

UNIVERSIDAD DON BOSCO



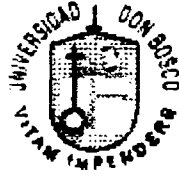
**“ ELABORACION DE UNA HERRAMIENTA ASISTIDA POR
COMPUTADORA PARA EL DISEÑO ELECTRICO Y EL
CALCULO DE TENSIONES MECANICAS DE
LINEAS DE TRANSMISION DE ALTO VOLTAJE ”
ANEXOS**

**TRABAJO DE GRADUACION
PREPARADO PARA LA FACULTAD DE INGENIERIA**

**PARA OPTAR EL GRADO DE:
INGENIERO ELECTRICISTA**

**PRESENTADO POR:
OTTO NAPOLEON TEVEZ ESCOBAR
JOSE MIGUEL VALENCIA ARTIGA**

**26 DE SEPTIEMBRE DE 1998
SOYAPANGO, EL SALVADOR, CENTRO AMERICA.**



UNIVERSIDAD DON BOSCO

**RECTOR
ING. FEDERICO HUGUET RIVERA**

**SECRETARIO GENERAL
PBRO. PEDRO JOSE GARCIA CASTRO**

**DECANO DE LA FACULTAD DE INGENIERIA
ING. CARLOS G. BRAN**

**ASESOR DEL TRABAJO DE GRADUACION
ING. PEDRO ALFONSO CHANCHAN J.**

**JURADO EXAMINADOR
ING. JOSE EDUARDO CONTRERAS
ING. EDGAR OMAR SOLORZANO LINARES**

INDICE

FORMAS PRINCIPALES

MENUS

Forma de presentación (frmSplash.frm)	1
Menú de inicio (Intro.frm)	2
Menú principal (Principal.frm)	2
Menú de parámetros (Param.frm)	3
Menú de la base de datos (frmbasedatos.frm)	5

DISEÑO

Entrada de datos (frmdatos.frm)	6
Método del momento eléctrico (frmcalcable.frm)	11
Momento Eléctrico (frmaisla.frm)	52
Angulo de desviación de la cadena de aisladores (frmangulo.frm)	77
Cálculo de la altura del cable de guarda y ángulo crítico de blindaje (frmblindaje.frm)	88
Figura final (frmvistazo.frm)	93
Efecto corona (frmcorona.frm)	100
Ruido audible y radio interferencia (frmruído.frm)	121

PLANTILLA DE FLECHAS

Cálculo de tensiones y flechas:	
plantilla de flechas curvas (frmtensiones.frm)	133
Vanos a desnivel: calculo de tensiones y flechas:	
plantilla de flechas curvas (frmvanosdes.frm)	166
Cálculo del numero de antivibradores y distancias (frmnumamorti.frm)	178

INTERACCIÓN DE LAS LÍNEAS DE TRANSMISIÓN CON LOS SISTEMAS ELÉCTRICOS DE POTENCIA: PARÁMETROS ELÉCTRICOS

Tipos de cables (frm cables)	187
Tabla de tipos de cables (frmgridcables.frm)	190
Resistencia (resi.frm)	191
Inductancia (Induct.frm)	200
Capacitancia (Capac.frm)	218
Constantes A, B, C y D (Ctes.frm)	254

FORMAS SECUNDARIAS RELACIONADAS CON LAS FORMAS PRINCIPALES (BASE DE DATOS)

FORMAS PARA LA BASES DE DATOS

Aisladores

(frmdataaisladores.frm; frmmasaisladores; frmmasaisladores2; frmmasaisladores3 y frmmasaisladores4)	264
---	-----

Herrajes

(frmherrajes.frm; frm masherrajes1.frm; frm masherrajes2.frm; frm masherrajes3.frm)	270
Estructuras (frmestruct.frm)	292

FORMAS DECLARADAS EN MODO MODAL

Datos para un circuito (dat1ckto.frm)	310
Datos para dos circuitos (dat2ckto.frm)	314
Nivel de contaminación (frmactualizar.frm)	318
Correcciones por altura (presión), temperatura y humedad (frmcorrecciones.frm)	320
Correcciones por altura y temperatura:	
Efecto corona (frmcorrecorona.frm)	324
Información sobre parámetros (frmdatosabcd.frm)	326

Actualización de números de antivibradores por vano (frminfnumamorti.frm)	326
Iteraciones requeridas (frmitera.frm)	327
Información de la longitud del cable y vano crítico (frmlonvanocri.frm)	328
Ajuste de la velocidad del viento (frmpacalpre.frm)	333
Ajuste de la presión del viento (frmprevdes.frm)	339
Visualización de distancias dos circuitos (frmverdis.frm)	343
Visualización de vanos a desnivel (frmvervanodes.frm)	344
<u>FORMA PARA BARRA DE MENU</u>	
Forma MDI (MDIfrm.frm)	353
Tutorial (Tutorial.frm)	355
Menú del tutorial (Tutor.frm)	357
 DECLARACION DE VARIABLES	
Codvariables (variables.bas)	358
Codediseño (Codediseño.bas)	359

FORMA DE PRESENTACION

(frmSplash.frm)

Option Explicit

```
Private Sub Form_KeyPress(KeyAscii As Integer)
Unload Me
End Sub
```

```
Private Sub Form_Load()
lblVersion.Caption = "Version " & 5 & "." & 0
Nombreactivo.Caption = "DELTA"
End Sub
```

```
Private Sub Frame1_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub imgLogo_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub lblCompany_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub lblCompanyProduct_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub lblLicenseTo_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub lblPlatform_Click()
Unload Me
Intro.Show
End Sub
```

```
Private Sub lblWarning_Click()
```

```
Unload Me  
Intro.Show  
End Sub
```

```
Private Sub Nombreproducto_Click()  
Unload Me  
Intro.Show  
End Sub
```

```
Private Sub Timer1_Timer()  
Unload Me  
Intro.Show  
End Sub
```

MENU DE INICIO

(Intro.frm)

```
Private Sub Delta_Click(Index As Integer)  
Unload Me  
Principio.Show  
End Sub
```

```
Private Sub SALIR_Click(Index As Integer)  
out = MsgBox("Desea abandonar la aplicación DELTA ?", 4, "Salir de DELTA")  
If out = vbYes Then  
End  
Else  
End If  
End Sub
```

```
Private Sub tutorial_Click()  
Unload Me  
Tutor.Show  
End Sub
```

MENU PRINCIPAL

(Principal.frm)

```
Private Sub btnfrmVcable_Click()  
Unload Me  
frmdatos.Show  
End Sub
```

```
Private Sub btnvfrmension_Click()  
Unload Me  
frmensiones.Show  
End Sub
```

```
Private Sub Delta1_Click()  
Unload Me  
Param.Show  
End Sub
```

```
Private Sub Delta4_Click()  
Unload Me  
Intro.Show  
End Sub
```

```
Private Sub Delta3_Click()  
Unload Me  
frmbasedatos.Show  
End Sub
```

```
Private Sub Delta5_Click()  
Unload Me  
Intro.Show  
End Sub
```

```
Private Sub Form_Load()  
MDIfrm.conductor.Enabled = False  
MDIfrm.Parámetros.Enabled = False  
MDIfrm.DISEÑO.Enabled = False  
MDIfrm.tensiones.Enabled = False  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
MDIfrm.conductor.Enabled = True  
MDIfrm.Parámetros.Enabled = True  
MDIfrm.DISEÑO.Enabled = True  
MDIfrm.tensiones.Enabled = True  
End Sub
```

MENU DE PARAMETROS

(Param.frm)

```
Private Sub Cap_Click()  
Unload Me  
Capac.Show
```



```
End Sub
```

```
Private Sub ctes_Click()  
Unload Me  
ctes.Show  
End Sub
```

```
Private Sub Ind_Click()  
Unload Me  
Induct.Show  
End Sub
```

```
Private Sub btniravolycab_Click()  
Unload Me  
frmcalcable.Show  
End Sub
```

```
Private Sub Form_Load()  
MDIfrm.conductor.Enabled = False  
MDIfrm.Parámetros.Enabled = False  
MDIfrm.DISEÑO.Enabled = False  
MDIfrm.tensiones.Enabled = False  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
MDIfrm.conductor.Enabled = True  
MDIfrm.Parámetros.Enabled = True  
MDIfrm.DISEÑO.Enabled = True  
MDIfrm.tensiones.Enabled = True  
End Sub
```

```
Private Sub iracapac_Click()  
Unload Me  
Capac.Show  
End Sub
```

```
Private Sub iraconductor_Click()  
Unload Me  
frmCables.Show  
End Sub
```

```
Private Sub iractes_Click()  
Unload Me  
ctes.Show  
End Sub
```

```
Private Sub Res_Click(Index As Integer)
```

```
Unload Me  
resi.Show  
End Sub
```

```
Private Sub iraind_Click()  
Unload Me  
Induct.Show  
End Sub
```

```
Private Sub irares_Click(Index As Integer)  
Unload Me  
resi.Show  
End Sub
```

```
Private Sub irainduct_Click()  
Unload Me  
Induct.Show  
End Sub
```

```
Private Sub iraresi_Click(Index As Integer)  
Unload Me  
resi.Show  
End Sub
```

```
Private Sub Salirparam_Click()  
Unload Me  
Principio.Show  
End Sub
```

