1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

dei25 departamento de engenharia informática 1995 – 2020

e-Voting

Voto Eletrónico na UC

Relatório realizado no âmbito da unidade curricular Sistemas Distribuídos, inserida no programa curricular do curso de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

2018284515	Ana Rita Rodrigues	analr@student.dei.uc.pt	PL2
2018233092	Dylan Perdigão	dgp@student.dei.uc.pt	PL2

Coimbra

2021

Introdução

No âmbito da unidade curricular Sistemas Distribuídos, foi-nos proposto desenvolver um sistema de voto eletrónico para estudantes, docentes e funcionários da Universidade de Coimbra cujo objetivo é ter um sistema operacional distribuído em várias máquinas que consigam comunicar entre elas. Cada máquina tem o seu propósito para efetuar diversas tarefas no processo de voto eletrônico, nomeadamente o registo das informações na base de dados, criação de eleições com respetivas listas, contas de eleitores, identificação dos eleitores nas mesas de voto e um terminal que permita proceder ao sufrágio. O armazenamento de todos os dados envolvidos nestes processos é feito através de uma base de dados SQLite.

O projeto é constituído pelo código fonte, pelos executáveis (*.jar*), pelo *Javadoc* que explica detalhadamente as classes, interfaces e métodos implementados e de um ficheiro (*.txt*) que contém as instruções de execução do sistema.

Arquitetura de software

Arquitetura do projeto

O código foi dividido em 3 *packages* sendo 2 deles *subpackages* do principal, contendo o servidor RMI, o servidor multicast, o terminal de voto e a consola de administração. Na *package* ***elections*** constam as classes das listas, dos departamentos, das eleições, das pessoas e do registo de votos. Na *package* ***others*** estão classes e interfaces utilitárias para o bom funcionamento dos quatro programas.

Arquitetura do servidor RMI

O servidor RMI serve de intermediário entre os restantes módulos do sistema e a base de dados. Ele é constituído maioritariamente por todos os métodos de inserção, atualização e consulta na base de dados constituída pelas seguintes tabelas:

- ***"candidacy"*** é a tabela relativa às listas que contém um ID, um nome, um tipo (permite diferenciar listas de estudantes, docentes e funcionários), o número de votos e o ID da eleição à qual se refere.
- ***"candidacy_person"*** é uma tabela intermediária que permite efetuar o relacionamento entre as pessoas e as listas, respetivamente através do número de cartão de cidadão e o ID da lista.
- ***"department"*** é a tabela relativa aos departamentos/mesas de voto. Ela é composta por um ID, do nome do departamento e de um booleano que permite saber se o departamento tem servidor multicast.
- ***"election"*** é a entidade referente às eleições, tendo elas um ID, um título (nome), um tipo (igual às listas), uma descrição, uma data de início, uma data de fim e de contadores para os votos brancos e nulos.
- ***"election_department"*** é uma tabela intermediária que efetua o relacionamento entre as eleições e os departamentos através dos seus IDs.
- ***"person"*** é a tabela relativa às pessoas, contendo o nome, o trabalho (sendo este referente aos tipos de listas e eleições), a palavra passe (sendo ela na realidade uma hash da concatenação do número de cartão de cidadão e da palavra passe original, isto é a forma de armazenarmos esta informação sensível de forma minimamente segura), o departamento em que trabalha, o numero de telemovel, a morada, o número de cartão de cidadão e sua respetiva data de validade.
- ***"voting_record"*** é a tabela fraca relativa aos registos de votos, sendo ela composta pela data do voto, pelo departamento onde foi realizado o voto, pelo número de cartão de cidadão do eleitor e pelo ID da eleição à qual se refere.
- ***"voting_terminal"*** é a tabela relativa ao estado dos terminais de voto, composta por um ID, um ID do departamento/mesa de voto que gere o terminal, um booleano que identifica o estado e informações sobre a pessoa e sobre a eleição que está a usar o terminal.

Arquitetura da consola de administração

A consola de administração é composta de um só processo que recebe os comandos e dados introduzidos pelo utilizador.

