



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Relatório Meta 1

Projeto de SD 2020-2021

E-voting

Sofia Meireles F. Costa - 2018296218 - sofiacosta@student.dei.uc.pt
Sofia Silva - 2018293871 - sofiasilva@student.dei.uc.pt

Índice

Multicast.....	2
Estrutura do protocolo.....	4
RMI	6
RMI Server:	6
RMI Client/Admin console	6
Distribuicao de tarefas	10
Testes.....	11

Multicast

Inicialmente definimos como seriam as duas multicast address, uma para descobrir e ser descoberto e outra para votar, tal como pedido no enunciado. De forma a garantir multicast addresses diferentes para todas as mesas, usamos a variável mesa na criação da address, "224.0.224."+mesa para o grupo de descobrir e ser descoberto e, da mesma forma, para o grupo de comunicação da intenção de voto "224.0.224."+mesa+100, assim, conseguimos garantir que não há conflitos. Ao definir desta forma, não nos foi possível arranjar uma maneira de garantir que ao iniciar o terminal de voto(multicast client) este ligar-se-á a uma mesa existente pois o join ao grupo multicast só será realizado posteriormente, já utilizando o valor inserido logo de seguida a correr o multicast client, uma vez que é o primeiro dado pedido ao correr o multicast client (ficando permanentemente bloqueado caso não seja criada uma mesa com esse número) .

Ao iniciar o multicast server, este apresentará a lista de mesas de voto que se encontrem ativas naquele momento, utilizando a ligação ao servidor RMI. Assim, o utilizador poderá escolher em que mesa pretende votar. A partir do momento em que se escolhe a mesa, essa mesa ligar-se-á ao ao grupo multicast e será iniciado o processo de voto. Será pedido ao eleitor o número do seu cartão de cidadão, que será enviado para o multicast client através do protocolo definido (explicitado mais à frente no documento). Caso esse número seja válido, o terminal desbloquear-se-á. De seguida, as mesas e os terminais de voto trocam mensagens através deste protocolo, de forma a permitir ao eleitor poder validar o seu login e iniciar o processo do voto. Caso o terminal desbloqueado fique inativo por cerca de 120 segundos, o mesmo será bloqueado. Na eventualidade de todos os terminais estarem ocupados no momento em que surge um novo eleitor, será pedido ao eleitor que espere até que algum deles se encontre disponível. Para isso, foi criado um ciclo que está constantemente a receber os pacotes e os ids dos terminais e respetiva

disponibilidade, caso esteja 5 segundos sem receber pacotes, volta a enviar mensagem a pedir a disponibilidade dos terminais.

No multicast server, existem duas threads, uma para ir recebendo os eleitores que cheguem à mesa de voto e procurar terminais livres para eles, e outra para tratar do voto do eleitor em si.

No multicast client, existem três threads, uma para enviar os terminais, onde no momento em que é encontrado um terminal livre, inicia a thread de envio dos dados que o eleitor inseriu, e por fim, a outra thread serve para receber as respostas do eleitor, distinguindo-as por type e enviando para a thread de envio para o server (type == 0 -> login, type == 1 -> envia a eleição escolhida, type == 2 -> envia a lista em que o eleitor votou).

Interface Multicast Server (Mesa de voto):

```
Mesas de voto:
1 - deec
2 - dei
3 - dec

Escolha uma mesa:
1
Insira o numero do CC:
11
Aguardando por terminais livres...
Dirija-se ao terminal de voto nº 73
Insira o numero do CC:
```

Interface Multicast Client (Terminal de voto):

```
ID Mesa:
1
Terminal 470 bloqueado
Terminal 470 desbloqueado
Numero do cartao de cidadao:
11
Password:
1234
Welcome to eVoting
Escolha a eleicao:
1 - DEE
2 - KS
1
1 - Lista A
2 - Lista B
3 - Voto Nulo
4 - Voto Branco
1
Voto realizado com sucesso.

Terminal 470 bloqueado
```

Estrutura do protocolo

Print mesas de voto disponíveis

- **type|item_list;item_count|3;item0_name|deec;item1_name|dei;item2_name|dec;**

Pede o id dos terminais (**server**)

- **"type|status;terminal|id;time|tempoNoMomento;"**

Envia terminais e respetiva disponibilidade (**client**)

