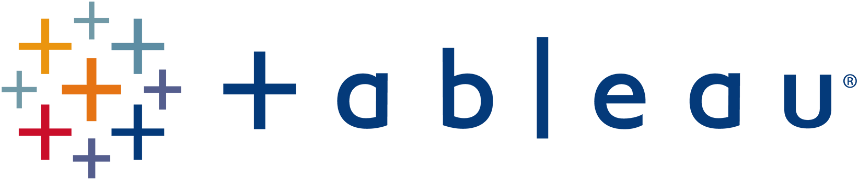
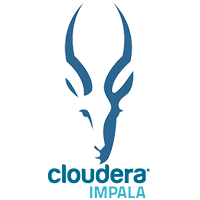


**PROGRAMA EXPERTO BIG DATA 2016-2017**

**TRABAJO FIN DE EXPERTO**

**BUSINESS INTELLIGENCE SOBRE BIG DATA**



AUTOR: PEDRO TOBARRA GUILLAMÓN (Estudiante Programa Experto Big Data U-TAD)

TUTOR: JULIO CONCA PASTOR (Solutions Specific Knowledge Analyst at Everis)



TABLA DE CONTENIDOS

[1. DESCRIPCIÓN 3](#_Toc492838458)

[2. OBJETIVOS 3](#_Toc492838459)

[3. TECNOLOGIAS UTILIZADAS 3](#_Toc492838460)

[4. ORIGEN DE LOS DATOS 3](#_Toc492838461)

[5. PLAN BASICO A SEGUIR 3](#_Toc492838462)

[6. EXPLORACION Y DEFINICION DE LOS GRÁFICOS A REALIZAR 4](#_Toc492838463)

[7. CARGADO Y PROCESADO EN SPARK DEL DATASET ELEGIDO 7](#_Toc492838464)

[8. CARGA DEL RESULTADO EN HIVE E IMPALA 19](#_Toc492838465)

[9. IMPLEMENTACIÓN DE LOS GRÁFICOS CON TABLEAU 24](#_Toc492838466)

# DESCRIPCIÓN

Los proyectos Big Data están englobados en dos grandes conjuntos, operacionales y de investigación. Los proyectos operacionales serían tipos de proyectos que ya se están realizando con las herramientas tradicionales, pero utilizando las tecnologías Big Data para realizarlos sobre más información, más rápido y/o de forma más económica. Son los proyectos que están realizando las empresas que quieren aproximarse al mundo Big Data. Business Intelligence, la habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios, sería uno de esos proyectos de mejora operacional. En este trabajo partiríamos de datos meteorológicos para transformarlos en conocimiento y poder consumirlos de una forma visual.

# OBJETIVOS

* Introducirse en uno de los casos de uso más comunes de introducción al Big Data para las empresas.
* Crear cuadros de mando con un enfoque más clásico (Herramientas de BI y HiveQL)

# TECNOLOGIAS UTILIZADAS

* Apache Spark versión 2.2.0 con Python versión 2.7.3 para el formateado de datos y la obtención del Dataframe que será utilizado en Hive como Datawarehouse.
* Jupyter Notebook para la introducción de comandos en PySpark
* HDFS para el almacenamiento de las tablas de Datawarehouse y del Datamart.
* Hive para la creación de las tablas del Datawarehouse y del Datamart y Cloudera Impala para la realización de consultas sobre el Datamart.
* Hue de Coudera Hadoop como interfaz visual con HDFS, Hive e Impala.
* Tableau Professional para la presentación visual de los datos.

# ORIGEN DE LOS DATOS

Los datos son obtenidos de estaciones meteorológicas repartidas en todo el mundo y cuyas mediciones son recogidas por la ‘National Oceanic and Atmosferic Administration NOAA’ de los Estados Unidos; que, además, son públicos y pueden descargarse desde el siguiente enlace:

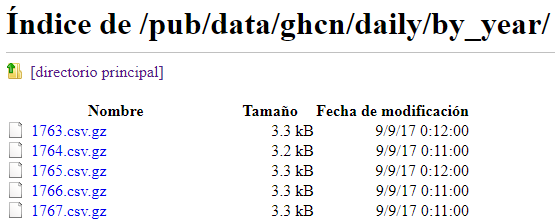
<ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/>

# PLAN BASICO A SEGUIR

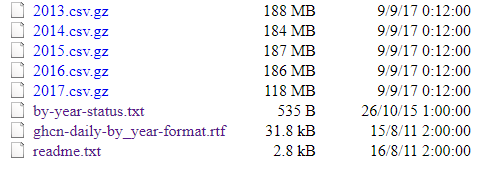
* Exploración y Definición de los gráficos a realizar.
* Carga y Procesado en Spark del dataset elegido.
* Carga del resultado en Hive e Impala.
* Implementación de los gráficos con Tableau.

# EXPLORACION Y DEFINICION DE LOS GRÁFICOS A REALIZAR

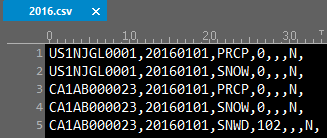
En la dirección <ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/> se pueden encontrar datos diarios de más de 50 magnitudes meteorologicas de más de 200 paises de todo el mundo. Estos datos han sido agrupados anualmente desde el año 1763 hasta la actualidad como se puede observar en la siguiente figura:



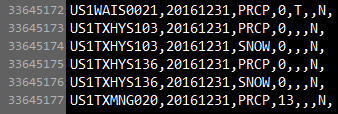
…

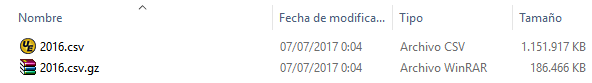


A modo de ejemplo se muestra un extracto del fichero de texto ‘2016.csv’ que resulta de descomprimir el fichero ‘2016.csv.gz’:



…

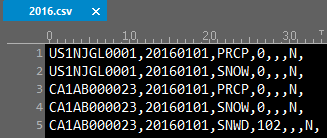




Como se puede observar en las figuras anteriores, el fichero de texto 2016.csv tiene 33.6 millones de líneas y ocupa 1.1 GB.

Siendo el año 2016 el año completo para el que más cantidad de datos había disponibles; se decidió escoger el fichero ‘2016.csv’ como fuente de datos para realizar este proyecto.

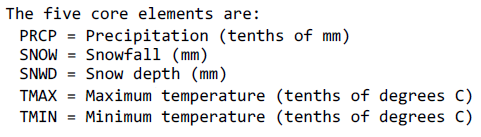
El formato de los datos del fichero ‘2016.csv’ es el siguiente:



‘US1NJGL001’ es el nombre de la estación meteorológica que recogió los datos. Concretamente las dos primeras letras del nombre de la estación corresponden al país en el que se encuentra la misma. Por ejemplo ‘US’ es para los Estados Unidos y ‘CA’ para Canada.

‘20160101’ es la fecha en la que se recogieron los datos en formato AñoMesDía.

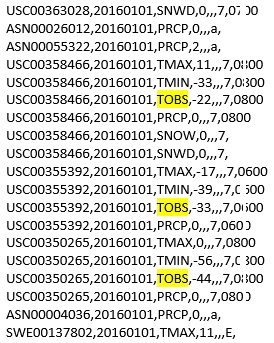
‘PRCP’, ‘SNOW’, SNWD’ son los nombres del tipo de magnitud correspondiente al dato. Se recogen más de 60 tipos de magnitudes meteorológicas distintas; pero las 5 magnitudes ‘core’ más importantes del dataset son:



Los valores ‘0’, ‘0’, ‘0’, ‘102’ son el valor de la magnitud medida y el resto de parámetros a la derecha del valor de la magnitud no serán objeto de este proyecto y serán, por tanto, eliminados en el procesamiento inicial de los datos.

Tras la exploración de la información disponible se decidió que la información a procesar y formatear sería la correspondiente estas 5 magnitudes core; eliminándose todas aquellas líneas que contuvieran datos de otras magnitudes.

