

Práctica 2: Acceso a base de datos e interfaces gráficas de usuario en Java

Asignatura: Ingeniería del Software-II **Curso:** 2008/2009

Sergio Ilarri
Dpto. de Informática e Ingeniería de Sistemas,
Universidad de Zaragoza
silarri@unizar.es

20 de octubre de 2008

1. Objetivos

Los objetivos de esta práctica son:

- Acceder a bases de datos desde Java utilizando *JDBC*.
- Diseñar interfaces gráficas de usuario en Java utilizando *Swing*.
- Manejar el API y la documentación de Java para encontrar la información necesaria.

El entorno de desarrollo a utilizar será JDK 1.5. Se recuerda la necesidad de trabajar en las máquinas locales del laboratorio de prácticas, y no directamente en la máquina *hendrix*.

2. Contexto del Problema

Se pretende desarrollar una aplicación que permita al usuario almacenar y gestionar en una base de datos información acerca de empleados de una empresa de desarrollo de software y los proyectos en los que trabajan. De cada empleado se almacenará la siguiente información: su nombre y apellido, su año de nacimiento, y su NIF. De cada proyecto se guardará el título, la fecha de inicio y fin, y una descripción del proyecto. Además, se almacenará información de qué empleados trabajan en qué proyectos.

2.1. Funcionalidades Requeridas

Se debe diseñar una aplicación Java con una interfaz gráfica de usuario que permita:

1. Introducir datos de empleados.
2. Introducir datos de proyectos.

3. Modificar datos de empleados y proyectos.
4. Asociar empleados a proyectos (un empleado puede estar asociado a varios proyectos si ha trabajado en varios).
5. Obtener la lista de empleados asociados a un proyecto.
6. Obtener la lista de proyectos en los que ha trabajado un empleado.
7. Listar todos los proyectos.
8. Listar todos los empleados.

En cada caso, se considerará la forma adecuada de interactuar con el usuario; por ejemplo, evitando que haya que escribir el nombre completo de un proyecto para poder ver los empleados que intervienen en él, evitando que haya tablas editables cuya actualización no se refleje en la base de datos, etc.

3. Acceso a Bases de Datos

Como recordaréis de clase, el paquete *java.sql* contiene la mayor parte de las interfaces y clases necesarias para acceder a bases de datos. Es importante empezar recordando las ideas fundamentales que vimos y echar un vistazo a la parte del API de Java correspondiente.

El gestor de base de datos a utilizar es Oracle® *9i Enterprise Sparc64 (9.0.1.3.0)*, disponible en la máquina *den*. Para conectarse con la base de datos Oracle desde Java, hay que utilizar la siguiente URL:

```
jdbc:oracle:thin:@den.cps.unizar.es:1521:vicious
```

El nombre del driver a utilizar al realizar la conexión es:

```
oracle.jdbc.driver.OracleDriver
```

y para poder utilizarlo desde código Java es imprescindible que en el *CLASSPATH* de Java se referencie el fichero *classes12.jar*, disponible en las máquinas del laboratorio en:

```
/usr/local/pkg/instantclient_10_2/
```

También es posible interactuar directamente con la base de datos utilizando el cliente *sqlplus* (útil para realizar pruebas):

```
sqlplus <USERNAME>@vicious.den.cps.unizar.es
```

4. Entrega de la Práctica

La práctica se realizará en parejas o de forma individual. Se debe entregar un fichero **p2miNIP.zip** (donde *miNIP* es el NIP de uno cualquiera de los dos miembros de la pareja de prácticas) que incluirá:

1. Un fichero *Autores.txt* que contendrá, exclusivamente, los nombres y apellidos de los autores de la práctica y sus respectivos NIPs. Opcionalmente, se incluirá también un resumen de las principales dificultades encontradas.
2. Un **shell script ejecuta**, que pone en ejecución la aplicación. Debe recibir dos argumentos: el nombre del usuario y la contraseña para acceder a la base de datos. Por ejemplo, *ejecuta pepito miPassword* lanzaría la aplicación conectando con la BD utilizando el nombre de usuario *pepito* y la contraseña *password*. El script *ejecuta* debe definir el *CLASSPATH* necesario (se deben referenciar los directorios propios con un *path* relativo) utilizando la opción *-cp* o *-classpath* de *java*. Al lanzar la aplicación Java correspondiente, **se deben crear automáticamente las tablas** necesarias en la base de datos (**borrando las anteriores**, si ya las hubiera) **e insertar tuplas de ejemplo** que permitan comprobar el correcto funcionamiento de la aplicación.
3. Un **shell script compila**, que compila todos los ficheros fuentes .java.
4. Un **directorio SRC**, que contiene los ficheros fuente Java utilizados en la implementación de la práctica.
5. Un **directorio CLASSES**, que contiene los ficheros de las clases Java. En este directorio es donde depositará las clases el shell script *compila*.

Es imprescindible seguir las convenciones de nombrado (incluido el uso adecuado de mayúsculas y minúsculas) y la estructura de ficheros y directorios (SRC, CLASSES, etc.) descrita. Al descomprimir el fichero .zip **se deben extraer los ficheros y directorios indicados en el directorio actual**. Para entregar el fichero .zip, se utilizará el comando *someter*:

```
someter is_07 p2miNIP.zip
```

La práctica debe entregarse antes del **15 de diciembre de 2008**. También es posible entregar la práctica en septiembre de 2009, pero en ese caso puntuará la mitad. Recordad que las reglas de la asignatura establecen que **para que la práctica sea corregida es necesario haber asistido a la correspondiente sesión de prácticas**, tanto si se entrega ahora como si se entrega en septiembre.

5. Criterios de Evaluación

La práctica tiene una valoración de 0.5 puntos sobre el total de 10 de la asignatura, de acuerdo con los siguientes criterios:

1. La práctica debe entregarse en los términos indicados anteriormente, funcionar correctamente, no haber sido copiada, y haber asistido a la sesión de prácticas de forma activa. El tratamiento de errores debe ser adecuado, de forma que si se producen debería informarse adecuadamente del tipo de error producido. La práctica debe funcionar en cualquier ordenador con Java 1.5 y acceso al servidor de Oracle (independientemente de si las tablas y/o la base de datos existen ya o no). En particular, hay que asegurarse también de que la práctica funciona correctamente en los ordenadores del laboratorio (vigilar aspectos como los permisos de ejecución de los scripts, juego de caracteres utilizado en los ficheros,

etc.). También es importante someter código limpio (donde se ha evitado introducir mensajes de depuración que no proporcionan información al usuario). Si se cumplen estas condiciones, se obtienen 5 puntos sobre 10.

2. Si la estructura es adecuada (clases estructuradas adecuadamente en paquetes, etc.), se obtienen 3 puntos extra sobre 10.
3. Los restantes 2 puntos (sobre 10) dependerán de una valoración global del trabajo realizado: la interfaz gráfica de usuario, la no utilización de posicionamiento absoluto, el uso de comentarios, estructura de los fuentes, etc.
4. Se penalizará si se detectan errores leves (si son graves, no se alcanzan los mínimos exigidos en el primer punto) en la implementación (por ejemplo, que determinada funcionalidad no funciona correctamente en ciertas circunstancias).