0.- Imports necesarios para el clustering de los datos

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import copy
import json
import joblib
import plotly.express as px
import warnings
from numpy import unique,where

from sklearn.preprocessing import OneHotEncoder, MinMaxScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from sklearn.cluster import KMeans
from sklearn_extra.cluster import KMedoids
from sklearn.cluster import AffinityPropagation
from sklearn.cluster import OPTICS
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score, davies_bouldin_score,
silhouette_samples
```

1.- Leer el fichero de datos de los pacientes con diabetes

Como en el análisis estadístico ya hemos limpiado y preprocesado el dataset solo tenemos que cargarlo y quedarnos con las variables relevantes

1.1.- Cargamos el fichero en la variable df

```
# Path al fichero de datos
file_path = 'diabetes_data_clean.csv'

# Leemos el fichero como un df de Pandas
df = pd.read_csv(file_path)

# Imprimimos la cabecera para ver si ha cargado
df.head()
```

```
   Diabetes_012  HighBP  HighChol  CholCheck  BMI  Smoker  Stroke  \
0             0       1         1          1   40       1       0
1             0       0         0          0   25       1       0
2             0       1         1          1   28       0       0
3             0       1         0          1   27       0       0
4             0       1         1          1   24       0       0
```

```
   HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare  \
0                     0             0       0  ...              1
1                     0             1       0  ...              0
2                     0             0       1  ...              1
3                     0             1       1  ...              1
4                     0             1       1  ...              1

   NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex  Age  
Education  \
0            0        5        18        15         1    0    9  
4
1            1        3         0         0         0    0    7  
6
2            1        5        30        30         1    0    9  
4
3            0        2         0         0         0    0   11  
3
4            0        2         3         0         0    0   11  
5

   Income
0       3
1       1
2       8
3       6
4       4

[5 rows x 22 columns]
```

1.2.- Obeservamos dimensiones del dataset y los nombres de columnas con sus tipos

```
# Numero de filas y columnas en el df
filas, columnas = df.shape
print(f"Numero de filas: {filas}")
print(f"Numero de columnas: {columnas}")
```

```
Numero de filas: 229781
Numero de columnas: 22
```

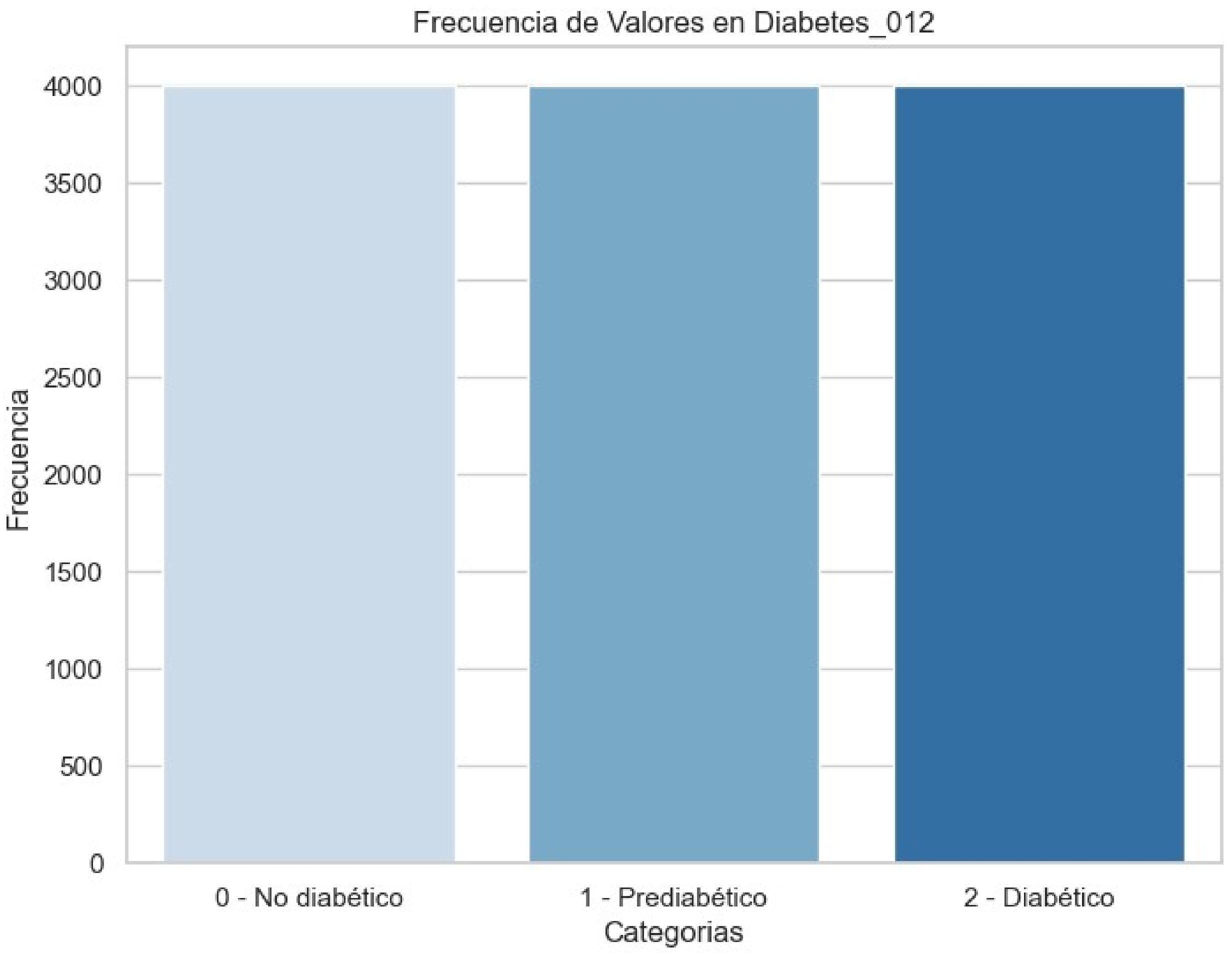
```
# Nombres de las columnas
nombres_columnas = df.columns
print("Columnas del df:", nombres_columnas)
```

```
Columnas del df: Index(['Diabetes_012', 'HighBP', 'HighChol', 
'CholCheck', 'BMI', 'Smoker',
       'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 
'Veggies',
       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
       'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
       'Income'],
      dtype='object')
```

1.3.- Reducción del dataset

A diferencia del K means el resto de algorimtos requieren de mucha RAM, más de la que se puede emplear en el desarrollo de este TFM. Por ellos vamos a quedarnos con una muestra significativa del dataset con 4000 de cada categoria de diabetes a partir de ahora para hacer el clustering

```python
# Función que coge un conjunto random de tamaño 5000 para cada grupo
def get_random_sample(group, n=5000):
    return group.sample(n=min(n, group.shape[0]), random_state=42)

clustering_df_original = copy.deepcopy(df)

df_sampled = df.groupby('Diabetes_012',
group_keys=False).apply(get_random_sample, n=4000)

df_sampled.reset_index(inplace=True)

# Definimos un mapeo de etiquetas para los valores en 'Diabetes_012'
etiquetas = {0: 'No diabético', 1: 'Prediabético', 2: 'Diabético'}

# Cereamos una copia a modificar
df_graficas = df_sampled.copy()
df_graficas['Diabetes_012'] =
df_graficas['Diabetes_012'].map(etiquetas)

# Gráfica de barras con barras estrechas, color azul, cuadrícula y
leyenda
plt.figure(figsize=(8, 6))
sns.set(style="whitegrid")
ax = sns.countplot(data=df_graficas, x='Diabetes_012',
palette="Blues")
plt.title("Frecuencia de Valores en Diabetes_012")
plt.xlabel("Categorias")
plt.ylabel("Frecuencia")

# Personaliza la leyenda
legend_labels = [f"{key} - {value}" for key, value in
etiquetas.items()]
ax.set_xticklabels(legend_labels)

plt.show()
```



```
print(df_sampled.value_counts("Diabetes_012"))
clustering_df = df_sampled.drop(columns=['Diabetes_012','index'])

clustering_df.to_csv('clustering_df_reducido.csv', index=False)

# Numero de filas y columnas en el df
filas, columnas = clustering_df.shape
print(f"Numero de filas: {filas}")
print(f"Numero de columnas: {columnas}")
```

```
Diabetes_012
0    4000
1    4000
2    4000
dtype: int64
Numero de filas: 12000
Numero de columnas: 21
```

2.- Definición del modelo KMeans (Basado en Centroides)

2.1.- Cálculo del número óptimo de clusters

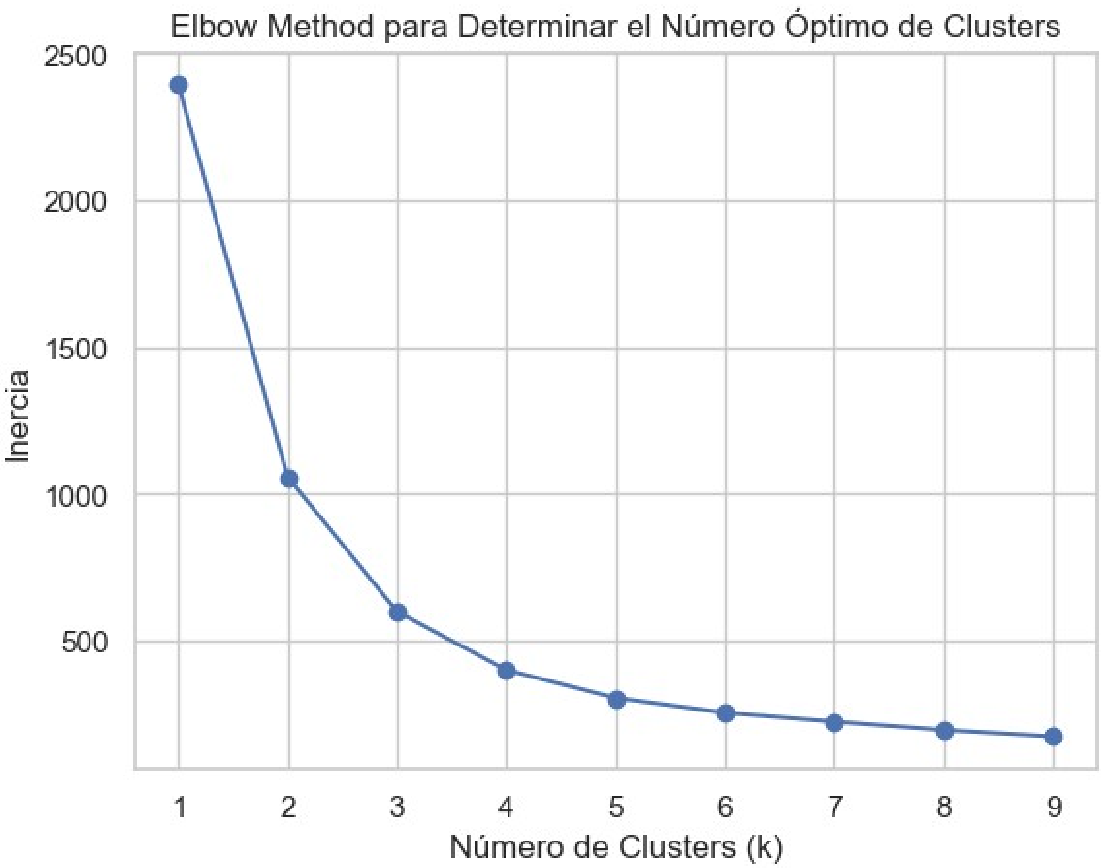
Para KMeans como vemos en el estado del arte, el método más empleado para determinar el número de clusters es el de la inercia.

```
# Lista para almacenar las inercias
inertias = []

# Números de clusters para probar
k_values = range(1, 10)

# Ignoramos los warning
warnings.filterwarnings("ignore", category=FutureWarning)

clustering_df.head()
```

```
   HighBP  HighChol  CholCheck  BMI  Smoker  Stroke  
HeartDiseaseorAttack  \
0       1         1          1   33       0       0   
0   
1       1         1          1   33       1       0   
0   
2       0         0          1   25       1       0   
0   
3       1         1          1   26       0       0   
0   
4       0         0          1   15       0       0   
0   

   PhysActivity  Fruits  Veggies  ...  AnyHealthcare  NoDocbcCost  
GenHlth  \
0             1       1        1  ...              1            0   
3   
1             0       1        1  ...              1            0   
3   
2             1       1        1  ...              1            0   
1   
3             1       0        1  ...              1            0   
2   
4             1       0        1  ...              1            0   
2   

   MentHlth  PhysHlth  DiffWalk  Sex  Age  Education  Income  
0         0         0         0    0    6          6       7  
1        14         0         1    0    9          6       7  
2         0         0         0    1    6          4       7  
3         0         0         0    0   13          3       3  
4         0         1         0    1    5          6       8  
```

```
[5 rows x 21 columns]
```

```python
clustering_df = pd.read_csv('clustering_df_reducido.csv')

columnas_deseadas = ['MentHlth', 'PhysHlth', 'BMI']
clustering_df = clustering_df.loc[:, columnas_deseadas]

for k in k_values:
    Clusterer = Pipeline([
        ('MMS', MinMaxScaler()),
        ('kmeans', KMeans(n_clusters=k, random_state=42))
    ])

    # Entrenar el modelo
    Clusterer.fit(clustering_df)

    # Obtener la inercia y almacenarla en la lista
    inertias.append(Clusterer.named_steps['kmeans'].inertia_)

# Graficar la curva de la inercia en función del número de clusters
para determinar el número óptimo
plt.plot(k_values, inertias, marker='o')
plt.xlabel('Número de Clusters (k)')
plt.ylabel('Inercia')
plt.title('Elbow Method para Determinar el Número Óptimo de Clusters')
plt.show()
```



2.2.- Definición de la pipeline para el KMeans

```
total_clusters = 3 # En base a la grafica anterior cogemos el codo de
la misma
kmeans_pipeline = Pipeline([
    ('MMS', MinMaxScaler()),
    ('kmeans', KMeans(n_clusters=total_clusters))
])
```

2.3.- Entrenamos el modelo con la pipeline y los datos del df

```
kmeans_pipeline.fit(clustering_df)
```

```
Pipeline(steps=[('MMS', MinMaxScaler()), ('kmeans',
KMeans(n_clusters=3))])
```

2.4.- Obtenemos las etiquetas (Cluster al que pertenece) para cada dato

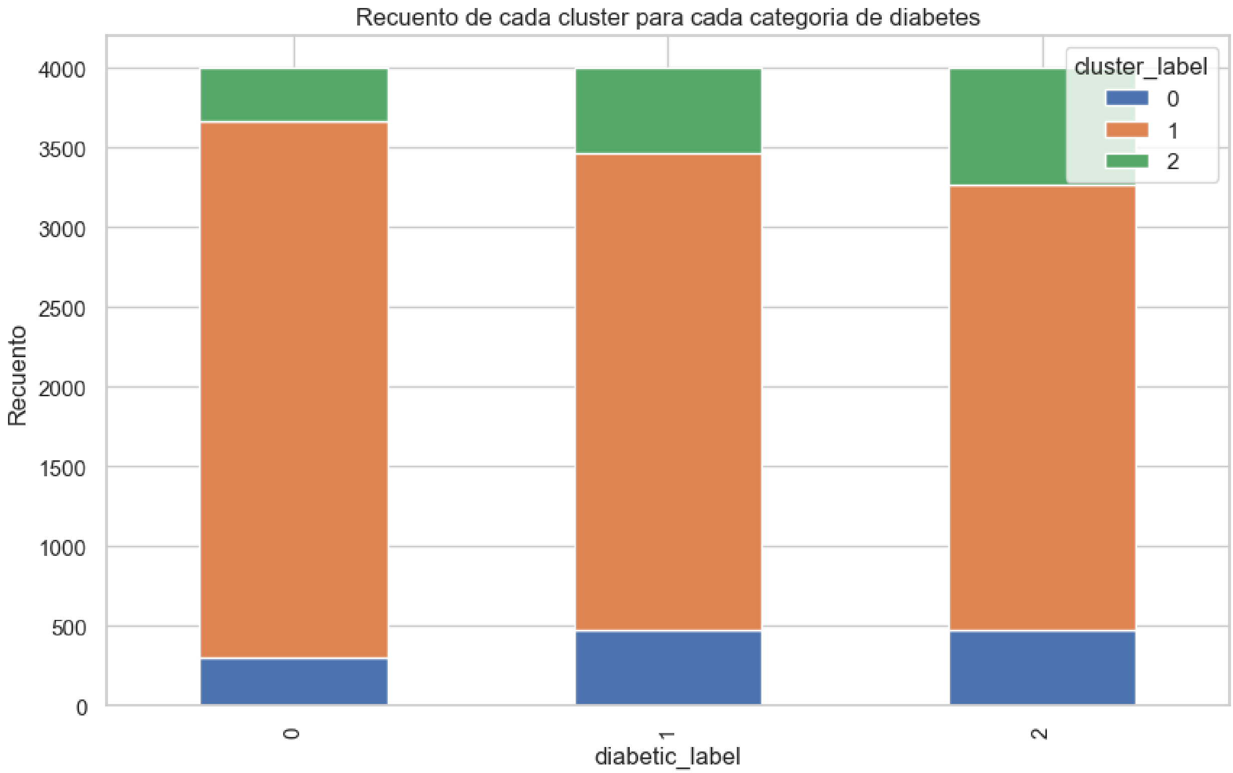
```
labels = kmeans_pipeline.predict(clustering_df)
```

2.5.- Agregar las etiquetas de clusters y de clasificacion al conjunto de datos

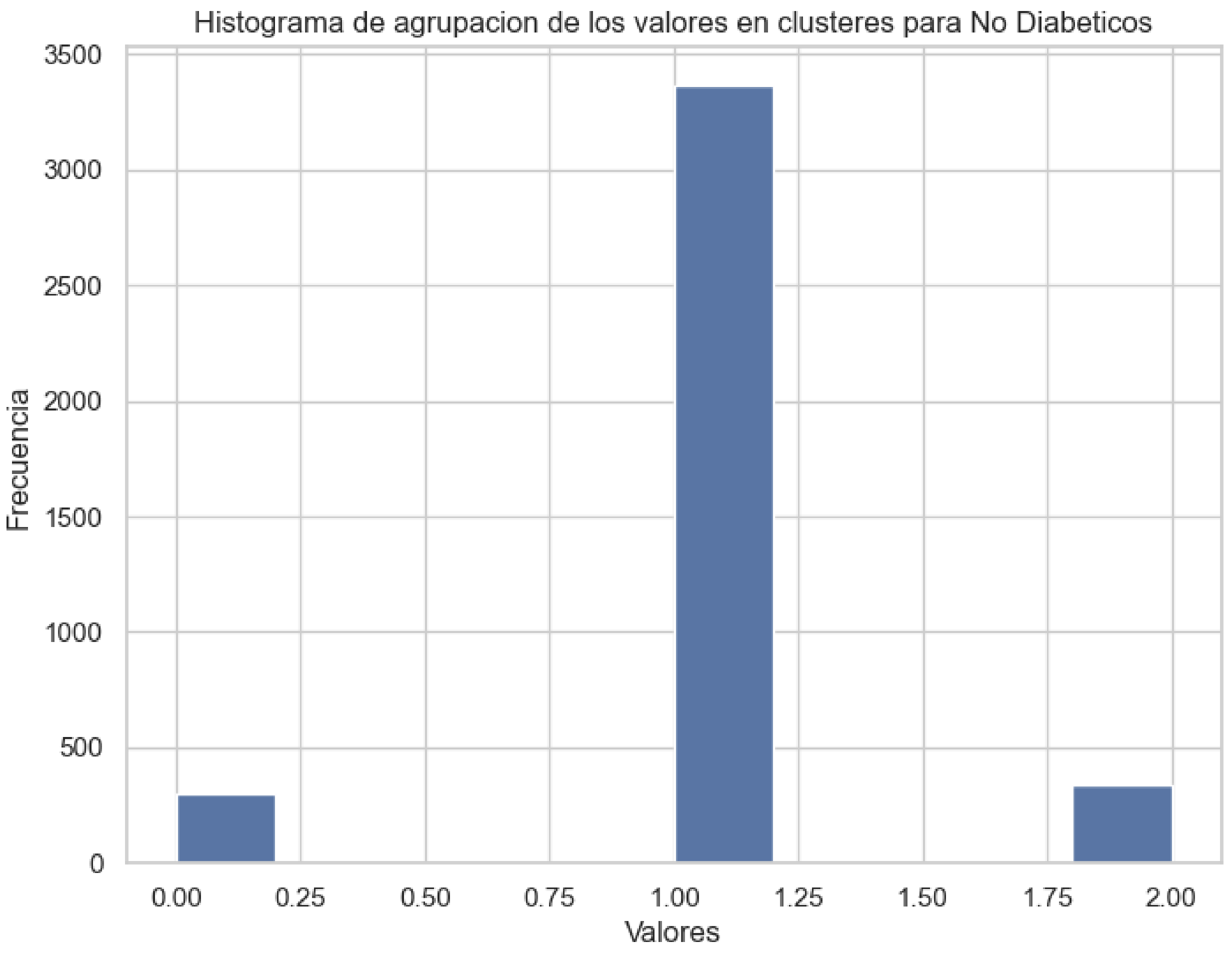
```
clustering_df["cluster_label"] = labels
clustering_df["diabetic_label"] = df_sampled["Diabetes_012"]

clustering_df.head()
clustering_df.to_csv('resultados/df_clusters/kmeans.csv')
```

2.6.- Resumen de pertenencia de clusters por grupo de diabetes

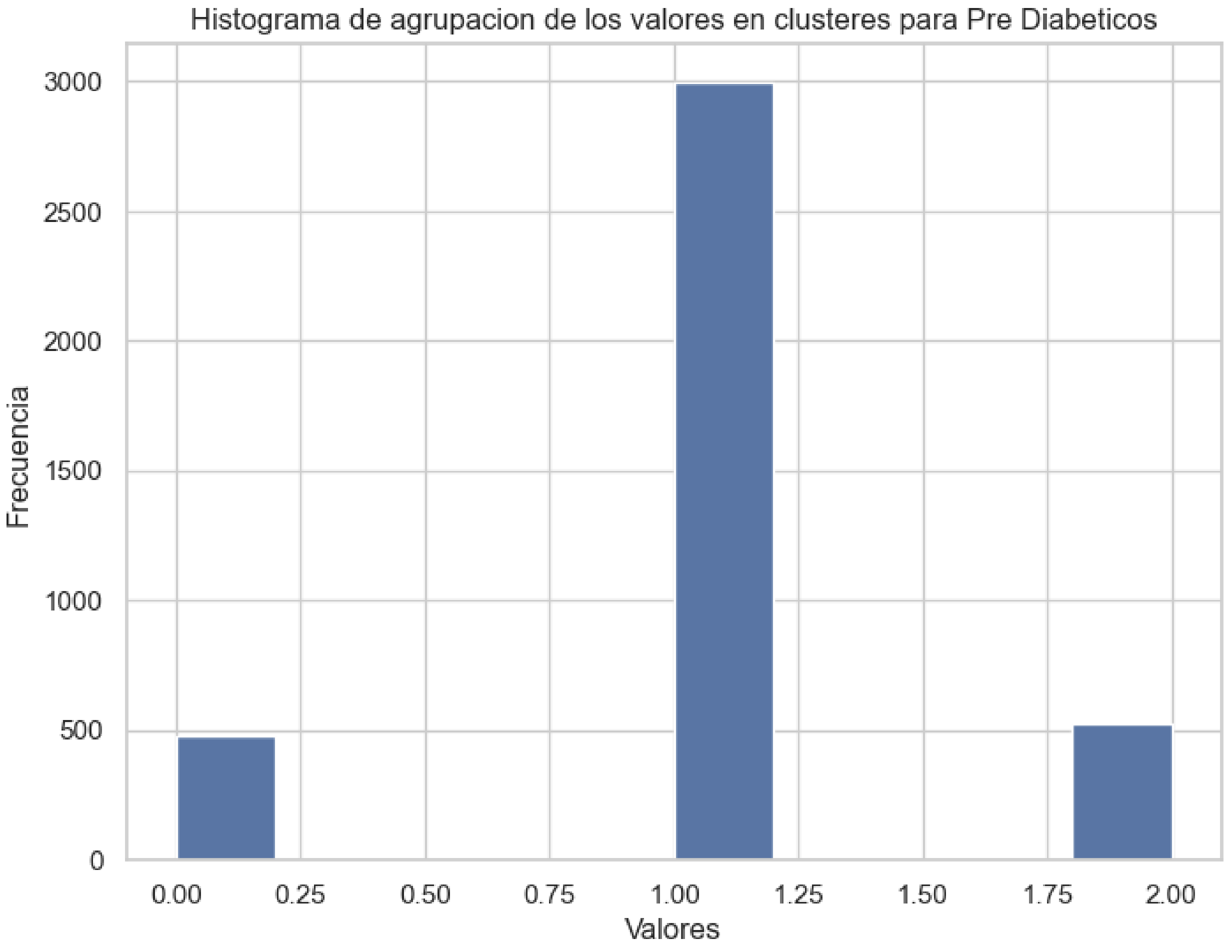
```

