



e-Voting

Voto Eletrónico na UC

Meta 2

Relatório realizado no âmbito da unidade curricular Sistemas Distribuídos, inserida no programa curricular do curso de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

2018284515	Ana Rita Rodrigues	analr@student.dei.uc.pt	PL2
2018233092	Dylan Perdigão	dgp@student.dei.uc.pt	PL2

Coimbra

2021

Introdução	3
Integração com Struts2	4
Controller	4
Model	4
View	5
Integração com WebSockets	6
Ligação ao WebSocket	6
Envio de mensagens	6
Receção de mensagens	6
Integração com APIs REST	7
Testes de Software	8
Conclusão	10

Introdução

No âmbito da unidade curricular Sistemas Distribuídos, foi-nos proposto desenvolver um sistema de voto eletrónico para estudantes, docentes e funcionários da Universidade de Coimbra cujo objetivo é ter um sistema operacional distribuído em várias máquinas que consigam comunicar entre elas. Cada máquina tem o seu propósito para efetuar diversas tarefas no processo de voto eletrónico, nomeadamente o registo das informações na base de dados, criação de eleições com respetivas listas, contas de eleitores, identificação dos eleitores nas mesas de voto e um terminal que permita proceder ao sufrágio. O armazenamento de todos os dados envolvidos nestes processos é feito através de uma base de dados *SQLite*.

Acabada a primeira meta, foi preciso adaptar o nosso código para a integração de *JSP* (*Java Server Page*) de modo a colocar o nosso projeto a correr na *web* através de um servidor *Apache Tomcat*. Deste modo o utilizador poderá votar de forma remota usando as suas credenciais registadas por um administrador ou usando a sua conta *Facebook* depois de previamente ter associado a sua conta *Facebook* à conta no sistema de votos.

O projeto é constituído pelo código fonte, pelos executáveis (*.jar* e *.war*), pelo *Javadoc* que explica detalhadamente as classes, interfaces e métodos implementados e de um ficheiro (*.txt*) que contém as instruções de execução do sistema.

Integração com Struts2

Na nossa implementação, decidimos recorrer ao uso da *framework Struts2*. O ficheiro central da framework reside no `struts.xml`, onde é efetuado todo o mapeamento de ações e redirecionamentos para uma determinada página *JSP*. O uso do *Struts2* é baseado na arquitetura *MVC* em que recorremos ao uso de *models*, *views* e *controllers*.

Controller

O controller é usado para fins de mapeamento de uma ação na qual por exemplo é preciso definir um *name* que age como identificador e a class da ação a que se refere.

Conforme as requests efetuadas no *browser*, o resultado da ação pode ser diferente, como podemos observar no seguinte excerto de código:

```
<action name="login" class="webServer.action.LoginAction" method="execute">
  <result name="isAdmin">/Login_menuAdministrator.jsp</result>
  <result name="success">/chooseElection.jsp</result>
  <result name="login">/Login.jsp</result>
</action>
```

Caso a *action* devolve `isAdmin`, a página redireciona para o menu de administrador, caso devolva `success`, é redirecionado para a página de escolha de eleição em que quer votar, finalmente caso a *action* devolve `login`, volta para a página de *login*.

O identificador é chamado no momento em que a *action* de uma *request* use o mesmo nome.

Model

O *model* é composto por uma *bean* que faz a ligação entre a *view* e o servidor *RMI*. Através de *getters* é possível enviar as informações solicitadas pela *view*.

View

A *view* é composta principalmente por código *JSP* sendo que é baseado em *HTML*, o que nos permitiu usar ambos.

Através das *beans* foi-nos possível adicionar à nossa página conteúdo dinâmico. Na *tag* desejada usa-se por exemplo:

```
${HeyBean.infoVotes}
```

Neste caso o *HeyBean* é o nome da *class Java* como o nome indica, da *bean*.

infoVote é o nome do getter sem o prefixo “*get*” e com letra minúscula no início da palavra.

