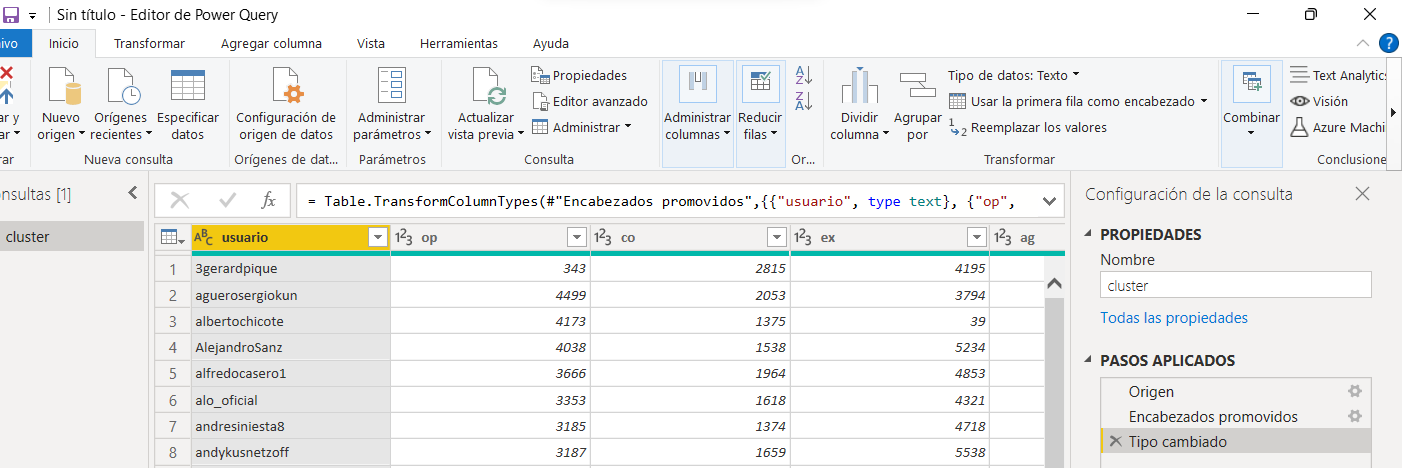
Los registros son agrupados en base a la similitud de sus características(las columnas), en la siguiente presentación comprendemos algunos conceptos necesarios para usar [cluster](https://docs.google.com/presentation/d/1UMfE4HCuYR3jCv2bM44KmOr8C0yIW17r/edit?usp=sharing&ouid=101199890097368716674&rtpof=true&sd=true). Como resultado de ejecutar el algoritmo tendremos:

* Los “centroides” de cada grupo serán unas “coordenadas” de cada uno de los conjuntos que se utilizarán para poder etiquetar nuevas muestras.
* Etiquetas para el conjunto de datos de entrenamiento. Cada etiqueta pertenece a uno de los k grupos formados.

1. Descargar e Instalar [**python**](https://www.python.org/downloads/)
2. Buscar en el menú inicio cmd y ejecutar como administrador y posterior a ello escribir “cd C:\Users\handres\AppData\Local\Programs\Python\PythonX” donde X es la versión de python(ejemplo:39 o 310) donde handres es el nombre de usuario de su computadora. en caso de no existir la ruta verificar la ruta en power bi archivo->opciones y python y en la consola de comandos ejecutar los siguientes comandos (**Nota:** si pip install librería no funciona probar usando python -m pip install librería)
   1. python -m pip install pandas
   2. python -m pip install numpy
   3. python -m pip install matplotlib
   4. python -m pip install seaborn
   5. python -m pip install --pre --extra-index https://pypi.anaconda.org/scipy-wheels-nightly/simple scikit-learn
3. Para conectarnos a una hoja de cálculo de google lo que debemos hacer es sobre la hoja de cálculo debemos ir a archivo y publicar en la web y seleccionamos todo el documento y en el tipo xls. El enlace generado lo copiamos (https://docs.google.com/spreadsheets/d/e/2PACX-1vR6yxL8Qy7ZaGhe4y2pE0WFLtsfd6NqMSq403tFFONVM0R1z3FBWUpyyhY2QDJCXS926Cx\_EAe9vZR7/pub?output=csv)
4. En power Bi vamos a obtener desde una web y pegamos el enlace y nos conectamos para posteriormente transformar datos
5. Verificamos que si power bi por defecto en los pasos aplicados de la parte izquierda a colocado tipo cambiado pulsamos sobre la x para quitar ese paso para mantener nuestro formato decimal
6. Si tenemos filas iniciales en blanco podemos usar la primera fila como encabezado o quitar dichas filas
7. A la columna que tiene el texto le llamaremos contenido y en la pestaña transformar seleccionamos la opción ejecutar script de python

