

¿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)



Estás en: Inicio » Tutoriales » Trabajando con JAXB y Eclipse



DESARROLLADO POR:
Carlos León Villamayor

Consultor tecnológico de desarrollo de proyectos informáticos.
Constructor de Adictos Al Trabajo 2.0

Ingeniero Técnico en Informática (cursando grado superior)

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



Fecha de publicación del tutorial: 2011-07-15



Share |

Regístrate para votar

Trabajando con JAXB y Eclipse

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Paso 1: Obtener Implementación de JAXB
- 4. Paso 2: Crear Proyecto Java en Eclipse
- 5. Paso 3: Creación de archivo XSD
- 6. Paso 4: Creación de clases accesoras
- 7. Paso 5: Creación de XML
- 8. Resultado
- 9. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil 2,4 Ghz Intel Core2 Duo P8600
- Sistema Operativo: Windows 7 Profesional x86 SP1
- Eclipse Helios (Revisar tutorial de Alex para su instalación)
- Versión de JRE 6 instalada en el sistema
- Implementación de JAXB 2.2.4

2. Introducción

En este tutorial realizaremos el proceso de generación de un schema (XSD) de XML utilizando JAXB y las herramientas proporcionadas por eclipse para realizarlo de forma gráfica.

Paso 1: Obtener Implementación de JAXB

Podemos descargar la distribución de JAXB desde <http://jaxb.java.net/>. En nuestro caso descargaremos la versión siguiente:

Catálogo de servicios Autentia

Últimas Noticias

- VII Autentia Cycling Day
- Autentia patrocina la charla sobre Java SE 7 en Madrid
- Alfresco Day 2011
- XVII Charla Autentia - Grails - Vídeos y Material
- ¡¡¡ 15 millones de descargas de tutoriales !!!

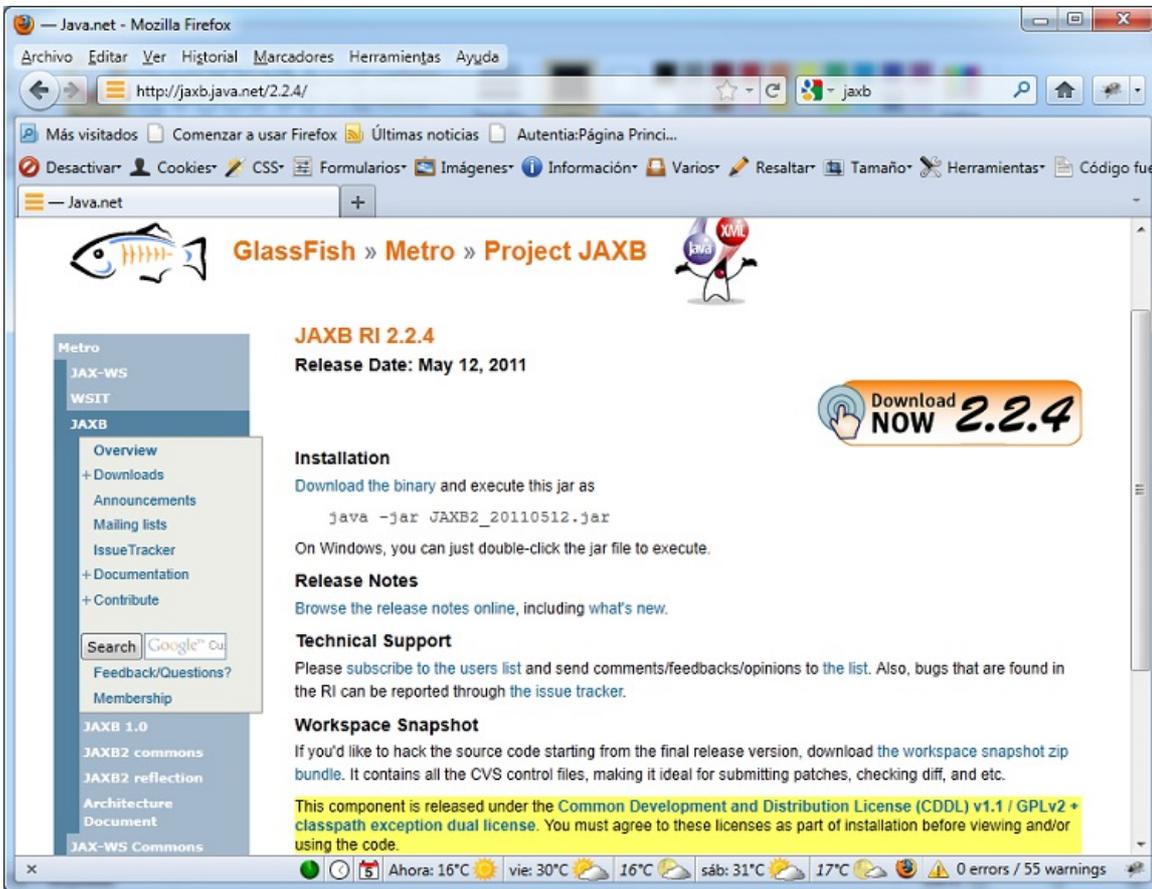
Histórico de NOTICIAS

Últimos Tutoriales

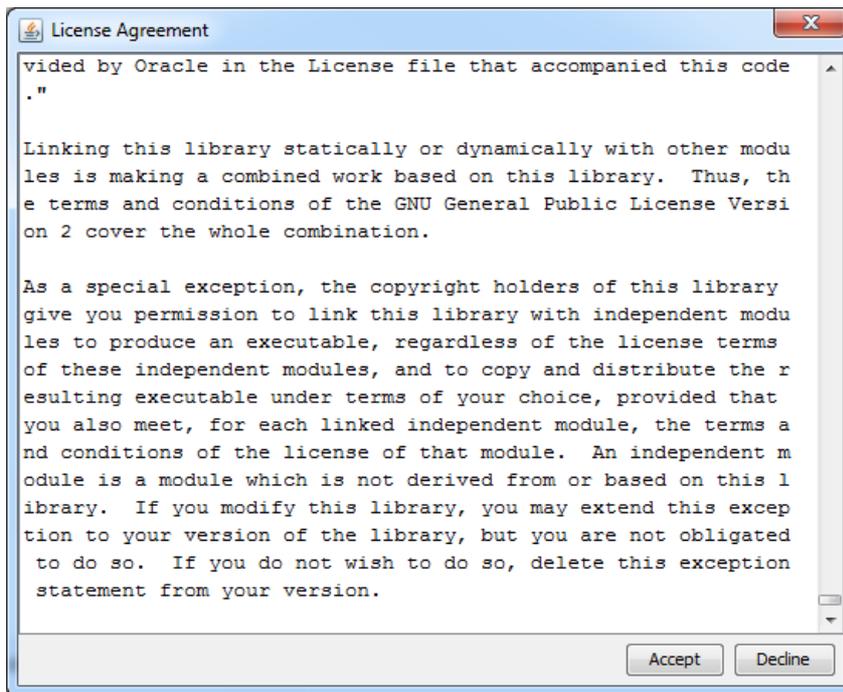
- Configurar Spring Security 3.1 para autenticarse contra un Active Directory
- Migración a ICEfaces 2.0
- Configuración de aplicaciones multientorno con Maven
- Primeros pasos con Scala
- Expression Language 2.2 en Tomcat 6: invocar a un método de un ManagedBean en JSF pasando parámetros.