Es decir que de un fichero csv de entrada:

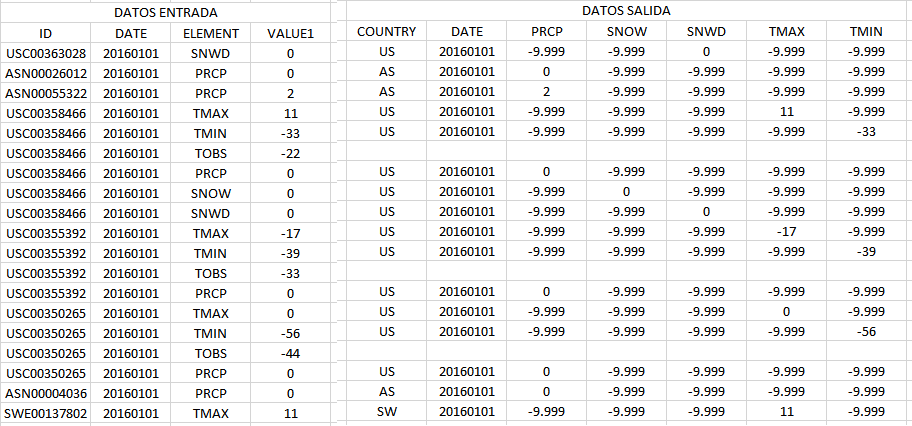


Se obtendría el siguiente fichero csv de salida:



Que solo contiene datos correspondientes a las magnitudes ‘core’ (las líneas de la magnitud TOBS se han eliminado) y donde se ha elegido el valor ‘-9999’ para distinguir aquellas magnitudes de las que no se tiene medida de las magnitudes de las que se tiene medida con valor ‘0’.

La misma transformación se muestra a través de las siguientes tablas:



Como se mostrará más adelante en este documento, las columnas de la tabla ‘DATOS SALIDA’ forman el dataframe que se obtendrá tras procesar los datos con Spark y Python; y que será el que se usará como tabla para contener el ‘datawarehouse’ de datos del proyecto.

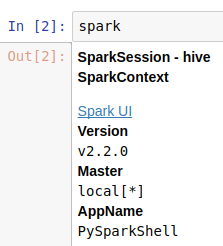
A partir de este datawarehouse, según lo acordado con el tutor, se pintará una gráfica de las temperaturas máximas diarias a lo largo de 2016 filtradas por países. Para ello, a partir del datawarehouse se creará un datamart específico para este propósito con las siguientes columnas : ‘COUNTRY,DATE,TMAX’.

# CARGADO Y PROCESADO EN SPARK DEL DATASET ELEGIDO

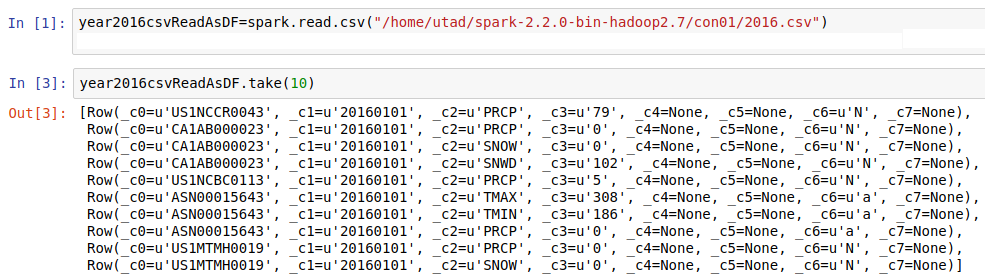
A continuación se iran exponiendo y explicando las instrucciones y transformaciones utilizadas para transformar el fichero ‘2016.csv’ en el dataframe comentando al final del apartado anterior.

Se ha utilizado Jupyter Notebook con Apache Spark versión 2.2.0 y Python versión 2.7.3.

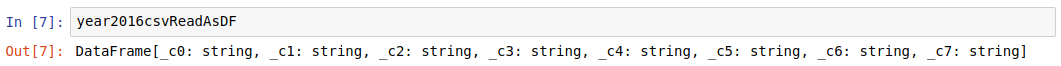
Los parámetros de la sesión son:

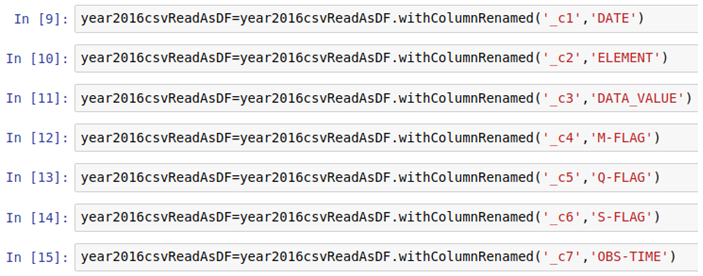


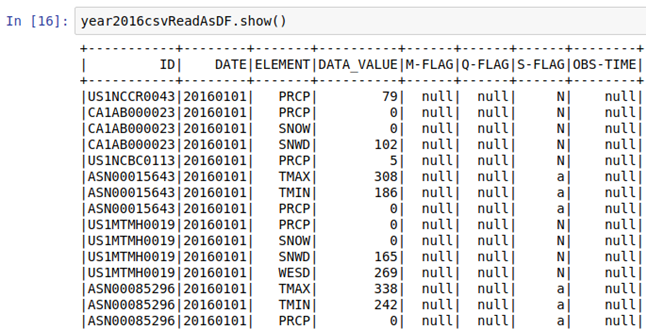
Cargamos el fichero 2016.csv como un dataframe separado en columnas:



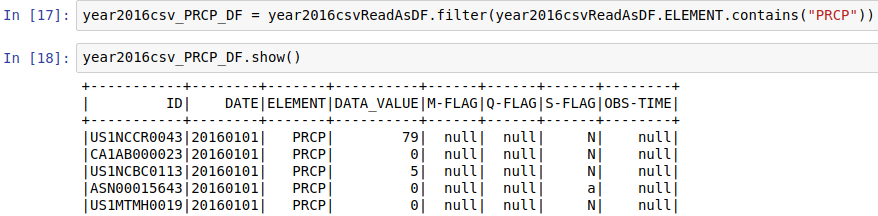
Renombramos las columnas del dataframe:

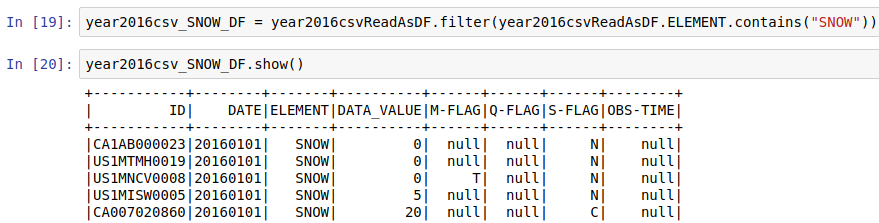


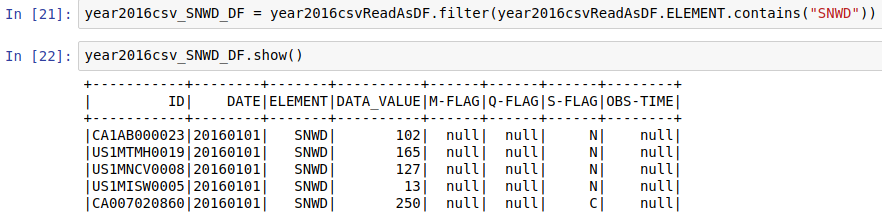


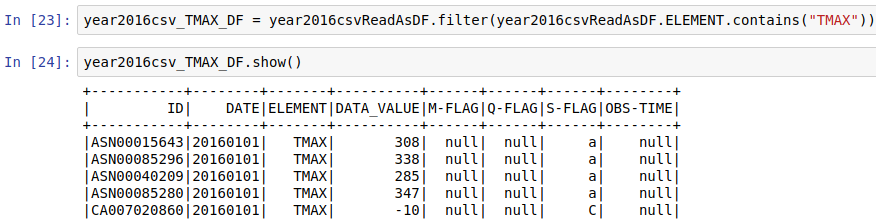


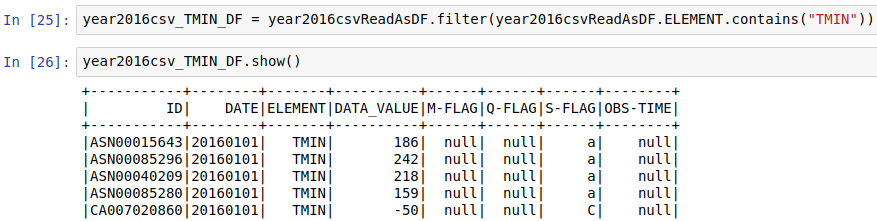
Creo 5 dataframes de manera que cada uno tenga datos de una de las cinco magnitudes core:



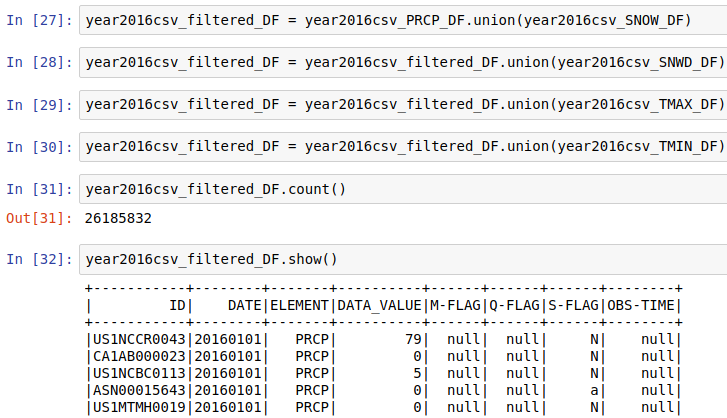


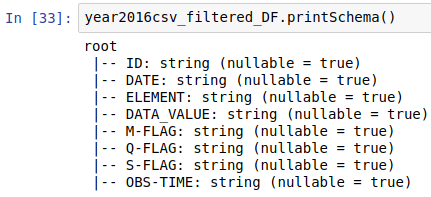




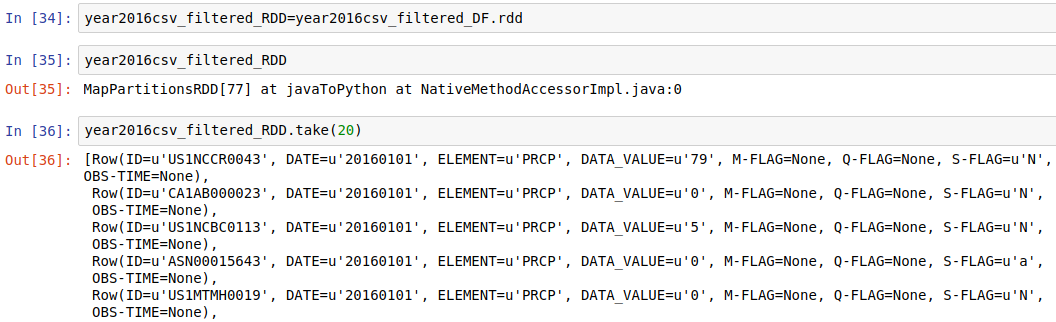


A continuación, uno los 5 dataframes uno a continuación del otro de manera que obtengo un dataframe solo con datos de las 5 magnitudes core una a continuación de la otra. Nótese que hemos pasado de un dataframe de casi 34 millones de líneas a uno de 26 millones:

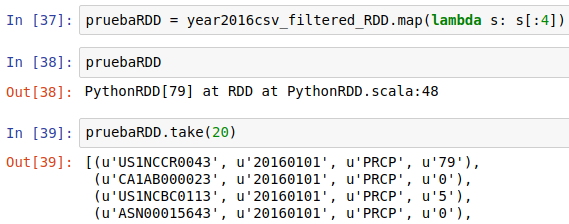




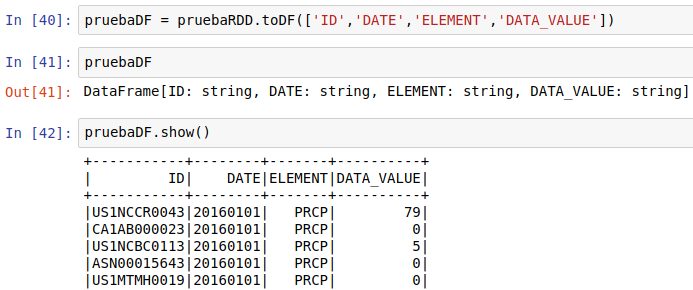
Paso el dataframe a formato RDD:

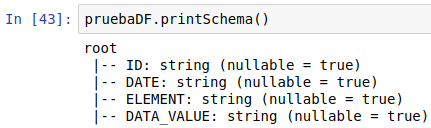


A continuación, me quedo con los cuatro primeros elementos de cada Row que son con los que voy a trabajar;

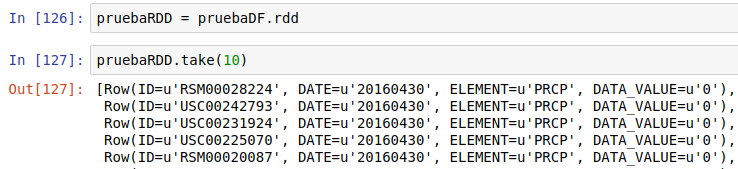


Vuelvo a pasar el RDD a formato dataframe nombrando las columnas:

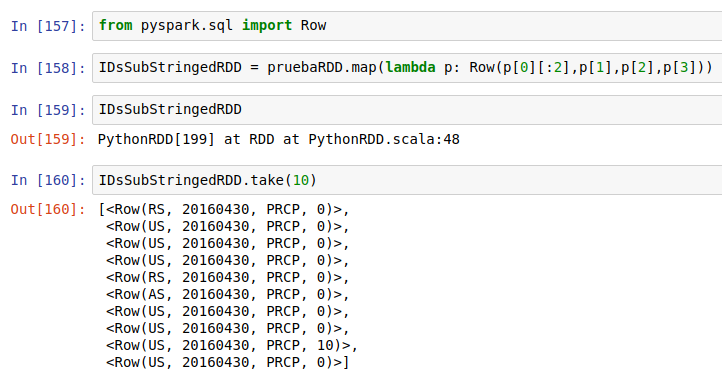




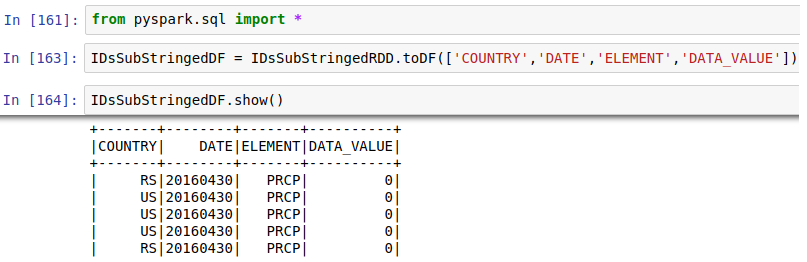
Ahora vuelvo a pasar el dataframe obtenido a formato RDD:



Ahora me quedo sólo con los 2 primeros caracteres de la primera columna pues necesito el nombre del país; NO el de la estación meteorológica:



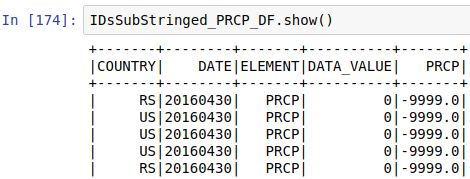
Paso el RDD obtenido a dataframe nombrando las columnas:



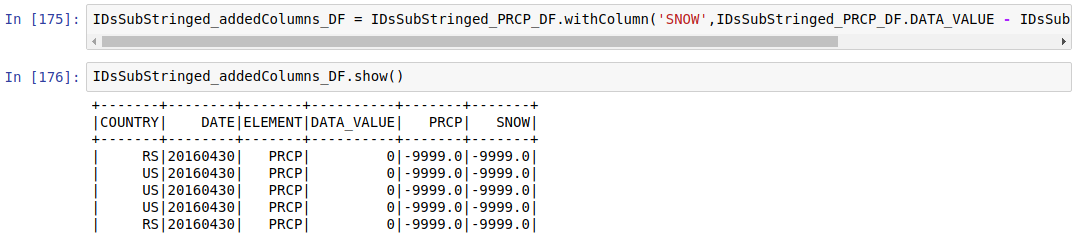
Añado una columna con el nombre de cada magnitud core rellenada con el valor ‘-9999’ para todas las filas de la tabla:

*IDsSubStringed\_PRCP\_DF = IDsSubStringedDF.withColumn('PRCP',IDsSubStringedDF.DATA\_VALUE - IDsSubStringedDF.DATA\_VALUE - 9999)*

