

Ejercicio 1

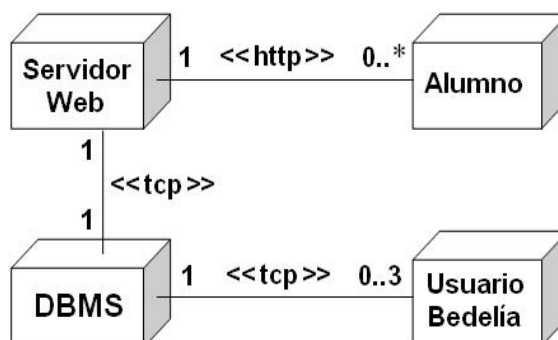
Para cada una de las siguientes afirmaciones sobre aplicaciones en **arquitecturas de 1 capa**, indique si es correcta o incorrecta de acuerdo a los conceptos vistos en el teórico. Fundamente todas sus respuestas.

- Si se decide cambiar el mecanismo de persistencia utilizado (por ejemplo, de archivos de texto a bases de datos), no es necesario recompilar código fuente en la interfaz de usuario.
- La definición de una clase que encapsula el acceso a la base de datos asegura que la arquitectura de la aplicación deja de estar en 1 capa y pasa a estar en 2 capas.
- Si la aplicación es *standalone* y cada requerimiento se resuelve en una sola consulta, entonces es posible migrarla a *cliente-servidor* sin necesitar modificar código fuente.
- El manejo de transacciones sobre la base de datos se reserva exclusivamente para aplicaciones en *arquitecturas de 2 y 3 capas*.
- Las aplicaciones que priorizan el comportamiento a nivel de la aplicación permiten un mejor aprovechamiento de los recursos del DBMS en términos de optimización de las consultas.
- Una reducción en la cantidad de consultas que ejecuta la aplicación implica un aumento en la cantidad de causas que pueden provocar excepciones de SQL.

Ejercicio 2

Se va a desarrollar una **aplicación en 1 capa** que permitirá a la bedelía de una pequeña institución educativa publicar los resultados de exámenes y a los alumnos la posibilidad de consultarlos por Internet.

La aplicación tendrá la **arquitectura física** dada en la figura. Los usuarios de la bedelía interactuarán con la aplicación mediante una **interfaz gráfica de ventanas** mientras que los alumnos lo harán mediante **páginas web**.



El esquema de la base de datos **MySQL** para la aplicación será el siguiente:

```
Examenes (codigo VARCHAR(45), materia VARCHAR(45), periodo VARCHAR(45))
Resultados (cedula INT, codigo VARCHAR(45), calificacion INT)
```

- La columna `codigo` en `Resultados` referencia al campo `codigo` en `Examenes`.

- Describa con palabras cuál es la arquitectura física de esta aplicación.
- Escriba un programa `Main` en **Java** que cree la base de datos según el esquema descrito y cargue en ella los datos de los siguientes exámenes por defecto:

Código	Materia	Período
MD2012Dic	Matemática discreta	Diciembre 2012
P12012Dic	Programación	Diciembre 2012
BD2012Dic	Bases de datos	Diciembre 2012
MD2013Feb	Matemática discreta	Febrero 2013

Ejercicio 3

En este ejercicio desarrollaremos el componente de la aplicación del ejercicio 2 que correrá en los equipos de los usuarios que trabajan en la bedelía. Habrá un *package* `accesoBD` que contendrá las clases vinculadas a la interacción con la base de datos y otro *package* `presentacion` que contendrá la interfaz de usuario.

- a) Implemente en **Java** la clase `Consultas` que define los textos de las consultas, la cual debe contar con los siguientes métodos:

```
public String listarExamenes ()
/* crea el texto de la consulta que obtiene un listado de todos */
/* los exámenes registrados en la BD */

public String insertarResultado ()
/* crea el texto de la consulta que inserta un resultado de examen */
```

- b) Implemente en **Java** la clase `AccesoBD` que encapsula el acceso a la base de datos, la cual debe contar con los siguientes métodos:

```
public List <Examen> listarExamenes (Connection con)
/* devuelve un listado de todos los exámenes registrados en la BD */
/* de cada examen se tiene código, asignatura y período */

public void ingresarResultado (Connection con, Resultado resu)
/* ingresa el resultado de un examen en la BD, los datos del */
/* resultado vienen almacenados en el objeto resu */
```

- c) Implemente en **Java** la clase `VentanaResultado` que le permita al usuario de bedelía ingresar el resultado de un nuevo examen, tomando en cuenta las siguientes consideraciones:

- La aplicación le muestra al usuario todos los exámenes y le pide que seleccione uno.
- El usuario selecciona el examen e ingresa en campos de texto la cédula y la calificación.
- El chequeo de formato de cédula y calificación se debe hacer a nivel de la interfaz gráfica.
- Se debe emitir un mensaje indicando el éxito o el fracaso del ingreso del resultado.
- Utilice un archivo de configuración para cargar *driver*, *url*, *usuario* y *password* de la BD.
- Se debe realizar **manejo de transacciones**.

- d) Observe la implementación que realizó en las partes anteriores e indique qué enfoque utilizó para centralizar el comportamiento. ¿Lo hizo a nivel **de la aplicación**? ¿Lo hizo **a nivel de la base de datos**? ¿Lo hizo de forma repartida entre ambos? Explique su respuesta.

Observación: Defina cualquier clase o método auxiliar que le pueda resultar de utilidad e inclúyala(s) en el (los) *package*(s) que corresponda(n).

Ejercicio 4

En este ejercicio desarrollaremos el componente web de la aplicación del ejercicio 2 que permitirá a los alumnos consultar los resultados de sus exámenes por Internet.

a) Agréguele a la clase Consultas del ejercicio 3 el siguiente método:

```
public String listarResultados ()
/* crea el texto de la consulta que obtiene todos los resultados */
/* de los exámenes rendidos por un estudiante, dada su cédula */
```

b) Agréguele a la clase AccesoBD del ejercicio 3 el siguiente método:

```
public List<Resultado> listarResultados (Connection con, int cedula)
/* devuelve un listado de todos los exámenes registrados en la BD */
/* correspondientes al número de cédula ingresado. En caso de no */
/* haber ninguno, devuelve el listado vacío */
```

c) Agréguele al package presentacion del ejercicio 3 la clase ServletResultados que le permita al estudiante obtener un listado de todos sus resultados de examen, tomando en cuenta las siguientes consideraciones:

- Debe realizar el chequeo de formato de la cédula del estudiante, la cual será recibida de la página JSP de ingreso.
- Forwardee a una página JSP de error en caso de que el formato sea inválido.
- En otro caso, obtenga de la base de datos el listado de resultados del estudiante y forwardee a la página JSP que los despliegue.
- Cargue del archivo **web.xml** el *driver*, *url*, *usuario* y *password* de la BD.
- Se debe realizar **manejo de transacciones**.

d) Cree dos páginas JSP, la primera de ellas le permitirá al estudiante ingresar su número de cédula y se lo pasará al servlet de la parte anterior. La segunda de ellas le mostrará el estudiante todos sus resultados de examen.

Resultados de exámenes

Ingrese su número de cédula sin puntos ni guión:

4175420

Resultados del estudiante:

Cédula	Código	Calificación
4175420	MD2012Dic	0
4175420	BD2012Dic	9
4175420	MD2013Feb	7

Observación: Para que el componente web pueda conectarse al DBMS, incorpore el archivo `mysql-connector-java-5.1.21-bin.jar` correspondiente al driver en la ubicación `WEB-INF\lib` del proyecto en eclipse en forma previa a generar el archivo **.war**