Práctica 2: Diseño, desarrollo e implantación de un sistema de información Web: diseño e implementación de la capa de persistencia de datos

Sistemas de Información

Grado de Informática
Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura

16 de octubre de 2019

1. Objetivos

Las aplicaciones Web son aplicaciones que se ejecutan en al menos un servidor Web y a las que los usuarios acceden desde un cliente de propósito general (por ejemplo: un navegador Web en un PC, teléfono móvil, etc.). Se caracterizan por poseer un contenido estático (html, pdf, etc.) y contenido dinámico, que se genera cuando los usuarios realizan diferentes peticiones. El contenido dinámico lo genera el contenedor/servidor de aplicaciones Web, consultando datos almacenados en diferentes fuentes de datos, las cuales generalmente son bases de datos relacionales

En esta práctica se usará un Sistema Gestor de Bases de Datos (SGBD) para trabajar conceptos básicos programación Web con persistencia de datos. Se recomienda el uso de un SGBD relacional compatible con JDBC (Java Data Base Connection); sin embargo, los diferentes grupos de prácticas pueden seleccionar otro tipo de gestor (como por ejemplo un sistema gestor de bases de datos NoSQL como MongoDB). Además, se completará el desarrollo del diseño de la vista que se implementó en la práctica 1. En mayor detalle, los objetivos de esta práctica son:

- Instalar/Configurar el entorno de trabajo, en concreto el SGBD seleccionado en caso de que este no se encuentre disponible en Hendrix (MySQL y Oracle están disponibles en Hendrix y disponéis de cuenta para acceder) o se prefiera trabajar con otro servidor (por ejemplo: MongoDB). En Hendrix disponéis de acceso a un servidor MySQL y a otro Oracle. También podéis optar por Postgresql.

- Diseñar el modelo Entidad-Relación de la base de datos requerida para almacenar los datos de la aplicación Web que se está construyendo.

- Transformar el modelo Entidad-Relación en un modelo relacional (en caso de que se haya seleccionado un SGBD relacional) o en el tipo de modelo (en red, jerárquico, en grafo, etc.) que emplee el SGBD seleccionado para la realización de esta práctica.

- Desarrollar (implementar) el modelo de persistencia de datos en el SGBD seleccionado.

- Diseñar e implementar la capa de acceso a datos de la aplicación. En caso de que se haya optado por una SGBD relacional se recomienda el uso de JDBC (o API equivalente, por ejemplo, en desarrollos con .Net ODBC) y los patrones DAO (Data Access Object) y VO (Value Object).

- Elaborar un informe o memoria de trabajo en el que se señalen, además de los aspectos de diseño y técnicos, la metodología empleada en el desarrollo de la práctica y el coste en horas de trabajo de esta, enumerando las decisiones tomadas.

El entorno de desarrollo propuesto emplea JDK 1.8, JEE, Tomcat y MySQL; no obstante, los diferentes grupos de prácticas pueden seleccionar tecnologías y herramientas análogas.

2. Contenidos

La práctica consta de dos bloques, de los cuales el primero consiste en el diseño del modelo Entidad-Relación para dar soporte al almacenamiento de datos del sistema de información Web que se está desarrollando y su transformación (por ejemplo, en un modelo relacional) e implementación en un SGBD. El segundo bloque se centra en el desarrollo de la capa de datos de la aplicación web que interactúa con el SGBD para recuperar, modificar, borrar y actualizar datos. Además, se requiere la elaboración de una memoria de trabajo/informe en el que se enumeren las decisiones tomadas en cada uno de los pasos y se resuma el trabajo que se ha realizado (metodología, coste en horas, etc.).

2.1 Configuración del entorno de trabajo (SGBD) y creación de la base de datos del sistema de información Web

Sistema gestor de bases de datos relacionales MySQL en Hendrix

Los detalles de acceso a una base de datos MySQL instalada en hendrix-mysql.cps.unizar.es se indican a continuación:

- **Host en el que se encuentra instalado**: hendrix-mysql.cps.unizar.es
- **Puerto**: 3306
- **Usuario**: el mismo que empleáis/empleabais en el sistema gestor de bases de datos MySQL en otras asignaturas (por ejemplo, Bases de Datos).
- **Clave**: se encuentra en el fichero README.mysql de vuestra cuenta en Hendrix.

