UNIVERSIDAD DE LA EMPRESA

Bases de Datos III
Práctico 5
Licenciatura en Informática
Ingeniería en Informática

Ejercicio 1

Para cada una de las siguientes afirmaciones sobre aplicaciones en ***arquitecturas de* 3 *capas***, indique si es correcta o incorrecta de acuerdo a los conceptos vistos en el teórico. Fundamente todas sus respuestas.

a) Si una aplicación utiliza el patrón **DAO**, pero ***no*** utiliza el patrón **Abstract Factory**, entonces sigue siendo una aplicación en 3 capas.

b) Si una aplicación utiliza el patrón **Abstract Factory** pero ***no*** utiliza el patrón **DAO**, entonces sigue siendo una aplicación en 3 capas.

c) Las aplicaciones en **3 *capas*** que persisten en DBMS suelen optimizar mejor sus recursos que las aplicaciones en **2 *capas*** porque la mayoría del comportamiento se realiza en la aplicación.

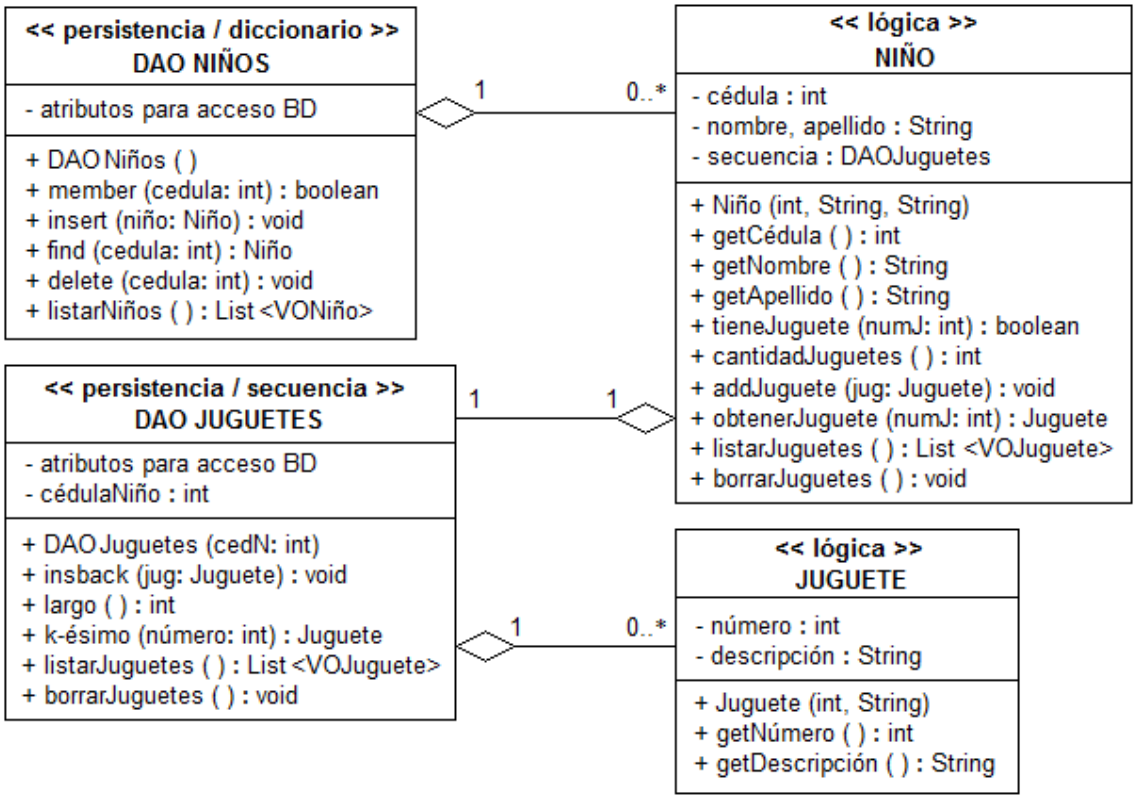
d) Cuando se aplica el patrón **DAO** en una aplicación en **3 *capas*** que persiste contra un DBMS, los métodos de una misma clase DAO acceden siempre a una misma tabla de la BD.

e) Las aplicaciones en **3 *capas*** logran una mayor independencia del mecanismo de persistencia utilizado que las aplicaciones en **2 *capas***.

f) Las aplicaciones en **2 *capas*** siempre son más eficientes que las aplicaciones en **3 *capas***.

