Cómo programar aplicaciones Cliente/Servidor en Visual Basic utilizando el Control WinSock.

Contenido

Introducción a Visual Basic.

Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del BASIC (Beginner´s All−purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo un lenguaje para principiantes.

Primero fue GW−BASIC, luego se transformó en QuickBASIC y actualmente se lo conoce como Visual Basic y la versión más reciente es la 6 que se incluye en el paquete Visual Studio 6 de Microsoft. Esta versión combina la sencillez del BASIC con un poderoso lenguaje de programación Visual que juntos permiten desarrollar robustos programas de 32 bits para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft, ya que el lenguaje sólo es compatible con Windows, un sistema operativo de la misma empresa.

Visual Basic ya no es más un lenguaje para principiantes sino que es una perfecta alternativa para los programadores de cualquier nivel que deseen desarrollar aplicaciones compatibles con Windows.

Conceptos básicos de programación Cliente/Servidor.

Un programa simple es un conjunto de instrucciones que generalmente devuelven un valor al usuario, ya sea numérico o una cadena de letras, este dato es el resultado de la acción del usuario sobre el programa, ya que el usuario fue el que solicitó el dato.

Al igual que un usuario se comunica con el programa por medio del teclado, dos programas se pueden comunicar entre sí por medio de un control especial que se denomina **WinSock Control**. Este control esta disponible en el lenguaje Visual Basic, y su nombre proviene de **Windows Sockets**.

El Winsock Control como opción predeterminada no se encuentra disponible en la barra de controles estándar de Visual Basic, para acceder a él debemos agregarlo manualmente mediante Proyecto> Componentes> y luego seleccionar WinSock Control y Aceptar. No es visible en tiempo de ejecución, lo que significa que solo nosotros sabemos que el control se encuentra en nuestra aplicación y cuáles son sus propiedades, aunque también se pueden definir en tiempo de ejecución.

Este tipo de aplicaciones Cliente/Servidor permiten comunicar programas entre sí, en consecuencia también permiten comunicar varias computadoras, porque habiendo un programa en la computadora llama Oscar_1 y otro en la computadora llama Daniel_1 ambos programas se pueden comunicar a través de Internet y compartir información, o adquirir información sin saber que el dueño de la computadora lo autoriza. Esto frecuentemente trae problemas al querer distinguir si un programa esta autorizado por el dueño o el encargado (administrador o sysop) para acceder al sistema. Varios administradores permiten el acceso, pero a la misma vez restringen las carpetas importantes de sus computadoras para no correr el riesgo de perder información vital. Si el programa que se usa para acceder a otra computadora esta autorizado por el encargado de la PC se puede denominar Herramienta de administración remota, en cambio si el acceso no es permitido o no se avisa que se quiere entrar y se hace a la fuerza el programa recibe el nombre de Troyano.

Troyanos

Un troyano es un aplicación disfrazada de un programa útil, consta de dos programas, el Servidor es el que se encarga de abrir un puerto en la PC a la que se quiere tener acceso y dejar el puerto a la escucha, es decir esperar a que se realice una conexión al puerto para dar el acceso a la máquina. Y el Cliente, este es el programa que se conecta al puerto que el Servidor dejó abierto, solicita que se realice la conexión y después comienza a transmitir información, pidiendo datos de la PC remota, tales como Información del sistema, contraseñas, archivos importantes, etc.

Se pueden utilizar de dos formas completamente distintas;

- Como herramienta de administración remota: que permite manipular el sistema a distancia, ideal para personas que necesitan urgente un archivo de la PC de su oficina y se encuentran en su casa. Se puede considerar como tal solo cuando el usuario tenga el acceso permitido a esa PC.

- Como herramienta para **hackear**: (Hackear: penetrar un sistema informático sin acceso) esta es la forma de utilización que prefiere cualquier persona con una conexión a Internet y ganas de espiar lo que hace otra persona conectada a Internet o a su Red privada, también llamada LAN (Local Area Network o Red de Area Local). Pudiendo acceder a sus archivos confidenciales, contraseñas, recursos compartidos, conversaciones que toman lugar en tiempo real, o borrar archivos fundamentales tales como por ejemplo: COMMAND.COM (dejando a la PC **víctima** sin poder arrancar, a menos que el usuario **atacado** sepa iniciar desde un disco de rescate o de inicio.

