

MANUAL TECNICO

Para la implementar el Sistema de Educación se utilizó Visual Basic 6.0, en donde se utilizó herramientas propias de este programa para de esta manera satisfacer los requerimientos iniciales del sistema. A continuación se describen los módulos del sistema y su respectiva programación:

CONTROL MASTER

Este módulo contiene lo que concierne a la administración del sistema: Acceso a los datos y obtención de los mismos, mediante la Base de Datos creada en SQL 2000 Server.

Las funciones que se utilizaron para acceder a los datos son las siguientes:

BUSCA ACCESO PARA ESTUDIANTES Y PROFESORES

```
Public Sub buscaacceso1(pval1 As String, pval2 As String)
    Dim abre As Integer

    Dim deco As String
    Dim dbt As ADODB.Connection
    Dim rpa As ADODB.Recordset, rea As ADODB.Recordset
    servopera = 0

    Set dbt = New ADODB.Connection
    Set rpa = New ADODB.Recordset
    Set rea = New ADODB.Recordset

    dbt.ConnectionString = "dsn=educacion"
    dbt.Open

    rea.Open "select * from Estudiantes", dbt, adOpenStatic, adLockReadOnly, adCmdText
    rpa.Open "select * from Profesores", dbt, adOpenStatic, adLockReadOnly, adCmdText

    deco = DecodeStr64(pval2)
    'Profesores
    With rpa
        Do While Not .EOF
            If LCase(pval1) = LCase(!Nombre_Usuario) And LCase(deco) = LCase(!Password) Then
```

```

    acepto = True
    idprofesor = !id_profesor
    idestudiante = ""
    apr = 1
    Exit Do
Else
    acepto = False
End If
.MoveNext
Loop
End With
'Estudiantes
If acepto = False Then
With rea
    Do While Not .EOF
        If LCase(pval1) = LCase(!Nombre_Usuario) And LCase(deco) = LCase(!Password) Then
            acepto = True
            idestudiante = !id_estudiante
            idprofesor = ""
            apr = 2
            Exit Do
        Else
            acepto = False
        End If
    .MoveNext
    Loop
End With
End If
rpa.Close
Set rpa = Nothing
rea.Close
Set rea = Nothing
dbt.Close
Set dbt = Nothing
End Sub

```

BUSCA ACCESO DIFERENTES MATERIAS O CURSOS PARA PROFESORES

```

Public Sub buscomateprof(idprofe As String)
Dim dbt As ADODB.Connection
Dim rbmp As ADODB.Recordset
Dim rsmateria As ADODB.Recordset
Dim rsidmate As ADODB.Recordset
Dim rsidniv As ADODB.Recordset
Dim rsniveles As ADODB.Recordset

Set dbt = New ADODB.Connection

```

```
Set rbmp = New ADODB.Recordset
Set rsmateria = New ADODB.Recordset
Set rsidmate = New ADODB.Recordset
Set rsidniv = New ADODB.Recordset
Set rsniveles = New ADODB.Recordset

dbt.ConnectionString = "dsn=educacion"
dbt.Open

If idprofe <> "" Then
    rbmp.Open "select * from ProfMateNiv where [Id_profesor]='" & idprofe & "' ", dbt,
adOpenStatic, adLockReadOnly, adCmdText
    With rbmp
        Do While Not .EOF
            rsidmate.Open "select [Id_Materia] from Materias where [Id_Materia]='" & rbmp!id_materia
& "' ", dbt, adOpenDynamic, adLockOptimistic, adCmdText 'Materias
            rsmateria.Open "select [Descripcion] from Materias where [Id_Materia]='" & rbmp!id_materia
& "' ", dbt, adOpenStatic, adLockReadOnly, adCmdText 'Materias
            rsidniv.Open "select [Id_Nivel] from Niveles where [Id_Nivel]='" & rbmp!id_nivel & "' ", dbt,
adOpenStatic, adLockReadOnly, adCmdText 'Niveles
            rsniveles.Open "select [Descripcion] from Niveles where [Id_Nivel]='" & rbmp!id_nivel & "' ",
dbt, adOpenStatic, adLockReadOnly, adCmdText 'Niveles

            desidmate = desidmate & rsidmate(0) & "~"
            desmate = desmate & rsmateria(0) & "~"
            desidniv = desidniv & rsidniv(0) & "~"
            desniv = desniv & rsniveles(0) & "~"

            rsidmate.Close
            rsmateria.Close
            rsidniv.Close
            rsniveles.Close
            .MoveNext
        Loop
    End With
    rbmp.Close
    Set rbmp = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
```

