

## Tarea 02-

### ► Concepto JPA

Java Persistence API (JPA) es la tecnología estándar de Java para gestionar entidades persistentes. Este enfoque intenta aplicar las ideas de la POO a las Bases de datos, de forma que las clases y los objetos de una aplicación puedan ser almacenados, modificados y buscados de forma eficiente en unidades de persistencia. Pero a pesar de existir bases de datos orientadas a objetos de forma nativa, éstas no tuvieron mucha efectividad ya que la tecnología dominante en lo referente a bases de datos siempre han sido los sistemas de gestión de bases de datos relacionales (RDBMS). De ahí que la solución propuesta para conseguir entidades persistentes hay sido realizar un mapeado del modelo de objetos al modelo relacional. JPA es una de estas tecnologías.

Además, se debe tener claro que JPA es solamente un estándar que necesita ser implementado por desarrolladores o empresas, existen bastantes implementaciones de JPA en forma de librerías Java disponibles para incluir en aplicaciones de escritorio o aplicaciones web. Entre algunas de las más populares encontramos Hibernate, Apache OpenJPA, Oracle TopLink, Cocobase PURE POJO, etc.

La forma de conseguir que las entidades sean persistentes es utilizando el denominado mapeo entidad-relación (ORM por sus siglas en inglés) donde se mapea cada entidad a una tabla SQL con los mismos atributos, pero no sólo eso, ya que JPA también implementa el mapeo de relaciones, tanto de herencia entre entidades, las clases embebidas o las múltiples relaciones entre entidades, como relaciones uno-a-uno, uno-a-muchos o muchos-a-muchos.



## ► Hibernate

Hibernate es un Framework 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, de tal modo que agiliza la relación entre la aplicación y nuestra base de datos SQL, de un modo que optimiza nuestro flujo de trabajo evitando caer en código repetitivo.

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, también generará 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.

Para programar con Hibernate, debemos considerar diferentes objetos, como lo son el Objeto de Configuración Hibernate, que representa un archivo de configuración para la conexión a la base de datos y el Mapeo de Clase. Además el Session Factory que permite crear un objeto Session a partir del archivo de configuración, después la Session se utiliza para obtener una conexión física con una base de datos, los objetos Transaction opcionales para gestionar o no las transacciones. El objeto Query que nos ayuda a realizar las consultas en la base de datos y crear objetos, y finalmente el objeto Criteria que se utilizan para crear y ejecutar consultas con objetos y recuperar objetos.

## ► Versant

Versant es un sistema gestor de Base de Datos Orientado a Objetos desarrollado por Versant Corporation, que simplifica el proceso de construir y lanzar aplicaciones transaccionales y distribuidas, que permite la definición personalizada de tipos de datos complejos, encapsulación de data y código, herencia de data y código, Reutilización de código, Polimorfismo, Identificación única de objetos, Almacenamiento persistente de información, Accesos concurrentes, manejo de transacciones.

Versant nos da la posibilidad de utilizar diferentes interfaces como son Java y C++, por lo que ~~atraves~~ de ellas podemos hacer uso de ellas. Además hace uso de XML para realizar las especificaciones de objetos.

Finalmente, podemos decir que Versant integra ambos **conceptos** (POO y BD) en un único modelo para almacenar y administrar la información, con el objetivo de simplificar el desarrollo de aplicaciones transaccionales.

## ► Bibliografía.

- <https://ifgeekthen.everis.com/es/gce-es-jpa-hibernate-porque-usarlo>
- <https://www.jtech.es/j2ee/2010-2011/restringido/jpa/sem03-apuntes.html>
- <https://silo.tips/download/versant-object-database-fundamentals-manual>
- [https://en.wikipedia.org/wiki/Versant\\_Object\\_Database](https://en.wikipedia.org/wiki/Versant_Object_Database).