Últimos Tutoriales del Autor

- Liferay 6: Cómo crear nuevos tipos de contenido



Descargaremos un jar que debemos ejecutar como se indica: `java -jar JAXB2_20110512.jar`



Si todo va bien se habrá descomprimido en el directorio `jaxb-ri-20110512` (en la misma ubicación del JAR)

4. Paso 2: Crear Proyecto Java en Eclipse

Ahora vamos a eclipse y creamos un nuevo proyecto java:

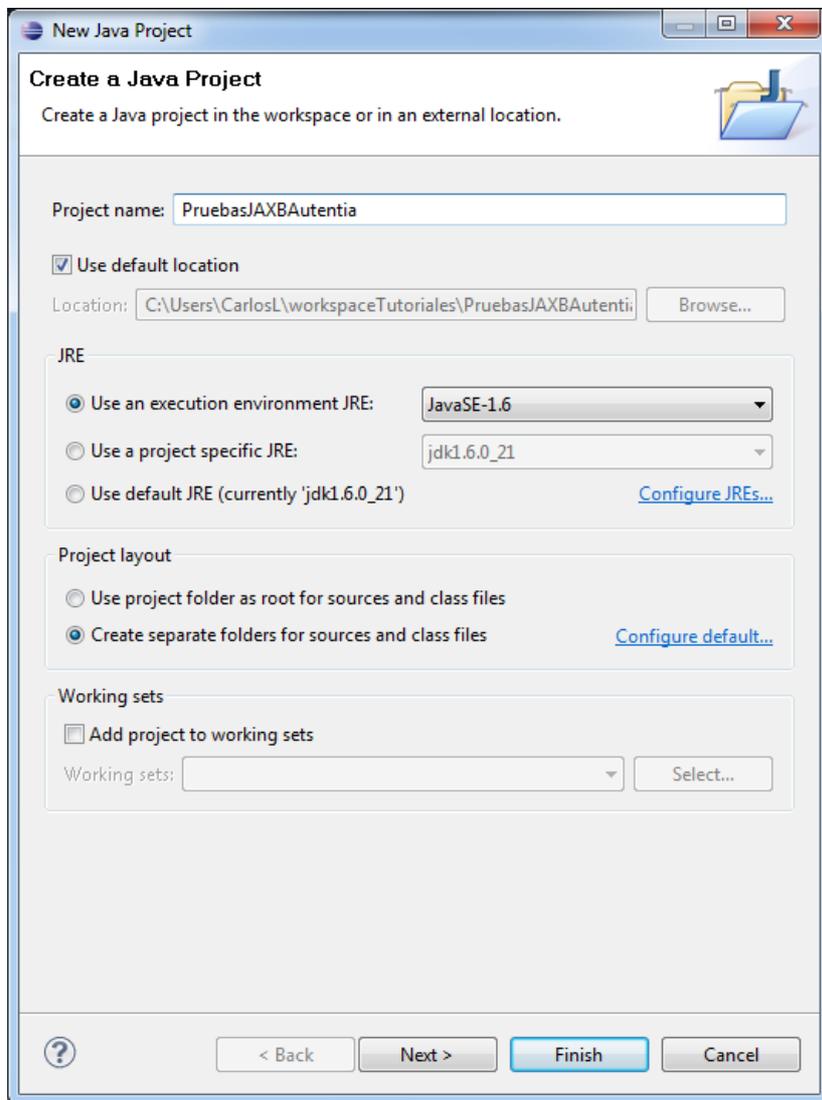
- Incluir Gadgets en Liferay 6.0.5: Cómo añadir Gadgets de forma sencilla
- Instalación de Apache Geronimo en Windows, e Integración con Eclipse Europa
- Profiling Java con Eclipse Test Performance Tools Platform (TPTP)
- Optimizando código Java con Eclipse Test Performance Tools Platform

Síguenos a través de:

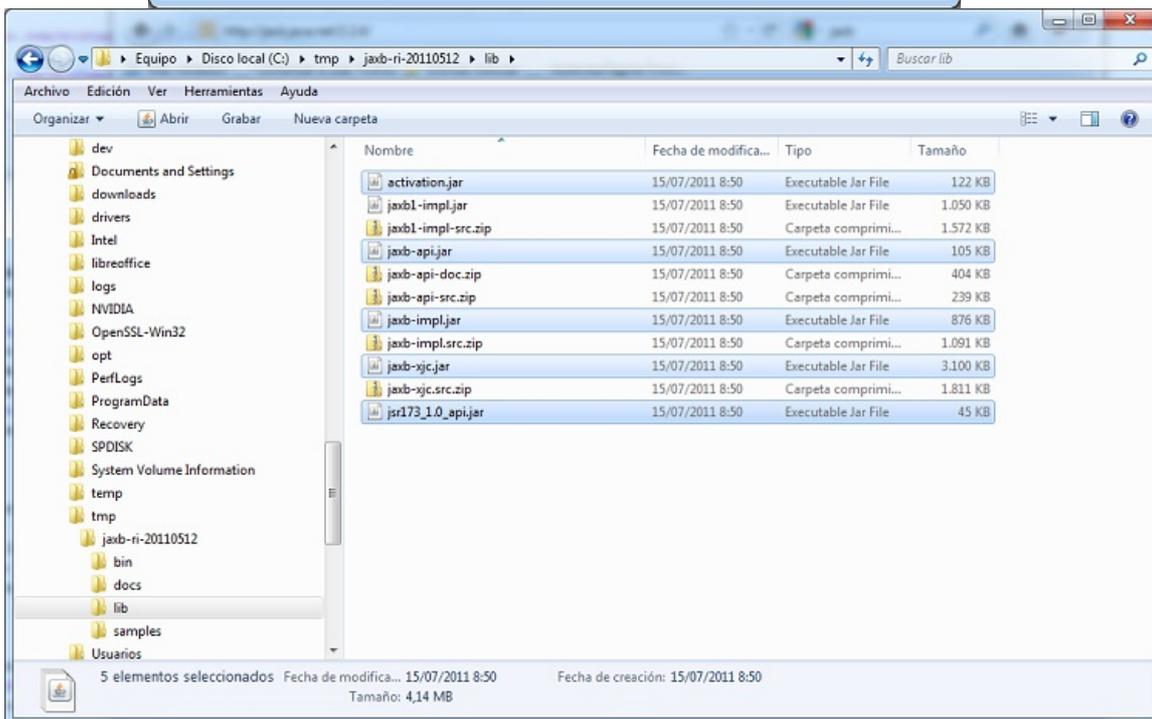
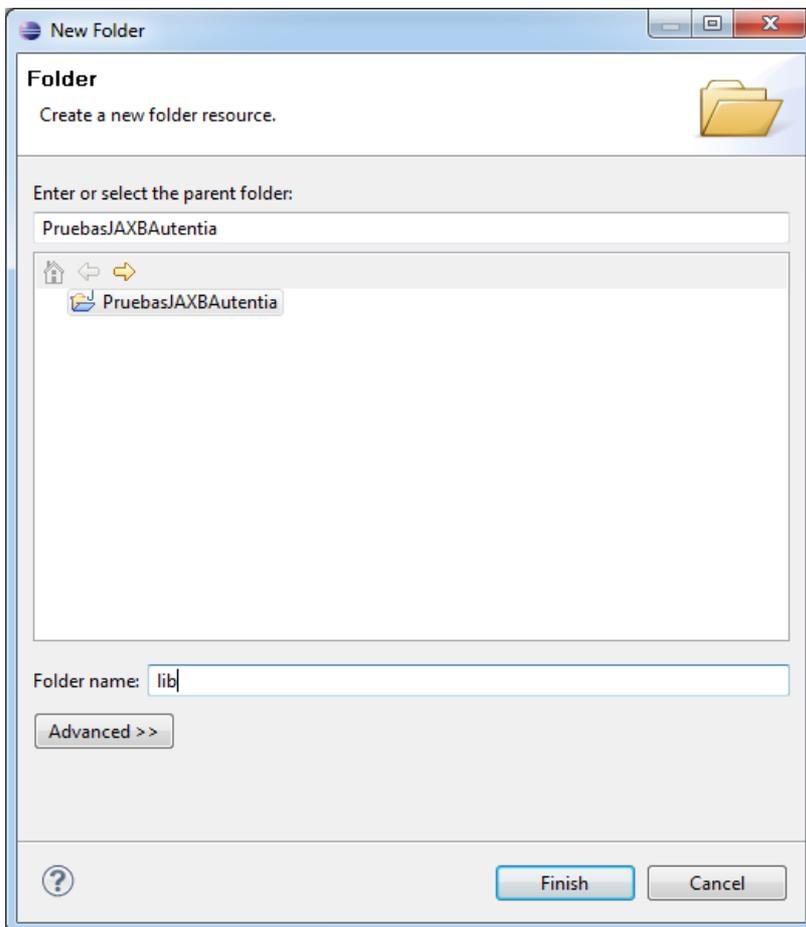


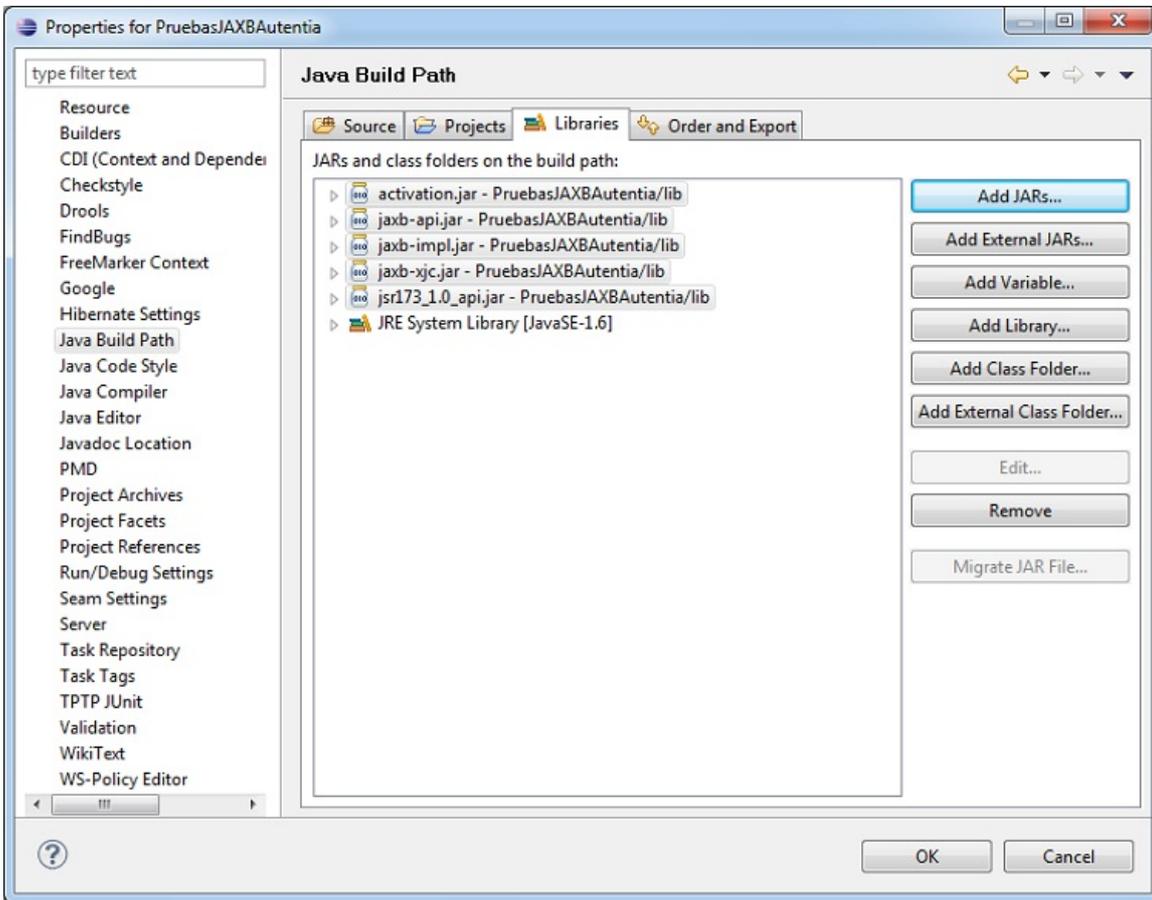
Últimas ofertas de empleo

- 2011-07-06
Otras Sin catalogar - LUGO.
- 2011-06-20
Comercial - Ventas - SEVILLA.
- 2011-05-24
Contabilidad - Especialista Contable - BARCELONA.
- 2011-05-14
Comercial - Ventas - TARRAGONA.
- 2011-04-13
Comercial - Ventas - VALENCIA.



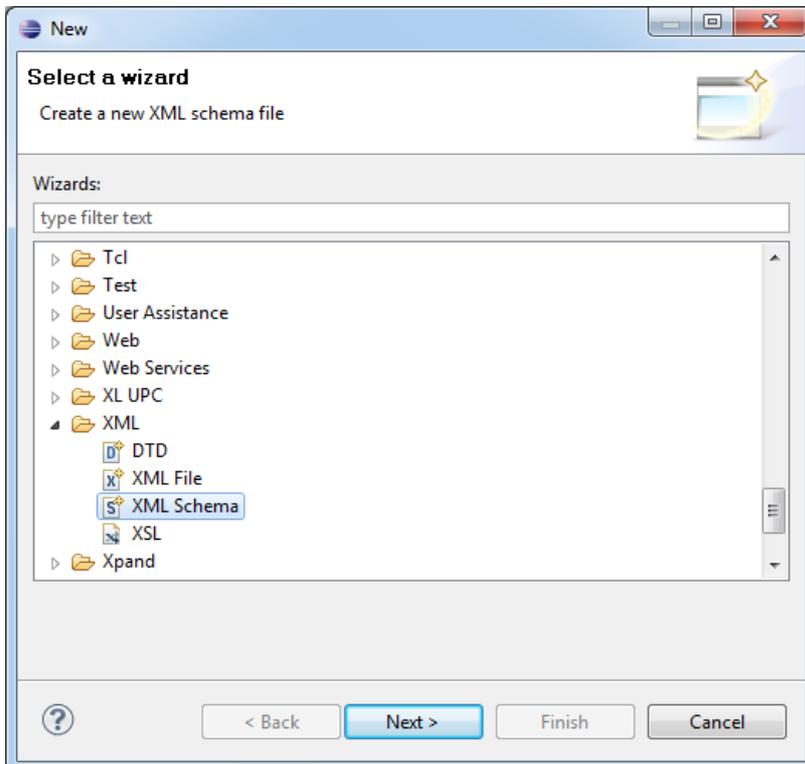
Dentro del proyecto creamos una carpeta llamada lib donde incluiremos las librerías de JAXB indicadas y las agregamos a nuestro buildpath de aplicación como se muestra en las imágenes siguientes:

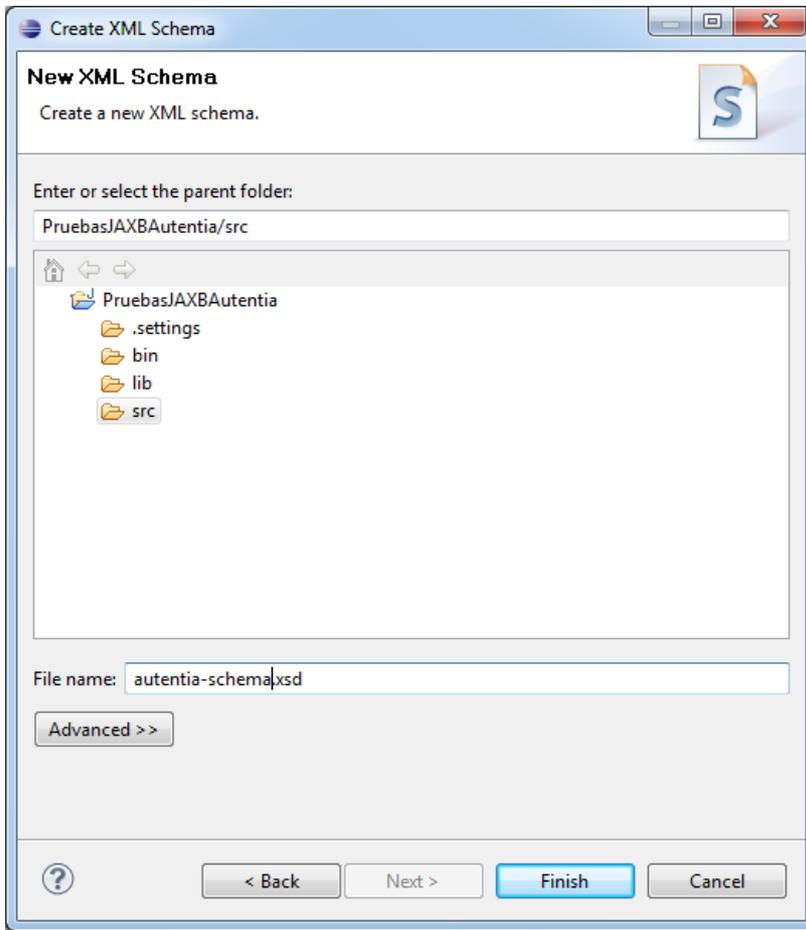




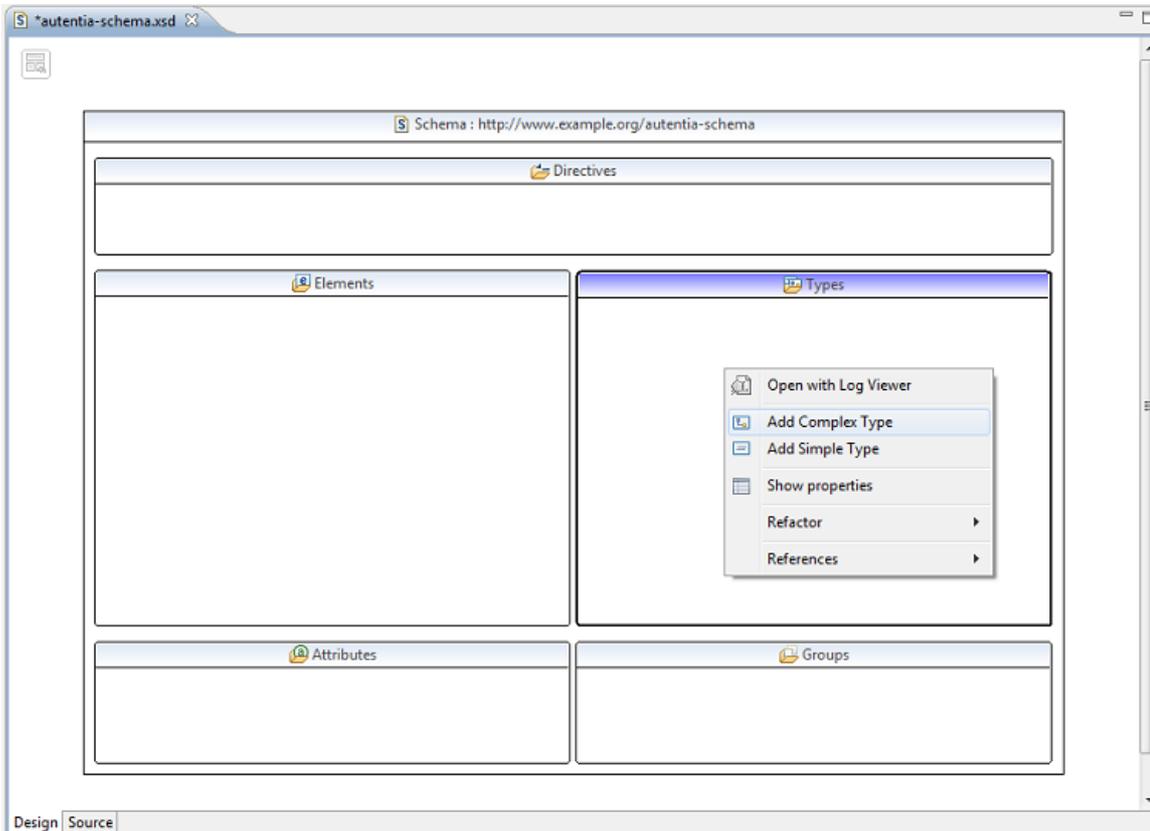
5. Paso 3: Creación de archivo XSD

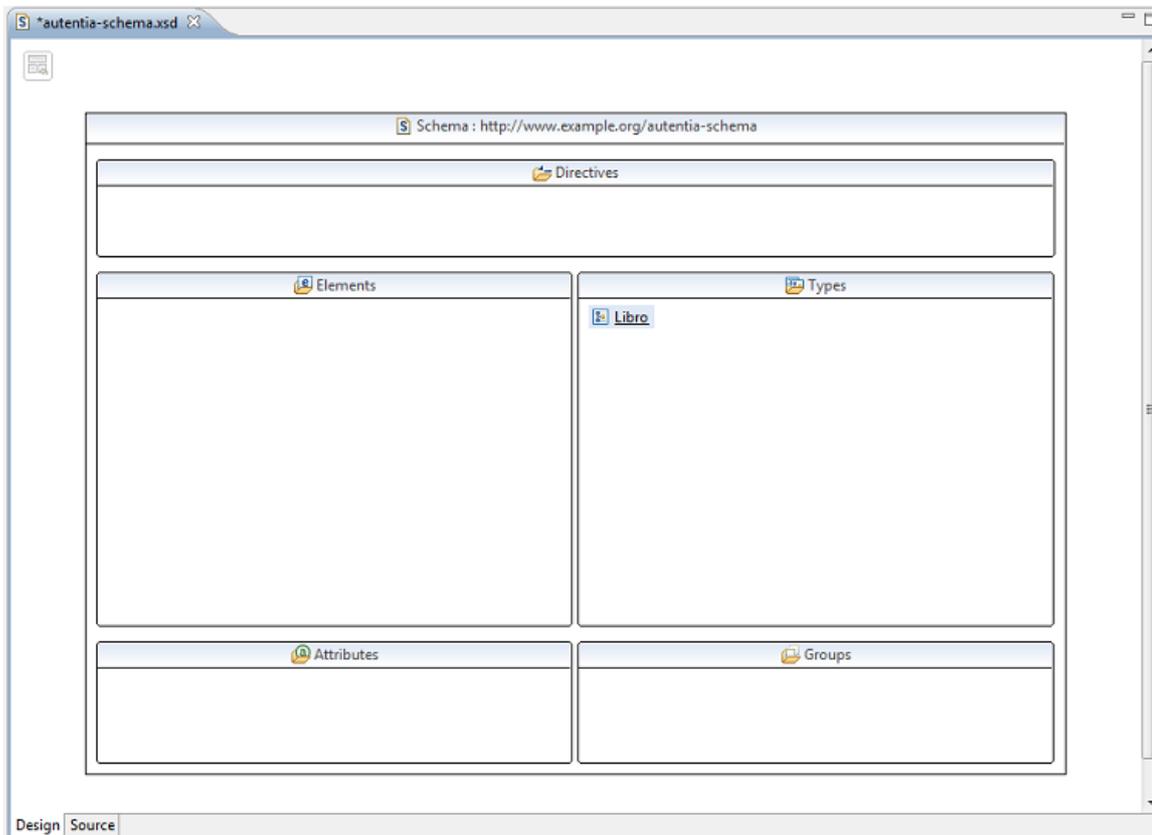
Ahora hacemos click derecho sobre la carpeta src y creamos un nuevo archivo XSD:



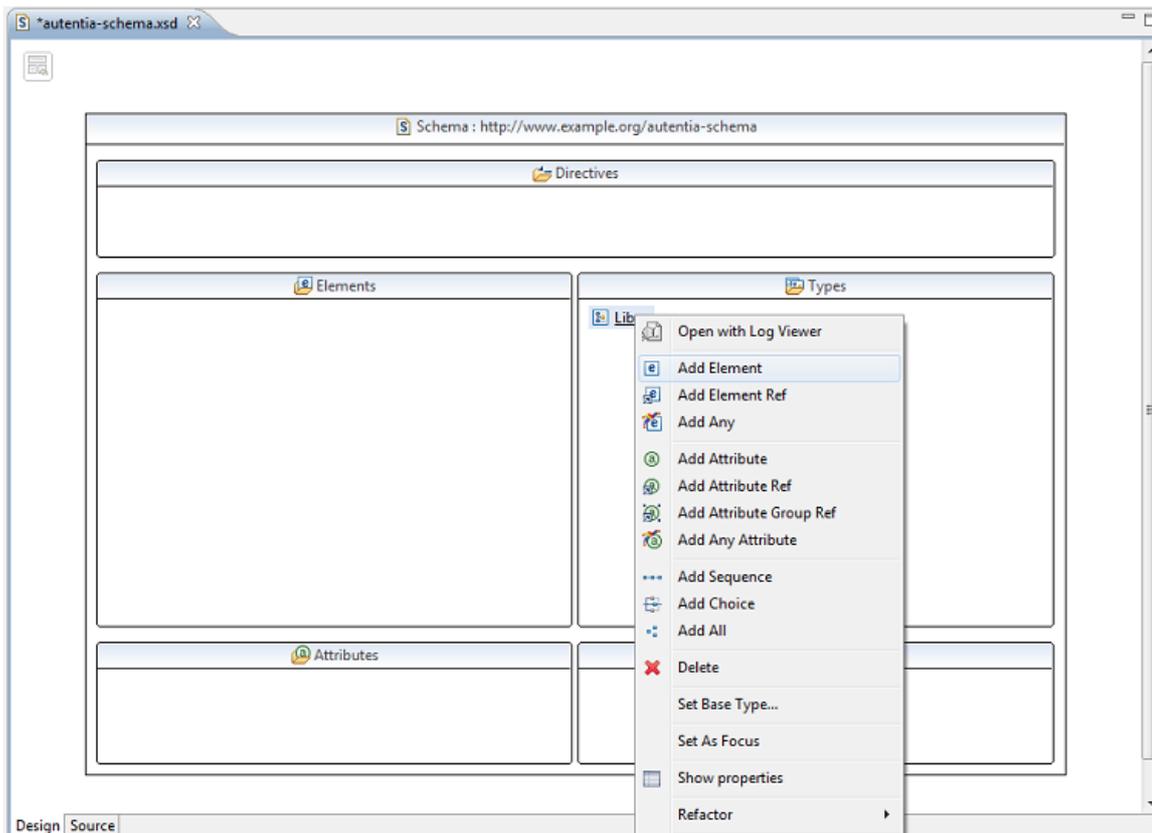


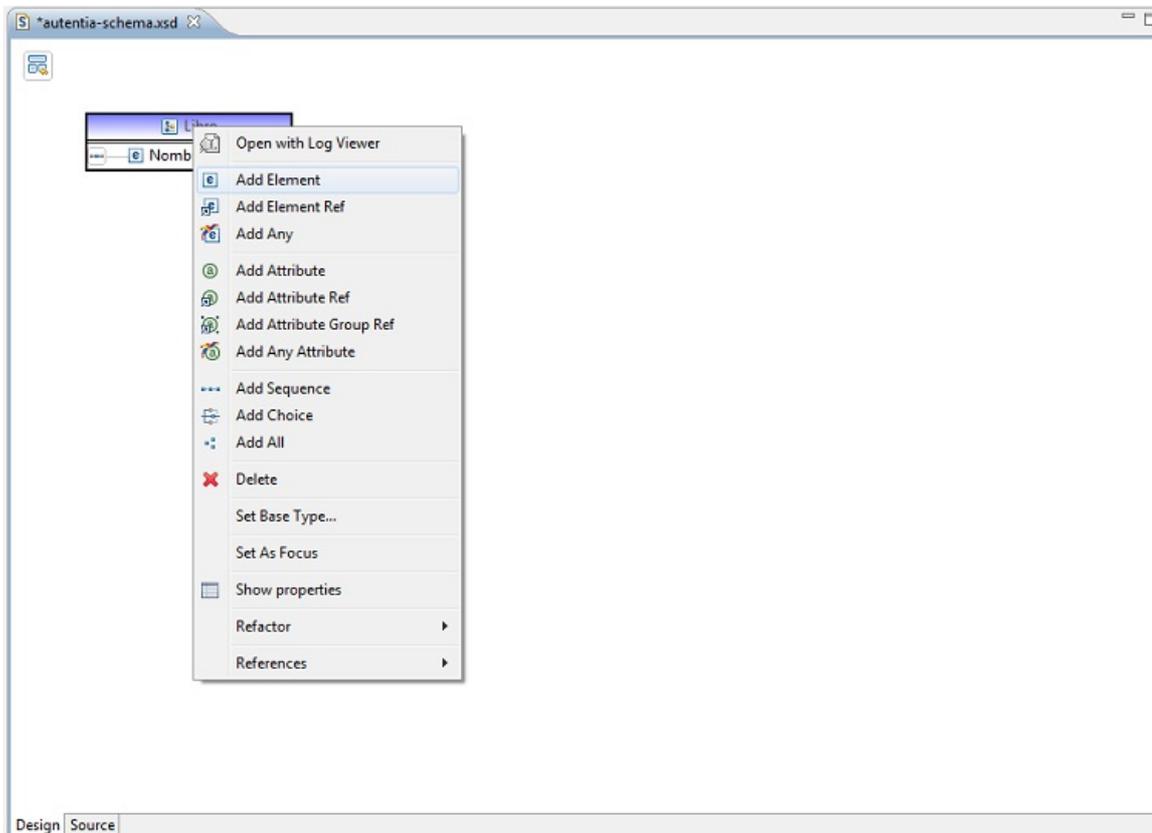
Una vez creado el XSD hacemos doble click y nos aparecerá la siguiente ventana donde comenzaremos a crear un nuevo tipo complejo que en nuestro caso se llamará Libro:



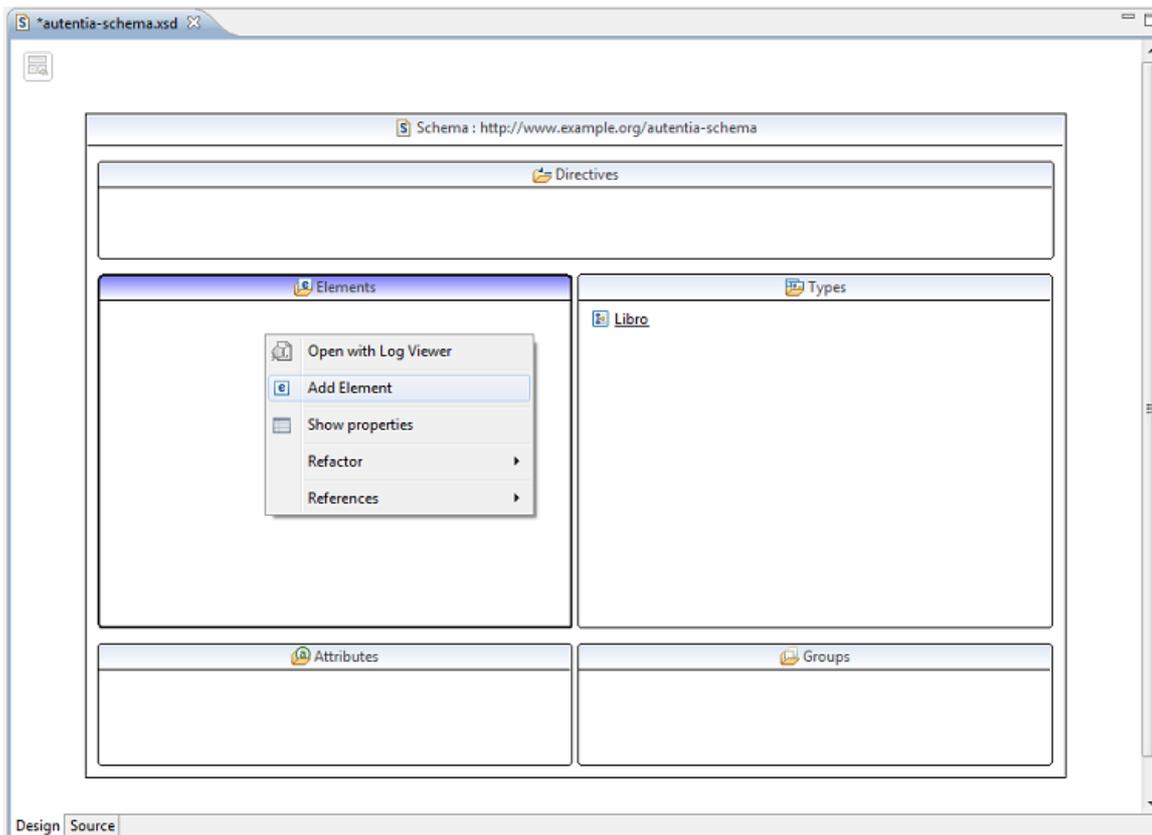


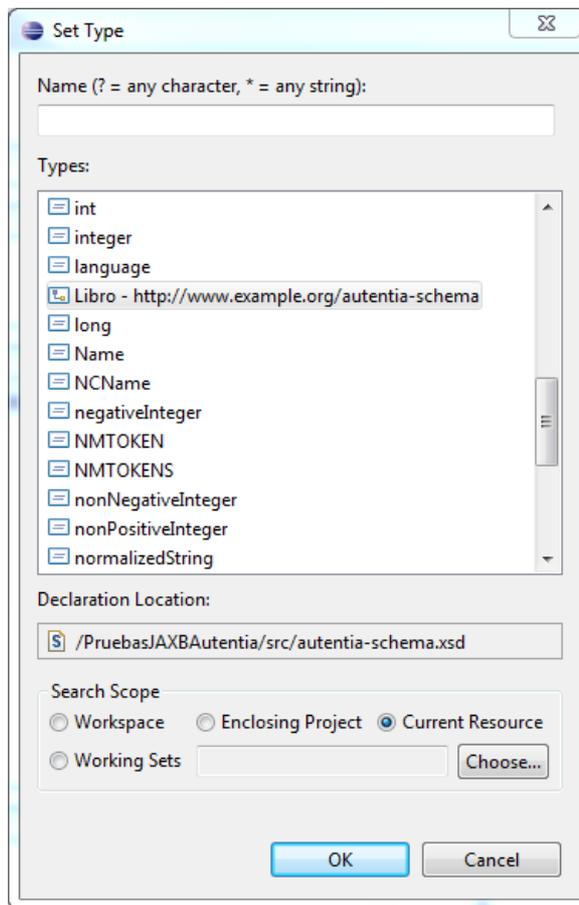
Ahora añadimos los elementos del tipo complejo haciendo click derecho sobre el tipo y pulsando sobre "Add Element", para el siguiente elemento nos cambiará la vista de diseño, pero la acción será la misma (click derecho...etc)





Ahora crearemos el elemento para almacenar el libro usando el tipo complejo creado en los pasos anteriores:



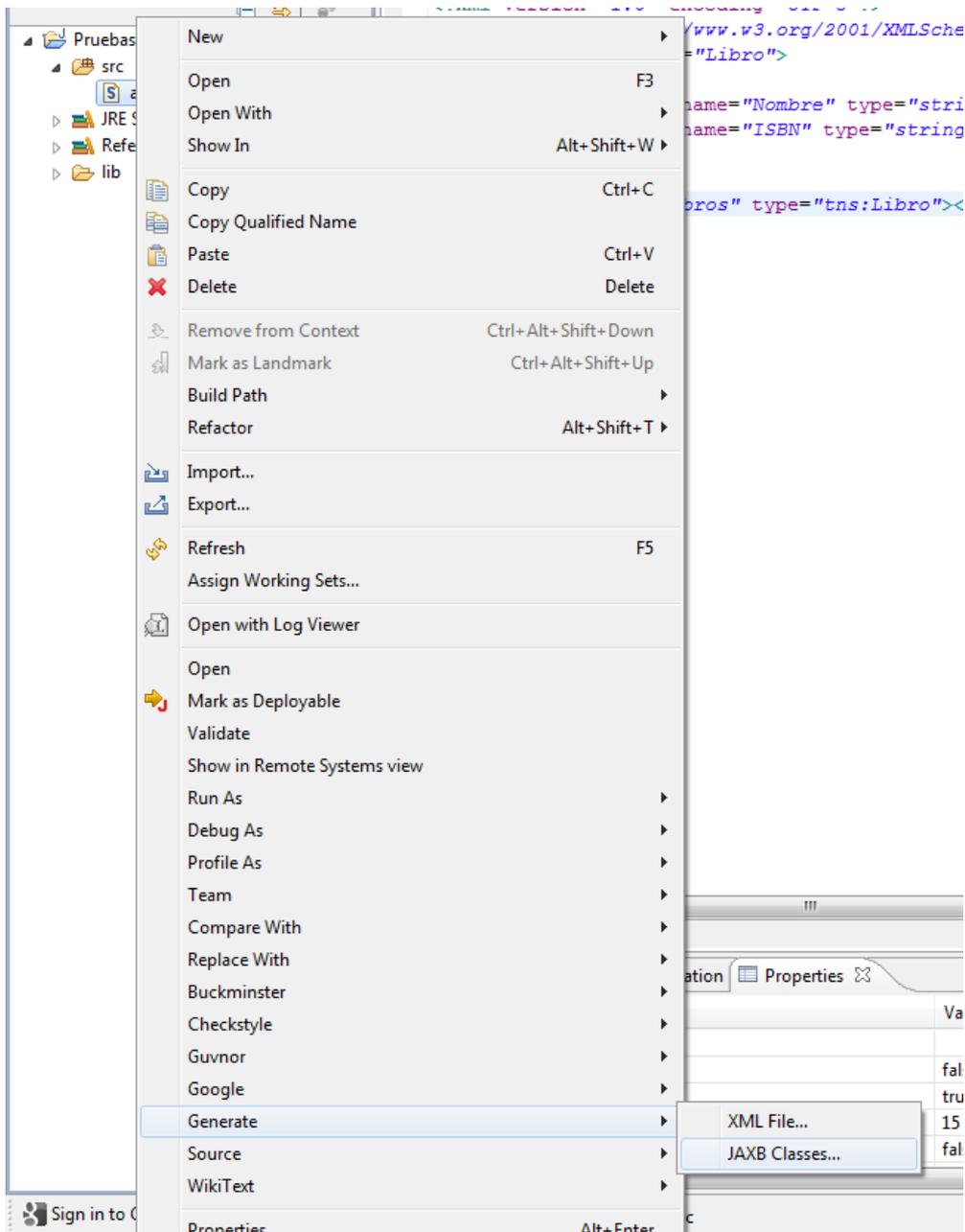


Si todo ha ido bien deberemos tener un xml parecido al siguiente:

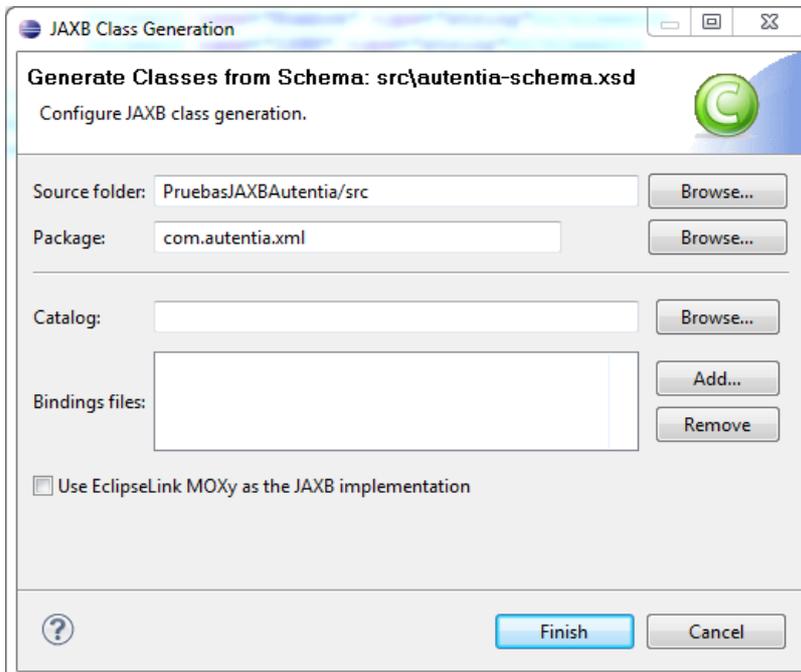
```
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <schema xmlns="http://www.w3.org/2001/XMLSchema"
03.       targetNamespace="http://www.example.org/autentia-schema"
04.       xmlns:tns="http://www.example.org/autentia-schema"
05.       elementFormDefault="qualified">
06.   <complexType name="Libro">
07.     <sequence>
08.       <element name="Nombre" type="string"/></element>
09.       <element name="ISBN" type="string"/></element>
10.     </sequence>
11.   </complexType>
12.   <element name="Libros" type="tns:Libro"/></element>
13. </schema>
```

6. Paso 4: Creación de clases accesoras

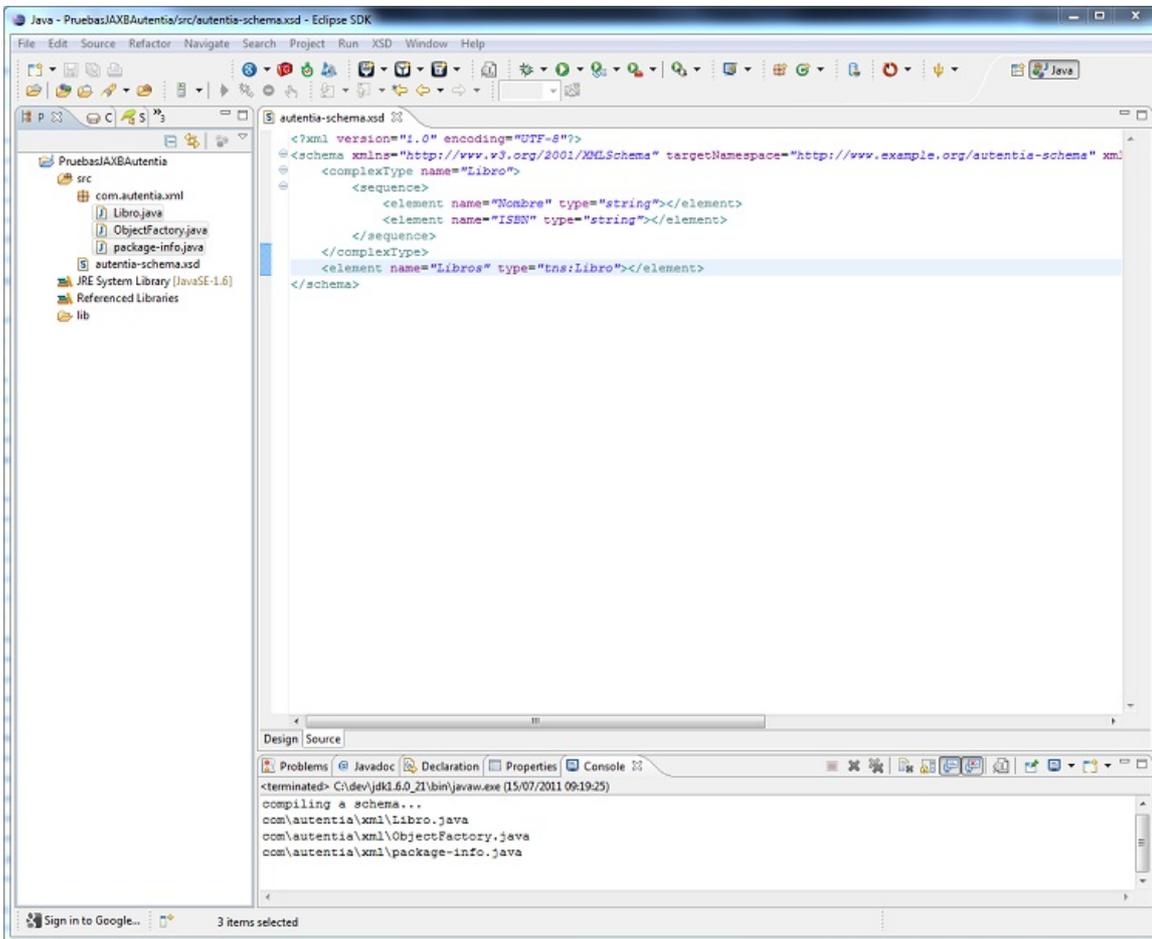
Ahora que tenemos definido el XSD haremos click derecho sobre este y pulsaremos en "Generate-->JAXB Classes..." como se muestra en la imagen:



En la siguiente pantalla solo tendremos que introducir el paquete donde queremos crear nuestras clases dentro de nuestro directorio de fuentes:

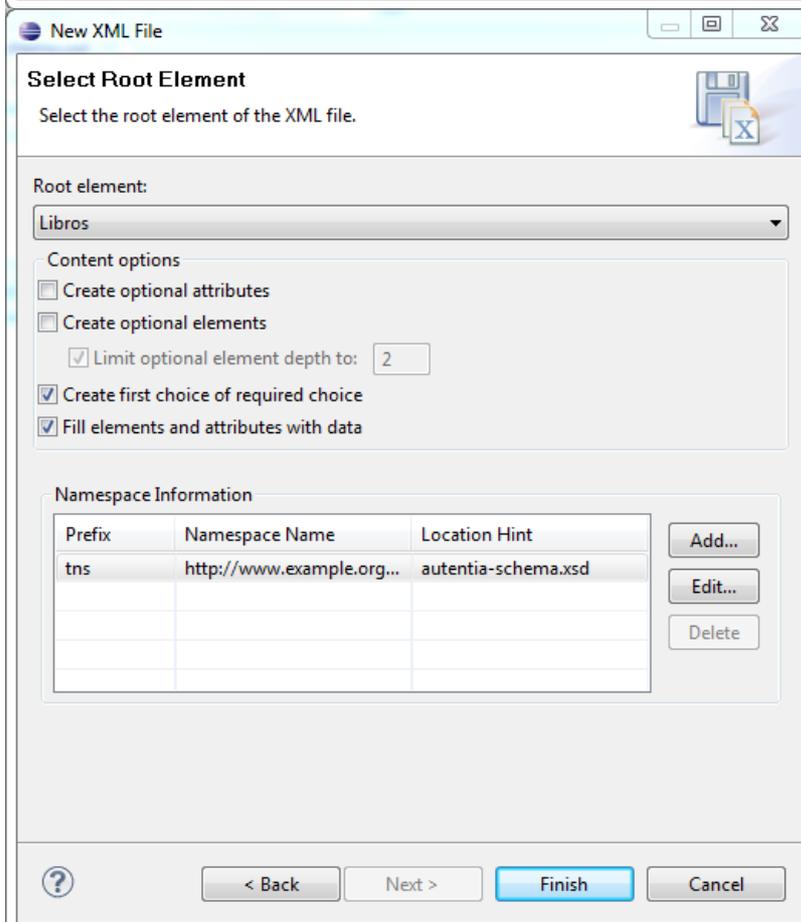
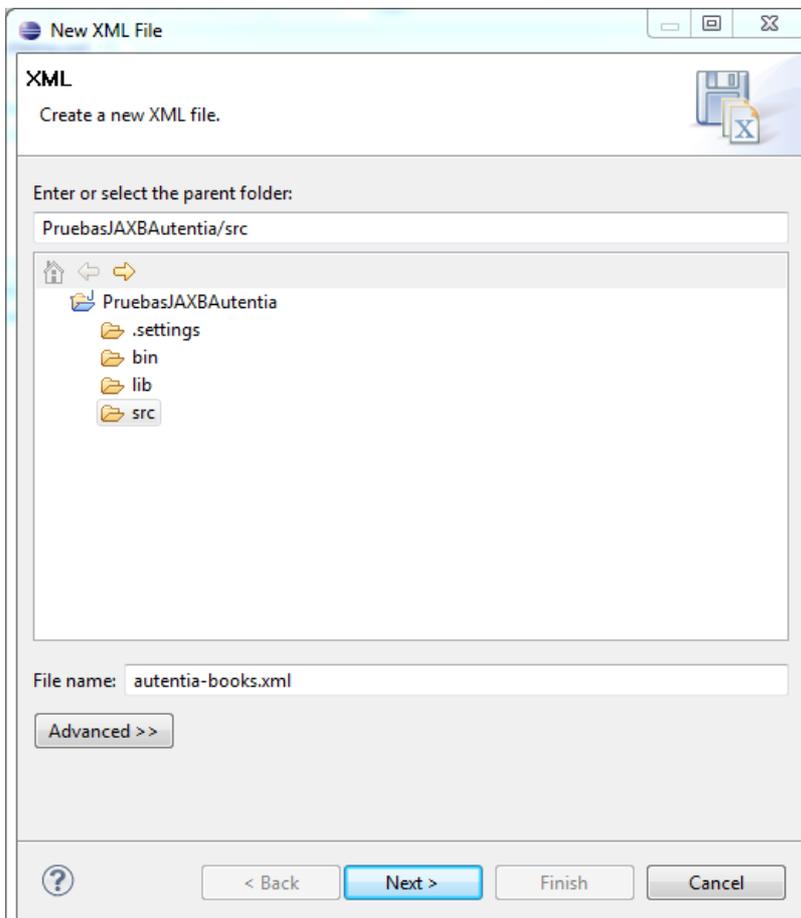


Si todo ha ido bien, veremos las clases incluidas en el paquete que hemos indicado previamente:



7. Paso 5: Creación de XML

Ahora volvemos a hacer click derecho sobre el XSD y pulsaremos en "Generate-->XML File..." como se muestra en la imagen:



Editamos el xml creado con lo siguiente:

```
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <tns:Libros xmlns:tns="http://www.example.org/autentia-schema"
03.           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04.           xsi:schemaLocation="http://www.example.org/autentia-schema autentia-
05.           schema.xsd ">
   <tns:Nombre>Libro1</tns:Nombre>
```

```
06. <tns:ISBN>ISBN1</tns:ISBN>
07. </tns:Libros>
```

8. Resultado

Por último creamos una clase ejemplo para obtener los datos del XML:

```
view plain print ?
01. import javax.xml.bind.JAXBContext;
02. import javax.xml.bind.JAXBElement;
03. import javax.xml.bind.JAXBException;
04. import javax.xml.bind.Unmarshaller;
05. import com.autentia.xml.Libro;
06. public class PruebaAutentia {
07.     public static void main(String[] args) {
08.         try {
09.             final JAXBContext jc = JAXBContext.newInstance(Libro.class.getPackage().getName(
10.                 final Unmarshaller u = jc.createUnmarshaller();
11.                 final JAXBElement<libro> menuElement = (JAXBElement<libro>)u.unmarshal(PruebaAut
books.xml"));
12.                 System.out.println("Nombre: " + menuElement.getValue().getNombre());
13.                 System.out.println("ISBN: " + menuElement.getValue().getISBN());
14.             } catch (JAXBException e) {
15.                 e.printStackTrace();
16.             }
17.         }
18.     }
19. </libro></libro>
```

Por último, la ejecución de esta clase nos debería como salida dar el nombre y el ISBN introducidos al crear el xml:

Nombre: Libro1

ISBN: ISBN1

6. Conclusiones

De esta forma se nos facilita la creación de esquemas xml al hacerlo mas visual, el nivel de complejidad irá aumentando según sea la definición del esquema, pero nos puede quedar mas claro utilizando estos mecanismos. Este método también es valido para crear las clases accesoras y hacer el Unmarshaller partiendo de un xsd ya definido anteriormente.

Con esto ya podemos leer nuestros xml de forma programática.

Cualquier duda o sugerencia podeis comentarlo.

Saludos.

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.

Enviar comentario

(Sólo para usuarios registrados)

>> **Regístrate** y accede a esta y otras ventajas <<

COMENTARIOS



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2011 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

[W3C XHTML 1.0](#) [W3C CSS](#) [XML RSS](#) [XML ATOM](#)