- **"type|terminal;status|on;id|numId;available|disponibilidade;time|tempoNoMomento;"**

Conecta a um id disponivel (**server**)

- **"type|status;terminal|connect;id|numId;"**

Dados login (**client**)

- **"type|login;server|id;username|numCC;password|password;"**

Caso o login seja efetuado com sucesso (**server**)

- **"type | status; server | id ; logged | on; msg | Welcome to eVoting"**

Caso o login seja invalido(**server**)

- **"type | status; server | id ; logged | off; msg | Falha ao fazer login. Tente de novo."**

Bloqueia o terminal (**client**)

- **"type|shutdown;server|id"**

Pede print das eleicoes (**client**)

- **"type|eleicoes;server|" + id + ";status|print"**

Envio lista eleicoes disponiveis (**server**)

- **"type|item_list;server| id ;item_count|numEleicoes;"**

Caso o eleitor já tenha votado naquela eleicao (**server**)

- **"type|eleicoes;server;id!numlid;eleicao|-1;"**

Caso não haja eleições a decorrer (**server**)

- **"type|eleicoes;server|id;item_count|0;"**

Escolhe eleicao (**client**)

- **"type|eleicoes;server|id;eleicao|escolha;"**;

Envio lista listas disponíveis (**server**)

- **"type|listas;server|id|numId;item_count|tamLista|tem_i_name|nomeLista ;"**

Caso não haja listas disponíveis de momento (**server**)

- **"type|listas;server|id;item_count|0;"**

Envia o voto (**client**)

- **"type|eleicoes;server|id;voto|escolheVoto ;"**

RMI

RMI Server:

No início é criado o servidor RMI, através do comando

`LocateRegistry.createRegistry()`. É criada e inicializada também a thread que controla as datas das eleições, que está explicada de forma mais completa em baixo. Relativamente ao funcionamento deste servidor, inicialmente é lido o ficheiro de objetos que contém a classe Dados, que contém todos os dados necessários ao funcionamento do programa. Este ficheiro é composto por diversos métodos que são chamados através da interface, sendo estes métodos que verificam dados, transformam dados em Strings para serem enviadas através dos protocolos e outros métodos que adicionam/eliminam e alteram os dados existentes no RMI. Existe também um método que guarda os dados no ficheiro de objetos, `guardaDados()`, que é chamado sempre que alguma alteração é feita na classe Dados.

Thread - a thread existente no RMI tem como objetivo organizar as eleições por três listas: lista de eleições concluídas, lista de eleições que ainda não começaram e lista de eleições que estão a decorrer. Para isto, de 15 em 15 segundos, percorrem-se as listas e caso as datas já não se enquadrem na lista em que estão, são mudadas para a correta.

Server secundário - de forma a evitar problemas no caso do RMI Server falhar, é criado um server secundário. Este é criado caso já exista um server no ip address e port indicadas. Para verificar se o server principal continua funcional, utilizamos o método `checkPing()`, que consiste em tentar conectar com o server. Caso esta conexão falhe 5 vezes seguidas, o RMI secundário passa a RMI primário, através do comando `LocateRegistry.createRegistry()`, usando o mesmo ip e a mesma port do primeiro RMI.

Executar em máquinas diferentes – de modo a executar o programa em diferentes máquinas ao mesmo tempo, é necessário atualizar o IP no código.

RMI Client/Admin console

A admin console conecta com o servidor RMI, através do comando

`Naming.lookup()`, que procura o *binding* “project” no RMI Registry que está a correr no IP/Port fornecido. Seguidamente é apresentado o seguinte menu ao utilizador:

```
Opcoes:
1 - Registrar eleitor
2 - Criar eleicao
3 - Registrar lista candidata
4 - Adicionar mesa de voto
5 - Eliminar mesa de voto
6 - Alterar propriedades de uma eleicao
7 - Saber em que local votou cada eleitor
8 - Consultar resultados detalhados de eleicoes passadas
9 - Número de eleitores que votaram em cada mesa em tempo real
10 - Imprimir mesas ativas/inativas
0 - Sair
```

Opção 1 - Registrar eleitor - Esta opção serve para registar um novo utilizador na base de dados. É pedido ao utilizador um nome, a sua função (tem de escolher a opção correspondente à sua função: Estudante, Docente ou Funcionário), uma password, a sigla do seu departamento, o número de telefone, a morada, o número de cartão de cidadão e a validade do mesmo. Para que o registo seja aceite, o cartão de cidadão tem de estar dentro de validade. Caso exista outro eleitor na base de dados com o mesmo número de cartão de cidadão, o pedido de registo é negado e a mensagem “Eleitor já registado” é imprimida na consola.

Opção 2 - Criar eleição - Nesta opção, são pedidos ao utilizador os dados necessários para adicionar uma nova eleição à base de dados. Estes dados são: a data/hora de início, a data/hora de fim, o título, a descrição, o tipo, isto é, para quem é que é a eleição (o utilizador escolhe um dos 4 tipos possíveis: Estudante, Docente, Funcionário, Todos) e a sigla departamento para o qual é feita a eleição (caso não seja para ser só de um departamento, o utilizador deve inserir “todos”). Nesta opção são verificadas as datas, quer o formato como a sua validade, isto é, se ainda não passaram ou se a data final é posterior à data de início.

Opção 3 - Registrar lista candidata - Aqui, são pedidos os dados necessários para criar uma lista candidata a uma determinada eleição. Para que este pedido seja possível, é necessário que já existam eleições que ainda não estejam a decorrer na base de dados. Caso não existam, a consola mostra a mensagem “Não existem eleições não começadas” e, caso existam, são imprimidas as eleições em que o utilizador pode adicionar uma lista candidata. Depois de selecionar a opção correspondente à eleição pretendida, são pedidos os seguintes dados: Nome da lista, função dos membros (mais uma vez são mostradas as opções: Estudante, Docente e Funcionário), número de membros da lista e, consoante este número, o número do Cartão de Cidadão dos membros. Se tudo estiver correto, é imprimida a mensagem “Lista inserida com sucesso”. Se não, aparecem mensagens que explicam o que estava incorreto ou em falta: “Lista não aceite. Possui candidatos

não registados.", "Lista não aceite. Possui candidatos de diferentes tipos" ou "Lista não aceite. Possui candidatos que não pertencem ao departamento."

Opção 4 - Adicionar um mesa de voto - Para adicionar uma mesa de voto, é necessário que existam eleições que ainda não estejam a decorrer. Esta primeira verificação e escolha da lista é feita da mesma forma que na opção 3. Depois de escolher a eleição, é pedido ao utilizador a sigla do departamento da mesa. Aqui são possíveis 3 cenários:

- a eleição já ter uma mesa no departamento escolhido e é imprimida a mensagem "Esta eleição já possui uma mesa neste departamento [sigla do departamento]";
- a mesa já existir, então a eleição é adicionada à mesa de voto do departamento escolhido e é imprimida a mensagem "Eleicao adicionada à mesa com sucesso";
- a mesa ainda não existir, então é criada uma nova mesa e é imprimida a mensagem "Mesa adicionada com sucesso".

Opção 5 - Eliminar mesa de voto - Esta opção tem praticamente a mesma estrutura que a opção 4, mas em vez de adicionar elimina uma eleição de uma mesa de voto. Aqui os cenários possíveis são os seguintes:

- a mesa de voto não é encontrada e é imprimida a mensagem "Mesa não encontrada";
- a mesa de voto só tem uma eleição, então é eliminada e é imprimida a mensagem "Mesa eliminada com sucesso";
- a mesa de voto tem mais eleições, então a eleição é eliminada da mesa e é imprimida a mensagem "Eleicao eliminada da mesa com sucesso".

Opção 6 - Alterar propriedades de uma eleição - Nesta opção o utilizador tem a possibilidade de alterar as propriedades de uma eleição à escolha. São mostradas as eleições que ainda não estão a decorrer, e é pedido ao utilizador que escolha a que pretende alterar. Seguidamente, são apresentadas as informações atuais da eleição selecionada e um menu com opções, como se pode ver na imagem seguinte.


```
Título: NEI
Descricao: Eleicao para os corpos gerentes do NEI/AAC.
Data de inicio: 6-6-2021 10:0
Data de fim: 6-6-2021 20:0
Tipo: EST
Departamento: DEI

0 que pretende editar?
1 - Título
2 - Descricao
3 - Data de inicio
4 - Data de fim
5 - Tipo
6 - Departamento
0 - Sair
```

Cada uma das opções permite que seja inserida a respetiva alteração, que depois de verificada e confirmada pelo utilizador, é atualizada na eleição escolhida.