Pegamos el script :

criterios=['op','co','ex'] #Columnas de criterios a analizar

cantidad\_cluster=5 #cantidad de conjuntos

nombre\_columna\_analizar="usuario" #columna a usar o primera columna

nombre\_nueva\_columna="Cluster" # Columna nueva

################################################

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sb

from sklearn.cluster import KMeans

from sklearn.metrics import pairwise\_distances\_argmin\_min

rango\_criterios = np.array(dataset[criterios])

rango\_criterios =np.ascontiguousarray(rango\_criterios, dtype=np.double) #

kmeans = KMeans(n\_clusters=cantidad\_cluster).fit(rango\_criterios)

centroids = kmeans.cluster\_centers\_

closest, \_ = pairwise\_distances\_argmin\_min(kmeans.cluster\_centers\_, rango\_criterios)

closest

total=[centroids]

columnas={}

contadortotal=0

nombrecategoria=''

centroides=pd.DataFrame(centroids)

for n in centroids:

columnas[contadortotal]=n

contadortotal=contadortotal+1

clusters\_finales = pd.DataFrame(columnas)

#grupos = grupos.apply(np.floor)

res=np.transpose(clusters\_finales)

grupos = res

pre\_res = [sub for sub in range(cantidad\_cluster)]

#grupos['Address'] = address

col\_name = list(grupos.columns)

col\_name.insert(0,nombre\_nueva\_columna)

grupos=grupos.reindex(columns = col\_name)

grupos[nombre\_nueva\_columna]=pre\_res

cabeceras=criterios

to\_insert=nombre\_nueva\_columna

int\_list = [to\_insert] + cabeceras

grupos.columns = int\_list

nombrescolumnas=list()

for columna in criterios:

nombrescolumnas.append(columna)

res.columns = [nombrescolumnas]

##########

columna\_de\_coeficientes=list()

for i in dataset[nombre\_columna\_analizar].index:

coeficientes=list()

for criterio in criterios:

coeficientes.append(dataset.loc[i, criterio])

conjunto\_coeficientes= np.array([coeficientes])

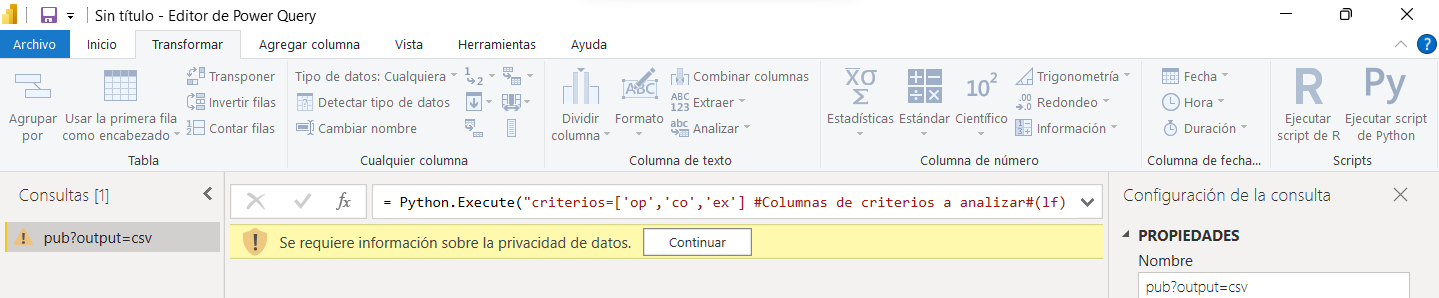
coeficiente\_por\_registro = kmeans.predict(conjunto\_coeficientes)

