



Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Publicação Electrónica

PROJECTO FINAL

REDE HOSPITALAR

Braga, Janeiro 2013



Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Publicação Electrónica

PROJECTO FINAL

REDE HOSPITALAR

Docente: João Saraiva

Discentes: Hugo Gomes (58536)
Marta Moreira (54459)
Telma Veloso (58521)

Conteúdo

1	Abordagem ao Problema	2
2	Regras de Validação	4
2.1	Expressões Regulares	6
3	Base de Dados Textual	8
4	Transformações	9
4.1	Dropdown Menus	9
4.1.1	Hospitais	9
4.1.2	Centros de Saúde	10
4.1.3	Farmácias	11
4.2	Tabelas	11
4.2.1	Hospitais	12
4.2.2	Centros de Saúde	13
4.3	Links	14
4.4	Formatação Simples de Texto	15
5	Manipulação Dinâmica	16
6	Conclusão	17
	Anexos	18
I	Schema e DTD	18
II	Exemplos XML	28
III	JavaScript	30

1 Abordagem ao Problema

Considerou-se uma aplicação destinada a suprir as necessidades de um veículo de emergência médica que opera exclusivamente no eixo composto pelos distritos de Braga, Porto e Viana do Castelo, e que alterna a sua actividade entre três tipos de unidades de saúde distintas: hospitais, centros de saúde e farmácias; assegurando, desta forma, cuidados urgentes de nível básico, intermédio e avançado. Pretendia-se que a referida aplicação permitisse aos utilizadores consultar todas as unidades de saúde existentes na área considerada, bem como obter direcções de forma a facilitar o encaminhamento dos doentes para uma dessas unidades, em caso de emergência, devendo ainda assegurar uma unidade alternativa, no caso de a primeira unidade não ter disponibilidade de atender a urgência em questão.



Figura 1: Página principal da aplicação GEOClinic.

Tomando em consideração os requisitos supracitados, construiu-se, numa primeira fase, o *template* da aplicação — denominada de GEOClinic —, tendo em consideração que esta deveria ser de fácil utilização e pouco complexa, a nível visual, de forma a tornar o mais directa e objectiva possível a obtenção dos dados requeridos por parte do utilizador. Optou-se, assim, por estruturá-la recorrendo a uma página principal com três separadores — um para cada tipo de unidade de saúde. Na página principal (Figura 1), fez-se um *embed* de um mapa com todas as unidades disponíveis, que, embora seja interactivo, se utilizou por razões meramente estéticas. A efectiva interacção com o utilizador foi implementada recorrendo, em cada uma das páginas referentes às unidades de saúde, à opção de pesquisa definida por intermédio de estruturas do tipo *dropdown menu*, que se considerou serem as mais adequadas no sentido de habilitar, numa fase posterior, o sistema de dinamização. Esta opção foi definida tendo por base o critério geográfico mais relevante para cada um dos tipos de unidades contemplados, sendo que se optou pelo distrito, para o caso dos hospitais, e pelos concelho e freguesia, para os centros de saúde e farmácias, respectivamente. No que concerne a apresentação da informação acerca das unidades de saúde seleccionadas pelo utilizador, pretendia-se garantir que a estrutura assumida fosse de leitura relativamente directa e, acima de tudo, organizada, no sentido de facilitar a navegação e simplificar a obtenção de elementos informativos importantes, em caso

de emergência. Neste sentido, a informação possivelmente mais dispersa e volumosa — aquela referente às urgências, às especialidades, tratamentos e meios de diagnóstico disponíveis — foi organizada em tabelas. A restante informação foi disponibilizada cumprindo uma formatação simultaneamente simples, agradável e pouco confusa, do ponto de vista do utilizador.

Recuperando a questão da obtenção de direcções, assumiu-se uma estratégia que contemplasse o funcionamento da aplicação em situações *online* e *offline* (Figura 2), com base na utilização do Google Maps para o primeiro caso e de ficheiros de extensão *.kml* gerados no Google Earth para a segunda, partindo-se do pressuposto de que o sistema necessita de possuir este programa instalado para poder usufruir da funcionalidade *offline*. Neste contexto, foi criado um *link* que apresenta ao utilizador a possibilidade de *download* do ficheiro *.kml* correspondente ao percurso pretendido. Convém referir que, no caso específico das direcções mapeadas entre um ponto de origem e a unidade de saúde seleccionada, se optou por fixar a posição de partida do veículo de emergência médica na Avenida Central, Braga.

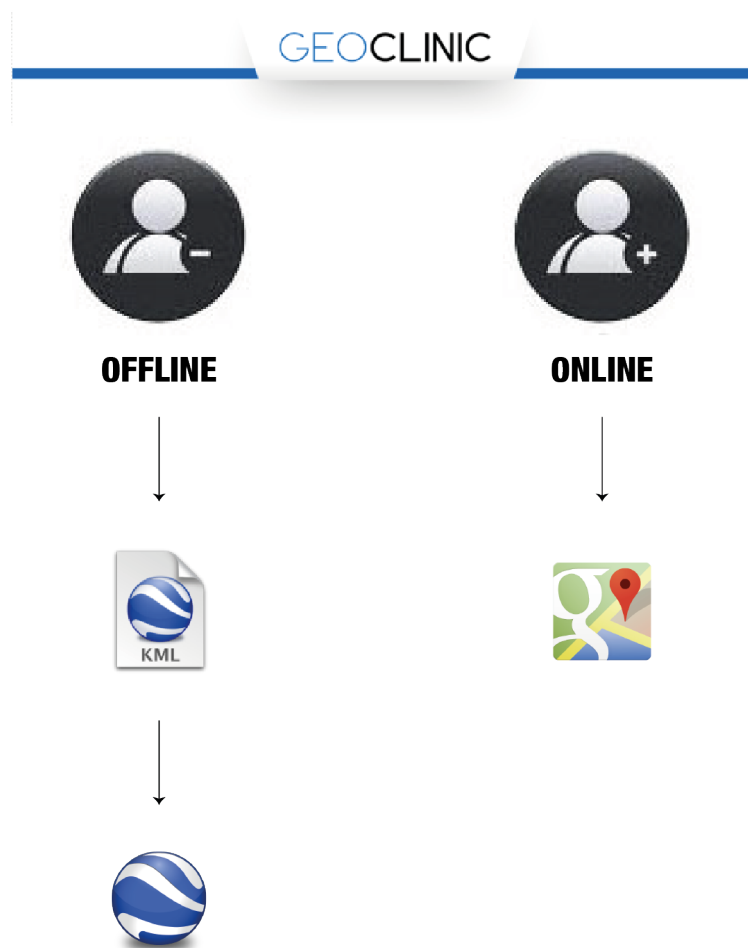


Figura 2: Uso alternativo da funcionalidade de obtenção de direcções utilizando a aplicação em modo *online* e *offline*.

