     5
57     5
48     5
63     4
66     4
55     4
62     4
59     3
56     3
65     3
67     3
61     2
69     2
72     1
81     1
64     1
70     1
68     1
Name: count, dtype: int64
```

In [76]:
```
#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
# nombreDataframe.columna.value_counts()
# nombreDataframe['columna'].value_counts()
df.DiabetesPedigreeFunction.value_counts()
```

Out[76]:
```
DiabetesPedigreeFunction
0.258    6
0.254    6
0.268    5
0.207    5
0.261    5
        ..
0.427    1
1.213    1
0.329    1
1.318    1
0.933    1
Name: count, Length: 517, dtype: int64
```

In [77]:
```
#Para conteo  de cada valor en una columna, en orden descendente usar función value_counts():
# nombreDataframe.columna.value_counts()
# nombreDataframe['columna'].value_counts()
df.Outcome.value_counts()
```

Out[77]:
```
Outcome
0    500
1    268
Name: count, dtype: int64
```

Consulta

In [78]:
```
# df.iloc[i]: Accede a la fila en la posición i.
# Acceder a la primera fila
df.iloc[1]
```

Out[78]:
```
Pregnancies                 1.000
Glucose                    85.000
BloodPressure              66.000
SkinThickness              29.000
Insulin                     0.000
BMI                        26.600
DiabetesPedigreeFunction    0.351
Age                        31.000
Outcome                     0.000
Name: 1, dtype: float64
```

In [79]:
```
# Acceder a las dos primeras filas
df.iloc[1:3]
```

Out[79]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1

In [80]:
```
# 1. Promedio de la función de parentesco en personas con y sin diabetes
df.groupby('Outcome')['DiabetesPedigreeFunction'].mean()
```

Out[80]:
```
