PROGRAMACIÓN WEB

Curso 2021/2022

PRÁCTICA 2: INTRODUCCIÓN A JDBC, DAO, DTO Y JSP

En esta práctica nos introducimos en el uso de buenas prácticas y patrones arquitecturales en código, trabajando en Java y J2EE contra bases de datos y haciendo uso de objetos de transferencia (DTOs) y objetos de accesos a datos (DAOs).

Se trabajará con el mismo equipo de tres personas que se constituyó para la práctica 1 (invariable e inseparable durante el presente curso académico). Entre las tecnologías que se practicarán en estos ejercicios contamos con acceso a MySQL (*driver* JDBC), J2EE (*javabeans*, JSP, etc.) y Apache Tomcat (configuración de web.xml) para su despliegue.

Tenga en cuenta los siguientes **aspectos normativos** referentes a esta práctica:

- Junto con el código fuente, se entregará un informe en formato PDF (máximo 4 páginas) que deberá explicar y fundamentar las decisiones de diseño e implementación que ha adoptado el equipo, así como la relación de fuentes consultadas y un análisis de las dificultades encontradas.
    - Indique en el informe el nombre de los integrantes del equipo, así como la pila de tecnología utilizada para el desarrollo.
    - Recuerde que el WAR entregado debe funcionar en el servidor Tomcat *standalone* (7.0) instalado en la UCO.
- El código debe estar completamente documentado, tanto las clases Java como los JSP y código HTML, si procede.
- La propuesta de ejercicios tiene margen de decisión por parte del equipo. Dado que se espera que estas decisiones sean distintas por cada equipo de prácticas, deberán estar justificadas pertinentemente en el informe.
- Sólo hará entrega de la práctica un miembro del equipo a través de Moodle. Deberá subir a Moodle esta práctica la misma persona que entregase la primera práctica.
- Se hará entrega a través de la tarea Moodle dispuesta para ello de un archivo ZIP que encapsule el proyecto Eclipse (preferiblemente) o ficheros fuente de la práctica.
    - Se debe tener especial cuidado en el nombrado de paquetes y clases.
    - El archivo PDF correspondiente al informe debe estar incluido en el raíz del directorio del proyecto.
    - Para cada ejercicio, se deberá entregar el ejecutable pertinente (JAR, WAR, etc.), incluido en el directorio raíz del ZIP de proyecto, así como un README.txt con las instrucciones pertinentes para su ejecución.

- El archivo ZIP que se entrega debe ser nombrado de la siguiente forma: GM<gm>_<login>.zip, donde <gm> es el número de grupo mediano y <login> es el nick de la UCO de la persona del equipo que hace la entrega de esta y el resto de prácticas.

El incumplimiento de las normas de entrega (contenido, nombrado, etc.) implicará la consideración de práctica no entregada. La fecha de entrega se dispondrá en la tarea Moodle correspondiente. Igualmente, el retraso en la entrega supondrá la no calificación en la práctica.

IMPORTANTE: No se admitirá entrega alguna fuera de plazo o a través de un medio distinto a la tarea de Moodle (p.ej. envío por email).

EJERCICIO 1

Partiendo de las clases gestoras de la práctica 1 (usuarios, críticas y espectáculos, al menos), se desea eliminar el uso de ficheros para el almacenamiento de usuarios, críticas y espectáculos. Para ello se ha habilitado a cada estudiante acceso a una base de datos en el servidor http://oraclepr.uco.es/

Registre una base de datos en dicho servidor.

Se llevará a cabo la reimplementación del ejercicio 2 de la práctica 1, en la que tanto usuarios, críticas y espectáculos estarán adecuadamente modelados como tablas y relaciones en la base de datos MySQL del estudiante.

En primer lugar, diseñe e implemente el esquema de base de datos relativo al ejercicio. Considere las tablas que sean necesarias para alojar datos referentes a la información descrita en el párrafo anterior. No olvide establecer una clave de acceso a la base de datos que pueda hacer pública.

Después, implemente el acceso a esta base de datos a través del *driver* JDBC. Para ello haga uso de buenas prácticas y codifique una capa de datos (*data*) diferenciada de la capa de negocio (*business*) para el ejercicio. Acceda a las tablas o vistas de la base de datos mediante objetos de acceso a datos (DAO) y utilice objetos de transferencia (DTO) para encapsular la semántica de los objetos del dominio de aplicación. La aplicación resultante debe ofrecer las mismas funcionalidades solicitadas para el ejercicio 2 de la práctica 1.

