



UNIVERSIDADE D  
COIMBRA



ucBusca:

Motor de pesquisa de páginas Web

Rodrigo Ferreira de Almeida Martins– 2017255325

Vitalina Holubenko – 2017255810

## 1. Protocolo

### Login:

- id - "id do usuário" ; type - login ; username - "nome do usuário" ; password - "password do usuário"

### Output:

- "id | " + id + " ; type | status ; logged | on ; msg | Welcome to ucBusca Admin" (para admins)

- "id | " + id + " ; type | status ; logged | on ; msg | Welcome to ucBusca User"; (utilizadores)

- "id | 000 ; type | status ; logged | off ; msg | No user found or password is incorrect"(falha no login)

### Registo:

- id - "id do usuário" ; type - register ; username - "nome do usuário" ; password - "password do usuário"

### Output:

- "type | status ; registered | negative ; msg | Failed to register, choose different username" (falha no registo)

- "type | status ; registered | affirmative ; msg | Registered" (registo foi bem sucedido)

### Inserção de um url para indexação recursiva:

- id - "id do usuário" ; type - urlInsert ; url - "url a indexar"

### Output:

- "type | status ; urlInsertion | successful ; msg | Indexation In Progress"

### Pesquisa de palavras:

- id - "id do usuário" ; type - termSearch ; terms - "termo 1, termo 2, termo 3,..."

### Output:

- Listagem de 10 (ou menos) resultados com o nome do site e uma pequena citação.

### Consultar lista de páginas com ligações para um página específica:

- id - "id do usuário" ; type - byReference ; url - "url a pesquisar"

### Output:

- Listagem de sites que mencionam o url indicado.

### Consular histórico de pesquisas:

- id - "id do usuário" ; type - userSearches

### Output:

- Listagem dos termos que o utilizador pesquisou.

Permitir acesso administrativo a um usuário:

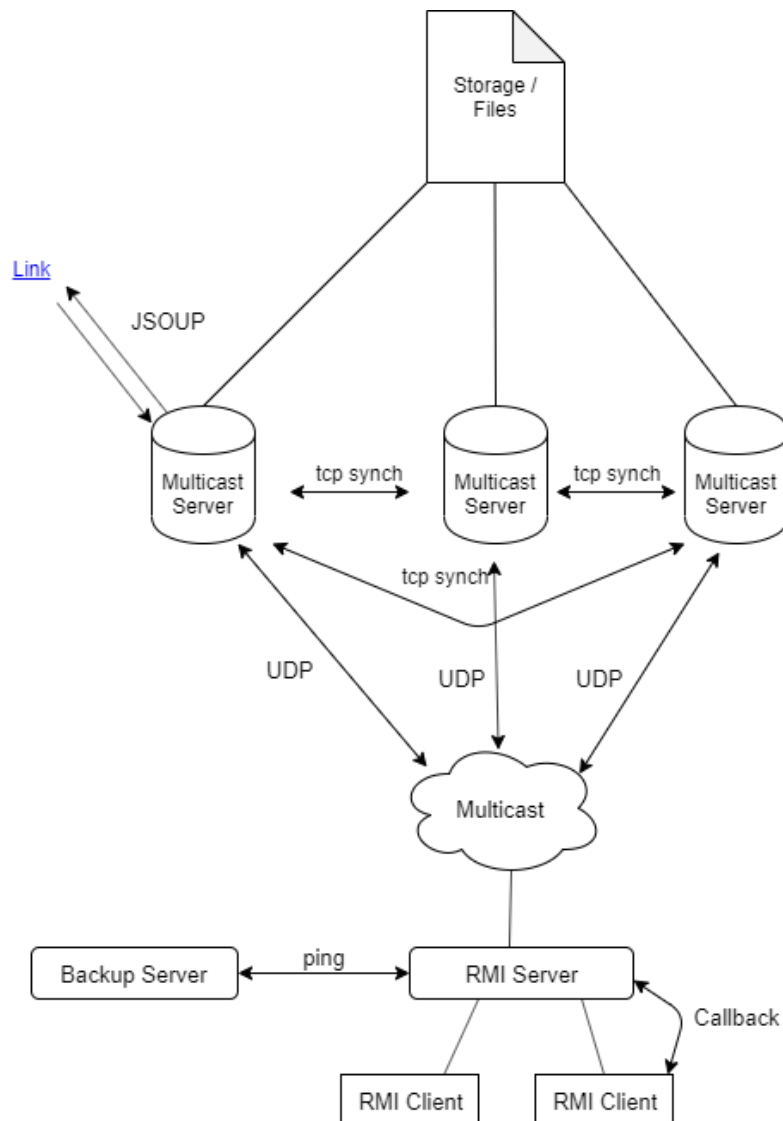
- id - "id do usuário" ; type - adminAccess ; user - "nome do usuário"

**Output:**

- "No user found" (No caso de não haver um utilizador com o nome indicado)"
- "Permission granted"

## Arquitetura do software

---



Para a integração da primeira meta 1 é implementado um Servidor Multicast com objetivo de processar pedidos que necessitem de acesso à informação armazenada nos servidores multicast. Existem vários Clientes RMI que comunicam com os Servidores Multicast via UDP através de um Servidor RMI, este processa os pedidos de todos os clientes RMI e envia para o servidor Multicast, que é escolhido pelo servidor RMI, o pedido através de um protocolo já pré-definido e explicado anteriormente. Os pedidos recebidos pelos servers serão tratados e consequentemente será devolvido um output desejado para o cliente RMI.

## Distribuição de tarefas pelos elementos do grupo

---

RMI Server e RMI cliente – Rodrigo Martins

Multicast – Vitalina Holubenko

## Teste de Software

---

### Requisitos Funcionais

Registar novo utilizador	PASS
Acesso protegido com password (exceto pesquisas)	PASS
Indexar novo URL introduzido por administrador	PASS
Indexar iterativamente ou recursivamente todos os URLs encontrados	PASS
Pesquisar páginas que contenham um conjunto de palavras	PASS
Resultados ordenados por número de ligações para cada página	PASS
Consultar lista de páginas com ligações para uma página específica	PASS
Consultar lista de pesquisas feitas pelo próprio utilizador	PASS
Dar privilégios de administrador a um utilizador	PASS
Página de administração atualizada em tempo real	FAIL
Notificação imediata de privilégios de administrador (online users)	PASS
Entrega posterior de notificações (offline users)	PASS

### Tratamento de Exceções

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
Em caso de avaria longa os clientes RMI ligam ao secundário (sessão mantida)	PASS
Avárias temporárias (<30s) dos servidores multicast são invisíveis para clientes	PASS
Pedidos são garantidamente processados por $N \geq 1$ servidores multicast	PASS
Pedidos de indexação são respondidos apenas por um servidor multicast	PASS

### Failover

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
EXTRA : STONITH	PASS

