

## 12. Integración Databricks con Power BI

Para la realización de este ejercicio es necesario tener la versión gratuita de Microsoft Power BI Desktop que se puede descargar desde <https://powerbi.microsoft.com/es-es/downloads/> y no necesita registro.

### 12.1. Creación del dataframe a partir del fichero SF-fire-calls

Al igual que en el ejemplo 9 Ejecución de Spark DataFrame en DataBriks crearemos el df con la información del fichero csv

#### 12.1.1. Verificamos el fichero

```
%fs ls /databricks-datasets/learning-spark-v2/sf-fire/sf-fire-calls.csv
```

```
%fs head databricks-datasets/learning-spark-v2/sf-fire/sf-fire-calls.csv
```

#### 12.1.2. Indicamos la ruta del fichero para el dataframe

```
from pyspark.sql.types import *
from pyspark.sql.functions import *

sf_fire_file = "/databricks-datasets/learning-spark-v2/sf-fire/sf-fire-calls.csv"
```

#### 12.1.3. Definimos la estructura

```
1 fire_schema = StructType([StructField('CallNumber', IntegerType(), True),
2                               StructField('UnitID', StringType(), True),
3                               StructField('IncidentNumber', IntegerType(), True),
4                               StructField('CallType', StringType(), True),
5                               StructField('CallDate', StringType(), True),
6                               StructField('WatchDate', StringType(), True),
7                               StructField('CallFinalDisposition', StringType(),
8 True),
9                               StructField('AvailableDtTm', StringType(), True),
10                              StructField('Address', StringType(), True),
11                              StructField('City', StringType(), True),
12                              StructField('Zipcode', IntegerType(), True),
13                              StructField('Battalion', StringType(), True),
14                              StructField('StationArea', StringType(), True),
15                              StructField('Box', StringType(), True),
16                              StructField('OriginalPriority', StringType(), True),
17                              StructField('Priority', StringType(), True),
```

```

18         StructField('FinalPriority', IntegerType(), True),
19         StructField('ALSUnit', BooleanType(), True),
20         StructField('CallTypeGroup', StringType(), True),
21         StructField('NumAlarms', IntegerType(), True),
22         StructField('UnitType', StringType(), True),
23         StructField('UnitSequenceInCallDispatch',
IntegerType(), True),
24         StructField('FirePreventionDistrict', StringType(),
True),
25         StructField('SupervisorDistrict', StringType(), True),
26         StructField('Neighborhood', StringType(), True),
27         StructField('Location', StringType(), True),
28         StructField('RowID', StringType(), True),
        StructField('Delay', FloatType(), True)]])

```

#### 12.1.4. Creamos el DF

```
fire_df = spark.read.csv(sf_fire_file, header=True, schema=fire_schema)
```

## 12.2. Utilizar spark para transformar el fichero y crear las dimensiones

### 12.2.1. Crear la columnas de fecha IncidentDate

Adicionalmente, muestro los datos para verificar que se ha realizado correctamente

```

fire_ts_df = (fire_df
              .withColumn("IncidentDate", to_timestamp(col("CallDate"),
"MM/dd/yyyy")) .drop("CallDate"))
fire_ts_df.columns
fire_ts_df.select("IncidentDate").show(5, False)

```

### 12.2.2. Creo la agrupación de la tabla de hechos con la información resumida

Adicionalmente cacheo el df ya que vamos a realizar varias consultas

```

fire_pbi_df=fire_ts_df.groupBy(year(col('IncidentDate')).alias("IncidentDate
Year"), 'CallType', 'Neighborhood', "Zipcode").count()
fire_pbi_df.cache()

```

### 12.2.3. Guardo el df como tabla

En el caso de que la tabla ya exista se puede utilizar %fs rm -r <ruta> para eliminar el fichero

```
fire_pbi_df.write.format("parquet").mode("overwrite").saveAsTable("FSCallsxB
arrioTipoYear")
```

