

Meta 2

eVoting

Voto Eletrónico na UC



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Sistemas Distribuídos

Departamento de Engenharia Informática

2020/2021 - 2º Semestre

Nome	Nº	Email	PL
Filipa Capela	2018297335	uc2018297335@student.uc.pt	PL4
Rodrigo Sobral	2018298209	uc2018298209@student.uc.pt	PL1

Índice

[Índice](#)

[Integração Struts2 - RMI](#)

[WebSockets](#)

[REST API - Facebook](#)

[Extras](#)

[Fase de Testagem](#)

Integração Struts2 - RMI

A associação aos servidores RMI foi feita com base em *Struts2*. Desta forma, e usando tecnologias associadas, tais como *HTML*, *CSS*, *XML*, *JSP* e *JAVA*, conseguimos construir uma interface web na qual, através de dois portais distintos, administradores e eleitores podem exercer as suas funções separadamente.

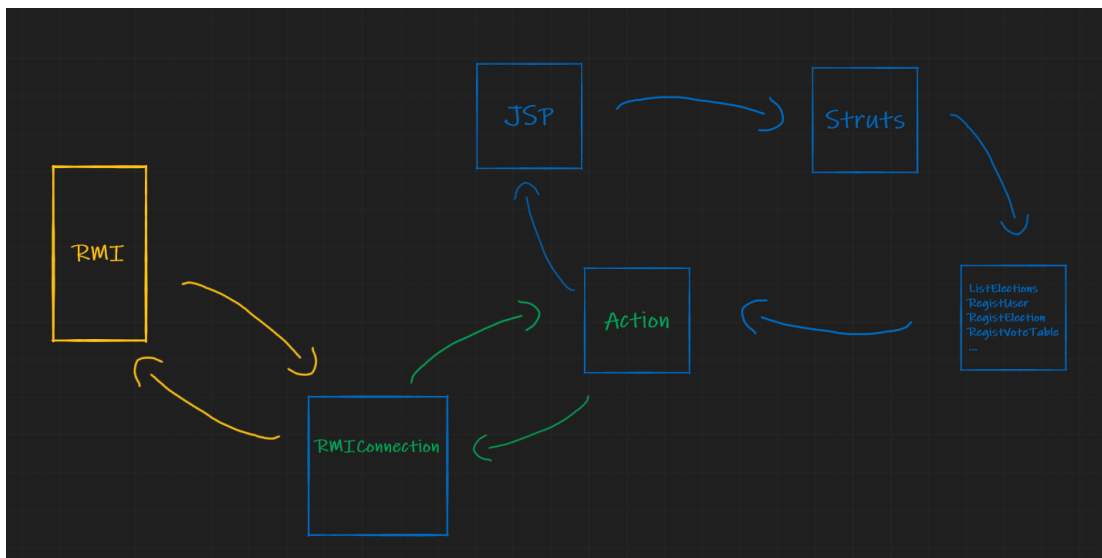
Para ter acesso a funcionalidades administrativas, basta adicionar manualmente */admin_menu* no *URL* raiz do *host*. A partir desse momento, o administrador pode criar eleitores, eleições, mesas de voto, fazer edições, remoções, consultas, entre outras funcionalidades.

Por outro lado, o que o eleitor se depara é com uma página de login, na qual tem de inserir o seu nome e *password* (definidos pelos administradores) para ter acesso a um painel no qual pode votar ou interagir com funcionalidades implementadas pela *REST API* e que serão abordadas brevemente.

Até o utilizador votar, tem de passar por uma filtragem, isto é:

- Primeiramente tem de indicar em qual mesa de voto pretende votar (estas mesas têm de ser previamente criadas e ativas, isto é, o *MultiCast Server* precisa de estar operacional).
- De seguida, são exibidas ao eleitor todas as eleições que decorrem na mesa de voto selecionada e que são direcionadas ao eleitor em causa, ou seja, um eleitor estudante não vai sequer ter acesso ou visibilidade às eleições de professores, portanto, nesta fase, o eleitor só tem de selecionar a eleição na qual quer votar e o seu número do cartão de cidadão de forma a se autenticar.
- Por fim, são listadas para o utilizador todas as candidaturas disponíveis e nesta fase o eleitor pode finalmente exercer o seu voto e ser reencaminhado de novo para o menu inicial.

Todas estas funcionalidades são implementadas segundo uma estrutura maioritariamente composta por 5 componentes: *jsp*s, *struts*, *actions* específicas, *action*, e *rmiconnection*.

