

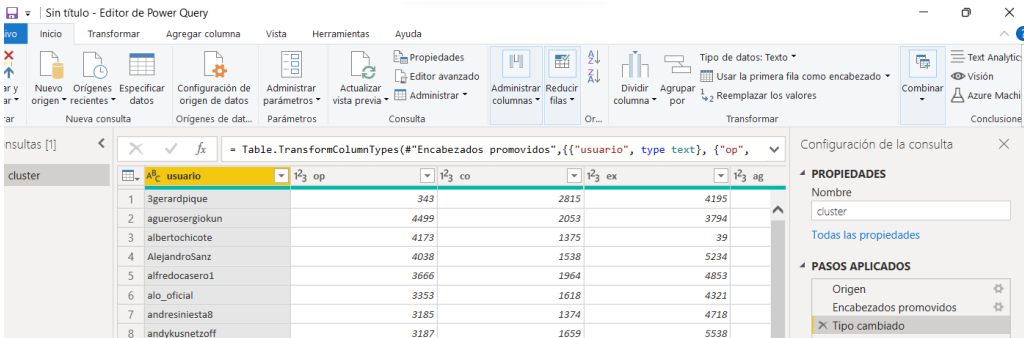
## GUÍA DE ANÁLISIS DE CLUSTER

### LIC. ANDRÉS PAZ

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](#). 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](#)
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)
  - a. python -m pip install pandas
  - b. python -m pip install numpy
  - c. python -m pip install matplotlib
  - d. python -m pip install seaborn
  - e. 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-1vR6yxL8Qy7ZaGhe4y2pE0WFLtsfd6NqMSq403tFFONVM0R1z3FBWUppyhY2QDJCXS926Cx\\_EAe9vZR7/pub?output=csv](https://docs.google.com/spreadsheets/d/e/2PACX-1vR6yxL8Qy7ZaGhe4y2pE0WFLtsfd6NqMSq403tFFONVM0R1z3FBWUppyhY2QDJCXS926Cx_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



	usuario	op	co	ex	ag
1	3gerardpique	343	2815		4195
2	aguerosergiokun	4499	2053		3794
3	albertochicote	4173	1375		39
4	AlejandroSanz	4038	1538		5234
5	alfredocaseri	3666	1964		4853
6	alo_oficial	3353	1618		4321
7	andresiniesta8	3185	1374		4718
8	andykusnetzoff	3187	1659		5538

## GUÍA DE ANÁLISIS DE CLUSTER

### LIC. ANDRÉS PAZ

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 :

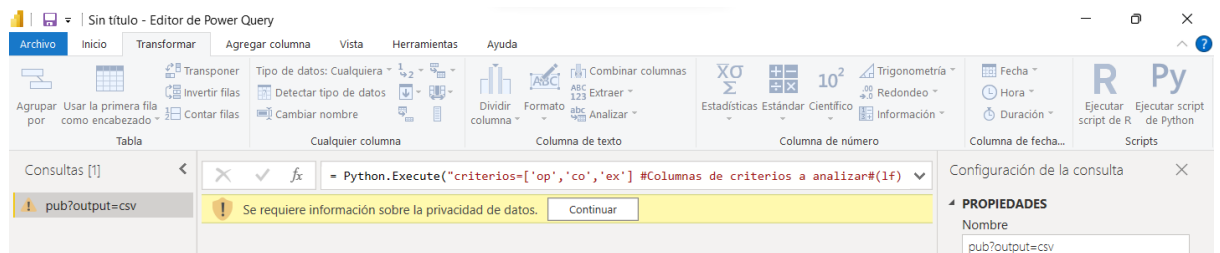
```
criterios=['op','co','ex'] #Columnas de criterios a analizar
cantidad_cluster=5 #cantidad de conjuntos
nombre_columna_analizar="usuario" #columna a usar o primera columna
nombre_nueva_columna="Cluster" # Columna nueva
#####
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import 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()
```

