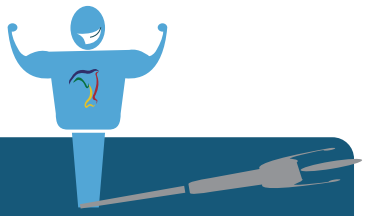
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

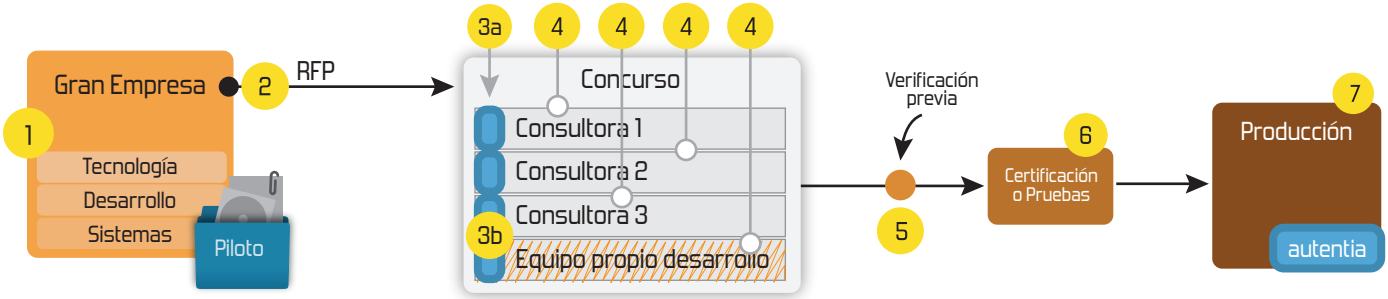
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Hosting Patrocinado por enREDados.com

Tutorial desarrollado por: Germán Jiménez Centeno

Puedes encontrarme en Autentia
Somos expertos en Java/J2EE
Contacta en info@autentia.com

autentia real business solutions

Descargar este documento en formato PDF datasrcSqlWebSph.pdf

Firma en nuestro libro de Visitas

Curso Web J2EE
Curso Avanzado en Desarrollo Web con J2EE

Mainframe Adapters
Use ODBC, JDBC, ADO, JCA, and .NET Access CICS, IMS, VSAM, Adabas, DB2

UML Modeling Tools
Superior Usability & Easy To Learn UML Data Modeling. Try It Free Now!

Master Java J2ee Oracle
Prácticas laborales 100% aseguradas Nuevo temario de Struts. Trabaja ya

Anuncios Goooooogle Anunciarse en este sitio

En Autentia trabajamos constantemente en el desarrollo de aplicaciones web para nuestros clientes. En esta línea, resulta fundamental el manejo y dominio de los distintos servidores para aplicaciones web disponibles que ofrecen mayor fiabilidad. Hoy os queremos mostrar cómo configurar un DataSource de SQLServer en WebSphere.

Índice de contenidos

1. Crear el datasource 1

1.1. Crear el usuario J2C 1

1.2. Crear el proveedor de JDBC 5

1.3. Creación del datasource 9

2. Conectarse desde java al datasource de SQL Server 19

1. Crear el datasource

En primer lugar hemos de abrir la consola de administración de WebSphere Express (http://<DIRECCION_SERVIDOR_WEBSPHERE>:7090/admin).

1.1. Crear el usuario J2C

Para comenzar hay que crear un usuario J2C en la consola de administración de WebSphere.

Un usuario J2C (Java 2 Connector) es un usuario que tiene permisos para hacer referencia a entradas de datos de autenticación de adaptadores de recursos, orígenes de datos (datasources) y otras configuraciones que requieran datos de autenticación.

Como necesitamos permisos para poder acceder a una base de datos mediante un datasource, nos dirgimos a Seguridad->Configuración de JAAS->Datos de autenticación de J2C:

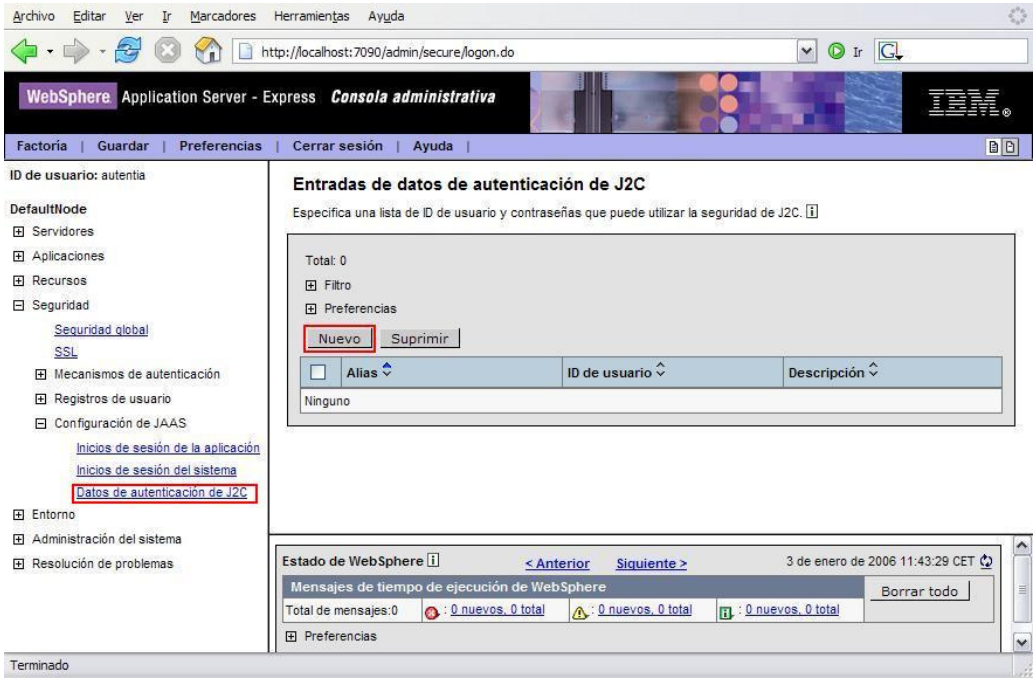


Imagen 1: insertar usuario J2C

Y pulsamos sobre el botón 'nuevo'.

En este punto tendremos una pantalla como la siguiente:

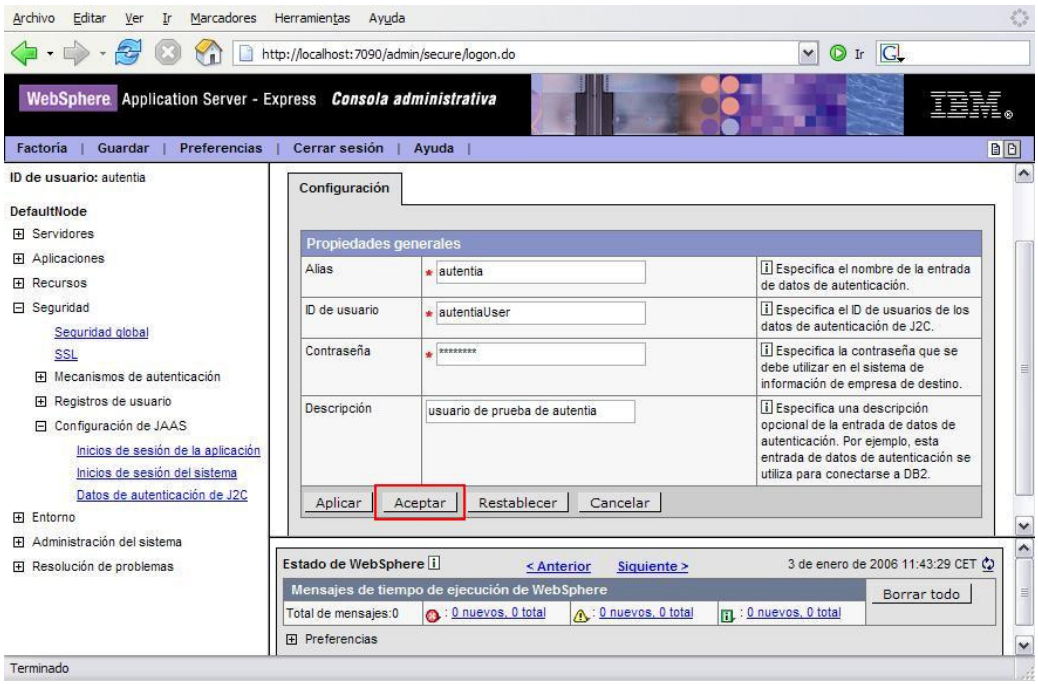


Imagen 2: insertar usuario J2C

Insertamos los datos necesarios:

- Alias: es el nombre que se le da al usuario dentro del contexto de websphere. Este nombre debe ser único para cada usuario insertado.
- ID de usuario: aquí se debe introducir el id de usuario para la base de datos a la que queremos acceder. Nosotros tenemos en nuestro servidor de Sql Server un usuario llamado AutentiaUser.
- Contraseña: Contraseña del usuario de la base de datos.
- Descripción: Si se desea se puede introducir una descripción para este usuario

Tras introducir los datos pulsamos en aceptar. Veremos una pantalla como la siguiente, donde se muestra el usuario recientemente creado:

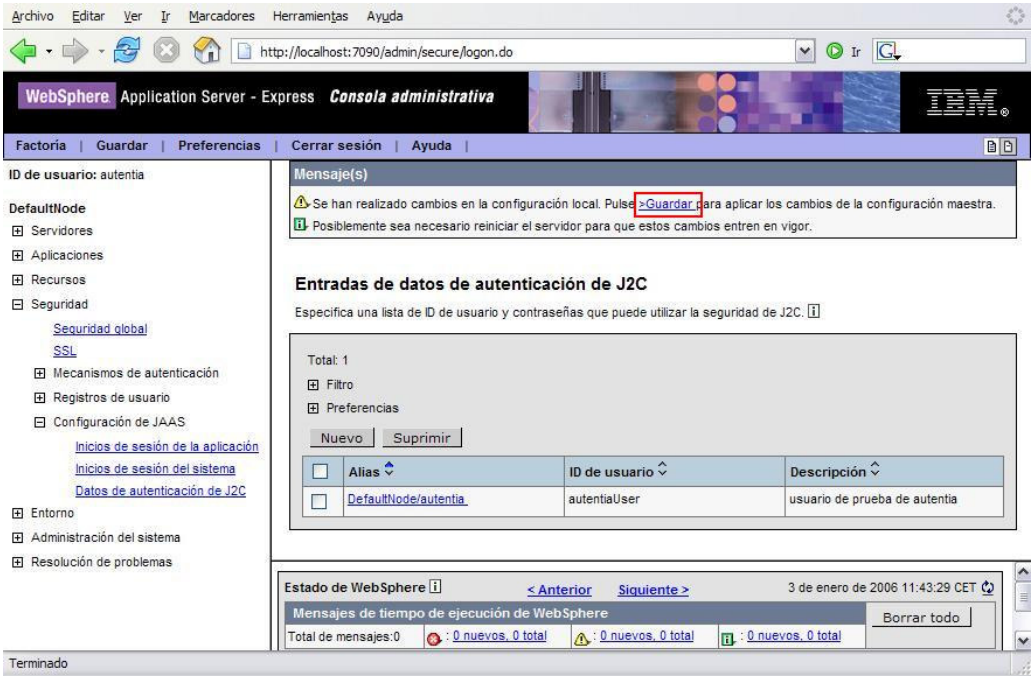


Imagen 3: insertar usuario J2C

Si nos fijamos bien, en la parte superior de la ventana aparece un mensaje avisándonos de que deberemos guardar los cambios. Pulsamos sobre guardar (recuadrado en rojo), y aparece un mensaje de confirmación. Volvemos a pulsar guardar:

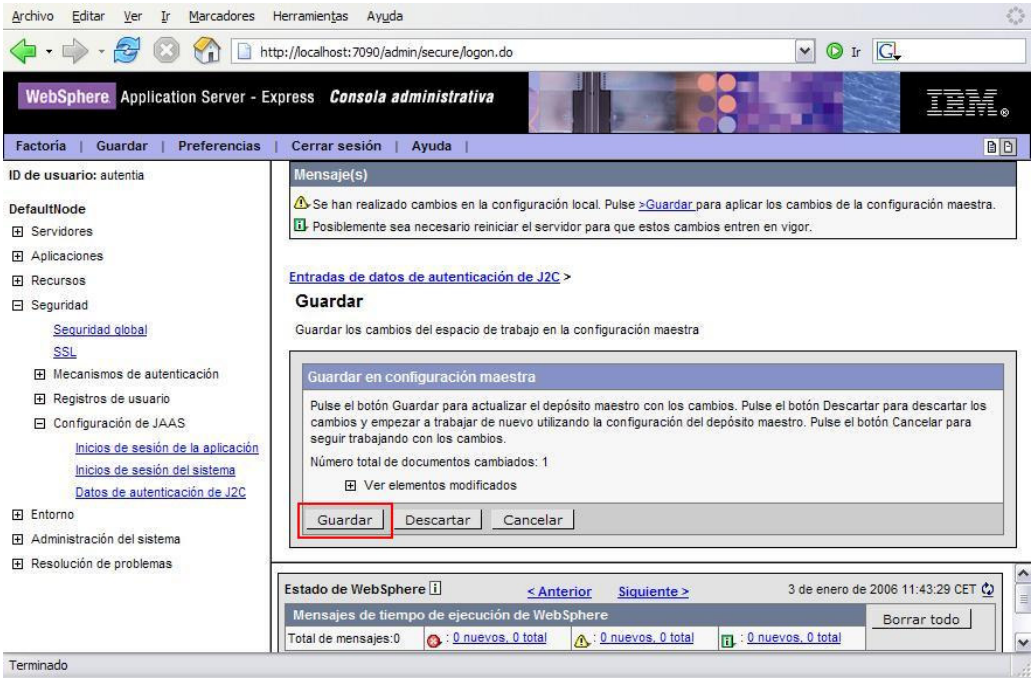


Imagen 4: insertar usuario J2C

Tras realizar los pasos anteriores ya tenemos creado un usuario que va a poder acceder a la base de datos. Ahora vamos a proceder a crear un Datasource en el servidor, al que luego accederemos desde cualquier aplicación.

1.2. Crear el proveedor de JDBC

El proveedor es simplemente el driver que vamos a usar para conectarnos con la base de datos.

Una vez creado el proveedor, es decir, una vez dado de alta el driver, podremos crear tantos datasources (fuentes de datos) que usen ese proveedor de JDBC, como queramos.

