



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Integração de Dados

Relatório

Trabalho Prático

2022/2023

Alexandre Ferreira – 2021138219

Carlos Pinto – 2021155627



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

1. Introdução

Com base neste trabalho, propomos a criação de um programa em Java composto por vários Wrappers, que serão responsáveis por obter dados de várias fontes. O objetivo principal é permitir ao usuário visualizar esses dados de forma integrada.

Além da visualização integrada dos dados, o programa oferece recursos de pesquisa, permitindo que o utilizador encontre informações específicas. Além disso, será possível adicionar novos dados que estejam em conformidade com os esquemas adotados e gerar ficheiros.

Para a implementação do projeto, usamos da linguagem de programação Java, expressões regulares e os APIs JDOM2 e SAXON, que foram estudados durante as aulas práticas. Essas ferramentas nos auxiliarão na extração, tratamento e integração dos dados provenientes de duas fontes principais.

As fontes de dados utilizadas serão o site "<https://pt.wikipedia.org/wiki/>" e o site "<https://www.bertrand.pt/>". A partir dessas fontes, iremos extrair informações relevantes sobre escritores famosos. As informações serão processadas e integradas em arquivo(s) XML, proporcionando uma visão unificada das informações coletadas.



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

2. Estrutura do Trabalho

O trabalho encontra-se dividido da seguinte forma:

- Escritor.java;
- Frame2.java;
- Global.java;
- HttpRequestFunctions.java
- ModeloG.java;
- Obras.java;
- TPID.java;
- Wrappers.java;
- XML.java;
- linhas.java;
- XPathFunctions.java;

Além disso, adicionamos 4 ficheiros às Libraries:

- jdom-2.0.6.jar;
- saxon9-s9api.jar;
- saxon9.jar;
- JDK 19 (Default).



3. Implementação dos Wrappers

3.1 Wrappers

O código fornecido contém uma classe chamada "Wrappers" que possui vários métodos responsáveis por obter dados de fontes heterogêneas e integrá-los em um programa em Java. Vou explicar cada um desses métodos em detalhes:

obtem_idautor(): Este método retorna o ID do autor, que é armazenado na variável global id. Ele é usado para obter um identificador único para cada autor.

obtem_datamorte(): Este método extrai a data de morte do arquivo "escritores.txt". Ele faz uso de expressões regulares para procurar a informação relevante no arquivo e retorna a data de morte do autor.

obtem_datanascimento(): Similar ao método anterior, este método extrai a data de nascimento do arquivo "escritores.txt" usando expressões regulares e retorna a data de nascimento do autor.

obtemurlObras(String nome): Este método recebe um nome de autor como entrada e retorna um array de Strings contendo URLs de obras relacionadas ao autor. Ele realiza uma pesquisa no site "https://www.bertrand.pt/" usando o nome do autor e extrai os URLs das obras encontradas no arquivo "obras.txt".

criaObras(String urlObras, String nome, int i): Este método recebe uma URL de uma obra, o nome do autor e um índice como entrada e retorna um objeto do tipo "Obras". Ele faz uma solicitação HTTP para a URL fornecida, extrai informações relevantes sobre a obra do arquivo "obra.txt" e cria um objeto "Obras" com essas informações.

criaEscritor(String nome): Este método recebe o nome de um autor como entrada e retorna um objeto do tipo "Escritor". Ele realiza uma solicitação HTTP para a página da Wikipedia do autor, extrai informações relevantes do arquivo "escritores.txt" e cria um objeto "Escritor" com essas informações.

obtem_nacionalidade(): Este método extrai a nacionalidade do autor do arquivo "escritores.txt" usando expressões regulares e retorna a nacionalidade do autor.

obtem_fotografia(String nome): Este método extrai a URL da fotografia do autor do arquivo "escritores.txt" usando expressões regulares e retorna a URL da fotografia.

obtem_genero(): Este método extrai o gênero literário do autor do arquivo "escritores.txt" usando expressões regulares e retorna o gênero literário.

obtem_ocupacoes(): Este método extrai as ocupações do autor do arquivo "escritores.txt" usando expressões regulares e retorna as ocupações.



Instituto Superior de Engenharia

Politécnico de Coimbra

obtem_premios(): Este método extrai os prémios recebidos pelo autor do arquivo "escritores.txt" usando expressões regulares e retorna os prémios.

obtem_isbn(): Este método extrai o número ISBN da obra do arquivo "obra.txt" usando expressões regulares e retorna o número ISBN.

obtem_titulo(): Este método extrai o título da obra do arquivo "obra.txt" usando expressões regulares e retorna o título.

obtem_capa(): Este método extrai a URL da capa da obra do arquivo "obra.txt" usando expressões regulares e retorna a URL da capa.

obtem_editora(): Este método extrai o nome da editora da obra do arquivo "obra.txt" usando expressões regulares e retorna o nome da editora.

4. Ficheiros XML

Função `lerDocumentoXML(String caminhoFicheiro)`: Esta função lê um documento XML a partir de um ficheiro no caminho especificado. Utiliza a biblioteca JDOM para construir um objeto `Document` a partir do ficheiro XML. Se ocorrer uma exceção durante a leitura do ficheiro, é criado um novo ficheiro vazio e a função retorna `null`.

Função `escreverDocumentoParaFicheiro(Document doc, String caminhoFicheiro)`: Esta função escreve um documento XML para um ficheiro no caminho especificado. Utiliza a biblioteca JDOM para formatar o documento de saída com indentação e codificação UTF-8. O documento é escrito no ficheiro especificado.

Função `readFileToString(String filePath)`: Esta função lê o conteúdo de um ficheiro como uma sequência de caracteres. Utiliza a classe `Files` da biblioteca Java para ler todos os bytes do ficheiro no caminho especificado. Em seguida, os bytes são decodificados usando a codificação UTF-8 e retornados como uma `string`.

Estas funções são úteis para ler, escrever e manipular documentos XML em Java, garantindo a correta codificação e formatação dos ficheiros.

5. Interface



Instituto Superior de Engenharia

Politécnico de Coimbra

A interface deste trabalho permite visualizar o conteúdo dos ficheiros XML e adicionar escritores. Também é possível realizar pesquisas utilizando XPath com base nos atributos disponíveis e, por fim, gerar outputs.

- **Ver Conteúdo XML:** Nesta opção, o utilizador pode aceder a todas as informações, caso existam, dos escritores ou obras guardados nos ficheiros "escritores.xml" e "obras.xml".
- **Efetuar Pesquisas XPath:** Nesta opção, é possível realizar pesquisas, tais como pesquisar pelo nome do autor e mostrar a informação relevante, pesquisar autores com uma nacionalidade específica, pesquisar as obras de um determinado autor, encontrar o escritor mais premiado, listar os livros publicados por uma determinada editora com preço acima de um valor mínimo e abaixo de um valor máximo.
- **Gerar Outputs:** Nesta parte, é possível gerar diferentes tipos de ficheiros de output:
 - Gerar um ficheiro HTML com uma tabela contendo duas colunas: nome dos autores e fotografias dos autores;
 - Gerar um ficheiro TXT que apresente a listagem de todos os autores;
 - Gerar um ficheiro XML com informações dos 5 livros mais caros.

Estas funcionalidades permitem explorar e manipular os dados contidos nos ficheiros XML de forma conveniente e eficiente.