BUSCA ACCESO DIFERENTES MATERIAS O CURSOS PARA ESTUDIANTES

```
Public Sub buscomateestu(idestu As String)
Dim dbt As ADODB.Connection
```

```
Dim rbmp As ADODB.Recordset
Dim rsmateria As ADODB.Recordset
Dim rsidmate As ADODB.Recordset
Dim rsidniv As ADODB.Recordset
Dim rsniveles As ADODB.Recordset
Dim rsmatinsc As ADODB.Recordset
Dim rsinsc As ADODB.Recordset
Dim rsinsniv As ADODB.Recordset
```

```
Dim codins As String
Dim insniv1 As String
```

```
'seteo lo que declare arriba
Set dbt = New ADODB.Connection
Set rbmp = New ADODB.Recordset
Set rsidmate = New ADODB.Recordset
Set rsmateria = New ADODB.Recordset
Set rsidniv = New ADODB.Recordset
Set rsniveles = New ADODB.Recordset
Set rsmatinsc = New ADODB.Recordset
Set rsinsc = New ADODB.Recordset
Set rsinsniv = New ADODB.Recordset
```

```
dbt.ConnectionString = "dsn=educacion"
dbt.Open
```

```
If idestu <> "" Then
    idpro1 = ""
    rsinsc.Open "SELECT [Cod_Inscripcion] From Inscripciones WHERE [Id_estudiante]= " & idestu
    & " ", dbt, adOpenStatic, adLockReadOnly, adCmdText
```

```
    codins = rsinsc(0)
    rsinsc.Close
```

```
    rsinsc.Open "SELECT [Id_nivel] From Inscripciones WHERE [Id_estudiante]= " & idestu & " ",
    dbt, adOpenStatic, adLockReadOnly, adCmdText
```

```
    insniv1 = rsinsc(0)
```

```
    rsmatinsc.Open "SELECT [id_profesor] From MatInsc WHERE [Cod_inscripcion]= " & codins &
    "" ", dbt, adOpenStatic, adLockReadOnly, adCmdText
    idpro1 = rsmatinsc(0)
```

```
    rbmp.Open "SELECT * From ProfMateNiv WHERE [Id_profesor]= " & idpro1 & "" and
    [Id_nivel]= " & insniv1 & "" ", dbt, adOpenStatic, adLockReadOnly, adCmdText
```

```
desidmate = ""
desmate = ""
desidniv = ""

With rbmp
Do While Not .EOF
    rsidmate.Open "SELECT [Id_Materia] From Materias WHERE [Id_Materia]= '" &
rbmp!id_materia & "' ", dbt, adOpenStatic, adLockReadOnly, adCmdText "Materias"
    rsmateria.Open "SELECT [Descripcion] From Materias WHERE [Id_Materia]= '" &
rbmp!id_materia & "' ", dbt, adOpenStatic, adLockReadOnly, adCmdText "Materias"
    rsidniv.Open "SELECT [Id_Nivel] From Niveles WHERE [Id_Nivel]= '" & rbmp!id_nivel & "' ",
dbt, adOpenStatic, adLockReadOnly, adCmdText "Niveles"
    rsniveles.Open "SELECT [Descripcion] From Niveles WHERE [Id_Nivel]= '" & rbmp!id_nivel &
"' ", dbt, adOpenStatic, adLockReadOnly, adCmdText "Niveles"

    desidmate = desidmate & rsidmate(0) & "~"
    desmate = desmate & rsmateria(0) & "~"
    desidniv = desidniv & rsidniv(0) & "~"
    desniv = desniv & rsniveles(0) & "~"

    rsidmate.Close
    rsmateria.Close
    rsidniv.Close
    rsniveles.Close
    .MoveNext
Loop
End With
rsinsc.Close
Set rdinsc = Nothing
rsmatinsc.Close
Set rsmatinsc = Nothing
rbmp.Close
Set rbmp = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
```

ACCESO A LA BASE DE DATOS PARA EXTRAER LISTADOS DE NOTAS DE LOS ESTUDIANTES

```
Public Sub listadonotas(lidmat As String, lidnivel As String, lidpo As String)
Dim dbt As ADODB.Connection
Dim rslistado As ADODB.Recordset
Dim rslistestu As ADODB.Recordset
Dim rslistnot As ADODB.Recordset
Dim rsnomestu As ADODB.Recordset
```

```
Dim nomauxes As String
Set dbt = New ADODB.Connection
Set rslistado = New ADODB.Recordset
Set rslistestu = New ADODB.Recordset
Set rslistnot = New ADODB.Recordset
Set rsnomestu = New ADODB.Recordset
dbt.ConnectionString = "dsn=educacion"
dbt.Open
If lidpo <> "" Then
    listnombres = ""
    listnot1 = ""
    rslistado.Open "SELECT * from MatInsc WHERE [Id_profesor]= " & lidpo & " and [Id_Nivel]=" & lidnivel & " and [Id_materia]= " & lidmat & " ", dbt, adOpenStatic, adLockReadOnly, adCmdText
    With rslistado
        Do While Not .EOF
            rslistestu.Open "SELECT [Cod_Inscripcion], [Id_Estudiante] From Inscripciones WHERE [Cod_Inscripcion]= " & rslistado!Cod_Inscripcion & " ", dbt, adOpenStatic, adLockReadOnly, adCmdText
            rsnomestu.Open "SELECT [Id_Estudiante], [Nombres], [Apellidos] from Estudiantes WHERE [Id_estudiante]= " & rslistestu(1) & " ", dbt, adOpenStatic, adLockReadOnly, adCmdText
            nomauxes = rsnomestu(0) & "_" & rsnomestu(1) & Chr(32) & rsnomestu(2)
            listnombres = listnombres & nomauxes & "|"
            rsnomestu.Close
            rslistestu.Close
            .MoveNext
        Loop
    End With
    rslistado.Close
    Set rslistado = Nothing
    dbt.Close
    Set dbt = Nothing
    Set rsnomestu = Nothing
    Set rslistestu = Nothing
End If
End Sub
```