2 Regras de Validação

Numa primeira fase, procedeu-se à análise da informação que se pretendia fazer constar da base de dados textual, de forma a inferir o esquema necessário à sua representação. Dado que se pretendia aproximar a aplicação criada à realidade anteriormente descrita e torná-la, nesse sentido, o mais completa possível, utilizou-se como fonte de informação o Portal da Saúde, página da responsabilidade do Ministério da Saúde que disponibiliza, entre outros, conteúdo respeitante às entidades de saúde pública existentes a nível nacional. Para além da informação constante da página, pretendia-se ainda associar a cada unidade um elemento correspondente às direcções da posição inicial do veículo de emergência até à respectiva unidade — armazenadas sob a forma de um *link* da aplicação Google Maps —, sendo que, no caso específico dos hospitais e dos centros de saúde, se definiu ainda um segundo elemento adicional, respeitante às direcções entre a respectiva unidade e o hospital mais próximo. Considerou-se, neste caso, que um paciente que seja encaminhado para um centro de saúde só poderá ser reencaminhado para um hospital e nunca para um outro centro de saúde, e que existe a possibilidade de um dado hospital não ter condições (quer humanas, quer logísticas) para receber o paciente transportado; por outro lado, considerou-se que um paciente apenas é encaminhado para uma farmácia quando a sua condição médica não inspira cuidados de maior e pode ser abordada pelos tratamentos disponíveis neste espaço (cuja utilidade se encontra limitada, em casos de urgência, à assistência em situações de choque anafilático [1] e epistaxe (sangramento nasal)), sendo que a inclusão desta unidade se deve, ainda, a garantir uma segurança para o fornecimento de fármacos, no caso improvável de a ambulância não estar devida ou suficientemente equipada.

Assim sendo, concluiu-se que os elementos básicos a incluir nas regras de validação de uma unidade de saúde seriam:

1. Nome (**nome**);
2. Morada (**morada**);
3. Coordenadas (**coordenadas**);
4. Direcções (**direccoes**);
5. Contacto (**contacto**);

Para todos as unidades de saúde, a Morada é decomposta nos elementos Endereço (**endereco**), Freguesia (**freguesia**), Concelho (**concelho**), Distrito (**distrito**) e Código Postal (**cpostal**), sendo que o Contacto, no caso dos hospitais e centros de saúde, é decomposto em Telefone (**telefone**), Fax (**fax**) e Email (**email**), e apenas em Telefone, no caso das farmácias. Os hospitais e os centros de saúde, em específico, são ainda respectivamente compostos por:

1. Hospital Mais Próximo (**hprox**);
 2. Urgência (**urgencia**);
-
1. Hospital Mais Próximo (**hprox**);
 2. Tratamento (**tratamento**);

3. Meios de Diagnóstico (*meiosdg*).

A Urgência, que pode ocorrer nenhuma ou mais vezes por cada unidade de saúde, faz referência ao Nível (*nivel*), Tipo (*tipo*), Horário Geral (*horariogeral*) e Especialidade (*especialidade*), sendo que, relativamente a estes últimos, o Horário Geral se divide em Dias (*dias*) e Horas (*horas*), e a Especialidade — que pode também ocorrer zero ou mais vezes — se subdivide na Descrição (*descricao*) e Horário (*horario*); por sua vez, tanto o Tratamento como os Meios de Diagnóstico sofrem a mesma subdivisão em Descrição e Horário.

Devido à sua extensão, tanto os Schemas como os DTD foram excluídos do corpo do relatório e podem ser consultados no Anexo I. Embora se tenha, também, construído os DTDs respeitantes a cada uma das unidades de saúde, apenas os Schemas foram utilizados para efeitos de validação, no âmbito deste trabalho (embora se tenha comprovado que os documentos XML produzidos são válidos para esses mesmos DTDs). No que concerne o tipo de dados utilizados, definiu-se a sua totalidade como sendo `xs:string`, à exceção dos atributos associados aos *empty elements* que contêm *links*, cujo tipo é `xs:anyURI`. A título de exemplo, apresenta-se uma instância de cada uma das estruturas básicas utilizadas na construção das regras:

1. Atributos

```
<xs:attribute name="value" type="xs:string"/>
```

2. Elementos Simples

```
<xs:element name="nome" type="xs:string"/>
```

3. Elementos Complexos

```
<xs:element name="morada">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="endereco"/>
      <xs:element ref="freguesia"/>
      <xs:element ref="concelho"/>
      <xs:element ref="distrito"/>
      <xs:element ref="postal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

4. Elementos Vazios

```
<xs:element name="direccoes">
  <xs:complexType>
    <xs:attribute name="value" type="xs:anyURI"/>
  </xs:complexType>
</xs:element>
```

2.1 Expressões Regulares

As expressões regulares foram implementadas recorrendo a construições do tipo padrão, definidas em XML utilizando construções do género:

```
<xs:element name="<nome>">
  <xs:simpleType>
    <xs:restriction base="<tipo>">
      <xs:pattern value="<expressão regular>" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Definiram-se expressões regulares para os elementos **telefone** e **fax** (de forma semelhante), **cpostal**, **coordenadas** e **horas**, de forma a apenas torná-los válidos mediante uma dada forma de construção, conforme se demonstra abaixo:

1. **Telefone:** os elementos **telefone** foram restringidos de forma a conterem exactamente nove dígitos.

```
<xs:element name="telefone">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{9}">
    </xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

2. **Código Postal:** os elementos **cpostal** foram restringidos de forma a compreenderem uma combinação de três dígitos — iniciada necessariamente por um 4 (dado que é o algarismo comum aos códigos postais dos distritos do Braga, Porto e Viana do Castelo) —, seguida de um hífen e de uma combinação de três dígitos.

```
<xs:element name="cpostal">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[4]\d{3}-\d{3}">
    </xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

3. **Coordenadas:** os elementos **coordenadas** foram restringidos a um padrão composto por um sinal negativo opcional ('?'), seguido de uma combinação de um ou mais dígitos ('\d+'), que pode opcionalmente ser seguido de um '.' e uma outra combinação de um ou mais dígitos. Esta parcela é separada

por uma vírgula de outra parcela semelhante, contando que as coordenadas geográficas encontram-se, normalmente, na forma '41.407684, -8.522659'. Adicionalmente, depois da vírgula podem surgir espaços em branco ('\s*').

```
<xs:element name="coordenadas">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="(\-?\d+(\.\d+)?),\s*(\-?\d+(\.\d+)?)">
    </xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

4. **Horas:** os elementos **horas** foram restringidos de forma a validarem apenas construções horárias do tipo '00:00' a '23:59', seguidas de um hífen e de uma construção semelhante, no sentido de denotar um horário de funcionamento que compreenda um horário de início e um horário de término de actividade.