Outcome
0    0.429734
1    0.550500
Name: DiabetesPedigreeFunction, dtype: float64
```

In [81]:
```
# 2. Edad promedio de las personas con diabetes
df[df['Outcome'] == 1]['Age'].mean()
```

Out[81]:
```
np.float64(37.06716417910448)
```

In [82]:
```
# 3. Porcentaje de personas mayores de 50 años con diabetes
(df[(df['Age'] > 50) & (df['Outcome'] == 1)].shape[0] / df[df['Age'] > 50].shape[0]) * 100
```

Out[82]:
```
46.913580246913575
```

In [83]:
```
df
```

Out[83]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Conclusiones

El análisis muestra que el riesgo familiar (DiabetesPedigreeFunction) se relaciona con la presencia de diabetes, aunque no es el único factor.
La mayoría de los pacientes son adultos jóvenes, pero los casos aumentan en edades mayores.

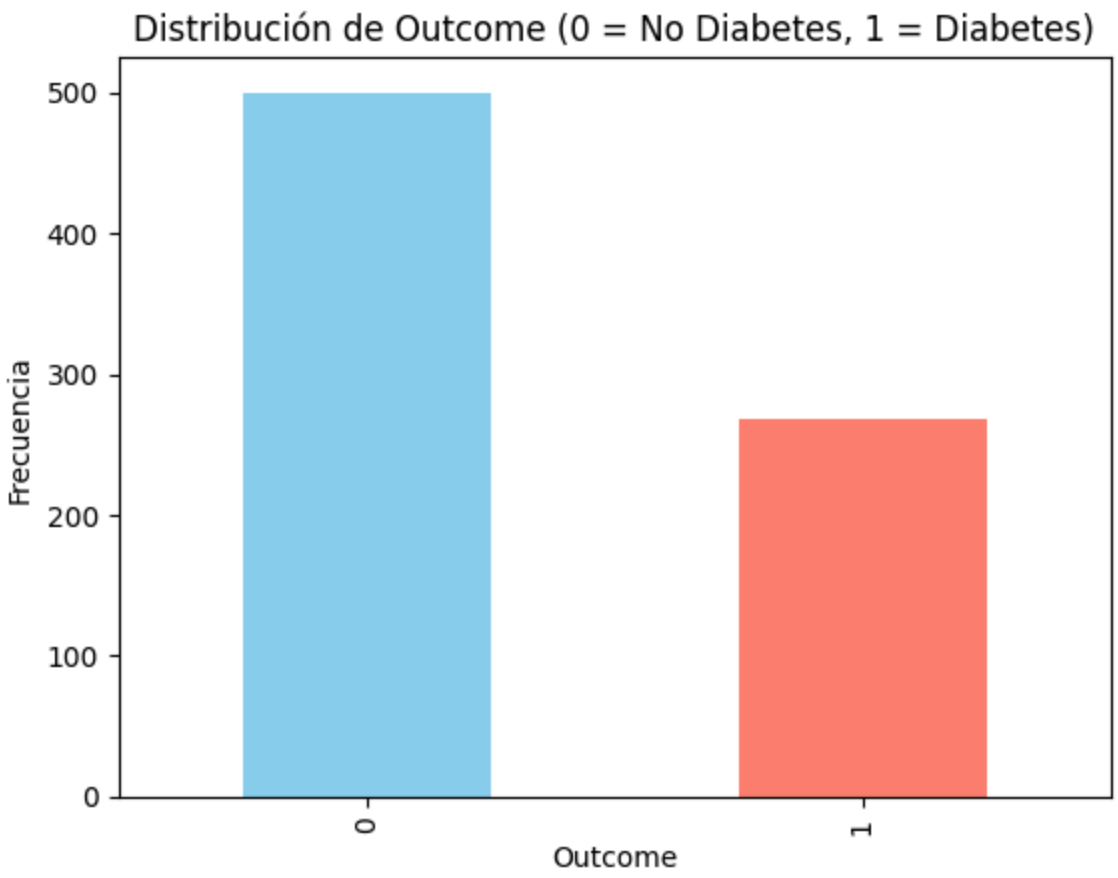
Visualización de Datos

Variable Categóricas

Variable 1

En este caso, la única variable categórica es **Outcome**, que indica si la persona tiene diabetes (1) o no (0).
Se muestra una gráfica de barras para visualizar la cantidad de casos en cada categoría.

In [91]:
```
# Gráfica de barras para la variable categórica Outcome
datosOutcome = df['Outcome'].value_counts()
datosOutcome.plot(kind='bar', color=['skyblue', 'salmon'])
plt.title('Distribución de Outcome (0 = No Diabetes, 1 = Diabetes)')
plt.xlabel('Outcome')
plt.ylabel('Frecuencia')
plt.show()
```



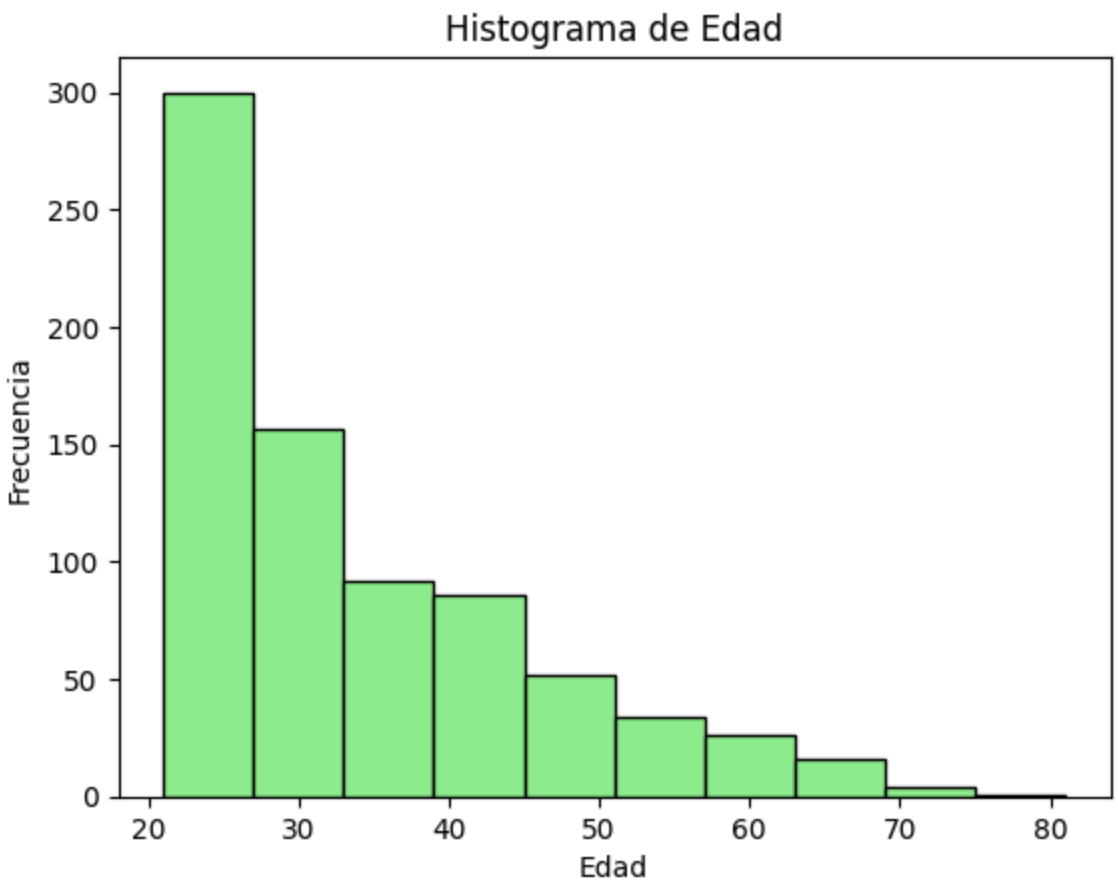
Análisis

Esta gráfica muestra la cantidad de personas con diabetes (1) y sin diabetes (0). Se observa que hay más personas sin diabetes, lo que indica que el conjunto de datos está desbalanceado hacia los casos negativos.

