

¿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

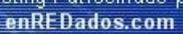
Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

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

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

		Hosting Patrocinado por  
---	---	---

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

Lanzamiento TNTConcept

[Autentia](#) da un paso más en su evolución: Lanzamiento de software propio. Ponemos a vuestra disposición el software que hemos construido para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño Saber más en: <http://tntconcept.sourceforge.net/>

<p>Tutorial desarrollado por: Francisco Javier Martínez Páez</p> <p>Puedes encontrarme en Autentia Somos expertos en Java/J2EE Contacta en fjmpaez@autentia.com</p>	<p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>  <p>real business solutions</p> <p>Catálogo de cursos</p>
--	--

Descargar este documento en formato PDF [interceptarEJB.pdf](#)

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

Java Struts o J2EE

¿experiencia en Java Struts o J2EE? Mejora tu carrera. Unete a nosotros www.soitsa-intesys.com/em...

Master-Java

15 perfiles informáticos remunerado, proyectos estrellas www.meca97.es

Diagram to Code with UML

O/R Mapping with Class & ER diagram Auto Generate .NET, PHP & Java Code www.visual-paradigm.com

JOPC Bridge

Full-featured Java OPC Access: Data Access, Alarms and Events www.netmodule.com

Anuncios Goooooogle

[Anunciarse en este sitio](#)

Interceptando un EJB en JBoss

Los ejemplos de este tutorial están hechos con el siguiente entorno de desarrollo:

- Jboss Eclipse IDE.
- JDK 1.5
- JBoss 4.0.5 GA

En este tutorial os vamos a enseñar (un poco) la arquitectura de EJBs en JBoss y a como modificarla, insertando un interceptor propio dentro de la cadena de interceptores del Proxy Cliente.

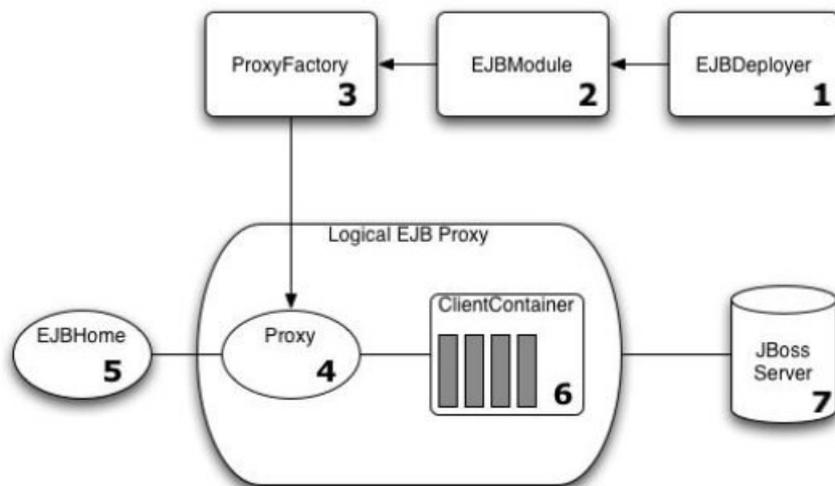
EJBs en JBoss

La arquitectura de EJBs en JBoss está basada en la funcionalidad que provee la clase Proxy del paquete java.lang.reflect (esta clase apareció en la versión 1.3)

La clase Proxy es una clase especial que nos permite interceptar un conjunto de métodos definidos en un interfaz.

Para una breve explicación ver: <http://www.javalobby.org/java/forums/t18631.html>

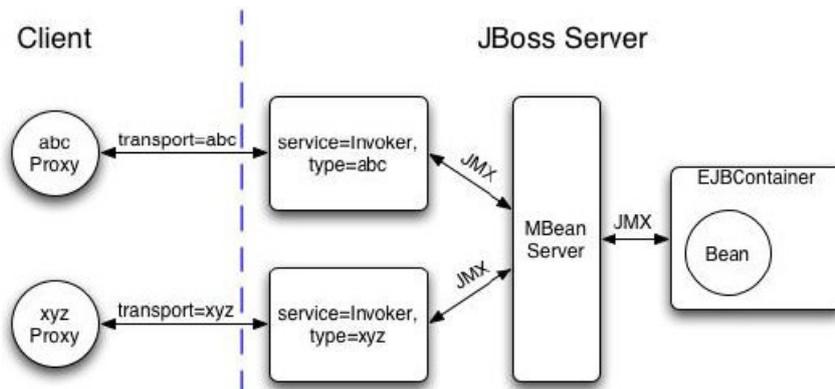
Cuando nosotros en JBoss, hacemos un lookUp del Interfaz Home o del Interfaz Remoto de un EJB, lo que obtenemos en realidad es una clase de tipo Proxy que encapsula la funcionalidad de ambos interfaces. La imagen que se muestra a continuación explica un poco toda esta arquitectura (esta imagen está obtenida de la documentación de JBoss)



Toda esta arquitectura nos ahorra el problema de tener que compilar los EJBs para crear los stubs y los skeletons.

