



UNIVERSIDADE DE  
COIMBRA

FACULDADE  
DE CIÊNCIAS  
E TECNOLOGIA

Licenciatura em Engenharia Informática  
Sistemas Distribuídos  
2020/2021 – 2º Semestre

eVoting: Voto Eletrónico na UC

*João Filipe Carnide de Jesus Nunes*    2017247442  
*Raul Filipe Enes Nogueira*    2017267634

[uc2017247442@student.uc.pt](mailto:uc2017247442@student.uc.pt)  
[uc2017267634@student.uc.pt](mailto:uc2017267634@student.uc.pt)

# 1. Introdução

Para a meta 1 do trabalho prático, foi proposta a realização de um sistema de votos eletrónicos para a Universidade de Coimbra. O objetivo geral do trabalho é um utilizador dirigir-se a uma mesa de voto, fazer a sua autenticação e, de seguida, ir a um terminal de voto para poder votar. Para isto foi utilizada uma arquitetura cliente-servidor usando RMI (para o servidor principal do sistema e as consolas de administração) e Multicast (para as mesas e terminais de voto).

Na meta 2 foi proposto a criação de um front-end web para a mesma aplicação da meta 1. Esta interface possibilitará que os diferentes utilizadores acessem ao serviço a partir de qualquer dispositivo com Internet no mundo, sem ser necessário a instalação de software cliente.

Para existir uma relação entre a meta 1 e 2, deverá ser possível cada utilizador poder exercer o voto através de modo local, pelas mesas e terminais de voto (Multicast), ou modo remoto, pelo browser. Ambas as formas de voto devem ter exatamente as mesmas funcionalidades para que não exista qualquer diferença no voto, exceto o seu local.

