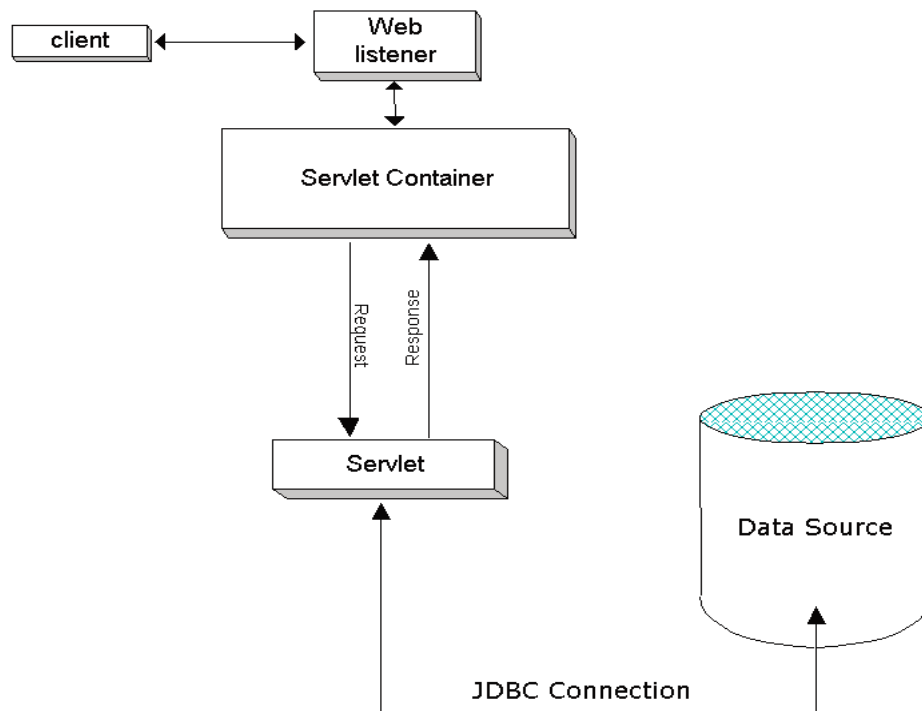




DESARROLLO DE APLICACIONES WEB CON SERVLETS

DESARROLLO DE APLICACIONES WEB CON SERVLETS

ACCESO A DATOS DESDE UN SERVLET



Vamos a realizar un ejercicio a modo de ejemplo para ilustrar el acceso a un servlet mediante la tecnología JDBC y el API Servlet.

La arquitectura de los servlets hace que la escritura de aplicaciones que se ejecuten en el servidor sea relativamente sencilla y, eso sí, sean aplicaciones muy robustas. La principal ventaja de utilizar servlets es que se puede programar sin dificultad, la información que va a proporcionar entre peticiones del cliente.

Es decir, se puede tener constancia de lo que el usuario ha hecho en peticiones anteriores e implementar funciones de tipo *rollback* o *cancel transaction* (suponiendo que el servidor de base de datos las soporte). Además, cada instancia del servlet se ejecuta dentro de un hilo de ejecución Java, por lo que se pueden controlar las interacciones entre múltiples instancias; y al utilizar el identificador de sincronización, se puede asegurar que los servlets del mismo tipo esperan a que se produzca la misma transacción, antes de procesar la petición; esto puede ser especialmente útil cuando mucha gente intenta actualizar al mismo tiempo la base de datos, o si hay mucha gente pendiente de consultas a la base de datos cuando ésta está en pleno proceso de actualización.

DESARROLLO DE APLICACIONES WEB CON SERVLETS

Este ejemplo consiste en un servlet, que junto con la página web asociada, y las dos tablas que utiliza, conforma un servicio completo de noticias o artículos.

Hay un número determinado de usuarios que tienen autorización para enviar artículos, y otro grupo de usuarios que pueden ver los últimos artículos. Por supuesto, que también se podría almacenar la fecha y la hora en que se colocan los artículos, y posiblemente también fuese útil la categorización, es decir, colocarlos dentro de una categoría como deportes, internacional, nacional, local, etc. De este modo, lo que aquí se esboza como la simple gestión de artículos, podría ser la semilla de un verdadero *servicio de noticias*. Pero el autor no pretende llegar a eso, sino simplemente presentar la forma en que se puede aunar la fuerza de JDBC y los servlets para poder acceder a una base de datos, introduciendo algunas características como la comprobación de autorización, etc...

Además, también se quiere almacenar las contraseñas para el acceso al sistema de artículos en una base de datos, en vez de en la configuración del servidor Web. Hay dos razones fundamentales para ello; por un lado, porque es previsible que mucha gente utilice este servicio, y por otro lado, que debe ser posible asociar cada envío con una persona en particular.

La inclusión en una base de datos ayudará a manejar gran cantidad de usuarios y además a controlar quién envía artículos. Una mejora que se puede hacer al sistema es el desarrollo de un applet JDBC que permita añadir y quitar usuarios, o asignar privilegios a quien haya enviado algún artículo al Sistema.

Las siguientes líneas de código muestran las sentencias utilizadas en la creación de las tablas e índices que se van a utilizar en la aplicación que se desarrollará para completar el Sistema de Artículos. El programa se ha pensado para atacar una base de datos Access, utilizando el puente JDBC-ODBC.

Tabla Usuarios

```
CREATE TABLE usuarios (  
    codusuario VARCHAR(16) NOT NULL,  
    nombre VARCHAR (60) NOT NULL,  
    empresa VARCHAR (60) NOT NULL,  
    admitirEnvio CHAR(1) NOT NULL,  
    clave VARCHAR (8) NOT NULL );  
CREATE INDEXcodusuario_key ON usuarios(codusuario) WITH PRIMARY;
```

DESARROLLO DE APLICACIONES WEB CON SERVLETS

USUARIOS

Nombre del campo	Tipo de datos		Descripción
CODUSUARIO	Texto	5	
NOMBRE	Texto	60	
EMPRESA	Texto	30	
ADMITIRENVIO	Texto	1	
CLAVE	Texto	8	

