Alquiler de Coches

**Estructura de Tablas Base de Datos**

La base de datos está compuesta por las siguientes tablas:

* **Precio\_Combustible ( Tipo\_Combustible, precio\_por\_litro )**
  + - Tabla de dos filas una con el precio de la gasolina y otra con el del diésel.
* **Modelos ( id\_modelo (PK), nombre, precio\_cada\_dia, capacidad\_deposito, tipo\_combustible (FK 🡪 Precio\_Combustible )**
* **Vehículos ( matricula (PK), id\_modelo (FK → Modelos ), color )**
* **Clientes ( NIF (PK), nombre, ape1, ape2, dirección )**
* **Reservas ( idReserva (PK), cliente (FK 🡪 Clientes ) ), matricula (FK 🡪 Vehículos), fecha\_ini, fecha\_fin).**
* **Facturas ( nroFactura (PK), importe, cliente (FK** 🡪 **Clientes ))**
* **Lineas\_Factura ( NroFactura (FK 🡪 Facturas), concepto, importe ) [PK compuesta NroFactura +Concepto]**

En la carpeta sql, proporcionada, hay un script SQL de carga que borra las tablas y las crea de nuevo, y el cual es llamado de manera automática al ejecutar el método **main**de la clase **Main**. rellena con varias filas de ejemplo. Se utilizan secuencias para implementar la clave primaria de Facturas, Modelos y Reservas.

A su vez, la práctica necesita de la librería ***user\_library***para Eclipse para funcionar adecuadamente. Recuerda que esta librería es la que se configuró en la Sesión 0 de prácticas.

**Descripción General de la práctica**

En src/lsi/ubu/servicios/ServicioImpl.java está el fichero que se te pide completar **empleando sentencias preparadas**. Del mismo, se pide implementar el método:

**public void** alquilar(String nifCliente, String matricula, Date fechaIni, Date fechaFin) **throws** SQLException

Es un método que dado el NIF de un cliente , la matrícula de un vehículo y las fechas de inicio y fin del alquiler inserta una fila en la tabla **Reservas** indicando que ese vehículo está alquilado para esas fechas concretas.

El método lanza sólo excepciones del tipo **AlquilerCochesException**, en las situaciones especificadas a continuación (Ver el fichero src/lsi/ubu/excepciones/AlquilerCochesException.java.):

1. Si el NIF del cliente no existe en la tabla de clientes lanzará una excepción **AlquilerCochesException**, con el código 1, y el mensaje “Cliente inexistente”.
2. Si la matricula no existe en la tabla de vehículos lanzará una excepción **AlquilerCochesException**, con el código 2, y el mensaje “Vehículo inexistente”.
3. Si el número de días es menor que uno lanzará una excepción **AlquilerCochesException**, con el código 3, y el mensaje “El número de días será mayor que cero”.
4. Si ese vehículo ya no está disponible lanzará una excepción **AlquilerCochesException**, con el código 4, y el mensaje “El vehículo no está disponible”.

El resto de las excepciones serán tratadas dentro del propio método. Toda excepción sea del tipo que sea retrocederá la transacción y se propagará al método que hace la llamada a la transacción.

Las excepciones que no sean del tipo **AlquilerCochesException** dejará el mensaje del error empleando el ***Logger*** ya configurado en el entorno de prácticas.

**Condiciones de entrega**

La práctica se debe entregar en una **única carpeta comprimida (ZIP)** en el buzón de entrega correspondiente **en el aula virtual** y contener una sola carpeta que se pueda descomprimir y que usando la función de Eclipse “*Open Projects from File System*” importe y se pueda ejecutar. Comprobad que esto es así ya que no se corregirá ni puntuará una práctica que no se importe y ejecute correctamente.

También hay que **incluir en la carpeta** de entrega un **PDF con los nombres completos de los participantes** en la práctica.

**No se admitirán entregas por correo electrónico**, así que no apuréis la fecha de entrega para evitar problemas técnicos a la hora de entregar.

**Pruebas automáticas**

Existen pruebas automáticas ya implementadas en la carpeta src/lsi/ubu/tests el método Tests.java. Es recomendable que las reviséis para poder comprobar que todo funciona correctamente cuando hayáis implementado el método que se os solicita.