Ejercicio 2

En este ejercicio vamos a migrar a **3 *capas*** la aplicación de la sala de juegos infantiles desarrollada en el práctico 4. Se hará de modo tal que la ***lógica de los requerimientos*** será resuelta del lado de la ***aplicación***. Se propone a continuación un diseño orientado a objetos para la aplicación.



Se van a mantener ***incambiados*** los siguientes elementos respecto a la versión en 2 capas:

- Se seguirá utilizando la misma ***base de datos*** que en el práctico anterior.
- Se seguirá utilizando la misma ***capa gráfica*** (ventanas, controladores y value objects) que en el práctico anterior).
- La ***Fachada*** seguirá teniendo los mismos métodos que en el práctico anterior, lo que cambiará en esta versión es la implementación interna de los métodos.

Al igual que en el práctico anterior, supondremos inicialmente que existirá un ***único usuario*** y ***no*** nos preocuparemos inicialmente por realizar manejo de ***concurrencia***.

a) Cree un proyecto para la aplicación con la estructura de packages dada en la figura.

b) Incorpore al package `excepciones` las clases correspondientes a las excepciones que implementó en el práctico anterior.

c) Incorpore al package `valueObjects` las clases `VONiño` y `VOJuguete` que implementó en el práctico anterior.

d) Implemente en el package `consultas` la clase `Consultas` que define los textos de todas las sentencias SQL que la aplicación ejecutará sobre la base de datos.

e) Implemente en el package `daos` las clases `DAONiños` y `DAOJuguetes` de acuerdo al diseño en UML propuesto. Estas serán las **únicas** clases desde donde se accederá a la BD. Respete **estrictamente** todos los encabezados de los métodos propuestos en UML.

f) Implemente en el package `logica` la nueva clase Fachada de modo que siga teniendo los **mismos** métodos que en el práctico anterior. Lo que cambiará será la implementación de dichos métodos, los cuales ahora harán uso de las cuatro clases definidas en el diagrama de UML y será accedida desde los clientes mediante **RMI**. Dado que ahora la aplicación es en ***3 capas***, la fachada **no** debe acceder directamente a la base de datos. La fachada tendrá el DAO de Niños como atributo. El acceso al DAO de Juguetes de cada niño **no** se realizará desde la fachada, sino que se hará en forma **interna** a la clase `Niño`.

g) Incorpore a los packages `ventanas` y `controladores` las mismas clases correspondientes a la capa gráfica que implementó en el práctico anterior. No debería necesitar modificar **ninguna** línea de código fuente en estas clases.

Ejercicio 3

Conteste las siguientes preguntas relativas a la aplicación en **3 *capas*** desarrollada en el ejercicio 2:

a) De acuerdo al diagrama de UML, ¿cuántas instancias de la clase DAONiños existen? ¿y cuántas instancias de la clase DAOJuguetes existen?

b) Explique cómo se mapeó el diseño en UML con las tablas de la BD en el ejercicio anterior.

c) Para cada uno de los 6 requerimientos que resuelve la ***Fachada*** (`nuevoNiño`, `nuevoJuguete`, `listarNiños`, `listarJuguetes`, `darDescripcion`, `borrarNiñoJuguetes`) conteste:

- ¿Cuántas sentencias SQL eran necesarias para resolver el requerimiento en **2 *capas***?
- ¿Cuántas sentencias SQL son ahora necesarias para resolver el requerimiento en **3 *capas***?

d) En general, la eficiencia de los requerimientos que resuelve la aplicación, ¿se vio beneficiada o perjudicada por la migración de **2 *capas*** a **3 *capas***? Explique brevemente.

