Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc479026702)

[1.1 Autores 3](#_Toc479026703)

[Javier Cia del Toro, Alberto Moraga Fernandez, Jorge Benitez Abad 3](#_Toc479026704)

[1.2 Planificación 3](#_Toc479026705)

[Se ha realizado la planificación mediante el uso de la herramienta GanttPro, adjuntamos captura para verificarlo: 3](#_Toc479026706)

[1.3 Entrega 3](#_Toc479026707)

[2. Descripción de las tecnologías 4](#_Toc479026708)

[2.1 Descripción de Hadoop 4](#_Toc479026709)

[Apache Hadoop busca la consecución de los siguientes objetivos: 4](#_Toc479026710)

[2.2 Descripción Spark 5](#_Toc479026711)

[3. Criterios de comparación 7](#_Toc479026712)

[3.1 Categoría A: General 7](#_Toc479026713)

[3.1.1 Criterio General A.1: Funcionalidad 7](#_Toc479026714)

[3.1.2 Criterio General A.2: Código abierto 7](#_Toc479026715)

[3.1.3 Criterio General A.3: Costos 8](#_Toc479026716)

[3.1.4 Criterio General A.4: Compatibilidad 8](#_Toc479026717)

[3.1.5 Criterio General A.5: Escalabilidad 8](#_Toc479026718)

[3.2 Categoría B: Técnico 8](#_Toc479026719)

[3.2.1 Criterio Técnico B.1: Procesamiento 8](#_Toc479026720)

[3.2.2 Criterio Técnico B.2: Rapidez 8](#_Toc479026721)

[3.2.3 Criterio Técnico B.3: Implementación 8](#_Toc479026722)

[3.2.4 Criterio Técnico B.4: Recuperación 8](#_Toc479026723)

[3.2.5 Criterio Técnico B.5: Rendimiento 9](#_Toc479026724)

[3.2.6 Criterio Técnico B.6: Seguridad 9](#_Toc479026725)

[3.2.7 Criterio Técnico B.7: Tolerancia a fallos 9](#_Toc479026726)

[3.2.8 Criterio Técnico B.8: Lenguaje 9](#_Toc479026727)

[3.2.9 Criterio General B.9: Sistema operativo 9](#_Toc479026728)

[3.2.10 Criterio General B.10: Aprendizaje automático 9](#_Toc479026729)

[3.2.11 Criterio General B.11: Portable 9](#_Toc479026730)

[3.3 Categoría C: Utilización 10](#_Toc479026731)

[3.3.1 Criterio Utilización C.1: Almacenamiento 10](#_Toc479026732)

[3.3.2 Criterio Utilización C.2: Productividad 10](#_Toc479026733)

[3.3.3 Criterio Utilización C.3: Usabilidad 10](#_Toc479026734)

[3.3.4 Criterio Utilización C.4: Gráficos 10](#_Toc479026735)

[4. Evaluación de los criterios por tecnología 10](#_Toc479026736)

[4.1 Evaluación de los criterios para Hadoop 10](#_Toc479026737)

[4.2 Evaluación de los criterios para Spark 11](#_Toc479026738)

[5. Comparación de las tecnologías 13](#_Toc479026739)

[6. Recomendaciones 16](#_Toc479026740)

[6.1 Situación 1 16](#_Toc479026741)

[6.1.1 Descripción de la situación 16](#_Toc479026742)

[6.1.2 Recomendación de tecnología a utilizar 16](#_Toc479026743)

[6.2 Situación 2 16](#_Toc479026744)

[6.2.1 Descripción de la situación 16](#_Toc479026745)

[6.2.2 Recomendación de tecnología a utilizar 16](#_Toc479026746)