## GUÍA DE ANÁLISIS DE CLUSTER

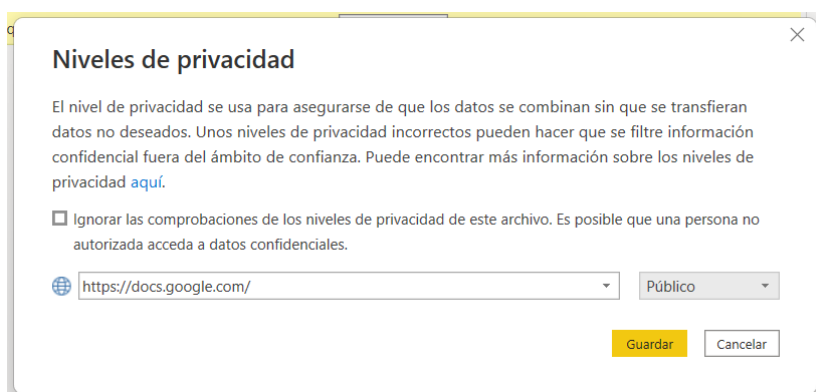
### LIC. ANDRÉS PAZ

```
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)
```

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



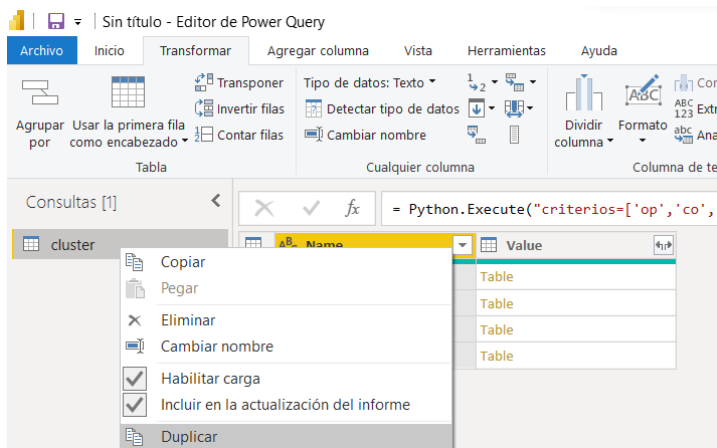
Si nos solicita nivel de privacidad solicitamos público



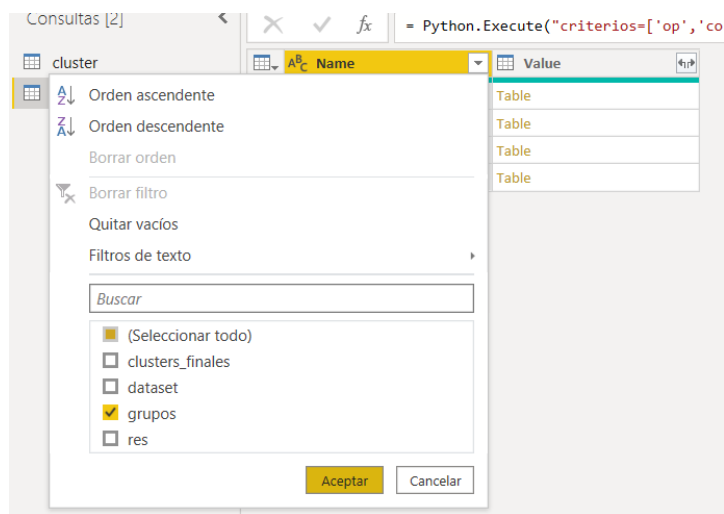
9. Una vez ejecutada el código de python verificamos el nombre de nuestra consulta “cluster” y con clic derecho la duplicamos la consulta a la cual llamaremos centroides.