```
<xs:element name="horas">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([01]?[0-9]|2[0-3]):[0-5][0-9]-([01]?[0-9]|2[0-3]):[0-5][0-9]">
    </xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

3 Base de Dados Textual

Dado o elevado volume de informação associado à representação da totalidade das unidades de saúde da área geográfica considerada — e de forma a facilitar a manutenção posterior da aplicação — optou-se por considerar três ficheiros de extensão `.xml` distintos (*vide* Anexo II). Conforme mencionado (Secção 2) anteriormente, a informação utilizada para popular as etiquetas foi retirada do Portal da Saúde, de forma a tornar a aplicação o mais fidedigna e aproximada da realidade possível. A única particularidade na construção da base prendeu-se com a necessidade de, para todos os *links* considerados, ter sido necessário substituir os símbolos *ampersand* (&) por `&`, na medida em que estes fazem parte do conjunto de caracteres especiais da linguagem XML e, caso não se proceda à referida modificação, ocorrerão erros ao nível do *parsing* da informação.

4 Transformações

No que concerne a informação constante da base de dados textual, e de acordo com a idealização gráfica da aplicação, as transformações foram empregues na criação de *dropdown menus*, tabelas e *links*, e na formatação simples de texto.

4.1 Dropdown Menus

Os *dropdown menus* foram utilizados no sentido de permitir a selecção de uma unidade de saúde específica, partindo de uma restrição geográfica. Assim, para cada uma das páginas referentes ao tipo de unidade de saúde, foram construídos dois menus, interligados e dinamizados através da utilização de JavaScript, conforme será explicitado na Secção 5.

4.1.1 Hospitais

Dada a distribuição geográfica dos hospitais considerados, optou-se por considerar o distrito como elemento constringente e, contando que cada hospital possui uma declaração do distrito em que se encontra localizado, aplicou-se um filtro, tendo em vista a eliminação dos elementos repetidos. Assim, aplicou-se um elemento cíclico para percorrer cada hospital e seleccionar todos os elementos XML do tipo `distrito`, que inclui o trecho `[not(distrito=preceding::distrito)]">`, destinado à eliminação das repetições.

```
<xsl:for-each select="/unidadesaude/hospital/morada[not(distrito=preceding::distrito)]">
```

Implementou-se ainda um elemento ordenador `<xsl:sort select="distrito"/>`, de forma a obter a informação alfabeticamente ordenada. A chamada dos valores às opções foi efectuada através do elemento `<xsl:value-of select="distrito"/>`, sendo que a estrutura completa da transformação que produz o *dropdown menu* referente ao distrito é definida através de:

```
<select id="distrito" style="display:block" onchange="go()">
  <option value="" selected="Y">Selecione um Distrito</option>
  <xsl:for-each select="/unidadesaude/hospital/morada[not(distrito=preceding::distrito)]">
    <xsl:sort select="distrito"/>
    <option>
      <xsl:value-of select="distrito"/><xsl:text> </xsl:text></option>
    </xsl:for-each>
</select>
```

O segundo *dropdown menu* construído, que lista os hospitais existentes para um dado distrito, consiste, na prática, em três menus distintos, devido a dificuldades ao nível da codificação em JavaScript. Neste cenário, tornou-se então necessário criar um menu para cada um dos distritos existentes, aplicando um elemento de restrição `<xsl:if>` a cada um deles. À semelhança do menu anterior, recorreu-se a um *loop* `<xsl:for-each>` para percorrer os elementos do tipo `hospital` e um `<xsl:sort>` para ordenar o *output*. A chamada dos valores foi introduzida através do elemento `<xsl:value-of select="nome"/>`, que dispensa a identificação dos nodos antecessores por estes estarem discriminados no elemento cíclico. A título de exemplo, apresentam-se as estruturas dos menus que permitem seleccionar o distrito pretendido e os hospitais do distrito de Braga, respectivamente.

1. Selecção do distrito.

```
<select id="distrito" style="display:block" onchange="go()">
  <option value="" selected="Y">Selecione um Distrito</option>
  <xsl:for-each select="/unidadesaude/hospital/morada[not(distrito=preceding::distrito)]">
    <xsl:sort select="distrito"/>
    <option>
      <xsl:value-of select="distrito"/><xsl:text> </xsl:text></option>
    </xsl:for-each>
  </select>
```

2. Selecção do hospital, considerando a escolha do distrito de Braga.

```
<select id="h_braga" style="display:none" onchange="go2()">
  <option value="" selected="Y">Selecione um Hospital</option>
  <xsl:for-each select="/unidadesaude/hospital">
    <xsl:sort select="nome"/>
    <xsl:if test="morada/distrito='Braga'">
      <option>
        <xsl:value-of select="nome"/><xsl:text> </xsl:text></option>
      </xsl:if>
    </xsl:for-each>
  </select>
```

Em ambos os exemplos de estrutura utilizados, as especificações **id**, **style** e **onchange** dizem respeito a elementos necessários à dinamização dos *dropdown menus* e serão devidamente justificadas na Secção 5.

4.1.2 Centros de Saúde

A abordagem à informação relativa aos centros de saúde foi semelhante à anteriormente descrita, variando apenas o critério geográfico de selecção, que passa a ser o concelho. Da mesma forma, foi necessário criar um *dropdown menu* para cada um dos concelhos considerados. De seguida, apresentam-se os menus implementados no contexto deste tipo de unidade de saúde em particular, considerando o exemplo do concelho de Barcelos, no segundo menu.

1. Selecção do concelho.

```
<select id="concelho" style="display:block" onchange="go()">
  <option value="" selected="Y">Selecione um Concelho</option>
  <xsl:for-each select="/unidadesaude/csaude/morada[not(concelho=preceding::concelho)]">
    <xsl:sort select="concelho"/>
    <option>
      <xsl:value-of select="concelho"/><xsl:text> </xsl:text></option>
    </xsl:for-each>
  </select>
```

2. Selecção do centro de saúde, considerando a escolha do concelho de Barcelos.

```
<select id="cs_barcelos" style="display:none" onchange="go2()">
  <option value="" selected="Y">Selecione um Centro de Saúde</option>
  <xsl:for-each select="/unidadesaude/csaude">
    <xsl:sort select="nome"/>
    <xsl:if test="morada/concelho='Barcelos'">
      <option>
        <xsl:value-of select="nome"/><xsl:text> </xsl:text></option>
      </xsl:if>
    </xsl:for-each>
  </select>
```

4.1.3 Farmácias

No que concerne as farmácias, o critério considerado foi, por sua vez, a freguesia, para cada qual se procedeu à criação de um *dropdown menu*, no seguimento da ideia apresentada até este ponto.

