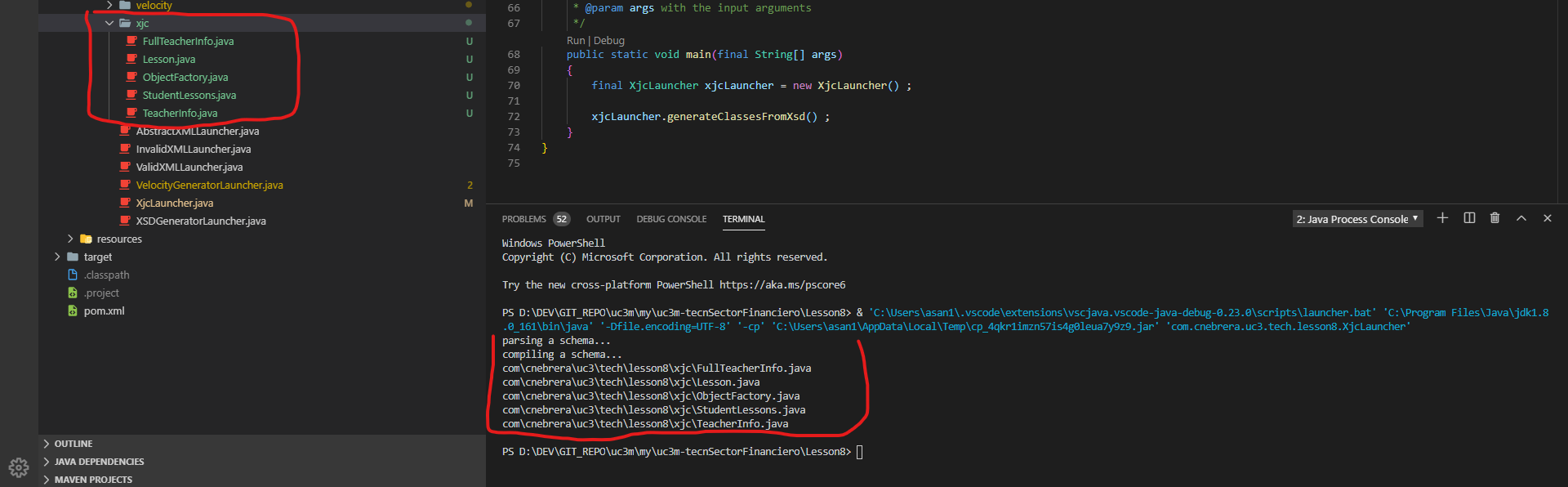
# Practica 8: Protocolos de mensajería

## Parte 2: Generación de código Java mediante XJC a partir de un XSD

El siguiente es el resultado de correr la clase XjcLauncher.java en la cual se generan unas clases a partir de un xsd



## Parte 3: Generación de instancias de clases a partir de XMLs y viceversa

Clase JAXBHandler:

package com.cnebrera.uc3.tech.lesson8.jaxb;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.IOException;

import java.io.OutputStream;

import javax.xml.XMLConstants;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Marshaller;

import javax.xml.bind.SchemaOutputResolver;

import javax.xml.bind.Unmarshaller;

import javax.xml.validation.Schema;

import javax.xml.validation.SchemaFactory;

import org.xml.sax.SAXException;

import com.cnebrera.uc3.tech.lesson8.util.Constants;

import com.cnebrera.uc3.tech.lesson8.xjc.StudentLessons;

/\*\*

 \* JAXB Handler

 \* --------------------------------------

 \* @author Francisco Manuel Benitez Chico

 \* --------------------------------------

 \*/

public class JAXBHandler

{

    /\*\* Attribute - JAXB Context \*/

    private final JAXBContext jaxbContext ;

    /\*\*

     \* Initialize the JAXB Context

     \* @throws JAXBException with an occurred exception

     \*/

    public JAXBHandler() throws JAXBException

    {

        // TODO 1

        this.jaxbContext = JAXBContext.newInstance(StudentLessons.class);

