

# Roadmap

## Índice

|                             |   |
|-----------------------------|---|
| 1 Puntos destacados.....    | 2 |
| 2 Certificación Sun.....    | 2 |
| 3 Recursos adicionales..... | 2 |
| 3.1 Bibliografía.....       | 2 |
| 3.2 Enlaces.....            | 3 |

## 1. Puntos destacados

- Hibernate es una **herramienta ORM** que permite trabajar con la capa de persistencia de una aplicación desde el modelo de objetos definido en capas superiores.
- Para utilizar Hibernate debemos entender bien el proceso de **configuración e inicio de Hibernate**, que nos va a permitir, entre otras cosas, determinar de forma transparente al programador, qué base de datos subyace en nuestra aplicación. Hay un *plug-in* de Hibernate para Eclipse que facilita dicha tarea de configuración, así como el uso general de Hibernate en nuestras aplicaciones J2EE.
- Uno de los elementos centrales de Hibernate es el **mapeado de las clases persistentes**, es decir, la definición de la correspondencia entre las clases del modelo de objetos y las tablas de la base relacional subyacente en la capa de persistencia de datos. Dicho mapeado se realiza en forma de ficheros *xml*.
- Los ficheros de mapeado nos permiten además, especificar las correspondencias entre las **relaciones** de las clases del modelo del objetos (**herencia, composición, asociaciones, ...**) y las tablas de la base de datos.
- La recuperación de objetos persistentes de la base de datos es otra de las partes interesantes de Hibernate. Hibernate proporciona tres formas de recuperar objetos persistentes (objetos que tienen su correspondiente fila en las tablas de datos) de la base de datos: utilizando **HQL** (un **lenguaje de consultas orientado a objetos** propio de Hibernate), utilizando el API **Criteria** de Hibernate, y mediante el uso de **SQL nativo** de la base de datos utilizada. Esta última opción resulta muy útil porque nos permite migrar de forma "limpia" y directa desde una aplicación basada en SQL/JDBH hacia Hibernate.

## 2. Certificación Sun

Sun no menciona explícitamente a Hibernate en ninguna de sus certificaciones. Es bastante lógico por otra parte ya que esta herramienta ORM, aunque muy utilizada en aplicaciones Java, es un estándar. No obstante, la nueva versión de Java EE (Java EE 5), incorpora un modelo de persistencia muy similar al de Hibernate, en su API [JPA](#) (Java Persistence API). Dicho API se incluye en el estándar [JSR 220](#) (EJB 3.0), para simplificar los EJBs de entidad.

## 3. Recursos adicionales

### 3.1. Bibliografía

- [Manual de referencia de Hibernate Tools 3.2.0](#)
- [Manual de referencia de Hibernate 3.1.3](#)
- Hibernate in Action, 2e. *Christian Bauer and Gavin King*. Ed. Manning Publications. 2005. Pinchando [aquí](#) os podéis descargar el código fuente de los ejemplos del libro. También están accesibles los capítulos [segundo](#) y [sexto](#) del libro.
- Professional Hibernate. *Eric Pugh and Joseph D. Gradecki*. Ed. Wrox. 2004. Tiene disponible el código fuente de todos los [ejemplos](#) del libro.

### 3.2. Enlaces

- 
- [Hibernate 3.1.3](#) Contiene, además de las librerías, un manual de referencia en pdf muy útil, que podéis descargar por separado en el apartado de Bibliografía anterior.
  - [HSQLDB 1.8.0](#) Base de datos en memoria. Útil para realizar pruebas.
  - [Hibernate Tools 3.2](#). Plugin de Hibernate para Eclipse

