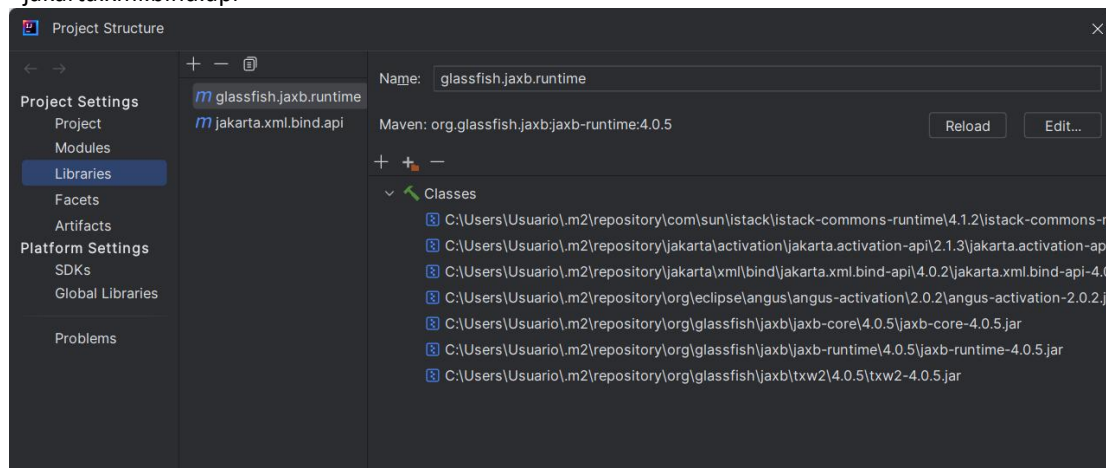


# SERIALIZACIÓN CON JABX

## 1. Implementación librerías jaxb

Añadir las librerías necesarias para JDK 20 desde opciones en la Estructura del proyecto las librerías:

- Glassfish.jaxb.runtime
- jakarta.xml.bind.api



## 2. Investigación de Mejores Prácticas con JAXB

### • Uso de Anotaciones:

- `@XmlRootElement`: Define la clase raíz para la serialización XML.
- `@XmlElement`: Marca los atributos o propiedades que serán elementos XML. Opcionalmente, puedes usar `@XmlElement (name="nombrePersonalizado")` para modificar el nombre en el XML.
- `@XmlAttribute`: Para atributos XML (si deseas que algunos campos se representen como atributos en vez de elementos).
- `@XmlTransient`: Excluye ciertos campos de la serialización.
- `@XmlAccessorType`: Permite especificar el modo de acceso a los campos y métodos (FIELD, PROPERTY, etc.).

### • Manejo de Colecciones y Tipos Complejos:

- Para colecciones como `List`, usa `@XmlElementWrapper` para definir un contenedor alrededor de los elementos y `@XmlElement` para los elementos individuales de la lista.
- Los tipos complejos deben ser anotados con `@XmlRootElement` o `@XmlType` para estructurar adecuadamente las clases anidadas.

- **Estrategias para Manejar Herencia:**

- Utiliza `@XmlSeeAlso` en las clases base para listar clases derivadas. Esto ayuda a JAXB a reconocer subtipos en la serialización.
- `@XmlElement` puede ser usado para especificar el nombre de la clase hija en el XML.

## 3. Creación clases y métodos

Mirar en las clases: Estudiante, Curso y JAXBUtil.

## 4. Depuración

Método marshal (Estudiante)

```
public class JAXBUtil {
    // Crea un archivo XML con las propiedades privadas de un solo estudiante y lo escribe
    // en el archivo pasado como parámetro
    1 usage
    public static void marshal(Estudiante estudiante, File file) throws JAXBException {
        // Inicializar contexto JAXB para la clase Estudiante.
        JAXBContext context = JAXBContext.newInstance(Estudiante.class);
        // Crear un marshaller (Marshaller):
        // El marshaller es el objeto que realizará la conversión del objeto Java en el archivo XML.
        Marshaller marshaller = context.createMarshaller();
        // Configurar el marshaller para un formato de salida más legible:
        // Esta configuración opcional permite que el XML resultante esté formateado
        // con saltos de línea e indentación para que sea más legible. Sin esta propiedad,
        // el XML se generaría en una sola línea.
        marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
        // Convertir el objeto en XML y escribirlo en el archivo:
        marshaller.marshal(estudiante, file);
    }
    // Devuelve un objeto de la clase Estudiante construido a partir de las
    // propiedades guardadas en el archivo pasado como parámetro en formato XML
}
```

