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

En este apartado se debe indicar el número de grupo y los nombres de los autores, poniendo en primer lugar al coordinador del grupo.

## 1.2 Planificación

En este apartado se debe incluir un enlace (URL) compartido a la planificación del trabajo utilizando una herramienta online de diagramación Gantt (por ejemplo, GanttPro, versión gratuita).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 15 horas. El peso de este trabajo en la calificación total de la asignatura es de un 10%, por tanto requiere de una dedicación de 15 horas del total de 150 horas de la asignatura.

<https://app.ganttpro.com/shared/token/5381d5471d9b6f1fd65ae42ad9a97666b9e701480398d76f7f50dd40974ae8d8>

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHub o en BitBucket creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Trabajo terminado: del trabajo terminado con el nombre TG2\_final.docx
* Presentación del trabajo: TG2\_final.pptx

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Descripción de las tecnologías ORM (Hibernate & Entity Framework)

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 Hibernate

**¿QUÉ ES HIBERNATE?**

Para introducirnos en el mundo de Hibernate primero tomaremos una definición:

“Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre deNHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en losbeansde las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL”.

En otras palabras, Hibernate es un Framework que agiliza la relación entre la aplicación y la base de datos. Para poder aprender a utilizarlo es necesario contar con los conocimientos básicos de base de datos y SQL  así como manejar el lenguaje Java.

**HISTORIA**

Hibernate fue una iniciativa de un grupo de desarrolladores dispersos alrededor del mundo conducidos por Gavin King.  
  
Hibernate se inició en 2001 por Gavin King con colegas de Cirrus Tecnologías como una alternativa al uso de beans de entidad estilo EJB2. Su objetivo original era ofrecer una mejor capacidad de persistencia de los ofrecidos por EJB2 simplificando las complejidades y complementa las características faltantes.  
  
A principios de 2003, el equipo de desarrollo Hibernate comenzó a dar comunicados de Hibernate2, que ofrecía muchas mejoras significativas con respecto a la primera versión.  
  
Tiempo después, JBoss Inc. (ahora parte de Red Hat) contrató a los principales desarrolladores de Hibernate y trabajó con ellos en brindar soporte al proyecto.  
  
En 2005, Hibernate versión 3.0 fue lanzado. Las características claves incluyen una nueva arquitectura Interceptor/Callback, filtros definidos por el usuario, y JDK 5.0 Anotaciones(Java metadatos de función).

En 2005 tambien fue lanzado **NHibernate** es la conversión de [Hibernate](https://es.wikipedia.org/wiki/Hibernate" \o "Hibernate) de lenguaje [Java](https://es.wikipedia.org/wiki/Java_(Sun)) a [C#](https://es.wikipedia.org/wiki/C_Sharp) para su integración en la plataforma .NET.  Su última versión estable es la 4.0.3

A partir de 2010, Hibernate 3(versión 3.5.0 o superior) era una aplicación certificada en la Java Persistence API 2.0 especificación a través de un contenedor para el módulo Core que proporcionaba la conformidad con el JSR 317 estándar.  
  
En diciembre de 2011, Hibernate Core 4.0.0 fué puest en libertad. Esto inluye nuevas características como el soporte multi-alquiler, la introducción de ServiceRegistry(un cambio importante en la forma de hibernación construye y gestiona "servicios"), una mejor apertura de sesión de SessionFactory, mejora la integración a través de org.hibernate.integrator.spi.integrator y detección automática, internacionalización de códigos de soporte y mensajes de registro, una división más clara entre el API, SPI y clases de implementación.  
  
En diciembre de 2012, Hibernate ORM 4.1.9 Se lanzó al mercado.

En 2012 también se inició el 5 de Hibernate. Contendrá JPA 2.1

La actual versión de Hibernate disponible es la 5.1.0 y su última versión estable es la 4.3.1.

