

Un programa de Chat

Creando el Server

Colocar un Memo.

Por comodidad el cambiamos el nombre por mMensajes.

Colocar un Edit1 (Cuadro de Texto). Por comodidad le cambiamos el nombre al Edit1 por eMensaje.

Colocar un Botón

Al Botón hacer Caption = Enviar Mensaje .Al presionarlo, se enviará el mensaje escrito en el Edit1 (que ahora tiene el nombre eMensaje). Hacer la propiedad Default=true. De este modo podremos dar Enter y será equivalente a presionar el Botón.

Vamos a la pestaña de componentes Internet y traemos un **ServerSocket**.

Debemos asignarle un número de puerto que no esté ya ocupado. Ejemplo 4000. (sino nos deja asignarlo es porque está activo). Desactivarlo, Active= false.

Vamos a propiedades. Activamos el componente. Active= true.

Luego el programa cliente deberá tener asignado el mismo puerto para comunicarse entre sí.

Limpiamos lo contenidos originales del Memo y del Edit.

Cambiamos el color a la Form1 para identificar el programa servidor. Propiedades – Color.

Guardamos el proyecto como ya sabemos.

Ahora seleccionamos el objeto ServerSocket y accedemos al Inspector de Objetos, pero la pestaña Events (Eventos). Damos doble click en **OnClientConnect**. Esto nos permite Programar las acciones a realizar cuando un Cliente se conecte. En este caso deseamos que aparezca en el Memo, la frase “Se ha conectado.. *La IP del cliente*”

```
mMensajes->Lines->Add(“Se ha conectado” + Socket->RemoteAddress);
```

Nota: La variable Socket trae toda la información del cliente que se ha conectado.

Pero si deseamos que aparezca en lugar de la IP de la computadora remota, el nombre de esta, se deberá lo siguiente.

```
mMensajes->Lines->Add(“Se ha conectado” + Socket->RemoteHost);
```

También deseamos que cuando se desconecte nos avise en el Memo. Para lo cual nuevamente seleccionado el objeto ServerSocket, vamos al Inspector de Objetos, pestaña Events, doble click en el evento OnClientDisconnect. Allí programamos las acciones a realizarse bajo este evento:

```
mMensajes->Lines->Add(“Se ha desconectado” + Socket->RemoteAddress);
```

ó

```
mMensajes->Lines->Add(“Se ha desconectado” + Socket->RemoteHost);
```

Ahora deseamos que cuando el cliente envíe un mensaje, este sea leído y mostrado en el Memo. Seleccionado el objeto ServerSocket, Inspector de Objetos, Events, doble click en el evento OnClientRead, y agregamos las acciones deseadas:

```
mMensajes->Lines->Add(Socket-> RemoteAddress + “: “+ Socket->ReceiveText());
```

ó

```
mMensajes->Lines->Add(Socket->RemoteHost + ":" + Socket->ReceiveText());
```

Ahora dentro del Botón (Enviar Mensaje), colocamos lo necesario para contestarle al Cliente:

Con lo siguiente, al presionar el botón, estamos enviando el contenido del cuadro de texto (el Edit llamado eMensaje) al cliente conectado:

```
ServerSocket1->Socket->Connections[0]->SendText(eMensaje->Text);
```

Pero como se desea que se refleje el mensaje enviado en el Memo, agregamos también dentro del botón lo siguiente:

```
mMensajes->Lines->Add(ServerSocket1->Socket->LocalHost + ":" + eMensaje->Text);
```

Además se desea que se limpie el contenido del Edit llamado eMensaje.

```
eMensaje->Clear();
```

Creando el Cliente

Ahora nos dedicaremos al cliente.

Los componentes necesarios son similares a los empleados en el servidor

Un Memo para el seguimiento de los mensajes entrantes y salientes. Al cual cambiaremos su nombre por mMensajes.

Un cuadro de texto Edit. Donde se coloca el texto a enviar.. Lo llamaremos eMensaje.

Limpiamos lo contenidos originales del Memo y del Edit.

Un botón con el texto Enviar Mensaje. Hacer la propiedad Default=true. De este modo podremos dar Enter y será equivalente a presionar el Botón.

Cambiaremos el color de la Form para diferenciar con el servidor.

Vamos agregar un cuadro de texto (Edit) donde se colocará la dirección IP del servidor al que debe conectarse. Y un Botón (con Caption= Conectar) para que ejecuta la acción de conectarse al servidor cuya IP pusimos en el Edit.

Al Edit que contiene la IP le cambiamos el nombre a: eIP. Si conocemos la IP del servidor podríamos cargar en la propiedad Text dicha IP, para que aparezca por defecto.

Es conveniente colocar un objeto Label con el tecto IP del Servidor junto al Edit llamado eIP.

El componente clave es el objeto ClientSocket que lo encontramos también en la pestaña Internet. Lo agregamos a la Form. Recordemos que tenemos que asignarle el mismo puerto. (Propiedad Port).

En la propiedad Address del ClientSocket no colocamos nada, ya que el valor de esa propiedad se cargará con el contenido del Edit llamado eIP, al presionar el Boton Conectar. Esto es util si queremos conectarnos a otras computadoras.

Por lo cual en dicho Botón deberán ir las siguientes instrucciones:

```
ClientSocket1->Address = eIP->Text;  
ClientSocket1->Active = true;
```

Al activar el ClientSocket este buscará a la IP indicada, si existe un servidor conectado al mismo puerto que el. En caso de existir van a establecer conexión, y ya podremos enviar mensajes entre cliente y servidor.

Para que se envíe el mensaje se deberá colocar en el Botón Enviar Mensaje lo siguiente:

ClientSocket1->Socket->SendText(eMensaje->Text);

Esta instrucción envía el texto al Servidor con el cual ya se esta conectado.

Y para que se refleje en el Memo:

mMensajes->Lines->Add(ClientSocket1->Socket->LocalHost + ":" + eMensaje->Text);

Donde LocalHost es el nombre de la computadora actual, donde esta el cliente.

Ahora cuando el Server nos está contestando, hay que reflejarlo en el Memo.

Entonces existe un evento de este componente (que dice que cuando este componente lee, es porque el Server nos esta mandando algo.

Damos doble clic en el evento OnRead y colocamos las siguientes instrucciones:

mMensajes->Lines->Add(Socket->RemoteHost + ":" + Socket->ReceiveText());

O sea que se refleje en el Memo, el mensaje enviado por el Server.

Además se desea que se limpie el contenido del Edit llamado eMensaje.

eMensaje->Clear();