

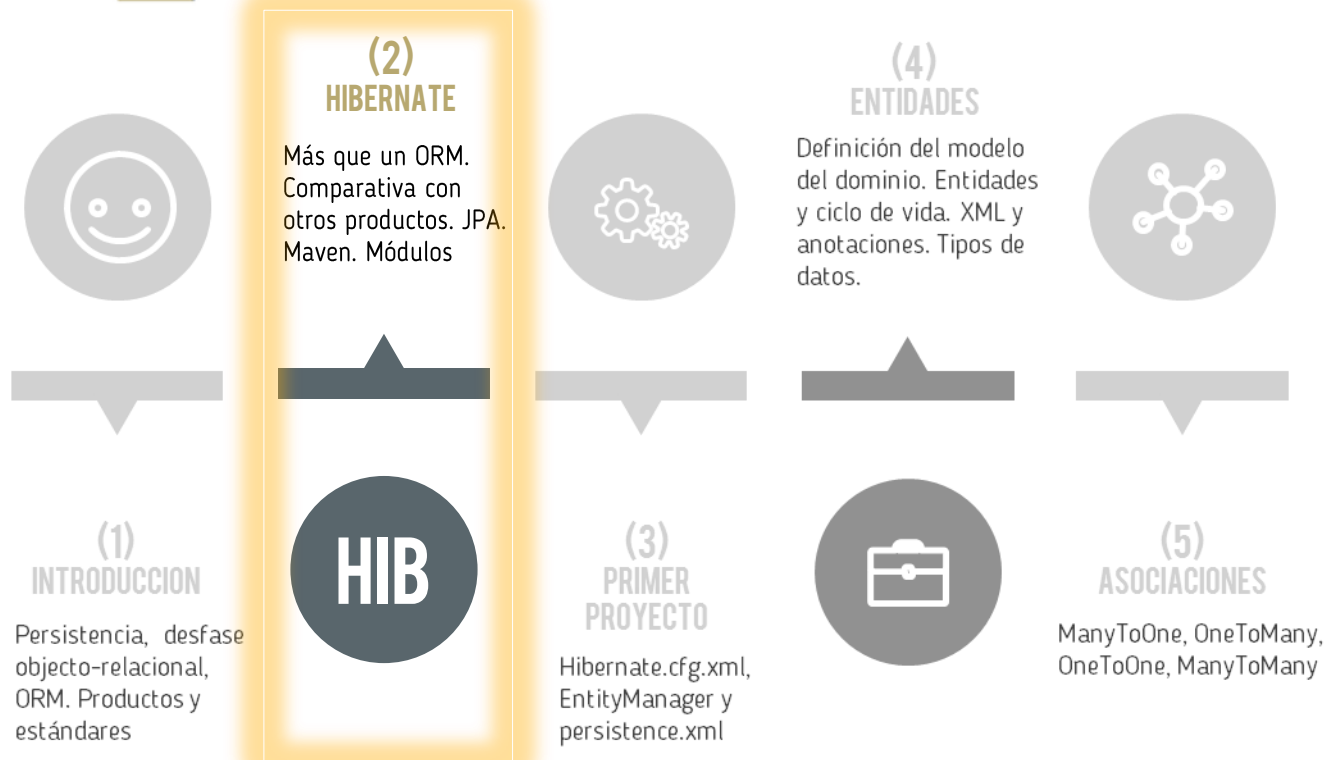


CURSO DE **HIBERNATE 5**


OpenWebinars



HIBERNATE





HIBERNATE

(7)

COLECCIONES



Mapeo de colecciones.
Tipos (list, set, map).
Colecciones ordenadas (sorted vs. ordered).

(9)

CONTEXTO DE PERSISTENCIA



Almacenamiento,
recuperación y borrado
de entidades.



(6)

ELEMENTOS AVANZADOS

Campos calculados,
herencia.



(8)

GENERACION DEL ESQUEMA

Customización del
proceso de
generación del
esquema.