# GUÍA DE ANÁLISIS DE CLUSTER

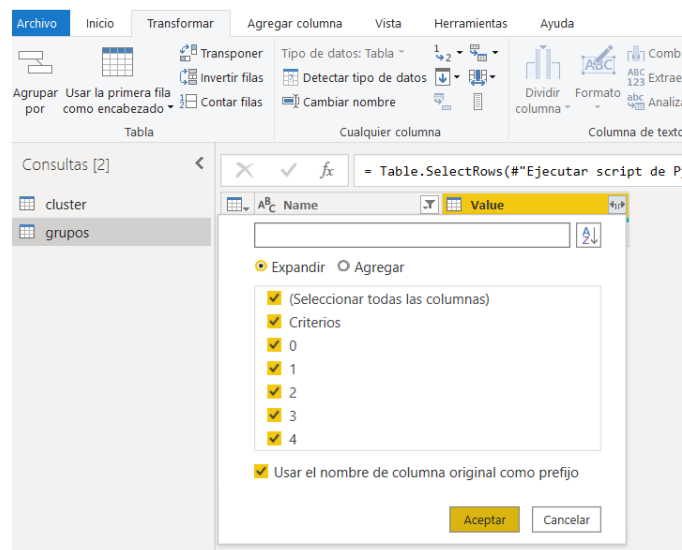
## LIC. ANDRÉS PAZ



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



## GUÍA DE ANÁLISIS DE CLUSTER

### LIC. ANDRÉS PAZ

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

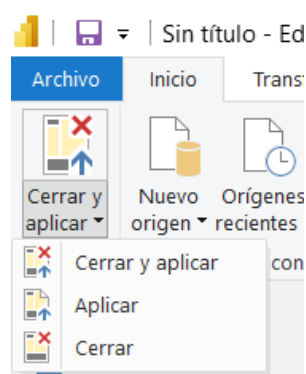
= Table.ExpandTableColumn("#Filas filtradas", "Value", {"Criterios", "0", "1", "2",				
ABC	Name	Value.Criterios	Value.0	Value.1
1	grupos	op	61.340030299999995	35.666764428571426
2	grupos	co	21.9647076	18.241653514285712
3	grupos	ex	29.9734743	48.6957212

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

ABC	Value.ne	Value.wordcount	Value.categoria	Value.clueter
1	9.841575	37.0945	7	3
2	10.362406	78.797	7	1
3	8.836979	49.2604	4	3
4	5.032231	80.4538	2	0
5	7.305968	47.0645	4	0
6	11.930417	40.1354	7	0
7	6.905591	91.5197	7	0
8	7.464646	66.2835	5	0
9	8.452791	145.1473	7	1
10	10.956591	177.5606	7	3
11	8.279847	97.2901	3	1
12	7.823535	32.8788	1	3
13	8.867197	125.3409	1	2
14	8.865303	128.8561	2	1
15	13.806343	178.1866	8	3
16	5.176515	58.5833	3	1
17	8.78189	49.2756	4	3
18	11.348613	201.0511	6	3
19	7.039767	109.6434	2	1
20	5.91062	70.1395	4	0

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

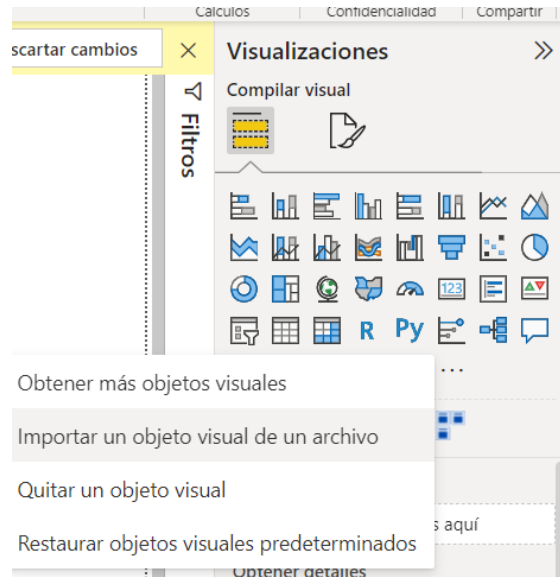
Nota: Si deseamos podemos eliminar la primera columna llamada name de las dos consultas anteriores ya que solo nos servían de identificador temporal



