



HITO GRUPAL PROGRAMACIÓN

2ª EVALUACIÓN



CHEDEY GALLEGO, JAVIER AHIJADO, DANIEL MANZANO

1º DAM
GETAFE

ÍNDICE

Fase 1.....	2
1. HABLAMOS DE FUENTES DE DATOS. DE GRANDES VOLÚMENES DE DATOS. POR EJEMPLO, DATA LAKE O SIMILAR. TAMBIÉN ES IMPORTANTE TRATAR LA DIFERENCIA ENTRE DATOS ESTRUCTURADOS Y NO ESTRUCTURADOS EN RELACIÓN AL BIG DATA.	2
2. ENTRE LAS HERRAMIENTAS MÁS INTERESANTES A LA HORA DE GESTIONAR GRANDES VOLÚMENES DE DATOS NOS ENCONTRAMOS CON HADOOP Y SPARK. HABRÍA QUE TRATAR SUS CARACTERÍSTICAS Y FINALIDAD.	2
SPARK.....	2
HADOOP.....	3
3. EXISTEN LENGUAJES DE PROGRAMACIÓN “RECOMENDABLES” PARA GESTIONAR DATOS. ENTRE ELLOS, ESTÁN PYTHON Y SCALA. SERÍA EXPLICAR BREVEMENTE POR QUÉ. .	4
SCALA.....	4
PYTHON.....	5
4. EN LA PARTE DE VISUALIZACIÓN DE DATOS, DE MOSTRAR DASHBOARDS NOS ENCONTRAMOS CON POWERBI Y TABLEAU ENTRE OTROS. DEBEMOS EXPLICAR QUÉ SON.	7
POWER BI	7
TABLEAU.....	7
FASE 2.....	8
EN ESTA SEGUNDA FASE SE REALIZA LA IMPLEMENTACIÓN DE LA INVESTIGACIÓN. EN CONCRETO SERÍA ACCEDER A UN VOLUMEN DE DATOS Y MOSTRARLO. PODRÍAMOS UTILIZAR SCALA O PYTHON Y MOSTRAR EL RESULTADO EN POWERBI O TABLEAU. LA IDEA ES QUE SEA ALGO MUY IMPACTANTE POR LA CALIDAD DE CONTENIDO TRATADO, VELOCIDAD DE ACCESO, VOLUMEN DE DATOS.....	8
FASE 3.....	13
PARA FINALIZAR, REALIZAMOS UNA EVALUACIÓN O CONSIDERACIONES DE CÓMO HAN EVOLUCIONADO EL ACCESO A DATOS EN LOS ÚLTIMOS AÑOS. DESDE ACCESO A FICHEROS, PASANDO POR BASE DE DATOS Y CONSUMIENDO APIS.	13

Fase 1

1. **HABLAMOS DE FUENTES DE DATOS. DE GRANDES VOLÚMENES DE DATOS. POR EJEMPLO, DATA LAKE O SIMILAR. TAMBIÉN ES IMPORTANTE TRATAR LA DIFERENCIA ENTRE DATOS ESTRUCTURADOS Y NO ESTRUCTURADOS EN RELACIÓN AL BIG DATA.**

Una fuente de datos en Big Data puede ser cualquier lugar donde se almacenan grandes volúmenes de datos, como un data Lake, bases de datos relacionales, archivos, y sensores.

Los datos Lake son repositorios centralizados diseñado para almacenar, procesar y proteger grandes cantidades de datos estructurados, semiestructurados o no estructurados. Puede almacenar datos en su formato nativo y procesar cualquier variedad de datos, ignorando los límites de tamaño.

En el Big Data se trata de la cantidad de datos generados por diferentes fuentes que excede la capacidad de procesamiento y almacenamiento convencionales.

¿Qué son los datos estructurados y los no estructurados?

- Datos estructurados: Son aquellos que tienen un formato claro y están organizados en una tabla o esquema.
- Datos no estructurados: Son aquellos que no tienen un formato claro o están organizados de manera libre.

La principal diferencia entre los dos, es que los datos estructurados son más fáciles de procesar y analizar, mientras que los datos no estructurados requieren un procesamiento adicional para ser utilizados.

La mayoría de los datos en Big Data son no estructurados y representan un gran desafío para su procesamiento y análisis.

