

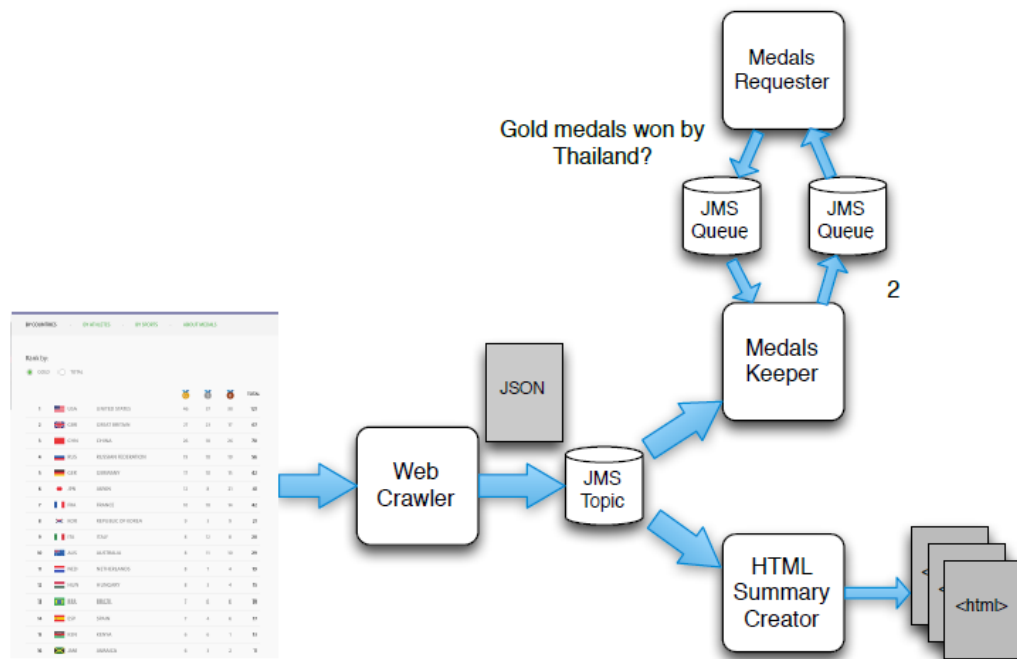


# UNIVERSIDADE DE COIMBRA

## Integração de Sistemas

Ana Inês Mesquita Fidalgo - 2013134819  
Diogo Miguel Costa Goncalves - 20131140087

October 13, 2016



## 1 Introdução

Neste trabalho era-nos pedido que implementássemos quatro aplicações. Um Web Crawler que nos permite ir buscar informação relativamente às medalhas dos Jogos Olímpicos. Extrai a informação para um XML e envia-a para um Java Message Service Topic. Este tópico contém mais duas aplicações que processam a informação, o MedalKeeper, que guarda a informação das medalhas, dos atletas e dos países em memória, e o Medal Requester que pede informação ao Medal Keeper. A outra aplicação é o HTML Summary Creator que a partir da informação retirada e guardada no XML cria um HTML.

## 2 Manual de Programador

### 2.1 Web Crawler

Utilizando os metodos da biblioteca Jsoup, extraímos a informacao relativa às medalhas de cada país e de cada atleta, guardando-as em classes Java. Utilizamos o metodo connect() para obter o DOM do website e percorremo-lo com o metodo select(). Para retirar a informação dos países acedemos aos objetos tr com a classe table-medal-countries e de seguida aos filhos. Para os atletas, acedemos a todos os objetos tr com classe type, depois o primeiro filho e depois os irmãos. Em seguida, geramos um ficheiro .xml com as classes em Java preenchidas e devidamente anotadas. Por fim, tentamos enviar a informação em forma de String para o tópico, esperando um tempo com crescimento exponencial, começando em 10 segundos, entre cada tentativa.//

As classes Mundo, XML e XSD foram criadas em Java, a partir dessas mesmas classes definimos a configuração do XML e desse o XSD. A classe superior é a classe Mundo, esta contém uma lista de Países (o mesmo formato pode ser encontrado no XML e é verificado pelo XSD). Cada País tem a variável ouro, prata e bronze (int no Java, byte para verificação do XSD), nome do país (atributo), abreviatura do nome (elemento) e uma lista de Modalidades (wrapper de elementos). Modalidade tem um atributo, nome e um elemento: lista de Pessoa. Cada obejcto da classe Pessoa tem três atributos: o nome, tipo de medalha e tipo de modalidade, todos estes no formato de String.

