



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Integração de Dados

2024/25 – Trabalho Prático

João Rosa - 2022131973

Índice

1. Descrição do Trabalho.....	3
2. Analisar as Fontes de Dados (S).....	4
3.1 Capital.....	4
3.2 Continente.....	4
3.3 Bandeira do País.....	5
3.4 Presidente / Chefe de Estado.....	5
3.5 Restante Informação.....	5
3. Definir o Esquema Global (G).....	6
3.1 Estrutura Hierárquica.....	6
4. Implementação dos Wrappers (M).....	7
5. Gerar / Manipular Ficheiro XML.....	8
5.1 Adicionar País.....	8
5.2 Eliminar um País.....	8
5.3 Editar/Eliminar Atributos.....	9
6. Validar Modelo (DTD/XSD).....	9
6.1 DTD (paísesValidacao.dtd).....	9
6.2 XSD (paísesValidacao.xsd).....	10
7. Fazer Pesquisas XPath.....	11
8. Gerar Ficheiro de Output (XSLT/XQuery).....	12
9. Interface Gráfico.....	13
10. Conclusão.....	15

1. Descrição do Trabalho

No âmbito da Unidade Curricular de **Integração de Dados** no ano letivo **2024/2025**, foi proposta a realização de um trabalho prático com o objetivo de criar uma aplicação de integração de dados que apresentasse uma visão unificada de informações relativas a países.

A informação foi extraída de dois sites a seguir apresentados, tratada e integrada em ficheiros XML. As fontes de dados usadas foram:

<https://pt.wikipedia.org/wiki/>

<https://pt.db-city.com/>

Os dados foram provenientes destas fontes de dados e o modelo global G é composto pelo ficheiro XML (países.xml) onde toda a informação pesquisada está organizada em elementos e atributos da maneira que achei mais correta e simplificada de forma a ser possível implementar todas as tarefas propostas.

O enunciado já propôs uma quantidade elevada de informação, logo decidi não incluir mais informação adicional devido ao facto de não achar relevante para a realização do projeto na sua totalidade.

Todos os valores numéricos foram devidamente tratados de forma a serem corretamente manipulados por pesquisas XPATH/XQUERY. Qualquer valor numérico ficou sem espaços, pontos (.) ou vírgulas (,) apenas para os valores decimais foi uniformizado o ponto (.).

Após o estudo das diferentes fontes de dados decidi usar apenas uma fonte para cada informação relevante. No capítulo seguinte justificarei a razão de ter escolhido a respetiva fonte de dados para cada dado extraído.

Os esquemas adotados nas vistas unificadas foram validados usando XSD e o DTD apropriados.

2. Analisar as Fontes de Dados (S)

Neste capítulo justificarei todas as minhas decisões no que toca à escolha das fontes de dados para cada informação proposta.

3.1 Capital

Para extrair as capitais de todos os países decidi usar a Wikipédia. Neste caso não existe uma justificação, simplesmente por ser a Wikipedia o site utilizado nas aulas práticas senti-me mais confortável para aplicar esta fonte de dados no primeiro campo a ser extraído.

```
String link = "https://pt.wikipedia.org/wiki/";
HttpRequestFunctions.httpRequest1(link, pais, "pais.html");
Scanner ler = new Scanner(new FileInputStream("pais.html"));
String linha;
String er1 = ">Capital<";
String er2 = ">([A-Za-zÁÉÍÓÚáéíóúÂÊÔâêôÀàÛüççÃÕãõ\\s]+)</a>";
```

3.2 Continente

Para extrair os continentes dos países inseridos pelo utilizador, decidi utilizar o site db-city devido à falta dessa informação na Wikipédia em diversos países.

```
public static String obter_continente(String pais) throws IOException {
    String link = "https://pt.db-city.com/";
    HttpRequestFunctions.httpRequest1(link, pais, "pais.html");
    Scanner ler = new Scanner(new FileInputStream("pais.html"));
    String linha;
    String er = "<th>Continent</th>\\s*<td>\\s*<a[^>]*>([<]+)</a>";
```

