Año 1 • Número 6 • **5,98** euros

TODO Programación

La Revista mensual para entusiastas de la programación

www.iberprensa.com

Año 2006
Windows Vs. Linux
¿Qué plataforma se impondrá?

Acceso a BB.DD. desde aplicaciones Java

Desarrollo J2EE con WebLogic Workshop 8.1

Introducción a Oracle PL/SQL

Studio PRESS

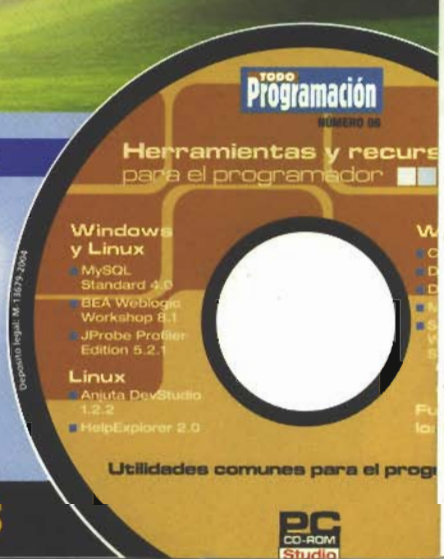
■ EN EL CD-ROM: MYSQL, ANJUTA, CAPE CLEAR, DB ARCHITECT, MASAL EDITOR, ETC.

Zona Linux

- **Mono 1.0:** Instalación paso a paso del entorno
- **C#:** Cómo intercambiar información entre las clases

Zona Windows

- **SharpDevelop:** Un entorno de desarrollo libre para .NET
- **Clase Environment:** Cómo controlar el entorno de .NET



CURSO PRÁCTICO DE PROGRAMACIÓN DE JUEGOS JAVA PARA MÓVILES

TODO Programación

La Revista mensual para entusiastas de la programación

DIRECTOR
Eduardo Toribio
etoribio@iberprensa.com

REDACCIÓN
Yenifer Trabadela
yenifer@iberprensa.com

COLABORADORES
Antonio M. Zugaldía
(azugaldia@iberprensa.com)
David Santo Orcero
(orcero@iberprensa.com)
Manuel Domínguez
(mdominguez@iberprensa.com)
Fernando Escudero
(fescudero@iberprensa.com)
José Manuel Navarro
(jnavarro@iberprensa.com)
Marcos Prieto
(mprieto@iberprensa.com)
Guillermo "el Guille" Som
(elguille@iberprensa.com)
Santiago Márquez
(smarquez@iberprensa.com)
Lorenzo Gil
(lgil@iberprensa.com)
José Rivera
(jrivera@iberprensa.com)
Jaime Anguiano
(janguiano@iberprensa.com)
Alejandro Serrano
(aserrano@iberprensa.com)

DISEÑO PORTADA
Antonio Gª Tomé

MAQUETACIÓN
Antonio Gª Tomé

DIRECTOR DE PRODUCCIÓN
Carlos Peropadre
cperopadre@iberprensa.com

SUSCRIPCIONES
Marisa Cogorro

SUSCRIPCIONES
Tel: 91 628 02 03
suscripciones@iberprensa.com

FILMACIÓN: Fotpreim Duvial
IMPRESIÓN: I. G. Printone
DUPLICACIÓN CD-ROM: M.P.O.

DISTRIBUCIÓN
S.G.E.L.
Avda. Valdelaparra 29 (Pol. Ind.)
28108 Alcobendas (Madrid)
Tel.: 91 657 69 00

EDITA: Studio Press
www.iberprensa.com

REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN
C/ del Río Ter, 7. Polígono "El Nogal"
28110 Algete (Madrid)
Tel.: 91 628 02 03*
Fax: 91 628 09 35
(Añada 34 si llama desde fuera de España.)

DEPÓSITO LEGAL: M-13679-2004
Número 06 • Año 1

EDITORIAL

Eduardo Toribio

Plataforma de futuro

Hemos intentado un curioso experimento, hemos cogido las dos plataformas más populares hoy en día. Windows y Linux, para enfrentarlas en una comparativa. Pero no en una comparativa tomando como parámetros las características actuales que ofrecen a sus usuarios, sino tratando de adelantarnos al tiempo para comparar cómo serán en un par de años, según entendemos nosotros evolucionarán ambos sistemas.

