Instituto Superior de Engenharia de Coimbra
Engenharia Informática
Integração de Dados

Trabalho Prático

Integração de Dados

2020/2021

P2

Integração de Dados com XML

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

Índice

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

Instituto Superior de Engenharia de Coimbra
Engenharia Informática
Integração de Dados

Introdução:

Com este trabalho criamos um programa em Java composto por vários Wrappers que obtêm dados de fontes heterogéneas, distribuídas e autónomas e possibilitam ao utilizador a visualização dos dados de forma integrada.

O utilizador conseguirá fazer várias pesquisas sobre vários jogadores, alterar/acrescentar alguns dados sobre esses jogadores e também gerar ficheiros de diferentes tipos com pesquisas de jogadores escolhidos.

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

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

Informação a ser integrada

Como referimos na introdução, o trabalho é baseado em integrar informação, neste caso relativamente a jogadores de futebol.

Foi nos proposta a seguinte lista de dados a integrar:

• nome completo do jogador

• nome pelo qual é mais conhecido

• fotografia (link)

• data de nascimento, idade

• nacionalidade

• altura

• peso

• pé preferencial

• posições onde joga

• clube atual

• clubes anteriores

• seleção nacional (se for o caso)

• prémios ganhos (taça, dat, etc)

• estado atual: ativo, reformado

• valor do contrato

• empresário

• ranking dado pelo site transfer markt

Decidimos seguir a recomendação e ainda acrescentar mais um parâmetro que foi dividir o ranking dado pelo transfer markt em dois o ranking global e o ranking do clube em que joga.

Ao longo da implementação ainda nos deparamos com o facto que alguns jogadores poderiam já ter acabado a carreira ou até mesmo terem falecido, e que alguns destes parâmetros não poderiam ser preenchidos. A solução para esse problema irá ser descrita no capítulo em que descrevemos a implementação dos Wrappers.

Fontes de dados

Foi nos proposto que os dados deveriam ser provenientes de páginas web.

Para além disso, ainda nos foi sugerido as seguintes 3 páginas web como fontes de dados:

- https://zerozero.pt/
- https://pt.wikipedia.org/wiki/
- https://www.transfermarkt.pt/

Após uma análise de como a informação estava estruturada nos diversos sites chegamos á conclusão de que o site que tinha a informação mais organizada era o transfermarket.

Outra conclusão que retiramos foi que o site zerozero quando lhe são solicitados muitos httprequest's de uma só vez ele bloqueia a possibilidade de pesquisa durante um certo período de tempo.

Por último constatamos que a wikipedia, apesar de ser o site que tem maior quantidade de informação sobre o jogador, não é dos melhores sites para realizar a integração de dados, pois a informação html dele não se encontra bem organizada e devido a ser um site em que existe informação de vários temas é difícil chegar á informação de um jogador se o nome deste for igual ao nome de outra personalidade.

Posto isto decidimos então utilizar principalmente o site transfermarket, um pouco do site wikipedia e decidimos não utilizar o site zerozero devido ao problema em cima descrito.

Por fim para contornar o problema do zerozero e da wikipedia decidimos recorrer a dois sites:

https://football.fandom.com/wiki/

https://pt-br.soccerwiki.org/

Destes sites foi retirada informação que não era possível no site do transfermarket.

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

ESQUEMA GLOBAL (G)

O sistema global onde são guardados os dados recolhidos é um ficheiro XML, em que é apresentada uma lista dos jogadores em que cada jogador tem as suas informações:

A imagem seguinte apresenta o modelo global utilizado.

```
<?xml version=""1.0" encoding=""utf-8"?>
<!DOCTYPE jogadores SYSTEM "jogadores.dtd">
<jogadores xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="jogadores.xsd">
    <jogador nome="Cristiano Ronaldo" numero="7">
        <nomec>Cristiano Ronaldo dos Santos Aveiro</nomec>
        <foto>https://img.a.transfermarkt.technology/portrait/header/8198-1568120625.jpg?lm=1</foto>
        <DataN>05/02/1985</DataN>
        <nacionalidade>Portugal</nacionalidade>
        <altura>187</altura>
        <peso>80</peso>
        <pepref>direito</pepref>
        <posicao>Ponta de Lança</posicao>
        <clube>Juventus FC</clube>
        <selecao>Portugal</selecao>
        <valor>45,00M €</valor>
        <empresario>Gestifute</empresario>
        <ranking tipo="global">96</ranking>
        <ranking tipo="clube">4</ranking>
        <premios>[2x Melhor jogador do mundo, 5x Vencedor Ballon d'Or, 3x O Futebolista do Ano da Europa, 18x Melhor marcador, 5x Jogador da época, 10x Futebolista do ano]</premios>
    </jogador>
    <jogador nome="Jordi Alba" numero="18">
        <nomec>Jordi Alba Ramos</nomec>
        <foto>https://img.a.transfermarkt.technology/portrait/header/69751-1454079452.jpg?lm=1</foto>
        <DataN>21/03/1989</DataN>
        <nacionalidade>Espanha</nacionalidade>
        <altura>170</altura>
        <peso>68</peso>
        <pepref>esquerdo</pepref>
        <posicao>Lateral Esquerdo</posicao>
        <clube>FC Barcelona</clube>
        <selecao>Espanha</selecao>
        <valor>20,00M €</valor>
        <empresario>InterStarDeporte</empresario>
        <ranking tipo="global">389</ranking>
        <ranking tipo="clube">15</ranking>
        <premios>[Sem Premios]</premios>
    </jogador>
</jogadores>
```

Neste modelo podemos ver que a raiz é o elemento folha jogadores sendo que dentro deste mesmo elemento se encontra uma lista dos vários jogadores, cada um com a informação integrada a partir dos Wrappers.

Para adicionar a informação neste modelo fizemos uma função chamada adicionaJogador, que está representada na imagem abaixo:

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

```java
public static Document adicionaJogador(Jogador jog, Document doc) throws SaxonApiException {
    Element raiz;
    XdmValue res = null;
    if (doc == null) {
        raiz = new Element("jogadores"); //cria <catalogo>...</catalogo>
        doc = new Document(raiz);
    } else {
        raiz = doc.getRootElement();
        String xp = "//jogador[@nome ='" + jog.getNome() + "']";
        System.out.println(xp);
        res = XPathFunctions.executaXpath(xp, xmlFile: "jogador.xml");
        if (res != null && res.size() > 0) {
            System.out.println( x: "O Jogador JÁ EXISTE");
            return doc;
        } else { //ADICIONA

            Element jogador = new Element("jogador");
            Attribute nome = new Attribute("nome", jog.getNome());
            jogador.setAttribute(nome);
            Attribute numero = new Attribute("numero", jog.getNumero());
            jogador.setAttribute(numero);
            Element nomec = new Element("nomec").addContent(jog.getNomec());
            Element foto = new Element("foto").addContent(jog.getLink_foto());
            Element datan = new Element("DataN").addContent(jog.getDatanasc());
            Element nacionalidade = new Element("nacionalidade").addContent(jog.getNacionalidade());
            Element altura = new Element("altura").addContent(jog.getAltura());
            Element peso = new Element("peso").addContent(jog.getPeso());
            Element pepref = new Element("pepref").addContent(jog.getPe_pref());
            Element posicao = new Element("posicao").addContent(jog.getPosicao());
            Element clube = new Element("clube").addContent(jog.getClube_atual());
            Element selecao = new Element("selecao").addContent(jog.getSelecao());
            Element valor = new Element("valor").addContent(jog.getValor_contrato());
            Element empresario = new Element("empresario").addContent(jog.getEmpresario());
            Element ranking = new Element("ranking").addContent(jog.getRanking_tm());
            Element rankingClube = new Element("ranking").addContent(jog.getRanking_clube());
            Attribute tipog = new Attribute("tipo", "global");
            Attribute tipoc = new Attribute("tipo", "clube");
            ranking.setAttribute(tipog);
            rankingClube.setAttribute(tipoc);
            Element premios = new Element("premios").addContent(jog.getPremios().toString());
            jogador.addContent(nomec);
            jogador.addContent(foto);
            jogador.addContent(datan);
            jogador.addContent(nacionalidade);
            jogador.addContent(altura);
            jogador.addContent(peso);
            jogador.addContent(pepref);
```