3.3 Bandeira do País

Neste caso decidi utilizar a Wikipedia por dois motivos: a qualidade das imagens é consideravelmente superior e porque me facilitava a execução da Expressão Regular.

```
public static String obtem_bandeira(String pais) throws IOException {
    String link = "https://pt.wikipedia.org/wiki/";
    HttpRequestFunctions.httpRequest1(link, pais, "pais.html");
    Scanner ler = new Scanner(new FileInputStream("pais.html"));
    String linha;
    String er = "<img alt=\"[A-Za-zÀÁÂÃÄÅËÉÊËÏÓÔÕÖÙÀÀÛÜÇÇÃÕãõ\\s]+\" src=\"([^\"]+)\" decoding\"";
```

3.4 Presidente / Chefe de Estado

Para encontrar o chefe de estado de cada país decidi usar a Wikipédia apenas para nivelar o número de vezes que faço pesquisas em cada uma das fontes de dados. Embora a informação estivesse melhor no db-city. Para encontrar o chefe de estado nos países através da Wikipédia sou obrigado a escrever o nome de cada tipo de chefe de estado, o que pode envolver o esquecimento de algum e não encontrar para alguns países.

```
public static String obtem_presidente(String pais) throws IOException {
    String link = "https://pt.wikipedia.org/wiki/";
    HttpRequestFunctions.httpRequest1(link, pais, "pais.html");
    Scanner ler = new Scanner(new FileInputStream("pais.html"));
    String linha;
    //>Presidente<|>Rei<
    //<td scope="row" style="vertical-align:
    String er1 = ">Presidente<|>Rei<|>Monarca<|>Imperador<|>Emir<";
    String er2 = "title=\"[A-Za-zÀÁÂÃÄÅËÉÊËÏÓÔÕÖÙÀÀÛÜÇÇÃÕãõ\\s]+\">([A-Za-zÀÁÂÃÄÅËÉÊËÏÓÔÕÖÙÀÀÛÜÇÇÃÕãõ\\s]+)<";
```

3.5 Restante Informação

Toda a restante informação foi retirada do site db-city devido ao facto de todos os países neste site, sem exceção, terem todo o tipo de dados organizado de forma muito mais intuitiva que a Wikipédia. Na maioria dos casos, tal como o N° de casos COVID registados, não é apresentado em 99% dos países, sendo mesmo obrigado a utilizar esta fonte de dados.

A partir do código é perceptível a organização efetuada sendo separado através de comentários a informação retirada de cada fonte de dados.

3. Definir o Esquema Global (G)

Neste capítulo mostrarei de forma simplificada de que forma organizei o meu documento XML mostrando a sua estrutura hierárquica. No capítulo 6 deste relatório serão demonstrados os códigos XSD e DTD apropriados a este XML de forma a validar corretamente o mesmo.

3.1 Estrutura Hierárquica



4. Implementação dos Wrappers (M)

Antes de mais, foi usada a função *HttpRequest* fornecida nas aulas práticas para aceder às 2 fontes de dados e gravá-las em disco. Foram analisadas ambas as fontes de forma a saber como implementar os Wrappers de forma correta e estruturada.

Como referido anteriormente, o ficheiro *países.xml* contém todas as informações relevantes encontradas com o recurso a **expressões regulares**, através da implementação de **Wrappers**.

Foram implementados 13 Wrappers (*Wrappers.java*) na sua totalidade como foi proposto. Destes 13, 8 retornam **String** e 4 **ArrayList<String>**.