En fin, no deja de ser un intento de adivinar cómo puede ser el futuro, pues en muchas facetas solo tenemos algunos esbozos de lo que serán o implementarán. Además de este reportaje, quiero aprovechar estas líneas para recomendaros un par de lecturas más, primero el artículo sobre acceso a la base de datos MySQL desde aplicaciones Java, y segundo, la entrega correspondiente al curso de programación de juegos Java para móviles, en la que ya vamos acabando con conceptos teóricos para meternos en la parte práctica con ejemplos incluidos.

SUSCRIPCIONES

Como oferta de lanzamiento existe la posibilidad de suscribirse durante un año (12 números) a **Todo Programación** por solo 61 euros lo que significa un ahorro del 15% respecto el precio de portada. Además de regalo se incluye un archivador para coleccionar y guardar las revistas con sus CD-ROMs.
Más información en: www.iberprensa.com

SERVICIO TÉCNICO

Todo Programación dispone de una dirección de correo electrónica y un número de Fax para formular preguntas relativas al funcionamiento del CD-ROM de la revista.
e-mail: todoprogramacion@iberprensa.com
Fax: 91 628 09 35

LECTORES

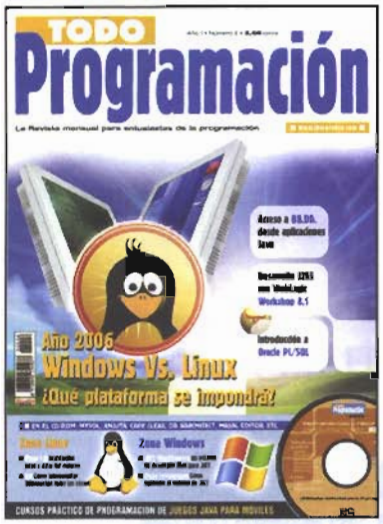
Comparte con nosotros tu opinión sobre la revista, envíanos tus comentarios, sugerencias, ideas o críticas.

Studio Press
(Todo Programación)
C/ Del Río Ter, Nave 13
Pol. "El Nogal"
28110 Algete. Madrid

DEPARTAMENTO DE PUBLICIDAD

Si le interesa conocer nuestras tarifas de publicidad no dude en ponerse en contacto con nuestro departamento comercial:
- **Tel. 91 628 02 03**
- **e-mail: publicidad@iberprensa.com**

Sumario



Número 6

A quién vamos dirigidos

Todo Programación (TP) es una revista para programadores escrita por programadores y con un enfoque eminentemente práctico. Trataremos de ser útiles al programador, tanto al profesional como al estudiante. Si hay algo cierto en este sector es que nunca podemos parar, vivimos en un continuo proceso de reciclaje. Ahí es donde tratará de encajarse TP: información, actualidad, cursos y prácticas de los lenguajes más demandados y formación en sistemas.

Ejemplos y código fuente

Cada CD-ROM de la revista incluye una carpeta denominada *fuentes* en la que se encuentra el material complementario para seguir cada uno de los cursos: por ejemplo, los listados completos, los ejemplos desarrollados en diversos lenguajes, compiladores, editores, utilidades y en general, cualquier herramienta que se mencione o cite en la respectiva sección. Todo con la finalidad de completar la formación y facilitar el seguimiento de cada artículo por parte del lector.

En portada

Año 2006: Windows Vs. Linux

¿Cuál es la plataforma de futuro?

10 Nos hemos propuesto realizar una comparativa "en potencia" entre Windows y Linux. Es decir, no comparamos según las características actuales sino que lo hacemos sobre ambos sistemas a medio plazo. Para ello hemos juntado a dos especialistas en cada sistema, firmas habituales de la casa: Fernando Escudero, recien llegado del Teched la parte Windows y Rocío Arango, consultora en software libre la parte Linux. Hemos seleccionado lo que entendemos son áreas más interesantes a cubrir y hemos desarrollado nuestra comparativa.

ZONA LINUX

Actualidad Desarrollo >>

18 Desde hace unas semanas disponemos de la versión definitiva de Mono 1.0, por tanto vamos a estudiar su proceso de instalación y algunos ejemplos de cómo trabajar sobre una de las tecnologías que provee Mono.

Mono: Clases en C# >>

22 En esta ocasión vamos a comprobar con ejemplos prácticos las diversas maneras que tienen los métodos de intercambiar información entre ellos. Después hablaremos de los métodos virtuales.