MENU DE BASE DE DATOS

(frmbasedatos.frm)

```
Private Sub btnverdataais_Click()  
Dim RetVal As Double  
RetVal = Shell("C:\Archivos de programa\Microsoft  
Office\Office\MSACCESS.EXE C:\Delta\bases\DataAisladores.mdb", 1)  
End Sub
```

```
Private Sub btnverdatacab_Click()  
Dim RetVal As Double  
RetVal = Shell("C:\Archivos de programa\Microsoft  
Office\Office\MSACCESS.EXE C:\Delta\bases\DataCables.mdb", 1)  
End Sub
```

```
Private Sub btnverdataestru_Click()
```

```
Dim RetVal As Double
RetVal = Shell("C:\Archivos de programa\Microsoft
Office\Office\MSACCESS.EXE C:\Delta\bases\Dataestruc.mdb", 1)
End Sub
```

```
Private Sub btnverdataherr_Click()
Dim RetVal As Double
RetVal = Shell("C:\Archivos de programa\Microsoft
Office\Office\MSACCESS.EXE C:\Delta\bases\DataAccesorios.mdb", 1)
End Sub
```

```
Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub
```

```
Private Sub Form_Load()
MDIfrm.conductor.Enabled = False
MDIfrm.Parámetros.Enabled = False
MDIfrm.DISEÑO.Enabled = False
MDIfrm.tensiones.Enabled = False
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
MDIfrm.conductor.Enabled = True
MDIfrm.Parámetros.Enabled = True
MDIfrm.DISEÑO.Enabled = True
MDIfrm.tensiones.Enabled = True
End Sub
```

DATOS

(frmdatos.frm)

```
Private Sub btnactual_Click()
frmactualizar.Show vbModal
End Sub
```

```
Private Sub btnrestaurar_Click()
textaltura.Text = 0
texttempe.Text = 25
texthumedad.Text = 11
End Sub
```

```
Private Sub btnverfrmvoltaje_Click()
Hide
```

```

frmcalcable.Show
End Sub

Private Sub btnverprincipio_Click()
Hide
Principio.Show
End Sub

Private Sub cbonivel_Click()
contaminacion = cbonivel.Text

Select Case contaminacion
Case "Ligera"
'kfe = 16
kfe = frmactualizar.textligera.Text
GoTo Line1
Case "Media"
'kfe = 20
kfe = frmactualizar.textmedia.Text
GoTo Line1
Case "Alta"
'kfe = 25
kfe = frmactualizar.textalta.Text
GoTo Line1
Case "Muy Alta"
'kfe = 31
kfe = frmactualizar.textmalta.Text
GoTo Line1
Case Else
Item = MsgBox("No se ha definido el nivel de contaminación: " &
contaminacion, 48, "Nivel de contaminación sin definir")
GoTo Line2
End Select

Line1:
ver = "Nivel de cotaminación: " & contaminacion & " ; Kfe = " & kfe & " mm/KV "
cbonivel.ToolTipText = ver
Line2:
cbonivel.SetFocus
End Sub

Private Sub Label1_Click()
msjuni = "Para el diseño de la línea de transmisión se puede empezar" &
Chr(10) & Chr(13)
msjuni = msjuni & "por ver la base de datos de que se dispone, para tener" &
Chr(10) & Chr(13)
msjuni = msjuni & "una idea clara de lo que se quiere realizar en dicho diseño"

```

```
INF = MsgBox(msjuni, vbInformation, "INFORMACION ADICIONAL (DELTA)")
End Sub
```

```
Private Sub ncf_Click()
ncfa = Int(Val(ncf.Text))
ncf.Text = ncfa
Select Case ncfa
```

```
Case Is = 0
    def = MsgBox("No existen Subconductores, por lo menos debe de existir un
conductor ", 48, "Valor negativo ")
Line1:
    ncf.Text = 1
```

```
Case Is < 1
    def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
    If def = vbYes Then
        ncf.Text = Abs(Int(ncf.Text))
    Else
        GoTo Line1
    End If
End Select
End Sub
```

```
Private Sub ncf_Change()
ncfa = Int(Val(ncf.Text))
ncf.Text = ncfa
Select Case ncfa
Case Is >= 6
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
    GoTo Line2
```

```
Case Is = 0
    def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
    ncf.Text = 1
```

```
Case Is < 1
    def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
    If def = vbYes Then
        ncf.Text = Abs(Int(ncf.Text))
```

Line2:

```

    Else
    GoTo Line1
    End If
End Select
End Sub

```

```

Private Sub spinum_SpinDown()
Dim dblrate
dblrate = ncf.Text
If dblrate > 1 Then
    ncf.Text = dblrate - 1
    Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub

```

```

Private Sub spinum_SpinUp()
ncf.Text = (ncf.Text) + 1
If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh
End Sub

```

```

Private Sub textaltura_Change()
If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And
Val(texthumedad.Text) = 11 Then
    btnrestaurar.Enabled = False
    GoTo final
    Else
    btnrestaurar.Enabled = True
End If
final:
End Sub

```

```

Private Sub textfp_Change()
fp = Val(textfp.Text)
If Abs(fp) > 1 Then
    fpnd = "El F.P. no puede ser mayor que uno" & Chr(10) & Chr(13)
    fpnd = fpnd & "Entre el dato correcto (-1<= F.P. <= 1)"
    deffp = MsgBox(fpnd, vbCritical, "Factor de Potencia fuera de rango
(DELTA)")
    textfp.Text = 0.85

```

```
    textfp.SelStart = 0
    textfp.SelLength = Len(textfp.Text)
End If
End Sub
```

```
Private Sub texthumedad_Change()
    If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And
Val(texthumedad.Text) = 11 Then
        btnrestaurar.Enabled = False
        GoTo final
    Else
        btnrestaurar.Enabled = True
    End If
final:
End Sub
```

```
Private Sub texttempe_Change()
    If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And texthumedad.Text =
11 Then
        btnrestaurar.Enabled = False
        GoTo final
    Else
        btnrestaurar.Enabled = True
    End If
final:
End Sub
```

MOMENTO ELECTRICO

(frmcalcable.frm)

Public cktos As Double

Private Sub btncalmme_Click()

Rem ***** CALCULO DE XL *****

Static pi

pi = 3.14159265358979

ncfa = Val(ncf.Text)

If uckto = True Then

***** inicio para un ckto *****

fre = 60

rmg = (Val(textrmg.Text)) / 1000

dab = 3.3528

dbc = 5.7368

dca = 5.7368

dmg = (dab * dbc * dca) ^ (1 / 3)

If ncf >= 2 Then

dis = 0.4

requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente

rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si

existe más de un conductor por fase

L = 0.2 * Log(dmg / rmgg)

XL = 2 * pi * fre * L * 10 ^ -3

textinduct.Text = XL

Else

requi = 1 'Radio equivalente

rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si

existe más de un conductor por fase

L = 0.2 * Log(dmg / rmgg)

XL = 2 * pi * fre * L * 10 ^ -3

textinduct.Text = XL

End If

Else ***** inicio de dos ckto *****

fre = 60

rmg = (Val(textrmg.Text)) / 1000

dac2 = 6.096

dbb2 = 7.9248

dca2 = 6.096

dhab = 3.3528


```

dhbc = 3.3528

dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

12) dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
rmgg = rmg

rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
If ncfa >= 2 Then
  dis = 0.4
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
Else
  requi = 1 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
End If
End If

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = Val(textresi.Text)
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

```

```

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
End Sub

```

```

Private Sub btnverdatos_Click()
Hide
frmdatos.Show
End Sub

```

```

Private Sub btnverfrmaisal_Click()
Rem Condiciones de comparación entre las pérdidas de P y V
caidapi = Val(textcaidap(0).Text)
caidavi = Val(textcaidav.Text)
caidap = Val(textcaidapn.Text)
caidav = Val(textcaidavn.Text)
Rem No se establecen las condiciones especificadas

```

```

If caidap > caidapi Or caidav > caidavi Then
Select Case caidap
Case Is > caidapi
simbol1 = ">"
Case Is < caidapi
simbol1 = "<"
Case Is = caidapi
simbol1 = "="
End Select

```

```

Select Case caidav
Case Is > caidavi
simbol2 = ">"
Case Is < caidavi
simbol2 = "<"
Case Is = caidavi
simbol2 = "="
End Select

```

```

    msjnc = "La selección del cable o el voltaje no satisfacen las condiciones
especificadas. " & Chr$(10) & Chr$(13)
    msjnc = msjnc & "Seleccione otro tipo de cable o cambie el nivel de voltaje. "
& Chr$(10) & Chr$(13)
    msjnc = msjnc & Chr$(10) & Chr$(13)
    msjnc = msjnc & "%P: Condición final " & caidap & simbol1 & caidapi & " %
condición de diseño" & Chr$(10) & Chr$(13)
    msjnc = msjnc & "%V: Condición final " & caidav & simbol2 & caidavi & " %
condición de diseño" & Chr$(10) & Chr$(13)

```

```

    MsgBox msjnc
    GoTo fin
End If

```

```

Rem Se cumplen las condiciones especificadas
If caidap <= caidapi And caidav <= caidavi Then
    msjnc = "La seleccion del cable y el voltaje satisfacen las condiciones
especificadas. " & Chr$(10) & Chr$(13)
    msjnc = msjnc & "Se debe de seleccionar el tipo de sobretensión con el que
se va a diseñar la línea: BIL o NBS. " & Chr$(10) & Chr$(13)
    msjnc = msjnc & Chr$(10) & Chr$(13)
    msjnc = msjnc & "%P: Condición final " & caidap & " <= " & caidapi & " %
condición de diseño" & Chr$(10) & Chr$(13)
    msjnc = msjnc & "%V: Condición final " & caidav & " <= " & caidavi & " %
condición de diseño" & Chr$(10) & Chr$(13)