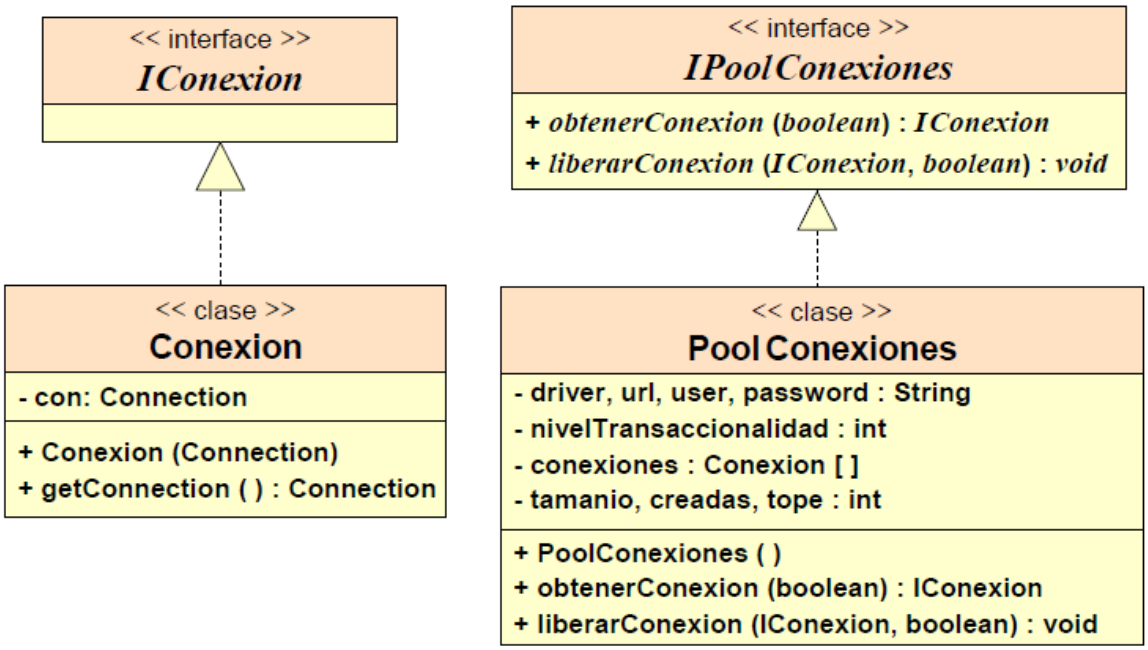
e) En general, las propiedades de mantenibilidad y reusabilidad de la aplicación, ¿se vieron beneficiadas o perjudicadas por la migración de **2 *capas*** a **3 *capas***? Explique brevemente.

f) Los beneficios y desventajas observados, se verán afectados de la misma manera en ***cualquier*** otra aplicación? ¿De qué factores depende? Explique brevemente.

Ejercicio 4

En este ejercicio incorporaremos manejo de ***concurrencia*** a la aplicación del ejercicio 2.

a) Agréguele al Project del ejercicio 2 un nuevo package para el ***pool de conexiones***. Incorpore a dicho package las interfaces `IPoolConexiones` e `IConexion` y las clases `PoolConexiones` y `Conexion` ya desarrolladas en el práctico 4. No debería necesitar modificarles **ninguna** línea de código.



b) Defina en la clase Fachada el atributo `private IPoolConexiones ipool;` correspondiente al pool de conexiones ***abstracto*** desarrollado en el práctico 4. Dentro del constructor de la Fachada, instancie el pool de conexiones de la siguiente manera:

```
String poolConcreto = /* cargo el nombre de la clase PoolConexiones
                         desde un archivo de configuración */
ipool = (IPoolConexiones) Class.forName(poolConcreto).newInstance();
```

Lo que hace este código es instanciar el pool de conexiones de modo tal que el nombre del pool concreto utilizado (en este caso `PoolConexiones`) **no** quede visible en el código fuente.