Los puertos que se dejan a la escucha generalmente sol altos, es decir puertos que pasan del número 500 o el 1000, para garantizar que ningún otro programa pueda estar usándolos y cancelar la conexión del troyano.

El uso de estos programas no es ilegal a menos que el usuario final opte por entrar a la máquina remota sin autorización. En dicho caso se puede proceder legalmente de acuerdo al país en el que se encuentre la PC **hackeada**, es decir la computadora a la que se infiltró el **hacker**. (hacker: persona interesada en el funcionamiento y vulnerabilidad de los sistemas operativos, lenguajes de programación y seguridad informática). Por ejemplo en EEUU se puede condenar a una larga sentencia por hacer eso, pero también hay que conocer la otra cara de la moneda, es el caso de Argentina, todavía no tiene tipificado en el código penal la intrusión en computadoras sin autorización, esto quiere decir que no es un delito condenable.

Aquí debajo una lista de los troyanos más conocidos del **Underground**, es decir de la sociedad hacker, o más bien todo aquello que sea difícil de encontrar para el usuario común. La palabra UnderGround significa debajo de la tierra, lo que para la mayoría significa algo oculto y qué otro ejemplo más conciso que los programas que usa un hacker. Cabe aclarar que el uso de estos programas comunmente denominados para hackers no convierte a nadie pero absolutamente nadie en hacker. El hacker no se hace de la noche a la mañana, es más, el hacker no se hace, nace

- NetBus: Este troyano o herramienta de administración remota fue uno de los más difundidos en Internet, ganó un gran número de usuarios adictos al programa por su sencillez de uso y la rapidez del mismo. El tamaño del servidor (el encargado de permitir el acceso a la máquina con o sin autorización) ahora parece grande en comparación con los troyanos nuevos.
  Tamaño del servidor: 495 KB aproximadamente.

- Back Oriffice 2000: Sin lugar a duda el troyano que más pánico causó en los últimos tiempos. Fue el preferido de todos por ser el primero que salió en Internet con una facilidad de uso impresionante y características que otros troyanos aun no imaginaban, como la renovada parte gráfica. En la última versión del programa se puede notar que fue programado para funciones de administración remota, ya que se nota la programación estructurada y concisa, sin botones de más, ni funciones innecesarias para el usuario final, la mejor versión hasta el momento.

- SubSeven: Otro troyano que causó un gran impacto, probablemente el más usado en la actualidad, ya que el programa servidor ocupa menos aun que el servidor del NetBus o el Back Oriffice. La parte gráfica es distinta a las demás, la complementan un gran juego de skins (texturas, colores, etc.) y mejor facilidad de uso, además incluye nuevas funciones como la desconexión de Internet del equipo remoto, el cuelgue del modem, el cambio de resolución de la pantalla, lista de los passwords que se encuentran en el cache (las contraseñas que el usuario escribió recientemente), y los passwords de la conexión telefónica a redes, es decir la contraseña de internet.
  Tamaño del servidor: 327 KB

- Cybersensor: Este troyano esta programado especialmente para funcionar bajo WindowsNT. No es tan conocido como los anteriores.
  Tamaño del servidor: 29.5 KB

- DeepThroat v2: Este programa también es bastante conocido, incluye muchas funciones muy parecidas al resto de los troyanos, como la de adquirir las contraseñas en el chache de la PC remota y las típicas funciones del resto.
  Tamaño del servidor: 304 KB

- Dolly Trojan: Excelente troyano, lástima que no se ganó el aprecio del público porque el servidor es muy grande. Evidentemente el programador no tenía la experiencia necesaria.
  Tama

- Girlfriend 1.35: Al contrario del Dolly Troyan este programa es muy pequeño, al igual que su servidor, por lo tanto no incluye tantas funciones.

- InCommand v1.0: Diferente a todos los demás este programa es de tamaño medio, pero lamentablemente no pudo adquirir la atención del usuario porque no tiene suficientes funciones.
  Tamaño del servidor: 168 KB

- NetSphere: Nuevamente, al igual que el Dolly este troyano posee un servidor muy grande por lo que se hace pesado el envío por Internet o por e−mail, lo que lleva a la gente a buscar algo menos pesado para enviar, recurriendo a otro troyano.

