Integração de Dados

Licenciatura em Engenharia Informática: 2º ano - 2º semestre 2021/2022

Trabalho Prático

Integração de Dados com XML – CIDADES DO MUNDO

***Nota prévia**: O enunciado é propositadamente vago, genérico e incompleto em alguns pontos. O que se pretende é que os alunos avaliem as várias opções existentes e escolham a que considerarem mais apropriada para cada uma das situações com que se depararem. Todas as escolhas devem ser referidas e devidamente justificadas no relatório a entregar.*

1. Objetivos

Com este trabalho pretende-se criar um programa em Java composto por vários Wrappers que obtenham dados de fontes heterogéneas, distribuídas e autónomas e possibilitem ao utilizador a visualização dos dados de forma integrada.

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

Para a realização deste trabalho deve usar a Linguagem Java, Expressões regulares e os APIs JDOM2 e SAXON estudados nas aulas práticas.

2. Resultados da Aprendizagem

Com este trabalho prático pretende-se que se adquiram as seguintes competências:

- Saber 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 e/ou XQuery
- Capacidade de efetuar transformações de ficheiros XML usando XSLT e/ou XQuery
- Capacidade de efetuar validação de ficheiros XML usando DTD e/ou XSD

3. Descrição do Trabalho

O objetivo do trabalho é criar uma aplicação de integração de dados que apresente uma visão unificada de informações relativas a cidades de diferentes países.

A informação deverá ser extraída dos dois sites a seguir apresentados, tratada e integrada em ficheiro(s) XML.

Fontes de dados

- https://pt.wikipedia.org/wiki/
- https://pt.db-city.com/

A palavra de pesquisa fornecida ao programa deve ter o formato: <cidade>, <pais>, por exemplo:

Barcelona, Espanha

Desta expressão de pesquisa deve separar a cidade o país e procurar a informação da cidade nos respetivos sites usando a função HttpRequest disponível no Moodle.

No site wikipedia:

```
HttpRequestFunctions.httpRequest1("https://pt.wikipedia.org/wiki/", cidade, "cidade.txt");
```

>> Depois, aplicar os Wrappers em **cidade.txt** para extrair os dados

No site DBCity:

```
HttpRequestFunctions.httpRequest1("https://pt.db-city.com/", país, "pais.txt");
```

>> De seguida com ajuda de ERs procurar no ficheiro ***pais.txt*** o link da cidade pretendida.
>> Fazer HttpRequest a esse link e criar um ficheiro **cidade2.txt**
>> Depois, aplicar os Wrappers em **cidade2.txt** para extrair os dados.

As fontes de dados são heterogéneas, autónomas e distribuídas e contêm informação relevante sobre o tema do trabalho.

O objetivo do trabalho prático consiste em efetuar **integração de dados** provenientes destas duas fontes de dados e construir um modelo global de dados usando XML. Este modelo de dados deve ser constituído por **um ficheiro XML** onde toda a informação pesquisada seja organizada em elementos/atributos, usando a hierarquia decidida pelos alunos como a mais correta para executar as tarefas propostas.

A informação de cada cidade/país que se pretende guardar no ficheiro é a seguinte (os alunos devem incluir informação adicional que achem relevante):

- Nome da cidade
- País a que pertence
- Indicação se a cidade é a capital do país
- Link para a imagem da bandeira do pais
- Língua(s) oficial(ais) do país
- Link para a imagem da bandeira da cidade
- Links para as imagens de monumentos/*landmarks* da cidade
- Área da cidade (valor numérico em km^2)
- Nº de habitantes da cidade
- Densidade populacional da cidade (nº de habitantes por km^2)
- Código Postal da cidade
- Presidente da Câmara da cidade
- Latitude e Longitude da cidade
- Altitude da cidade em metros
- Clima da cidade
- Fuso Horário da cidade
- Website da cidade
- Cidades geminadas

NOTA: Os **valores numéricos** devem ser tratados para poderem ser corretamente manipulados por pesquisas XPATH/XQUERY. Este tratamento pode passar por eliminar o separador dos milhares, eliminar espaços em branco, colocar o separador decimal como . (ponto) em vez de , (virgula), etc.

EXEMPLOS (do site DBcity):

Número de habitantes Lisboa	507.220 habitantes

>> O Wrapper deve devolver: 507220 (eliminar o separador dos milhares) e posteriormente colocado no XML: `<habitantes>507220</habitantes>`

Densidade populacional Lisboa	5.981,4 /km²

>> O Wrapper deve devolver: 5981.4 (eliminar o separador dos milhares e alterar o separador decimal que no java é o . e não a ,) e posteriormente colocado no XML: `<densidade uni="hab_km2">5981.4</densidade>`

Superfície Lisboa	8.480 hectares 84,80 km²

>> O Wrapper deve devolver: 84.80 (o separador decimal do java é o . e não a ,) e posteriormente colocado no XML: `<area uni="km2">84.80</area>`

Por exemplo, a aplicação de todos os Wrappers com a introdução de Lisboa, Portugal deve devolver informação relevante:

- Nome da cidade: Lisboa
- País: Portugal
- Capital: true
- Bandeira país: https://dwpt1kkww6vki.cloudfront.net/img/drapeau/120/170.png
- Lingua oficial: Português, Mirandese
- Bandeira cidade: https://upload.wikimedia.org/wikipedia/commons/thumb/1/16/Bandeira_municipal_de_Lisboa.png/90px-Bandeira_municipal_de_Lisboa.png
- Lista de imagens de Monumentos: [https://upload.wikimedia.org/wikipedia/commons/thumb/4/41/Lisbon_%2836831596786%29_%28cropped%29.jpg/280px-Lisbon_%2836831596786%29_%28cropped%29.jpg, https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/Rua_Augusta_Arch_-_April_2019_%28cropped%29.jpg/139px-Rua_Augusta_Arch_-_April_2019_%28cropped%29.jpg, …]
- Área: 84.80
- Num habitantes: 507220
- Densidade pop: 5981.4
- C Postal: 1100
- Presidente: Fernando Medina
- Latitude: 38.7071
- Longitude: -9.13549
- Altitude: 4
- Clima: Clima mediterrânico (Classificação climática de Köppen-Geiger: Csa)
- FUSO: UTC +0:00 (Europe/Lisbon)
- Website: https://www.lisboa.pt
- Cidades geminadas: [Rio de Janeiro, São Paulo, Praia, …]

Os alunos devem analisar as diferentes fontes de dados e apresentar um estudo das mesmas, decidindo e justificando onde vão retirar a informação. Obrigatoriamente, devem ser usadas **as duas** fontes de dados fornecidas.

O esquema a adotar na vista unificada deve ser decidido pelos alunos e validado usando o XSD e o DTD apropriado.
Depois de realizado o processo de integração dos dados, o utilizador poderá fazer pesquisas sobre a vista unificada.

No moodle encontra-se uma lista de cidades que podem servir de teste da aplicação. Depois de terminada a aplicação, os alunos devem testar com outros nomes e avaliar a funcionalidade do sistema implementado.

4. TAREFAS A REALIZAR

Encontram-se em seguida as **tarefas principais** a desenvolver neste trabalho prático. As descrições são genéricas e os exemplos apresentados servem apenas para uma melhor compreensão do que é pretendido. Os alunos devem ser criativos e apresentar uma solução integradora completa e funcional que permita efetuar uma grande diversidade de pesquisas.

4.1. ANALISAR AS FONTES DE DADOS (S)

A primeira parte do trabalho consiste em analisar as fontes de dados e verificar onde pode ser encontrada a informação sobre as cidades.

Todas as situações de exceção devem avaliadas e as decisões tomadas devem ser justificadas no relatório. Por exemplo:

- Caso encontre informação duplicada nas várias fontes
- Se uma cidade não for encontrada
- Se algum dos atributos pedido não existir para algumas cidades
- ...

4.2. DEFINIR O ESQUEMA GLOBAL (G)

Defina um modelo global para a recolha dos dados. Este modelo deve ser baseado num ficheiro XML com a estrutura hierárquica adequada ao problema proposto. Isto é, o aluno deve analisar qual a estrutura do ficheiro que considera mais adequada no que se refere ao nível de ramificação e à escolha de elementos ou atributos para guardar os dados. O esquema a adotar na vista unificada decidido pelos alunos deve ser sempre validado usando o XSD e o DTD apropriado.

4.3. IMPLEMENTAR WRAPPERS (MAPEAMENTOS M)

Implementar os *Wrappers* que permitam obter a informação relevante de cada fonte de dados. Estes *Wrappers* devem ser implementados usando expressões regulares. No relatório deve ser descrito detalhadamente cada um dos wrappers, indicando que informação é retirada por cada um deles da fonte de dados em que cada um opera. Para cada atributo a encontrar, deve(m) ser selecionada(s) a(s) fonte(s) de dado(s) relevante(s). No caso de encontrar inconsistências ou conflitos os alunos terão de propor uma solução.

Para saber como implementar os Wrappers deve analisar a estrutura das páginas HTML onde vai procurar a informação.

Use a função *httpRequest* dada nas aulas práticas para aceder às páginas e gravá-las em disco.

O número e a estrutura dos wrappers depende da forma e da quantidade de informação que se quer encontrar e deve ser analisada pelos estudantes.

