

iBei: Leilões Invertidos

SD 2016/2017 — Meta 2 — 17 de dezembro de 2016 (23:59)

Resumo

Num *leilão invertido* os compradores indicam o que pretendem adquirir e são os vendedores a fazer as licitações. Ganha o vendedor que oferecer o valor mais baixo. Neste tipo de leilões o preço vai descendo à medida que as licitações são feitas, ao contrário do habitual. Este projeto consiste em clonar um sistema típico de leilão (e.g., eBay) por forma a realizar leilões invertidos.

1 Objetivos do projecto

No final do projeto o aluno deverá ter:

- Desenvolvido uma interface web para um sistema de leilões invertidos.
- Ter integrado a interface web com o projecto desenvolvido na primeira meta.
- Dominado Struts2, Java Servlets, JSPs, and JavaBeans.
- Seguido a arquitectura MVC para desenvolvimento web.
- Aplicado Websockets para comunicar assincronamente com os clientes.
- Integrado a aplicação com serviços REST externos.

2 Visão geral

Nesta segunda fase do projecto, irão criar um frontend web para a vossa aplicação. Esta nova interface irá possibilitar os vossos utilizadores a aceder ao vosso serviço 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 DataServer, anteriormente desenvolvido em RMI.

Os utilizadores deverão ter as mesmas funcionalidades, independentemente da interface que usem. Portanto a interface web deverá listar os leilões e deverá permitir a criação de leilões e licitações. Como o aspecto interactivo é muito importante na web, o vosso projecto deverá mostrar alterações em tempo real, nomeadamente no

dinheiro recolhido, e através de notificações quando são enviadas mensagens ou licitações. Como os utilizadores estão cada vez mais exigentes, técnicas menos robustas, como meta-refresh e iframes ocultas, não serão aceites.

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 dois serviços externos: o Facebook e o eBay. Ao criarem um leilão, deve ser criado um post no facebook do autor com o link para o leilão.

3 Arquitetura

A Figura 1 mostra a arquitetura geral do projecto. 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, garantido a interoperabilidade dos dados com os clientes da primeira meta.

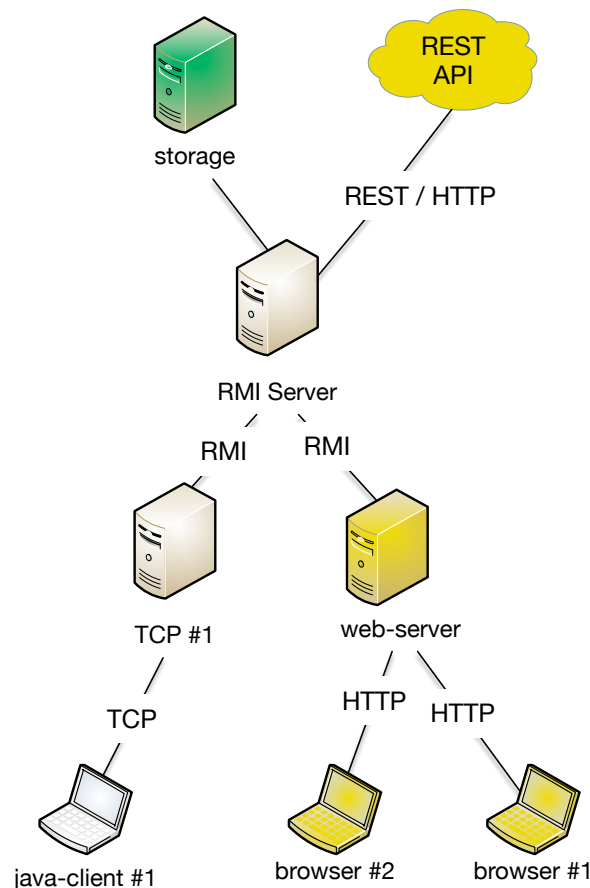


Fig. 1: Arquitectura do projecto

Devem implementar uma aplicação web que corra num servidor HTTP (Apache Tomcat) e que actue 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 deverão ponderar fazer alguns dos pedidos via AJAX em vez de carregar a página toda.

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

4 Interface Web

Usando uma arquitectura MVC, deverão implementar os seguintes requisitos funcionais usando Struts:

- **Criar um novo leilão.** Cria-se um leilão começando por identificar o artigo que se pretende comprar. Para simplificar, considera-se que cada artigo tem um código EAN/ISBN que o identifica univocamente. Cada leilão deve igualmente ter um título, uma descrição e quaisquer detalhes adicionais que o utilizador considere necessários. Para criar o leilão, o comprador indica o preço máximo que está disposto a pagar, bem como a data, hora e minuto em que o leilão termina.
- **Pesquisar leilões existentes por código.** Deve poder-se listar os leilões que estão a decorrer, pesquisando pelo código EAN/ISBN do artigo. Esta listagem apresenta apenas os detalhes suficientes de cada leilão que obedeça ao critério da pesquisa, e pode seleccionar-se um desses leilões para consultar os respetivos detalhes.
- **Consultar detalhes de um leilão.** Para qualquer leilão escolhido, deve poder-se obter todos os detalhes relativos à descrição do artigo, ao término do leilão, às mensagens trocadas e ao histórico de licitações efetuadas nesse mesmo leilão.
- **Listar todos os leilões em que o utilizador tenha atividade.** Um utilizador deve poder listar os leilões nos quais tem ou teve alguma atividade, seja como criador do leilão seja como licitador. Esta listagem sumaria os detalhes de cada leilão e pode seleccionar-se um desses leilões para consultar os respetivos detalhes.
- **Efetuar uma licitação num leilão.** Um vendedor pode propor um preço mais baixo num determinado leilão, desde que o leilão não tenha terminado e a licitação seja a mais baixa até ao momento e seja inferior ao preço máximo.
- **Editar propriedades de um leilão.** O comprador pode alterar todas as descrições textuais relativas a um leilão, sendo que todas as versões anteriores devem ficar guardadas e poder ser consultadas posteriormente para referência.
- **Escrever mensagem no mural de um leilão.** Cada leilão deve ter um mural onde poderão ser escritos comentários, questões e esclarecimentos relativos ao leilão.
- **Término do leilão na data, hora e minuto marcados.** No momento indicado pelo comprador (data, hora e minuto) o leilão termina. Determina-se aí o vencedor e fecha-se a possibilidade de realizar mais licitações. Os detalhes desse leilão são atualizados e podem ser consultados posteriormente.

Grupos de 3 elementos terão de fazer as seguintes funcionalidades extra:

- **Um administrador pode cancelar um leilão.** Um administrador deve poder cancelar um leilão se tal for necessário. O leilão continua a poder ser consultado pelos utilizadores, mas está dado como encerrado e não podem ser feitas licitações.
- **Um administrador pode banir um utilizador.** Um administrador deve poder banir um utilizador se tal for necessário. Todos os leilões criados por esse utilizador são cancelados. Todas as licitações efetuadas por esse utilizador são apagadas. Note que, ao apagar uma licitação num leilão, quaisquer licitações inferiores a essa devem ser igualmente apagadas exceto a melhor delas, cujo valor se torna igual ao valor da que for apagada. Automaticamente é criada uma mensagem no mural dos leilões afetados lamentando o incómodo.
- **Um administrador pode obter estatísticas de atividade na aplicação.** Um administrador deve poder consultar estatísticas da utilização da aplicação: top 10 utilizadores com mais leilões criados, top 10 utilizadores que mais leilões venceram, número total de leilões nos últimos 10 dias.

5 Notificações em Tempo Real

De forma a que a vossa aplicação seja responsiva, deverão usar o poder das WebSockets para fazer push de informação para o cliente assim que possível. Deverão usar WebSockets para:

- **Entrega imediata de mensagens a utilizadores que estejam ligados.** Utilizadores que estejam ligados à aplicação recebem imediatamente as mensagens publicadas no mural de um leilão (server push). O criador de um leilão recebe todas as mensagens relativas a esse leilão. Todos os utilizadores que tiverem escrito num mural passam a receber mensagens escritas nesse mesmo mural.
- **Entrega de mensagens a utilizadores desligados assim que estes se liguem.** Qualquer utilizador que devesse ter recebido uma mensagem, mas não se encontrasse ligado à aplicação, recebe a notificação assim que se ligar a próxima vez.
- **Listagem de utilizadores ligados atualmente.** Deve ser possível obter a lista completa de todos os utilizadores que se encontrem ligados à aplicação.
- **Notificação imediata de licitação melhor.** Um vendedor que tenha feito uma licitação num leilão é notificado imediatamente sempre que houver outra licitação melhor do que a sua, caso esteja ligado à aplicação.