2. **ENTRE LAS HERRAMIENTAS MÁS INTERESANTES A LA HORA DE GESTIONAR GRANDES VOLÚMENES DE DATOS NOS ENCONTRAMOS CON HADOOP Y SPARK. HABRÍA QUE TRATAR SUS CARACTERÍSTICAS Y FINALIDAD.**

SPARK

Spark es un marco de procesamiento de Apache el cual facilita grandes dimensiones de datos. Spark tiene una función que le permite calcular esas grandes cantidades de datos en un corto espacio de tiempo.

¿Cómo instalar Spark?

1. Primero tienes que descargar la última versión de Spark desde el sitio web.
2. A continuación, tienes que descomprimir el archivo descargado en tu ordenador o máquina.
3. Después, tienes que configurar la aplicación para acceder a Spark desde una línea de comandos.

4. Por último, para verificar que la instalación se ha realizado correctamente tienes que usar el comando 'spark-shell' en la línea de comandos de la aplicación.

Características de Spark

- Spark utiliza procesamiento de memoria, lo que significa que almacena datos en una memoria RAM para hacer más rápido el movimiento de datos.
- Spark es escalable, es decir que puede procesar grandes cantidades de datos.
- Spark consta de una API, (Interfaz de programación unificada) en lenguajes de programación como Scala, Python o Java. Esto facilita que los programadores puedan escribir código de aplicación de manera consistente.
- Spark posee un motor SQL el cual permite ejecutar sentencias SQL para gestionar bases de datos.
- Spark es capaz de procesar y ejecutar datos de manera instantánea lo que hace que sea a tiempo real.
- Spark puede procesar datos almacenados en sistemas como HDFS, HBase y Cassandra lo que hace que se integre con otros sistemas como Hadoop.
- Spark es una aplicación rápida y sencilla.

La finalidad de esta herramienta es mejorar la eficiencia y escalabilidad a la hora de procesar datos grandes cantidades de datos y hacerlo de una manera que sea accesible y fácil de usar para los programadores.

HADOOP

Hadoop es un tipo de software que emplea código abierto para procesar datos de gran volumen al igual que Spark. Hadoop divide sus tareas en dos partes principales: la primera es HDFS (Hadoop Distributed File System) la cual se encarga de almacenar los datos y MapReduce se encarga del procesamiento de datos.

¿Cómo instalar Hadoop?

1. Primero, debes entrar a la página web y descargar la versión de Hadoop más reciente.
2. Segundo, tienes que descomprimir el archivo en el directorio de tu computadora que desees.
3. Tercero, tienes que establecer la ruta de acceso tanto a Java como a la carpeta de Hadoop.
4. Después tienes que configurar los archivos de configuración de Hadoop.
5. Finalmente, verificamos mediante el comando "Hadoop version" si hemos instalado correctamente el programa.

Características de Hadoop

- Hadoop tiene una alta escalabilidad, lo cual permite el procesamiento de datos paralelamente con una amplia gama de servidores.
- Hadoop es tolerante a fallos, es decir, puede funcionar a pesar de que uno o dos nodos fallen.
- Facilita la distribución de datos entre diferentes nodos lo que hace más eficiente y útil la aplicación.

- Es abierto de manera que un gran número de programadores tengan acceso al programa.
- Es flexible, por lo que incluye variedad de imágenes, videos, texto...
- Por último, Hadoop es económico ya que permite procesar datos en servidores de bajo costo.

La finalidad de Hadoop es procesar datos de grandes dimensiones en distintos servidores lo que hace ideal para tareas relacionadas con el Big Data.

Además, Hadoop hace que el almacenamiento de datos se efectúe en diferentes formatos lo que hace variar desde la recopilación de datos en redes sociales hasta recetas médicas.

3. EXISTEN LENGUAJES DE PROGRAMACIÓN “RECOMENDABLES” PARA GESTIONAR DATOS. ENTRE ELLOS, ESTÁN PYTHON Y SCALA. SERÍA EXPLICAR BREVEMENTE POR QUÉ.

SCALA

Scala es un lenguaje de programación de propósito general, diseñado para el desarrollo utilizando patrones de una forma concisa, elegante y con tipos. Es de código abierto, integra principios de orientación a objetos y la programación funcional, permitiendo a los programadores ser más productivos y aprovechar los conocimientos y estructuras de otros lenguajes como Java.