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

WRAPPERS (MAPEAMENTOS M)

Os wrappers foram criadoos para que fosse possível retirar a informação das diversas fontes de dados.

Alguns dos Wrappers implementados foram:

- Wrapper Nacionalidade:

```
public static String WrapperNacionalidade(String nome) throws IOException {
    String link2 = encontraLinkTransferProfil(nome);
    HttpRequestFunctions.httpRequest1(link2, pesquisa: "", outFile: "transfer.txt");

    String erl = "<span itemprop=\"nationality\">([^<]+)</span>";
    Pattern p = Pattern.compile(erl);
    Matcher m;
    Scanner ler = new Scanner(new FileInputStream("transfer.txt"));
    while (ler.hasNextLine()) {
        String linha = ler.nextLine();
        m = p.matcher(linha);
        if (m.find()) {
            ler.close();
            return m.group( group: 1);
        }
    }
    return "Nacionalidade não encontrada";
}
```

Este Wrapper retira a Nacionalidade do jogador do site transfermarket

- Wrapper Altura

```
public static String WrapperAltura(String nome) throws IOException {

    String link = encontraLinkSoccerW(nome);
    HttpRequestFunctions.httpRequest1(link, pesquisa: "", outFile: "soccerw.txt");

    String erl = "cm[)]</th><td class=\"left\">([0-9]+)</td>";
    Pattern p = Pattern.compile(erl);
    Matcher m;
    Scanner ler = new Scanner(new FileInputStream("soccerw.txt"));
    while (ler.hasNextLine()) {
        String linha = ler.nextLine();
        m = p.matcher(linha);
        if (m.find()) {
            ler.close();
            return m.group( group: 1);
        }
    }
    return "Altura não encontrada";
}
```

Este Wrapper retira a altura do jogador do site Soccerwiki

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

- Wrapper Nome Completo

```
public static String WrapperNomeCompleto1(String nome) throws IOException {
    String link = encontraLinkWikiF(nome);
    HttpRequestFunctions.httpRequest1(link, pesquisa: "", outFile: "wikif.txt");
    String er1 = "<b>Full.+</b>";
    String er2 = "<td colspan=\"2\">([a-zA-Z\\sãõéíàáóçú]+)";

    Pattern p = Pattern.compile(er1);
    Pattern p2 = Pattern.compile(er2);

    Matcher m;
    Scanner ler = new Scanner(new FileInputStream("wikif.txt"));
    while (ler.hasNextLine()) {
        String linha = ler.nextLine();
        m = p.matcher(linha);
        if (m.find()) {
            linha = ler.nextLine();
            linha = ler.nextLine();
            m = p2.matcher(linha);
            if (m.find()) {
                ler.close();
                return m.group( group: 1);
            }
        }
    }
    return "Nome Completo não encontrado";
}
```

Este Wrapper retira o nome completo do site Wiki Football

GERAR / MANIPULAR FICHEIRO XML: ACRESCENTAR, EDITAR E ELIMINAR DADO

As funções de manipulação do ficheiro XML que foram implementadas foram as seguintes:

- Remove Jogador:

```
public static Document removeJogador(String nome, Document doc) {
    Element raiz;
    if (doc == null) {
        System.out.println( x: "Ficheiro não existe, nada a remover");
        return null;
    } else {
        raiz = doc.getRootElement();
    }
    List todosJogadores = raiz.getChildren( cname: "jogador");
    boolean found = false;
    for (int i = 0; i < todosJogadores.size(); i++) {
        Element jogador = (Element) todosJogadores.get(i); //obtem livro i da Lista
        if (jogador.getAttributeValue( attname: "nome").equals(nome)) {
            jogador.getParent().removeContent(jogador);
            System.out.println( x: "Jogador removido com sucesso!");
            found = true;
        }

    }
    if (!found) {
        System.out.println("Jogador " + nome + " não foi encontrado");
        return null;
    }
    return doc;

}
```

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

- Altera Clube

```
public static Document alteraClube(String nome, String novoClube, Document doc) {
    Element raiz;
    if (doc == null) {
        System.out.println( x: "Ficheiro não existe, nada a alterar");
        return null;
    } else {
        raiz = doc.getRootElement();
    }
    List todosJogadores = raiz.getChildren( cname: "jogador");
    boolean found = false;
    for (int i = 0; i < todosJogadores.size(); i++) {
        Element jogador = (Element) todosJogadores.get(i); //obtem jogador i da Lista
        if (jogador.getAttributeValue( attname: "nome").equals(nome)) {
            jogador.getChild( cname: "clube").setText(novoClube);
            found = true;
        }

    }
    if (!found) {
        System.out.println("Jogador " + nome + " não foi encontrado");
        return null;
    } else {
        System.out.println( x: "Clube alterado!");
    }
    return doc;
}
```

- Altera idade

```
public static Document alteraIdade(String nome, String novoIdade , Document doc) {
    Element raiz;
    if (doc == null) {
        System.out.println( x: "Ficheiro não existe, nada a alterar");
        return null;
    } else {
        raiz = doc.getRootElement();
    }
    List todosJogadores = raiz.getChildren( cname: "jogador");
    boolean found = false;
    for (int i = 0; i < todosJogadores.size(); i++) {
        Element jogador = (Element) todosJogadores.get(i); //obtem jogador i da Lista
        if (jogador.getAttributeValue( attname: "nome").equals(nome)) {
            jogador.getChild( cname: "idade").setText(novoIdade);
            found = true;
        }

    }
    if (!found) {
        System.out.println("Jogador " + nome + " não foi encontrado");
        return null;
    } else {
        System.out.println( x: "Idade alterado!");
    }
    return doc;
}
```

```java
public static Document alteraNacionalidade(String nome, String novoNacionalidade , Document doc) {
    Element raiz;
    if (doc == null) {
        System.out.println( x: "Ficheiro não existe, nada a alterar");
        return null;
    } else {
        raiz = doc.getRootElement();
    }
    List todosJogadores = raiz.getChildren( cname: "jogador");
    boolean found = false;
    for (int i = 0; i < todosJogadores.size(); i++) {
        Element jogador = (Element) todosJogadores.get(i); //obtem jogador i da Lista
        if (jogador.getAttributeValue( attname: "nome").equals(nome)) {
            jogador.getChild( cname: "nacionalidade").setText(novoNacionalidade);
            found = true;
        }

    }
    if (!found) {
        System.out.println("Jogador " + nome + " não foi encontrado");
        return null;
    } else {
        System.out.println( x: "Nacionalidade alterado!");
    }
    return doc;
}
```

VALIDAR O MODELO G

Para validar o nosso modelo global(Ficheiro Xml) usamos o ficheiros DTD e XSD.

