

PROYECTO UF3: HIBERNATE – Por Gerard Luque Oliver

1. Introducción

1.1 Descripción del programa

Este programa utiliza Hibernate, un framework de mapeo objeto-relacional (ORM), para realizar consultas avanzadas a una base de datos que contiene información sobre libros, autores, editoriales y temas. El objetivo principal es mostrar cómo consultar y presentar datos relacionales de forma eficiente utilizando Hibernate Query Language (HQL).

1.2 Objetivo

El código realiza tres tareas principales:

1. Recuperar y mostrar información básica de libros y autores.
2. Mostrar toda la información detallada de libros, autores, editoriales y temas.
3. Realizar consultas basadas en parámetros ingresados por el usuario.

2. Estructura del programa

2.1 Configuración inicial

El programa comienza estableciendo la conexión con la base de datos utilizando una `SessionFactory` y una `Session` proporcionada por Hibernate:

```
SessionFactory sf = HibernateUtil.getSessionFactory();  
Session session = sf.openSession();
```

Explicación:

- **SessionFactory:** Es una fábrica de sesiones que crean objetos `Session` para interactuar con la base de datos.
- **Session:** Representa la conexión con la base de datos y permite ejecutar consultas y transacciones.

2.2 Primera Consulta: Mostrar ISBN, título y nombre del autor

```
String hql = "SELECT l, a FROM Libro l JOIN l.idAutor a";  
Query q = session.createQuery(hql);  
List lista = q.list();  
Iterator iter = lista.iterator();  
System.out.println("---- MUESTRA EL ISBN, EL TÍTULO Y EL NOMBRE DEL AUTOR DE LOS LIBROS ----");  
System.out.println("");  
while (iter.hasNext()) {  
    Object[] partes = (Object[]) iter.next();  
    Libro l = (Libro) partes[0];  
    Autor a = (Autor) partes[1];  
    System.out.println("-> ISBN: " + l.getIsbn()  
        + "-> TÍTULO DEL LIBRO: " + l.getTitulo()  
        + "-> NOMBRE DEL AUTOR: " + a.getNombreAutor());  
}
```

Explicación paso a paso:

Consulta HQL:

```
String hql = "SELECT l, a FROM Libro l JOIN l.idAutor a";
```

Libro l: Se seleccionan las entidades de tipo Libro.

JOIN l.idAutor a: Se realiza una unión con la entidad Autor asociada.

Ejecución de la consulta:

```
Query q = session.createQuery(hql);  
List lista = q.list();
```

La consulta se ejecuta y los resultados se almacenan en una lista.

Iteración y presentación de resultados:

Cada elemento de la lista es un objeto compuesto (Object[]) que contiene las entidades Libro y Autor:

```
while (iter.hasNext()) {  
    Object[] partes = (Object[]) iter.next();  
    Libro l = (Libro) partes[0];  
    Autor a = (Autor) partes[1];  
    System.out.println("-> ISBN: " + l.getIsbn()  
        + "-> TÍTULO DEL LIBRO: " + l.getTitulo()  
        + "-> NOMBRE DEL AUTOR: " + a.getNombreAutor());  
}
```

2.3 Segunda Consulta: Mostrar información detallada sin ID

```
String hql2 = "SELECT l, a, e, t\n" +  
    "FROM Libro l\n" +  
    "JOIN l.idAutor a\n" +  
    "JOIN l.idEditorial e\n" +  
    "JOIN l.idTema t";  
Query q2 = session.createQuery(hql2);  
List lista2 = q2.list();  
Iterator iter2 = lista2.iterator();  
System.out.println(" ---- MUESTRA TODA LA INFORMACIÓN DE LIBROS Y AUTORES PARA EL USUARIO  
FINAL ----");  
System.out.println("");  
while (iter2.hasNext()) {  
    Object[] partes2 = (Object[]) iter2.next();  
    Libro l2 = (Libro) partes2[0];  
    Autor a2 = (Autor) partes2[1];  
    Editorial e = (Editorial) partes2[2];  
    Tema t = (Tema) partes2[3];  
    System.out.println("-> ISBN: " + l2.getIsbn()  
        + "-> TÍTULO: " + l2.getTitulo()  
        + "-> Nº EJEMPLARES: " + l2.getNumeroEjemplares()  
        + "-> NOMBRE AUTOR: " + a2.getNombreAutor()  
        + "-> EDITORIAL: " + e.getNombreEditorial()  
        + "-> TEMA: " + t.getNombreTema()  
    );  
}
```

Explicación paso a paso:

Consulta HQL:

```
String hql2 = "SELECT l, a, e, t\n" +  
    "FROM Libro l\n" +  
    "JOIN l.idAutor a\n" +  
    "JOIN l.idEditorial e\n" +  
    "JOIN l.idTema t";
```

Recupera información de las entidades Libro, Autor, Editorial y Tema.