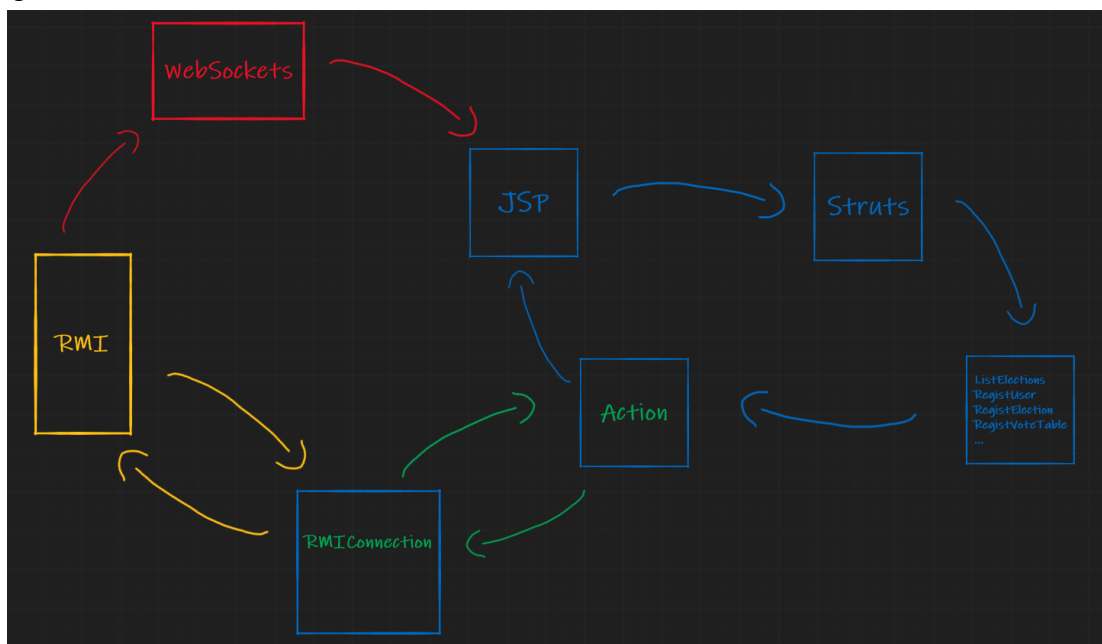


Neste modelo, a interface (*JSP*) recebe inputs e envia-os para as diversas actions através do *struts*, os dados inseridos são analisados e enviados para o servidor RMI através do *Action* e do *RMIConnection*. Posteriormente uma resposta de confirmação é dada e uma reação é disparada e executada na interface.

WebSockets

Como, a implementação dos WebSockets é apenas necessária para renderização de eleições, mesas de voto, podemos conectá-la diretamente ao servidor RMI, desta forma conseguiremos sempre a informação atualizada.

Além das funcionalidades descritas acima, e ainda através de *websockets*, imprimimos a quantidade de administradores e eleitores logados e a correspondente listagem números de cartão de cidadão dos eleitores para que, desta forma, seja assegurada a sua anonimidade.



REST API - Facebook

Para a integração da *REST API* do *Facebook* no nosso sistema eleitoral, utilizámos a biblioteca *ScribeJava* que trata do *OAuth*.

Foi necessário adquirir uma conta de developer e criar uma app para obtermos o *App ID* e o *App Secret* para ser utilizado no *service*. Este é utilizado para nos devolver o *URL* de autorização para o utilizador fazer a autenticação. Após tal se realizar, é chamada uma *action* num *callback*, e é devolvido um *code* que será utilizado para gerar o *ACCESS_TOKEN*. É feito um *request* para nos devolver o *nome* do utilizador e o *id* associado à sua conta do *Facebook*. De seguida, se tivermos a associar a conta do utilizador ao *Facebook*, guardamos no objeto *User*, as variáveis obtidas através do *request*. No caso do *login*, é feita uma pesquisa na lista de utilizadores e é feita uma comparação com o *id* obtido através do *request* e o *id* guardado, para ter a confirmação se o utilizador fez a associação com o *Facebook*, e prosseguir com o voto.

Em relação à publicação do *link* da eleição, foi utilizado o *plugin share button*. Nesse post é partilhado o *link* da página de *login* da aplicação *web*, uma vez que, para um utilizador votar, necessita de iniciar sessão na aplicação **eVoting**.


Extras

O nosso *website* pode ser hospedado usando uma ferramenta chamada *NGROK*. Esta devolve-nos dois domínios públicos automatizados (um em *HTTP* e outro em *HTTPS*) e, desta forma, conseguimos ter acesso ao nosso *website* a partir de qualquer dispositivo com acesso à *internet*, incluindo *tablets* e *smartphones*.

Fase de Testagem

Testes	Resultados
Validação dos Inputs Gerais	✓
Verificação de CCs, Departamentos, Faculdades ou Eleições já registadas(os)	✓
Verificação de Datas, números de telemóvel, validades do CC, entre outros	✓
Registar pessoas (estudantes, docentes, ou funcionários)	✓
Login protegido com password (acesso restrito a ações e a páginas)	✓

Criar eleição (incl. integração com a meta 1)	✓
Criar listas de candidatos a uma eleição	✓
Listar eleições e consultar detalhes de cada uma delas	✓
Adicionar mesas de voto a uma eleição (incl. integração com a meta 1)	✓
Alterar propriedades de uma eleição	✓
Votar (incl. integração com a meta 1)	✓
Saber em que local votou cada eleitor	✓
Eleição termina corretamente na data, hora e minuto marcados	✓
Consultar resultados detalhados de eleições passadas	✓
Página de uma eleição mostra eleitores em tempo real	✓
Páginas de administração mostram o estado das mesas de voto (da meta 1)	✓
Listar utilizadores online	✓
Associar conta ao Facebook	✓
Login com o Facebook	✓
Partilha da página de uma eleição no Facebook	✓
Partilha dos resultados de uma eleição com link no Facebook	✗
Uso de AJAX	✓
Utilização em smartphone ou tablet	✓
Uso de HTTPS	✓
O projeto corre distribuído por várias máquinas	✓
Código HTML e Java estão separados	✓
A aplicação não apresenta erros/exceções/avarias	✗ / ✓

Código legível e bem comentado	
--------------------------------	---