Hay ciertos aspectos que sobresalen de Scala:

- El software Scala al igual que otros lenguajes soporta el paradigma de orientación a objetos, para muchos el paradigma debe contar con características como: encapsulamiento, herencia, polimorfismo, todos los tipos predefinidos son objetos, las operaciones se realizan enviando mensajes a objetos y por último todos los tipos definidos por usuarios son objetos.
- En el lenguaje Scala hay funciones, es decir, es funcional, lo que lo hace ideal para el procesamiento paralelo de datos.
- También hay valores de primera clase u objetos que pueden asignarse a diferentes variables. Comparte esta característica con Java, con la que tiene una buena interoperabilidad. Por esto, es muy usado en Big Data y aplicaciones como Spark, uno de los marcos de trabajo de Big data más populares y utilizados en la industria.
- Scala es un lenguaje multiparadigma, ya que soporta la programación orientada a objetos y la programación funcional, y es el primero en realizarlo en la JVM (Java Virtual Machine), lo que significa que puede utilizar bibliotecas y herramientas Java existentes.
- Uno de los objetivos del software Scala es tener un lenguaje que pueda crecer y escalar con la demanda. Scala es adecuado para utilizar como un lenguaje de script o para realizar aplicaciones empresariales. La abstracción de componentes, la sintaxis sucinta, y el soporte para los paradigmas de orientación a objetos y funcional (arquitectura de multinúcleo), hacen al lenguaje escalable y capaz de manejar grandes cantidades de datos.
- Otra característica de la tecnología de Scala es que tiene una sintaxis muy precisa y fácil de entender, lo que facilita la escritura de código y la lectura de código existente. En

comparación con Java, Scala es más legible y conciso. El compilador de Scala puede generar y trabajar para un mejor código como `String ()`, `equals ()`, etcétera.

- Las aplicaciones de Scala proporcionan varias bibliotecas y, por lo tanto, se pueden usar para construir muchos frameworks.

En los últimos años, la tecnología de Scala viene experimentando un gran crecimiento y ha pasado a convertirse en un estándar para muchas empresas, startups y universidades de todo el mundo.

Además, no solo se utiliza Scala para crear nuevos proyectos, sino también para otras herramientas de gran impacto en el mercado como los frameworks, como por ejemplo:

- *Play Framework*: Un framework de alta velocidad para el desarrollo con Java y Scala.
- *Apache Spark*: Motor de análisis para el procesamiento de datos a gran escala.
- *Akka*: Framework para aplicaciones reactivas, concurrentes y distribuidas con mayor facilidad.
- *Apache Kafka*: Herramienta para construir canalizaciones de datos y aplicaciones de transmisión en tiempo real.

Asimismo, también existen grandes librerías que promueven y hacen que sea más fácil la programación funcional, destacan:

- *Cats*: Librería ligera, modular y extensible para programación funcional.
- *Scalaz*: Conjunto de estructuras puramente funcionales para complementar las de Scala.
- *Magnolia*: Una Macro genérica para la materialización automática de clases de tipos para tipos de datos compuestos.
- *ZIO*: Este último creando un amplio y potente ecosistema que va a hacer en mi opinión más interesante Scala en el futuro.

Hay que destacar que el software Scala también se utiliza en muchos proyectos open source para acceso a datos y comunicación entre APIs.

PYTHON

Python es un lenguaje de programación recomendable para gestionar datos ya que se trata de un lenguaje en desarrollo constante, su código es simple, contiene inmensas bibliotecas y más características que explicaremos a continuación. Tiene un código y una sintaxis simple que hace que sea muy fácil de aprender, además de ser de código abierto, ya que esto hace que cualquier persona tenga acceso a sus recursos de forma gratuita. Python destaca por las siguientes características:

- **Simplicidad.** Python es conocido por hacer que los programas funcionen en la menor cantidad de líneas de código. Identifica y asocia automáticamente los tipos de datos y resulta un lenguaje fácil de usar y toma menos tiempo en la codificación. Tampoco hay limitación para el procesamiento de datos. Puede calcular datos en cualquier tipo de equipo y entorno. Anteriormente se argumentaba que Python era más lento que algunos de sus homólogos como Java y Scala, pero con la plataforma Anaconda se ha puesto al día demostrando que es rápido tanto en desarrollo como en ejecución.