1. Selecção da freguesia.

```
<select id="freguesia" style="display:block" onchange="go3()">
  <option value="" selected="Y">Selecione uma Freguesia</option>
  <xsl:for-each select="/unidadesaude/farmacia/morada[not(freguesia=preceding::freguesia)]">
    <xsl:sort select="freguesia"/>
    <option>
      <xsl:value-of select="freguesia"/><xsl:text> </xsl:text></option>
    </xsl:for-each>
  </select>
```

2. Selecção da farmácia, considerando a escolha da freguesia da Sé.

```
<select id="se" style="display:none" onchange="go9()">
  <option value="" selected="Y">Selecione uma Farmácia</option>
  <xsl:for-each select="/unidadesaude/farmacia">
    <xsl:sort select="nome"/>
    <xsl:if test="morada/freguesia='Sé'">
      <option>
        <xsl:value-of select="nome"/><xsl:text> </xsl:text></option>
      </xsl:if>
    </xsl:for-each>
  </select>
```

4.2 Tabelas

Tendo em consideração que a aplicação se destina a situações de emergência, um dos critérios essenciais era o de garantir a disponibilização da informação da forma mais inteligível e organizada possível, pelo que

se julgou a estruturação em tabelas como sendo a forma mais adequada à representação dos casos específicos em que o volume de informação provava ser relativamente elevado, como sejam os caso da descrição das urgências e especialidades disponíveis, nas unidades hospitalares, e os tratamentos e meios de diagnóstico, nos centros de saúde.

4.2.1 Hospitais

No contexto dos hospitais, construíram-se as transformações relativas à informação das urgências — tipo, nível e horário geral de funcionamento — e das especialidades — descrição e horário de funcionamento. Dado que nem todos os hospitais possuem serviço de urgência e a respectiva definição das especialidades disponíveis nesses serviços, tornou-se necessário adoptar uma estratégia capaz de assegurar que as tabelas apenas seriam criadas e populadas no caso de os elementos **urgencia** e **especialidade** efectivamente existirem. A solução encontrada foi a de envolver a tabela correspondente às urgências numa estrutura condicional que testa se o nodo **urgencia** é não vazio, bem como associar um processo semelhante para as especialidades, mas envolto, por sua vez, num *loop* condicional que pressupõe a existência de uma urgência.

De forma a simplificar e tornar mais claros os exemplos apresentados, ao nível das transformações aplicadas, dispensaram-se, aqui, alguns elementos HTML de formatação, como sejam a definição das fontes, os parágrafos e a indentação.

1. Transformação relativa ao serviço de urgência.

```
<xsl:if test="urgencia != ''">
  <b>Urgência</b>
  <table border="1">

    <tr>
      <th align="center"><b>Nível</b>&#160;</th>
      <th align="center"><b>Tipo</b>&#160;</th>
      <th align="center"><b>Horário Geral</b></th>
    </tr>

    <xsl:for-each select="urgencia[boolean(*)]">
      <tr>
        <td align="center"><xsl:value-of select="nivel"/></td>
        <td><xsl:value-of select="tipo"/></td>
        <td><xsl:value-of select="horariogeral/dias"/>&#160; <xsl:value-of select="hora
          riogeral/horas"/></td>
      </tr>
    </xsl:for-each>

  </xsl:if>

</table>
```

2. Transformação relativa às especialidades inerentes ao serviço de urgência.

```

<xsl:for-each select="urgencia[boolean(*)]">
  <xsl:if test="especialidade != ''">
    <b>Especialidades</b></font>
  </xsl:if>
</xsl:for-each>

<xsl:for-each select="urgencia[boolean(*)]">
  <xsl:if test="especialidade != ''">

    <table border="1">
      <tr>
        <th align="center"><b>Descrição</b></font>&#160;</th>
        <th align="center"><b>Horário</b></font>&#160;</th>
      </tr>
      <xsl:for-each select="especialidade">
        <tr>
          <td><xsl:value-of select="descricao"/></td>
          <td><xsl:value-of select="horario/dias"/>&#160;<xsl:value-of select="horario/horas"/></td>
        </tr>

      </xsl:for-each>
    </table>
  </xsl:if>
</xsl:for-each>

```

4.2.2 Centros de Saúde

A construção das tabelas relativas à informação contida nos nodos `tratamento` e `meiosdg` foi conduzida de forma semelhante à descrita anteriormente, distinguindo-se apenas pelo facto de nenhum dos nodos considerados ser descendente de outro que não a raiz.

1. Transformação relativa aos tratamentos disponíveis.

```

<xsl:if test="tratamento != ''">
  <b>Tratamentos Disponíveis</b>
  <table border="1">

    <tr>
      <th align="center"><b>Descrição</b>&#160;</th>
      <th align="center"><b>Horário</b>&#160;</th>
    </tr>

    <xsl:for-each select="tratamento[boolean(*)]">
      <tr>
        <td><xsl:value-of select="descricao"/></td>

```

```

        <td><xsl:value-of select="horario/horas"/></td>
      </tr>
    </xsl:for-each>

  </table>

```

2. Transformação relativa aos meios de diagnóstico disponíveis.

```

<xsl:if test="meiosdg != ''">
  <b>Meios de Diagnóstico Disponíveis</b>
  <table border="1">

    <tr>
      <th align="center"><b>Descrição</b>&#160;</th>
      <th align="center"><b>Horário</b>&#160;</th>
    </tr>

    <xsl:for-each select="meiosdg[boolean(*)]">
      <tr>
        <td><xsl:value-of select="descricao"/></td>
        <td><xsl:value-of select="horario/horas"/></td>
      </tr>
    </xsl:for-each>

  </table>

```

4.3 Links

No contexto deste trabalho, foram definidos dois tipos de *links*, sendo que apenas aqueles cujo valor se encontra armazenado na base de dados foi sujeite a transformação, isto é, os *links* referentes à obtenção de direcções quando em modo de funcionamento *online*. A implementação transformada da hiperligação foi efectuada declarando um atributo `href` ao qual foi atribuído o valor contido no elemento `hprox` — um endereço para o respectivo percurso entre a unidade considerada e o hospital mais próximo. O mesmo procedimento foi utilizado para o *link* que permite obter direcções a partir do local de origem fixo e a unidade até à qual se pretende viajar, substituindo-se, neste caso, o elemento `hprox` por `direccoes`. Segue-se um exemplo da implementação deste elemento:

1. Transformação relativa ao *link* que permite obter direcções até aos hospital mais próximo da unidade consultada.

```

<a><xsl:attribute name="href">
  <xsl:value-of select="hproximo/@value"/></xsl:attribute>
  <u>Hospital Mais Próximo | Direcções</u>
</a>

```

Os casos dos centros de saúde e das farmácias foram abordados nos mesmos parâmetros, destacando-se que as farmácias compreendem apenas a funcionalidade de obtenção de direcções do ponto de origem até ao estabelecimento, dado que, conforme adiantado na Secção 2, se partiu do princípio de que as situações encaminhadas para este tipo de unidades não requerem a prestação de cuidados de nenhuma das outras unidades.