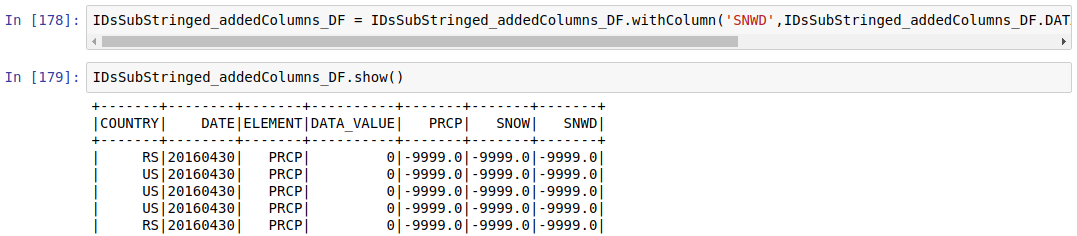




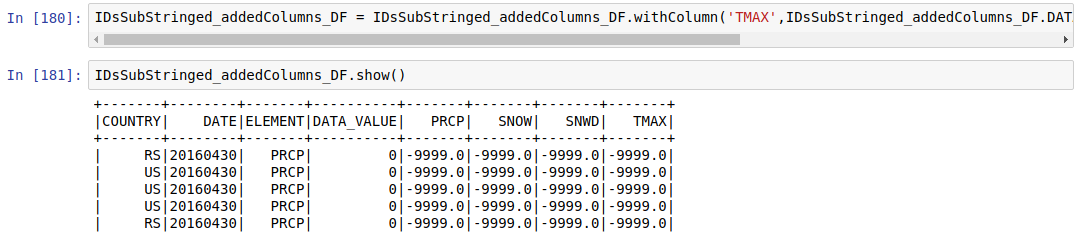
*IDsSubStringed\_addedColumns\_DF = IDsSubStringed\_PRCP\_DF.withColumn('SNOW',IDsSubStringed\_PRCP\_DF.DATA\_VALUE - IDsSubStringed\_PRCP\_DF.DATA\_VALUE - 9999)*



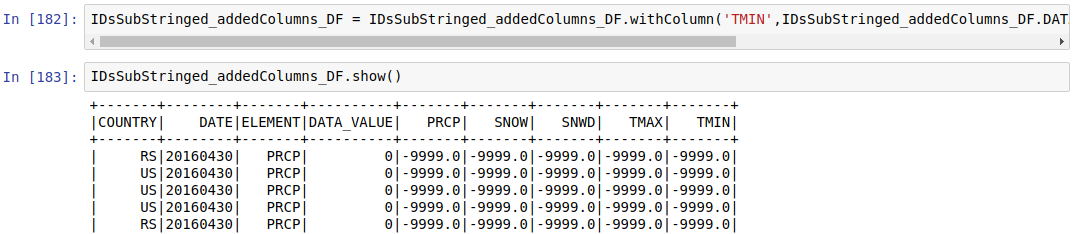
*IDsSubStringed\_addedColumns\_DF = IDsSubStringed\_addedColumns\_DF.withColumn('SNWD',IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - 9999)*



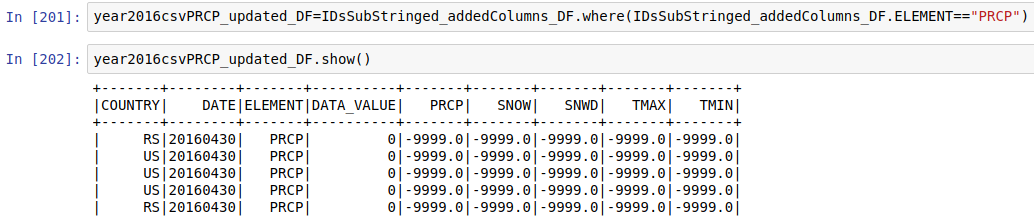
*IDsSubStringed\_addedColumns\_DF = IDsSubStringed\_addedColumns\_DF.withColumn('TMAX',IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - 9999)*



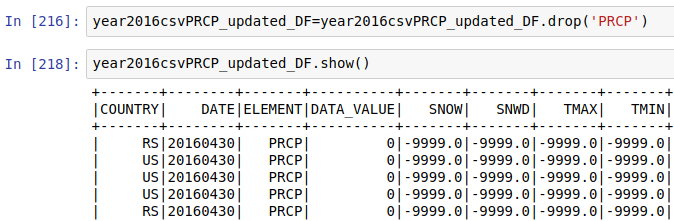
*IDsSubStringed\_addedColumns\_DF = IDsSubStringed\_addedColumns\_DF.withColumn('TMIN',IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - IDsSubStringed\_addedColumns\_DF.DATA\_VALUE - 9999)*



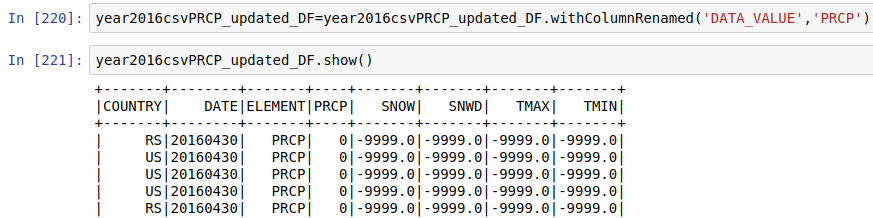
Creo un dataframe cogiendo sólo solo las líneas ‘PRCP’:



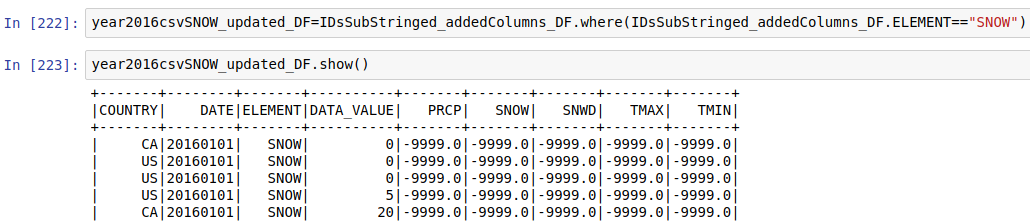
Elimino la columna ‘PRCP’:

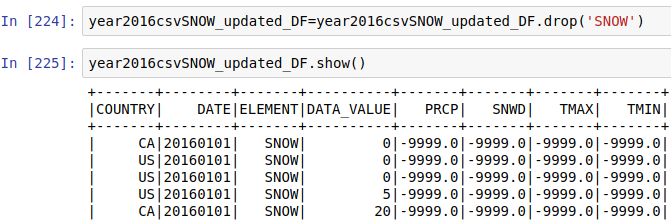


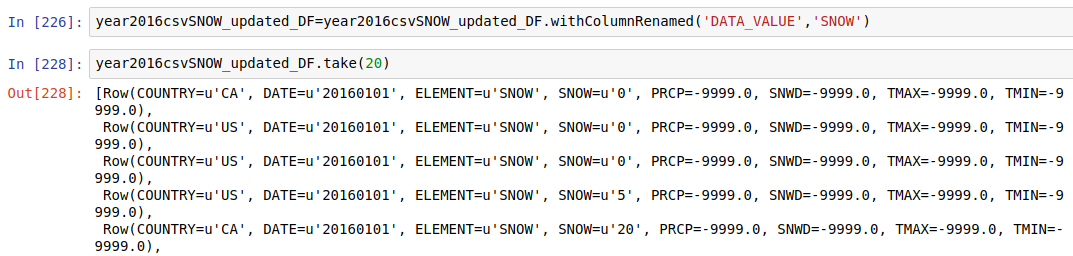
Cambio el nombre de la columna ‘DATA\_VALUE’ por el de ‘PRCP’:



Realizo esta misma operación para las otras cuatro magnitudes core:

























Unimos los 5 dataframes obtenidos uno detrás de otro:

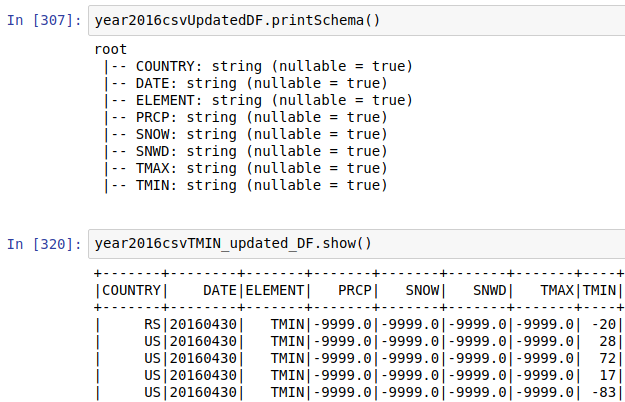






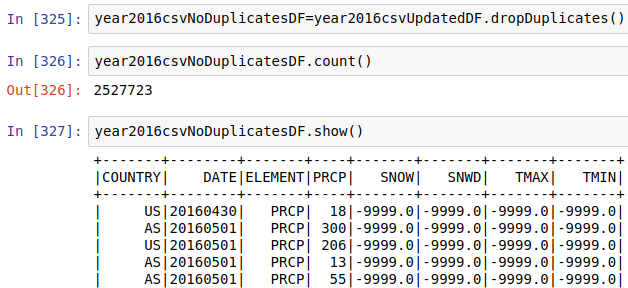


Y obtengo un dataframe con las siguientes características:

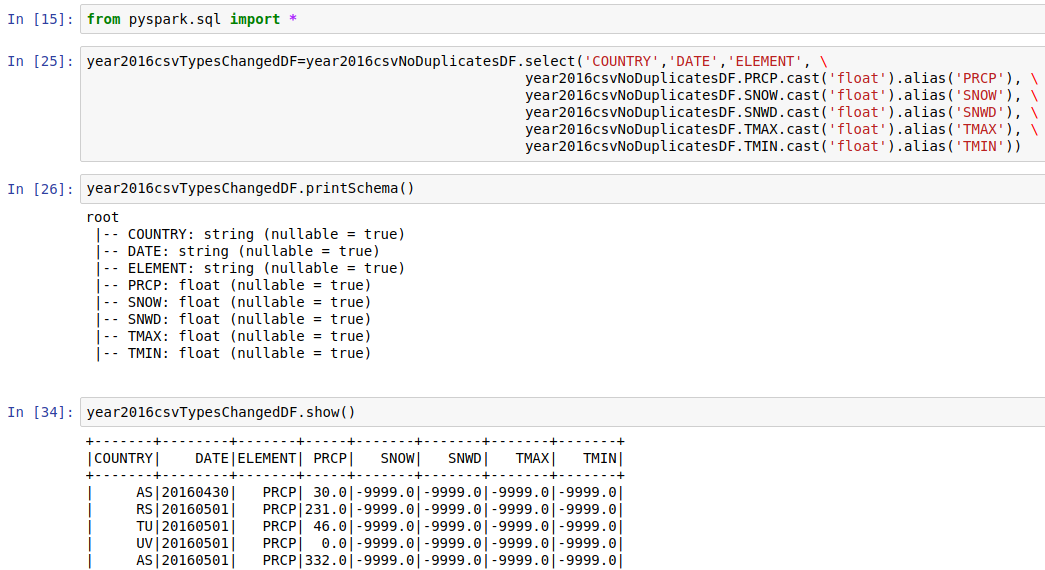




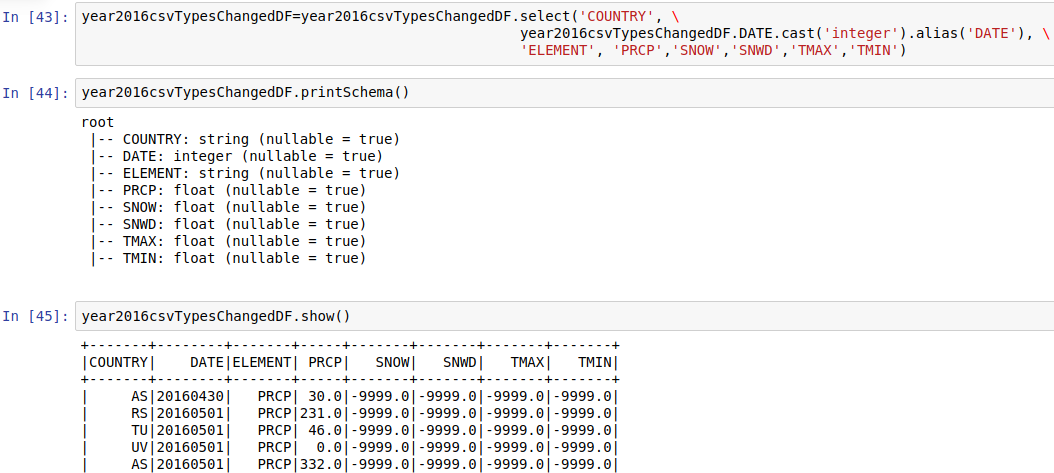
Elimino las líneas duplicadas:



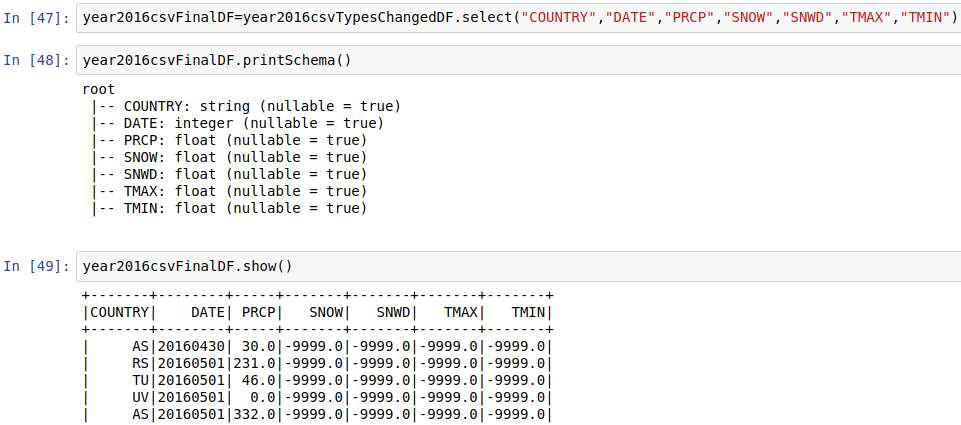
Cambio los tipos de las magnitudes core de ‘string’ a ‘float’:



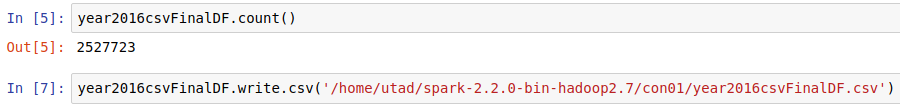
Cambio el tipo de la fecha de ‘string’ a ‘integer’:



Creo el dataframe final ya sin la columna ‘ELEMENT’:

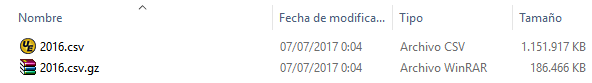


Finalmente guardo el dataframe como un fichero de texto ‘csv’ que utilizaré para crear el datamart en Hive en el siguiente capítulo:

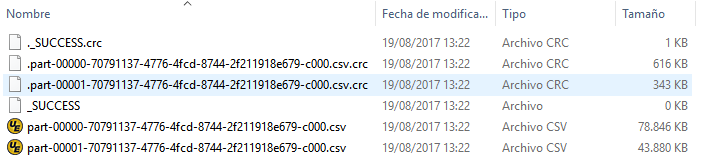


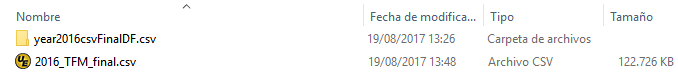
Nótese que antes del procesado de los datos teníamos un fichero csv de texto de 1.1 GB y 33 millones de líneas y al final de la transformación hemos obtenido uno de 122 MB y 2.5 millones de líneas.





…





**NOTA**: en la figura anterior el fichero de texto 2016\_TFM\_final.csv es el resultado de unir los ficheros   
‘part-00000-70791137-4776-4fcd-8744-2f211918e679-c000.csv’  
 y ‘part-00001-70791137-4776-4fcd-8744-2f211918e679-c000.csv’ con un editor de texto.

En un proyecto profesional se tendría que haber pasado el dataframe final desde Spark a Hive sin necesidad de descargarlo en un fichero texto. Debido a dificultades técnicas no se pudo ejecutar Spark versión 2.2.0 en la máquina virtual CentOS 6.7 de Cloudera. Por ello la transformación que se ha descrito se ha realizado en una máquina virtual ‘ubuntu 14.04 LTS’. Al tener que cambiar de máquina virtual para continuar el proyecto se me permitió cargar el dataframe final en Hive desde un fichero en HFDS en lugar de leerlo directamente desde Spark.



# CARGA DEL RESULTADO EN HIVE E IMPALA

En esta sección se describe la creación del datawarehouse en hive a partir del dataframe obtenido en el apartado anterior, la creación del datamart de temperaturas maximas diarias a partir del datawarehouse y la consulta a realizar en Impala sobre el datamart para alimentar en la siguiente sección a Tableau con los resultados de dicha consulta para pintar la gráfica de temperaturas máximas diarias mundiales filtradas por países.