CONTENIDO DEL CD-ROM

Herramientas y recursos para el programador

64 Este mes nuestro CD viene repleto de utilidades interesantes para el programador. Para ambas plataformas encontramos el gestor de bases de datos MySQL. Para los usuarios Windows aplicaciones tan útiles como Cape Clear, DB Arachitect o MaSal Editor. Finalmente, para desarrolladores Linux, Anjuta DevStudio, uno de los IDEs de desarrollo más populares del momento. Además recordad que en la sección *Código Fuente* los ejemplos y utilidades que se destacan relativas a cada artículo de la revista.

TALLER PRÁCTICO

Acceso a BB.DD. Desde aplicaciones Java >>

35 A lo largo de dos entregas vamos a estudiar cómo acceder a la base de datos MySQL desde aplicaciones Java. Lo haremos con ejemplos prácticos y paso a paso para que resulte lo más sencillo posible.

SharpDevelop, un IDE libre para .NET >>

39 Estamos ante una magnífica aplicación libre para entornos Windows, con ella podemos desarrollar para .NET sin necesidad de las herramientas de Microsoft

Análisis Software: Desarrollo J2EE con WebLogic Workshop 8.1 >>

43 Workshop 8.1 es la nueva versión de un entorno de desarrollo Java para J2EE operativo en Windows y Linux, que Bea tiene la firme decisión de popularizar lo máximo posible entre la comunidad de desarrolladores Java con una nueva licencia de distribución.

.net ZONA WINDOWS

Controlar nuestro entorno de .NET por medio de la clase Environment >>

27 Las aplicaciones deberían disponer de información de cada usuario que las utiliza, para de esta forma poder personalizarlas todo lo posible. Para que nuestras aplicaciones de .NET dispongan de esa información, entre otras, tenemos la clase Environment.

Programación orientada a objetos en .NET >>

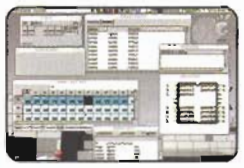
31 La conocida como POO nos permite escribir código menos propenso a fallos, además de la reutilización del mismo de forma más conveniente. Veamos una introducción a la POO en .NET.

NOTICIAS

6. **Sun** lanza la versión definitiva de Java Studio Creator.
6. **Herramientas** SELESTA INTROSCOPE.
7. **Las** soluciones AMP de VERITAS soportan Java Enterprise System.
7. **Petición** de trabajos para el II Congreso JavaHispano.
8. **Resultados** Sun.
8. **IBM** en la Universidad.
8. **Open** Group
8. **Acuerdo** entre Macromedia y Bellwave.
9. **Portátil** Toshiba Satellite A50.
9. **HP** Maestro.
9. **Cámara** digital multifunción DV5500 de Yukai.

Y ADEMÁS...

C para Hackers: Con X de HEXadecimal >>



56 Vamos a repasar algunos conceptos fundamentales que serán de gran interés para el programador C: los códigos binarios y hexadecimal, el álgebra de Boole y los operadores binarios y su uso.

Juegos Java: Los MIDLets y la configuración CLDC >>

51 Necesitamos explicar algunos conceptos teóricos más antes de poder pasar al primer programa desarrollado por nosotros para dispositivos móviles que, cómo no, será el archiconocido Hola Mundo.

Bases de datos: Introducción a Oracle PL/SQL >>

60 Una vez vistos el lenguaje SQL estándar y la forma de normalizar bases de datos, nos vamos a introducir en la forma de programar desde el propio SQL tomando como ejemplo el lenguaje PL/SQL de Oracle.

CUADERNOS DE PRINCIPIANTES

El acceso a los ficheros >>

46 Vamos a introducir al lector en lo que es un sistema de ficheros, los distintos tipos de sistemas que existen para Windows y Linux y cómo condicionan al programador a la hora de acceder y organizar la información en un determinado sistema.

NIVEL DE DIFICULTAD:

MATERIAL INCLUIDO EN EL CD-ROM:
Ejemplos ■ SQL de los ejemplos ■ J2SDK + Netbeans (Windows y Linux)

Acceso a BB.DD. desde aplicaciones Java JDBC 1.0

MANUEL DOMÍNGUEZ
mdominguez@iberprensa.com

Con la aparición de aplicaciones distribuidas, el acceso a un único conjunto común de datos se hace muy importante.

Con Java tenemos la opción de crear un programa multiplataforma para entornos heterogéneos, y para el acceso a unos únicos datos de forma distribuida están los sistemas gestores de bases de datos (SGBD). Uniendo ambas cosas crearemos aplicaciones que sumen las ventajas de los dos.

