

¿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
enREDados.com



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



CoNcept

Lanzado

TNTConcept versión 0.6 (12/07/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, 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>Autor del tutorial: Cristhian Kirs Herrera Basurto</p>	<p>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</p>	
	<p>§ Lugar de residencia: Quito - Ecuador</p>	<p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>	
	<p><i>Cuento con experiencia en el área de desarrollo de software y en la docencia académica. Dentro de la construcción de software he manejado las etapas de: análisis, diseño, personalización e implementación de aplicaciones bajo ambientes Cliente / Servidor e Internet.</i></p> <p>Cristhian.Herrera@gmail.com / cherrera@kruqer.com.ec</p>	 <p>real business solutions</p> <p>Catálogo de cursos</p>	

Descargar este documento en formato PDF [EJB3vsSpring.pdf](#)

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

Fecha de creación del tutorial: 2007-10-17

[Comparativa entre EJB 3 y Spring Framework](#)

[Sobre el presente documento:](#)

[Introducción](#)

[Enterprise JavaBeans](#)

[Spring Framework](#)

[Criterio de Comparación](#)

[Fuente:](#)

[Referencias del documento original:](#)

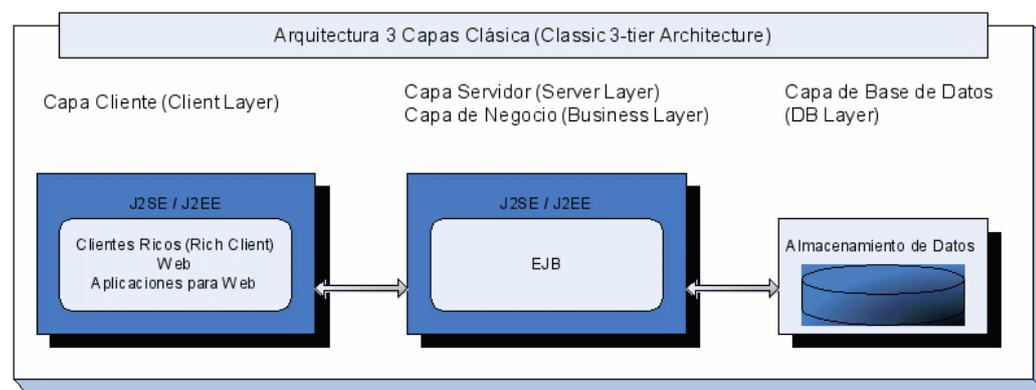
Comparativa entre EJB 3 y Spring Framework

Sobre el presente documento:

Hace algún tiempo que encontré una comparativa de EJB3 versus Spring Framework, realizada por Janis Graudins y Larissa Zaitseva, la misma que me he permitido traducir pues la considero muy ilustrativa y una buena ayuda a la hora de determinar cual de las alternativas emplear. He omitido la sección de "abstract" del documento original y en general en la traducción he procurado ser fiel al espíritu del documento en inglés pero en algunos puntos de la tabla de comparación final he agregado algunas apreciaciones propias cuando he considerado que lo que dice no es del todo correcto.

Introducción

Las aplicaciones basadas en el lenguaje de programación JAVA comprenden una parte significativa del universo de desarrollo de software. Java fue inicialmente creado por SUN Corporation y ha logrado dentro la comunidad Open-Source la aceptación y el soporte de verdaderos gigantes de las TI como BEA Systems, IBM Corporation y JBOSS (Red Hat). JAVA es especialmente popular en el desarrollo de grandes aplicaciones empresariales. Las comparativas muestran que el 25,3% de las grandes compañías usan JAVA en sus aplicaciones más importantes (octubre 2005) [7]. La plataforma Java 2 Enterprise Edition (J2EE) provee grandes oportunidades para el desarrollo de sistemas distribuidos, se utiliza en la mayoría de las aplicaciones empresariales y la tecnología Enterprise JavaBeans (EJB) es una parte muy importante de la misma. Usualmente la arquitectura de una aplicación J2EE contiene varias capas separadas como se puede apreciar en la figura. La capa de servidor típicamente contiene componentes de servidor con lógica de negocio, estos son manejados por un contenedor EJB (de acuerdo con la implementación de la especificación EJB). El contenedor EJB es parte del servidor de aplicaciones (usualmente el contenedor EJB y el servidor de aplicaciones no pueden ser separados y son proporcionados por el mismo proveedor). El servidor de aplicaciones provee el ciclo de vida de los componentes, así como servicios de seguridad y manejo de transacciones.