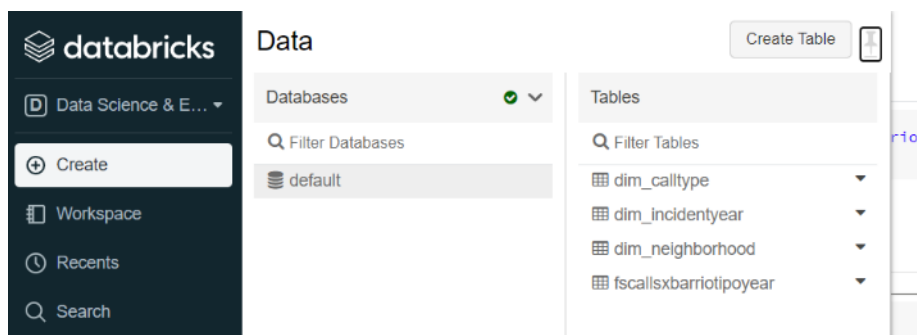
#### 12.2.4. Creamos las dimensiones para filtrar en Power BI

No es necesario ya que se pueden generar desde la herramienta de presentación, pero así mantenemos la arquitectura de análisis de datos más correcta

```
dim_neighborhood_df=fire_pbi_df.select('Neighborhood').distinct()
dim_calltype_df=fire_pbi_df.select('CallType').distinct()
dim_incidentyear_df=fire_pbi_df.select('IncidentDateYear').distinct()

dim_neighborhood_df.write.format("parquet").mode("overwrite").saveAsTable("dim_neighborhood")
dim_calltype_df.write.format("parquet").mode("overwrite").saveAsTable("dim_calltype")
dim_incidentyear_df.write.format("parquet").mode("overwrite").saveAsTable("dim_incidentyear")
```

