

¿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)



Los años enseñan las cosas que los días jamás llegan a conocer

Adictos AL Trabajo

Powered by Autentia

Hosting patrocinado por

enREDADOS

- Inicio Quienes somos Tutoriales Formación Comparador de salarios Comic Charlas Más

Estas en: Inicio Tutoriales Primeros pasos con JBoss Seam

Últimas Noticias

- » Autentia patrocina el primer Agile Open Spain
- » Nuevas funcionalidades en la web de www.adictosaltrabajo.com
- » Disponible la primera versión de los plugins para integrar Maven y Bugzilla.
- » Historia de la Informática. Capítulo 76. 1999
- » Historia de la Informática. Capítulo 74. 1997
- » Comentando el libro: Guía de Estilo, Protocolo y Etiqueta en la empresa.
- » Comentando el libro: "El viaje a la felicidad. Nuevas claves científicas" de Eduardo Punset
- » Lanzamiento del nuevo Web de Autentia
- » ¿Es necesario iQué ofrece esta Web?

+Noticias Destacadas

- » Autentia patrocina el primer Agile Open Spain
- » Lanzamiento del nuevo Web de Autentia
- » Autentia cumple 6 años
- » Grupo XING

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en AdictosALTrabajo...

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)

Tutorial desarrollado por

Juan Alonso Ramos



Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión (cursando Ingeniería Informática)

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosALTrabajo.com es el Web de difusión de conocimiento de Autentia.



[Catálogo de cursos](#)

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

Fecha de creación del tutorial: 2009-09-21

Primeros pasos con JBoss Seam

Índice de contenidos.

1. Introducción.
2. Entorno
3. Configuración del proyecto (seam setup)
4. Generando la estructura del proyecto (seam new-project)
5. Generando la funcionalidad del proyecto (seam generate-entities)
6. Despliegue del proyecto
7. Probando la aplicación
8. Conclusiones.

1. Introducción

Este tutorial pretende ser una guía introductoria al Framework JBoss Seam utilizando el seam-gen. El seam-gen es una utilidad que trae JBoss Seam para crear el código fuente de la aplicación con el único requisito de tener una base de datos del que sacar la lógica de negocio en base al modelo de datos especificado. El proceso de creación no puede ser más sencillo, nos irá preguntando el nombre del proyecto, ubicación del servidor de aplicaciones, la base de datos que queremos utilizar, nombres de paquetes, empaquetamiento del proyecto (ear/war), etc. Con esto generará la aplicación que nos servirá de base para continuar desarrollando la lógica de negocio y demás funcionalidades del proyecto. Iremos viendo los pasos necesarios para crear una aplicación web de una sencilla biblioteca.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Acer Aspire 5920G (Core Duo T8300 2.4GHz, 3GB RAM, 320 GB HD).
- Sistema operativo: Windows Vista
- JBoss Seam 2.2.0.GA
- JBoss AS 5.1.0.GA
- JDK 1.6
- Apache Ant 1.7.1

3. Configuración del proyecto (seam setup)

Lo primero será descargarnos el framework JBoss Seam. Para ello vamos a la web oficial y bajamos el zip. Pinchar[aquí](#)

La instalación es sencilla, basta con descomprimir el zip en un directorio, por ejemplo en D:\Desarrollo\jboss-seam-2.2.0.GA

También debemos tener instalado el servidor de aplicaciones JBoss AS. Para ello lo descargamos de la web [desdeaquí](#). La instalación al igual que de Seam es tan sencilla como descomprimir el zip, por ejemplo D:\Servidores\jboss-5.1.0.GA

Sólo nos queda ejecutar el script de creación de las propiedades del proyecto localizado en la raíz del proyecto, llamado setup. Para ello abrimos una consola y escribimos **seam setup**. En este punto es necesario que esté instalado **Apache Ant**. A continuación Seam nos irá preguntando por una serie de propiedades que se necesitan para crear la estructura del proyecto y que debemos ir configurando.

