Departamento de Engenharia Informática

1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

ucBusca: Motor de pesquisa

Meta 2

Sistemas Distribuídos

2019-2020

Inês Silva Lamelas Van Brabant Moreira - 2004101168

Índice

1. Introdução

No seguimento da meta anterior, nesta meta pretende simular-se um motor de busca através do desenvolvimento de uma interface web para a aplicação da meta 1. Desta forma, será possível aceder à plataforma através do web browser num dispositivo. O projeto implica uma integração do RMI (desenvolvido na meta 1) com um servidor web e os utilizadores web acedem à mesma informação que os utilizadores na aplicação desktop, de modo a haver interoperabilidade.

2. Funcionalidades desenvolvidas

Registo
Os clientes podem emitir um pedido de criação de conta, bastando fornecer um username e uma password.

Login
Os clientes podem emitir um pedido de login, sendo necessários o username e a password. Se não tiverem feito o registo anteriormente, ou se já estiverem registados, ao tentarem submeter o pedido, irão manter-se na mesma view. Todas as funcionalidades, exceto a pesquisa, requerem o login efetuado.

Indexar novo URL
Um utilizador registado, que seja administrador e que tenha feito login, poderá inserir manualmente um URL.

Indexar recursivamente todos os URLs encontrados
Ao indexar o primeiro URL, serão indexados todos os links associados a esse URL. Para simplificar, o número de links a visitar foi limitado a 20.

Pesquisar páginas que contenham um conjunto de termos
Qualquer utilizador pode pesquisar um conjunto de termos. No resultado, aparecerão, o título da página, a descrição, o URL, o número de conexões e, para os utilizadores com login efetuado, as conexões para este URL.

Resultados de pesquisa ordenados por importância
Visualizável nos resultados da pesquisa.

Consultar lista de páginas com ligação para uma página específica
Visualizável nos resultados da pesquisa, apenas para utilizadores com login efetuado.

Consultar pesquisas realizadas anteriormente
Qualquer utilizador com login efetuado pode visualizar o seu histórico de pesquisas.

Dar privilégios de administrador a um utilizador
Um utilizador com privilégio de administrador pode conceder esse privilégio a outro utilizador. Caso o utilizador esteja presente na lista de utilizadores registados, será possível conceder-lhe esse privilégio e o administrador receberá uma notificação afirmativa.

Entrega posterior de notificações a utilizadores offline
Caso um cliente esteja offline, a mensagem será guardada numa lista de mensagens onde consta o ID do cliente e o texto com a notificação. Assim, quando o cliente fizer login, receberá de imediato as notificações.

Logout
Quando um utilizador digita “Logout”, é encaminhado para a página inicial (sem login efetuado).

Língua original dos resultados de pesquisa
Para cada resultado de pesquisa, aparece o idioma em que a página está escrita.

3. Arquitetura

Neste projeto, o servidor web comunica com o servidor multicast através da ligação por RMI. A aplicação web corre num servidor HTTP (Apache Tomcat) e comporta-se como um cliente RMI. Os clientes ligam-se ao servidor web através de navegadores para efetuarem pedidos de páginas através de HTTP.
Na elaboração do projeto foi utilizada a framework struts2.

3.1. Model

Cada uma das seguintes classes da pasta **model** permite encapsular os objetos e propriedades necessários num só objeto. Com o auxílio de getters e setters, serão definidos nas respectivas classes da pasta **actions** os valores desejados para efetuar os pedidos dos clientes. É em cada uma destas classes que é feito o lookup que permite invocar os métodos RMI anteriormente criados.

RegistryBean.java
Para efetuar o registo. Guarda a confirmação de que aquele cliente está registado.

LoginBean.java
Para efetuar o login. Guarda o username, a password, altera o estado de login para True, caso o cliente já esteja registado e esteja offline e guarda a informação sobre se é admin ou não.

LogoutBean.java
Para efetuar o logout. Altera o estado de login para False.

SearchBean.java
Gere os pedidos de pesquisa de clientes online e offline. Tem também os métodos para verificação do idioma de cada resultado de pesquisa e para tradução de páginas (obtenção do dicionário de idiomas do Yandex, procura pela língua portuguesa e tradução de texto para português).

MySearchesBean.java
Permite aceder ao histórico de pesquisas de clientes com login efetuado.

AdminPrivBean.java
Permite ao administrador atribuir privilégios de administração a outros utilizadores.

URLBean.java
Permite ao administrador indexar um novo URL.

NotificationsBean.java
Se o cliente que receber privilégios de administração estiver offline, será guardada a mensagem na sua página das notificações.

3.2. Controller

RegistryAction.java
Define na classe RegistryBean, através de setters, o parâmetro de registo (se o utilizador está registado, ou não). Depois irá chamar a função do RegistryBean, através do método execute(), encarregue de fazer o registo do utilizador. O resultado da chamada remota de RMI é interpretada nesta classe e só em caso de sucesso o utilizador conseguirá ser encaminhado para a página de Login.

LoginAction.java
Define através de setters, os parâmetros do username e password; chama o método do LoginBean, através do método execute(), encarregue de fazer o login do utilizador; após receber o resultado e caso este seja de sucesso, define através de um getter, o parâmetro de admin e coloca o utilizador na sessão, bem como o estado de login confirmado; isto fará com que o utilizador seja encaminhado para a página principal.

LogoutAction.java
Define o parâmetro de login, no LogoutBean. Irá chamar a função do LogoutBean, através do método execute(), para ir buscar à sessão o nome e a password do utilizador, que depois irá definir através de setters; depois irá alterar o estado de login, na sessão, para False. Para este caso, apenas foi definido o caso de sucesso e o utilizador será encaminhado para a página inicial (comum a todos os utilizadores).

SearchAction.java
Define através de setters, o parâmetro da pesquisa e se o utilizador é anónimo ou registado. Irá chamar a função do SearchBean, através do método execute e, dependendo do resultado da chamada remota RMI (se o campo da pesquisa está, ou não, vazio), irá verificar se o utilizador está online ou é anónimo (para permitir os dois tipos de pesquisa). Em ambos os casos (sucesso, ou não), será encaminhado para a página de pesquisa, onde o resultado dependerá daquilo que foi enviado.
No método execute é também obtido, através de uma chamada remota RMI, o idioma de cada página dos resultados de uma pesquisa.

MySearchesAction.java
Define através de setter, o parâmetro do username, obtido da sessão; depois chama o método do MySearchesBean, para obter a lista de pesquisas efetuadas pelo utilizador; consoante o resultado, guarda as pesquisas numa variável através de um getter. O utilizador será encaminhado para a página do histórico de pesquisas, que poderá conter, ou não, resultados.

AdminPrivAction.java
Define os parâmetros do username do administrador e do outro utilizador através de setters; depois chama a função do AdminPrivBean para obter o resultado da atribuição de privilégios admin. em caso de sucesso, o administrador voltará para a página principal.

URLAction.java
Define o parâmetro do URL a indexar e do username do administrador, na classe URLBean, através dum setter; através do método execute(), chama a função do URLBean encarregue de verificar se o URL foi indexado. Em caso de sucesso, o administrador será encaminhado para a página principal.

NotificationsAction.java
Em caso de sucesso, no método execute(), define o parâmetro username através dum setter e o parâmetro da mensagem recebida, através dum getter, na classe NotificationsBean. Em caso de sucesso, encaminhará o utilizador para a página onde será exibida a mensagem.
TranslatedAction.java

Esta classe invoca os métodos da classe SearchBean, para tradução de páginas para portugês. Depois, encaminharia o utilizador para uma nova página com os resultados traduzidos.

3.3. JSP (WebContent)

Aqui encontra-se o código HTML, ou seja, as views que correspondem à interface.