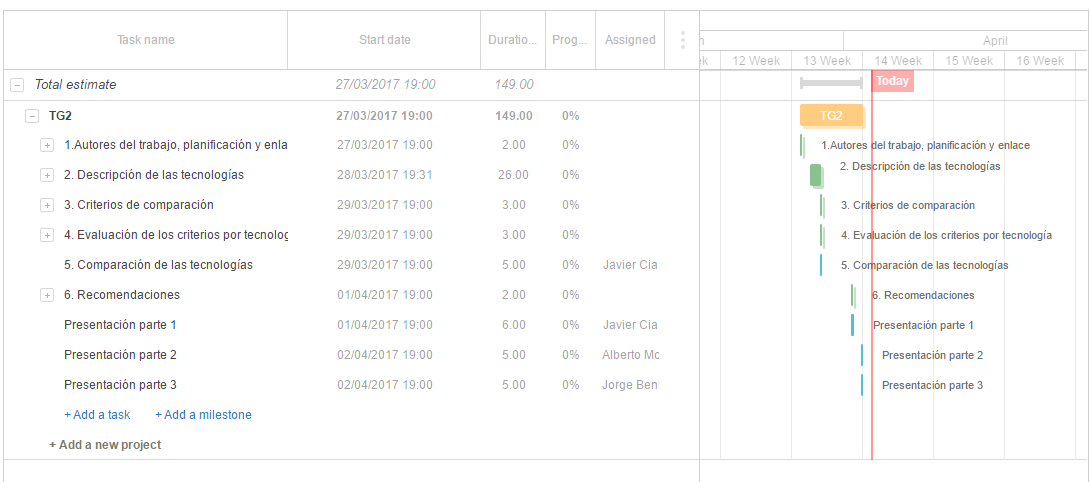
# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

## Javier Cia del Toro, Alberto Moraga Fernandez, Jorge Benitez Abad

## 1.2 Planificación

## Se ha realizado la planificación mediante el uso de la herramienta GanttPro, adjuntamos captura para verificarlo:



En cuanto a la distribución de apartados es la siguiente:

- Alberto Moraga: Apartados 2 y 6, presentacion parte 2(15 horas)

- Jorge Benítez: Apartados 3 y 4, presentación parte 3(19 horas)

- Javier Cia: Apartados 1 y 5, presentación parte 1(16 horas)

Por ultimo añadimos el [enlace](https://app.ganttpro.com/shared/token/2b93dd044de07de4933ee7160e25989283c9176026dccefa5b44aecfd0ee9087) para que se pueda comprobar la veracidad de la captura y la planificación en detalle:

## 1.3 Entrega

Añadimos el [enlace del repositorio](https://github.com/javierCia1/TG2) para esta entrega.

# 2. Descripción de las tecnologías

## 2.1 Descripción de Hadoop

La tecnología Apache Hadoop es una de las soluciones pioneras en el ámbito de Big Data, nacida en el seno de las principales empresas tecnológicas de Internet (inspirada en Google MR y Google FS, extendida por Yahoo!). Es ya una solución madura y robusta, y es utilizada por organizaciones como eBay, LinkedIn, Facebook o Twitter.

Apache Hadoop es un framework que permite el tratamiento distribuido de grandes cantidades de datos y trabajar con miles de máquinas de forma distribuida. Es decir,

Las principales características de Hadoop son:

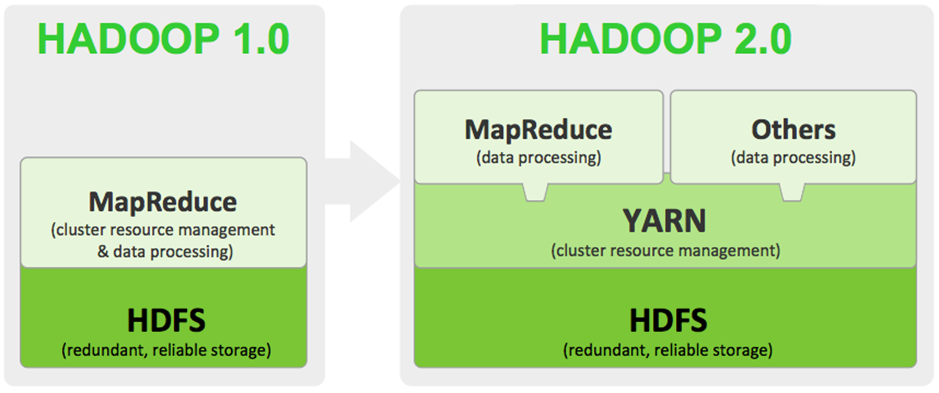
* Económico: Está diseñado para ejecutarse en equipos de bajo coste formando clústeres. Estos clústeres pueden llevarnos a pensar en miles de nodos de procesamiento disponibles para el procesado de información.
* Escalable: Si se necesita más poder de procesamiento o capacidad de almacenamiento solo hay que añadir más nodos al clúster de forma sencilla.
* Eficiente: Hadoop distribuye los datos y los procesa en paralelo en los nodos donde los datos se encuentran localizados.
* Confiable: Es capaz de mantener múltiples copias de los datos y automáticamente hacer un redespliegue de las tareas. El aspecto clave de Hadoop es que en lugar de mover los datos hacia donde se hace el procesamiento, Hadoop mueve el procesamiento a donde están los datos.
* Consta de 2 Componentes centrales: HDFS (almacenamiento de archivos) y MapReduce (infraestructura de programación)