APLICACIONES CON JAVA Y MYSQL

Java incorpora una API diseñada para el acceso a bases de datos, llamada JDBC (*Java Database Conectivity*) Estas bases de datos deben estar controladas por sistemas gestores que sean compatibles con dicha API de Java; esto es, que proporcionen un controlador sobre el cual nuestro programa Java pueda acceder a los datos (se llama controlador JDBC). Actualmente Oracle, Informix..., en definitiva, todos los SGBD comerciales, proporcionan conectividad con Java a través de sus respectivos controladores JDBC. Estos sistemas son caros, pesados y complejos de administrar y no son una alternativa demasiado viable para aplicaciones ligeras y sencillas como contabilidad familiar, gestión, TPV, almacén, fontanería, talleres... por lo cual hemos de buscar alternativas. En el mundo del software libre existen muchas posibilidades como por ejemplo MySQL o PostgreSQL. Nosotros utilizaremos MySQL junto a Java para los ejemplos de este minicurso de dos entregas, aunque la forma de trabajar es idéntica para cualquiera que se use. MySQL es menos sofisticado que Oracle o Informix pero infinitamente más ligero, barato y sencillo.

ÍNDICE CURSO C PARA HACKERS

PREPARANDO EL ENTORNO

Antes de introducirnos en el código de un programa que acceda a bases de datos, necesitaremos instalar algunas herramientas en nuestro PC. Las que hemos seleccionado se encuentran en Windows y en Linux, de forma gratuita, por lo que las explicaciones valen para ambos sistemas. Necesitaremos:

- Un entorno integrado de desarrollo que nos facilitará la creación de programas Java y concretamente de los interfaces de usuario.
- La plataforma de desarrollo de Java, proporcionada por Sun.
- Un controlador JDBC para poder acceder a bases de datos MySQL

Y daremos por supuesto que tenemos bien instalado y configurado MySQL y sabemos crear una base de datos en él. Además tendremos usuarios y permisos suficientes para los ejemplos que se explicarán. Todo esto debe estar funcionando correctamente

Instalación de Netbeans IDE y J2SE SDK para Windows

NetBeans IDE es un entorno de desarrollo software muy completo que permite crear plantillas para muchos tipos de aplicaciones Java, interfaces gráficos de forma rápida y es extensible mediante plugins. J2SE SDK es la plataforma de desarrollo de Java, puesta a disposición de todos por Sun Microsystems.

El primero de ellos se puede descargar de *http://www.netbeans.org* y el segundo de *http://java.sun.com*. De todas formas, de este último sitio también se puede descargar un paquete llamado *J2SDK 1.4.2 – NetBeans Cobundle* que contiene las dos cosas en una, lo que nos resultará más cómodo. Concretamente el fichero usado en este cursillo *es j2sdk-1_4_2-nb-3_5_1-bin-windows.exe* sobre el que solo hace falta hacer doble clic para comenzar a instalar todo. En el CD-ROM de la revista encontrarás esa versión y otra para Linux.

La instalación resulta sencilla. Simplemente se nos pedirá un lugar donde deseamos instalar NetBeans y Java 2 Software Development Kit y las ventanas típicas para aceptar la licencia del programa. En pocos toques de ratón todo estará instalado sin problemas. El uso de NetBeans es bastante intuitivo y excede el objetivo de este artículo. Los ejemplos que presentamos están compuestos por un único fichero Java, por lo no hay que hacer demasiadas cosas para que todo funcione bien.

Instalación de Connector/J Para MYSQL

Connector/J es el controlador JDBC para acceso desde Java a bases de datos MySQL. Se distribuye como un fichero JAR y se puede descargar de *http://www.MySQL.com*. Lo único que hay que hacer con el fichero que obtendremos de este sitio es descomprimirlo. Al hacerlo encontraremos un fichero llamado *mysql-*

Web oficial de MySQL.

Con Java podemos crear un **programa multiplataforma** para entornos heterogéneos

connector-java-xxxx-bin.jar donde (xxxx) será un número que dependerá de la versión que se esté utilizando.

Para que Java acceda a las clases de este paquete podemos hacer dos cosas: modificar la variable de entorno *CLASSPATH* para que apunte a dicho fichero o, lo que resulta más cómodo en la mayor parte de las ocasiones, colocar dicho fichero JAR en el directorio */jre/lib/ext* que se encuentra dentro del directorio donde hayamos instalado J2SDK. Con esto ya está finalizada la instalación del controlador JDBC para MySQL.

