

Ultimate Tic Tac Toe

-David Puga Mendivil A01332391

-Dennis Omar Lugo A01214271

-Cynthia García Velasco A01332329

Tabla Comparativa Frameworks

JPA	myBatis	Hibernate	JDBC
Este framework trata de conservar las ventajas de la orientación a objetos interactuando con la base de datos a través de objetos, permitiendo usar los “ <i>Plain Old Java Object</i> ” (POJOs).	Herramienta que permite mapear sentencias de SQL y procedimientos almacenados con objetos a partir de ficheros XML. De igual forma permite que el modelo de datos y el modelo de objetos se encuentren independientes.	Busca principalmente evitar problemas entre los modelos de datos que coexisten en la aplicación: el usado en la memoria de la computadora y el usado en las bases de datos.	Permite la ejecución de operaciones sobre bases de datos desde Java, independientemente del sistema operativo donde sea ejecutado o de la base de datos a la cual se accede.
Utiliza archivos XML para configurar el mapeo entre los objetos de Java y las tablas en la base de datos. Utiliza un language similar a SQL, Java Persistence Query Language (JPQL).	Permite utilizar todas las funcionalidades de la base de datos (procedimientos almacenados, vistas). Es una herramienta indicada para bases de datos legadas o desnormalizadas.	Ofrece también el lenguaje HQL (Hibernate Query Language) que soporta características avanzadas como paginación y perfiles dinámicos, así como queries polimorficos.	Únicamente soporta SQL y se debe especificar en el código el mapeo para asignar la representación de datos de un modelo de objetos a un modelo de datos relacional y su esquema de base de datos correspondiente.

Para un mejor entendimiento de cada uno de los frameworks previamente mencionados se realizaron pruebas de concepto de cada uno de ellos para una mejor comprensión.

El código de las pruebas de concepto se encuentra dentro de repositorio GitHub donde se encuentra este archivo.

- PoC JPA:

En primera instancia se creó un nuevo proyecto en NetBeans de Java Application.

Se debe corroborar que exista el driver MySQL (Connector/J driver) dentro del proyecto.

Dentro de este proyecto se crearon las entidades a partir de la base de datos. Para esto se utilizó el esquema de base de datos diseñado previamente y se utilizó la opción de “Entity Classes from Database”.

A través de esta opción se crearon las clases entidades para Jugador y Juego y un archivo XML que contenía los datos necesarios para la conexión a base como usuario y contraseña.

En la clase principal instanciamos los queries utilizados a través de un EntityManager para poder traer datos de la base.

- PoC myBatis:

En primera instancia se creó un nuevo proyecto en NetBeans de Java Application.

Se corroboró que existiera el driver MySQL (Connector/J driver) dentro de las librerías del proyecto, así mismo se importó la librería de myBatis (link de descarga para myBatis: <https://github.com/mybatis/mybatis-3/releases>).

Se creó un nuevo archivo XML, llamado SQLConfig.xml. Dentro de este archivo se definieron los datos necesarios para la conexión con la base de datos. De igual forma se crearon dos archivos XML Jugador y Juegos donde se mapearan los objetos a las sentencias de SQL y dos clases de objeto (POJO) Java Jugador y Juego las cuales mapean los datos del XML.

Finalmente cargamos nuestro mapeador SQLConfig.xml a nuestra aplicación en la clase principal y mandamos a llamar a los datos de nuestra base.

- PoC Hibernate

En primera instancia se creó un nuevo proyecto en NetBeans de Java Application.

Se importaron las librerías necesarias para el uso de Hibernate (link de descarga en caso de no tener las librerías de hibernate <http://hibernate.org/orm/downloads/>).

Se creó un nuevo archivo XML, dentro de este archivo se definirán los datos necesarios para la conexión con la base de datos como usuario y contraseña. Así mismo creamos dos archivos XML, Jugador y Juego con la terminación hbm a los cuales mapearemos la base de datos. Después creamos las clases objeto Java a las cuales esta mapeando el XML, Juego y Jugador.

A través de una clase auxiliar HibernateUtil leemos nuestro mapeo especificado dentro de nuestros XML para poder establecer la conexión.

En la clases principal traemos o insertamos los datos a la base, para esto se realiza una llamada a una sesión (session) que llama a la clase HibernateUtil para permitir la conexión a base de datos.

- PoC JDBC

En primera instancia se creó un nuevo proyecto en NetBeans de Java Application.

Se debe corroboró que existiera el driver MySQL (Connector/J driver) dentro del proyecto.

Para el conectar la base de datos utilizamos una clase Conexión.java que manda los datos necesarios para establecer la conexión con sql. Después a través de la conexión utilizamos una clase Servicio.java, que se encarga obtener la información de las tablas de la base de datos a través de la ejecución de los queries especificados. Finalmente en la clase principal llamamos a los metodos de la clase Servicio que se encargan de traer los datos de la base.