Propiedades del campo

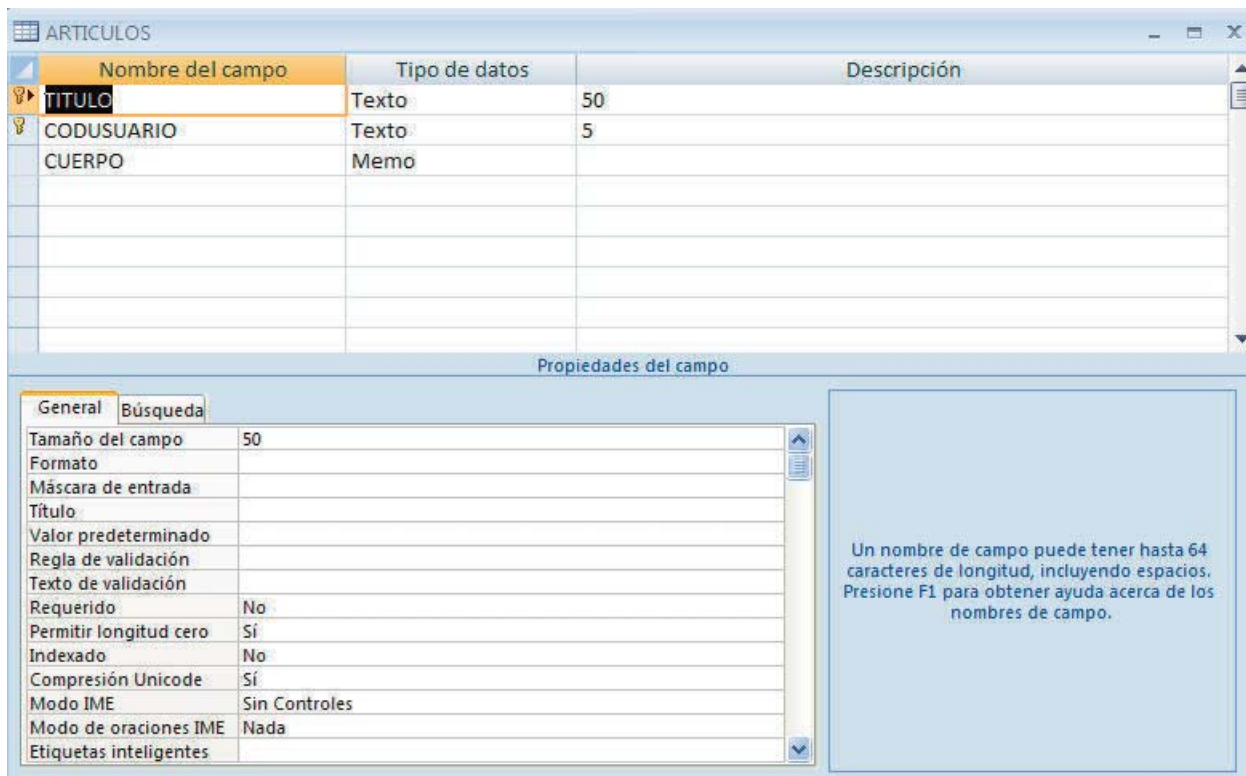
General		Búsqueda
Tamaño del campo	50	
Formato		
Máscara de entrada		
Título		
Valor predeterminado		
Regla de validación		
Texto de validación		
Requerido	No	
Permitir longitud cero	Sí	
Indexado	Sí (Sin duplicados)	
Compresión Unicode	Sí	
Modo IME	Sin Controles	
Modo de oraciones IME	Nada	
Etiquetas inteligentes		

Un nombre de campo puede tener hasta 64 caracteres de longitud, incluyendo espacios. Presione F1 para obtener ayuda acerca de los nombres de campo.

Tabla Artículos

```
CREATE TABLE articulos (
    titulo VARCHAR(255) NOT NULL,
    codusuario VARCHAR(16) NOT NULL,
    cuerpo memo );
CREATE INDEX articulo_key ON articulos(codusuario);
CREATE INDEX titulo_key ON articulos(titulo,codusuario);
```

DESARROLLO DE APLICACIONES WEB CON SERVLETS



Nombre del campo	Tipo de datos	Tamaño	Descripción
TITULO	Texto	50	
CODUSUARIO	Texto	5	
CUERPO	Memo		

Propiedades del campo

General	Búsqueda
Tamaño del campo	50
Formato	
Máscara de entrada	
Título	
Valor predeterminado	
Regla de validación	
Texto de validación	
Requerido	No
Permitir longitud cero	Sí
Indexado	No
Compresión Unicode	Sí
Modo IME	Sin Controles
Modo de oraciones IME	Nada
Etiquetas inteligentes	

Un nombre de campo puede tener hasta 64 caracteres de longitud, incluyendo espacios. Presione F1 para obtener ayuda acerca de los nombres de campo.

Como se puede observar, solamente hay dos tablas.

Si se quisiesen implementar las categorías, sería necesario incorporar una nueva tabla, o añadir un campo más a la tabla de artículos. La clave primaria de la tabla de usuarios es el identificador de cada usuario, y en la tabla de artículos hay un índice compuesto, formado por el título del artículo y el usuario que lo envió. Se usa un tipo MEMO (soportado por Access, que puede ser BLOB en otros motores), para guardar el cuerpo del artículo. Una mejora, en caso de convertirlo en un sistema de noticias, consistiría en añadir la fecha en que se ha enviado la noticia, como ya se ha comentado en un párrafo anterior, guardarla en un campo en la tabla de artículos y añadir el campo al índice compuesto de esa tabla, con lo cual se podrían presentar las noticias correspondientes a una fecha determinada sin aparente dificultad.