Para ello pulsamos sobre el enlace Proveedores de JDBC (Recursos->Proveedores de JDBC), y llegaremos a la siguiente pantalla:

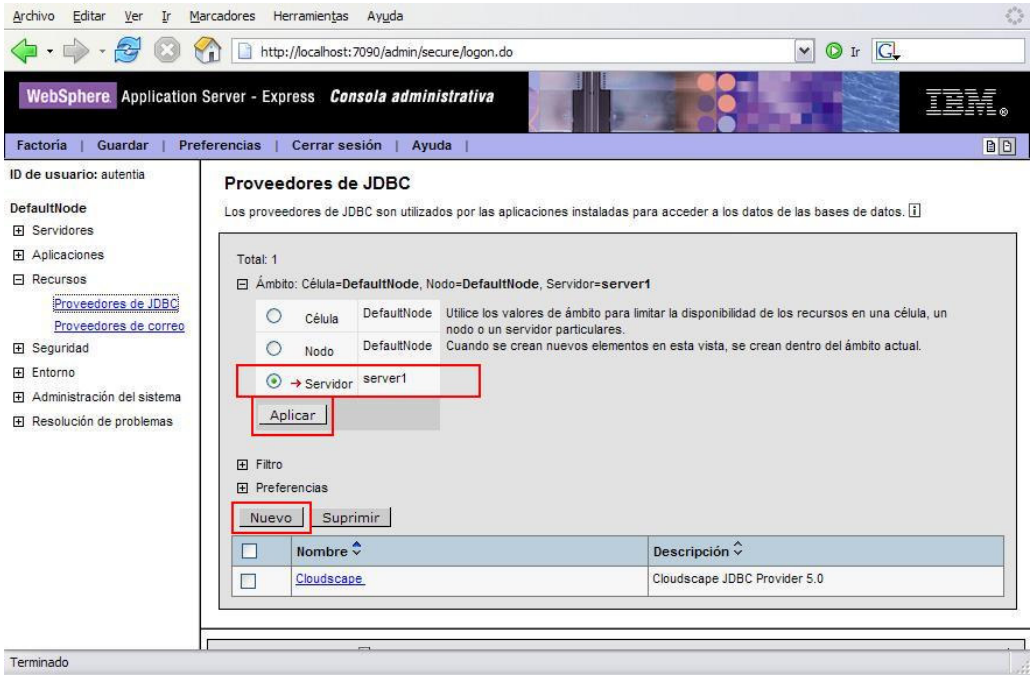


Imagen 5: crear un datasource

Se puede apreciar que se puede elegir en que ámbito crear el datasource. Para este documento se ha realizado sobre el ámbito 'servidor', pero se puede también hacer sobre el ámbito 'Nodo'.

Pulsamos entonces sobre el botón nuevo que aparece un poco más abajo que nos llevara al siguiente menú:

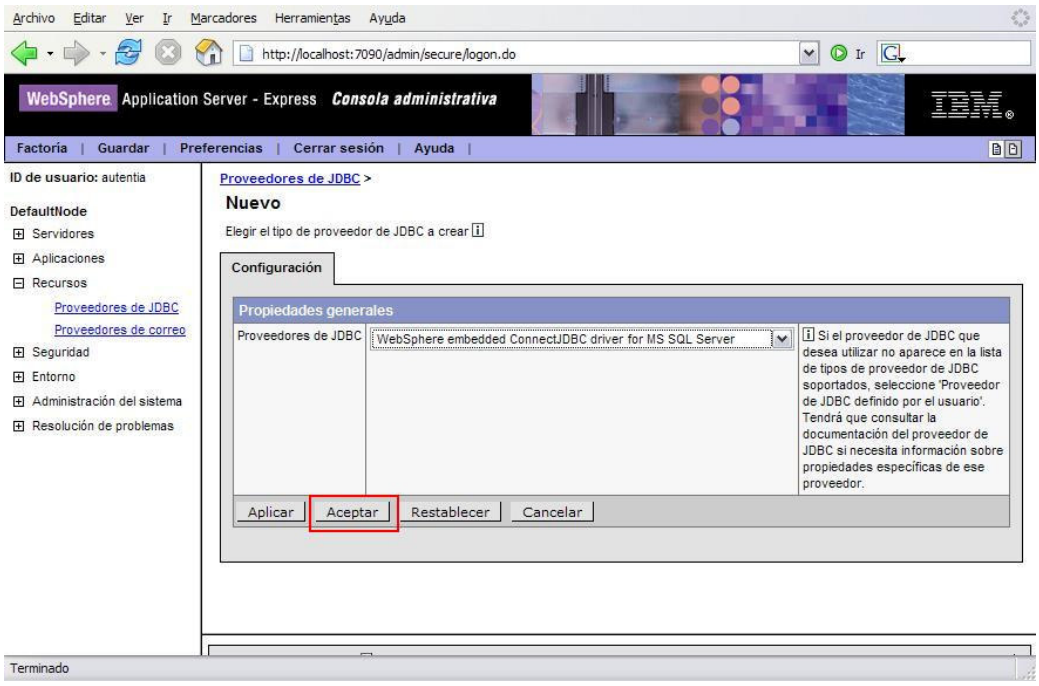


Imagen 6: crear un datasource

Desde aquí seleccionamos el proveedor del driver JDBC que se va a utilizar. Como el servidor al que nos queremos conectar es SQL Server elegimos el penúltimo driver que aparece en la lista desplegable (*WebSphere embedded ConnectJDBC driver for MS SQL Server*).

Si nos fijamos en la lista desplegable podemos observar que existen 2 drivers con este nombre; uno terminado en (XA) y otro sin esta terminación. Los drivers jdbc XA son los que se utilizan para realizar transacciones distribuidas. Nosotros usaremos aquí la versión que no utiliza transacciones distribuidas (aquella que no lleva XA en su nombre).

Pulsamos sobre aceptar y llegamos a la próxima pantalla:

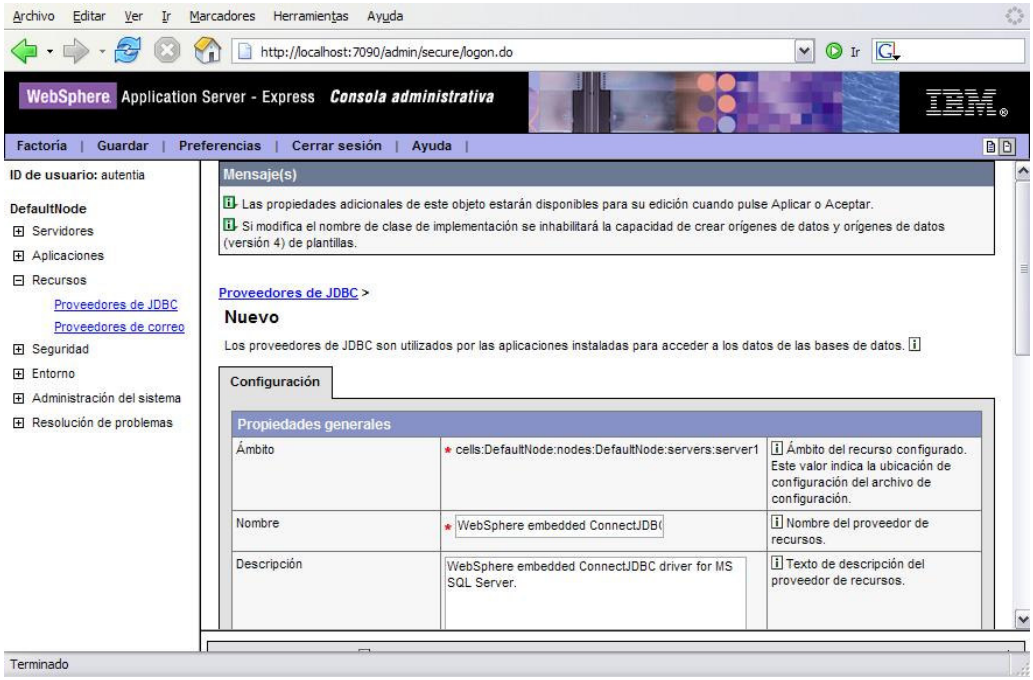


Imagen 7: crear un datasource

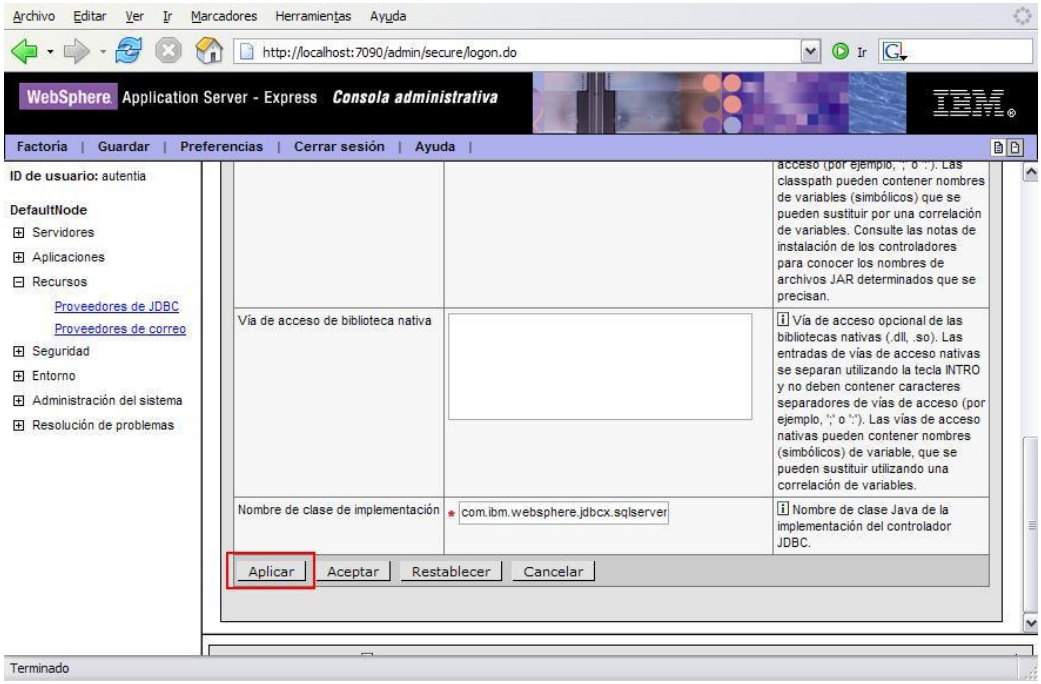


Imagen 8: crear un datasource

En este punto no es necesario cambiar nada, y podemos pulsar sobre Aplicar.

1.3. Creación del datasource

Ahora en la parte inferior de la pantalla han aparecido más opciones como muestra la siguiente captura:

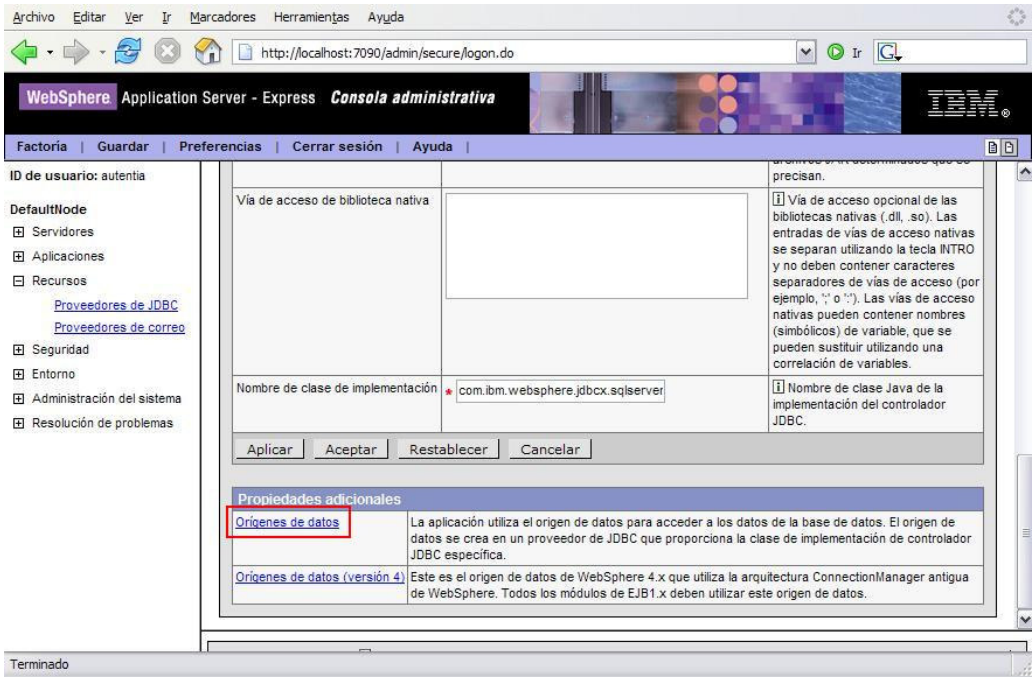


Imagen 9: crear un datasource

En recuadro rojo esta resaltado el link sobre el que deberemos pulsar ahora para definir el datasource en sí:

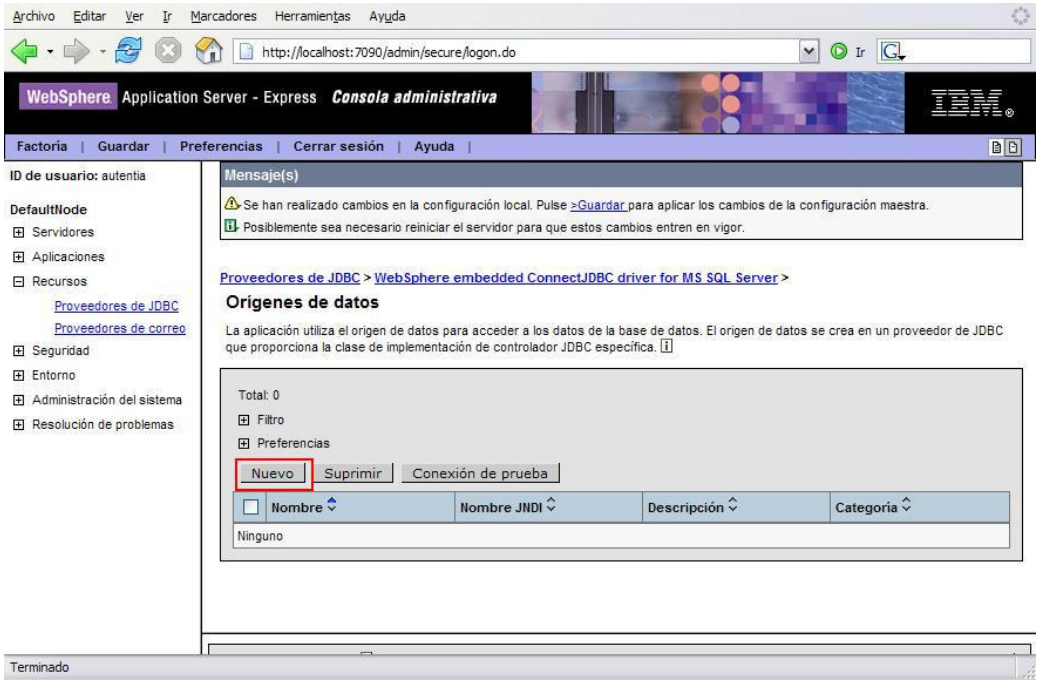


Imagen 10: crear un datasource

Pulsamos sobre nuevo e introducimos los siguientes datos:

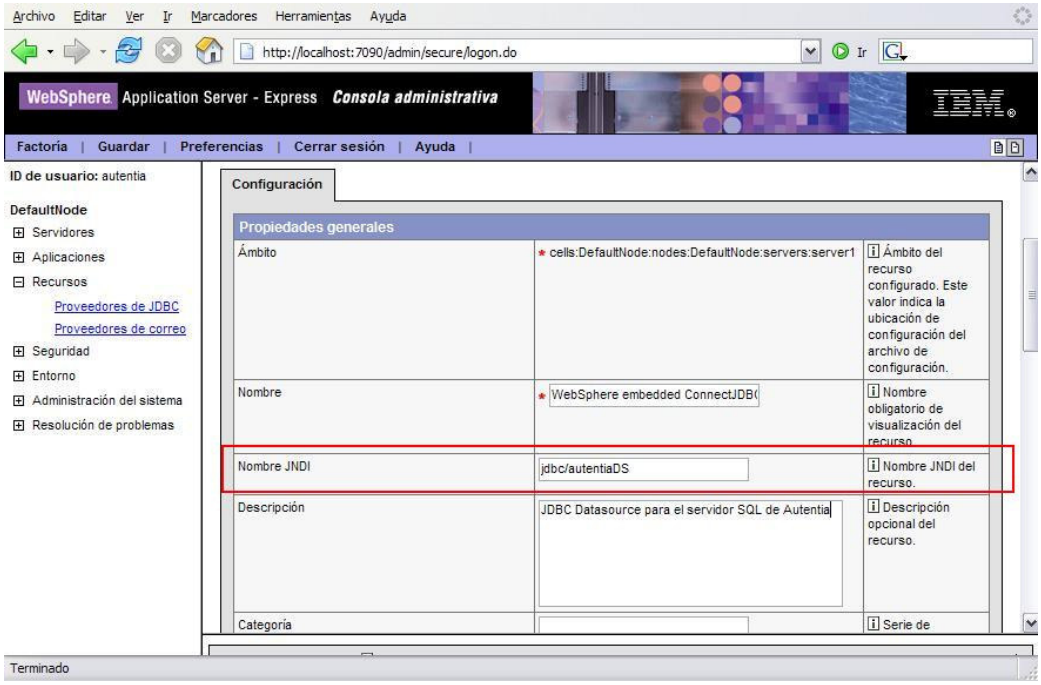


Imagen 11: crear un datasource

Dentro del dato recuadrado en rojo (Nombre JNDI) insertamos el nombre con el cual más tarde haremos referencia en la aplicación mediante JNDI. Nosotros lo hemos llamado 'jdbc/autentiaDS'.

Si seguimos haciendo scroll hacia abajo (pues hay más opciones que no caben en la captura) insertaremos más datos que se explican a continuación:

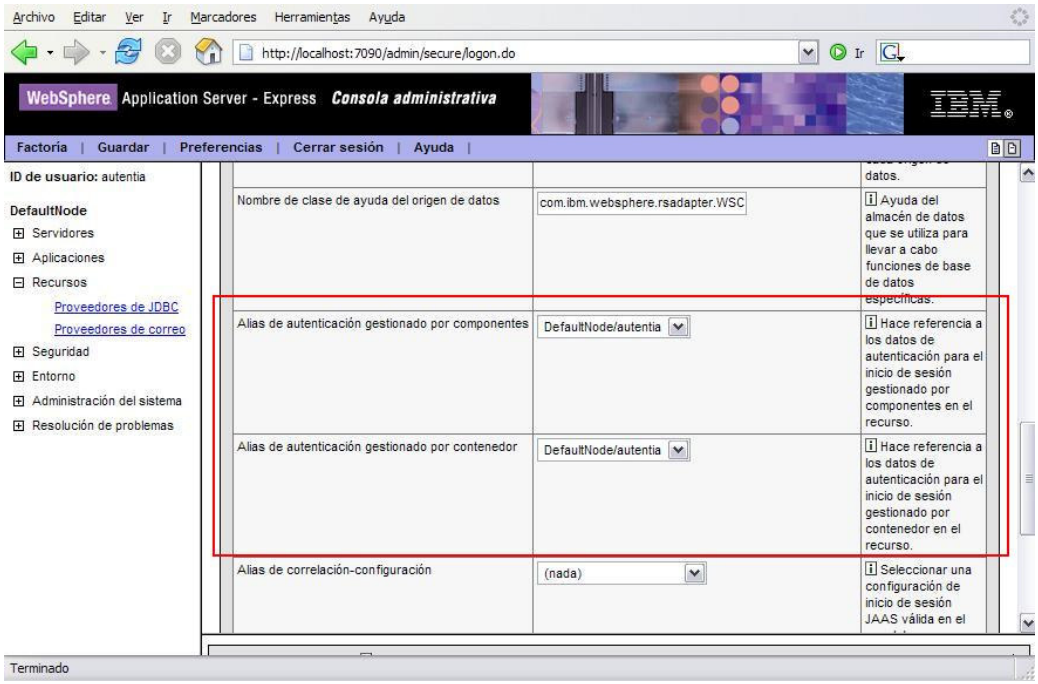


Imagen 12: crear un datasource

Recuadrados en rojo hay 2 listas desplegables cuyos valores iniciales son '(nada)' ('Alias de autenticación gestionada por componentes' y 'Alias de autenticación gestionado por contenedor'). Desplegamos las listas y seleccionamos el alias del usuario J2C que creamos al principio del documento, y pulsamos aplicar:

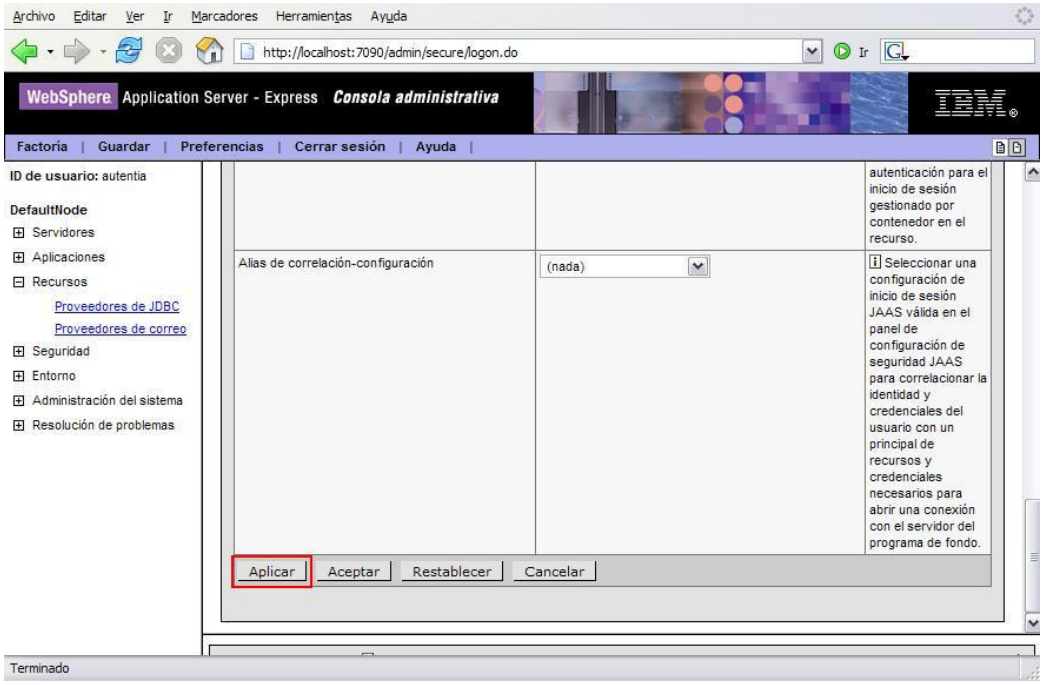


Imagen 13: crear un datasource

Tras pulsar aplicar, en la parte inferior de la pantalla vuelven a aparecer 2 opciones más: pinchamos sobre 'propiedades personalizadas' y continuamos.

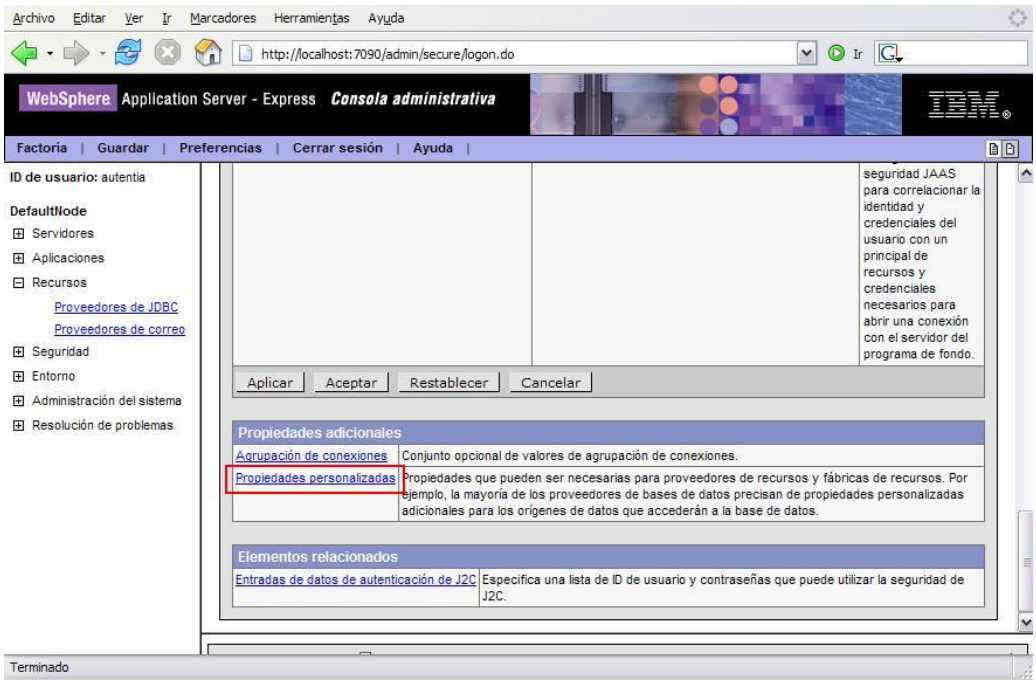


Imagen 14: crear un datasource

Ahora hemos de rellenar los datos referentes a la conexión con la base de datos, como son el nombre de la misma, la dirección del servidor.. Más abajo se explica cada opción con más detalle:

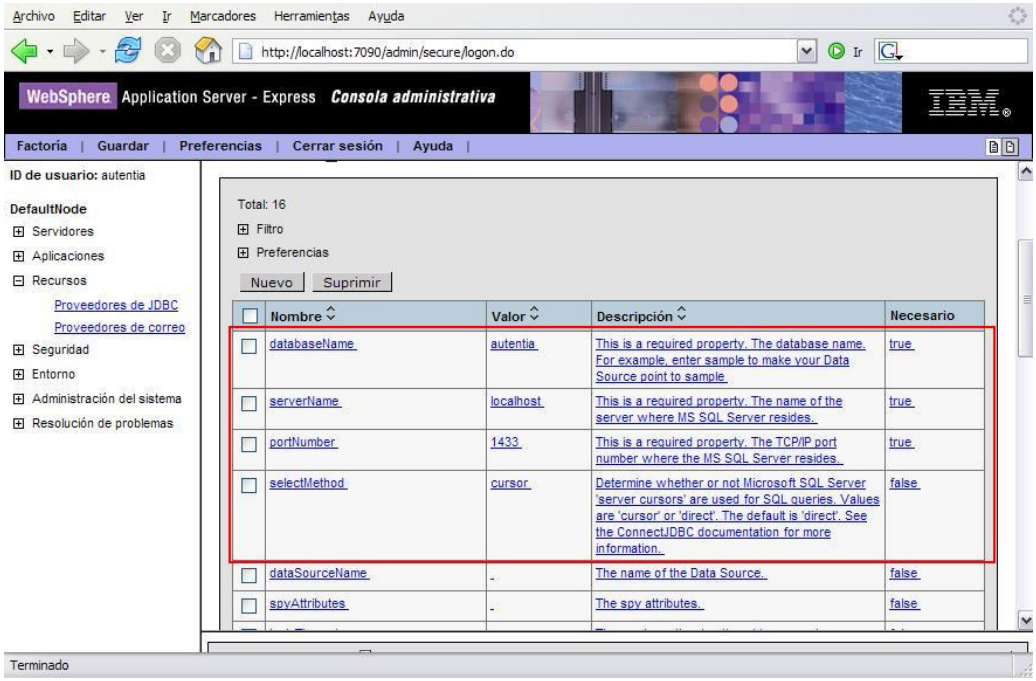


Imagen 15: crear un datasource

En la captura anterior vemos varios parámetros rodeados por un cuadrado rojo, que son aquellos que se han rellenado. El resto pueden quedarse vacíos (son opcionales). Los parámetros rellenados son:

- **databaseName**: nombre de la base de datos a la cual nos conectaremos con este datasource.
- **serverName**: Dirección o nombre del servidor. Para el ejemplo es "localhost" puesto que el servidor de SQL Server está en la misma máquina que el WebSphere. También puede introducirse la dirección IP del servidor, su URL (www.unadireccioncualquiera.com), o el nombre de la máquina para redes Windows (p. ej. SERVIDOR, SQLSERVER, AUTENTIASERVER,...). Consultar al administrador de la red para más información acerca de este dato.
- **portNumber**: por defecto ya viene el 1433, que es el puerto de escucha de SQL Server por defecto, pero habrá que introducir otro si así fuera necesario.
- **selectMethod**: se ha utilizado para este documento el método 'cursor'. También es posible establecer 'direct' (ver opciones de SQL Server).

Si es necesario rellenar más parámetros debe hacerse desde aquí, y en caso de necesitar algún parámetro que no esté listado en esta sección, se pueden añadir más mediante el botón 'Nuevo', que esta en la parte superior.

Y tras finalizar de insertar los parámetros necesarios, vamos a la parte superior de la ventana y pulsamos el link 'Guardar':

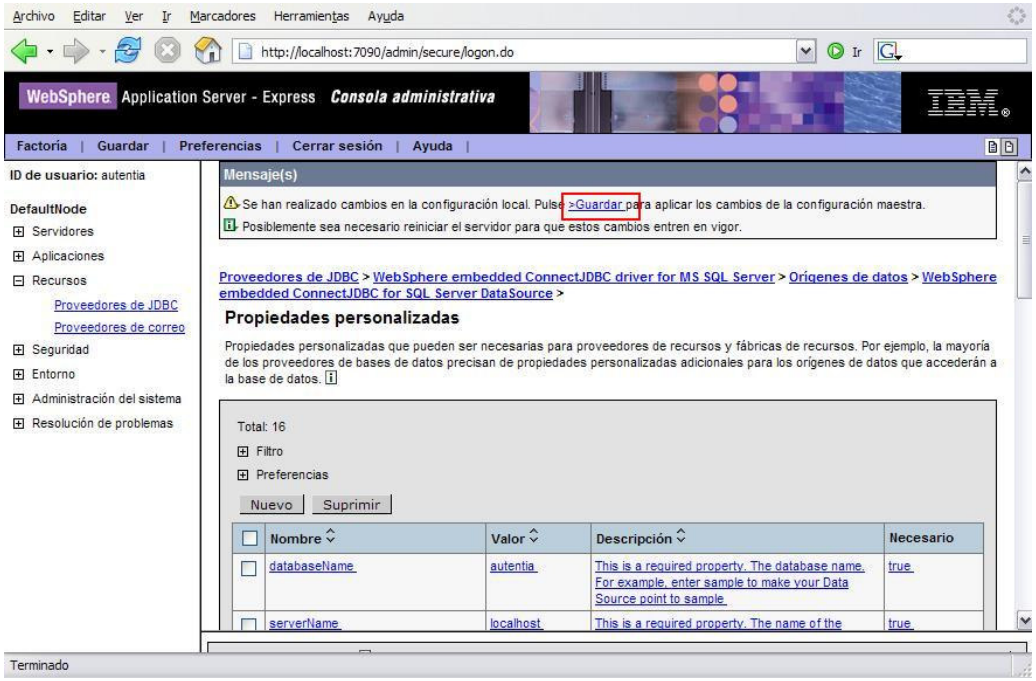


Imagen 16: crear un datasource

Seguidamente otra pantalla nos pide confirmación al guardar los cambios. Pulsamos en guardar de nuevo:

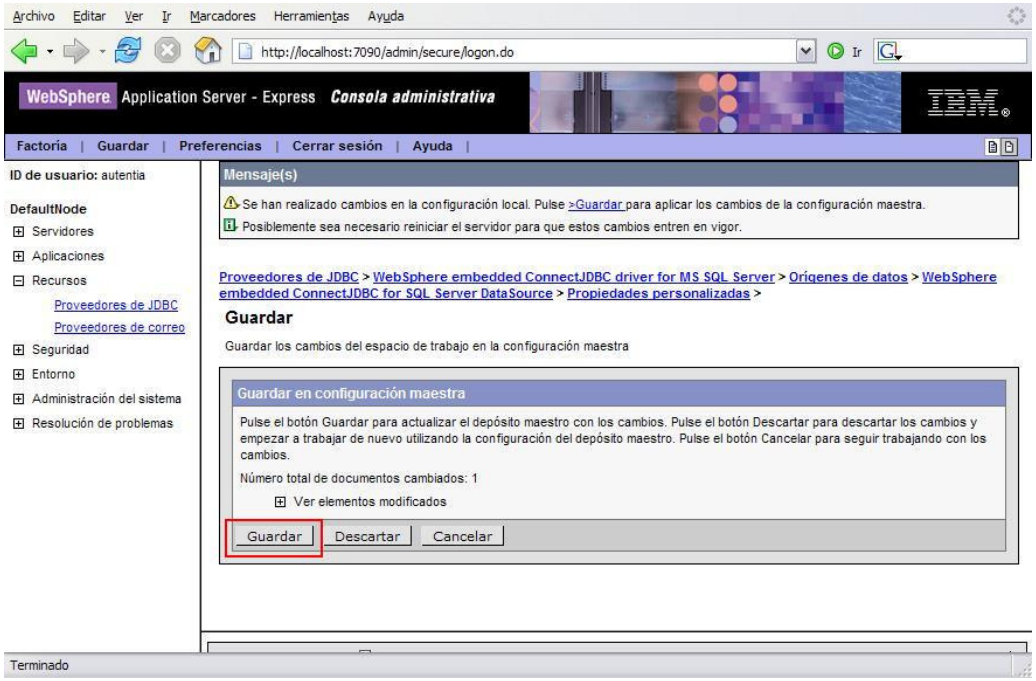


Imagen 17: crear un datasource

En este punto ya tenemos creado el datasource. Podemos hacer una prueba del mismo yendo a 'Proveedores de JDBC -> WebSphere embedded ConnectJDBC driver for MS SQL Server'. Veremos el datasource que acabamos de crear. Lo seleccionamos mediante el checkbox, y pulsamos sobre el botón 'conexión de prueba':

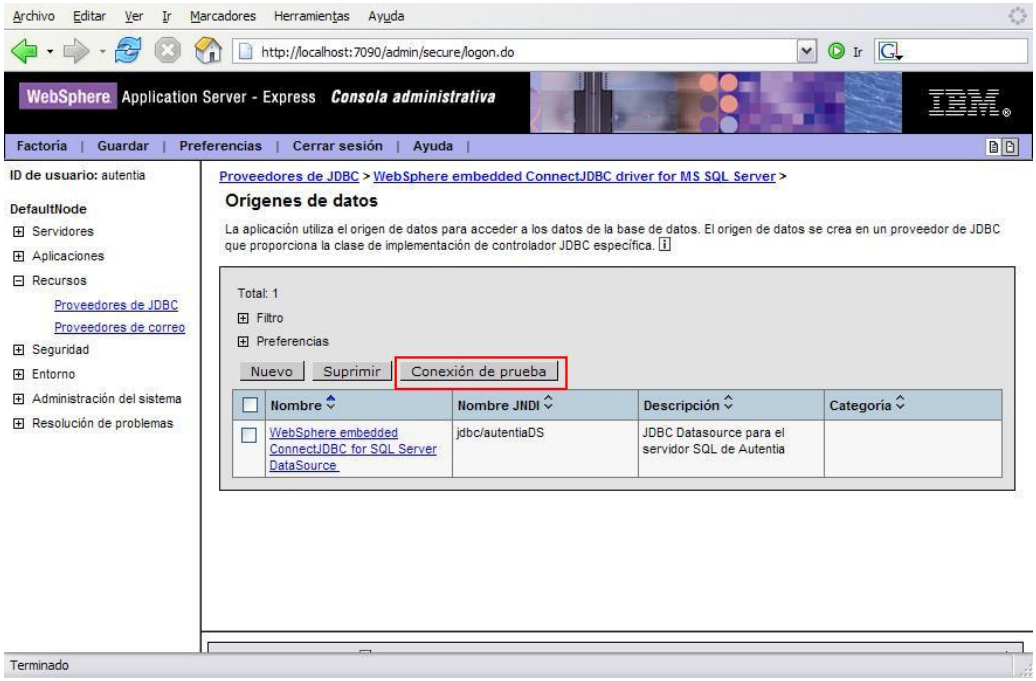


Imagen 18: crear un datasource

Si todo ha ido bien el sistema nos tiene que devolver un mensaje en la parte superior como el siguiente:

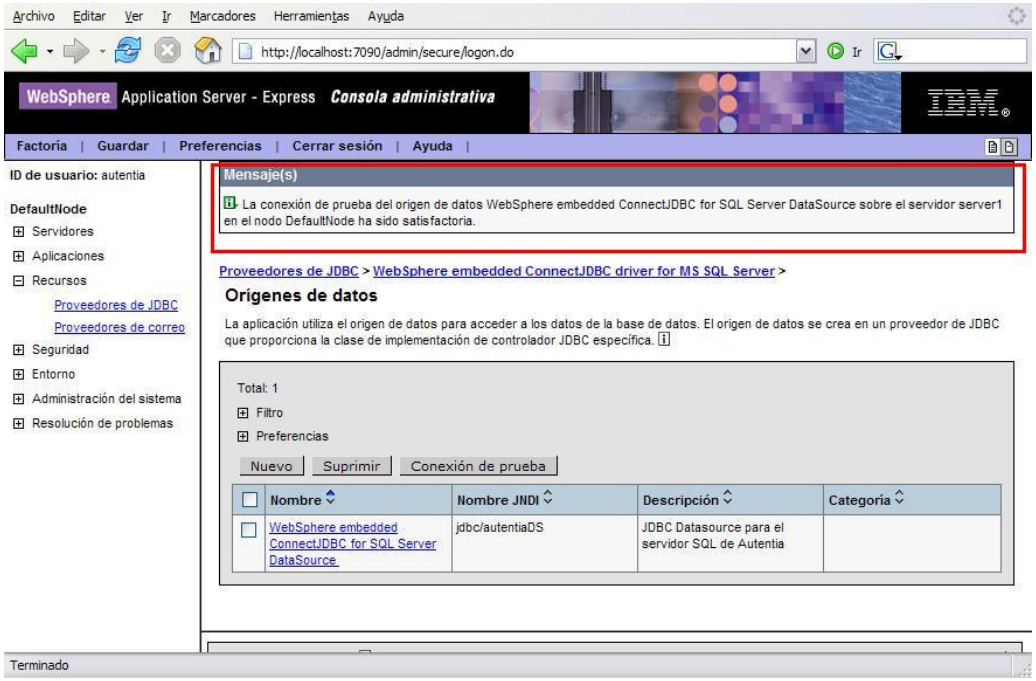


Imagen 19: crear un datasource

2. Conectarse desde java al datasource de SQL Server

Tras la creación del datasource vamos a proceder a mostrar un sencillo ejemplo de java para conseguir un objeto Connection (que representa una conexión con el datasource, y por tanto con la base de datos).