Utilice dos ficheros de propiedades:

- *config.properties*: almacena los parámetros de configuración y cadena de acceso a la base de datos (nombre de servidor, puerto, usuario, contraseña).
- *sql.properties*: almacena las cadenas específicas de MySQL para consulta y actualización de datos en SQL.

Se asumirá que, al menos, el fichero *config.properties* estará alojado en el mismo directorio donde se ejecute el programa.

De este ejercicio deberá entregarse el programa ejecutable resultante en formato JAR.

Para el desarrollo de este ejercicio se podrán llevar a cabo las modificaciones (orientadas a la mejora) que el equipo de desarrollo considere conveniente sobre el entregable de la práctica 1. Reporte convenientemente estos cambios y mejoras en el informe entregado.

EJERCICIO 2

Tomando como ejemplo simplificado el patrón MVC aplicado para la funcionalidad de acceso (*login*) de usuario, se desea implementar una aplicación web que realice el registro y acceso de usuarios para una web. Esta aplicación se desarrollará estrictamente conforme al patrón MVC para todas las funcionalidades del dominio de negocio.

Todos los datos de la aplicación se alojarán en base de datos, a la que se accederá mediante DAOs y serán manipulados con objetos de transferencia propios del dominio. La base de datos utilizada será la misma que para el ejercicio 1 (salvo por las tablas/atributos que fueran necesarias añadir).

Al acceder inicialmente, la aplicación mostrará una ventana con dos opciones: registrarse o acceder. En el caso del registro, se debe permitir la creación de un nuevo usuario desde cero. Se deberán realizar las comprobaciones necesarias para verificar que el registro de usuario se realiza de forma consistente (p.ej. que no existe otro usuario ya registrado con el mismo correo electrónico). Debe contemplarse la existencia de dos tipos de usuarios registrados: administrador y espectador. La información del usuario quedará almacenada en la base de datos.

En el caso de la funcionalidad de acceso, el usuario podrá logarse con o sin éxito, accediendo a la tabla de usuario de la base de datos para comprobar si se encuentra registrado. Si el acceso es correcto, se creará el *CustomerBean* correspondiente y se redireccionará a la página inicial. Para el caso del usuario “espectador”, la página principal mostrará un mensaje de bienvenida al usuario, la fecha actual, y la fecha en la que el usuario se registró. Para el caso del usuario “administrador”, se mostrará el listado de los nombres de usuario (nick) de cada tipo y la fecha de su última conexión. Para ello, guarde en base de datos información de los accesos de los usuarios registrados (log). Además, la página principal (para cualquiera de los dos tipos de usuarios registrados) mostrará dos opciones: “desconectar” y “modificar datos”. Si pulsara sobre “desconectar”, se realizaría la desconexión del usuario que había iniciado la sesión. En caso de que el acceso fuera incorrecto, se mostrará un mensaje de error, dando opción a volverlo a intentar.

En el caso de la funcionalidad de “modificar datos”, la aplicación mostrará los campos del registro de usuario, posibilitando la modificación de aquellos que se consideren permitidos. Por ejemplo, no se podrá cambiar el correo electrónico por tratarse del identificador ni tampoco el tipo de usuario.

Para la implementación de este ejercicio se hará uso de un fichero de propiedades, denominado *sql.properties*, en el que se guardarán las consultas necesarias para el acceso

a la base de datos desde los DAO. Los parámetros de configuración (cadena de conexión a base de datos) se guardarán en el archivo *web.xml* asociado a la aplicación. Tanto los controladores como las vistas se implementarán como JSP. Se omitirá cualquier detalle de implementación referido a la interfaz de usuario, no siendo necesario formateo de estilos.

Igualmente, considérese que toda página debe redireccionar a la página de error correcta. Vigile que las recomendaciones de diseño (patrones, controles de acceso, etc.), de estructura de páginas y clases, y de despliegue se cumplen satisfactoriamente.

De este ejercicio deberá entregarse la aplicación web resultante en formato WAR. Igualmente, entregue en el ZIP una imagen con la estructura de la base de datos, que incluya tablas, sus atributos y relaciones entre las tablas. La base de datos debe tener contenido de ejemplo realista, contando con varias tuplas y omitiendo nombres y valores al estilo “Usuario1”, “UsuarioPrueba”, “Etiqueta1” para los valores de los atributos.