



Instituto Superior de
Engenharia de Coimbra

Relatório Final
Integração de Dados com XML
2017/2018

Integração de
Dados

Realizado por:

- João Pedro - 21250086
- Miguel Ferreira - 21230708

Índice

Introdução	2
Classes	3
Esquemas Globais dos Ficheiros XML	5
Funcionalidades do Programa	6
Interface Gráfica	10
Conclusão	12

Introdução

O trabalho prático da disciplina de Integração de Dados consiste na criação de um programa em *Java* composto por vários *Wrappers* que obtenham dados de fontes heterogêneas, distribuídas e autónomas de tal forma que possibilitem ao utilizador a visualização dos dados de forma integrada.

O utilizador terá ainda a possibilidade de fazer pesquisas, realizar alterações em dados que respeitem os esquemas adotados e gerar ficheiros com a informação selecionada.

Para a realização deste trabalho foi usada a linguagem *Java*, expressões regulares para a obtenção de dados e a biblioteca *API JDOM2* para manipulação de ficheiros XML estudado nas aulas práticas.

Com este trabalho pretende-se praticar e assimilar de uma melhor forma as seguintes competências:

- Analisar uma situação típica de integração de dados e apresentar propostas válidas para um modelo de integração funcional, eficaz e correto;
- Capacidade de criação e manipulação de XML;
- Utilização de expressões regulares;
- Capacidade de realização de pesquisa de informação em ficheiros XML usando XPath ou XQuery;
- Capacidade de efetuar transformações de ficheiros XML;
- Capacidade de efetuar validação de ficheiros XML usando DTD e/ou XSD.

Classes

Para este trabalho identificámos 2 classes que armazenam as informações que servem de base para este trabalho, sendo elas:

1. Classe Autores;
2. Classe Obras.

1. Classe Autores

Esta classe contém as propriedades/informações relativas a cada autor que o utilizador quiser “investigar”.

As informações que a classe armazena são:

- Nome;
- Link da sua foto que está exposta no site *Wikipédia*;
- Data de Nascimento;
- Data de Falecimento;
- Prémios;
- Géneros;
- Prémios;
- Identificador único;

O protótipo do construtor deste objecto é o seguinte:

```
Autor aux = new Autor (nome, data_nasc, data_morte, premios, generos,  
local_nasc, link_foto);
```

2. Classe Obras

Esta classe contém as propriedades/informações relativas a cada obra de um autor.

As seguintes informações que a classe armazena são:

- Código do Autor;
- Nome do Autor;
- Título da Obra/Livro;
- ISBN do Livro (código identificativo do livro);
- Preço do Livro;
- Ano de Edição do Livro;
- Editora do Livro;
- Foto da Capa.

O protótipo do construtor deste objecto é o seguinte:

```
Obra aux = new Obra(codigo_autor, autor, titulo, isbn, preco, ano, editor,  
foto_capa);
```

Esquemas Globais dos Ficheiros XML

Ficheiro XML dos Autores:

```
<autores>
  <Autor id="xxx">
    <Nome></>
    <Data_Nascimento></>
    <Data_Falecimento></>
    <Nacionalidade></>
    <Generos></>
    <Premios></>
    <Local_Nascimento></>
    <Link_Foto></>
  <Autor/>
</autores>
```

Ficheiro XML das Obras:

```
<Obras>
  <Livro>
    <Codigo_Autor></>
    <Autor></>
    <Titulo></>
    <ISBN></>
    <Ano></>
    <Preco></>
    <Editor></>
    <Foto_Capa></>
  <Livro/>
</Obras>
```

Funcionalidades do Programa

1. *Wrappers*

- **procuraNome**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e devolve o nome capturado pela expressão regular.

- **procuraNasc**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e devolve a data de nascimento capturada pela expressão regular.

- **procuraMorte**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e devolve a data de morte caso o escritor já tenha falecido.

- **procuraNacionalidade**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e procura a linha da nacionalidade devolvida pela primeira expressão regular. De seguida vai procurar a nacionalidade do autor com a segunda expressão regular.

- **procuraGenero**(String nome);

Este *wrapper* acede à página da *Wikipédia* de cada autor que o utilizador introduziu e devolve o(s) género(s) capturado(s) pela expressão regular.

- **procuraPremios**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e procura a linha do prémio devolvida pela primeira expressão regular.

- **procuralink_foto**(String nome);

Este *wrapper* acede à página da *Wikipédia* do autor que o utilizador introduziu e devolve o link da fotografia do autor capturado pela expressão regular.

- **adicionaAutor**(Autor aux)

Esta função adiciona os dados do autor que o utilizador introduziu e que foram recolhidos pelo programar, de acordo com os *wrapper's*, ao ficheiro "autores.xml".