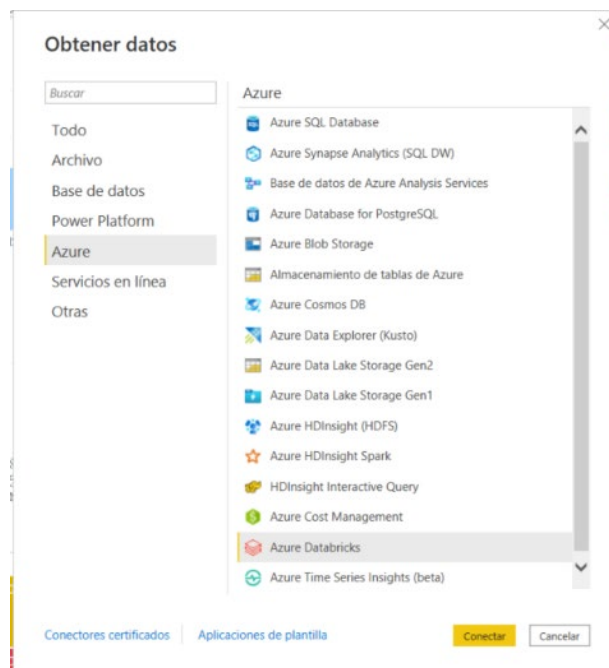
Resultado



### 12.3. Enlazar con Power BI

#### 12.3.1. Indica a Power BI que vamos a utilizar Azure DataBricks

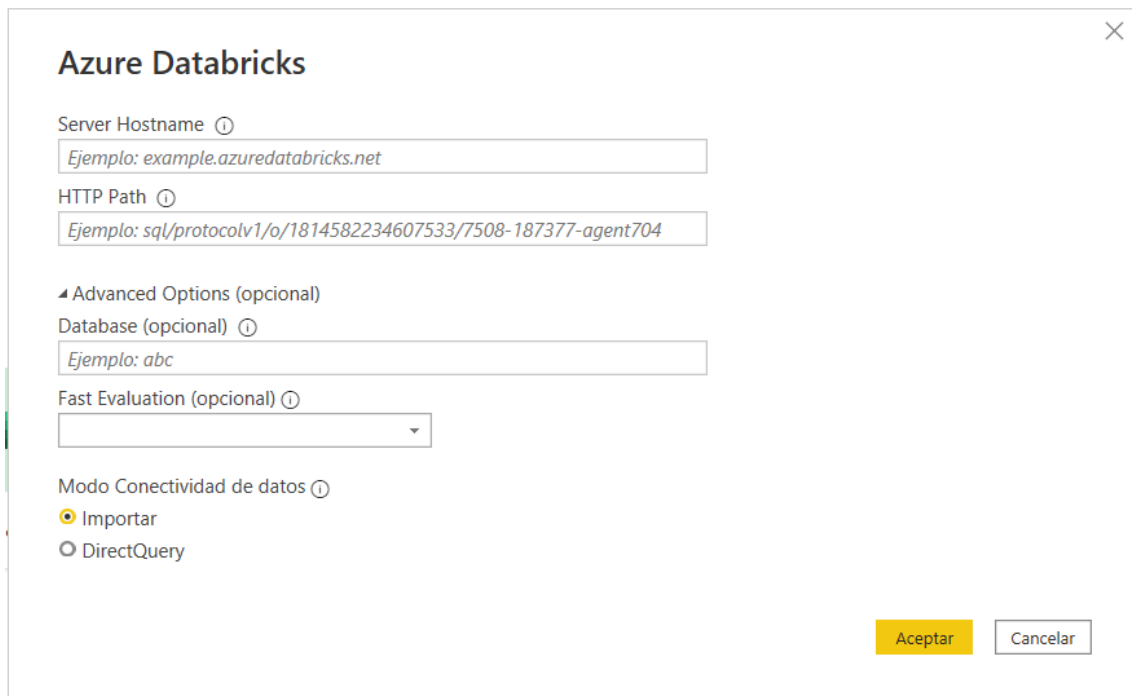
En un fichero de Pbi nuevo, mostrar la ventana de orígenes de datos INICIO>Obtener Datos>Mas..



Seleccionar Azure > Azure DataBricks

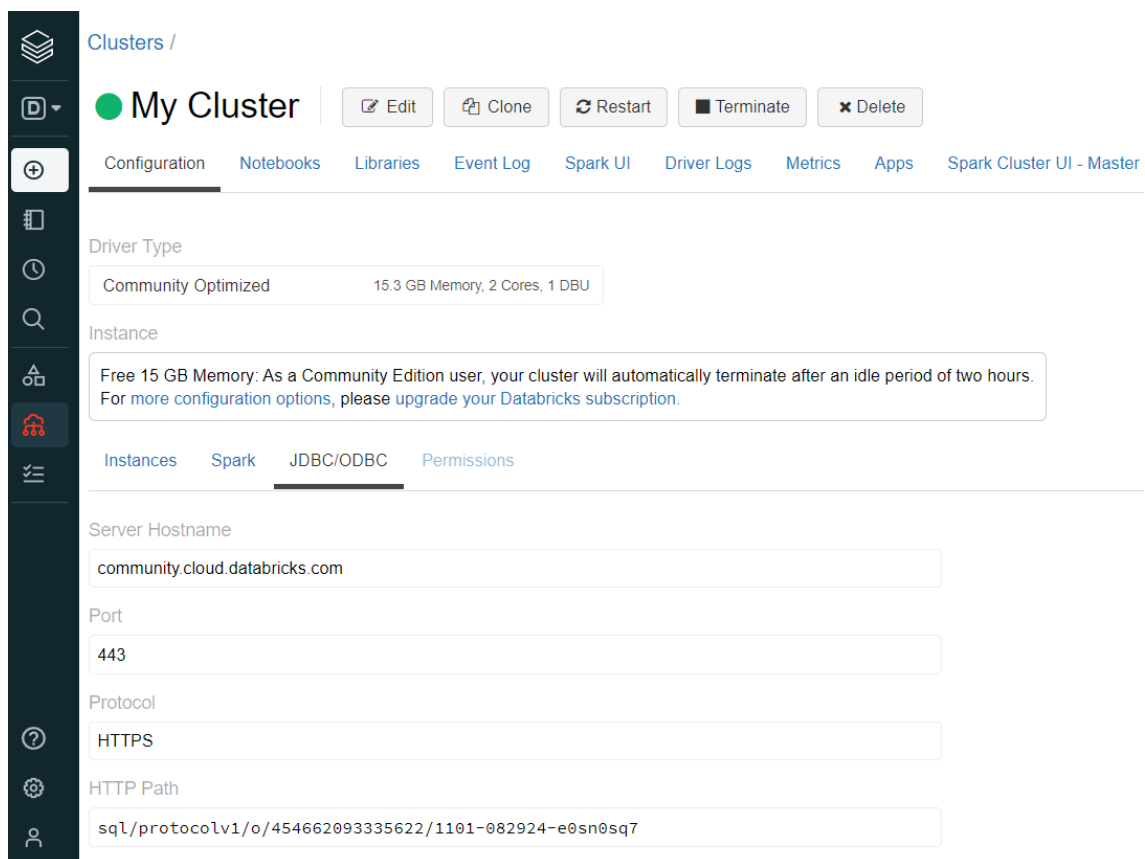
## 12.3.2. Configurar los datos de enlace

Necesitamos rellenar los datos de Server Hostname y HTTP Path.



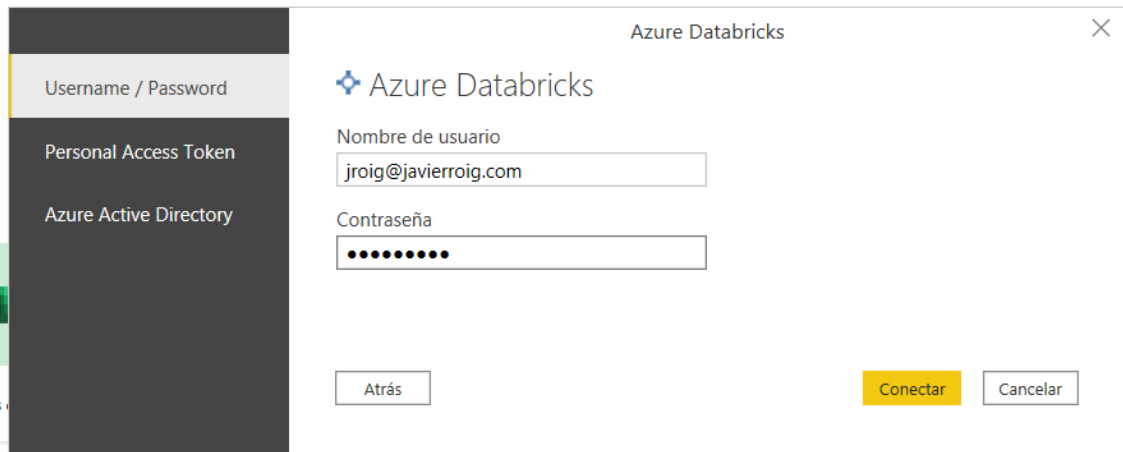
The screenshot shows the 'Azure Databricks' configuration window. It includes fields for 'Server Hostname' (with an example: `example.azure.databricks.net`) and 'HTTP Path' (with an example: `sql/protocolv1/o/1814582234607533/7508-187377-agent704`). There are also 'Advanced Options' for 'Database' (example: `abc`) and 'Fast Evaluation' (a dropdown menu). At the bottom, there are radio buttons for 'Modo Conectividad de datos' with 'Importar' selected and 'DirectQuery' as an option. 'Aceptar' and 'Cancelar' buttons are at the bottom right.

Esa información la tenemos en DataBricks>Compute>Cluster> My Cluster. En la parte de Instance, seleccionar JDBC



The screenshot shows the Databricks Clusters page for 'My Cluster'. The 'Configuration' tab is active. Under 'Driver Type', 'Community Optimized' is selected with '15.3 GB Memory, 2 Cores, 1 DBU'. Under 'Instance', a message states: 'Free 15 GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please upgrade your Databricks subscription.' Below this, the 'JDBC/ODBC' tab is selected. It shows the 'Server Hostname' as `community.cloud.databricks.com`, 'Port' as `443`, 'Protocol' as `HTTPS`, and 'HTTP Path' as `sql/protocolv1/o/454662093335622/1101-082924-e0sn0sq7`.

