



FCTUC

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Sistemas Distribuidos

Meta 1



DROP MUSIC

Trabalho Realizado por:

| | |
|------------------|------------|
| -João Figueiredo | 2015248636 |
| -Pedro Chicória | 2015262771 |

Índice

| | |
|------------------------------|----|
| Sistemas Distribuidos Meta 1 | 1 |
| Índice | 2 |
| Rmi client | 3 |
| Rmi Server | 3 |
| Multicast Server | 3 |
| Protocolo | 5 |
| Callbacks | 9 |
| Tratamento de exceções | 9 |
| Failover | 9 |
| Distribuição de trabalho | 9 |
| Checklist | 10 |

Rmi client

O *rmi client* tem como objectivo estabelecer uma porta de comunicação entre o utilizador da nossa aplicação e o *rmi server*. No *rmi client* encontram-se todos os menus da nossa aplicação, neles é possível realizar diversas operações tais como a inserção de informação relativas a albums, artistas, musicas e playlists. É ainda possível realizar sobre esses objectos operações de edição, consulta e remoção.

O *rmi client* realiza essas operações recorrendo a métodos remotos que se encontram localizados no *server interface*, estes métodos são a forma como o *rmi client* comunica com o *rmi server*.

Rmi Server

O *rmi server* implementa os métodos remotos especificados no *server interface*, é dentro destes métodos que a comunicação é realizada entre o *rmi server* e o *multicast server*.

Estes comunicam com recurso a duas multicast sockets e a datagram packets, onde uma das quais será responsável por enviar informação do *rmi server* para o *multicast server* e outra pela recepção da resposta. Esta comunicação segue um protocolo que a semelhança do enunciado do mesmo segue um padrão de **chave | valor**.

Multicast Server

O multicast é responsável por receber os pedidos das funções remotas do *rmi server* e tratar de os satisfazer. O servidor multicast têm acesso a todas as classes de objectos usados para modelar as necessidades de construção da nossa aplicação. As classes criadas foram as seguintes:

- User
 - private String username;
 - private String password;
 - private boolean privilege;
 - private ArrayList<Playlist> playlists;
- Album

```
-private String album_name;  
-private String record_label;  
-private ArrayList<Review> reviews;  
-private ArrayList<Music> musics;
```

- Artist

```
-private String artist_name;  
-private int age;  
-private String music_style;  
-private String biography;  
-private ArrayList<Album> albums;  
-private ArrayList<String> usernamesToNotificate;
```

- Music

```
-private String music_name;  
-private String musical_gender;  
-private String lyrics;  
-private String composer;  
-private String release_date;
```

- Playlist

```
-private String name;  
-private String creator;  
-private boolean privilege;  
-private ArrayList<Music> musics;
```

- Review

```
-private String username;  
-private String text;  
-private int rating;
```

Para armazenar os dados foram criadas duas ArrayLists, uma da classe Users e outra da classe Artist. Com estas duas classes é nos possível obter qualquer informação que o utilizador possa solicitar. Estas são guardas posteriormente sob a forma de um ficheiro de objectos. O multicast retorna então uma mensagem sobre o formato de string ao *rmi server* seguindo o protocolo estabelecido.

Protocolo

Todas as variâncias do protocolo aceites entre o *rmi server* e o *multicast* serão especificadas de seguida.

rmi server -> multicast server

- "type|check_login;username|" + username + ";password|" + password
 - "type|check_user_privilege;username|" + username
 - "type|check_user;username|" + username
 - "type|check_artist;artist_name|" + artist_name
 - "type|check_album;album_name|" + album_name
 - "type|check_music;music_name|" + music_name
 - "type|check_review;album_name|" + album_name + ";username|" + username
 - "type|check_playlist;playlist_name|" + playlist_name
 - "type|check_user_playlist;playlist_name|" + playlist_name + ";username|" + username
- username
- "type|insert_user;username|" + username + ";password|" + password
 - "type|insert_user_privilege;username|" + username
 - "type|insert_artist;artist_name|" + artist_name + ";artist_age|" + artist_age + ";artist_music_style|" + artist_music_style + ";artist_biography|" + artist_biography
 - "type|insert_album;album_name|" + album_name + ";album_record_label|" + album_record_label + ";artist_name|" + artist_name
 - "type|insert_music;music_name|" + music_name + ";music_gender|" + music_gender + ";music_lyrics|" + music_lyrics + ";music_composer|" + music_composer + ";music_release_date|" + music_release_date + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|insert_review;username|" + username + ";description|" + description + ";rating|" + rating + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|insert_playlist;playlist_name|" + playlist_name + ";playlist_privilege|" + playlist_privilege + ";creator|" + creator
 - "type|insert_music_playlist;playlist_name|" + playlist_name + ";music_name|" + music_name + ";album_name|" + album_name + ";artist_name|" + artist_name + ";creator|" + creator
 - "type|edit_user;user_to_edit|" + user_to_edit + ";username|" + username + ";password|" + password + ";privilege|" + privilege

