

## Exercício 1

- Escreva uma expressão regular que encontre todas as palavras começadas e terminadas por vogais (maiúsculas e/ou minúsculas).  
`\b[aeiouAEIOU][a-zA-Z]*[aeiouAEIOU]\b`
- Escreva uma expressão regular que encontre todas as palavras com 10 caracteres numéricos e/ou alfabéticos (maiúsculos ou minúsculos).  
`\b[a-zA-Z0-9]{10}\b`
- Expressão regular que capture todas as palavras começadas e terminadas pela letra 'a' (ou 'A'), sendo o restante conteúdo letras minúsculas ou maiúsculas.  
`\b[aA][a-zA-Z]*[aA]\b`
- Escreva uma ER que encontre palavras de tamanho 6 caracteres, começadas com três dígitos e terminadas com três vogais.  
`\b[0-9]{3}[aeiou]{3}\b`
- Escreva uma ER que encontre cadeias binárias que sejam repetições da sequência 011  
011 011011011 0111011011011011  
`\b(011)+\b`

## Exercício 2

`\b(a*b)*\b`

ab  
bbbb  
ababa  
abbaa  
ababab  
aaaaaa  
bababa

`\b(a?b*c+){2,3}\b`

ae  
bbbe  
abcabc  
bcbcbc  
be  
ccc  
abccccbc

`\b[01]?.[bc]*\b`

0Xbb  
1Xcbc  
11c  
01  
000ee  
01c  
11bc

## Exercício 3

`\ba?b*s[0-9]*\b`

Select one or more:

- ☐ bbs  
☒ s222  
☐ ass333  
☒ abs001  
☐ aabs9

`\b[abc]?a[bc]*\b`

Select one or more:

- ☒ abc  
☐ aaabc  
☐ bccbc  
☒ abbbbbb  
☒ ac

`^[test]+[ar]*s[REre]{2}s?s[diDI]+?\$`

Select one or more:

- ☒ esta re DI?  
☒ test re di?  
☐ es a re DI?  
☒ testar er di?  
☐ ta r di?

## Exercício 4 - XPath

```
<?xml version="1.0" encoding="UTF-8"?>
<concurso>
  <concorrentes>
    <concorrente id="c01">
      <nome>Filipe Melo</nome>
      <idade>18</idade>
      <cidade>Coimbra</cidade>
      <categorias>
        <cat tipo="teatro" resultado="reprovado" nota="10"/>
        <cat tipo="dança" resultado="reprovado" nota="5"/>
      </categorias>
    </concorrente>
    <concorrente id="c02">
      <nome>Carla Fontes</nome>
      <idade>28</idade>
      <cidade>Coimbra</cidade>
      <categorias>
        <cat tipo="musica" resultado="aprovado" nota="7"/>
        <cat tipo="dança" resultado="aprovado" nota="8"/>
      </categorias>
    </concorrente>
    <concorrente id="c03">
      <nome>Joana Martins</nome>
      <idade>53</idade>
      <cidade>Lisboa</cidade>
      <categorias>
        <cat tipo="teatro" resultado="aprovado" nota="7"/>
        <cat tipo="dança" resultado="aprovado" nota="9"/>
        <cat tipo="musica" resultado="reprovado" nota="3"/>
      </categorias>
    </concorrente>
    <concorrente id="c04">
      <nome>Ana Ferro</nome>
      <idade>55</idade>
      <cidade>Coimbra</cidade>
      <categorias>
        <cat tipo="teatro" resultado="aprovado" nota="8"/>
        <cat tipo="dança" resultado="reprovado" nota="3"/>
        <cat tipo="musica" resultado="reprovado" nota="5"/>
      </categorias>
    </concorrente>
  </concorrentes>
  <juri>
    <jurado>
      <nome>Sara Silva</nome>
      <local>Lisboa</local>
    </jurado>
    <jurado>
      <nome>André Poeiras</nome>
      <local>Coimbra</local>
    </jurado>
    <jurado>
      <nome>Luis Guerra</nome>
      <local>Porto</local>
    </jurado>
  </juri>
</concurso>
```

## XPATH

- **Nomes (texto) dos concorrentes e dos jurados que vivem em Coimbra.**

`//concorrente[cidade='Coimbra']/nome/text() | //jurado[local='Coimbra']/nome/text()`

- **Nomes (texto) dos concorrentes que moram em Lisboa e têm mais de 30 anos.**

`//concorrente[cidade = 'Lisboa' and idade >30]/nome/texto()`

- **Localidades (elementos cidade) onde moram os concorrentes com mais de 50 anos.**

`//concorrente[idade > 50]/cidade`

- **Qual a idade do concorrente mais velho?**

`//concorrente[idade = max(//idade)]/idade`

- **Todos os concorrentes (elementos nome) que ficaram aprovados a mais do que uma categoria (cat).**

`//concorrente[count(categorias/cat[@resultado = 'aprovado']) > 1]/nome`

- **Quantos concorrentes foram aprovados a pelo menos duas categorias?**

`count(//concorrente[count(categorias/cat[@resultado = 'aprovado']) >= 2])`

- **Id do concorrente com a classificação mais alta à categoria (cat) 'teatro'**

`//concorrente[categorias/cat[@tipo='teatro']/@nota = max(//cat[@tipo='teatro']/@nota)]/@id`

**Transcreva do ficheiro o que é devolvido pela seguinte expressão XPath:**

`/curso/descendant::concorrente[last()]/preceding-sibling::concorrente[position()<3]//cat[@tipo="dança"]//ancestor::* /nome`

`<nome>Joana ...</nome>`

`<nome>Carla ...</nome>`

### Exercício 5 - Escreva o DTD para validar

- Elemento `<concorrente>`  
`<!ELEMENT concorrente (nome,idade,cidade,categorias)>`
- Elemento `<categorias>`  
`<!ELEMENT categorias (cat)+>`
- Elemento `<nome>`  
`<!ELEMENT nome(#PCDATA)>`
- Atributo **resultado** como uma enumeração  
`<!ATTLIST cat resultado (aprovado|reprovado) #REQUIRED>`
- Atributo **tipo** como string  
`<!ATTLIST cat tipo CDATA #REQUIRED>`