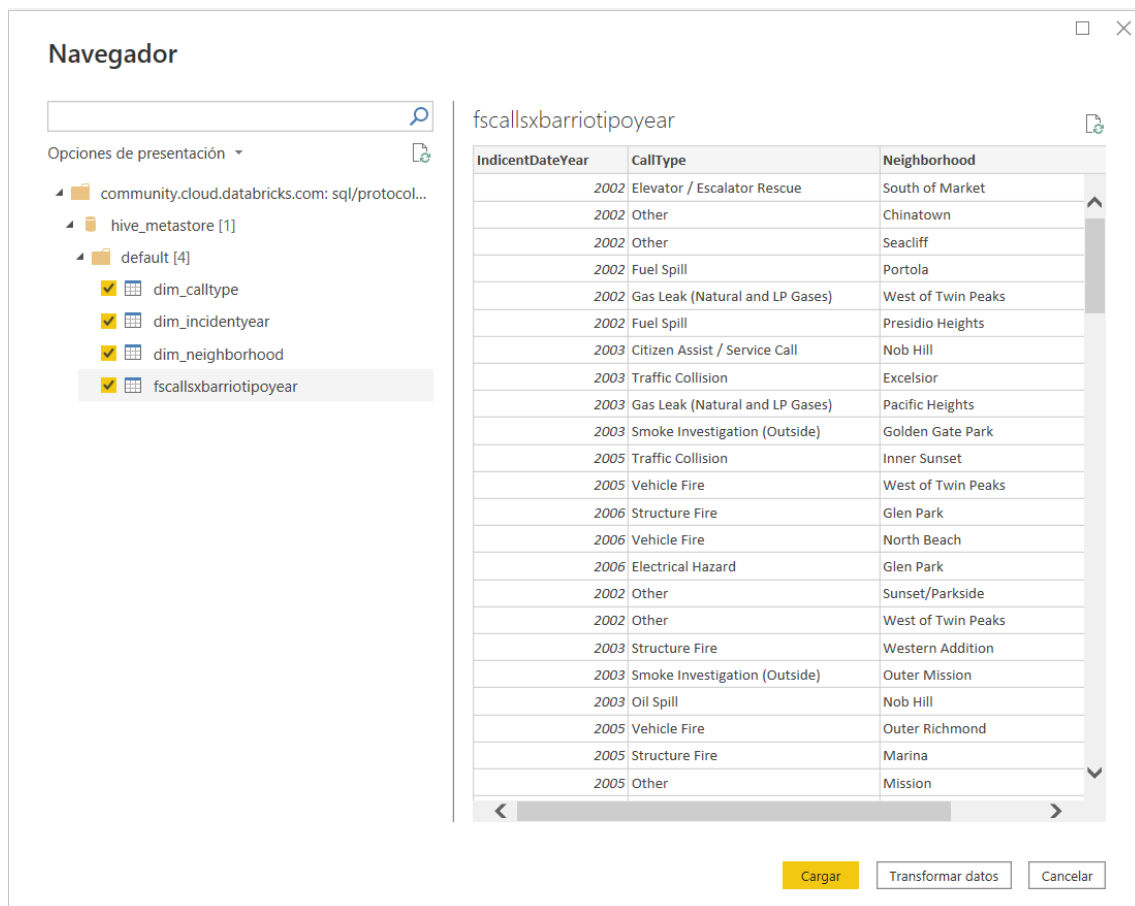
Una vez introducidos los datos de acceso, indicar el usuario y contraseña de DataBricks



Azure Databricks login window. The window has a sidebar on the left with three options: 'Username / Password' (selected), 'Personal Access Token', and 'Azure Active Directory'. The main area is titled 'Azure Databricks' and contains a login form. The form has two input fields: 'Nombre de usuario' (Username) with the value 'jroig@javierroig.com' and 'Contraseña' (Password) with masked characters. Below the fields are three buttons: 'Atrás' (Back), 'Conectar' (Connect), and 'Cancelar' (Cancel).