Se crea y se utiliza en hive la base de datos ‘con01’ (CON01 es la denominación de U-TAD para el proyecto)





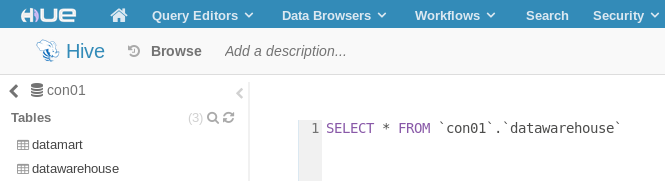
Se crea la estructura vacía de la tabla datawarehouse:

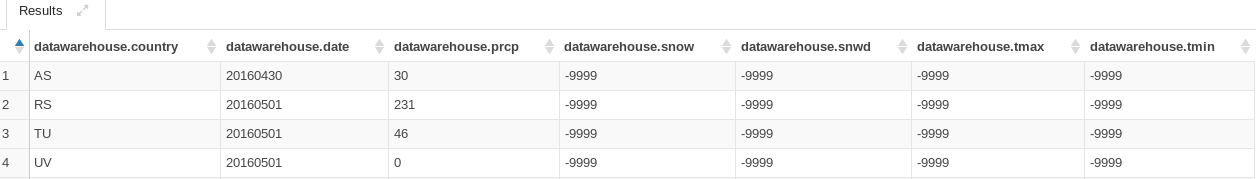


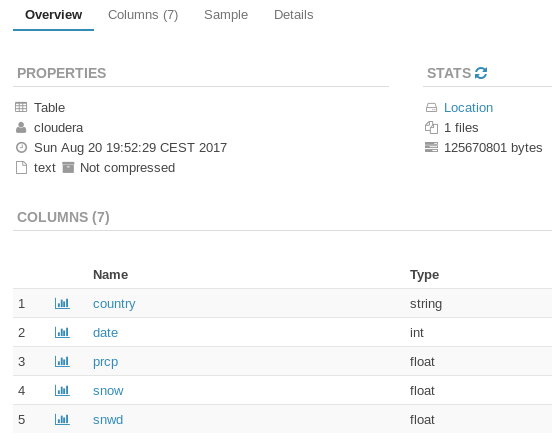
Se realiza la carga del fichero csv obtenido en el apartado anterior desde el HDFS al ‘datamart’:

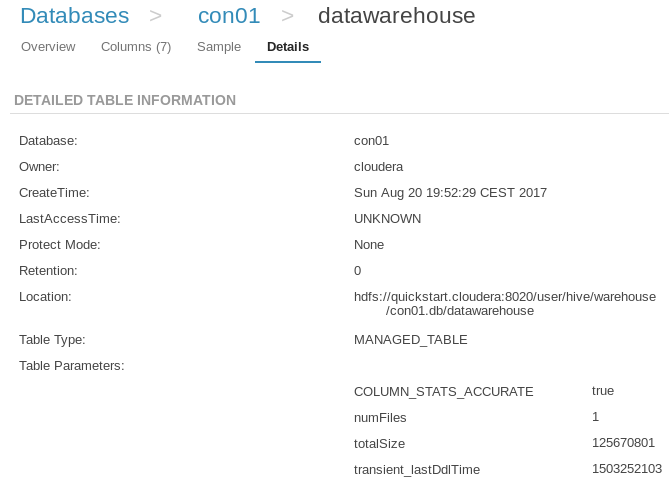


Con esto ya tendríamos creado el datawarehouse cuyas principales características se muestran a continuación:

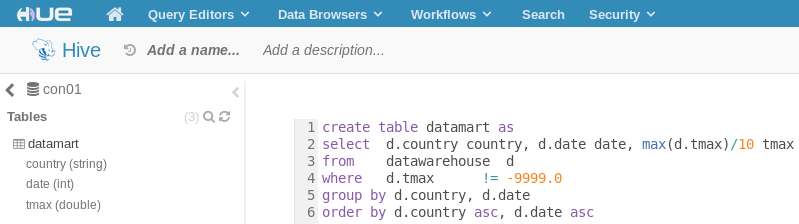




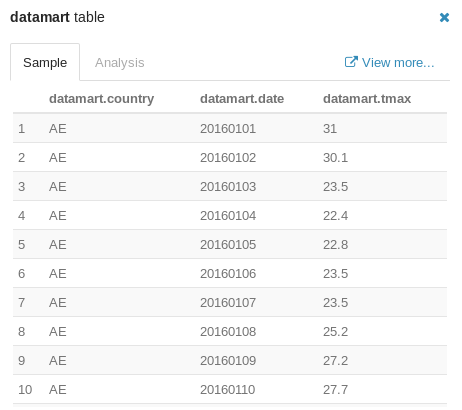


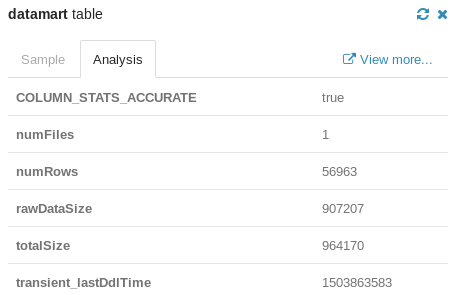


Una vez creado el datawarehouse en Hive, estamos en condiciones de crear la tabla datamart de temperaturas diarias máximas mundiales durante el año 2016 mediante la siguiente consulta:

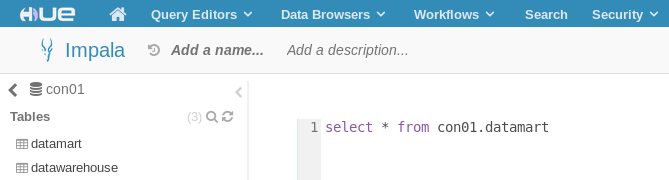


Y los principales parámetros del datamart se muestran a continuación:





A continuación, ya podemos realizar una consulta con Impala sobre el datamart para obtener los datos de temperaturas máximas mundiales diarias durante 2016 filtradas por países:



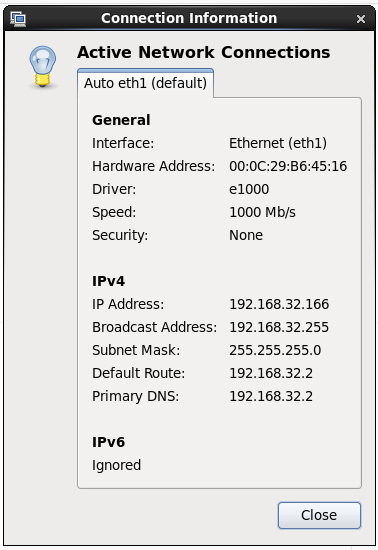


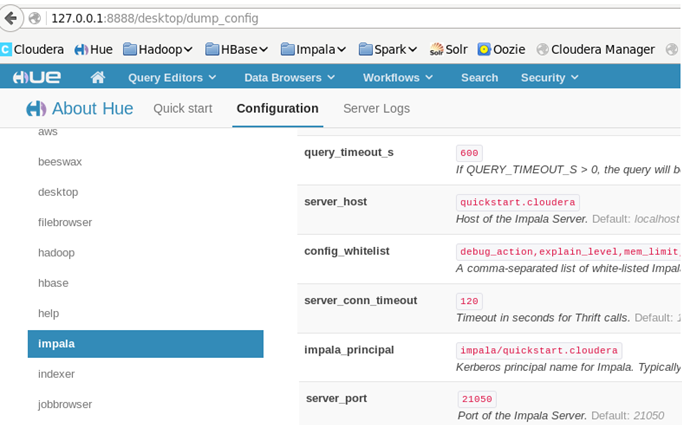
Esta será la query que en la siguiente sección veremos que hay que realizar desde Tableau para acceder a los datos del datamart y poder pintar la gráfica de temperaturas máximas mundiales diarias durante 2016 filtradas por países.

