

UA05. ORM

5.1. Herramientas

Object Relational Mapping (ORM) es la herramienta que nos sirve para transformar representaciones de datos de los Sistemas de Bases de Datos Relacionales, a representaciones (Modelos) de objetos. Dado a que los RDBMS carecen de la flexibilidad para representar datos no escalares, la existencia de un ORM es fundamental para el desarrollo de sistemas de software robustos y escalables.

En el modelo relacional, cada fila en la tabla se mapea a un objeto y cada columna a una propiedad.

Las herramientas ORM pues, actúan como un puente que conecta las ventajas de los **RDBMS**

<https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos_relacionales> con la buena representación de estos en un lenguaje Orientado a Objetos, o, dicho en otras palabras, nos lleva de la base de datos al lenguaje de programación.

Características

Las herramientas ORM facilitan el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos **XML o anotaciones** que permiten establecer estas relaciones. Gracias a las ORM, podemos conectar con una base de datos relacional para extraer la información contenida en objetos de programas que están almacenados. Para ello, sólo tendremos que definir la forma en la que establecer la correspondencia entre las clases y las tablas una sola vez (indicando qué propiedad se corresponde con cada columna, qué clase con cada tabla,

etc.). Una vez hecho esto, podremos utilizar **POJO's**

<<https://www.arquitecturajava.com/que-es-un-pojo-en-java/>> de nuestra aplicación e indicar a la ORM que los haga persistentes, consiguiendo que una sola herramienta pueda leer o escribir en la base de datos utilizando **VO's**

<<https://leanmind.es/es/blog/primitive-obsesion-value-objects/>> directamente.

Una herramienta ORM permite tomar un objeto Java y hacerlo persistente, carga el objeto de la base de datos a memoria y permite hacer consultas a las tablas de la base de datos.

Ventajas de ORM

- Ayudan a reducir el tiempo de desarrollo de software. La mayoría de las herramientas ORM disponibles, permiten la creación del modelo a través del esquema de la base de datos, es decir, el usuario crea la base de datos y la herramienta automáticamente lee el esquema de tablas y relaciones y crea un modelo ajustado.
- Abstracción de la base de datos.
- Reutilización.
- Permiten persistir objetos a través de un método `save` o `persist` y generar el SQL correspondiente.
- Permiten recuperar los objetos persistidos a través de un método `load` o `get`.
- Lenguaje propio para realizar las consultas.
- Independencia de la base de datos.
- Incentivan la portabilidad y escalabilidad de los programas de software.

Desventajas de ORM

- Tiempo utilizado en el aprendizaje. Este tipo de herramientas suelen ser complejas por lo que su correcta utilización lleva un tiempo que hay que emplear en ver el funcionamiento correcto y ver todo el partido que se le puede sacar.
- Menor rendimiento (aplicaciones algo más lentas). Esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero deberá de transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos.

- Sistemas complejos. Normalmente la utilidad de ORM desciende con la mayor complejidad del sistema relacional.

Herramientas más utilizadas

Entre las herramientasORM más relevantes encontramos las siguientes:

- **Hibernate** <<https://hibernate.org/orm/>> es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de **NHibernate**) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Utiliza archivos declarativos (**XML**) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU **LGPL**.
- El **Java Persistence API (JPA)** <<https://www.oracle.com/java/technologies/persistence-jsp.html>> es una especificación de Sun Microsystems para la persistencia de objetos Java a cualquier base de datos relacional. Esta API fue desarrollada para la plataforma **JEE** e incluida en el estándar de EJB 3.0, formando parte de la Java Specification Request JSR 220.
- **iBatis** <<https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/98>> es un framework de persistencia desarrollado por la Apache software Foundation (actualmente fuera de ella). Al igual que que el resto de los proyectos desarrollados por la ASF, iBatis es una herramienta de código libre. iBatis sigue el mismo esquema de uso que Hibernate; se apoya en ficheros de mapeo XML para persistir la información contenida en los objetos en un repositorio relacional.

Obra publicada con **Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0** <<http://creativecommons.org/licenses/by-nc-sa/4.0/>>