Variable 2: Edad

Numérica

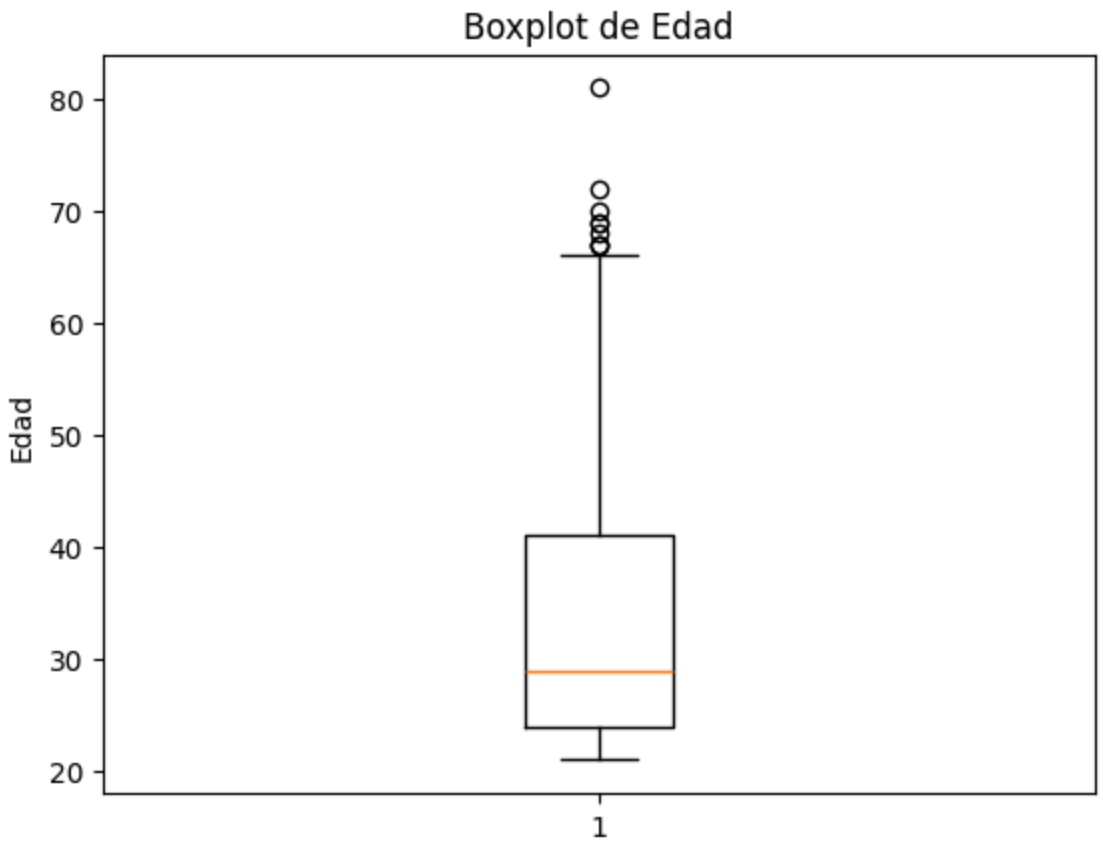
In [96]:
```
# Histograma de la variable Age
plt.hist(df['Age'], bins=10, color='lightgreen', edgecolor='black')
plt.title('Histograma de Edad')
plt.xlabel('Edad')
plt.ylabel('Frecuencia')
plt.show()
```



Análisis

El histograma muestra cómo se distribuyen las edades. La mayoría de las personas tienen entre 20 y 40 años, y hay pocos casos mayores de 60. Esto indica que la mayoría de los pacientes son adultos jóvenes.