(10)

TRANSACCIONES

Control de concurrencia.
Patrones y antipatrones.



HIBERNATE

(12) ENVERS



Introducción a la
auditoria de entidades.



(11) CONSULTAS HPQL VS JPQL

Consultas con
parámetros,
Anotaciones. SQL nativo



A thick, light blue diagonal line runs from the top right towards the bottom left, separating the white background on the left from the solid blue background on the right.

1.

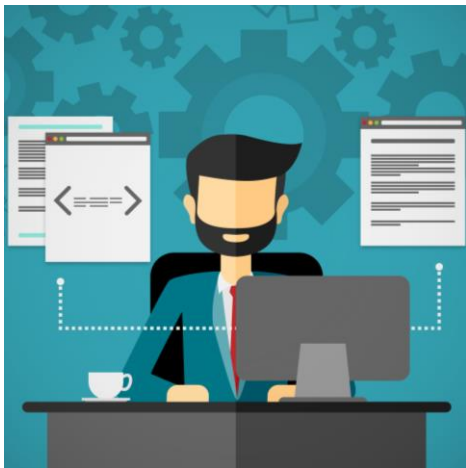
HIBERNATE COMO ORM

¿Qué ventajas
tenemos?



PRODUCTIVIDAD

Nos permite centrarnos en el desarrollo de la lógica de negocio, ahorrándonos una gran cantidad de código.



MANTENIBILIDAD

El ORM de Hibernate, al reducir el número de líneas de código que tenemos que escribir, nos permite realizar refactorizaciones de forma más fácil.



RENDIMIENTO

Aunque la persistencia manual podría permitir un rendimiento mayor, Hibernate aporta muchas optimizaciones, como el almacenamiento en caché.



INDEPENDENCIA

Hibernate nos permite aislarnos del RDBMS concreto que estemos usando.

2.

**HIBERNATE, NO
SOLO UN ORM**

¿Qué otros
módulos existen?

HIBERNATE ENTITYMANAGER

También conocido como *JPA Provider*, este módulo es el que nos permite utilizar Hibernate como implementación del estándar JPA

Hibernate ORM



Domain model persistence for relational databases

More →

HIBERNATE VALIDATOR

Hibernate
proporciona una
implementación del
estándar JSR 303
Bean Validation.
Proporciona una
validación declarativa
para nuestras clases
modelo.

Hibernate Validator

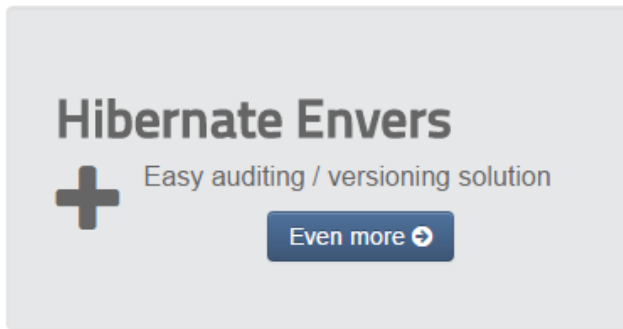


Annotation based constraints for
your domain model

More →

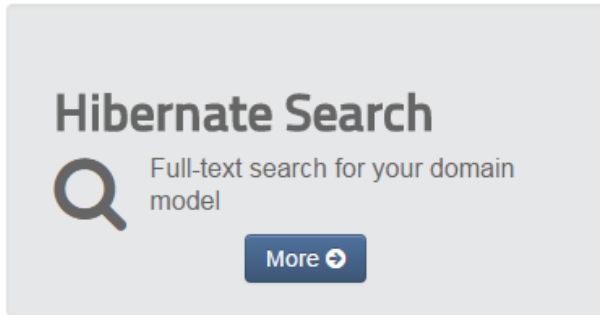
HIBERNATE ENVERS

Nos permite auditar los cambios en nuestra persistencia, manteniendo múltiples versiones de los datos en nuestra base de datos SQL.



