



FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

RELATÓRIO DA SEGUNDA META DE SISTEMAS DISTRIBUÍDOS



Rui Pedro Costa Linhares

2015227420

rlinhares@student.dei.uc.pt

Tiago Guilherme Lopes Gomes

2015245562

tglgomes@student.dei.uc.pt

22.10.2017

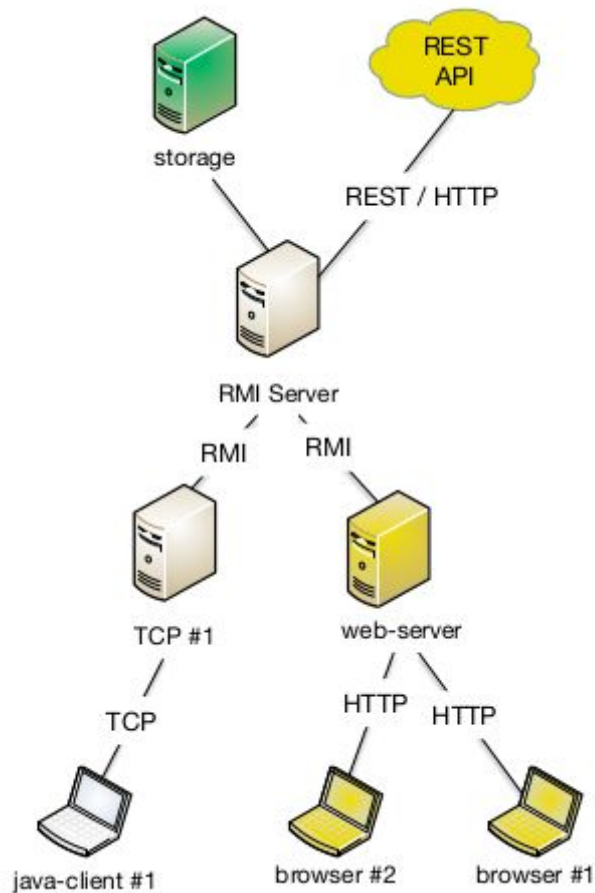
1.Introdução

Para a Meta 1 do Trabalho Prático de Sistemas Distribuídos, foi pedido aos alunos que criassem um Sistema de voto eletrónico para as eleições da Universidade de Coimbra. O objetivo é um eleitor dirigir-se a uma mesa de voto, autenticar-se e votar no terminal de voto. Para isso, foi usada uma arquitetura com recurso a dois servidores RMI, um ou mais servidores TCP e terminais de voto.

Na Meta 2 foi pedido para criar um frontend Web para a nossa aplicação sendo que esta nova interface irá possibilitar que os utilizadores acedam ao nosso serviço a partir de quase qualquer dispositivo com Internet no planeta, sem necessitar de instalação de software cliente.

Como a interoperabilidade é um requisito muito importante, utilizadores Web deverão aceder à mesma informação que os utilizadores na aplicação desktop. Para tal, o servidor Web deverá comunicar com o servidor RMI desenvolvido na meta 1. Os utilizadores deverão ter as mesmas funcionalidades, independentemente da interface que usem. Portanto a interface Web deverá listar as eleições, deverá permitir a criação de eleições e votar.

2.Arquitetura



1. Arquitetura do Sistema

A arquitetura implementada inclui dois servidores RMI, um primário e um secundário; Servidores TCP que representam as mesas de voto e por fim, clientes TCP como terminais de voto. Foram implementados também um servidor web e criadas interfaces web para utilização na internet.

1.RMI Server

É provavelmente a peça mais importante do sistema, sendo que é aqui que estão a decorrer todas as principais operações de registo e alteração de dados. Estabelece comunicação através de duas Interfaces RMI com a Consola de Administrador e com os Servidores TCP. Durante a execução do RMI server é instanciada uma entidade de RMI Registry. O RMI server implementa também um server UDP para comunicação entre o server primário e o secundário com o objetivo de confirmar a existência de um servidor primário a funcionar. Devem

assim ser corridas duas instâncias do RMI server. Em caso de avaria, o servidor secundário assume o papel de primário e continue a execução do programa sem que a avaria seja visível para o cliente. Ao iniciar o RMI server ele vai verificar se já existe alguma instância no RMIRegistry, e se existir vai ficar como servidor secundário. Isto permite a existência de um failover que garante a consistência da utilização por parte das consolas de administração, TCP servers e utilizadores web. No RMIServer estão implementados os métodos declarados na interface RMI que serão depois invocadas pelos servidores TCP e Web de modo a responder aos requests dos clientes.