-
- "type|edit_artist;artist_to_edit|" + artist_to_edit + ";artist_name|" + artist_name + ";artist_age|" + artist_age + ";artist_music_style|" + artist_music_style + ";artist_biography|" + artist_biography
 - "type|edit_album;album_to_edit|" + album_to_edit + ";album_name|" + album_name + ";album_record_label|" + album_record_label + ";artist_name|" + artist_name
 - "type|edit_music;music_to_edit|" + music_to_edit + "music_name|" + music_name + ";music_gender|" + music_gender + ";music_lyrics|" + music_lyrics + ";music_composer|" + music_composer + ";music_release_date|" + music_release_date + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|edit_review;review_to_edit|" + review_to_edit + "username|" + username + ";description|" + description + ";rating|" + rating + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|edit_playlist;playlist_to_edit|" + playlist_to_edit + ";playlist_name|" + playlist_name + ";playlist_privilege|" + playlist_privilege + ";creator|" + creator
 - "type|remove_user;username|" + username
 - "type|remove_artist;artist_name|" + artist_name
 - "type|remove_album;album_name|" + album_name + ";artist_name|" + artist_name
 - "type|remove_music;music_name|" + music_name + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|remove_review;album_name|" + album_name + ";username|" + username + ";artist_name|" + artist_name
 - "type|remove_playlist;playlist_name|" + playlist_name + ";username|" + username
 - "type|search_user"
 - "type|search_artist"
 - "type|search_album"
 - "type|search_album_by_name;album_name|" + album_name
 - "type|search_album_by_artist;artist_name|" + artist_name
 - "type|search_music"
 - "type|search_music_by_album;album_name|" + album_name + ";artist_name|" + artist_name
 - "type|search_music_by_playlist;playlist_name|" + playlist_name + ";creator|" + creator
 - "type|search_public_playlist"

-
- "type|search_private_playlist;username|" + username
 - "type|details_user;username|" + username
 - "type|details_artist;username|" + artist_name
 - "type|details_album;album_name|" + album_name + ";artist_name|" + artist_name
 - "type|details_music;music_name|" + music_name + ";album_name|" + album_name + ";artist_name|" + artist_name
 - "type|details_review;album_name|" + album_name + ";artist_name|" + artist_name

multicast server -> rmi server

- type|check_login;status|on
- type|check_user_privilege;status|on
- type|check_user;status|on
- type|check_artist;status|on
- type|check_album;status|on
- type|check_music;status|on
- type|check_review;status|on
- type|check_playlist;status|on
- type|check_user_playlist;status|on
- type|insert_artist;status|on
- type|insert_user;status|on
- type|insert_user_privilege;status|on
- type|insert_album;status|on
- type|insert_review;status|on
- type|insert_music;status|on
- type|insert_playlist;status|on
- type|insert_music_playlist;status|on
- type|edit_user;status|on
- type|edit_album;status|on
- type|edit_artist;status|on
- type|edit_music;status|on
- type|edit_review;status|on
- type|edit_playlist;status|on
- type|remove_user;status|on

- type | remove_artist;status | o
- type | remove_album;status | on
- type | remove_music;status | on
- type | remove_review;status | on
- type | remove_playlist;status | on
- type | search_user;list_count | " + Integer.toString(users.size()) + ";user_" + Integer.toString(counter) + "|" + user.getUsername()
 - type | search_artist;list_count | " + Integer.toString(artists.size()) + ";artist_" + Integer.toString(counter) + "|" + artist.getArtist_name()
 - type | search_album" + ";list_count | + ;album_" + Integer.toString(counter) + "|" + album.getAlbum_name() type | search_album_by_name;item_count | + album_" + Integer.toString(counter) + "|" + album.getAlbum_name()
 - type | search_music;list_count | + ;music_" + Integer.toString(counter) + "|" + music.getMusic_name()
 - type | search_music_by_album;list_count | + ;music_" + Integer.toString(counter) + "|" + music.getMusic_name()
 - type | search_user;list_count | + ;music_" + Integer.toString(counter) + "|" + music.getMusic_name() type | search_public_playlist;list_count | + ;playlist_" + Integer.toString(counter) + "|" + playlist.getName() type | search_private_playlist;list_count | + ;playlist_" + Integer.toString(counter) + "|" + playlist.getName()
 - type | details_user+ ";password |" + user.getPassword() + ";privilege |" + user.getPrivilege().toString()
 - type | details_artist+ ;age |" + Integer.toString(artist.getAge()) + ";music_style |" + artist.getMusic_style() + ";biography |" + artist.getBiography();
 - type | album_details + ";record_label |" + album.getRecord_label()
 - type | album_music + ;musical_gender |" + music.getMusical_gender() + ";lyrics |" + music.getLyrics() + ";composer |" + music.getComposer() + ";release_date |" + music.getRealease_date();
 - type | reviews_details;list_count | + ;username |" + review.getUsername()+ ";description |" + review.getText(); + ";rating |" + Integer.toString(review.getRating());

Callbacks

Foi criada no *rmi server* uma *ArrayList* da classe *client interface*, para verificar quais os utilizadores que se encontram online a cada momento. Desta forma é possível notificá-los, assim que o seu privilegio dentro da aplicação ou alguma alteração nos dados da aplicação desde que para esta segunda opção eles tenham privilegio de editor.

Tratamento de excepções

Sempre que ocorrer alguma excepção no *rmi client*, este irá tentar fazer *lookup* do servidor *rmi* para detectar qual é que se encontra activo e se re-conectar. Desta forma asseguramos que para avarias menores a 30 segundos o *client* não se aperceba que um dos servidores *rmi* falhou.

Failover

O servidor *rmi* esta constantemente a realizar *pings* de forma a se assegurar que o segundo servidor *rmi* esta ligado. Caso um dos servidores *crash* o outro irá passar a assumir o papel de servidor primário.

Distribuição de trabalho

O trabalho foi dividido da seguinte forma pelos elementos do grupo:

- Pedro Chicória - Responsável pelo servidor *Rmi*
- João Figueiredo - Responsável pelo servidor *Multicast*

Checklist

| | |
|--|---|
| Requisitos Funcionais | |
| Registar novo utilizador | ✓ |
| Login protegido com password | ✓ |
| Introduzir artistas, álbuns e músicas | ✓ |
| Pesquisar álbuns por artista e por título de álbum | ✓ |
| Consultar detalhes de álbum (incluindo músicas e críticas) | ✓ |
| Editar detalhes de álbum (incluindo músicas) | ✓ |
| Escrever crítica sobre um álbum (com pontuação) | ✓ |
| Consultar detalhes de artista (e.g., discografia, biografia) | ✓ |
| Dar privilégios de editor a um utilizador | ✓ |
| Notificação imediata de privilégios de editor (online users) | ✓ |
| Notificação imediata de re-edição de descrição de álbum (online users) | ✓ |
| Entrega posterior de notificações (offline users) | |
| Upload de ficheiro para associar a uma música existente | ✓ |
| Partilhar um ficheiro musical e permitir o respetivo download | ✓ |
| Grupos de 3: Criar grupo de amigos (-3) | |
| Grupos de 3: Owner de cada grupo gere os membros (-3) | |
| Grupos de 3: A partilha de informação está limitada aos grupos (-3) | |
| Grupos de 3: Editores que não são owners podem alterar dados (-3) | |
| Tratamento de Exceções | |
| Avaria de um servidor RMI não tem qualquer efeito nos clientes | ✓ |
| Não se perde/duplica músicas se os servidores RMI falharem | |
| O serviço funciona desde que haja um servidor multicast disponível | ✓ |
| Avarias temporárias (<30s) dos N servidores são invisíveis para clientes | ✓ |
| Pedidos são sempre processados por $N \geq 1$ servidores multicast | ✓ |
| Pedidos de leitura são respondidos apenas por um servidor multicast | |
| Failover | |
| Servidor RMI primário testa periodicamente o secundário | ✓ |
| Em caso de avaria longa os clientes RMI ligam ao secundário | ✓ |
| Servidor RMI secundário substitui o primário em caso de avaria longa | ✓ |
| O failover é invisível para utilizadores (não perdem a sessão) | ✓ |
| O servidor original, quando recupera, torna-se secundário | ✓ |
| Relatório | |
| Arquitetura de software detalhadamente descrita | ✓ |
| Detalhes do funcionamento do servidor UDP (protocolo multicast, etc.) | ✓ |
| Detalhes do funcionamento do servidor RMI (interface, javadocs, etc.) | ✓ |
| Distribuição de tarefas pelos elementos do grupo | ✓ |
| Testes de software (tabela com descrição + pass/fail de cada teste) | ✓ |