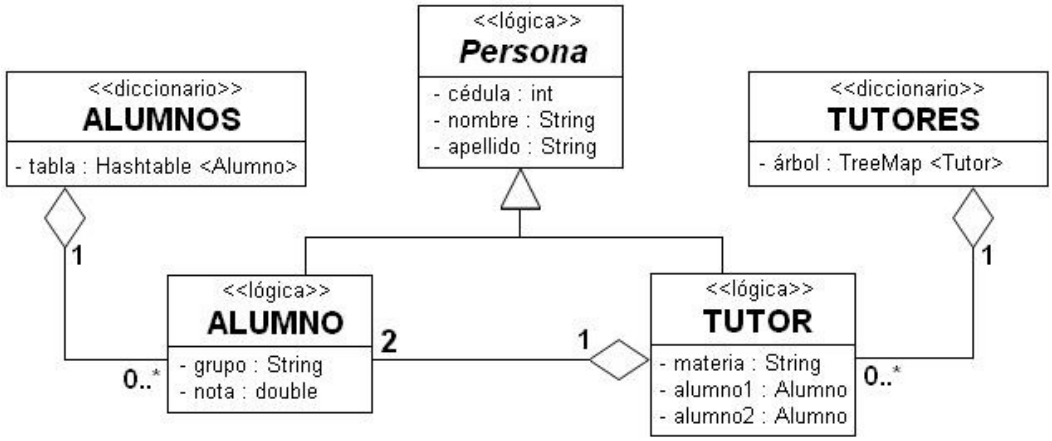
c) Incorpore a la clase Fachada el uso del ***Pool de Conexiones*** de la siguiente manera:

- En cada método de la Fachada, solicite una conexión abstracta al `IPoolConexiones`.
- Pase dicha conexión a los métodos de las clases DAO que resuelvan cada requerimiento. Deberá ***agregar*** un nuevo parámetro de tipo `IConexión` a cada método de los DAO y usarlo internamente para resolver la misma acción que resolvía anteriormente. Dentro del DAO deberá ***castear*** la conexión abstracta hacia una `Conexión` concreta.
- Al finalizar la ejecución del requerimiento, devuelva la conexión al `IPoolConexiones` liberándola en forma exitosa (`true`) o fallida (`false`), según corresponda.

d) ¿Por qué es beneficioso que el nombre del pool concreto utilizado **no** quede visible en la Fachada? Explique brevemente.

Ejercicio 5

Considere el siguiente diseño (parcial) en notación UML para una determinada aplicación:



a) Explique brevemente el significado de cada una de las ***multiplicidades*** del diagrama

b) De acuerdo con este diseño, ¿cuántas instancias de cada diccionario existen?

c) Proponga un esquema de tablas de una base de datos relacional que ***mapee*** este diseño orientado a objetos de acuerdo con las técnicas de mapeo vistas en el teórico. Justifique.

d) Modifique el diagrama propuesto de modo que ahora los datos se persistan en la BD anterior en vez de permanecer almacenados en memoria (aplicación del patrón ***D.A.O*** en forma pura).

e) Modifíquelo nuevamente, de modo que los datos se sigan persistiendo en la BD relacional, pero ahora se haga combinando ***D.A.O*** con ***Abstract Factory***. ¿Cuántas familias y cuántos tipos de productos se tienen en este momento?

f) Ahora suponga que también se desea tener la posibilidad de persistir tanto en la BD anterior como en una estructura de archivos XML. Agregue al diagrama las clases e interfaces necesarias para reflejarlo. ¿Cuántas familias y cuántos tipos de productos se tienen ahora?