

Licenciatura em Engenharia Informática

Sistemas Distribuídos

2020/2021 – 2º Semestre

eVoting: Voto Eletrónico na UC

*João Filipe Carnide de Jesus Nunes 2017247442* [*uc2017247442@student.uc.pt*](mailto:uc2017247442@student.uc.pt) *Raul Filipe Enes Nogueira 2017267634* [*uc2017267634@student.uc.pt*](mailto:uc2017267634@student.uc.pt)

**1. Introdução**

Para esta meta 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 puder 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).

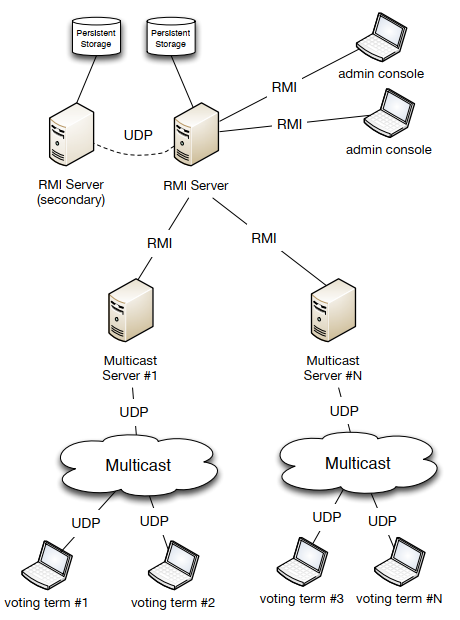
**2. Arquitetura**

Figura - Arquitetura do Trabalho

A arquitetura implementada 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. Existe também uma base de dados ligada com os servidores RMI que é atualizada sempre que são feitas alterações no sistema (criação de utilizadores, adição de votos, etc.).

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:

* Registar 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)

(raul)

4. Terminal de Voto

(raul)

**3. Distribuição de Tarefas**

Mediante a distribuição de tarefas, o trabalho foi distribuído do seguinte modo: um membro do grupo fez a implementação do RMI Server e da Admin Console e o outro membro fez a implementação do Multicast Server (Mesas de Voto) e dos Terminais de Voto.

Contudo, sempre que fosse necessário, em caso de problemas numa dada parte do trabalho que um dos membros do grupo não conseguisse resolver sozinho, ambos se juntavam e tentavam resolver esse problema juntos.

**4. Testes de Software**

|  |  |
| --- | --- |
| **Requisitos Funcionais** | **Pass/Fail** |
| Registar novo utilizador (estudante, docente, ou funcionário) |  |
| Criar eleição |  |
| Gerir listas de candidatos a uma eleição |  |
| Criar mesas de voto |  |
| Gestão automática de terminais de voto, por Multicast |  |
| Identificar eleitor na mesa de voto e desbloquear um terminal de voto |  |
| Login de eleitor no terminal de voto |  |
| Votar (escolher, uma só vez, uma lista no terminal de voto) |  |
| Editar propriedades de uma eleição |  |
| Saber em que local votou cada eleitor |  |
| Consola de administração mostra mesas de voto on/off e votantes |  |
| Consola de administração atualizada em tempo real nas eleições |  |
| Eleição termina corretamente na data, hora e minuto marcados |  |
| Consultar resultados detalhados de todas as eleições passadas |  |

|  |  |
| --- | --- |
| **Tratamento de Exceções** | **Pass/Fail** |
| Avaria de um servidor RMI não tem qualquer efeito nos clientes |  |
| Não se perde/duplica votos se os servidores RMI falharem |  |
| Não se perde/duplica votos se a comunicação Multicast tiver falhas |  |
| Avarias temporárias (<30s) dos 2 RMIs são invisíveis para clientes |  |
| Terminal de voto bloqueado automaticamente após 60s sem uso |  |
| Crash de terminal de voto é recuperado |  |

|  |  |
| --- | --- |
| **Failover** | **Pass/Fail** |
| O servidor RMI secundário testa periodicamente o primário |  |
| Em caso de avaria longa os servidores Multicast ligam ao secundário |  |
| Servidor RMI secundário substitui o primário em caso de avaria |  |
| Os dados são os mesmos em ambos os servidores RMI |  |
| O failover é invisível para clientes/eleitores (não perdem a sessão) |  |
| O servidor original, quando recupera, torna-se secundário |  |