- **Compatibilidad.** Hadoop es la plataforma de Big data de código abierto más popular y la compatibilidad inherente de Python es otra razón más para preferirlo a otros lenguajes.
- **Facilidad de aprendizaje.** En comparación con otros idiomas, Python es fácil de aprender incluso para los programadores con menos experiencia. Es un primer idioma ideal debido a tres razones principales: cuenta con amplios recursos de aprendizaje, garantiza un código legible y se rodea de una gran comunidad. La gran comunidad Python ofrece la seguridad de saber que, caso de encontrar problemas en el desarrollo, habrá otros que puedan echar una mano para ayudar a resolverlos.
- **Paquetes de gran alcance.** Python tiene un poderoso conjunto de paquetes para una amplia gama de necesidades de análisis y ciencia de datos. Algunos de los paquetes populares que le dan a este idioma una ventaja son NumPy, Pandas, Scipy, Scikit-learn, PyBrain, Tensorflow, Cython, PyMySQL, BeautifulSoup o iPython.
- **Visualización de datos.** Aunque R es mejor en lo que respecta a la visualización de datos, con paquetes recientes, Python para Big data ha mejorado su oferta en este espacio. Ahora existen API que pueden ofrecer buenos resultados.

Otra gran ventaja del lenguaje es su gran velocidad de procesamiento. Para analizar y procesar grandes cantidades de datos, este requisito es necesario.

Los analistas de datos utilizan Python porque es un gran lenguaje de programación para todas las necesidades, ofrece visualización para datos complejos, lo que simplifica su trabajo, por ello cada vez hay más profesionales de Data Science que usan Python para Big data.

Para finalizar, hay que tener en cuenta que es un lenguaje orientado a objetos que permite utilizar estructuras de datos como diccionarios, listas y mucho más. También admite operaciones de computación científica como matrices. Así que, con Python, podrás simplificar las operaciones y hacer que sean mucho más rápidas.

Finalmente, aunque hay muchos más motivos por los que Python y Big data son la combinación perfecta, es importante mencionar que tiene soporte para el procesamiento de datos. Es decir, tiene una función integrada que permite procesar los datos estructurados, no estructurados y semiestructurados.

¿Cómo se usa para el análisis de datos?

La forma más común de usar Python para el análisis de datos es para crear y gestionar rápidamente varias estructuras de datos. Por ejemplo, la biblioteca Panda ofrece una gran cantidad de herramientas para analizar, manipular e incluso representar las estructuras de datos y conjuntos de datos complejos.

Por otro lado, Scikit-Learn es perfecta para el análisis de los datos de redes sociales o campañas de marketing. Con ella, tendrás a tu disposición herramientas avanzadas para hacer analíticas y darles sentido a esos datos. Por ejemplo, para tomar decisiones según el éxito de las campañas o para ver el tipo de usuarios que interactúa con tus redes.

Finalmente, con Python, tú mismo podrás escribir tus propios algoritmos para el análisis de datos e integrarlos en tus herramientas.

4. EN LA PARTE DE VISUALIZACIÓN DE DATOS, DE MOSTRAR DASHBOARDS NOS ENCONTRAMOS CON POWERBI Y TABLEAU ENTRE OTROS. DEBEMOS EXPLICAR QUÉ SON.

PowerBI y Tableau son herramientas de visualización de datos que permiten crear dashboards y presentar información de manera visual y atractiva.

POWER BI

Es una herramienta de análisis de datos y visualización de información desarrollada por Microsoft. Ofrece una amplia gama de funciones para visualizar datos, crear informes y compartirlos con otros usuarios.

Ventajas

- Personalizar temas, formato y diseño
- Flexibilidad de extracción de datos importantes.
- Innovación de crear informes con datos interactivos.
- Multiplataforma de informes optimizados para dispositivos.

Desventajas

- No tiene capacidad de publicar informes con datos asociados.
- Algunos datos pueden quedar fuera de visualizaciones.
- Tiene pocas fuentes de datos.

¿Para qué sirve Power BI?

- Para conectar diferentes orígenes de datos, ya sea desde Salesforce, Excel, Google Analytics y otros tipos de bases de datos que están en la nube.
- Crean y organizan modelos de datos con ayuda de DAX: con un conjunto de funciones, operadores y constantes para poder crear información nueva, algo parecido a las fórmulas de Excel.
- Se crean informes personalizados integrando varios objetos visuales para manejar los datos modernos y crear gráficas interactivas.
- Se puede hacer pronósticos con los datos que se han trabajado; de esta manera también se pueden ver tendencias y oportunidades de algunos negocios.
- Se construyen informes interactivos paginados, esto se refiere a que la información va a acomodarse en un tamaño de página impresa, a lo que sería más fácil leer la página.

TABLEAU

Tableau es un tipo de plataforma que permite tener un campo visual de los datos a utilizar para resolver un problema, además permite a los usuarios sacar el máximo de dichos datos.

