1 Spring

O código do projeto foi implementado na linguagem Java (por questão de familiaridade com a linguagem e o uso dos Handouts como base) utilizando o framework Spring. A vantagem do uso do Spring parte da facilidade de organizar um MVC utilizando este framework. Sua configuração envolve tres arquivos **.xml**, incluindo o **web.xml**(para que a compilação funcione de forma correta),o **pom.xml** (indica que o seu projeto foi de fato convertido para Maven e resolve dependências de terceiros), e o **spring-context.xml** (habilita o uso de anotações e o local dos JSPs).Este projeto utiliza o recurso Maven por este ter um mecanismo de compilação (build) bem integrado.

2 MVC

O projeto utiliza a arquitetura MVC com o intuito de separar dados ou lógica de negócios (Model) da interface do usuário (View) e do fluxo da aplicação (Controller). A ideia é permitir que uma mesma lógica de negócios possa ser acessada e visualizada através de várias interfaces. o MVC retira a lógica de negócios e a interface de dentro dos servlets e põe em 2 camadas, a Model usando classes de Java simples e a View com JSPs. Os servlets, portanto atuarão somente como a camada de controle.

2.1 Model

A parte de model do software do chat esta localizada em um pacote com o nome de mvc.model. Este pacote contem quatro classes. A **ChatDAO.java** que contem todos os dados das mensagens que tem relação com o banco de dados em SQL da tabela *Mensagem*(As mensagens do chat são armazenadas em uma tabela que é organizada por um ID, as mensagens e o usuários escrevendo as mensagens) para limpar o chat e "zerar" esta tabela com mensagens, um botão esta disponível na página de bate papo para iniciar um bate papo novo, a **Parametros.java** é uma classe com construtores e com métodos de acesso (get e set) com as informações das mensagens, a **Usuario.java** é uma classe com construtores e com métodos de acesso (get e set) com as definições dos perfis dos usuários e a **UsuarioDAO.java** contem todos os dados dos usuários que tem relação com o banco de dados SQL da tabela *Usuario*(Os usuários registrados são armazenados em uma tabela que é organizada por um login , uma senha e uma foto).

2.2 View

A parte de view do projeto está alocada na pasta *WebContent/WEB-INF/views* onde estão todos os arquivos JSP.Temos cinco arquivos JSP: **formulario-registro.jsp** responsável pela página de registro dos usuários, **formulario-login.jsp** responsável pela página de login dos usuários cadastrados,**menu.jsp** responsável pela página de menu principal do Chat que dá acesso ao bate papo e as **dadosTela.jsp** e **telaChat.jsp** responsáveis pela página principal de bate papo.

2.3 Controller

O software contem um pacote chamado mvc.controller. Este pacote contem as classes **AutorizadorInterceptor.java** que funciona como um filtro, onde qualquer requisição deve passar por um filtro que analisa se a ação pode continuar. Esse filtro vai verificar se a requisição já está logada na sessão ou senão se está numa lista de exclusão. A segunda classe deste pacote é a **chatController.java**, responsável pelo controle de todas as chamadas relacionadas às mensagens e ao usuário.

3 Ajax

Para que a página fosse atualizada o tempo todo para os usuários que utilizam o chat, foi utilizado o Ajax, esta decisão foi tomada pois o Ajax pode fazer o update dá página web constantemente sem recarregar a página inteira. Sua rotina funciona como um loop que recarrega a pagina de tempos em tempos(tempos muito curtos, para a página estar em constante update).

4 Utilização

O chat pode ser utilizado no IP local (minha assinatura do Bluemix expirou). As páginas possíveis de acesso são divididas em: **("Chat_Insper/")** que é redirecionada para a página de registro,**("Chat_Insper/registro")**que serve para se cadastrar no chat, **("Chat_Insper/login")**para fazer login no chat, **("Chat_Insper/efetuaLogin")** página de menu e **("Chat_Insper/ChatInsper")**página de bate papo.