UNIVERSIDAD PRIVADA DE TACNA
INGENIERIA DE SISTEMAS

TITULO:

Comparativa de tecnologias ORM

CURSO:

BASE DE DATOS 2

DOCENTE:

Ing. Patrick Cuadros Quiroga

Integrantes:

Briset Celia Garcia Salazar	(2018062496)
Mathius Omar Farfan Colque	(2015050991)
Roger Alex Aria Vela	(2015050623)
Daniel Alejandro Gonzalez Franco	(2015052599)
Deivis Jhonatan Flores Navarro	(2018060916)
Luis Fernando Flores Querieco	(2018062394)

Tacna - Perú
2021

Resumen

En el presente artículo se explicará sobre el ORM que sirve para la conversión de datos a un formato específico para almacenarse en una base de datos. Además, se mencionarán los motivos principales para el uso de ORM, siendo el principal de estos la gran simplificación en el manejo de la base de datos. Se explicarán tres tecnologías ORM que se usan en la actualidad: Hibernate, Eclipse y Mybatis. Hibernate bajo licencia GNU LGPL para Java, que facilita el mapeo de atributos en una base de datos tradicional, busca solucionar el problema de la diferencia entre el modelo POO y el modelo relacional detallando cómo es su modelo de datos, qué relaciones existen y qué forma tienen.EclipseLink que provee un extensible framework que permite a los programadores interactuar con varios servicios de datos. Y Mybatis que es un framework de persistencia que soporta SQL, procedimientos almacenados y mapeos avanzados.

Palabras Clave: ORM, Mapeo, base de datos, Hibernate, Eclipse, Mybatis, SQL

Abstract

This article will explain about the ORM that is a tool that is used to convert data to a specific format to be stored in a database. In addition, the main reasons for the use of ORM will be mentioned, the main one being the great simplification in the management of the database. Three ORM technologies in use today will be explained: Hibernate, Eclipse, and Mybatis. Hibernate under the GNU LGPL license for Java, which facilitates the mapping of attributes in a traditional database, seeks to solve the problem of the difference between the OOP model and the relational model by detailing what its data model is like, what relationships exist and what form it has. EclipseLink that provides an extensible framework that allows programmers to interact with various data services. And Mybatis which is a persistence framework that supports SQL, stored procedures and advanced mappings.

Keywords: ORM, Mapping, database, Hibernate, Eclipse, Mybatis, SQL

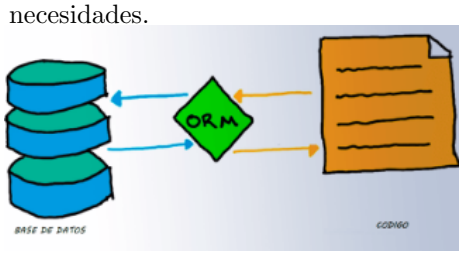
1 Introducción

Todo programador que desarrolla software de aplicación tal como es el software empresarial, educativo, videojuegos, medicina, telecomunicaciones, automatización industrial, diseño asistido, etc., tarde o temprano se enfrenta con la necesidad de guardar la información que actúa como entrada o salida de los sistemas que desarrolla, busca que esta información persista incluso después de que la aplicación termina su ejecución. Los frameworks de mapeo objeto relacional u ORM por sus siglas en inglés (Object-Relational Mapping) juegan un papel muy importante para el mejoramiento del proceso de persistencia de datos entre un lenguaje de programación y una base de datos. Los programadores profesionales siempre pretenden realizar bien su trabajo y además están en constante búsqueda de nuevas herramientas, metodologías, estándares que permitan mejorarlo.

2 ¿Qué es un ORM?

ORM (del inglés Object Relational Mapping). Un ORM te permite convertir los datos de tus objetos en un formato correcto para poder guardar la información en una base de datos (mapeo) creándose una base de datos virtual donde los datos que se encuentran en nuestra aplicación quedan vinculados a la base de datos (persistencia).

Transformar toda la información que recibes de la base datos, principalmente en tablas, en los objetos de tu aplicación y viceversa. A esto se le denomina mapeo. Utilizando un ORM este mapeo será automático, es más, será independiente de la base de datos que estés utilizando en ese momento pudiendo cambiar de motor de base de datos según tus necesidades.