GUARDAR EXAMENES DE ACUERDO A LOS DOCENTES Y MATERIAS ASIGNADAS

```
Public Sub examenes(eidmat As String, eidniv As String, eidpro As String, pastem As String, edesexa As String)
Dim dbt As ADODB.Connection
Dim rsexsa As ADODB.Recordset
Dim rexa, numexa As Integer

numexa = 1
```

```
Set dbt = New ADODB.Connection
Set rsex = New ADODB.Recordset
dbt.ConnectionString = "dsn=educacion"
dbt.Open
If eidpro <> "" Then
    rsex.Open "SELECT * from Examatprofniv WHERE [Id_profesor]= " & eidpro & " ", dbt,
adOpenDynamic, adLockOptimistic, adCmdText
    rsex = rsex.RecordCount + 1
    With rsex
        If Not (.BOF And .EOF) Then
            Do While Not .EOF
                numexa = numexa + 1
                'slastrecord = .Bookmark
                .MoveNext
            Loop
        End If
        .AddNew
        !id_profesor = Trim(eidpro)
        !id_materia = Trim(eidmat)
        !id_nivel = Trim(eidniv)
        !Id_examen = numexa 'Trim(pastem & Trim(Str(rsex)) & "x") & numexa
        !descripcion = Trim(edesexa)
        desexa = Trim(edesexa)
        .Update
    End With
    idexamen = numexa 'Trim(pastem & Trim(Str(rsex)) & "x")
    rsex.Close
    Set rsex = Nothing
End If
dbt.Close
Set rsex = Nothing
End Sub
```

GUARDAR LAS PREGUNTAS QUE CONTIENEN LOS EXAMENES

```
Public Sub guardapreguntas(idexam As String, idpreg As String, preg As String, opc1 As String,
opc2 As String, opc3 As String, opc4 As String, opc5 As String, respu As String, tipopreg As
String, valorpreg As String, tiemposresp As String)
Dim dbt As ADODB.Connection
Dim pregexa As ADODB.Recordset

Set dbt = New ADODB.Connection
Set pregexa = New ADODB.Recordset

dbt.ConnectionString = "DSN=educacion"
dbt.Open
```

```
If idexam <> "" Then
    pregexa.Open "SELECT * from Examenes_linea ", dbt, adOpenDynamic, adLockOptimistic,
    adCmdText

    With pregexa
        If Not (.BOF And .EOF) Then
            'slastrecord = .Bookmark
            .MoveLast
        End If
        .AddNew
        !Id_examen = Trim(idexam)
        !descripcion = desexa
        !Id_pregunta = Val(idpreg)
        !pregunta = Trim(preg)
        !op1 = Trim(opc1)
        !op2 = Trim(opc2)
        !op3 = Trim(opc3)
        !op4 = Trim(opc4)
        !op5 = Trim(opc5)
        !respuesta = Val(respu)
        !tipopregunta = Trim(tipopreg)
        !Valor_pregunta = Val(valorpreg)
        !tiempo_respuesta = Val(tiemporesp)
        .Update
    End With
    pregexa.Close
    Set pregexa = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
```

FUNCION QUE PERMITE ACCEDER A LOS EXAMENES QUE SOLICITA EL ESTUDIANTE PARA CONTESTARLOS

```
Public Sub solicitaexa(soidpro As String, soidnivel As String, soidmate As String, soidestud As String)
    Dim dbt As ADODB.Connection
    Dim rsexamp As ADODB.Recordset
    Dim rspun As ADODB.Recordset
    Dim contes, nexa, cont As Integer
    Dim contestado(10) As String
    Set dbt = New ADODB.Connection
    Set rsexamp = New ADODB.Recordset
    Set rspun = New ADODB.Recordset
```



```
dbt.ConnectionString = "dsn=educacion"
dbt.Open
idsoexamen = ""
contestados = ""
contes = 0
cont = 0
If soidpro <> "" Then
    rsexamp.Open "SELECT * from Examatprofniv WHERE [Id_profesor]= " & soidpro & " and
[Id_materia]= " & soidmate & " and [Id_nivel]= " & soidnivel & " ", dbt, adOpenStatic,
adLockReadOnly, adCmdText
    rspunt.Open "SELECT * from Puntajes WHERE [Id_estudiante]= " & soidestud & " ", dbt,
adOpenStatic, adLockReadOnly, adCmdText
    With rsexamp
        If Not (.BOF And .EOF) Then
            Do While Not .EOF
                nexa = nexa + 1
                .MoveNext
            Loop
        Else
            nexa = 1
        End If
        .MoveFirst

        Do While Not .EOF
            cont = cont + 1
            With rspunt
                If contes <> 0 Then
                    .MoveFirst
                End If
                If Not (rspunt.BOF And rspunt.EOF) Then
                    Do While Not .EOF
                        If rsexamp!Id_examen = rspunt!Id_examenlinea And rsexamp!contestado = "1" Then
                            contes = contes + 1
                            contestado(contes) = rsexamp!Id_examen
                        End If
                        .MoveNext
                    Loop
                Else
                    idsoexamen = idsoexamen & rsexamp!Id_examen & "_" & rsexamp!descripcion & "|"
                End If
            End With
            If contes <> 0 Then
                If rsexamp!Id_examen <> contestado(contes) Then
                    idsoexamen = idsoexamen & rsexamp!Id_examen & "_" & rsexamp!descripcion & "|"
                End If
            End If
        End If
    End With
End If
```

```
.MoveNext
Loop
End With
rspunt.Close
rsexamp.Close
Set rspunt = Nothing
Set rsexamp = Nothing
contestados = Str(contes)
End If
dbt.Close
Set dbt = Nothing
End Sub
Public Sub pidepregunta(pidexa As String)
Dim dbt As ADODB.Connection
Dim preg As ADODB.Recordset
Set dbt = New ADODB.Connection
Set preg = New ADODB.Recordset

dbt.ConnectionString = "dsn=educacion"
dbt.Open

npreg = 0
tpreg = 0

If pidexa <> "" Then
    preg.Open "SELECT * from Examenes_linea WHERE [Id_examen]= " & pidexa & " ", dbt,
adOpenStatic, adLockReadOnly, adCmdText
    npreg = preg.RecordCount + 1

    If npreg > 0 Then
        With preg
            Call limpiaexamen
            Do While Not .EOF
                idpregunta = idpregunta & !Id_pregunta & "|"
                pregunta = pregunta & !pregunta & "|"
                op1 = op1 & !op1 & "|"
                op2 = op2 & !op2 & "|"
                op3 = op3 & !op3 & "|"
                op4 = op4 & !op4 & "|"
                op5 = op5 & !op5 & "|"
                respuesta = respuesta & !respuesta & "|"
                tipopregunta = tipopregunta & !tipopregunta & "|"
                valorpregunta = valorpregunta & !Valor_pregunta & "|"
                tiemposrespuesta = tiemposrespuesta & !tiempo_respuesta & "|"
                tpreg = tpreg + !tiempo_respuesta
            .MoveNext
        Loop
    End If
End If
End Sub
```

```
End With
End If
preg.Close
Set preg = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
Private Sub limpiaexamen()

idpregunta = ""
pregunta = ""
op1 = ""
op2 = ""
op3 = ""
op4 = ""
op5 = ""
respuesta = ""
tipopregunta = ""
valorpregunta = ""
tiemporespuesta = ""
tpreg = 0
End Sub
```