Com as *tags JSP* é possível percorrer elementos de um *array* recorrendo a ciclos “*forEach*” e também temos a possibilidade de usar controlo de fluxos com branches “*choose-when-otherwise*”. Com “*form*” e várias categorias de “*input*” conseguimos guardar informações nos *beans* através de *setters* para serem tratados e enviados por exemplo ao servidor RMI.

Integração com WebSockets

A integração com Websockets foi fundamental para enviarmos conteúdo html para a página do browser para o administrador receber, em tempo real, informações sobre o estado das mesas de voto (e respetivos terminais de voto), contagem dos votos com a distinção do trabalho da pessoa (Estudante, Docente, Funcionário).

Ligação ao WebSocket

Através de código em *JavaScript*, foi-nos possível, criar um WebSocket no endereço:

```
ws://<host>:8080/webserver/webServer/ws
```

Sendo `webserver` o nome do *artefacto* “`.war`” (*Web archive*) e `webServer/ws` a diretoria contendo a *class* Java do *WebSocket*.

O funcionamento é relativamente simples. Ao carregar a página, é efetuada a ligação no endereço mencionado anteriormente, são ativados listeners (`onOpen`, `onMessage`, `onError`, `onClose`) e é adicionado o WebSocket num Set contendo as ligações todas.

Envio de mensagens

Os *Callbacks* chamam um método estático no *WebSocket.java* que irão tentar enviar a mensagem (normalmente texto em código *HTML*) em parâmetro para todas as sessões dos *WebSockets* que estão guardados no *Set*. Em caso de erro, é terminada a sessão e removido o *WebSocket* do *Set*.

Receção de mensagens

Voltando ao *browser*, o listener é ativado ao receber uma mensagem, sendo ela introduzida numa `<div>` *HTML*.

Integração com APIs REST

A integração com a *API Rest* permitiu-nos fazer com que um determinado utilizador pudesse associar a sua conta de *Facebook* à sua conta na aplicação (para poder votar) e consequentemente poder efetuar o *login* a partir do *Facebook*.

Além disso, aquando do voto, um utilizador pode ainda partilhar no seu “*feed*” do facebook informações acerca da eleição que acabou de votar. No entanto dada a impossibilidade de se realizar o “*post*” de forma automática, decidimos recorrer a um botão no *JSP* que faz o redirecionamento para a página de *Facebook* do utilizador para que este possa finalizar o processo de partilha.

O conteúdo na página partilhada via *Facebook* depende do estado da eleição. Se a eleição estiver a decorrer apresenta detalhes acerca desta como o título, tipo, datas de início e fim e respetivas listas a ela associadas. Se a eleição estiver terminada apresenta os resultados finais (número de votos em cada lista, número de votos brancos, números de votos nulos e respetivas percentagens).

Para **associar** e fazer **login** com a conta do facebook utilizámos uma *action* que comunicou com a *API* do Facebook a fazer o pedido para associar / fazer *login* de um determinado utilizador. Se o utilizador ainda não tiver autorizado o acesso da aplicação ao facebook será redirecionado para uma página onde pode fornecer essa autorização. Uma vez fornecida, será redirecionado de volta para o *website*. Depois, a *API* devolve um *token* referente ao utilizador em específico, que o irá representar univocamente para *logins* futuros.

Para impedir que a mesma pessoa se associe a várias contas distintas no *website*, colocámos uma *constraint* na base de dados que torna o *token* único.

Testes de Software

De notar que, para além de todos os testes realizados que constam na tabela, todos os botões da aplicação foram testados de modo a executarem o pretendido.

Sempre que o *RMI* vai abaixo a aplicação consegue voltar a reconectar-se a ele.

Para uma fácil utilização todos os campos têm em *placeholder* exemplos do conteúdo que se deve colocar e para prevenir problemas todos os inputs estão protegidos.