```

```

    MsgBox msjnc
    btnverfrmaisla.SetFocus
End If
Hide
frmaisal.Show
fin:
End Sub

```

```

Private Sub btnverinduct_Click()
rastreo4 = "activo L frmcalcable"
Hide
Induct.Show
End Sub

```

```

Private Sub btnverprincipio_Click()
Hide
Principio.Show
End Sub

```

```

Private Sub btnverresi_Click()
rastreo3 = "activo R frmcalcable"

```

```

Hide
resi.Show
End Sub

Private Sub cbovoltaje_Click()
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)
If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If

Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

```

```

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

```

```

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

```

```

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

```

```

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

```

ter:

```
End Sub
```

```
Private Sub cbovoltaje_Change()
```

```
Rem Cálculo de las corrientes
```

```
v = Val(cbovoltaje.Text) * 10 ^ 3
```

```
p = Val((textpoten.Text) * 10 ^ 6)
```

```
fs = Val(textfs.Text)
```

```
fp = Val(textfp.Text)
```

```
If v = 0 Then
```

```
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
```

```
    cbovoltaje.SetFocus
```

```
    GoTo ter
```

```
End If
```

```
If fp = 0 Then
```

```
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
```

```
    textfp.SetFocus
```

```
    GoTo ter
```

```
End If
```

```
ncf = Val(ncf.Text)
```

```
If uckto.Value = True Then
```

```
    uc = 1
```

```
Else
```

```
    uc = 2
```

```
End If
```

```

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If

Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = (Val(textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = Val(textresi2.Text) / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)
p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
ter:
End Sub

Private Sub Command1_Click()
rastreo5 = "activo Cab frmcalcable"
Hide
frmCables.Show
End Sub

Private Sub dbci50_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbci50.SelectedItem

```

End Sub

```
Private Sub dckto_Click()
If dckto.Value = True Then
    cktos = 2
    lblckto.Enabled = True
    lbluckto.Enabled = False
Else
    cktos = 1
    lbluckto.Enabled = True
    lblckto.Enabled = False
End If
```

```
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)
If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If
```

```
If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
    textfp.SetFocus
    GoTo ter
End If
```

```
ncf = Val(ncf.Text)
If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If
```

```
If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
```

```
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
```

```

Rem *** Para P futura ***
pfu = (Val(textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem *****

Rem ***** CALCULO DE XL *****
Static pi
pi = 3.14159265358979
ncfa = Val(ncf.Text)

If uckto = True Then
'***** inicio para un ckto *****
fre = 60
rmg = (Val(textrmg.Text)) / 1000

dab = 3.3528
dbc = 5.7368
dca = 5.7368
dmg = (dab * dbc * dca) ^ (1 / 3)
If ncf >= 2 Then
dis = 0.4
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textinduct.Text = XL
Else
requi = 1 'Radio equivalente
rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textinduct.Text = XL
End If
Else '***** inicio de dos ckto *****
fre = 60
rmg = (Val(textrmg.Text)) / 1000

If rmg = 0 Then
GoTo ter
End If

dac2 = 6.096
dbb2 = 7.9248

```



```

dca2 = 6.096

dhab = 3.3528
dhbc = 3.3528

dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

12) dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
rmgg = rmg

rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
If ncfa >= 2 Then
  dis = 0.4
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
Else
  requi = 1 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
End If
End If

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

```

```

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

```

ter:

```
End Sub
```

```

Private Sub Form_Activate()
textpoten.Text = frmdatos.textpoten.Text
textfp.Text = frmdatos.textfp.Text
textcaidap(0).Text = frmdatos.textcaidap(0).Text
textcaidav.Text = frmdatos.textcaidav.Text
textfs.Text = frmdatos.textfs.Text
textkm.Text = frmdatos.textkm.Text
cbovoltaje.Text = frmdatos.cbovoltaje.Text
ncf.Text = frmdatos.ncf.Text

If frmdatos.uckto.Value = True Then
    uckto.Value = True
Else
    dckto.Value = True
End If

dbci50.Text = Val(dtacables.Recordset("AMP (50 °C)"))
Rem Cálculo de P futura
años = Val(textaños.Text)
tasa = Val(texttasa.Text)
p = Val(textpoten.Text)
pfu = Val(p * (1 + (tasa / 100)) ^ (años))
textpotenfu.Text = pfu

textresi2.Text = textresi.Text
End Sub

Private Sub Form_Load()

```

```
cktos = 1
```

```
textpoten.Text = frmdatos.textpoten.Text
textfp.Text = frmdatos.textfp.Text
textcaidap(0).Text = frmdatos.textcaidap(0).Text
textcaidav.Text = frmdatos.textcaidav.Text
textfs.Text = frmdatos.textfs.Text
textkm.Text = frmdatos.textkm.Text
cbovoltaje.Text = frmdatos.cbovoltaje.Text
ncf.Text = frmdatos.ncf.Text
```

```
If frmdatos.uckto.Value = True Then
    uckto.Value = True
Else
    dckto.Value = True
End If
End Sub
```

```
Private Sub ncf_Click()
    ncfa = Int(Val(ncf.Text))
    ncf.Text = ncfa
    Select Case ncfa
        Case Is = 0
            def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
            ncf.Text = 1
            Case Is < 1
                def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
                If def = vbYes Then
                    ncf.Text = Abs(Int(ncf.Text))
                Else
                    GoTo Line1
                End If
            End Select
```

```
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)
If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If
```

```

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

!*****
Rem ***** CALCULO DE XL *****
Static pi
pi = 3.14159265358979
ncfa = Val(ncf.Text)

If uckto = True Then
    !***** inicio para un ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

    dab = 3.3528
    dbc = 5.7368
    dca = 5.7368
    dmg = (dab * dbc * dca) ^ (1 / 3)
    If ncf >= 2 Then
        dis = 0.4
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    
```

```

    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    L = 0.2 * Log(dmg / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
Else
    requi = 1 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    L = 0.2 * Log(dmg / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
End If
Else '***** inicio de dos ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

    dac2 = 6.096
    dbb2 = 7.9248
    dca2 = 6.096

    dhab = 3.3528
    dhbc = 3.3528

    dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
    dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
    dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

    dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
    dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
    daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

    dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)
    rmgg = rmg

    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    If ncfa >= 2 Then
        dis = 0.4
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
        'aumento por el caso de 2ckts:
        rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
        L = 0.2 * Log(dmg / rmgdckto)
        XL = 2 * pi * fre * L * 10 ^ -3
        textinduct.Text = XL

```

```

Else
requi = 1 'Radio equivalente
rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
'aumento por el caso de 2ckts:
rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
L = 0.2 * Log(dmg / rmgdckto)
XL = 2 * pi * fre * L * 10 ^ -3
textinduct.Text = XL
End If
End If

```

```

Rem ***** METODO DEL MOMENTO ELECTRICO *****

```

```

ncfa = Val(ncf.Text)

```

```

km = Val(textkm.Text)

```

```

textresi2.Text = textresi.Text

```

```

producto = cktos * ncfa

```

```

textresi2.Text = textresi2.Text / producto

```

```

resiac = Val(textresi2.Text)

```

```

XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)

```

```

v = Val(cbovoltaje.Text) * 10 ^ 3

```

```

fp = Val(textfp.Text)

```

```

ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))

```

```

tanfp = Tan(ang2)

```

```

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P

```

```

textcaidapn.Text = Format(caidap, "0.000000")

```

```

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2

```

```

textcaidavn.Text = Format(caidav, "0.000000")

```

```

ter:

```

```

End Sub

```

```

Private Sub ncf_Change()

```

```

ncfa = Int(Val(ncf.Text))

```

```

ncf.Text = ncfa

```

```

Select Case ncfa

```

```

Case Is >= 6

```

```

mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")

```

```

GoTo Line2

```

```

    Case Is = 0
    def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")

```

```

Line1:

```

```

    ncf.Text = 1
    Case Is < 1
    def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")

```

```

    If def = vbYes Then
    ncf.Text = Abs(Int(ncf.Text))

```

```

Line2:

```

```

    Else
    GoTo Line1
    End If
End Select

```

```

Rem Cálculo de las corrientes

```

```

v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

```

```

If v = 0 Then

```

```

    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter

```

```

End If

```

```

If fp = 0 Then

```

```

    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")

```

```

    textfp.SetFocus
    GoTo ter

```

```

End If

```

```

ncf = Val(ncf.Text)

```

```

If uckto.Value = True Then

```

```

    uc = 1
    Else
    uc = 2

```

```

End If

```

```

If dckto.Value = True Then

```

```

    uc = 2
    Else
    uc = 1

```

```

End If

```

```

Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = (Val(textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

!*****
Rem ***** CALCULO DE XL *****
Static pi
pi = 3.14159265358979
ncfa = Val(ncf.Text)

If uckto = True Then
    !***** inicio para un ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

    If rmg = 0 Then
        GoTo ter
    End If

    dab = 3.3528
    dbc = 5.7368
    dca = 5.7368
    dmg = (dab * dbc * dca) ^ (1 / 3)
    If ncf >= 2 Then
        dis = 0.4
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
        L = 0.2 * Log(dmg / rmgg)
        XL = 2 * pi * fre * L * 10 ^ -3
        textinduct.Text = XL
    Else
        requi = 1 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
        L = 0.2 * Log(dmg / rmgg)
        XL = 2 * pi * fre * L * 10 ^ -3
        textinduct.Text = XL
    End If
Else !***** inicio de dos ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

```



```

dac2 = 6.096
dbb2 = 7.9248
dca2 = 6.096

dhab = 3.3528
dhbc = 3.3528

dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

12) dmrg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
rmgg = rmg

rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
If ncfa >= 2 Then
  dis = 0.4
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmrg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
Else
  requi = 1 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmrg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
End If
End If

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text

