



**FCTUC** **FACULDADE DE CIÊNCIAS  
E TECNOLOGIA**  
UNIVERSIDADE DE COIMBRA

iVote

Bruno Grifo N°2014228262

Fábio Antunes N°2014206491

# Arquitetura do Projeto

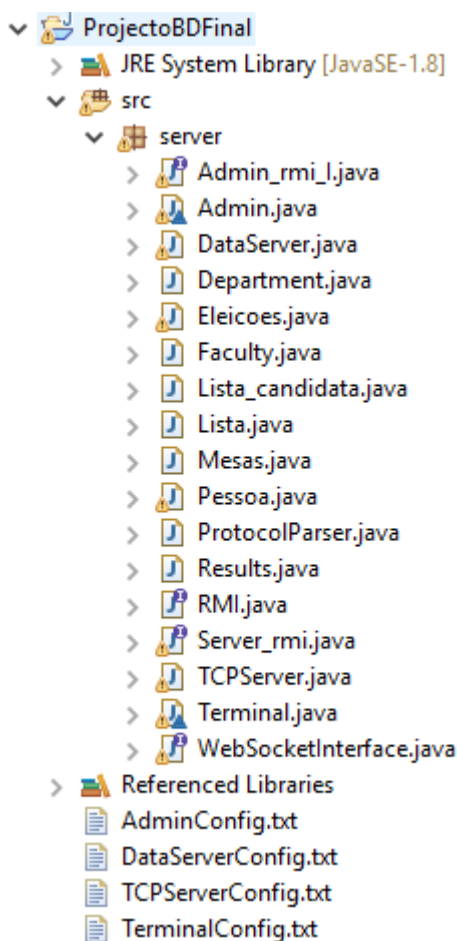
O nosso projeto é constituído por 2 partes distintas, mas que se completam. Temos a aplicação web implementada com um servidor Tomcat e a parte desktop provida de um servidor RMI e um servidor TCP. Ambos acedem remotamente a uma base de dados com uma interface RMI.

O servidor de RMI é um processo que acede aos dados armazenados na base de dados MySQL e responde aos pedidos efetuados pelo servidor de TCP e pelo *WebServer* Tomcat provenientes dos clientes ligados à aplicação.

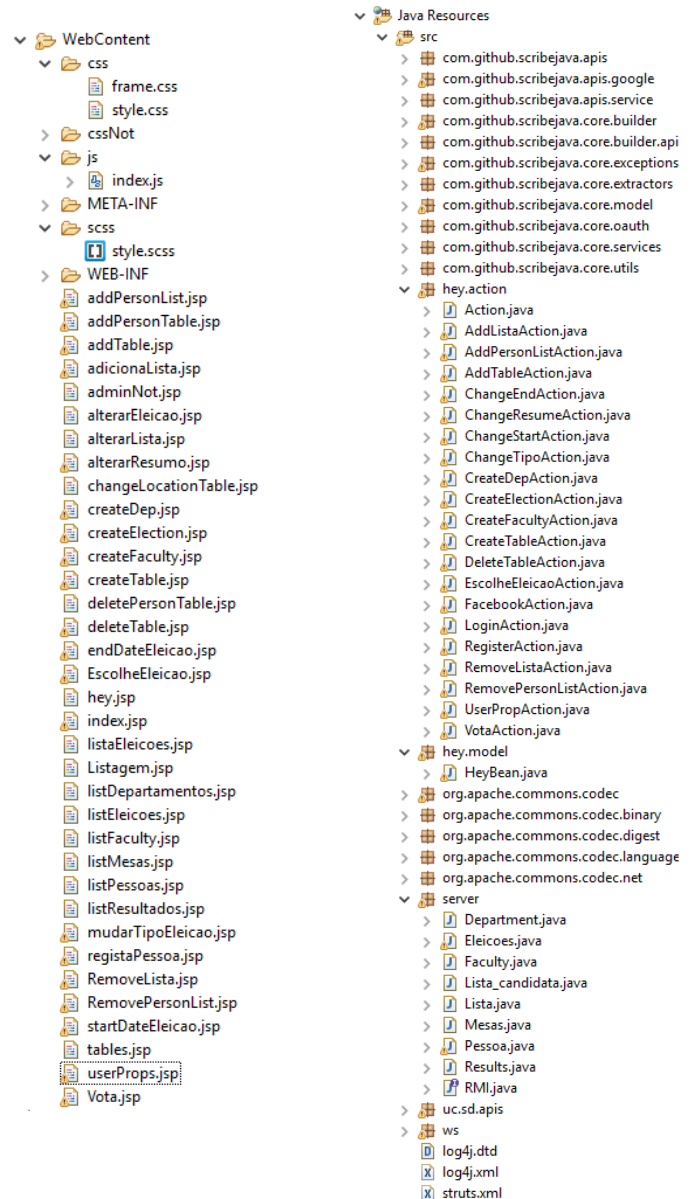
O servidor de TCP e o *WebServer* servem de intermediários entre os clientes e os dados disponíveis na aplicação, interpretando os pedidos dos clientes, obtendo a resposta proveniente do servidor de RMI que, posteriormente, é enviada para o correspondente utilizador. O *WebServer* server faz uso de várias *actions* dedicadas à realização das várias funcionalidades disponibilizadas aos utilizadores e administradores.

Estrutura:

## Meta 1



## WebServer



## Funcionamento da Base de dados (RMI)

O servidor RMI (DataServer.java) implementa as interfaces RMI.java, Server\_rmi.java e Admin\_rmi\_i.java, que a interface RMI.java é utilizada para a aplicação poder invocar métodos na base de dados e as restantes interfaces servem para fazer *call backs*.

## Funcionamento do Servidor TCP

O servidor de TCP é provido de várias *threads* que são criadas quando um novo utilizador se liga a plataforma, essa *thread* permanece em funcionamento até que o cliente se desliga da aplicação. Cada *thread* é responsável apenas por um cliente.

O servidor realiza o *parsing* das mensagens recebidas dos clientes e do tratamento da sua informação, caso as mensagens enviadas pelos clientes sejam inválidas, o servidor retorna uma mensagem de erro.

O servidor faz *lookup* ao *rmi registry* para obter uma referencia remota para o servidor de RMI podendo, assim, posteriormente invocar métodos no servidor remoto obtendo os dados importantes para responder aos pedidos dos utilizadores na app da meta 1.

# Testes Realizados

Visto que a implementação das websockets e REST não foram bem sucedidos pela nossa parte, não temos a integração dos mesmos no relatório.

55	Requisitos Funcionais	55	0	65
5	Registar pessoas (estudantes, docentes, ou funcionários)	5		5
5	Login protegido com password (acesso a ações e a páginas)	5		5
5	Criar eleição (incl. integração com a meta 1)	5		5
5	Criar listas de candidatos a uma eleição	5		5
5	Listar eleições e consultar detalhes de cada uma delas	5		5
5	Adicionar mesas de voto a uma eleição (incl. integração com a meta 1)	5		5
5	Alterar propriedades de uma eleição	5		5
5	Votar (incl. integração com a meta 1)	5		5
5	Saber em que local votou cada eleitor	5		5
5	Eleição termina corretamente na data, hora e minuto marcados	5		5
5	Consultar resultados detalhados de eleições passadas	5		5
0	Grupos de 3: Alterar dados pessoais (-5)	0		5
0	Grupos de 3: Gerir membros de cada mesa de voto (-5)	0		0
0	Grupos de 3: Considerar eleições de departamento e faculdade (-5)	0		5
15	WebSockets	15	0	0
5	Página de uma eleição mostra eleitores em tempo real	5		
5	Páginas de administração mostram o estado das mesas de voto (da meta 1)	5		
5	Listar utilizadores online	5		
20	REST	20	0	0
5	Associar conta existente ao Facebook	5		
5	Login com o Facebook	5		
5	Partilha da página de uma eleição no Facebook sendo o post atualizado no fim	5		
5	Post no Facebook de um eleitor assim que vote numa eleição	5		
0	Grupos de 3: deve ser possível desassociar um utilizador do facebook (-5)	0		
10	Relatório	10	0	4
2	Arquitetura do projecto Web detalhadamente descrita	2		2
2	Integração do Struts com o servidor RMI	2		0
2	Integração de WebSockets com Struts e RMI	2		0
2	Integração de APIs REST no projecto	2		0
2	Testes de software (tabela: descrição e pass/fail de cada teste)	2		2
	Extra (até 5 pontos)	0	0	2
	Utilização de HTTPS (4 pts)			
	Utilização em smartphone ou tablet (2pts)			2