Todas las invocaciones a los métodos de la clase Proxy son delegadas sobre el ClientContainer, que está compuesto por una cadena de Interceptores (clases que extienden la clase *org.jboss.proxy.Interceptor* que serán los manejadores reales de la funcionalidad). Nosotros podemos añadir o quitar funcionalidad modificando la cadena de interceptores.

El único que no podemos eliminar es el último interceptor, que será el que gestione la comunicación con el servidor. Este último interceptor podrá ser RMI, RMI en cluster, HTTP, HTTP en cluster, CORBA etc... En el lado servidor, existirá un componente (detached invoker) que hablará con el cliente (último invoker de la cadena) y nos pondrá en contacto con el contenedor de EJBs a través del servidor JMX de MBeans: (la imagen también está obtenida de la documentación de JBoss)



Toda esta arquitectura de EJBs es modificable en el fichero de configuración:
<RUTA_JBOSS>\server<TU_CONFIGURACION>\conf\standardjboss.xml

También podemos cambiar la configuración de esta arquitectura para un EJB usando su descriptor jboss.xml.

Un EJB que no trabaja en Domingo.

Para explicar un poco esto, vamos a generarnos un EJB de sesión sin estado que sólo podrá ser invocado de lunes a sábado, porque los domingos descansa. Para ello crearemos un interceptor que colocaremos en la cadena de interceptores del ClientContainer en primer lugar para evitar que sea invocado en domingo.

Primero nos crearemos el interceptor que denominaremos SundayInterceptor y que extenderá *org.jboss.proxy.Interceptor* (no olvidéis importar en el proyecto la librería <RUTA_JBOSS>\client\jbossall-client.jar)

```
package com.autentia.jboss.interceptor;

import java.util.Calendar;
import java.util.Date;
import org.jboss.invocation.Invocation;
import org.jboss.proxy.Interceptor;

public class SundayInterceptor extends Interceptor {

    public Object invoke(Invocation invocation) throws Throwable {
        System.out.println("INVOCANDOME AL INTERCEPTOR:"
            +invocation.getMethod().getName());

        Date fechaActual = new Date();
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(fechaActual);
        int dia = calendar.get(Calendar.DAY_OF_WEEK);

        if(Calendar.SUNDAY == dia) {
            throw new SecurityException("LO SIENTO, PERO ES DOMINGO " +
                "Y ESTOY DESCANSANDO.");
        }
        /* SI NO ES DOMINGO, LLAMO AL SIGUIENTE
        * INTERCEPTOR DE LA CADENA */
    }
}
```

```

    }
    return getNext().invoke(invocation);
}
}

```

Ahora creamos el EJB. Primero el Bean:

```

package com.autentia.jboss.interceptor;

import java.rmi.RemoteException;

import javax.ejb.EJBException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;

import javax.ejb.CreateException;

public class InterceptedBean implements SessionBean {

    public InterceptedBean() {
        super();
    }

    public void setSessionContext(SessionContext ctx)
        throws EJBException,
        RemoteException {
    }

    public void ejbRemove() throws EJBException, RemoteException {
    }

    public void ejbActivate() throws EJBException, RemoteException {
    }

    public void ejbPassivate() throws EJBException, RemoteException {
    }

    public void ejbCreate() throws CreateException {
    }

    public void unMetodo() throws java.rmi.RemoteException {
        System.out.println("Invocando unMetodo()");
    }
}

```

Luego el Interfaz Home:

```

package com.autentia.jboss.interceptor;

public interface InterceptedHome
    extends javax.ejb.EJBHome
{

    public Intercepted create()
        throws javax.ejb.CreateException,java.rmi.RemoteException;
}

```

Luego el Interfaz Remoto:

```

package com.autentia.jboss.interceptor;

public interface Intercepted
    extends javax.ejb.EJBObject
{

    public void unMetodo( )
        throws java.rmi.RemoteException;
}

```

Ahora crearemos los descriptores:

El ejb-jar.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejb-jar PUBLIC
"-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN"
"http://java.sun.com/dtd/ejb-jar_2_0.dtd">
<ejb-jar>
  <description>Interceptando</description>
  <display-name>Interceptando</display-name>
  <enterprise-beans>
    <session>
      <description>Intercepted Bean</description>
      <display-name>Intercepted Bean</display-name>
      <ejb-name>Intercepted</ejb-name>
      <home>com.autentia.jboss.interceptor.InterceptedHome</home>
      <remote>com.autentia.jboss.interceptor.Intercepted</remote>
      <ejb-class>com.autentia.jboss.interceptor.InterceptedBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
</ejb-jar>

```

El jboss.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jboss PUBLIC "-//JBoss//DTD JBOSS 3.0//EN"

```

```

"http://www.jboss.org/j2ee/dtd/jboss_3_0.dtd">
<jboss>
  <enterprise-beans>
    <session>
      <ejb-name>Intercepted</ejb-name>
      <jndi-name>ejb/Intercepted</jndi-name>
      <configuration-name>Standard Stateless SessionBean</configuration-name>
      <invoker-bindings>
        <invoker>
          <invoker-proxy-binding-name>
            stateless-paco-invoker
          </invoker-proxy-binding-name>
        </invoker>
      </invoker-bindings>
    </session>
  </enterprise-beans>
</invoker-proxy-bindings>
<invoker-proxy-binding>
  <name>stateless-paco-invoker</name>
  <invoker-mbean>jboss:service=invoker,type=jrmp</invoker-mbean>
  <proxy-factory>org.jboss.proxy.ejb.ProxyFactory</proxy-factory>
  <proxy-factory-config>
    <client-interceptors>
      <home>
        <interceptor>
          com.autentia.jboss.interceptor.SundayInterceptor
        </interceptor>
        <interceptor>org.jboss.proxy.ejb.HomeInterceptor</interceptor>
        <interceptor>org.jboss.proxy.SecurityInterceptor</interceptor>
        <interceptor>org.jboss.proxy.TransactionInterceptor</interceptor>
        <interceptor call-by-value="false">
          org.jboss.invocation.InvokerInterceptor
        </interceptor>
        <interceptor call-by-value="true">
          org.jboss.invocation.MarshallingInvokerInterceptor
        </interceptor>
      </home>
    </bean>
    <interceptor>
      com.autentia.jboss.interceptor.SundayInterceptor
    </interceptor>
    <interceptor>org.jboss.proxy.ejb.StatelessSessionInterceptor</interceptor>
    <interceptor>org.jboss.proxy.SecurityInterceptor</interceptor>
    <interceptor>org.jboss.proxy.TransactionInterceptor</interceptor>
    <interceptor call-by-value="false">
      org.jboss.invocation.InvokerInterceptor
    </interceptor>
    <interceptor call-by-value="true">
      org.jboss.invocation.MarshallingInvokerInterceptor
    </interceptor>
  </client-interceptors>
</proxy-factory-config>
</invoker-proxy-binding>
</invoker-proxy-bindings>
</jboss>

```

Empaquetamos el EJB y lo desplegamos en el servidor de aplicaciones.

Nos creamos una clase cliente para invocarlo:

```

package com.autentia.jboss.interceptor;

import java.rmi.RemoteException;
import java.util.Properties;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;

public class ClienteEJB {

    public static void main(String[] args) {
        Context ctx = null;
        Properties props = new Properties();
        props.setProperty(Context.INITIAL_CONTEXT_FACTORY,
            "org.jnp.interfaces.NamingContextFactory");
        props.setProperty(Context.PROVIDER_URL,"jnp://localhost:1099");
        props.setProperty(Context.URL_PKG_PREFIXES,
            "org.jboss.naming:org.jnp.interfaces");

        try {
            ctx = new InitialContext(props);
        } catch (NamingException e) {
            e.printStackTrace();
        }

        Intercepted remoto=null;
        try{
            Object obj=ctx.lookup("ejb/Intercepted");
            InterceptedHome home=
                (InterceptedHome)
                PortableRemoteObject.narrow(obj,InterceptedHome.class);
            remoto=(Intercepted)home.create();
        }catch (Exception e){

```

```

        System.out.println("Excepcion invocando el EJB->" + e.toString());
    }
    try {
        remoto.unMetodo();
    } catch (RemoteException e) {
        e.printStackTrace();
    }
}
}
}

```

Ejecutamos el cliente: (como hoy es viernes, no debería haber ningún problema:) Mostramos la consola del lado cliente:

```

Problems Javadoc Declaration Search Progress Console X Call Hierarchy
<terminated> ClienteEJB (2) [Java Application] C:\Archivos de programa\Java\jdk1.5.0_08
INVOCANDOME AL INTERCEPTOR:create
INVOCANDOME AL INTERCEPTOR:unMetodo