INTRODUCCION A JDBC 1.0

Para todos los ejemplos de este artículo vamos a utilizar una base de datos llamada *pruebas* que contendrá una única tabla llamada *personas*. La estructura de esta tabla se puede ver en la Tabla 1; el SQL asociado se encuentra en el CD-ROM que acompaña a la revista. Para conectarnos vamos a utilizar un usuario llamado *usuarioTP* que va a tener permiso completo para esta base de datos y se podrá conectar desde cualquier lugar. Como dijimos, la creación de la base de datos y los permisos correspondientes excede el cometido de este artículo y se supone que ya se sabe hacer. Además, dado lo famagoso del código de los interfaces gráficos, únicamente mostraremos las porciones del código que hacen operaciones vía JDBC, el resto se puede consultar en el CD-ROM de la revista.

En cualquier caso, siempre que se desee utilizar la API JDBC en un programa Java, se debe importar los paquetes *java.sql.** y, opcional aunque recomendable, *javax.sql.**

Conversión de tipos SQL a tipos Java

El primer problema que nos encontramos cuando obtenemos datos de una base de datos mediante JDBC, es que los tipos SQL soportados por el sistema gestor de bases de datos no son los mismos que los tipos de Java, por lo que hay que hacer una conversion. Esto no es demasiado problemático puesto que la API de JDBC incorpora mecanismos para obtener el tipo deseado, pero hay que conocer cuáles son las posibilidades de conversión. En la Tabla 2 vemos un resumen muy escueto de los tipos SQL más usados y su posible conversión a tipos Java. Sabiendo esto, no tendremos problemas a la hora de utilizar los datos obtenidos. El listado completo de compatibilidades entre tipos se puede obtener de *http://java.sun.com*.

Conexión

Si tenemos una base de datos y queremos acceder a ella desde Java, el primer paso es conectarnos a ella. Debemos crear en memoria una instancia del controlador JDBC que estemos usando, en nuestro caso, *Connector/J*. Esto se hace con el método *Class.forName(nombreDelControlador)*. Para nuestros ejemplos, será:

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
```

De este modo estamos en disposición de conectar a una base de datos MySQL. El siguiente paso y más importante es crear la conexión propiamente dicha. La clase *Connection* será la que utilizaremos y haremos uso del gestor de controladores de Java para crear la conexión correcta. La forma de hacer esto sería usando *DriverManager.getConnection(URL, usuario, clave);* donde *usuario* y *clave* son cadenas de texto que identifican al usuario de la base de datos a la que queremos acceder y URL es una cadena de texto con la forma *jdbc:<subprotocolo>://<host>:<puerto></bd>* y que en nuestro caso será *"jdbc:mysql://localhost/prueba"*. Así que para realizar la conexión debemos usar todo esto de la siguiente forma:

```
Connection miConexion = DriverManager.getConnection("jdbc:mysql://localhost/prueba", "usuarioTP", "unaClave");
```

Y a partir de ahí, si todo ha ido bien, tendremos una conexión establecida con la base de datos y podremos usar los métodos de la clase *Connection* cuando sea necesario. Por ejemplo, a la hora de cerrar la conexión con la base de datos deberemos usar el método *Connection.close();* lo cual liberará los recursos. Siguiendo con las instrucciones de más arriba, nosotros deberíamos hacer *miConexion.close()* cuando hayamos terminado.

Con estas tres sencillas reglas, cargar el controlador, abrir una conexión y cerrarla, podemos ir probando que todo funciona bien. El fichero *Ejemplo1.java* implementa un ejemplo ilustrativo de una aplicación que conecta con nuestra base de datos y desconecta a nuestro antojo.

Vista de Ejemplo1.java.

En el Listado 1 se muestran las líneas JDBC que se han utilizado. El código es sencillo así que lo mejor es observar directamente el código fuente del ejemplo que está en el CD de la revista.

El fichero *Ejemplo2.java* muestra el mismo programa pero ahora pide la introducción de la clave en un cuadro de texto. Es conveniente hacerlo siempre así puesto

Listado 1. Proceso de conexión a la base de datos