# Agrupa por 'diabetic_label' y 'cluster_label' y cuenta las
ocurrencias
df_agrupado = clustering_df.groupby(['diabetic_label',
'cluster_label']).size().unstack()

# Crear el gráfico de barras
df_agrupado.plot(kind='bar', stacked=True, figsize=(10, 6))

# Configurar el título y etiquetas
plt.title('Recuento de cada cluster para cada categoria de diabetes')
plt.xlabel('diabetic_label')
plt.ylabel('Recuento')

# Mostrar el gráfico
plt.show()
```



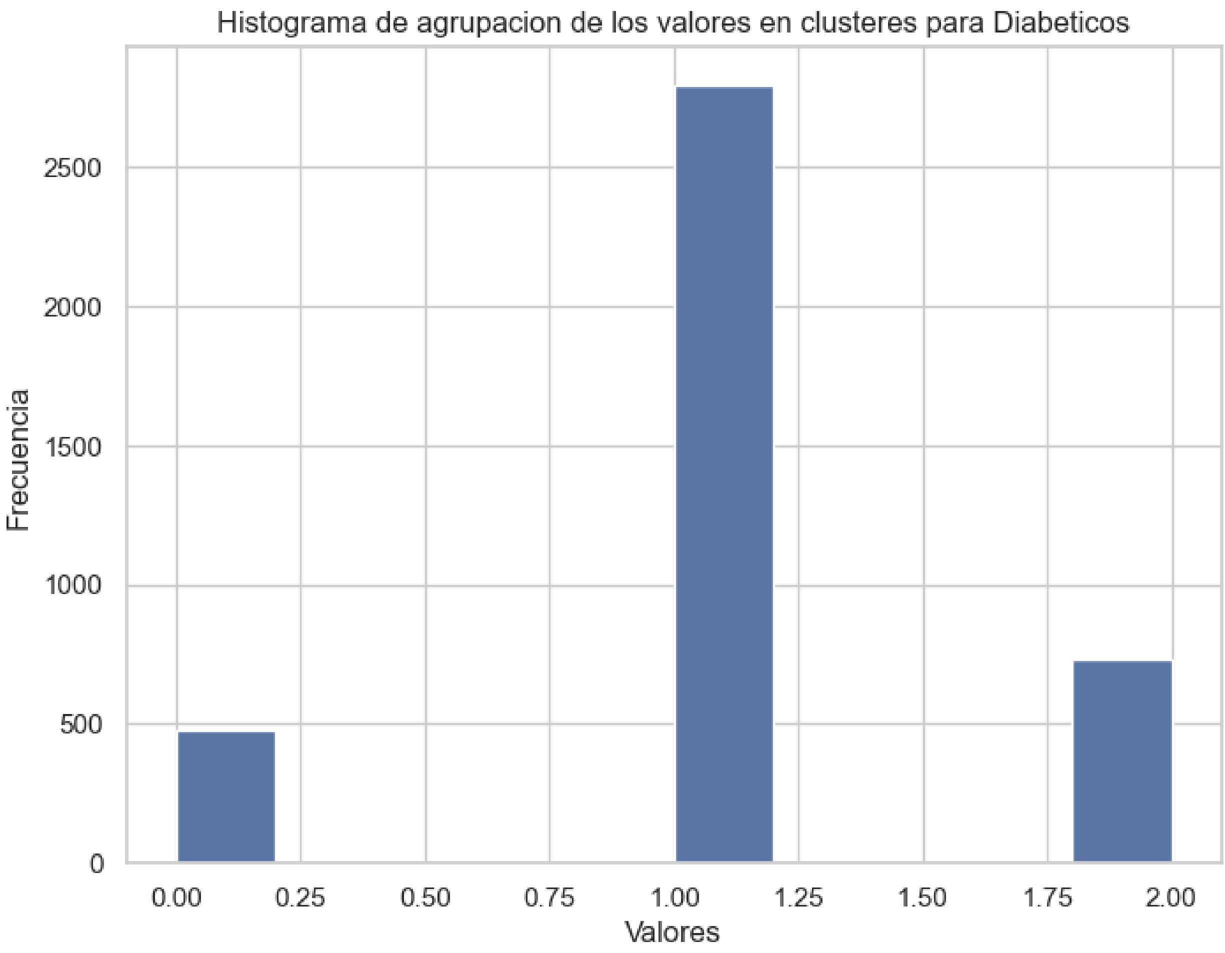
```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 0]
['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```



```python
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para Pre Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```

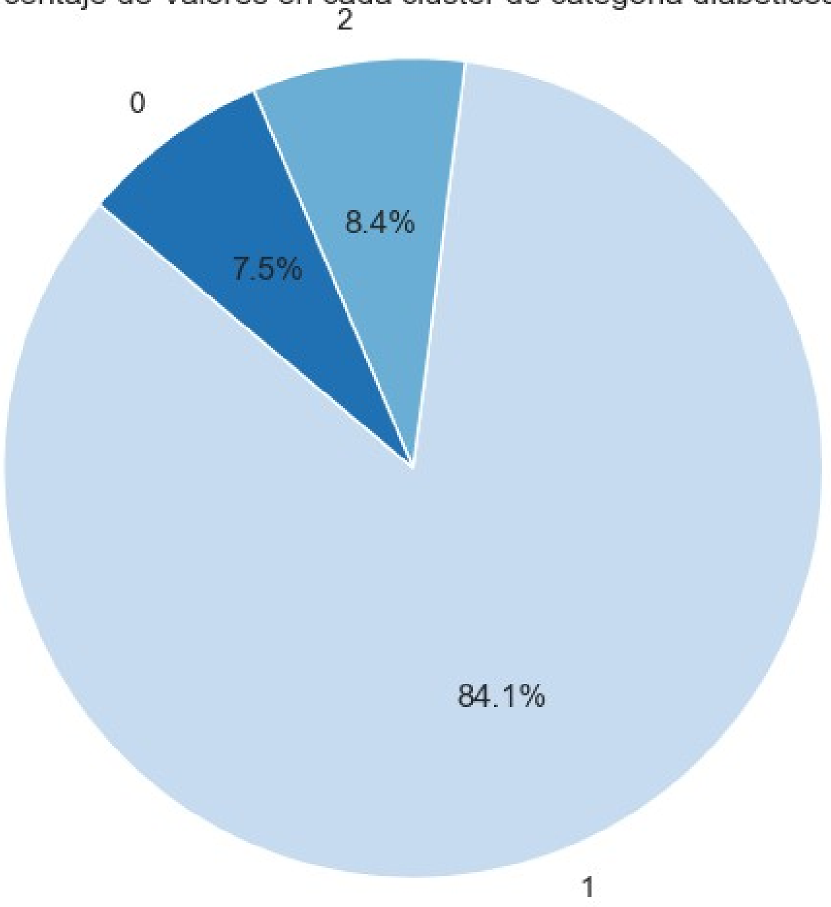


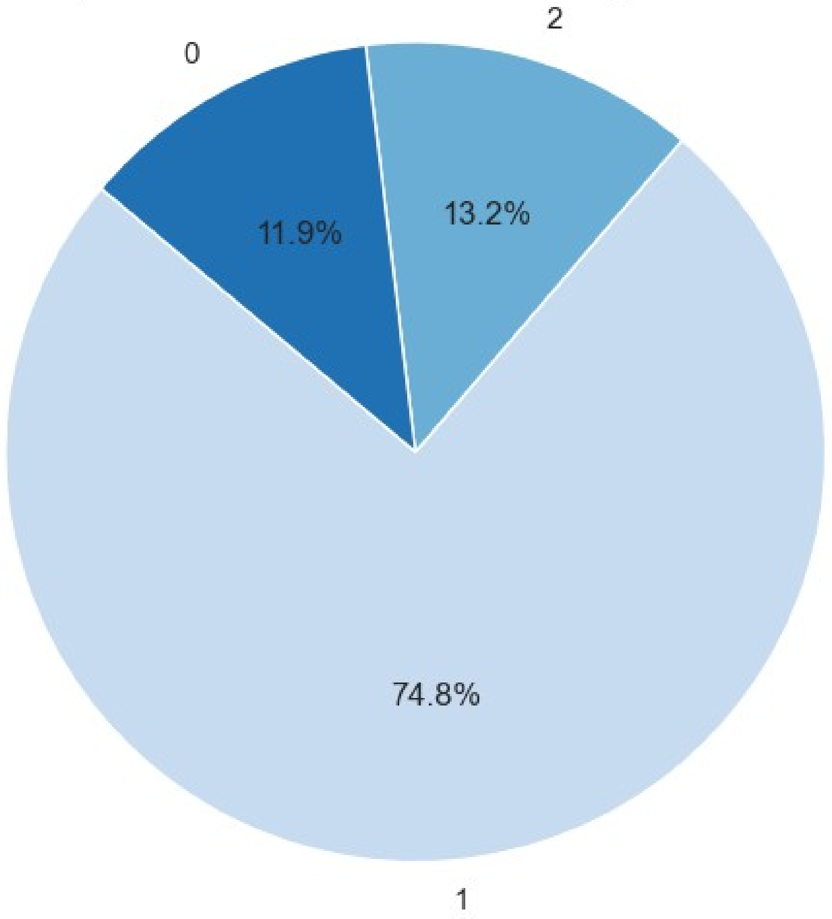
```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 2]
['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para
Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```



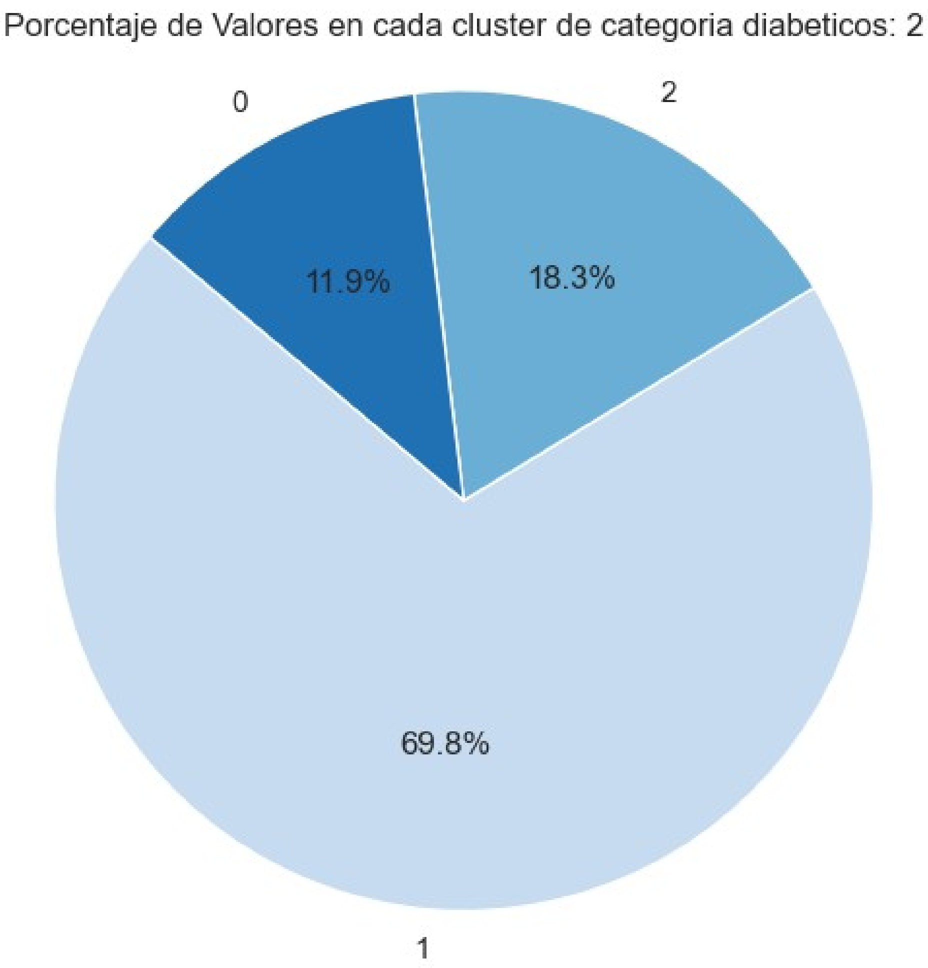
2.7.- Resumen general de pertenencia de las categorias de diabetes a los clusters

```
for diabetic_value in clustering_df['diabetic_label'].unique():

    # Pintamos el pie plot para esta categoria de diabetes
    df_class =  clustering_df[clustering_df['diabetic_label'] ==
diabetic_value]
    plt.figure(figsize=(6, 6))
    labels = df_class['cluster_label'].value_counts().index
    sizes = df_class['cluster_label'].value_counts().values
    plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140,
colors=sns.color_palette("Blues", n_colors=total_clusters))
    plt.title(f"Porcentaje de Valores en cada cluster de categoria
diabeticos: {diabetic_value}")
    plt.axis('equal')
    plt.show()
```


Porcentaje de Valores en cada cluster de categoria diabeticos: 0



Porcentaje de Valores en cada cluster de categoria diabeticos: 1


Porcentaje de Valores en cada cluster de categoria diabeticos: 2

2.8.- Análisis estadístico de los clusters

```
for cluster_number in list(range(total_clusters)):
    # Guardamos las descripción a un csv
    description_df = clustering_df[clustering_df['cluster_label'] ==
cluster_number].drop(columns=['diabetic_label',
'cluster_label']).describe()
    description_df.to_csv(f'resultados\\kmeans\\
kmeans_estadisticas_cluster{cluster_number}.csv', sep=',',
float_format='%.2f', index=False)

    # Guardamos la moda a un csv
    mode_df = clustering_df[clustering_df['cluster_label'] ==
cluster_number].drop(columns=['diabetic_label',
'cluster_label']).mode()
    mode_df.to_csv(f'resultados\\kmeans\\
kmeans_moda_cluster{cluster_number}.csv', sep=',',
float_format='%.2f', index=False)
```

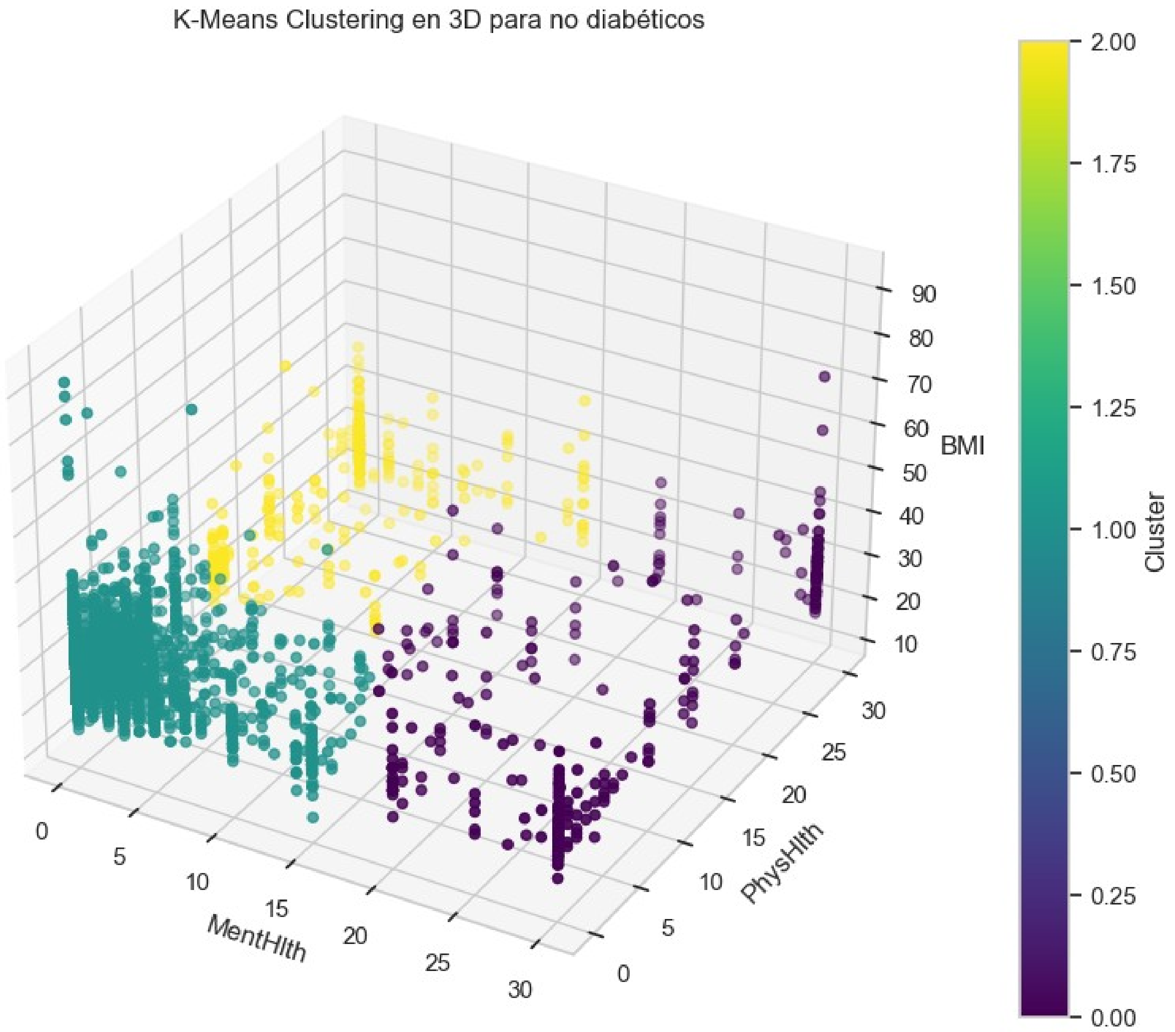
2.9.- Representamos los clusters para cada categoría de enfermedad y las tres variables deseadas

Escogemos tres pero podrían ser cualquiera del dataset

```
# Variables a comparar
var1 = "MentHlth"
var2 = "PhysHlth"
var3 = "BMI"
```

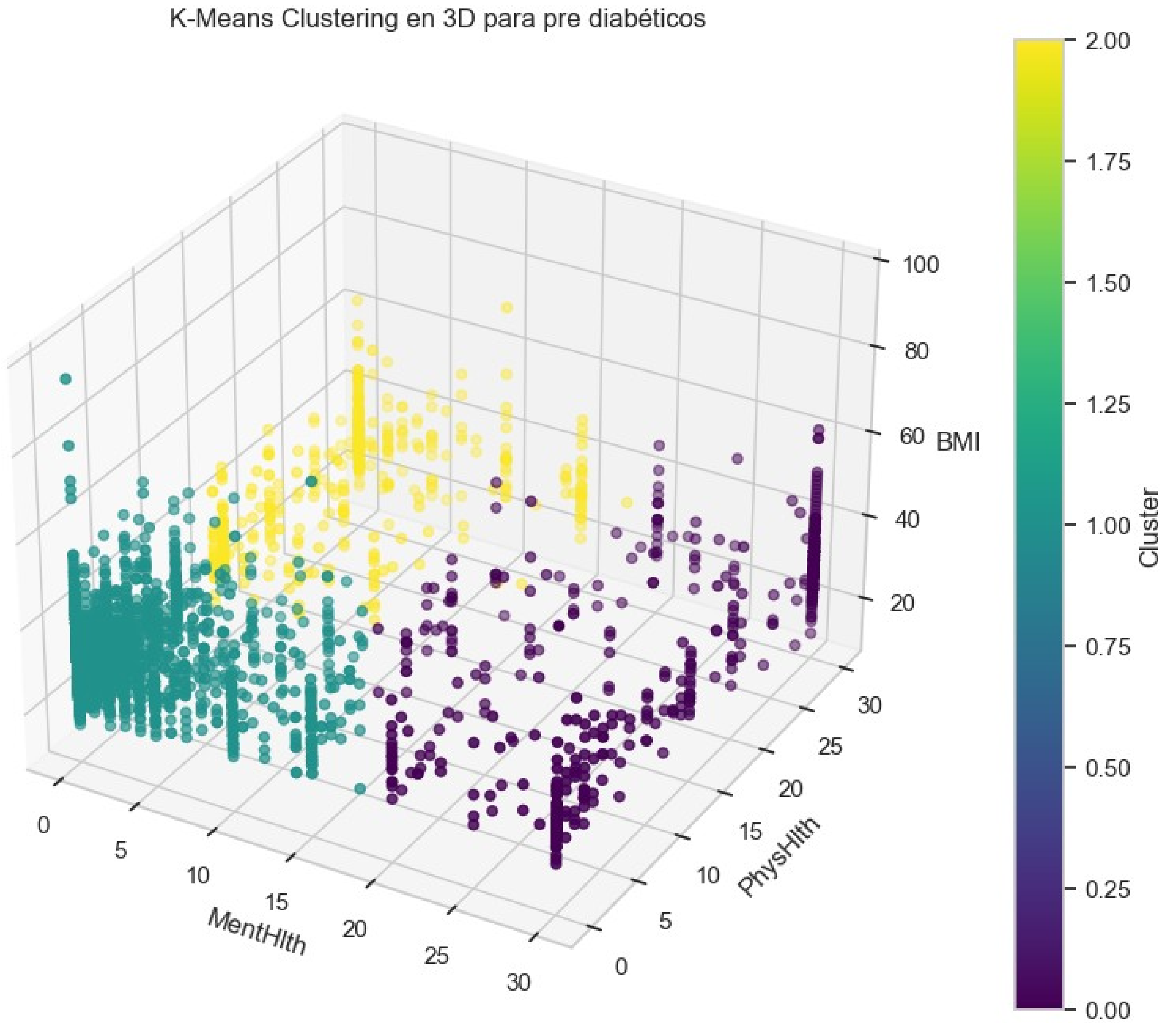
2.10.1.- No diabetico 3D Plot

```
# Filtramos los no diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 0]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Means Clustering en 3D para no diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



2.10.2.- Pre diabetico 3D Plot

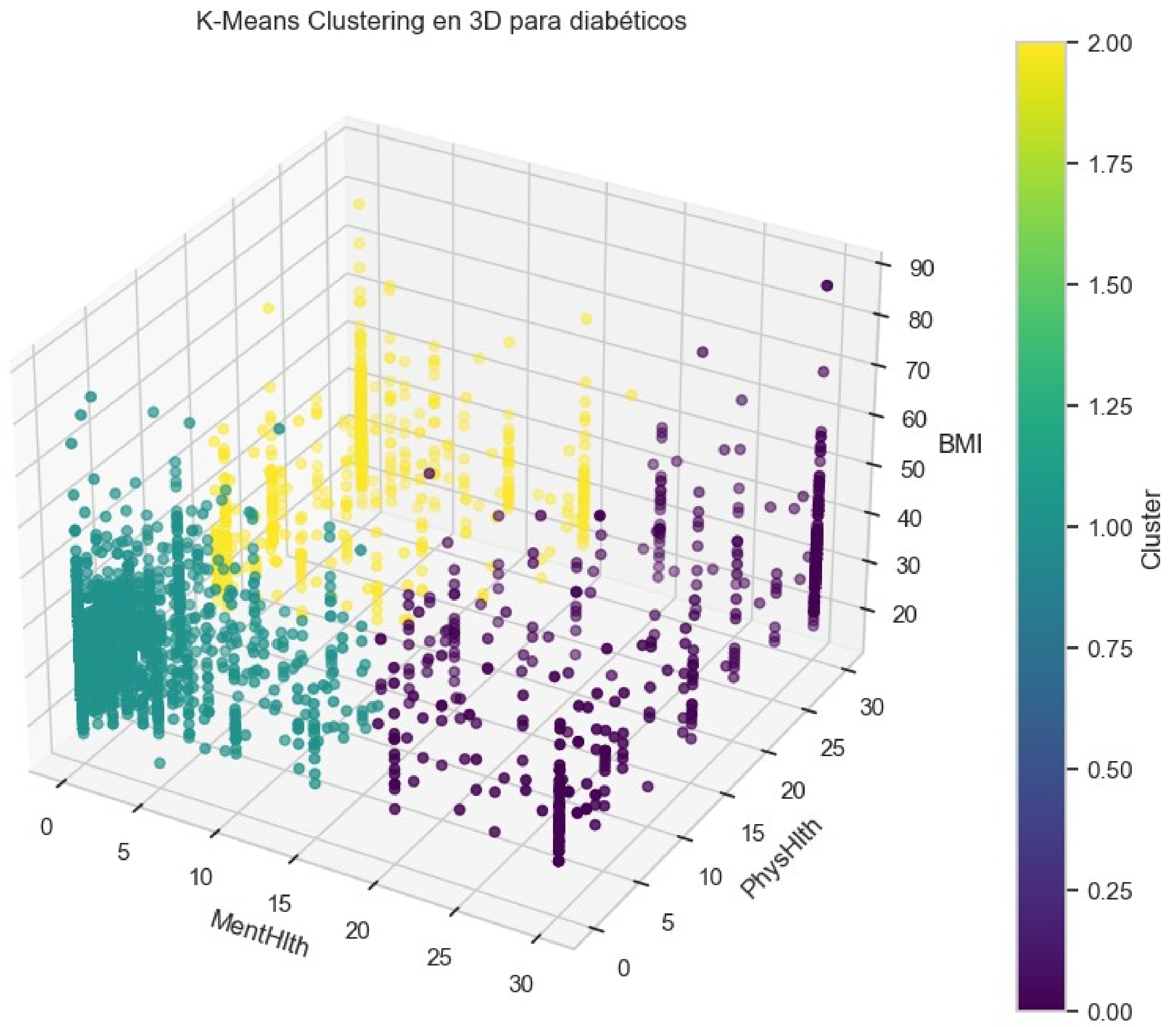
```
# Filtramos los pre diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 1]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Means Clustering en 3D para pre diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



