

**Integração de Dados – 2016/17**

**Exame – Época Especialíssima**

**CTSPs - TPSI** - 2º ano/1º semestre Duração: 90 minutos 17/11/2017

**NOME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Nº ALUNO: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **[10%] Analise a seguinte situação.**

Um colégio possui as seguintes bases de dados sobre os seus alunos e professores

Alunos\_1(id, nome, idturma)

Alunos\_2(id, nome, idturma)

Notas(id\_aluno, listaNotas)

Turmas(id, ano, numero\_alunos)

Professores(idprof, idturma, nome, grupo)

Pretende-se um sistema de integração de dados com uma vista unificada que permita optimizar as seguintes pesquisas:

* dado **o nome** de um aluno obter a lista de notas
* dado **o id de um professor** obter a lista com os nomes dos alunos da(s) turma(s) que lecciona

**Apresente um sistema de integração de dados capaz de responder às restrições indicadas. Descreva os componentes do sistema de integração**

|  |
| --- |
|  |

1. **[20%] Construa as expressões regulares para cada uma das situações apresentadas de seguida.**
2. ER que encontre todas as palavras de um texto que comecem por letra maiúscula, terminem na letra ‘a’ e tenham entre 5 a 10 caracteres**.**

|  |
| --- |
| **\b[A-Z][a-zA-Z]{3,8}a\b** |

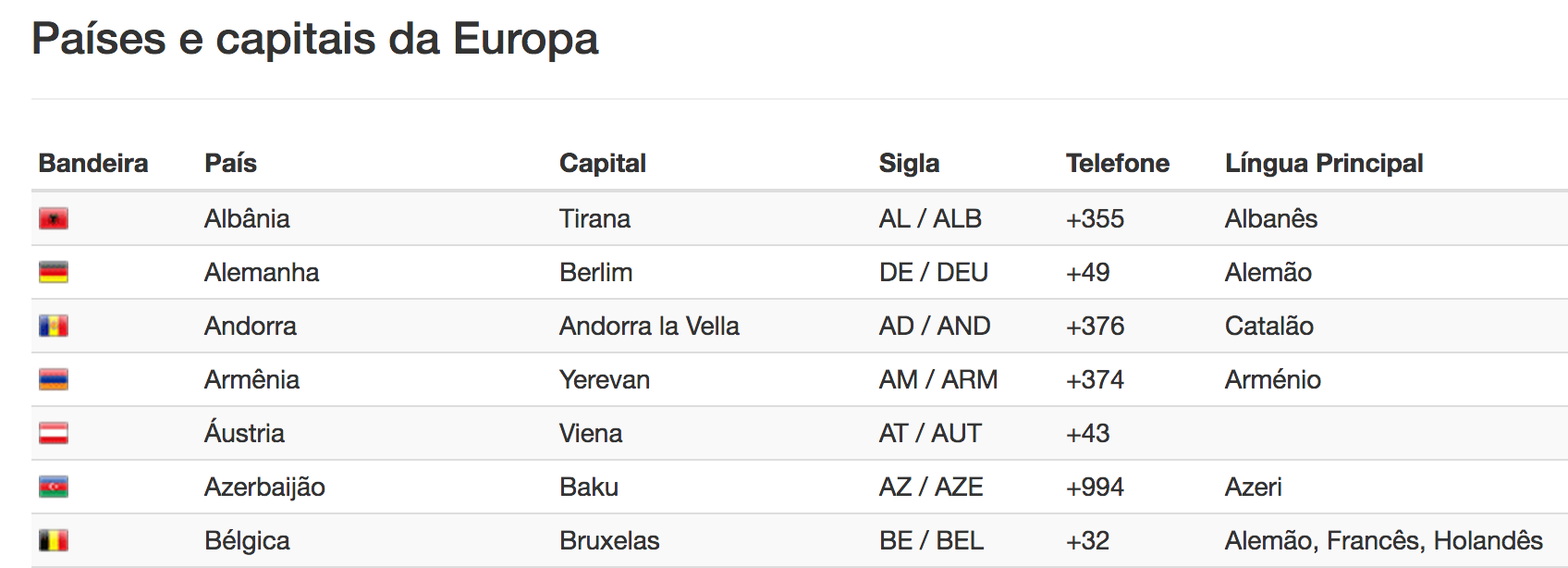
1. ER que encontre cadeias binárias com sequências de três zeros seguidos de três uns: 000111 ou 000111000111 ou 000111000111000111, etc**.**

|  |
| --- |
| **(000111)+** |

1. ER que valide endereços de email. Assuma que o nome do utilizador pode ter letras, dígitos, o ponto e o underscore**,** seguido do **@** e dos vários domínios (caracteres seguidos de ponto). O último domínio deve ter dois ou três caracteres: alice19\_a@isec.pt ana.melo12@uni.mas.edu Ana\_Bela.1999@mydomain.gov.us ....

|  |
| --- |
| **[a-zA-Z\.\_0-9]+@([a-z]+\.)+[a-z]{2,3}** |

Analise o seguinte código de uma página da internet sobre países



...

|  |  |
| --- | --- |
| 475 | <tr><td><i class="flag al albania"></i></td><td class=”pais”>Albânia</td><td class=”capital”>Tirana</td><td>AL / ALB</td><td>+355</td><td class="idioma">Albanês</td></tr> |
| 476 | <tr><td><i class="flag de alemanha"></i></td><td class=”pais”>Alemanha</td><td class=”capital”>Berlim</td><td>DE / DEU</td><td>+49</td><td class="idioma">Alemão</td></tr> |
| 477 | <tr><td><i class="flag ad andorra"></i></td><td class=”pais”>Andorra</td><td class=”capital”>Andorra la Vella</td><td>AD / AND</td><td>+376</td><td class="idioma" >Catalão</td></tr> |

...

d) Escreva uma expressão regular que permita extrair os idiomas de todos os países

* Apresente uma solução com lista de representantes

|  |
| --- |
| **<td class=”idioma”>([a-zA-Zãêíé.....]+)</td>** |

* Apresente uma solução com lista negada

|  |
| --- |
| **<td class=”idioma”>([^<]+)</td>** |

e) Escreva uma expressão regular que permita extrair a **capital** e **código telefónico** da Alemanha. Identifique os grupos da ER.

|  |
| --- |
| **class=”pais”>Alemanha</td><td class=”capital”>([^<]+)</td><td>[^<]+</td><td>(\+[0-9]+)</td>** |

1. **[10%] DTD**

Valide o ficheiro fornecido em **anexo 1** usando DTD. Considere que o atributo **tipo** é uma enumeração de 3 valores (Ciclo1, Ciclo2, Ciclo3). Este atributo é obrigatório. As restantes situações valide-as de acordo com o ficheiro dado.

1. **[20%] XSD**

a) Analise o seguinte XSD e escolha a(s) opção(ões) que é(são) validada(s) pelo XSD fornecido.

<xsd:element name="artigo">

<xsd:complexType>

<xsd:sequence minOccurs="0" maxOccurs="2">

<xsd:element name="nome" type="xsd:string" maxOccurs="2"/>

<xsd:element name="preco" type="xsd:double" minOccurs="0" maxOccurs="2"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

**B**

**A**

**C**

**D**

<artigo>

<nome>Caneta</nome>

</artigo>

<artigo>

<nome>Caneta</nome>

<nome>Lápis</nome>

<preco>12.2</preco>

</artigo>

<artigo>

<nome>Caneta</nome>

<preco>12.2</preco>

</artigo>

<artigo>

<preco>12.5</preco>

<preco>62.5</preco>

</artigo>

**E**

**H**

**G**

**F**

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<preco>32.2</preco>

<nome>Caneta</nome>

<nome>Caderno</nome>

</artigo>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<nome>Caneta</nome>

<preco>32.2</preco>

</artigo>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<nome>Caneta</nome>

<preco>32.2</preco>

<nome>Caneta</nome>

</artigo>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<nome>Caneta</nome>

<preco>32.2</preco>

<preco>32.2</preco>

<preco>32.2</preco> </artigo>

b) Analise o seguinte XSD e escolha a(s) opção(ões) que é(são) validada(s) pelo XSD fornecido.

<xsd:element name="artigo">

<xsd:complexType>

<xsd:choice maxOccurs="2">

<xsd:element name="nome" type="xsd:string" minOccurs="0" maxOccurs="2"/>

<xsd:element name="preco" type="xsd:double" minOccurs="0" maxOccurs="2"/>

</xsd:choice>

</xsd:complexType>

</xsd:element>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<nome>Caneta</nome>

<preco>32.2</preco>

<nome>Caneta</nome>

</artigo>

<artigo>

<nome>Lápis</nome>

<nome>Caneta</nome>

<preco>12.2</preco>

<preco>32.2</preco>

</artigo>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<nome>Caneta</nome>

<preco>32.2</preco>

<preco>32.2</preco>

<preco>32.2</preco> </artigo>

<artigo>

<nome>Lápis</nome>

<preco>12.2</preco>

<preco>32.2</preco>

<nome>Caneta</nome>

<nome>Caderno</nome>

</artigo>

**B**

**D**

**C**

**A**

<artigo>

<preco>12.5</preco>

<preco>62.5</preco>

</artigo>

<artigo>

<nome>Caneta</nome>

</artigo>

<artigo>

<nome>Caneta</nome>

<preco>12.2</preco>

</artigo>

<artigo>

<nome>Caneta</nome>

<nome>Lápis</nome>

<preco>12.2</preco>

</artigo>

**E**

**F**

**G**

**H**

Analise o ficheiro fornecido em **anexo** **1** e responda às seguintes questões:

1. Escreva o XSD que valide o elemento **avaliacao**. Deve ser uma restrição que permita apenas números inteiros entre 0 e 20.

**<xsd:element name=”avaliação”>**

**<xsd:simpleType>**

**<xsd:restriction base="xsd:integer">**

**<xsd:minInclusive value="0"/>**

**<xsd:maxInclusive value="20"/>**

**</xsd:restriction>**

**</xsd:simpleType>**

1. Escreva um tipo de dados XSD que possa ser usado para validar os elementos **professor** e **docente**.

**<xsd:complexType name=”tipoDados”>**

**<xsd:sequence>**

**<xsd:element name=”nome” type=”xsd:string”/>**

**<xsd:element name=”contacto” type=”xsd:string” maxOcuurs=”5”/>**

**</xsd:sequence>**

**</xsd:completType>**

1. Usando os tipos de dados anteriormente definidos, escreva o XSD que valide os elementos **professor** e **docente**

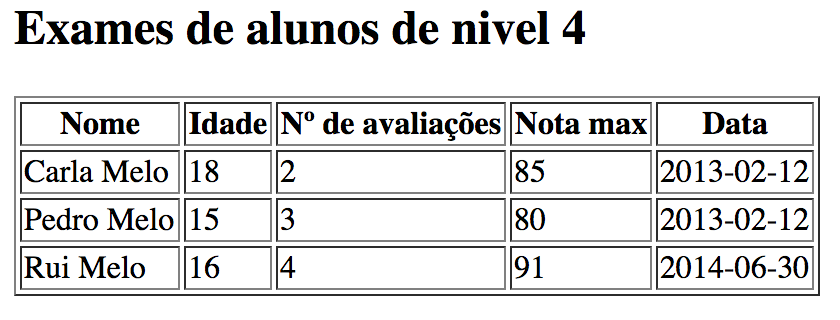
**<xsd:element name=”professor” type=”tipoDados”/>**

**<xsd:element name=”docente” type=”tipoDados”/>**

1. **[20%] XSLT**

Escreva um ficheiro XSLT que permita transformar o ficheiro XML fornecido no **anexo 2** no seguinte ficheiro HTML.

Deve mostrar apenas os alunos de nível 4, ordenados por ordem alfabética do nome. Na tabela deve ser visualizado o nome, a idade, o nº de exames que realizaram, a nota máxima obtida nas várias avaliações e a respectiva data.



**<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">**

**<xsl:output method="html"/>**

**<xsl:template match="exames">**

**<html>**

**<body>**

**<h2>Exames de alunos de nivel 4</h2>**

**<table border="1" >**

**<tr><th>Nome</th><th>Idade</th>**

**<th>Nº de avaliações</th>**

**<th>Nota max</th><th>Data</th></tr>**

**<xsl:for-each select="aluno">**

**<xsl:sort select="nome"/>**

**<xsl:if test="nivel='4'">**

**<tr><td><xsl:value-of select="nome"/></td>**

**<td><xsl:value-of select="idade"/></td>**

**<td><xsl:value-of select="count(avaliacoes/exame)"/></td>**

**<td><xsl:value-of select="max(avaliacoes/exame/@nota)"/></td>**

**<td><xsl:value-of select="avaliacoes/exame[max(../exame/@nota)=@nota]/@data"/></td>**

**</tr>**

**</xsl:if>**

**</xsl:for-each>**

**</table>**

**</body>**

**</html>**

**</xsl:template>**

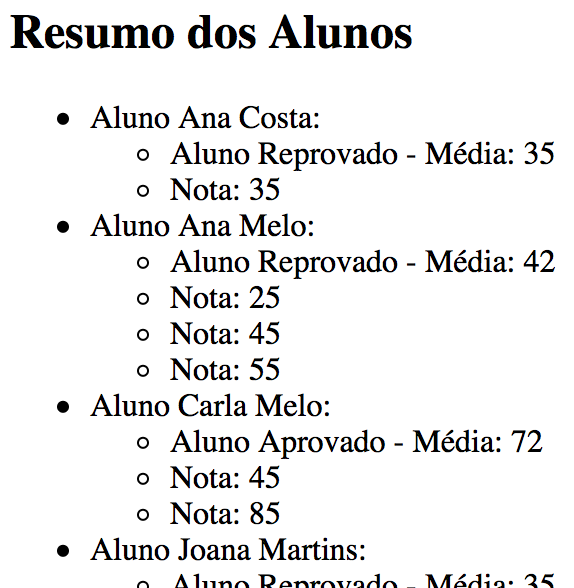
**</xsl:stylesheet>**

1. **[20%] XQuery**

Analise o ficheiro XML **exames.xml** fornecido em anexo

1. Escreva uma query que transforme o ficheiro **exames.xml** no seguinte output HTML

Os alunos estão ordenados por ordem alfabética. Para cada aluno deve ser calculada a média das notas. Se a média for superior ou igual a 50% deve aparecer a string “**Aluno Aprovado**”, caso contrário aparece a string “**Aluno reprovado**”, seguido da média e da lista de notas ordenadas por ordem crescente.



|  |
| --- |
| **<html><body><h2>Resumo dos Alunos</h2>**  **<ul>**  **{**  **for $a in doc("exames.xml")//aluno[nível=4]**  **let $media := avg($a//exame/@nota)**  **order by $a/nome**  **return if ($media >= 50) then**  **(<li>Aluno {$a/nome/text()}: </li>,**  **<ul>**  **<li>Aluno Aprovado - Média: {round($media)}</li>**  **{**  **for $ex in $a//exame**  **order by $ex/@nota**  **return <li>Nota: {data($ex/@nota)}</li>**  **}**  **</ul>)**  **else**  **(<li>Aluno {$a/nome/text()}: </li>,**  **<ul>**  **<li>Aluno Reprovado - Média: {round($media)}</li>**  **{**  **for $ex in $a//exame**  **order by $ex/@nota**  **return <li>Nota: {data($ex/@nota)}</li>**  **}**  **</ul>)**  **}</ul></body></html>** |