

DropMusic: arquivo de música com partilha de ficheiros

Sistemas Distribuídos 2018/19 — Meta 2 — 17 de dezembro de 2018 (21:59)

Resumo

Este projeto tem como objetivo criar um sistema de gestão e partilha de músicas com funcionalidades semelhantes aos arquivos AllMusic.com e IMDb.com, acrescentando a possibilidade de partilhar ficheiros entre utilizadores tal como o serviço Dropbox. O sistema deverá ter todas as informações relevantes sobre as músicas, tais como os autores, álbuns, grupos musicais, compositores, letras, concertos, e outros que considere importantes. Ao consultar um álbum, um utilizador obtém a listagem de músicas com as respetivas informações, críticas escritas por outros utilizadores, bem como informações gerais sobre os géneros musicais e a data de lançamento. Apenas os editores podem acrescentar e alterar informações sobre álbuns e músicas, sendo que os restantes utilizadores podem apenas consultar essas informações. É possível a cada utilizador transferir ficheiros musicais (MP3, FLAC, etc.) que ficarão associados à sua própria conta mas que poderão ser partilhados com outros utilizadores.

1 Objetivos do projecto

No final do projeto cada estudante deverá ter:

- Desenvolvido uma interface Web para um sistema de gestão e partilha de músicas.
- Ter integrado a interface Web com a aplicação desenvolvida na primeira meta.
- Dominado Struts2, JavaServer Pages e JavaBeans.
- Seguido a arquitetura MVC para desenvolvimento Web.
- Aplicado WebSockets para comunicar assincronamente com os clientes.
- Integrado a aplicação com serviços REST externos, usando OAuth.

2 Visão geral

Nesta segunda meta do projecto, os estudantes irão criar um *frontend* Web para a aplicação DropMusic. Esta nova interface irá possibilitar que os utilizadores acessem ao serviço a partir de quase qualquer dispositivo com Internet no planeta, sem necessitar de instalação de software cliente. Como a interoperabilidade é um requisito muito importante, utilizadores Web deverão aceder à mesma informação que os utilizadores

na aplicação desktop. Para tal, o servidor Web deverá comunicar com o servidor RMI desenvolvido na meta 1.

Os utilizadores deverão ter as mesmas funcionalidades, independentemente da interface que usem. Portanto a interface Web deverá listar álbuns, músicas, *etc.*, e deverá permitir a inserção de informações e de críticas. Como o aspeto interativo é muito importante na Web, o vosso projeto deverá mostrar algumas alterações em tempo real, nomeadamente ao conceder privilégios de editor e ao escrever uma crítica, através de notificações que envolvem a meta 1. Como os utilizadores estão cada vez mais exigentes, técnicas menos robustas como meta-refresh e iframes ocultas não serão consideradas.

Finalmente, as aplicações de hoje em dia não existem isoladas umas das outras. Através de APIs REST e OAuth, irão integrar a vossa aplicação com um serviço externo: a Dropbox. Deverá ser possível associar ficheiros musicais que estejam na Dropbox a músicas existentes no DropMusic e posteriormente partilhá-las com outros utilizadores.

3 Arquitetura

A Figura 1 mostra a arquitetura geral do projeto. Os elementos a amarelo referem-se à segunda meta do projecto, enquanto os outros se referem à primeira meta. O servidor Web deverá ligar-se por RMI ao servidor de dados, garantindo a interoperabilidade com os clientes da primeira meta.

Devem desenvolver uma aplicação Web que corra num servidor HTTP (Apache Tomcat) e que atue como um cliente RMI para com o servidor RMI. Os clientes irão usar browsers para se ligarem ao servidor Web para pedirem páginas através de HTTP. Para melhorar a experiência de utilização poderão ponderar fazer alguns dos pedidos via AJAX em vez de carregar a página toda.

A comunicação real-time para o browser deverá ser construída usando WebSockets. Isto inclui notificações e alterações de valores em tempo real.

4 Interface Web

Pretende-se criar uma aplicação Web que disponibilize as mesmas funcionalidades da meta 1 através da Internet. Usando uma arquitetura MVC, deverão implementar os seguintes requisitos funcionais usando Struts2:

1. **Registar pessoas.** Os utilizadores devem poder registar-se, sendo que o acesso à aplicação deve estar protegido com username e password. Deverá guardar toda a informação pessoal que considere necessária bem como uma password (código de acesso). Para simplificar, considere que o primeiro utilizador a registar-se tem automaticamente privilégios de editor em todo o sistema.
2. **Gerir artistas, álbuns e músicas.** A informação está organizada por artistas, álbuns e músicas, sendo que apenas os editores podem inserir, alterar e remover dados relativos a estas entidades. Os restantes utilizadores podem apenas ler a informação existente.