```

```

producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

```

```

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

```

```

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

```

```

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

```

ter:

```
End Sub
```

```
Private Sub spinum_SpinDown()
```

```
Dim dblrate
```

```
dblrate = ncf.Text
```

```
If dblrate > 1 Then
```

```
    ncf.Text = dblrate - 1
```

```
Else
```

```
    ncf.Text = 1
```

```
End If
```

```
ncf.Refresh
```

```
End Sub
```

```
Private Sub spinum_SpinUp()
```

```
ncf.Text = (ncf.Text) + 1
```

```
If dblrate >= 6 Then
```

```
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar  
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de  
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
```

```
End If
```

```
ncf.Refresh
```

```
End Sub
```

```
Private Sub textaños_Click()
```

```
años = Val(textaños.Text)
```

```
tasa = Val(texttasa.Text)
```

```
p = Val(textpoten.Text)
```

```
pfu = p * (1 + (tasa / 100)) ^ (años)
```

```
textpotenfu.Text = pfu
End Sub
```

```
Private Sub textaños_Change()
años = Val(textaños.Text)
tasa = Val(texttasa.Text)
p = Val(textpoten.Text)
pfu = p * (1 + (tasa / 100)) ^ (años)
```

```
textpotenfu.Text = pfu
End Sub
```

```
Private Sub textfp_Click()
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)
If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If
```

```
If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)
```

```
If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If
```

```
If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
```

```

Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
ter:
End Sub

Private Sub textfp_Change()
fp = Val(textfp.Text)
If Abs(fp) >= 1 Then
    fpnd = "El F.P. no puede ser mayor que uno" & Chr(10) & Chr(13)
    fpnd = fpnd & "Entre el dato correcto (-1<= F.P. <= 1)"
    deffp = MsgBox(fpnd, vbCritical, "Factor de Potencia fuera de rango
(DELTA)")
    textfp.Text = 0.85
    textfp.SelStart = 0
    textfp.SelLength = Len(textfp.Text)
End If

Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

```

```

If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
    textfp.Text = 0.85
    textfp.SelStart = 0
    textfp.SelLength = Len(textfp.Text)
    'textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = (Val(textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = Val(textresi2.Text) / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
ter:
End Sub

Private Sub textfs_Click()
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If

```

```

Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu
ter:
End Sub

Private Sub textfs_Change()
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)

```

```

    ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
    textifu.Text = ifu
ter:
End Sub

Private Sub textinduct_Click()
Rem ***** METODO DEL MOMENTO ELECTRICO *****

    ncfa = Val(ncf.Text)
    km = Val(textkm.Text)

    textresi2.Text = textresi.Text
    producto = cktos * ncfa
    textresi2.Text = textresi2.Text / producto

    resiac = Val(textresi2.Text)
    XL = Val(textinduct.Text)

    p = Val((textpoten.Text) * 10 ^ 6)
    v = Val(cbovoltaje.Text) * 10 ^ 3
    fp = Val(textfp.Text)
    ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
    tanfp = Tan(ang2)

    caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
    textcaidapn.Text = Format(caidap, "0.000000")

    caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
    textcaidavn.Text = Format(caidav, "0.000000")

End Sub

Private Sub textinduct_Change()
Rem ***** METODO DEL MOMENTO ELECTRICO *****
    ncfa = Val(ncf.Text)
    km = Val(textkm.Text)

    textresi2.Text = Val(textresi.Text)
    producto = cktos * ncfa
    textresi2.Text = textresi2.Text / producto

    resiac = Val(textresi2.Text)
    XL = Val(textinduct.Text)

    p = Val((textpoten.Text) * 10 ^ 6)
    v = Val(cbovoltaje.Text) * 10 ^ 3
    fp = Val(textfp.Text)

```



```

ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

```

End Sub

```

Private Sub textkm_Click()
Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

```

End Sub

```

Private Sub textkm_Change()
Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = Val(textresi2.Text) / producto

resiac = Val(textresi2.Text)

```

```
XL = Val(textinduct.Text)
```

```
p = Val((textpoten.Text) * 10 ^ 6)
```

```
v = Val(cbovoltaje.Text) * 10 ^ 3
```

```
fp = Val(textfp.Text)
```

```
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
```

```
tanfp = Tan(ang2)
```

```
caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")
```

```
caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
```

```
End Sub
```

```
Private Sub textpoten_Click()
```

```
años = Val(textaños.Text)
```

```
tasa = Val(texttasa.Text)
```

```
p = Val(textpoten.Text)
```

```
pfu = p * (1 + (tasa / 100)) ^ (años)
```

```
textpotenfu.Text = pfu
```

```
Rem Cálculo de las corrientes
```

```
v = Val(cbovoltaje.Text) * 10 ^ 3
```

```
p = Val((textpoten.Text) * 10 ^ 6)
```

```
fs = Val(textfs.Text)
```

```
fp = Val(textfp.Text)
```

```
If v = 0 Then
```

```
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
```

```
    cbovoltaje.SetFocus
```

```
    GoTo ter
```

```
End If
```

```
If fp = 0 Then
```

```
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
```

```
    textfp.SetFocus
```

```
    GoTo ter
```

```
End If
```

```
If fp = 0 Then
```

```
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
```

```
    textfp.SetFocus
```

```
    GoTo ter
```

```

End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
ter:
End Sub

```

```

Private Sub textpoten_Change()
años = Val(textaños.Text)
tasa = Val(texttasa.Text)
p = Val(textpoten.Text)
pfu = p * (1 + (tasa / 100)) ^ (años)

textpotenfu.Text = pfu
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1
Else
    uc = 2
End If

If dckto.Value = True Then
    uc = 2
Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

Rem ***** METODO DEL MOMENTO ELECTRICO *****

```

```

ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = Val(textresi2.Text) / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
ter:
End Sub

Private Sub textpotenfu_Click()
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)

If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If

If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos
(DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)

If uckto.Value = True Then
    uc = 1

```

```

Else
uc = 2
End If

```

```

If dckto.Value = True Then

```

```

    uc = 2

```

```

    Else

```

```

        uc = 1

```

```

End If

```

```

Rem *** Para P actual ***

```

```

i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)

```

```

texti.Text = i

```

```

Rem *** Para P futura ***

```

```

pfu = Val((textpotenfu.Text) * 10 ^ 6)

```

```

ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)

```

```

textifu.Text = ifu

```

```

ter:

```

```

End Sub

```

```

Private Sub textpotenfu_Change()

```

```

Rem Cálculo de las corrientes

```

```

v = Val(cbovoltaje.Text) * 10 ^ 3

```

```

p = Val(textpoten.Text) * 10 ^ 6

```

```

fs = Val(textfs.Text)

```

```

fp = Val(textfp.Text)

```

```

If v = 0 Then

```

```

    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")

```

```

    cbovoltaje.SetFocus

```

```

    GoTo ter

```

```

End If

```

```

If fp = 0 Then

```

```

    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")

```

```

    textfp.SetFocus

```

```

    GoTo ter

```

```

End If

```

```

ncf = Val(ncf.Text)

```

```

If uckto.Value = True Then

```

```

    uc = 1

```

```

    Else

```

```

        uc = 2

```

```

End If

```

```

If dckto.Value = True Then

```

```

    uc = 2
    Else
    uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu
ter:
End Sub

Private Sub textresi2_Click()
Rem ***** METODÓ DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

End Sub

Private Sub textresi2_Change()
Rem ***** METODÓ DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

```

```

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

End Sub

Private Sub textrmg_Click()
Rem ***** CALCULO DE XL *****
  Static pi
  pi = 3.14159265358979
  ncfa = Val(ncf.Text)

  If uckto = True Then
    !***** inicio para un ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

    dab = 3.3528
    dbc = 5.7368
    dca = 5.7368
    dmrg = (dab * dbc * dca) ^ (1 / 3)
    If ncf >= 2 Then
      dis = 0.4
      requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
      rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
      L = 0.2 * Log(dmrg / rmgg)
      XL = 2 * pi * fre * L * 10 ^ -3
      textinduct.Text = XL
    Else
      requi = 1 'Radio equivalente
      rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
      L = 0.2 * Log(dmrg / rmgg)
      XL = 2 * pi * fre * L * 10 ^ -3
      textinduct.Text = XL
    End If
  Else
    !***** inicio de dos ckto *****

```



```

fre = 60
rmg = (Val(textrmg.Text)) / 1000

dac2 = 6.096
dbb2 = 7.9248
dca2 = 6.096

dhab = 3.3528
dhbc = 3.3528

dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)
rmgg = rmg

rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
If ncfa >= 2 Then
  dis = 0.4
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
Else
  requi = 1 'Radio equivalente
  rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  L = 0.2 * Log(dmg / rmgdckto)
  XL = 2 * pi * fre * L * 10 ^ -3
  textinduct.Text = XL
End If
End If

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)

```

```

km = Val(textkm.Text)

textresi2.Text = textresi.Text
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")

End Sub

Private Sub textrmg_Change()
Rem ***** CALCULO DE XL *****
    Static pi
    pi = 3.14159265358979
    ncfa = Val(ncf.Text)

If uckto = True Then
    !***** inicio para un ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000
    If rmg = 0 Then
        pre = MsgBox("El Radio Medio Geométrico no pued ser cero, defina el
RMG", vbCritical, "Falta de dato: RMG (DELTA)")
        textrmg.SetFocus
        GoTo fin
    End If
    dab = 3.3528
    dbc = 5.7368
    dca = 5.7368
    dmg = (dab * dbc * dca) ^ (1 / 3)
    If ncf >= 2 Then
        dis = 0.4
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente

```

```

    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    L = 0.2 * Log(dmg / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
    Else
    If rmg = 0 Then
        pre = MsgBox("El Radio Medio Geométrico no pued ser cero, defina el
RMG", vbCritical, "Falta de dato: RMG (DELTA)")
        textrmg.SetFocus
        GoTo fin
    End If
    requi = 1 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    L = 0.2 * Log(dmg / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
    End If
Else '***** inicio de dos ckto *****
    fre = 60
    rmg = (Val(textrmg.Text)) / 1000

    dac2 = 6.096
    dbb2 = 7.9248
    dca2 = 6.096

    dhab = 3.3528
    dhbc = 3.3528

    dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
    dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
    dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

    dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
    dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
    daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

    dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)
    rmgg = rmg

    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    If ncfa >= 2 Then
        dis = 0.4
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente

```

```

    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    L = 0.2 * Log(dmg / rmgdckto)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
    Else
    requi = 1 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    L = 0.2 * Log(dmg / rmgdckto)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
    End If
End If

Rem ***** METODO DEL MOMENTO ELECTRICO *****
ncfa = Val(ncf.Text)
km = Val(textkm.Text)

textresi2.Text = Val(textresi.Text)
producto = cktos * ncfa
textresi2.Text = textresi2.Text / producto

resiac = Val(textresi2.Text)
XL = Val(textinduct.Text)

p = Val((textpoten.Text) * 10 ^ 6)
v = Val(cbovoltaje.Text) * 10 ^ 3
fp = Val(textfp.Text)
ang2 = Val(Atn(-fp / Sqr(-fp * fp + 1)) + 2 * Atn(1))
tanfp = Tan(ang2)

caidap = 100 * resiac * km * p / (v * fp) ^ 2 'Fórmula %P
textcaidapn.Text = Format(caidap, "0.000000")

caidav = p * km * 100 * (resiac + (XL * tanfp)) / v ^ 2
textcaidavn.Text = Format(caidav, "0.000000")
fin:
End Sub

Private Sub texttasa_Click()
años = Val(textaños.Text)
tasa = Val(texttasa.Text)

```

```
p = Val(textpoten.Text)
pfu = Val(p * (1 + (tasa / 100)) ^ (años))
```

```
textpotenfu.Text = pfu
End Sub
```

```
Private Sub texttasa_Change()
años = Val(textaños.Text)
tasa = Val(texttasa.Text)
p = Val(textpoten.Text)
pfu = Val(p * (1 + (tasa / 100)) ^ (años))
```

```
textpotenfu.Text = pfu
End Sub
```

```
Private Sub uckto_Click()
If uckto.Value = True Then
    cktos = 1
    lbluckto.Enabled = True
    lblckto.Enabled = False
Else
    cktos = 2
    lblckto.Enabled = True
    lbluckto.Enabled = False
End If
```

```
Rem Cálculo de las corrientes
v = Val(cbovoltaje.Text) * 10 ^ 3
p = Val((textpoten.Text) * 10 ^ 6)
fs = Val(textfs.Text)
fp = Val(textfp.Text)
If v = 0 Then
    pel = MsgBox("Defina el voltaje", vbCritical, "Falta de datos (DELTA)")
    cbovoltaje.SetFocus
    GoTo ter
End If
```

```
If fp = 0 Then
    pel = MsgBox("Defina el factor de potencia (F.P.)", vbCritical, "Falta de datos (DELTA)")
    textfp.SetFocus
    GoTo ter
End If
ncf = Val(ncf.Text)
```

```
If uckto.Value = True Then
    uc = 1
```

```

Else
uc = 2
End If

If dckto.Value = True Then
uc = 2
Else
uc = 1
End If
Rem *** Para P actual ***
i = (p * fs / (fp * v * Sqr(3))) / (ncf * uc)
texti.Text = i
Rem *** Para P futura ***
pfu = Val((textpotenfu.Text) * 10 ^ 6)
ifu = (pfu * fs / (fp * v * Sqr(3))) / (ncf * uc)
textifu.Text = ifu

!*****
Rem ***** CALCULO DE XL *****
Static pi
pi = 3.14159265358979
ncfa = Val(ncf.Text)

If uckto = True Then
!***** inicio para un ckto *****
fre = 60
rmg = (Val(textrmg.Text)) / 1000

dab = 3.3528
dbc = 5.7368
dca = 5.7368
dmg = (dab * dbc * dca) ^ (1 / 3)
If ncf >= 2 Then
dis = 0.4
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textinduct.Text = XL
Else
requi = 1 'Radio equivalente
rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textinduct.Text = XL

```

```

End If
Else ***** inicio de dos ckto *****
  fre = 60
  rmg = (Val(textrmg.Text)) / 1000

  dac2 = 6.096
  dbb2 = 7.9248
  dca2 = 6.096

  dhab = 3.3528
  dhbc = 3.3528

  dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
  dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
  dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

  dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
  dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
  daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

  dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)
  rmgg = rmg

  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  If ncfa >= 2 Then
    dis = 0.4
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    L = 0.2 * Log(dmg / rmgdckto)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
  Else
    requi = 1 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    L = 0.2 * Log(dmg / rmgdckto)
    XL = 2 * pi * fre * L * 10 ^ -3
    textinduct.Text = XL
  End If
End If

```

Rem ***** METODO DEL MOMENTO ELECTRICO *****

$ncfa = \text{Val}(ncf.\text{Text})$

$km = \text{Val}(\text{textkm}.\text{Text})$

$\text{textresi2}.\text{Text} = \text{textresi}.\text{Text}$

$\text{producto} = \text{cktos} * ncfa$

$\text{textresi2}.\text{Text} = \text{textresi2}.\text{Text} / \text{producto}$

$\text{resiac} = \text{Val}(\text{textresi2}.\text{Text})$

$XL = \text{Val}(\text{textinduct}.\text{Text})$

$p = \text{Val}((\text{textpoten}.\text{Text}) * 10 ^ 6)$

$v = \text{Val}(\text{cbovoltaje}.\text{Text}) * 10 ^ 3$

$fp = \text{Val}(\text{textfp}.\text{Text})$

$\text{ang2} = \text{Val}(\text{Atn}(-fp / \text{Sqr}(-fp * fp + 1)) + 2 * \text{Atn}(1))$

$\text{tanfp} = \text{Tan}(\text{ang2})$

$\text{caidap} = 100 * \text{resiac} * km * p / (v * fp) ^ 2$ 'Fórmula %P

$\text{textcaidapn}.\text{Text} = \text{Format}(\text{caidap}, "0.000000")$

$\text{caidav} = p * km * 100 * (\text{resiac} + (XL * \text{tanfp})) / v ^ 2$

$\text{textcaidavn}.\text{Text} = \text{Format}(\text{caidav}, "0.000000")$

ter:

End Sub

CORRDINACION DE AISLAMIENTO

(frmaisla.frm)

Public voltajemax As Single

```
Private Sub btmcaldismin_Click()
  btnverfrmcorre.Enabled = True
  btnverangulo.Enabled = True
```

```
If optBIL.Value = True Then
  GoTo Line1 ' Diseño por Rayo
Else
  GoTo Line2 ' Diseño por maniobra
End If
```

Line1: ' ***** Inicio: diseño por Rayo *****

```
porcent = (Val(cboBILpor.Text)) / 100 ' Determina "sigma"
BIL = (Val(cboBIL.Text)) * 1000 ' Determina el BIL
espin = Val((Mid(cboespin.Text, 1, 7))) ' Determina el F. espinterométrico
fstorre = Val(cbofstorre.Text) ' Fact./Seg. torre
```

```
If fstorre = 0 Then
  pel = MsgBox("El factor de seguridad (F.S.) no ha sido definido, entre el
dato", vbCritical, "Dato faltante (DELTA)")
  cbofstorre.SetFocus
  GoTo terminar
End If
```

```
If espin = 0 Then
  pel = MsgBox("El factor espinterométrico no ha sido definido, entre el dato",
vbCritical, "Dato faltante (DELTA)")
  cboespin.SetFocus
  GoTo terminar
End If
```

```
If BIL = 0 Then
  pel = MsgBox("El BIL no ha sido definido, entre el dato", vbCritical, "Dato
faltante (DELTA)")
  cboBIL.SetFocus
  GoTo terminar
End If
```

Rem **** Correcciones por altura, temperatura y humedad: rayo ****

```

hume = Val(texthumedad.Text)
tempe = Val(texttempe.Text)
altura = Val(textaltura.Text)

```

```

If tempe = 25 And altura = 0 And hume = 11 Then
  GoTo sincronrección
Else
  GoTo corrección
End If

```

sincorrección:

```
vcf = (BIL * fstorre) / (1 - 1.3 * (porcent)) 'vcf= V(50%) fórmula
```

```
disminparis = ((vcf / 1000) / (espin * 500)) ^ (1 / 0.6) 'Distancia mínima: París
(fórmula)
```

```
disminleroy = 8 / ((3400 * espin / (vcf / 1000)) - 1)
```

```

textvcf.Text = Format(vcf / 1000, "0.00000")
textdisminparis.Text = Format(disminparis, "0.00000")
textdisminleroy.Text = Format(disminleroy, "0.00000")
GoTo Line3

```

corrección:

```

k3 = calhumedad(hume)
presión = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
k1 = presión / 76 'b/bo
k2 = (273 + 25) / (273 + tempe)