# GUÍA DE ANÁLISIS DE CLUSTER

## LIC. ANDRÉS PAZ

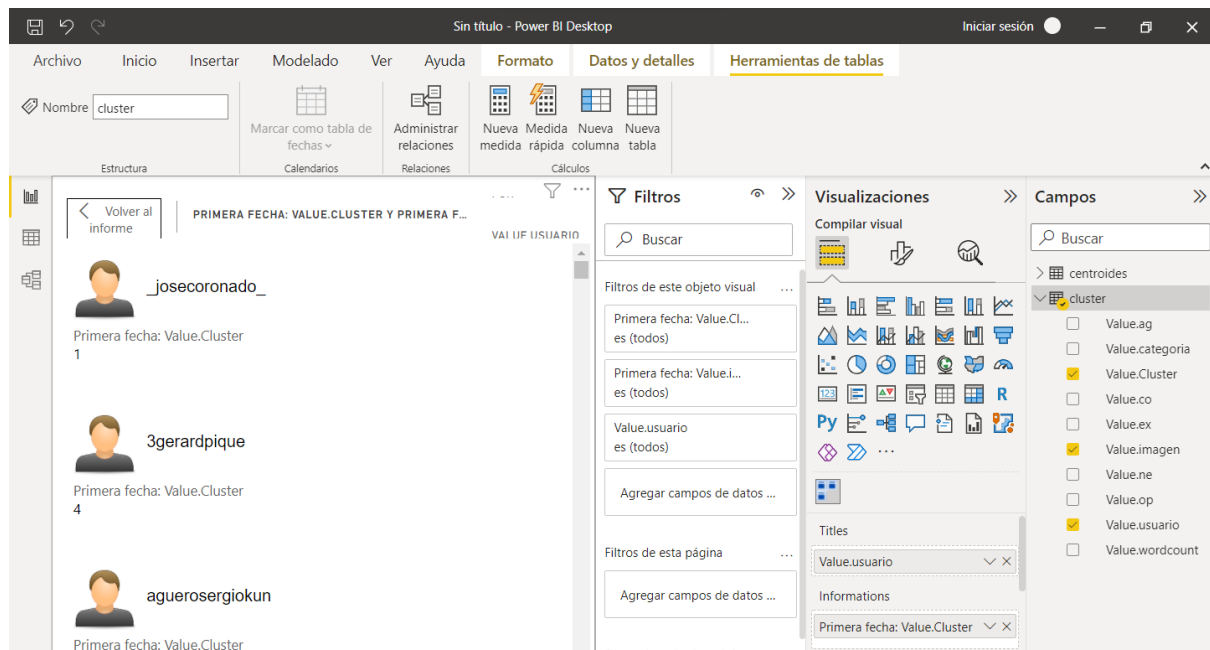
Para visualizar la información descargamos la visualización [Tarjetas de información múltiple](#) y en la sección de visualizaciones los podemos importar



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

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

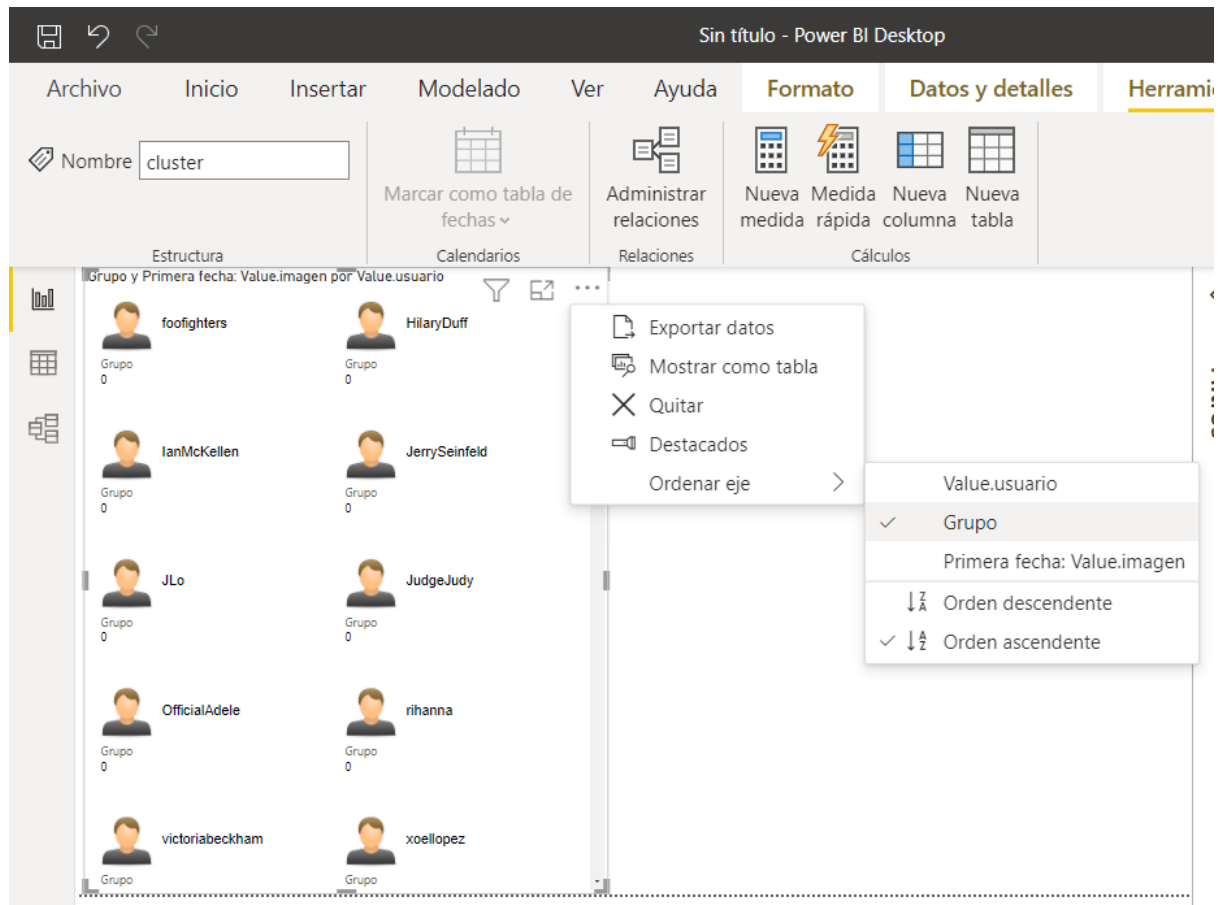
Damos doble clic sobre Primera Fecha Value.Cluster y lo renombramos a Grupo