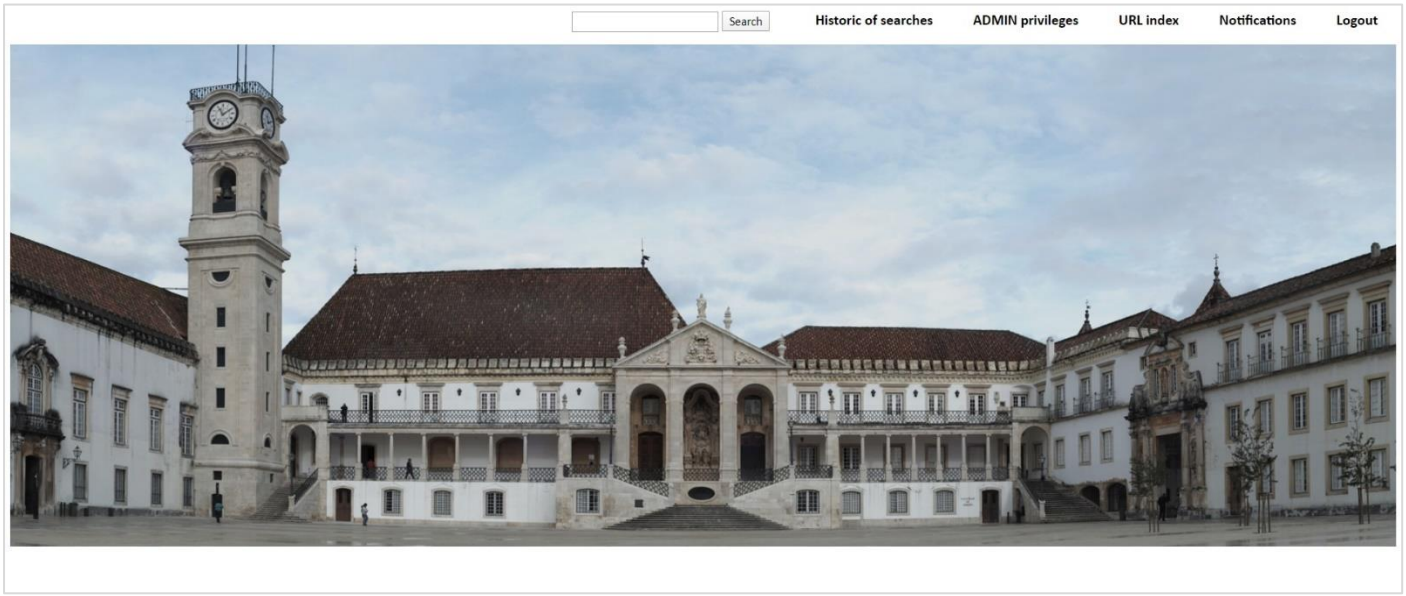
index1.jsp
Página inicial, visível para qualquer utilizador (com ou sem login efetuado). Aqui é possível realizar uma pesquisa, ou aceder às funcionalidades de registo e login.

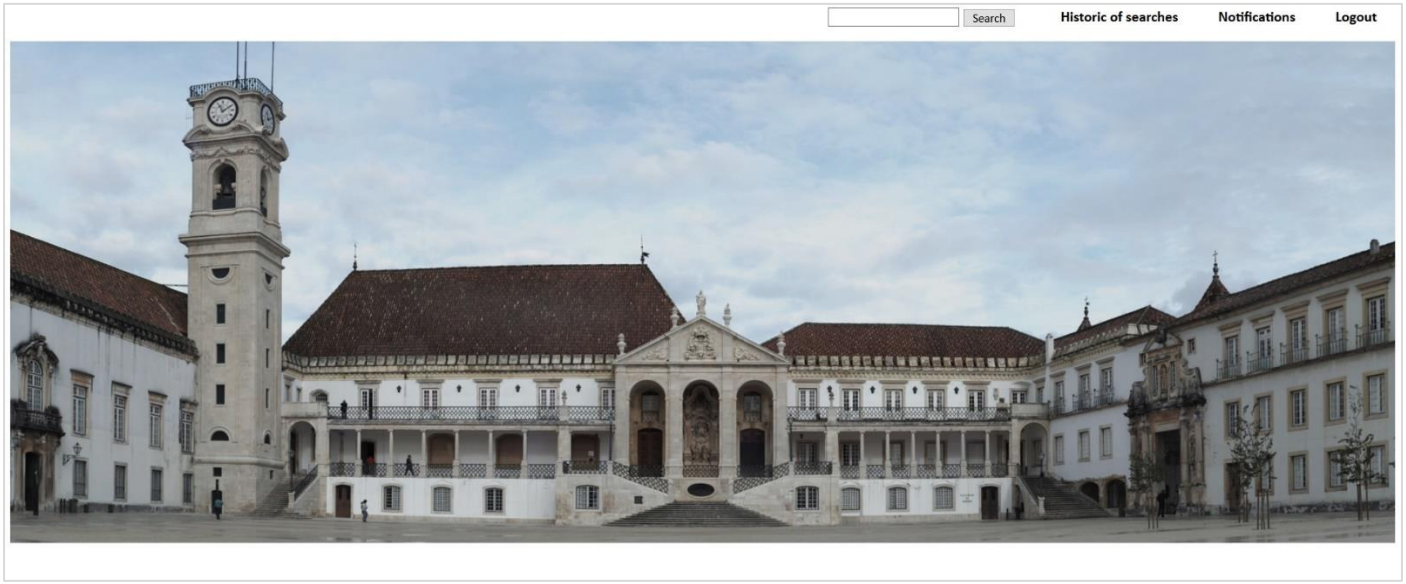
registry.jsp
Ao clicar em *Registry*, no **index1.jsp,** o utilizador será encaminhado para a página abaixo.