## 2. Arquitetura

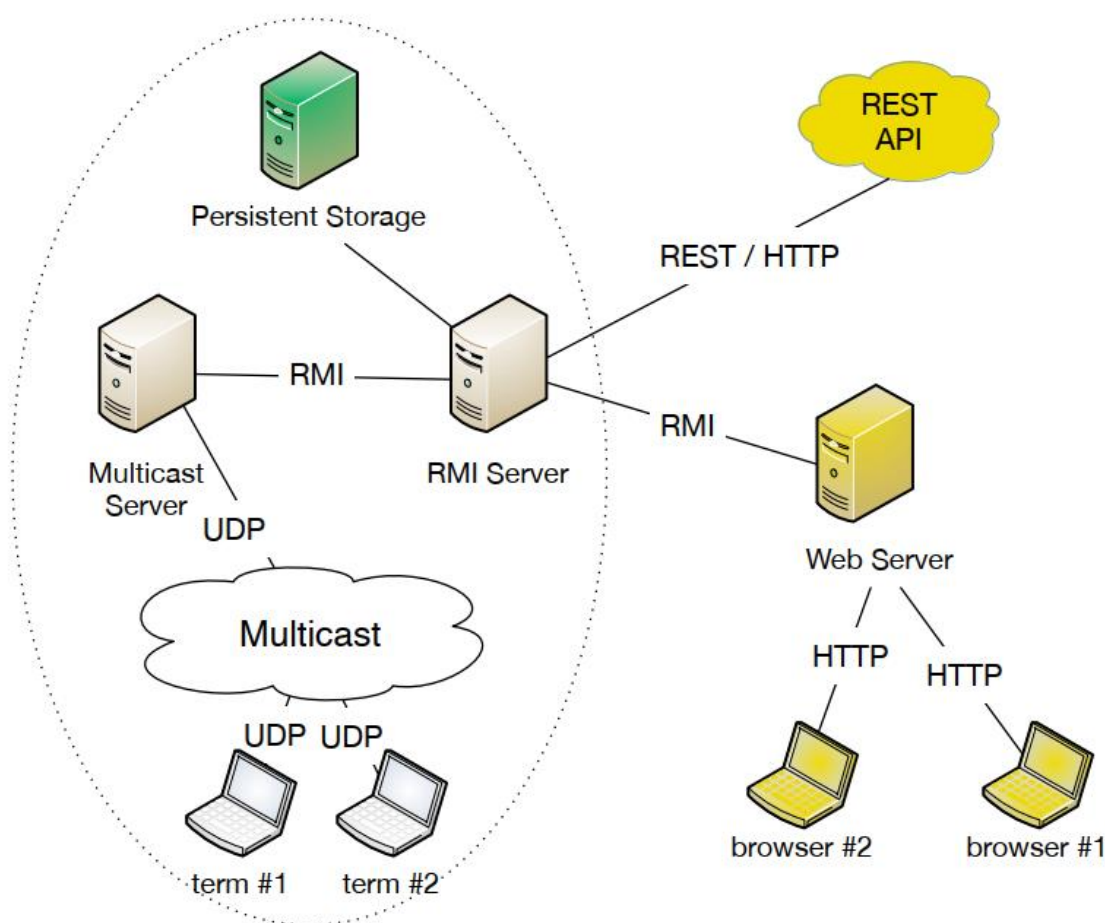


Figura 1 - Arquitetura de software do projeto (meta 1 + 2)

A arquitetura implementada em ambas as metas consiste em dois servidores RMI (um primário e um secundário); clientes RMI que são as consolas de administração; servidores Multicast que representam as mesas de voto; clientes Multicast que representam os terminais de voto; servidores Web e interfaces para a utilização na internet através de HTTP; uma API REST que se liga com o servidor RMI para que seja possível aos eleitores associarem a sua conta de Facebook (através de OAuth), para que seja possível fazer login pelo Facebook sem a necessidade de nome e password e poderem partilhar a eleição em que votaram e, posteriormente, partilhar também os resultados de eleições passadas.

## 1. Servidor RMI

Este servidor é o centro do sistema, tudo o que é criado, editado e autenticado passa por este servidor RMI. Sempre que é criada ou editada uma funcionalidade do sistema, o RMI, por invocação remota, recebe esta informação e atualiza a base de dados à qual esta ligada. Para fazer a comunicação entre as consolas de administração e as mesas de voto foram criadas interfaces para, a partir de métodos implementados neste servidor, fosse feita a invocação remota destes métodos que fossem necessários nos diferentes clientes.

No que diz respeito à ligação entre o servidor RMI primário e o secundário estes estão conectados através de uma ligação UDP. Para que não exista *failover*, o servidor secundário está, de 5 em 5 segundos, a enviar mensagens ao primário para ver se este continua a funcionar de maneira segura, respondendo o servidor primário com uma mensagem a dizer que está tudo operacional. Caso uma destas mensagens que o servidor primário envia não seja recebida pelo secundário, significa que algum problema aconteceu ao servidor primário. Assim, o servidor secundário toma controlo do sistema, tornando-se este o novo servidor primário. Quando o servidor primário inicial voltar ao normal, este passa a secundário e faz as mesmas funções que o secundário inicial fazia (envio de mensagens para certificação do servidor primário).

## 2. Consola de Administração

As consolas de administração são responsáveis pela criação, edição e consulta de vários agentes presentes no sistema de uma eleição. Através da invocação de métodos presentes numa interface, que são implementados no servidor RMI, de modo sincronizado para não existir duplicação ou perda de dados importantes, as consolas de administração implementam as seguintes funcionalidades:

- Registrar pessoas
- Criar eleições
- Gerir listas de candidatos a uma eleição
- Gerir mesas de voto
- Alterar propriedades de uma eleição
- Saber em que local votou cada eleitor
- Mostrar o estado das mesas de voto
- Mostrar eleições em tempo real
- Consultar resultados detalhados de eleições passadas

### 3. Servidor Multicast (Mesa de Voto)

Em servidor Multicast tiveram de ser criados dois grupos, um para a deteção de máquinas (terminais de voto) e outro para comunicar com a mesa de voto correta. Para a deteção de máquinas estas devem comunicar através do endereço 224.0.224.0 e porto 4321. Aquando de conectados o Multicast Server atribui, aos terminais, um endereço de uma mesa de voto que é único por cada uma (endereço gerado aleatoriamente) e um porto que será 4321 ou 4322 conforme se a mesa de voto já tem ou não algum terminal conectado.

Em cada mesa de voto, também são realizados logouts de forma que um utilizador não possa dar login caso já esteja logged in.

### 4. Terminal de Voto

Os terminais de voto, inicialmente, encontram-se bloqueados até receberem uma mensagem de desbloqueio conforme a autenticação de um utilizador na mesa voto através de uma HashMap com a informação do utilizador (nome e número). Cada terminal está desbloqueado, no máximo por 120 segundos. Caso estes 120 segundos sejam atingidos, ou o utilizador vote no terminal, este recebe uma mensagem de bloqueio e volta a esperar por uma nova mensagem de desbloqueio. Para isto foram utilizadas exceções do tipo SocketTimeout para ir verificando se o socket da mesa de voto recebia mensagens do terminal de voto para, caso não recebesse, mandar mensagem de bloqueio.