```
private void clicEnConectar(java.awt.event.ActionEvent evt){
...
DriverManager.getConnection("jdbc:mysql://localhost/pruebas",
"usuarioTP", "clave");
...
}
```

Tabla 1. Estructura de la tabla "personas"

Nombre del campo	Tipo	Clave primaria	Valor por defecto
Id	Int(11)	Si	0
Nombre	Varchar(50)	No	""
Apellido1	Varchar(50)	No	""
Apellido2	Varchar(50)	No	""

Tabla 2. Compatibilidad entre tipos

Tipo SQL	Tipo Java óptimo	Otros tipos Java posibles		
Integer	Int	Long	String	Object
Real	Float	Double	String	Object
Bit	Boolean	Byte	String	Object
Char	Char	Byte	Short	Object
Date	Date	String	Byte[]	Object
Time	Time	String	Byte[]	Object
varchar	String	Byte[]	Object	

que el *bytecode* Java se desensambla fácilmente y cualquier clave que estuviese formando parte del código se puede descubrir con rapidez.

Vista de Ejemplo2.java.

Consulta

Bien, ya sabemos abrir y cerrar una conexión JDBC; ahora necesitamos poder realizar consultas (en SQL) a dicha base de datos. Para ello utilizaremos la clase *Statement*, también de la API JDBC. Una consulta siempre se monta sobre una conexión establecida, por lo que tenemos que hacer uso del objeto *miConexion* que hemos creado líneas atrás. Éste nos proporciona el método *createStatement()* cuyo cometido es devolver un objeto *Statement*

Así, para nuestro caso, deberíamos hacer

```
Statement consulta =
miConexion.createStatement();
```

El objeto *consulta*, de tipo *Statement*, nos proporcionará ahora una serie de métodos que tienen la finalidad de lanzar consultas SQL contra la base de datos deseada, a través de la conexión abierta. Los métodos son muchos y muy variados, pero se suelen usar con frecuencia tres de ellos. *executeQuery(SQL)*, que toma como parámetro una cadena de texto que es la consulta SQL que deseamos lanzar contra MySQL, devuelve los registros producidos por la consulta, se usa para consultas de tipo *SELECT*. *executeUpdate(SQL)*, cuya finalidad es la misma que el método anterior pero que no devuelve el resultado de la consulta sino el número de registros de la base de datos que se han visto afectados por ella, por ejemplo, consultas de tipo *UPDATE*, *DELETE* o *INSERT*. Por último, al igual que con la conexión, la consulta se debe cerrar cuando no es necesaria. Para ello está el método *close()*.

El fichero *Ejemplo3.java* mejora el *Ejemplo2.java* añadiéndole una consulta que permite saber el número de personas que hay en nuestra base de datos. No vamos a ver de nuevo cómo se conecta o desconecta porque el ejemplo es igual en ese aspecto; lo que sí nos interesa es ver cómo se realiza esa consulta

De nuevo el código es sencillo por lo que lo mejor es abrir el código fuente del CD de la revista, compilarlo y probarlo.

Vista de Ejemplo3.java.

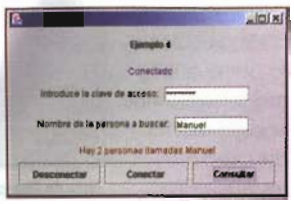
En el Listado 2, se muestran las líneas correspondientes al código JDBC de este ejemplo.

Por su parte, el fichero *Ejemplo4.java* mejora a *Ejemplo3.java*, añadiéndole la opción de introducir un nombre en un cuadro de texto y averiguar cuántas personas hay en la base de datos cuyo nombre coincida con el especificado. La estructura es la misma, la consulta SQL se pasa al objeto *Statement* que consulta la base de datos;

Listado 2. Proceso de consulta a la base de datos

```
private void clicEnConsultar(java.awt.event.ActionEvent evt)
...
consulta = conexion.createStatement();
resultado = consulta.executeQuery("select count(id) as contador from
personas");
if (resultado.next()) {
if (resultado.getInt("contador") == 1)
this.jLabel4.setText("Hay 1 persona en la BB.DD.");
else
this.jLabel4.setText("Hay " + resultado.getInt("contador") + " personas
en la BB.DD.");
}
consulta.close();
consulta = null;
...
}
```

pero es interesante ver cómo se puede ir complicando la aplicación a medida que interactuemos más con la base de datos



Vista de Ejemplo4.java.

Tratamiento de resultados

Hasta este momento solo sabemos crear una instancia del controlador JDBC, conectar y desconectar de la base de datos, abrir y cerrar consultas y ejecutarlas, lo que nos devolverá el número de registros afectados o el resultado de su ejecución. No es suficiente aún para poder realizar una aplicación útil; necesitamos poder recorrer los resultados obtenidos y acceder a cada uno de sus campos. La clase *ResultSet* es la que almacena los resultados de la consulta y la que nos va a permitir hacer esto; es por tanto el tipo de datos devuelto por *Statement.executeQuery(SQL)* y en nuestro caso se obtendría con una línea como:

```
ResultSet r =
consulta.executeQuery("select *
from personas");
```

La clase *ResultSet* se implementa como una tabla con tantas columnas y filas como columnas y registros se hayan obtenido de la realización de la consulta SQL. La forma de acceder a cada fila (registro de la BB.DD.) es mediante el avance de un puntero que, en principio, no apunta a ningún registro y que irá avanzando desde el primero hasta el último hacia delante, hasta que no queden más registros en el resultado. Cada vez que se avance dicho puntero y hasta que se haga de nuevo, se pueden consultar cada uno de los campos de la tupla que está siendo apuntada en ese momento. Avanzamos el puntero con el método *ResultSet.next()* que devolverá *CIERTO* si quedan registros por visitar y *FALSE* en caso contrario. Si devuelve *CIERTO*, además incrementa el puntero al siguiente resultado de la tabla de resultados, con lo que es fácil recorrer los resultados de haber realizado una consulta, con un código como:

```
While(r.next()) {
  //leo los resultados
}
```

Donde *leo los resultados* se refiere a leer los campos de ese registro. Por ejemplo, en nuestro caso los campos son *id, nombre, apellido1 y apellido2* y podríamos acceder a cada uno de ellos para el registro apuntado en ese momento. Esto se hace con los métodos *getXXX(nombreColumna)* donde XXX puede sustituirse por *Float, Int, Bytes, Long...* en cuyo caso devolverá esos valores; *nombreColumna* indica el campo que se va a leer de dicho registro. En nuestro caso el siguiente código:

```
int i = getInt("id");
```

Nos devolvería el valor del campo *"id"* para el registro actual, como un valor entero. *getString("id");* nos lo devolvería como una cadena de caracteres. De ahí la importancia de saber la conversión entre datos SQL y datos Java. No se podría, por ejemplo, leer el apellido de una persona de los resultados obtenidos como si fuera un *boolean*.

Vista de Ejemplo5.java.

Vista de Ejemplo6.java.

Con esto ya podemos hacer una aplicación Java que interactúe con bases de datos, aunque de forma muy tosca aún. Una vez hayamos terminado de trabajar con los resultados, debemos indicarlo cerrando dichos resultados; para ello usamos el método *ResultSet.close()*. En nuestro caso *r.close();*

El fichero *Ejemplo5.java* actualiza a *Ejemplo4.java*. Añade la opción de cambiar el nombre a personas de la base de datos. Básicamente se trata de un ejemplo ilustrativo de cómo usar sentencias SQL del tipo *UPDATE* en Java.

Por otro lado, el fichero *Ejemplo6.java*, el último de esta entrega, modifica *Ejemplo5.java* de tal forma que al realizar la búsqueda de de personas por su nombre no solo muestra el número de coincidencias, sino que muestra en la ventana los datos de aquellas personas que se han encontrado. Así se muestra lo que hemos explicado sobre cómo acceder a los campos de un *ResultSet*.

CONCLUSIONES

Con pocas líneas nos hemos ahorrado el tener que pensar un formato de fichero para almacenar nuestros datos. Además, el código generado es el mismo para cualquier SGBD que utilicemos; solo hay que modificar dos líneas: la que crea una instancia del controlador JDBC en memoria, y la que crea la conexión con la base de datos. Esto nos permite tener una aplicación muy independiente de la localización física y lógica de los datos.

En la siguiente entrega veremos cómo hacer este acceso a los datos más eficiente.

Listado 3. Proceso de consulta y muestra de datos