El acceso vía línea de comandos a MySQL desde Hendrix se realiza del siguiente modo:

mysql -h hendrix-mysql.cps.unizar.es -u username -p nombreDeLaBD

La base de datos nombreDeLaBD almacenará la información propia del sistema (por ejemplo, datos de usuarios) y es necesario realizar su diseño e implementación.

Sistema gestor de bases de datos relacionales Oracle en Hendrix

Los detalles de acceso a una base de datos Oracle instalada en hendrix-oracle.cps.unizar.es se indican a continuación:

- **Host en el que se encuentra instalada**: hendrix-oracle.cps.unizar.es
- **Usuario**: el mismo que empleáis/empleabais en el sistema gestor de bases de datos Oracle en otras asignaturas (como por ejemplo Bases de Datos), vuestro NIA.
- **Clave**: se encuentra en el fichero README.oracle de vuestra cuenta en Hendrix.

El acceso vía línea de comandos a Oracle desde Hendrix se realiza del siguiente modo

sqlplus nip@vicious.hendrix-oracle.cps.unizar.es

Para acceder se usará la clave que se os indica en el fichero README.oracle.

Creación de tablas en un sistema gestor de bases de datos relacionales

Para la creación de las tablas de la base de datos requeridas se emplearán instrucciones análogas a las siguientes en caso de emplear un SGBD relacional:

```
-- ------------------Tabla usuario-------------------------

CREATE TABLE usuario(

login VARCHAR(20) NOT NULL,

nombre VARCHAR(20) NOT NULL,

CONSTRAINT usuarioPK PRIMARY KEY(login)).

-- ------------------Tabla comentario-------------------------

CREATE TABLE comentario(

comentarioId BIGINT NOT NULL AUTO_INCREMENT,

texto VARCHAR(250) NOT NULL,

usuarioId VARCHAR(20),

CONSTRAINT comentarioPK PRIMARY KEY(comentarioId)

CONSTRAINT usuarioFK FOREIGN KEY(usuarioId)

REFERENCES usuario(login));
```

2.2 Diseño e implementación de la capa de acceso a datos del sistema de información Web.

El patrón de diseño *Value Object* (VO) o *Transfer Object* (TF)

El principal propósito de este patrón es agrupar diferentes valores de propiedades en un objeto para facilitar su envío y recepción. Además, de este modo se facilita la legibilidad y mantenimiento del código (resulta más sencillo añadir nuevos valores de propiedades que se utilizan frecuentemente con los ya agrupados). Si el patrón se implementa en Java, las clases representando a los Value Object suelen extender de la interfaz Serializable.

Más información en la documentación del patrón en el sitio web de Oracle (http://www.oracle.com/technetwork/java/transferobject-139757.html, accedido el día 6 de octubre de 2017).

El patrón de diseño *Data Access Object* (DAO)

El principal propósito de este patrón es abstraer la implementación del acceso a la fuente de datos a las clases que implementan la lógica de negocio. De este modo la lógica de negocio delega en la clase DAO la carga y almacenamiento de los datos y además es posible cambiar la fuente de datos (por ejemplo, pasar de ficheros a un SGBD relacional o a un SGBD NoSQL) sin que las clases que implementan la lógica de

negocio se vean afectadas.
Más información en la documentación del patrón en el sitio web de Oracle (http://www.oracle.com/technetwork/java/dataaccessobject-138824.html, accedido el día 6 de octubre de 2017)

3. Entrega de la práctica

La práctica se realizará en grupos de tres personas. Cuando se finalice la práctica se debe entregar en un fichero denominado práctica2_NIP1_NIP2_NIP3.tar o práctica1_NIA1_NIA2_NIA3.zip (donde NIA1, NIA2 y NIA3 son los NIAs de los autores de la práctica) con el siguiente contenido:
1. Un fichero de texto denominado autores.txt que contendrá el NIA, los apellidos y el nombre de los autores de la práctica en las primeras líneas del fichero. Por ejemplo:

NIA	Apellidos	Nombre
545689	Rodríguez Quintela	Sabela
745689	Caamaño Cives	Antón
677893	López Marín	Adrián

2. Un directorio denominado fuentes/scripts, que contendrá los ficheros empleados para la creación de las tablas o estructura requerida por el sistema gestor de bases de datos seleccionado.
3. Un fichero pdf denominado memoriaPractica2.pdf cuya extensión no excederá de 10 páginas donde se indiquen: los aspectos más relevantes del desarrollo de la práctica (modelo entidad relación, modelo relacional o equivalente, diseño de las clases implementadas en la capa de persistencia de datos, etc.), la metodología de trabajo empleada para el desarrollo de la práctica (recursos, herramientas utilizadas, distribución del trabajo, las decisiones tomadas, planificación, distribución de tareas, etc.), el coste (horas empleadas en cada uno de los apartados) y las principales dificultades encontradas durante la realización de esta.
Al descomprimir el fichero .tar o .zip, se deben extraer los ficheros y directorios indicados en el directorio practica2_NIA1_NIA2_y_NIA3. Es importante seguir las convenciones de nombrado y la estructura de ficheros y directorios descrita.

Para la realización de esta práctica se han planificado 2 sesiones de prácticas. Por tanto, la fecha límite de entrega es el día anterior (a las 23:59) a la quinta sesión de prácticas (Práctica 3) en el laboratorio (29 o 30 de octubre). Para la entrega del fichero .tar o .zip, se utilizará la plataforma Moodle 2 del Anillo Digital Docente de la Universidad de Zaragoza (http://moodle2.unizar.es).

4. Criterios de corrección

Una vez realizadas las prácticas y entregadas estas, cada grupo debe presentarselas al profesorado de prácticas en la siguiente sesión de prácticas. Al realizar la presentación el profesorado le formulará cuestiones sobre las decisiones de diseño e implementación que se han realizado.

La práctica debe entregarse en los términos indicados anteriormente, funcionar correctamente en la máquina virtual proporcionada (vigilar aspectos como por ejemplo los permisos de ejecución de los scripts, el juego de caracteres utilizado en los ficheros, etc.) y no haber sido copiada. También es importante someter código limpio (donde se ha evitado introducir mensajes de depuración y comentarios que no proporcionan información al usuario). Los criterios de corrección se dividen en dos bloques:

- *Requisitos mínimos a cumplir (7 puntos)*
    - Sintaxis y diseño del modelo Entidad-Relación: 2 puntos.
    - Transformación del modelo conceptual Entidad-Relación en el modelo lógico seleccionado (modelo relacional o equivalente): 1 punto.
    - Implementación del modelo lógico en el SGBD seleccionado (elección del tipo adecuado de los atributos de los diferentes componentes, creación de índices para acelerar el acceso, configuración del SGBD para tener en cuenta el idioma, restricciones de clave primaria y foráneas adecuadas, etc.): 2 puntos.
    - Implementación de la capa de acceso a datos de la aplicación Web (documentación y sangrado de los ficheros fuentes .java o equivalente, estructuración de las diferentes clases en paquetes, uso de patrones de diseño): 2 puntos.
- *Otros requisitos (3 puntos)*
    - Aspectos relacionados con la presentación del informe (estructura, redacción, puntuación, índice, referencias y bibliografía ...), justificación de las afirmaciones recogidas en él e inclusión de un apartado metodología y costes para la elaboración de la práctica: 2 puntos.
    - Originalidad, aspectos novedosos que se han considerado: 1 punto.

ANEXO I:

Ficheros de datos del modelo de la base de datos TRAFAIR que se os ha proporcionado anteriormente accesible en la siguientes direcciones:

- https://drive.google.com/file/d/1mzrqb96eCQwV1rvF-eGjibdPpKdJl085/view?usp=sharing (Dump de la base datos de TRAFAIR en Zaragoza en formato ZIP que contiene ficheros CSV).
- https://drive.google.com/file/d/1Nftt3n1-1Q_-qSG6T0JDbK0AapXR4feA/view?usp=sharing (Dump de la TRAFAIR en Zaragoza creado con la herramienta pgAdmin, para el rol trafairz con los parámetros de backup por defecto).