2.10.3.- Diabetico 3D Plot

```
# Filtramos los diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 2]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')
```

```
# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Means Clustering en 3D para diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



2.10.- Evaluación de calidad

```
evaluacion_calidad = {}

clustering_df = pd.read_csv('clustering_df_reducido.csv')

# Indice de silueta
silhouette_avg = silhouette_score(clustering_df,
kmeans_pipeline.named_steps['kmeans'].labels_)
print("El índice de silueta promedio es:", silhouette_avg)

# Indice de Davies-Bouldin
davies_bouldin = davies_bouldin_score(clustering_df,
kmeans_pipeline.named_steps['kmeans'].labels_)
print("El índice de Davies-Bouldin es:", davies_bouldin)

evaluacion_calidad["kmeans"] = {"silueta": silhouette_avg,
"davies_bouldin": davies_bouldin}
```

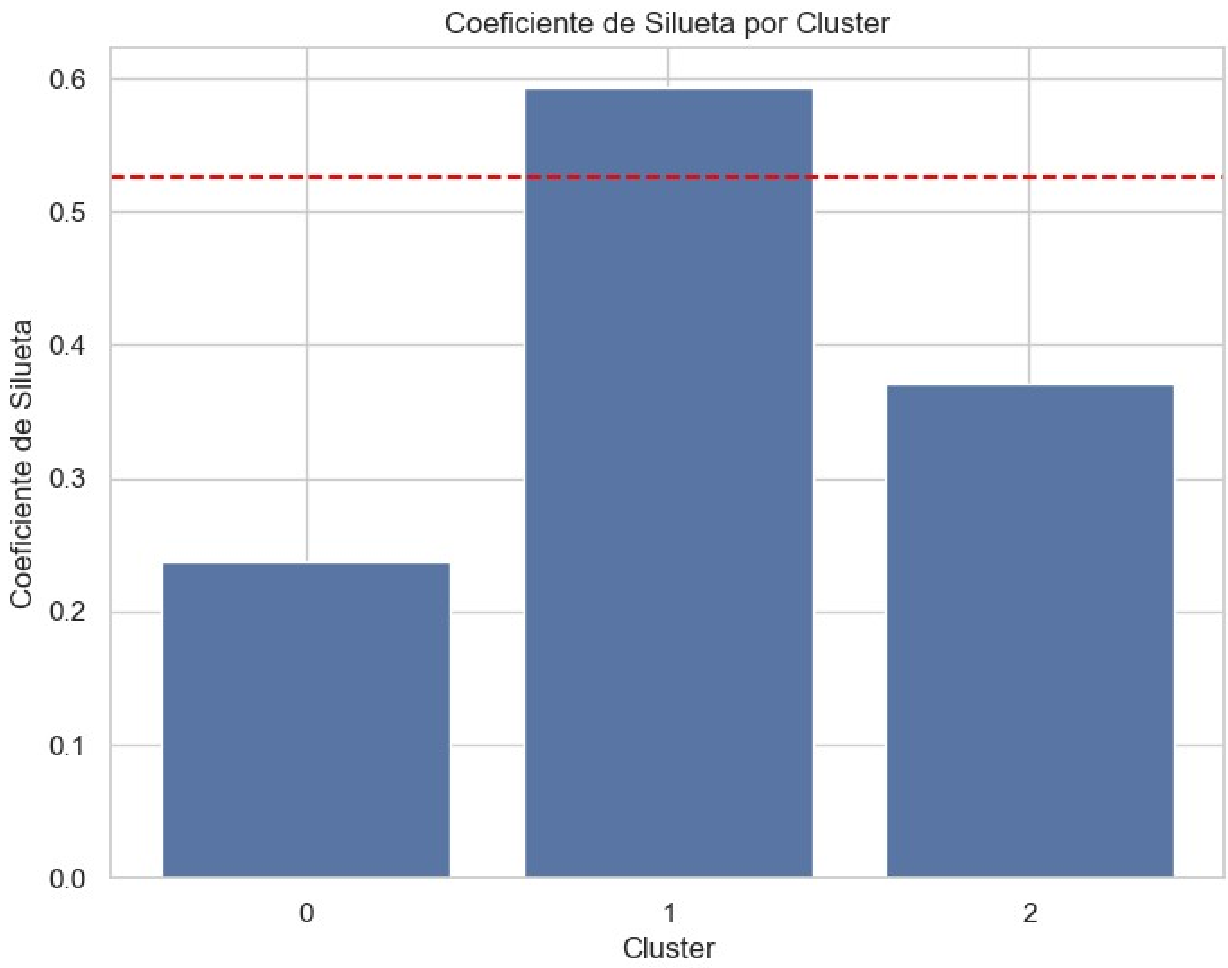
```
El índice de silueta promedio es: 0.5261373140682846
El índice de Davies-Bouldin es: 0.9704896757029348
```

El índice de silueta es una medida de qué tan bien están separados los clusters en un conjunto de datos. Toma valores en el rango [-1, 1]. Cuanto más cercano a 1 sea el valor, mejor será la separación, indicando que los puntos están bien ubicados en sus clusters y lejos de los otros clusters. Un valor cercano a 0 indica superposición en la ubicación de los clusters, mientras que valores negativos generalmente indican que los puntos podrían estar asignados al cluster incorrecto.

El índice de Davies-Bouldin es otra métrica de validación interna para evaluar la calidad de los clusters. Mide la "compacidad" dentro de los clusters y la separación entre diferentes clusters. Un valor más bajo indica una mejor partición.

```
cluster_siluetes = {}

# Calculamos el coeficiente de Silueta promedio
silhouette_avg = silhouette_score(clustering_df,
kmeans_pipeline.named_steps['kmeans'].labels_)

# Calculamos el coeficiente de Silueta para cada muestra
sample_silhouette_values = silhouette_samples(clustering_df,
kmeans_pipeline.named_steps['kmeans'].labels_)

# Creamos una lista para almacenar los coeficientes de Silueta
promedio por cluster
cluster_silhouette_avg = []

list_siluetes = []
# Creamos un bucle para obtener el coeficiente de Silueta para cada
cluster
for cluster in
```

```
range(max(kmeans_pipeline.named_steps['kmeans'].labels_)+1):
    # Obtenemos las muestras pertenecientes al cluster actual
    cluster_silhouette_values =
sample_silhouette_values[kmeans_pipeline.named_steps['kmeans'].labels_
== cluster]

    # Calculamos el coeficiente de Silueta promedio para el cluster
actual
    avg_silhouette = np.mean(cluster_silhouette_values)

    cluster_dict = {'category': cluster, 'value': avg_silhouette}
    list_siluetes.append(cluster_dict)

    # Almacenamos el coeficiente de Silueta promedio en la lista
    cluster_silhouette_avg.append(avg_silhouette)

cluster_siluetes["kmeans"] = list_siluetes

# Creamos el gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(range(len(cluster_silhouette_avg)), cluster_silhouette_avg,
color="#5975A4")
plt.title('Coeficiente de Silueta por Cluster')
plt.xlabel('Cluster')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range(len(cluster_silhouette_avg)))
plt.axhline(y=silhouette_avg, color="red", linestyle="--")  # Línea
para indicar el valor promedio global de Silueta
plt.show()
```



2.11.- Save models

```
joblib.dump(kmeans_pipeline,
'resultados/trained_models/kmeans.joblib')

['resultados/trained_models/kmeans.joblib']

clustering_df

       HighBP  HighChol  CholCheck  BMI  Smoker  Stroke
HeartDiseaseorAttack  \
0           1         1          1   33       0       0
0
1           1         1          1   33       1       0
0
2           0         0          1   25       1       0
0
3           1         1          1   26       0       0
0
4           0         0          1   15       0       0
0
```

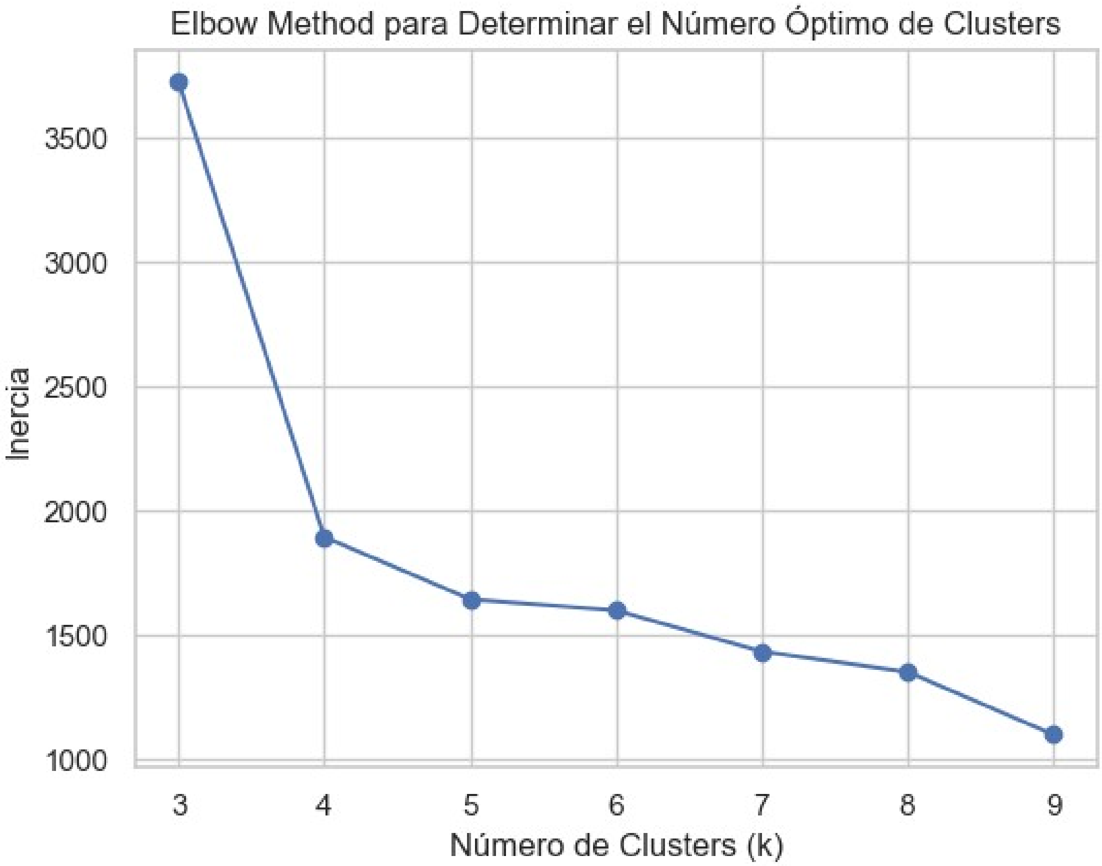
```
...       ...       ...        ...  ...     ...     ...
...
11995       1         1          1   29       1       0
0
11996       1         1          1   17       1       0
1
11997       0         0          1   22       1       0
0
11998       1         1          1   27       0       0
1
11999       1         1          1   25       0       0
1

       PhysActivity  Fruits  Veggies  ...  AnyHealthcare  NoDocbcCost
\
0                 1       1        1  ...              1            0

1                 0       1        1  ...              1            0

2                 1       1        1  ...              1            0

3                 1       0        1  ...              1            0

4                 1       0        1  ...              1            0

...             ...     ...      ...  ...            ...          ...

11995             0       1        0  ...              1            0

11996             0       0        0  ...              1            1

11997             0       1        1  ...              1            0

11998             1       1        1  ...              0            0

11999             1       1        1  ...              1            0


       GenHlth  MentHlth  PhysHlth  DiffWalk  Sex  Age  Education
Income
0            3         0         0         0    0    6          6
7
1            3        14         0         1    0    9          6
7
2            1         0         0         0    1    6          4
7
3            2         0         0         0    0   13          3
3
4            2         0         1         0    1    5          6
8
```

```
...        ...       ...       ...       ...  ...  ...        ...
...
11995        4         0         0         1    1   12          2
2
11996        5         2         4         1    1    8          3
1
11997        4         0        30         1    1   13          4
5
11998        4         0         0         1    0   13          4
1
11999        2         0         0         0    1   10          6
6

[12000 rows x 21 columns]
```

3.- KMedoids (Basado en Centroides)

3.1.- Cálculo del número óptimo de clusters

```
# Lista para almacenar las inercias
inertias = []

# Números de clusters para probar
k_values = range(3, 10)

# Ignoramos los warning
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=UserWarning)

clustering_df = pd.read_csv('clustering_df_reducido.csv')

columnas_deseadas = ['MentHlth', 'PhysHlth', 'BMI']
clustering_df = clustering_df.loc[:, columnas_deseadas]

for k in k_values:
    Clusterer = Pipeline([
        ('MMS', MinMaxScaler()),
        ('kmedoids', KMedoids(n_clusters=k, random_state=42))
    ])

    # Entrenar el modelo
    Clusterer.fit(clustering_df)

    # Obtener la inercia y almacenarla en la lista
    inertias.append(Clusterer.named_steps['kmedoids'].inertia_)

# Graficar la curva de la inercia en función del número de clusters
para determinar el número óptimo
plt.plot(k_values, inertias, marker='o')
plt.xlabel('Número de Clusters (k)')
plt.ylabel('Inercia')
plt.title('Elbow Method para Determinar el Número Óptimo de Clusters')
plt.show()
```



3.2.- Definición de la pipeline para el KMedoids

```
total_clusters = 4 # En base a la grafica anterior cogemos el codo de
la misma
kmedoids_pipeline = Pipeline([
    ('MMS', MinMaxScaler()),
    ('kmedoids', KMedoids(n_clusters=total_clusters))
])
```

3.3.- Entrenamos el modelo con la pipeline y los datos del df

```
kmedoids_pipeline.fit(clustering_df)

Pipeline(steps=[('MMS', MinMaxScaler()), ('kmedoids',
KMedoids(n_clusters=4))])
```

3.4.- Obtenemos las etiquetas (Cluster al que pertenece) para cada dato

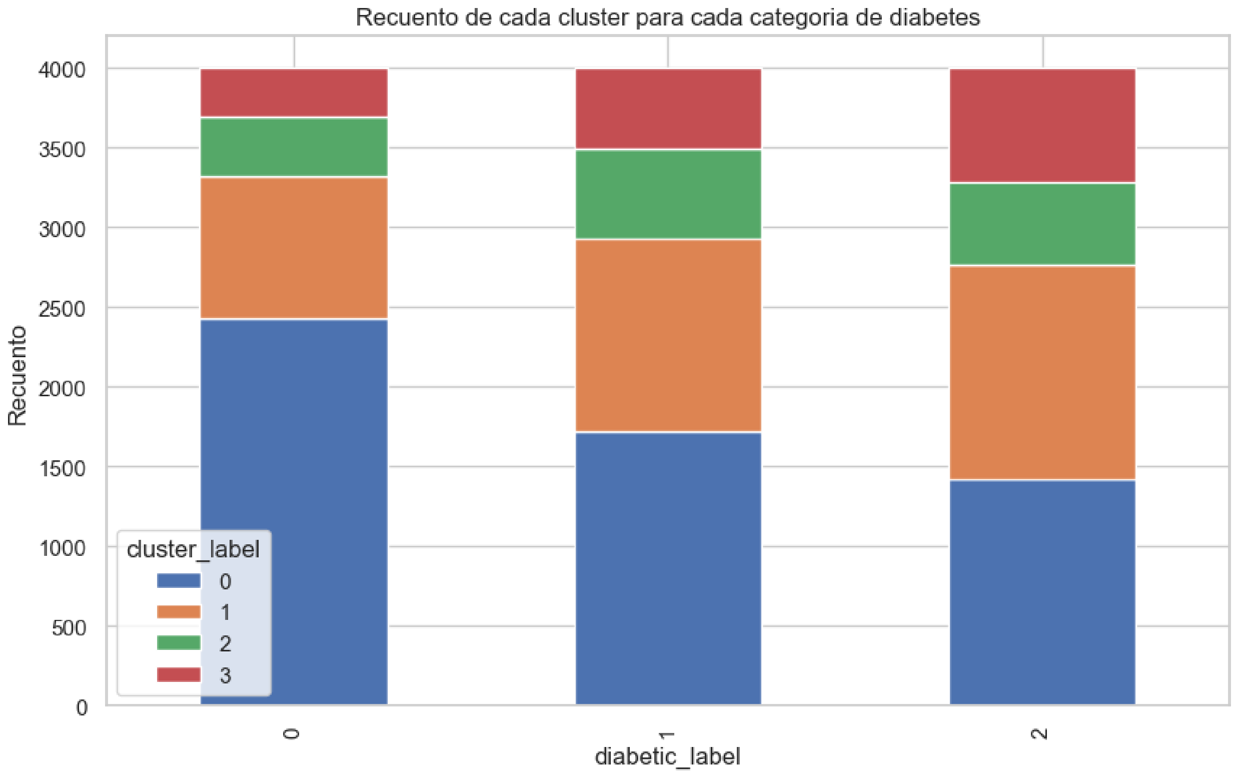
```
labels = kmedoids_pipeline.predict(clustering_df)
```

3.5.- Agregar las etiquetas de clusters y de clasificacion al conjunto de datos

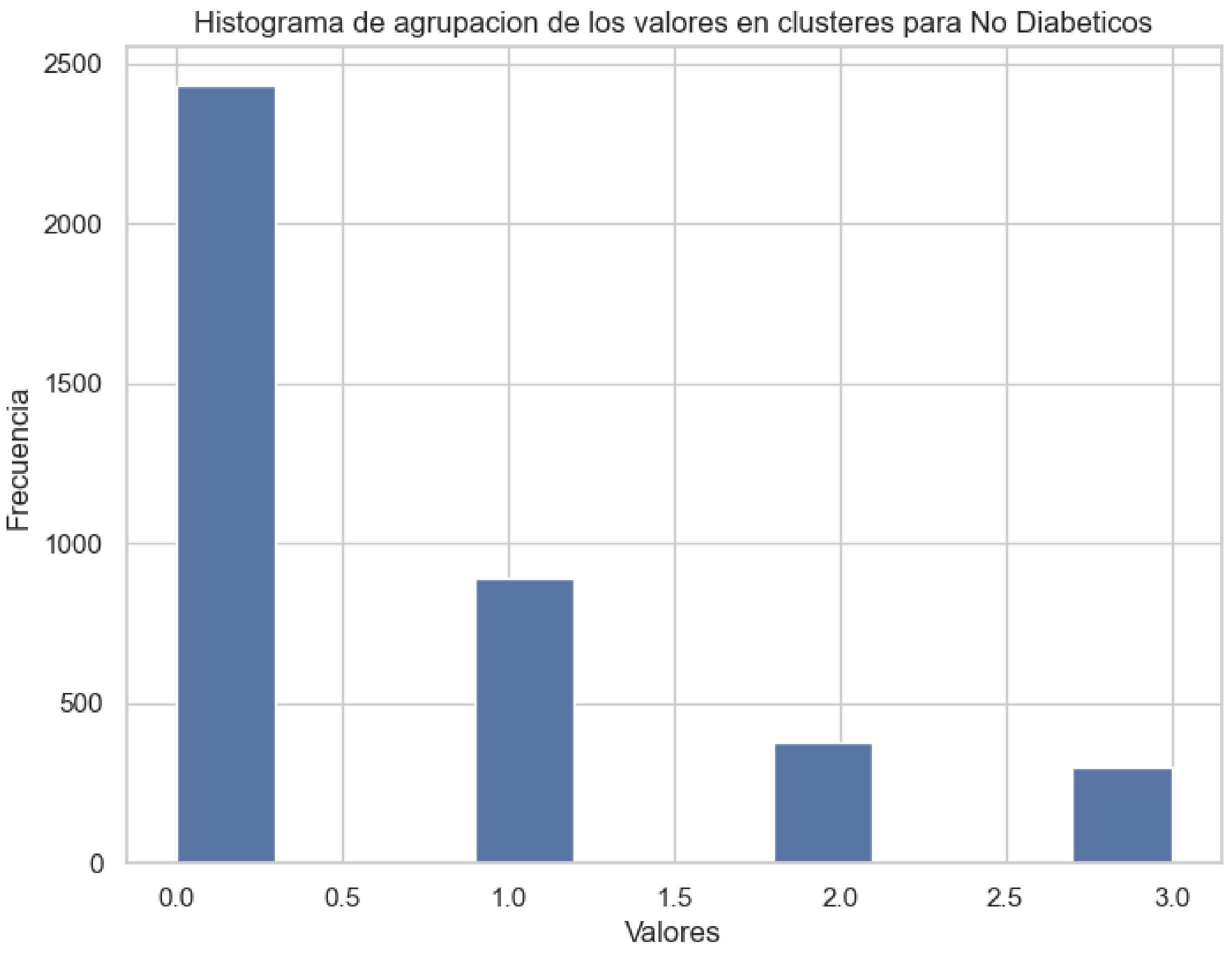
```
clustering_df["cluster_label"] = labels
clustering_df["diabetic_label"] = df_sampled["Diabetes_012"]