## Apache Hadoop busca la consecución de los siguientes objetivos:

* Almacenamiento: Mediante un repositorio de almacenamiento independiente y autónomo.
* Performance: Aprovechando las capacidades de multiprocesamiento en paralelo.
* Robustez: Mediante mecanismos de fallo parcial y recuperación.
* Actualización: Utilizando las capacidades de migración de datos en tiempo real.
* Productividad: Implementando los procesos en lenguajes ampliamente extendidos
* Funcionalidad: Proporcionando algoritmos y técnicas de inferencia de conocimiento.

Tiene muchas herramientas que permiten interactuar con él:

* HDFS: Es el sistema de archivos como tal en el cual se almacenan los datos, está compuesto por el nodo maestro, que se encarga de almacenar los metadatos, y por una serie de nodos esclavos que se encargan de almacenar los datos.
* MapReduce: Es la herramienta propia de Hadoop para procesar los datos que están almacenados en HDFS, en este trabajo es bastante importante ya que Hadoop no va a competir con Spark en todos sus aspectos, sino principalmente en este (cuando expliquemos Spark veremos a que nos referimos).
* Flume: Sirve para mover grandes cantidades de logs a un respositorio central
* Hive: Sirve para organizar los datos de una manera que sean más accesibles para su procesamiento.
* Sqoop: Sirve para transferir datos desde bases de datos tradicionales (con datos estructurados) a HDFS, y viceversa.
* Y un amplio etcétera de herramientas que sirven para interactuar con HDFS.



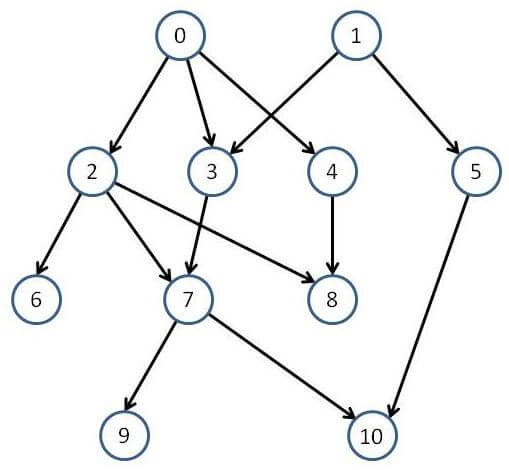
http://www.dineshonjava.com

## 2.2 Descripción Spark

Spark es un framework orientado al procesamiento de grandes cantidades de datos. Es un proyecto liderado por Databricks e ideado para reducir los tiempos de ejecución de MapReduce mediante la carga de datos en memoria.

Spark mantiene la escalabilidad lineal y la tolerancia a fallos de MapReduce, pero amplía sus bondades gracias a varias funcionalidades: DAG y RDD.

DAG (Directed Acyclic Graph)



geekytheory.com

DAG (Grafo Acíclico Dirigido) es un grafo dirigido que no tiene ciclos, es decir, para cada nodo del grafo no hay un camino directo que comience y finalice en dicho nodo. Un vértice se conecta a otro, pero nunca a sí mismo**.**

En comparación con MapReduce, el cual crea un DAG con dos estados predefinidos (Map y Reduce), los grafos DAG creados por Spark pueden tener cualquier número de etapas. Spark con DAG es más rápido que MapReduce por el hecho de que no tiene que escribir en disco los resultados obtenidos en las etapas intermedias del grafo. MapReduce, sin embargo, debe escribir en disco los resultados entre las etapas Map y Reduce.