```

```
vcf = (BIL * fstorre) / (1 - 1.3 * (porcent)) 'vcf= V(50%) fórmula
```

```
vcfcorre = (vcf * k3) / (k1 * k2) 'Corrección de vcf
```

```
disminparis = ((vcfcorre / 1000) / (espin * 500)) ^ (1 / 0.6) 'Distancia mínima:
París (fórmula)
```

```
disminleroy = 8 / ((3400 * espin / (vcfcorre / 1000)) - 1)
```

```

textvcf.Text = Format(vcfcorre / 1000, "0.00000")
textdisminparis.Text = Format(disminparis, "0.00000")
textdisminleroy.Text = Format(disminleroy, "0.00000")

```

```
GoTo Line3 ' ***** fin de Diseño por rayo *****
```

Line2: ' ***** Inicio: diseño por Maniobra *****

```
porcent = Val(cboNBSpor.Text) / 100 ' Determina "sigma"
```

```

NBS = Val(cboNBS.Text) * 1000      ' Determina el NBS
espin = Val(Mid(cboespin.Text, 1, 7)) ' Determina el F. espinterométrico
fstorre = Val(cbofstorre.Text)    ' Fact./Seg. torre

If fstorre = 0 Then
    pel = MsgBox("El factor de seguridad (F.S.) no ha sido definido, entre el
dato", vbCritical, "Dato faltante (DELTA)")
    cbofstorre.SetFocus
    GoTo terminar
End If

If espin = 0 Then
    pel = MsgBox("El factor espinterométrico no ha sido definido, entre el dato",
vbCritical, "Dato faltante (DELTA)")
    cboespin.SetFocus
    GoTo terminar
End If

If NBS = 0 Then
    pel = MsgBox("El NBS no ha sido definido, entre el dato", vbCritical, "Dato
faltante (DELTA)")
    cboNBS.SetFocus
    GoTo terminar
End If

Rem **** Correcciones por altura, temperatura y humedad: rayo ****

hume = Val(texthumedad.Text)
tempe = Val(texttempe.Text)
altura = Val(textaltura.Text)

If tempe = 25 And altura = 0 And hume = 11 Then
    GoTo sincorrecciónnbs
Else
    GoTo correcciónnbs
End If

sincorrecciónnbs:

vcf = (NBS * fstorre) / (1 - 1.3 * (porcent))      'vcf= V(50%) fórmula
disminparis = ((vcf / 1000) / (espin * 500)) ^ (1 / 0.6) 'Distancia mínima: París
(fórmula)
disminleroy = 8 / ((3400 * espin / (vcf / 1000)) - 1)

textvcf.Text = Format(vcf / 1000, "0.00000")
textdisminparis.Text = Format(disminparis, "0.00000")
textdisminleroy.Text = Format(disminleroy, "0.00000")

```

GoTo Line3

correcciónnbs:

'Establece la primera aproximación de D sin corrección
 $vcf = (NBS * fstore) / (1 - 1.3 * (porcent))$ 'vcf= V(50%) fórmula
 $disminleroy = 8 / ((3400 * espin / (vcf / 1000)) - 1)$

$k3 = \text{calhmedad(hume)}$
 $\text{presión} = 76 * 10^{(-\text{altura} / 18336)}$ 'presión cmHg=b
 $k1 = \text{presión} / 76$ 'b/bo
 $k2 = (273 + 25) / (273 + \text{tempe})$

Rem Determinación de los coeficientes m y w-n

$m = \text{calm}(NBS / 1000)$
 $w = \text{calwn}(\text{disminleroy})$

$vcf = (NBS * fstore) / (1 - 1.3 * (porcent))$ 'vcf= V(50%) fórmula
 $vcfcorre = (vcf * (k3)^w) / ((k1)^m * (k2)^w)$ ' Corrección de vcf
 $disminparis = ((vcfcorre / 1000) / (espin * 500))^{(1 / 0.6)}$ 'Distancia mínima:

París (fórmula)

$disminleroy = 8 / ((3400 * espin / (vcfcorre / 1000)) - 1)$ 'Distancia mínima:

G - Leroy (fórmula)

$\text{textvcf.Text} = \text{Format}(vcfcorre / 1000, "0.00000")$
 $\text{textdisminparis.Text} = \text{Format}(disminparis, "0.00000")$
 $\text{textdisminleroy.Text} = \text{Format}(disminleroy, "0.00000")$

Rem ***** fin de Diseño por maniobra *****

Line3:

Rem ***** Cálculo de la cadena de aisladores *****

$\text{voltajemax} = \text{Val}(\text{textvolmax.Text})$ ' lee Vmáx
 $\text{espesor} = (\text{Val}(\text{textespesor.Text})) / 1000$ ' lee espesor
 $\text{lonais} = \text{voltajemax} / 150$ 'Longitud de la cadena de ais

If espesor = 0 Then

$\text{pre} = \text{MsgBox}(\text{"No ha definido el tipo de aislador a utilizar "}, \text{vbCritical}, \text{"Falta de datos (DELTA)"})$

btndatosaisla.SetFocus

GoTo terminar

End If

$\text{numais} = \text{lonais} / \text{espesor}$ 'Número de aisladores

```

textlongitud.Text = lonais
textnumais.Text = Int(numais) + 1

lonaisreal = (textnumais.Text) * espesor 'Longitud real
textlonreal.Text = Format(lonaisreal, "0.00000")

relon = Val(cborelon.Text) 'Lee la relación de longitud
discorregida = relon * lonaisreal 'distancia torre-conductor
textdiscorregida.Text = Format(discorregida, "0.00000")

Rem ***** Fin de cálculo de la cadena de aisladores *****

Rem ***** Distancia entre fases y altura mínima *****

fk = Val(Mid(cbofk.Text, 1, 7))

If textflecha.Text = "" Then
    opera = MsgBox("No se puede calcular las distancias mínimas sin el dato de
la flecha", vbExclamation, "Flecha")
    textflecha.SetFocus
    GoTo terminar
End If

flecha = textflecha.Text
volnom = textvolnom.Text
disff = fk * ((flecha + lonaisreal) ^ 0.5) + volnom / 150
hmin = 5.3 + volnom / 150

texthmin.Text = Format(hmin, "0.00000")
textdisff.Text = Format(disff, "0.00000")
terminar:
End Sub
Private Sub btndatosaisla_Click()
textespesor.Text = frmdataaisladores!textespesor.Text
textancho.Text = frmdataaisladores!textancho.Text
frmdataaisladores.Show vbModal
End Sub

Private Sub btnrestaurar_Click()
textaltura.Text = 0
texttempe.Text = 25
texthumedad.Text = 11
End Sub

Private Sub btnverangulo_Click()
Hide
frmangulo.Show

```

```
End Sub
```

```
Private Sub btnverfrmcorre_Click()  
frmcorrecciones.Show vbModal  
End Sub
```

```
Private Sub btnverfrmestruct_Click()  
Hide  
frmestruc.Show vbModal  
End Sub
```

```
Private Sub btnverfrmvoltaje_Click()  
Hide  
frmcalcable.Show  
End Sub
```

```
Private Sub btnverprincipo_Click()  
Unload Me  
Principio.Show  
End Sub
```

```
Private Sub cboBIL_Click()  
BIL = Val(cboBIL.Text)
```

```
Select Case BIL
```

```
Case 450
```

```
    cboNBS.List(0) = 185  **** NBS  
    cboNBS.List(1) = ""  
    cboNBS.List(2) = ""
```

```
    cboNBS.Text = 185  
    GoTo sandia
```

```
Case 550
```

```
    cboNBS.List(0) = 230  **** NBS  
    cboNBS.List(1) = ""  
    cboNBS.List(2) = ""
```

```
    cboNBS.Text = 230  
    GoTo sandia
```

```
Case 650
```

```
    cboNBS.List(0) = 275  **** NBS  
    cboNBS.List(1) = ""  
    cboNBS.List(2) = ""
```

```
    cboNBS.Text = 275
```

GoTo sandia

Case 750

cboNBS.List(0) = 325 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 325

GoTo sandia

Case 850

Select Case voltajemax

Case 245

cboNBS.List(0) = 360 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 360

GoTo sandia

Case 300

cboNBS.List(0) = 750 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 750

GoTo sandia

End Select

Case 950

Select Case voltajemax

Case 300

cboNBS.List(0) = 750 **** NBS

cboNBS.List(1) = 850

cboNBS.List(2) = ""

cboNBS.Text = 750

GoTo sandia

Case 362

cboNBS.List(0) = 850 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 850

GoTo sandia

End Select

Case 1050

Select Case voltajemax

Case 245

cboNBS.List(0) = 460 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 460
GoTo sandia

Case 300

cboNBS.List(0) = 850 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 850
GoTo sandia

Case 362

cboNBS.List(0) = 850 **** NBS
cboNBS.List(1) = 950
cboNBS.List(2) = ""

cboNBS.Text = 850
GoTo sandia

End Select

Case 1175

Select Case voltajemax

Case 362

cboNBS.List(0) = 950 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 950
GoTo sandia

Case 420

cboNBS.List(0) = 950 **** NBS
cboNBS.List(1) = 1050
cboNBS.List(2) = ""

cboNBS.Text = 950

GoTo sandia

End Select

Case 1300

Select Case voltajemax

Case 420

cboNBS.List(0) = 1050 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 1050

GoTo sandia

Case 525

cboNBS.List(0) = 1050 **** NBS

cboNBS.List(1) = 1175

cboNBS.List(2) = ""

cboNBS.Text = 1050

GoTo sandia

End Select

Case 1425

cboNBS.List(0) = 1050 **** NBS

cboNBS.List(1) = 1175

cboNBS.List(2) = 1300

cboNBS.Text = 1050

GoTo sandia

Case 1550

cboNBS.List(0) = 1175 **** NBS

cboNBS.List(1) = 1300

cboNBS.List(2) = 1425

cboNBS.Text = 1175

GoTo sandia

Case 1800

cboNBS.List(0) = 1300 **** NBS

cboNBS.List(1) = 1425

cboNBS.List(2) = 1550

cboNBS.Text = 1300

GoTo sandia

Case 1950

```
cboNBS.List(0) = 1550    **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 1550
GoTo sandia
```