Desafortunadamente, las versiones anteriores de EJB fueron demasiado complejas y nuevos componentes de manejo tecnológico aparecieron. Spring Framework, en su versión 1.0 liberada en Marzo de 2004, es un producto gratuito de carácter Open-Source, el mismo que es un contenedor liviano de componentes de negocio que pueden ser utilizados como una alternativa frente a EJB. En general, Spring Framework provee algunos servicios adicionales tales como Spring Web Model View Controller (MVC), pero estos están fuera del alcance de este documento.

A continuación se describe brevemente algunas de las características básicas de EJB y de Spring Framework.

Enterprise JavaBeans

Una de las metas de la arquitectura EJB es la de poder escribir de manera fácil aplicaciones de negocio orientadas a objetos y distribuidas, basadas en el lenguaje de programación JAVA [1]. Desafortunadamente, las versiones 1.0 a 2.1 de EJB fueron demasiado complejas y no alcanzaron esta meta. El propósito de EJB 3 es el de proveer el soporte de la arquitectura de EJB y al mismo tiempo reducir la complejidad para el desarrollo de aplicaciones empresariales. Para simplificar la arquitectura EJB se realizaron los siguientes cambios [3]:

- Se introduce las anotaciones de metadatos (metadata annotations) [10] las mismas que pueden ser usadas en combinación con el descriptor de despliegue (deployment descriptor) ó separadas del mismo, para anotar aplicaciones EJB (especificar tipos de componentes, comportamiento, etc.), como una manera de encapsular dependencias del ambiente de trabajo y recursos.
- Se elimina el requerimiento de especificar una interfaz "home"
- En los enterprise beans se elimina la necesidad de implementar una interfaz específica (javax.ejb.EnterpriseBean)
- Se simplifican los tipos de enterprise beans (Los entity beans fueron removidos)
- La existencia de interceptores reemplaza la necesidad de implementar interfaces tipo callback¹.
- Los valores por defecto se emplean lo menos posible (se usa la aproximación de configuración por excepción).
- Se reducen los requerimientos para el manejo de excepciones

Como contrapunto se introducen en EJB 3 las anotaciones de metadatos y de interceptores como las siguientes:

- La persistencia de entidades (Entity Persistence) fue simplificada y soportada para modelar dominios de negocio medianos a grandes, además ahora es posible proveer contenedores EJB 3 livianos que pueden ser usados en una capa cliente fuera de la caja del servidor de aplicaciones.
- Se mejora en EJB QL el soporte para consultas y sentencias SQL nativas
- Se provee de un servicio de temporizador (Timer Service) manejado por el contenedor EJB el mismo que permite ejecutar Enterprise Beans en eventos de tiempo específicos.
- En EJB3 se puede usar AOP a través de interceptores.

Spring Framework

El principal objetivo de Spring Framework es el constituirse en una alternativa sencilla y fácil ante EJB. La simplificación del desarrollo de aplicaciones y de sus respectivas pruebas (testing) es una de las claves del éxito de Spring. Este Framework se sustenta en dos características básicas en su núcleo: Inversión de Control (Inversion of Control IoC) y la Programación Orientada a Aspectos (Aspect-orient programming) [11].

Usualmente, los objetos obtienen las referencias de otros objetos requeridos por sí mismos (tal como en EJB 2.0 los beans obtienen los recursos necesarios usando JNDI). La inversión de control permite inyectar las dependencias en un bean al momento de su creación usando un manejador externo. El bean sólo necesita definir la propiedad requerida en su código así como el método de establecimiento (set() method). La fuente primaria de la inyección de dependencias es un archivo de configuración en formato XML. Por ejemplo *productService* necesita realizar alguna operación sobre *customerService*, la referencia de *customerService* debe ser inyectada en la propiedad *customer* de *com.article.ProductService* en la siguiente figura se muestra un ejemplo de cómo se mapearía esta configuración en el respectivo archivo XML:

```
<beans>
<bean id="customerService" class="com.article.CustomerServiceImpl"/>
<bean id="productService" class="com.article.ProductServiceImpl">
<property name="customer" ref="customerService"/>
</bean>
</beans>
```

La Programación Orientada a Aspectos (Aspect-Orient Programming AOP) permite implementar la mayoría de los servicios comunes (como manejo de transacciones, seguridad, logging, etc.) que pueden ser aplicados en múltiples componentes. En el caso del uso de AOP no se requiere ningún conocimiento acerca de cómo han sido enmascarados (wrapped²) los servicios. AOP es usada en Spring [8] para:

- Proveer servicios de aplicación (enterprise services) declarativos. Ejemplo declarar el manejo de transacciones
- Permitir a los usuarios la facilidad de implementar sus propios aspectos personalizados

Spring provee un número de servicios adicionales que son basados en IoC y AOP. Estos servicios deben ser comparados con sus equivalentes en EJB para poder tener un buen criterio de evaluación.