clustering_df.head()
clustering_df.to_csv('resultados/df_clusters/kmedoids.csv')
```

3.6.- Resumen de pertenencia de clusters por grupo de diabetes

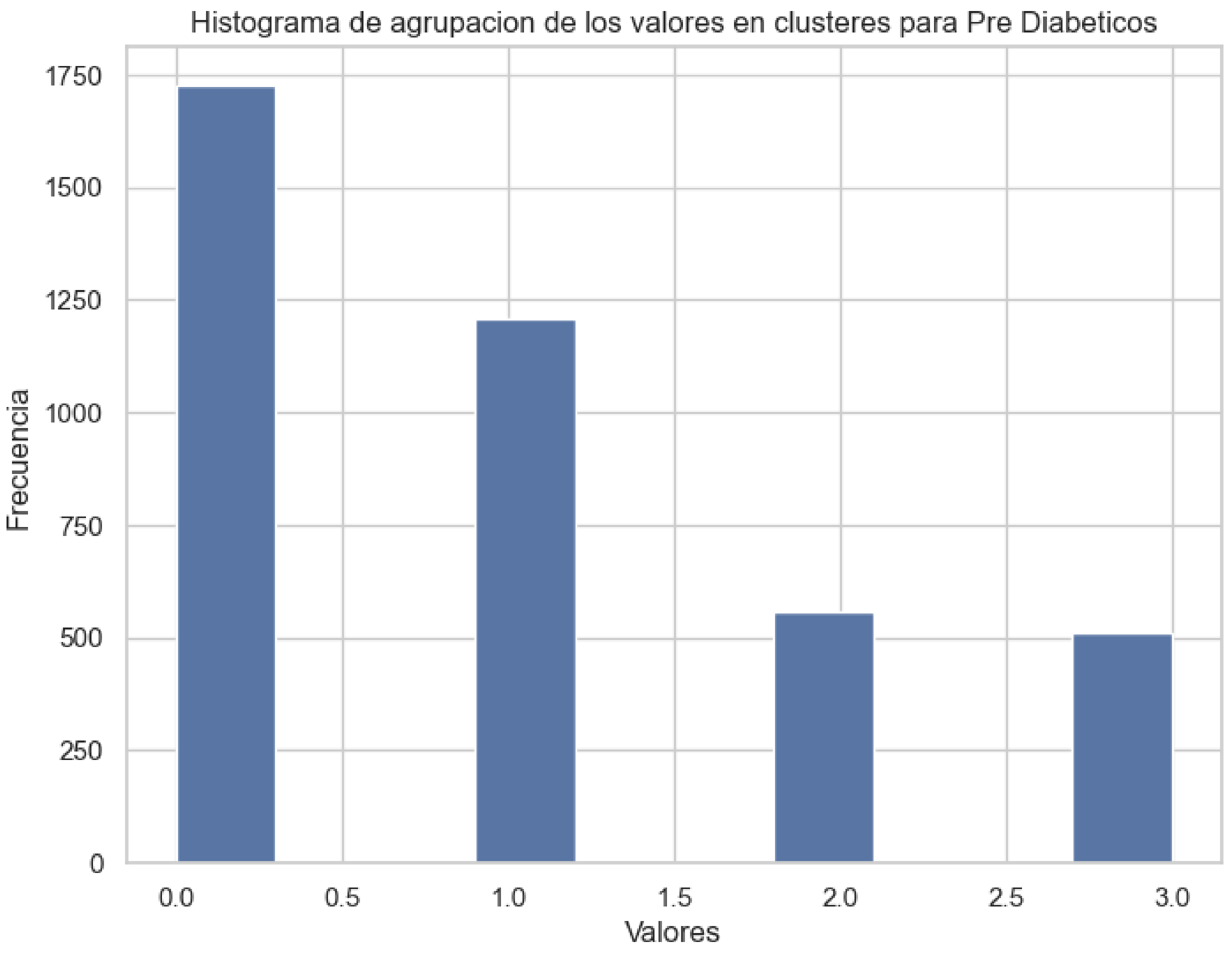
```

# Agrupa por 'diabetic_label' y 'cluster_label' y cuenta las
ocurrencias
df_agrupado = clustering_df.groupby(['diabetic_label',
'cluster_label']).size().unstack()

# Crear el gráfico de barras
df_agrupado.plot(kind='bar', stacked=True, figsize=(10, 6))

# Configurar el título y etiquetas
plt.title('Recuento de cada cluster para cada categoria de diabetes')
plt.xlabel('diabetic_label')
plt.ylabel('Recuento')

# Mostrar el gráfico
plt.show()
```



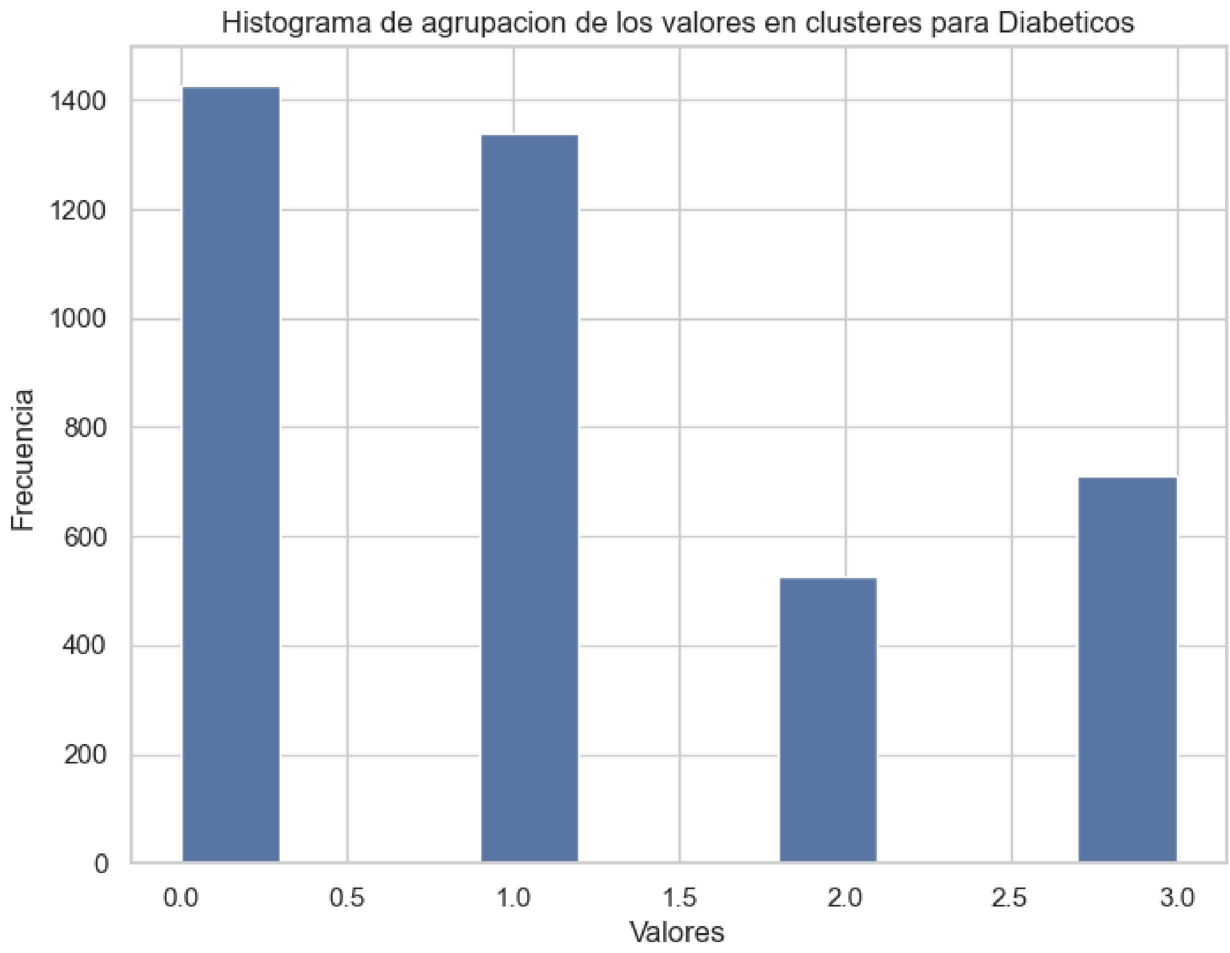
```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 0]
['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```



```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]
['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para
Pre Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```

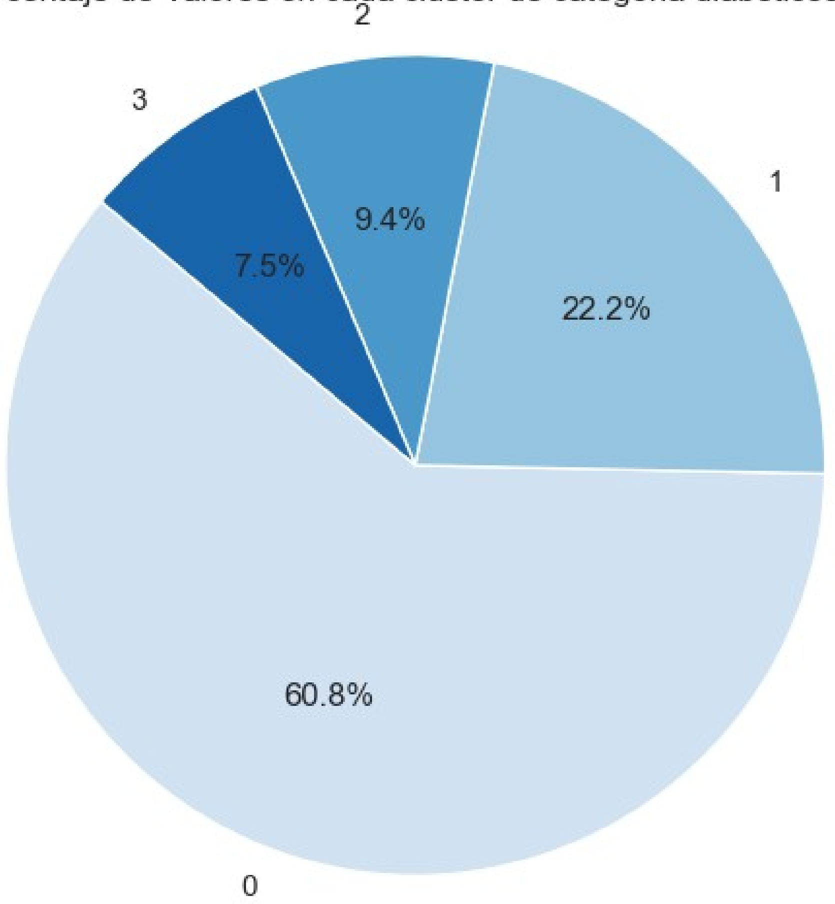


```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 2]
['cluster_label']
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4")
plt.title("Histograma de agrupacion de los valores en clusteres para
Diabeticos")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.show()
```

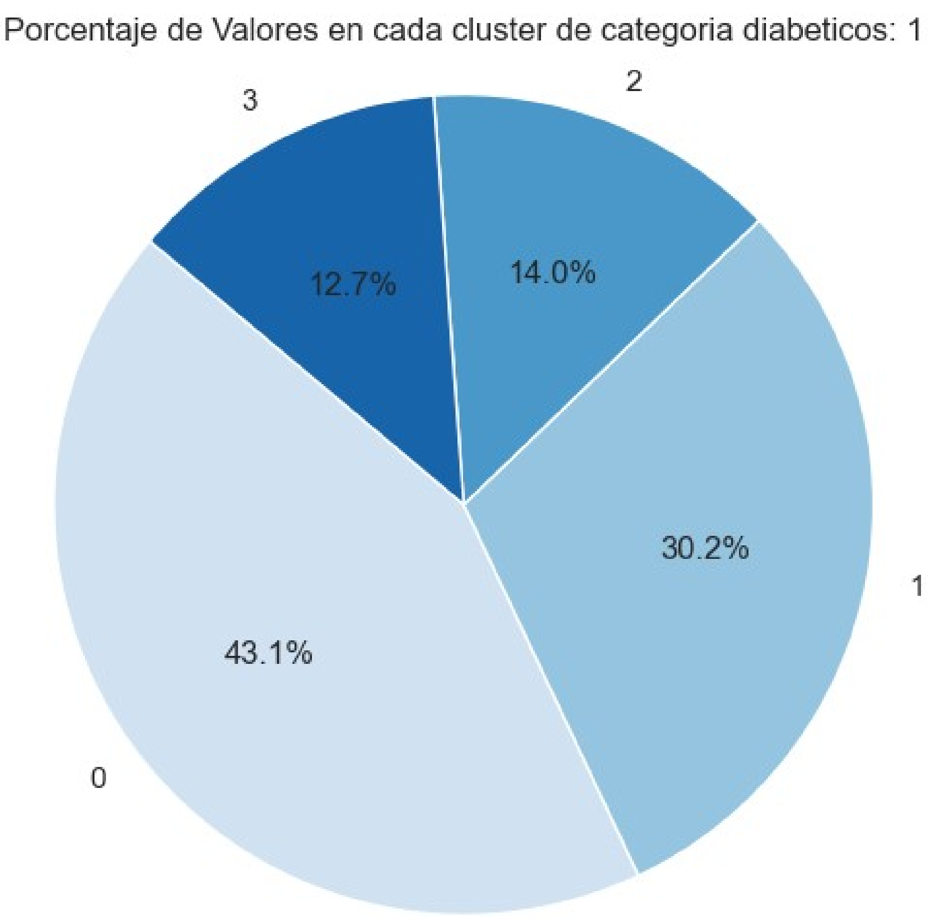


3.7.- Resumen general de pertenencia de las categorias de diabetes a los clusters

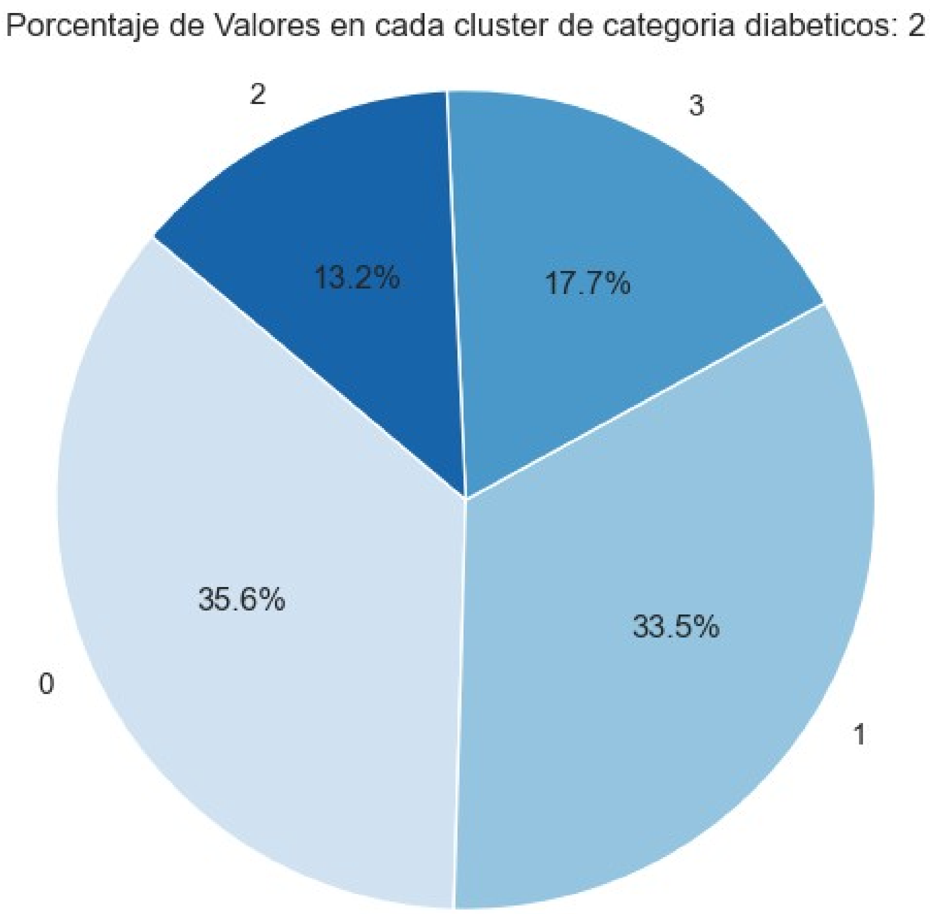
```
for diabetic_value in clustering_df['diabetic_label'].unique():

    # Pintamos el pie plot para esta categoria de diabetes
    df_class =  clustering_df[clustering_df['diabetic_label'] ==
diabetic_value]
    plt.figure(figsize=(6, 6))
    labels = df_class['cluster_label'].value_counts().index
    sizes = df_class['cluster_label'].value_counts().values
    plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140,
colors=sns.color_palette("Blues", n_colors=total_clusters))
    plt.title(f"Porcentaje de Valores en cada cluster de categoria
diabeticos: {diabetic_value}")
    plt.axis('equal')
    plt.show()
```



Porcentaje de Valores en cada cluster de categoria diabeticos: 0



Porcentaje de Valores en cada cluster de categoria diabeticos: 1



3.8.- Análisis estadístico de los clusters

```
for cluster_number in list(range(total_clusters)):
    # Guardamos las descripción a un csv
    description_df = clustering_df[clustering_df['cluster_label'] ==
cluster_number].drop(columns=['diabetic_label',
'cluster_label']).describe()
    description_df.to_csv(f'resultados\\kmedoids\\
kmedoids_estadisticas_cluster{cluster_number}.csv', sep=',',
float_format='%.2f', index=False)

    # Guardamos la moda a un csv
    mode_df = clustering_df[clustering_df['cluster_label'] ==
cluster_number].drop(columns=['diabetic_label',
'cluster_label']).mode()
    mode_df.to_csv(f'resultados\\kmedoids\\
kmedoids_moda_cluster{cluster_number}.csv', sep=',',
float_format='%.2f', index=False)
```

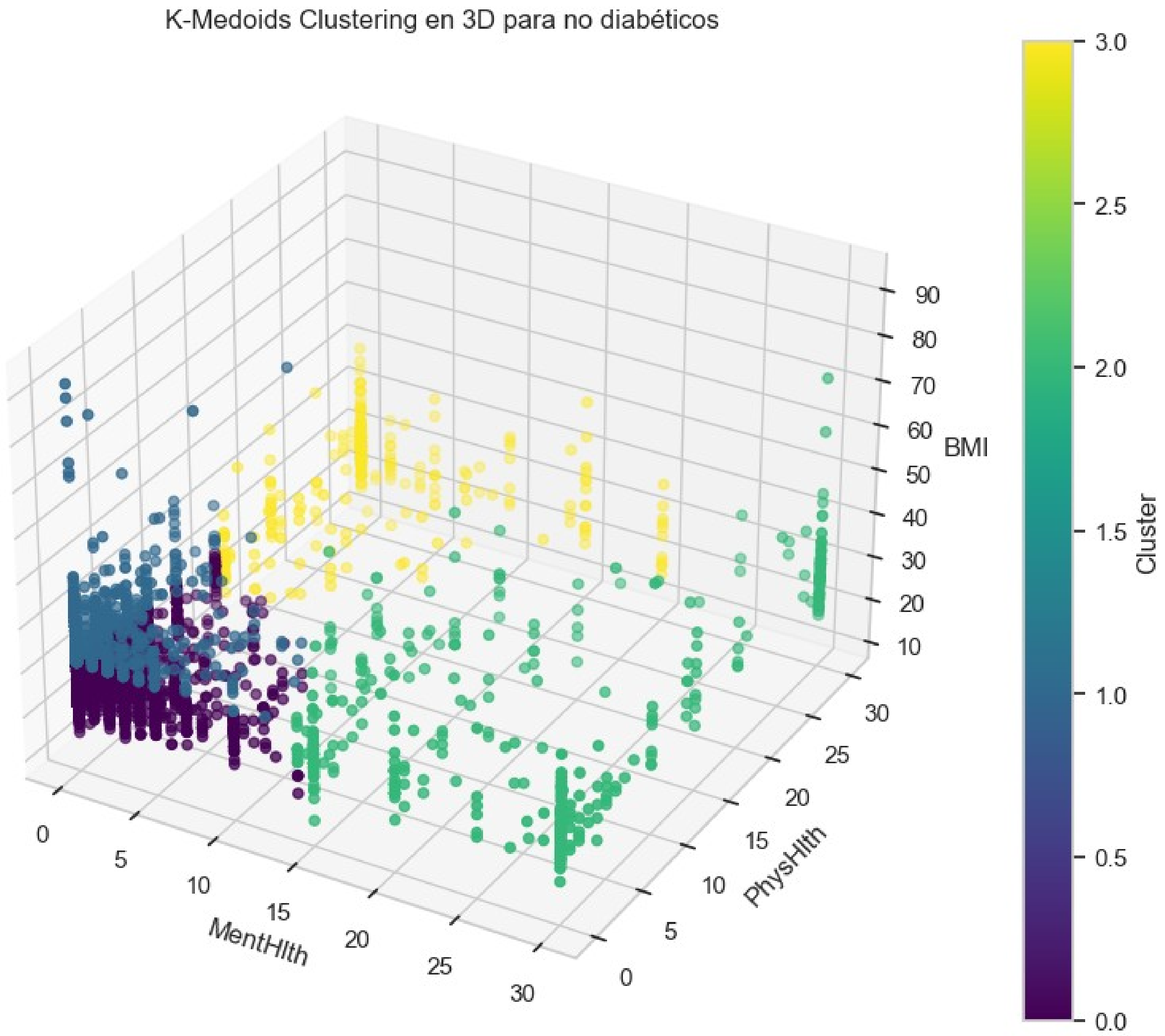
3.9.- Representamos los clusters para cada categoría de enfermedad y las tres variables deseadas

Escogemos tres pero podrían ser cualquiera del dataset