Obtém cada uma das informações pretendidas a partir do seguinte código em que o país com o nome seja igual à *String* *pais*.

```
String capital = Wrappers.obtem_capital(pais); //wikipedia
String continente = Wrappers.obtem_continente(pais); //db-city
String bandeira = Wrappers.obtem_bandeira(pais); //wikipedia
ArrayList linguas = Wrappers.obtem_linguas(pais); //db-city
String area = Wrappers.obtem_area(pais); //db-city
String habitantes = Wrappers.obtem_habitantes(pais); //db-city
String densidade = Wrappers.obtem_densidade(pais); //db-city
String presidente = Wrappers.obtem_presidente(pais); //Wikipedia
ArrayList religioes = Wrappers.obtem_religiao(pais); //db-city
ArrayList cidades = Wrappers.obtem_cidades(pais); //db-city
ArrayList fronteiras = Wrappers.obtem_frenteiras(pais); //db-city
String covid = Wrappers.obtem_covid(pais); //db-city
```

Durante a realização destes Wrappers, notei falta de alguma informação em ambas as fontes de dados. De forma a resolver isso utilizei a outra fonte de dados disponível, o que no geral resolveu a falta de informação. Ou seja, ambas **as fontes de dados juntas forneceram toda a informação necessária** e relevante.

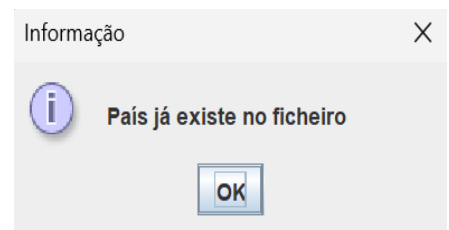
5. Gerar / Manipular Ficheiro XML

Após a Implementação dos Wrappers, foram criadas algumas opções de manipulação do ficheiro XML.

5.1 Adicionar País

Adiciona um País inserido pelo utilizador e as respectivas informações ao ficheiro XML. Sendo que este não permite a repetição de países e caso o ficheiro não esteja criado, é criado aquando da inserção do primeiro país.

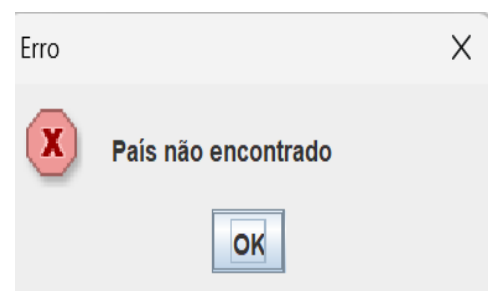
The form titled 'Adicionar País' features a text input field labeled 'País:' and a button labeled 'Adicionar País' at the bottom.



5.2 Eliminar um País

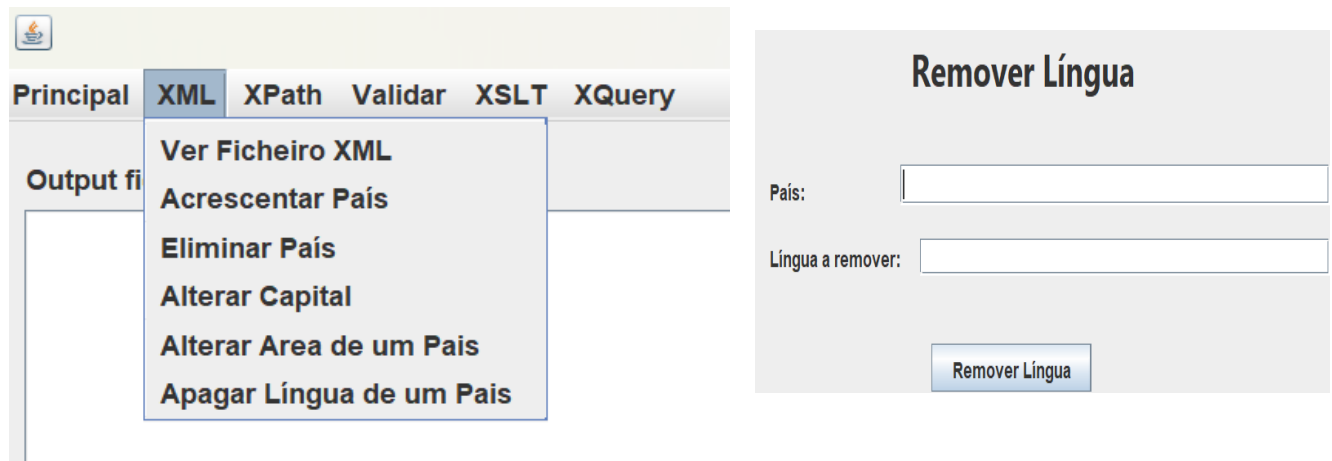
Permite eliminar um país, e as respectivas informações, escolhido pelo utilizador através do nome. Exibe uma mensagem de erro quando esse país não existe no ficheiro XML.