Arquitetura dos servidores multicast

O servidor multicast é composto por duas threads, uma para fazer pings no seu grupo de multicast e outra para escutar as mensagens provenientes dos terminais de voto. Em paralelo corre o processo principal que espera ações do utilizador.

Arquitetura dos terminais de voto

O terminal de voto corre com uma única thread que fica à escuta de instruções da parte do servidor multicast.

Funcionamento do servidor Multicast

O servidor é configurado com o seu endereço IPv4 e com o ID do departamento onde está em serviço. Ele cria o seu grupo de multicast, liga-se com o socket e fica à escuta de mensagens, nomeadamente de pings dos terminais associados à mesma rede. Depois de iniciado aparecem as eleições em que é possível votar no seu departamento. Depois é pedida a identificação do eleitor, sendo a pesquisa feita por nome, cargo, departamento onde trabalha, número de telemóvel, morada ou número de cartão de cidadão. A informação introduzida é enviada ao RMI para pesquisar na Base de Dados as pessoas correspondentes. Depois de escolher a pessoa, é enviado mensagem pelo protocolo multicast para desbloquear um dos terminais para a pessoa votar com os respetivos dados e listas em que pode votar. Quando o utilizador efetua o login no terminal de voto ele recebe as credenciais por multicast para serem verificadas. É enviada resposta sempre com o protocolo com o sucesso ou insucesso do login. O eleitor pode realizar o seu voto que é recebido novamente por multicast e é registado na Base de Dados pelo RMI.

Protocolo Multicast

O protocolo multicast é composto por pares de chave-valor separados por um “;” e sendo os pares separados por um “|”. O protocolo é constituído pelos seguintes pares chaves-valores obrigatórios

- O valor da chave ***sender*** que permite identificar o remetente pode ser decomposta da seguinte forma:

  {***multicast*** | ***voteterm***} **-** {***ID***} **-** {***ID departamento***}

- O valor da chave ***destination*** que permite identificar o destinatário, ela segue a mesma formatação do sender.

- O valor da chave ***message*** indica o tipo de ação pretendida pelo destinatário.

Mais especificamente podemos distinguir várias categorias de mensagens:

1. O ***“ping”*** é comum ao servidor multicast e ao terminal de voto que permite notificar o destinatário que está a funcionar.
2. A ***“request_id”*** quando é enviada do terminal de voto para o servidor multicast, é acompanhada por ***“required_id”*** (ID desejado), ela permite efetuar o requerimento do ID e registar o terminal caso esse ID não exista. Quando ela é enviada do servidor multicast para o terminal de voto, tem adicionalmente um ***“allowed_id”***, que permite enviar o ID se ele for aceite ou *not_available* caso contrário.
3. O ***“findTerminals”*** é enviado do servidor multicast quando inicializado para todos os terminais de voto no grupo multicast, para descobrir os terminais que já se encontravam ativos antes do servidor se ligar. O objetivo principal desta funcionalidade é permitir a recuperação do servidor multicast de forma invisível para os terminais de voto.
4. O ***“found”*** é enviado como resposta à mensagem especificada no ponto 3 para informar o servidor multicast que o terminal de voto foi encontrado no grupo de multicast.
5. O ***“login”*** é enviado do terminal de voto para o servidor multicast, ele é acompanhado das chaves **“username”** e **“password”** que permitem enviar

respetivamente o número de cartão de cidadão e o código de acesso para verificar se o login está correto.

6. O **"*vote"*** é enviado do terminal de voto para o servidor multicast, e permite enviar o voto do eleitor, ele é acompanhado com ***"id_candidacy"*** (ID da lista em que votou), "***id_election"*** (ID da eleição), ***"cc***" (número de cartão de cidadão do eleitor) e ***"dep"*** (ID do departamento onde o eleitor votou).
7. O ***"identify"*** é enviado do servidor de multicast para o terminal de voto para enviar o ***"cc"*** (número de cartão de cidadão) e informações sobre a eleição, nomeadamente o ***"election"*** (ID da eleição), ***"arrayList"*** (com as listas e o voto branco e nulo), ***"arraylds"*** (com os IDs das listas). Isto acontece sempre que um terminal deve ser desbloqueado para um eleitor especificado.
8. O ***"voteOk"*** é enviado do servidor de multicast para o terminal de voto com o **cc** (número de cartão de cidadão) e ***"election"*** (ID da eleição) para confirmar a receção do voto, referido no ponto 2.
9. O ***"logged in"*** confirma o sucesso do login efetuado no terminal de voto e envia o ***"cc"*** (número de cartão de cidadão) do servidor multicast para o terminal de voto, para que o terminal possa confirmar que esta mensagem se refere ao utilizador em questão.
10. O ***"wrong password"*** é enviado para o terminal de voto caso o login seja inválido.
11. O ***"timeout"*** é enviado do terminal de voto para o servidor multicast quando o utilizador fica 60 segundos sem realizar o voto.