login.jsp
Ao clicar em *Login*, no **index1.jsp**, ou no botão *Submit*, do **registry.jsp**, o utilizador será encaminhado para a página abaixo. Caso o utilizador ainda não esteja registado ou já tenha login efetuado, ao fazer *Submit* manter-se-á na mesma página.

index2.jsp

Ao fazer submit, no **login.jsp**, o utilizador será encaminhado para esta página, onde poderão aparecer algumas das funcionalidades, dependendo se é administrador ou não.



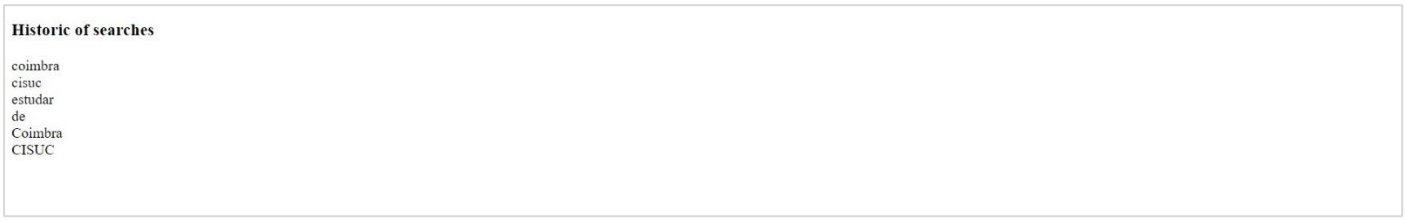


URLindex.jsp

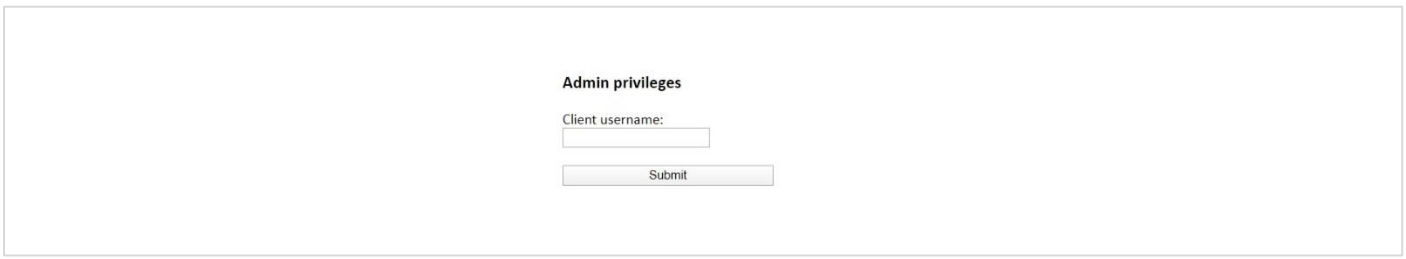
Ao clicar em *URL index,* o utilizador será encaminhado para esta página, onde poderá inserir o URL a indexar. Ao fazer *Submit*, voltará à página **index2.jsp**.

SearchResults.jsp
Ao efetuar uma pesquisa (*Submit*), o utilizador será encaminhado para esta página de resultados.

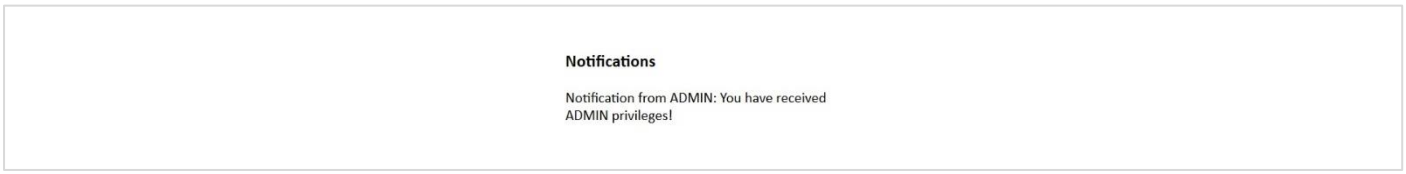
MySearches.jsp
Ao clicar em *Historic of searches*, o utilizador será encaminhado para esta página, onde será apresentado o seu histórico de pesquisas.



