

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Engenharia Informática e de Computadores



Infra-Estruturas de Sistemas Distribuídos

3º Trabalho Prático

Arquitectura JEE

Enterprise Java Beans (EJB) e Java Message Service (JMS)

Desenvolvido por:

Ricardo Neto (#26657)



Análise do Problema

O trabalho proposto tem o objectivo de permitir consolidar os conhecimentos relativos à arquitectura JEE, bem como, praticar a implementação de *Enterprise JavaBeans* como forma de acesso aos serviços expostos pela plataforma, nomeadamente o serviço de gestão de mensagens (JMS) e de gestão de concorrência (*ConcurrencyManagement*).

O objectivo final do trabalho é semelhante ao dos trabalhos anteriores, contudo, são apresentados requisitos ao nível da tecnologia a utilizar que alteram em grande parte a estrutura de classes utilizada.

Entre as diferenças mais significativas ao nível dos requisitos destaca-se a inclusão de um serviço de controlo de palavras impróprias que deverá notificar os utilizadores de que foram utilizadas palavras impróprias e deverá guardar histórico com informação acerca da sua actividade.

No que respeita ao serviço de tradução, este deverá traduzir mensagens de utilizadores num idioma comum a ser definido administrativamente.



Solução Proposta

De acordo com os requisitos do enunciado e seguindo as indicações dada pelo docente, no sentido de tentar utilizar uma maior diversidade de tipos de objectos disponíveis, foi implementada a arquitectura em baixo mostrada, sendo a mesma detalhada de seguida.

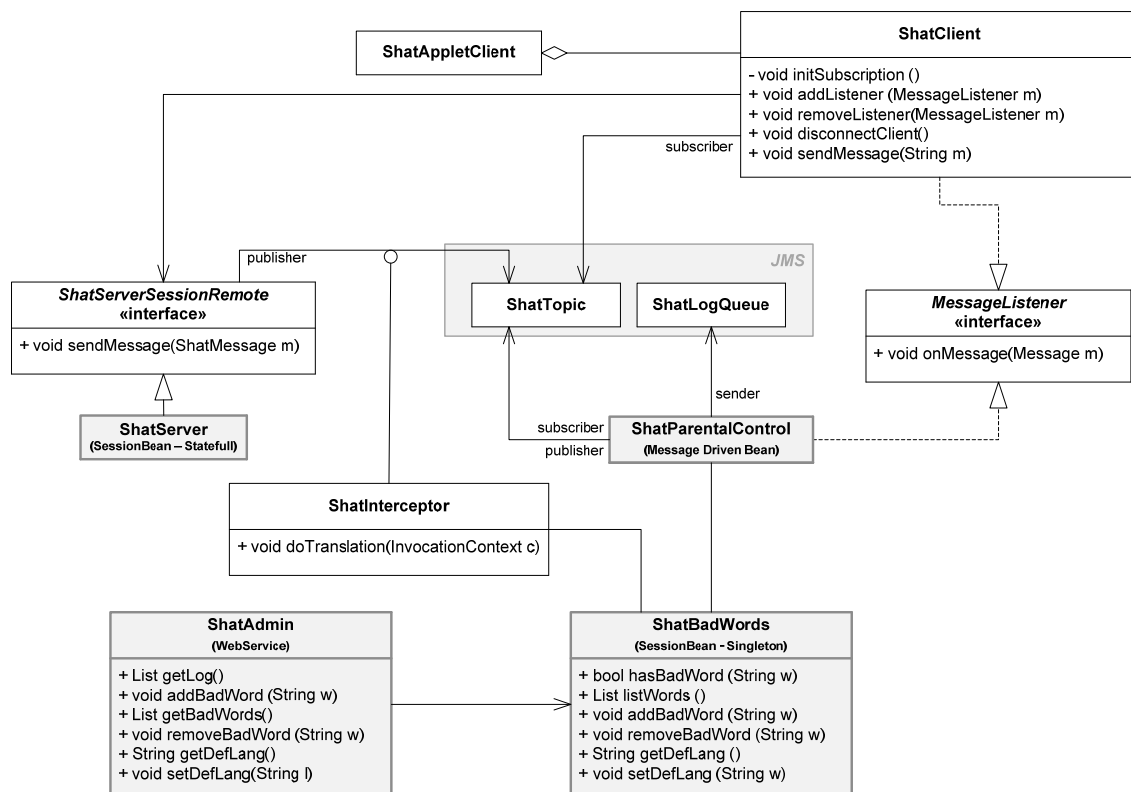


Diagrama 1 - Diagrama de classes

ShatServerSession é uma *Session Bean* responsável por receber as mensagens dos clientes e publica-las em *ShatTopic*. Para a sua implementação optou-se por defini-la como *Statefull* com base na premissa de que não vai existir um número muito elevado de clientes. Assim, possibilita-se que um cliente faça o aproveitamento da sua instância entre chamadas e que os objectos locais da *bean* sejam mantidos penalizando a memória ocupada no contentor mas optimizando a rapidez no acesso. A publicação de mensagens no tópico é interceptada por *ShatInterceptor* que será responsável por alterar a mensagem, traduzindo-a para o idioma utilizado.

A identificação de mensagens com termos impróprios é feita através da *MessageDrivenBean* *ShatParentalControl* que subscreve o tópico. Assim, quando notificada por uma mensagem ter sido publicada, verifica o seu conteúdo junto da classe *ShatBadWords*, esta responsável por manipular os repositórios em memória relativos a termos impróprios e idioma utilizado. Caso a



mensagem contenha um termo impróprio é enviada uma mensagem informativa para o tópico e registada na *ShatLogQueue* informação acerca da ocorrência.

No que respeita à administração dos termos impróprios, bem como, de qual o idioma a ser utilizado, foi implementado o *WebService*, materializado na classe *ShatAdmin*, que manipulará o repositório.

A classe *ShatBadWords* é, conforme indicado, responsável por manipular os repositórios em memória. Desta forma, foi delegado ao contentor a gestão da concorrência no acesso aos seus métodos, estando os mesmos classificados quanto ao tipo de operação a realizar sobre as estruturas de dados.

A aplicação cliente foi implementada com recurso a uma *Applet* através da classe *ShatAppletClient*. Esta mantém a estrutura utilizada nos trabalhos anteriores sendo que utiliza uma instância de *ShatClient* para comunicar com o contentor.