Criterio de Comparación

El propósito de la comparación es mostrar diferencias entre EJB y Spring. Para llegar a esta meta se han seleccionado los siguientes criterios

1. Manejo de transacciones (Transaction Manager), la comparación debe permitir comparar las diferentes clases de implementación de transacciones que están soportadas
2. Oportunidades de criterios de transacción incluidos (Transaction Opportunities), atributos soportados, niveles de isolation³), soporte de transacciones anidadas.
3. Manejo de entidades de persistencia (Entity Persistente) que permitan evaluar funcionalidad para persistencia de objetos Object-Relational Mappings (ORM)
4. AOP (Interceptors) muestran como se provee funcionalidad para programación orientada a aspectos
5. Configuración de aplicaciones (Application Configuration), la posibilidad de instalar la configuración de la aplicación y servicios declarativos
6. Seguridad (Security) la comparación muestra como se ofrecen diferentes niveles de seguridad.
7. Flexibilidad de servicios (Service Flexibility), evaluar la posibilidad de reemplazar servicios por otros
8. Servicios de integración (Service Integration), detecta las facilidades de integración en especial con los servidores de aplicaciones.

9. Funcionalidad adicional (Additional Functionality), describe funcionalidad adicional provista por el framework,
10. Criterios para testing (Testability Criterion), en este criterio se analiza la facilidad para realizar test en forma conjunta y de todos los componentes por separado.
11. Madurez de la tecnología y soporte (Technology Maturity and support). Este criterio analiza la madurez del producto y si la compañía ofrece soporte para el mismo.
12. Precio (Price), posible precio y costos asociados del producto.
13. Documentación (Documentation), investiga si se provee de documentación, ejemplos y de soporte para los mismos.

Cada criterio fue evaluado en un rango desde 0.0 (no soportado) a 1.0 (completamente soportado)

El resultado de la comparación se muestra en la siguiente tabla:

Nota	EJB 3.	Spring Framework	Nota
0.7	Manejo de Transacciones (Transaction Manager)		0.9
	Sólo se soporta JTA Transaction Manager. Este es el manejador de transacciones que se usa en forma primaria en las aplicaciones de negocio.	JTA es una de las alternativas disponibles. Pueden usarse diferentes ORM como Hibernate, JDO, JDBC, ODBC	
0.6	Oportunidad de Transacciones (Transaction Opportunities)		0.8
	Sólo los atributos de una transacción son soportados. No se soportan transacciones anidadas.	Soporta atributos de transacción y diferentes niveles de isolation. Las transacciones anidadas son soportadas sólo si el manejo de transacciones las implementa.	
0.9	Persistencia de Entidades (Entity Persistente)		0.7
	Define su propio manejo de persistencia, posibilidad de emplear anotaciones en ORM, EJB QL y sentencias SQL nativas, además de integración con Hibernate	Usa implementaciones ORM de terceros como Hibernate, IBATIS, JDO, OJB. La calificación parece ser un poco sesgada a favor de EJB3, desde mi perspectiva la nota debería ser inversa. Simplemente Spring provee una mayor diversidad de alternativas para el manejo de persistencia.	
1.0	Programación Orientada a Aspectos [AOP (Interceptors)]		0.9
	Interceptores por defecto pueden ser especificados y aplicados a todos los componentes, interceptores de callback, Los interceptores pueden ser implementados en la misma clase o en una clase separada. Pueden ser seteados (establecidos) usando anotaciones en el descriptor de despliegue.	Provee servicios de aplicación en forma declarativa, aspectos personalizados pueden ser definidos.	
1.0	Configuración de la aplicación (Application configuration)		0.8

