# GUÍA PRÁCTICA

**Nombre:**

**Paul Rodríguez**

**Carrera:**

**Big Data**

**Materia:**

**Marcos de Referencia**

**Ciclo:**

**M3A**

**Fecha:**

**24/04/2024**

**Periodo:**

**Abril - Agosto**

## 1. Datos Generales

| | |
|---|---|
| **Carrera:** | **Tecnología Superior en Big Data** |
| **Período académico:** | **Abril – Agosto 2023** |
| **Asignatura:** | **Marcos de Referencia para Big Data** |
| **Unidad Nº:** | **2** |
| **Tema:** | **Introducción a la Big Data con Python.** |
| **Ciclo-Paralelo:** | **M3A** |
| **Fecha de inicio de la Unidad:** | |
| **Fecha de fin de la Unidad** | |
| **Práctica Nº:** | **2** |
| **Horas:** | **10** |
| **Docente:** | **Ing. Verónica Chimbo. Mgtr.** |

## 2. Contenido

### 2.1 Introducción

Es un hecho natural que cada día generamos más y más datos, y que su captura, almacenamiento y procesamiento son piezas fundamentales en una gran variedad de situaciones, ya sean de ámbito empresarial o con la finalidad de realizar algún tipo de investigación científica.

Para conseguir estos objetivos, es necesario habilitar un conjunto de tecnologías que permitan llevar a cabo todas las tareas necesarias en el proceso de análisis de grandes volúmenes de información o big data. Estas tecnologías impactan en casi todas las areas de las tecnologías de la información y comunicaciones, también conocidas como TIC. Desde el desarrollo de nuevos sistemas de almacenamiento de datos —como serían las memorias de estado sólido o SSD (Solid State Disk), que permiten acceder de forma eficiente a grandes conjuntos de datos— o el desarrollo de redes de computadores más rápidas y eficientes —basadas por ejemplo, en fibra óptica, que permiten compartir

gran cantidad de datos entre multiples servidores—, hasta nuevas metodologías de programación que permiten a los desarrolladores e investigadores usar estos nuevos componentes de hardware de una forma relativamente sencilla.

### 2.2 Objetivo de la Guía

1. Comprender los diferentes componentes hardware de una arquitectura de big data.
2. Conocer el stack de software típico de gestión de una arquitectura de big data.
3. Entender cómo se almacenan y distribuyen los datos masivos en un sistema de archivos distribuido.
4. Entender las diferentes jerarquías de memoria para poder procesar datos masivos de forma eficiente.

5. Ser capaz de diferenciar los diferentes tipos de procesamiento distribuido: modelo batch (por lotes) frente al modelo streaming (secuencial).

### 2.3 Materiales, herramientas, equipos y software

Computador personal, Google Colab.

### 2.4 Procedimiento

Para instalar Hadoop 3.5.5 en Windows, sigue los siguientes pasos:

1. **Requisitos previos:**
- Asegúrate de tener instalada una versión de Java (Java Development Kit, JDK) compatible con Hadoop. Puedes descargar JDK desde el sitio web de Oracle.
- Configura la variable de entorno **JAVA_HOME** para apuntar al directorio de instalación de JDK.
2. **Descarga Hadoop:**
- Ve al sitio web de Apache Hadoop (https://hadoop.apache.org/releases.html) y busca la versión 3.5.5.
- Descarga el archivo binario hadoop-3.5.5.tar.gz.
3. **Descomprime el archivo:**
- Crea una carpeta en tu sistema donde deseas instalar Hadoop.
- Descomprime el archivo hadoop-3.5.5.tar.gz en la carpeta que has creado.
4. **Configuración de Hadoop:**
- En la carpeta de instalación de Hadoop, abre el archivo etc/hadoop/hadoop-env.cmd en un editor de texto.
- Establece la variable de entorno JAVA_HOME en la ubicación de tu instalación de JDK:

```
set JAVA_HOME=C:\ruta\a\JDK
```

- Guarda los cambios y cierra el archivo.
5. **Configuración del archivo de configuración de Hadoop:**
- En la carpeta de instalación de Hadoop, abre el archivo etc/hadoop/core-site.xml en un editor de texto.
- Añade la siguiente configuración para definir la ubicación del sistema de archivos Hadoop (HDFS):

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

- Guarda los cambios y cierra el archivo.

6. Configuración del archivo de configuración de Hadoop:

- En la carpeta de instalación de Hadoop, abre el archivo etc/hadoop/hdfs-site.xml en un editor de texto.
- Añade la siguiente configuración para definir la ubicación de almacenamiento de datos de Hadoop (HDFS):

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

- Guarda los cambios y cierra el archivo.