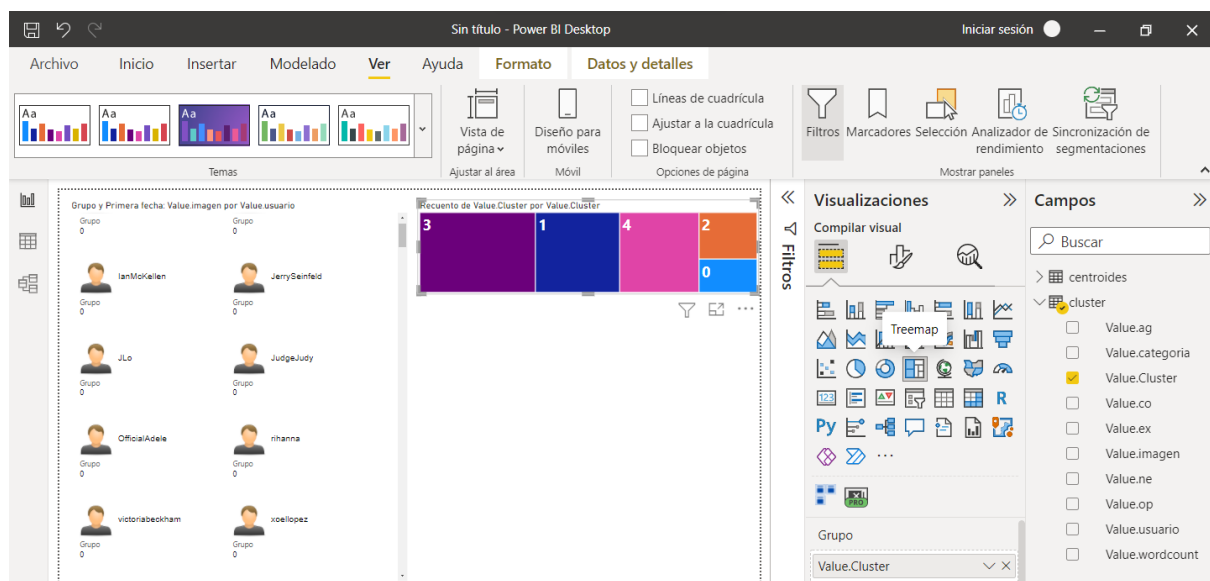
# GUÍA DE ANÁLISIS DE CLUSTER

## LIC. ANDRÉS PAZ

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



## GUÍA DE ANÁLISIS DE CLUSTER

### LIC. ANDRÉS PAZ

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