- **Ubicación del proyecto:** D:\Proyectos\tutoriales\
- **Ubicación del servidor JBoss:** D:\Servidores\jboss-5.1.0.GA
- **Dominio utilizado del servidor JBoss:** default
- **Ubicación del servidor Glassfish:** D:\Servidores\Sun\AppServer\AppServer
- **Dominio utilizado del servidor Glassfish:** domain1
- **Nombre del proyecto:** biblioteca
- **¿Usar ICEFaces en lugar de RichFaces?:** n
- **Skin utilizado:** ruby
- **Empaquetamiento del proyecto (ear/war):** ear
- **Nombre del paquete baes:** com.autentia.tutoriales
- **Nombre del paquete de los beans de sesión:** com.autentia.tutoriales.action
- **Nombre del paquete de los beans de entidad:** com.autentia.tutoriales.entity
- **Nombre del paquete de las clases de test:** com.autentia.tutoriales.test
- **Base de datos utilizada:** mysql
- **Directorio al driver de mysql:** D:\Drivers\mysql-connector-java-5.1.8-bin.jar
- **Dialecto de Hibernate:** org.hibernate.dialect.MySQLDialect
- **Clase del Driver JDBC para MySQL:** com.mysql.jdbc.Driver
- **Clase del DataSource JDBC para MySQL:** com.mysql.jdbc.jdbc2.optional.MysqlDataSource
- **URL de conexión a la base de datos:** jdbc:mysql://localhost:3306/biblioteca
- **Usuario de la base de datos:** biblioteca
- **Password de la base de datos:** *****
- **Nombre de la base de datos:** -
- **¿Usar todas las tablas que están en la base de datos?:** y
- **¿Borrar la base de datos en cada despliegue?:** n

Si todo ha ido bien saldrá la siguiente traza:

Google

Web

www.adictosaltrabajo.com

Últimos tutoriales

- 2009-09-21
[Primeros pasos con JBoss Seam](#)
- 2009-09-21
[Integración con sistemas de Bug Tracking desde NetBeans 6.7..](#)
- 2009-09-14
[EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.](#)
- 2009-09-12
[Instalación de Liferay en Tomcat existente](#)
- 2009-09-11
[Release Bugzilla Maven Plugin](#)
- 2009-09-11
[Enlazar Bugzilla con MavenChangesPlugin](#)
- 2009-09-08
[Sobre las reglas de codificación o... ¿de dónde salen esos caracteres "raros"?](#)
- 2009-08-28
[Cómo hacer deploy del site de Maven en SourceForge](#)
- 2009-08-26
[Ordenación por cantidades en informe cruzado](#)
- 2009-08-20
[Selenium IDE-Incorporando while en los test](#)
- 2009-08-14
[Blender y JMonkeyEngine. Exportación de archivos Blender y uso de los mismos en JMonkeyEngine](#)
- 2009-08-14
[5º tutorial TNT Concept Versión 0.16.1 Gestión de informes, vacaciones y utilidades](#)
- 2009-08-14
[Joomla 1.5. Instalación y configuración](#)
- 2009-08-13
[Introducción a los diagramas EPC \(Event-Driven Process Chain\)](#)
- 2009-08-10
[Blender. Animaciones avanzadas y renderización](#)

```

view plain print ?
01. [propertyfile] Creating new property file: D:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.properties
02. [echo] Installing JDBC driver jar to JBoss AS
03. [copy] Copied 1 empty directory to 1 empty directory under D:\Servidores\jboss-5.1.0.GA\server\default\lib
04.
05. init:
06.
07. init-properties:
08. [echo] D:/Servidores/jboss-5.1.0.GA
09.
10. validate-workspace:
11.
12. validate-project:
13.
14. settings:
15. [echo] JBoss AS home: D:/Servidores/jboss-5.1.0.GA
16. [echo] GlassFish home: D:/Servidores/Sun/AppServer/AppServer
17. [echo] Project name: biblioteca
18. [echo] Project location: D:/Proyectos/tutoriales
19. [echo] Project type: ear
20. [echo] IceFaces: n
21. [echo] Action package: com.autentia.tutoriales.action
22. [echo] Model package: com.autentia.tutoriales.entity
23. [echo] Test package: com.autentia.tutoriales.test
24. [echo] JDBC driver class: com.mysql.jdbc.Driver
25. [echo] JDBC DataSource class: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
26. [echo] Hibernate dialect: org.hibernate.dialect.MySQLDialect
27. -- [echo] JDBC URL: jdbc:mysql://localhost:3306/biblioteca
28. [echo] Database username: biblioteca
29. [echo] Database password: *****
30. [echo]
31. [echo] Type 'seam create-project' to create the new project
32. BUILD SUCCESSFUL
    
```

También se habrá generado un build.properties en D:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen para que con una tarea de Ant se genere todo el proyecto

Antes de ejecutar el script de creación del proyecto necesitamos tener una base de datos de donde Seam pueda sacar la estructura necesaria. La base de datos de la biblioteca que vamos a crear es muy sencilla, tendrá tres tablas (libro, autor y reserva). El script de creación de la base de datos y las tablas es el siguiente:

```

view plain print ?
01. --
02. -- Create schema biblioteca
03. --
04. CREATE DATABASE IF NOT EXISTS biblioteca;
05. USE biblioteca;
06.
07. --
08. -- Definition of table `libro`
09. --
10. CREATE TABLE `libro` (
11. `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
12. `nombre` varchar(45) NOT NULL,
13. `autor` varchar(45) NOT NULL,
14. `isbn` varchar(45) NOT NULL,
15. `fecha_publicacion` date NOT NULL,
16. `editorial` varchar(45) NOT NULL,
17. PRIMARY KEY (`id`),
18. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
19.
20. --
21. -- Definition of table `socio`
22. --
23. CREATE TABLE `socio` (
24. `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
25. `nombre` varchar(45) NOT NULL,
26. `apellidos` varchar(45) NOT NULL,
27. `numero_socio` varchar(45) NOT NULL,
28. `fecha_alta` date NOT NULL,
29. PRIMARY KEY (`id`),
30. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
31.
32. --
33. -- Definition of table `reserva`
34. --
35. CREATE TABLE `reserva` (
36. `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
37. `fecha_reserva` date NOT NULL,
38. `id_libro` int(10) unsigned NOT NULL,
39. `id_socio` int(10) unsigned NOT NULL,
40. PRIMARY KEY (`id`),
41. CONSTRAINT `fk_reserva_socio` FOREIGN KEY (`id_socio`) REFERENCES `socio` (`id`),
42. CONSTRAINT `fk_reserva_libro` FOREIGN KEY (`id_libro`) REFERENCES `libro` (`id`),
43. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
    
```

4. Generando la estructura del proyecto (seam new-project)

Ya estamos en disposición de crear nuestro proyecto de biblioteca. Para ello desde la consola invocamos la tarea de `ant seam new-project` que crea el proyecto en base a las propiedades configuradas anteriormente.

Los fuentes generados estarán en el directorio que le indicamos en el paso anterior, D:\Proyectos\tutoriales\biblioteca

Nombre	Fecha modificación	Tipo	Tamaño
settings	18/09/2009 13:48	Carpeta de archi...	
bootstrap	18/09/2009 13:48	Carpeta de archi...	
lib	18/09/2009 13:48	Carpeta de archi...	
nbproject	18/09/2009 13:48	Carpeta de archi...	
resources	18/09/2009 13:48	Carpeta de archi...	
src	18/09/2009 13:48	Carpeta de archi...	
view	18/09/2009 13:48	Carpeta de archi...	
.classpath	18/09/2009 13:48	Archivo CLASSP...	3 KB
.project	18/09/2009 13:48	Archivo PROJECT	1 KB
biblioteca.iml	18/09/2009 13:48	Archivo IML	12 KB
biblioteca.ipr	18/09/2009 13:48	Archivo IPR	15 KB
biblioteca.launch	18/09/2009 13:48	Archivo LAUNCH	1 KB
build.properties	18/09/2009 13:48	Archivo PROPER...	1 KB
build.xml	18/09/2009 13:48	Archivo XML	21 KB
build-dev.properties	18/09/2009 13:48	Archivo PROPER...	1 KB
build-prod.properties	18/09/2009 13:48	Archivo PROPER...	1 KB
debug-glassfish-biblioteca.launch	18/09/2009 13:48	Archivo LAUNCH	1 KB
debug-jboss-biblioteca.launch	18/09/2009 13:48	Archivo LAUNCH	1 KB
deployed-jars-ear.list	18/09/2009 13:48	Archivo LIST	1 KB
deployed-jars-war.list	18/09/2009 13:48	Archivo LIST	1 KB
explode.launch	18/09/2009 13:48	Archivo LAUNCH	2 KB
glassfish-build.xml	18/09/2009 13:48	Archivo XML	11 KB
glassfish-readme.txt	18/09/2009 13:48	Documento de t...	3 KB
hibernate-console.properties	18/09/2009 13:48	Archivo PROPER...	1 KB
seam-gen.properties	18/09/2009 13:48	Archivo PROPER...	2 KB
validate.xml	18/09/2009 13:48	Archivo XML	5 KB

Como se puede ver nos ha creado los ficheros de configuración de Eclipse (.classpath y .project) y también de Netbeans (carpeta nbproject) para que podamos abrir el proyecto en nuestro IDE favorito.

Si navegamos por la estructura de directorios que nos ha creado vemos algunas clases interesantes, por ejemplo las clases Authenticator y AuthenticatorBean dentro del directorio src/main/com/autentia/tutoriales/entity para securizar nuestra biblioteca y que únicamente puedan acceder usuarios autorizados. Por defecto añade el siguiente código:

- 2009-08-10 [Gestión de Calidad, tablón y seguimiento en TNT Concept Versión 0.16.1](#)
- 2009-08-10 [Cómo hacer una página web](#)
- 2009-08-06 [Tips And Tricks JUnit Spring](#)
- 2009-08-03 [Instalación de VirtualBox PUEL](#)
- 2009-08-03 [Gestión de contactos y pedidos en TNT Concept versión 0.16.1](#)
- 2009-08-03 [Comentando el libro: La estrategia del océano azul](#)
- 2009-07-30 [Funciones esenciales para crear un juego.](#)
- 2009-07-30 [2º tutorial TNT Concept versión 1.16.1](#)
- 2009-07-29 [Hibernate Search, Bridges, Analizadores y más](#)
- 2009-07-24 [Migración de EJB3 a JPA y Spring.](#)
- 2009-07-20 [Directorio de ejemplos de jMonkey Engine](#)
- 2009-07-19 [JSR-179 Location API para J2ME: Posicionamiento geográfico en nuestras aplicaciones.](#)
- 2009-07-16 [Gestión de Usuarios en TNT Concept versión 0.16.1](#)
- 2009-07-16 [Continuación del Tutorial: JMonkeyEngine, Creación de nuestro primer juego.](#)
- 2009-07-16 [Como implementar el Scene Monitor para analizar las escenas en JMonkeyEngine](#)
- 2009-02-26 [Transformaciones de escena en JMonkeyEngine](#)
- 2009-07-15 [Detalles del juego de la moto en JMonkeyEngine.](#)
- 2009-07-14 [JMonkeyEngine, Creación de nuestro primer juego.](#)
- 2009-07-13 [Ajax tests con Selenium: prototype.js e ICEfaces.](#)
- 2009-07-08 [AOP con AspectJ y Maven](#)
- 2009-07-07 [Instalación y configuración de Eclipse Galileo](#)
- 2009-07-07 [Iniciarse en el manejo de JME, Creación de un Cloth.](#)
- 2009-07-06 [Primeros pasos con Blender: Pintando nuestra mascota en 3D](#)
- 2009-07-06 [DBUnit-Exportar e Importar BBDD](#)
- 2009-07-05 [JMeter, Pruebas de stress sobre aplicaciones web: Grabando y reproduciendo navegaciones](#)
- 2009-07-02 [Axis2: Invocación de Servicios Web usando distintos MEP](#)
- 2009-07-02 [Instalación OpenOffice](#)
- 2009-07-02 [Juegos 3D en Java: Blender y JMonkeyEngine](#)
- 2009-06-20

```

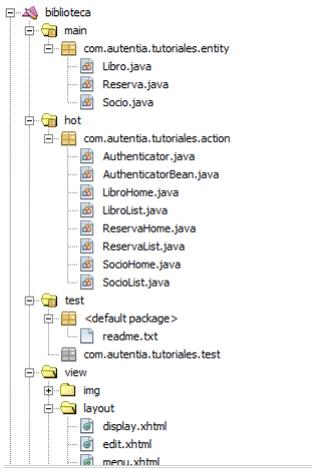
view plain print ?
01. @Stateless
02. @Name("authenticator")
03. public class AuthenticatorBean implements Authenticator {
04.     @Logger private Log log;
05.
06.     @In Identity identity;
07.     @In Credentials credentials;
08.
09.     public boolean authenticate()
10.     {
11.         log.info("authenticating {0}", credentials.getUsername());
12.         //write your authentication logic here,
13.         //return true if the authentication was
14.         //successful, false otherwise
15.         if ("admin".equals(credentials.getUsername()))
16.         {
17.             identity.addRole("admin");
18.             return true;
19.         }
20.         return false;
21.     }
22. }
23. }
    
```

5. Generando la funcionalidad del proyecto

Ahora vamos a crear las entidades de nuestro modelo y la lógica de negocio para las operaciones CRUD (alta, baja, modificación y listado) de las entidades a partir de las tablas de la base de datos. Para ello ejecutamos la tarea de ant **seam generate-entities**

Esta tarea genera el código de las entidades Libro, Usuario y Reserva y los componentes Seam que se encargarán de la lógica de negocio.

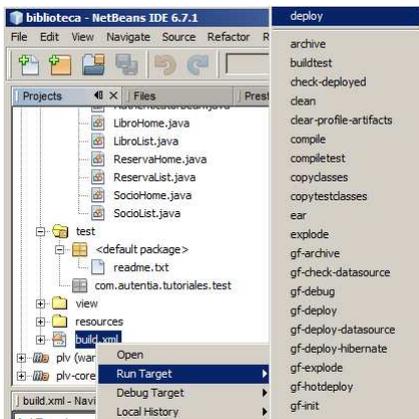
Por otro lado crea toda la parte web, pantalla de login, página de inicio de la aplicación, menú de navegación, páginas de creación y listado de cada entidad, reglas de navegación, configuración de componentes, seguridad, etc.



6. Despliegue del proyecto

Con lo que ya tenemos podemos desplegar la aplicación en un servidor de aplicaciones (recordar que dijimos en la configuración del proyecto que queríamos empaquetar la aplicación como un ear). En nuestro caso utilizaremos JBoss ya que los fuentes generados están preparados para este servidor aunque si queremos desplegar en Glassfish no habría problema, se incluye un script de Ant (glassfish-build.xml), que lo permite.

Gracias al build.xml que se encuentra en la raíz del proyecto, el despliegue se reduce a ejecutar la tarea **deploy**. Podemos ejecutarla a través de Netbeans (o Eclipse) o bien desde la consola ejecutando **ant deploy**



La tarea deploy se encarga de compilar el proyecto, empaquetarlo como un ear y echarlo al directorio deploy del servidor JBoss. También copiará el datasource necesario para la conexión a la base de datos.

7. Probando la aplicación

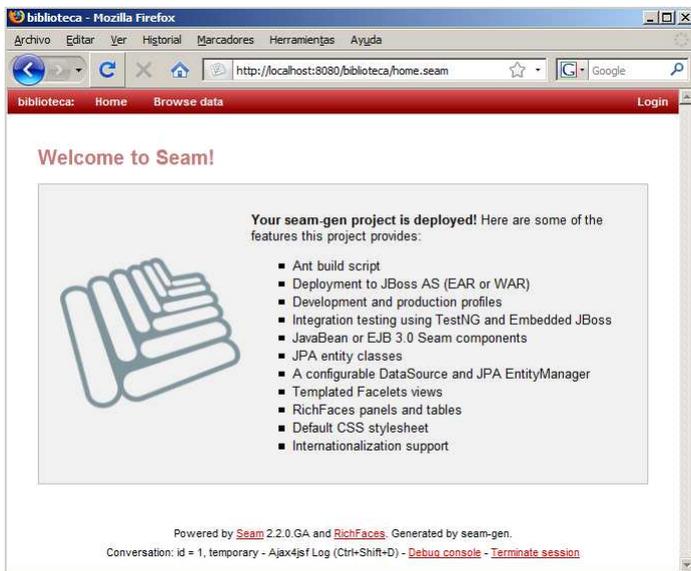
Antes de arrancar el servidor será necesario copiar el driver MySQL al directorio server/default/lib de JBoss. Para arrancar ejecutamos el script D:\Servidores\jboss-5.1.0.GA\bin\run.bat y entramos en la aplicación mediante <http://localhost:8080/biblioteca>

- STAX (Xml Pull Parser): Streaming API para XML
- 2009-06-15 Configuración de la desconexión de usuarios con ICEFaces
- 2009-06-10 LWUIT: Una librería gráfica tipo AWT o Swing para J2ME
- 2009-06-10 Mapas mentales con XMind
- 2009-02-26 Redimensionar Imágenes en Windows Vista
- 2009-06-08 UploadFile con Icefaces + Hibernate + Anotaciones
- 2009-06-05 Habilitar exportación en Liferay

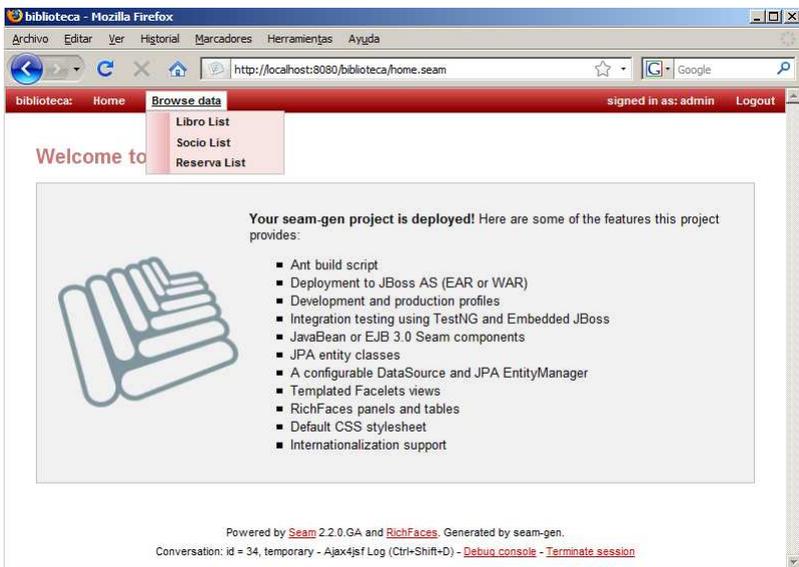
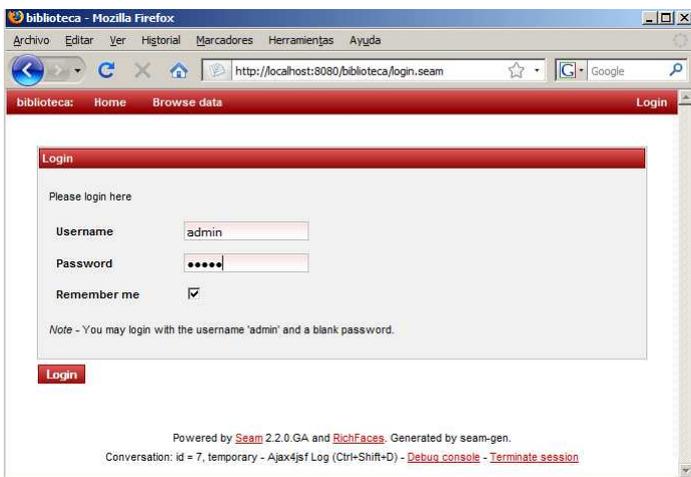
Últimas ofertas de empleo

- 2009-07-31 T. Información - Operador (día / noche) - BARCELONA.
- 2009-06-25 Atención a cliente - Call Center - BARCELONA.
- 2009-06-19 Otras - Ingeniería (minas, puentes y puertos) - VALENCIA.
- 2009-06-17 Comercial - Ventas - ALICANTE.
- 2009-06-03 Comercial - Ventas - VIZCAYA.

Anuncios Google

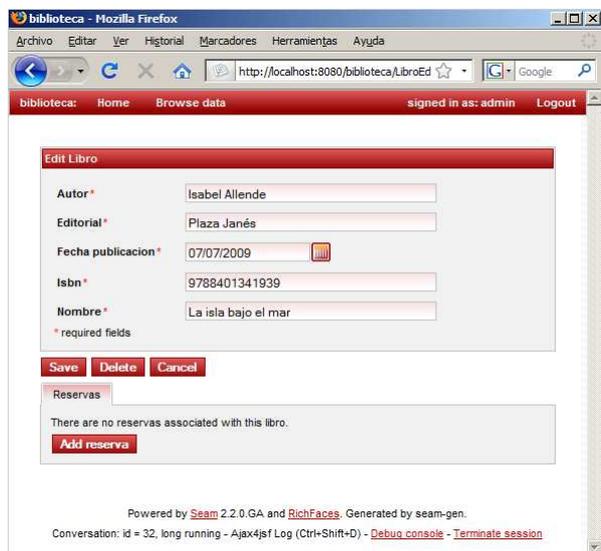


Nos logamos en la aplicación pinchando en la opción 'Login' situada en la parte superior derecha. Por defecto Seam configura la seguridad de manera que permite ver los listados sin necesidad de estar logado pero sí que lo requiere para dar de alta, modificar o borrar una entidad. Esta lógicamente podremos configurarlo en función de nuestras necesidades. Observad también que los textos no están en inglés. Para ponerlos en español basta con crear un messages_es.properties y añadir un nuevo soporte para el idioma español en el faces-config.xml

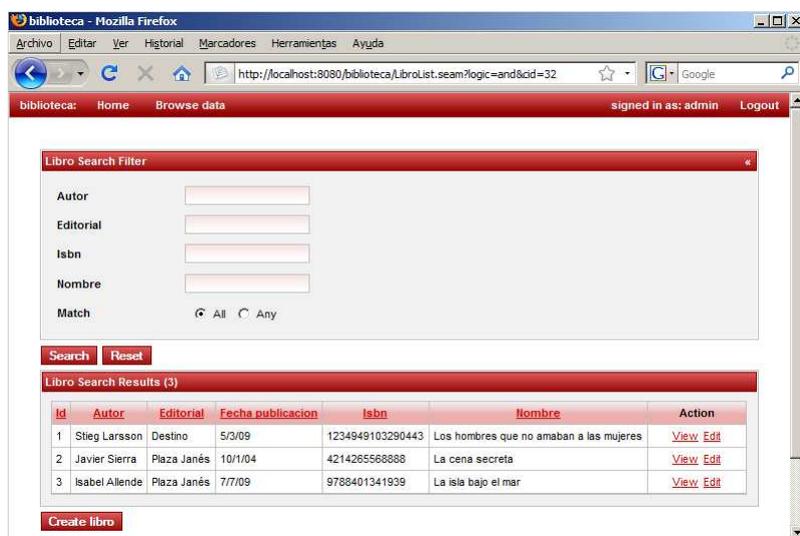


Ya sólo nos queda probar la aplicación que nos ha generado Seam. Para ello accedemos a las opciones 'Libro List' y 'Socio List' para dar de alta unos cuantos datos con los que probar.

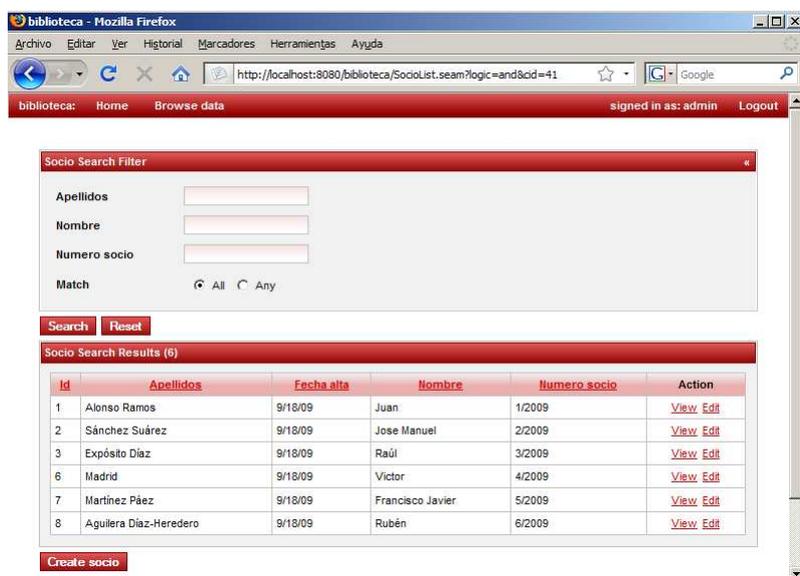
Alta de un libro



Listado de libros



Listado de socios



Realizada un reserva, pestaña libro

Reserva Details

Fecha reserva 9/21/09

Edit Done

Libro Socio

Id	Autor	Editorial	Fecha publicacion	Isbn	Nombre	Action
3	Isabel Allende	Plaza Janés	7/7/09	9788401341939	La isla bajo el mar	View

Powered by [Seam 2.2.0.GA](#) and [RichFaces](#). Generated by seam-gen.
Conversation: id = 54, temporary - [Ajax4jsf Log \(Ctrl+Shift+D\)](#) - [Debug console](#) - [Terminate session](#)

Realizada un reserva, pestaña socio

Reserva Details

Fecha reserva 9/21/09

Edit Done

Libro Socio

Id	Apellidos	Fecha alta	Nombre	Numero socio	Action
1	Alonso Ramos	9/18/09	Juan	1/2009	View

Powered by [Seam 2.2.0.GA](#) and [RichFaces](#). Generated by seam-gen.
Conversation: id = 54, temporary - [Ajax4jsf Log \(Ctrl+Shift+D\)](#) - [Debug console](#) - [Terminate session](#)

La funcionalidad que se ha generado automáticamente contiene todo lo necesario para realizar login/logout de usuarios, crear, editar, borrar, listar, ordenar los campos del listado, paginación de los resultados, buscador avanzado de cada entidad por cualquiera de sus atributos, etc. Como véis sin escribir ni una sola línea de código. Únicamente tendríamos que internacionalizar los textos y cambiar los patrones de formateo de fechas ya que están en formato MM/dd/yyyy y tendríamos una aplicación completa eso sí, al gusto de Seam.

Esto puede que no nos cuadre con nuestros requisitos aunque sí que nos sirve para ver cómo montan la arquitectura del proyecto que se centraliza en los componentes anotados con @Name. Por ejemplo para el caso de la entidad Libro, nos ha creado los componentes LibroHome y LibroList. El primero se encarga de la creación de nuevas instancias de tipo Libro extendiendo la clase EntityHome que proporciona la funcionalidad para realizar todas las operaciones necesarias para los CRUD. El segundo componente se encarga de añadir los parámetros de búsqueda de la entidad para listar los libros extendiendo de EntityQuery que ya se encarga de lanzar las queries.

Componente LibroHome

```

view plain print ?
01. package com.autentia.tutoriales.action;
02.
03. import com.autentia.tutoriales.entity.*;
04. import java.util.ArrayList;
05. import java.util.List;
06. import org.jboss.seam.annotations.Name;
07. import org.jboss.seam.framework.EntityHome;
08.
09. @Name("libroHome")
10. public class LibroHome extends EntityHome<Libro> {
11.
12.     public void setLibroId(Integer id) {
13.         setId(id);
14.     }
15.
16.     public Integer getLibroId() {
17.         return (Integer) getId();
18.     }
19.
20.     @Override
21.     protected Libro createInstance() {
22.         Libro libro = new Libro();
23.         return libro;
24.     }
25.
26.     public void load() {
27.         if (isIdDefined()) {
28.             wire();
29.         }
30.     }
31.
32.     public void wire() {
33.         getInstance();
34.     }
35.
36.     public boolean isWired() {
37.         return true;
38.     }
39.
40.     public Libro getDefinedInstance() {
41.         return isIdDefined() ? getInstance() : null;
42.     }
43.
44.     public List<Reserva> getReservas() {
45.         return getInstance() == null ? null : new ArrayList<Reserva>(
46.             getInstance().getReservas());
47.     }
48.
49. }

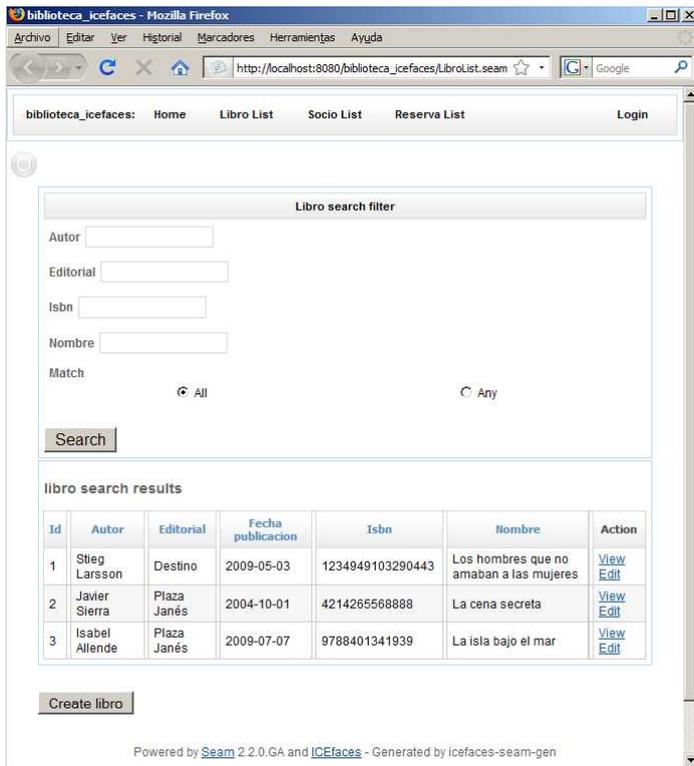
```

Componente LibroList

```

view plain print ?
01. package com.autentia.tutoriales.action;
02.
03. import com.autentia.tutoriales.entity.*;
04. import org.jboss.seam.annotations.Name;
05. import org.jboss.seam.framework.EntityQuery;
06. import java.util.Arrays;
07.
08. @Name("libroList")
09. public class LibroList extends EntityQuery<Libro> {
10.
11.     private static final String EJBQL = "select libro from Libro libro";
12.
13.     private static final String[] RESTRICTIONS = {
14.         "lower(libro.autor) like lower(concat(#{libroList.libro.autor}, '%')",
15.         "lower(libro.editorial) like lower(concat(#{libroList.libro.editorial}, '%')",
16.         "lower(libro.isbn) like lower(concat(#{libroList.libro.isbn}, '%')",
17.         "lower(libro.nombre) like lower(concat(#{libroList.libro.nombre}, '%'))";
18.
19.     private Libro libro = new Libro();
20.
21.     public LibroList() {
22.         setEjbql(EJBQL);
23.         setRestrictionExpressionStrings(Arrays.asList(RESTRICTIONS));
24.         setMaxResults(25);
25.     }
26.
27.     public Libro getLibro() {
28.         return libro;
29.     }
30. }
    
```

Si queremos generar la interfaz con ICEFaces en lugar de Richfaces únicamente tendríamos que cambiar el parámetro 'icefaces=y' dentro del build.properties generado dentro de D:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.properties.



8. Conclusiones.

Como aspecto más importante a destacar de la utilidad seam-gen de JBoss Seam es su facilidad de uso y la flexibilidad que te da, ya que permite configurar el proyecto a nuestro gusto. También es verdad que no aporta todo lo que necesitamos pero es útil para empezar con un proyecto con JBoss Seam desde cero.

Como aspecto negativo está que no nos crea el proyecto para maven, espero que esto lo mejoren en futuras versiones.

Un saludo.

Juan

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo
 Malo
 Regular
 Bueno
 Muy bueno

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial, con tu ayuda, podemos ofrecerte un mejor servicio.

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

■ Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.

- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.  ADD THIS BLOG TO MY FAVORITES



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿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?

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

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Gestión de contenidos

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
--------	---------	-------	---------	------------	-------	-----

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.