

# ¿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)

 Powered by 	Hosting Patrocinado por <b>enREDados.com</b> 
---	--

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



CoNcept

Nuevo

## TNTConcept versión 0.2 ( 26/04/2007 )

**Autentia** da un paso más en su evolución, hemos lanzado una nueva versión con más de 50 mejoras. Ponemos a vuestra disposición el software que hemos construido para nuestra gestión interna, llamado TNTConcept (au**Te**NTia).

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

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

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

### PKS Software GmbH

Migration of ADABAS C applications to relational databases  
[www.pks.com](http://www.pks.com)

### Master-Java

15 perfiles informáticos remunerado, proyectos estrellas  
[www.meca97.es](http://www.meca97.es)

### Altio

Real-time data visualisation, interaction and integration.  
[www.altio.com](http://www.altio.com)

Anuncios 

### PATRÓN DE DISEÑO PROXY

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

- Jboss Eclipse IDE.
- JDK 1.5

### INTRODUCCIÓN

Desde hace unos años están apareciendo gran cantidad de herramientas que introducen nuevos conceptos como la orientación a aspectos (AOP), la interceptación, etc... . En el fondo, todas ellas están basadas en algo no tan nuevo como es el patrón Proxy. Podríamos definir un proxy, como un objeto que fuerza que todas las llamadas al objeto al que se desea invocar pasen previamente a través de él. Esto nos permite realizar acciones que podríamos llamar transversales a la funcionalidad realizada por el objeto final invocado, como por ejemplo, trazabilidad, seguridad, transaccionalidad etc... . Es decir, si por ejemplo nosotros invocamos al método *sayHello* del objeto *obj* de la clase *Saludo*, podríamos "interceptar" esa llamada a través del Proxy, para comprobar si realmente el que invoca al método tiene permisos para realizar la acción, sin tener que modificar para ello el método *sayHello* . En esto se basan gran cantidad de herramientas como Spring, Acegi, la arquitectura de EJBs de gran cantidad de servidores de aplicaciones como JBoss, etc.

Para ello, usaremos la clase *java.lang.reflect.Proxy* que lleva ya con nosotros desde la versión 1.3 de la J2SE.

Sin más, vamos a realizar un ejemplo sencillo, para demostrar todo esto:

### EL EJEMPLO

Nos crearemos un proyecto nuevo en nuestro Eclipse, y generaremos primero un Interfaz que denominaremos Manager, que define toda la funcionalidad que proveemos puede tener la lógica de negocio de las clases de nuestra aplicación:

```
package com.autentia.adictos.proxy;
```

```
public interface Manager {
    public void save(Object obj);
    public void remove(Integer id);
}
```

Vamos a construir ahora una clase implemente el interfaz:

```
package com.autentia.adictos.proxy;
```

```
public class UserManagerImpl implements Manager {
```

```

public void save(Object obj) {
    System.out.println("I save user Objects");
}

public void remove(Integer id) {
    System.out.println("I remove user objects");
}
}

```

Vamos ahora a contruir el Manejador o Handler, que será el que será previamente invocado a los métodos de los managers:

```

package com.autentia.adictos.proxy;

import java.lang.reflect.InvocationHandler;
import java.lang.reflect.Method;

public class ManagerHandler implements InvocationHandler {

    Manager realManager = null;

    public ManagerHandler(Manager realManager) {
        this.realManager = realManager;
    }

    /* Este es el método callback que será invocado previamente a cada método de los managers. */
    public Object invoke(Object proxy, Method method, Object[] args)
        throws Throwable {

        System.out.println("A Manager is being invoked:"+method.getName());

        // Continuamos invocando al Manager real.
        return method.invoke(realManager,args);
    }
}