```
# Variables a comparar
var1 = "MentHlth"
var2 = "PhysHlth"
var3 = "BMI"
```

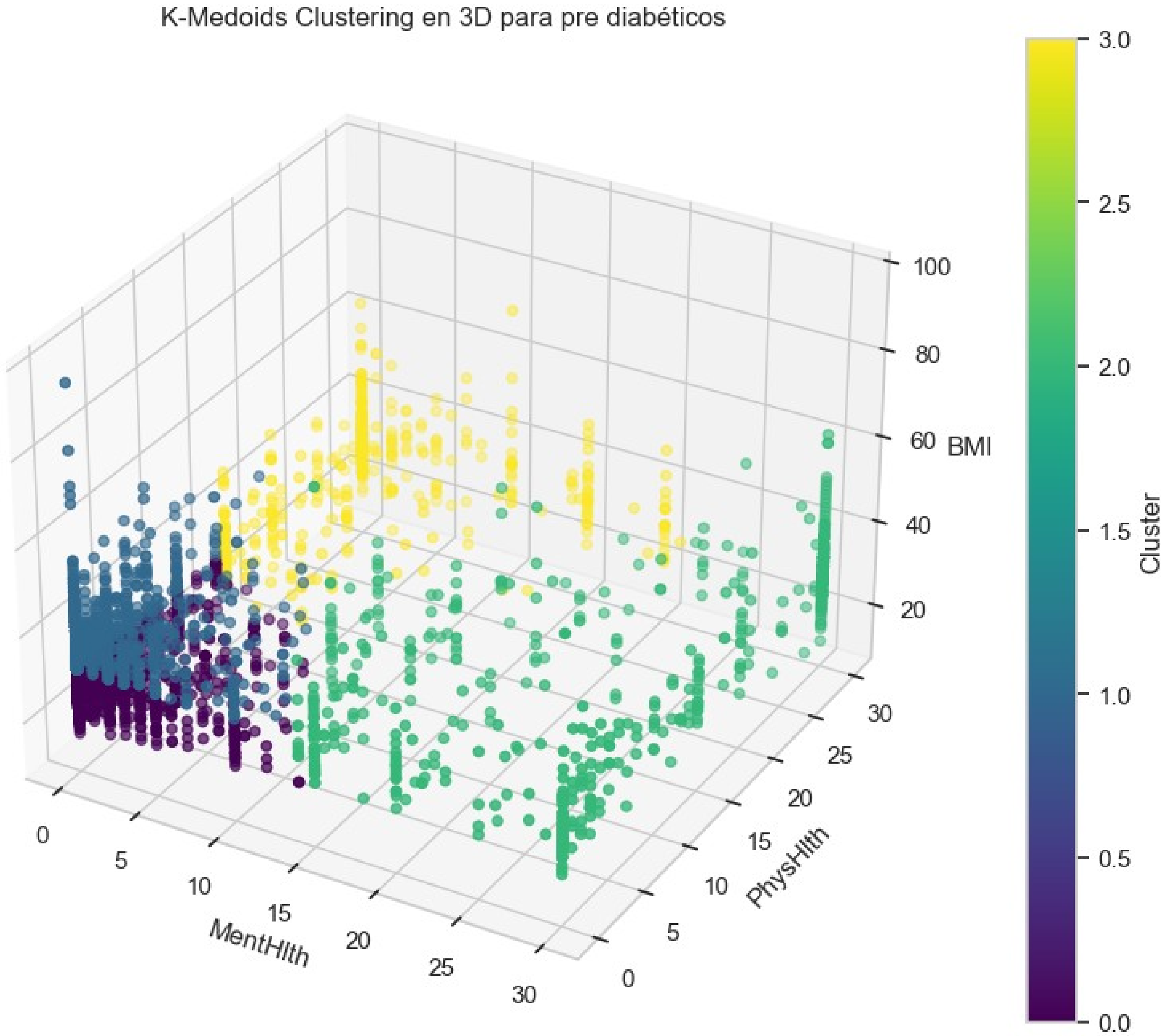
3.9.1.- No diabetico 3D Plot

```
# Filtramos los no diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 0]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Medoids Clustering en 3D para no diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



3.9.2.- Pre diabetico 3D Plot

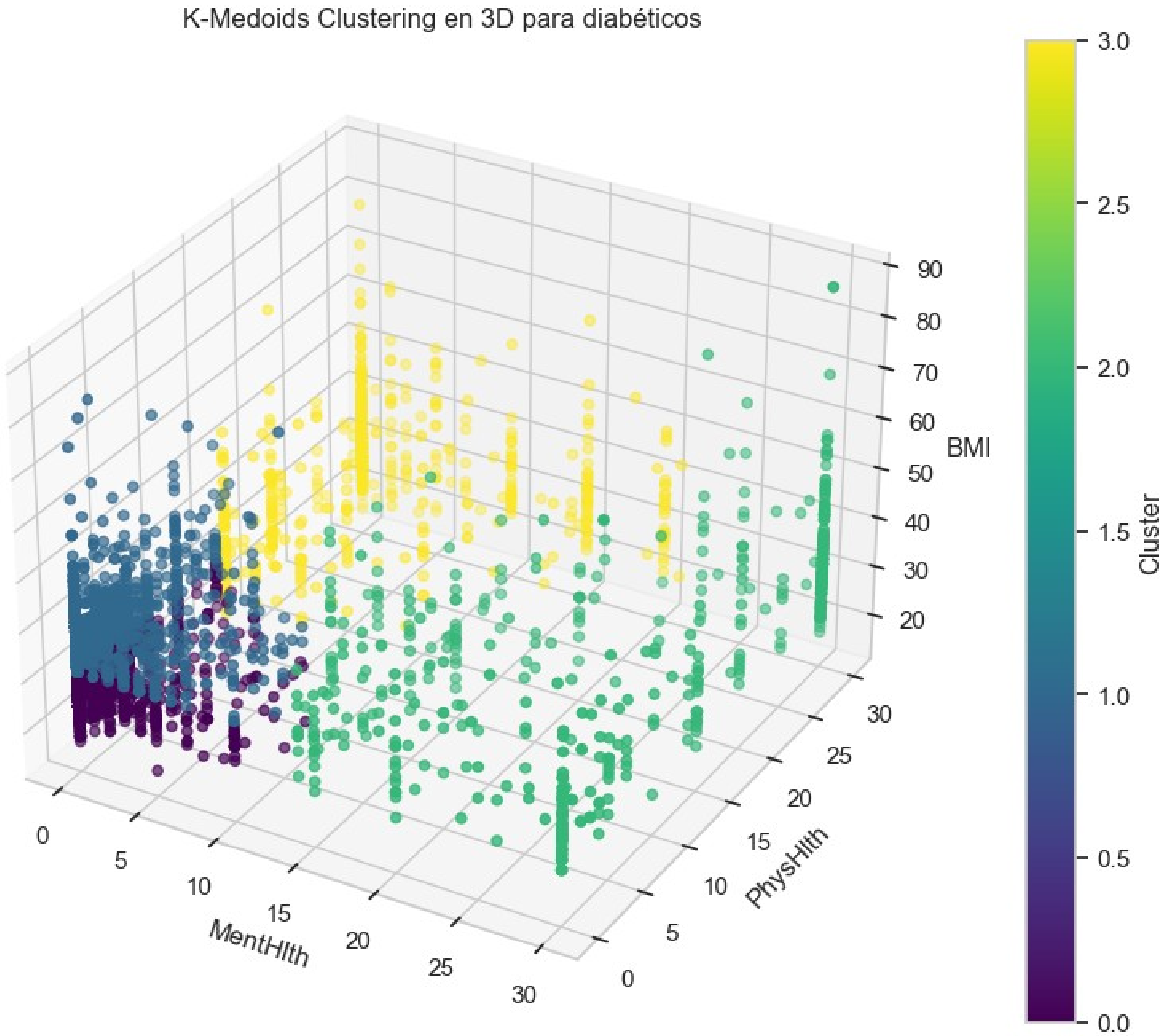
```
# Filtramos los pre diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 1]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Medoids Clustering en 3D para pre diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



3.9.3.- Diabetico 3D Plot

```
# Filtramos los diabeticos
clustering_df_filtrado = clustering_df[clustering_df['diabetic_label']
== 2]

# Creamos una figura 3D
fig = plt.figure(figsize=(15, 8))
ax = fig.add_subplot(111, projection='3d')
```

```python
# Scatter plot 3D
scatter = ax.scatter(clustering_df_filtrado[var1],
clustering_df_filtrado[var2], clustering_df_filtrado[var3],
c=clustering_df_filtrado["cluster_label"], cmap='viridis', marker='o')

# Añadimos etiquetas y titulos
ax.set_xlabel(var1)
ax.set_ylabel(var2)
ax.set_zlabel(var3)
ax.set_title('K-Medoids Clustering en 3D para diabéticos')

# añadimos leyenda de clusteres como barra de colores
fig.colorbar(scatter, ax=ax, label='Cluster')

# Mostramos la gráfica
plt.show()
```



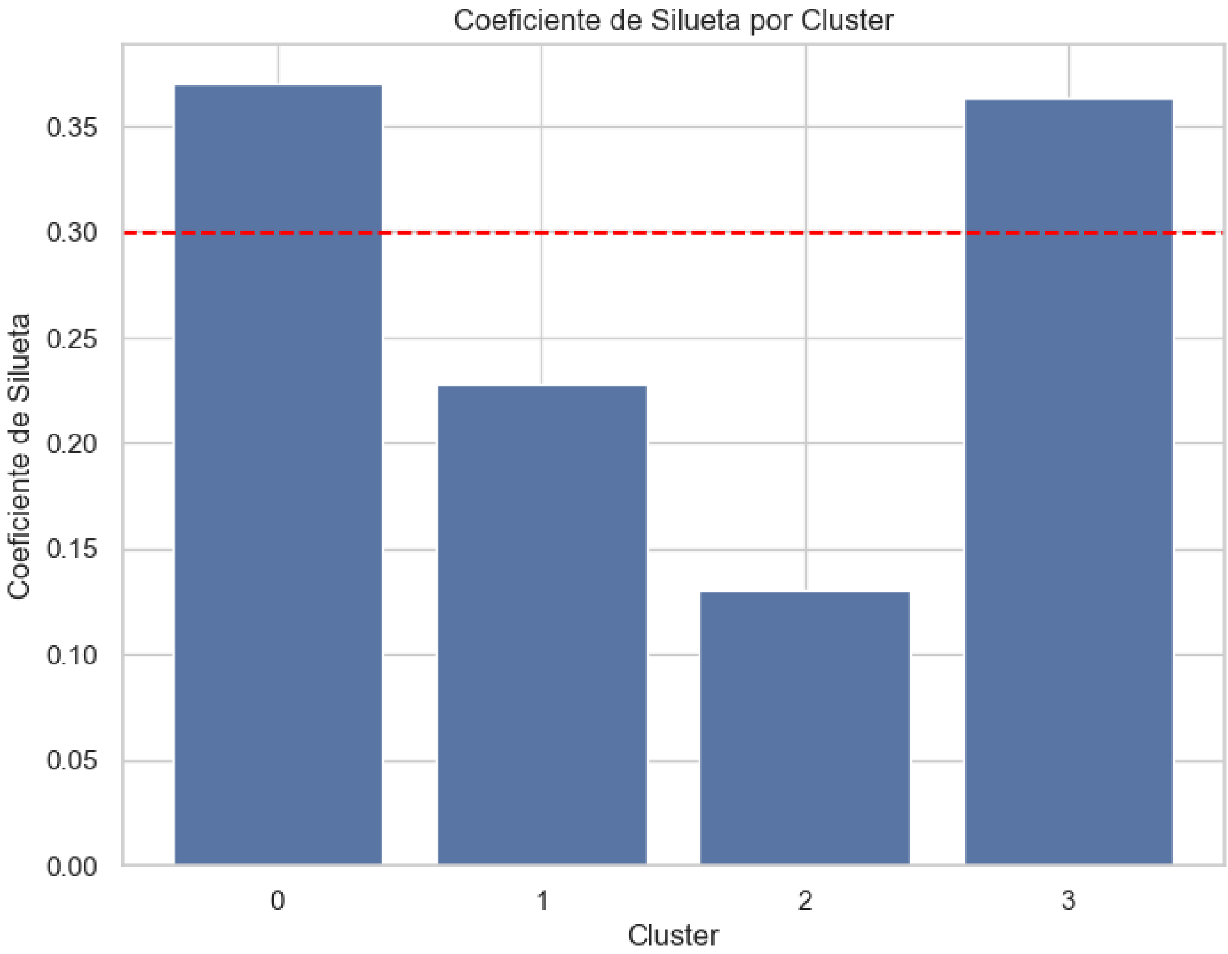
3.10.- Evaluación de calidad

```
clustering_df = pd.read_csv('clustering_df_reducido.csv')

# Indice de silueta
silhouette_avg = silhouette_score(clustering_df,
kmedoids_pipeline.named_steps['kmedoids'].labels_)
print("El índice de silueta promedio es:", silhouette_avg)

# Indice de Davies-Bouldin
davies_bouldin = davies_bouldin_score(clustering_df,
kmedoids_pipeline.named_steps['kmedoids'].labels_)
print("El índice de Davies-Bouldin es:", davies_bouldin)

evaluacion_calidad["kmedoids"] = {"silueta": silhouette_avg,
"davies_bouldin": davies_bouldin}
```

```
El índice de silueta promedio es: 0.2991221312391194
El índice de Davies-Bouldin es: 1.1530872792512559
```

```
# Calculamos el coeficiente de Silueta promedio
silhouette_avg = silhouette_score(clustering_df,
kmedoids_pipeline.named_steps['kmedoids'].labels_)

# Calculamos el coeficiente de Silueta para cada muestra
sample_silhouette_values = silhouette_samples(clustering_df,
kmedoids_pipeline.named_steps['kmedoids'].labels_)

# Creamos una lista para almacenar los coeficientes de Silueta
promedio por cluster
cluster_silhouette_avg = []

list_siluetes = []
# Creamos un bucle para obtener el coeficiente de Silueta para cada
cluster
for cluster in
range(max(kmedoids_pipeline.named_steps['kmedoids'].labels_)+1):
    # Obtenemos las muestras pertenecientes al cluster actual
    cluster_silhouette_values =
sample_silhouette_values[kmedoids_pipeline.named_steps['kmedoids'].lab
els_ == cluster]

    # Calculamos el coeficiente de Silueta promedio para el cluster
actual
    avg_silhouette = np.mean(cluster_silhouette_values)

    cluster_dict = {'category': cluster, 'value': avg_silhouette}
    list_siluetes.append(cluster_dict)

    # Almacenamos el coeficiente de Silueta promedio en la lista
    cluster_silhouette_avg.append(avg_silhouette)
```

```
cluster_siluetes["kmedoids"] = list_siluetes

# Creamos el gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(range(len(cluster_silhouette_avg)), cluster_silhouette_avg, color="#5975A4")
plt.title('Coeficiente de Silueta por Cluster')
plt.xlabel('Cluster')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range(len(cluster_silhouette_avg)))
plt.axhline(y=silhouette_avg, color="red", linestyle="--")  # Línea para indicar el valor promedio global de Silueta
plt.show()
```



3.11.- Save Models

```
joblib.dump(Clusterer, 'resultados/trained_models/kmedoids.joblib')
```

```
['resultados/trained_models/kmedoids.joblib']
```

4.- AffinityPropagation (Basado en el concepto de "mensajeo")

A diferencia de algoritmos de clusterización como k-means o k-medoids, propagación de afinidad no requiere que el número de grupos estén determinados o estimados antes de correr el algoritmo.

4.1.- Definición de la pipeline para el AffinityPropagation

```
clustering_df = pd.read_csv('clustering_df_reducido.csv')

columnas_deseadas = ['MentHlth', 'PhysHlth', 'BMI']
clustering_df = clustering_df.loc[:, columnas_deseadas]

ap_pipeline = Pipeline([
    ('MMS', MinMaxScaler()),
    ('ap', AffinityPropagation(damping=0.7))
])
```

4.2.- Entrenamos el modelo con la pipeline y los datos del df

```
ap_pipeline.fit(clustering_df)
```

```
Pipeline(steps=[('MMS', MinMaxScaler()),
                ('ap', AffinityPropagation(damping=0.7))])
```

4.3.- Obtenemos las etiquetas (Cluster al que pertenece) para cada dato

```
labels = ap_pipeline.predict(clustering_df)
```

4.4.- Agregar las etiquetas de clusters y de clasificacion al conjunto de datos

```
clustering_df["cluster_label"] = labels
clustering_df["diabetic_label"] = df_sampled["Diabetes_012"]

clustering_df.head()
clustering_df.to_csv('resultados/df_clusters/affinitypropagation.csv')
```

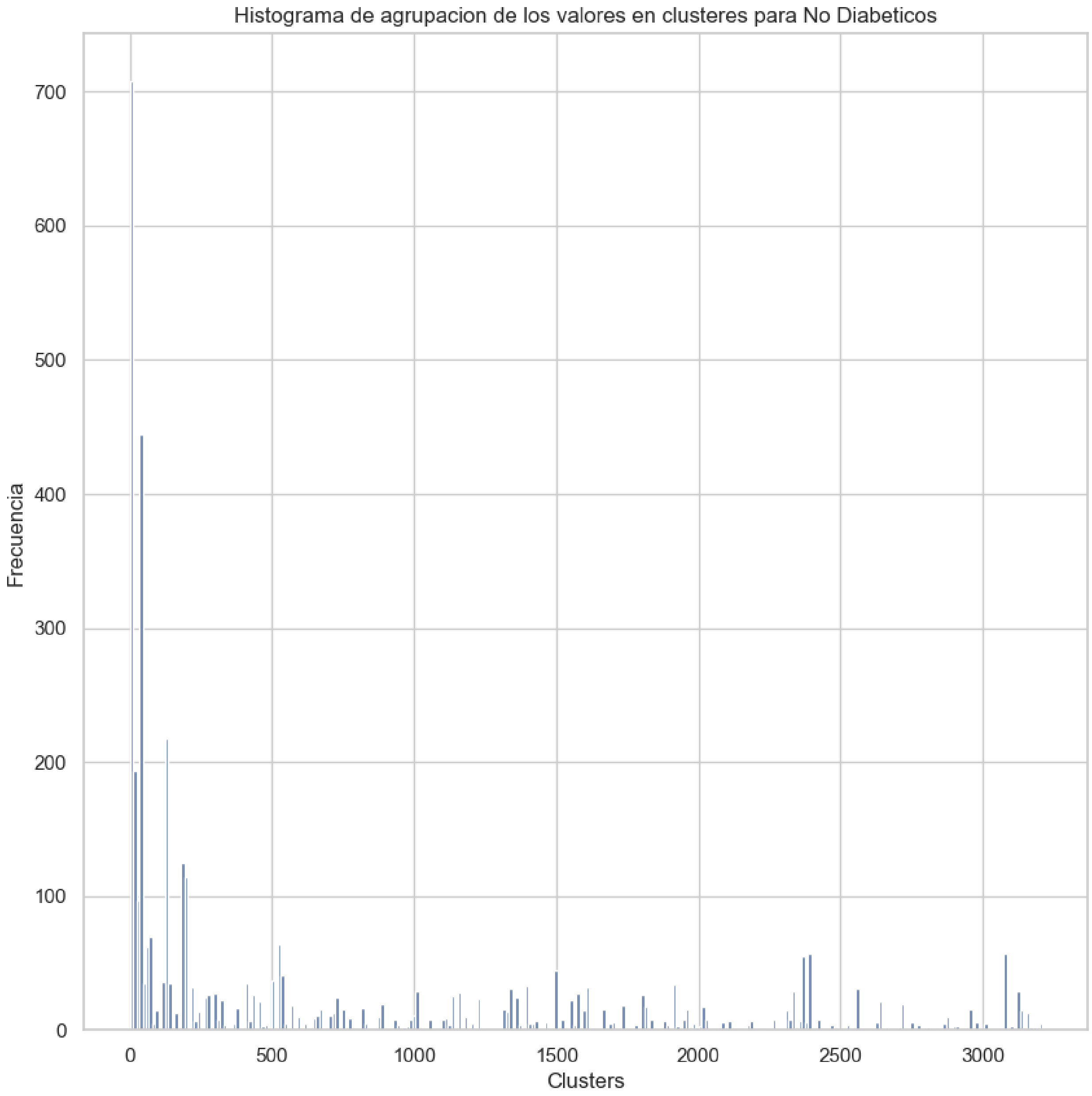
4.5.- Número de clusters encontrados por el algoritmo

```
# Miramos el número de clusters que el algoritmo ha encontrado
clusters = unique(labels)
print(f"Numero de clusters: {len(clusters)}")
```

```
Numero de clusters: 284
```

4.6.- Resumen de pertenencia de clusters por grupo de diabetes

```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 0]
['cluster_label']
plt.figure(figsize=(10, 10))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()


# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] ==
0],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"branchvalues":"total","domain":{"x":[0,1],"y":
[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%
{parent}<br>id=%{id}<extra></extra>","ids":
["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","2
5","27","28","30","32","34","35","38","39","43","45","46","49","50","5
2","60","62","63","68","69","76","89","91","101","113","118","121","12
8","129","131","135","143","160","167","181","183","184","185","187","
193","194","195","196","201","202","203","215","217","231","241","266"
,"267","269","272","278","297","302","308","317","321","323","324","33
4","367","377","378","381","408","417","421","422","433","462","470","
482","501","502","528","535","550","552","575","576","591","615","617"
```

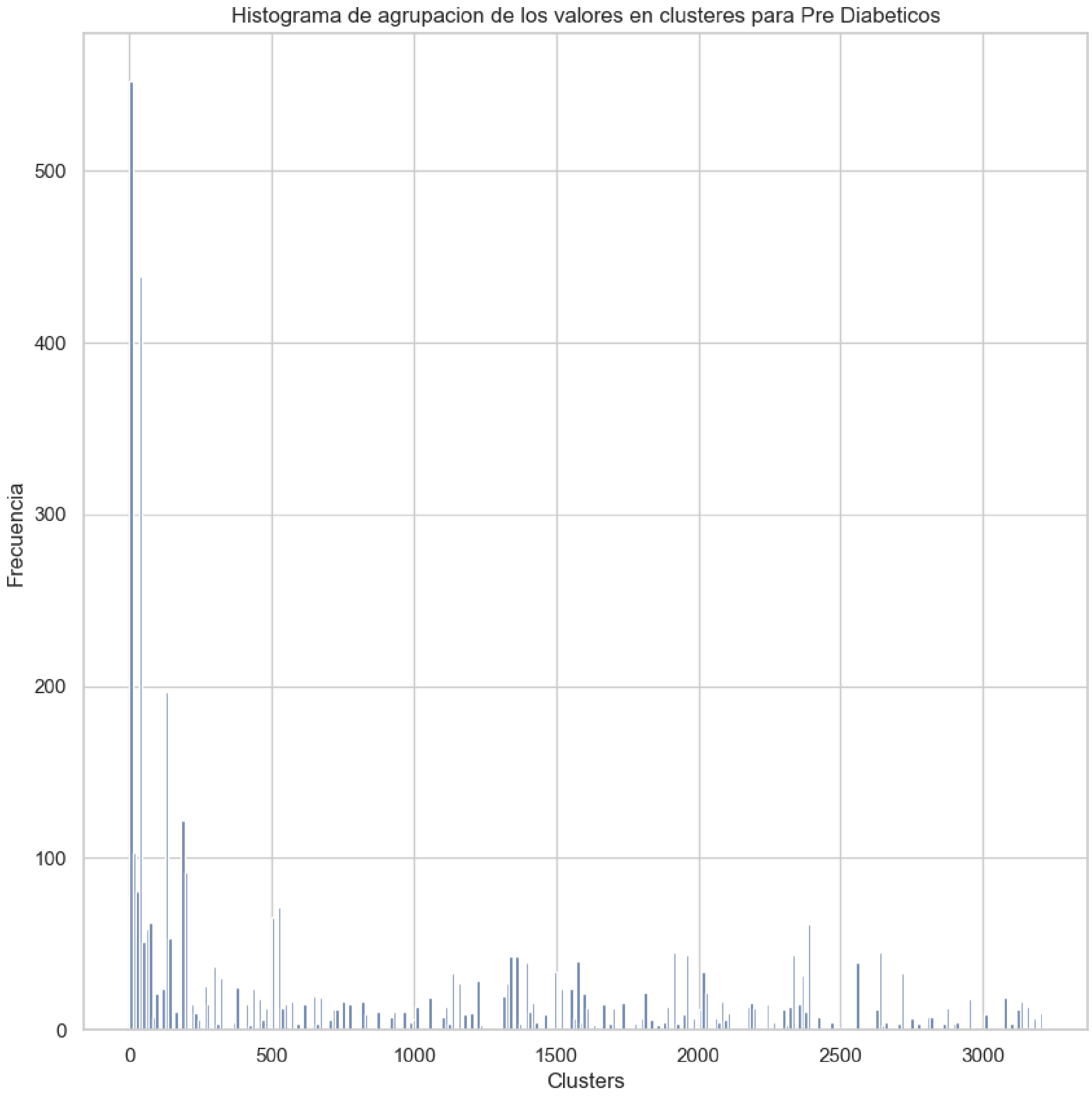
```
,"647","665","675","676","708","712","717","723","733","746","752","77
1","775","804","820","833","875","876","888","890","924","927","930","
944","969","972","988","1004","1010","1038","1052","1054","1107","1111
","1116","1119","1139","1141","1156","1185","1199","1209","1221","1225
","1226","1229","1234","1322","1329","1335","1342","1359","1362","1364
","1375","1393","1397","1400","1406","1410","1415","1433","1459","1493
","1502","1517","1523","1550","1551","1564","1572","1575","1577","1580
","1582","1598","1602","1605","1612","1639","1668","1670","1690","1695
","1740","1782","1785","1805","1808","1811","1837","1861","1881","1893
","1917","1922","1951","1956","1965","1985","1989","2000","2011","2015
","2022","2026","2068","2088","2100","2105","2110","2172","2173","2183
","2200","2242","2263","2299","2312","2319","2327","2332","2334","2337
","2351","2369","2372","2383","2385","2386","2390","2403","2420","2466
","2525","2558","2562","2611","2626","2633","2656","2703","2713","2719
","2727","2744","2749","2753","2775","2803","2818","2863","2874","2877
","2902","2915","2933","2953","2954","2980","3016","3084","3106","3122
","3126","3134","3136","3161","3165","3184","3209"],"labels":
["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","2
5","27","28","30","32","34","35","38","39","43","45","46","49","50","5
2","60","62","63","68","69","76","89","91","101","113","118","121","12
8","129","131","135","143","160","167","181","183","184","185","187","
193","194","195","196","201","202","203","215","217","231","241","266"
,"267","269","272","278","297","302","308","317","321","323","324","33
4","367","377","378","381","408","417","421","422","433","462","470","
482","501","502","528","535","550","552","575","576","591","615","617"
,"647","665","675","676","708","712","717","723","733","746","752","77
1","775","804","820","833","875","876","888","890","924","927","930","
944","969","972","988","1004","1010","1038","1052","1054","1107","1111
","1116","1119","1139","1141","1156","1185","1199","1209","1221","1225
","1226","1229","1234","1322","1329","1335","1342","1359","1362","1364
","1375","1393","1397","1400","1406","1410","1415","1433","1459","1493
","1502","1517","1523","1550","1551","1564","1572","1575","1577","1580
","1582","1598","1602","1605","1612","1639","1668","1670","1690","1695
","1740","1782","1785","1805","1808","1811","1837","1861","1881","1893
","1917","1922","1951","1956","1965","1985","1989","2000","2011","2015
","2022","2026","2068","2088","2100","2105","2110","2172","2173","2183
","2200","2242","2263","2299","2312","2319","2327","2332","2334","2337
","2351","2369","2372","2383","2385","2386","2390","2403","2420","2466
","2525","2558","2562","2611","2626","2633","2656","2703","2713","2719
","2727","2744","2749","2753","2775","2803","2818","2863","2874","2877
","2902","2915","2933","2953","2954","2980","3016","3084","3106","3122
","3126","3134","3136","3161","3165","3184","3209"],"name":"","parents
":
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
```