Tableau ayuda a ver y comprender datos ya sean estudiantes, docentes, analistas, científicos lo que hace esto una plataforma eficaz, flexible y segura.

La parte esencial de Tableau es el Desktop. Esto permite ver, analizar y visualizar los datos a través de un panel dentro de la plataforma. En dicha plataforma se utilizan gráficos o imágenes para que los usuarios puedan comprender la utilización de sus datos. En la aplicación, puedes introducir las filas o columnas que deseas observar y la plataforma te pinta el gráfico de los datos que hayas introducido. Esto hace que la plataforma sea asequible para cualquier persona que no sepa hacer código ya que no requiere que muestres por consola el gráfico mediante la consola de comandos.

Ventajas

- Es intuitivo y fácil: La aplicación ha sido creada de manera flexible, intuitiva y sencilla. Esto significa que no será necesario que poseas conocimientos de programación para usar la aplicación.
- Trabaja con múltiples fuentes de datos: En Tableau puedes extraer, transformar y subir datos a través de diferentes archivos tales como: Excel, PDF, CSV o incluso de gran dificultad como Google Analytics.
- Saca el máximo provecho de la aplicación: Tableau analiza, manipula y almacena datos desde diferentes herramientas. Dichas herramientas como el dashboard sacan el máximo rendimiento de los datos.
- Conecta tus datos en instantes: A través de aplicaciones como Tableau Online o Tableau Reader permiten que los datos sean transferidos en tiempo real.

Desventajas

- No se pueden utilizar estructuras tales como COUNTD y MEDIAN por lo que hay que cambiar el formato en la aplicación.
- A medida que uses los mezcladores de conexiones, Tableau perderá dinamismo en los filtros.
- Solo se pueden compartir los filtros con la misma clave primaria

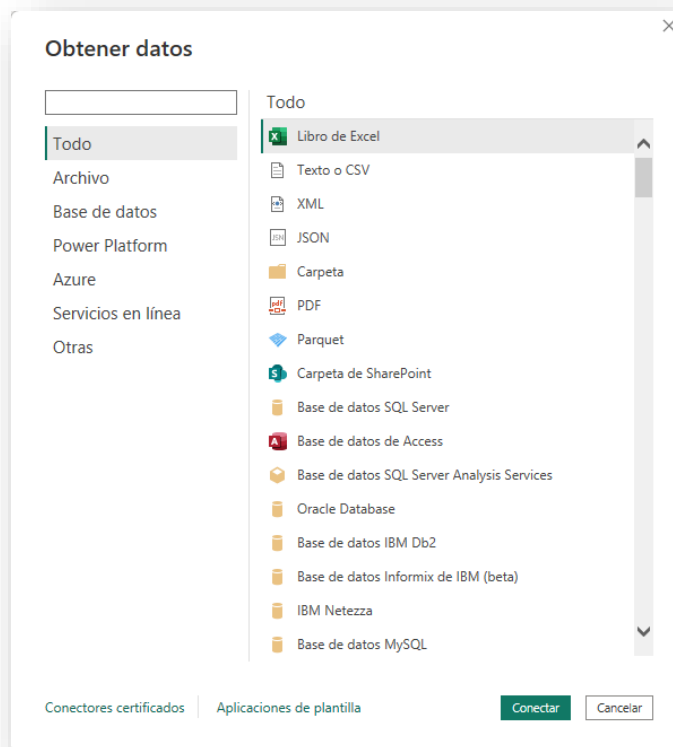
FASE 2

EN ESTA SEGUNDA FASE SE REALIZA LA IMPLEMENTACIÓN DE LA INVESTIGACIÓN. EN CONCRETO SERÍA ACCEDER A UN VOLUMEN DE DATOS Y MOSTRARLO. PODRÍAMOS UTILIZAR SCALA O PYTHON Y MOSTRAR EL RESULTADO EN POWERBI O TABLEAU. LA IDEA ES QUE SEA ALGO MUY IMPACTANTE POR LA CALIDAD DE CONTENIDO TRATADO, VELOCIDAD DE ACCESO, VOLUMEN DE DATOS....

Prueba 1

Para mostrar datos en una gráfica de Power BI, lo primero que debemos hacer es descargar un CSV. En nuestro caso hemos escogido un documento de emisiones de gases invernadero por región.

Dentro de Power BI, hemos escogido la opción Obtener datos, y seleccionado el CSV que previamente hemos descargado.