The form titled 'Remover País' features a text input field labeled 'País:' and a button labeled 'Remover País' at the bottom.



5.3 Editar/Eliminar Atributos

Permite alterar a **capital** ou a área de um **país** definido pelo utilizador para um valor introduzido pelo mesmo. É possível também remover a **língua** desejada do array do país introduzido pelo utilizador.



6. Validar Modelo (DTD/XSD)

Neste capítulo mostrarei como faço a validação do meu ficheiro `paises.xml` utilizando o ficheiro DTD e XSD apropriados. Esta tarefa foi executada utilizando API JDOM2 dado nas aulas práticas.

6.1 DTD (`paisesValidacao.dtd`)

```
<!ELEMENT paises (paisDados+)>
<!ELEMENT paisDados (pais, capital, bandeira,
linguas, area, populacao, densidade, chefe_estado,
religioes, cidades_importantes, fronteiras, casos_covid)>
<!ATTLIST paisDados nome CDATA #REQUIRED>

<!ELEMENT pais (#PCDATA)>
<!ELEMENT capital (#PCDATA)>
<!ELEMENT bandeira (#PCDATA)>
<!ELEMENT linguas (língua+)>
<!ELEMENT lingua (#PCDATA)>
<!ELEMENT area (#PCDATA)>
<!ELEMENT populacao (#PCDATA)>
<!ELEMENT densidade (#PCDATA)>
<!ELEMENT chefe_estado (#PCDATA)>
<!ELEMENT religioes (religiao*)>
<!ELEMENT religiao (#PCDATA)>
<!ELEMENT cidades_importantes (cidade*)>
<!ELEMENT cidade (#PCDATA)>
<!ELEMENT fronteiras (pais_frenteira*)>
<!ELEMENT pais_frenteira (#PCDATA)>
<!ELEMENT casos_covid (#PCDATA)>
```

```
switch (result) {
    case 0:
        JOptionPane.showMessageDialog(this,
            "Erro em ficheiros",
            "Erro",
            JOptionPane.ERROR_MESSAGE);
        break;

    case 1:
        JOptionPane.showMessageDialog(this,
            "Ficheiro paises.xml valido por DTD",
            "De acordo com DTD",
            JOptionPane.INFORMATION_MESSAGE);
        break;

    case -1:
        JOptionPane.showMessageDialog(this,
            "Ficheiro paises.xml invalido por DTD",
            "Nao está de acordo com DTD",
            JOptionPane.INFORMATION_MESSAGE);
        break;

    default:
        JOptionPane.showMessageDialog(this,
            "Resultado imprevisto",
            "Erro",
            JOptionPane.ERROR_MESSAGE);
        break;
```

