Bases de Datos II

Trabajo Práctico Integrador: Etapa 1

Introducción
Esta primer etapa del Trabajo Práctico Integrador estará basada en la implementación de una aplicación en Java orientado al estudio de la persistencia, tanto en términos conceptuales, como en el uso de tecnología en particular. La aplicación consistirá en una arquitectura multicapa clásica, debiéndose implementar una capa de servicios y otra de acceso a datos subyacente. Por cuestiones de simplicidad se omitirá en esta implementación una capa superior a la capa de servicios como podría ser una API REST o de presentación vía algún framework MVC clásico.

La cátedra entregará un proyecto modelo implementado con Maven y el framework Spring y una configuración de Hibernate inicial, así como una serie de test cases que se deben satisfacer. El alumno deberá implementar todo lo necesario para que estos test cases pasen. En términos concretos, al correr los comandos

```
./createDatabase.sh
mvn clean install
```

desde la línea de comandos debe obtenerse un build exitoso en donde pasen todos los tests. El script `createDatabase.sh` debe contener los comandos MySQL para crear la base de datos correspondiente al grupo (más detalles al final del TP). De ser necesario, en este link los usuarios del SO Windows pueden encontrar documentación acerca de cómo instalar soporte para bash en ese Sistema Operativo.

Modelo de dominio de la aplicación a implementar
La aplicación a implementar (que se llamará AirBDB) consistirá en una versión simplificada de la aplicación de reservas online Airbnb. El términos generales, la aplicación deberá permitir modelar usuarios que poseen propiedades, y también pueden hacer reservas sobre las mismas. Las reservas serán realizadas por un usuario concreto, sobre una propiedad y tendrán una fecha de inicio, una fecha de fin, y un estado (**A confirmar**, **Confirmada**, **Cancelada** y **Terminada**). Las propiedades pertenecerán a una ciudad en particular y pueden ser de dos tipos: **departamento**, en cuyo caso se conoce el número de habitaciones, y **habitación privada**, en cuyo caso se conoce el número de camas. Ambos tipos de propiedad se conoce un nombre, descripción, precio por noche y la capacidad. Una vez terminada una reserva, un usuario podrá hacer una evaluación de la misma, calificando su experiencia con una descripción y otorgando un puntaje de 1 a 5.

Nota de implementación: los estados de las reservas pueden ser implementados con un tipo enumerativo (enum).

Requisitos detallados que deben satisfacerse en esta etapa

1. Definir el método `HibernateConfiguration.getGroupNumber()` para que devuelva un Integer con el número de grupo.
2. Escribir todo el código necesario para que la aplicación compile y todos los tests de la clase `AirBdbServiceTestCase` pasen. Esto implicará codificar una implementación concreta de la interfaz `AirBdbService` cuyos métodos están descriptos vía Javadoc en la misma y también a través de los tests en `AirBdbServiceTestCase`.
3. Deben proveer un script bash que inicialice la base de datos en sí y cualquier otra configuración que sea necesaria (por ejemplo, usuarios y permisos) llamado `createDatabase.sh`
4. Correr el script anterior y luego `mvn clean install` debe resultar en un build exitoso en donde todos los tests pasen
5. El código tiene que estar subido en un repositorio privado de BitBucket y la forma de entrega en principio va a ser solamente a través de este medio. El repositorio privado deberá compartirse con el ayudante designado para el grupo.

Requisitos técnicos para esta etapa

1. Tener instalado JDK 8 (`javac -version` desde la línea de comandos debe funcionar)
2. Tener instalado Maven 3.5.0 (`mvn --version` desde la línea de comando debe funcionar)
3. Tener instalado MySQL 5.7 (Importante: no utilizar MariaDB)
4. Es altamente recomendable Utilizar un IDE para Java, preferentemente IntelliJ o Eclipse.