

Tipo : Enunciado de Laboratorio
Capítulo : Arquitecturas Web Services
Duración : 60 minutos

I. OBJETIVOS

Implementar una arquitectura Web Services según requerimiento.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Windows 7
- Eclipse
- Código base del laboratorio

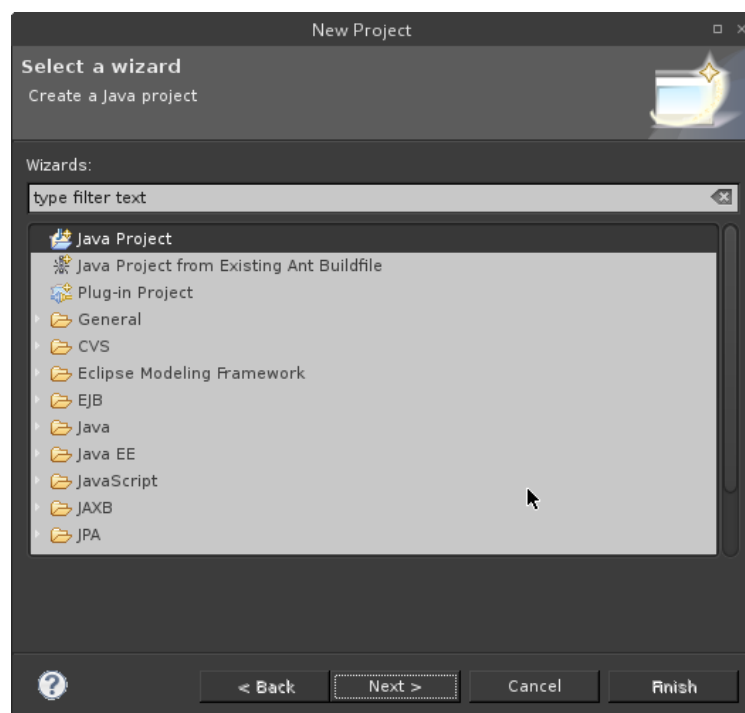
III. EJECUCIÓN DEL LABORATORIO

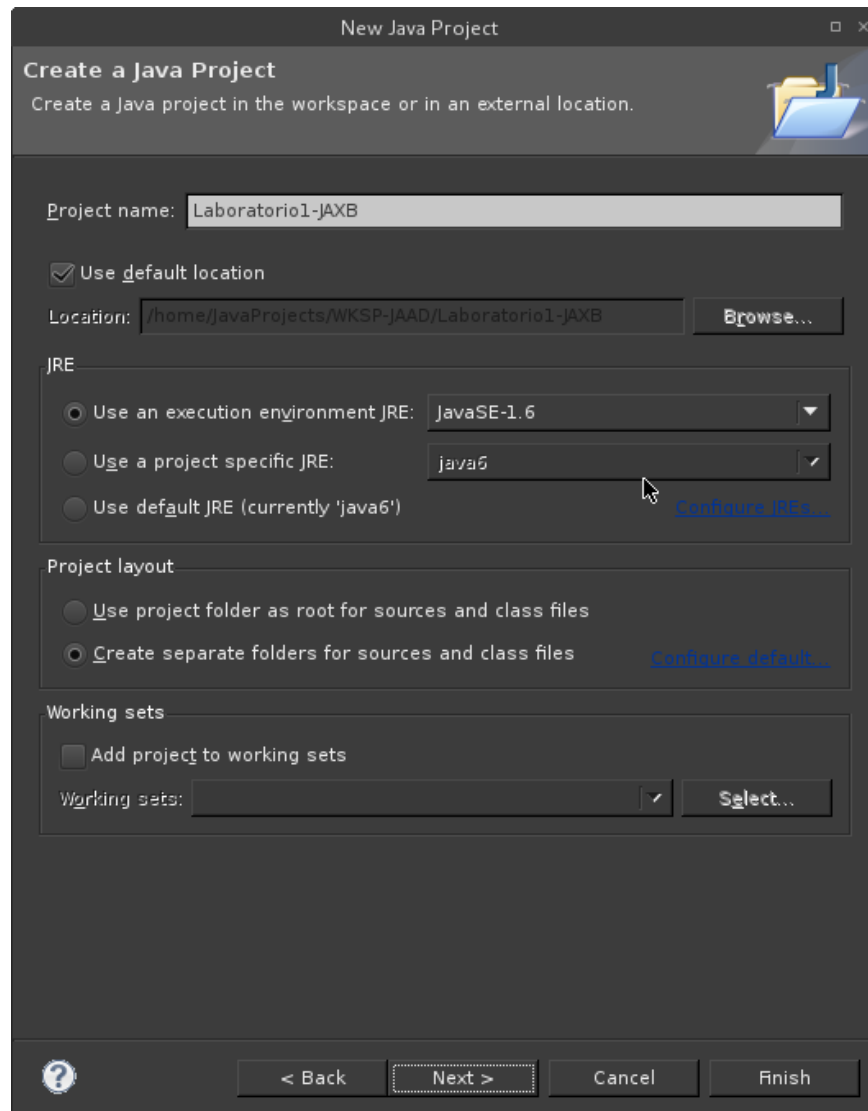
- **Ejercicio 2.1:** Entender las definiciones de servicios para exponer funcionalidades.

IV. EVALUACIÓN

1. Creación del proyecto Eclipse

Cree un nuevo proyecto en la IDE Eclipse con el nombre “**Laboratorio1-JAXB**” mediante las siguientes opciones: **File > New > Java Project**

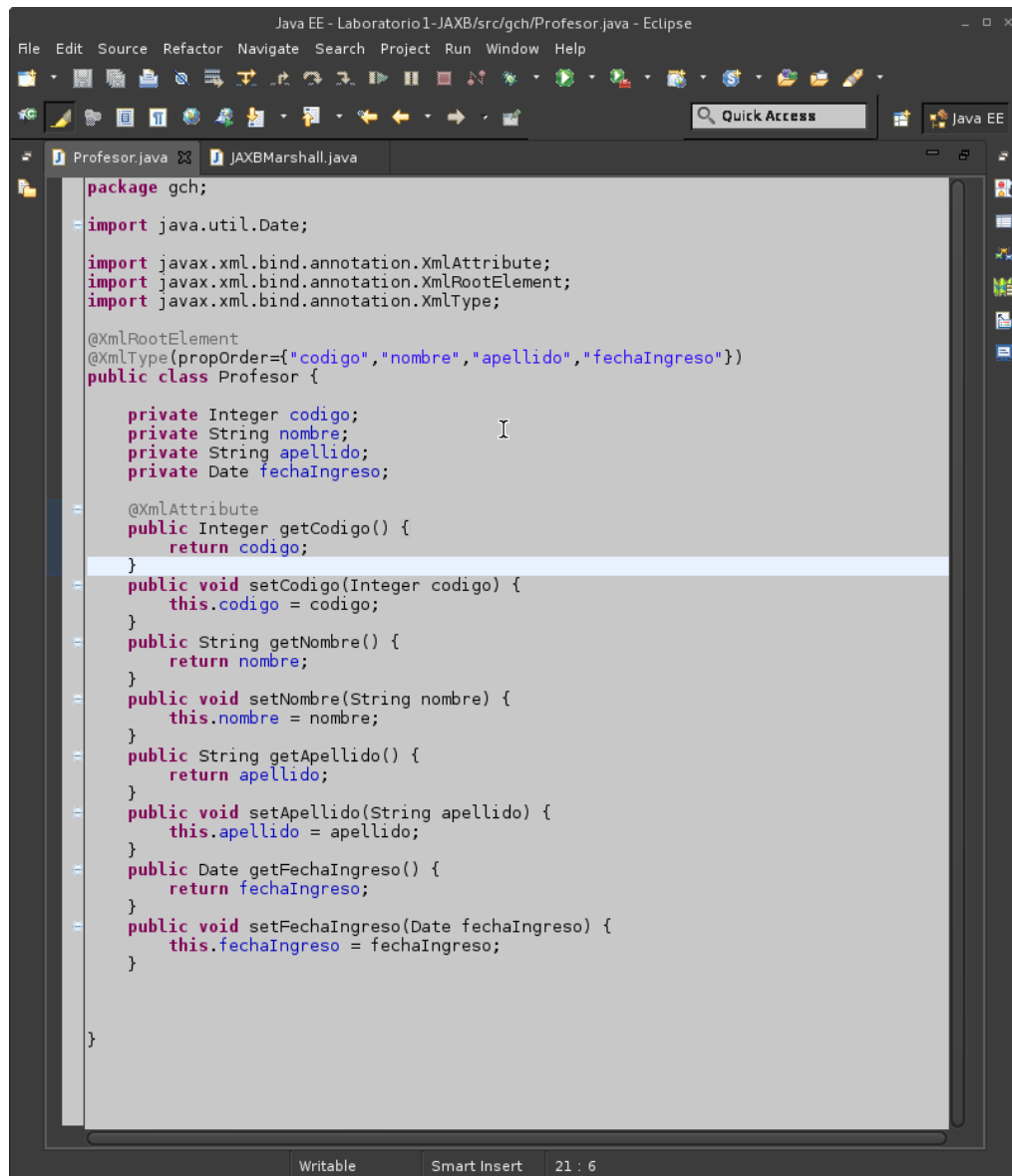




2. Defina las entidades mediante JAXB

Cree la entidad Profesor como clase Java indicando las siguientes anotaciones que son parte de la especificación JAXB:

- **@XmlRootElement:** Anotación utilizada para definir una clase Java como elemento raíz de un esquema XML.
- **@XmlType:** Anotación utilizada para definir detalles adicionales de una clase Java tales como namespace y orden de elementos XML.
- **@XmlElement:** Anotación utilizada para definir una propiedad de una clase Java como un elemento XML.
- **@XmlAttribute:** Anotación utilizada para definir una propiedad de una clase Java como atributo XML.



```
Java EE - Laboratorio1-JAXB/src/gch/Profesor.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

package gch;

import java.util.Date;

import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement
@XmlType(propOrder={"codigo","nombre","apellido","fechaIngreso"})
public class Profesor {

    private Integer codigo;
    private String nombre;
    private String apellido;
    private Date fechaIngreso;

    @XmlAttribute
    public Integer getCodigo() {
        return codigo;
    }

    public void setCodigo(Integer codigo) {
        this.codigo = codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public Date getFechaIngreso() {
        return fechaIngreso;
    }

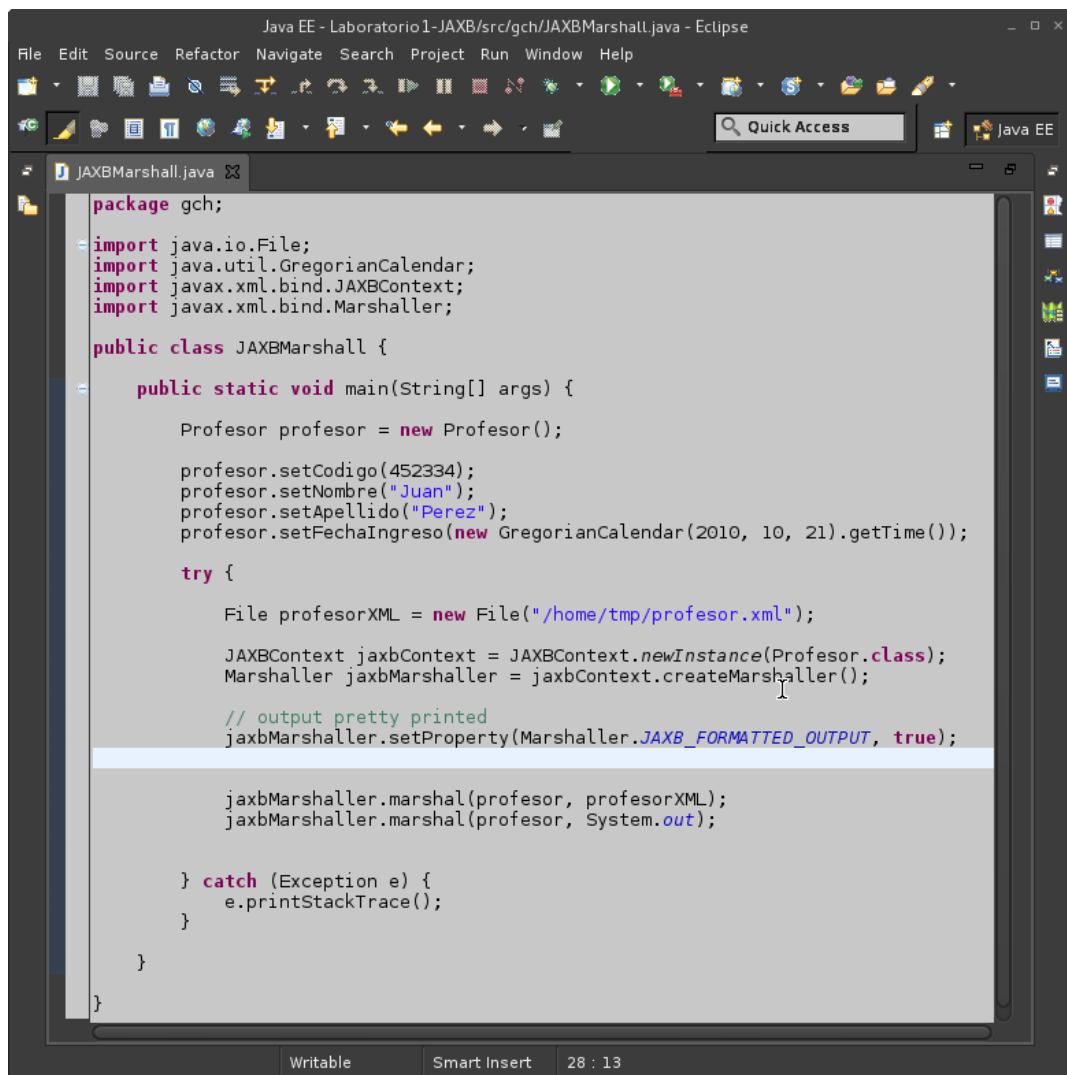
    public void setFechaIngreso(Date fechaIngreso) {
        this.fechaIngreso = fechaIngreso;
    }

}

Writable Smart Insert 21 : 6
```

3. Convierta la entidad JAXB hacia XML

Cree la clase Java para realizar el “marshalling” o la transformación de una entidad JAXB hacia un archivo XML (profesor.xml).



```
package gch;

import java.io.File;
import java.util.GregorianCalendar;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;

public class JAXBMarshall {

    public static void main(String[] args) {

        Profesor profesor = new Profesor();

        profesor.setCodigo(452334);
        profesor.setNombre("Juan");
        profesor.setApellido("Perez");
        profesor.setFechaIngreso(new GregorianCalendar(2010, 10, 21).getTime());

        try {

            File profesorXML = new File("/home/tmp/profesor.xml");

            JAXBContext jaxbContext = JAXBContext.newInstance(Profesor.class);
            Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

            // output pretty printed
            jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

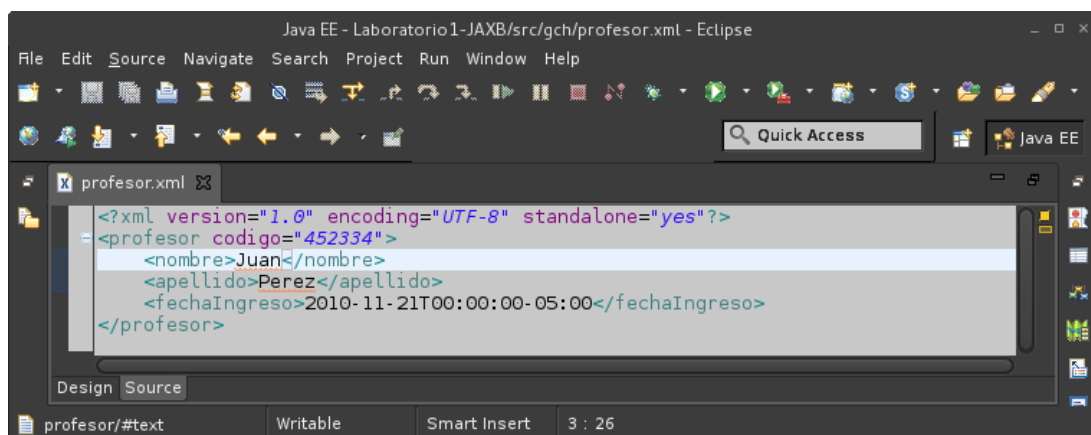
            jaxbMarshaller.marshal(profesor, profesorXML);
            jaxbMarshaller.marshal(profesor, System.out);

        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```

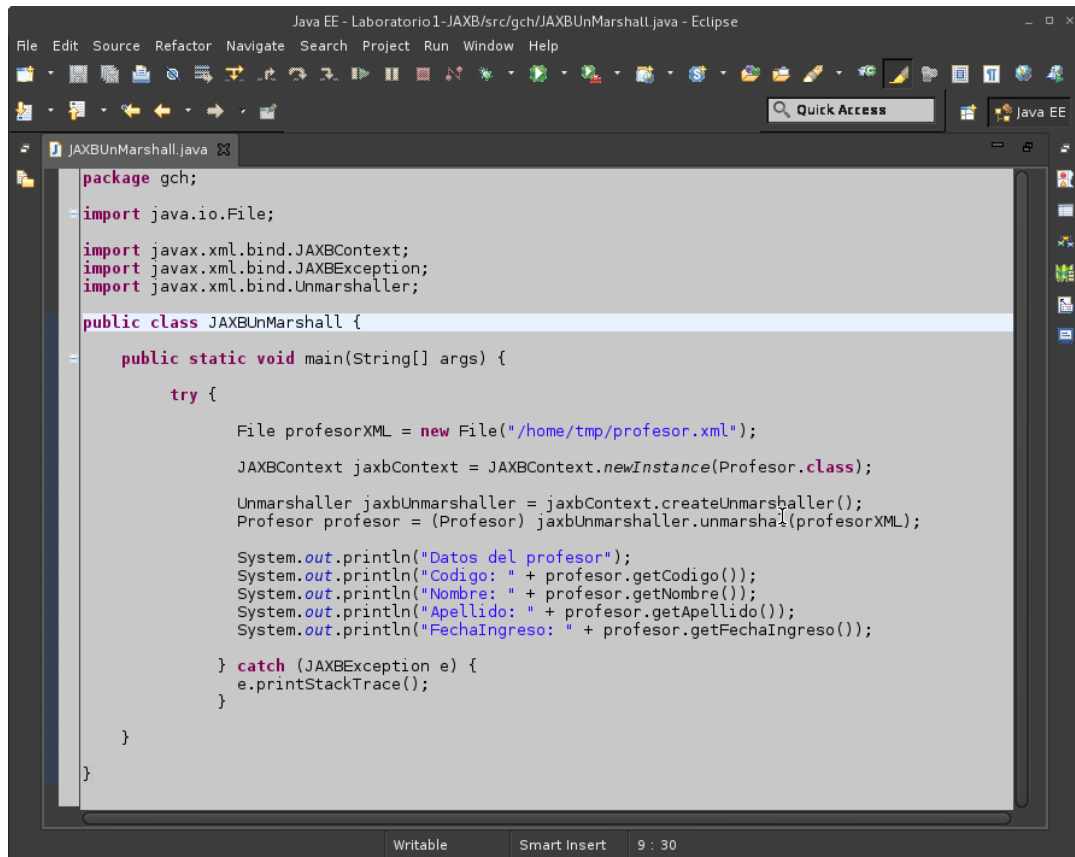
Al abrir el archivo XML y revisar su contenido podremos visualizar que es el mismo que se definió en el objeto JAXB.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profesor codigo="452334">
  <nombre>Juan</nombre>
  <apellido>Perez</apellido>
  <fechaIngreso>2010-11-21T00:00:00-05:00</fechaIngreso>
</profesor>
```

4. Convierta la entidad en formato XML hacia un objeto JAXB

Cree la clase Java para realizar el “unmarshalling” o la transformación de un archivo XML (profesor.xml) hacia un objeto JAXB.



```
package gch;

import java.io.File;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;

public class JAXBUnMarshall {

    public static void main(String[] args) {

        try {

            File profesorXML = new File("/home/tmp/profesor.xml");

            JAXBContext jaxbContext = JAXBContext.newInstance(Profesor.class);

            Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
            Profesor profesor = (Profesor) jaxbUnmarshaller.unmarshal(profesorXML);

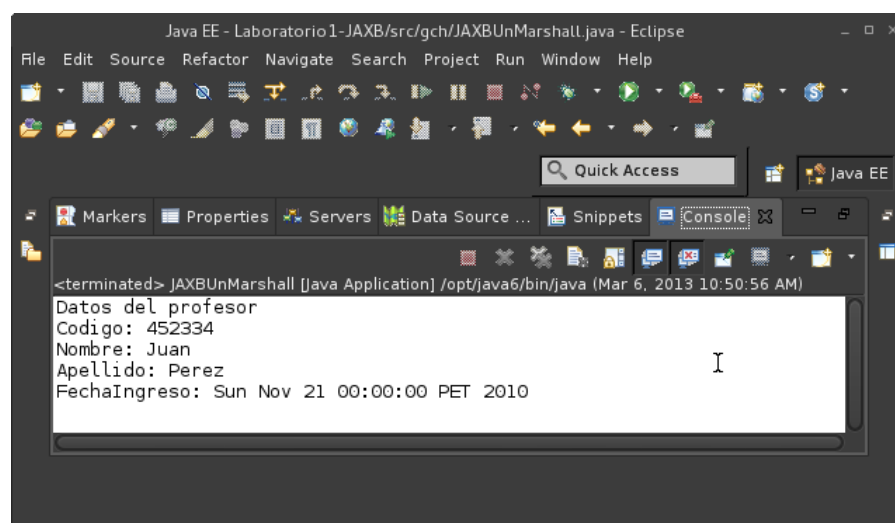
            System.out.println("Datos del profesor");
            System.out.println("Codigo: " + profesor.getCodigo());
            System.out.println("Nombre: " + profesor.getNombre());
            System.out.println("Apellido: " + profesor.getApellido());
            System.out.println("FechaIngreso: " + profesor.getFechaIngreso());

        } catch (JAXBException e) {
            e.printStackTrace();
        }

    }

}
```

La consola del Eclipse nos mostrará el contenido de las propiedades del objeto Java leído desde el archivo XML mediante JAXB.



```
<terminated> JAXBUnMarshall [Java Application] /opt/java6/bin/java (Mar 6, 2013 10:50:56 AM)
Datos del profesor
Codigo: 452334
Nombre: Juan
Apellido: Perez
FechaIngreso: Sun Nov 21 00:00:00 PET 2010
```