Opção 7 - Saber em que local votou cada eleitor - Neste opção é possível ver, dentro da eleição escolhida, o local (mesa de voto) e a hora em que cada eleitor votou. Caso não existam votos na eleição escolhida, aparece a mensagem “Nao existem votos”. Caso existam, é apresentada uma lista com a informação pretendida no seguinte formato:

```
"nCC: [número do CC do eleitor] - mesa: [mesa de voto em que o eleitor votou] - [data e hora em que o eleitor votou]"
```

Exemplo:

```
nCC: 99 - mesa: dei - 5-4-2021 5:9
nCC: 11 - mesa: dei - 5-4-2021 4:23
nCC: 1 - mesa: dei - 5-4-2021 5:53
```

Opção 8 - Consultar resultados detalhados de eleições passadas - Esta opção permite que o utilizador consulte os resultados de uma eleição passada à sua escolha. Depois de escolher a eleição passada desejada, são apresentadas as informações da eleição (título, descrição, data de início, data de fim, tipo e departamento) seguidas da informação sobre os votos, como se pode ver no exemplo abaixo:

```

Insira o número da eleição:
1
Titulo: NEI
Descricao: Eleicao para os corpos gerentes do NEI/AAC.
Data de inicio: 6-6-2021 10:0
Data de fim: 6-6-2021 20:0
Tipo: EST
Departamento: DEI

Total de Votos: 3
Lista AAA: 1 votos (33,33%)
Lista DD: 1 votos (33,33%)
Votos em branco: 0 votos (0,00%)
Votos nulos: 1 votos (33,33%)

```

Opção 9 - Número de eleitores que votaram em cada mesa em tempo real -

Nesta opção é possível verificar o número de votos em tempo real numa determinada mesa de voto, numa dada eleição. O formato é o seguinte:

```
"No de votos na mesa [local da mesa], eleicao [nome da eleicao]: [nº de votos]"
```

Opção 10 - Imprimir mesas ativas/inativas - Aqui o utilizador pode visualizar as mesas ativas e inativas e os respetivos terminais.

Proteções - Como forma de evitar que o nosso programa terminasse devido a erros de input, acrescentámos algumas proteções, para além das já mencionadas em cima, sendo elas: verificar se o input é um número inteiro, verificar se a opção selecionada está dentro dos valores possíveis, verificar se as datas estão no formato correto e verificar se o número de telemóvel tem 9 dígitos.

Distribuição de tarefas

Sofia Meireles	Sofia Silva
RMI	Multicast e Failover

Testes

Registar novo utilizador (estudante, docente, ou funcionário)	<i>pass</i>
Criar eleição	<i>pass</i>
Gerir listas de candidatos a uma eleição	<i>pass</i>
Criar mesas de voto	<i>pass</i>
Gestão automática de terminais de voto, por Multicast	<i>pass</i>
Identificar eleitor na mesa de voto e desbloquear um terminal de voto	<i>pass</i>
Login de eleitor no terminal de voto	<i>pass</i>
Votar (escolher, uma só vez, uma lista no terminal de voto)	<i>pass</i>
Editar propriedades de uma eleição	<i>pass</i>
Saber em que local votou cada eleitor	<i>pass</i>
Consola de administração mostra mesas de voto on/off e votantes	<i>pass</i>
Consola de administração atualizada em tempo real nas eleições	<i>pass</i>
Eleição termina corretamente na data, hora e minuto marcados	<i>pass</i>
Avaria de um servidor RMI não tem qualquer efeito nos clientes	<i>pass</i>
Consultar resultados detalhados de todas as eleições passadas	<i>pass</i>
Não se perde/duplica votos se os servidores RMI falharem	<i>fail</i>
Não se perde/duplica votos se a comunicação Multicast tiver falhas	<i>fail</i>
Avárias temporárias (<30s) dos 2 RMIs são invisíveis para clientes	<i>pass</i>
Terminal de voto bloqueado automaticamente após 120s sem uso	<i>pass</i>
Crash de terminal de voto é recuperado	<i>fail</i>

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