En la tabla de usuarios se guarda el identificador por el que el sistema va a reconocer al usuario, junto con la contraseña que elija para comprobar su identificación, más su nombre completo, la empresa a que pertenece y si ese usuario tiene autorización para el envío de artículos al Sistema. Si un usuario no dispone de autorización para el envío, solamente podrá acceder a la lectura de artículos. Y por supuesto, si alguien no aparece en esta tabla, no tendrá acceso alguno al Sistema.

DESARROLLO DE APLICACIONES WEB CON SERVLETS

Lo primero que hay que desarrollar es la página que va a dar acceso al sistema, para ver la forma de procesar los datos que van a llegar.



Control de Artículos

Introduce el nombre por el que te reconoce este sistema:

Introduce tu contraseña:

Selecciona la acción que quieres realizar:

Para el envío de artículos se requiere autorización especial

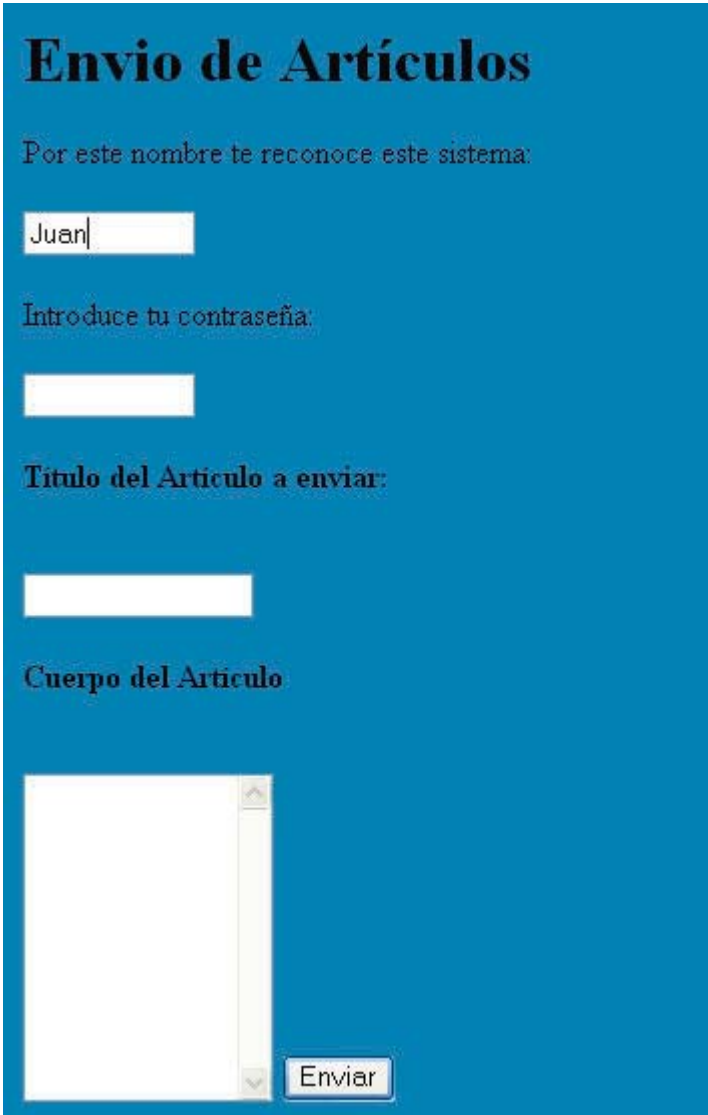
```
<html>
<head>
  <title>Control de Artículos</title>
</head>
<body bgcolor=gold>
  <form action="/examples/servlet/ServNotArt" method="get">
    <h1>Control de Artículos</h1>
    <p>Introduce el nombre por el que te reconoce este sistema:</p><br>
    <input type="text" size="5" maxlength="5" name="Usuario" id="Usuario"></input><br>
    <p>Introduce tu contraseña:</p><br>
    <input type="password" size="8" maxlength="8" name="Clave" id="Clave"></input><br>
    <p>Selecciona la acción que quieres realizar:
    <input type="submit" value="Leer_Articulos" name="boton">
    <input type="submit" value="Enviar_Articulos" name="boton"></p>
    <p><i>Para el envío de artículos se requiere autorización especial</i></p>
  </form>
</body>
</html>
```

