Preguntas típicas para consultor Big Data (spark, hadoop)

## En que componentes se divide el framework Hadoop?

HDFS, MapReduce, Yarn.

### Explique cada uno de ellos.

HDFS: Hadoop Data File System es el sistema de almacenamiento de Hadoop

MapReduce: Es la unidad de procesamiento de Hadoop.

Yarn: Es la unidad de gestión de recursos de Hadoop.

### ¿Qué es el factor de replicación y para qué sirve?

El factor de replicación es el número de copias que se hace de un archivo/información y sirve para que en caso de error o fallo no perdamos la información.

## Diferencia entre un NameNode y un DataNode.

NameNode es el nodo maestro en HDFS que maneja los metadatos mientras que el DataNode es el nodo esclavo que guarda la información tal y como se indica en el NameNode.

## ¿Qué es Big Data?

Es el conjunto de tecnologías que sirven para recopilar, analizar y gestionar cantidades ingentes de datos para procesarlos e identificar patrones.

## ¿Cuáles son las V’s que definen Big Data? (Menciona 4)

Volumen, Velocidad, Variedad, Veracidad y Valor.

## ¿Qué es Spark?

Spark es un framework de programación para procesar datos distribuidos.

### En que se diferencian Spark con Hadoop MapReduce

Spark puede ser entre 10 y 100 veces más rápido, permite Streaming, Machine Learning y análisis complejos y procesa en memoria mientras MapReduce procesa en el disco local.

### ¿Qué es un RDD?

Es la forma básica de datos en Spark que representa una colección de elementos inmutable que se puede operar en paralelo.

#### RDD vs DataFrame vs Dataset

Dataset es una extensión de DataFrame y un DataFrame es como un RDD pero con los datos ya organizados.

#### ¿Cuándo se debería utilizar RDD en vez de Dataset (o DataFrame)?

Cuando necesitemos funcionalidad de bajo nivel y control.

### Existen 2 tipos de operaciones básicas que se pueden realizar con los RDD y DataFrames, menciónalas y define cada una.

Transformaciones: Producen un nuevo RDD de uno ya existente, pero solo cuando una acción sea ejecutada.

Acciones: Son operaciones que devuelven resultados no RDDs.

### ¿Qué diferencia existe entre las narrow y wide transformations? Menciona un ejemplo de cada una.

Narrow Transformation: Solo es necesaria una sola partición para cada dato transformado.

Wide Transformation: Los resultados de una partición pueden encontrarse en diferentes particiones de su padre.

### ¿Qué es un broadcast variable y para qué se utilizan?

Son variables compartidas de solo lectura que están en caché y disponibles en todos los nodos para ser usadas para cualquier tarea.

### ¿Qué son los accumulators?

Son variables usadas para agregar información a través de los nodos (para hacer conteos por ejemplo)

### ¿Qué es Data shuffle/shuffling? ¿Por qué ocurre? ¿Cómo evitarlo?

Es el proceso por el cual se redistribuye los datos entre las particiones entre etapas. Se puede evitar usando reduceByKey o combineByKey en vez de groupByKey.

### Tipos de persistencia de un RDD/DataFrame/DataSet - persist().

Solo memoria, memoria y disco, solo memoria serializable, memoria y disco serializable y solo disco

### ¿Cuál es la diferencia entre cache y persist?

Persist permite más control sobre cómo y dónde es guardada la información.

### ¿Qué es checkpointing?

Es un proceso por el cual se crean streams tolerantes a fallos con un sistema de escritura de records.

### Spark SQL vs DataFrame/DataSet

No hay diferencia de ejecución entre ambas. DataFrames/DataSets suelen resultar más claros visualmente.

### ¿Cuántas APIs distintas (por lenguaje de programación) posee Spark?

Python, R, Java, Scala y SQL

### Si tuvieras que hacer un trabajo de limpieza en ficheros de texto (muy grandes) con que API de spark lo harías y por qué

Los DF de Scala o Python, ya que ambos se ejecutan igual y resultan en la misma performance.

### ¿Qué es el Catalyst Optimizer?

<https://blog.bi-geek.com/spark-sql-optimizador-catalyst/>

### ¿Qué es project tungsten?

<https://community.cloudera.com/t5/Community-Articles/What-is-Tungsten-for-Apache-Spark/ta-p/248445>

### ¿Qué es un ejecutor (executor)?

Son los encargados de correr las tareas y devolver el resultado.

### ¿Por qué el rendimiento de Spark Scala es mejor que el de PySpark?

<https://medium.com/@ketanvatsalya/a-scenic-route-through-pyspark-internals-feaf74ed660d>

## Tipos de formatos de datos columnares

### Ventajas/Desventajas

Los datos se pueden comprimir fácilmente y las operaciones se pueden realizar más rápido. Como desventaja las bbdd columnares no reemplazan otras bbdd, solo las complementan.

## ¿En qué consiste el particionamiento de los datos a nivel físico (hdfs) y para qué se utiliza?

Consiste en dividir los datos para así mejorar mantenimiento, performance y gestión de estos.

## ¿Qué es Hive?

### Características

### ¿Qué es el particionado y el bucketing?

## ¿Qué es Impala?

### ¿En qué se diferencian Hive e Impala?

### ¿Cuándo utilizarías cada uno de estos?

## ¿Qué es sqoop? ¿Para qué se utiliza?

### Sqoop vs Hive

### Sqoop vs Spark

### Sqoop1 vs Sqoop2

## ¿Qué es cloudera?

## ¿Qué es Hue? ¿Qué es Ambari?

### ¿Para qué se utilizan?

## ¿Qué es una función lambda?

## ¿Qué es una tupla?

## ¿Qué es programación orientada a objetos?

### ¿Qué sabes de la programación orientada a objetos?

### ¿Qué es la herencia de clases? Ejemplos

### ¿Qué es el polimorfismo? Ejemplos

## ¿Qué es programación funcional?

<https://blog.nearsoftjobs.com/programaci%C3%B3n-funcional-con-scala-4708b6b56e30>

### ¿Qué has usado/aplicado de programación funcional?

### Características de la programación funcional.

### ¿Qué son las high order functions?

### ¿Qué son las monads?

### ¿Qué es currying?

<https://youtu.be/psmu_VAuiag>

## ¿Qué son las funciones lambda?

## ¿Cuáles son las principales características del lenguaje Scala?

### ¿Qué es una case class?

### ¿Qué son los traits?

### ¿Qué es un singleton object y cómo se definen?

### ¿Qué es un companion object y cómo se definen?

### ¿Qué son las clases implícitas (implicit classes)?

## Diferencia entre Clase y Objeto (indistintamente de lenguaje de programación)

## ¿Qué es Jenkins y para qué se utiliza?

## ¿Qué es Git y para que se utiliza?

### ¿Qué es una rama?

### ¿Cuáles son las principales que crearías para tu proyecto?

### ¿Qué es un pull request/merge request?

### ¿Cómo trabajas en Git?

<https://nvie.com/posts/a-successful-git-branching-model/>