4.4. GERAR / MANIPULAR FICHEIRO XML: ACRESCENTAR, EDITAR E ELIMINAR DADOS

Depois de implementados os *wrappers*, os dados devem ser guardados num ficheiro XML usando o modelo escolhido. Deverá ser possível

- Adicionar uma nova cidade, desde que esta não exista no ficheiro XML.
- Se o ficheiro não existir ainda, deve ser criado com a inserção da primeira cidade.
- Eliminar uma cidade (usar nome da cidade como palavra de pesquisa).
- Editar/alterar alguns atributos do ficheiro XML (área, código postal, clima, ...)

4.5. VALIDAR O MODELO G

Os ficheiros do modelo **G** devem ser validados usando os XSD/DTD escolhidos.

Esta tarefa deve ser feita usando o API JDOM2 dado nas aulas práticas.

4.6. FAZER PESQUISAS XPATH

Permitir ao utilizador efetuar diferentes pesquisas sobre o ficheiro XML:

- Pesquisar pelo nome da cidade e mostrar a informação relevante (país, é capital?, área, nº de habitantes, ...)
- Pesquisar cidades de um dado país
- Pesquisar cidades com número de habitantes superior a um valor introduzido
- Pesquisar cidades com um determinado clima
- Pesquisar todas as capitais existentes no ficheiro
- (outras pesquisas propostas pelos alunos terão cotação adicional)

4.7 GERAR FICHEIROS DE OUTPUT (XSLT/XQUERY)

O programa deve possibilitar ao utilizador gerar ficheiros de resultados. Estes ficheiros devem ser transformações do ficheiro XML da vista global.
quatro transformações **obrigatórias**:

- Gerar ficheiro HTML de fotos das bandeiras dos países das cidades inseridas no ficheiro (sem repetições)
- Gerar ficheiro TXT que mostre a listagem das cidades de um dado país
- Gerar um ficheiro XML com 5 cidades mais populosas
- Pedir ao utilizador o nome de uma cidade e gerar um ficheiro HTML que mostre as imagens dos monumentos/landscapes dessa cidade
- Os alunos devem propor no mínimo **mais três** transformações adicionais. Devem implementar as transformações usando as duas tecnologias dadas nas aulas - XSLT e XQuery – optando pela que for mais adequada em cada situação.

4.8. INTERFACE GRÁFICO

A aplicação deve ter uma interface amigável e intuitiva, disponibilizando ao utilizador um conjunto de opções, por exemplo, sugere-se a seguinte estrutura:

- Opções gerais
  - Ver conteúdo do ficheiro XML
  - Validar modelo de dados (DTD e XSD)
  - Sair da aplicação

- Alterar dados do modelo XML (efetue sempre a validação do modelo em cada uma das opções)
  - Eliminar uma cidade do ficheiro (usar nome como palavra de pesquisa)
  - Acrescentar uma cidade que não exista no ficheiro
    - pedir <cidade, pais> e usar os Wrappers para obter os dados da web
  - Alterar alguns atributos de uma cidade

- Efetuar Pesquisas XPATH
  - ...

- Gerar Outputs
  - ...

5. NORMAS PARA REALIZAÇÃO DO TRABALHO

O trabalho deverá ser realizado **individualmente ou em grupos de dois alunos**.

O trabalho vale 6 valores e é necessário um mínimo de 35% para aprovação na Unidade Curricular.

O trabalho final deve ser entregue até **12 de Junho de 2022** às 23h55 GMT.
>>>>>> DATA ÚNICA DE ENTREGA PARA TODAS AS ÉPOCAS DE EXAME <<<<<<

A entrega dos trabalhos deverá ser feita usando a plataforma Moodle. Deve ser submetido um ficheiro compactado cujo nome deve conter a identificação dos elementos do grupo de trabalho:
Por exemplo: **a22222_AnaMatos_a33333_RuiMelo_P1.zip**

O ficheiro deve conter o projeto Java com a implementação da aplicação e todos os ficheiros DTD, XSD, XSLT, XQuery, etc que foram implementados.

Os trabalhos serão sujeitos a **defesa obrigatória** nas aulas das semanas 13 a 24 de Junho.

6. CRITÉRIOS DE AVALIAÇÃO

O trabalho vale **6 valores** na nota final da Unidade Curricular. Para aprovação na UC é necessário ter um mínimo de 35% neste trabalho.

O trabalho será avaliado segundo os seguintes critérios:

- Qualidade e correção na implementação das tarefas solicitadas
- Funcionalidade do programa
- Originalidade e diversificação dos conteúdos abordados, nomeadamente as funcionalidades extras
- Justificação das opções tomadas
- Qualidade do relatório entregue
- Qualidade da defesa

Bom trabalho!