### 12.3.3. Seleccionar las tablas

Selecciona las tablas y pulsar en Transformar datos para verificar que la información se importa en el formato correcto

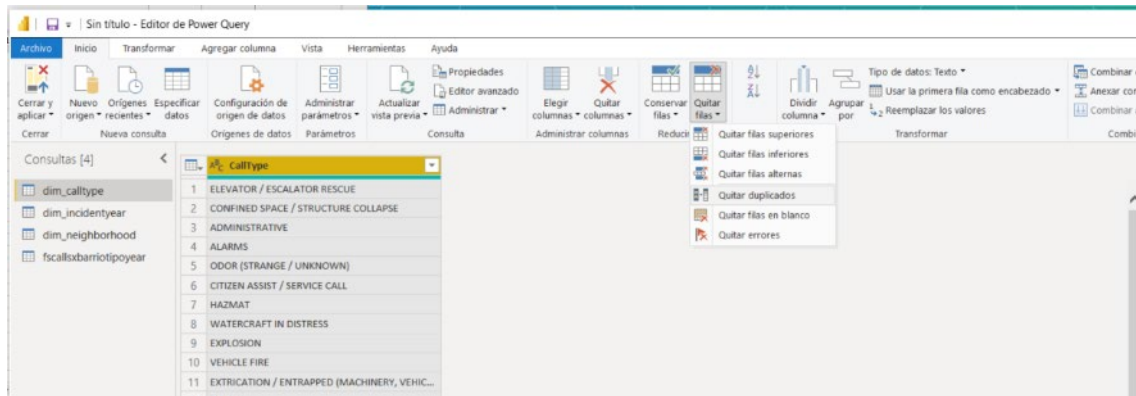
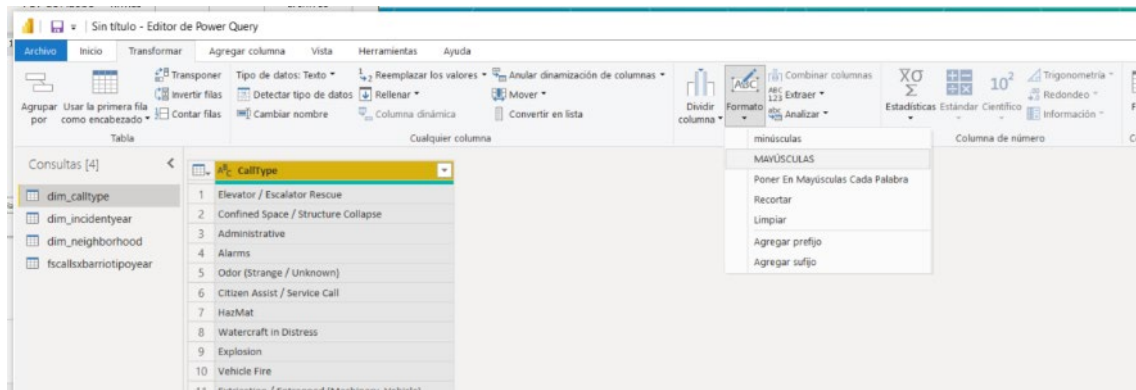


Azure Databricks Navigator window. The window is titled 'Navegador'. On the left, there is a tree view showing the file structure. The tree is expanded to show the 'default' folder, which contains several tables. The table 'fscallsxbarriotipyear' is selected. On the right, the table data is displayed in a table format. The table has three columns: 'IndicentDateYear', 'CallType', and 'Neighborhood'. The data is as follows:

IndicentDateYear	CallType	Neighborhood
2002	Elevator / Escalator Rescue	South of Market
2002	Other	Chinatown
2002	Other	Seacliff
2002	Fuel Spill	Portola
2002	Gas Leak (Natural and LP Gases)	West of Twin Peaks
2002	Fuel Spill	Presidio Heights
2003	Citizen Assist / Service Call	Nob Hill
2003	Traffic Collision	Excelsior
2003	Gas Leak (Natural and LP Gases)	Pacific Heights
2003	Smoke Investigation (Outside)	Golden Gate Park
2005	Traffic Collision	Inner Sunset
2005	Vehicle Fire	West of Twin Peaks
2006	Structure Fire	Glen Park
2006	Vehicle Fire	North Beach
2006	Electrical Hazard	Glen Park
2002	Other	Sunset/Parkside
2002	Other	West of Twin Peaks
2003	Structure Fire	Western Addition
2003	Smoke Investigation (Outside)	Outer Mission
2003	Oil Spill	Nob Hill
2005	Vehicle Fire	Outer Richmond
2005	Structure Fire	Marina
2005	Other	Mission