In [97]:
```
# Boxplot de la variable Age
plt.boxplot(df['Age'])
plt.title('Boxplot de Edad')
plt.ylabel('Edad')
plt.show()
```



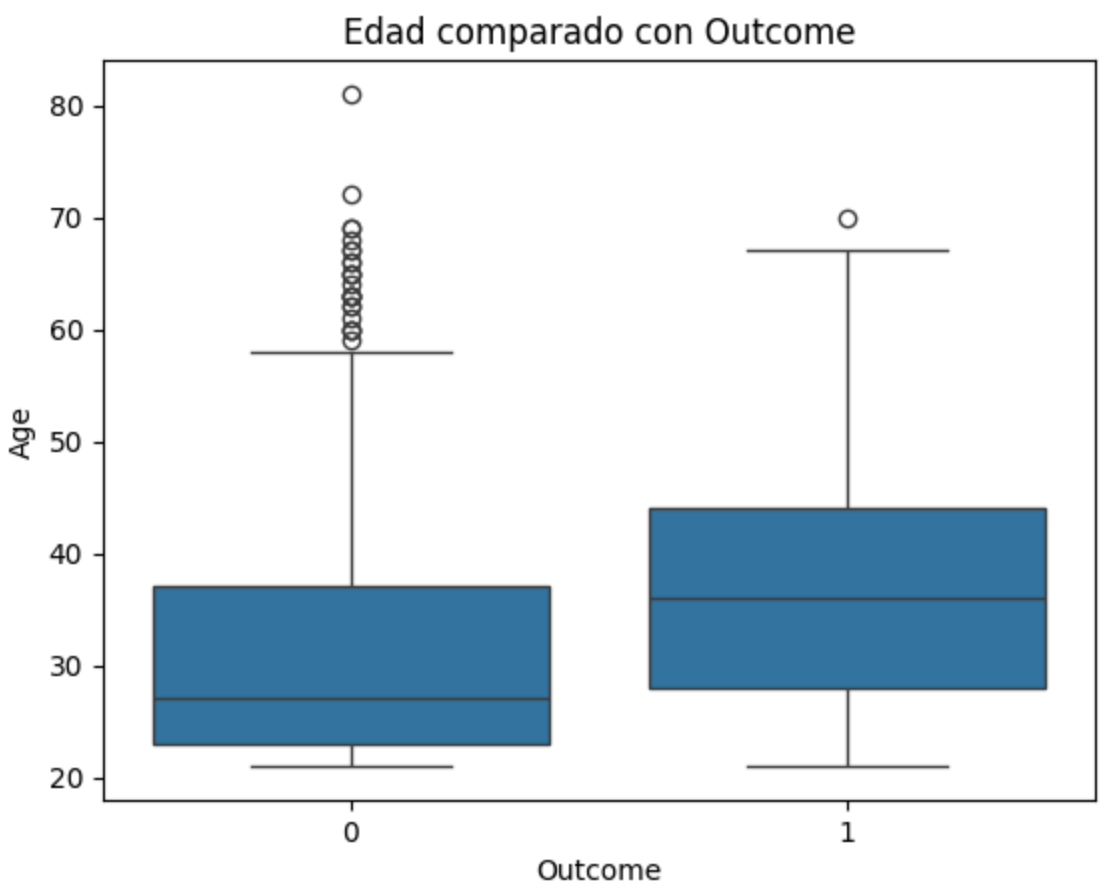
Análisis

El diagrama muestra que la edad tiene una mediana alrededor de los 30 años, y que los valores están concentrados entre los 20 y 45 años. No hay muchos valores extremos.