RDD (Resilient Distributed Dataset)

Apache Spark mejora con respecto a los demás sistemas en cuanto a la computación en memoria. Surge debido a que las herramientas existentes tienen problemas que hacen que se manejen los datos ineficientemente a la hora de ejecutar algoritmos iterativos y procesos de minería de datos. En ambos casos, mantener los datos en memoria puede mejorar el rendimiento considerablemente.

El almacenamiento de los datos en memoria caché hace que los algoritmos de machine learning ejecutados que realizan varias iteraciones sobre el conjunto de datos de entrenamiento sea más eficiente. Además, se pueden almacenar versiones transformadas de dichos datos.

Las características principales de Spark son:

* Open Source
* Integrado con YARN (Hadoop 2.0)
* Evita accesos a disco y permite compartir datos entre procesos. Mejores tiempos de respuesta en procesos batch.
* Posibilidad de trabajar en Streaming.
* Acceso de datos ‘online’.
* Ideal para tratamiento de grafos y minería de datos.

Para concluir y que las cosas queden claras vamos a hacer un pequeño resumen:

¿QUÉ ES SPARK?

* No es una versión modificada de Hadoop
* Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.
* De 10 a 40 veces más rápido que Hadoop.
* Sistema escalable y con tolerancia a los fallos.
* Integrado en procesos Batch e Interactivos.
* Compatible con la API del sistema de almacenamiento de Hadoop

# 3. Criterios de comparación

En los sub-apartados de este apartado se deben indicar cada uno de los criterios (también llamados factores, propiedades, características, indicadores, etc.), que se usarán en la comparación.

Los criterios deben organizarse en categorías (al menos 3 categorías). El número de criterios totales no puede ser inferior a 20.

Las categorías dependerán del tipo de tecnología, pueden ser categorías como “General”, “Utilidades”, “Rendimiento”, etc.

Los criterios a definir en cada categoría también dependerán del tipo de tecnologías a comparar. En el siguiente apartado hay algunos ejemplos.

## 3.1 Categoría A: General

### 3.1.1 Criterio General A.1: Funcionalidad

Nombre del criterio: Funcionalidad de la herramienta.

Descripción: nivel de utilidades con las que cuenta la herramienta.

Tipo de valor: Texto (Alta/Baja)

### 3.1.2 Criterio General A.2: Código abierto

Nombre del criterio: Código abierto.

Descripción: si el software tiene un desarrollo y una distribución libre.

Tipo de valor: Booleano (Sí/No)

### 3.1.3 Criterio General A.3: Costos

Nombre del criterio: Costos.

Descripción: la herramienta tiene un determinado nivel de coste monetario o no monetario.

Tipo de valor: Texto (Alto/Bajo)

### 3.1.4 Criterio General A.4: Compatibilidad

Nombre del criterio: Compatibilidad.

Descripción: es compatible con otros softwares relacionados con este campo.

Tipo de valor: Booleano (Sí/No)

### 3.1.5 Criterio General A.5: Escalabilidad

Nombre del criterio: Escalabilidad.

Descripción: la herramienta reacciona y/o se adapta sin perder calidad.

Tipo de valor: Texto (Buena/Mala)

## 3.2 Categoría B: Técnico

### 3.2.1 Criterio Técnico B.1: Procesamiento

Nombre del criterio: Procesamiento.

Descripción: técnica de procesamiento de los datos.

Tipo de valor: Texto

### 3.2.2 Criterio Técnico B.2: Rapidez

Nombre del criterio: Velocidad.

Descripción: velocidad a la hora de procesar los datos.

Tipo de valor: Texto (Alta/Baja)

### 3.2.3 Criterio Técnico B.3: Implementación

Nombre del criterio: Implementación.

Descripción: ejecuta los métodos y medidas necesarias para llevar a cabo la tarea de la mejor forma posible.

Tipo de valor: Booleano (Sí/No)

### 3.2.4 Criterio Técnico B.4: Recuperación

Nombre del criterio: Recuperación de falla.

Descripción: capacidad de recuperar datos en el caso de que se produzca alguna falla.

Tipo de valor: Booleano (Sí/No)

### 3.2.5 Criterio Técnico B.5: Rendimiento

Nombre del criterio: Rendimiento.