AdminPriv.jsp
ao clicar em *ADMIN privileges*, o administrador será encaminhado para esta página, onde poderá colocar o nome do utilizador a quem deseja atribuir privilégios de administração. Ao fazer *Submit,* caso o utilizador não exista, manter-se-á na mesma página. Caso contrário, voltará a **index2.jsp**.



Notifications.jsp
Se um utilizador receber notificação de privilégios de administração enquanto estiver offline, quando fizer login, já entrará na página como administrador e, se clicar em *Notifications,* irá encontrar esta mensagem.



Translated.jsp
Ao clicar no botão *Translated*, na página SearchResults, o utilizador seria encaminhado para esta página, onde apareceriam os resultados das pesquisas traduzidos para português.

4. Struts2

Em struts.xml é definido o que fazer em cada ação das páginas .jsp, isto é, consoante a ação escolhida pelo utilizador, irá invocar diferentes métodos nas actions. Depois, de acordo com o retorno desse método, é declarado para que página .jsp o utilizador deve ser direcionado. Desta forma, é feita a junção das três componentes da arquitetura. No mapeamento de Struts, neste ficheiro é declarado que classe responde a que ação, que método tem de ser executado e que view deve ser mostrado.

Nos ficheiros .jsp são definidas actions, com o auxílio de tags html. Deste modo, quando o utilizador submeter uma determinada action com o auxílio de forms, nas struts.xml são chamadas as funções apropriadas. De seguida, com o uso de funções setters, nas classes Bean, são definidos os valores dos atributos necessários à operação em questão. Finalmente, nesses mesmos ficheiros, são invocados remotamente os métodos RMI necessários para realizar cada operação. Se o processamento do pedido for bem efetuado, o método correspondente nas classes Action irá definir na session do utilizador em questão, o parâmetro resposta.

Nesta arquitetura, nenhuma das variáveis persiste assim que a acção for executada, logo é necessário o uso de sessions. Deste modo, há variáveis que são sempre conhecidas por todas as classes de Actions e seus métodos, uma vez que as classes implementam SessionAware.

5. Serviços Rest

Foi usado o Yandex para obter o idioma de cada resultado de pesquisa. Esta verificação foi feita nas classes SearchBean e SearchAction (ver pontos 3.1 e 3.2). Também foram implementados os métodos para tradução das páginas para português, na classe SearchBean.

6. Testes de software

Descrição dos testes	Resultado
Requisitos funcionais	
Registry	Utilizador novo – registo com sucesso; encaminhado para a página Login.jsp.
	Utilizador existente – permanecerá no ecrã Registry.jsp.
Login	Utilizador offline – login com sucesso; encaminhado para a página principal (index2.jsp)

	Utilizador não registado – mantém-se na página Login.jsp
	Utilizador online – manter-se-á na página Login.jsp
ADMIN Privileges	Utilizador existente – atribuição efetuada com sucesso; administrador volta para a página principal.
	Utilizador inexistente – administrador mantém-se na página AdminPriv.jsp
	Utilizador offline – quando se conectar, já entrará na página como ADMIN e verá a notificação de privilégios de ADMIN ao aceder à página Notifications.jsp
URL index	Depois de indexar o URL, o administrador volta para a página principal (index2.jsp)
Search	Pesquisa encontrada – encaminhado para a página SearchResults.jsp, onde aparecerão os resultados.
	Pesquisa não encontrada – encaminhado para a página SearchResults.jsp, onde não aparecerão resultados.
	Para cada resultado, aparece o idioma da página.
Historic of searches	Efetuou pesquisas – aparece a lista de todas as pesquisas efetuadas em SearchResults.jsp
	Não efetuou pesquisas – aparece a lista sem resultados
Notifications	Tem notificações – será encaminhado para a página Notifications.jsp
	Não tem notificações – mantém-se na página principal (index2.jsp)
Logout	É encaminhado para a página inicial index1.jsp