

Ejercicios de Comunicación

Índice

1 Chat con XML y AJAX (*).....	2
2 Applet para el chat (*).....	3
3 Autocompletar con jMaki (*).....	3

1. Chat con XML y AJAX (*)

Vamos a añadir a la aplicación de chat realizada en anteriores sesiones la posibilidad de obtener la lista de mensajes publicados en XML. Esto nos permitirá acceder al chat mediante una interfaz web utilizando AJAX, o utilizar un cliente rico. Se pide:

a) Crear un JSP llamado `listaMensajesXml.jsp` que formatee la lista de mensajes publicados en el chat en XML. Debe utilizar un formato como el siguiente:

```
<mensajes>
<mensaje>
  <nick>Ana</nick>
  <texto>Hola</texto>
</mensaje>
<mensaje>
  <nick>Jose</nick>
  <texto>Que tal?</texto>
</mensaje>
</mensajes>
```

Importante

Se deberá hacer que el tipo MIME de este documento sea `text/xml`, ya que de lo contrario el cliente no lo reconocerá como tal y no podremos analizarlo.

Actualizar el servlet `ListaMensajesServlet` para que tome como entrada un parámetro `formato`. Los posibles valores para el formato serán:

- 0: Formato Web (HTML)
- 1: Formato XML (para utilizar AJAX)
- 2: Formato serializado (para comunicación con cliente ricos Java)

Si no se especifica este parámetro, por defecto el *servlet* servirá la versión web del chat.

Al especificar el formato 1 (XML), se deberá redirigir la salida al JSP que acabamos de crear, proporcionando la lista de mensajes como un atributo de la petición. Publicar una serie de mensajes mediante la interfaz web ya existente. Consultar la lista de mensajes en XML accediendo a la siguiente dirección:

`http://localhost:8080/cw-sesion04-chat/servlet/ListaMensajesServlet?formato=1`

b) En el directorio `cw-sesion10-ajax` de las plantillas de la sesión se proporciona el fichero `chatAjax.html` que contiene la página web de la versión AJAX del chat, que obtiene los mensajes del servidor en el formato visto en el punto anterior. Observar el código fuente de esta página. En él encontraremos el código que hace la petición HTTP al servidor y analiza la respuesta recibida. Copiar este fichero al directorio `chat` de nuestra aplicación de chat, probar a acceder ella mediante esta interfaz y comprobar que el chat funciona correctamente.

Nota

Se recomienda probar la aplicación en un navegador real, ya que el que está integrado en Eclipse podría no funcionar correctamente con determinadas funciones de Javascript.

2. Applet para el chat (*)

Vamos a utilizar un cliente rico para acceder la aplicación de chat. Esta aplicación cliente ya está desarrollada en el proyecto `cw-sesion10-cliente`. Se pide:

- a) La aplicación invoca cada 5 segundos al servlet `ListaMensajesServlet`. Espera recibir como respuesta un objeto `ColaMensajes`, del cual mostrará todos los mensajes en el área de texto de la aplicación. Implementar en el servlet `ListaMensajesServlet` la opción de devolver serializado el objeto `ColaMensajes` al cliente.
- b) Probar el cliente rico para acceder al chat. Comprobar que se puede acceder correctamente desde las diferentes interfaces que hemos creado.

3. Autocompletar con jMaki (*)

Vamos a crear una aplicación que nos permita consultar determinados atributos del ámbito del contexto. Tendremos un campo de texto en el que tendremos que introducir el nombre del atributo, y que debe permitirnos autocompletar el texto introducido con todas las coincidencias encontradas. Utilizar el *widget* `Autocompleter` de `Scriptaculous` para implementar dicho campo de texto.