```
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","",""],"type":"treemap","valu
es":
[134,179,3,92,30,15,72,55,9,119,16,98,71,8,9,14,3,10,20,41,124,178,9,3
,16,114,7,12,6,10,25,6,31,48,12,10,5,9,6,19,9,8,47,3,19,148,35,6,7,4,8
,2,103,8,2,24,35,31,10,6,6,5,27,7,14,5,16,3,5,21,21,6,8,2,3,6,11,4,5,1
2,3,2,4,31,5,2,26,21,3,4,9,28,64,41,4,1,11,8,10,2,3,9,11,9,7,11,7,2,4,
24,7,9,1,8,1,17,5,7,3,10,10,1,2,6,4,2,3,8,11,29,1,4,4,8,4,5,4,11,14,28
,10,1,4,1,8,11,3,1,16,14,23,8,14,6,4,4,12,16,5,4,1,5,7,6,9,36,2,6,7,15
,4,13,4,6,4,2,14,1,30,2,1,7,9,5,6,19,3,1,26,13,5,8,1,7,4,34,3,8,7,9,5,
2,2,1,14,4,8,2,6,2,4,3,3,1,7,1,1,8,1,15,2,6,15,9,5,7,24,31,6,1,2,54,1,
8,4,4,10,21,1,6,21,2,2,9,11,1,1,3,3,4,2,1,5,3,7,3,3,1,15,1,6,5,57,3,4,
25,8,7,13,2,2,5]}],"layout":{"legend":{"tracegroupgap":0},"template":
{"data":{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
```

```
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

```python
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]
```

```
['cluster_label']
plt.figure(figsize=(10, 10))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
Pre Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] ==
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



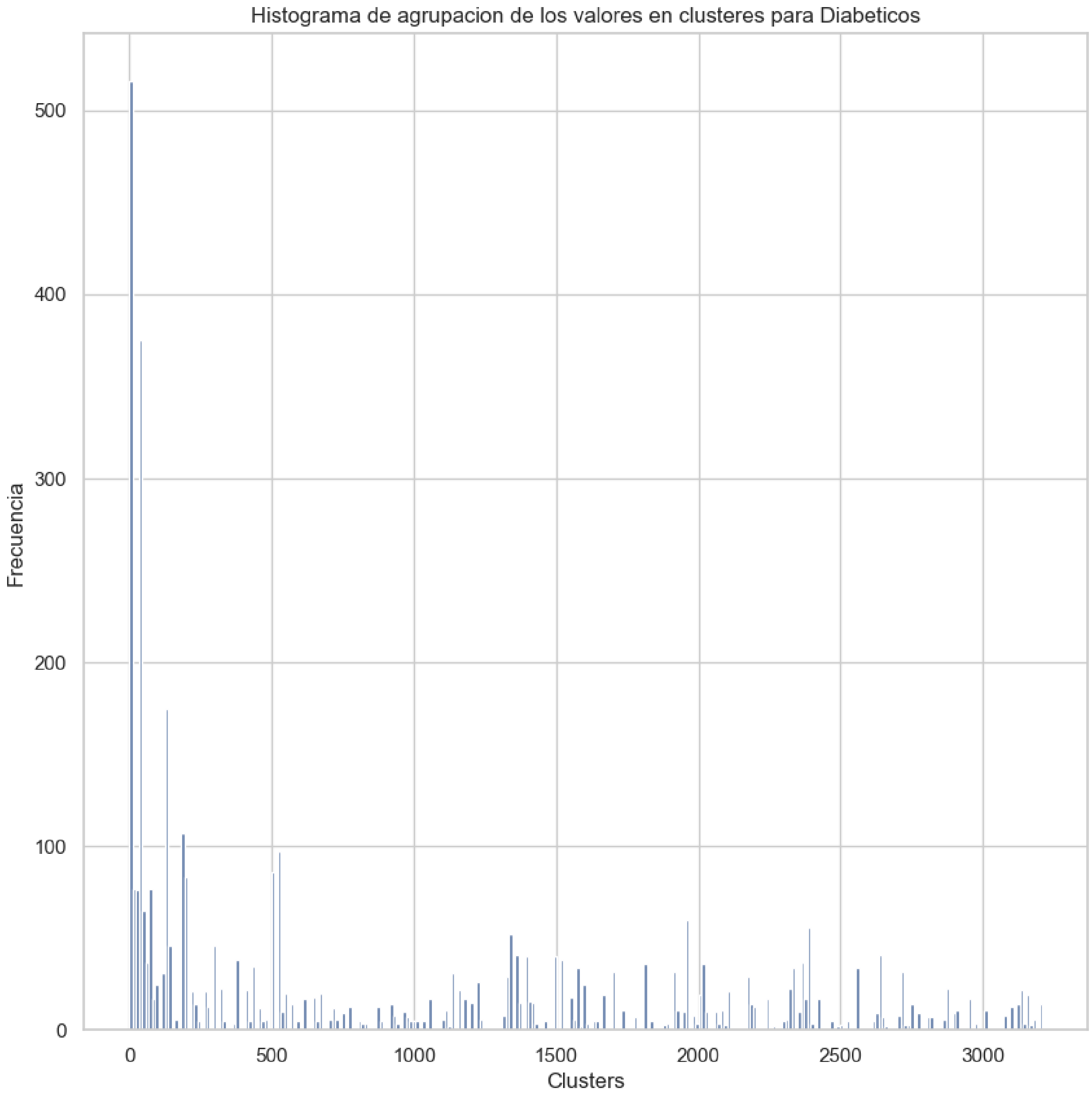
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","25","27","28","30","32","34","35","38","39","43","45","46","49","50","52","60","62","63","68","69","76","89","91","101","113","118","121","128","129","131","135","143","160","167","181","183","184","185","187","193","194","195","196","201","202","203","215","217","231","241","266","267","269","272","278","297","302","308","317","321","323","324","367","377","378","381","408","417","421","422","433","462","470","482","501","502","528","535","550","552","575","576","591","615","617","647"

```
,"665","675","676","708","712","717","723","733","746","752","771","77
5","804","820","833","875","876","890","924","927","930","944","969","
972","988","1004","1010","1038","1052","1054","1107","1111","1116","11
19","1139","1141","1156","1185","1199","1209","1221","1225","1226","12
29","1234","1245","1322","1329","1335","1342","1359","1362","1364","13
75","1393","1397","1400","1406","1410","1415","1433","1459","1493","15
02","1517","1523","1550","1551","1564","1572","1575","1577","1580","15
82","1598","1602","1605","1612","1628","1639","1668","1670","1690","16
95","1740","1782","1785","1805","1808","1811","1837","1861","1881","18
93","1917","1922","1951","1956","1965","1985","1989","2000","2011","20
15","2016","2022","2026","2057","2064","2068","2088","2100","2105","21
10","2172","2173","2183","2197","2200","2242","2263","2299","2300","23
12","2319","2327","2332","2334","2337","2351","2369","2372","2383","23
85","2386","2390","2420","2466","2488","2525","2558","2562","2611","26
26","2633","2649","2656","2701","2703","2713","2719","2727","2744","27
49","2753","2775","2803","2818","2863","2874","2877","2902","2915","29
33","2953","2954","2980","3016","3084","3106","3122","3126","3134","31
36","3148","3161","3165","3184","3209"],"labels":
["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","2
5","27","28","30","32","34","35","38","39","43","45","46","49","50","5
2","60","62","63","68","69","76","89","91","101","113","118","121","12
8","129","131","135","143","160","167","181","183","184","185","187","
193","194","195","196","201","202","203","215","217","231","241","266"
,"267","269","272","278","297","302","308","317","321","323","324","36
7","377","378","381","408","417","421","422","433","462","470","482","
501","502","528","535","550","552","575","576","591","615","617","647"
,"665","675","676","708","712","717","723","733","746","752","771","77
5","804","820","833","875","876","890","924","927","930","944","969","
972","988","1004","1010","1038","1052","1054","1107","1111","1116","11
19","1139","1141","1156","1185","1199","1209","1221","1225","1226","12
29","1234","1245","1322","1329","1335","1342","1359","1362","1364","13
75","1393","1397","1400","1406","1410","1415","1433","1459","1493","15
02","1517","1523","1550","1551","1564","1572","1575","1577","1580","15
82","1598","1602","1605","1612","1628","1639","1668","1670","1690","16
95","1740","1782","1785","1805","1808","1811","1837","1861","1881","18
93","1917","1922","1951","1956","1965","1985","1989","2000","2011","20
15","2016","2022","2026","2057","2064","2068","2088","2100","2105","21
10","2172","2173","2183","2197","2200","2242","2263","2299","2300","23
12","2319","2327","2332","2334","2337","2351","2369","2372","2383","23
85","2386","2390","2420","2466","2488","2525","2558","2562","2611","26
26","2633","2649","2656","2701","2703","2713","2719","2727","2744","27
49","2753","2775","2803","2818","2863","2874","2877","2902","2915","29
33","2953","2954","2980","3016","3084","3106","3122","3126","3134","31
36","3148","3161","3165","3184","3209"],"name":"","parents":
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
```

```
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
],"type":"treemap","values":
[90,123,2,50,34,10,98,12,13,120,3,68,28,4,3,8,2,6,4,58,121,168,7,1,9,1
32,13,4,19,15,30,6,23,42,15,6,8,11,10,10,4,10,68,13,4,112,54,6,5,6,2,5
,99,10,6,6,13,29,13,14,11,4,11,10,6,4,21,1,4,11,32,5,4,10,2,12,6,5,13,
2,10,7,8,1,2,24,18,6,13,14,52,72,13,12,3,14,3,4,4,11,20,4,5,14,6,5,5,2
,12,5,12,10,5,2,17,9,8,3,1,8,6,5,1,11,2,5,6,14,2,7,12,8,7,7,4,23,10,27
,9,7,3,3,4,13,9,3,1,20,27,38,5,12,12,19,4,7,23,9,4,7,16,5,9,14,20,3,21
,7,17,7,19,7,11,3,5,18,3,9,4,3,1,2,13,4,13,16,3,1,7,13,9,6,3,5,14,45,4
,9,12,32,7,1,2,10,20,11,3,22,5,2,5,17,6,3,7,10,4,16,8,5,15,5,5,7,3,4,1
0,20,11,13,15,9,23,11,8,10,44,8,5,1,2,23,16,1,12,45,3,5,3,1,12,21,1,2,
3,4,4,8,8,4,6,7,4,5,1,15,3,2,9,19,4,4,8,14,3,2,14,2,7,10]}],"layout":
{"legend":{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
```

```
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 2]
['cluster_label']
plt.figure(figsize=(10, 10))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] ==
2],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","25","27","28","30","32","34","35","38","43","45","46","49","50","52","60","62","63","68","69","76","89","91","101","113","118","121","128","129","131","135","143","160","167","181","183","184","185","187","193","194","195","196","201","202","203","215","217","231","241","266","267","269","272","278","297","302","308","317","321","323","324","334","367","377","378","381","408","417","421","422","433","462","470","482","501","502","528","535","550","552","575","576","591","615","617","647
```

```
","665","675","676","708","712","717","723","733","746","752","771","7
75","804","820","833","875","876","888","890","924","927","930","944",
"969","972","988","1004","1010","1038","1052","1054","1107","1111","11
16","1119","1139","1141","1156","1185","1199","1209","1221","1225","12
26","1229","1234","1245","1322","1329","1335","1342","1359","1362","13
64","1375","1393","1397","1400","1406","1410","1415","1433","1459","14
93","1502","1517","1523","1550","1551","1564","1572","1575","1577","15
80","1582","1598","1602","1605","1612","1628","1639","1668","1670","16
95","1740","1782","1785","1805","1808","1811","1837","1861","1881","18
93","1917","1922","1951","1956","1965","1985","1989","2000","2011","20
15","2016","2022","2026","2057","2064","2068","2088","2100","2105","21
10","2172","2173","2183","2197","2200","2242","2263","2299","2300","23
12","2319","2327","2332","2334","2337","2351","2369","2372","2383","23
85","2386","2390","2403","2420","2466","2488","2501","2525","2558","25
62","2611","2626","2633","2649","2656","2701","2703","2713","2719","27
27","2744","2749","2753","2775","2803","2818","2863","2874","2877","29
02","2915","2933","2953","2954","2980","3016","3084","3106","3122","31
26","3134","3136","3148","3161","3165","3184","3209"],"labels":
["0","1","2","3","4","5","6","7","10","11","12","16","17","21","24","2
5","27","28","30","32","34","35","38","43","45","46","49","50","52","6
0","62","63","68","69","76","89","91","101","113","118","121","128","1
29","131","135","143","160","167","181","183","184","185","187","193",
"194","195","196","201","202","203","215","217","231","241","266","267
","269","272","278","297","302","308","317","321","323","324","334","3
67","377","378","381","408","417","421","422","433","462","470","482",
"501","502","528","535","550","552","575","576","591","615","617","647
","665","675","676","708","712","717","723","733","746","752","771","7
75","804","820","833","875","876","888","890","924","927","930","944",
"969","972","988","1004","1010","1038","1052","1054","1107","1111","11
16","1119","1139","1141","1156","1185","1199","1209","1221","1225","12
26","1229","1234","1245","1322","1329","1335","1342","1359","1362","13
64","1375","1393","1397","1400","1406","1410","1415","1433","1459","14
93","1502","1517","1523","1550","1551","1564","1572","1575","1577","15
80","1582","1598","1602","1605","1612","1628","1639","1668","1670","16
95","1740","1782","1785","1805","1808","1811","1837","1861","1881","18
93","1917","1922","1951","1956","1965","1985","1989","2000","2011","20
15","2016","2022","2026","2057","2064","2068","2088","2100","2105","21
10","2172","2173","2183","2197","2200","2242","2263","2299","2300","23
12","2319","2327","2332","2334","2337","2351","2369","2372","2383","23
85","2386","2390","2403","2420","2466","2488","2501","2525","2558","25
62","2611","2626","2633","2649","2656","2701","2703","2713","2719","27
27","2744","2749","2753","2775","2803","2818","2863","2874","2877","29
02","2915","2933","2953","2954","2980","3016","3084","3106","3122","31
26","3134","3136","3148","3161","3165","3184","3209"],"name":"","paren
ts":
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"",""],"type":"treemap","values":
[75,125,1,40,43,18,87,6,13,108,2,51,23,1,8,8,2,4,4,50,112,138,5,6,114,
13,8,22,22,20,2,15,61,15,1,17,6,19,13,6,12,79,17,5,74,46,5,1,6,1,7,77,
16,8,1,4,35,17,15,3,7,14,14,5,5,12,4,7,6,36,10,1,2,1,12,8,5,4,22,2,14,
17,5,2,3,35,12,5,6,18,68,97,10,14,6,13,1,5,3,14,18,5,7,13,6,5,5,2,6,4,
5,8,5,5,4,4,10,3,2,3,14,5,3,4,10,7,5,5,5,5,10,7,6,3,8,2,22,9,22,17,10,
5,8,2,11,5,6,1,8,29,44,8,14,6,21,15,4,28,8,6,10,15,4,5,29,11,5,33,2,16
,6,11,8,13,2,2,15,10,2,2,5,5,5,14,32,11,6,1,1,14,22,5,1,3,4,32,11,10,1
6,44,8,4,2,17,27,8,1,10,2,8,4,11,3,6,15,24,5,14,9,4,17,2,1,4,6,7,16,19
,8,7,10,12,25,17,19,7,30,4,17,5,2,3,5,20,14,5,9,41,7,2,4,4,11,21,3,3,6
,8,9,7,7,6,15,8,9,11,1,12,5,4,11,8,13,4,10,11,11,4,19,3,6,14]}],"layou
t":{"legend":{"tracegroupgap":0},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
```

```
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
```

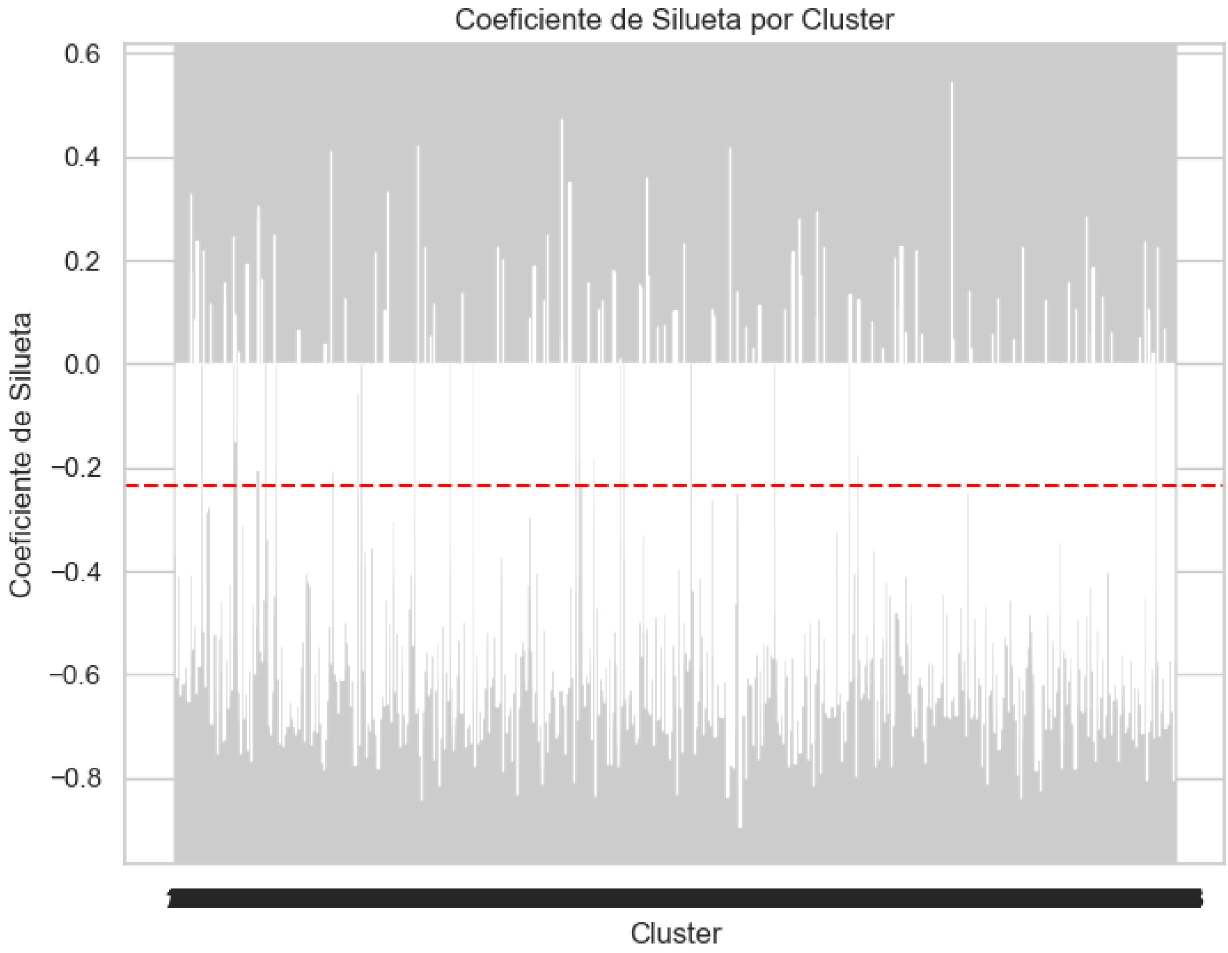
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}

4.7.- Evaluación de calidad

```
clustering_df = pd.read_csv('clustering_df_reducido.csv')

# Indice de silueta
silhouette_avg = silhouette_score(clustering_df,
ap_pipeline.named_steps['ap'].labels_)
print("El índice de silueta promedio es:", silhouette_avg)

# Indice de Davies-Bouldin
davies_bouldin = davies_bouldin_score(clustering_df,
ap_pipeline.named_steps['ap'].labels_)
print("El índice de Davies-Bouldin es:", davies_bouldin)

evaluacion_calidad["affinitypropagation"] = {"silueta":
silhouette_avg, "davies_bouldin": davies_bouldin}
```

```
El índice de silueta promedio es: -0.2356539244826847
El índice de Davies-Bouldin es: 3.1002663310808183
```

```
# Calculamos el coeficiente de Silueta promedio
silhouette_avg = silhouette_score(clustering_df,
ap_pipeline.named_steps['ap'].labels_)

# Calculamos el coeficiente de Silueta para cada muestra
sample_silhouette_values = silhouette_samples(clustering_df,
ap_pipeline.named_steps['ap'].labels_)

# Creamos una lista para almacenar los coeficientes de Silueta
promedio por cluster
cluster_silhouette_avg = []

list_siluetes = []
# Creamos un bucle para obtener el coeficiente de Silueta para cada
cluster
for cluster in range(max(ap_pipeline.named_steps['ap'].labels_)+1):
    # Obtenemos las muestras pertenecientes al cluster actual
    cluster_silhouette_values =
sample_silhouette_values[ap_pipeline.named_steps['ap'].labels_ ==
cluster]