6.2 XSD (paísesValidacao.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

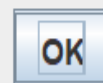
  <xs:element name="países">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="paisDados" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="pais" type="xs:string"/>
              <xs:element name="capital" type="xs:string"/>
              <xs:element name="bandeira" type="xs:string"/>
              <xs:element name="linguas">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="lingua" type="xs:string" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="area" type="xs:decimal"/>
              <xs:element name="populacao" type="xs:integer"/>
              <xs:element name="densidade" type="xs:decimal"/>
              <xs:element name="chefe_estado" type="xs:string"/>
              <xs:element name="religioes">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="religiao" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="cidades_importantes">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="cidade" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="fronteiras">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="pais_frenteira" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="casos_covid" type="xs:integer"/>
            </xs:sequence>
            <xs:attribute name="nome" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
int result = ValidarXML.validarDocumentoXSD("paísesXSD.xml", "paísesValidacao.xsd");
switch (result) {
  case 0:
    JOptionPane.showMessageDialog(this,
      "Erro em ficheiros",
      "Erro",
      JOptionPane.ERROR_MESSAGE);
    break;
  case 1:
    JOptionPane.showMessageDialog(this,
      "Ficheiro países.xml valido por XSD",
      "De Acordo com XSD",
      JOptionPane.INFORMATION_MESSAGE);
    break;
  case -1:
    JOptionPane.showMessageDialog(this,
      "Ficheiro países.xml invalido por XSD",
      "Nao está de Acordo com XSD",
      JOptionPane.INFORMATION_MESSAGE);
    break;
  default:
    JOptionPane.showMessageDialog(this,
      "Resultado imprevisto",
      "Erro",
      JOptionPane.ERROR_MESSAGE);
    break;
}
```

De Acordo com XSD



Ficheiro países.xml valido por XSD



7. Fazer Pesquisas XPath

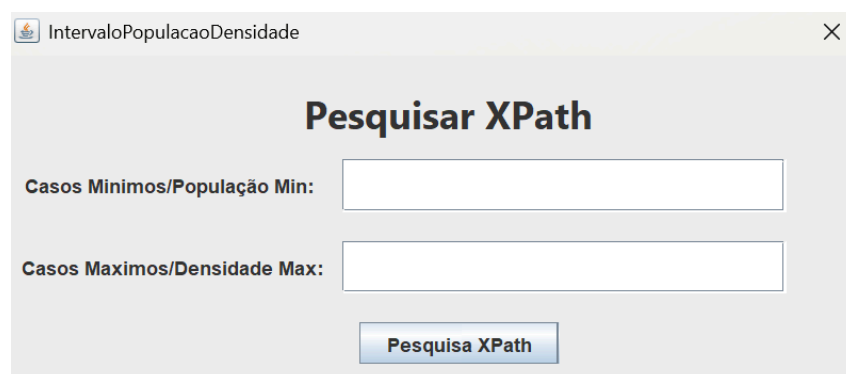
É permitido ao utilizador a realização de diferentes pesquisas nos ficheiros XML, das quais:

- Mostrar a informação relevante de um país.
- Pesquisar cidades importantes de um dado país.
- Pesquisar apenas os países com número de habitantes superior ao introduzido.
- Pesquisar países que tenham uma religião definida pelo utilizador.
- Mostrar todas as capitais existentes no ficheiro.
- Mostrar o nome dos países com nº de casos covid num intervalo definido pelo utilizador.
- Pesquisar os países que tenham no mínimo 3 cidades importantes.
- Pesquisar o nome dos países que tenham mais de uma língua oficial.
- Mostrar os países que partilham fronteira com outro que fale a mesma língua.
- Mostrar os países com uma população superior a X mas com uma densidade populacional inferior a Y (sendo X e Y introduzidos pelo utilizador).

Nestes casos, de forma a não sobrecarregar o ficheiro, utilizei apenas uma JDialog como é possível verificar na imagem abaixo.