- Ficheiro DTD

```
<!ELEMENT nomec (#PCDATA)>
<!ELEMENT foto (#PCDATA)>
<!ELEMENT DataN (#PCDATA)>
<!ELEMENT nacionalidade (#PCDATA)>
<!ELEMENT altura (#PCDATA)>
<!ELEMENT peso (#PCDATA)>
<!ELEMENT pepref (#PCDATA)>
<!ELEMENT posicao (#PCDATA)>
<!ELEMENT clube (#PCDATA)>
<!ELEMENT selecao (#PCDATA)>
<!ELEMENT valor (#PCDATA)>
<!ELEMENT empresario (#PCDATA)>
<!ELEMENT ranking (#PCDATA)>
<!ELEMENT premios (#PCDATA)>
<!ELEMENT jogador (nomec,foto,DataN,nacionalidade,altura,peso,pepref,posicao,clube,selecao,valor,empresario,ranking+,premios)>
<!ELEMENT jogadores (jogador)+>

<!ATTLIST jogador  nome CDATA #REQUIRED>
<!ATTLIST jogador  numero CDATA #REQUIRED>
<!ATTLIST ranking  tipo (global|clube) #REQUIRED>
<!ATTLIST jogadores xmlns:xsi CDATA #IMPLIED>
<!ATTLIST jogadores xsi:noNamespaceSchemaLocation CDATA #IMPLIED>
```

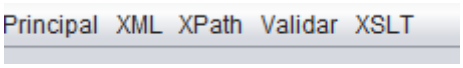
Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

- Ficheiro XSD

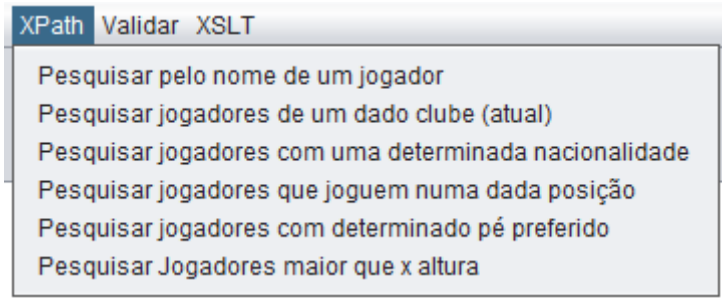
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="nomec" type="xsd:string"/>
    <xsd:element name="foto" type="xsd:string"/>
    <xsd:element name="DataN" type="xsd:string"/>
    <xsd:element name="nacionalidade" type="xsd:string"/>
    <xsd:element name="altura" type="xsd:string"/>
    <xsd:element name="peso" type="xsd:string"/>
    <xsd:element name="pepref" type="xsd:string"/>
    <xsd:element name="posicao" type="xsd:string"/>
    <xsd:element name="clube" type="xsd:string"/>
    <xsd:element name="selecao" type="xsd:string"/>

    <xsd:element name="valor" type="xsd:string"/>
    <xsd:element name="empresario" type="xsd:string"/>
    <xsd:element name="premios" type="xsd:string"/>

