

Integração de Sistemas

1º Semestre

**XML and XML Manipulation, Java Message Service and Message Oriented Middleware**



Trabalho realizado por:

* Daniel Coelho Bastos 2009109605
* Tiago Martins Andrade 2010165760

2015/2016

# Índice

[Introdução 3](#_Toc432621566)

[Overview 4](#_Toc432621567)

[Aplicações 5](#_Toc432621568)

[1. Web Crawler 5](#_Toc432621569)

[2. HTML Summary Creator 6](#_Toc432621570)

[3. Price Keeper 7](#_Toc432621571)

[3.1. Queue/Temporary Queue 8](#_Toc432621572)

[4. Cliente/Price Requester 9](#_Toc432621573)

[Testes 10](#_Toc432621574)

[Conclusão 11](#_Toc432621575)

# Introdução

Este projecto tem como objetivo a aprendizagem e compreensão de competências para a implementação de um sistema de parsing e comunicação entre servidores para aceitar resquests de preços de produtos dos sites <http://www.pixmania.pt> e <http://www.worten.pt>.

Os produtos em questão são smartphones e o objetivo é aceitar pedidos através da pesquisa do seu nome e devolver o preço.

As linguagens de programação utilizadas são o Java, XML, XSD e XSL. Os IDE’s utilizados foram o IntelliJ e o Eclipse. Como Servidor de Aplicação foi utilizado o Wildfly que fornece a API, o modelo de arquitetura e o ambiente de execução definidos na especificação JAVA EE que inclui o container web e diversos outros subsistemas Java EE (Segurança, computação e acessos distribuídos, JAX-RS, JAX-WS, JSF, JMS, JTA, CDI, EJB entre outros).

# Overview

O sistema divide-se em 4 partes:

* [Web Crawler](#_Web_Crawler)
* [HTML Summary Creator](#_HTML_Summary_Creator)
* [Price Keeper](#_Price_Keeper)
* [Price Requester](#_Clients/Requesters)

A informação e preço dos smartphones é obtida pelo WebCrawler e enviada para o Price Keeper através do Java Message Service Wildfly. O Price Keeper faz a gestão entre a Queue dos requests feitos pelos Clientes/Price Requesters e as Temporary Queues criadas para lhes enviar as respostas. Existe ainda o HTML Summary Creator que cria uma página HTML para apresentar a informação obtida pelo WebCrawler.

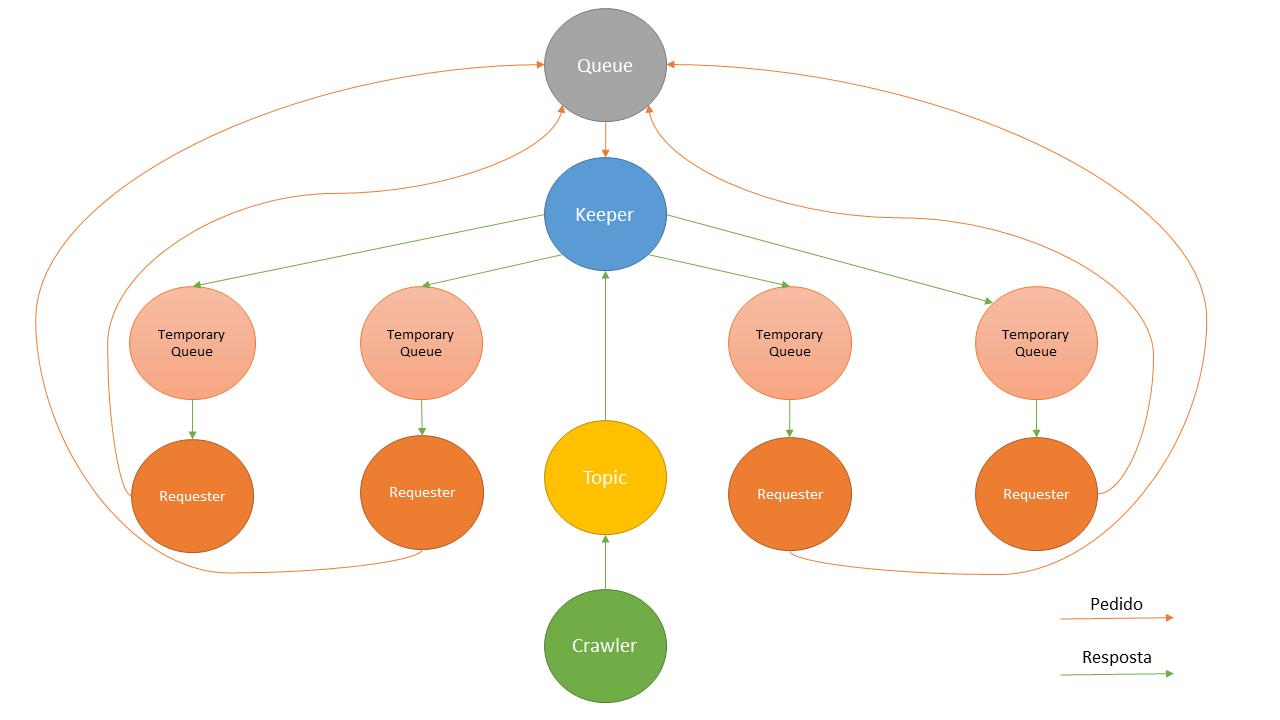


Figura - Organização das Aplicações

# Aplicações

## Web Crawler

O Web Crawler é a aplicação que faz o parsing de um website e envia a informação para o servidor Wildfly. Para enviar a informação tenta 5 vezes fazer a conexão. Caso exista conexão ao servidor Wildfly envia a informação como mensagem, caso não exista transforma-o num XML. Utilizámos a biblioteca Jsoup para fazer o parsing das várias páginas HTML do site.