Tamaño del servidor: 621 KB

- Master Angel 97: Este troyano es uno de los menos conocidos, pero no deja de ser muy bueno.

Comenzando a programar con WinSock Control.

Protocolos TCP/IP y UDP

Como anteriormente quedó aclarado dos programas se pueden conectar entre sí a través de internet o de una LAN.

Internet usa el protocolo TCP/IP que significa Transmision Control Protocol / Internet Protocol, es el que se encarga de recibir paquetes de información y redirigirlos al usuario final que los solicitó. Este protocolo es el preferido por todos ya que posee una característica que UDP le envidia, TCP/IP puede verificar que el paquete de información haya llegado con éxito al destinatario final, concretando así la transacción.

Por el contrario UDP no puede hacer esto, solo manda el paquete con la información y no verifica que haya llegado satisfactoriamente, poniendo de esta manera en peligro al paquete, ya que puede no llegar entero al destinatario y por lo tanto no sirve si el paquete no llega en su totalidad.

Todas las máquinas que están conectadas a Internet tienen asignadas un número que se forma con 4 cifras de 3 dígitos (que no pueden superar al número 255). Ejemplo del IP máximo que se puede encontrar: 255.255.255.255

Propiedades, métodos y eventos de WinSock.

Una vez que tenemos el WinSock control en nuestra barra de controles en Visual Basic ya podemos comenzar a ver las propiedades, eventos y métodos más importantes del control. Para agregarlo manualmente ir a Proyecto> Componentes> y luego seleccionar WinSock Control y Aceptar. Como mencionamos anteriormente este control no es visible en tiempo de ejecución.

Primero abrimos un proyecto (EXE Estándar) y colocamos en control en cualquier parte del formulario. Vamos a comenzar por ver las propiedades, estas pueden ser puestas en tiempo de diseño como también en tiempo de ejecución. A continuación las propiedades más importantes;

Lista de propiedades más importantes

LocalIP: Devuelve la dirección IP de la máquina local en el formato de cadena con puntos de dirección IP (xxx.xxx.xxx.xxx).

LocalHostName: Devuelve el nombre de la máquina local.

RemoteHost: Establece el equipo remoto al que se quiere solicitar la conexión.

LocalPort: Establece el puerto que se quiere dejar a la escucha.

RemotePort: Establece el número del puerto remoto al que se quiere conectar.

State: Verifica si el Control WinSock esta siendo utilizado o no.

Estas son algunas de las propiedades más importantes, y a continuación la sintaxis de cada propiedad.

Objeto.Propiedad = Valor

Donde Objeto va el nombre del Control WinSock, el nombre predeterminado cuando lo incluimos en alguna aplicación es WinSock1. Luego le sigue la propiedad que deseamos asignar y finalmente el valor que la misma tomará.

Entonces por ejemplo si queremos probar la propiedad LocalIP debemos seguir el ejemplo 1.

Ejemplo 1_____

Crear un Proyecto (EXE Estándar) y agregar el WinSock Control. Luego agregar una etiqueta vacía, es decir un Label. Despues introducimos el siguiente código.

```
Private Sub Form_Load()

Label1.caption = WinSock1.LocalIP

End Sub
```

Este simple ejemplo nos da de forma rápida nuestro IP, aunque no estemos conectados a Internet el IP aparece igual, solo que siempre va a tomar el valor : 127.0.0.1

Ahora que sabemos manejar las propiedades podemos seguir con los Métodos. A continuación la lista de algunos de los Métodos más importantes del Control WinSock.

Lista de Métodos más importantes_____

Accept: Sólo para las aplicaciones de servidor TCP. Este método se utiliza para aceptar una conexión entrante cuando se está tratando un evento ConnectionRequest.

GetData: Recupera el bloque actual de datos y lo almacena en una variable de tipo Variant.

Listen: Crea un socket y lo establece a modo de escucha.

SendData: Envía datos a un equipo remoto.

Lista de Eventos más importantes_____

ConnectionRequest: Se produce cuando el equipo remoto solicita una conexión. Sin este evento no se puede llevar a cabo la conección.