Descripción: utilidad de la herramienta en proporción a los recursos que va a necesitar.

Tipo de valor: Texto (Alto/Bajo)

### 3.2.6 Criterio Técnico B.6: Seguridad

Nombre del criterio: Seguridad.

Descripción: confianza que proporciona la herramienta.

Tipo de valor: Texto (Alta/Baja)

### 3.2.7 Criterio Técnico B.7: Tolerancia a fallos

Nombre del criterio: Tolerancia a fallos.

Descripción: capacidad de minimizar la producción de errores.

Tipo de valor: Booleano (Sí/No)

### 3.2.8 Criterio Técnico B.8: Lenguaje

Nombre del criterio: Lenguaje de programación.

Descripción: lenguajes necesarios para la utilización de la herramienta.

Tipo de valor: Texto

### 3.2.9 Criterio General B.9: Sistema operativo

Nombre del criterio: Sistema operativo.

Descripción: sistemas operativos en los que se puede ejecutar.

Tipo de valor: Texto

### 3.2.10 Criterio General B.10: Aprendizaje automático

Nombre del criterio: Aprendizaje automático.

Descripción: compatible con el uso de técnicas capaces de generalizar comportamientos en base a una información suministrada.

Tipo de valor: Booleano (Sí/No)

### 3.2.11 Criterio General B.11: Portable

Nombre del criterio: Portable.

Descripción: posibilidad de utilizar la herramienta sin instalación previa.

Tipo de valor: Booleano (Sí/No)

## 3.3 Categoría C: Utilización

### 3.3.1 Criterio Utilización C.1: Almacenamiento

Nombre del criterio: Almacenamiento.

Descripción: la herramienta almacena realmente los datos.

Tipo de valor: Booleano (Sí/No)

### 3.3.2 Criterio Utilización C.2: Productividad

Nombre del criterio: Productividad.

Descripción: capacidad de realizar un trabajo óptimo sin consumir una cantidad innecesaria de recursos, incluido el tiempo.

Tipo de valor: Texto (Buena/Mala)

### 3.3.3 Criterio Utilización C.3: Usabilidad

Nombre del criterio: Usabilidad.

Descripción: la herramienta es sencilla de utilizar en todos sus aspectos.

Tipo de valor: Booleano (Sí/No)

### 3.3.4 Criterio Utilización C.4: Gráficos

Nombre del criterio: Datos en gráficos.

Descripción: capacidad de la herramienta en procesar datos obtenidos mediante gráficos.

Tipo de valor: Booleano (Sí/No)

## 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para Hadoop

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Funcionalidad | Alta |
| Criterio A.2: Código abierto | Sí |
| Criterio A.3: Costos | Bajo |
| Criterio A.4: Compatibilidad | Sí |
| Criterio A.5: Escalabilidad | Buena |
| Criterio B.1: Procesamiento | Por lotes |
| Criterio B.2: Rapidez | Baja |
| Criterio B.3: Implementación | Sí |
| Criterio B.4: Recuperación | Sí |
| Criterio B.5: Rendimiento | Alto |
| Criterio B.6: Seguridad | Alta |
| Criterio B.7: Tolerancia a fallos | Sí |
| Criterio B.8: Lenguaje | Java |
| Criterio B.9: Sistema operativo | Plataforma cruzada |
| Criterio B.10: Aprendizaje automático | No |
| Criterio B.11: Portable | No |
| Criterio C.1: Almacenamiento | Sí |
| Criterio C.2: Productividad | Mala |
| Criterio C.3: Usabilidad | No |
| Criterio C.4: Gráficos | No |

- Costos: aunque Hadoop sea de código libre, sus costos de funcionamiento son bajos ya que utiliza fundamentalmente memoria interna.

- Rapidez: la velocidad de Hadoop es relativamente buena, pero la de Spark es superior.

- Seguridad: Hadoop proporciona todos los beneficios en los avances obtenidos en los proyectos de seguridad. Además, consta de Hadoop YARN.

- Productividad: por motivos de velocidad, la productividad de Hadoop no es tan buena como la de Spark.