```
if (jDialog6.getTitle().equals("Informações relevantes de um País")) {  
    xp = "//paisDados[pais='" + jTextField9.getText() + "']";  
}  
if (jDialog6.getTitle().equals("Cidades Importantes de um País")) {  
    xp = "//paisDados[@nome='" + jTextField9.getText() + "']/cidades_importantes/cidade";  
}  
if (jDialog6.getTitle().equals("numero de habitantes superior")) {  
    xp = "//paisDados[populacao > " + jTextField9.getText() + "]/@nome";  
}  
if (jDialog6.getTitle().equals("Países com Religiao")) {  
    xp = "//paisDados[religioes/religiao='" + jTextField9.getText() + "']/@nome";  
}
```

Nos casos em que não é necessário uma JDialog para a inserção da pesquisa pelo utilizador (todas as capitais no ficheiro, países com mais de 3 cidades importantes e países com mais de 1 língua oficial) o código está diretamente no JMenu e não no botão presente na JDialog. Utilizei a janela abaixo para os restantes 2 casos.



8. Gerar Ficheiro de Output (XSLT/XQuery)

O utilizador consegue gerar ficheiros HTML, TXT ou XML com os resultados pretendidos com recurso a ficheiros **XSLT** e **XQuery**. Estas são as transformações permitidas:

- **XSLT:**

- Gerar **HTML** com uma tabela com os nomes e bandeiras dos países sem repetições (para não confundir a bandeira com o fundo web, foi criada uma pequena borda à volta das bandeiras devido às bandeiras que têm branco no seu limite); **–OBRIGATÓRIO–**
 - transfPaísesBandeiras.xsl ->> HTMLPaísesBandeiras.html
- Gerar **HTML** com lista dos Países Fronteira de um determinado País definido pelo utilizador (neste caso é gerado um XML auxiliar para isolar apenas o país a pesquisar e depois é aplicado o XSLT nesse ficheiro temporário); **–OBRIGATÓRIO–**
 - fronteiras.xsl ->> fronteiras.html
- Gerar **HTML** com todos os países e a respetiva Densidade Populacional;
 - paisesDensidade.xsl ->> HTMLPaísesDensidade.html

- **XQuery:**

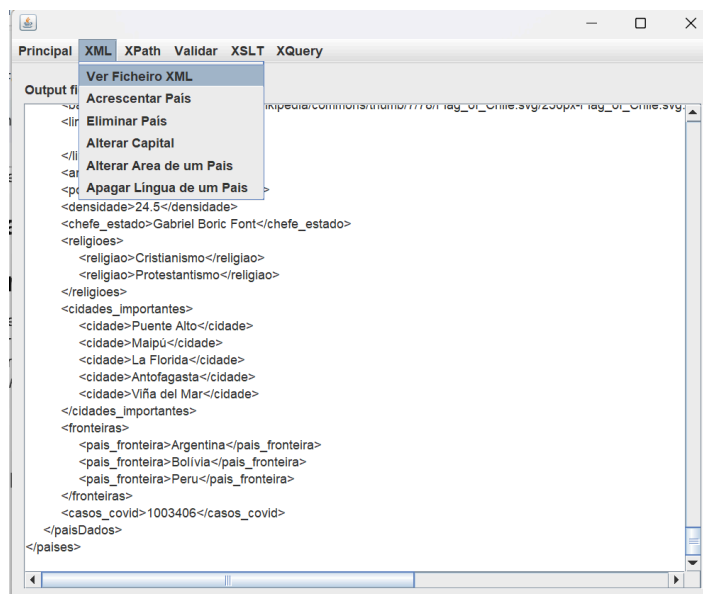
- Gerar um **TXT** com todas as cidades importantes de um país inserido pelo utilizador (logo de seguida é possível ver o ficheiro TXT na interface do programa); **–OBRIGATÓRIO–**
 - cidadesImportantes.xq ->> cidadesOutput.txt
- Gerar um **XML** com o top 5 países mais populosos a partir do ficheiro. (logo de seguida é possível ver o ficheiro XML na interface do programa) **–OBRIGATÓRIO–**
 - top5Populacao.xq ->> top5.xml
- Gerar **XML** com todos os países que falam uma determinada língua inserida pelo utilizador (logo de seguida é possível ver o ficheiro XML na interface do programa);
 - PaísesPorLingua.xq ->> paisesLingua.xml
- Gerar **XML** com todos os países com casos de Covid superior à média dos países existentes no ficheiro (logo de seguida é possível ver o ficheiro XML na interface do programa);
 - paisesAcimaMedia.xq ->> paisesAcimaMedia.xml

9. Interface Gráfico

Neste capítulo mostrarei a interface do meu projeto e como a mesma consegue proporcionar ao utilizador a possibilidade de executar tudo o que seja do seu interesse de forma amigável e intuitiva.

Nota: Foi usado essencialmente o exemplo gráfico das aulas práticas!

Como podemos ver na imagem abaixo, a aplicação possui 6 menus (**Principal, XML, XPath, Validar, XSLT e XQuery**) e 1 área de texto para mostrar os ficheiros (**XML ou TXT**), sendo que cada um dos menus possui itens correspondentes ao mesmo.



Nesta imagem é possível verificar também algumas funcionalidades desta aplicação. Ao serem seleccionadas aparecerá uma nova janela onde será possível reproduzir a funcionalidade escolhida como é possível verificar na imagem abaixo.

Alterar Area

País:

Alterar area para:

Como referido anteriormente, com o objetivo de tornar a aplicação mais intuitiva, em todas as janelas JDialog, após carregar no botão é gerada uma JoptionPane que informa o utilizador se a funcionalidade pretendida foi executada com normalidade. Além disso, como vai ser possível verificar nas imagens abaixo, em todas as funcionalidades em que existem alterações ou criação de ficheiros, esse ficheiro será apresentado ao utilizador. Se for XML ou TXT será mostrado no output automaticamente (na área de output da aplicação), se for HTML abrirá a página web correspondente.






Cidades Importantes de um País

País:

Lista de Países e Bandeiras

Output ficheiro XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<top5Populacao>
  <pais>
    <nome>China</nome>
    <populacao>1395380000</populacao>
  </pais>
  <pais>
    <nome>Indonésia</nome>
    <populacao>264162000</populacao>
  </pais>
  <pais>
    <nome>Brasil</nome>
    <populacao>208325000</populacao>
  </pais>
  <pais>
    <nome>Rússia</nome>
    <populacao>143965000</populacao>
  </pais>
  <pais>
    <nome>Alemanha</nome>
    <populacao>82886000</populacao>
  </pais>
</top5Populacao>
```