PERMITE ALMACENAR LAS RESPUESTAS A CADA UNA DE LAS PREGUNTAS

```
Public Sub almacenarespuesta(respidexam As String, repidestu As String, puntestu As String,
contestu As Boolean)
Dim dbt As ADODB.Connection
Dim rsrespuesta As ADODB.Recordset
Dim rsexam As ADODB.Recordset
Dim ayuda As String

Set dbt = New ADODB.Connection
Set rsrespuesta = New ADODB.Recordset
Set rsexam = New ADODB.Recordset

dbt.ConnectionString = "dsn=educacion"
dbt.Open
If respidexam <> "" Then
rsrespuesta.Open "SELECT * from Puntajes", dbt, adOpenDynamic, adLockOptimistic,
adCmdText
rsexam.Open "SELECT * from Examatprofniv", dbt, adOpenDynamic, adLockOptimistic,
adCmdText

With rsrespuesta
If Not (.BOF And .EOF) Then
```

```
'slastrecord = .Bookmark
.MoveLast
End If
.AddNew
!id_examenlinea = Trim(respidexam)
ayuda = !id_examenlinea
!id_estudiante = Trim(repidestu)
!Puntaje = Val(puntestu)
!contestado = 0 'contestu
.Update
End With

With rsexam
Do While Not .EOF
If rsexam!Id_examen = ayuda Then
!contestado = 1

Exit Do
End If
.MoveNext
Loop
.Update
End With
rsrespuesta.Close
Set rsrespuesta = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
```

ACCEDER A LOS EXAMENES QUE HAN SIDO CONTESTADOS

```
Public Sub vercontestados(vidp As String, vidniv As String, vidmat As String)
Dim dbt As ADODB.Connection
Dim rsconex As ADODB.Recordset

Set dbt = New ADODB.Connection
Set rsconex = New ADODB.Recordset

dbt.ConnectionString = "DSN=educacion"
dbt.Open

If vidp <> "" Then
rsconex.Open "SELECT * from Examatprofniv WHERE [Id_profesor]= "" & vidp & "" and
[Id_materia]= "" & vidmat & "" and [Id_nivel]= "" & vidniv & "" ", dbt, adOpenStatic,
adLockReadOnly, adCmdText
```

```
With rsconex
  Do While Not .EOF
    enviexacon = enviexacon & !Id_examen & "_" & !descripcion & "|"
    .MoveNext
  Loop
End With
rsconex.Close
Set rsconex = Nothing
End If
dbt.Close
Set dbt = Nothing
End Sub
```

VISUALIZAR LOS PUNTAJES DE CADA EXAMEN

```
Public Sub verpuntajes(pidpro1 As String, pidnivel1 As String, pidmat1 As String, pidestu1 As String)
  Dim dbt As ADODB.Connection
  Dim preg As ADODB.Recordset
  Dim preg1 As ADODB.Recordset

  Dim sqlwhat As String, sqlwhere As String
  Dim sqlstring As String, vl As String

  Set dbt = New ADODB.Connection
  Set preg = New ADODB.Recordset
  Set preg1 = New ADODB.Recordset

  dbt.ConnectionString = "DSN=educacion"
  dbt.Open

  vl = "-1"
  sqlwhat = "SELECT Examatprofniv.Id_examen as idexaml, Examatprofniv.Descripcion as descripl, Puntajes.Puntaje as puntajel FROM Examatprofniv" _
    & " INNER JOIN Puntajes ON Examatprofniv.Id_examen = Puntajes.Id_examenlinea "
  sqlwhere = "WHERE (((Examatprofniv.Id_profesor)='" & pidpro1 & "') AND ((Examatprofniv.Id_materia)='" & pidmat1 & "') AND ((Examatprofniv.Id_nivel)='" & pidnivel1 & "') AND ((Puntajes.Id_estudiante)='" & pidestu1 & "')) 'AND ((Puntajes.contestado)=' " & vl & "'))"
  sqlstring = sqlwhat & sqlwhere

  preg.Open sqlstring, dbt, adOpenStatic, adLockReadOnly, adCmdText
  totpunta = ""

  With preg
    Do While Not .EOF
      totpunta = totpunta & !idexaml & "_" & !descripl & "_" & Str(!puntajel) & "|"
    
```

```
.MoveNext  
Loop  
End With  
preg.Close  
Set preg = Nothing  
dbt.Close  
Set dbt = Nothing  
End Sub
```