# IMPLEMENTACIÓN DE LOS GRÁFICOS CON TABLEAU

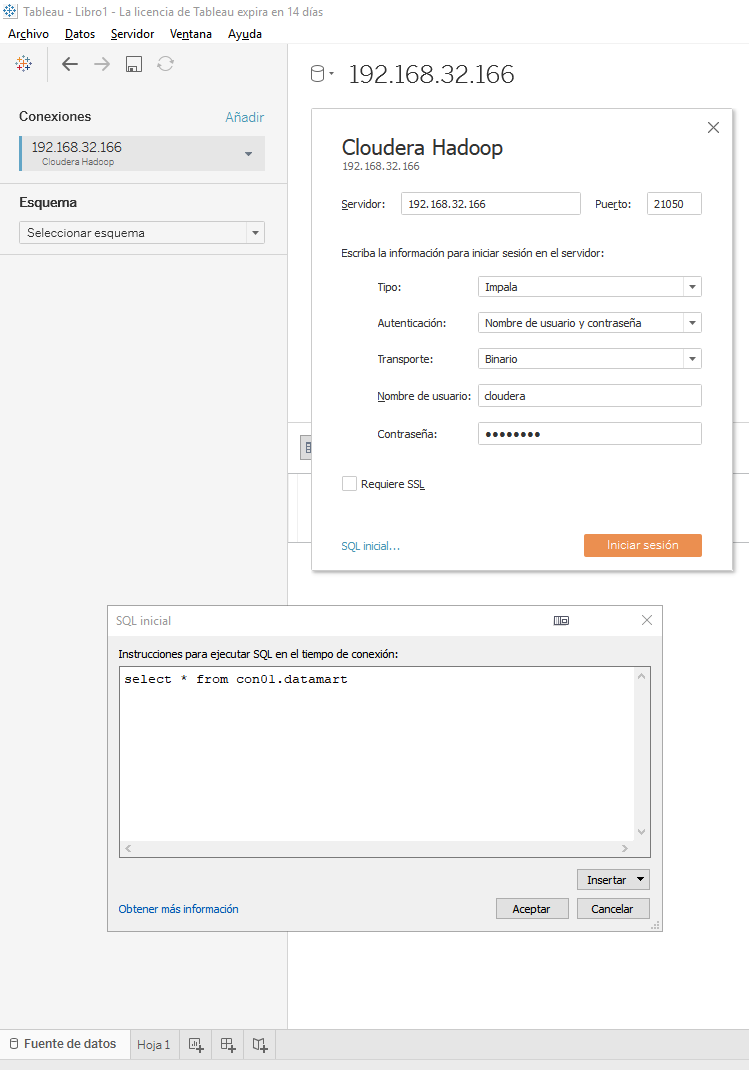
Para realizar esta etapa se ha trabajado con la herramienta ‘Tableau Desktop Professional Edition’ versión 10.3.2 ejecutándose sobre la máquina huésped con sistema operativo ‘Windows 10 Home 64-bit’ y Hive e Impala abiertos sobre la máquina virtual de Cloudera Linux CentOS versión 6.7

Para conectar las 2 máquinas y que Tableau pueda leer el datamart con Impala primero hay que averiguar la dirección IP de la máquina virtual y el puerto donde se está ejecutando Impala.





Con la información obtenida ya puede configurarse la conexión de Tableau con Impala y la query de descarga del datamart según se muestra en la siguiente figura:



Y antes de haber intentado la conexión hay que descargar los drivers ODBC desde la página web de Cloudera e instalarlos en el sistema operativo huésped Windows según se muestra en las siguientes figuras:

<https://www.cloudera.com/downloads.html>



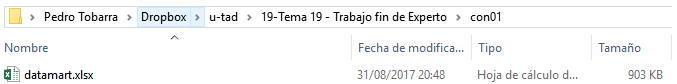
cid:image006.png@01D324F8.9AB0E550

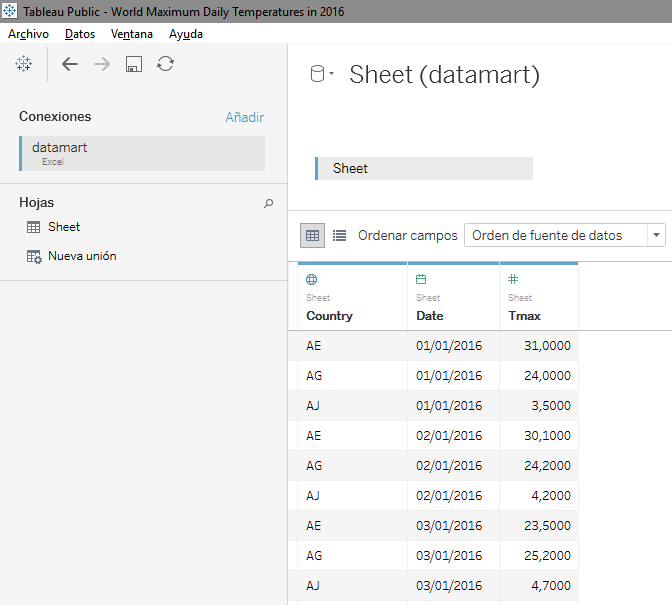
**NOTA**: después de haber realizado las operaciones descritas en este capítulo, conseguí realizar la conexión entre las dos máquinas, pero no conseguí que se descargaran en tableau los resultados de la query. Tras intentar resolver el problema con mi tutor; decidimos que como los parámetros para conectar las dos máquinas estaban puestos correctamente, lo mejor era que continuara el proyecto descargando el resultado de la query en una hoja Excel desde Impala-HUE que será lo que mostraré a continuación. Pero en un proyecto profesional de Big Data se descargarían los datos directamente entre las dos máquinas.

Después de lo explicado, como ya no necesito hacer una conexión entre las dos máquinas, continuaré trabajando con ‘Tableau Desktop Public Edition’ versión 10.3.2.

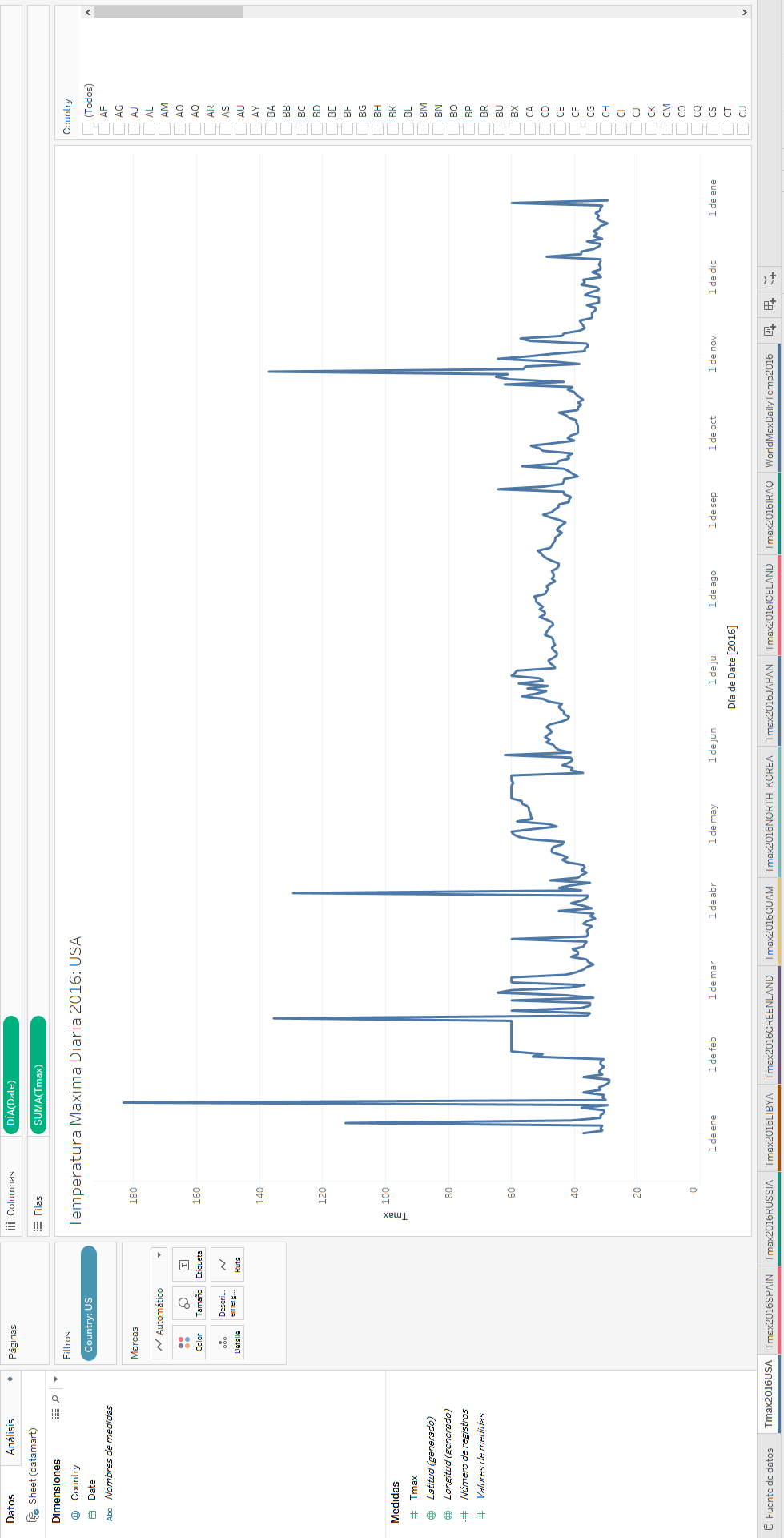
En la figura de la página siguiente se puede ver el resultado de la descarga del resultado de la query en Tableau que, a su vez, fue descargada en un directorio local de la máquina virtual de cloudera desde HUE-Impala. Este directorio local era compartido por la máquina huésped Windows y por la máquina virtual Cloudera mediante la aplicación Dropbox, la cual se ejecutaba en las dos máquinas.