DESARROLLO DE APLICACIONES WEB CON SERVLETS

Lo primero que se necesita ver es cómo se van a procesar los datos del formulario.

Si se quiere enviar un artículo, no es necesario introducir el nombre y clave en esta página, ya que el servlet enviará una nueva página para la introducción del contenido del artículo, y ya sobre esa sí que se establecen las comprobaciones de si el usuario está autorizado o no a enviar artículos al sistema.

La página que envía el servlet, es la que reproduce la imagen siguiente. Esta página se genera en el mismo momento en que el usuario solicita la inserción de un artículo. En caso de que haya introducido su identificación en la página anterior, el servlet la colocará en su lugar, si no, la dejará en blanco.



Envio de Artículos

Por este nombre te reconoce este sistema:

Juan|

Introduce tu contraseña:

Título del Artículo a enviar:

Cuerpo del Artículo

DESARROLLO DE APLICACIONES WEB CON SERVLETS

La imagen reproduce la página ya rellena, lista para la inserción de un nuevo artículo en el sistema. Cuando se pulsa el botón de envío de artículo y el servlet recibe la petición de inserción del artículo en el sistema, es cuando realiza la comprobación de autorización, por un lado de si es un usuario reconocido para el sistema y, en caso afirmativo, si está autorizado el envío de artículo, o solamente puede leerlos.

Debe recordarse que los nombres de los campos de entrada de datos se pueden llamar con *getParameter()* y recoger la información que contienen. Los parámetros *nombre* y *clave* en el formulario se mapean en las variables *usuario* y *clave* en el servlet, que serán las que se utilicen para realizar las comprobaciones de acceso del usuario.

El código completo del servlet es el siguiente:

```
import java.io.*;
import java.net.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ServNotArt extends HttpServlet {

    String DBurl = "jdbc:odbc:articulos";
    String usuarioGet = "";
    String usuarioPost = "";
    String claveGet = "";
    String clavePost = "";
    Connection con;
    DatabaseMetaData metaData;

    // Este método es el que se encarga de establecer e inicializar
    // la conexión con la base de datos
    public void init( ServletConfig conf ) throws ServletException {
        SQLWarning w;

        super.init( conf );
        try {
            Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
```



DESARROLLO DE APLICACIONES WEB CON SERVLETS

```

con = DriverManager.getConnection( DBurl,usuarioPost,clavePost );
if( (w = con.getWarnings()) != null ) {
    while( w != null ) {
        log( "SQLWarning: "+w.getSQLState()+"\t'+
            w.getMessage()+"\t'+w.getErrorCode()+"\t' );
        w = w.getNextWarning();
    }
}
} catch( ClassNotFoundException e ) {
    throw new ServletException( "init" );
} catch( SQLException e ) {
    try {
        con = DriverManager.getConnection( DBurl,usuarioGet,claveGet );
    } catch( SQLException ee ) {
        ee.printStackTrace();
        while( e != null ) {
            log( "SQLException: "+e.getSQLState()+"\t'+
                e.getMessage()+"\t'+e.getErrorCode() );
            e = e.getNextException();
        }
        throw new ServletException( "init" );
    }
}
}
}