Iteración y presentación de resultados:

Cada elemento de la lista es un objeto compuesto (Object[]) que contiene las entidades Libro, Autor, Editorial y Tema. Los resultados contienen datos relacionados, presentados sin mostrar identificadores:

```
System.out.println("");  
while (iter2.hasNext()) {  
    Object[] partes2 = (Object[]) iter2.next();  
    Libro l2 = (Libro) partes2[0];  
    Autor a2 = (Autor) partes2[1];  
    Editorial e = (Editorial) partes2[2];  
    Tema t = (Tema) partes2[3];  
    System.out.println("-> ISBN: " + l2.getIsbn()  
        + " -> TITULO: " + l2.getTitulo()  
        + " -> Nº EJEMPLARES: " + l2.getNumeroEjemplares()  
        + " -> NOMBRE AUTOR: " + a2.getNombreAutor()  
        + " -> EDITORIAL: " + e.getNombreEditorial()  
        + " -> TEMA: " + t.getNombreTema()  
    );  
}
```

2.4 Tercera Consulta: Filtro por ISBN

```
String isbnBuscado = "3238845533";  
String hql3 = "SELECT l, a, e, t FROM Libro l "  
    "JOIN l.idAutor a JOIN l.idEditorial e JOIN l.idTema t WHERE l.isbn = :isbn";  
Query q3 = session.createQuery(hql3);  
q3.setParameter("isbn", isbnBuscado);  
List lista3 = q3.list();  
Iterator iter3 = lista3.iterator();  
System.out.println(" ---- INFORMACIÓN DE LIBROS Y AUTORES PARA EL USUARIO FINAL ----");  
System.out.println("");  
while (iter3.hasNext()) {  
    Object[] partes3 = (Object[]) iter3.next();  
    Libro l3 = (Libro) partes3[0];  
    Autor a3 = (Autor) partes3[1];  
    Editorial e = (Editorial) partes3[2];  
    Tema t = (Tema) partes3[3];  
    System.out.println("-> ISBN: " + l3.getIsbn()  
        + " -> TITULO: " + l3.getTitulo()  
        + " -> Nº EJEMPLARES: " + l3.getNumeroEjemplares()  
        + " -> NOMBRE AUTOR: " + a3.getNombreAutor()  
        + " -> EDITORIAL: " + e.getNombreEditorial()  
        + " -> TEMA: " + t.getNombreTema()  
    );  
}  
}
```

Explicación paso a paso:

Consulta HQL con parámetro:

```
String hql3 = "SELECT l, a, e, t FROM Libro l " +  
"JOIN l.idAutor a JOIN l.idEditorial e JOIN l.idTema t WHERE l.isbn = :isbn";
```

Utiliza un parámetro para filtrar la consulta por ISBN.

Configuración del parámetro:

```
q3.setParameter("isbn", isbnBuscado);
```

Asocia el valor proporcionado por el usuario a la consulta para evitar inyecciones SQL.

Presentación de resultados:

La estructura es similar a las consultas anteriores, pero con resultados filtrados.

```
while (iter3.hasNext()) {  
    Object[] partes3 = (Object[]) iter3.next();  
    Libro l3 = (Libro) partes3[0];  
    Autor a3 = (Autor) partes3[1];  
    Editorial e = (Editorial) partes3[2];  
    Tema t = (Tema) partes3[3];  
    System.out.println("-> ISBN: " + l3.getIsbn()  
        + " -> TITULO: " + l3.getTitulo()  
        + " -> Nº EJEMPLARES: " + l3.getNumeroEjemplares()  
        + " -> NOMBRE AUTOR: " + a3.getNombreAutor()  
        + " -> EDITORIAL: " + e.getNombreEditorial()  
        + " -> TEMA: " + t.getNombreTema()  
    );  
}
```