Comparación con Outcome

In [111…
```
sns.boxplot(df, x="Outcome", y="Age")
plt.title("Edad comparado con Outcome")
```

Out[111…
```
Text(0.5, 1.0, 'Edad comparado con Outcome')
```



Análisis

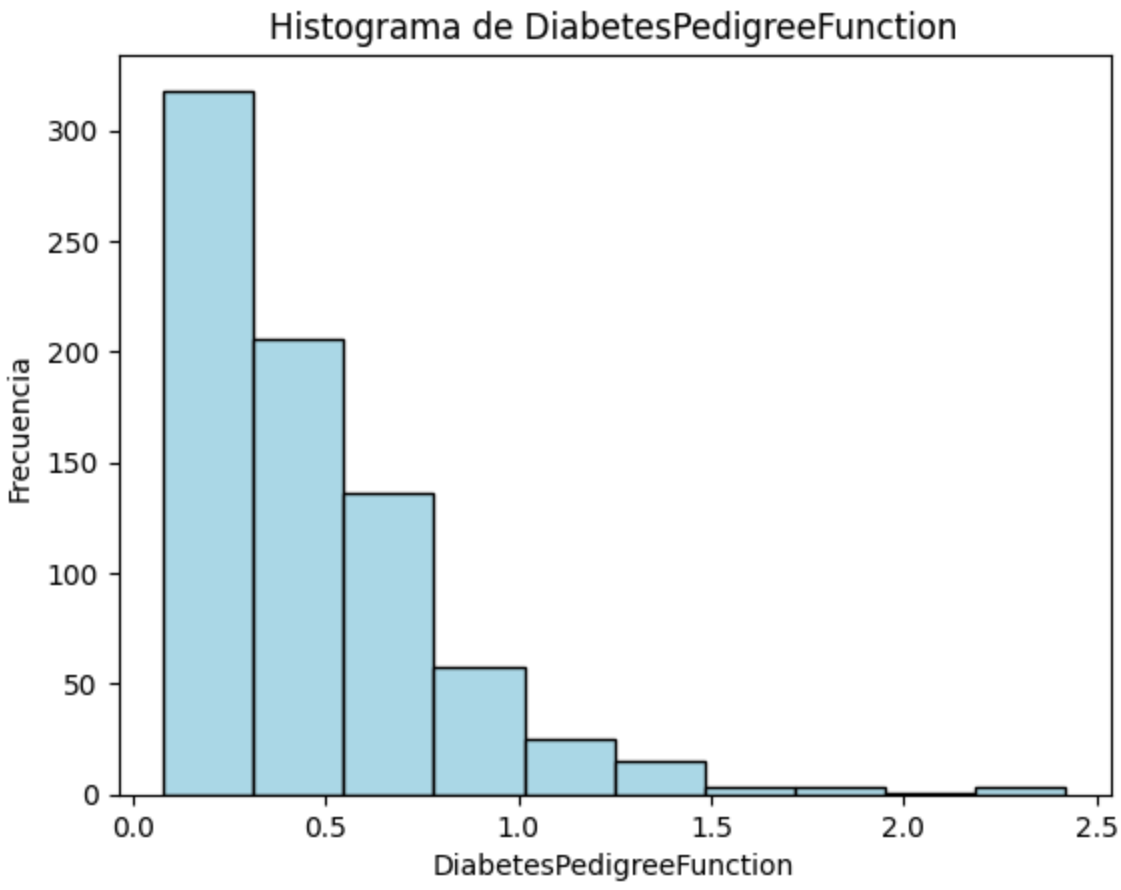
En la comparación de la variable Age con Outcome, se observa que las personas que tienen diabetes (Outcome = 1) son, en promedio, más grandes de edad que las que no la tienen (Outcome = 0). El promedio de edad para quienes no tienen diabetes es de alrededor de 31 años, mientras que para quienes sí tienen diabetes es de aproximadamente 37 años.

Esto significa que la edad influye en la probabilidad de tener diabetes, ya que conforme las personas son mayores. La diferencia no es enorme, pero sí muestra una tendencia clara: a mayor edad, mayor riesgo de desarrollar diabetes.

