2021 - 2022

ACTIVIDAD GUIADA 1

Máster Universitario en Big Data y Ciencia de Datos

03MBID – Procesamiento de datos masivos

Curso 2021-2022 – Ed. octubre

viu Universidad Internacional de Valencia

1 . Descripción general

Actividades Guiadas

DESCRIPCIÓN	
Introducción	El procesamiento masivo de datos no siempre se puede realizar con tecnologías tradicionales. En muchas ocasiones se tienen que utilizar tecnologías Big Data como Hadoop MapReduce y Spark.
Objetivo	Conocer el modelo de procesamiento MapReduce y las principales herramientas Big Data. Desarrollar programas Big Data utilizando los frameworks Hadoop MapReduce y Spark según lo que diga el enunciado.
Trabajo previo	Lectura del material docente de la parte específica que se encuentra disponible desde el comienzo del curso en la carpeta: Recursos y materiales>1. Materiales docentes Visualización de las videoconferencias sobre instalación del entorno, ecosistema Hadoop y ecosistema Spark que se encontrarán disponibles en: Videoconferencias>grabaciones
Metodología	En las videoconferencias teóricas se expondrá al alumno conocimientos, material e indicaciones suficientes para que pueda elaborar una unidad didáctica basada en el aprendizaje y enseñanza por competencias en matemáticas e informática. En la videoconferencia de actividad guiada se establecerá las pautas concretas y la dinámica que el alumnado deberá seguir para realizar la actividad propuesta. Las actividades se centrarán en poner en práctica y asentar los conocimientos adquiridos en la videoconferencia teórica anterior.
Tarea para el e-portfolio	Desarrollo de 3 programas Big data utilizando los frameworks que se indican en los siguientes enunciados. **Programa 1**: Desarrollar un programa utilizando el framework Hadoop MapReduce con java o con Hadoop Streaming. Dado un dataset que contenga entradas con la forma “cliente;dineroGastado”. Crea un programa llamado clientesQueGastanMucho que indique los clientes que gastaron en total más de 1000 euros. Se valorará positivamente la optimización del programa, por ejemplo a través de la funcionalidad Combiner. Ejemplo: Entrada Salida

Alice;100 Alice
Alice;950
Bob;150
Bob;700
Alice;300
Carol;200

Notar que Alice gastó más de 1000 euros (100+950+300), por ello figura en la salida. En cambio, ni Bob ni Carol gastaron más de 1000 euros, 850 y 200 euros respectivamente, y por ello no figuran en la salida. El programa debería funcionar para cualquier gasto y número de personas.

Programa 2: Desarrollar un programa utilizando el framework Spark con la API RDD y los lenguajes python o java.

Dado un dataset que contenga entrada con la forma "persona;método_pago;dinero_gastado". Crea un único programa llamado personaYMetodosDePago que:

a) Por cada persona indique en cuántas compras pagó más de 1500 euros con tarjeta de crédito. La solución se tiene que guardar en un archivo comprasCreditoMayorDe1500.
b) Por cada persona indique en cuántas compras pagó menos de 1500 euros con tarjeta de crédito. La solución se tiene que guardar en un archivo comprasCreditoMenorDe1500

Se valorará positivamente la optimización del programa, por ejemplo a través de la funcionalidad Combiner o el equivalente de Spark.

Ejemplo:

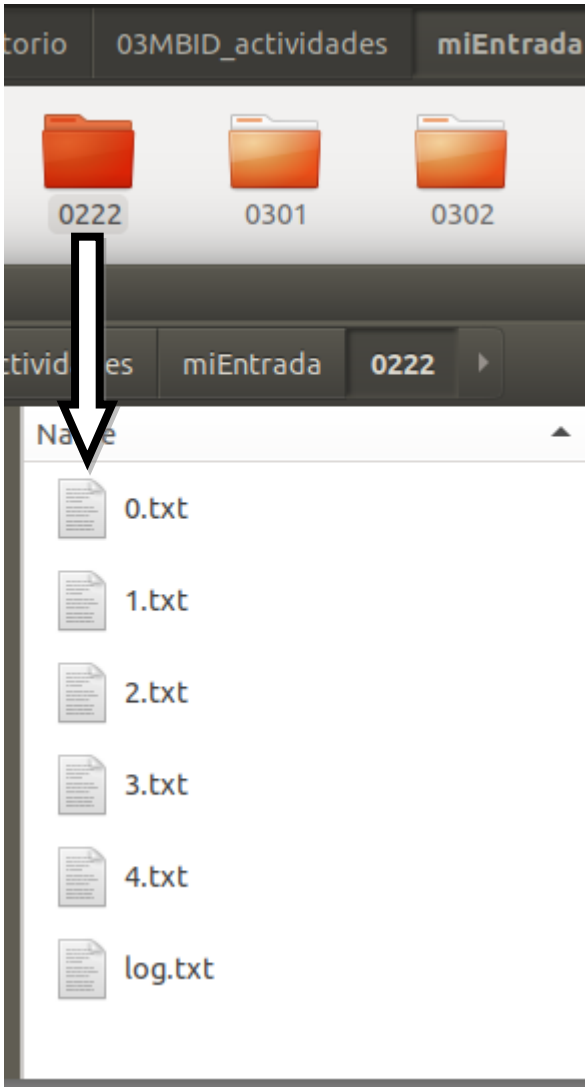
Entrada	Salida (a)	Salida (b)
Alice;Tarjeta de crédito;1000	Alice;2	Alice;1
Alice;Tarjeta de crédito;1800	Bob;0	Bob;0
Alice;Tarjeta de crédito;2100		
Bob;Bizum;2000		