2. Consola de Administrador

A consola de administrador é a parte do sistema onde através de chamadas RMI pela interface AdminRMIImplements, conseguimos gerir todos os dados do sistema. Todos os métodos da interface referida são synchronized para impedir duplicação ou perda de dados. A consola de administrador implementa as seguintes funcionalidades:

- Registar Pessoas
- Criar e Gerir Departamentos
- Criar Eleições
- Gerir Listas de candidatos
- Gerir Mesas de voto
- Alterar Eleições
- Verificar o estado das mesas de voto
- Saber onde votou cada eleitor
- Consultar Eleições em tempo real

3. TCP Server e TCP Client

O TCP server vai servir como uma mesa de voto, onde os eleitores são identificados e posteriormente autenticados, agora no terminal de voto (TCP Client). O TCP server comunica com o RMI server através de chamadas RMI pela interface TCPserverRMIImplements e com o TCP Client através de sockets TCP. Cada conexão com um terminal de voto é uma thread diferente para que as mesas de voto consigam atender vários terminais de voto. Para a troca de mensagens entre o TCP server e o TCP client foi desenvolvido um protocolo para as mensagens trocadas:

1. O Eleitor é identificado no Terminal de Voto. É autorizada a abertura de um terminal de voto.
2. O Eleitor abre o terminal de voto e autêntica-se enviando a seguinte mensagem para o TCP server:
`type|auth;uc|*numerouc*;password|*password*`
3. O TCP server confirma os dados e devolve o boletim de voto:
`Lista A|1;LISTA B|2;LISTA C|3`
4. O eleitor escolhe vota enviando a seguinte mensagem:
`type|vote;option|*número da lista*`

5. O eleitor escolhe vota enviando a seguinte mensagem:

`type|vote;option|*número da lista*`

6. O TCP server responde com:

`TYPE|VOTE;VOTE|CONFIRMED`

No terminal de voto, após 120s sem input, o terminal é fechado.

4. Servidor Web

O servidor Web foi implementado seguindo o MVC: Model, View e Controller. Corre num servidor Tomcat 8.5.0. Para esta segunda meta e para a criação e utilização de uma interface web, foi utilizada a framework Struts2. Deste modo as Actions correspondem ao Controller, os JSP files às Views e o iVotasBean ao Model.

Organizando os ficheiros desta maneira, há a garantia de independência entre o backend e o frontend.

Aquando da inicialização do bean, este liga-se ao RMI e servirá como um intermédio entre este e a interface web (e correspondentes ações). É assim responsável por todas as chamadas de métodos efetuadas ao RMI. Assim, quando o utilizador quer realizar alguma operação, esta será tratada pelas ações que recorrem ao bean para fazer os pedidos ao servidor. Assim todos os dados são tratados pelo bean para depois estarem ao dispor do utilizador através das views.

3. Integração das duas Metas (Struts2 com Server RMI)

Para a integração das duas metas, foi utilizado (já referido anteriormente) o modelo MVC. O Web server corre através do Tomcat server utilizando struts2 para a interface web. Deste modo, é garantida a possibilidade das actions obterem dados inseridos pelos utilizadores nas views e que o bean consiga realizar os requests ao RMI Server.

4. Tabela de testes

Requisitos Funcionais	55
Registar pessoas (estudantes, docentes, ou funcionários)	PASS
Login protegido com password (acesso a ações e a páginas)	PASS
Criar eleição (incl. integração com a meta 1)	PASS
Criar listas de candidatos a uma eleição	PASS
Listar eleições e consultar detalhes de cada uma delas	PASS
Adicionar mesas de voto a uma eleição (incl. integração com a meta 1)	PASS
Alterar propriedades de uma eleição	PASS
Votar (incl. integração com a meta 1)	PASS
Saber em que local votou cada eleitor	PASS
Eleição termina corretamente na data, hora e minuto marcados	PASS
Consultar resultados detalhados de eleições passadas	PASS

WebSockets	15
Página de uma eleição mostra eleitores em tempo real	FAIL
Páginas de administração mostram o estado das mesas de voto (da meta 1)	FAIL
Listar utilizadores online	FAIL

REST	20
Associar conta existente ao Facebook	FAIL
Login com o Facebook	FAIL
Partilha da página de uma eleição no Facebook sendo o post atualizado no fim	FAIL
Post no Facebook de um eleitor assim que vote numa eleição	FAIL

6. Distribuição de Tarefas

O trabalho foi uniformemente distribuído pelos membros, sendo as funcionalidades implementadas distribuídas igualmente pelos membros do grupo.