ESTUDIANTE Y PROFESOR

Estos módulos contienen varias herramientas que son iguales para los dos, a continuación se describen algunas:

VOZ

Declaraciones

```
Option Explicit  
Public CLOSINGAPPLICATION As Boolean ' Bandera del estado de la aplicación  
Public wStream As Object
```

Activa el reproductor del Audio

```
Private Sub cmdTalk_Click()  
Dim rc As Long ' Retorna un código de variable  
Dim iPort As Integer ' Puerto Local  
Dim itm As Integer ' Actual Item  
If (Not wStream.Playing And wStream.PlayDeviceFree And _  
Not wStream.Recording And wStream.RecDeviceFree) Then ' Estado de validación del  
dispositivo de audio  
wStream.Playing = True ' Reproduciendo el audio  
cmdtalk.Caption = "&Escuchando"  
Screen.MousePointer = vbHourglass  
iPort = wStream.StreamInQueue  
Do While (iPort <> NULLPORTID) ' Mientras los puertos del socket tienen datos  
en el reproductor de audio  
inLight.Picture = ImgIcons.ListImages(speakON).Picture  
inLight.Refresh  
For itm = 0 To ConnectionList.ListCount - 1 ' Busco el usuario actual para ejecutar los  
datos de sonido  
If (ConnectionList.ItemData(itm) = iPort) Then ' si lo encuentro  
ConnectionList.TopIndex = itm ' obtengo el índice del mismo  
ConnectionList.Selected(itm) = True ' Selecciono al miembro para presentar quien  
está hablando actualmente...  
Exit For
```

```
End If
Next

rc = wStream.PlayWave(Me.hWnd, iPort) ' ejecuto los datos wave que están en el
puerto...
Call wStream.RemoveStreamFromQueue(iPort) ' Remuevo el ID del Puerto desde la
cola de espera (playwave)
iPort = wStream.StreamInQueue
inLight.Picture = ImgIcons.ListImages(speakOFF).Picture
inLight.Refresh
Loop ' busco el siguiente socket que está en al cola de espera
(playwave)
ConnectionList.TopIndex = 0
If (ConnectionList.ListCount > 0) Then
ConnectionList.Selected(0) = True ' Deselecciono al item posterior
ConnectionList.Selected(0) = False ' deselecciono al item actualmente seleccionado
End If
Screen.MousePointer = vbDefault
cmdtalk.Caption = "&Hablar"
wStream.Playing = False ' Apago la ejecución del reproductor de audio
End If
End Sub
```

Activa la grabación del audio

```
Private Sub cmdTalk_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

Dim rc As Long

If (Not wStream.Playing And _
Not wStream.Recording And _
wStream.RecDeviceFree And _
wStream.PlayDeviceFree) Then ' Chequeo el estado del dispositivo de audio
wStream.Recording = True
cmdtalk.Caption = "&Hablando"
Screen.MousePointer = vbHourglass
outLight.Picture = ImgIcons.ListImages(mikeON).Picture
outLight.Refresh

rc = wStream.RecordWave(Me.hWnd, TCPSocket) ' Grabando voz y envio a todos los
sockets conectados

outLight.Picture = ImgIcons.ListImages(mikeOFF).Picture
outLight.Refresh
Screen.MousePointer = vbDefault
cmdtalk.Caption = "&Hablar"
```

```

    If Not wStream.Playing And _
        wStream.PlayDeviceFree And _
        wStream.RecDeviceFree Then          ' Está el dispositivo de audio libre?
        Call cmdTalk_Click                   ' Activa el Playback De Cualquier Mensaje Entrante...
    End If
End If
End Sub

```

Función que permite la grabación del sonido

```

Private Sub cmdTalk_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    wStream.Recording = False              ' Para la Grabación
End Sub

```

Controla que exista la conexión

```

Private Sub connecta_Click()
Dim Idx As Long
Idx = InstanceTCP(TCPSocket)              ' Control de Instacia de TCP

    If (Idx > 0) Then
        connecta.Enabled = False
        ConnectionList.Enabled = False

        On Error Resume Next
        If Not Connect(TCPSocket(Idx), txtServer.Text, VOICEPORT) Then ' Intento conectar
            Unload TCPSocket(Idx)      ' Conexión fallada descargo las instancias del control
        End If

        ConnectionList.Enabled = True
        connecta.Enabled = True
    End If
End Sub

```

Form Load, lo que se cargará al acceder el formulario

```

Private Sub Form_Load()

    Dim rc As Long
    Dim Idx As Long
    Dim TCPidx As Long

    CLOSINGAPPLICATION = False
    Call InitServerList(txtServer)          ' Obtengo la lista del servidor en comun
    txtServer.Text = txtServer.List(0)     ' Despliego el primer nombre en la lista
    imgStatus = ImgIcons.ListImages(phoneHungUp).Picture

    Set wStream = New WaveStream