Case 2100

```
cboNBS.List(0) = 1425    **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 1425
GoTo sandia
```

Case 2400

```
cboNBS.List(0) = 1550    **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 1550
GoTo sandia
```

Case Else

```
BIL = cboBIL.Text
```

```
sandia:
```

```
sandia2:
```

```
End Select
```

```
End Sub
```

```
Private Sub cboBIL_Change()
```

```
BIL = Val(cboBIL.Text)
```

```
Select Case BIL
```

Case 450

```
cboNBS.List(0) = 185    **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 185
GoTo sandia
```

Case 550

```
cboNBS.List(0) = 230    **** NBS
```

```
cboNBS.List(1) = ""  
cboNBS.List(2) = ""
```

```
cboNBS.Text = 230  
GoTo sandia
```

Case 650

```
cboNBS.List(0) = 275  **** NBS  
cboNBS.List(1) = ""  
cboNBS.List(2) = ""
```

```
cboNBS.Text = 275  
GoTo sandia
```

Case 750

```
cboNBS.List(0) = 325  **** NBS  
cboNBS.List(1) = ""  
cboNBS.List(2) = ""
```

```
cboNBS.Text = 325  
GoTo sandia
```

Case 850

```
Select Case voltajemax  
  Case 245  
    cboNBS.List(0) = 360  **** NBS  
    cboNBS.List(1) = ""  
    cboNBS.List(2) = ""
```

```
  cboNBS.Text = 360  
  GoTo sandia
```

Case 300

```
  cboNBS.List(0) = 750  **** NBS  
  cboNBS.List(1) = ""  
  cboNBS.List(2) = ""
```

```
  cboNBS.Text = 750  
  GoTo sandia
```

```
End Select
```

Case 950

```
Select Case voltajemax  
  Case 300  
    cboNBS.List(0) = 750  **** NBS  
    cboNBS.List(1) = 850
```

cboNBS.List(2) = ""

cboNBS.Text = 750
GoTo sandia

Case 362

cboNBS.List(0) = 850 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 850
GoTo sandia

End Select

Case 1050

Select Case voltajemax

Case 245

cboNBS.List(0) = 460 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 460
GoTo sandia

Case 300

cboNBS.List(0) = 850 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""

cboNBS.Text = 850
GoTo sandia

Case 362

cboNBS.List(0) = 850 **** NBS
cboNBS.List(1) = 950
cboNBS.List(2) = ""

cboNBS.Text = 850
GoTo sandia

End Select

Case 1175

Select Case voltajemax

Case 362

cboNBS.List(0) = 950 **** NBS

```
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 950
GoTo sandia
```

Case 420

```
cboNBS.List(0) = 950 **** NBS
cboNBS.List(1) = 1050
cboNBS.List(2) = ""
```

```
cboNBS.Text = 950
GoTo sandia
```

End Select

Case 1300

Select Case voltajemax

Case 420

```
cboNBS.List(0) = 1050 **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
```

```
cboNBS.Text = 1050
GoTo sandia
```

Case 525

```
cboNBS.List(0) = 1050 **** NBS
cboNBS.List(1) = 1175
cboNBS.List(2) = ""
```

```
cboNBS.Text = 1050
GoTo sandia
```

End Select

Case 1425

```
cboNBS.List(0) = 1050 **** NBS
cboNBS.List(1) = 1175
cboNBS.List(2) = 1300
```

```
cboNBS.Text = 1050
GoTo sandia
```

Case 1550

```
cboNBS.List(0) = 1175 **** NBS
cboNBS.List(1) = 1300
```

cboNBS.List(2) = 1425

cboNBS.Text = 1175

GoTo sandia

Case 1800

cboNBS.List(0) = 1300 **** NBS

cboNBS.List(1) = 1425

cboNBS.List(2) = 1550

cboNBS.Text = 1300

GoTo sandia

Case 1950

cboNBS.List(0) = 1550 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 1550

GoTo sandia

Case 2100

cboNBS.List(0) = 1425 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 1425

GoTo sandia

Case 2400

cboNBS.List(0) = 1550 **** NBS

cboNBS.List(1) = ""

cboNBS.List(2) = ""

cboNBS.Text = 1550

GoTo sandia

Case Else

BIL = cboBIL.Text

sandia:

sandia2:

End Select

End Sub

Private Sub Form_Activate()

textvolnom.Text = frmcalcable!cbovoltaje.Text

```
If flecha = Empty Then
    flecha = 6
End If
textflecha.Text = flecha 'Flecha es una variable pública
End Sub
```

```
Private Sub Form_Load()
textaltura.Text = frmdatos.textaltura.Text
texttempe.Text = frmdatos.texttempe.Text
texthumedad.Text = frmdatos.texthumedad.Text
End Sub
```

```
Private Sub optBIL_Click()
If optBIL.Value = True Then
    lbIBIL.Enabled = True
    cboBIL.Enabled = True
    lbIBIL1.Enabled = True
    lbIBILpor.Enabled = True
    cboBILpor.Enabled = True
    lbINBS.Enabled = False
    cboNBS.Enabled = False
    lbINBS1.Enabled = False
    lbINBSpor.Enabled = False
    cboNBSpor.Enabled = False
Else
    lbIBIL.Enabled = False
    cboBIL.Enabled = False
    lbIBIL1.Enabled = False
    lbIBILpor.Enabled = False
    cboBILpor.Enabled = False
    lbINBS.Enabled = True
    cboNBS.Enabled = True
    lbINBS1.Enabled = True
    lbINBSpor.Enabled = True
    cboNBSpor.Enabled = True
End If
End Sub
```

```
Private Sub optNBS_Click()
If optNBS.Value = True Then
    lbIBIL.Enabled = False
    cboBIL.Enabled = False
    lbIBIL1.Enabled = False
    lbIBILpor.Enabled = False
    cboBILpor.Enabled = False
    lbINBS.Enabled = True
    cboNBS.Enabled = True
End If
End Sub
```

```

    lbINBS1.Enabled = True
    lbINBSpor.Enabled = True
    cboNBSpor.Enabled = True
    Else
    lbIBIL.Enabled = True
    cboBIL.Enabled = True
    lbIBIL1.Enabled = True
    lbIBILpor.Enabled = True
    cboBILpor.Enabled = True
    lbINBS.Enabled = False
    cboNBS.Enabled = False
    lbINBS1.Enabled = False
    lbINBSpor.Enabled = False
    cboNBSpor.Enabled = False
End If
End Sub

Private Sub textaltura_Change()
    If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And
Val(texthumedad.Text) = 11 Then
        btnverfrmcorre.Enabled = False
        btnrestaurar.Enabled = False
        GoTo final
    Else
        btnverfrmcorre.Enabled = True
        btnrestaurar.Enabled = True
    End If
final:
End Sub

Private Sub textfuga_Click()
    Rem ***** Cálculo de la cadena de aisladores por contaminación *****
    If contaminacion = "" Then
        GoTo abajo
    Else
        GoTo fun
    End If

fun:
    Select Case contaminacion
    Case "Ligera"
        'kfe = 16
        kfe = frmactualizar.textligera.Text
        GoTo Line1
    Case "Media"
        'kfe = 20
        kfe = frmactualizar.textmedia.Text

```



```

    GoTo Line1
Case "Alta"
    'kfe = 25
    kfe = frmactualizar.textalta.Text
    GoTo Line1
Case "Muy Alta"
    'kfe = 31
    kfe = frmactualizar.textmalta.Text
    GoTo Line1
Case Else
    Item = MsgBox("No se ha definido el nivel de contaminación: " &
contaminacion, 48, "Nivel de contaminación sin definir")
    GoTo Line2
End Select

```

Line1:

```

    espesor = (Val(textespesor.Text)) / 1000 'lee espesor
    disfuga = Val(textfuga.Text)
    volmax = Val(textvolmax.Text)

    If volmax = 0 Then
        pel = MsgBox("Defina el voltaje máximo en KV !", vbDefaultButton1, "Voltaje
no definido (DELTA)")
        textvolmax.SetFocus
        GoTo abajo
    End If

    If disfuga = 0 Then
        pel = MsgBox("La distancia de fuga no puede ser cero, defina el dato !",
vbCritical, "Distancia de fuga no definida (DELTA)")
        textfuga.SetFocus
        GoTo abajo
    End If

    napcc = kfe * volmax / disfuga '# de aislads./corrección de contaminación

    napccmsj = Int(napcc) + 1
    msjais = "Nivel de contaminación: " & contaminacion & " (" & kfe & " mm/KV) " &
" ; Aisladores necesarios: " & napccmsj & " "
    textnumais.ToolTipText = msjais

    napccmsjreal = napccmsj * espesor 'Longitud vertical real
    msjlon = "Nivel de contaminación: " & contaminacion & " ; Longitud de la
cadena: " & napccmsjreal & " m "
    textlonreal.ToolTipText = msjlon