4.4 Formatação Simples de Texto

Toda a informação não incluída nas Subsecções 4.1 - 4.3 - que consiste basicamente nas informações logísticas das unidades de saúde — foi simplesmente tratada como texto corrido. Neste sentido, as transformações aplicadas foram básicas e directas, estando as atenções mais focadas na formatação através de código HTML. A título de exemplo, expõem-se os casos da transformação do nome de uma dada unidade e do contacto da mesma, sendo que, neste contexto, se incluem os elementos HTML.

1. Transformação relativa ao nome de uma unidade de saúde

```
<font color="#0066FF" face="Arial, Helvetica, sans-serif" size="5"><xsl:value-of select="nome"/></font>
```

2. Transformação relativa ao contacto de uma unidade de saúde.

```
<span style="padding-left: 20px; display:block">
  <font color="#0066FF" face="Arial, Helvetica, sans-serif" size="2.8"><b>Contac
to</b></font>&#160;
</span>
```

```
<span style="padding-left: 40px; display:block">
  <font color="#0066FF" face="Arial, Helvetica, sans-serif" size="2.5"><b>Telef
one:</b></font>&#160;
  <font face="Arial, Helvetica, sans-serif" size="2.3"><xsl:value-of select="con
tacto/telefone"/></font>
```

```
<font color="#0066FF" face="Arial, Helvetica, sans-serif" size="2.5"><b>Fax:<
/b></font>&#160;
<font face="Arial, Helvetica, sans-serif" size="2.3"><xsl:value-of select="con
tacto/fax"/></font>
```

```
<font color="#0066FF" face="Arial, Helvetica, sans-serif" size="2.5"><b>Email:
</b></font>&#160;
<font face="Arial, Helvetica, sans-serif" size="2.3"><xsl:value-of select="con
tacto/email"/></font>
```

```
</span>
```

5 Manipulação Dinâmica

A criação dos menus *dropdown* foi inicialmente pensada de forma a dinamizar a opção de pesquisa, nomeadamente através da influência da escolha do utilizador relativa ao critério geográfico de selecção, que deveria ditar o conteúdo do menu respeitante às unidades de saúde disponíveis. Numa primeira fase, procurou-se estabelecer a interacção entre as estruturas referidas através de um código JavaScript universal, tentativa que não sucedeu e forçou a procura de um método de implementação alternativo. A solução construída não é, reconhecidamente, a melhor forma de atingir o objectivo da dinamização, mas realiza eficazmente a acção pretendida. Esta baseia-se na associação de uma identificação única (*id*), a um estilo (*style*) e a um evento (*onchange*) que dispara uma função aquando da modificação do respectivo elemento — neste caso, sempre que o utilizador selecciona uma opção —, ao *select* da estrutura de cada um dos menus. Da mesma forma, cada declaração completa de informação a apresentar acerca das unidades de saúde é envolvida numa estrutura do tipo *form*, na qual se declaram igualmente uma identificação, um estilo e um evento. À excepção dos menus de localização geográfica, todos os restantes elementos foram inicializados com um *style* que os torna ocultos da aplicação.

Com base neste cenário, criaram-se dois *scripts* (para cada uma das páginas referentes às unidades de saúde) do tipo "text/javascript", que acomodam duas funções distintas: *go()* e *go2()* (Anexo III). A função *go()* parte da declaração de uma variável que vai armazenar o valor seleccionado pelo utilizador no primeiro menu *dropdown* para compará-la iterativamente com as localizações geográficas disponíveis. Sempre que a comparação suceder, o *style* do menu correspondente à selecção é alterado de forma a que este fique visível, permitindo assim que o utilizador proceda a uma segunda selecção, desta feita de uma das unidades de saúde que compõem o conjunto restringido pela selecção anterior. Por sua vez, a função *go2()* compreende um conjunto de variáveis que armazenam os valores seleccionados pelo utilizador e, seguindo a mesma linha de raciocínio iterativo, estabelece tentativas de comparação até que exista correspondência, alterando, de seguida, o *style* de visibilidade do elemento correspondente, possibilitando, por fim, a consulta da informação da unidade pretendida.

6 Conclusão

Implementou-se com sucesso uma aplicação destinada a suprir as necessidades de um veículo de emergência médica relativamente às funcionalidades de navegação e consulta de informação acerca das unidades de saúde disponíveis no seu perímetro de influência. Considera-se que o sistema desenvolvido se revela, acima de tudo, prático e *user friendly*, sendo que a implementação procurou potenciar a simplicidade de manuseamento e a descomplicação visual, factores decisivos em situações que requeiram o acesso rápido à informação, como é o caso da assistência prestada pela tripulação especializada de uma rede de ambulâncias. Uma outra vantagem do sistema desenvolvido prende-se com a capacidade de funcionamento na presença e na ausência de ligação à rede, devido à construção de ficheiros de extensão *.kml* que veiculam a informação relacionada com a obtenção de direcções independentemente do estado da conexão.

No que concerne possíveis melhorias a efectuar, aponta-se como principal lacuna a forma como a manipulação dinâmica foi implementada. Pese embora o desempenho da aplicação não seja directamente afectado ou comprometido pela abordagem adoptada, no caso de, por exemplo, se pretender adicionar informação à base de dados textual, esta será reconhecida e apresentada nos menus *dropdown*, mas não será possível aceder a informação detalhada sem antes definir as estruturas correspondentes aos elementos acrescentados.

Anexos

I Schema e DTD

Os Schemas e DTDs construídos no sentido de validar a informação contida nos documentos XML foram:

1. Hospitais

(a) Schema

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- definition of attributes -->

  <xs:attribute name="value" type="xs:string"/>

  <!-- definition of simple elements -->

  <xs:element name="nome" type="xs:string"/>
  <xs:element name="endereco" type="xs:string"/>
  <xs:element name="freguesia" type="xs:string"/>
  <xs:element name="concelho" type="xs:string"/>
  <xs:element name="distrito" type="xs:string"/>
  <xs:element name="email" type="xs:string"/>
  <xs:element name="nivel" type="xs:string"/>
  <xs:element name="tipo" type="xs:string"/>
  <xs:element name="descricao" type="xs:string"/>
  <xs:element name="dias" type="xs:string"/>
  <xs:element name="href" type="xs:string"/>

  <!-- definition of complex elements -->

  <xs:element name="hproximo">
    <xs:complexType>
      <xs:attribute name="value" type="xs:anyURI"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="direccoes">
    <xs:complexType>
      <xs:attribute name="value" type="xs:anyURI"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="morada">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="endereco"/>
        <xs:element ref="freguesia"/>
        <xs:element ref="concelho"/>
        <xs:element ref="distrito"/>
        <xs:element ref="cpostal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="cpostal">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[4]\d{3}-\d{3}">
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="coordenadas">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="(\-?\d+(\.\d+)?),\s*(\-?\d+(\.\d+)?)">
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="contacto">
<xs:complexType>
<xs:sequence>
  <xs:element ref="telefono"/>
  <xs:element ref="fax" minOccurs="0"/>
  <xs:element ref="email" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="telefono">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{9}">
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="fax">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{9}">
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="urgencia">
<xs:complexType>
<xs:sequence>
  <xs:element ref="nivel"/>

```