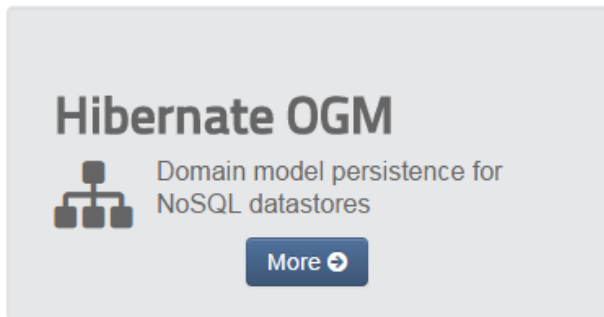
HIBERNATE SEARCH

Podremos realizar búsquedas de texto completo a través de un índice mantenido dentro de una base de datos de Apache Lucene.



HIBERNATE OGM

Se trata del mapeador objeto-grid, y está orientado hacia bases de datos NoSQL. Tiene soporte para su uso con JPA.



3.

HIBERNATE FRENTE A OTROS ORMs

O a lo mejor no
son ORMs



Ventajas

- Mayor rendimiento
- Funcionalidades concretas

Desventajas

- Mayor tiempo desarrollo.
- Mantenimiento costoso.
- Errores durante el desarrollo



HIBERNATE

- Desarrollo más veloz.
- Uso de entidades en lugar de *queries*.
- Eliminación errores ejecución.

- Curva aprendizaje más pronunciada.
- Posibilidad de menor rendimiento, si bien existe posibilidad de *caché*.



MyBatis



HIBERNATE

Es muy simple. El empaquetado ocupa poco espacio

Genera el SQL por nosotros, lo cual significa que no tenemos que invertir nuestro tiempo en hacerlo.

Es flexible, ofrece un tiempo de desarrollo muy rápido

Es altamente escalable, y ofrece una caché más avanzada.

Utiliza SQL, que si bien es un estándar, puede tener elementos concretos de un RDBMS

Utiliza HQL (o JPQL), que es más independiente del RDBMS utilizado. Es fácil cambiar de base de datos con Hibernate.

Mapea un *ResultSet* del API JDBC en nuestros objetos POJO, de forma que no debemos preocuparnos por las tablas relacionales.

Hibernate mapea nuestros objetos POJO en tablas de una base de datos.

Es fácil usar procedimientos almacenados

Es algo dificultoso usar procedimiento almacenados.

4.

HIBERNATE y JPA

¿Somos
distintos?
¿Somos iguales?

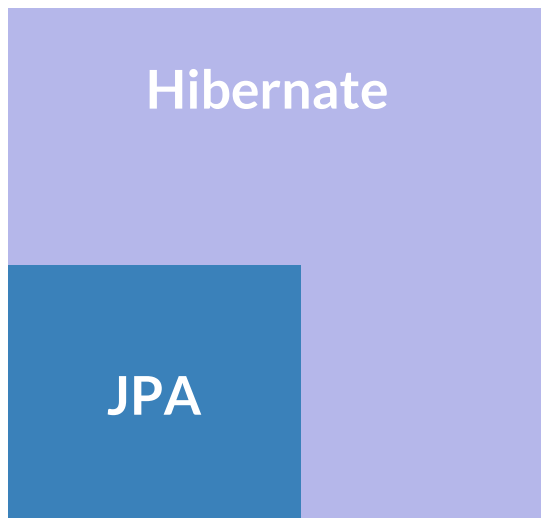
HIBERNATE Y JPA

- ▶ Si JPA es el *baile*, Hibernate es el *bailarín*.
- ▶ Si JPA es el *arte*, Hibernate es el *artista*.
- ▶ JPA es la especificación para estandarizar ORMs en Java.
- ▶ Hibernate es un *vendor* de una implementación del estandar JPA.