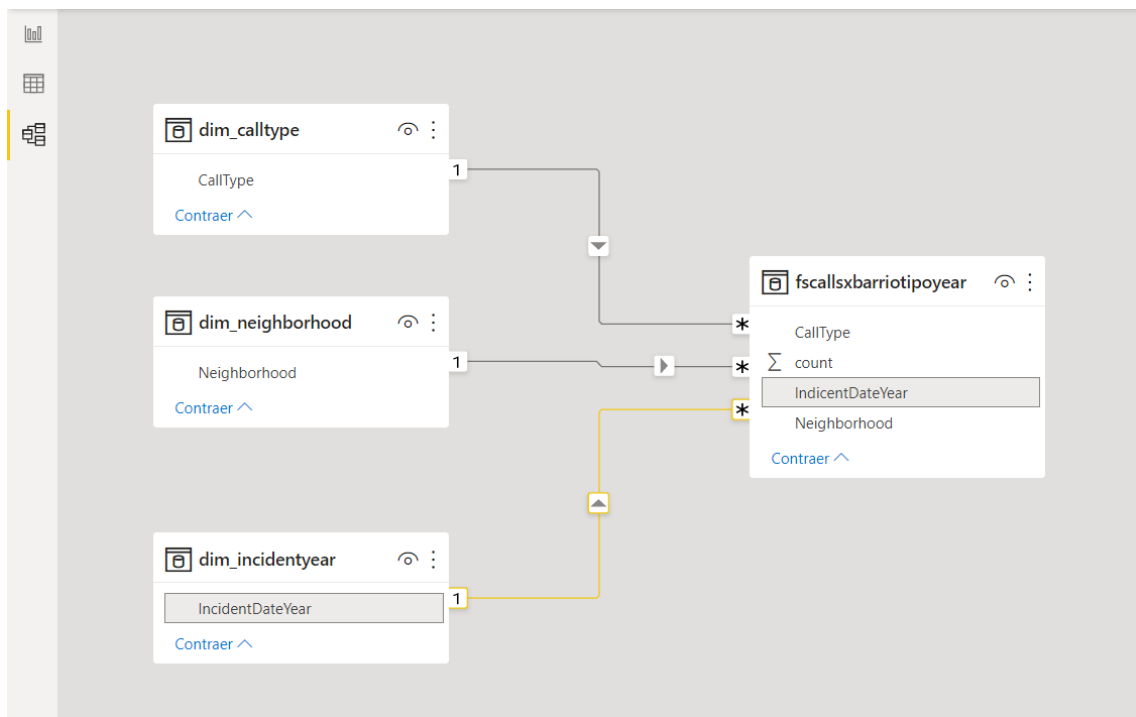
At the bottom of the window, there are three buttons: 'Cargar' (Load), 'Transformar datos' (Transform data), and 'Cancelar' (Cancel).

Por ejemplo, transformamos en mayúsculas y quitamos duplicados la columna CallType de la tabla dim\_CallType y la columna Neighborhood de la tabla dim\_neighborhood. En spark.sql el comando `distinct()` distingue entre mayúsculas y minúsculas en cambio, Power BI no.



Una vez realizadas las transformaciones necesarias pulsar en la pestaña inicio en Cerrar y Aplicar

Accediendo desde la vista Modelo ( y organizando las tablas) debería quedar similar a la imagen. En el caso de que falte alguna relación, se pueden crear desde el botón Administrar relaciones o pinchando y arrastrando un campo sobre otro



## 12.4. Crear una presentación sencilla con PowerBI

Utilizando el campo count en el campo de valor de los diferentes objetos visuales y los de las tablas de dimensiones para filas, columnas o ejes, se pueden crear diferentes visualizaciones de información