```

        <xs:element ref="tipo"/>
        <xs:element ref="horariogeral"/>
        <xs:element ref="especialidade" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="especialidade">
<xs:complexType>
<xs:sequence>
    <xs:element ref="descricao" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="horario" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horario">
<xs:complexType>
<xs:sequence>
    <xs:element ref="dias"/>
    <xs:element ref="horas"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horariogeral">
<xs:complexType>
<xs:sequence>
    <xs:element ref="dias"/>
    <xs:element ref="horas"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horas">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:pattern value="([01]?[0-9]|2[0-3]):[0-5][0-9]-([01]?[0-9]|2[0-3]):[0-5][0-9]">
    </xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="unidadesaude">
    <xs:complexType>
<xs:sequence>

<!-- hospital -->

<xs:element name="hospital" maxOccurs="unbounded">
    <xs:complexType>
    <xs:sequence>
        <xs:element ref="nome"/>
        <xs:element ref="morada"/>

```

```

        <xs:element ref="coordenadas"/>
        <xs:element ref="direccoes"/>
        <xs:element ref="hproximo"/>
        <xs:element ref="contacto"/>
        <xs:element ref="urgencia" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>

```

(b) DTD

```

<!DOCTYPE unidadesaude [
<!ELEMENT unidadesaude (hospital+)>
    <!ELEMENT hospital (nome,morada,coordenadas,direccoes,hproximo,contacto,urgencia)>
    <!ELEMENT nome (#PCDATA)>
    <!ELEMENT morada (endereco,freguesia,concelho,distrito,cpostal)>
        <!ELEMENT endereco (#PCDATA)>
        <!ELEMENT freguesia (#PCDATA)>
        <!ELEMENT concelho (#PCDATA)>
        <!ELEMENT distrito (#PCDATA)>
        <!ELEMENT cpostal (#PCDATA)>
    <!ELEMENT coordenadas (#PCDATA)>
    <!ELEMENT direccoes (#PCDATA)>
        <!ATTLIST direccoes value CDATA #IMPLIED>
    <!ELEMENT hproximo (#PCDATA)>
        <!ATTLIST hproximo value CDATA #IMPLIED>
    <!ELEMENT contacto (telefone,fax,email)>
        <!ELEMENT telefone (#PCDATA)>
        <!ELEMENT fax (#PCDATA)>
        <!ELEMENT email (#PCDATA)>
    <!ELEMENT urgencia (nivel,tipo,horariogeral,especialidade*)>
        <!ELEMENT nivel (#PCDATA)>
        <!ELEMENT tipo (#PCDATA)>
        <!ELEMENT horariogeral (dias,horas)>
            <!ELEMENT dias (#PCDATA)>
            <!ELEMENT horas (#PCDATA)>
        <!ELEMENT especialidade (descricao,horario)>
            <!ELEMENT descricao (#PCDATA)>
            <!ELEMENT horario (dias,horas)>
                <!ELEMENT dias (#PCDATA)>
                <!ELEMENT horas (#PCDATA)>
    ]>

```

2. Centros de Saúde

(a) Schema

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

```

```

<!-- definition of attributes -->

<xs:attribute name="value" type="xs:string"/>

<!-- definition of simple elements -->

<xs:element name="nome" type="xs:string"/>
<xs:element name="endereco" type="xs:string"/>
<xs:element name="freguesia" type="xs:string"/>
<xs:element name="concelho" type="xs:string"/>
<xs:element name="distrito" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="descricao" type="xs:string"/>
<xs:element name="dias" type="xs:string"/>

<!-- definition of complex elements -->

<xs:element name="hproximo">
  <xs:complexType>
    <xs:attribute name="value" type="xs:anyURI"/>
  </xs:complexType>
</xs:element>

<xs:element name="direccoes">
  <xs:complexType>
    <xs:attribute name="value" type="xs:anyURI"/>
  </xs:complexType>
</xs:element>

<xs:element name="morada">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="endereco"/>
      <xs:element ref="freguesia"/>
      <xs:element ref="concelho"/>
      <xs:element ref="distrito"/>
      <xs:element ref="cpostal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="cpostal">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[4]\d{3}-\d{3}">
      </xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="coordenadas">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="(\-?\d+(\.\d+)?),\s*(\-?\d+(\.\d+)?)">

```

```

        </xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="contacto">
<xs:complexType>
<xs:sequence>
    <xs:element ref="telefone"/>
    <xs:element ref="fax" minOccurs="0"/>
    <xs:element ref="email" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="telefone">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:pattern value="\d{9}">
    </xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="fax">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:pattern value="\d{9}">
    </xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="tratamento">
<xs:complexType>
<xs:sequence>
    <xs:element ref="descricao"/>
    <xs:element ref="horario"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="meiosdg">
<xs:complexType>
<xs:sequence>
    <xs:element ref="descricao"/>
    <xs:element ref="horario"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horario">
<xs:complexType>
<xs:sequence>

```

```

        <xs:element ref="dias"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horariogeral">
<xs:complexType>
<xs:sequence>
    <xs:element ref="dias"/>
    <xs:element ref="horas"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="horas">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:pattern value="([01]?[0-9]|2[0-3]):[0-5][0-9]-([01]?[0-9]|2[0-3]):[0-5][0-9]">
    </xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="unidadesaude">
    <xs:complexType>
<xs:sequence>

<!-- centro de saúde -->

<xs:element name="csaude" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nome"/>
            <xs:element ref="morada"/>
            <xs:element ref="coordenadas"/>
            <xs:element ref="direccoes"/>
            <xs:element ref="hproximo"/>
            <xs:element ref="contacto"/>
            <xs:element ref="tratamento" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="meiosdg" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>

```

(b) DTD

```

<!DOCTYPE unidadesaude [
<!ELEMENT unidadesaude (csaude+)>
<!ELEMENT csaude (nome,morada,coordenadas,direccoes,hproximo,contacto,tratamento*,