	Usa anotaciones de metadatos en forma primaria, pero es posible sobrescribirlas en el descriptor de despliegue	En forma primaria se usan archivos XML de configuración, posibilidad de usar Jakarta Commons Attributes ó las anotaciones J2SE estándares	
0.9	Seguridad (Security)		0.6
	Soporta seguridad declarativa por medio de anotaciones de metadatos y declaraciones en el descriptor de despliegue	Provee integración con la solución Open Source Acegi Security Framework, el mismo que soporta seguridad declarativa basada en el uso de IoC y AOP.	
0.7	Flexibilidad de Servicios (Services flexibility)		1.0
	Depende de la implementación de EJB. Si el servidor provee una estructura modular entonces sólo se requieren los servicios que este puede usar.	Cualquier servicio puede ser ensamblado usando un archivo XML de configuración	
0.9	Integración de Servicios (Services Integration)		0.7
	El servidor de aplicaciones contiene una implementación de EJB y tiene la oportunidad de optimizar rendimiento. Existe soporte para clustering	Spring Framework es desarrollado en forma separada de un servidor de aplicaciones y resulta más dificultoso optimizar su integración. No se aplica si no se usa un servidor de aplicaciones. En este punto no estoy de acuerdo con los autores originales del artículo, si bien es cierto Spring no está previsto para hacer uso de todas las facilidades de un servidor de aplicaciones, se trata de un framework abierto que puede ser fácilmente integrado con otros frameworks y adicionalmente una aplicación Spring puede ser publicada en cualquier servidor de aplicaciones del mercado.	
N/A	Funcionalidad Adicional (Additional Functionality)		N/A
	Depende de la implementación EJB proporcionada por el proveedor En ambos casos el estudio debería mostrar alguna puntuación o mejor eliminar el criterio.	Provee oportunidades de integración con varios productos Open-Source, Spring MVC	
0.8	Testeo (Testability)		1.0
	La mayor parte de los componentes pueden ser testeados fuera del contenedor, pero el servicio de contenedor de objetos sólo puede ser testeado dentro del contenedor (Por ejemplo EntityManager)	Todos los componentes pueden ser testeados fuera del contenedor	
0.7	Madurez de la Tecnología y Soporte (Technology maturity, support)		0.6
	EJB es un estándar, creado por expertos que provienen de diferentes proveedores (incluidos Oracle, BEA, IBM) y en sus implementaciones son completamente compatibles. El estándar como tal se mantiene únicamente en estado PFD	La tecnología de Open-Source es soportada por Interface21. Es relativamente madura (2 años desde la liberación 1.0), pero no es un estándar.	

			Tampoco parece ser una justificación del todo valedera, pues el número de proyectos y por ende el número de desarrolladores con experiencia en Spring es, por ahora, mayor al de EJB3 y por tanto hay una mayor aplicación y madurez en esta tecnología.
0.3	Precio (Price)		0.9
	Primariamente los productos son pagos, la implementación de EJB de JBOSS es gratuita	Productos Open-Source son gratuitos.	El criterio como tal no parece estar bien empleado, el que un producto tengo 0 costo no implica que por eso deba tener una mejor puntuación en ese rubro.
0.8	Documentación (Documentation)		0.5
	Los diferentes proveedores producen excelente y extensa documentación. Se presentan varios ejemplos. Y los vendedores dan el soporte del caso	La documentación en formato Javadoc no contiene todos los detalles técnicos, pocos ejemplos detallados.	La documentación de algunos proveedores no es tan completa como podría esperarse, especialmente si hay una paga de por medio en el servicio.

Los resultados de la comparación muestran que Spring Framework es especialmente preferible para usarse en compañías pequeñas, en conjunto con otros productos Open-Source. Es un framework muy simple, conveniente y flexible, pero la mismo tiempo muy poderoso. Se recomienda el uso de Spring en los casos donde un contenedor de aplicaciones pesado no es necesario.

En el otro extremo EJB 3 puede ser utilizado por compañías cuyo plan de mantenimiento de aplicaciones basadas en EJB es un compromiso a largo tiempo y buscan que las nuevas versiones de una aplicación sean por completo compatibles con las versiones anteriores. La integración de EJB con el servidor de aplicaciones provee grandes oportunidades en escalamiento, y optimización del programa de desarrollo.