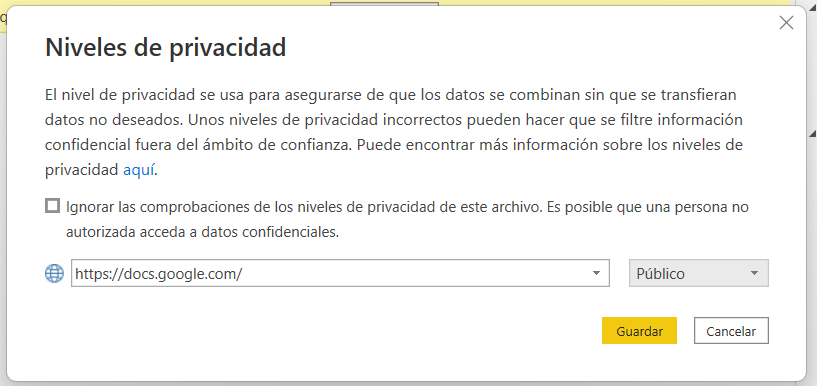
columna\_de\_coeficientes.append(coeficiente\_por\_registro)

dataset[nombre\_nueva\_columna] = pd.DataFrame(columna\_de\_coeficientes,columns=[nombre\_nueva\_columna],dtype ='string')

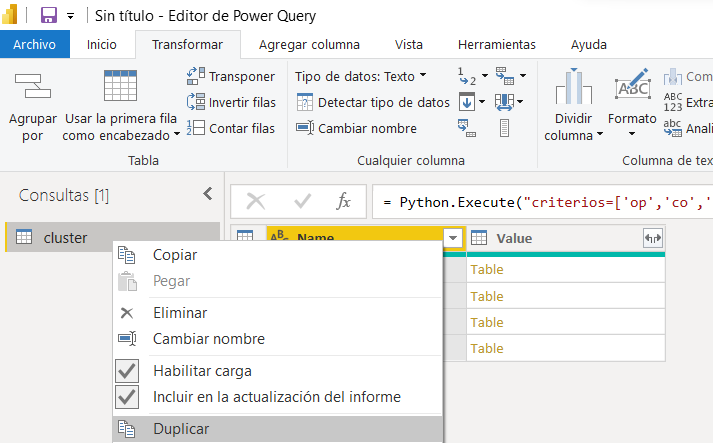
print(dataset)

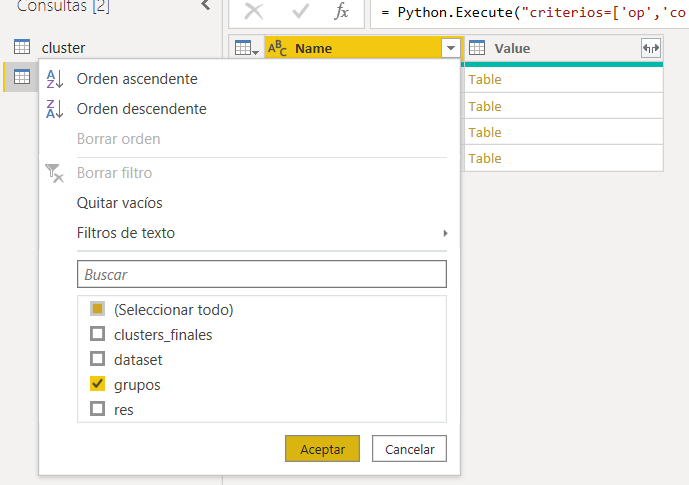
1. Si notifica sobre privacidad de los datos pulsamos sobre continuar y sobre las listas de niveles de publicidad en todas dejamos público y continuar