```

```

public void service( HttpServletRequest req,HttpServletResponse res )
    throws ServletException,IOException {
    String usuario = req.getParameter( "usuario" );
    String autorizado;
    String accion = req.getParameter( "accion" );

    try {
        autorizado = autorizacion( req );
        if( accion.equals( "Leer Articulos" )
            && !autorizado.equals( "ACCESO DENEGADO" ) ) {
            leerArticulo( req,res );
        } else if( accion.equals( "Enviar" )

```



DESARROLLO DE APLICACIONES WEB CON SERVLETS

```

&& autorizado.equals( "POST" ) ) {
    enviarArticulo( req,res );
} else if( accion.equals("Envío de Artículos" ) ) {
    if( usuario == null )
        usuario = " ";
    PrintWriter out = new PrintWriter( res.getOutputStream() );
    out.println( "<HTML>" );
    out.println( "<HEAD><TITLE>Envío de Artículos</TITLE></HEAD>" );
    out.println( "<BODY>" );
    out.println( "<CENTER><H1>Envío de Artículos</H2></CENTER>" );
    out.println( "<FORM ACTION=/examples/servlet/ServNotArt " );
    out.println( "METHOD=POST>" );
    out.println( "<H2>Envío de Artículos</H2>" );
    out.println( "<P>Por este nombre te reconoce el Sistema: <BR>" );
    out.println( "<INPUT NAME=usuario TYPE=text VALUE='"+usuario+"' " );
    out.println( "SIZE=16 MAXLENGTH=16><BR>" );
    out.println( "Introduce tu contraseña: <BR>" );
    out.println( "<INPUT NAME=clave TYPE=password SIZE=8 MAXLENGTH=8>" );
    out.println( "</P>" );
    out.println( "<H3>Título del Artículo a enviar:</H3>" );
    out.println( "<P><INPUT NAME=titulo TYPE=text SIZE=25" );
    out.println( "MAXLENGTH=50></P>" );
    out.println( "<H3>Cuerpo del Artículo</H3>" );
    out.println( "<P><TEXTAREA NAME=cuerpo ROWS=10 COLS=50>" );
    out.println( "</TEXTAREA></P>" );
    out.println( "<P><INPUT NAME=accion TYPE=submit VALUE='Enviar'>" );
    out.println( "</FORM>" );
    out.println( "</BODY></HTML>" );
    out.flush();
} else {
    PrintWriter out = new PrintWriter( res.getOutputStream() );
    out.println( "<html>" );
    out.println( "<head><title>Acceso Denegado</title></head>" );
    out.println( "<body>" );
    out.println( "Se ha producido un error de acceso:<br>" );
    out.println( "El usuario o clave que has introducido no " );
    out.println( "son válidos o<br>" );

```



DESARROLLO DE APLICACIONES WEB CON SERVLETS

```
        out.println( "no tienes acceso a esta funcionalidad." );
        out.println( "</body></html>" );
        out.flush();
    }
} catch( SQLException e ) {
    while( e != null ) {
        log( "SQLException: "+e.getSQLState()+"\t"+
            e.getMessage()+"\t"+e.getErrorCode()+"\t" );
        e = e.getNextException();
    }
    // Aquí habría que insertar el código necesario para restablecer la
    // conexión llamando a init() de nuevo y volviendo a realizar la
    // llamada al método service(req,res)
}
}

// Se cierra la conexión con la base de datos
public void destroy() {
    try {
        con.close();
    } catch( SQLException e ) {
        while( e != null ) {
            log( "SQLException: "+e.getSQLState()+"\t"+
                e.getMessage()+"\t"+e.getErrorCode()+"\t" );
            e = e.getNextException();
        }
    } catch( Exception e ) {
        e.printStackTrace();
    }
}

// Este método ejecuta la consulata a la base de datos y devuelve el
// resultado, para formatear la salida y presentar el resultado de
// la consulta de artículos al usuario
public void leerArticulo( HttpServletRequest req,HttpServletResponse res )
    throws IOException,SQLException {
    Statement stmt = con.createStatement();
```



DESARROLLO DE APLICACIONES WEB CON SERVLETS

```
String consulta;
ResultSet rs;

res.setStatus( res.SC_OK );
res.setContentType( "text/html" );
consulta = "SELECT articulos.cuerpo, articulos.titulo, " );
consulta += articulos.usuario, usuarios.nombre, usuarios.empresa " ;
consulta += "FROM articulos, usuarios WHERE " );
consulta += articulos.usuario=usuarios.usuario";
rs = stmt.executeQuery( consulta );

PrintWriter out = new PrintWriter( res.getOutputStream() );
out.println( "<HTML>" );
out.println( "<HEAD><TITLE>Artículos Enviados</TITLE></HEAD>" );
out.println( "<BODY>" );

while( rs.next() ) {
    out.println( "<H2>" );
    out.println( rs.getString(1) );
    out.println( "</H2><p>" );
    out.println( "<|>Enviado desde: "+rs.getString(5)+"</|><BR> " );
    out.println( "<B>"+rs.getString(2)+"</B>, por "+rs.getString(4) );
    out.println( "<HR>" );
}
out.println( "</BODY></HTML>" );
out.flush();
rs.close();
stmt.close();
}

public void enviarArticulo( HttpServletRequest req, HttpServletResponse res )
    throws IOException, SQLException {
    Statement stmt = con.createStatement();
    String consulta = "";
    String usuario = req.getParameter( "usuario" );

    PrintWriter out = new PrintWriter( res.getOutputStream() );
```

DESARROLLO DE APLICACIONES WEB CON SERVLETS

```
res.setStatus( res.SC_OK );
res.setContentType( "text/html" );
out.println( "<HTML>" );
out.println( "<HEAD><TITLE>Envío Realizado</TITLE></HEAD>" );
out.println( "<BODY>" );

consulta = "INSERT INTO articulos VALUES( ";
consulta += req.getParameter( "titulo" )+"', '"+usuario+"', ";
consulta += req.getParameter("cuerpo")+"')";
int result = stmt.executeUpdate( consulta );

if( result != 0 ) {
    out.println( "Tu artículo ha sido aceptado e insertado" );
    out.println( " correctamente." );
} else {
    out.println( "Se ha producido un error en la aceptación de tu " );
    out.println( " artículo.<BR>" );
    out.println( "Contacta con el Administrador de la base de datos, " );
    out.println( "o consulta<BR>" );
    out.println( "el fichero <!--log--> del servlet." );
}
out.println( "</BODY></HTML>" );
out.flush();
stmt.close();
}

// Devuelve la información del Servlet
public String getServletInfo() {
    return "Servlet JDBC (Tutorial de Java), 1998";
}

public String autorizacion( HttpServletRequest req ) throws SQLException {
    Statement stmt = con.createStatement();
    String consulta;
    ResultSet rs;
    String valido = "";
    String usuario = req.getParameter( "usuario" );
```



DESARROLLO DE APLICACIONES WEB CON SERVLETS

```
String clave = req.getParameter( "clave" );
String permiso="";

consulta = "SELECT admitirEnvio FROM usuarios WHERE usuario = '"+usuario;
consulta += "' AND clave = '"+clave+"'";
rs = stmt.executeQuery( consulta );

while( rs.next() ) {
    valido = rs.getString(1);
}
rs.close();
stmt.close();

if( valido.equals( "" ) ) {
    permiso = "ACCESO DENEGADO";
} else {
    // Permiso sólo para lectura de artículos
    if( valido.equals( "N" ) ) {
        permiso = "GET";
    // Permiso para lectura y envío de artículos
    } else if( valido.equals( "S" ) ) {
        permiso = "POST";
    }
}
return permiso;
}
```