- Usabilidad: al trabajar con el lenguaje Java, Hadoop se convierte en una herramienta complicada de utilizar, sobre todo a la hora de realizar análisis de datos lógicos.

- Gráficos: Hadoop no consta de la capacidad de analizar datos localizados en gráficas.

## 4.2 Evaluación de los criterios para Spark

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Funcionalidad | Baja |
| Criterio A.2: Código abierto | Sí |
| Criterio A.3: Costos | Alto |
| Criterio A.4: Compatibilidad | Sí |
| Criterio A.5: Escalabilidad | Buena |
| Criterio B.1: Procesamiento | Híbrido (por lotes y de flujo) |
| Criterio B.2: Rapidez | Alta |
| Criterio B.3: Implementación | Sí |
| Criterio B.4: Recuperación | Sí |
| Criterio B.5: Rendimiento | Alto |
| Criterio B.6: Seguridad | Baja |
| Criterio B.7: Tolerancia a fallos | Sí |
| Criterio B.8: Lenguaje | Scala, Java, Python, R |
| Criterio B.9: Sistema operativo | Windows, OS X, Linux |
| Criterio B.10: Aprendizaje automático | Sí |
| Criterio B.11: Portable | No |
| Criterio C.1: Almacenamiento | No |
| Criterio C.2: Productividad | Buena |
| Criterio C.3: Usabilidad | Sí |
| Criterio C.4: Gráficos | Sí |

- Costos: aunque Spark sea de código libre, sus costos de funcionamiento son altos ya que trabaja principalmente con memoria RAM.

- Rapidez: la velocidad de Spark es mayor que la de Hadoop ya que realiza todas las operaciones en tiempo real. Lee los datos del clúster, realiza todas las operaciones analíticas necesarias y escribe los resultados en el clúster. Spark puede ser hasta 10 veces más rápido en procesamiento por lotes y hasta 100 veces más rápido para análisis en memoria que Hadoop. La rapidez de Spark es perfecta para el análisis de datos en transmisión, pero para la realización de operaciones de datos estáticos es preferible la utilización de Hadoop.

- Seguridad: Spark necesita recurrir a Hadoop YARN para obtener beneficios de seguridad.

- Productividad: gracias a su gran velocidad, la productividad de Spark es mejor que la de Hadoop.

- Usabilidad: gracias a la cantidad de lenguajes que utiliza, Spark se convierte en una herramienta bastante sencilla de utilizar.

- Gráficos: Spark tiene la capacidad de realizar análisis sobre datos gráficos.

# 5. Comparación de las tecnologías

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | HADOOP | SPARK | COMENTARIOS |
| Criterio A.1: Funcionalidad | X |  | Es cierto que Hadoop al ser una herramienta que tiene bastante más funcionalidades a parte del procesamiento de datos(que es en lo que compite con Spark) vence en este apartado sin duda. |
| Criterio A.2: Código abierto | - | - | Ambos son de código abierto |
| Criterio A.3: Costos | X |  | Ambos son de código abierto, pero Hadoop utiliza memoria interna mientras que Spark utiliza memoria RAM, lo que incrementa en este ultimo caso mucho el coste del Hardware. |
| Criterio A.4: Compatibilidad | - | - | Ambos desarrollados por Apache y tienen prácticamente las mismas interfaces con otras extensiones para Big Data. |
| Criterio A.5: Escalabilidad |  | X | Spark está diseñado para ser más escalable(siéndolo también este) que Hadoop. |
| Criterio B.1: Procesamiento |  | X | El procesamiento de Spark nos ofrece dos opciones, híbrido y por lotes, mientras que el procesamiento de Hadoop no nos ofrece la posibilidad de procesamiento de flujo. |
| Criterio B.2: Rapidez |  | X | La rapidez de Spark es definitivamente mas rápido. |
| Criterio B.3: Implementación | - | - |  |
| Criterio B.4: Recuperación | - | X | Spark tiene una más alta tolerancia a fallos ya que está específicamente diseñado para esto. |
| Criterio B.5: Rendimiento | - | X | Spark tiene un rendimiento mucho mayor que Hadoop |
| Criterio B.6: Seguridad | X | - | Hadoop posee YARN, pero Spark tiene que acceder a YARN(de Hadoop) para asegurar sus operaciones |
| Criterio B.7: Tolerancia a fallos | - | X | Spark tiene una más alta tolerancia a fallos ya que está específicamente diseñado para esto. |
| Criterio B.8: Lenguaje |  | X | Hadoop solo ofrece la opción de Java mientras Spark 4 lenguajes, uno de ellos es Scala, mucho mas escalable que Java- |
| Criterio B.9: Sistema operativo |  | X | Spark ofrece mas sistemas operativos |
| Criterio B.10: Aprendizaje automático |  | X | Spark da opción de diseñar aplicaciones de aprendizaje automatico |
| Criterio B.11: Portable | - | - |  |
| Criterio C.1: Almacenamiento | X |  | Hadoop en este aspecto es mejor, ya que utiliza la memoria interna. |
| Criterio C.2: Productividad |  | X | Spark es mas productivo |
| Criterio C.3: Usabilidad |  | X | Spark permite mas opciones de lenguaje por lo que es mas usable. |
| Criterio C.4: Gráficos |  | X | Spark permite análisis sobre datos graficos |