    }

    /\*\*

     \* @param file           with the file

     \* @return a new instance of StudentLessons with the filled values from the XML

     \* @throws JAXBException with an occurred exception

     \* @throws SAXException  with an occurred exception

     \*/

    public StudentLessons convertToObject(final File file) throws JAXBException, SAXException

    {

        // TODO 2

        final Unmarshaller jaxbUnmarshaller = this.jaxbContext.createUnmarshaller() ;

        // TODO 3

        // Set validator

        final SchemaFactory sf = SchemaFactory.newInstance(XMLConstants.W3C\_XML\_SCHEMA\_NS\_URI) ;

        final Schema schema = sf.newSchema(new File(Constants.XSD\_FILE\_INPUT)) ;

        jaxbUnmarshaller.setSchema(schema) ;

        // TODO 4

        // Unmarshal the content of the file

        return (StudentLessons) jaxbUnmarshaller.unmarshal(file) ;

    }

    /\*\*

     \* @param studentLessons with the instance of StudentLessons

     \* @return a string with the XML content

     \* @throws JAXBException with an occurred exception

     \*/

    public String convertToXml(final StudentLessons studentLessons) throws JAXBException

    {

        // TODO 5

        final Marshaller jaxbMarshaller = this.jaxbContext.createMarshaller() ;

        // TODO 6

        // Output pretty printed

        jaxbMarshaller.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true) ;

        // TODO 7

        // Output Stream

        final OutputStream outputStream = new ByteArrayOutputStream() ;

        // TODO 8

        // Marshal the StudentLessons object

        jaxbMarshaller.marshal(studentLessons, outputStream) ;

        // TODO 9

        return outputStream.toString() ;

    }

    /\*\*

     \* @param namespaceUri      with the namespace URI

     \* @param suggestedFileName with the suggested file name

     \* @return the generated schema

     \* @throws IOException      with an occurred exception

     \*/

    public void generateSchema(final String namespaceUri, final String suggestedFileName) throws IOException

    {

        // TODO 10

        // TODO 11

        // TODO 12

    }

}

Resultado de ejecución de ValidXMLLauncher.java:

PS D:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8> & 'C:\Users\asan1\.vscode\extensions\vscjava.vscode-java-debug-0.23.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk1.8.0\_161\bin\java' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\asan1\AppData\Local\Temp\cp\_4qkr1imzn57is4g0leua7y9z9.jar' 'com.cnebrera.uc3.tech.lesson8.ValidXMLLauncher'

com.cnebrera.uc3.tech.lesson8.xjc.StudentLessons@7cc355be

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<studentLessons>

    <studentName>Paco Benitez</studentName>

    <studentLesson lessonId="1">

        <lessonFrom>2016-09-12</lessonFrom>

        <lessonTo>2016-09-18</lessonTo>

        <lessonName>Latencia</lessonName>

        <lessonTeacher>

            <teacherFirstName>Carlos</teacherFirstName>

            <teacherLastName>Nebrera</teacherLastName>

            <teacherCity>Madrid</teacherCity>

            <teacherCountry>Spain</teacherCountry>

        </lessonTeacher>

        <lessonMark>10</lessonMark>

    </studentLesson>

    <studentLesson lessonId="6">

        <lessonFrom>2016-10-24</lessonFrom>

        <lessonTo>2016-10-30</lessonTo>

        <lessonName>In Memory Grid</lessonName>

        <lessonTeacher>

            <teacherFirstName>Florentino</teacherFirstName>

            <teacherLastName>Sainz</teacherLastName>

            <teacherCity>Madrid</teacherCity>

            <teacherCountry>Spain</teacherCountry>

        </lessonTeacher>

        <lessonMark>9</lessonMark>

    </studentLesson>

</studentLessons>

Resultado de ejecución de InvalidXMLLauncher.java

Acá como primer ejemplo nos está diciendo que ' lessonName ' es esperado, y en el invalid.xml el lessonName esta comentado

PS D:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8> cd 'd:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8'; & 'C:\Users\asan1\.vscode\extensions\vscjava.vscode-java-debug-0.23.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk1.8.0\_161\bin\java' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\asan1\AppData\Local\Temp\cp\_4qkr1imzn57is4g0leua7y9z9.jar' 'com.cnebrera.uc3.tech.lesson8.InvalidXMLLauncher'

Exception in thread "main" javax.xml.bind.UnmarshalException

 - with linked exception:

[org.xml.sax.SAXParseException; systemId: file:/D:/DEV/GIT\_REPO/uc3m/my/uc3m-tecnSectorFinanciero/Lesson8/src/main/resources/invalid.xml; lineNumber: 11; columnNumber: 24; cvc-complex-type.2.4.a: Invalid content was found starting with element 'lessonTeacher'. One of '{lessonName}' is expected.]

        at javax.xml.bind.helpers.AbstractUnmarshallerImpl.createUnmarshalException(AbstractUnmarshallerImpl.java:335)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.UnmarshallerImpl.createUnmarshalException(UnmarshallerImpl.java:563)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.UnmarshallerImpl.unmarshal0(UnmarshallerImpl.java:249)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.UnmarshallerImpl.unmarshal(UnmarshallerImpl.java:214)

        at javax.xml.bind.helpers.AbstractUnmarshallerImpl.unmarshal(AbstractUnmarshallerImpl.java:157)

        at javax.xml.bind.helpers.AbstractUnmarshallerImpl.unmarshal(AbstractUnmarshallerImpl.java:162)

        at javax.xml.bind.helpers.AbstractUnmarshallerImpl.unmarshal(AbstractUnmarshallerImpl.java:171)

        at javax.xml.bind.helpers.AbstractUnmarshallerImpl.unmarshal(AbstractUnmarshallerImpl.java:189)

        at com.cnebrera.uc3.tech.lesson8.jaxb.JAXBHandler.convertToObject(JAXBHandler.java:61)

        at com.cnebrera.uc3.tech.lesson8.AbstractXMLLauncher.handleXML(AbstractXMLLauncher.java:33)

        at com.cnebrera.uc3.tech.lesson8.InvalidXMLLauncher.main(InvalidXMLLauncher.java:26)

Caused by: org.xml.sax.SAXParseException; systemId: file:/D:/DEV/GIT\_REPO/uc3m/my/uc3m-tecnSectorFinanciero/Lesson8/src/main/resources/invalid.xml; lineNumber: 11; columnNumber: 24; cvc-complex-type.2.4.a: Invalid content was found starting with element 'lessonTeacher'. One of '{lessonName}' is expected.

        at com.sun.org.apache.xerces.internal.util.ErrorHandlerWrapper.createSAXParseException(ErrorHandlerWrapper.java:203)

        at com.sun.org.apache.xerces.internal.util.ErrorHandlerWrapper.error(ErrorHandlerWrapper.java:134)

        at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:396)

        at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:327)

        at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:284)

        at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator$XSIErrorReporter.reportError(XMLSchemaValidator.java:453)

        at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.reportSchemaError(XMLSchemaValidator.java:3231)

        at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.handleStartElement(XMLSchemaValidator.java:1791)

        at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.startElement(XMLSchemaValidator.java:741)

        at com.sun.org.apache.xerces.internal.jaxp.validation.ValidatorHandlerImpl.startElement(ValidatorHandlerImpl.java:568)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.ValidatingUnmarshaller.startElement(ValidatingUnmarshaller.java:86)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.SAXConnector.startElement(SAXConnector.java:153)

        at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.startElement(AbstractSAXParser.java:509)

        at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl.scanStartElement(XMLNSDocumentScannerImpl.java:374)

        at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentContentDriver.next(XMLDocumentFragmentScannerImpl.java:2784)

        at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(XMLDocumentScannerImpl.java:602)

        at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl.next(XMLNSDocumentScannerImpl.java:112)

        at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:505)

        at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:842)

        at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:771)

        at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser.java:141)

        at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1213)

        at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(SAXParserImpl.java:643)

        at com.sun.xml.internal.bind.v2.runtime.unmarshaller.UnmarshallerImpl.unmarshal0(UnmarshallerImpl.java:243)

        ... 8 more

PS D:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8>

Si quitamos estos comentarios, arroja un xml valido:

PS D:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8> cd 'd:\DEV\GIT\_REPO\uc3m\my\uc3m-tecnSectorFinanciero\Lesson8'; & 'C:\Users\asan1\.vscode\extensions\vscjava.vscode-java-debug-0.23.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk1.8.0\_161\bin\java' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\asan1\AppData\Local\Temp\cp\_4qkr1imzn57is4g0leua7y9z9.jar' 'com.cnebrera.uc3.tech.lesson8.InvalidXMLLauncher'

com.cnebrera.uc3.tech.lesson8.xjc.StudentLessons@7cc355be

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<studentLessons>

    <studentName>Paco Benitez</studentName>

    <studentLesson lessonId="1">

        <lessonFrom>2016-09-12</lessonFrom>

        <lessonTo>2016-09-18</lessonTo>

        <lessonName>Latencia</lessonName>

        <lessonTeacher>

            <teacherFirstName>Carlos</teacherFirstName>

            <teacherLastName>Nebrera</teacherLastName>

            <teacherCity>Madrid</teacherCity>

            <teacherCountry>Spain</teacherCountry>

        </lessonTeacher>

        <lessonMark>7</lessonMark>

    </studentLesson>

    <studentLesson lessonId="4">

        <lessonFrom>2016-10-03</lessonFrom>

        <lessonTo>2016-10-09</lessonTo>

        <lessonName>Mensajeria de ultima milla I</lessonName>

        <lessonTeacher>

            <teacherFirstName>Carlos</teacherFirstName>

            <teacherLastName>Nebrera</teacherLastName>

            <teacherCity>Madrid</teacherCity>

            <teacherCountry>Spain</teacherCountry>

        </lessonTeacher>

        <lessonMark>10</lessonMark>

    </studentLesson>

    <studentLesson lessonId="6">

        <lessonFrom>2016-10-24</lessonFrom>

        <lessonTo>2016-10-30</lessonTo>

        <lessonName>In Memory Grid</lessonName>

        <lessonTeacher>

            <teacherFirstName>Florentino</teacherFirstName>

            <teacherLastName>Sainz</teacherLastName>

            <teacherCity>Madrid</teacherCity>

            <teacherCountry>Spain</teacherCountry>

        </lessonTeacher>

        <lessonMark>9</lessonMark>

    </studentLesson>

</studentLessons>

## Parte 4: Generación de XSD a partir de clases anotadas con JAXB

Clase JAXBHandler.java:

package com.cnebrera.uc3.tech.lesson8.jaxb;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.IOException;

import java.io.OutputStream;

import javax.xml.XMLConstants;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Marshaller;

import javax.xml.bind.SchemaOutputResolver;

import javax.xml.bind.Unmarshaller;

import javax.xml.validation.Schema;

import javax.xml.validation.SchemaFactory;

import org.xml.sax.SAXException;

import com.cnebrera.uc3.tech.lesson8.util.Constants;

import com.cnebrera.uc3.tech.lesson8.xjc.StudentLessons;

/\*\*

 \* JAXB Handler

 \* --------------------------------------

 \* @author Francisco Manuel Benitez Chico

 \* --------------------------------------

 \*/

public class JAXBHandler

{

    /\*\* Attribute - JAXB Context \*/

    private final JAXBContext jaxbContext ;

    /\*\*

     \* Initialize the JAXB Context

     \* @throws JAXBException with an occurred exception

     \*/

    public JAXBHandler() throws JAXBException

    {

        // TODO 1

        this.jaxbContext = JAXBContext.newInstance(StudentLessons.class);

    }

    /\*\*

     \* @param file           with the file

     \* @return a new instance of StudentLessons with the filled values from the XML

     \* @throws JAXBException with an occurred exception

     \* @throws SAXException  with an occurred exception

     \*/

    public StudentLessons convertToObject(final File file) throws JAXBException, SAXException

    {

        // TODO 2

        final Unmarshaller jaxbUnmarshaller = this.jaxbContext.createUnmarshaller() ;

        // TODO 3

        // Set validator

        final SchemaFactory sf = SchemaFactory.newInstance(XMLConstants.W3C\_XML\_SCHEMA\_NS\_URI) ;

        final Schema schema = sf.newSchema(new File(Constants.XSD\_FILE\_INPUT)) ;

        jaxbUnmarshaller.setSchema(schema) ;

        // TODO 4

        // Unmarshal the content of the file

        return (StudentLessons) jaxbUnmarshaller.unmarshal(file) ;

    }

    /\*\*

     \* @param studentLessons with the instance of StudentLessons

     \* @return a string with the XML content

     \* @throws JAXBException with an occurred exception

     \*/

    public String convertToXml(final StudentLessons studentLessons) throws JAXBException

    {

        // TODO 5

        final Marshaller jaxbMarshaller = this.jaxbContext.createMarshaller() ;

        // TODO 6

        // Output pretty printed

        jaxbMarshaller.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true) ;

        // TODO 7

        // Output Stream

        final OutputStream outputStream = new ByteArrayOutputStream() ;

        // TODO 8

        // Marshal the StudentLessons object

        jaxbMarshaller.marshal(studentLessons, outputStream) ;

        // TODO 9

        return outputStream.toString() ;

    }

    /\*\*

     \* @param namespaceUri      with the namespace URI

     \* @param suggestedFileName with the suggested file name

     \* @return the generated schema

     \* @throws IOException      with an occurred exception

     \*/

    public void generateSchema(final String namespaceUri, final String suggestedFileName) throws IOException

    {

        // TODO 10

        final SchemaOutputResolver schemaOutputResolver = new MySchemaOutputResolver();

        // TODO 11

        this.jaxbContext.generateSchema(schemaOutputResolver) ;

        // TODO 12

        schemaOutputResolver.createOutput(namespaceUri, suggestedFileName) ;

    }

}

Clase MySchemaOutputResolver.java:

package com.cnebrera.uc3.tech.lesson8.jaxb;

import java.io.File;

import java.io.IOException;

import javax.xml.bind.SchemaOutputResolver;

import javax.xml.transform.Result;

import javax.xml.transform.stream.StreamResult;

/\*\*

 \* My Schema Output Resolver

 \* --------------------------------------

 \* @author Francisco Manuel Benitez Chico

 \* --------------------------------------

 \*/

public class MySchemaOutputResolver extends SchemaOutputResolver

{

    /\*\*

     \* @param namespaceURI      with the namespace URI

     \* @param suggestedFileName with the suggested file name

     \* @return an instance of Result

     \* @throws IOException with an occurred exception

     \*/

    @Override

    public Result createOutput(final String namespaceURI, final String suggestedFileName) throws IOException

    {

        // TODO 1

        final File file = new File(suggestedFileName) ;

        final StreamResult result = new StreamResult(file) ;

        result.setSystemId(file.toURI().toURL().toString()) ;

        return result ;

    }

}

Fichero generado schema1.xsd

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="studentLessons">

    <xs:complexType>

      <xs:sequence>

        <xs:element name="studentName" type="xs:string"/>

        <xs:element name="studentLesson" type="lesson" maxOccurs="unbounded"/>

      </xs:sequence>

    </xs:complexType>

  </xs:element>

  <xs:complexType name="lesson">

    <xs:sequence>

      <xs:element name="lessonFrom" type="xs:date"/>

      <xs:element name="lessonTo" type="xs:date"/>

      <xs:element name="lessonName" type="xs:string"/>

      <xs:element name="lessonTeacher" type="fullTeacherInfo"/>

      <xs:element name="lessonMark" type="xs:integer"/>

    </xs:sequence>

    <xs:attribute name="lessonId" type="xs:integer" use="required"/>

  </xs:complexType>

  <xs:complexType name="fullTeacherInfo">

    <xs:complexContent>

      <xs:extension base="teacherInfo">

        <xs:sequence>

          <xs:element name="teacherCity" type="xs:string"/>

          <xs:element name="teacherCountry" type="xs:string"/>

        </xs:sequence>

      </xs:extension>

    </xs:complexContent>

  </xs:complexType>

  <xs:complexType name="teacherInfo">

    <xs:sequence>

      <xs:element name="teacherFirstName" type="xs:string"/>

      <xs:element name="teacherLastName" type="xs:string"/>

    </xs:sequence>

  </xs:complexType>

</xs:schema>

## Parte 5: Generación de código con velocity

A continuación, se ejecuta la clase VelocityGeneratorLauncher.java con los valores en comentarios, y se puede evidenciar en la clase generada VelocityClassExample.java que el atributo con tipo List esta marcado con error debido a que la clase no esta importando la clase nativa correspondiente:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This class has been automatically generated using Velocity!

Thu Dec 05 16:31:20 CET 2019

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package com.cnebrera.uc3.tech.lesson8.velocity ;

public class VelocityClassExample

{

   /\*\* Attribute - myListStringValues \*/

   private List<String> myListStringValues ;

    /\*\*

     \* Public Constructor

     \*/

    // TODO 5

    {

        this.myListStringValues = new ArrayList<String>() ;

        // Set values from Velocity

// TODO 6

    }

    /\*\*

     \* @return myListStringValues as string

     \*/

    public String toString()

    {

        return myListStringValues.toString() ;

    }

    /\*\*

     \* @param args with the input arguments

     \*/

    public static void main(final String[] args)

    {

        final VelocityClassExample myExample = new VelocityClassExample() ;

        System.out.println(myExample.toString()) ;

    }

}

A continuación, se ejecuta la clase VelocityGeneratorLauncher.java con sin los valores en comentarios, y se puede evidenciar en la clase generada VelocityClassExample.java de forma correcta:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This class has been automatically generated using Velocity!

Thu Dec 05 16:40:54 CET 2019

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package com.cnebrera.uc3.tech.lesson8.velocity ;

import java.util.List ;

import java.util.ArrayList ;

public class VelocityClassExample

{

   /\*\* Attribute - myListStringValues \*/

   private List<String> myListStringValues ;

    /\*\*

     \* Public Constructor

     \*/

    // TODO 5

    {

        this.myListStringValues = new ArrayList<String>() ;

        // Set values from Velocity

// TODO 6

    }

    /\*\*

     \* @return myListStringValues as string

     \*/

    public String toString()

    {

        return myListStringValues.toString() ;

    }

    /\*\*

     \* @param args with the input arguments

     \*/

    public static void main(final String[] args)

    {

        final VelocityClassExample myExample = new VelocityClassExample() ;

        System.out.println(myExample.toString()) ;

    }

}

La plantilla de velocity con la cual se generó la clase:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This class has been automatically generated using Velocity!

${now}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package ${packageName} ;

#foreach($classImport in $classImports)

import ${classImport} ;

#end

public class ${className}

{

   /\*\* Attribute - myListStringValues \*/

   private List<String> myListStringValues ;

    /\*\*

     \* Public Constructor

     \*/

    // TODO 5

    {

        this.myListStringValues = new ArrayList<String>() ;

        // Set values from Velocity

// TODO 6

    }

    /\*\*

     \* @return myListStringValues as string

     \*/

    public String toString()

    {

        return myListStringValues.toString() ;

    }

    /\*\*

     \* @param args with the input arguments

     \*/

    public static void main(final String[] args)

    {

        final ${className} myExample = new ${className}() ;

        System.out.println(myExample.toString()) ;

    }

}