Y cargamos todos los datos:

greenhouse-gas-emissions-by-region-industry-and-household-year-ended-2021-csv.csv

Origen de archivo: 1252: Europeo occidental (Windows) | Delimitador: Coma | Detección del tipo de datos: Basado en las primeras 200 filas

region	anzsic_descriptor	anzsic_descriptor2	category	gas	units	me
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Auckland	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d
Bay of Plenty	Forestry and logging, fishing, and agricultural support s...	Total	Total	Carbon dioxide equivalents	Kilotonnes	Carbon d

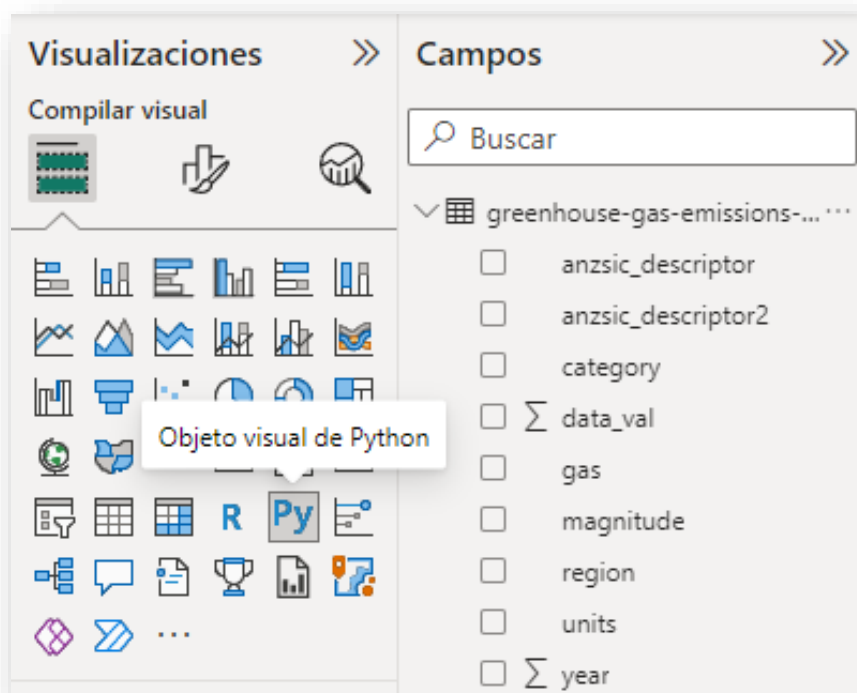
Extraer tabla mediante ejemplos | Cargar | Transformar datos | Cancelar

Escribimos el código para mostrar la gráfica.

