**Análisis del tráfico en la ciudad de Valencia y su relación con la contaminación atmosférica II. MapReduce, Hadoop y AWS**

**1. Realizar las siguientes consultas**

* 1. **(1 punto).** Escribir una función *mapper* para que lea, directamente de la base de datos de MongoDB, para cada documento de tráfico, el identificador de estación y el valor de tráfico para cada fecha (una línea para cada hora del mes para cada uno de los sensores). Escribir una función *reducer* que calcule, para cada sensor, el valor medio de tráfico del mes; escribir en el fichero de salida primero el valor y luego el identificador del sensor sensor para que se pueda ordenar usando la función tubería y la orden *sort* de Linux. El resultado debe ser el mismo que el de la consulta 2.2 de la parte de MongoDB del proyecto.

*Pon aquí el código de tu software para la función Mapper*

*Pon aquí el código de tu software para la función Reducer*

*Copia y pega aquí la ventana de la terminal con la orden y los resultados*

* 1. **(1 punto).** Usar la función -*appendToFile* para subir todos los ficheros originales de tráfico a HDFS como un solo fichero. Usar la clase MRJob para escribir un código que realice la operación de mapeo y reducción necesaria para obtener el mismo resultado que en la consulta anterior. La solución se debe escribir en HDFS. Desde HDFS muestra las primeras 20 líneas de la solución.

*Haz un copiar-pegar de la ventana de la terminal con la orden desde HDFS para subir los ficheros*

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Escribe un programa en Python para que genere un fichero a partir de la base de datos de MongoDB donde figuren el identificador de tráfico, la fecha y el valor para cada lectura de cada sensor de tráfico, sube este fichero a HDFS (ESTE SERÁ EL FICHERO AL QUE HACE REFERENCIA EL RESTO DE CONSULTAS). Utiliza la clase MRJob para escribir un código que realice la operación de mapeo y reducción sobre el fichero anterior grabado en HDFS necesaria para obtener, para cada sensor de tráfico, los valores de temperatura máximo y mínimo incluyendo la fecha (incluyendo día y hora) exacta en que se produjo cada uno de estos valores. La salida debe ser por la terminal y estar ordenada por sensor y los resultados máximo y mínimo se deben ver a la vez. Como ejemplo, para comprobar la solución, para el sensor A10, el menor valor, 0, se produce el 13 de mayo a las 3:03 (en este caso es el primer cero que encuentra de toda la serie) y el mayor se produce el 3 a las 14:00, con un valor de 3210.

*Pon aquí el código de la función para subir el fichero a HDFS*

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Utiliza la clase MRJob para escribir un código que realice la operación de mapeo y reducción, sobre el fichero de tráfico grabado en HDFS, necesaria para obtener, para cada sensor de tráfico y día del mes, el valor medio de tráfico. Graba la salida en un directorio de HDFS, pásala a local y muestra los valores para los días del mes de la estación “A111”. El resultado debe ser el mismo que el de la consulta 2.3 de MongoDB.

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Haz un copiar-pegar de la ventana de la terminal con la orden desde HDFS para pasar el fichero de salida a local*

*Copia y pega la ventana del editor de texto con la solución*

* 1. **(1 punto).** Añadir un paso (step) al código de la consulta anterior para saber, para cada sensor de tráfico, qué día ha sido el de mayor valor medio de tráfico. La salida debe ser por la terminal ordenada por sensor. Como ejemplo, para comprobar la solución, para el sensor A102, el día es el 17 y el valor es 318.26.

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Escribe un código en Python para que lea directamente de la base de datos de MongoDB, para cada documento de estación meteorológica, el identificador de estación, la fecha y el valor de NO2 para cada fecha (una línea para cada hora del mes para cada una de las estaciones). En el mismo código subir este fichero a HDFS. Desde HDFS lee las primeras líneas del fichero generado.

*Pon aquí el código de la función MapReduce*

*Copia y pega la ventana de la terminal con la lectura del fichero desde HDFS*

* 1. **(1 punto).** Usando la clase MRJob, escribir un código similar al de la consulta 1.5 para saber, para cada estación, qué día ha sido el de mayor valor medio de NO2, ordenando la salida por estación meteorológica. Como ejemplo, para comprobar la solución, para la estación Boulevard, el día es el 5 y el valor de NO2 es 21.875.

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Usa la clase MRJob para escribir el código que lea el fichero de tráfico guardado en HDFS y genere, para cada día de la semana, el valor medio de tráfico considerando todos los sensores de tráfico. Obtener el resultado en la terminal ordenándolo por día de la semana. Como ejemplo, para comprobar la solución, el lunes (día 0) el resultado es 812.769.

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Obtén el mismo resultado que la consulta anterior, pero para el fichero de las estaciones meteorológicas grabado en HDFS. Como ejemplo, para comprobar la solución, el lunes (día 0) el resultado es 13.78.

*Pon aquí el código de la función MapReduce*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código*

*Copia y pega la ventana de la terminal con la solución*

* 1. **(1 punto).** Realizar la consulta anterior subiendo el fichero de estaciones a S3 usando la librería de Python *BOTO3*, generando un clúster en EMR dentro de Amazon AWS. Guardar en resultado en S3, descargarlo y mostrar el resultado. Cuando acabes recuerda revisar que el clúster generado se ha borrado y elimina los ficheros de S3 para que no sigas incurriendo en gastos.

*Pon aquí el código de tu software para pasar a S3 con BOTO3*

*Haz un copiar-pegar de la ventana de la terminal con la orden de ejecución del código y el final de la ejecución que debe mostrar que se ha efectuado correctamente*

*Copia y pega la ventana del editor de texto con la solución*

**Conclusiones (Evaluación de la competencia de responsabilidad y toma de decisiones)**

Se deberán presentar las conclusiones relativas a los datos con los que se ha trabajado (precisión, granularidad -resolución espacial y temporal-, confusión, valor, validez para los objetivos perseguidos, etc.), la plataforma o framework usado para su almacenamiento y/o análisis tanto de forma local como en la nube (facilidad de uso, generación de resultados, etc.), de los resultados obtenidos (correlaciones observadas, consecuencias, predicciones, agrupaciones, etc.) y sobre la bibliografía consultada por el alumno.

**Bibliografía (Evaluación de la competencia de responsabilidad y toma de decisiones)**

Se deberá presentar la bibliografía correspondiente que tendrá que contener todas las búsquedas realizadas por el alumno, tanto de carácter teórico en libros y artículos como de carácter técnico (búsqueda en internet de soluciones para la correcta instalación de las herramientas, para la generación de código -p.e. en blogs, plataforma Stack Overflow, información libre de diferentes cursos, etc.), todas estas referencias deberán estar adecuadamente referenciadas en el texto del proyecto de forma que no haya bibliografía sin su correspondiente referencia en el texto o en el código elaborado. Por último, en el apartado de conclusiones debe haber una parte sobre la bibliografía usada donde el alumno deberá considerar la importancia, fiabilidad y sencillez de uso o implementación de las diferentes fuentes usadas.