1. **“Tenemos un proyecto de Big Data, porque tenemos mucha información” (data minning, analytics, big data). Como se sabe que es un proyecto de big data.**
   1. **¿Qué preguntas harías para darte cuenta? ¿Qué mirarías?**

*3 V’s de big data. Depende del proyecto y estructura de la empresa*

Volumen: No entran en un disco/rack. Hay que partir los datos para poder analizarlos. Pasa a ser un sistema distribuido.

Velocidad: Un sistema convencional no llega a procesarlos/consumirlos. Hay que distribuir el procesamiento.

Variedad: Tipos de datos.

1. **Plataformas de Big Data**
   1. **Hadoop**
   2. **Spark**
2. **Que es Hadoop. Beneficios / Para que sirve**
3. **Componentes**
4. **High Availability – porque/como**
5. **Fault Tolerant – porque/como – HDFS & YARN– Como se detectan los fallos, como se recuperan**
6. **Es fault tolerant, pero tiene una gran contra… SPOF - NN**
7. **Standby NN, Secondary NN**
8. **Diferencias H1 vs H2**
9. **MR – Como corre un job**
   1. **Donde se guardan los resultados intermedios? En el FS local**
   2. **Que es un combiner? Cuand se usa?**
   3. **Como se elige el mapper? Data Locality**
10. **HDFS – data write (subir un archivo a HDFS)**
    1. **Que interfaz hay? CLI, simil POSIX**
    2. **Default block size**
    3. **Si archivo tiene tamaño más pequeño que el block size?**
    4. **Problema de (muchos) archivos pequeños? Overhead muy grande**
    5. **Data Read**
    6. **HDFS, acceso a dos usuarios al mismo tiempo, que pasa?**
11. **YARN**
    1. **Funciona solo con MR?**
    2. **Como se ejecutan los Jobs en yarn?**
12. **Contras de Hadoop**
    1. **Archivos pequeños**
    2. **Datos relacionales – BD relacionales**
    3. **Ad hoc queries**
    4. **Joins**
    5. **Online analytics**
    6. **Random read, random update no se pueden 🡪 es append only**
13. **HBase, para accesos randoms a files**
14. **Para qué Hive/Pig? Similitudes, Diferencias**

* **Son ambos abstracciones de MR (Hive/Pig por debajo tienen MR/spark/tez)**
* **Presentan interfaces a documentos**
* **Para Hive necesitas un esquema, para Pig no.**
* **Para Pig, el acceso en la información, es procedural**
* **Hive tiene interfaz simil SQL, a nivel tabla, a nivel batch**
* **Hive para acceso a la información, creas tablas externas y accedes a la info que necesitas, datawarehouse capabilities sobre Hadoop**
* **Tabla internal se borra la data y la info del metastore**
* **Pig para programación, para ETLs, limpieza, para acceso a toda la información**
  1. **Que es el metastore, para que sirve**
  2. **es HCatalog 🡪 tiene el archivo de db externos con las que tiene que interactuar**
  3. **Diferencias Tabla interna / tabla externa**
  4. **Que tipo de datos soporta Hive 🡪 para serializar, deserealizar (serder, avro)**
     1. **Avro, que es? (Pig no lo soporta) para que sirve?**

1. **Plataforma Spark vs Hadoop, diferencias, cuando usar cada una?**

* **Para procesamiento distribuido paralelo.**
* **Spark guarda info en memoria de cálculos intermedios**
* **Spark usa YARN como resource manager (necesita algunos componentes de Hadoop para instalar)**
* **Hadoop los guarda en HDFS**
  + **Cuando se usa cada uno? Tipos de proyectos**
    - **Para datos en streaming (micro batching) 🡪 Spark**
    - **Para procesamiento batch, y para el procesamiento de toda la info 🡪 Hadoop**
  1. **Procesamiento dentro de Spark, como accede a la información?**
  2. **Que tipo de abstracción de datos tiene?**
     1. **Puede leer archivos de datos de HDFS, FS**
  3. **RDD – Resilient distributed Dataset – porque cada letra?**
  4. **Como accede a los RDD dentro de Spark? SparkContext / HiveContext**
  5. **Componentes dentro de Spark**
     1. **SparkContext**
     2. **SparkSQL**
     3. **GraphQL**
     4. **MLib**
     5. **SparkCore**
  6. **Que operaciones hay? Que características tiene? Procesamiento Lazy. Ejemplos**
     1. **Transformaciones**
     2. **Acciones**
  7. **Como se ejecutan los Jobs dentro de Spark? Driver, .., etc**
  8. **Configuraciones de Spark / Modos de distribución**
     1. **Standalone**
     2. **Distributed**
     3. **Semi distributed**
  9. **DataFrames**
  10. **Ejecución de Jobs en spark**
  11. **Responsabilidades, que pasa si se cae un nodo cuando lo están leyendo, …**

1. **Plataformas para: - para que sirven, cuando se usa cada una**
   1. **Analisis: Hadoop (MR), Spark, Pig**
   2. **Storage: HDFS, Hive**
   3. **Ingestión: Sqoop,**
   4. **Visualizacion:**
   5. **Persistencia: BD NoSQL**
   6. **Procesamiento en streaming: Lambda, Spark, Storm, Flink**
2. **BD NoSQL, diferencias con RDBMS, Acid vs. Base, teorema CAP, ¿cuáles son, categorías?**
   1. **Teorema CAP – formas de clasificarlas**
      1. **Que dice**
      2. **Características - siempre hay un trade off**
      3. **Que DB hay en cada vértice?**
   2. **Acid vs. Base. Explicar cada sigla**
      1. **ACID**
      2. **BASE**
         1. **Consistencia eventual**
         2. **Availability**
   3. **Tipos de Datos**
   4. **Clasificación según formato de datos**
      1. **Documental**
      2. **Graph**
      3. **Columnar**
      4. **Key Value**
3. **Datawarehouse vs. DataLake: Raw data/pre ETL.**
4. **Arquitectura Lambda**
   1. **Que es?**
   2. **Para que sirve?**
   3. **Que cosas soporta Lamba(Storm)? Que corren encima? Que ya lo incorpora(Spark)?**
   4. **Contra**
   5. **Que tecnologías se usan para arquitectura Lambda**
      1. **Storm**
      2. **Flink**
5. **Si hay que leer datos de una relacional para meterlo en HDFS y cruzarlo con más información, que plataforma usarías?**
   1. **Sqoop**
      1. **Arquitectura**
      2. **Componentes**
      3. **Operaciones**
      4. **Como se hace un read**
6. **Para ingestar datos de logs**
   1. **Logmessage???**
7. **Para consumir mensajes de streaming**
   1. **Cola de mensajería Kafka**
8. **Como administras tantos componentes**
   1. **Zookeeper**
9. **Como armarías una plataforma**
   1. **Que preguntas harías para armar una plataforma de big data?**