

**ucBusca: Motor de pesquisa de páginas Web**

**Projeto de Sistemas Distribuídos**

David Jesus Vaz Cortesão Silva – 2008109004

Rui Mário Coupina Reis – 2013134606

Arquitetura de Software

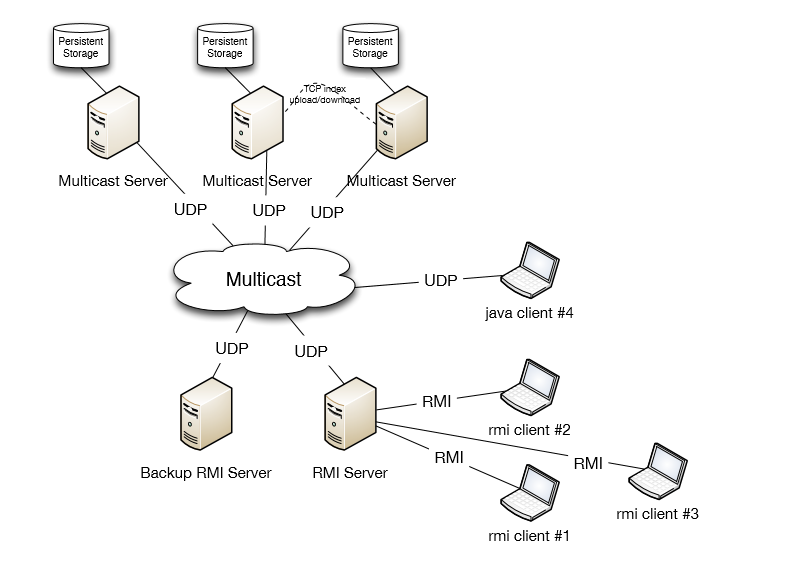
A arquitetura é composta por múltiplos clientes RMI, 2 servidores RMI ligados a uma rede Multicast com vários servidores Multicast.

O software está dividido em três componentes:

Client RMI, aqui estão implementadas todas as funções menu com que o utilizador vai interagir, e é este componente o responsável por garantir conexão com o server RMI (caso esteja disponível).

Server RMI, responsável por comunicar com a rede Multicast, através da qual, são enviadas mensagens(PROTOCOLO) para os servidores Multicast. É também responsável por enviar notificações aos clientes RMI, mantendo informações sobre os utilizadores online.

Servidor Multicast, é aqui que estão implementadas todas as funcionalidades necessárias para o correto desempenho da aplicação.



Funcionamento Servidor RMI

Os servidores RMI (principal e backup) funcionam de forma idêntica, sendo inicializados no registo na porta 7000 e 7001 respetivamente, através dessa interface o cliente vai fazer chamadas às suas funções que vão retornar na maior parte dos casos o resultado da operação (success / fail).

É o servidor RMI que envia notificações ao cliente, através da informação guardada (HashMap<username , ClientInterface>) quando um utilizador inicia sessão (função *subscribe*();), quando é necessário enviar uma notificação (ex: novo admin), o HashMap é percorrido e é enviada a notificação a todos os clientes *online.*

Os Servidores estão ligado à rede multicast na porta 4321 e ip 224.3.2.1;

Funcionamento Servidor Multicast

Os servidores Multicast vão todos dar join à rede Multicast 224.3.2.1 porta 4321 e através duma thread vão ficar à espera de mensagens (PROTOCOLO) do servidor RMI e passar essa mensagem para um HashMap depois através da identificação do tipo da mensagem(agora um HashMap) é identificada a função a ser executada e enviado o retorno dessa mensagem para o servidor RMI que a vai transmitir ao cliente.

Testes de Software

|  |  |  |
| --- | --- | --- |
| REQUISITO FUNCIONAL | PROTOCOLO | PASS/FAIL |
| Registar novo utilizador | Type | registar | PASS |
| Acesso protegido com password | Type | login | PASS |
| Indexar novo URL (admin) | Type |addUrl | PASS |
| Indexar iterativamente ou recursivamente URLs | Type |addUrlRec | PASS |
| Pesquisar páginas que contenham conjunto de palavras | Type |checkWords | PASS |
| Resultados ordenados por número de ligações para cada página | Type |tabelaLigacoes | PASS |
| Consultar lista de páginas com ligações para uma página específica | Type |ligacoesALinks | PASS |
| Consultar lista de pesquisas feitas pelo próprio utilizador | Type |mostraConsultas | PASS |
| Dar privilégios de admin a um utilizador | Type |admin\_give | PASS |
| Página de administração atualizada em tempo real | Type |admin\_check | PASS |
| Notificação imediata de privilégios de administrador | (enviada pelo *RMI server*) | PASS |
| Entrega posterior de notificações (offline users) | (enviada pelo *RMI server*) | PASS |

Tratamento de Exceções

|  |  |
| --- | --- |
| EXCEÇÃO | PASS/FAIL |
| Avaria de um servidor RMI não tem efeito visível nos clientes | PASS |
| Servidor RMI secundário testa e substitui o primário em caso de avaria longa | PASS |
| Avarias temporárias (<30s) dos servidores multicast são invisíveis para os clientes | FAIL |
| Pedidos são garantidamente processados por N>=1 servidores multicast | PASS |
| Pedidos de indexação são respondidos apenas por um servidor multicast | FAIL |

FAILOVER

|  |  |
| --- | --- |
| FAILOVER | PASS/FAIL |
| O serviço funciona se houver pelo menos um servidor multicast disponível | PASS |
| Os servidores multicast recuperam de disco o seu estado se avariarem | PASS |
| Cada servidor multicast replica a sua parte do índice por outros servidores (TCP) | FAIL |
| Cada servidor distribui URLs para serem indexados por outros servidores | FAIL |
| O servidor RMI original, quando recupera, torna-se secundário | PASS |

Distribuição de tarefas

Quanto à divisão de tarefas decidimos distribuir o mesmo número de tarefas pelos dois elementos, não só por acharmos que se trata da opção mais justa, mas também pelo facto de com este método podermos ter a oportunidade de ficar com um bom conhecimento em relação a todos os aspetos da aplicação.