## ORM (Object-Relational Mapping - Mapeo Objeto-Relacional)

**ORM** es una técnica de programación que facilita la conversión de datos entre un sistema de tipos de lenguaje de programación orientado a objetos (como Java) y una base de datos relacional. En esencia, actúa como un puente, permitiendo a los desarrolladores interactuar con los datos de la base de datos utilizando objetos del lenguaje de programación en lugar de escribir sentencias SQL directamente.

El problema que resuelve el ORM es la "discrepancia de impedancia objeto-relacional". Las bases de datos relacionales organizan los datos en tablas con filas y columnas, mientras que los lenguajes orientados a objetos estructuran los datos en objetos con sus propiedades y relaciones. Un ORM se encarga de traducir esta diferencia, mapeando automáticamente las clases a tablas, los objetos a filas y las propiedades de los objetos a columnas, y viceversa.

## DAO (Data Access Object - Objeto de Acceso a Datos)

Un **DAO** es un patrón de diseño que se utiliza para separar la lógica de acceso a datos de la lógica de negocio de una aplicación. Su objetivo principal es encapsular todas las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) relacionadas con una entidad específica.

Mientras que un ORM se encarga de la traducción *automática* entre objetos y datos relacionales, un DAO es una capa *manual* o *programática* que define cómo se accede y manipula esa información. Es decir, un DAO contendrá métodos como guardar(objeto), obtenerPorId(id), actualizar(objeto), eliminar(objeto), y estos métodos, internamente, podrían usar un ORM o incluso JDBC directamente para interactuar con la base de datos. Un DAO es el responsable de cómo se "habla" con la fuente de datos, sin que el resto de la aplicación tenga que saber los detalles de implementación (si es SQL, NoSQL, etc.).

## Hibernate y JPA

**Hibernate** es una popular **implementación de ORM** para el lenguaje de programación Java. Fue iniciado en 2001 por Gavin King y posteriormente fue desarrollado y mantenido por JBoss, que ahora forma parte de Red Hat.

**JPA (Java Persistence API)**, por otro lado, **es un estándar** (una especificación) de Java para la persistencia de datos relacionales en aplicaciones Java. Define un conjunto de interfaces y anotaciones que permiten a los desarrolladores definir cómo se mapean los objetos Java a las tablas de una base de datos. JPA no es una implementación en sí misma, sino un contrato que las implementaciones de ORM (como Hibernate) deben seguir.

Cuando se dice "Usaremos JPA e Hibernate", significa que se utilizará el estándar JPA para definir cómo se persisten los objetos Java, y Hibernate será la implementación concreta de ORM que se encargará de llevar a cabo esa persistencia. Esto permite que el código sea más portable, ya que se programa contra el estándar JPA, y se podría cambiar la implementación de ORM subyacente (por ejemplo, de Hibernate a EclipseLink) sin tener que reescribir gran parte del código de persistencia.

En resumen:

* **ORM**: Es la **técnica** de mapeo entre objetos y datos relacionales.
* **DAO**: Es un **patrón de diseño** para encapsular la lógica de acceso a datos. Un DAO puede usar un ORM para sus operaciones.
* **JPA**: Es el **estándar** (API) de Java para el mapeo objeto-relacional.
* **Hibernate**: Es una **implementación** concreta del estándar JPA, un ORM muy potente y ampliamente utilizado en el ecosistema Java.