Los primeros capítulos del libro trabajan con SQL plano y  simplemente estan orientados a ir asentando ideas sobre los principios de ingenieria de software que se tratan .Ahora bien ha habido gente que me ha comentado que debía haber sido mas explicito y comentar que el uso de SQL plano es una mala práctica. Así pues he decidido añadir
unos artículos adicionales al blog que cubran este tema. Vamos a ver el siguiente programa sencillo .

```
package com.arquitecturajava;

//omitimos imports

public class PrincipalSentencia {

public static void main(String[] args) {

boolean valido = validarUsuario("usuario1", "clave1");

if (valido) {

System.out.println("el acceso es correcto");
 }
 }

public static boolean validarUsuario(String nombre, String clave) {

Connection conexion = null;
 Statement sentencia = null;
 ResultSet rs = null;
 boolean valido = false;
```

```
 try {
 Class.forName("com.mysql.jdbc.Driver");
 conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/arquitecturaj
ava","root","blog");
 sentencia = conexion.createStatement();
 String consulta = "select nombre from Usuarios where nombre='"
 + nombre + "' and clave='" + clave + "'";
 rs = sentencia.executeQuery(consulta);
 if (rs.next())
 valido = true;

rs.close();
 rs = null;
 sentencia.close();
 sentencia = null;
 conexion.close();
 conexion = null;

} catch (ClassNotFoundException e) {

e.printStackTrace();
 } catch (SQLException e) {

e.printStackTrace();
 } finally {

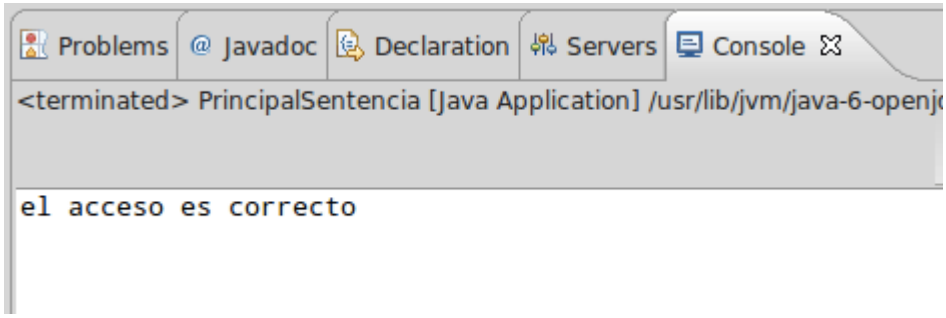
if (rs != null) {
 try {
 rs.close();
 } catch (SQLException e) {
```

```
 //log
 }
 rs = null;
 }
 if (sentencia != null) {
 try {
 sentencia.close();
 } catch (SQLException e) {
 //log
 }
 sentencia = null;
 }
 if (conexion != null) {
 try {
 conexion.close();
 } catch (SQLException e) {
 //log
 }
 conexion = null;
 }
 }

return valido;

}
}
```

Este programa se encarga de validar a un usuario a traves del método validarUsuario (String,String).  Si el usuario es válido el método devuelve “true” y el programa imprime por pantalla “el acceso es correcto”.




El método main unicamente tiene el siguiente bloque de código que invoca al método de validarUsuario.

```
boolean valido = validarUsuario("usuario1", "clave1");

if (valido) {

System.out.println("el acceso es correcto");
 }
```

El método validar construye la siguiente SQL y la ejecuta

select nombre from Usuarios where nombre=’usuario1′ and clave=’clave1′

Si la consulta devuelve algún registro el método devuelve “true” y  el mensaje de “acceso válido” aparecera por pantalla. Ahora bien si en vez de pasar como clave “clave1” pasáramos lo siguiente:

```
boolean valido = validarUsuario("usuario1", "' or 0=0 #");

if (valido) {
```

```
System.out.println("el acceso es correcto");
 }
```

Curiosamente nuestro programa también nos permitiría el acceso ya que la consulta construida sería la siguiente.
select nombre from Usuarios where nombre=’usuario1′ and clave=” or 0=0 #’

En donde 0=0 siempre se cumple como condición y por lo tanto siempre nos dejara acceder. Este es un ejemplo sencillo de como el código que hemos construido es claramente malo ya que permite inyectar código en las consultas SQL modificando sus propiedades . En posteriores post veremos como solventar este problema