
Integração de Sistemas

Relatório projeto 2

Bruno Madureira 2011161942 / Fábio Moura 2013118534 - 10 de Novembro de 2014



Índice

Introdução	pag 3
Arquitetura usada	pag 4
Camada Lógica.....	pag5
Camada de Apresentação.....	pag 5
Divisão do trabalho.....	pag 6
Bibliografia	pag 6

Introdução

O objectivo deste trabalho foi desenvolver uma aplicação com 3 camadas usando a tecnologia Java EE. Esta aplicação estava dividida entre a parte Web, os Enterprise Java Beans e um mecanismo de persistência de dados.

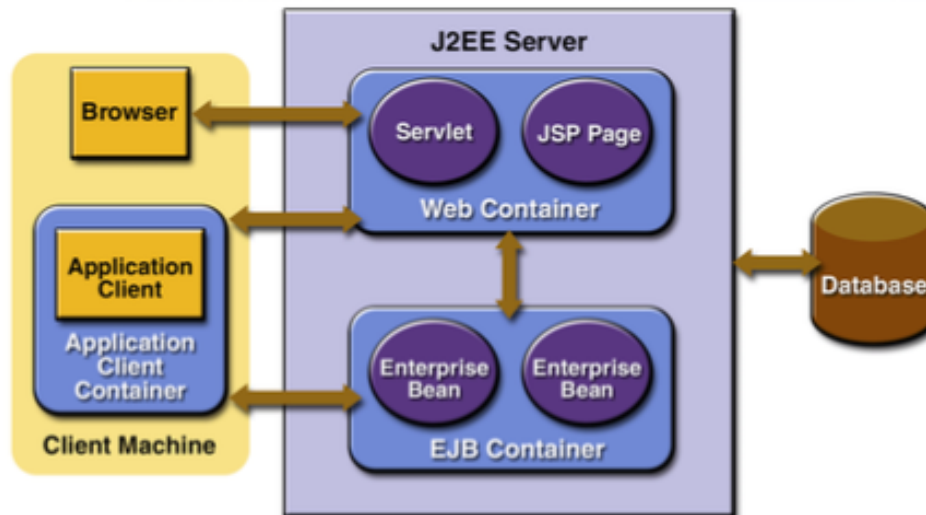
Para elaborar a parte Web usamos uma Servlet e vários Jsp. Esta camada comunicava com a camada responsável pela persistência através da camada intermédia.

A camada intermédia é construída por vários EE Beans normais e um EE bean message-oriented. Esta camada recebe pedidos da camada web e envia a resposta. Esta camada para conseguir aceder a base de dados usa a camada mais abaixo, a camada de persistência.

A camada de persistência é responsável por criar a base de dados, persistir dados e enviar dados contidos na base de dados as camadas superiores.

Esta aplicação pedia que comunicasse-mos com o Crawler do trabalho 1, esta comunicação é feita através do tópico criado anteriormente e do message Driven EE Bean desta nova aplicação. Após popular a base de dados os clientes podiam ligar se a nossa aplicação através de um browser, sendo que tem a possibilidade de se registar, fazer login e de verem noticia usando varias formas de pesquisa.

Arquitetura usada



Camada Logica

JPA: Esta camada lida com o problema da persistência dos dados. Esta camada tem 3 packages, todas elas com classes com anotações JPA. A primeira package chama-se "data" e contém dados relativos ao utilizador. A package generated contém as classes geradas a partir do xml sem anotações. A última package "toDatabase" contém o mesmo conteúdo que a package generated, mas neste caso as classes já contêm anotações de persistência. Tivemos problemas ao fazer unmarshall já com as anotações por isso fizemos unmarshall para as classes do projecto antigo e depois convertimos as para as novas classes anotadas.

Nota importante : Como base de dados usamos PostgreSQL.

EJB: Como a parte web é suficiente para guardar a informação relativa a cada user decidimos usar um bean Stateless. Sendo assim cada chamada ao bean é reconhecida com uma nova, não necessita de guardar dados. Este bean serve de "ponte" entre a camada web e a de persistência.

Message Driven Bean: Este bean está continuamente a espera de mensagens no tópico criado no trabalho anterior. Ao receber uma mensagem, faz unmarshall da mesma e persiste os dados na base de dados com o auxílio da camada de JPA.

Camada de Apresentação

Conteúdo Web: A interface web é constituída por vários Jsp e por um Css. Tentamos ao máximo fazer uma interface limpa e amigável.

Servlets: A Servlet serviu de interface entre os Jps e a os EJBs. Os pedidos feitos pelos users nas Jps eram tratados pela Servlet, esta identifica o pedido e redireciona ao EE bean adequado. Ao receber a resposta a Servlet envia a Jsp/dados adequados ao utilizador.

Divisão do trabalho

	Horas de esforço	Parte do Trabalho
Bruno Madureira	5 estudo 30 implementação	Configuração do Jboss Camada Logica Relatório
Fabio Moura	2 estudo 35 implementação	Interface Web Message Driven-Bean Relatório

De notar que a divisão não foi estrita e que ambos os elementos ajudaram nas partes um do outro.

Bibliografia

<http://eai-course.blogspot.pt/2014/10/an-enterprise-application-repository-to.html>
http://www.tutorialspoint.com/java/java_sending_email.html
<http://www.mastertheboss.com/jboss-server/jboss-configuration/jboss-mail-service-configuration>
<http://www.tutorialspoint.com/servlets/>
http://en.wikipedia.org/wiki/JavaServer_Pages
<http://www.w3schools.com/html/>