

Nome: _____ Curso: _____ Num Mec: _____

A. Schema XML:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd" elementFormDefault="qualified"
  xmlns="http://tempuri.org/XMLSchema.xsd" xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="cursos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="curso">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="guid" type="xs:integer" />
              <xs:element name="codigo" type="xs:string" />
              <xs:element name="nome" type="xs:string" />
              <xs:element name="grau" type="tipoGrau" />
              <xs:element name="bolonha">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="0" />
                    <xs:maxInclusive value="1" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="vagas">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="0" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="template">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9]" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="guid" type="xs:integer" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="tipoGrau">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Licenciatura" />
      <xs:enumeration value="Mestrado" />
      <xs:enumeration value="Mestrado Integrado" />
      <xs:enumeration value="Doutoramento" />
      <xs:enumeration value="Curso de Especialização Tecnológica" />
      <xs:enumeration value="Licenciatura Bi-etápica" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

1. Atendendo ao XML Schema definido em (A), construa um documento XML válido com 3 elementos do tipo “curso”.
2. Ainda sobre o mesmo Schema representado em (A), que valores são válidos no elemento “bolonha”?
3. Que alterações deverão ser feitas nas regras do elemento para que o conjunto de valores aceites sejam apenas inteiros entre 10 e 20 inclusivé?

4. Supondo que o elemento “*curso*” pode ser reutilizado noutras partes do schema, que modificações deverão ser feitas ao Schema representado em (A) para que este elemento seja um elemento global?
Escreva a parte do schema que reflecte essas alterações.

Nota: as partes comuns poderão ser representadas como “...”.

Exemplo:

<xs:schema ...>

B. XML:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cursos>
  <curso guid="96">
    <codigo>450/2007</codigo>
    <nome>Administração e Gestão Pública</nome>
    <grau>Mestrado</grau>
    <bolonha>sim</bolonha>
    <vagas>40</vagas>
  </curso>
  <curso guid="54">
    <guid>54</guid>
    <codigo>0300-9002</codigo>
    <nome>Administração Pública</nome>
    <grau>Licenciatura</grau>
    <bolonha>sim</bolonha>
    <vagas>60</vagas>
  </curso>
  <curso guid="104">
    <guid>104</guid>
    <codigo>0300-0003</codigo>
    <nome>Administração Pública</nome>
```

```
    <grau>Licenciatura</grau>
    <bolonha>não</bolonha>
    <vagas>0</vagas>
  </curso>
  <curso guid="14">
    <codigo>0300-9011</codigo>
    <nome>Biologia</nome>
    <grau>Licenciatura</grau>
    <bolonha>sim</bolonha>
    <vagas>66</vagas>
  </curso>
  <curso guid="37">
    <guid>37</guid>
    <codigo>0300-0048</codigo>
    <nome>Biologia</nome>
    <grau>Licenciatura</grau>
    <bolonha>não</bolonha>
    <vagas>0</vagas>
  </curso>
</cursos>
```

5. O documento XML apresentado em (B) é válido para o XML Schema apresentado em (A)?
- Caso não seja, que erro(s) está(ão) presente(s)?
6. Represente a expressão XPATH para escolher apenas elementos “*curso*” com vagas disponíveis no documento (B).
7. Supondo que é necessário uma nova representação da informação do documento XML (B) mas com a informação dos elementos filho do elemento “*curso*” como atributos de um elemento curso.
- Elabore o XSLT necessário para transformar o documento original do documento pedido.
 - Os cursos deverão estar ordenados por nome.

Exemplo do resultado esperado:

```
<?xml version="1.0" encoding="utf-8"?>
<cursos>
  <curso guid="96" codigo="450/2007" nome="Administração e Gestão Pública" vagas="40"
    grau="Mestrado" bolonha="sim" />
  <curso guid="54" codigo="0300-9002" nome="Administração Pública" vagas="60"
    grau="Licenciatura" bolonha="sim" />
</cursos>
```

1. Following the XML Schema represented in (A) write a valid XML document with three “curso” elements.
2. With the XML Schema represented in (A), what are the valid values to the “bolonha” element?
3. What changes should be made in the Schema in order to allow only values between 10 and 20 (inclusive)?
4. Supposing that “curso” element can be reused in other contexts in the same document, what are the needed modifications in the XML Schema represented in (A) to make such element global?
5. The XML document represented in (B) is a valid document with XML Schema represented in (A)?
 - a. If not, write what is(are) the error(s)?
6. What is the XPath expression that selects only the “curso” elements with “vagas” >0 in the XML document (B)?
7. Suppose that you need a new representation of the XML document (B) but all the child elements of “curso” must be represented as attributes.
 - a. Write the XSLT code necessary to do that.
 - b. The “cursos” should be ordered by “nome”