# 6. Recomendaciones

## 6.1 Situación 1

### 6.1.1 Descripción

Nos encontramos ante el caso de que el mayor banco de inversión chino, el ZHUZHU, decide hacer uso del Big Data para posicionarse por delante se sus rivales más directos.

Han decidido que cogiendo los valores de la bolsa (cada minuto) a través del Machine Learning serán capaces de predecir con mucha exactitud el siguiente valor de bolsa diferente. Es decir predecir el futuro.

### 6.1.2 Recomendación de tecnología a utilizar

Debe indicarse la tecnología propuesta para esa situación.

Debe incluirse una tabla como la siguiente, mostrando las ventajas, respecto a los criterios, que ofrece cada tecnología en esa situación concreta.

Incluir sólo los criterios sobre los que se aprecien ventajas de una de las tecnologías frente a otra. No incluir criterios que no sean relevantes para la decisión (por ejemplo, el criterio “autor” seguramente no será relevante).

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Hadoop | Spark |
| Madurez | Hadoop es un sistema mucho mas maduro | Aunque en la situación actual es un sistema que ha madurado mucho, no llega alos niveles de Hadoop |
| Online | No |  |
| Adecuado para machine learning | Los algoritmos de Machine Learning proporcionan capacidades que no son fáciles de explotar en Hadoop Map Reduce | Tiene una biblioteca (MLIB) de Machine Learning |
| Sistema de almacenamiento | Hadoop posee el sistema de almacenamiento distribuido HDFS | No |
| Rapidez | MAPREDUCE tiene que mover datos dentro/ fuera de disco | Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.  De 10 a 40 veces más rápido que Hadoop. |

Aquí ante la posibilidad de una gran inversión por parte del banco para la implementación y poder contar con expertos en la tecnología Spark y Hadoop, la solución propuesta es ejecutar la herramienta Spark en la parte “superior” de Hadoop. No obstante, cabe recordar que Spark no tiene un sistema de almacenamiento independiente y Spark debe cargar los datos desde algún lugar para procesarlos. Por lo tanto al integrarlo con Hadoop, el banco Zhuzhu puede aprovechar las ventajas de ambas tecnologías.

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

Una empresa de paquetería decide realizar un estudio de todas las direcciones de envío donde van repartiendo para realizar una optimización de los recorridos que realizan los transportistas. Con ello reducirán el tiempo de entrega, algo vital en esta industria.

### 6.2.2 Recomendación de tecnología a utilizar

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Hadoop | Spark |
| Madurez | Hadoop es un sistema mucho mas maduro | Aunque en la situación actual es un sistema que ha madurado mucho, no llega alos niveles de Hadoop |
| Online | No | Acceso de datos en streaming |
| Sistema de almacenamiento | Hadoop posee el sistema de almacenamiento distribuido HDFS | No |
| Rapidez | MAPREDUCE tiene que mover datos dentro/ fuera de disco | Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.  De 10 a 40 veces más rápido que Hadoop. |

Como en este caso no depende de tanta velocidad de procesamiento, los accesos a los datos no tienen por qué ser en streaming, el procesamiento no debe ser en tiempo real.

La solución adoptada es Spark.

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)