



# JAVA WEB SERVICES



**JAXB**

**ERIC GUSTAVO CORONEL CASTILLO**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)



# Contenido

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>2</b>	<b>LEER DOCUMENTO XML.....</b>	<b>4</b>
2.1	DOCUMENTO XML .....	4
2.2	CLASE LIBRERIA.JAVA .....	5
2.3	CLASE LIBRO .....	6
2.4	CLASE DE PRUEBA .....	7
<b>3</b>	<b>GENERAR DOCUMENTO XML.....</b>	<b>8</b>
3.1	CLASE PRODUCTO.JAVA.....	8
3.2	CLASE DE PRUEBA .....	9
<b>4</b>	<b>PRACTICA DE LABORATORIO .....</b>	<b>10</b>
4.1	CONSULTA DE DATOS .....	10
4.2	PROCESAR DATOS .....	10
4.3	RETO .....	10



# 1 INTRODUCCIÓN

JAXB es una tecnología Java que permite mapear clases de Java a representaciones XML. JAXB provee dos funciones fundamentales:

1. **marshal.** Consiste en presentar un objeto Java en XML. Java Object → XML
2. **unmarshal.** Consiste en presentar XML en un objeto Java. XML → Java Object

La versión actual trabaja con anotaciones con las cuáles se indica cuál será la raíz del documento XML, los elementos y los atributos. Las principales anotaciones son:

- @XmlRootElement
- @XmlElement
- @XmlAttribute



## 2 LEER DOCUMENTO XML

Para la demostración se estará desarrollando un ejemplo para leer un archivo XML de nombre librería.xml.

### 2.1 Documento XML

El nombre del archivo sería librería.xml.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<libreria>
  <nombre>LIBRERIA TECNICA</nombre>
  <libros>
    <libro isbn="3478653245985">
      <titulo>Fundamentos de Programación</titulo>
      <autor>Ricardo Marcelo</autor>
    </libro>
    <libro isbn="7843569823659">
      <titulo>Curso de programación java web</titulo>
      <autor>Gustavo Coronel</autor>
    </libro>
    <libro isbn="6578430965237">
      <titulo>Mineria de Datos</titulo>
      <autor>Sergio Matsukawa</autor>
    </libro>
  </libros>
</libreria>
```



## 2.2 Clase Libreria.java

```
package lecturaxml;

import java.util.ArrayList;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlElementWrapper;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement(name = "libreria")
@XmlType(propOrder = {"nombre", "libros"})
public class Libreria {

    private String nombre;
    private ArrayList<Libro> libros = new ArrayList();

    public Libreria() {
    }

    @XmlElement(name = "nombre")
    public String getNombre() {
        return nombre;
    }

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

    @XmlElementWrapper(name = "libros")
    @XmlElement(name = "libro")
    public ArrayList<Libro> getLibros() {
        return libros;
    }

    public void setLibros(ArrayList<Libro> libros) {
        this.libros = libros;
    }
}
```



## 2.3 Clase Libro

```
package lecturaxml;

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

@XmlRootElement(name = "libro")
@XmlType(propOrder = {"isbn", "titulo", "autor"})
public class Libro {

    private String isbn;
    private String titulo;
    private String autor;

    public Libro() {
    }

    @XmlAttribute(name = "isbn")
    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    @XmlElement(name = "titulo")
    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    @XmlElement(name = "autor")
    public String getAutor() {
        return autor;
    }

    public void setAutor(String autor) {
        this.autor = autor;
    }
}
```



## 2.4 Clase de Prueba

```
package lecturaxml;

import java.io.File;
import java.util.ArrayList;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;

public class LecturaXML {

    public static void main(String[] args) throws JAXBException {

        JAXBContext context = JAXBContext.newInstance(Libreria.class);
        Unmarshaller unmarshaller = context.createUnmarshaller();
        File file = new File("libreria.xml");
        Libreria libreria = (Libreria) unmarshaller.unmarshal(file);

        System.out.println("Nombre: " + libreria.getNombre());

        ArrayList<Libro> libros = libreria.getLibros();

        for (Libro l : libros) {
            System.out.println(l.getIsbn() + " | "
                + l.getTitulo() + " | " + l.getAutor());
        }

    }

}
```

La ejecución debe dar el siguiente resultado:

```
run:
Nombre: LIBRERIA TECNICA
3478653245985 | Fundamentos de Programación | Ricardo Marcelo
7843569823659 | Curso de programación java web | Gustavo Coronel
6578430965237 | Minería de Datos | Sergio Matsukawa
```



## 3 GENERAR DOCUMENTO XML

### 3.1 Clase Producto.java

```
package generaxml;

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

@XmlRootElement
public class Producto {

    private int id;
    private String nombre;
    private double precio;

    public Producto() {
    }

    public Producto(int id, String nombre, double precio) {
        this.id = id;
        this.nombre = nombre;
        this.precio = precio;
    }

    public int getId() {
        return id;
    }

    @XmlAttribute
    public void setId(int id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

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

    public double getPrecio() {
        return precio;
    }

    @XmlElement
    public void setPrecio(double precio) {
        this.precio = precio;
    }
}
```





```
}  
  
}
```

## 3.2 Clase de Prueba

```
package generaxml;  
  
import java.io.File;  
import javax.xml.bind.JAXBContext;  
import javax.xml.bind.JAXBException;  
import javax.xml.bind.Marshaller;  
import javax.xml.bind.PropertyException;  
  
public class GeneraXML {  
  
    public static void main(String[] args) throws PropertyException, JAXBException {  
  
        Producto producto = new Producto(1000, "Refrigeradora", 4790.00);  
  
        File file = new File("E:\\\\producto.xml");  
        JAXBContext jaxbContext = JAXBContext.newInstance(Producto.class);  
        Marshaller jaxbMarshaller = jaxbContext.createMarshaller();  
  
        jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);  
  
        // Escribir en el archivo  
        jaxbMarshaller.marshal(producto, file);  
        // Salida por consola  
        jaxbMarshaller.marshal(producto, System.out);  
  
    }  
}
```

Salida por consola:

```
run:  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<producto id="1000">  
    <nombre>Refrigeradora</nombre>  
    <precio>4790.0</precio>  
</producto>
```



---

## 4 Practica de Laboratorio

Para esta práctica de laboratorio se utilizara la base de datos Eureka.

### 4.1 Consulta de Datos

Desarrollar una aplicación que permita consultar los movimientos de una cuenta y permita generar el resultado en un archivo XML.

### 4.2 Procesar Datos

Hacer una aplicación que permita leer los datos de clientes desde un archivo XML y los inserte en la base de datos, si esta duplicado genere un archivo XML con los registros duplicados.

### 4.3 Reto

Desarrollar un ejemplo para generar un documento tipo Cabecera-Detalle, por ejemplo, una factura o boleta.