- **procuraLinkWook**(String nome)

Este *wrapper* acede à página da Wook na pesquisa do autor indicado pelo utilizador e devolve um *ArrayList*(conjunto de valores) de *Strings* que são os *links* para a páginas dos livros do autor capturado pela expressão regular.

- **procuralsbn**(String res)

Este *wrapper* acede à página de um livro do autor no site da Wook e devolve o ISBN do livro capturado pela expressão regular.

- **procuraCod_Autor**(String nome)

Este *wrapper* acede à página da Wook na pesquisa de um autor e devolve o código da página do autor capturado pela expressão regular.

- **procuraTitulo**(String res)

Este *wrapper* acede à página de cada livro de cada autor no site da Wook e devolve o título de cada livro do autor capturado pela expressão regular.

- **procuraEditor**(String res)

Este *wrapper* acede à página de cada livro de cada autor no site da Wook e devolve o editor de cada livro de cada autor capturado pela expressão regular.

- **procuraPreco**(String res)

Este *wrapper* acede à página do livro de cada autor no site da Wook e devolve o preço do livro de cada autor capturado pela expressão regular.

- **adicionaObra**(Obra aux)

Esta função adiciona os dados das obras de cada autor (que foi introduzido pelo utilizador) e guarda-os em disco no ficheiro “obras.xml”.

- **leituraWiki**(String nome)

Esta função recebe o nome de um autor e vai buscar os dados da *Wikipédia* utilizando os *wrappers* e guarda essa informação no ficheiro “autores.xml”.

- **leituraWook**(String res)

Esta função recebe um nome de um autor, introduzido previamente pelo utilizador, e vai buscar os dados de cada livro de cada autor da Wook utilizando os *wrappers* e guarda essa informação no ficheiro “obras.xml”.

- **validarDocumentoDTD**(String xmlFile)

Esta função valida o ficheiro “autores.xml” e “obras.xml” com DTD.

- **validarDocumentoXSD**(String xmlFile)

Esta função valida o ficheiro “obras.xml” e “autores.xml” com XSD.

- **removeEscritor**(String nome)

Esta função vai ao ficheiro “autores.xml” e elimina o autor que contém o nome igual à String *nome*. Se encontrar o autor, vai também ao ficheiro “obras.xml” eliminar a informação que esteja relacionada cujo o autor seja igual à String *nome*.

- **mudaNomeAutor**(String procura, String novoNome)

Esta função vai ao ficheiro “autores.xml” e muda o nome do autor que contém o nome igual à String *procura*. Se encontrar o autor, vai também ao ficheiro “obras.xml” mudar o nome do autor.

- **pesquisaemFicheiro**(String nome_ficheiro, String pesquisa)

o Esta função vai efetuar a pesquisa no ficheiro passado como argumento na String *nome_ficheiro*.

- **pesquisaemAutores**(String pesquisa)

Esta função devolve a chamada da função *pesquisaemFicheiro* com o nome do ficheiro “autores.xml” e a String *pesquisa* passados como argumento.

- **pesquisaemObras**(String pesquisa)

Esta função devolve a chamada da função *pesquisaemFicheiro* com o nome do ficheiro “obras.xml” e a String *pesquisa* passados como argumento.

- **obterAutorporTitulooulsbn** (String titulooulsbn)

Esta função devolve a chamada da função *pesquisaemObras* com a expressão XPath para procurar a nacionalidade.

- **mostraAutores**()

Esta função devolve a chamada da função *pesquisaemAutores* com a expressão XPath para procurar todo o conteúdo do ficheiro “autores.xml”.

- **mostraAutor**(String procura)

Esta função devolve os resultados dos dados do autor que foi introduzido pelo utilizador. Esta função é-nos útil para que se o autor introduzido pelo utilizador já estiver no ficheiro *autores.xml* ele não faz nenhum pedido à *Wikipédia* e mostra logo os dados do autor.

- **mostraObras**(String procura)

Retorna as informações das obras do autor introduzido, que estão guardadas no ficheiro *obras.xml*.

- **mostraObr()**

Esta função devolve a chamada da função *pesquisaemObras* com a expressão XPath para procurar todo o conteúdo do ficheiro “obras.xml”.

- **transformaHTML**(String fxml, String fxsl)

Esta função utiliza um ficheiro xsl para transformar o ficheiro “escritores.xml” num ficheiro “fotos.html” com as fotos de cada escritor.

- **verificaObras**(String procura)

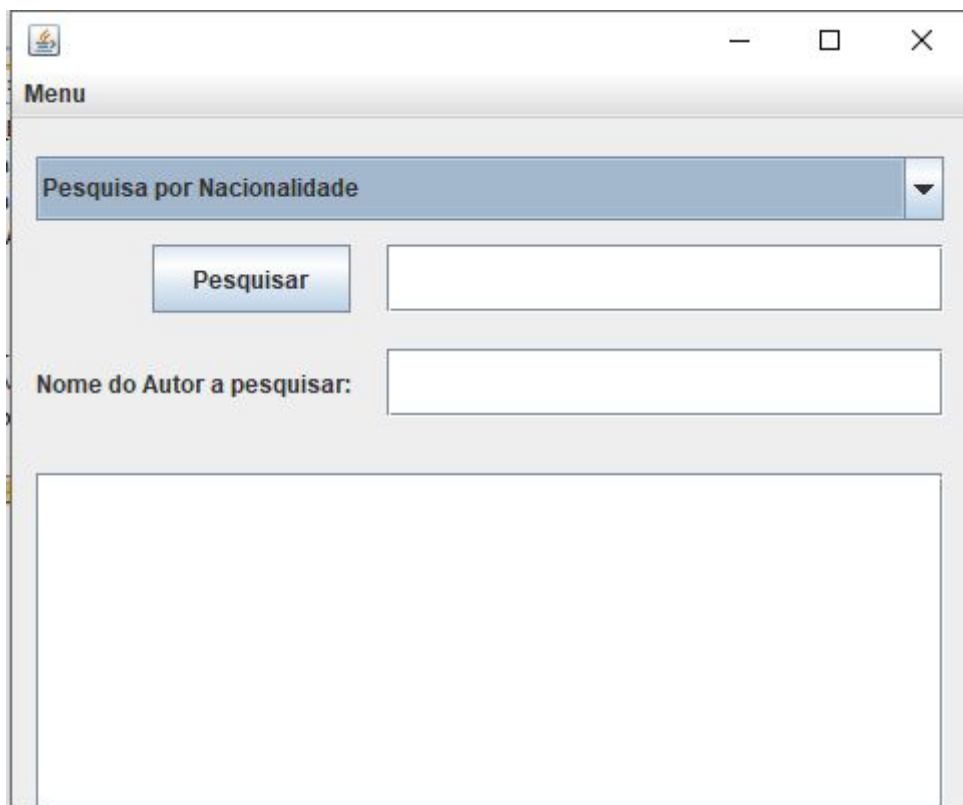
Função para verificar se já existem obras do autor que o utilizador introduziu e que é mandado por parâmetro na função (*procura*), se já existirem obras daquele autor o programa retorna os resultados *via* XPath com as obras de determinado autor.

- **verificaAutor**(String procura)