Motivos para usar un ORM:

- Lo primero que el ORM funciona como una capa intermedia totalmente separada de la base de datos. Esto permite que puedas centrarte únicamente en el desarrollo de la aplicación.
- Hace que una migración sea más sencilla. Si en algún momento se quiere cambiar la aplicación, no habría que reconstruir todas las consultas: podríamos migrar esa aplicación a otra base de datos sin problema.
- El programador no necesita saber un lenguaje para cada base de datos: el ORM unifica el lenguaje.
- Nos permite tener una mayor velocidad a la hora de hacer tareas básicas en cuanto al acceso a los datos.
- Es un código más legible y con menos líneas.
- Otra ventaja importante es la seguridad. Al ser una capa independiente a los datos, nos permite protegerlos de ciberataques al no estar en el mismo nivel.
- Nos evita escribir a mano las consultas de SQL necesarias.
- Facilita el trabajo con acciones simples y básicas de acceso a los datos. Como funciones básicas de bases de datos consideramos lo que se conoce como CRUD: Create, Read, Update y Delete.
- Por último y principalmente, una de las ventajas más atractiva para los desarrolladores a la hora de usarlo es que un ORM nos guarda y carga toda la información de una base de datos relacional automáticamente.

Principales ORMs en el mercado actual:

- Java: Hibernate, MyBatis, iBatis, Ebean, etc.
- .NET: Entity Framework, nHibernate, MyBatis.Net, etc.
- PHP: Doctrine, Propel, Rocks, Torpor, etc.
- Python: Peewee, SQLAlchemy, PonyORM, Elixir, etc.

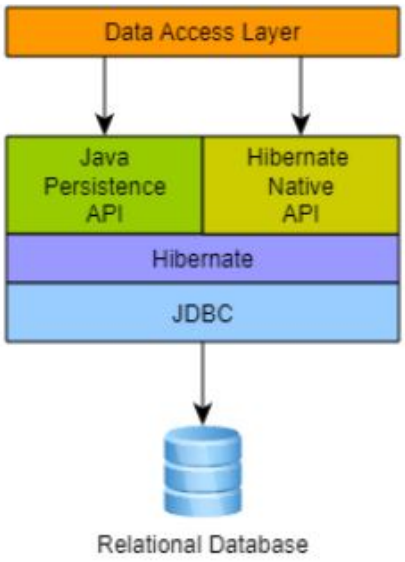
3 Comparativa de tecnologías ORM

3.1 Hibernate

Basicamente es una herramienta de mapeo objeto-relacional (ORM) bajo licencia GNU LGPL para Java, que facilita el mapeo de atributos en una

base de datos tradicional, y el modelo de objetos de una aplicación mediante archivos declarativos o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

3.1.1 Arquitectura de Hibernate



Hibernate se encuentra entre la capa de acceso a datos de la aplicación Java y la base de datos relacional, como se puede ver en el diagrama anterior. La aplicación Java hace uso de las API de Hibernate para cargar, almacenar, consultar, etc. datos de dominio.

3.1.2 Sobre su funcionamiento

En la mayoría de programas actuales existen dos modelos de datos que coexisten: el usado en la memoria de la computadora (orientación a objetos), y el usado en las bases de datos (modelo relacional). Hibernate busca solucionar el problema de la diferencia entre ambos modelos. Para lograr esto, nos permite detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen mediante un documento XML, o mediante anotaciones donde corresponde un atributo de una clase, con una columna de una tabla. En la actualidad es una tarea simple, pues existen herramientas que lo hacen por nosotros.

Hibernate permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por 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, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

3.2 EclipseLink

Es proyecto open source (Eclipse Persistence Services) de Eclipse Foundation. Provee un extensible framework que permite a los programadores interactuar con varios servicios de datos, incluyendo bases de datos, web services, Object XML mapping (OXM), and Enterprise Information Systems (EIS)

Proporciona una implementación JPA completa y compatible con JPA.Proporciona un cumplimiento completo de todas las funciones obligatorias y muchas de las funciones opcionales.También admite funciones de EclipseLink que no se describen en la especificación JPA, como caché a nel de objeto, coordinación de caché distribuida, amplias opciones de ajuste de rendimiento, compatiidad mejorada con Oracle Database, asignaciones avanzadas, opciones de bloqueo optimistas y pesimistas anotaciones extendidas y sugerencias de consulta.

EclipseLink es considerado como el framework sucesor de TopLink pues está basado en el código fuente y las pruebas realizadas por Oracle a este proyecto antes de donarlo a la fundación de Eclipse.