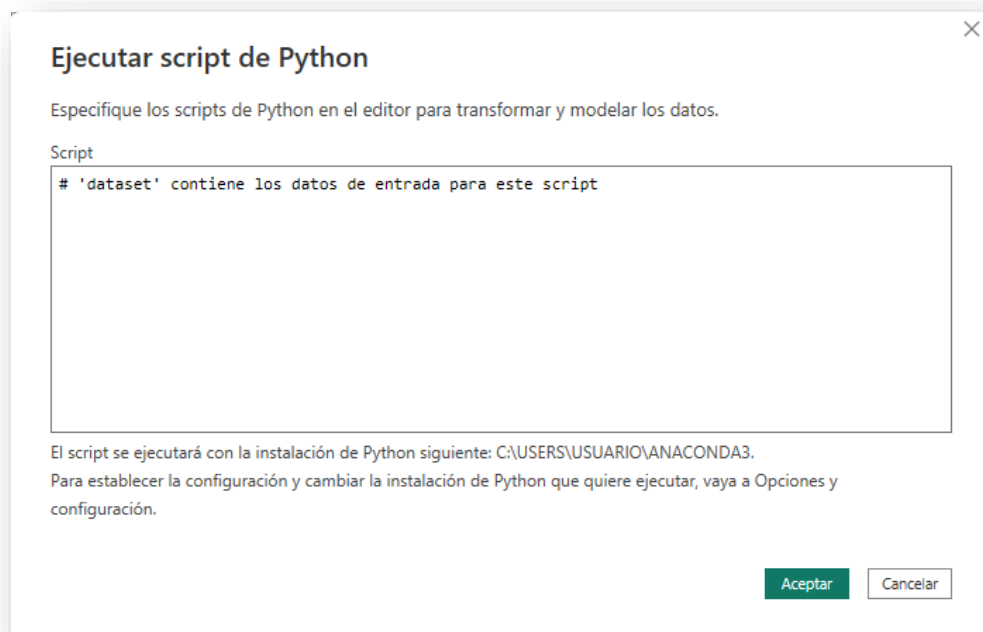
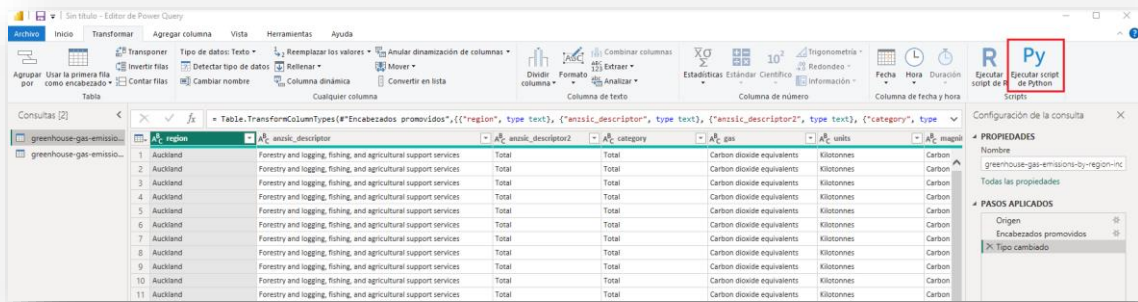
```
1 # El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como un preambulo
2
3 # dataset = pandas.DataFrame(category, region)
4 # dataset = dataset.drop_duplicates()
5
6 # Pegue o escriba aquí el código de script:
7 import matplotlib.pyplot as plt
8 import pandas as pd
9
10 df = pd.read_csv("D:\PROGRAMACIÓN\greenhouse-gas-emissions-by-region-industry-and-household-year-ended-2021-csv.csv")
11
12 df.plot(kind='bar', x='category', y='region', color='green')
13
14
15 plt.title('Emisiones')
16
17 plt.xlabel('Categoría')
18
19 plt.ylabel('Region')
20
21 plt.show()
```

Prueba 2

Desde la parte de visualizaciones en Power BI, seleccionamos el objeto visual de Python, y cargamos los datos de nuestro CSV.

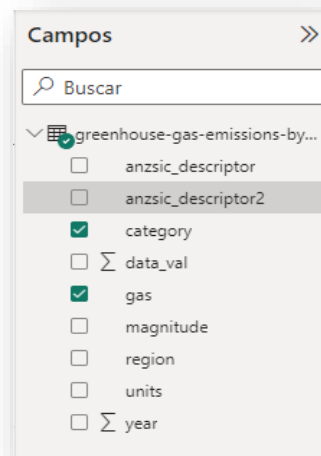


A continuación, seleccionamos Transformar datos, y lo ejecutamos como un script de Python.



Prueba 3

Hemos probado con otros datos y otro código para ver si funcionaba. Los campos eran los siguientes:



Hemos adaptado el siguiente código con los datos de nuestros campos.

```
1 # El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta
2
3 # dataset = pandas.DataFrame(anzsic_descriptor, anzsic_descriptor2, category, data_val, gas)
4 # dataset = dataset.drop_duplicates()
5
6 # Pegue o escriba aquí el código de script:
7 plt.plot(dataset.category., dataset.gas, 'g--')
8 plt.title('Emisiones')
9 plt.xlabel('Categorías')
10 plt.ylabel('Gas')
11 plt.show()
```

FASE 3

PARA FINALIZAR, REALIZAMOS UNA EVALUACIÓN O CONSIDERACIONES DE CÓMO HAN EVOLUCIONADO EL ACCESO A DATOS EN LOS ÚLTIMOS AÑOS. DESDE ACCESO A FICHEROS, PASANDO POR BASE DE DATOS Y CONSUMIENDO APIS.