País	Bandeira
Polónia	
Dinamarca	
Suiça	
Indonésia	
Portugal	

Output ficheiro XML:

Berlim (Capital)
 Hamburgo
 Munique
 Colônia
 Frankfurt am Main

10. Conclusão

Em suma, este trabalho prático não só consolidou os meus conhecimentos teóricos lecionados na disciplina de Integração de Dados, como também me proporcionou experiência prática valiosa em diversas áreas, incluindo:

- Desenvolvimento de wrappers utilizando expressões regulares para extração de dados de fontes heterogêneas, como HTML e XML;
- Criação, manipulação e validação de documentos XML, utilizando DTD e XSD para garantir a conformidade com os padrões estabelecidos;
- Desenvolvimento de uma interface gráfica intuitiva para interação com os dados, facilitando a pesquisa, edição e visualização das informações;
- Utilização de tecnologias como XSLT e XQuery para transformar os dados em diferentes formatos e realizar consultas.

Este projeto permitiu-me aplicar os conhecimentos adquiridos em aula a um problema real, consolidando a minha aprendizagem e preparando-me para os desafios do mundo profissional na área de integração de dados.

Fazendo uma análise total do trabalho realizado, posso concluir que foram realizadas todas as tarefas propostas no enunciado do trabalho prático com precisão, até uma ou outra pesquisa a mais que achei relevante.

Tendo também em consideração que apliquei maioritariamente conteúdos e funções lecionados nas aulas posso concluir que, adquirir na sua totalidade todos os conhecimentos lecionados nesta disciplina. Considerando os pontos mencionados, acredito que desempenhei um bom trabalho com um excelente aproveitamento.