ESPECIFICACIÓN JPA

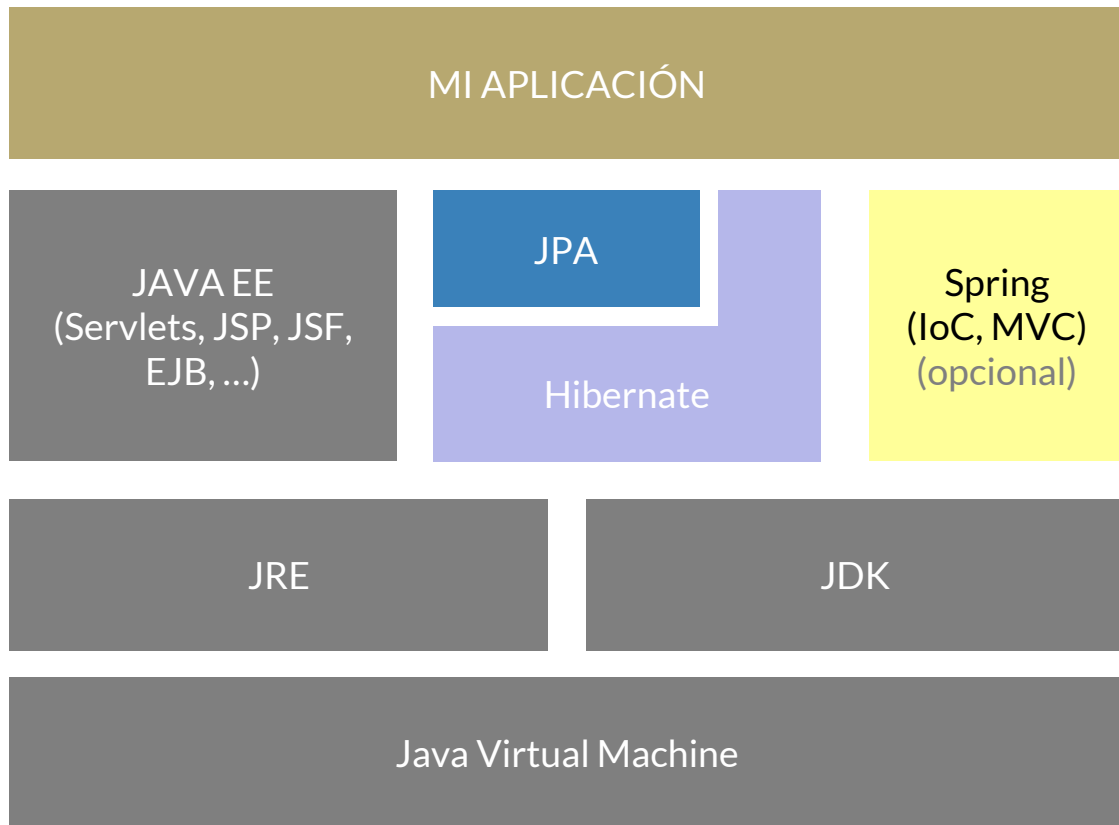
- ▶ Una facilidad para especificar como nuestros objetos Java se relaciona con el esquema de una base de datos
- ▶ Una api sencillo para realizar las operaciones CRUD
- ▶ Un lenguaje y un API para realizar consultas sobre los datos (JPQL).
- ▶ Elementos de optimización (actualización de entidades, captura de asociaciones, caché, ...)

HIBERNATE Y JPA



Las funcionalidades de Hibernate van más allá de la especificación JPA. Además, el documento que define JPA está basado, a grandes rasgos, en como trabaja Hibernate.

ARQUITECTURA TÍPICA





5.

HIBERNATE y
MAVEN

HIBERNATE Y MAVEN

- El uso de Hibernate implica incluir en nuestro proyecto varias librerías.

10 jars

~ 300 packages

~ 6000 clases

HIBERNATE Y MAVEN

- ▶ Maven nos permite gestionar las dependencias mediante un fichero descriptor xml (pom.xml).

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.2.9.Final</version>
</dependency>
```