Práctica 6

Vilchis Domínguez Miguel Alonso

1. Contexto:

Los usuarios de tu chat no se sienten cómodos teniendo que actualizar la lista de contactos conectados, además aprovecharemos para implementar la funcionalidad de inicio de sesión.

2. Objetivo de la práctica

Mejorar la experiencia del usuario en el sistema.

3. Especificaciones:

- Se implementará la funcionalidad de registro, para tener un control sobre las personas que hacen uso del chat, así como tener la opción de personalizar la sesión de cada usuario en un futuro. La ventana de login ahora tendrá la opción de registro, en donde se mostrará una nueva ventana que pedirá el nombre de usuario, la contraseña y un tercer campo para que se repita la contraseña (las contraseñas se deben ocultar). En esta misma ventana habrá un botón para mandar los datos al servidor de ubicación, (antes de mandar los datos se debe verificar que las contraseñas coincidan). Si el servidor de ubicación no tiene un usuario que coincida conel nombre de usuario recibido, entonces agregará al usuario con la contraseña a un archivo que servirá como base de datos y le regresará una lista con los usuarios conectados para que el cliente los muestre (como si se hubiera logueado en el sistema). Si el nombre de usuario ya está registrado, entonces se debe mostrar un mensaje del lado del cliente donde indique que el nombre de usuario ya está ocupado, y que se debe escoger otro.

- El archivo de los usuarios registrados seguirá el mismo patrón que el archivo descrito en el práctica 1, es decir se guardará el nombre del usuario seguido de dos puntos y la contraseña a la que se le sumará 5 a cada código ascii que la componga.

$$usuario : password$$

- Ahora en la etapa de login si se trabaja de manera local contendrá el puerto local, puerto del directorio, nombre de usuario y contraseña. Mientras que si se trabaja en red se pedirán los campos de ip del directorio de ubicación, nombre de usuario y contraseña. En ambos casos se mandarán los datos al servidor de ubicación (a través del DirectoryChannel) para que se lleve acabo la validación del nombre de usuario con la contraseña escrita. Si son válidos los datos proporcionados, el servidor de ubicación regresará la lista de contactos conectados para que el cliente lo despliegue en una nueva ventana. De no ser válido el servidor de ubicación regresará la cadena " combinacion no valida", en cuyo caso, la ventana de login debe mostrar un aviso informando del error.

- La actualización de los clientes activos ahora se llevará acabo en el servidor de ubicación por lo que se debe designar un thread que cada 5 seg haga uso del método *ping_wrapper* el cual regresará True.
  El metodo *ping_wrapper* debe ser incluido en el ApiServer de los clientes del chat (Para hacer esto se debe crear un request channel del lado del servidor de ubicaciones para cada cliente conectado cuando el cliente se conecte, y se debe remover el request Channel cuando el cliente se desconecte).
  Si el cliente por alguna razón se detuvo, al intentar hacer uso de su metodo ping se producirá una excepción sobre la conexión rechazada, en ese caso se debe remover al cliente de la lista de contactos activos.

- Al cerrarse la ventana de lista de contactos se deben cerrar todas las ventanas de conversación y se debe hacer uso del metodo de desconexión del servidor de ubicación para que remueva al cliente de la lista de contactos activos.
- La actualización de los contactos conectados del lado del cliente ahora será responsabilidad del servidor de ubicación y no del usuario. Por lo que, cada que se realice un cambio sobre la lista de contactos activos el servidor de ubicación a través del requestChannel de cada cliente actualizará dicha lista.
- Se debe de hacer todo el manejo de excepciones para que al ejecutar el chat este no produzca ningún tipo de error.
- Al notar que se produce un error, como la falta de acceso a la camara web o al micrófono, así como la disponibilidad del otro cliente (que por alguna razón ya no se pueda mandar mensaje). Se debe de mandar una alerta al cliente informando en términos no técnicos sobre el error.
- La estructura del proyecto sigue siendo la misma:

```
|
|− −Constants
                |− −Constants.py
                |− −AuxiliarFunctions.py
|− −GUI
                |− −LoginWindow.py
                |− −ListContactWindow.py
                |− −ChatWindow.py
                |− −CallWindow.py
|− −Channel
                |− −ApiServer.py
                |− −ApiClient.py
                |− −Channel.py
                |− −RecordAudio.py
                |− −RecordVideo.py
                |− −DirectoryChannel.py
|− −Directory
                |− −Directory.py
|− −GraphicalUserInterface.py
```

Los cambios que deben considerarse son:

- En *ApiServer.py*, se debe agregar el método $ping_wrapper$ el cual será consultado por el servidor de ubicaciones. Además de agregar el método $update_contact_list$ que será usada por el servidor de ubicaciones cuando se produzca un cambio en la lista de contactos conectados, este método debe estar asociado a una señal que actualice la lista en la interfaz.
- En *Directory.py* se debe agregar el método $access_wrapper$, el cual recibirá como parámetro el nombre de usuario y contraseña, y debe validar los campos, si son válidos debe regresar la lista de contactos conectados, en caso contrario la cadena çombinacion no valida".
- En *Constants* se debe agregar la cadena Çombinación no válidaz todas las cadenas relacionadas a los errores desplegados en el manejo de excepciones.
- En *DirectoryChannel* se debe dar soporte para la consulta del metodo $access_wrapper$ con el metodo $access$.

4. Preguntas para el reporte:

- Se ha agregado un aumento de tráfico por las funcionalidades agregadas, si se tienen 3 clientes conectados al sistema que solo intercambian mensajes de texto y cada 20 segundos un cliente le responde a los 2 con los que está hablando. Supongase que cada mensaje de texto ocupa 2 paquetes, y los pings 1 paquete ¿Cuál es el porcentaje de tráfico que se genera por conversación vs el tráfico que se genera por las funcionalidades agregadas?
- En el escenario anterior, ¿Cuántos clientes intercambiando mensajes entre si deben existir para que el porcentaje de tráfico por conversación sea mayor ? (Supóngase que cada cliente le contesta a todos los contactos conectados cada 30 segundos)

5. Entrega

Fecha de entrega: 3 de Octubre del 2016

La entrega es por parejas, la práctica con la estructura mencionada debe estar almacenada en un depósito de github llamado Redes2017, en la rama Practica6, el readme debe contener el nombre de los 2 integrantes del equipo.

Deben mandar el link del depósito con el asunto *[Redes-17]Practica6* al siguiente correo:

mvilchis@ciencias.unam.mx

Nota: Basta con que un integrante del equipo suba su código al depósito.
Nota: A las 11:59 del díal de la entrega se descargará el contenido de los depósitos, para evaluación, asegurense que se encuentre su versión final para esa hora.