Variable 3: DiabetesPedigreeFunction

Numérica

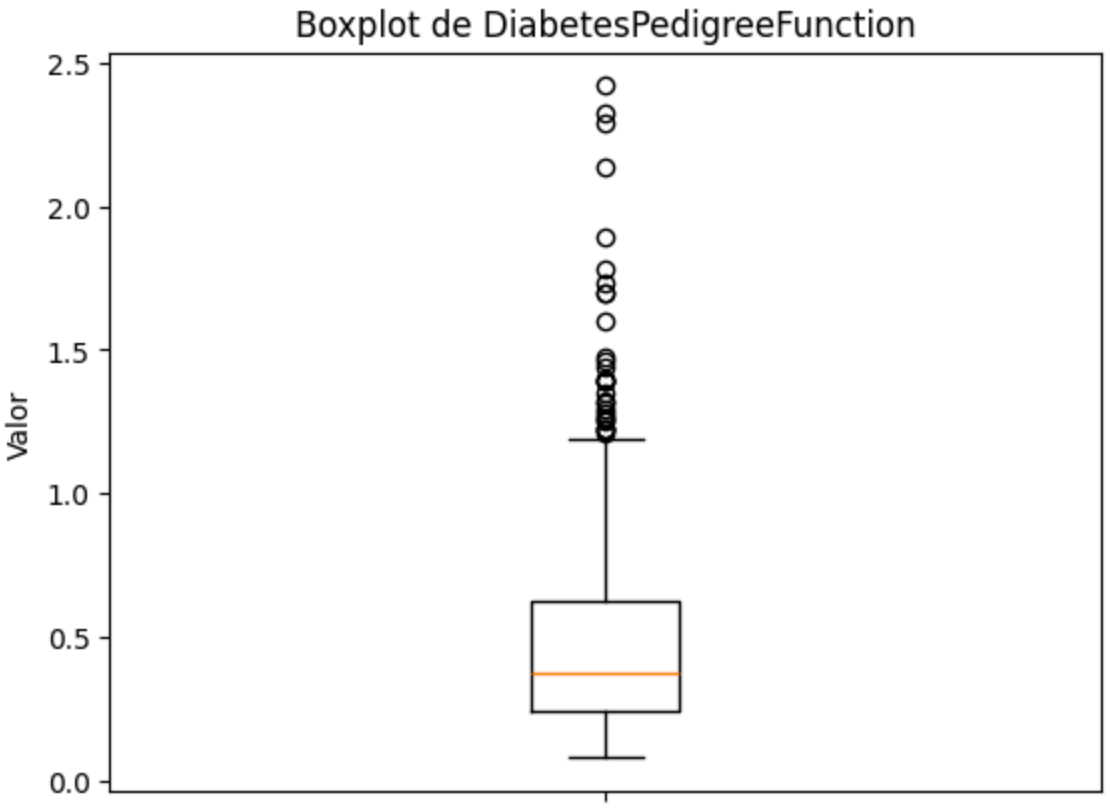
In [93]:
```
# Histograma de la variable DiabetesPedigreeFunction
plt.hist(df['DiabetesPedigreeFunction'], bins=10, color='lightblue', edgecolor='black')
plt.title('Histograma de DiabetesPedigreeFunction')
plt.xlabel('DiabetesPedigreeFunction')
plt.ylabel('Frecuencia')
plt.show()
```



Análisis

Esta gráfica muestra los valores del "riesgo hereditario de diabetes". La mayoría de los valores están entre 0.1 y 0.6, lo que sugiere que la mayoría tiene un riesgo bajo o medio.

In [98]:
```
# Boxplot de la variable DiabetesPedigreeFunction
plt.boxplot(df['DiabetesPedigreeFunction'])
plt.title('Boxplot de DiabetesPedigreeFunction')
plt.ylabel('Valor')
plt.show()
```



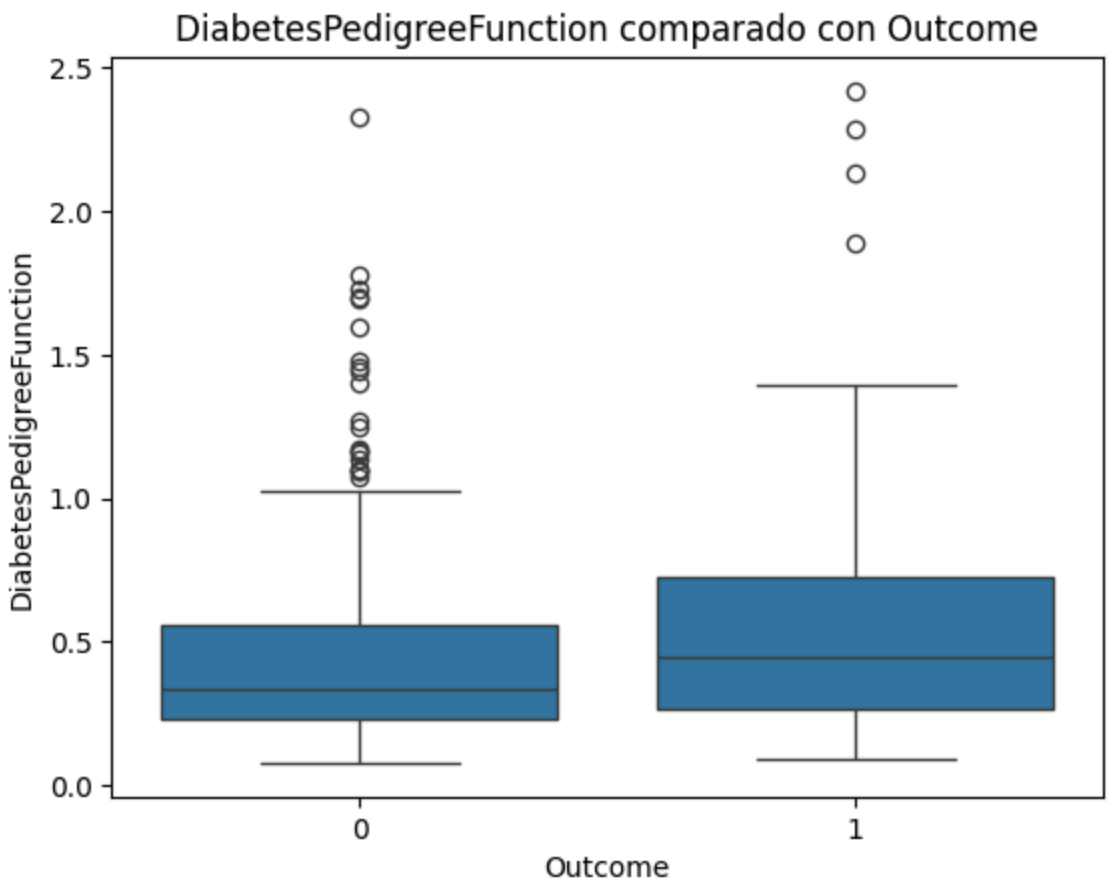
Análisis

El diagrama confirma que la mayoría de los valores son bajos, y se observan algunos puntos extremos arriba de 2. Estos casos podrían ser personas con alto riesgo familiar de diabetes.

Comparación con Outcome

In [113…
```
sns.boxplot(df, x="Outcome", y="DiabetesPedigreeFunction")
plt.title("DiabetesPedigreeFunction comparado con Outcome")