Grupos de 3 elementos terão de fazer a seguinte funcionalidade extra:

- **Mostrar contador de utilizadores na página de cada leilão.** Na página de detalhe de cada leilão, deve estar um contador que seja imediatamente actualizado assim que entre ou saia da página de detalhe. Pretende-se mostrar o número de interessados a cada momento.

6 Integração com serviços REST

Este projecto deverá ser integrado com outros dois serviços web: o Facebook e o eBay. O Facebook será usado para dinamizar a partilha social dos leilões, bem como fornecer uma alternativa ao login por username e password. O eBay será usado para obter um preço de referência para as licitações.

Para usar a API do Facebook é necessário usar autenticação OAuth. Para tratar deste processo devem usar a biblioteca Scribejava. Não serão aceites bibliotecas que façam a integração em Java quer do Facebook, quer do eBay.

As funcionalidades a implementar com recurso a REST são:

- **Associar conta ao Facebook.** Qualquer utilizador logado poderá associar a sua conta do sistema à sua conta do Facebook. Este passo permitirá fazer login com a sua conta do facebook e que os posts sejam criados automaticamente.
- **Login com o Facebook.** Um utilizador que ainda não tenha feito login, deverá poder fazer com o Facebook, entrando automaticamente na sua conta, sem ter de inserir username ou password.
- **Post no Facebook** Assim que um utilizador cria um novo leilão, o sistema deverá automaticamente criar no Facebook um novo post com o link para a página do leilão, tudo sem interacção do utilizador.
- **Mostrar preço mais baixo do eBay** Na página de detalhe de um leilão, deve aparecer o preço mais baixo desse item à venda no eBay. Para tal deve ser usada a Finding API do eBay.

Grupos de 3 deverão ainda fazer a seguinte funcionalidade:

- **Um administrador deverá desassociar um utilizador da conta de Facebook** Na página de administrador, deverá ser possível escolher um utilizador com Facebook associado e remover essa associação.

6.1 Relatório

Devem alocar tempo para a escrita do relatório no final do projecto, tendo em conta os passos anteriores. Devem escrever o relatório de modo a que um novo colega se junte ao grupo e perceba a solução criada, as decisões técnicas efectuadas e possa introduzir novos componentes ou modificar os que já existem. O relatório deve incluir:

- Arquitectura de Software detalhadamente descrita. Deverá ser focada a estrutura de processos, threads e sockets usadas, bem como a organização do código.
- Detalhes sobre a integração do Struts com o Servidor RMI da primeira meta.
- Detalhes sobre a implementação de WebSockets e a sua integração com o servidor RMI.
- Detalhes sobre a integração de serviços REST.
- Descrição dos testes feitos à plataforma.

7 O que irão aprender

Este projecto presuppõe que os alunos aprendam competências práticas relativas a:

- Programar para a Web em Java
- Arquitectura MVC com structs
- Integração de Servidor web com RMI.
- Uso de Websockets
- Consumir APIs REST
- Autenticar usando OAuth

8 Entrega do projecto.

O projecto deverá ser entregue num ficheiro ZIP. Esse ficheiro deverá conter um ficheiro README com toda a informação necessária para instalar e executar o projecto sem a presença dos alunos. Projectos sem este ficheiro, ou sem informações suficientes **não serão considerados**. Projectos que não executem correctamente também não serão avaliados.

Dentro do ficheiro ZIP deverá também estar um PDF 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 o 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 todo código fonte do projecto. A ausência deste elemento levará à anulação do projecto.

O ficheiro ZIP deverá ser entregue na plataforma inforestudante até ao dia 17 de dezembro de 2016 (23:59).

<http://inforestudante.uc.pt>