

Para eliminar el problema de la inyección de SQL a nivel de JDBC debemos de hacer uso de consultas parametrizadas .Estas consultas no concatenan las variables a la consulta SQL sino que usan una sintaxis específica para pasar a la consulta SQL un conjunto de parámetros predeterminado. Vamos a ver a continuación una parte del código del post anterior modificado.

```
Connection conexion = null;
PreparedStatement sentencia = null;
ResultSet rs = null;
boolean valido = false;
try {
    Class.forName("com.mysql.jdbc.Driver");
    conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/arquitecturajava", "root", "blog");
    String consulta = "select nombre from Usuarios where nombre =? and
clave =?";
    sentencia = conexion.prepareStatement(consulta);
    sentencia.setString(1, nombre);
    sentencia.setString(2, clave);
    rs = sentencia.executeQuery();
    if (rs.next())
        valido = true;
```

Como podemos ver la consulta se construye de otra forma utilizando el carácter de interrogación "?" para definir la posición de cada uno de los parámetros.

```
String consulta = "select nombre from Usuarios where nombre =? and clave =?";
```

El valor de los parámetros será pasado mas adelante a traves del siguiente bloque de código

```
sentencia = conexion.prepareStatement(consulta);  
sentencia.setString(1, nombre);  
sentencia.setString(2, clave);
```

De esta forma la consulta es primero precompilada para mas adelante recibir los valores de los parámetros pero sin posibilidad de que la consulta (tablas ,indices y joins) sea modificada. Asi pues si intentamos ejecutar otra vez el programa MAIN intentando inyectar SQL como se muestra a continuación.

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

```
if (valido) {
```

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

```
}
```

Simplemente no recibiremos ningún resultado ya que no hay ningún usuario con la clave ="or 0=0" y el ResultSet de retorno vendra vacío.De esta manera hemos evitado inyección de SQL utilizando consultas parametrizadas.