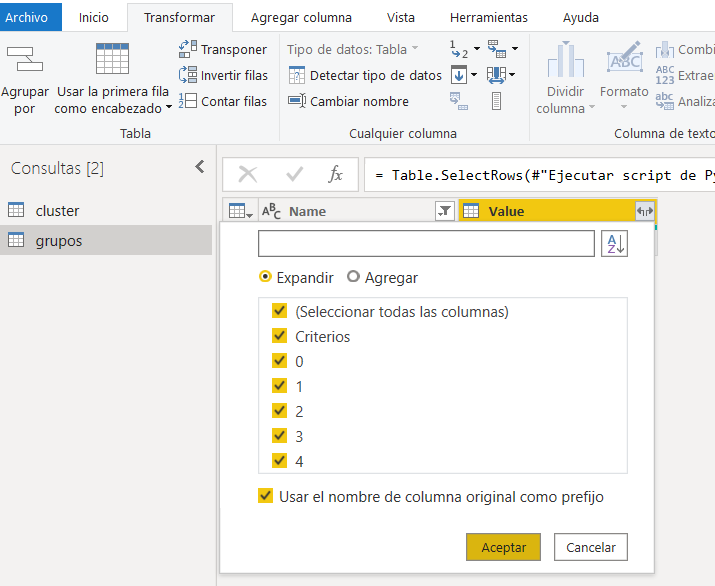
Si nos solicita nivel de privacidad solicitamos público 

1. Una vez ejecutada el código de python verificamos el nombre de nuestra consulta “cluster” y con clic derecho la duplicamos la consulta a la cual llamaremos centroides.

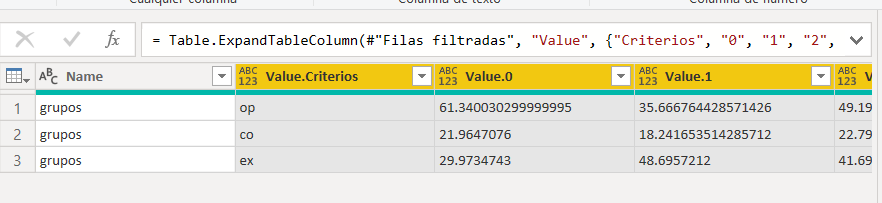


En la consulta centroides filtramos únicamente la columna grupos 

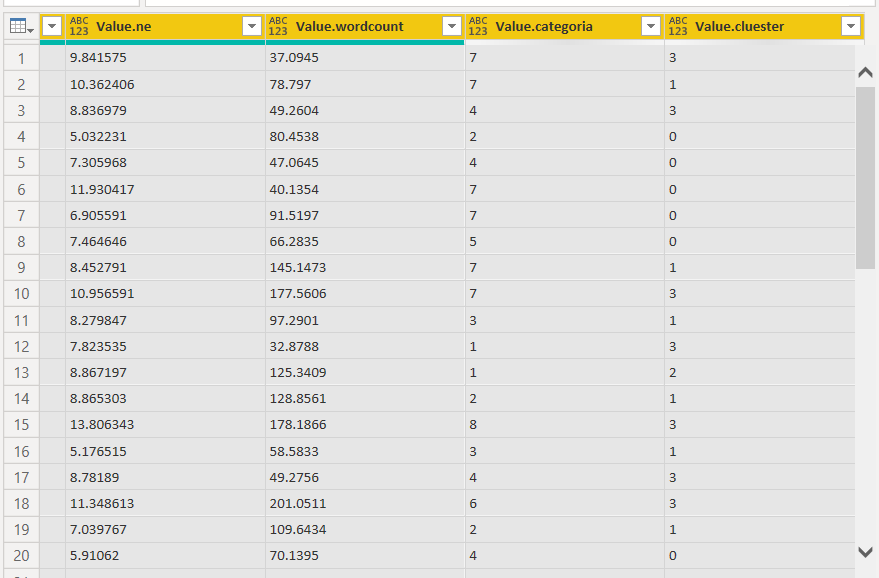
Y desplazamos en el botón a la derecha de value y seleccionamos todas las columnas y aceptar