```

```

meiosdg*)>
  <![ELEMENT nome (#PCDATA)]>
  <![ELEMENT morada (endereco,freguesia,concelho,distrito,cpostal)]>
    <![ELEMENT endereco (#PCDATA)]>
    <![ELEMENT freguesia (#PCDATA)]>
    <![ELEMENT concelho (#PCDATA)]>
    <![ELEMENT distrito (#PCDATA)]>
    <![ELEMENT cpostal (#PCDATA)]>
  <![ELEMENT coordenadas (#PCDATA)]>
  <![ELEMENT direccoes (#PCDATA)]>
  <![ATTLIST direccoes value CDATA #IMPLIED]>
  <![ELEMENT hproximo (#PCDATA)]>
    <![ATTLIST hproximo value CDATA #IMPLIED]>
  <![ELEMENT contacto (telefone,fax,email)]>
    <![ELEMENT telefone (#PCDATA)]>
    <![ELEMENT fax (#PCDATA)]>
    <![ELEMENT email (#PCDATA)]>
  <![ELEMENT tratamento (descricao,horario)]>
    <![ELEMENT descricao (#PCDATA)]>
    <![ELEMENT horario (dias,horas)]>
      <![ELEMENT dias (#PCDATA)]>
      <![ELEMENT horas (#PCDATA)]>
  <![ELEMENT meiosdg (descricao,horario)]>
    <![ELEMENT descricao (#PCDATA)]>
    <![ELEMENT horario (dias,horas)]>
      <![ELEMENT dias (#PCDATA)]>
      <![ELEMENT horas (#PCDATA)]>
]>

```

3. Farmácias

(a) Schema

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- definition of attributes -->

  <xs:attribute name="value" type="xs:string"/>

  <!-- definition of simple elements -->

  <xs:element name="nome" type="xs:string"/>
  <xs:element name="endereco" type="xs:string"/>
  <xs:element name="freguesia" type="xs:string"/>
  <xs:element name="concelho" type="xs:string"/>
  <xs:element name="distrito" type="xs:string"/>
  <xs:element name="fax" type="xs:integer"/>
  <xs:element name="email" type="xs:string"/>
  <xs:element name="dias" type="xs:string"/>
  <xs:element name="hrefer" type="xs:string"/>

  <!-- definition of complex elements -->

  <xs:element name="direccoes">

```

```

<xs:complexType>
  <xs:attribute name="value" type="xs:anyURI"/>
</xs:complexType>
</xs:element>

<xs:element name="morada">
<xs:complexType>
<xs:sequence>
  <xs:element ref="endereco"/>
  <xs:element ref="freguesia"/>
  <xs:element ref="concelho"/>
  <xs:element ref="distrito"/>
  <xs:element ref="cpostal"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="cpostal">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[4]\d{3}-\d{3}">
  </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="coordenadas">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="(\-?\d+(\.\d+)?),\s*(\-?\d+(\.\d+)?)">
  </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

<xs:element name="contacto">
<xs:complexType>
<xs:sequence>
  <xs:element ref="telefone"/>
  <xs:element ref="fax" minOccurs="0"/>
  <xs:element ref="email" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="telefone">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{9}">
  </xs:pattern>
  </xs:restriction>
</xs:simpleType>
</xs:element>

```

```

<xs:element name="unidadesaude">
  <xs:complexType>
<xs:sequence>

<!-- farmácia -->

<xs:element name="farmacia" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nome"/>
      <xs:element ref="morada"/>
      <xs:element ref="coordenadas"/>
      <xs:element ref="direccoes"/>
      <xs:element ref="contacto" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>

```

(b) DTD

```

<!DOCTYPE unidadesaude [
<!ELEMENT unidadesaude (farmacia+)>
<!ELEMENT farmacia (nome,morada,coordenadas,direccoes,contacto*)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT morada (endereco,freguesia,concelho,distrito,cpostal)>
    <!ELEMENT endereco (#PCDATA)>
    <!ELEMENT freguesia (#PCDATA)>
    <!ELEMENT concelho (#PCDATA)>
    <!ELEMENT distrito (#PCDATA)>
    <!ELEMENT cpostal (#PCDATA)>
  <!ELEMENT coordenadas (#PCDATA)>
  <!ELEMENT direccoes (#PCDATA)>
    <!ATTLIST direccoes value CDATA #IMPLIED>
  <!ELEMENT contacto (telefone)>
    <!ELEMENT telefone (#PCDATA)>
]>

```

II Exemplos XML

Apresentam-se, de seguida, exemplos do conteúdo de cada um dos documentos XML produzidos, correspondentes a um hospital, um centro de saúde e uma farmácia, respectivamente.

1. Hospital Santa Maria Maior

```
<hospital>
<nome>Hospital Santa Maria Maior</nome>
<morada>
  <endereco>Campo da República</endereco>
  <freguesia>Barcelos</freguesia>
  <concelho>Barcelos</concelho>
  <distrito>Braga</distrito>
  <cpostal>4754-909</cpostal>
</morada>
<coordenadas>41.533688,-8.61631</coordenadas>
<direccoes value="https://maps.google.com/maps?saddr=Avenida+Central,+Braga,+Portugal&addr=Novo+Hospital+de+Braga+%4041.567684,-8.399391&hl=pt-PT&sll=41.55121,-8.413296&sspn=0.016058,0.04227&geocode=FXUFegId04N__ynx1hgmwP4kDTGChFsAyCJeBg%3BFcRFegId4dV__w&oq=Av&t=h&doflg=ptk&mra=&z=14">
</direccoes>
<hproximo value="https://maps.google.com/maps?saddr=Hospital+Santa+Maria+Maior+de+Barcelos+%4041.53318,-8.61639&addr=Unidade+Hospitalar+de+Famalic%C3%A3o+%4041.412296,-8.521133&hl=pt-PTie=UTF8&ll=41.473345,-8.567963&spn=0.129906,0.308647&sll=41.473345,-8.570709&sspn=0.129906,0.308647&geocode=Ffy-eQId0oZ8_w%3BFcjmdwIdU_p9_w&mra=ls&t=m&z=12">
</hproximo>
<contacto>
  <telefone>253809200</telefone>
  <fax>253817379</fax>
  <email>secadm@hbarcelos.min-saude.pt</email>
</contacto>
<urgencia>
  <nivel>Medico-cirúrgica</nivel>
  <tipo>Pediátrica</tipo>
  <horariogeral>
    <dias>2ª Feira-Domingo</dias>
    <horas>00:00-00:00</horas>
  </horariogeral>
</urgencia>
<urgencia>
  <nivel>Medico-cirúrgica</nivel>
  <tipo>Geral</tipo>
  <horariogeral>
    <dias>2ª Feira-Domingo</dias>
    <horas>00:00-00:00</horas>
  </horariogeral>
<especialidade>
  <descricao>Medicina Interna</descricao>
  <horario>
    <dias>2ª Feira-Domingo</dias>
    <horas>00:00-00:00</horas>
  </horario>
</especialidade>
```

```

<especialidade>
  <descricao>Cirurgia Geral</descricao>
  <horario>
    <dias>2ª Feira-Domingo</dias>
    <horas>00:00-00:00</horas>
  </horario>
</especialidade>
  </urgencia>
</hospital>

```

2. Centro de Saúde de Famalicão