Notar que si bien Bob hace una compra superior a 1500 euros, no la hace con tarjeta de crédito.

Programa 3: Desarrollar un programa utilizando el framework Spark con la API RDD y los lenguajes python o java.

Dado un dataset que contiene información sobre los videos de Youtube (https://netsg.cs.sfu.ca/youtubedata/), crear un programa llamado CategoriaDeVideosMasVista que obtenga cuál es la categoría de videos más vista de la plataforma Youtube y el número total de visualizaciones que hay en esa categoría. El programa debe recibir dos parámetros de entrada: la carpeta en la que está el dataset y la carpeta en la que se

guardará el resultado. En la carpeta donde está el dataset se tienen que descomprimir ALGUNO de los archivos 0222.zip, 0301.zip, etc. **Importante**: si la persona que hace la actividad dispone de pocos recursos computaciones, entonces se recomienda que únicamente descomprima algún .zip pequeño para que pueda desarrollar el programa. La carpeta de datos de entrada debería quedar como se ve a continuación:



Los datos de entrada están en los archivos 0.txt, 1.txt, etc y cada fila contiene la información de un video tabulada con el siguiente formato: id del video de youtube, usuario que subió el video, número de días desde que se subió el video y la fecha en la que obtuvieron los datos, categoría del video, longitud del video, número de visitas del video, puntuación del video, número de puntuaciones del video, número de comentarios del video, y una lista de ids de videos relacionados.

Se valorará positivamente la optimización del programa, por ejemplo a través de la funcionalidad Combiner o el equivalente de Spark.

Ejemplo:

<table>
<tr><td></td><td>Entrada
… Gadgets & Games … 30
… Gadgets & Games … 10
… Music … 90
… Sports … 20
… Music … 50
… Gadgets & Games … 95

Salida
Music;140

Notar que la categoría “Music” es la que más visitas tiene: 90 en un video + 50 en otro video, es decir, en total 140 visitas. Por otro lado, la categoría “Gadgets & Games” tiene menos visitas: en total 135 visitas obtenidas de 30 + 10 + 95. Y la categoría que menos visitas tiene es Sports con sólo 20 visitas.

El programa debe funcionar independientemente del número de categorías y para cualquier cantidad de filas que se pueda llegar a tener.</td></tr>
<tr><td>Forma de entrega</td><td>La memoria de cada actividad se subirá al site de la asignatura en formato zip (ningún otro formato será admitido). La actividad debe tener 3 carpetas, una por cada programa, que contengan: el programa con un comentario que incluya tanto el nombre del autor y como el comando/instrucciones para ejecutar ese programa. Si el comentario/instrucciones no se pueden poner como comentario dentro del programa, también se puede subir en un documento pdf.

Las memorias se realizarán de manera individual y cada alumno es responsable de subir a la plataforma la actividad.

En cualquier caso, las entregas se realizarán dentro de los plazos establecidos en el calendario de la asignatura en 1» o 2» Convocatoria.

Las entregas sólo serán válidas si se realizan a través del site de la asignatura:
• Actividades>Actividad 1: Programación</td></tr>
<tr><td colspan="2">Fecha de entrega</td></tr>
<tr><td>1ª Convocatoria</td><td>23/01/2022 hasta las 23:59</td></tr>
<tr><td>2ª Convocatoria</td><td>03/04/2022 hasta las 23:59</td></tr>
</table>

RÚBRICA DE EVALUACIÓN DE LAS ACTIVIDADES GUIADAS (20%)		
	Adecuación de la funcionalidad al enunciado (65%)	**Optimización y diseño del programa (35%)**
Muy competente	Los programas hacen exactamente lo que se pide en el enunciado	Los programas implementan funcionalidades de optimización, son escalables y están bien diseñados.
Competente	Los programas suelen hacer lo que se pide exactamente en el enunciado	Los programas suelen implementar funcionalidades de optimización, son escalables y están más o menos bien diseñados.
Aceptable	Los programas sólo a veces hacen lo que se pide exactamente en el enunciado	Los programas sólo a veces implementan funcionalidades de optimización, son escalables y están más o menos bien diseñados.
Aún no Competente	Los programas no suelen hacer lo que se pide exactamente en el enunciado	Los programas o bien no implementan funcionalidades de optimización, o bien no son escalables y o bien no están correctamente diseñados.

viu .es