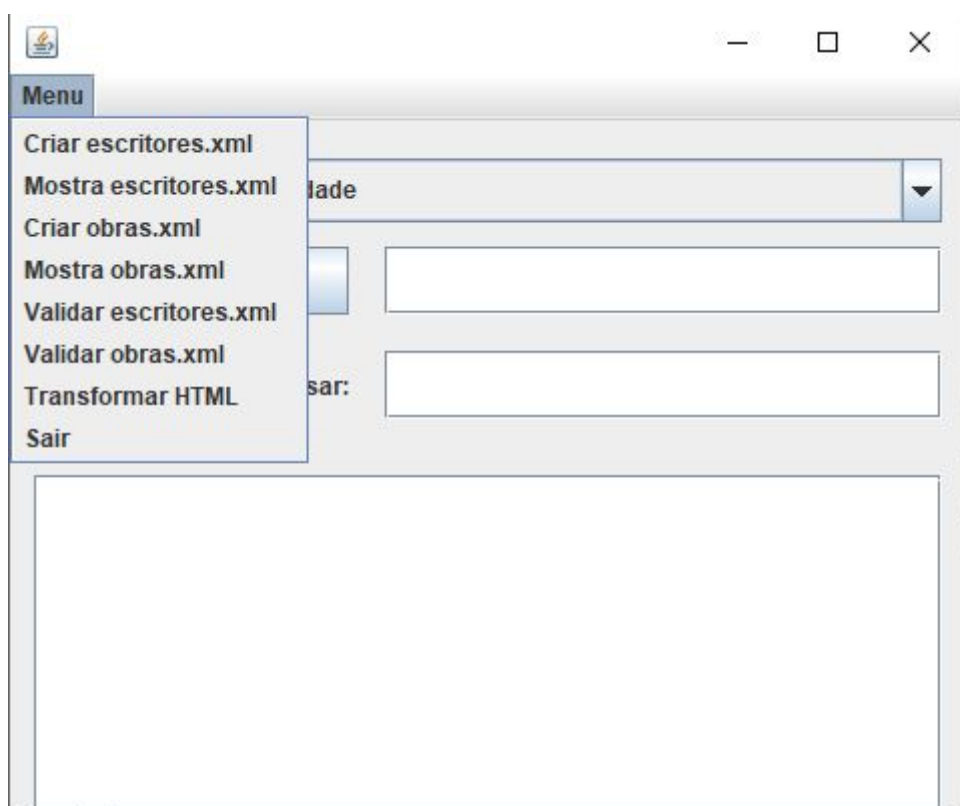
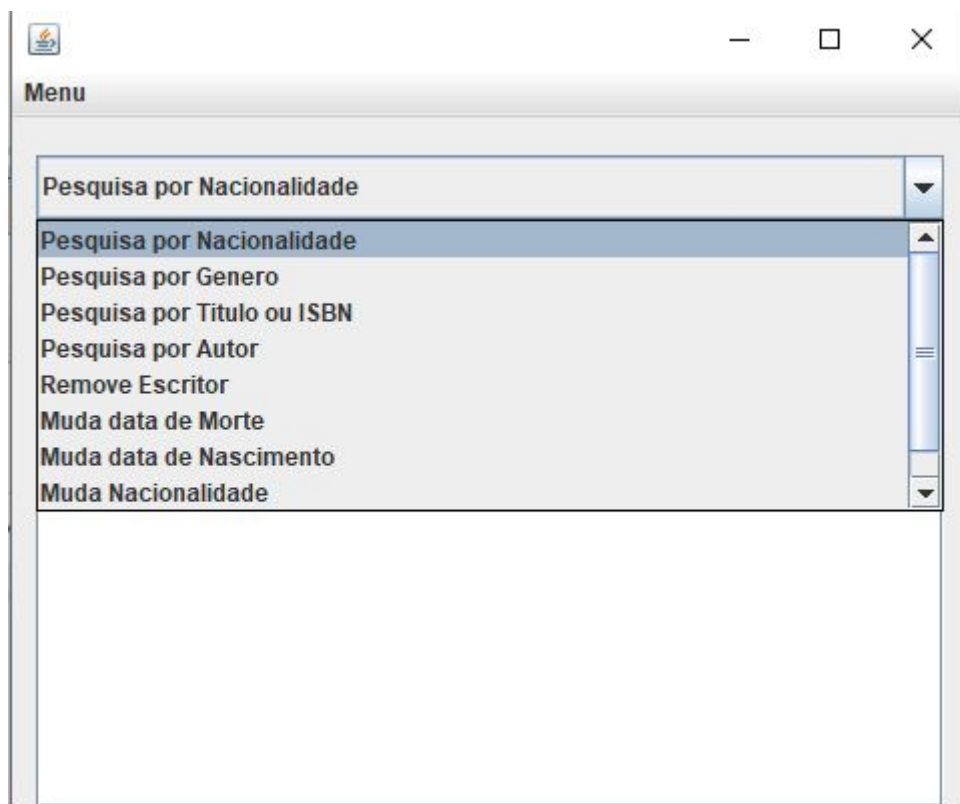
Função que verifica se já existe o autor que o utilizado quer procurar no ficheiro *autores.xml*, se já existir o programa retorna as informações do autor (que é passado por argumento para a função) que o utilizador quer investigar.

Interface Gráfica

Na interface podemos criar, validar e mostrar o conteúdo dos dois ficheiros XML. São feitas também as transformações com recurso ao XSLT. Temos também uma *comboBox* com opções para remoção e edição de conteúdos bem como opções para pesquisa de informação.



The screenshot shows a window titled "Menu" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a search interface. At the top, there is a blue header bar with the text "Pesquisa por Nacionalidade" and a dropdown arrow on the right. Below this, there is a button labeled "Pesquisar" and a text input field. Further down, there is a label "Nome do Autor a pesquisar:" followed by another text input field. At the bottom of the window, there is a large, empty rectangular area, likely intended for displaying search results.



Conclusão

Terminado o trabalho prático podemos concluir que foi um trabalho que ainda levou algum tempo até estar concluído mas em que podemos aplicar o que foi lecionado ao longo do semestre na disciplina de Integração de Dados e assimilando melhor os conceitos da linguagem de Java.

A criação das expressões regulares para obter a informação dos sites foi trabalhosa, provavelmente a parte mais demorada do trabalho, pois a informação nos sites nem sempre estava estruturada da mesma forma, o que aumentou a dificuldade e os problemas para obtermos as informações que eram necessárias.

Também na criação dos ficheiros *.xml* tivemos alguns problemas pois nem sempre conseguimos escrever a informação corretamente ou a informação não era escrita no ficheiro *.xml* sem conseguirmos perceber a aparente razão.