    # Calculamos el coeficiente de Silueta promedio para el cluster
actual
    avg_silhouette = np.mean(cluster_silhouette_values)

    cluster_dict = {'category': cluster, 'value': avg_silhouette}
    list_siluetes.append(cluster_dict)
    # Almacenamos el coeficiente de Silueta promedio en la lista
    cluster_silhouette_avg.append(avg_silhouette)

cluster_siluetes["affinitypropagation"] = list_siluetes

# Creamos el gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(range(len(cluster_silhouette_avg)-1),
cluster_silhouette_avg[1:], color="#5975A4")
plt.title('Coeficiente de Silueta por Cluster')
plt.xlabel('Cluster')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range(len(cluster_silhouette_avg)))
plt.axhline(y=silhouette_avg, color="red", linestyle="--")  # Línea
para indicar el valor promedio global de Silueta
plt.show()
```



4.8.- Save Models

```
joblib.dump(ap_pipeline, 
'resultados/trained_models/affinitypropagation.joblib')

['resultados/trained_models/affinitypropagation.joblib']
```

5.- OPTICS (Basado en el concepto de "Reachability Distance")

5.1.- Definición de la pipeline para el OPTICS

```
clustering_df = pd.read_csv('clustering_df_reducido.csv')

columnas_deseadas = ['MentHlth', 'PhysHlth', 'BMI']
clustering_df = clustering_df.loc[:, columnas_deseadas]

optics_pipeline = Pipeline([
    ('MMS', MinMaxScaler()),
    ('optics', OPTICS(eps=0.75, min_samples=20))
])
```

5.2.- Entrenamos el modelo con la pipeline y los datos del df

```
optics_pipeline.fit(clustering_df)

C:\Users\Marcos\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-
packages\Python38\site-packages\sklearn\cluster\_optics.py:932: 
RuntimeWarning:

divide by zero encountered in divide


Pipeline(steps=[('MMS', MinMaxScaler()),
                ('optics', OPTICS(eps=0.75, min_samples=20))])
```

5.3.- Obtenemos las etiquetas (Cluster al que pertenece) para cada dato

```
labels = optics_pipeline.fit_predict(clustering_df)

C:\Users\Marcos\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-
packages\Python38\site-packages\sklearn\cluster\_optics.py:932: 
RuntimeWarning:

divide by zero encountered in divide
```

5.4.- Agregar las etiquetas de clusters y de clasificacion al conjunto de datos

```
clustering_df["cluster_label"] = labels
clustering_df["diabetic_label"] = df_sampled["Diabetes_012"]

clustering_df.head()
clustering_df.to_csv('resultados/df_clusters/optics.csv')
```

5.5.- Número de clusters encontrados por el algoritmo

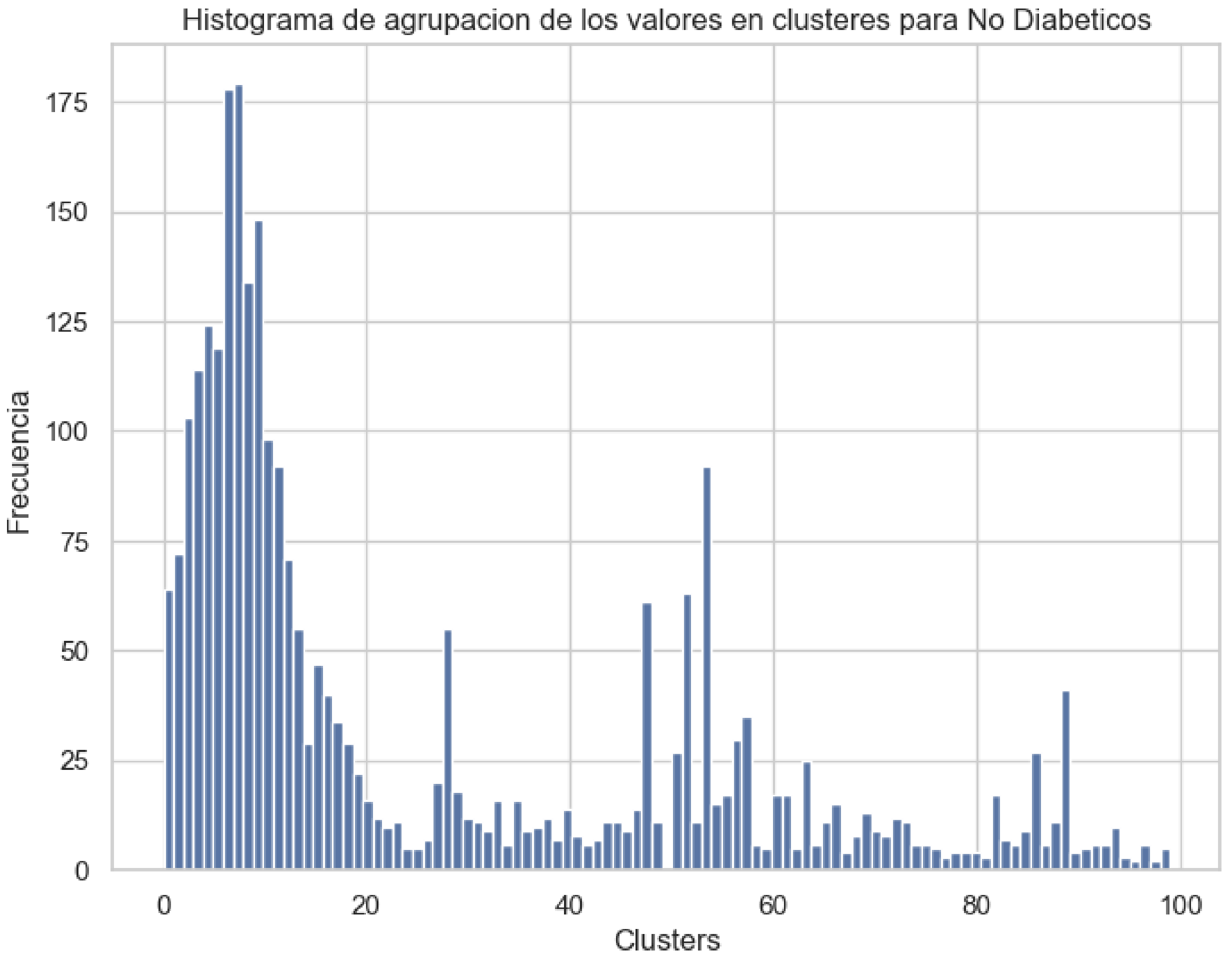
Cuando el algoritmo OPTICS asigna un punto a un cluster con un valor de -1, significa que ese punto puede ser considerado como un "punto de ruido" o un outlier. En términos de distancia, un valor de -1 indica que el punto está fuera del alcance de los clusters identificados, por lo que no pertenece claramente a ningún cluster específico.

```
# Miramos el número de clusters que el algoritmo ha encontrado
clusters = unique(labels)
print(f"Numero de clusters: {len(clusters)}")

# Numero de outliers
outlayers = len(clustering_df[clustering_df['cluster_label'] == -1])
print(f"Numero de outlayers: {outlayers}")
```

```
Numero de clusters: 101
Numero de outlayers: 3943
```

5.6.- Resumen de pertenencia de clusters por grupo de diabetes

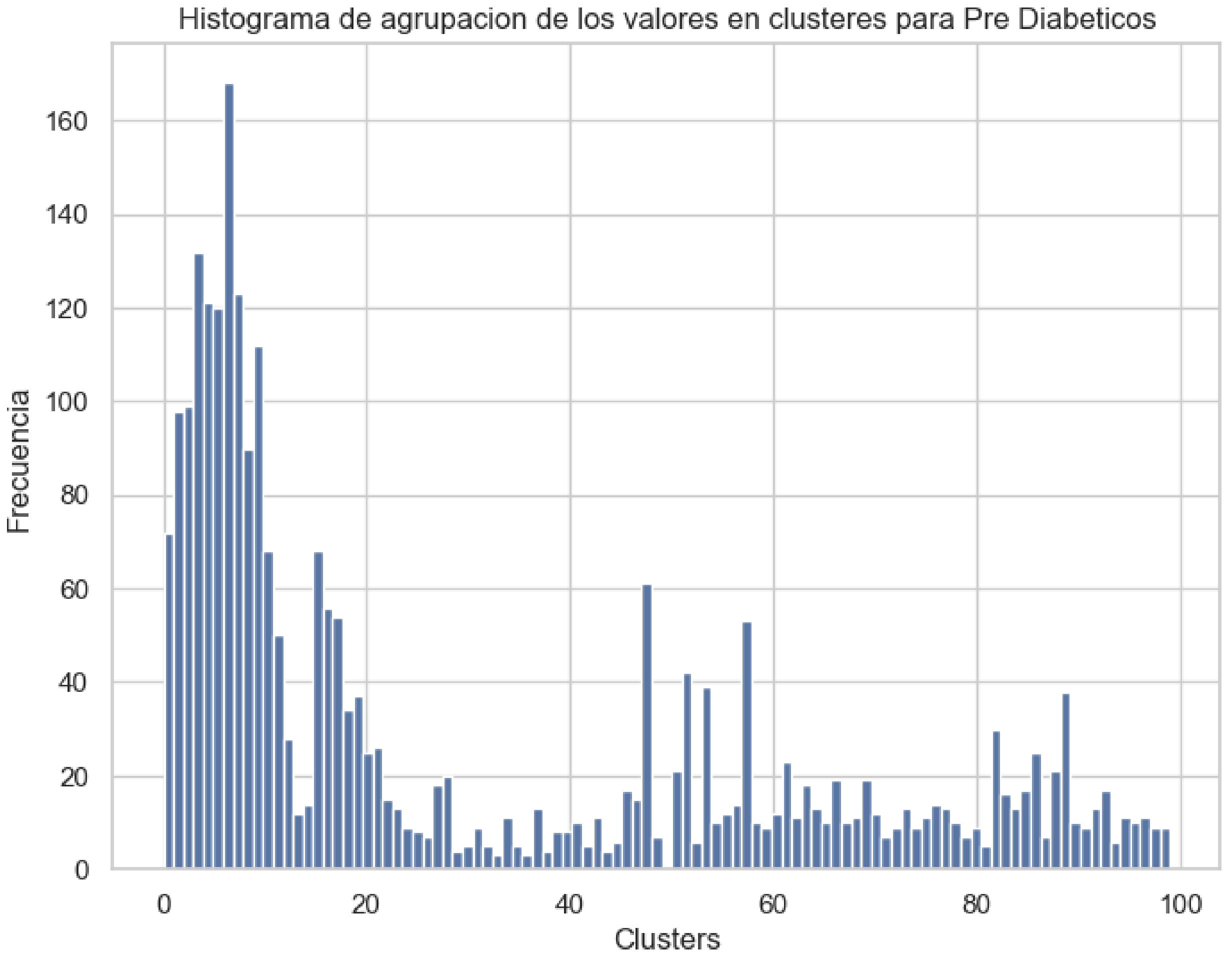
```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 0]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 0]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"branchvalues":"total","domain":{"x":[0,1],"y":
[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%
{parent}<br>id=%{id}<extra></extra>","ids":
["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15"
,"16","17","18","19","20","21","22","23","24","25","26","27","28","29"
,"30","31","32","33","34","35","36","37","38","39","40","41","42","43"
,"44","45","46","47","48","49","50","51","52","53","54","55","56","57"
,"58","59","60","61","62","63","64","65","66","67","68","69","70","71"
,"72","73","74","75","76","77","78","79","80","81","82","83","84","85"
,"86","87","88","89","90","91","92","93","94","95","96","97","98","99"
],"labels":
["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15"
,"16","17","18","19","20","21","22","23","24","25","26","27","28","29"
,"30","31","32","33","34","35","36","37","38","39","40","41","42","43"
,"44","45","46","47","48","49","50","51","52","53","54","55","56","57"
,"58","59","60","61","62","63","64","65","66","67","68","69","70","71"
,"72","73","74","75","76","77","78","79","80","81","82","83","84","85"
,"86","87","88","89","90","91","92","93","94","95","96","97","98","99"
],"name":"","parents":
```

```
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","",""],"type":"treemap","values":
[64,72,103,114,124,119,178,179,134,148,98,92,71,55,29,47,40,34,29,22,1
6,12,10,11,5,5,7,20,55,18,12,11,9,16,6,16,9,10,12,7,14,8,6,7,11,11,9,1
4,61,11,27,63,11,92,15,17,30,35,6,5,17,17,5,25,6,11,15,4,8,13,9,8,12,1
1,6,6,5,3,4,4,4,3,17,7,6,9,27,6,11,41,4,5,6,6,10,3,2,6,2,5]}],"layout"
:{"legend":{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

```python
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
Pre Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```

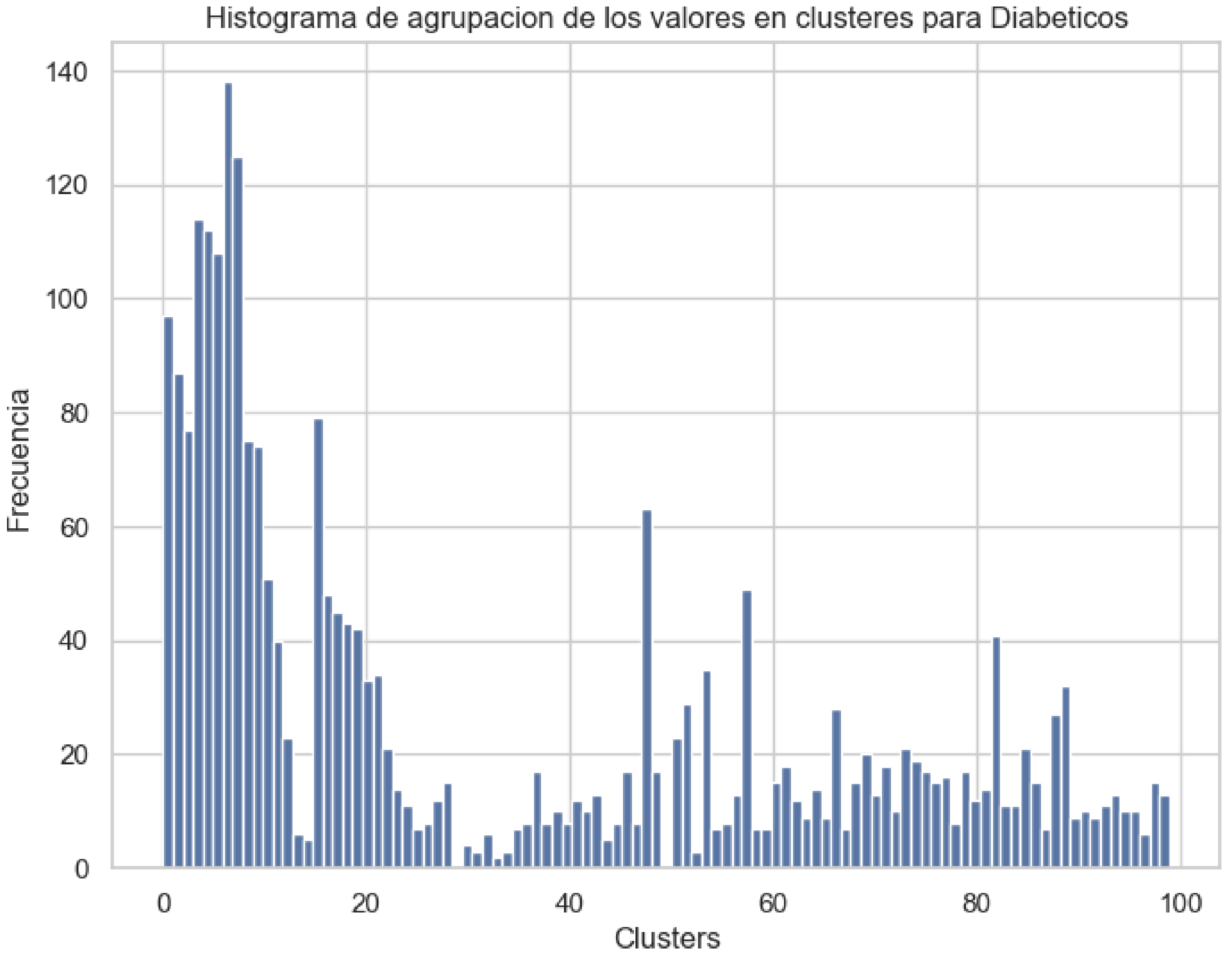


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"branchvalues":"total","domain":{"x":[0,1],"y":
[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%
{parent}<br>id=%{id}<extra></extra>","ids":
["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15"
,"16","17","18","19","20","21","22","23","24","25","26","27","28","29"
,"30","31","32","33","34","35","36","37","38","39","40","41","42","43"
,"44","45","46","47","48","49","50","51","52","53","54","55","56","57"
,"58","59","60","61","62","63","64","65","66","67","68","69","70","71"
,"72","73","74","75","76","77","78","79","80","81","82","83","84","85"
```

```
,"86","87","88","89","90","91","92","93","94","95","96","97","98","99"
],"labels":
["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15"
,"16","17","18","19","20","21","22","23","24","25","26","27","28","29"
,"30","31","32","33","34","35","36","37","38","39","40","41","42","43"
,"44","45","46","47","48","49","50","51","52","53","54","55","56","57"
,"58","59","60","61","62","63","64","65","66","67","68","69","70","71"
,"72","73","74","75","76","77","78","79","80","81","82","83","84","85"
,"86","87","88","89","90","91","92","93","94","95","96","97","98","99"
],"name":"","parents":
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","",""],"type":"treemap","values":
[72,98,99,132,121,120,168,123,90,112,68,50,28,12,14,68,56,54,34,37,25,
26,15,13,9,8,7,18,20,4,5,9,5,3,11,5,3,13,4,8,8,10,5,11,4,6,17,15,61,7,
21,42,6,39,10,12,14,53,10,9,12,23,11,18,13,10,19,10,11,19,12,7,9,13,9,
11,14,13,10,7,9,5,30,16,13,17,25,7,21,38,10,9,13,17,6,11,10,11,9,9]}],
"layout":{"legend":{"tracegroupgap":0},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
```

```
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
```

```
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}

# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 2]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 2]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```


Histograma de agrupacion de los valores en clusteres para Diabeticos

```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25","26","27","28","30","31","32","33","34","35","36","37","38","39","40","41","42","43","44","45","46","47","48","49","50","51","52","53","54","55","56","57","58","59","60","61","62","63","64","65","66","67","68","69","70","71","72","73","74","75","76","77","78","79","80","81","82","83","84","85","86","87","88","89","90","91","92","93","94","95","96","97","98","99"],"labels":["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25","26","27","28","30","31","32","33","34","35","36","37","38","39","40","41","42","43","44","45","46","47","48","49","50","51","52","53","54","55","56","57","58","59","60","61","62","63","64","65","66","67","68","69","70","71","72","73","74","75","76","77","78","79","80","81","82","83","84","85","86","87","88","89","90","91","92","93","94","95","96","97","98","99"],"name":"","parents":
```

```
["","","","","","","","","","","","","","","","","","","","","","","",
"","","","","","","","","","","","","","","","","","","","","","","","
","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","",
"","","","","",""],"type":"treemap","values":
[97,87,77,114,112,108,138,125,75,74,51,40,23,6,5,79,48,45,43,42,33,34,
21,14,11,7,8,12,15,4,3,6,2,3,7,8,17,8,10,8,12,10,13,5,8,17,8,63,17,23,
29,3,35,7,8,13,49,7,7,15,18,12,9,14,9,28,7,15,20,13,18,10,21,19,17,15,
16,8,17,12,14,41,11,11,21,15,7,27,32,9,10,9,11,13,10,10,6,15,13]}],"la
yout":{"legend":{"tracegroupgap":0},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

5.7.- Evaluación de calidad

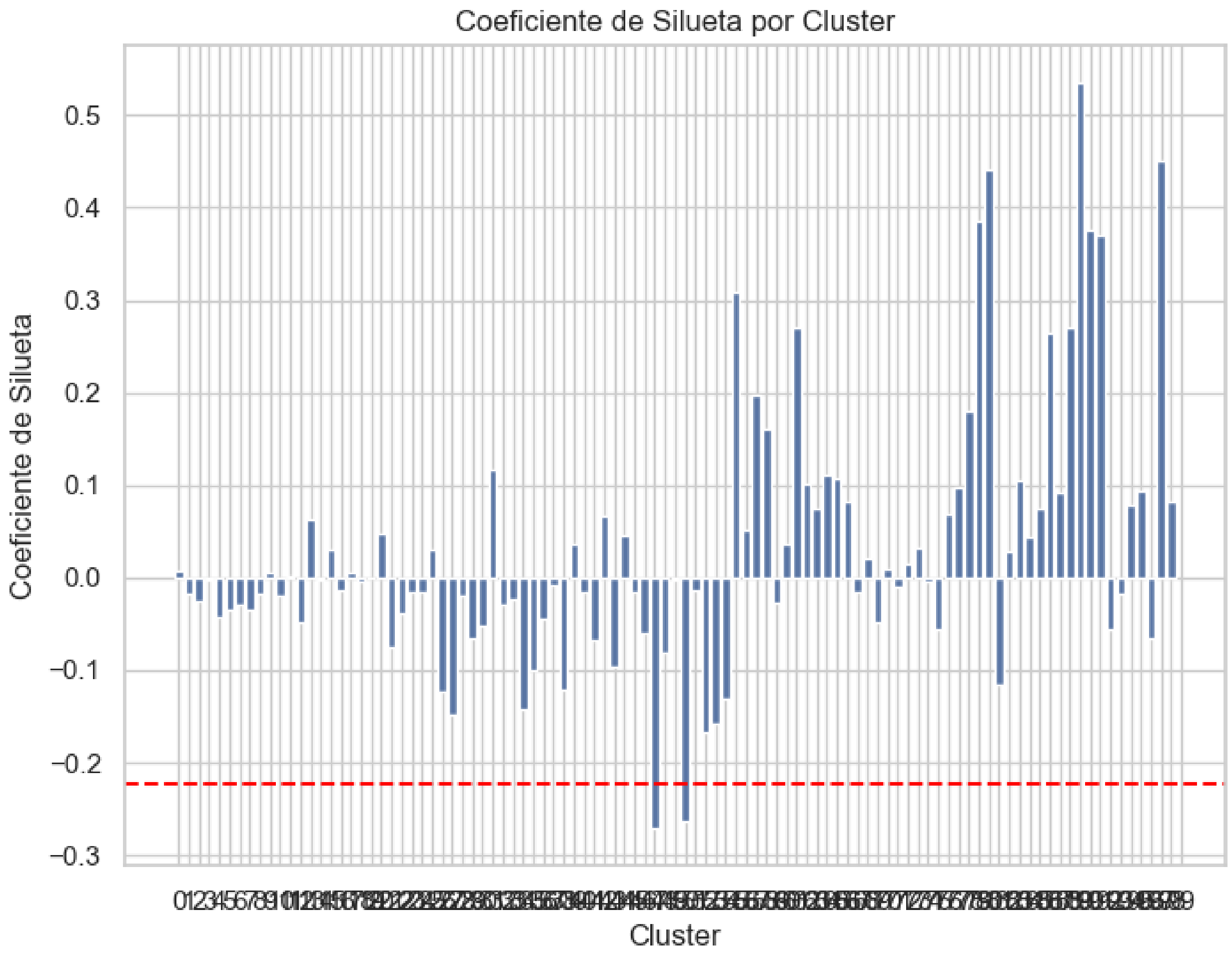
```python
clustering_df = pd.read_csv('clustering_df_reducido.csv')

# Indice de silueta
silhouette_avg = silhouette_score(clustering_df,
optics_pipeline.named_steps['optics'].labels_)
print("El índice de silueta promedio es:", silhouette_avg)
```

```python
# Indice de Davies-Bouldin
davies_bouldin = davies_bouldin_score(clustering_df,
optics_pipeline.named_steps['optics'].labels_)
print("El índice de Davies-Bouldin es:", davies_bouldin)

evaluacion_calidad["optics"] = {"silueta": silhouette_avg,
"davies_bouldin": davies_bouldin}
```

```
El índice de silueta promedio es: -0.22409647554173692
El índice de Davies-Bouldin es: 4.887898025271002
```

```python
# Calculamos el coeficiente de Silueta promedio
silhouette_avg = silhouette_score(clustering_df,
optics_pipeline.named_steps['optics'].labels_)

# Calculamos el coeficiente de Silueta para cada muestra
sample_silhouette_values = silhouette_samples(clustering_df,
optics_pipeline.named_steps['optics'].labels_)

# Creamos una lista para almacenar los coeficientes de Silueta
promedio por cluster
cluster_silhouette_avg = []

list_siluetes = []
# Creamos un bucle para obtener el coeficiente de Silueta para cada
cluster
for cluster in
range(max(optics_pipeline.named_steps['optics'].labels_)+1):
    # Obtenemos las muestras pertenecientes al cluster actual
    cluster_silhouette_values =
sample_silhouette_values[optics_pipeline.named_steps['optics'].labels_
== cluster]

    # Calculamos el coeficiente de Silueta promedio para el cluster
actual
    avg_silhouette = np.mean(cluster_silhouette_values)

    cluster_dict = {'category': cluster, 'value': avg_silhouette}
    list_siluetes.append(cluster_dict)

    # Almacenamos el coeficiente de Silueta promedio en la lista
    cluster_silhouette_avg.append(avg_silhouette)

cluster_siluetes["optics"] = list_siluetes

# Creamos el gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(range(len(cluster_silhouette_avg)-1),
cluster_silhouette_avg[1:], color="#5975A4")
```

```
plt.title('Coeficiente de Silueta por Cluster')
plt.xlabel('Cluster')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range(len(cluster_silhouette_avg)))
plt.axhline(y=silhouette_avg, color="red", linestyle="--")  # Línea
para indicar el valor promedio global de Silueta
plt.show()
```