```

Out[113…
```
Text(0.5, 1.0, 'DiabetesPedigreeFunction comparado con Outcome')
```



Análisis

Al comparar la variable DiabetesPedigreeFunction (riesgo hereditario) con Outcome, se nota que las personas con diabetes presentan valores promedio más altos en esta función. Esto quiere decir que quienes tienen antecedentes familiares más fuertes de diabetes tienen mayor probabilidad de padecerla.

Análisis de Correlación

A continuación se obtiene la matriz de correlación entre las variables numéricas visualizado mediante un mapa de calor.

In [105…

```
#seleccionar variables numéricas
variables_numericas = df.select_dtypes(include='number')
matriz_correlacion = variables_numericas.corr().round(2)
matriz_correlacion
```

Out[105…

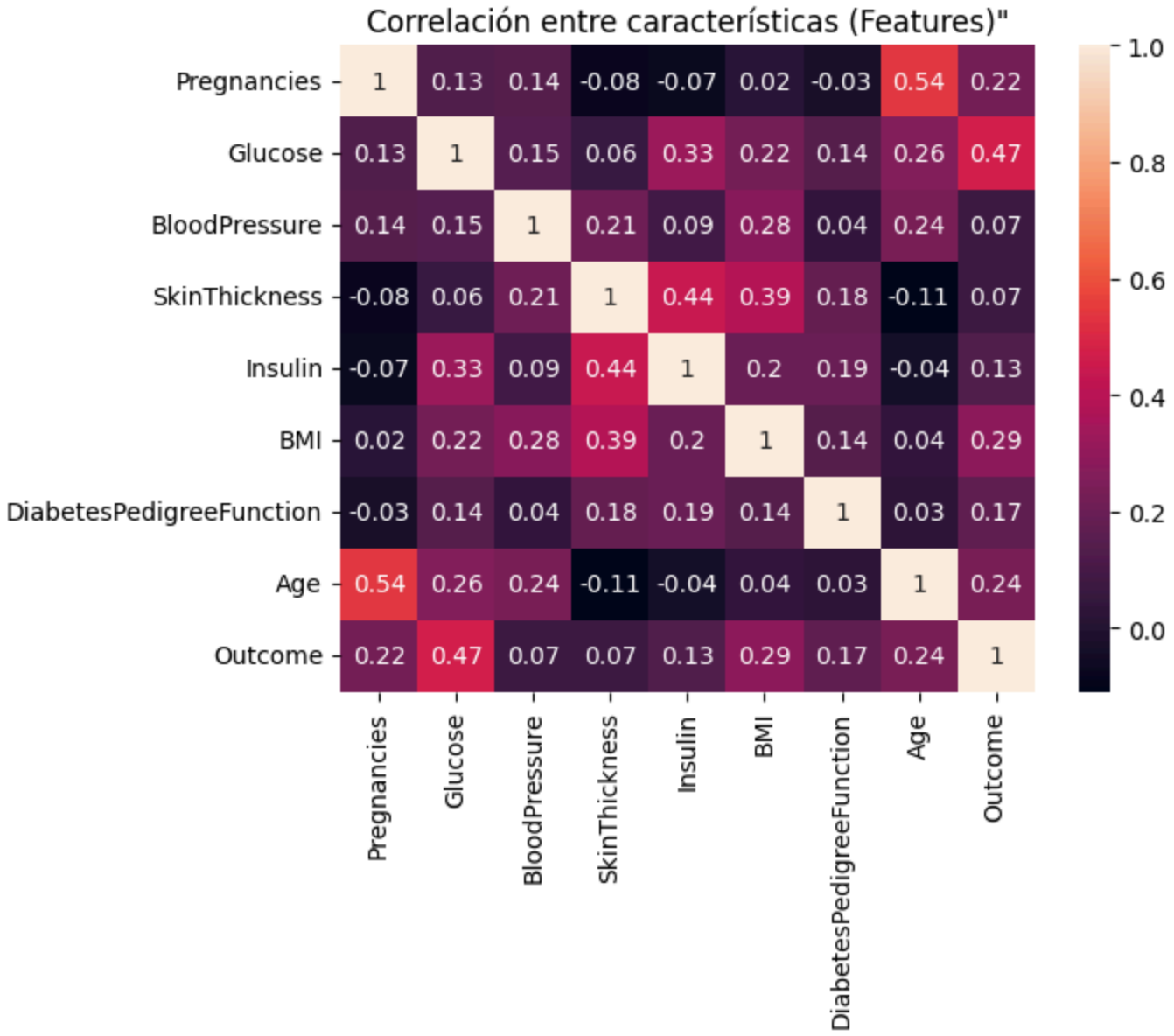
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.00	0.13	0.14	-0.08	-0.07	0.02	-0.03	0.54	0.22
Glucose	0.13	1.00	0.15	0.06	0.33	0.22	0.14	0.26	0.47
BloodPressure	0.14	0.15	1.00	0.21	0.09	0.28	0.04	0.24	0.07
SkinThickness	-0.08	0.06	0.21	1.00	0.44	0.39	0.18	-0.11	0.07
Insulin	-0.07	0.33	0.09	0.44	1.00	0.20	0.19	-0.04	0.13
BMI	0.02	0.22	0.28	0.39	0.20	1.00	0.14	0.04	0.29
DiabetesPedigreeFunction	-0.03	0.14	0.04	0.18	0.19	0.14	1.00	0.03	0.17
Age	0.54	0.26	0.24	-0.11	-0.04	0.04	0.03	1.00	0.24
Outcome	0.22	0.47	0.07	0.07	0.13	0.29	0.17	0.24	1.00

In [104…

```
#Mostrar mapa de calor.
sns.heatmap(matriz_correlacion, annot=True)
plt.title('Correlación entre características (Features)"')
```

Out[104…

```
Text(0.5, 1.0, 'Correlación entre características (Features)"')
```



Análisis

El mapa de calor muestra la correlación entre todas las variables numéricas.

La variable Age tiene correlación positiva moderada (0.24) con Outcome, indicando que la edad también influye.

Las demás variables (como la DiabetesPedigreeFunction anteriormente analizada) tienen correlaciones bajas o casi nulas.

Conclusiones

En general, el análisis del conjunto de datos de diabetes muestra que:

La mayoría de las personas no tienen diabetes.

Los pacientes suelen ser adultos jóvenes, aunque los casos aumentan con la edad.

El riesgo hereditario (DiabetesPedigreeFunction) suele ser bajo, pero hay algunos casos altos que podrían influir en el diagnóstico.

In [ ]: