



ESCOLA  
SUPERIOR  
DE TECNOLOGIA  
E GESTÃO

## **Licenciatura em Engenharia Informática**

### **Processamento Estruturado de Informação**

Trabalho Prático 1

Grupo: 19

8130372 - 8140176 - 8140211

**Bruno Oliveira<sup>1</sup>, Óscar Oliveira<sup>2</sup>**

[<sup>1</sup>bmo@estg.ipp.pt](mailto:bmo@estg.ipp.pt), [<sup>2</sup>oao@estg.ipp.pt](mailto:oao@estg.ipp.pt)

# Índice

<b>Exemplo de aplicação do schema</b>	<b>12</b>
<b>Consultas XQuery implementadas</b>	<b>13</b>
<b>Transformações XSL implementadas</b>	<b>16</b>
<b>Aplicação desenvolvida em linguagem Java</b>	<b>18</b>
<b>Requisitos não implementados</b>	<b>18</b>
<b>Principais dificuldades e conclusões</b>	<b>19</b>

## 1. Identificação de schemas

### Schema ESTG Mapa

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/epocasAvaliacao">http://namespace.com/epocasAvaliacao</a> <a href="http://namespace.com/curso">http://namespace.com/curso</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	ESTG_Mapas.xsd

Para o elemento: Mapa

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
Descrição	Mapa de exames
Descrição visual	
Descrição da composição do elemento	O elemento Mapa é composto por outros dos elementos complexos, o elemento “epocasAvaliacao” e o elemento “cursos”
Elementos Filho	“cursos” e “epocasAvaliacao”
Exemplo XML	<pre>&lt;mapa&gt;   &lt;epocasAvaliacao&gt; [13 lines]    &lt;cursos&gt; [194 lines] &lt;/mapa&gt;</pre>
Estrutura XSD	O elemento mapa é do tipo complexType que contém uma sequencia de elementos, cujo nome é epocasAvaliacao e cursos;
Dependências de em relação a outras elementos (identificar schema)	xmlns:epav=" <a href="http://namespace.com/epocasAvaliacao">http://namespace.com/epocasAvaliacao</a> "; xmlns:cur=" <a href="http://namespace.com/curso">http://namespace.com/curso</a> ";

## Schema EpocasAvaliacaoType

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/exame">http://namespace.com/exame</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	EpocasAvaliacaoType.xsd

Para o elemento: epocasAvaliacao

Namespace	<a href="http://namespace.com/epocasAvaliacao">http://namespace.com/epocasAvaliacao</a>
Descrição	Epocas de Avaliação
Descrição visual	
Descrição da composição do elemento	O elemento epocasAvaliacao é composto por outros 3 elementos complexos, o elemento “epocaNormal”, o elemento “epocaRecurso” e o elemento “epocaEspecial”
Elementos Filho	“epocaNormal” , “epocaRecurso” e “epocaEspecial”
Exemplo XML	<pre> &lt;epocasAvaliacao&gt;   &lt;epocas semestre="1"&gt;     &lt;epocaNormal&gt; [3 lines]     &lt;epocaRecurso&gt; [3 lines]     &lt;epocaEspecial&gt; [3 lines]   &lt;/epocas&gt;   &lt;epocas semestre="2"&gt;     &lt;epocaNormal&gt; [3 lines]     &lt;epocaRecurso&gt; [3 lines]     &lt;epocaEspecial&gt; [3 lines]   &lt;/epocas&gt; &lt;/epocasAvaliacao&gt; </pre>
Estrutura XSD	O elemento epocasAvaliacao é do tipo complexType que contém uma sequência de 3 elementos , “epocaNormal” , “epocaRecurso” e “epocaEspecial” que por sua vez têm uma sequência de 2 elementos cada que são: “dataInicio” e “dataFim”
Dependências de em relação a outras elementos (identificar schema)	Os elementos não tem dependências.

## Schema CursoType

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/curso">http://namespace.com/curso</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	cursoType.xsd

Para o elemento: Curso

Namespace	<a href="http://namespace.com/curso">http://namespace.com/curso</a>
Descrição	Este schema serve para validar a sintaxe de um XML de curso e todos os seus elementos filho.
Descrição visual	<p>Visual representation of the XSD schema for the 'cursoType' element. The diagram shows the 'cursoType' element with attributes and child elements. The 'cursoType' element is of type 'xs:string' and contains the elements 'nome', 'codigo', 'coordenador', and 'unidadesCurriculares'. Each of these elements is also of type 'xs:string'. The 'unidadesCurriculares' element is of type 'uc:unidadeCurricularType' and contains the element 'unidadeCurricular' of type 'uc:unidadeCurricularType'.</p>
Descrição da composição do elemento	O elemento “curso” é constituído por 4 elementos, sendo eles o nome, o código, coordenador e o elemento complexo unidadesCurriculares.
Elementos Filho	“nome”, “codigo”, “coordenador”, “unidadesCurriculares”.
Exemplo XML	<pre>&lt;curso&gt;   &lt;nome&gt;Licenciatura em Engenharia Informática&lt;/nome&gt;   &lt;codigo&gt;LEI&lt;/codigo&gt;   &lt;coordenador&gt;Ricardo Santos&lt;/coordenador&gt;   &lt;unidadesCurriculares&gt; [107 lines] &lt;/curso&gt;</pre>
Estrutura XSD	O elemento curso é do tipo complexType que contém uma sequência de 3 elementos mais um complexo que por sua vez têm uma sequência de mais elementos
Dependências de em relação a outras elementos (identificar schema)	xmlns:uc="https://namespace.com/unidadecurricular"; O schema “cursoType.xsd” depende de “unidadesCurricularType” para validar unidades curriculares.

**Schema unidadeCurricularType**

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/epocas">http://namespace.com/epocas</a> <a href="http://namespace.com/types">http://namespace.com/types</a> <a href="https://namespace.com/unidadecurricular">https://namespace.com/unidadecurricular</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	unidadeCurricularType.xsd

Para o elemento: unidadeCurricular

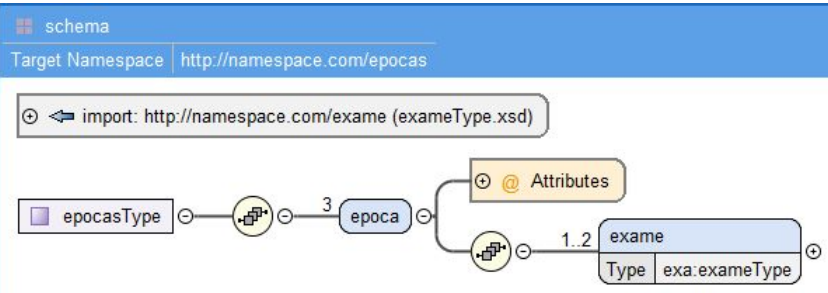
Namespace	<a href="https://namespace.com/unidadecurricular">https://namespace.com/unidadecurricular</a>
Descrição	Este schema serve para validar a sintaxe de um XML de unidadeCurricular e todos os seus elementos filho.
Descrição visual	
Descrição da composição do elemento	O elemento “unidadeCurricular” é constituído por 7 elementos, sendo eles o nome, areaCientifica, anoCurricular, semestre, responsavel, tipologias, epocas.
Elementos Filho	“nome”, “areaCientifica”, “anoCurricular”, “semestre”, ”responsavel”, ”tipologias”, ”epocas”.

Exemplo XML	<pre> &lt;unidadeCurricular cod="PEI"&gt;   &lt;nome&gt;Processamento estruturado de informação&lt;/nome&gt;   &lt;areaCientifica&gt;Informática&lt;/areaCientifica&gt;   &lt;anoCurricular&gt;2&lt;/anoCurricular&gt;   &lt;semestre&gt;1&lt;/semestre&gt;   &lt;responsavel&gt;     &lt;nome&gt;Bruno Oliveira&lt;/nome&gt;     &lt;cod&gt;001&lt;/cod&gt;   &lt;/responsavel&gt;   &lt;tipologias&gt;     &lt;tipologia&gt;TP&lt;/tipologia&gt;   &lt;/tipologias&gt;   &lt;epocas&gt;     &lt;epoca nome="Normal"&gt;       &lt;exame tipologia="Tp"&gt;         &lt;hora&gt;14:00:00&lt;/hora&gt;         &lt;dia&gt;2018-01-15&lt;/dia&gt;         &lt;tipoSala&gt;Informática&lt;/tipoSala&gt;         &lt;sala&gt;Auditório 1&lt;/sala&gt;         &lt;semana&gt;1&lt;/semana&gt;       &lt;/exame&gt;     &lt;/epoca&gt;     &lt;epoca nome="Recurso"&gt; [8 lines]     &lt;epoca nome="Especial"&gt; [8 lines]   &lt;/epocas&gt; &lt;/unidadeCurricular&gt; </pre>
Estrutura XSD	<p>O elemento unidadeCurricular é do tipo complexType que contém uma sequência de elementos, sendo eles: “nome”, “areaCientifica”, “anoCurricular”, “semestre”, “responsavel”, “tipologias”, “epocas”, sendo que, “responsavel” é do tipo complexType e contém um sequência de 2 elementos, “nome” e “cod”.</p>
Dependências de em relação a outras elementos (identificar schema)	<p>xmlns:ep="http://namespace.com/epocas"  xmlns:typ="http://namespace.com/types"  O schema “unidadeCurricular.xsd” depende de “epocas.xsd” para validar epocas e de “types.xsd” para validar o elemento tipologia.</p>

## Schema EpocasType

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/exame">http://namespace.com/exame</a> <a href="http://namespace.com/epocas">http://namespace.com/epocas</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	EpocasType.xsd

Para o elemento: epocas

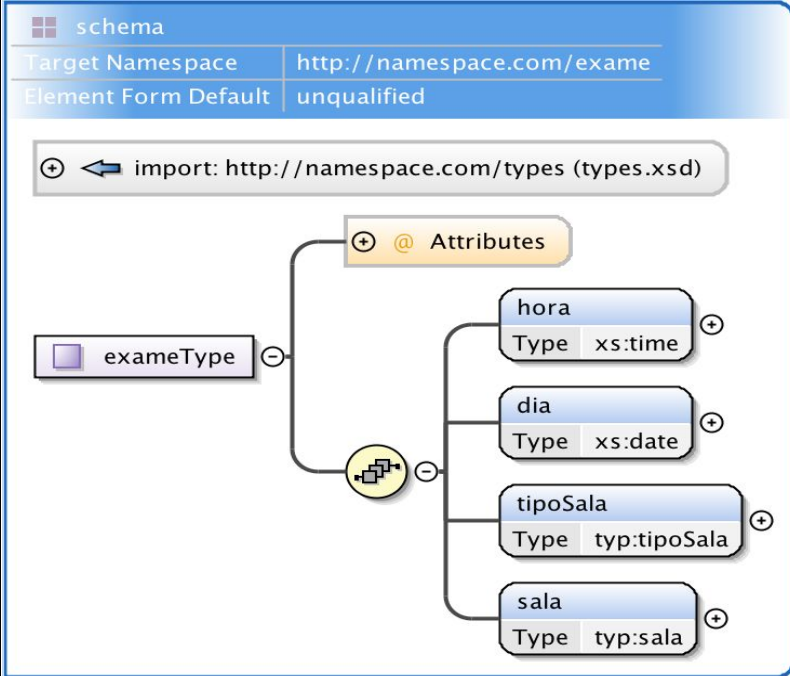
Namespace	<a href="http://namespace.com/epocas">http://namespace.com/epocas</a>
Descrição	Este schema serve para validar a sintaxe de um XML de epocas e todos os seus elementos filho.
Descrição visual	 <p>The visual representation shows a sequence of 3 'epoca' elements. Each 'epoca' element contains an 'exame' element. The 'exame' element is of type 'exa:exameType' and has a cardinality of 1..2. The 'epoca' element has a cardinality of 3. The 'exame' element has an attribute 'nome'.</p>
Descrição da composição do elemento	O elemento “epocas” é constituído por 3 elementos “epoca”, que se referem a cada epoca de exames(normal, recurso e especial)
Elementos Filho	“epoca”
Exemplo XML	<pre> &lt;epocas&gt;   &lt;epoca nome="Normal"&gt; [8 lines]   &lt;epoca nome="Recurso"&gt; [8 lines]   &lt;epoca nome="Especial"&gt; [8 lines] &lt;/epocas&gt; </pre>
Estrutura XSD	O elemento “epocas” é do tipo complexType que contém uma sequência de de 3 elementos “epoca”.
Dependências de em relação a outras elementos (identificar schema)	<p>xmlns:exa="<a href="http://namespace.com/exame">http://namespace.com/exame</a>"</p> <p>O schema “epocasType.xsd” depende de “exame.xsd” para validar exame.</p>



## Schema ExamesType

Namespace	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> <a href="http://namespace.com/exame">http://namespace.com/exame</a>
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	ExamesType.xsd

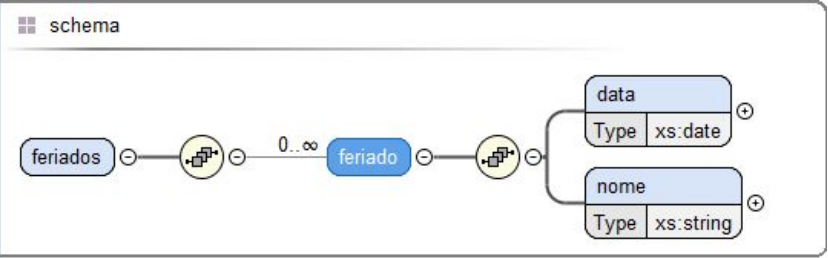
Para o elemento: Exame

Namespace	<a href="http://namespace.com/exame">http://namespace.com/exame</a>
Descrição	Este schema serve para validar a sintaxe de um XML que tenho elementos exame
Descrição visual	
Descrição da composição do elemento	O elemento “exame” é constituído por 5 elementos, sendo eles a hora, dia, tipoSala, sala e semana.
Elementos Filho	“hora”, “dia”, “tipoSala”, “sala”
Exemplo XML	<pre>&lt;exame tipologia="P"&gt;   &lt;hora&gt;14:00:00&lt;/hora&gt;   &lt;dia&gt;2018-03-15&lt;/dia&gt;   &lt;tipoSala&gt;Informática&lt;/tipoSala&gt;   &lt;sala&gt;Auditório 1&lt;/sala&gt; &lt;/exame&gt;</pre>
Estrutura XSD	O elemento exame é do tipo complexType que contém uma sequência de 5 elementos.
Dependências de em relação a outras elementos (identificar schema)	Este elemento depende do elemento épocas que está definido no schema EpocasType.xsd

## Schema feriados

Namespace	http://www.w3.org/2001/XMLSchema
Propriedades	elementFormDefault="unqualified" version="1.0"
Nome do schema	feriados.xsd

Para o elemento: feriados

Namespace	http://www.w3.org/2001/XMLSchema
Descrição	Feriados de um ano civil
Descrição visual	
Descrição da composição do elemento	O elemento “feriados” é constituído por um elemento “feriado”, que se referem a cada feriado num ano civil.
Elementos Filho	“feriado”
Exemplo XML	<pre> &lt;feriados xmlns:xs="http://www.w3.org/2001/XMLSchema"   &lt;feriado&gt;     &lt;data&gt;2018-01-01&lt;/data&gt;     &lt;nome&gt;Ano Novo&lt;/nome&gt;   &lt;/feriado&gt;   &lt;feriado&gt; [3 lines]   &lt;feriado&gt; [3 lines]   &lt;feriado&gt; [3 lines]   &lt;feriado&gt; [3 lines] &lt;/feriados&gt; </pre>
Estrutura XSD	O elemento “feriados” é do tipo complexType que contém uma sequência de de elementos “feriado” do tipo complexType, que por sua vez tem uma sequência de 2 elementos, data e nome do respectivo feriado.
Dependências de em relação a outras elementos (identificar schema)	Este elemento não tem dependências.

## 2. Exemplo de aplicação do schema

Exemplos de documentos XML e explicação do cenário que lhes deu origem.

```
<mapa xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="ESTG_Mapa.xsd">
  <epocasAvaliacao>
    <epocas semestre="1">
      <epocaNormal> [3 lines]
      <epocaRecurso> [3 lines]
      <epocaEspecial> [3 lines]
    </epocas>
    <epocas semestre="2">
      <epocaNormal> [3 lines]
      <epocaRecurso> [3 lines]
      <epocaEspecial> [3 lines]
    </epocas>
  </epocasAvaliacao>
  <cursos>
    <curso>
      <nome>Licenciatura em Engenharia Informática</nome>
      <codigo>LEI</codigo>
      <coordenador>Ricardo Santos</coordenador>
      <unidadesCurriculares>
        <unidadeCurricular cod="PEI">
          <nome>Processamento estruturado de informação</nome>
          <areaCientifica>Informática</areaCientifica>
          <anoCurricular>1</anoCurricular>
          <semestre>2</semestre>
          <responsavel>
            <nome>Bruno Oliveira</nome>
            <cod>001</cod>
          </responsavel>
          <tipologias>
            <tipologia>TP</tipologia>
          </tipologias>
          <epocas>
            <epoca nome="Normal">
              <exame tipologia="T">
                <hora>14:00:00</hora>
                <dia>2018-01-15</dia>
                <tipoSala>Informática</tipoSala>
                <sala>Auditório 1</sala>
              </exame>
            </epoca>
          </epocas>
        </unidadeCurricular>
      </unidadesCurriculares>
    </curso>
  </cursos>
</mapa>
```

O documento ESTG\_Mapa.xml é validado com o documento XSD ESTG\_Mapa.xsd , que por sua vez é validado pelos documentos XSD, epocasAvaliacao.xsd, cursos.xsd.

Cursos.xsd depende do schema unidadeCurricular.xsd, que por sua vez depende de épocas.xsd e de type.xsd. Epocas depende do schema exame.xsd, que por sua vez depende de types.xsd.

Este é o xml geral do nosso trabalho onde ficam armazenados todos os dados relativamente à calendarização de exames, de todos os cursos, para cada época, da ESTG.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <feriados xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="feriados.xsd">
3   <feriado>
4     <data>2018-01-01</data>
5     <nome>Ano Novo</nome>
6   </feriado>
7   <feriado>
8     <data>2018-03-30</data>
9     <nome>Sexta feira Santa</nome>
10  </feriado>
11  <feriado>
12    <data>2018-04-01</data>
13    <nome>Pascoa</nome>
14  </feriado>
15  <feriado>
16    <data>2018-04-25</data>
17    <nome>25 de Abril</nome>
18  </feriado>
19  <feriado>
20    <data>2018-05-01</data>
21    <nome>Dia do Trabalhador</nome>
22  </feriado>
23 </feriados>
24

```

O documento feriados.xml é validado com o documento XSD feriados.xsd, o cenário que deu origem a este xml , foi a possibilidade de coincidirem feriados com épocas de exame .

### 3. Consultas XQuery implementadas

- A primeira expressão implementada com recurso a xquery foi a consulta do código, nome e regente das disciplinas existentes no mapa, e encontra-se presente no ficheiro “codNomeRegente.xqy”.

```
declare function local:codNomeRegente()
{
  for $x in doc('ESTG_Mapa.xml')/mapa/cursos
  return <line>

  {
    for $i in $x/curso/unidadesCurriculares/unidadeCurricular
    return <line><cod>{data($i/@cod)}</cod><nome>{data($i/nome)}</nome><responsavel>{data($i/responsavel/nome)}</responsavel></line>
  } </line>
};

local:codNomeRegente()
```

#### Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<line>
  <line>
    <cod>PEI</cod>
    <nome>Processamento estruturado de informação</nome>
    <responsavel>Bruno Oliveira</responsavel>
  </line>
  <line>
    <cod>ED</cod>
    <nome>Estrutura de Dados</nome>
    <responsavel>Ricardo Santos</responsavel>
  </line>
  <line>
    <cod>ASI</cod>
    <nome>Administração Sistemas Computadores</nome>
    <responsavel>João Magalhães</responsavel>
  </line>
  <line>
    <cod>ED</cod>
    <nome>Estrutura de Dados</nome>
    <responsavel>Ricardo Santos</responsavel>
  </line>
</line>
```

- A segunda expressão implementada com recurso a xquery foi a consulta do número de disciplinas existentes no mapa para um determinado docente(código), e encontra-se presente no ficheiro “unidadesCurricularesDocente.xqy”.

```

declare function local:unidadesCurricularesDocente($cod){
    let $count:=0
    return count(doc('ESTG_MAPA.xml')/mapa/cursos/curso/unidadesCurriculares/unidadeCurricular/responsavel[cod=$cod])
};

local:unidadesCurricularesDocente("001")

```

### **Output:**

```

<?xml version="1.0" encoding="UTF-8"?>
<UnidadesdoDocente>1</UnidadesdoDocente>

```

- A terceira expressão implementada com recurso a xquery foi a consulta do número de exames existentes numa determinada semana, para um dado curso, semestre e unidade curricular, e encontra-se presente no ficheiro “numExams.xqy”.

```

declare function local:calcWeek($semana, $semestre){
    let $dataInicioNormal:= doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaNormal/dataInicio
    let $dataFinNormal:=doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaNormal/dataInicio
    let $dataInicioEspecial:= doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaEspecial/dataInicio
    let $dataFinEspecial:=doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaEspecial/dataInicio
    let $dataInicioRecurso:= doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaRecurso/dataInicio
    let $dataFinRecurso:=doc('ESTG_MAPA.xml')/mapa/epocasAvaliacao/epocas[@semestre = $semestre]/epocaRecurso/dataInicio

    return if ((xs:date($semana)>= xs:date($dataInicioNormal))or (xs:date($semana)<= xs:date($dataFinNormal))) then
        let $s:="Normal"
        return ($s)
    else if ((xs:date($semana)>= xs:date($dataInicioRecurso))or (xs:date($semana)<= xs:date($dataFinRecurso)))then
        let $s:="Recurso"
        return ($s)
    else if((xs:date($semana)>= xs:date($dataInicioEspecial))or (xs:date($semana)<= xs:date($dataFinEspecial)))then
        let $s:="Especial"
        return ($s)
    else
        ("")
};

declare function local:calcWeekExams($semana, $semestre, $curso, $anoCurricular){
    let $x:=local:calcWeek($semana, $semestre)

    return <NumeroExams>{count (doc('ESTG_MAPA.xml')/mapa/cursos/curso[codigo=$curso]/unidadesCurriculares/unidadeCurricular
        [semestre = $semestre and anoCurricular = $anoCurricular]/epocas/epoca[@nome=$x]/exame)}</NumeroExams>
};

local:calcWeekExams("2018-01-15", "1", "LEI", "1")

```

### **Output:**

```

<?xml version="1.0" encoding="UTF-8"?>
<NumeroExams>3</NumeroExams>

```



## 4. Transformações XSL implementadas

Detalhar todos os documentos XSL criados (estrutura, expressões XPath), assim como apresentar exemplos gerados.

O documento XSL criado, gera o mapa de exames para todos os cursos e para todas as épocas de cada curso. Segue abaixo a estrutura do XSL ESTG\_GERALNOVO:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  >
  <xsl:import href="templateSemestre1NormalRecurso.xsl" />
  <xsl:import href="templateSemestre2NormalRecurso.xsl" />
  <xsl:import href="templateEpocaEspecial.xsl" />
  <xsl:template match="/">
    <html>
      <head>
        <link rel="stylesheet" href="mapaGeralStyles.css"/>
        <link href='https://fonts.googleapis.com/css?family=Aladin' rel='stylesheet'/>
      </head>
      <body>
        <header><div id="txt">Mapa de Exames</div><div id="image"></img></div></header>
        <div class="semestre"> <h3>Semestre 1</h3></div>
        <xsl:for-each select="mapa/cursos/curso">

          <xsl:call-template name="semestre1"/>

        </xsl:for-each>
        <div class="semestre"><h3>Semestre 2</h3></div>
        <xsl:for-each select="mapa/cursos/curso">
          <xsl:call-template name="semestre2"/>
        </xsl:for-each>
        <hr/>
        <div class="semestre"><h3>Epoca Especial</h3></div>
        <xsl:for-each select="mapa/cursos/curso">
          <xsl:call-template name="epocaEspecial"/>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Segue abaixo a estrutura XSL do 1º semestre, época normal e época recurso :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:import href="typ.xsl"/>
  <xsl:template name="semestre1">

    <h3><xsl:value-of select="nome"/></h3>
    <div class="epocasDesc">Epoca Normal - <xsl:value-of select="../../epocasAvaliacao/epocas[@semestre='1']/epocaNormal"/></div>
    <table border="1">
      <xsl:call-template name="table"/>
      <xsl:for-each select="unidadesCurriculares/unidadeCurricular">
        <xsl:if test="semestre = 1">
          <xsl:for-each select="epocas/epoca[@nome='Normal']/exame">
            <xsl:call-template name="for"/>
          </xsl:for-each>
        </xsl:if>
      </xsl:for-each>
    </table>
    <div class="epocasDesc">Epoca Recurso - <xsl:value-of select="../../epocasAvaliacao/epocas[@semestre='1']/epocaRecurso"/></div>
    <table border="1">
      <xsl:call-template name="table"/>
      <xsl:for-each select="unidadesCurriculares/unidadeCurricular">
        <xsl:if test="semestre = 1">
          <xsl:for-each select="epocas/epoca[@nome='Recurso']/exame">
            <xsl:call-template name="for"/>
          </xsl:for-each>
        </xsl:if>
      </xsl:for-each>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

Segue abaixo a estrutura XSL do 2º semestre, época normal e época recurso :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:import href="typ.xsl"/>
  <xsl:template match="/" name="semestre2">

    <h3><xsl:value-of select="nome" /></h3>
    <div class="epocasDesc">Epoca Normal - <xsl:value-of select="../../epocasAvaliacao/epocas[@semestre='2']/epocaNormal"/></div>
    <table border="1">
      <xsl:call-template name="table"/>
      <xsl:for-each select="unidadesCurriculares/unidadeCurricular">
        <xsl:if test="semestre = 2">
          <xsl:for-each select="epocas/epoca[@nome='Normal']/exame">
            <xsl:call-template name="for"/>
          </xsl:for-each>
        </xsl:if>
      </xsl:for-each>
    </table>
    <div class="epocasDesc">Epoca Recurso - <xsl:value-of select="../../epocasAvaliacao/epocas[@semestre='1']/epocaRecurso"/></div>
    <table border="1">
      <xsl:call-template name="table"/>
      <xsl:for-each select="unidadesCurriculares/unidadeCurricular">
        <xsl:if test="semestre = 2">
          <xsl:for-each select="epocas/epoca[@nome='Recurso']/exame">
            <xsl:call-template name="for"/>
          </xsl:for-each>
        </xsl:if>
      </xsl:for-each>
    </table>
    <hr/>
  </xsl:template>
</xsl:stylesheet>
```

Segue abaixo a estrutura XSL de commons:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:template name="table">
    <tr>
      <th>Dia</th>
      <th>Ano</th>
      <th>Unidade Curricular</th>
      <th>Hora</th>
      <th>Sala</th>
      <th>Responsavel</th>
    </tr>
  </xsl:template>
  <xsl:template name="for">
    <tr>
      <td><xsl:value-of select="dia"/> -
      </td>
      <td><xsl:value-of select="../../anoCurricular"/></td>
      <td><xsl:value-of select="../../nome"/></td>
      <td><xsl:value-of select="hora"/></td>
      <td><xsl:value-of select="sala"/></td>
      <td><xsl:value-of select="../../responsavel/nome"/></td>
    </tr>
  </xsl:template>
</xsl:stylesheet>
```



Segue abaixo a estrutura XSL do 1º e 2º semestre, época especial :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:import href="typ.xsl"/>
  <xsl:template match="/" name="epocaEspecial">

    <h3><xsl:value-of select="name"/></h3>
    <div class="epocasDesc">Epoca Especial - <xsl:value-of select=".../epocasAvaliacao/epocas[@semestre='1']/epocaEspecial"/></div>
    <table border="1">
      <xsl:call-template name="table"/>
      <xsl:for-each select="unidadesCurriculares/unidadeCurricular">
        <xsl:for-each select="epocas/epoca[@name='Especial']/exame">
          <xsl:call-template name="for"/>
        </xsl:for-each>
      </xsl:for-each>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

Segue abaixo o output da execução do ficheiro XSL:

# Mapa de Exames

**P.PORTO**

ESCOLA  
SUPERIOR  
DE TECNOLOGIA  
E GESTÃO

## Semestre 1

### Licenciatura em Engenharia Informática

Epoca Normal – 2018-01-15 2018-01-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-01-15	1	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-01-15	1	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

Epoca Recurso – 2018-06-15 2018-06-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-06-15 -	1	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-06-15 -	1	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

### Licenciatura em Redes Computadores

Epoca Normal – 2018-01-15 2018-01-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-01-15	2	Administração Sistemas Computadores	14:00:00	Auditório 2	João Magalhães
2018-01-15	2	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-01-15	2	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

### Época Recurso – 2018-06-15 2018-06-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-01-15 -	2	Administração Sistemas Computadores	14:00:00	Auditório 2	João Magalhães
2018-06-15 -	2	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-06-15 -	2	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

## Semestre 2

### Licenciatura em Engenharia Informática

### Época Normal – 2018-09-15 2018-09-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-01-15	1	Processamento estruturado de informação	14:00:00	Auditório 1	Bruno Oliveira

### Época Recurso – 2018-06-15 2018-06-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-03-15 -	1	Processamento estruturado de informação	14:00:00	Auditório 1	Bruno Oliveira

### Licenciatura em Redes Computadores

### Época Normal – 2018-09-15 2018-09-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
-----	-----	--------------------	------	------	-------------

### Época Recurso – 2018-06-15 2018-06-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
-----	-----	--------------------	------	------	-------------

## Epoca Especial

### Licenciatura em Engenharia Informática

### Época Especial – 2018-09-15 2018-09-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-07-15 -	1	Processamento estruturado de informação	14:00:00	Auditório 1	Bruno Oliveira
2018-09-15 -	1	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-09-15 -	1	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

### Licenciatura em Redes Computadores

### Época Especial – 2018-09-15 2018-09-26

Dia	Ano	Unidade Curricular	Hora	Sala	Responsavel
2018-01-15 -	2	Administração Sistemas Computadores	14:00:00	Auditório 2	João Magalhães
2018-09-15 -	2	Estrutura de Dados	10:00:00	P10	Ricardo Santos
2018-09-15 -	2	Estrutura de Dados	14:00:00	Auditório 1	Ricardo Santos

## 5. Aplicação desenvolvida em linguagem Java

Classes desenvolvidas:

- Erros - Classe usada para instanciar uma lista de erros;
- XmlApprover - Classe que contém os caminhos para os ficheiros xml e xsd's respetivos. Nela estão presentes também as três funções para validar as regras de negócio apresentadas no enunciado, bem como funções auxiliares para as regras de negócio;
- XPathEvaluator - Classe que contém métodos utilizados, para a execução de funções XPath com Java;
- Docente - Classe usada para instanciar uma lista de docentes;
- Exame - Classe usada para instanciar uma lista de exames;
- Feriados - Classe para instanciar uma lista de feriados;
- Feriado - Classe que contém os atributos relativos a um feriado;

### Classe XmlApprover

Métodos:

- public boolean validaXML() {} - Método responsável por validar a sintaxe de um ficheiro XML, recorrendo ao respetivo XSD;
- public void read() {} - Método responsável por ler um ficheiro XML e efetuar parse para uma variável do tipo Document;
- public boolean verifyIfcodEcists(ArrayList<Docente> arrayTemp, String cod) {} - Método que percorre um ArrayList de docentes e mediante o código de um docente passado por parâmetro, retorna true se o código existir na lista de docentes;
- public ArrayList<Docente> sortArray() {} - Método Responsável por ordenar uma lista de exames de Docentes;
- private ArrayList<Feriado> convertHolidaysXMLtoList() {} - Método responsável por percorrer um ficheiro XML com os feriados, e converter para um ArrayList do Tipo Feriados;
- public ArrayList<Docente> convertExamsMaptoList() throws ParseException {} - Método responsável por percorrer um ficheiro XML com o mapa de exames, e retorna um ArrayList desordenado dos docentes com os exames de cada um;
- public ArrayList<Errors> runXMLMAP() {} - Método responsável por percorrer o XML de mapas e retorna um ArrayList com os erros relativos a existência de exames em feriados.
- public boolean approveDate(String data) {} - Método responsável, por verificar se a data de um exame que é recebido por parâmetro, existe no Array com os feriados. Caso exista, retorna true;
- public boolean approveNumberExamsDayUC() {} - Método responsável por verificar se uma unidade curricular pode ter no máximo dois registos de avaliação (Teórica e Prática) que terão de ser realizados no mesmo dia. Esta função foi resolvida com recurso a xQuery;
- public boolean approveTeacheronExam() {} - Método responsável por verificar se um docente tem exames à mesma hora e no mesmo dia a Unidades Curriculares diferentes;
- public boolean approveExamHolidays() {} - Método responsável por verificar se existe algum exame marcado em feriados;

## **6. Requisitos não implementados**

Todos os requisitos foram implementados com sucesso.

## **7. Principais dificuldades e conclusões**

Tivemos dificuldade nos seguintes tópicos:

- Desenvolver uma estrutura de XML, de forma a termos a menor redundância possível;
- Desenvolver os métodos em java necessários;
- Namespaces do xsd;