```

Ahora crearemos una Factoría de Managers, que creará los mismos encapsulados dentro de un Proxy:

```

package com.autentia.adictos.proxy;

import java.lang.reflect.Proxy;

public class FactoryManager {

    public Manager createManager(Class claseManager) {

        Manager realManager = null;

        try {
            // Creamos un objeto de la clase que recibimos.
            realManager = (Manager)claseManager.newInstance();
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

        /* Creamos el CallBack Handler y le pasamos el objeto real para ser invocado posteriormente en su método invoke. */

        ManagerHandler handler = new ManagerHandler(realManager);

        // Creamos el proxy.
        Class[] interfacesQueEncapsulo = new Class[] {Manager.class};
        return (Manager)Proxy.newProxyInstance(
            claseManager.getClassLoader(),
            interfacesQueEncapsulo,handler);
    }
}

```

Creamos una clase para probar lo que hemos hecho:

```

package com.autentia.adictos.proxy;

public class Test {

    public static void main(String[] args) {
        // Creamos ahora la factoría
        FactoryManager factory = new FactoryManager();

        // Usamos la factoría para crear un Proxy sobre UserManagerImpl
        Manager manager = factory.createManager(UserManagerImpl.class);

        Object obj = new Object();
        // Invocamos a los métodos:
        manager.save(obj);
        manager.remove(1);
    }
}

```

Mostremos la salida estándar:

```

Java - Test.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Search Progress Call Hierarchy Synchronize Console x
<terminated> Test (5) [Java Application] C:\Archivos de programa\Java\jdk1.5.0_08\bin\javaw.exe (26-abr-2007 14:07:33)
A Manager is being invoked:save
I save user Objects
A Manager is being invoked:remove
I remove user objects

```

Como podéis ver, el código del Handler es invocado previamente al código del UserManagerImpl. Curioso ¿no?. ¿Os suena esto?. Supongo que si conocías ya el tema de AOP, AspectJ etc..., verás que se parece sospechosamente. Ahora, en el manejador podríamos trazar lo que ocurre, o guardar en base de datos un registro, o autenticar o autorizar, o abrir una transacción, invocar al método y cerrar una transacción, etc... Suponed éste código:

```

public Object invoke(Object proxy, Method method, Object[] args)
    throws Throwable {
    openTransaction();
    Object obj = method.invoke(realManager,args);
    closeTransaction();
    return obj;
}

```

Creo que esto habla por sí sólo.

El único problema es que el hacer que nuestros Managers implementen un interfaz es algo rígido, por eso es mejor usar frameworks que nos proporcionan todo esto de una manera más flexible, como Spring, Acegi, EJB 3.0 etc..., pero creo que es más bonito conocer como se hacen las herramientas o en que se basan antes de usarlas, y no creer que es magia....



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](mailto: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
<a href="#">EJB´s y Orion</a>	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
<a href="#">Programación AOP en el lenguaje DE</a>	Cristobal González nos habla de Introducción a la "Programación Orientada a Aspectos". Extraído de los apéndices de su libro "Introducción al lenguaje de programación DE (del que es autor)", actualmente en preparación.
<a href="#">Interceptando un EJB en JBoss</a>	En este tutorial os vamos a enseñar la arquitectura de EJBs en JBoss y a como modificarla, insertando un interceptor propio dentro de la cadena de interceptores del Proxy Cliente.
<a href="#">Aplicación de Patrones de Diseño en Java</a>	En este tutorial os mostramos como las técnicas avanzadas de diseño ( como patrones de diseño ) contribuyen a la contrucción de aplicaciones profesionales en Java.
<a href="#">Técnicas básicas y poco comentadas en Java</a>	Os mostramos como realizar algunas cosas simples en Java: Formateo de decimales y enteros, gestión de preferencias y comparación entre objetos de nuevas clases
<a href="#">Novedades en Java 1.5</a>	Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.
<a href="#">TNTConcept Beta 0.2 y anticipo de la siguiente</a>	Este documento hace referencia a la nueva versión Beta 0.2 de nuestra herramienta gratuita TNTConcept. Se tratará de explicar aspectos técnicos y funcionales de las nuevas operativas incluidas en esta nueva versión.

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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

[Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE](#)



**¿Buscas un hospedaje de calidad  
con soporte JAVA?**

[www.AdictosAlTrabajo.com](http://www.AdictosAlTrabajo.com) Optimizado 800X600