

# Ejercicios de consultas, movimientos y restricciones en la consulta

## Índice

1 Consulta simple de una BD.....	2
2 Movimiento por el ResultSet.....	2
3 Actualización de la base de datos.....	2

## 1. Consulta simple de una BD

En el proyecto tenéis una clase llamada *Consultas*. Debéis implementar los métodos correspondientes. Para ello, llamaréis al método del DAO correspondiente y luego mostráis por la consola el listado. Nota: para obtener la fecha llamada a *getDate* y para la hora *getTime*.

Implementad también dos métodos adicionales. El primero realiza una consulta por el aeropuerto de salida y nos devuelve los vuelos que salen de ese aeropuerto. Se mostrarán los vuelos obtenidos. Para indicar el valor de un campo hay que usar la sentencia SQL *where nombre='pepito'*, usando las comillas simples en el caso de cadenas de caracteres. Controlad si no ha habido datos en la consulta. El siguiente método realiza una búsqueda por nombre de hotel, permitiendo introducir un nombre parcial. Para ello, podéis usar *where nombre like '%ote%'*.

## 2. Movimiento por el ResultSet

Vamos a realizar un ejercicio para ver cómo nos podemos mover por el *ResultSet*. Para ello, vamos a centrarnos en el DAO de usuario. Tendremos un objeto *ResultSet* como miembro de la clase e implementaremos los siguientes métodos:

- *public void consultaUsuarios ()*: creará el *ResultSet* ejecutando una sentencia con el tipo de scroll *INSENSITIVE*.
- *public boolean isLastUsuario ()*: devuelve cierto si es el último elemento del *ResultSet*.
- *public boolean isFirstUsuario ()*: devuelve cierto si es el primer elemento del *ResultSet*.
- *public Usuario lastUsuario ()*: devuelve el último usuario.
- *public Usuario firstUsuario ()*: devuelve el primer usuario.
- *public Usuario getNextUsuario ()*: devuelve el siguiente usuario o *null* si el *ResultSet* está vacío.
- *public Usuario getPreviousUsuario ()*: devuelve el usuario anterior o *null* si el *ResultSet* está vacío.

Para poder moveros por el *ResultSet* tenéis que crear la sentencia como *ResultSet.TYPE\_SCROLL\_INSENSITIVE* y *ResultSet.CONCUR\_READ\_ONLY*. El tipo de consulta *SENSITIVE* no funciona en la versión actual de MySQL o del driver.

## 3. Actualización de la base de datos

Vamos a realizar un ejercicio donde actualicemos la BD. Hay que realizar un método, *void actualizaEuros()*, que cambiará el valor en el campo *precio* de la tabla *hotel* convirtiéndolo a

euros. Para ello, primero tenéis que convertir la columna *precio* en real y luego realizar la conversión. Podéis usar los siguientes comandos SQL:

- *alter table hotel change precio precio double(10,3)*
- *update hotel set precio = precio/166.386*

Debéis implementar un método en el DAO hotel. El método devolverá los registros que se han visto afectados y se mostrarán por la salida estándar.