Quando este programa se inicia se tiver um ficheiro XML guardado localmente tenta imediatamente o seu envio.

Para a eventualidade de os websites estarem offline fizemos uma cópia dos mesmos localmente através da Extensão para Firefox *Scrapbook*.

Para validação do XML criado adicionámos o ficheiro *smartphones.xsd*. Usámos o site <http://www.freeformatter.com/xsd-generator.html> para gerar o xsd correspondente ao xml criado.

## HTML Summary Creator

A aplicação HTML Summary Creator (*MainHtmlCreator.java*) recebe o xml e gera uma página HTML (*full\_smartphones\_list.html*).

Para gerar a página HTML primeiro valida o xml recebido através do xsd. De seguida, caso a validação seja positiva, o xml em conjunto com o xsl (*xsl\_stylesheet.xsl*) geram o ficheiro.

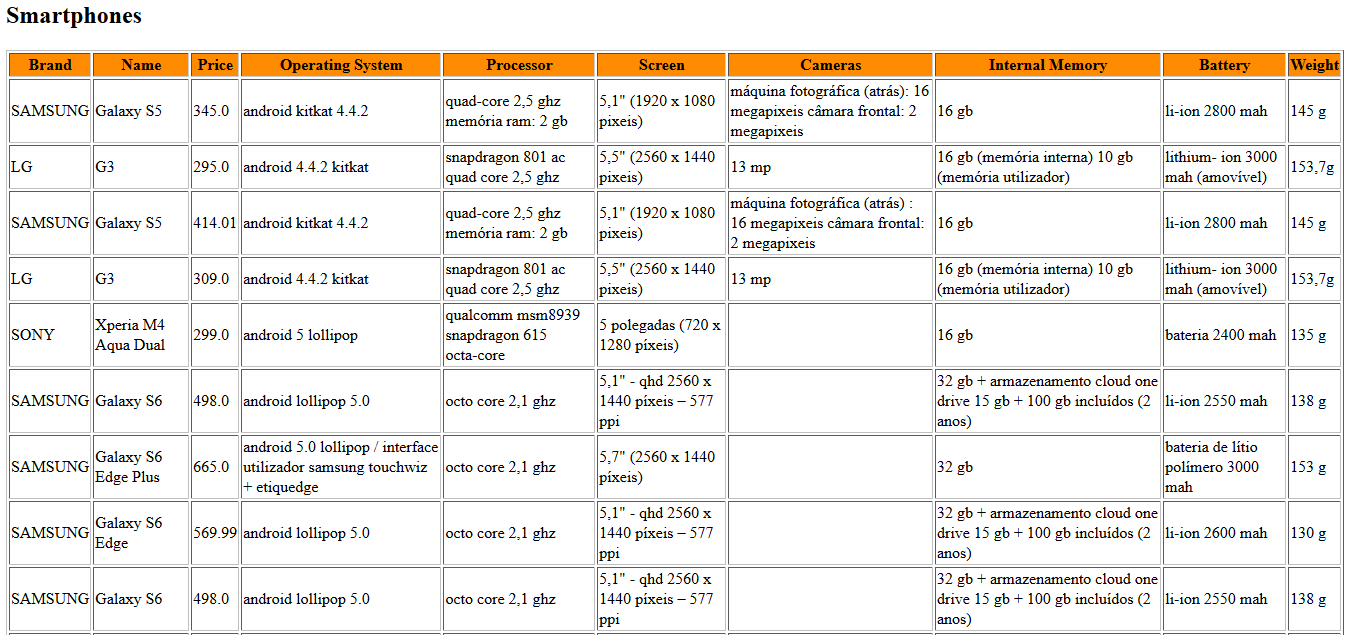


Figura - Página HTML gerada pelo HTML Summary Creator

## Price Keeper

MainKeeper:

FatherKeeper

TopicListener

QueueListener

Responder

ListOfThings

### Queue/Temporary Queue

## Cliente/Price Requester

A aplicação Requester (*MainRequester.java*) implementa uma pesquisa pelo nome de um smartphone com o objetivo de saber qual o seu preço.

Esta liga-se ao Price Keeper e envia as pesquisas para a Queue. Após enviar uma pesquisa esta fica à espera da resposta. A resposta será recebida através de uma Temporary Queue.

# Testes

Para avaliar o funcionamento das várias aplicações fizemos uma série de testes.

Webcrawler:

* Se a conexão com o website falhar utilizar a cópia local do site
* Se a conexão com o servidor Wildfly falhar criar o ficheiro xml e guardar lá a informação
* Se ao iniciar existir um ficheiro xml guardado em sistema enviar esse ficheiro em vez de fazer o parsing e de seguida apagá-lo.

HTML Summary Creator:

* Receber a mensagem (xml)
* Validar/Não validar o xml através do schema (xsd)
* Criar o ficheiro HTML com as informações recebidas no xml.

Price Keeper:

* Fazer as threads

Cliente/Requester:

* Fazer pesquisa por um termo
* Fazer pesquisa por mais do que um termo
* Receber o preço correto em relação ao smartphone pesquisado

# Conclusão

Ao realizar este projecto ficámos com conhecimentos sólidos sobre a edição e manipulação de documentos xml. Também nos deu a oportunidade para explorar o serviço de comunicação de mensagens Wildfly.