Connect: Se produce cuando el equipo local se conecta al equipo remoto y se establece una conexión.

Close: Se produce cuando el equipo remoto cierra la conexión. Las aplicaciones deben usar el método Close para cerrar correctamente una conexión TCP.

DataArrival: Se produce cuando llegan nuevos datos. Este evento es importante, ya que debemos hacer algo con la información que llega.

La sintaxis de los métodos y eventos es igual a la sintaxis de las propiedades, por lo cual no vamos a hacer referencia a ella.

Programando la primera aplicación Cliente/Servidor.

Conociendo las propiedades, métodos y eventos del Control WinSock podemos pasar a la engorrosa labor de la programación.

Para poder programar la siguiente aplicación necesitan tener el Control WinSock en el formulario, eso siempre es fundamental para que el programa ande.

Para enterder el correcto funcionamiento del protocolo TCP/IP vamos a empezar por programar la aplicación Servidor a la cual luego se conectará el Cliente.

Comenzamos por crear un proyecto nuevo (EXE estándar) para el Servidor, y agregamos la siguiente lista de controles al formulario principal. La ubicación de dichos controles es a gusto del programador, siempre tratando de que el usuario final este a gusto con el producto y que se pueda manejar libremente sin problemas por el entorno del mismo.

- WinSock Control
- 2 cajas de texto (TextBox)
- 2 botones.

A continuación hace falta que cambiemos algunas propiedades de los controles, debajo la lista de controles con las respectivas propiedades a cambiar.

Control ***(nombre predeterminado)***	**Propiedad** ***(nuevo valor)***
WinSock1	LocalPort = 888
Text1	Text =
Text2	Text =
Command1	Caption = Escuchar
Command2	Caption = Enviar

Para que el ejemplo funcione a la perfección conviene que seguir la ubicación de los controles como esta indicado en la figura 1.

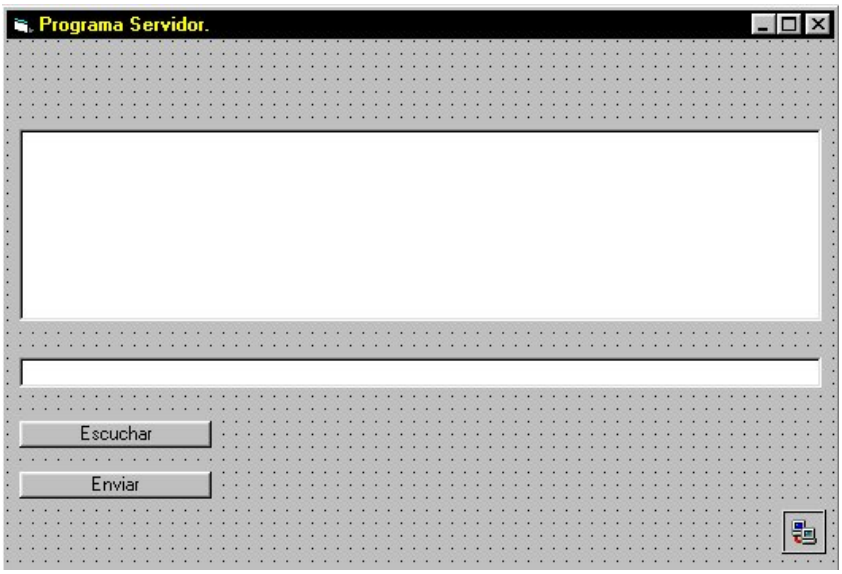

Figura 1

Una vez hecho esto podemos empezar a tipear código. El sangrado del programa es una cuestión de entendimiento para el programador, algunos recurren a éste como otros no, eso también queda a criterio del que programa.

En el Evento Click del Command1 incluimos el siguiente código; (sólo lo que esta en **NEGRITA**, el resto es en modo de ayuda, ya que aparece cuando se hace doble click en algun control).

Private Sub Command1_Click()

Winsock1.Listen

End Sub

Esto hace que el Control WinSock empiece a funcionar, escuchando el puerto que se indicó en las propiedades de dicho control. Este puerto es el 888. Ahora si realizamos todo a la perfección el puerto 888 esta siendo vigilado para aceptar conexiones remotas.

Luego en el Evento DataArrival del WinSock;

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

Dim datos As String

Winsock1.GetData datos

Text1.Text = Text1.Text + datos

End Sub

Datos queda transformada en una variable de cadena, y WinSock almacena los datos que recibe del Cliente en el buffer y luego ingresan a la variable datos, dicha variable mostrará su contenido en el control TextBox (Text1).

En el evento ConnectionRequest;

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)

Winsock1.Close

Winsock1.Accept requestID

End Sub

Este evento es muy importante, permite aceptar la petición de conexión. Sin este evento el resto del programa no tendría efecto.

En el evento Click del command2;

Private Sub Command2_Click()

Dim enviar As String

enviar = Text2.Text

Winsock1.SendData enviar

End Sub

Esto permite enviar el texto que se introduzca en el TextBox número 2.

Por ahora este es un simple programa Servidor, lo que hace es: designar un puerto, dejarlo a la escucha para aceptar conexiones, si se realiza una petición de conexión aceptarla, y por último enviar datos al Cliente y recibir los datos que éste mande.

Para seguir programando el **Cliente** hace falta crear un nuevo proyecto y en el formulario principal incluir la siguiente lista de controles:

- WinSock Control
- 3 cajas de texto (TextBox)
- 2 botones.

Como lo hicimos anteriormente hace falta cambiar algunas propiedades. Debajo la lista de controles con las respectivas propiedades para cambiar.

Control ***(nombre predeterminado)***	**Propiedad** ***(nuevo valor)***
WinSock1	RemotePort = 888
Text1	Text =
Text2	Text =
Text3	Text =
Command1	Caption = Conectar
Command2	Caption = Enviar

Para tener una referencia de cómo situar los controles conviene seguir la Figura 2.

Figura 2

En el método del command1;

Private Sub Command1_Click()

Winsock1.RemoteHost = Text3.Text

Winsock1.Connect

End Sub

El evento connect permite conectar al programa servidor que esta esperando la solicitud, este evento requiere un parámetro fundamental, el IP o nombre de host el cual es introducido previamente a la conexión en el cuadro de texto número 3 (Text3).

En el evento DataArrival del WinSock Control;

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

Dim datos As String

Winsock1.GetData datos

Text1.Text = Text1.Text + datos

End Sub

Esto permite a la aplicación (a través de WinSock) recibir información del servidor y mostrarla en pantalla.

En el método del command2;

Private Sub Command2_Click()

Dim enviar As String

enviar = Text2.Text

Winsock1.SendData enviar

End Sub

Estas instrucciones son necesarias para enviar información al servidor.

Este ejemplo del primer programa Cliente / Servidor es muy simple, para utilizarlo al máximo es necesario por ejemplo poner las propiedades de los TextBox en Multiline, lo que hace que si los datos recibidos exceden el tamaño del TextBox estos datos vayan directo a la línea de abajo.

Bibliografía.

Toda la información aquí expuesta fue escrita por Insomnia y fue tomada de dos diferentes fuentes, libros e Internet.

Libros.

Baltazar & Mariano Birnios, *Creación de Aplicaciones Multimedia con Visual Basic*, editorial MP Ediciones, primera edición, Buenos Aires, 1998.

Baltazar & Mariano Birnios, *Microsoft Visual Basic Manual de Referencia*, editorial MP Ediciones, primera edición, Buenos Aires, 1999.

Internet.

En Inglés.

http://msdn.microsoft.com/vb
http://www.vb−herlper.com

http://www.vbexplorer.com

http://www.cgvb.com

http://www.vb−world.net

http://www.planet−source−code.com/vb

http://www.programmersheaven.com/

En Castellano.

http://www.programacion.net/

http://www.abcdatos.com

http://www.monografias.com

Revisando estas direcciones de Internet se pueden adquirir algunos tutoriales, la mayoría en Ingles, pero no dejan de ser buenos. Estos tutoriales son una especie de guía en formato de texto, paso por paso, son muy famosos entre programadores y gente que recién empieza por la sencillez y los prácticos ejemplos que brindan.

Aplicaciones Cliente/Servidor en Visual Basic.

Aplicaciones Cliente/Servidor en Visual Basic.

_____2

_____1