```



```

Call wStream.InitACMCodec(WAVE_FORMAT_GSM610, TIMESLICE)

cmdtalk.Enabled = False           ' Deshabilito el control hasta conectar
Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
inLight.Picture = ImgIcons.ListImages(speakNO).Picture
outLight.Picture = ImgIcons.ListImages(mikeNO).Picture
Option1.Value = True
Call Listen(TCPsocket(0))         ' Escuchando para la conexión TCP/IP
End Sub

```

CAPTURA DE VIDEO

```

Public Const WS_BORDER = &H800000
Public Const WS_CAPTION = &HC00000
Public Const WS_SYSMENU = &H80000
Public Const WS_CHILD = &H40000000
Public Const WS_VISIBLE = &H10000000
Public Const WS_OVERLAPPED = &H0&
Public Const WS_MINIMIZEBOX = &H20000
Public Const WS_MAXIMIZEBOX = &H10000
Public Const WS_THICKFRAME = &H40000
Public Const WS_OVERLAPPEDWINDOW = (WS_OVERLAPPED Or WS_CAPTION Or
WS_SYSMENU Or WS_THICKFRAME Or WS_MINIMIZEBOX Or WS_MAXIMIZEBOX)
'Public Const SWP_NOMOVE = &H2
'Public Const SWP_NOSIZE = 1
Public Const SWP_NOZORDER = &H4
Public Const HWND_BOTTOM = 1
'Public Const HWND_TOPMOST = -1
'Public Const HWND_NOTOPMOST = -2
Public Const SM_CYCAPTION = 4
Public Const SM_CXFRAME = 32
Public Const SM_CYFRAME = 33
Public Const WS_EX_TRANSPARENT = &H20&
Public Const GWL_STYLE = (-16)
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hWnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

'// Manipulación de Memoria
Declare Function IStrCpy Lib "kernel32" Alias "IstrcpyA" (ByVal lpString1 As Long, ByVal lpString2
As Long) As Long
Declare Function IStrCpyn Lib "kernel32" Alias "IstrcpynA" (ByVal lpString1 As Any, ByVal
lpString2 As Long, ByVal iMaxLength As Long) As Long
Declare Sub RtlMoveMemory Lib "kernel32" (ByVal hpvDest As Long, ByVal hpvSource As Long,
ByVal cbCopy As Long)
Declare Sub hmemcpy Lib "kernel32" (hpvDest As Any, hpvSource As Any, ByVal cbCopy As
Long)

```

```
// Manipulación de ventanas
Declare Function SetWindowPos Lib "user32" (ByVal hWnd As Long, ByVal hWndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As Boolean
Declare Function GetSystemMetrics Lib "user32" (ByVal nIndex As Long) As Long
Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hWnd As Long, ByVal lpString As String) As Long

Public hWndC As Long      ' Cabecera para laventana de captura

Function MyFrameCallback(ByVal hWnd As Long, ByVal lpVHdr As Long) As Long
    Debug.Print "FrameCallBack"
    Dim VideoHeader As VIDEOHDR
    Dim VideoData() As Byte
    RtlMoveMemory VarPtr(VideoHeader), lpVHdr, Len(VideoHeader)
    ReDim VideoData(VideoHeader.dwBytesUsed)

    //Copia datos dentro del array
    RtlMoveMemory VarPtr(VideoData(0)), VideoHeader.lpData, VideoHeader.dwBytesUsed
    Debug.Print VideoHeader.dwBytesUsed
    Debug.Print VideoData
End Function

Function MyYieldCallback(hWnd As Long) As Long
    Debug.Print "Yield"
End Function

Function MyErrorCallback(ByVal hWnd As Long, ByVal iID As Long, ByVal ipstrStatusText As Long) As Long
    If iID = 0 Then Exit Function
    Dim sStatusText As String
    Dim usStatusText As String
    sStatusText = String$(255, 0)
    IStrCpy StrPtr(sStatusText), ipstrStatusText
    sStatusText = Left$(sStatusText, InStr(sStatusText, Chr$(0)) - 1)
    usStatusText = StrConv(sStatusText, vbUnicode)
    LogError usStatusText, iID
End Function

Function MyStatusCallback(ByVal hWnd As Long, ByVal iID As Long, ByVal ipstrStatusText As Long) As Long

    If iID = 0 Then Exit Function
    Dim sStatusText As String
    Dim usStatusText As String
```

```
sStatusText = String$(255, 0)
IStrCpy StrPtr(sStatusText), ipstrStatusText
sStatusText = Left$(sStatusText, InStr(sStatusText, Chr$(0)) - 1)
usStatusText = StrConv(sStatusText, vbUnicode)

frmVideo.StatusBar.SimpleText = usStatusText
Debug.Print "Status: ", usStatusText, IID

Select Case IID '
End Select
End Function

Sub ResizeCaptureWindow(ByVal lwnd As Long)

Dim CAPSTATUS As CAPSTATUS
Dim lCaptionHeight As Long
Dim lX_Border As Long
Dim lY_Border As Long
lCaptionHeight = GetSystemMetrics(SM_CYCAPTION)
lX_Border = GetSystemMetrics(SM_CXFRAME)
lY_Border = GetSystemMetrics(SM_CYFRAME)
If capGetStatus(lwnd, VarPtr(CAPSTATUS), Len(CAPSTATUS)) Then
SetWindowPos lwnd, HWND_BOTTOM, 0, 0, _
CAPSTATUS.uiImageWidth + (lX_Border * 2), _
CAPSTATUS.uiImageHeight + lCaptionHeight + (lY_Border * 2), _
SWP_NOMOVE Or SWP_NOZORDER
End If
Debug.Print "Resize Window."
End Sub

Function MyVideoStreamCallback(lwnd As Long, lpVHdr As Long) As Long
Beep
End Function

Function MyWaveStreamCallback(lwnd As Long, lpVHdr As Long) As Long
Debug.Print "WaveStream"
End Function

Sub LogError(txtError As String, IID As Long)
frmVideo.StatusBar.SimpleText = txtError
Debug.Print "Error: ", txtError, IID
End Sub
```