**CARACTERISTICAS:**

Estas son solo algunas de las características que nos ofrece Hibernate

* Herramienta de Mapeo objeto relacional (ORM) para Java.
* Facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante ficheros declarativos (XML) o anotaciones en los beans
* Software libre
* Busca solucionar el problema entre el modelo de memoria de una aplicación (orientado a objetos) y el usado en base de datos (modelo relacional).
* Hibernate permite a la aplicación manipular los datos de la base operando sobre los objetos, que tiene las características de la POO.
* Hibernate convertirá los datos entre los tipos utilizados en Java y los definidos por SQL.
* Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias.
* Mantiene la portabilidad entre los motores de base de datos, con un ligero incremente en el tiempo de ejecución.
* Es flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
* También puede crear una base de datos a partir de la información disponible.
* Además, ofrece un lenguaje de consulta de datos llamado HQL, además de un API para construir las consultas programáticamente (criteria).
* Puede utilizarse en aplicaciones Java independientes (o standalones) o bien en aplicaciones Java EE, mediante el componente Hibernate Annotations.

## 2.2 Descripción de Entity Framework

# 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 Precio de la herramienta

Nombre del criterio: Precio.

Descripción: Valor por el cual se puede adquirir el software para poder utilizarlo.

Tipo de valor: Texto libre.

### 3.1.2 Criterio A.2: Nombre

### 3.1.n Criterio A.n: Nombre

## 3.2 Categoría B: REQUERIMIENTOS

3.2.1 Supported Database Management Systems

Nombre del criterio: Sistemas gestores soportados

Descripción: Sistema gestores de bases de datos que son soportados por la herramienta.

Tipo de valor: Texto libre.

### 3.2.2 Criterio B.2: Nombre

### 3.2.n Criterio B.n: Nombre

## 

## 3.3 Categoría c: RENDIMIENTO/EFICIENCIA

3.3.1 TTL

Nombre del criterio:TTL

Descripción: Este es el tiempo entre el comienzo del índice de referencia y la ejecución de la primera consulta.

Tipo de valor: Numérico

3.3.2 SQ(Symple Query)

Nombre del criterio: Consulta Sencilla

Descripción: Tiempo medio que tarda en realizar una consulta sencilla.

Tipo de valor: Numérico

3.3.3 CQ(Complex Query)

Nombre del criterio: Consulta Compleja

Descripción: Tiempo medio que tarda en realizar una consulta compleja, incluye projections, joins, grouping and sorting.

Tipo de valor: Numérico

3.3.4 EL (eager loading)

Nombre del criterio: una consulta con la carga ansiosa

Descripción: rendimiento ante una consulta con una carga ansiosa de datos.

Tipo de valor: Numérico

3.3.5 CUD(creation, update and delete)

Nombre del criterio: creación, actualización y borrado de operaciones.

Descripción: rendimiento de la ejecución de las operaciones básicas como insertar, actualizar y borrar.

Tipo de valor: Numérico

3.3.6 CPU

Nombre del criterio: el tiempo de uso de la CPU

Descripción: es la cantidad de tiempo en que la unidad central de proceso es usada por las instrucciones del programa.

Tipo de valor: Numérico

3.3.7 RAM

Nombre del criterio: uso de memoria media

Descripción: Porcentaje medio de memoria RAM utilizada por el software durante su ejecución.

Tipo de valor: Numérico.

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología 1

Debe incluir al menos una tabla con la siguiente estructura.

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Nombre |  |
| Criterio A.2: Nombre |  |
| Criterio A.n: Nombre |  |
| … |  |
| Criterio B.1: Nombre |  |
| Criterio B.2: Nombre |  |
| … |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

La 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).

## 4.2 Evaluación de los criterios para la tecnología 2

# 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 | TECNOLOGÍA A | TECNOLOGÍA B | COMENTARIOS |
| A.1 |  |  |  |
| A.2 |  |  |  |
| … |  |  |  |
| B.1 |  |  |  |
| B.2 |  |  |  |
| … |  |  |  |

# 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.)