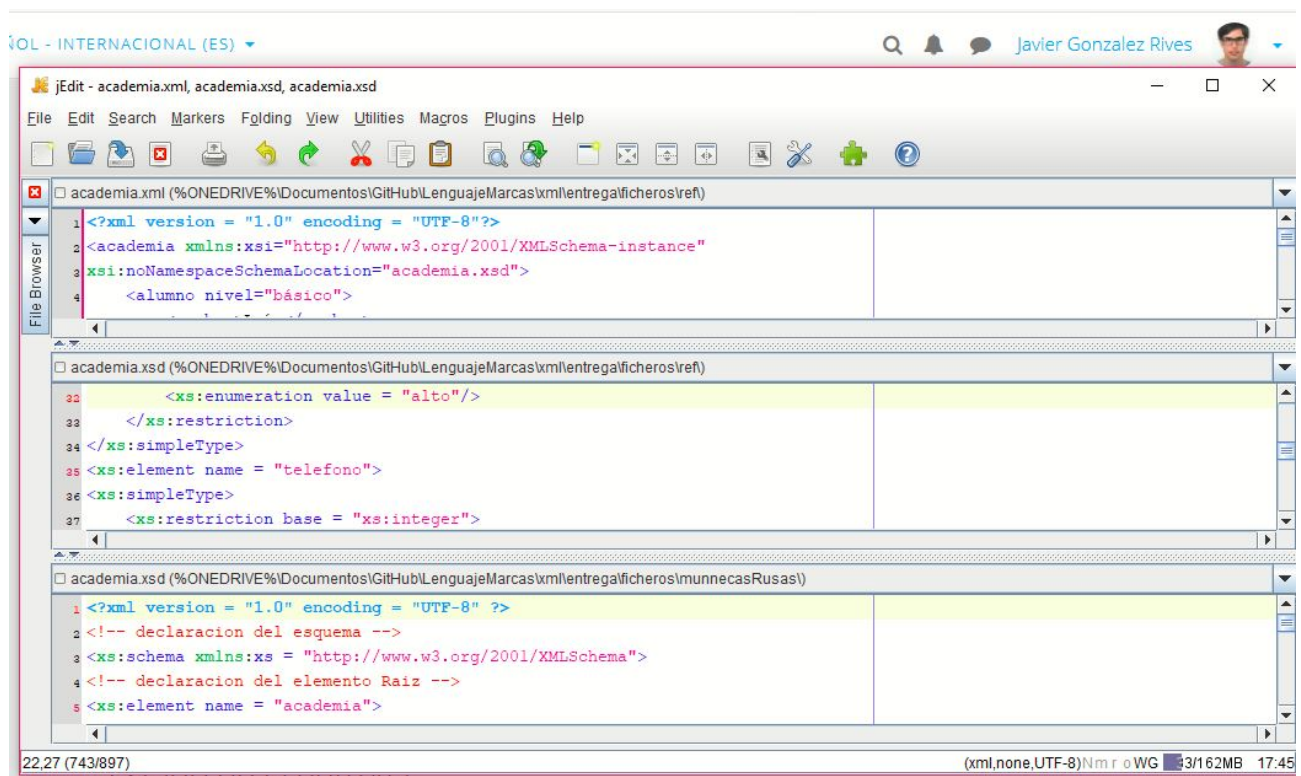


XML XSD, ESQUEMAS ENTREGA.

Como Se ha realizado esta tarea.

Para la realización de esta tarea se ha utilizado el editor de texto Jedit con el plugin para xml, este añade muchas funcionalidades que han sido muy útiles para la realización de esta tarea, algunas de estas es la propia validación del documento o el auto completado.

Ademas se ha usado una pagina web esterna para tener una doble validación del documento (<https://xmlvalidation.com/>).



Método muñecas rusas.

Este método consiste en adaptar la estructura del documento xml directamente a esquema, esto nos permite una rápida creación del documento sin necesidad de darle muchas complicaciones al diseño. El inconveniente es que la legibilidad de este es complicada debido a que todos los elementos salen entremezclados.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<!-- declaracion del esquema -->
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
<!-- declaracion del elemento Raiz -->
<xs:element name = "academia">
<xs:complexType>
<xs:sequence>
```

```
<!-- declaracion del alumno :0 -->
<xs:element name = "alumno" maxOccurs = "unbounded">
<xs:complexType>
  <xs:all>
    <!-- sub elementos del elemento alumno -->
    <!-- tanto el elemento nombre como el elemento apellido
llevan
una restriccion en el numero de caracteres-->
    <xs:element name = "nombre" >
      <xs:simpleType>
        <xs:restriction base = "xs:string">
          <xs:maxLength value = "50"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name = "apellidos" >
      <xs:simpleType>
        <xs:restriction base = "xs:string">
          <xs:maxLength value = "50"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element maxOccurs="1" name="fecha_nacimiento"
type="xs:date"/>
    <!-- sub elemento compuesto notas -->
    <xs:element maxOccurs="1" name="notas">
      <xs:complexType>
        <xs:sequence>
          <!-- trim usa un rango en el que pueden estar las notas
-->
            <xs:element name = "trim" minOccurs="0"
maxOccurs="unbounded">
              <xs:simpleType>
                <xs:restriction base = "xs:decimal">
                  <xs:minInclusive value = "0"/>
                  <xs:maxInclusive value = "10"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- sub elemento compuesto telefonos -->
      <xs:element name = "telefonos">
        <xs:complexType>
          <xs:sequence>
            <xs:element name = "telefono"
minOccurs = "0" maxOccurs = "unbounded">
              <xs:simpleType>
                <xs:restriction base = "xs:integer">
                  <xs:maxInclusive value = "9999999999"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:restriction>
        </xs:simpleType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:all>
<!-- atributos de alumno -->
<xs:attribute name = "nivel" type = "xs:string" use =
"required" />
</xs:complexType>
</xs:element>
<!-- fin del elemento principal -->
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- definicon de nivel -->
<xs:simpleType name = "nivel">
    <xs:restriction base = "xs:string">
        <xs:enumeration value = "básico"/>
        <xs:enumeration value = "intermedio"/>
        <xs:enumeration value = "alto"/>
    </xs:restriction>
</xs:simpleType>

</xs:schema>
```

Modelo por clonación o por referencias.

Este modelo consiste en la creación y definición de los elementos por un lado y la creación de la estructura por otro referenciando a los elementos anteriormente mencionados.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
  <!-- nombre, limite de 50 caracteres -->
  <xs:element name = "nombre">
    <xs:simpleType>
      <xs:restriction base = "xs:string">
        <xs:maxLength value = "50"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <!-- apellido limite de 50 caracteres -->
  <xs:element name = "apellidos">
    <xs:simpleType>
      <xs:restriction base = "xs:string">
        <xs:maxLength value = "50"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name = "fecha_nacimiento" type = "xs:date"/>
  <xs:element name = "trim">
    <xs:simpleType>
      <xs:restriction base = "xs:decimal">
        <xs:minInclusive value = "0"/>
        <xs:maxInclusive value = "10"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:simpleType name = "nivel">
    <xs:restriction base = "xs:string">
      <xs:enumeration value = "básico"/>
      <xs:enumeration value = "intermedio"/>
      <xs:enumeration value = "alto"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name = "telefono">
    <xs:simpleType>
      <xs:restriction base = "xs:integer">
        <xs:maxInclusive value = "9999999999"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <!-- Elementos compuestos -->
  <xs:element name = "notas">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "trim" minOccurs = "1" maxOccurs
```

```
= "unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name = "telefonos">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "telefono" minOccurs = "1"
maxOccurs = "unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name = "alumno">
  <xs:complexType>
    <xs:all>
      <xs:element ref = "nombre"/>
      <xs:element ref = "apellidos"/>
      <xs:element ref = "fecha_nacimiento"/>
      <xs:element ref = "notas"/>
      <xs:element ref = "telefonos"/>
    </xs:all>
    <xs:attribute name = "nivel" type = "nivel"/>
  </xs:complexType>
</xs:element>
<xs:element name = "academia">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "alumno" minOccurs = "1" maxOccurs =
"unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```