**7. Configuración de los archivos de configuración de Hadoop:**

- Copia los archivos etc/hadoop/core-site.xml y etc/hadoop/hdfs-site.xml en la carpeta etc/hadoop y pégalo en la carpeta etc/hadoop en la carpeta de instalación de Hadoop.
- Configuración del archivo de configuración de Hadoop:

- En la carpeta de instalación de Hadoop, abre el archivo etc/hadoop/mapred-site.xml en un editor de texto.
- Añade la siguiente configuración para definir el framework de ejecución de MapReduce:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- Guarda los cambios y cierra el archivo.

  Configuración del archivo de configuración de Hadoop:

- En la carpeta de instalación de Hadoop, abre el archivo etc/hadoop/yarn-site.xml en un editor de texto.

  Añade la siguiente configuración para definir la capacidad de recursos del clúster YARN:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class<
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

- Guarda los cambios y cierra el archivo.

1. **Configuración de la variable de entorno Hadoop:**

- Añade la siguiente variable de entorno a tu sistema:

```
HADOOP_HOME=C:\ruta\a\Hadoop
```

2. **Formatea el sistema de archivos Hadoop (HDFS):**

- Abre una ventana de comandos y navega hasta la carpeta de instalación de Hadoop.
- Ejecuta el siguiente comando para formatear el sistema de archivos Hadoop:

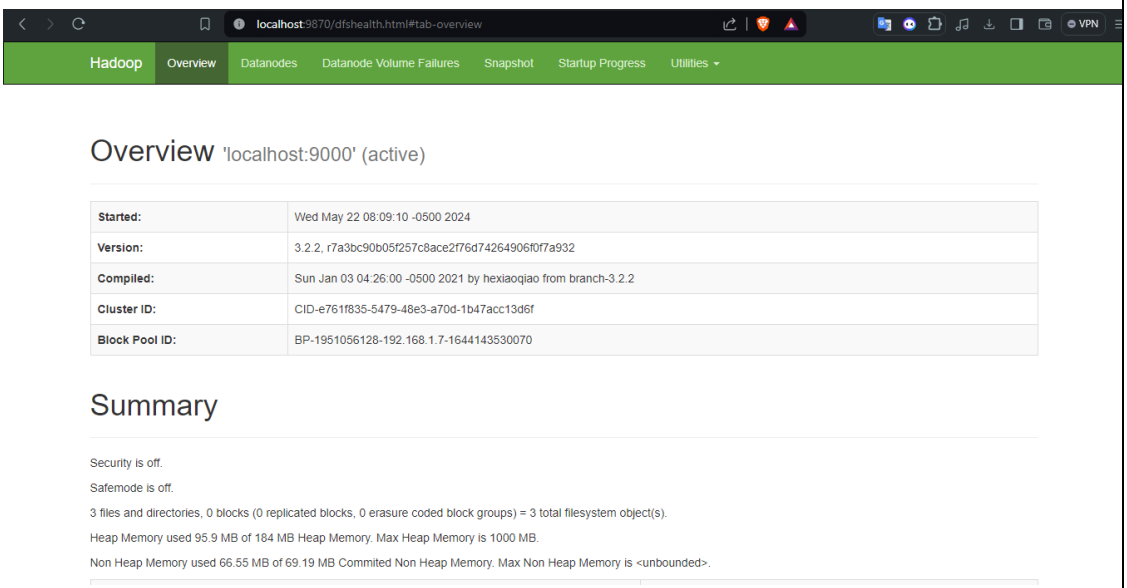
```
bin/hdfs namenode -format
```

3. Inicia los demonios de Hadoop:

- Ejecuta el siguiente comando para iniciar los demonios de Hadoop:

```
sbin/start-dfs.cmd
```

4. Verifica la instalación:
    - Abre un navegador web y ve a la siguiente URL: http://localhost:9870
    - Deberías ver la interfaz web del NameNode de Hadoop.




**Responder a las siguientes preguntas:**

1. **¿Qué es Hadoop y para qué se utiliza?**

   Hadoop es un framework de código abierto que permite el procesamiento distribuido de grandes conjuntos de datos en clústeres de computadoras utilizando modelos de programación sencillos
2. **¿Cuáles son los componentes principales de Hadoop?**
    - Hadoop Distributed File System (HDFS): sistema de archivos distribuido y tolerante a fallos
    - MapReduce: modelo de programación para procesar grandes conjuntos de datos en paralelo
    - Hadoop Common: utilidades compartidas que soportan los otros módulos de Hadoop
3. **¿Cuál es la diferencia entre Hadoop MapReduce y Hadoop Distributed File System (HDFS)?**

HDFS es el sistema de archivos distribuido de Hadoop, mientras que MapReduce es el modelo de programación para procesar datos en paralelo]. HDFS almacena los datos de manera distribuida en los nodos del clúster, mientras que MapReduce divide las tareas en subtareas que se ejecutan en paralelo en los nodos.