```

Mostramos la consola del lado servidor:

```

Problems Javadoc Declaration Search Progress Console X Call Hierarchy
JBoss 4.0.5 (LOCAL) [JBoss 4.0.x] C:\Archivos de programa\Java\jdk1.5.0_08\bin\javaw
18:51:30,679 INFO [STDOUT] Invocando unMetodo()

```

Vamos a modificar el interceptor para que no trabaje los viernes y volvemos a empaquetar y desplegar el ejb:

```

....
    if(Calendar.FRIDAY == dia) {
        throw new SecurityException("LO SIENTO, PERO ES DOMINGO " +
            "Y ESTOY DESCANSANDO.");
    }
....

```

Ejecutamos el cliente:

Mostramos la consola del lado cliente:

```

<terminated> ClienteEJB (2) [Java Application] C:\Archivos de programa\Java\jdk1.5.0_08\bin\javaw.exe (23-mar-2007 18:58:52)
INVOCANDOME AL INTERCEPTOR:create
Excepcion invocando el EJB->java.lang.SecurityException: LO SIENTO, PERO ES DOMINGO Y ESTOY DESCANSANDO.
Exception in thread "main" java.lang.NullPointerException
    at com.autentia.jboss.interceptor.ClientEJB.main(ClientEJB.java:35)

```

El lado servidor no muestra nada porque la llamada ha sido interceptada.

Al colocar el interceptor tanto en home como en remoto, como el método create() es parte del home, no se permite ni crear el ejb. Vamos a modificar el jboss.xml eliminando el interceptor de la cadena de interceptores de <home> y vamos a dejar sólo el de <bean>, reempaquetamos, redespelgamos y volvemos a ejecutar el cliente a ver que pasa:

```

Problems Javadoc Declaration Search Progress Console Call Hierarchy Synchronize Console X
<terminated> ClienteEJB (2) [Java Application] C:\Archivos de programa\Java\jdk1.5.0_08\bin\javaw.exe (23-mar-2007 19:05:50)
INVOCANDOME AL INTERCEPTOR:unMetodo
Exception in thread "main" java.lang.SecurityException: LO SIENTO, PERO ES DOMINGO Y ESTOY DE
    at com.autentia.jboss.interceptor.SundayInterceptor.invoke(SundayInterceptor.java:20)
    at org.jboss.proxy.ClientContainer.invoke(ClientContainer.java:100)
    at $Proxy1.unMetodo(Unknown Source)
    at com.autentia.jboss.interceptor.ClientEJB.main(ClientEJB.java:35)

```

Vemos que ahora, la interrupción se produce al invocar al método unMetodo() que pertenece al interfaz remoto.

Pues esto es todo, espero que os haya ayudado a entender un poco la arquitectura de EJBs en JBoss...

Si necesitáis ayuda: <http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](http://creativecommons.org/licenses/by-nc-nd/2.5/).
Puedes opinar sobre este tutorial aquí



Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación



[Autentia S.L.](#) Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
 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	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
Guía rápida de instalación de JBOSS Application Server 4.	En este manual veremos paso a paso la forma de instalar en tu equipo JBoss Application Server 4.
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
MBeans y JBoss	Este tutorial tiene como finalidad familiarizarse con el servidor de aplicaciones JBoss y su API JMX
JBoss en cluster. Ejemplo de alta disponibilidad	Se pretende en este tutorial enseñar a todo aquel que esté interesado en este poderoso servidor de aplicaciones las distintas posibilidades que ofrece JBoss para manejar temas de clustering, alta disponibilidad etc... en aplicaciones J2EE.
WebServices con Axis y JBoss	En este tutorial os mostramos como realizar servicios web utilizando Axis y el contenedor de aplicaciones web JBoss
mod_jk en WindowsXP / Apache2-JBoss	Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando Apache2 y JBoss
XMBeans y JBoss	Este tutorial es una continuación de una anterior llamado MBeans y JBoss
EJB´s y Orion	Recreación de la guía paso a paso de como crear una aplicación Web con EJB´s y Servlets y su despliegue con ANT sobre Orion
Instalar JBoss	Os mostramos como instalar en servidor gratuito de aplicaciones JBOSS así como a automatizar su arranque y parada.

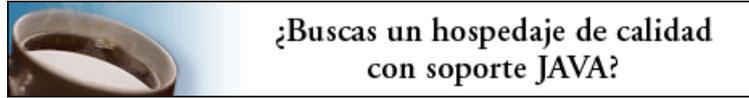
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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 Optimizado 800X600