Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc445388848)

[1.1 Autores 2](#_Toc445388849)

[1.2 Planificación 2](#_Toc445388850)

[1.3 Entrega 2](#_Toc445388851)

[2. Descripción de las tecnologías 3](#_Toc445388852)

[2.1 Descripción de la tecnología 1 3](#_Toc445388853)

[2.2 Descripción de la tecnología 2 3](#_Toc445388854)

[3. Criterios de comparación 4](#_Toc445388855)

[3.1 Categoría A: Nombre 4](#_Toc445388856)

[3.1.1 Criterio A.1: Nombre 4](#_Toc445388857)

[3.1.2 Criterio A.2: Nombre 5](#_Toc445388858)

[3.1.n Criterio A.n: Nombre 5](#_Toc445388859)

[3.2 Categoría B: Nombre 5](#_Toc445388860)

[3.2.1 Criterio B.1: Nombre 5](#_Toc445388861)

[3.2.2 Criterio B.2: Nombre 5](#_Toc445388862)

[3.2.n Criterio B.n: Nombre 5](#_Toc445388863)

[3.3 Categoría Z: Nombre 5](#_Toc445388864)

[3.3.1 Criterio Z.1: Nombre 5](#_Toc445388865)

[3.3.2 Criterio Z.2: Nombre 5](#_Toc445388866)

[3.3.n Criterio Z.n: Nombre 5](#_Toc445388867)

[4. Evaluación de los criterios por tecnología 6](#_Toc445388868)

[4.1 Evaluación de los criterios para la tecnología 1 6](#_Toc445388869)

[4.2 Evaluación de los criterios para la tecnología 2 6](#_Toc445388870)

[5. Comparación de las tecnologías 7](#_Toc445388871)

[6. Recomendaciones 9](#_Toc445388872)

[6.1 Situación 1 9](#_Toc445388873)

[6.1.1 Descripción de la situación 9](#_Toc445388874)

[6.1.2 Recomendación de tecnología a utilizar 9](#_Toc445388875)

[6.2 Situación 2 9](#_Toc445388876)

[6.2.1 Descripción de la situación 9](#_Toc445388877)

[6.2.2 Recomendación de tecnología a utilizar 9](#_Toc445388878)

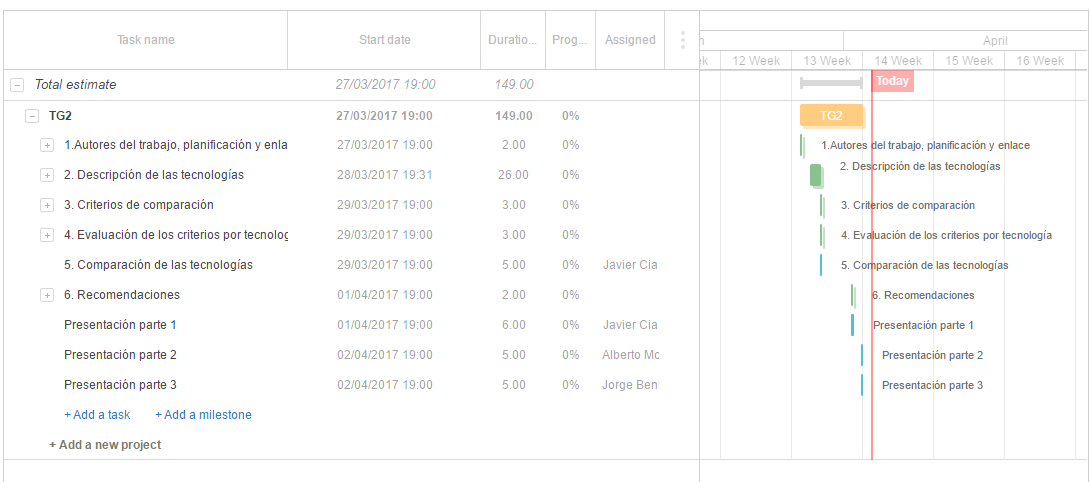
# 1. Autores del trabajo, planificación y entrega

## 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

En los siguientes apartados se debe describir brevemente cada tecnología a comparar.

Se pueden incluir imágenes copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada una se indique la fuente (al menos la URL).

## 2.1 Descripción de la tecnología 1

## 2.2 Descripción de la tecnología 2

# 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

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

Esta tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

|  |  |  |  |
| --- | --- | --- | --- |
| 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

Deben platearse posibles situaciones de uso, y recomendar justificadamente una u otra tecnología en función de la situación. Al menos 2 situaciones diferentes.

## 6.1 Situación 1

### 6.1.1 Descripción de la situación

*Una posible situación en el caso de comparar dos herramientas CASE, podría ser el caso de una empresa de desarrollo muy interesada en tecnologías open source, que programa sólo en Java, con equipos de desarrollo pequeños, que utiliza UML como notación, etc, etc*…

### 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 | Ventajas tecnología 1 | Ventajas tecnología 2 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

### 6.2.2 Recomendación de tecnología a utilizar

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

(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.)