- Al instalar las librerías pandas y matplotlib desde la consola del ordenador, el comando pip nos daba error:

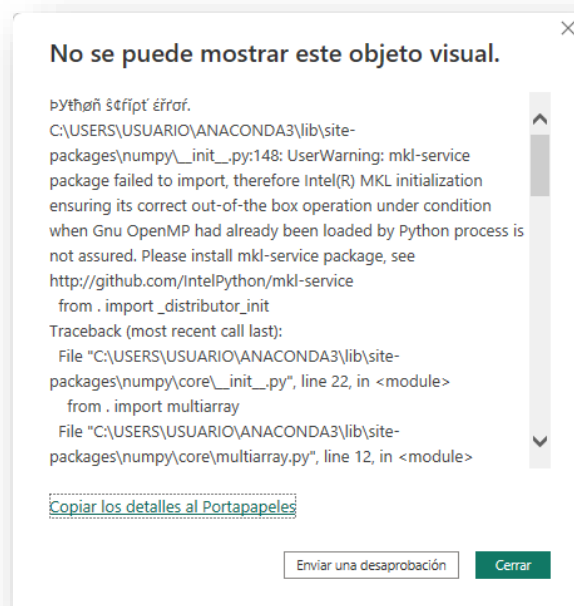
```
Microsoft Windows [Versión 10.0.22621.1105]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>pip install pandas
"pip" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

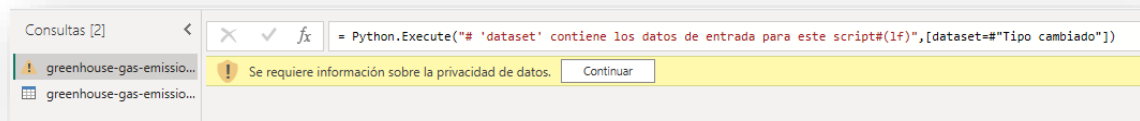
C:\Users\Usuario>pip install matplotlib
"pip" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Usuario>
```

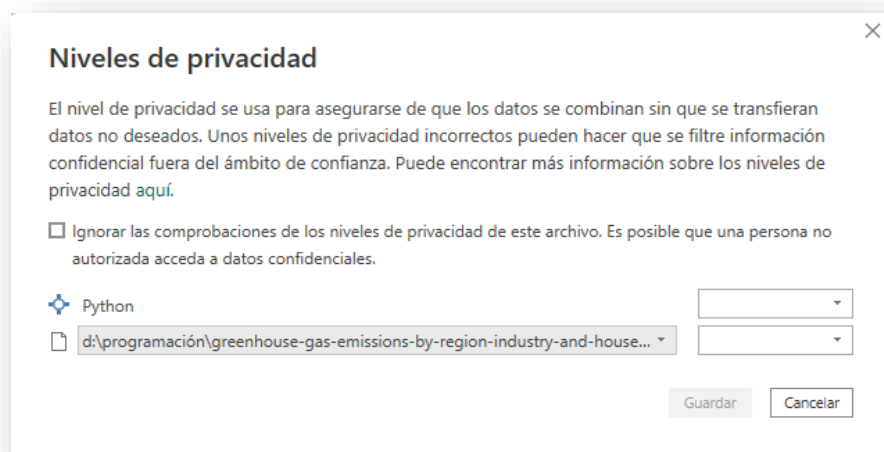
1. Al ejecutar la Prueba 1 nos daba el siguiente error, en el que no nos permitía mostrar la gráfica.



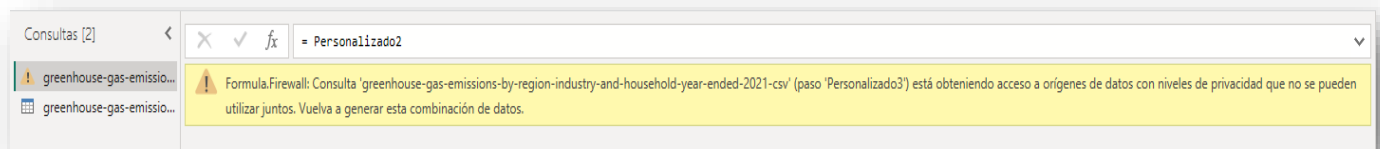
2. En la Prueba 2, una vez ejecutamos el CSV como un script de Python, nos aparece el siguiente error:



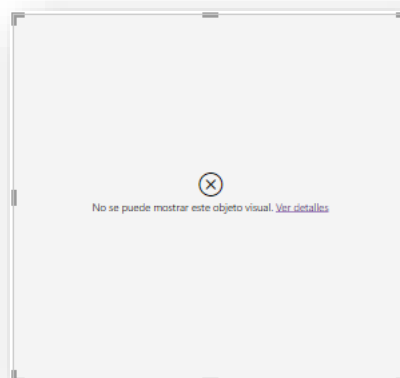
Seleccionamos la ruta del archivo



Y nos apareció otro error.



3. Muestra un mensaje de que no se puede mostrar este objeto visual.



Y nos da un error de sintaxis.