Debug Main x

Threads & Variables Console

\*main\*@1 in group "main": RUNNING

marshal23, JAXBUtil

main:13, Main

Evaluate expression (Intro) or add a watch (Ctrl+Mayús+Intro)

- > estudiante = (Estudiante@1416)
- > file = (File@1417) "JAXBUtilPruebasEstudiante.txt"
- > context = (JAXBContextImpl@1418) "jar:file:/C:/Users/Usuario/.m2/repository/org/glassfish/jaxb/jaxb-runtime/4.0.5/jaxb-runtime-4.0.5.jar"
- > marshaller = (MarshallerImpl@1419)

Método unmarshal (Estudiante)

```
public static Estudiante unmarshal(File file) throws JAXBException {
    JAXBContext context = JAXBContext.newInstance(Estudiante.class);
    Unmarshaller unmarshaller = context.createUnmarshaller();
    return (Estudiante) unmarshaller.unmarshal(file);
}
```

Debug Main x

Threads & Variables Console

\*main\*@1 in group "main": RUNNING

marshal31, JAXBUtil

main:14, Main

Evaluate expression (Intro) or add a watch (Ctrl+Mayús+Intro)

- > file = (File@1417) "JAXBUtilPruebasEstudiante.txt"
- > context = (JAXBContextImpl@1650) "jar:file:/C:/Users/Usuario/.m2/repository/org/glassfish/jaxb/jaxb-runtime/4.0.5/jaxb-runtime-4.0.5.jar"
- > unmarshaller = (UnmarshallerImpl@1651)

Método marshalCurso (Curso)

```
public static void marshalCurso(Curso curso, File file) throws JAXBException {
    JAXBContext context = JAXBContext.newInstance(Curso.class);
    Marshaller marshaller = context.createMarshaller();
    marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
    marshaller.marshal(curso, file);
}
```

Debug Main x

Threads & Variables Console

\*main\*@1 in group "main": RUNNING

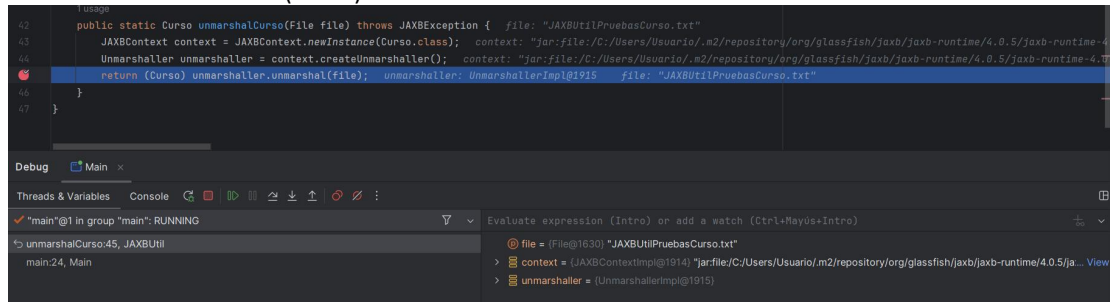
marshalCurso39, JAXBUtil

main:23, Main

Evaluate expression (Intro) or add a watch (Ctrl+Mayús+Intro)

