

# Ejercicios de Consultas: HQL y SQL

## Índice

|                                   |   |
|-----------------------------------|---|
| 1 Ejercicio 1: Consultas HQL..... | 2 |
| 2 Ejercicio 2: Consultas SQL..... | 3 |

Se proporciona el fichero **hib-sesion4.doc** como plantilla para que contestéis a las preguntas formuladas en los ejercicios.

Vamos a utilizar el proyecto `hib-sesion4`, que se encuentra en el fichero **sesion04-ejercicios.zip**. En dicho proyecto realizaremos las modificaciones que se piden en los ejercicios propuestos.

## 1. Ejercicio 1: Consultas HQL

Utilizando el lenguaje HQL, se pide:

- Implementa los métodos *private List listPujas()*, *private List listPersons()*, y *private List listArticulos()*, en los ficheros *PujaManager.java*, *PersonManager.java* y *ArticuloManager*, respectivamente. Dichos métodos devuelven una lista con todas las *pujas*, *personas* y *artículos* respectivamente. Los botones etiquetados como "Print Pujas", "Print persons" y "Print Articulos" os servirán para probar los resultados de las consultas.
- Implementa el método *public String print\_pujasDePerson(Long idPerson)* en el fichero *PersonManager.java* que, dado el identificador de una persona, **recupera todas las pujas asociadas a dicha persona**. Asocia el método implementado al botón con nombre *PRINT PUJAS HQL*. En el área de texto del interfaz gráfico, se imprimirá la siguiente información: el identificador y el nombre de la persona, para cada una de las pujas: el identificador y el nombre del artículo, la cantidad pujada, y la fecha de la puja. Así, por ejemplo, para la persona con identificador 3, la información que se mostrará por pantalla será:

Listado de Pujas de la persona: 1

Nombre persona: Pepa

ARTICULO(ID) CANTIDAD FECHA

-----  
Bufanda (2) 1000 12/oct/2007

Gorra (3) 100 14/ago/2006

Y para la persona con identificador 5 se mostrará por pantalla:

Listado de Pujas de la persona: 5

La persona 5 NO ha hecho ninguna puja

- Implementa el método *public List hql\_pujas(Long cantidad)* en el fichero *PujaManager.java* que, dada una cantidad monetaria, **recupera todas las pujas cuya**

**cantidad pujada supere o iguale dicha cantidad.** Las pujas estarán ordenadas ascendentemente por cantidad y por fecha de puja. La información concreta a recuperar para cada puja será: la cantidad monetaria, la fecha de la puja, el nombre e identificador del artículo pujado, y el nombre de la persona que realizó la puja.

- Implementa el método *public String print\_hql\_pujas\_on\_Console(Long cantidad)* en el fichero *PujaManager.java*, que imprima en el área de texto del interfaz gráfico la información de la consulta anterior. Asocia el método implementado al botón con nombre *PRINT PUJAS HQL*. Así, por ejemplo, si llamamos al método con la cantidad 300, aparecerá por pantalla lo siguiente:

Listado de Pujas mayores o iguales a 300 euros:

CANTIDAD FECHA ARTICULO(ID) PUJADOR

-----  
300 10/oct/2007 Mechero (1) Juan

1000 12/oct/2007 Bufanda (2) Pepa

4000 05/dic/2007 Libro (5) Maria

Si no hay pujas superiores o iguales a la cantidad deseada, aparecerá el mensaje: "No se han encontrado pujas".

- **OPCIONAL.** Implementa el método *public List hql\_getArticulosByEstado()* en el fichero *PujaManager.java* que obtenga, de entre todas las pujas, el número de artículos pujados con un determinado estado, y el nombre de dichos artículos. Implementa el método *public String print\_hql\_getArticulosByEstado()*, que imprima en el área de texto del interfaz gráfico el resultado de la consulta anterior. El formato es el siguiente:

Resultado de la consulta:

ESTADO ARTICULO NÚMERO DE PUJAS

-----  
Casi nuevo Mechero 3

Impecable Bufanda 2

## 2. Ejercicio 2: Consultas SQL

Utilizando el lenguaje SQL, implementa el método *public String print\_pujasDePerson\_SQL(Long idPerson)* en el fichero *PersonManager.java*. Es el mismo

método que en el ejercicio 1, pero sustituye la *query* HQL por SQL. Utiliza una SQL con nombre. El nombre de la *query* será *print\_pujasDePerson\_SQL* y recuerda que debes incluirlo en el fichero de mapeado, en este caso, en el fichero *Person.hbm.xml*. Asocia el método implementado al botón con nombre *PRINT PUJAS SQL*.

#### Nota

En este ejercicio utiliza el tipo *BigInteger* para los elementos que hacen referencia al identificador del artículo y la cantidad, y que forman parte del resultado devuelto por la sentencia SQL. ¿Puedes explicar por qué tenemos que hacer ésto cuando en la clase *Person.java* hemos definido estos campos de tipo *Long*?