5.8.- Save Models

```
joblib.dump(optics_pipeline,
'resultados/trained_models/optics.joblib')

['resultados/trained_models/optics.joblib']
```

6.- DBSCAN (Basado en el concepto de "Densidad")

6.1.- Definición de la pipeline para el DBSCAN

```python
clustering_df = pd.read_csv('clustering_df_reducido.csv')

columnas_deseadas = ['MentHlth', 'PhysHlth', 'BMI']
clustering_df = clustering_df.loc[:, columnas_deseadas]

dbscan_pipeline = Pipeline([
    ('MMS', MinMaxScaler()),
    ('dbscan', DBSCAN(eps=0.12, min_samples=2))
])
```

6.2.- Entrenamos el modelo con la pipeline y los datos del df

```python
dbscan_pipeline.fit(clustering_df)
```

```
Pipeline(steps=[('MMS', MinMaxScaler()),
                ('dbscan', DBSCAN(eps=0.12, min_samples=2))])
```

6.3.- Obtenemos las etiquetas (Cluster al que pertenece) para cada dato

```python
labels = dbscan_pipeline.fit_predict(clustering_df)
```

6.4.- Agregar las etiquetas de clusters y de clasificacion al conjunto de datos

```python
clustering_df["cluster_label"] = labels
clustering_df["diabetic_label"] = df_sampled["Diabetes_012"]

clustering_df.head()
clustering_df.to_csv('resultados/df_clusters/dbscan.csv')

clustering_df.head()
```

```
   MentHlth  PhysHlth  BMI  cluster_label  diabetic_label
0         0         0   33              0               0
1        14         0   33              0               0
2         0         0   25              0               0
3         0         0   26              0               0
4         0         1   15              0               0
```

6.5.- Número de clusters encontrados por el algoritmo

```python
# Miramos el número de clusters que el algoritmo ha encontrado
clusters = unique(labels)
print(f"Numero de clusters: {len(clusters) - 1 }")
```

```
# Numero de outliers
outlayers = len(clustering_df[clustering_df['cluster_label'] == -1])
print(f"Numero de outlayers: {outlayers}")
```

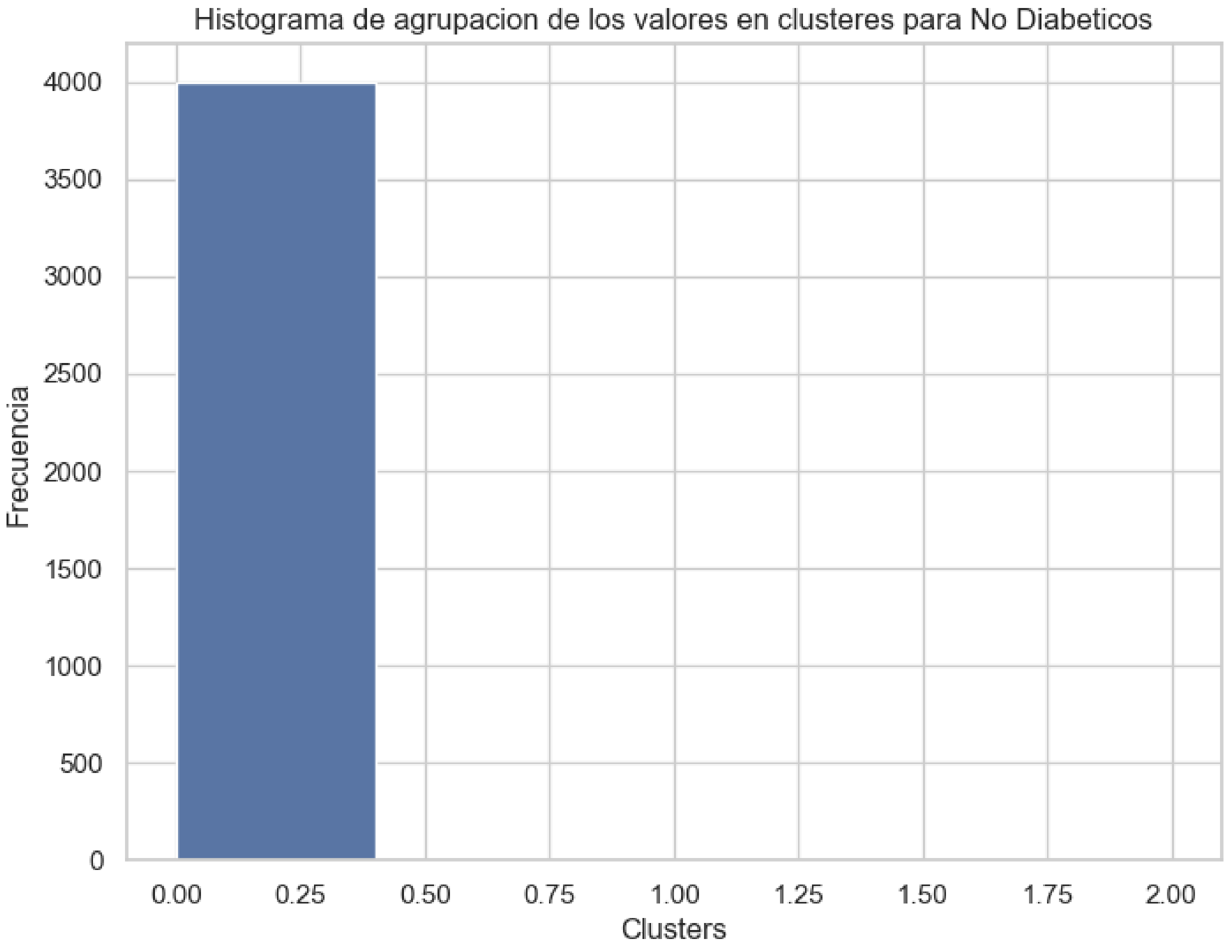
```
Numero de clusters: 4
Numero de outlayers: 15
```

```
num_elementos_cluster_label =
len(clustering_df[clustering_df['cluster_label'] == 2])

print(f"Número de elementos en el cluster con etiqueta 1:
{num_elementos_cluster_label}")
```

```
Número de elementos en el cluster con etiqueta 1: 7
```

6.6.- Resumen de pertenencia de clusters por grupo de diabetes

```
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 0]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 0]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```

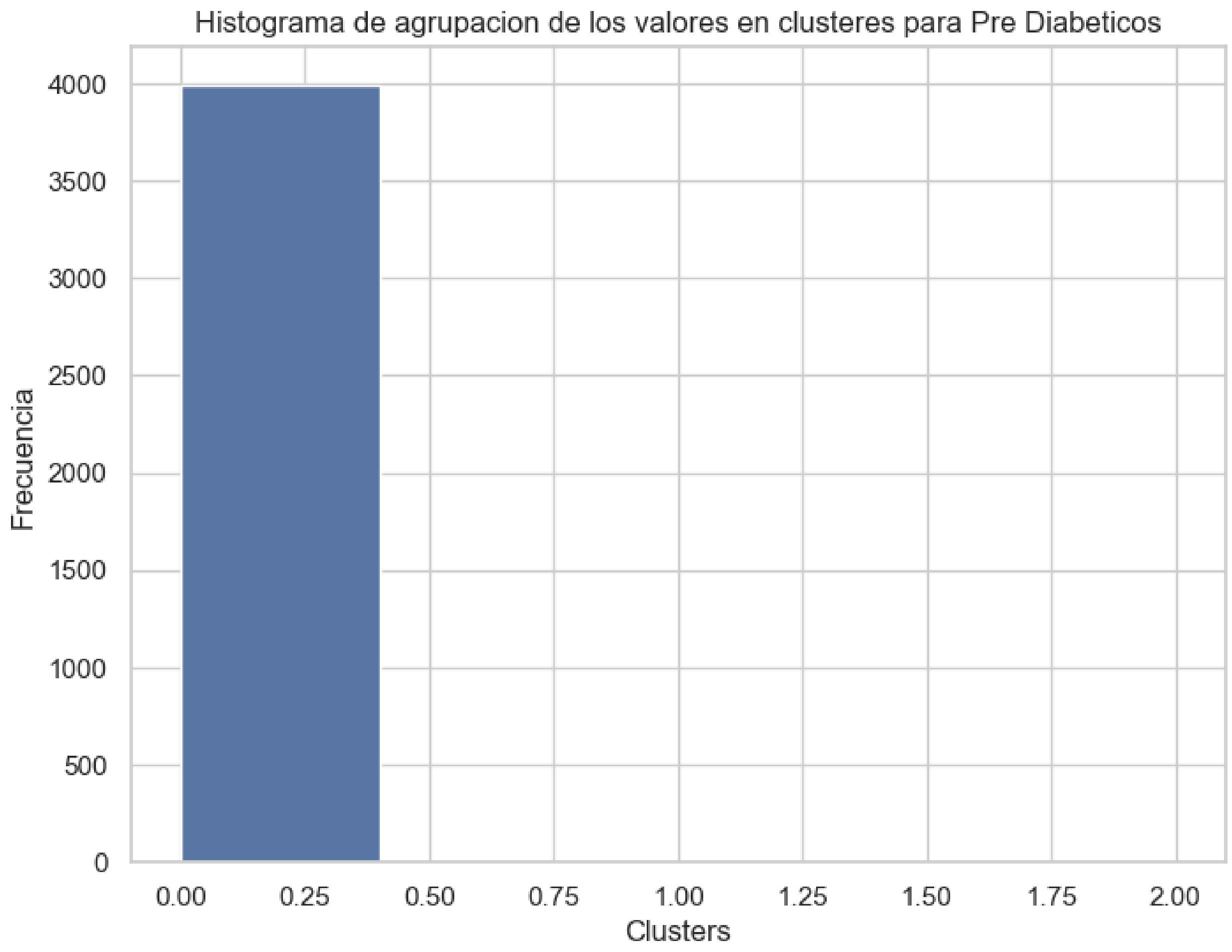


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"branchvalues":"total","domain":{"x":[0,1],"y":
[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%
{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2"],"labels":
["0","1","2"],"name":"","parents":
["","",""],"type":"treemap","values":[3994,1,1]}],"layout":{"legend":
{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
```

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
```

```
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
```

```
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}

# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
Pre Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



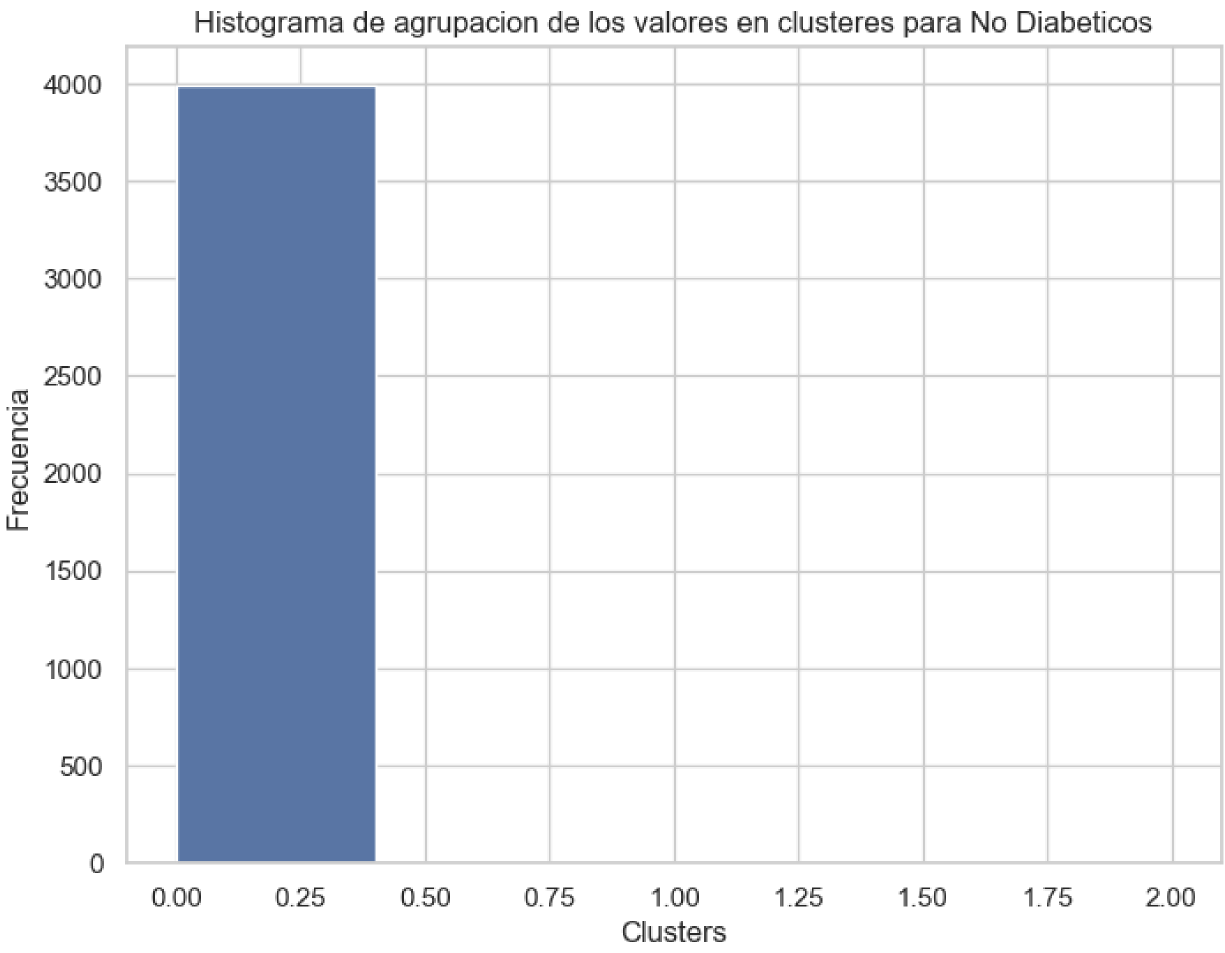
```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2"],"labels":["0","1","2"],"name":"","parents":["","",""],"type":"treemap","values":[3988,3,5]}],"layout":{"legend":{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
```

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
```

```
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
```

```
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

```python
# Miramos el histograma de cada categoria en detalle
data = clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -1]['cluster_label'] # Quitamos los
outliers
plt.figure(figsize=(8, 6))
plt.hist(data, color="#5975A4", bins=len(clusters))
plt.title("Histograma de agrupacion de los valores en clusteres para
No Diabeticos")
plt.xlabel("Clusters")
plt.ylabel("Frecuencia")
plt.show()

# Visualización de los valores de cluster_label
ax= px.treemap(clustering_df[clustering_df['diabetic_label'] == 1]
[clustering_df['cluster_label'] != -
1],path=['cluster_label'],title="Distribución de pacientes en
clusteres")
ax.show()
```



```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"branchvalues":"total","domain":{"x":[0,1],"y":
[0,1]},"hovertemplate":"labels=%{label}<br>count=%{value}<br>parent=%
{parent}<br>id=%{id}<extra></extra>","ids":["0","1","2"],"labels":
["0","1","2"],"name":"","parents":
["","",""],"type":"treemap","values":[3988,3,5]}],"layout":{"legend":
{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
```

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
```

```
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
```

```
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribución de pacientes en clusteres"}}}
```

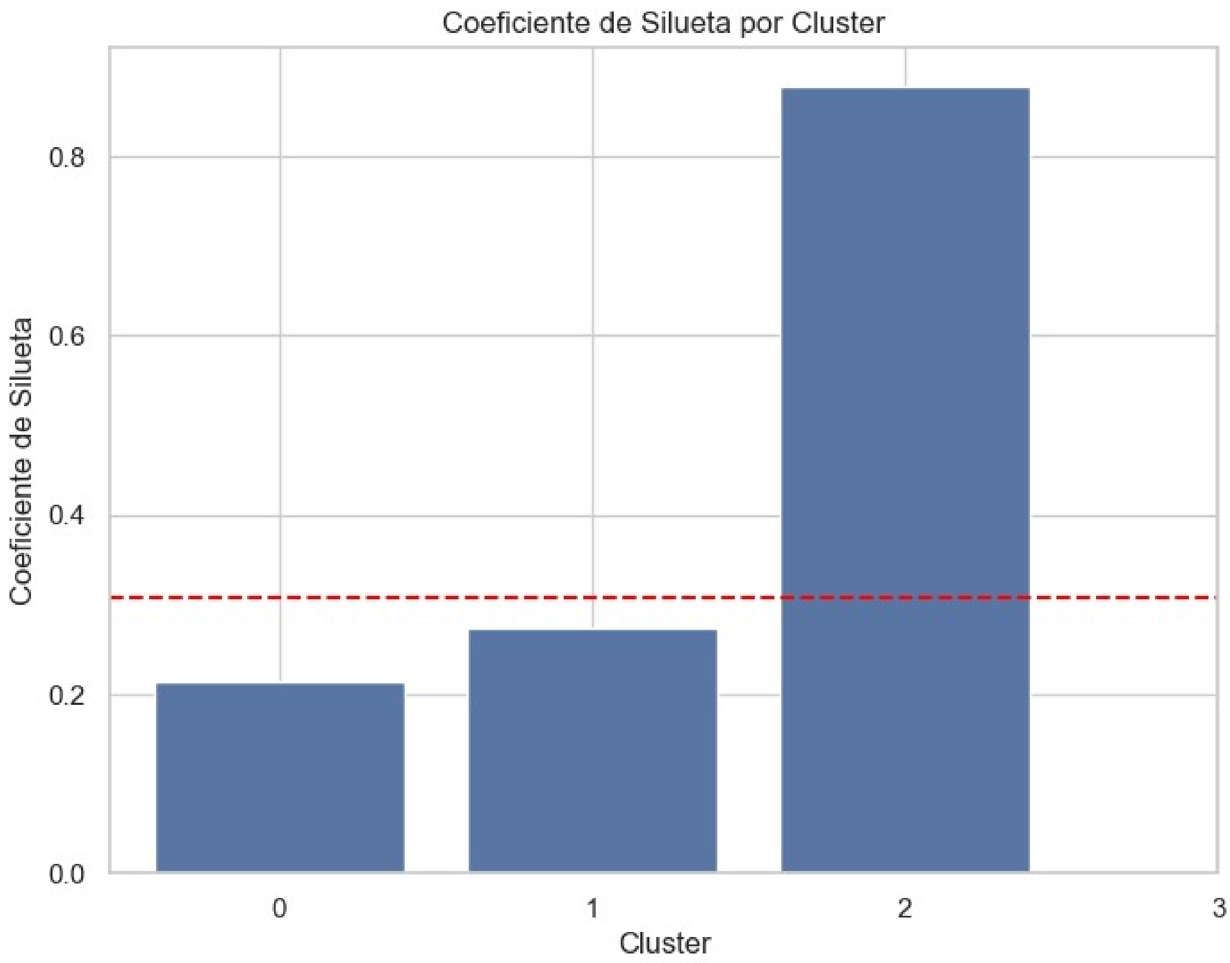
6.9.- Evaluación de calidad

```
clustering_df = pd.read_csv('clustering_df_reducido.csv')

# Indice de silueta
silhouette_avg = silhouette_score(clustering_df,
dbscan_pipeline.named_steps['dbscan'].labels_)
print("El índice de silueta promedio es:", silhouette_avg)

# Indice de Davies-Bouldin
davies_bouldin = davies_bouldin_score(clustering_df,
dbscan_pipeline.named_steps['dbscan'].labels_)
print("El índice de Davies-Bouldin es:", davies_bouldin)

evaluacion_calidad["dbscan"] = {"silueta": silhouette_avg,
"davies_bouldin": davies_bouldin}
```

```
El índice de silueta promedio es: 0.30716259536814916
El índice de Davies-Bouldin es: 0.9247490139934845
```

```
# Calculamos el coeficiente de Silueta promedio
silhouette_avg = silhouette_score(clustering_df,
dbscan_pipeline.named_steps['dbscan'].labels_)

# Calculamos el coeficiente de Silueta para cada muestra
sample_silhouette_values = silhouette_samples(clustering_df,
dbscan_pipeline.named_steps['dbscan'].labels_)

# Creamos una lista para almacenar los coeficientes de Silueta
promedio por cluster
cluster_silhouette_avg = []

list_siluetes = []
```

```
# Creamos un bucle para obtener el coeficiente de Silueta para cada
cluster
for cluster in
range(max(dbscan_pipeline.named_steps['dbscan'].labels_)+1):
    # Obtenemos las muestras pertenecientes al cluster actual
    cluster_silhouette_values =
sample_silhouette_values[dbscan_pipeline.named_steps['dbscan'].labels_
== cluster]

    # Calculamos el coeficiente de Silueta promedio para el cluster
actual
    avg_silhouette = np.mean(cluster_silhouette_values)

    cluster_dict = {'category': cluster, 'value': avg_silhouette}
    list_siluetes.append(cluster_dict)

    # Almacenamos el coeficiente de Silueta promedio en la lista
    cluster_silhouette_avg.append(avg_silhouette)

cluster_siluetes["dbscan"] = list_siluetes

# Creamos el gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(range(len(cluster_silhouette_avg)-1),
cluster_silhouette_avg[1:], color="#5975A4")
plt.title('Coeficiente de Silueta por Cluster')
plt.xlabel('Cluster')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(range(len(cluster_silhouette_avg)))
plt.axhline(y=silhouette_avg, color="red", linestyle="--")  # Línea
para indicar el valor promedio global de Silueta
plt.show()
```



```
with open('resultados/evaluacion_calidad.json', 'w') as archivo_json:
    json.dump(evaluacion_calidad, archivo_json)

with open('resultados/cluster_siluetes.json', 'w') as archivo_json:
    json.dump(cluster_siluetes, archivo_json)
```

6.10.- Save Models

```
joblib.dump(dbscan_pipeline,
'resultados/trained_models/dbscan.joblib')
```

```
['resultados/trained_models/dbscan.joblib']
```