EclipseLink admite una serie de estándares de persistencia que incluyen:

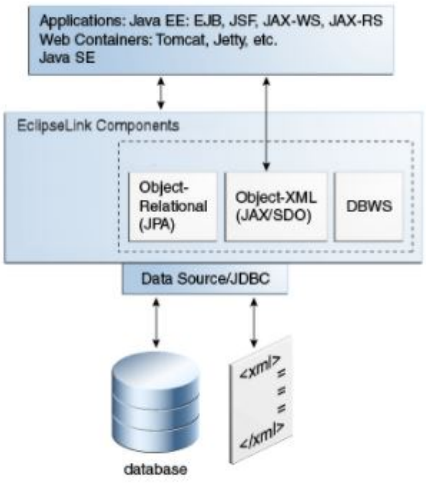
- API de persistencia de Java (JPA).
- Arquitectura Java para enlaces XML (JAXB).
- Arquitectura de conector Java (JCA)
- Objetos de datos de servicio (SDO)

EclipseLink ofrece soporte para:

- Objeto - Relacional (JPA)
- NoSQL (base de datos NoSQL y EIS)
- MOXy: Objeto-XML (JAXB) y Objeto-JSON

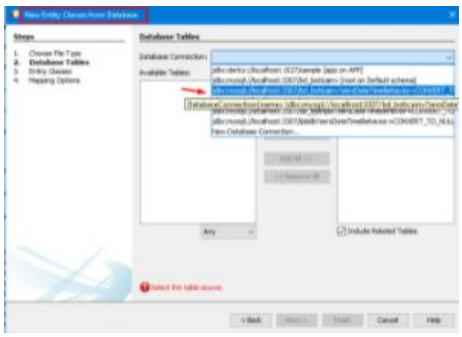
- DBWS: Servicios web de base de datos
- Objetos de datos de servicio (SDO)

3.2.1 Arquitectura de EclipseLink



3.2.2 Implementación

- La librería de eclipse link se agrega en el Build Path del proyecto. El proyecto contiene ahora todos los bloques necesarios para acceder a la base de datos.
- Ahora, debemos de generar Entidades desde las tablas, aquí seleccionaremos la coxión con la base de datos y podemos seleccionar las tablas de la base de datos.



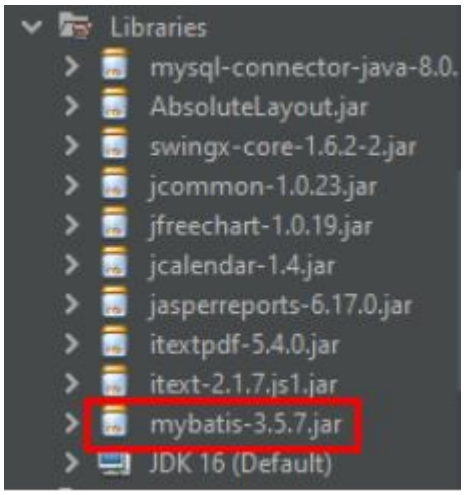
3.3 MyBatis

Es un framework de persistencia que soporta SQL, procedimientos almacenados y mapeos avanzados. MyBatis elimina casi todo el código JDBC, el establecimiento manual de los parámetros y la obtención de resultados. MyBatis puede configurarse con XML o anotaciones y permite mapear mapas y POJOs (Plain Old Java Objects) con registros de base de datos.

La potencia de MyBatis reside en los Mapped Statements. Aquí es donde está la magia. Para lo potentes que son, los ficheros XML de mapeo son relativamente simples. Sin duda, si los comparas con el código JDBC equivalente comprobarás que ahorras el 95

Los ficheros XML de mapeos SQL solo tienen unos pocos elementos de alto nivel (en el orden en el que deberían definirse):

- cache – Configuración de la caché para un namespace.
- cache-ref – Referencia a la caché de otro namespace.
- resultMap – El elemento más complejo y potente que describe como cargar tus objetos a partir de los ResultSets.
- parameterMap – Deprecada! Antigua forma de mapear parámetros. Se recomienda el uso de parámetros en línea. Este elemento puede ser eliminado en futuras versiones. No se ha documentado en este manual.
- sql – Un trozo de SQL reusable que puede utilizarse en otras sentencias.
- insert – Una sentencia INSERT.
- update – A Una sentencia UPDATE.
- delete – Una sentencia DELETE.
- select – Una sentencia SELECT.
- Para usar MyBatis sólo tienes que incluir el ficheromybatis-x.x.x.jaren el classpath. Puedes obtenerlo en : https://blog.mybtis.org



