



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

# **Integração de Dados**

Relatório

Trabalho Prático

2023/2024

Pedro Paiva– 2021134365

Rodrigo Lourenço– 2021155662



**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.

As fontes de dados utilizadas serão o site "<https://pt.wikipedia.org/wiki/>" e o site <https://www.countryreports.org/>. A partir dessas fontes, iremos extrair informações relevantes sobre países. 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:

- Factos.java;
- JDOMFunctions\_Validar.java;
- JDOMFunctions\_XSLT.java;
- HttpRequestFunctions.java
- Jframe.java;
- Pais.java;
- SaxonFunctions\_XQuery.java
- Tpp.java;
- Wrappers.java;
- ModeloXML.java;
- ModeloXML2.java
- utils.java;
- ValidarXML.java
- XPathFunctions.java;
- XMLJDomFunctions.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.

#### **obtemBandeiraLink(String pesquisa)**

Esta função obtém o link da bandeira de um país a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a tag <img> que contém o atributo src com o link da imagem.

**obtemISO(String pesquisa)** Esta função obtém o código ISO 3166-2 de um país a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a seção "ISO 3166-2".

**obtemCapital(String pesquisa)** Esta função obtém a capital de um país a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a tabela "Infobox" e extraíndo o valor da célula associada à capital.

**obtemMoeda(String pesquisa)** Esta função obtém a moeda de um país a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a seção "Moeda".

#### **obtemDominioInternet(String pesquisa)**

Esta função obtém o domínio da internet de um país (por exemplo, ".pt" para Portugal) a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a seção "Internet TLD".



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

**Pop(String pesquisa):** Esta função obtém a população de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Population".

**Area(String pesquisa)** Esta função obtém a área total de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Total Area".

**IdiomasOficiais(String pesquisa)** Esta função obtém os idiomas oficiais de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Language".

**PaísesVizinhos(String pesquisa)** Esta função obtém os países vizinhos de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Location".

**CrescimentoPop(String pesquisa)** Esta função obtém a taxa de crescimento populacional de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Population Growth Rate".

**CidadesmaisPop(String pesquisa)** Esta função obtém as cidades mais populosas de um país a partir da pesquisa fornecida. Ela faz isso acessando o Country Reports (arquivado em 2023-11-30), pesquisando o país e procurando a linha "Population in Major Urban Areas".

**obtemContinente(String pesquisa)** Esta função obtém o continente onde um país está localizado a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando o globo terrestre. O continente é extraído do texto ao lado do globo.



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

**obtemPresidente(String pesquisa)** Esta função obtém o presidente de um país a partir da pesquisa fornecida. Ela faz isso acessando a Wikipedia, pesquisando o país e procurando a seção "Presidente" na tabela "Infobox".

**Wrappers.criaPais(pesquisa)** Esta função cria um objeto Pais com os detalhes de um país a partir da pesquisa fornecida. Ela obtém a bandeira, código ISO, população, continente e presidente (se disponível) do país e os armazena no objeto Pais.

**Wrappers.criaPaisFactos(pesquisa)** Esta função cria um objeto Factos com detalhes mais extensos de um país a partir da pesquisa fornecida.



## 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.

Função `escreverDocumentoString(Document doc)`: Esta função recebe um documento XML e transforma-o numa string usando o método `getPrettyFormat()` que cria um formato para saída legível por humanos.

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



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

## 5. Validar DTD e XSD

A classe ValidarXML oferece duas funções para validar documentos XML:

### **ValidarDocumentoDTD:**

- Lê e verifica se o arquivo DTD existe
- Associa o DTD ao ficheiro XML

### **ValidarDocumentoXSD:**

- Lê e verifica se o arquivo XSD existe
- Associa o XSD ao ficheiro XML





## 6. Interface

A interface deste trabalho permite visualizar o conteúdo dos ficheiros XML e adicionar países. 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 países ou factos nos ficheiros "países.xml" e "factos.xml".
- Efetuar Pesquisas XPath: Nesta opção, é possível realizar pesquisas, tais como pesquisar pelo nome do país e mostrar a informação relevante, pesquisar os factos de um país, pesquisar o país mais populoso, pesquisar países com maior população do que a inserida num continente.
  - Validar DTD & XSD
- Gerar Outputs: Nesta parte, é possível gerar diferentes tipos de ficheiros de output



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra