

Projeto de Sistemas Distribuídos

Struts2, WebSockets, RMI, OAuth

Gonçalo Amaral - 2015249122

Henrique Ferrer - 2016250818

Arquitetura de software

Para esta meta, a nova arquitetura é constituída por um server Struts2, WebSocketServer, RMI Server, MulticastServer e Client.

Desenvolvemos uma aplicação web que corre num servidor HTTP (Tomcat), este server comunica com o Server RMI que por consequente comunica com o servidor de dados por multicast.

Este servidor web está dividido por controladores, views, rotas etc, sendo estes primeiros a base para a funcionalidade deste webserver. As rotas definem quais os “links” que podem ser utilizados e a que controlador e ação estão relacionada a esse link. Os controladores fazem a recolha de informação e lógica das funcionalidades da aplicação e as views são necessárias para a apresentação gráfica no browser. A utilização de uma arquitetura MVC tem várias vantagens uma delas a facilidade na leitura, manutenção e desenvolvimento de novas features.

Integração com RMI

Para reutilizar os métodos desenvolvidos para a meta 1, foi feito um Service que integra as Interfaces. Foi criado um singleton para reduzir a utilização de memória e pedidos ao Registry.

Integração com WebSockets e RMI

Criamos um WebSocket Server que comunica com o RMI server por tcp (protocolo) e um websocket client em javascript. Desta forma conseguimos integrar as notificações em real-time.

REST e REST WebServices

Para toda a aplicação são feitos pedidos GET e POST para as routes dependendo do que queremos fazer, ou seja caso queiramos obter um view, por exemplo, fazemos um GET request e caso queiramos fazer submit de um form podemos realizar um pedido POST. Isto ajuda muito também na integração de third-party authentication tools.

Testes de software

100		Nota Final:	100				
50	Requisitos Funcionais		50				
5	Registar novo utilizador**		5				
5	Login protegido com password (acesso a todas as páginas)		5				
5	Introduzir artistas, álbuns e músicas		5				
5	Remover artistas que não tenham álbuns nem músicas <---		5				
5	Pesquisar álbuns por artista e por título de álbum		5				
5	Consultar detalhes de álbum (incluindo músicas e críticas)**		5				
5	Editar detalhes de álbum (incluindo músicas)		5				
5	Escrever crítica sobre um álbum (com pontuação)		5				
5	Consultar detalhes de artista (e.g., discografia, biografia)		5				
5	Dar privilégios de editor a um utilizador**		5				
0	Grupos de 3: Criar grupo de amigos (-5)						
0	Grupos de 3: Owner de cada grupo gere os membros (-5)						
0	Grupos de 3: A partilha de informação está limitada aos grupos (-5)						
0	Grupos de 3: Editores que não são owners podem alterar dados (-5)						
15	WebSockets		15				
5	Notificação imediata de privilégios de editor**		5				
5	Notificação imediata de re-edição de descrição textual**		5				
5	Atualização imediata da pontuação média das críticas		5				
25	REST		25				
5	Associar conta de utilizador à Dropbox		5				
5	Associar ficheiro contido na Dropbox a uma música existente		5				
5	Partilhar música através da Dropbox		5				
5	Tocar uma música diretamente na página do DropMusic		5				
5	Login com a conta da Dropbox		5				
10	Relatório		10				
2	Arquitetura do projeto Web detalhadamente descrita		2				
2	Integração de Struts2 com o servidor RMI		2				
2	Integração de WebSockets com Struts2 e RMI		2				
2	Integração de REST WebServices no projeto		2				
2	Testes de software (tabela: descrição e pass/fail de cada teste)		2				
	Extra (até 5 pontos)		0				
	Utilização de HTTPS (4 pts)		ngrok, heroku				
	Utilização em smartphone ou tablet (2pts)		ngrok				
	STONITH nos servidores RMI (3p)						
	Outros (a propor pelos alunos)		ajax, passe encriptada				