Funcionalidade		Testes realizados	Pass / Fail
Menus Administrador	Registar Utilizador	Não é possível registar 2 utilizadores com o mesmo número de cartão de cidadão	
		O número de telemóvel tem 9 dígitos e começam por 91, 92, 93 ou 96	
		O utilizador só pode escolher o departamento em que frequenta dentro de 11 opções possíveis, que lhe serão apresentadas	
		O utilizador só pode escolher o cargo que ocupa dentro de 3 opções possíveis que lhe são apresentadas	
		Para efeitos de segurança, aquando do registo de um utilizador, será aplicada uma hash à concatenação do número de cidadão com a password	
		A data do cartão de cidadão tem que ser superior à data atual	
	Criar eleição	Data de fim é sempre superior à data de início	
		Não permite inserir um título vazio	
	Gerir eleição	Apenas se pode adicionar candidatos caso a eleição ainda não esteja a decorrer	
		Apenas se podem adicionar candidatos cujo tipo de lista corresponda ao trabalho da pessoa	
		Apenas se podem adicionar candidatos cujo departamento onde a eleição é restringida seja onde ele trabalha	
		Apenas se pode editar eleições que ainda não estejam a decorrer	
	Gerir mesas de voto	Não podem ser adicionadas mesas de voto a eleições que estão restringidas a um único departamento	
		Só é possível inserir e remover mesas de voto a eleições que ainda não começaram	
		Não é possível inserir mesas de voto a eleições que estejam a decorrer	

	Registo de votos (Integração com a primeira meta)	Mostra o local, data, hora, eleição em que uma pessoa votou já votou	
	Resultados das eleições (Integração com a primeira meta)	Mostra os resultados das eleições acabadas na data e hora marcada	
	Estado das Mesas de Voto (Integração com a primeira meta)	As mesas ligam e desligam e é mostrado ao utilizador na consola de administração	
	Contagem dos votos (Integração com a primeira meta)	É contabilizado o voto com a respetiva mesa de voto e trabalho da pessoa	
	Utilizadores online (Integração com a primeira meta)	São mostrados os utilizadores online no sistema de voto quer na mesa de voto associada a um departamento quer no browser	
Navegação pelos menus		Só consegue navegar se o RMI estiver ligado	
Login		Conseguir fazer login depois RMI ir a baixo e voltar a ser ligado	
Logout		Enquanto um determinado utilizador não proceder ao logout a sessão fica sempre iniciada e navega sempre nas páginas a que tem acesso	
Login com Facebook		Conseguir fazer login com o Facebook depois RMI ir a baixo e voltar a ser ligado	
Associar Facebook		Associar facebook se o RMI for abaixo e voltar a ser ligado	
Partilhar		Partilhar uma Eleição se o RMI for abaixo e voltar a ser ligado	
Consultar a Partilha		Caso a eleição ainda esteja a decorrer são apresentados detalhes acerca desta bem como as respetivas listas	
		Caso a eleição já tenha terminado são apresentados os resultados	
		Consultar uma partilha se o RMI for abaixo e voltar a ser ligado	
Votar		Por questões de confidencialidade, não se sabe em que lista votou cada eleitor	
		Votar se o RMI for abaixo e voltar a ser ligado	
		Apenas se pode votar em eleições a decorrer	
		Apenas estudantes votam em listas de estudantes, docentes em listas de docentes e funcionários em listas de funcionários	

Conclusão

Com base no que foi apresentado, e com o projeto concluído, é de realçar que foram atingidos os principais objetivos do funcionamento do nosso sistema distribuído.

Como foi referido anteriormente, a aplicação tem a finalidade de ser usada como ferramenta de voto eletrónico na comunidade estudantil da Universidade de Coimbra.

Posto isto, foi um projeto que nos proporcionou um conhecimento mais profundo no mundo da programação, mais especificamente da linguagem Java e com a framework *struts2*, das quais retirámos competências que certamente nos vão proporcionar um melhor futuro como Engenheiros Informáticos.