UNIVERSIDAD DE LA EMPRESA

Bases de Datos III

Trabajo Obligatorio
Curso 2019
Licenciatura e Ingeniería en Informática
Total: 100 puntos - Mínimo de aprobación: 60 puntos

Indicaciones Generales

- Los grupos deberán tener cuatro (4) integrantes. En caso de que la cantidad total de estudiantes de la clase no sea múltiplo de 4, se admitirá excepcionalmente algún grupo de tres (3) integrantes.
- La fecha de entrega es el ***Miércoles 13 de noviembre de 2019***. Se debe entregar el proyecto en eclipse (vía e-mail a la casilla del docente) conteniendo:
  1. Un programa `main` de prueba que se encargue de crear la base de datos para el reality show ***RuPaul's Drag Race***, según las indicaciones del ejercicio 2 del práctico 4.
  2. La estructura de packages definida en el ejercicio 2 del práctico 5, conteniendo la implementación de todas las clases e interfaces implementadas entre los ejercicios 2 y 4 del práctico 5.
  3. Los packages adicionales que sean necesarios para contener la implementación de las nuevas clases e interfaces adicionales para un ***segundo mecanismo de persistencia*** que se solicita en este trabajo obligatorio (ver más adelante, puntos 5 al 9 de la letra).
- Al día siguiente de la entrega, es decir el ***Jueves 14 de noviembre de 2019*** cada grupo realizará en clase una ***demostración*** de la ejecución de la aplicación construida. Durante la demostración de la ejecución, cada grupo deberá correr su aplicación en un ambiente ***distribuido***, utilizando varios equipos para la ejecución (un equipo que albergue la base de datos, otro que albergue las capas lógica y de persistencia, y otros dos equipos que alberguen la capa gráfica) y mostrando la ejecución de todas las funcionalidades.

> **No** se requiere realizar presentación alguna, solamente la demostración de la ejecución.

Características del Trabajo

El trabajo obligatorio consiste en la implementación **completa** de la aplicación resultante de la realización de los siguientes ejercicios de los prácticos del curso (más otros puntos adicionales que se solicitan más adelante):

1. Ejercicio 2 del Práctico 4 (creación de la base de datos para ***RuPaul's Drag Race***).
2. Ejercicio 4 del Práctico 4 (creación del Pool de Conexiones para manejo de concurrencia).
3. Ejercicio 2 del Práctico 5 (implementación de la aplicación en 3 capas persistiendo en la base de datos creada en el punto 1).
4. Ejercicio 4 del Práctico 5 (incorporación de manejo de concurrencia a la aplicación en 3 capas).

Se deberán ir resolviendo los ejercicios indicados en forma ***incremental*** y siguiendo el orden en que fueron propuestos. El resultado de todo lo anterior deberá ser una ***aplicación en 3 capas*** que tendrá las siguientes características:

- Debe poseer la ***arquitectura física 3-Tier*** descrita en el Ejercicio 2 del Práctico 4.
- Debe realizar manejo de concurrencia utilizando ***Pool de Conexiones***.
- Debe utilizar los patrones ***Facade***, ***Value Object***, ***MVC***, ***DAO*** vistos en el curso.

Una vez completada la implementación anterior, se deberá incorporar a la aplicación la utilización del patrón ***Abstract Factory***, para permitir también persistir toda la información en un ***segundo*** mecanismo de persistencia, que consistirá en una estructura de ***archivos serializados*** en disco.

La estructura de archivos serializados será la siguiente: Habrá un archivo separado por cada temporada, conteniendo los siguientes datos de la temporada (número, año, cantidad de capítulos). El nombre del archivo será generado a partir del número de la temporada. Por cada temporada, habrá a su vez otro archivo conteniendo los datos de **todas** las Drag Queens participantes en dicha temporada (número de participante, nombre y cantidad de victorias). La nomenclatura para los nombres de los archivos se ilustra en los siguientes ejemplos:

`temporada-1.txt` – archivo que contiene los datos de la temporada número 1.

`temporada-2019.txt` – archivo que contiene los datos de la temporada número 2019.

`dragqueens-1.txt` – archivo que contiene las drag queens de la temporada número 1.

`dragqueens-2019.txt` – archivo que contiene las drag queens de la temporada número 2019.

La aplicación deberá ser capaz de persistir ***indistintamente*** en uno u otro mecanismo de persistencia (la base de datos del certamen o la estructura de archivos serializados). En el código fuente de la Fachada, **no** deberá haber indicación explícita del mecanismo de persistencia utilizado. El cambio de un mecanismo a otro deberá manejarse con ***archivos de configuración***, del modo visto en el teórico. Se deberá agregar a la aplicación desarrollada hasta el momento la implementación que se detalla en los siguientes puntos:

5. Implementación de las clases `ConexionArchivo` y `PoolConexionesArchivo`. Estas clases deben implementar las interfaces `IConexion` e `IPoolConexiones` vistas en el curso y corresponden a un ***segundo*** pool de conexiones, que tendrá una implementación ***diferente*** a la desarrollada para la base de datos. Servirá para controlar el manejo de concurrencia sobre la estructura de archivos correspondiente al segundo mecanismo de persistencia.

- Los métodos `obtenerConexion` y `liberarConexion` tendrán el comportamiento usual de un ***monitor de lectura y escritura*** similar al utilizado en Taller II. Si la conexión se pide para un requerimiento que **modifica** los archivos, se considerará como un ***escritor***. En cambio, si es un requerimiento que **no** modifica archivos, se considerará como un ***lector***.
- La clase `PoolConexionesArchivo` deberá definir aquellos atributos que sean necesarios para el manejo usual de un monitor de lectura y escritura.

6. Creación de las interfaces `IDAOTemporadas` e `IDAODragQueens`. Se usarán en el contexto del patrón ***Abstract Factory***, para migrar de mecanismo de persistencia. Los cabezales de sus métodos serán los mismos que ya poseen las clases DAO hasta el momento.

7. Adaptación de las clases `DAOTemporadas` y `DAODragQueens`, de modo que ahora implementen las interfaces anteriores. Sus métodos ya implementados no sufrirán ningún cambio. Además, en la clase `Temporada`, cambiar el atributo de tipo `DAODragQueens` por uno de tipo `IDAODragQueens`.

8. Implementación de las clases `DAOTemporadasArchivo` y `DAODragQueensArchivo`. Estas clases también implementarán las interfaces `IDAOTemporadas` e `IDAODragQueens` y definirán el acceso a la estructura de archivos serializados usada como segundo mecanismo de persistencia. Tendrán los mismos cabezales que las clases anteriores, pero en vez de persistir sobre la BD, lo harán sobre la estructura de archivos. Se permite definir clases y/o métodos auxiliares para leer y escribir en los archivos, siempre y cuando se utilicen **únicamente** en forma interna a `DAOTemporadasArchivo` y `DAODragQueensArchivo`.

9. Creación de la interface `FabricaAbstracta` e implementación de las clases `FabricaMySQL` y `FabricaArchivo` del patrón ***Abstract Factory*** visto en el teórico. La `FabricaAbstracta` será utilizada del modo visto en clase y las clases `FabricaMySQL` y `FabricaArchivo` instanciarán las clases DAO que acceden al mecanismo concreto de persistencia correspondiente a cada una, también del modo visto en el curso. La fábrica se utilizará desde la `Fachada` para instanciar el DAO de Temporadas que corresponda y desde el método constructor de la clase `Temporada` para instanciar el DAO de Drag Queens que corresponda.