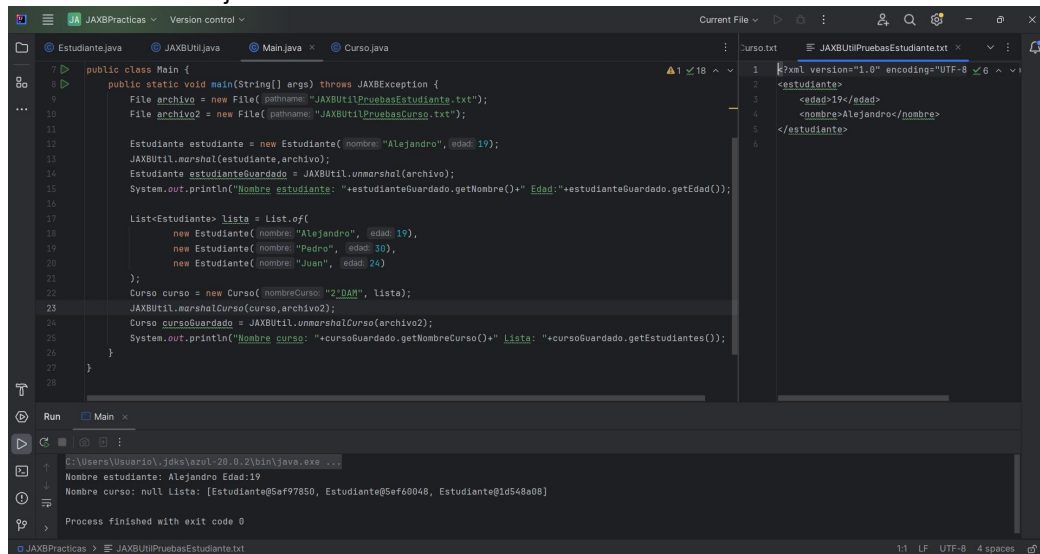
- > curso = (Curso@1906)
- > file = (File@1630) "JAXBUtilPruebasCurso.txt"
- > context = (JAXBContextImpl@1907) "jar:file:/C:/Users/Usuario/.m2/repository/org/glassfish/jaxb/jaxb-runtime/4.0.5/jaxb-runtime-4.0.5.jar"
- > marshaller = (MarshallerImpl@1908)

## Método unmarshalCurso (Curso)

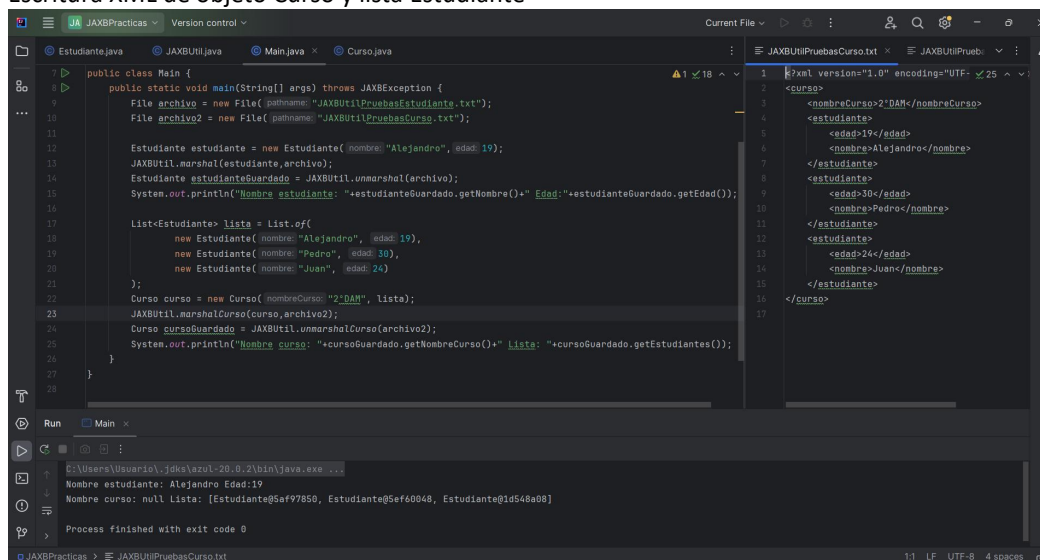


## 5. Resultados clase Main

### Escritura XML de objeto Estudiante



### Escritura XML de objeto Curso y lista Estudiante



## 6. Errores y soluciones

### Error 1:

En las clases Estudiante y Curso no funcionará los métodos de JAXB si las propiedades privadas contienen '@XmlElement' y además tenemos getters asignados a estas.