```
private void clicEnConsultar(java.awt.event.ActionEvent evt) {
...
consulta = conexion.createStatement();
resultado = consulta.executeQuery("select * from personas where
nombre='"+this.jTextField1.getText()+"'");
while (resultado.next()) {
String CadenaAux = new String(resultado.getString("Apellido1"));
CadenaAux = CadenaAux + " " + resultado.getString("Apellido2");
CadenaAux = CadenaAux + ", " + resultado.getString("Nombre");
CadenaAux = CadenaAux + '\n';
this.jTextArea1.append(CadenaAux);
contador++;
}
...
consulta.close();
consulta = null;
...
}
```

La mejor formación técnica apta para todos

FORMACIÓN 100% GARANTIZADA

1 Aprende a programar tus propios juegos

Programación y diseño de Videojuegos

- La primera obra para aprender a desarrollar un videojuego 3D completo incluyendo gráficos, animaciones, menús y toda la programación.
- Formación en la herramienta de programación Blitz 3D, incluida en el primer CD-ROM.
- Se incluye el desarrollo paso a paso con todo el código fuente de un juego de acción 3D: Zone of Fighters.

COMPOSICIÓN: 20 coleccionables en dos tomos de 200 páginas y 20 CD-ROMS
NIVEL: Principiantes a nivel medio
TEMARIO: Zona gráficos, Zona desarrollo, Blitz 3D, Historia del videojuego, cuestionario y desarrollo del juego Zone of Fighters
REQUERIMIENTOS: Pentium 200 o superior. 64 Mb RAM. Windows 98/Millenium/XP
PVP: ~~120 euros~~.
OFERTA SUSCRIPCIÓN: 107,82 euros
Incluye las tapas archivadoras de regalo.

2 Creatividad publicitaria aplicada al diseño gráfico y web

Diseño publicitario

- Una obra práctica que cubre todos los aspectos y temas necesarios para dominar el diseño gráfico publicitario actual.
- Aprende las técnicas, los métodos y los trucos que utilizan los profesionales del diseño en el día a día de una forma sencilla, práctica y eficaz.
- Realiza los ejercicios paso a paso, empezando a diseñar de forma profesional con un estilo actual.

COMPOSICIÓN: 20 coleccionables en dos tomos de 200 páginas y 20 CD-ROMS
NIVEL: Principiantes a nivel medio
TEMARIO: Teoría de la publicidad, diseño editorial, diseño corporativo, diseño web y zona de tutoriales
REQUERIMIENTOS: Pentium 200 o superior. 64 Mb RAM. Windows 98/Millenium/XP
PVP: ~~120 euros~~.
OFERTA SUSCRIPCIÓN: 101,85 euros
Incluye las tapas archivadoras de regalo.

3 El primer curso para usuarios de videocámaras digitales

Vídeo digital

- Curso orientado a usuarios de videocámaras digitales y entusiastas de la edición digital.
- Aprende a trabajar con los programas más importantes de tratamiento de vídeo.
- Cómo grabar, editar y montar correctamente tus propios vídeos.
- No son necesarios conocimientos previos para seguir este curso.

COMPOSICIÓN: 20 coleccionables en dos tomos de 200 páginas y 20 CD-ROMS
NIVEL: Principiantes
TEMARIO: Teoría de vídeo digital, formatos, compresiones, técnicas de edición, montaje, efectos, transiciones. Prácticas de producción de vídeo. Tutoriales sobre las herramientas que se incluyen en los CD-ROMs
REQUERIMIENTOS: Pentium 200 o superior. 64 Mb RAM. Windows 98/Millenium/XP
PVP: ~~120 euros~~.
OFERTA SUSCRIPCIÓN: 99,95 euros
Incluye las tapas archivadoras de regalo.

Cupón de pedido (Fotocopie y rellene el siguiente cupón de pedido)

Nombre: ..... Apellidos: .....

Dirección: ..... Tel.: ..... E-mail: ..... @ .....

Población: ..... C.P.: ..... Provincia: .....

Corta por la línea de puntos, pégalo y envíalo al Apartado F.D. 1 (28110 Algete). (No necesita sello).

(No olvides firmar tu cupón). ¡Para menores se requiere firma paterna). Oferta válida durante 6 meses desde la publicación. La información que nos facilites será incluida en un fichero automatizado, tienes el derecho de acceso, rectificación y cancelación dirigiéndote a Studio Press, S.L. C/ del Río Ter, Nave 13. Polígono "El Nogal", 28110 Algete (Madrid). (Ley de protección de datos).

- ☐ Adjunto cheque a nombre de STUDIO PRESS, S.L.
- ☐ Domiciliación bancaria, código de la cuenta:

☐☐☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐☐☐☐☐☐☐

☐ Visa ☐ 4B ☐ Master Card Fecha de caducidad: ☐☐ ☐☐

(Visa electrón no válida) ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Firma

Pedido de:
- ☐ Programación y diseño de Videojuegos 107,82 €
- ☐ Diseño publicitario 101,85 €
- ☐ Vídeo Digital 99,95 €

Total Pedido ..... €
Marque con una "X" su pedido.

Por Teléfono: 916280203 (de 9 a 14h y de 15h. a 18 h.)

Por Fax: 916280935

Por Correo electrónico: suscripciones@iberprensa.com