3.3.1 Pasos para usarlo en el proyecto

Se crea un fichero de configuración XML contiene la configuración delcorede MyBatis, incluyendo el DataSource para obtener instancias de conexión a la base de datos y tambi un TransactionManager para determinar cómo deben controlarse las transacciones

```
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD 3.5.7//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
    <environments default="development">
        <envoronment id="development">
            <transactionmanager type="JDBC"/>
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
                <property name="url" value="jdbc:mysql://localhost:3306/bd_boticainv"/>
                <property name="username" value="root"/>
                <property name="password" value="x8y4o2s5s7y6nx"/>
            </dataSource>
        </envoronment>
    </environments>
    <mappers>
        <mapper resource="mybatis/mappers/ContactoMappers.xml"/>
    </mappers>
</configuration>
```

Creamos una clase para una instancia SqlFactory. Se puede obtener una instancia de SqlSessionFactory mediante un SqlSessionFactoryBuilder. Un SqlSessionFactoryBuilder puede construir una instancia de SqlSessionFactory a partir de un fichero de configuración XML o de una instancia personalizada de la clase Configuration.

```
public class MybatisUtil {
    private String resource="Mybatis/Mybatis-config.xml";
    private SqlSession session = null;

    public SqlSession getSession(){
        try{
            Reader reader = Resources.getResourceAsReader(resource);
            SqlSessionFactory sqlMapper = new SqlSessionFactoryBuilder().build(reader);
            session = sqlMapper.openSession();
        }catch(IOException ex)
        {
            ex.printStackTrace();
        }
        return session;
    }
```

Creamos una clase mapeada. LosMapped Statementsson una materia muy densa, para que te hagas una idea de qué se está ejecutando realmente proporcionaremos como lo usamos en nuestro proyecto.

y por ultimo creamos una interface para los metodos en la clase mapeada

```
public interface ContMapper {
    void insertarEmpleado(ClsEEmpleado empleado);
    List<ClsEEmpleado>getEmpleados();
}
```

4 Conclusiones

- El uso de las ORM podría ser una alternativa cuando quisiéramos trasladar un modelo conceptual a un esquema relacional de las bd. a su vez este deberíamos usar ORM según las particularidades de cada problema que vamos a modelar.
- ventajas que ofrecen los ORM se encuentran: rapidez en el desarrollo, abstracción de la base de datos, reutilización, seguridad, mantenimiento del código, lenguaje propio para realizar las consultas.
- Hibernate es una implementación muy popular de JPA. No implementa JPA2.2 pero tiene casi todas sus características mientras que eclipse es una implementación de código abierto de JPA 2.2
- Se podría decir que Hibernate es muy maduro y está bien documentado, mientras que Eclipse no es muy maduro, pero también está bien documentad.
- En el caso de MyBatis es eficiente, admite construcción SQL dinámica y compleja, admite integración con transacciones Spring y AOP, el conjunto de resultados tiene un paquete ligero de Mapper y admite almacenamiento en caché, pero, la base de datos multiplataforma no es compatible, aún necesita escribir instrucciones SQL.

5 Recomendaciones

- Para la implementacion de hibernate en apache netbeans 12.3 se debe de tener en cuenta la version 238 y el unpack 200.
- Se debe verificar que la conexión o el driver esten correctamente conectados.
- Debemos de verificar que todas nuestras librerias y JAR necesarias esten añadidas al classpath del nuestro proyecto.

References

[1] Gracia del Busto, IH y Yanes Enriquez, IO (2012). Mapeo Objeto / Relacional (ORM). Telemática , 10 (3), 1–7. Obtenido de https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/23

[2] ESIC BUSINESS & MARKETING SCHOOL, (2018).El ORM como herramienta eficiente de trabajo, Obtenido de https://www.esic.edu/rethink/tecnologia/el-orm-como-herramienta-eficiente-de-trabajo

[3] mybatis, (2021). Ficheros XML de mapeo. Obtenido de https://mybatis.org/mybatis-3/es/sqlmap-xml.html

[4] Jlebrijo. (2010). JPA: Implementando La Persistencia Con EclipseLink. obtenido de https://blog.lebrijo.com/jpa-implementando-la-persistencia-con-eclipselink/

[5] Callejas, M., Peñaloza, D., & Alarcón, A. (2011, junio). valuación y análisis de rendimiento de los frameworks de persistencia Hibernate y Eclipselink. https://doi.org/10.30554/ventanainform.24.155.2011

[6] The Eclipse Foundation. (2017, 30 octubre). Comprensión de EclipseLink, 2.4. Comprensión de EclipseLink, 2.4. https://www.eclipse.org/eclipselink/documentation/2.4/concepts/blocks001.htm

[7] Equipo Geek. (2019). ¿Qué es Java Hibernate? ¿Por qué usarlo?. obtenido de https://ifgeekthen.everis.com/es/que-es-java-hibernate-por-que-usarlo