4. **¿Cuáles son las ventajas de utilizar Hadoop?**
    - Permite procesar grandes volúmenes de datos estructurados y no estructurados
    - Es escalable, tolerante a fallos y se ejecuta en hardware commodity
    - Tiene un bajo costo de almacenamiento y procesamiento
    - Permite el procesamiento distribuido en paralelo
5. **¿En qué lenguaje está escrito Hadoop?**

    Hadoop está escrito principalmente en Java
6. **¿Qué es un clúster Hadoop?**

    Un clúster Hadoop es un grupo de máquinas (nodos) que trabajan juntas para procesar datos en paralelo utilizando HDFS y MapReduce.
7. **¿Cuál es la diferencia entre un NameNode y un DataNode en Hadoop?**

    El NameNode es el nodo maestro en HDFS que gestiona el espacio de nombres del sistema de archivos y regula el acceso a los archivos por parte de los clientes. Los DataNodes son nodos esclavos que gestionan el almacenamiento adjunto a los nodos y realizan operaciones de lectura/escritura de archivos bajo la coordinación del NameNode.
8. **¿Cómo se maneja la tolerancia a fallos en Hadoop?**

    Hadoop maneja la tolerancia a fallos replicando los datos en múltiples nodos DataNode. Si un nodo falla, HDFS automáticamente redirige las solicitudes a otra réplica.
9. **¿Cuál es la diferencia entre Hadoop 1 y Hadoop 2 (YARN)?**

    Hadoop 2 (con YARN) separa el procesamiento de datos (MapReduce) del gestor de recursos, permitiendo ejecutar otros modelos de procesamiento además de MapReduce. Hadoop 1 solo soportaba MapReduce.
10. **¿Hadoop es adecuado para procesar datos en tiempo real?**

No, Hadoop no es adecuado para procesamiento de datos en tiempo real debido a su arquitectura batch. Herramientas como Apache Storm o Apache Spark Streaming son más apropiadas para streaming de datos en tiempo real.

11. **¿Cuál es el papel de Apache Hive en el ecosistema de Hadoop?**

    Apache Hive es un data warehouse construido sobre Hadoop que permite consultar y gestionar grandes conjuntos de datos almacenados en HDFS y sistemas de archivos compatibles usando un lenguaje similar a SQL llamado HiveQL.

12. **¿Es necesario saber programar para utilizar Hadoop?**

    No es necesario saber programar para utilizar Hadoop, ya que herramientas como Hive permiten interactuar con Hadoop usando SQL. Pero para desarrollar aplicaciones MapReduce personalizadas sí se requieren conocimientos de programación.

13. **¿Hadoop se ejecuta solo en servidores Linux?**

    Hadoop se ejecuta principalmente en sistemas operativos Linux, aunque también es posible ejecutarlo en Windows.

14. **¿Cuáles son algunos casos de uso comunes para Hadoop?**

    - Análisis de grandes volúmenes de datos web y de redes sociales
    - Procesamiento de datos de sensores y dispositivos IoT
    - Análisis de registros de aplicaciones y sistemas
    - Almacenamiento y análisis de datos genómicos

15. **¿Cuál es la diferencia entre Hadoop y Apache Spark?**

    La principal diferencia es que Hadoop utiliza MapReduce como modelo de programación para procesar datos en paralelo, mientras que Apache Spark usa su propio motor de procesamiento en memoria que es más rápido que MapReduce, especialmente para aplicaciones iterativas.

## 2.5 Resultados esperados

Instalación de Hadoop en los pcs para el procesamiento de datos masivos.

## 2.6 Bibliografía

1. Vegas Lozano, Esteban , autor; Universitat Oberta de Catalunya, disponible: http://cvapp.uoc.edu/autors/MostraPDFMaterialAction.do?id=165727
2. Hall, Mark A ; Frank, Eibe ; Witten, Ian H, Data mining: practical machine learning tools and techniques, 2011


## 3. Firmas de Responsabilidad

| ESTUDIANTE | DOCENTE | DIRECTORA DE CARRERA |
|---|---|---|
| Firma | **Nombre: Ing. Verónica Chimbo. Mgtr.**<br>Firma | **Nombre: Ing. Verónica Segarra.**<br>Firma |
| **Fecha:** | **Fecha:** | **Fecha:** |