Fuente:

- "Comparative Analysis of EJB3 and Spring Framework", Janis Graudins, Larissa Zaitseva, Documento original publicado en la siguiente dirección <http://ecet.ecs.ru.acad.bg/cst06/Docs/cp/SIII/IIIA.18.pdf>

Referencias del documento original:

- [1] L. DeMichiel, "Enterprise JavaBeans Specification, Version 2.1", [Online document], Sun Microsystems, November 12, 2003, 646 pages, Available at HTTP: <http://java.sun.com/products/ejb/docs.html>

- [2] L. DeMichiel, M. Keith "JSR 220: Enterprise JavaBeans, Version 3.0. EJB 3.0 Simplified API", [Online document], Sun Microsystems, December 18, 2005, 59 pages, Available at HTTP: <http://jcp.org/aboutJava/communityprocess/pfd/jsr220/index.html>
- [3] L. DeMichiel, M. Keith "JSR 220: Enterprise JavaBeans, Version 3.0. EJB Core Contracts and Requirements", [Online document], Sun Microsystems, December 18, 2005, 526 pages, Available at HTTP: <http://jcp.org/aboutJava/communityprocess/pfd/jsr220/index.html>
- [4] J. Gaudins "Comparing analysis of Java application servers", Scientific proceedings of Riga Technical University, 2004, p. 118 " 125.
- [5] J. Gaudins, L. Zaitseva "Application Server Evaluation Method", Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing" (CompSysTechAçã,,ç05), 2005, p.IIIB.6-1 " IIIB.6-6.
- [6] J. Gaudins, L. Zaitseva "Application Server Selection for definite systemsâ,,ç class", Proceedings of 19th International conference "Systems" for Automation of Engineering and Research" (SAER-2005), 2005, p.230 " 235.
- [7] S. Hamm, "Java? It's So Nineties", [Online document], BusinessWeek Online, December 13, 2005, Available at HTTP: http://www.businessweek.com/technology/content/dec2005/tc20051213_042973.htm
- [8] R. Johnson, J. Hoeller, A. Arendsen "Spring. Java/J2EE Application Framework", [Online document], 2004-2005, 290 pages, Available at HTTP: <http://www.springframework.org/documentation>
- [9] R. Lambert, "An Introduction to the Spring Framework", [Online document], Chicago Java Users Group, June 21, 2005, Available at HTTP: <http://cjug.org/presentations/2005/June21/Spring-Framework-Intro-Rob-Lambert.ppt>
- [10] R. Mordani "Common Annotations for the Java Platform", [Online document], Sun Microsystems, October 12, 2005, 32 pages, Available at HTTP: <http://jcp.org/aboutJava/communityprocess/pfd/jsr250/index.html>
- [11] C. Walls, R. Breidenbach "Spring in Action", Manning Publications, 2005, 472 pages

1 Mecanismo de programación para devolver el control del flujo del programa a un punto en particular del mismo, luego de realizada una acción determinada.

2 Wrapped: Término en inglés que significa envoltura y que en el contexto del texto denota la acción de enmascarar u ocultar una implementación.

3 ISOLATION: En manejo de persistencia el término hace referencia a garantizar que las transacciones que son dependientes unas de otras se graben todas o ninguna.



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

[Block]

[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
Spring: definición dinámica de Beans	Este tutorial habla sobre la modificación dinámica de los beans del contexto para simplificar la configuración de Spring
EJB 3.0: Resurrection	Este tutorial nos va a presentar las nuevas funcionalidades que nos aportan los EJB 3.0.
Interceptando un EJB en JBoss	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.
JSF y comparativa con Struts	Os mostramos los pasos necesarios para empezar a utilizar JSF (Java Server Faces) y su comparación / relación con Struts
Introducción a Spring Web Flow	Spring Web Flow es un módulo de extensión del framework Spring, que facilita la implementación del flujo de páginas de una aplicación web
EJB 3.0, un ejemplo práctico con Maven y JBoss	Este tutorial presenta un ejemplo sencillo donde se verá como desarrollar EJBs de sesión y de entidad, inyección de dependencias, llamar a los EJBs desde una aplicación Web, definición de un DataSource, y como configurarlo y hacerlo funcionar en JBoss, y
EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss	En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss
Despliegue gráfico de EJBs	Os mostramos como crear y desplegar de un modo gráfico un EJB de sesión el el servidor de aplicaciones de referencia de Sun
Comparativa entre Hibernate y EJB3 en la Capa de Persistencia	El presente documento pretende dar algunas luces a la comparativa entre la opción de usar Hibernate y/ó EJB3 para la capa de persistencia
URLs amigables con Spring MVC	En este tutorial se va a hacer un ejemplo práctico utilizando Spring MVC para la configuración de URLs amigables de nuestra aplicación

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](#)