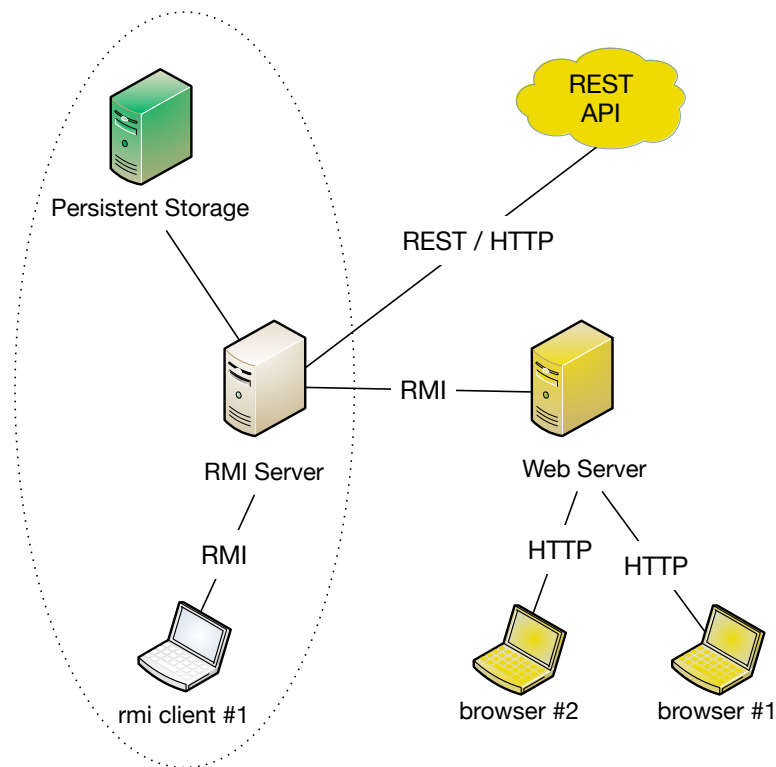


Fig. 1: Arquitetura do projecto

3. **Pesquisar músicas.** Um utilizador deve poder pesquisar por álbuns e por artistas introduzindo dados parciais tais como o nome do artista, o género musical ou o título do álbum. O resultado da pesquisa é uma lista de itens na qual se pode escolher um item para obter mais detalhes.
4. **Consultar detalhes sobre álbum e sobre artista.** Deverá ser possível consultar todos os detalhes de um álbum, tais como a descrição, a lista de músicas, críticas escritas por outros utilizadores, a pontuação média atribuída pelos utilizadores, etc. A lista de músicas deverá mostrar informações tais como título, compositor, duração, etc. Deve ser também possível aceder aos detalhes sobre cada artista.
5. **Escrever crítica a um álbum.** Qualquer utilizador (editor ou não) deverá poder escrever uma crítica a um álbum, que consiste numa pontuação e numa justificação textual que não exceda 300 caracteres. Todas as críticas a um dado álbum aparecem listadas ao consultar os detalhes desse álbum, sendo igualmente apresentada a pontuação média.
6. **Dar privilégios de editor a um utilizador.** Aos utilizadores que tenham privilégios de *editor* é possível conceder esses mesmos privilégios a outros utilizadores. Para tal, escolhem o utilizador que pretendem promover a editor e esse utilizador passa a poder alterar informações no sistema.

7. **Criar grupo de amigos.** Trabalhos apresentados por 3 alunos têm o requisito de criação e gestão de grupos de amigos. Os grupos têm owners, que confirmam ou recusam a pertença ao grupo após pedido. O owner inicial é o utilizador que criou o grupo, sendo que ele pode adicionar mais owners. **Grupos de 3 estudantes.**
8. **O owner de cada grupo faz a gestão dos membros.** É possível a um owner de um grupo adicionar e remover utilizadores do grupo, confirmando e recusando a pertença ao grupo após pedido. **Grupos de 3 estudantes.**
9. **A partilha de informação é limitada aos membros de cada grupo.** O objetivo dos grupos é permitir partilha limitada ao grupo. Quando nova informação é adicionada por um utilizador, este pode indicar se a partilha é pública ou restringida a um ou mais dos grupos a que pertence. **Grupos de 3 estudantes.**
10. **Dentro de cada grupo há editores que podem não ser owners.** Em qualquer grupo existe o papel de editor (um owner é editor por defeito, mas um editor pode não ser owner). Apenas os editores podem acrescentar e alterar informações sobre álbuns e músicas, sendo que os restantes utilizadores podem apenas consultar essas informações. **Grupos de 3 estudantes.**

5 Notificações em tempo real

De forma a que as páginas da vossa aplicação sejam atualizadas instantaneamente (server push), deverão usar WebSockets para fazer push de informação para o cliente assim que esteja disponível. Deverão usar WebSockets para:

- **Notificação imediata de privilégios de editor.** Quando um utilizador é promovido a editor deve receber imediatamente (em tempo real) essa informação se estiver ligado à aplicação, sem precisar de realizar nenhuma operação.
- **Notificação imediata de re-edição de descrição textual.** A descrição textual incluída nos detalhes de um álbum (ou de um artista) pode ser alterada por um editor, tal como especificado anteriormente. Sempre que um editor altere uma descrição textual todos os editores anteriores são notificados de imediato se estiverem ligados à aplicação. Por exemplo, um editor que altere a descrição de um álbum será notificado sempre que essa mesma descrição seja novamente alterada por outro editor qualquer.
- **Atualização imediata da pontuação média das críticas.** Qualquer utilizador (editor ou não) que escreva uma crítica a um álbum classifica-o com uma pontuação. A pontuação média das críticas a um álbum deve ser atualizada nos browsers de todos os clientes que estiverem a consultar esse mesmo álbum. Vale pontos extra atualizar também o texto das diversas críticas a álbuns que forem feitas. Este requisito não deve integrar com a meta 1, ou seja, aplica-se apenas a clientes que estejam ligados através de browsers.

6 Integração com serviço REST

Este projeto deverá ser integrado com a Dropbox. A Dropbox será usada para dinamizar a partilha de músicas, bem como fornecer uma alternativa ao login por username e password. Para usar a API da Dropbox é necessário usar autenticação OAuth. Para tratar deste processo devem usar a biblioteca Scribejava. Não serão aceites bibliotecas que façam a integração em Java com a Dropbox.

As funcionalidades a implementar com recurso a REST são:

- **Associar conta à Dropbox.** Qualquer utilizador com login efetuado poderá associar a sua conta no sistema à sua conta da Dropbox. Este passo permitirá depois fazer login com a conta da Dropbox e aceder aos ficheiros que lá se encontram.
- **Associar ficheiro contido na Dropbox a uma música existente.** Deverá ser possível a um utilizador escolher uma música existente no DropMusic e associar a essa música um ficheiro que esteja na sua diretoria da Dropbox.
- **Partilhar música através da Dropbox.** Um utilizador que tenha associado ficheiros a músicas pode escolher utilizadores individuais com os quais pretenda partilhar esses mesmos ficheiros. Ao aceitarem a partilha, os utilizadores ficam com as músicas partilhadas na sua própria diretoria Dropbox (*i.e.*, a partilha de ficheiros é feita através da Dropbox).
- **Playback de uma música diretamente no browser.** Um utilizador que tenha acesso a ficheiros musicais que estejam associados a músicas no DropMusic deverá poder fazer playback diretamente no browser.
- **Login com a conta da Dropbox.** Um utilizador que ainda não tenha feito login, deverá poder fazê-lo com o Dropbox, entrando automaticamente na sua conta DropMusic sem ter de inserir username ou password.

6.1 Relatório

Devem reservar tempo para a escrita do relatório no final do projeto, tendo em conta os passos anteriores. Devem escrever o relatório de modo a que um novo colega que se junte ao grupo possa perceber a solução criada, as decisões técnicas e possa adicionar novos componentes ou modificar os que existem. **O relatório pode ser inteiramente escrito em Javadoc no código-fonte apresentado pelos estudantes.** Deve incluir:

- Arquitetura de software detalhadamente descrita. Deverá ser focada a estrutura de models, views, controllers, processos, threads e sockets usadas, bem como a organização do código.
- Detalhes sobre a integração do Struts2 com o Servidor RMI da primeira meta.
- Detalhes sobre a programação de WebSockets e a sua integração com o servidor RMI.
- Detalhes sobre a integração com o serviço REST.
- Descrição dos testes feitos à plataforma.

7 Entrega do projeto

O projeto deverá ser entregue num arquivo ZIP. Esse arquivo deverá conter um ficheiro README.TXT com toda a informação necessária para instalar e executar o projeto sem a presença dos alunos. Projetos sem informações suficientes, que não compilem ou não executem corretamente **não serão avaliados**.

Dentro do ficheiro ZIP deverá também estar um Javadoc/PDF/HTML com o relatório. O relatório deve seguir a estrutura fornecida, dado que a avaliação irá incidir sobre cada um dos pontos. Também no ficheiro ZIP deverá existir um ficheiro WAR com a aplicação Web pronta a executar, bem como os deliverables da meta anterior prontos a correr.

Finalmente, o ficheiro ZIP deverá ter também **uma pasta com o código fonte completo do projeto**. A ausência deste elemento levará à anulação do projeto.

O ficheiro ZIP com o projeto deverá ser entregue na plataforma Inforestudante até ao dia **17 de dezembro de 2018 (21:59)**, via <https://inforestudante.uc.pt>