GRABA MOVIMIENTOS DEL MOUSE

```
Private Sub cmdRecord_Click()  
On Error GoTo Error  
  
    'Calcule el número total de Muestras para ser Grabado  
    Samples = SPS * Val(InputBox("Número de segundos para la grabación :", "No Cambiar la  
Resolución de la Pantalla mientras Graba"))  
  
    If Samples <= 0 Then Exit Sub      'Aborta la grabación sin nada grabado  
    i = 0                             'Inicializa contador de muestras  
    ReDim Cursor(Samples)             'Redimenciono el estdo del array del cursor  
    UpdateControls False, False, True, False 'Actualiza controles  
    UnSaved = True                   'Movimientos todavía no grabados  
    HW = CBool(chkHide.Value)        'Graba la opción para aplicar a la reproducción  
    If HW Then Me.Hide               'Debe Esconder la Ventana mientras está Grabando  
    Exit Sub  
  
Error:  
MsgBox "Error, Tiempo de Grabación demasiado largo!", vbCritical, "Error"  
End Sub  
Private Sub cmdPlay_Click()  
    'Verifique para ver si la actual resolución empareja la resolución del archivo grabado  
    If Res <> CurrentResolution Then  
        If MsgBox("Su resolución de la pantalla actual no empareja la resolución del archivo a  
reproducir. ¿Está seguro que desea continuar?", vbCritical Or vbYesNo, "No se adapta a la  
resolución") = vbNo Then Exit Sub  
    End If  
  
    i = 1                             'Inicializa el reproductor  
    If j <= 0 Then Exit Sub            'Aborta el reproductor si no se está ejecutando  
    UpdateControls False, False, False, True 'Actualiza Controles  
    If HW Then Me.Hide               'Debe Esconder la Ventana mientras está Grabando  
End Sub  
Private Sub mnuFileExit_Click()  
    DoEvents  
    Unload Me  
End Sub  
Private Sub mnuFileNew_Click()  
    FreshForm  
End Sub  
Private Sub mnuFileOpen_Click()  
    ComDlg.ShowOpen      'Abre el cuadro de dialogo para recuperar el archivo grabado  
    FN = ComDlg.FileName  'Obtiene el nombre del archivo escogido  
    If FN = "" Then Exit Sub 'Verifica si ha seleccionado el archivo  
    LoadFile FN          'Ahora lee el archivo  
End Sub
```

```
Private Sub mnuFileSave_Click()  
    If FN = "" Then mnuFileSaveAs_Click: Exit Sub  
    SaveFile FN  
End Sub  
  
Private Sub mnuFileSaveAs_Click()  
    ComDlg.DialogTitle = "GUARDAR ARCHIVO"  
    ComDlg.ShowSave      'Abre el cuadro de diálogo para guardar el archivo  
    FN = ComDlg.FileName  'Obtiene el nombre del archivo escogido  
    If FN = "" Then Exit Sub 'Verifica si ha seleccionado el archivo  
    SaveFile FN          'Ahora lee el archivo  
End Sub  
  
Private Sub tmrRecord_Timer()  
    Record  
End Sub  
Private Sub tmrPlay_Timer()  
    Play  
End Sub
```

Estas funciones son llamadas en las líneas de código anteriores.

```
Public Sub Record()  
  
    GetCursorPos Cursor(i).pos          'Grabamos posición del cursor  
    Esc = CBool(GetAsyncKeyState(vbKeyEscape)) 'Monitoreamos la tecla ESC en caso  
de que el Usuario quiere parar  
    Cursor(i).LButton = CBool(GetAsyncKeyState(vbLeftButton)) 'Estado del Botón izquierdo  
    Cursor(i).MButton = CBool(GetAsyncKeyState(vbMiddleButton)) 'Estado del Botón centro  
    Cursor(i).RButton = CBool(GetAsyncKeyState(vbRightButton)) 'Estado del Botón derecho  
  
    If (i < Samples) And (Not Esc) Then  
        i = i + 1  
    Else  
        j = i - 1  
        UpdateControls True, True, False, False  
        MsgBox "Grabación Terminada.", vbInformation, "Terminado!"  
        Res = CurrentResolution  
        UpdateInfo FN, Res, CStr(Now), j / SPS, HW  
        frmMouserregistra.Show  
    End If  
  
End Sub  
  
Public Sub Play()  
  
    SetCursorPos Cursor(i).pos.X, Cursor(i).pos.Y
```

```
'regenera los eventos del botón izquierdo del mouse
If (Not pLB) And (Cursor(i).LButton) Then mouse_event MOUSEEVENTF_LEFTDOWN, 0, 0, 0,
0
If (pLB) And (Not Cursor(i).LButton) Then mouse_event MOUSEEVENTF_LEFTUP, 0, 0, 0, 0

'regenera los eventos del botón centro del mouse
If (Not pMB) And (Cursor(i).MButton) Then mouse_event MOUSEEVENTF_MIDDLEDOWN, 0, 0,
0, 0
If (pMB) And (Not Cursor(i).MButton) Then mouse_event MOUSEEVENTF_MIDDLEUP, 0, 0, 0,
0

'regenera los eventos del botón derecho del mouse
If (Not pRB) And (Cursor(i).RButton) Then mouse_event MOUSEEVENTF_RIGHTDOWN, 0, 0,
0, 0
If (pRB) And (Not Cursor(i).RButton) Then mouse_event MOUSEEVENTF_RIGHTUP, 0, 0, 0, 0

'en caso de parar presionamos la tecla ESC
Esc = CBool(GetAsyncKeyState(vbKeyEscape))

If (i < j) And (Not Esc) Then
    pLB = Cursor(i).LButton
    pMB = Cursor(i).MButton
    pRB = Cursor(i).RButton
    i = i + 1
Else
    UpdateControls True, True, False, False
    MsgBox "Reproducción Terminada.", vbInformation, "¡ Terminación !"
    frmMousseregistra.Show
End If

End Sub
```

TOMAR CONTROL DEL ESTUDIANTE (PROFESOR)

```
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal Y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Const SRCCOPY = &HCC0020

Dim RecDib As New cDIBSection
Dim ZLib As New clsZLib
Dim i As Long
Dim tmpPos() As String
Dim pos(1) As Long
Const CRate = 5
```

```
Private Sub Command1_Click()
'### Sale de la Captura
    Unload frmCapture
    frmMain.Visible = True
    modClient.ReconnectToServer
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
'### Envía el código de la tecla precionada
'### y espera para responder
Dim bKeys As Byte
    bKeys = CByte(KeyCode)
    TCP_KB.SendData bKeys
    Do Until KbReply
        DoEvents
    Loop
    KbReply = False
End Sub

Private Sub Form_Load()
'### Obtiene el reciente desktop del servidor
    modClient.Reply = False
    frmMain.wskMain.SendData modClient.GetRect
'### Espera para replicar
    Do Until modClient.Reply
        DoEvents
    Loop
    modClient.Reply = False
'### pone la forma en la esquina superior izquierda
    Me.Top = 0
    Me.Left = 0
'### Siempre en el tope
    SetWindowPos Me.hWnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE + SWP_NOSIZE
'### Ventana principal no visible
    frmMain.Visible = False
'### mismos tamaños como desktop de los servidores
    Me.Width = Me.ScaleX(dx, vbPixels, vbTwips)
    Me.Height = Me.ScaleY(dy, vbPixels, vbTwips)
'### el picturebox es más pequeño
    pVideo.Width = Me.Width - 100
    pVideo.Height = Me.Height - 200
    Call CapConnect
End Sub

Private Sub pVideo_KeyDown(KeyCode As Integer, Shift As Integer)
'### Envía el código de la tecla precionada
'### y espera para responder
Dim bKeys As Byte
```

```
bKeys = CByte(KeyCode)
TCP_KB.SendData bKeys
Do Until KbReply
    DoEvents
Loop
KbReply = False
End Sub
```

```
Private Sub pVideo_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
'### envía la posición y el botón para un click del mouse
' ###se puede agregar funcionalidad para decir al servidor si el cambio-importante
' ###es precionado
    Select Case Button
        Case 1
            TCP_Mouse.SendData X & ";" & Y & ";" & 1
        Case 2
            TCP_Mouse.SendData X & ";" & Y & ";" & 3
    End Select
Do Until MouseReply
    DoEvents
Loop
End Sub
```

```
Private Sub TCP_DataArrival(ByVal BytesTotal As Long)
Dim Ret As Long
Dim ByteArray() As Byte
'### Bueno, Obtengo una serie de bytes comprimidos
    TCP.GetData ByteArray, vbByte
'### Descomprimo eso
    Call ZLib.DecompressByte(ByteArray)
    DoEvents
'### Realizo un DIB desde la serie de bytes
    Call RecDib.ParseByte(ByteArray)
'### y finalmente ponemos el blit a la posición real
    Ret = BitBlt(pVideo.hdc, pos(0), pos(1), dx / CRate, dy / CRate, RecDib.hdc, 0, 0, SRCCOPY)
'### envío algunas réplicas - no importa lo que usted envíe
'### Yo pienso que sería mejor enviar un byte
    TCP.SendData "a"
End Sub
```

```
Private Sub TCP_KB_DataArrival(ByVal BytesTotal As Long)
'### El servidor envía una réplica al comando del teclado
    KbReply = True
End Sub
```

```
Private Sub TCP_Mouse_DataArrival(ByVal BytesTotal As Long)
'### El servidor envía una réplica al comando del mouse
```



```
    MouseReply = True
End Sub

Private Sub TCP_Set_DataArrival(ByVal BytesTotal As Long)
'### el servidor nos dice la posición real para el blitting
Dim tmp As String
    TCP_Set.GetData tmp, vbString
    tmpPos = Split(tmp, ";")
    pos(0) = CLng(tmpPos(0))
    pos(1) = CLng(tmpPos(1))
    TCP_Set.SendData "a"
End Sub

Public Sub CapConnect()
'### conecta todos los datos - a los sockets
Dim strHost As String
    strHost = modClient.RemoteIP
    TCP.Connect strHost, 2213
    TCP_Set.Connect strHost, 2214
    TCP_Mouse.Connect strHost, 2215
    TCP_KB.Connect strHost, 2216
'### crea u dispositivo independiente de bitmap con 16 colores
    RecDib.Colors = 16
    Call RecDib.Create(dx / CRate, dy / CRate)
    i = 0
End Sub
```