    <xsd:element name="ranking">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute ref="tipo" use="required"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="jogador">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="nomec"/>
                <xsd:element ref="foto"/>
                <xsd:element ref="DataN"/>
                <xsd:element ref="nacionalidade"/>
                <xsd:element ref="altura"/>
                <xsd:element ref="peso"/>
                <xsd:element ref="pepref"/>
                <xsd:element ref="posicao"/>
                <xsd:element ref="clube"/>
                <xsd:element ref="selecao"/>
                <xsd:element ref="valor"/>
                <xsd:element ref="empresario"/>
                <xsd:element ref="ranking" maxOccurs="2"/>
                <xsd:element ref="premios"/>
            </xsd:sequence>
            <xsd:attribute ref="nome" use="required"/>
            <xsd:attribute ref="numero" use="required"/>
```

PESQUISAS XPATH

As pesquisas Xpath foram diretamente implantadas na interface gráfica:



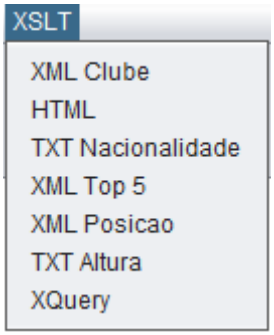
Sendo que as pesquisas foram as seguintes :



Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

FICHEIROS DE OUTPUT (XSLT/XQUERY)

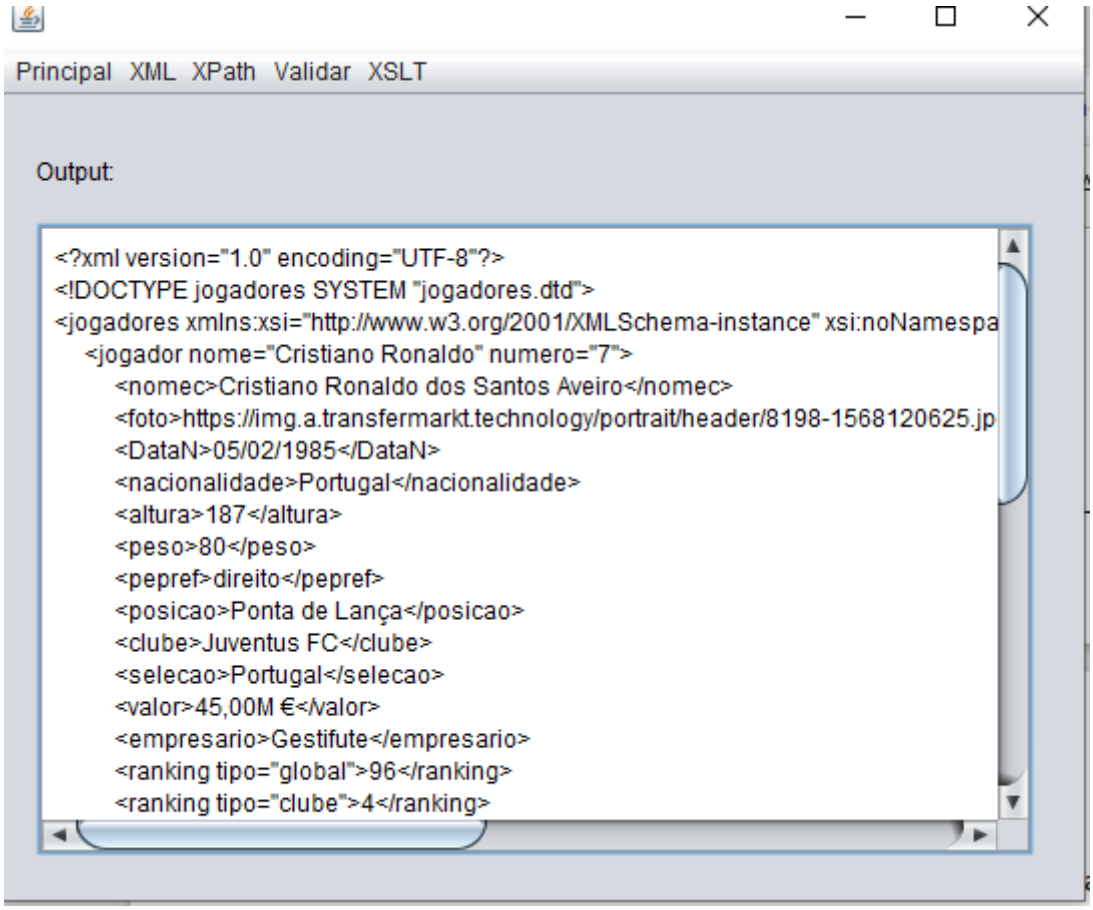
Os ficheiros de output também foram diretamente implantados na interface gráfica.



Foram usadas funções tanto em XSLT como XQUERY para gerar estes mesmos ficheiros de outputs.

INTERFACE GRÁFICO

Foi por fim implementada uma interface gráfica para que fosse disponibilizada todas as funções acima referidas o mais intuitivo e amigável possível para o utilizador.



Diogo Lopes- 2018019746
Leonardo Marques – 2019123778

Conclusão

Com este trabalho prático conseguimos aprender 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

Para além disto achamos o tema do trabalho bastante interessante, o que nos fez quer saber mais sobre cada uma das competências adquiridas.

Diogo Lopes- 2018019746
Leonardo Marques – 2019123778