

Práctica 4.1 - HIBERNATE

Realizar un proyecto en Java que utilizando Hibernate, realice la siguientes tareas:

Conectarse a una base de datos MySql llamada **Practica4_orm** con el usuario **root4** y la contraseña **Password4**. El comportamiento del programa hará que cada vez que se ejecute el programa se vuelvan a crear las tablas mapeadas desde cero.

Crea tres clases llamadas **Heroe**, **Poder** y **Villano** y mediante Hibernate (con correspondencia hbm o JPA) mapea dichas clases contra las tablas mostradas a continuación.

Tabla para la **clase heroe**, **clase poder** y **clase villano**, respectivamente.

Information	Information	Information
Table: heroes Columns: HeroeId int AI PK Alias varchar(255) Ataque float Debilidades varchar(255) Defensa float Fortalezas varchar(255) Nivel int Nombre varchar(255) Vida float	Table: poderes Columns: PoderId int AI PK Descripcion varchar(255) Nombre varchar(255)	Table: villanos Columns: VillanoId int AI PK Alias varchar(255) Ataque float Debilidad varchar(255) Defensa float Nivel int Nombre varchar(255) Vida float

NOTA: Las claves principales de cada tabla se definirán como auto-numéricas en la clase.

El programa tendrá un menú donde según se elija se realizarán las siguientes tareas:

0 – Salir del programa

1 – Cargar grupo de héroes o villanos.

(Se preguntará si héroe o villano). Introducirá **al menos 4** héroes y 4 villanos según elección. Independientemente de la elección cargará **al menos 2 poderes**.(4 si se eligen villanos y luego héroes)

2 – Mostrar todos los héroes o villanos.

(Se preguntará si héroe o villano). Mostrará por pantalla todos los héroes o villanos según opción, pero siempre se muestran todos los poderes.

3 – Seleccionar un héroe o villano a modificar.

(Se preguntará si héroe o villano). Se preguntará por id para identificar el objeto y posteriormente se permitirá **modificar sus valores de ataque, defensa y Vida los cuales se actualizarán en la BD.**

Práctica 4.1 - HIBERNATE

4 – Eliminar héroe o villano.

(Se preguntará si héroe o villano). Se preguntará por id para **eliminarlo**.

5 – COMBATE.

Se seleccionará 1 héroe y 1 villano (de forma aleatoria por id) y se mostrarán por pantalla indicando quién vencería en combate, calculando el combate de la siguiente forma:

5.1 – $\text{heroes.Vida} = \text{heroes.Vida} - (\text{villanos.Ataque} - \text{heroes.Defensa})$

5.2 – $\text{villanos.Vida} = \text{villanos.Vida} - (\text{heroes.Ataque} - \text{villanos.Defensa})$

Por cada cálculo realizado se mostrará el resumen del tipo:

ATAQUE 2 (ronda de ataque)

Heroe SPIDERMAN - Def= 9, Ataq= 18, PV=60. (6 de daño, PVtemp 48).

Villano Dr. OCTOPUS – Def=6, Ataq=15, PV=50. (12 de daño, Pvtemp 26).

Mientras el valor de Vida de uno de los dos objetos sea > 0, se preguntará si continúa el combate o se retiran.

a) Si hay retirada, se guardan los valores de vida en la BD y se vuelve al menú.

b) Si se continúa se vuelve a generar una nueva ronda de combate.

Al finalizar el combate se mostrará un resumen como los anteriores pero indicando el vencedor y mostrando el **nombre real** del perdedor.

ATAQUE 5 (ronda final)

Heroe SPIDERMAN - Def= 9, Ataq= 18, PV=60. (6 de daño, PVtemp 30).

Villano Dr. OCTOPUS – Def=6, Ataq=15, PV=50. (12 de daño, Pvtemp -10).

SPIDERMAN gana el combate contra Dr. OCTOPUS y lo identifica como OTTO OCTAVIUS

NOTA: Los valores de vida tras un combate se guardan en la BD.

Sobre valores de registros:

Los valores de Ataque estarán entre 10 y 20.

Los valores de Defensa estarán entre 0 y 10.

Los valores de Vida estarán entre 30 y 100.

OPCIONAL PARA SUBIR NOTA:

A) Se puede hacer que por cada ronda de ataque donde se haya hecho daño (puede pasar que no se supere la defensa y en ese caso no se aplica) se disminuya la defensa en 1, informando de este hecho.

Práctica 4.1 - HIBERNATE

B) El personaje que gane un combate se le incrementa en 1 su valor de nivel y 3 su valor de ataque.

Notas sobre la práctica

Se valorará la estructuración del código , tanto en métodos como en comentarios.

Que los distintos menús utilizados en el programa sean simples y permitan no solo elegir opción sino volver al anterior.