

Definición de claves primarias en Hibernate

Hibernate proporciona varios mecanismos para definir las claves primarias de las entidades. A continuación se describen estos mecanismos tanto en **archivos de mapeo Hibernate (.hbm.xml)** como en **anotaciones JPA**.

1. Definición de claves en archivos de mapeo .hbm.xml

En los archivos de mapeo XML (.hbm.xml), la clave primaria se identifica mediante la etiqueta <id>, con una subetiqueta <generator>. El tipo de clave se especifica con el atributo “class” dentro de <generator>.

Ejemplo básico en .hbm.xml

```
<id name="id" column="id" type="long">
  <generator class="identity"/>
</id>
```

Valores más habituales del atributo “class” en <generator>:

- `<generator class="identity"/>`: significa que Hibernate delega la generación de claves a la base de datos. Esto es común cuando se tienen columnas con propiedades como AUTO_INCREMENT en MySQL, o SERIAL en PostgreSQL. Hibernate confía en la base de datos para generar un nuevo identificador único cuando se inserta una nueva fila.
 - `<generator class="increment"/>`: Hibernate calcula el próximo valor de la clave primaria incrementando el último valor utilizado. Es una estrategia totalmente gestionada por Hibernate, no por la base de datos. Es útil cuando no tienes un mecanismo de generación automático en la base de datos, pero no es seguro en entornos con múltiples sesiones simultáneas. JPA no soporta este tipo de estrategia.
 - `<generator class="sequence"/>`: usa secuencias de la base de datos para generar valores. Requiere que la base de datos soporte la funcionalidad de secuencias (como en PostgreSQL y Oracle).
 - `<generator class="table"/>`: usa una tabla especial para almacenar los valores generados. Menos eficiente.
 - `<generator class="native"/>`: Hibernate elige automáticamente entre identity, sequence o table, según la base de datos subyacente. Es la opción recomendada cuando se trabaja con múltiples bases de datos.
 - `<generator class="assigned"/>`: no genera ningún ID automáticamente. El desarrollador debe asignar manualmente un valor antes de guardar la entidad.
-

2. Definición de claves en anotaciones JPA

En JPA con Hibernate, la clave primaria de una entidad se identifica mediante la anotación **@Id**. Si la clave es autogenerada, se añade además la anotación **@GeneratedValue(strategy = ...)**.

Ejemplo en JPA con Hibernate

```
@Entity
public class Producto {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    // Otros atributos y métodos...
}
```

Valores más habituales de **@GeneratedValue(strategy = ...)**:

- **@GeneratedValue(strategy = GenerationType.IDENTITY)** : equivalente a `<generator class="identity"/>`, Hibernate delega la generación del ID a la base de datos.
 - **@GeneratedValue(strategy = GenerationType.AUTO)** : equivalente a `<generator class="native"/>`. Hibernate elige automáticamente la mejor estrategia según la base de datos.
 - **@GeneratedValue(strategy = GenerationType.SEQUENCE)** : equivalente a `<generator class="sequence"/>`. Usa una secuencia de la base de datos para generar valores. Requiere **@SequenceGenerator**.
 - **@GeneratedValue(strategy = GenerationType.TABLE)** : equivalente a `<generator class="table"/>`. Usa una tabla especial para gestionar los IDs.
-

3. Equivalencias entre definición de claves en las anotaciones y en los archivos de mapeo.

ANOTACIONES Valor de "strategy" en @GeneratedValue(strategy = ...)	ARCHIVOS DE MAPEO Valor de "class" en <code><generator class="assigned"/></code>	Significado
IDENTITY	identity	Usa AUTO_INCREMENT en MySQL, SERIAL en PostgreSQL, etc.
AUTO	native	Hibernate elige la mejor estrategia.
SEQUENCE	sequence	Requiere una base de datos con soporte de secuencias.
TABLE	table	Usa una tabla especial para gestionar IDs. Menos eficiente.
SIN anotación @GeneratedValue	assigned	El ID debe asignarse manualmente.
No soportado en JPA	increment	Hibernate calcula manualmente el siguiente valor a partir del último valor insertado.