### 2.1.1 TopicSender

Esta classe vai permitir que seja enviada toda a informação retirada do Web Crawler, em forma de String, para um Tópico criado no Wildfly.

### 2.1.2 TopicSSincro

A informação retirada será alojada numa lista no Medal Keeper, para isto, é necessário um receiver que irá buscar a informação ao Tópico do Wildfly, e envia-lá para o Medal Keeper.

## 2.2 HTML Creator

Esta aplicação verifica a existência de informação no tópico, onde foi alojada informação a partir do Web Crawler. O HTML Creator retira a informação em formato String, verifica a sua validade com o XSD, e cria um ficheiro HTML com a informação ordenada com ajuda do XSL e guarda no ficheiro Table.html.

## 2.3 Medal Keeper

No Medal Keeper ficará alojada toda a informação retirada do site dos jogos olímpicos. Esta informação é retirada do tópico a partir da classe TopicSSincro. Quando a mensagem é recebida no Medal Keeper esta é primeiramente verificada com o XSD criado anteriormente de modo a verificar que a nossa mensagem é válida. Depois de ser validada, é efectuado um unmarshall para criarmos as classes java, e toda a informação de países, medalhas e pessoas fica alojada em uma lista, a lista "a", que servirá no futuro para responder a diversos pedidos feitos pelo Medal Requester.

### 2.3.1 QueueReceiver

Quando o Medal Keeper recebe um request do Medal Requester, este pedido fica alojado em uma Queue criada no Wildfly. Esta classe permite ao Medal Keeper receber esse pedido da PlayQueue. Quando esta Message é recebida o Medal Keeper irá procurar a resposta na função "requester" a partir da string property da mensagem e da sua int property, sendo que cada um destes parâmetros correspondem à query da qual queremos obter informação e o seu tipo, tipo 1 se é um país, tipo 2 se é uma modalidade e tipo 3 se é uma pessoa, além disto cada pedido tem o seu próprio id para garantir que lhe é enviada a resposta correcta. Depois de encontrada a informação relativa à query a pesquisar, é criada uma mensagem nova, na qual a string property corresponde à resposta, e esta é enviada.

### 2.3.2 ReqSendKeeper

Esta classe é a classe responsável por enviar a resposta final para uma queue temporária que foi criada aquando foi feito o pedido. A mensagem de resposta é enviada juntamente juntamente com o seu destino, a queue temporária.

## 2.4 Medal Requester

Esta classe é que irá realizar os pedidos à classe Medal Keeper, onde está alojada a informação retirada do site dos Jogos Olímpicos. Primeiramente é realizado um pedido através de interação user. Este pedido é realizado na função pedido() que está alojada na classe Medal Keeper. O utilizador insere o tipo e o que procura, sendo gerada assim uma string. A partir desta string retiramos a query, o que o utilizador pretende procurar, e o tipo de procura.

### 2.4.1 Requests

A classe Requests é responsável por enviar o pedido para a queue PlayQueue criada no Wildfly. Tendo obtido a query e o tipo de pedido anteriormente, estes são enviados para a função send-create() alojada nesta classe. Nesta função é criada uma mensagem com string property igual a query e uma int property igual ao tipo. De seguida é criada uma queue temporária, para onde será enviada uma resposta futuramente e é definido que a reply da mensagem tem a sua referência. Sendo finalmente enviado o pedido para a PlayQueue.

Depois de o Medal Keeper enviar a sua resposta para a queue temporária, é feito o receive desta resposta também nesta classe a partir da referência da queue temporária e é impressa a resposta ao pedido do Utilizador.

O Medal Keeper continua à escuta de novos pedidos feitos pelo Medal Requester, e o Medal Requester fica à escuta de respostas para os seus pedidos. Sendo que o Medal Keeper continua à escuta do tópico, para ter a sua informação sempre actualizada, caso ocorra alguma mudança de informação no site dos Jogos Olímpicos.