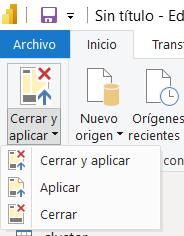
Con esto obtendremos la tabla que nos permite identificar las características de cada cluster según las columnas que seleccionamos y la cantidad de grupos

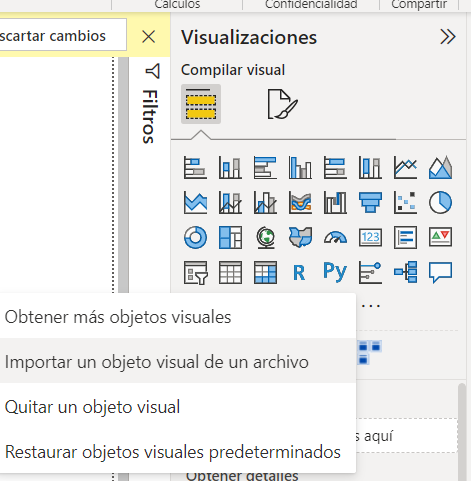


Ahora bien vamos a realizar el mismo proceso con la consulta cluster pero ahora de la lista solo seleccionamos dataset y seleccionamos todos los campos de value y aceptar



Esto nos devolverá la tabla de datos con una columna adicional llamada cluster que identifica a que grupo pertenece cada persona.

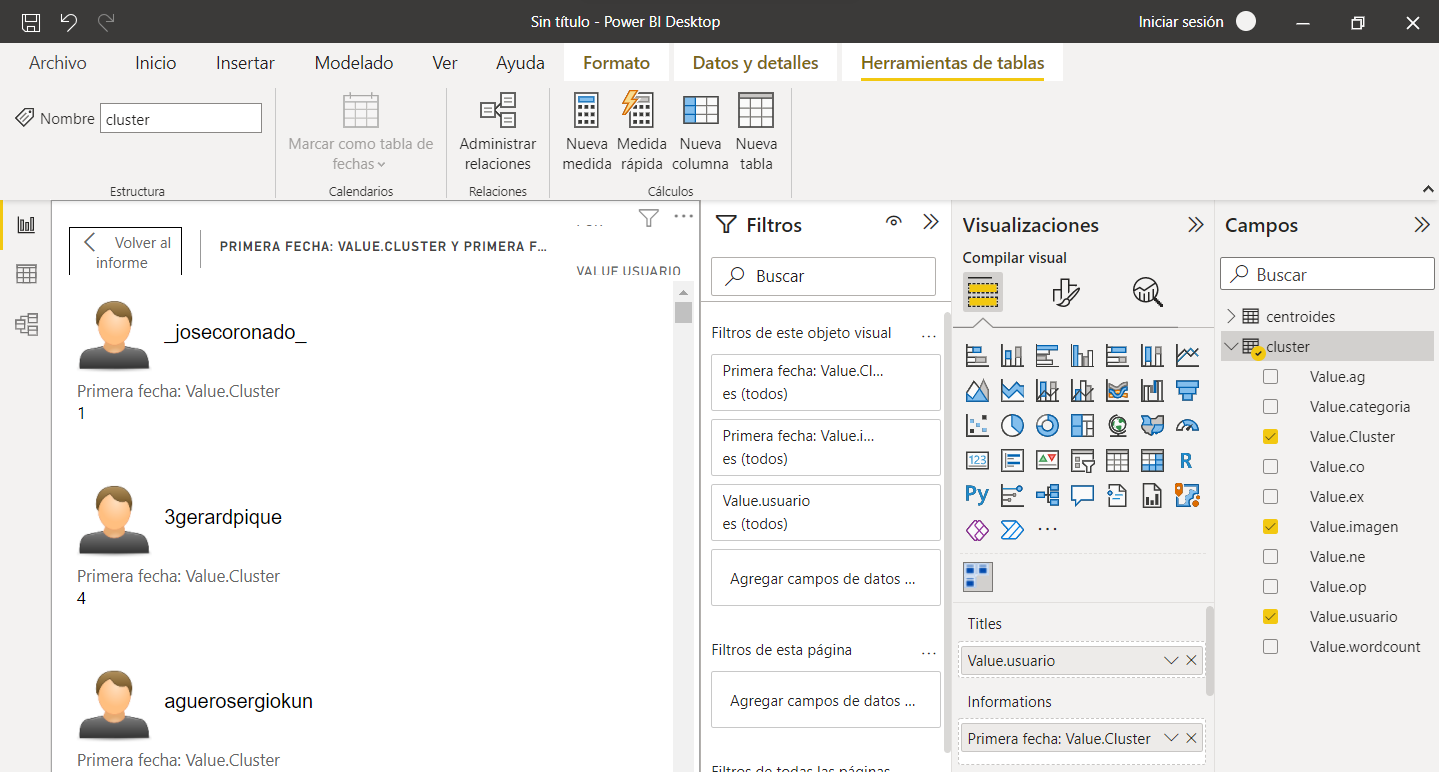
Nota: Si deseamos podemos eliminar la primera columna llamada name de las dos consultas anteriores ya que solo nos servían de identificador temporal

Para visualizar la información descargamos la visualización [Tarjetas de información múltiple](https://appsource.microsoft.com/es/product/power-bi-visuals/WA200002893?src=office&tab=Overview) y en la sección de visualizaciones los podemos importar

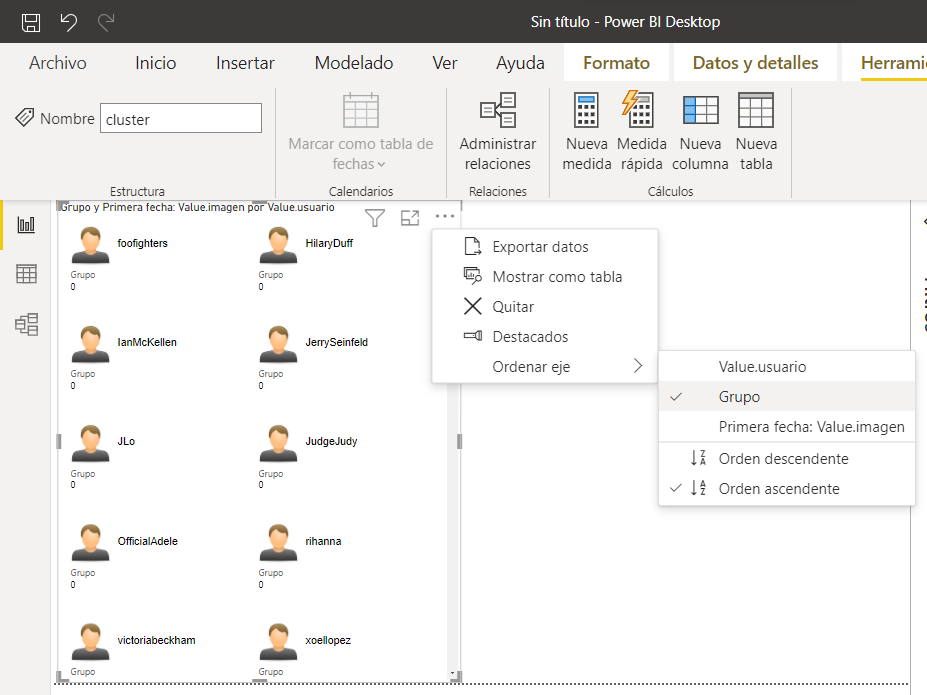
Seleccionaremos la nueva visualización importada y arrastramos en las opciones de la visualización:

* En title : Usuario
* Informations: Cluster
* Image: Imagen

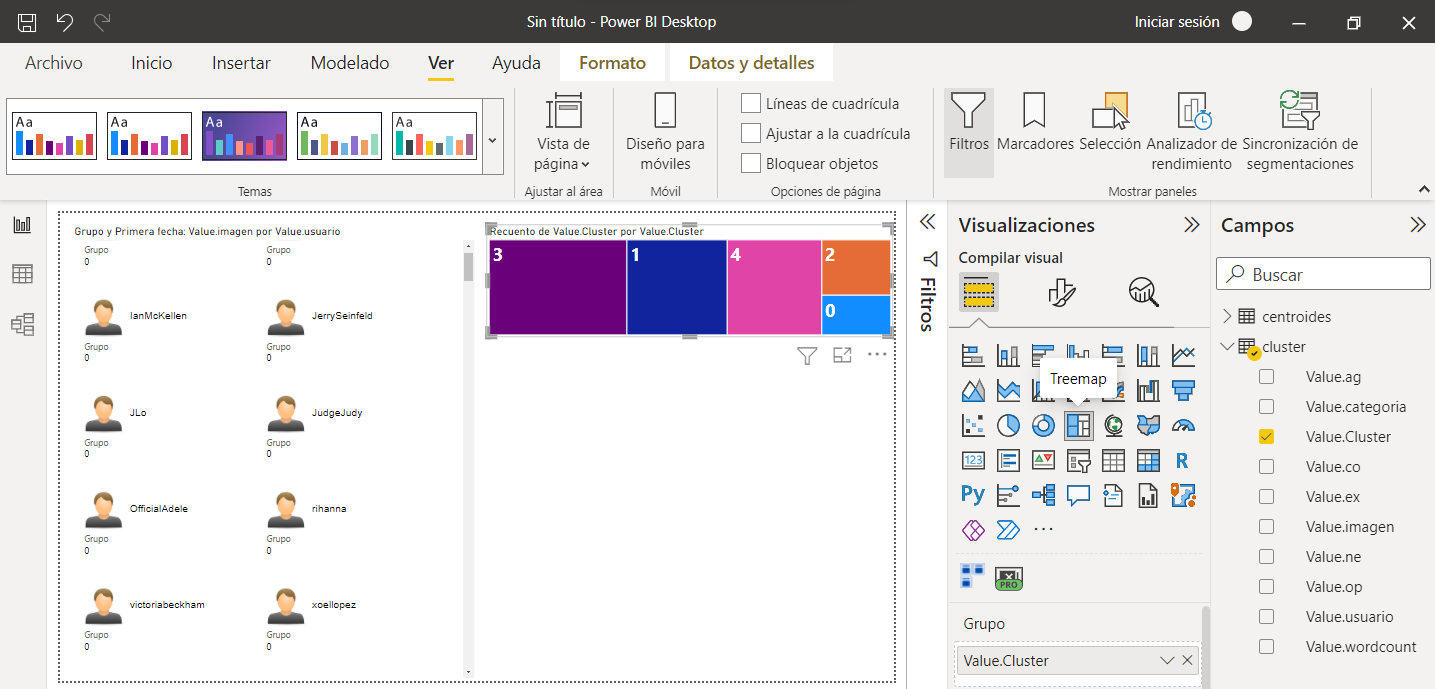
Damos doble clic sobre Primera Fecha Value.Cluster y lo renombramos a Grupo



En las opciones de orden ordenaremos por grupo de forma descendente



Añadiremos una visualización de treemap y activamos de los campos cluster para filtrar los cluster deseados



En las opciones de formato de visualización podremos configurar el tamaño de letra de cada cluster y si deseamos ver la cantidad de elementos en cada cluster podemos activar etiqueta de datos y dar formato al gusto