Saldrá que tenemos las propiedades duplicadas.

```
C:\Users\Usuario\.jdk\azul-20.0.2\bin\java.exe ...
Exception in thread "main" org.glassfish.jaxb.runtime.v2.runtime.IllegalAnnotationsException: 2 counts of IllegalAnnotationExceptions
La clase tiene dos propiedades con el mismo nombre "edad"
  this problem is related to the following location:
    at public int Estudiante.getEdad()
    at Estudiante
  this problem is related to the following location:
    at private int Estudiante.edad
    at Estudiante
La clase tiene dos propiedades con el mismo nombre "nombre"
  this problem is related to the following location:
    at public java.lang.String Estudiante.getNombre()
    at Estudiante
  this problem is related to the following location:
    at private java.lang.String Estudiante.nombre
    at Estudiante

at org.glassfish.jaxb.runtime.v2.runtime.IllegalAnnotationsException$Builder.check(IllegalAnnotationsException.java:83)
at org.glassfish.jaxb.runtime.v2.runtime.JAXBContextImpl.getTypeInfoSet(JAXBContextImpl.java:421)
at org.glassfish.jaxb.runtime.v2.runtime.JAXBContextImpl.<init>(JAXBContextImpl.java:255)
at org.glassfish.jaxb.runtime.v2.runtime.JAXBContextImpl$JAXBContextBuilder.build(JAXBContextImpl.java:1115)
at org.glassfish.jaxb.runtime.v2.ContextFactory.createContext(ContextFactory.java:144)
at org.glassfish.jaxb.runtime.v2.JAXBContextFactory.createContext(JAXBContextFactory.java:44)
at jakarta.xml.bind.ContextFinder.find(ContextFinder.java:368)
at jakarta.xml.bind.JAXBContext.newInstance(JAXBContext.java:605)
at jakarta.xml.bind.JAXBContext.newInstance(JAXBContext.java:546)
at JAXBUtil.marshal(JAXBUtil.java:13)
at Main.main(Main.java:13)
```

La solución es o quitar los getters o cambiar '@XmlElement' de las propiedades a los getters.

### Error 2:

Si no tenemos '@XmlRootElement' puesto a la clase saltará este error.

```
C:\Users\Usuario\.jdk\azul-20.0.2\bin\java.exe ...
Exception in thread "main" jakarta.xml.bind.MarshalException
- with linked exception:
[com.sun.istack.SAXException2: no se ha podido canalizar el tipo "Estudiante" como un elemento, porque le falta una anotación @XmlRootElement]
at org.glassfish.jaxb.runtime.v2.runtime.MarshallerImpl.write(MarshallerImpl.java:268)
at org.glassfish.jaxb.runtime.v2.runtime.MarshallerImpl.marshal(MarshallerImpl.java:197)
at jakarta.xml.bind.helpers.AbstractMarshallerImpl.marshal(AbstractMarshallerImpl.java:95)
at JAXBUtil.marshal(JAXBUtil.java:23)
at Main.main(Main.java:13)
Caused by: com.sun.istack.SAXException2: Create breakpoint: no se ha podido canalizar el tipo "Estudiante" como un elemento, porque le falta una anotación @XmlRootElement
at org.glassfish.jaxb.runtime.v2.runtime.XMLSerializer.reportError(XMLSerializer.java:213)
at org.glassfish.jaxb.runtime.v2.runtime.ClassBeanInfoImpl.serializeRoot(ClassBeanInfoImpl.java:311)
at org.glassfish.jaxb.runtime.v2.runtime.XMLSerializer.childAsRoot(XMLSerializer.java:456)
at org.glassfish.jaxb.runtime.v2.runtime.MarshallerImpl.write(MarshallerImpl.java:265)
... 4 more
```

Pon '@XmlRootElement' encima de la clase.

### Error 3:

Tipo de datos no soportado

Mensaje de error: JAXB can't handle class...

Causa: JAXB no puede manejar tipos como Map.

Solución: Cambia a tipos de datos soportados (ej., usa List en lugar de Map).