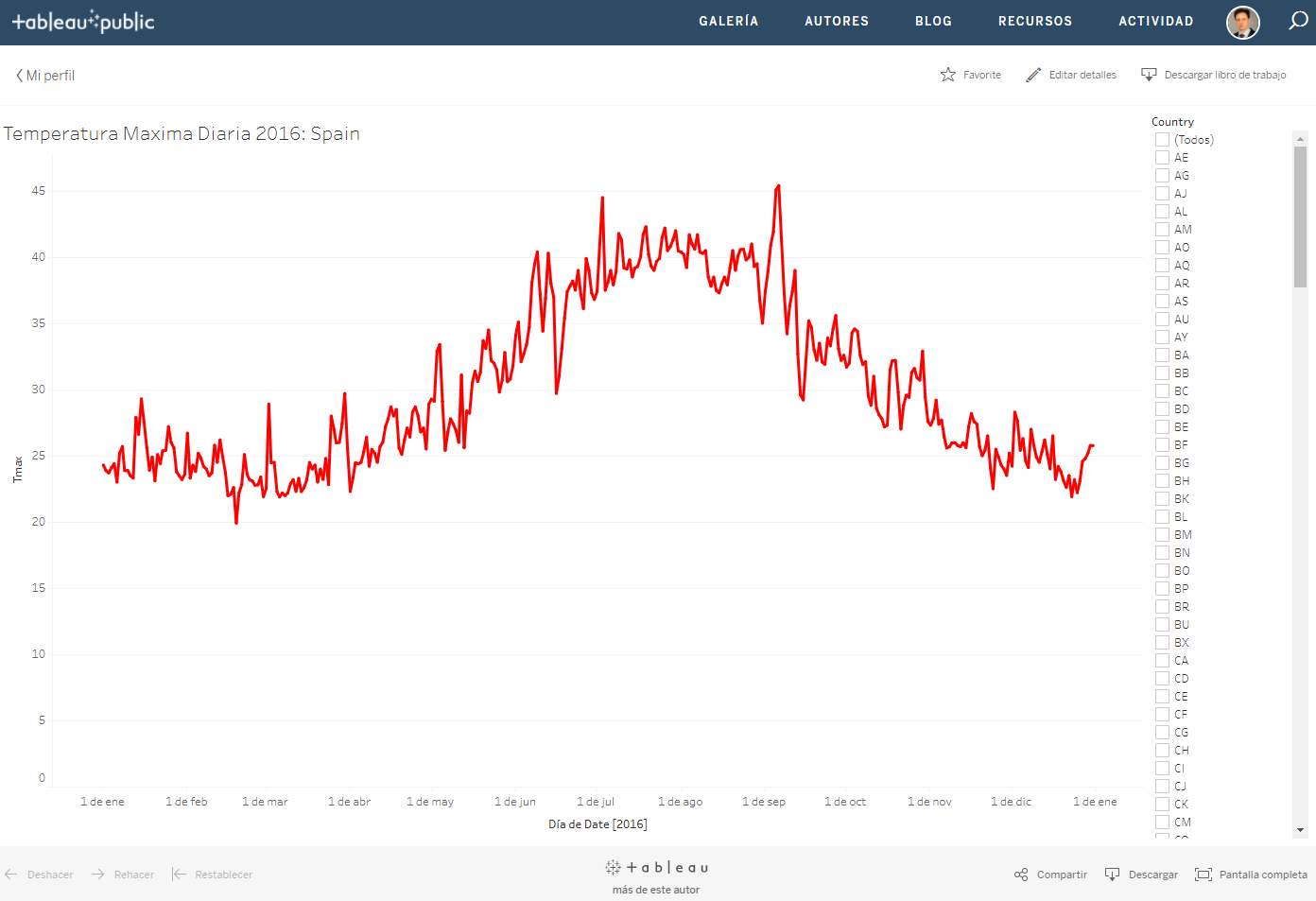
Nótese que después de todas las transformaciones realizadas en este proyecto, el datamart queda reducido a un fichero Excel de 3 columnas y 57.000 filas que ocupa 903 KB.

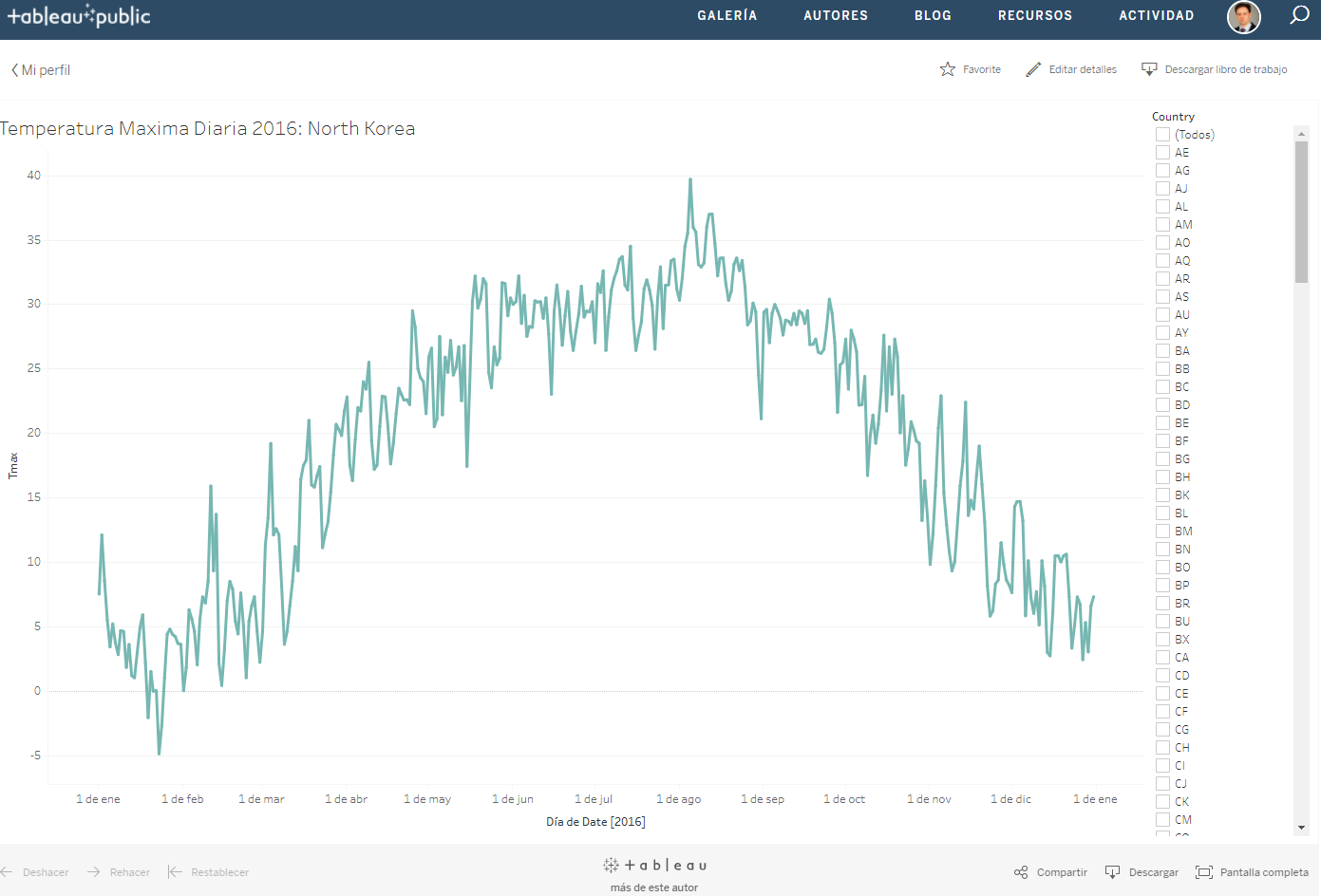




Una vez accesibles los datos desde Tableau podemos pasar a pintar la gráfica filtrada por países. En la página siguiente se muestra la configuración completa de la gráfica de temperaturas diarias máximas durante 2016 en Estados Unidos y luego, a modo de curiosidad, las gráficas de las temperaturas de otros países.









Los datos utlizados por Tableau, el fichero del libro de trabajo, las gráficas en formato imagen y pdf, y los datos de las tabulaciones cruzadas de las gráficas, son públicos y accesibles desde el perfil público de Tableau del autor de este proyecto al que se puede acceder desde el siguiente enlace:

[https://public.tableau.com/profile/pedro.tobarra.guillamon#!/](https://public.tableau.com/profile/pedro.tobarra.guillamon%23!/)