### 5. Servidor Web

O servidor Web foi implementado usando uma arquitetura MVC (Model, View e Controller), correndo num servidor Tomcat. Para a criação e utilização de uma interface web foi utilizada a framework Struts2. As diferentes páginas foram criadas através de Java Server Pages (correspondente às Views), as ações de submissões de qualquer informação para o RMI ou entre páginas correspondem ao Controller e o bean corresponde ao Model (para fazer a ligação com o RMI e mais funcionalidades).

## **3. Integração de Struts com o servidor RMI**

Para a integração com as duas metas, foi utilizada uma arquitetura MVC. O servidor Web corre através do servidor Tomcat utilizando Struts para o desenvolvimento de JSP's (interface web). Assim, é garantida a possibilidade das actions obterem os dados inseridos pelos diferentes utilizadores nas views e que o bean (o model) consiga realizar pedidos ao servidor RMI, como, por exemplo, a verificação de utilizadores, o envio de votos feitos online, entre outros.

Para aceder à página de administração apenas o utilizador com nome e password *admin* pode aceder a todas as funcionalidades semelhantes à consola de administração da meta 1. Para que um utilizador normal não possa aceder às páginas correspondentes a páginas de administração foi criado um interceptor que envia para a página de login caso o utilizador tente aceder a páginas exclusivas ao administrador.

Após entrar na página de votar são mostradas todas as eleições disponíveis para determinado utilizador através de *radio buttons*. Após a seleção da eleição são mostradas todas as opções de voto, também através de *radio buttons*. Após a confirmação do voto, este é adicionado à eleição bem como sendo guardada a hora e local de voto, que neste caso será considerado como voto online.

## 4. Integração de WebSockets com Struts e RMI

Foram criados dois WebSockets um para enviar as mensagens da atividade do utilizador e outro que lê estas mensagens e mostra-as ao utilizador. Este web socket de escrita envia as mensagens de login de um utilizador, notificação de voto numa dada eleição, número total de votos até ao momento numa dada eleição e notificação de logout por parte de um utilizador.

Durante a implementação do WebSocket de escrita obtivemos um problema relacionado com o envio das mensagens, que se devia ao facto de estarmos a enviar a mensagem sem o socket já estar conectado. Depois de alguma pesquisa descobrimos que os WebSockets têm um event listener chamado onopen(), que executa uma ação aquando do WebSocket já estar conectado.

## 5. Integração de APIs REST no projeto

Para a realização deste ponto utilizámos as bibliotecas disponibilizadas na ficha 8:

- async-http-client-1.9.31.jar
- json\_simple-1.1.jar
- slf4j-api-1.7.13.jar

Também foi criada a aplicação “testeREST” na plataforma FacebookDevelopers, em que possibilitamos redirecionamento para a página <https://eden.dei.uc.pt/~fmduarte/echo.php> de forma que seja gerado um código de confirmação para o utilizador poder dar login com o Facebook ou mesmo para associar a sua conta ao seu Facebook.

Aquando da associação da conta é devolvido um FacebookID, único por utilizador, que é guardado num atributo do utilizador para mais tarde ser detetado quando for feito um login através do Facebook.

Para o login através desta plataforma é necessário ter conta registada e já associada ao FacebookID com que se pretende dar login. Caso esta última condição não se verifique o utilizador é retornado para a página inicial de login, caso contrário é identificado o utilizador e dado login para a página de voto.

## 6. Testes de Software

Requisitos Funcionais	Pass/Fail
Registrar pessoas (estudantes, docentes ou funcionários)	Pass
Login protegido com password (acesso restrito 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	Pass/Fail
Página de uma eleição mostra eleitores em tempo real	Pass
Páginas de administração mostram o estado das mesas de voto (da meta 1)	Fail
Listar utilizadores online	Pass

REST	Pass/Fail
Associar conta ao Facebook	Pass
Login com o Facebook	Pass
Partilha da página de uma eleição no Facebook	Fail
Partilha dos resultados de uma eleição com link no Facebook	Fail