```

<csaude>
<nome>Centro de Saúde de Famalicão</nome>
<morada>
  <endereco>Avenida 25 de Abril</endereco>
  <freguesia>Vila Nova de Famalicão</freguesia>
  <concelho>Braga</concelho>
  <distrito>Braga</distrito>
  <cpostal>4760-101</cpostal>
</morada>
<coordenadas>41.407684, -8.522969</coordenadas>
<direccoes value="https://maps.google.com/maps?saddr=Avenida+Central,+Braga,+Portugal&addr=Centro+de+
Sa%C3%BAde+de+Famalic%C3%A3o+%4041.407627,-8.522944&hl=pt-PT&ie=UTF8&sll=41.55006,-8.384514&am
p;sspn=0.55806,1.352692&geocode=FXUFegId04N_ynx1hgmwP4kDTGChFsAyCJeBg%3BFYvU
dwIdQPN9_w&oq=Av&t=h&doflg=ptk&mra=ls&z=11"></direccoes>
<hproximo value="https://maps.google.com/maps?saddr=Centro+de+Sa%C3%BAde+de+Famalic%C3%A3o+%4041.407629,-8.5
22944&daddr=Unidade+Hospitalar+de+Famalic%C3%A3o+%4041.412296,-8.521133&hl=pt-PT&ie=UTF8&sll=
41.417371,-8.499813&sspn=0.065009,0.154324&geocode=FY3UdwIdQPN9_w%3BFcjmdwIdU_p9_w&mra=ls&t=m
&z=16"></hproximo>
<contacto>
  <telefone>252330230</telefone>
  <fax>252330231</fax>
  <email>urbana@csfamalicao1.min-saude.pt</email>
</contacto>
<tratamento>
  <descricao>Pensos</descricao>
  <horario>
    <dias>2ª Feira-6ª Feira</dias>
    <horas>08:30-17:30</horas>
  </horario>
</tratamento>
<tratamento>
  <descricao>Injectáveis</descricao>
  <horario>
    <dias>2ª Feira-6ª Feira</dias>
    <horas>08:30-17:30</horas>
  </horario>
</tratamento>
<tratamento>
  <descricao>Algalias</descricao>
  <horario>
    <dias>2ª Feira-6ª Feira</dias>
    <horas>08:30-17:30</horas>
  </horario>

```

```

</tratamento>
<tratamento>
  <descricao>Aspiração de Secreções</descricao>
  <horario>
    <dias>2ª Feira-6ª Feira</dias>
    <horas>08:30-17:30</horas>
  </horario>
</tratamento>
<tratamento>
  <descricao>Remoção de Pontos</descricao>
  <horario>
    <dias>2ª Feira-6ª Feira</dias>
    <horas>08:30-17:30</horas>
  </horario>
</tratamento>
</csaude>

```

3. Farmácia Adaúde

```

<farmacia>
<nome>Farmácia Adaúfe</nome>
<morada>
  <endereço>Rua da Pegada</endereço>
  <freguesia>Adaúfe</freguesia>
  <concelho>Braga</concelho>
  <distrito>Braga</distrito>
  <cpostal>4710-556</cpostal>
</morada>
<coordenadas>41.451347, -8.294488</coordenadas>
<direccoes value="https://maps.google.com/maps?saddr=Avenida+Central,+Braga,+Portugal&addr=Farm%C3%A
1cia+Adaúfe+%4041.585879,-8.402036&hl=pt-PT&ie=UTF8&sll=41.742627,-8.410034&sspn=1.034899
,2.469177&geocode=FXUFegId04N__ynx1hgmwP4kDTGChFsAyCJeBg%3BFdeMegIdjMt__w&t=h&doflg=ptk&mra
=ls&z=13"></direccoes>
<contacto>
  <telefone>253679121</telefone>
</contacto>
</farmacia>

```

III JavaScript

A título de clarificação, apresentam-se as funções `go()` e `go2()`, neste caso específico, relativas às unidades hospitalares.

1. Função `go()`

```

<script type = "text/javascript">

  function go() {

    var x = document.getElementById("distrito").value;

    if (x == "Braga") {

      document.getElementById("h_braga").style.display="block";
    }
  }

```

```
    } else if (x == "Porto"){

        document.getElementById("h_porto").style.display="block";

    } else {

        document.getElementById("h_viana").style.display="block";

    }
}

</script>
```

2. Função go2()

```
<script type = "text/javascript">

    function go2() {

        var hospb = document.getElementById("h_braga").value;
        var hospp = document.getElementById("h_porto").value;
        var hospv = document.getElementById("h_viana").value;

        if (hospb == "Hospital de Braga") {

            document.getElementById("hdb").style.display="block";

        } else if (hospb == "Hospital Santa Maria Maior") {

            document.getElementById("hsmm").style.display="block";

        } else if (hospb == "Unidade Hospitalar de Fafe") {

            document.getElementById("hhf").style.display="block";

        } else if (hospb == "Unidade Hospitalar de Famalicão") {

            document.getElementById("uhf").style.display="block";

        } else if (hospb == "Unidade Hospitalar de Guimarães") {

            document.getElementById("hhg").style.display="block";

        } else if (hospp == "Centro Hospitalar de Vila Nova de Gaia") {

            document.getElementById("chvng").style.display="block";

        } else if (hospp == "Hospital Geral de Santo António") {

            document.getElementById("hgsa").style.display="block";

        } else if (hospp == "Hospital Joaquim Urbano") {
```

```
document.getElementById("hju").style.display="block";

} else if (hospp == "Hospital Nossa Senhora da Conceição de Valongo") {

    document.getElementById("hns cv").style.display="block";

} else if (hospp == "Hospital Padre Américo") {

    document.getElementById("hpa").style.display="block";

} else if (hospp == "Hospital Pedro Hispano") {

    document.getElementById("hph").style.display="block";

} else if (hospp == "Hospital São Gonçalo de Amarante") {

    document.getElementById("hsga").style.display="block";

} else if (hospp == "Hospital São João") {

    document.getElementById("hsj").style.display="block";

} else if (hospp == "Hospital de Magalhães Lemos") {

    document.getElementById("hml").style.display="block";

} else if (hospp == "Instituto Português de Oncologia do Porto") {

    document.getElementById("ipop").style.display="block";

} else if (hospp == "Unidade Hospitalar da Póvoa de Varzim") {

    document.getElementById("uhpv").style.display="block";

} else if (hospp == "Unidade Hospitalar de Santo Tirso") {

    document.getElementById("uhst").style.display="block";

} else if (hospp == "Unidade Hospitalar de Vila do Conde") {

    document.getElementById("uhvc").style.display="block";

} else if (hospv == "Hospital Conde de Bertiandos") {

    document.getElementById("hcb").style.display="block";

} else if (hospv == "Hospital Santa Luzia de Viana do Castelo") {

    document.getElementById("hslvc").style.display="block";

}

}

</script>
```

Referências

- [1] INFARMED. Administração de vacinas em farmácia. http://www.infarmed.pt/portal/page/portal/INFARMED/MAIS_NOVIDADES/DETALHE_NOVIDADE?itemid=3676059.