Como el objetivo de este documento es mostrar cómo se crea dicha conexión, los ejemplos de java no seguirán ninguno de los patrones de diseño que se esperaría en un desarrollo serio; eso queda en manos del lector de este documento.

Para mostrar estos ejemplos se ha utilizado como entorno de desarrollo IBM Studio Site Developer 5.1.2, que viene instalado junto

con el servidor de aplicaciones WebSphere.

```
package com.autentia.ejemplo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Hashtable;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.sql.DataSource;

/**

* @author autentia

*

*/

public class Conexion {

public Conexion() {

}

public Connection getConnection() {

try {

Hashtable env = new Hashtable();

env.put(Context.INITIAL_CONTEXT_FACTORY,

"com.ibm.websphere.naming.WsnInitialContextFactory");

InitialContext ctx = new InitialContext(env);

DataSource ds = (DataSource) ctx.lookup("jdbc/autentiaDS");

return ds.getConnection();

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public String getUsers() {

StringBuffer strRet = new StringBuffer(

"<tr><th>Nombre</th><th>Apellidos</th><th>Descripcion</th></tr>");

PreparedStatement pstmt;

ResultSet rs;
```

```
Connection c = this.getConnection();
try {
pstmt = c.prepareStatement(
"SELECT USER_NOMBRE, USER_APELLIDOS, USER_DESC " +
"FROM USUARIOS");

rs = pstmt.executeQuery();
while (rs.next()) {
strRet.append("<tr>");
strRet.append("<td>");
strRet.append(rs.getString(1));
strRet.append("</td>");
strRet.append("<td>");
strRet.append(rs.getString(2));
strRet.append("</td>");
strRet.append("<td>");
strRet.append(rs.getString(3));
strRet.append("</td>");
strRet.append("</tr>");
}
} catch (Exception e) {
strRet.append(e.getMessage());
}
return strRet.toString();
}

public String getPC() {
StringBuffer strRet = new StringBuffer("<tr><th>Nombre PC</th></tr>");
PreparedStatement pstmt;
ResultSet rs;

Connection c = this.getConnection();
try {
pstmt = c.prepareStatement("SELECT PC_NOMBRE FROM autentiapc.dbo.PC");
rs = pstmt.executeQuery();
while (rs.next()) {
strRet.append("<tr>");
strRet.append("<td>");
strRet.append(rs.getString(1));
strRet.append("</td>");
strRet.append("<td>");
```

```
strRet.appe nd("</tr>");
}
} catch (Exception e) {
strRet.append(e.getMessage());
}
return strRet.toString();
}

public String getALL() {
StringBuffer strRet = new StringBuffer(
"<tr><th>Nombre</th><th>Apellidos</th><th>Descripcion</th><th>Nombre PC</th></tr>");
PreparedStatement pstmt;
ResultSet rs;

Connection c = this.getConnection();
try {
pstmt = c.prepareStatement(
"SELECT U.USER_NOMBRE, U.USER_APELLIDOS, U.USER_DESC, P.PC_NOMBRE " +
"FROM autentiapc.dbo.PC P, USUARIOS U " +
"WHERE P.USER_UID = U.USER_UID");

rs = pstmt.executeQuery();
while (rs.next()) {
strRet.append("<tr>");
strRet.append("<td>");
strRet.append(rs.getString(1));
strRet.append("</td>");
strRet.append("<td>");
strRet.append(rs.getString(2));
strRet.append("</td>");
strRet.append("<td>");
strRet.append(rs.getString(3));
strRet.append("</td>");
strRet.append("<td>");
strRet.append(rs.getString(4));
strRet.append("</td>");
strRet.append("</tr>");
}
} catch (Exception e) {
strRet.append(e.getMessage());
}
```

```
return strRet.toString();

}


}
```

Fijémonos ahora en el método getConnection que es aquel que nos da la conexión:

```
Hashtable env = new Hashtable();

env.put(Context.INITIAL_CONTEXT_FACTORY,

"com.ibm.websphere.naming.WsnInitialContextFactory");

InitialContext ctx = new InitialContext(env);
```

Este fragmento de código es la forma definida por IBM para obtener el contexto de WebSphere dentro de la aplicación java, así que siempre debería obtenerse de esta manera.

```
DataSource ds = (DataSource) ctx.lookup("jdbc/autentiaDS");

return ds.getConnection();
```

Mediante este último fragmento obtendremos el objeto que representa el datasource en java, y del cual obtendremos la conexión que nos permitirá realizar operaciones sobre la base de datos. El nombre jdbc/autentiaDS es el nombre JNDI que le dimos al datasource en WebSphere al crearlo.

Como último comentario decir que aunque el datasource accede por defecto a la base de datos 'autentia' (que se indicó en la creación), se pueden acceder a otras bases de datos del servidor mediante sus nombres, como se puede observar con el fragmento de código mostrado a continuación:

```
pstmt = c.prepareStatement(

"SELECT U.USER_NOMBRE, U.USER_APELLIDOS, U.USER_DESC, P.PC_NOMBRE " +

"FROM autentiapc.dbo.PC P, USUARIOS U " +

"WHERE P.USER_UID = U.USER_UID");
```

'Usuarios' es una tabla que se encuentra dentro de la base de datos 'autentia', mientras que 'PC' es otra tabla que se encuentra en otra base de datos del servidor, a saber, 'autentiapc'. La forma de acceder a esta tabla es mediante el nombre de recurso completo 'autentiapc.dbo.PC'.

Y finalmente vamos a mostrar el resultado de ejecutar la clase anterior en la aplicación web:

Nombre	Apellidos	Descripcion
german	jimenez	autentia
daniel	hernandez	autentia
miguel	mena	autentia
alberto	carrasco	autentia

Nombre PC
autentiaserver
pc-mena
pc-alberto
pc-german
pc-daniel

Nombre	Apellidos	Descripcion	Nombre PC
miguel	mena	autentia	pc-mena
alberto	carrasco	autentia	pc-alberto
german	jimenez	autentia	pc-german
daniel	hernandez	autentia	pc-daniel

Si desea contratar formación, consultoria o desarrollo de piezas a medida puede contactar con

J2EE, EJBs, Struts...

Autentia S.L. Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
mod_jk en Linux / Apache2-JBoss	El conector mod_jk se encarga de enviar las peticiones dinámicas de Apache2 a un servidor de aplicaciones JBoss
mod_jk en WindowsXP / Apache2-JBoss	Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando Apache2 y JBoss
mod_jk en WindowsXP / ISS-JBoss	Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando ISS y JBoss
mod_jk en Ubuntu / Apache2-JBoss	Os mostramos como instalar el conector mod_jk sobre la distribución linux Ubuntu utilizando Apache2 y JBoss
Instalación de WebSphere Server	En este tutorial os mostramos como instalar la versión Express 5.1 de WebSphere Server sobre Windows
Primeros pasos con WebSphere	Os mostramos como empezar a manejar WebSphere Server Express 5.1 : arranque y entrada a la consola principal de administración
Desplegar aplicaciones en WebSphere	En este tutorial os mostramos como desplegar una aplicación web en WebSphere Server Express 5.1
Arrancar una aplicación en WebSphere	En este nuevo tutorial os mostramos como arrancar una aplicación que se encuentra desplegada en un WebSphere Server Express 5.1
Configurar ClassLoader para WebSphere	En este nuevo tutorial sobre WebSphere, os mostramos las particularidades y como configurar el ClassLoader cuando se trabaja con WebSphere Server Express 5.1
Instalación del conector para WebSphere	En este tutorial os mostramos como instalar un conector que de soporte a un WebSphere Server Express 5.1

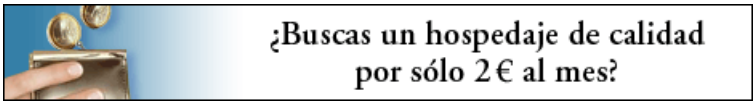
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE



www.AdictosAlTrabajo.com Opimizado 800X600