Funcionamento do servidor RMI

Quando um RMI se liga, ele tenta enviar pings via UDP para um servidor RMI, quando não recebe resposta conta uma falha. Quando chega às 5 falhas considera que o RMI está morto e portanto deve ser ele a tornar-se o primário. Quando o servidor é primário, ele lê e responde aos pings enviados pelo servidor secundário. Os pings deixam de ser perdidos e deixam de contar como falhas para qualquer RMI secundário que esteja a testar.

Como mencionado anteriormente o servidor RMI primário efetua todas as operações com a base de dados tendo ele métodos específicos para todos os tipos de consulta, atualização e inserção de dados.

Quando se conecta ao servidor RMI pela primeira vez, o servidor multicast envia um objeto remoto que é guardado no servidor RMI como callback. Este callback é usado para que o servidor RMI possa pingar regularmente o servidor multicast, verificando o seu estado de funcionamento.

As consolas de administração quando pretendem receber informação em tempo real enviam um callback que o servidor RMI usa para imprimir informação do lado da consola de administração.

Distribuição de tarefas

O trabalho foi distribuído pela paridade das tarefas, ou seja um elemento ficou com as funcionalidades identificadas com um número par e o segundo elemento foi encarregado das tarefas com número ímpar, isto é um total de 7 tarefas para cada um. Em paralelo das funcionalidades foi preciso dividir o trabalho para a implementação das ligações entre os diferentes servidores/clientes e comunicações com a base de dados e respetivas estruturas de dados. Finalmente no processo de debugging dividimos as tarefas de forma a ter uma pessoa a escrever a Javadoc e outra a resolver alguns bugs existentes, sendo que no final ambos trabalharam no processo de testes à plataforma descritos no próximo parágrafo.

Testes feitos à plataforma

Funcionalidade	Testes realizados	Pass/Fail
Menus da Consola de administração	No **Menu Principal**, introduzir uma opção inválida	
	No "**Registar Pessoas**", introduzir uma opção inválida	
	No "**Criar Eleicao**", introduzir uma opção inválida	
	No "**Gerir Eleição**", introduzir uma opção inválida	
	No "**Gerir Mesas de Voto**", introduzir uma opção inválida	
	No "**Local em que cada eleitor votou**", introduzir uma opção inválida	
	No "**Consultar resultados detalhados de eleições passadas**", introduzir uma opção inválida	
	No "**Consultar estado das mesas de voto e respetivos terminais de voto**", introduzir uma opção inválida	
Registar novo utilizador	É impossível registar 2 utilizadores com o mesmo número de cartão de cidadão	
	O número de telemóvel tem 9 dígitos e começam por 91, 92, 93 ou 96	
	A validade do cartão de cidadão é superior à data atual	
	O utilizador só pode escolher o departamento em que frequenta dentro de 11 opções possíveis, que lhe serão apresentadas	
	O utilizador só pode escolher o cargo que ocupa dentro de 3 opções possíveis que lhe são apresentadas	
	Para efeitos de segurança, aquando do registo de um utilizador, será aplicada uma hash à concatenação do número de cidadão com a password	
Criar eleição	Data de fim é sempre superior à data de início	
	Não permite inserir um título vazio	

	Não permite inserir uma descrição vazia	
Gerir Listas de Candidatos a uma eleição	Apenas se pode adicionar candidatos caso a eleição ainda não esteja a decorrer	
	Apenas se podem adicionar candidatos cujo tipo de lista corresponda ao trabalho da pessoa	
	Apenas se podem adicionar candidatos cujo departamento onde a eleição é restringida seja onde ele trabalha	
Gerir eleições	Apenas se pode editar eleições que ainda não estejam a decorrer	
Criar mesas de voto	Não podem ser adicionadas mesas de voto a eleições que estão restringidas a um único departamento	
	Só é possível inserir e remover mesas de voto a eleições que ainda não começaram	
	Não é possível inserir mesas de voto a eleições que estejam a decorrer	
	Ao iniciar uma mesa de voto, é logo atribuído um departamento	
	Há um número limitado de mesas de voto que se pode ligar	
	Por default quando uma mesa de voto é criada, são listadas todas as eleições que não têm departamento restringido	
Identificar eleitor na mesa de voto e desbloquear um terminal de voto	O terminal de voto apenas fica desbloqueado caso o eleitor ainda não tenha votado antes naquela eleição	
	Pode-se identificar a pessoa por qualquer campo	
	Permite pesquisar por partes do nome e morada (regex)	
	O terminal de voto fica bloqueado passados 60s de inatividade	
Login de eleitor no terminal de voto	Quando um terminal de voto é desbloqueado apenas à pessoa que se identificou na mesa de voto pode votar	
	Caso password errada voltar a pedir para inserir	
	Caso password nula volta a pedir para inserir	

Votar	Quando uma eleição não está restringida a um determinado departamento, o eleitor pode votar em qualquer mesa de voto	
	Apenas estudantes votam em listas de estudantes, docentes em listas de docentes e funcionários em listas de funcionários	
	Quando uma eleição não tem listas não deve ser possível votar	
	Apenas dá para votar em eleições a decorrer	
Editar propriedades de uma eleição	Editar apenas eleições que não estejam a decorrer	
Saber em que local votou cada eleitor	Por questões de confidencialidade, não se sabe em que lista votou cada eleitor	
	Print da data e local onde votou cada eleitor	
Consola de administração mostra mesas de voto on/off e votantes	Print em tempo real de mesas de voto e respetivos terminais de voto que se ligam e desligam em tempo real	
Consola de administração atualizada em tempo real nas eleições	Print dos resultados em tempo real apenas de eleições a decorrer	
Consultar resultados detalhados de todas as eleições passadas	Apenas é possível consultar resultados de eleições que já tenham terminado	
	Os resultados contêm o número total de votos de cada lista e o número de nulos e brancos, bem como as suas percentagens	
Outros	Caso uma mesa de voto vá abaixo o terminal de voto consegue enviar o voto, quando ele se reconecta	
	Sempre que há avarias no servidor primário RMI o servidor multicast consegue reconectar-se sem quaisquer problemas visíveis	
	Sempre que há avarias no servidor primário RMI a consola de administração continua a receber updates em tempo real	

	Sempre que há avarias no servidor primário RMI a consola de administração consegue reconectar-se sem quaisquer problemas visíveis	
	Crash do terminal de voto recuperado, sempre que um terminal de voto vai abaixo é possível voltar a ligá-lo no mesmo sítio e com a mesma informação de quando se desligou	
	Quando um servidor multicast vai abaixo o RMI é notificado através de um callback, enviando-lhe constantemente pings a verificar se está a correr, e dá update na base de dados	
Falhas de rede	Terminais de voto tentam restabelecer a ligação todos os 5 segundos	
	Mesas de voto tentam restabelecer a ligação todos os 5 segundos	

Conclusão

Com base no que foi apresentado, e com a primeira meta do projeto concluída, é de realçar que foram atingidos os principais objetivos do funcionamento do nosso sistema distribuído.

Como foi referido anteriormente, a aplicação tem a finalidade de ser usada como ferramenta de voto eletrónico na comunidade estudantil da Universidade de Coimbra.

Posto isto, foi um projeto que nos proporcionou um conhecimento mais profundo no mundo da programação, mais especificamente da linguagem Java, e do qual retirámos competências que certamente nos vão proporcionar um melhor futuro como Engenheiros Informáticos.