```

Line2:

abajo:

```
End Sub
```

```
Private Sub textfuga_Change()
```

```
Rem ***** Cálculo de la cadena de aisladores por contaminación *****
```

```
If contaminacion = "" Then
```

```
    GoTo abajo
```

```
Else
```

```
    GoTo fun
```

```
End If
```

fun:

```
Select Case contaminacion
```

```
Case "Ligera"
```

```
    'kfe = 16
```

```
    kfe = frmactualizar.textligera.Text
```

```
    GoTo Line1
```

```
Case "Media"
```

```
    'kfe = 20
```

```
    kfe = frmactualizar.textmedia.Text
```

```
    GoTo Line1
```

```
Case "Alta"
```

```
    'kfe = 25
```

```
    kfe = frmactualizar.textalta.Text
```

```
    GoTo Line1
```

```
Case "Muy Alta"
```

```
    'kfe = 31
```

```
    kfe = frmactualizar.textmalta.Text
```

```
    GoTo Line1
```

```
Case Else
```

```
    Item = MsgBox("No se ha definido el nivel de contaminación: " &  
contaminacion, 48, "Nivel de contaminación sin definir")
```

```
    GoTo Line2
```

```
End Select
```

Line1:

```
espesor = (Val(textespesor.Text)) / 1000 'lee espesor
```

```
disfuga = Val(textfuga.Text)
```

```
volmax = Val(textvolmax.Text)
```

```
If volmax = 0 Then
```

```
    pel = MsgBox("Defina el voltaje máximo en KV !", vbDefaultButton1, "Voltaje  
no definido (DELTA)")
```

```
    'textvolmax.SetFocus
```

```
    GoTo abajo
```

```
End If
```

```

If disfuga = 0 Then
    pel = MsgBox("La distancia de fuga no puede ser cero, defina el dato ! ",
vbCritical, "Distancia de fuga no definida (DELTA)")
    textfuga.SetFocus
    GoTo abajo
End If

napcc = kfe * volmax / disfuga '# de aislads./corrección de contaminación

napccmsj = Int(napcc) + 1
msjais = " Nivel de contaminación: " & contaminacion & " (" & kfe & " mm/KV) " &
"; Aisladores necesarios: " & napccmsj & " "
textnumais.ToolTipText = msjais

napccmsjreal = napccmsj * espesor 'Longitud vertical real
msjlon = " Nivel de contaminación: " & contaminacion & " ; Longitud de la
cadena: " & napccmsjreal & " m "
textlonreal.ToolTipText = msjlon

Line2:
abajo:
    End Sub

Private Sub texthumedad_Change()
    If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And
Val(texthumedad.Text) = 11 Then
        btnverfrmcorre.Enabled = False
        btnrestaurar.Enabled = False
        GoTo final
    Else
        btnverfrmcorre.Enabled = True
        btnrestaurar.Enabled = True
    End If
final:
    End Sub

Private Sub textnumais_Change()
    Rem datos
    nroais = Val(textnumais.Text)      '# de aisladores
    altuaisla = Val(textespesor.Text) / 1000 'Altura del aislador de mm a mts
    relameca = Val(cborelon.Text)      'factor mecánico debido a herrajes
    Rem Cálculo de longitudes
    Ircadena = nroais * altuaisla      'Longitud de la cadena de aisladores
    Ircadenacorreg = relameca * Ircadena 'Longitud real de toda la cadena (con
herrajes)
    textlonreal.Text = Ircadena
    textdiscorregida.Text = Ircadenacorreg

```

End Sub

```
Private Sub texttempe_Change()
If Val(textaltura.Text) = 0 And Val(texttempe.Text) = 25 And texthumedad.Text =
11 Then
    btnverfrmcorre.Enabled = False
    btnrestaurar.Enabled = False
    GoTo final
Else
    btnverfrmcorre.Enabled = True
    btnrestaurar.Enabled = True
End If
final:
End Sub
```

```
Private Sub textvcf_Change()
    espin = Val((Mid(cboespin.Text, 1, 7))) 'Determina el F. espinterométrico
    vcfcorre = Val(textvcf.Text)          'VCF de KV a V

    disminparis = (vcfcorre / (espin * 500)) ^ (1 / 0.6) 'Distancia mínima: París
    (fórmula)
    disminleroy = 8 / ((3400 * espin / (vcfcorre)) - 1)    'Distancia mínima: G - Leroy
    (fórmula)
```

```
textdisminparis.Text = Format(disminparis, "0.00000")
textdisminleroy.Text = Format(disminleroy, "0.00000")
End Sub
```

```
Private Sub textvolmax_Change()
Rem ***** Inicio BIL *****
voltajemax = Val(textvolmax.Text)
Select Case voltajemax
```

```
Case 123
    cboBIL.List(0) = 450    '*** BIL
    cboBIL.List(1) = 550
    cboBIL.List(2) = ""
    cboBIL.List(3) = ""
    cboBIL.List(4) = ""
    cboBIL.List(5) = ""

    cboBIL.Text = 450
```

```
Case 145
    cboBIL.List(0) = 450    '*** BIL
    cboBIL.List(1) = 550
    cboBIL.List(2) = 650
```

```
cboBIL.List(3) = ""  
cboBIL.List(4) = ""  
cboBIL.List(5) = ""
```

```
cboBIL.Text = 450
```

Case 170

```
cboBIL.List(0) = 550 **** BIL  
cboBIL.List(1) = 650  
cboBIL.List(2) = 750  
cboBIL.List(3) = ""  
cboBIL.List(4) = ""  
cboBIL.List(5) = ""
```

```
cboBIL.Text = 550
```

Case 245

```
cboBIL.List(0) = 650 **** BIL  
cboBIL.List(1) = 750  
cboBIL.List(2) = 850  
cboBIL.List(3) = 950  
cboBIL.List(4) = 1050  
cboBIL.List(5) = ""
```

```
cboBIL.Text = 650
```

Case 300

```
cboBIL.List(0) = 850 **** BIL  
cboBIL.List(1) = 950  
cboBIL.List(2) = 1050  
cboBIL.List(3) = ""  
cboBIL.List(4) = ""  
cboBIL.List(5) = ""
```

```
cboBIL.Text = 850
```

Case 362

```
cboBIL.List(0) = 950 **** BIL  
cboBIL.List(1) = 1050  
cboBIL.List(2) = 1175  
cboBIL.List(3) = ""  
cboBIL.List(4) = ""  
cboBIL.List(5) = ""
```

```
cboBIL.Text = 950
```

Case 420

```

cboBIL.List(0) = 1050   **** BIL
cboBIL.List(1) = 1175
cboBIL.List(2) = 1300
cboBIL.List(3) = 1425
cboBIL.List(4) = ""
cboBIL.List(5) = ""

```

```

cboBIL.Text = 1050

```

Case 525

```

cboBIL.List(0) = 1175   **** BIL
cboBIL.List(1) = 1300
cboBIL.List(2) = 1425
cboBIL.List(3) = 1550
cboBIL.List(4) = ""
cboBIL.List(5) = ""

```

```

cboBIL.Text = 1175

```

Case 765

```

cboBIL.List(0) = 1425   **** BIL
cboBIL.List(1) = 1550
cboBIL.List(2) = 1800
cboBIL.List(3) = 1950
cboBIL.List(4) = 2100
cboBIL.List(5) = 2400

```

```

cboBIL.Text = 1425

```

Case Else

```

voltajemax = Val(textvolmax.Text)

```

```

cboBIL.List(0) = ""      **** BIL
cboBIL.List(1) = ""
cboBIL.List(2) = ""
cboBIL.List(3) = ""
cboBIL.List(4) = ""
cboBIL.List(5) = ""

```

```

cboNBS.List(0) = ""     **** NBS
cboNBS.List(1) = ""
cboNBS.List(2) = ""
cboNBS.List(3) = ""
cboNBS.List(4) = ""

```

```

If optBIL.Value = True Then

```

```

        cboBIL.Text = ""
    Else
        cboNBS.Text = ""
    End If

End Select ' ***** fin de BIL *****

Rem ***** Cálculo de la cadena de aisladores por contaminación *****
If contaminacion = "" Then
    GoTo abajo
Else
    GoTo fun
End If

fun:
Select Case contaminacion
Case "Ligera"
    'kfe = 16
    kfe = frmactualizar.textligera.Text
    GoTo Line1
Case "Media"
    'kfe = 20
    kfe = frmactualizar.textmedia.Text
    GoTo Line1
Case "Alta"
    'kfe = 25
    kfe = frmactualizar.textalta.Text
    GoTo Line1
Case "Muy Alta"
    'kfe = 31
    kfe = frmactualizar.textmalta.Text
    GoTo Line1
Case Else
    Item = MsgBox("No se ha definido el nivel de contaminación: " &
contaminacion, 48, "Nivel de contaminación sin definir")
    GoTo Line2
End Select

Line1:
    espesor = (Val(textespesor.Text)) / 1000 'lee espesor
    disfuga = Val(textfuga.Text)
    volmax = Val(textvolmax.Text)

    If volmax = 0 Then
        pel = MsgBox("Defina el voltaje máximo en KV !", vbDefaultButton1, "Voltaje
no definido (DELTA)")
        textvolmax.SetFocus

```

```

    GoTo abajo
End If

If disfuga = 0 Then
    GoTo abajo
End If

napcc = kfe * volmax / disfuga '# de aislads./corrección de contaminación

napccmsj = Int(napcc) + 1
msjais = "Nivel de contaminación: " & contaminacion & " (" & kfe & " mm/KV) " &
"; Aisladores necesarios: " & napccmsj & " "
textnumais.ToolTipText = msjais

napccmsjreal = napccmsj * espesor 'Longitud vertical real
msjlon = "Nivel de contaminación: " & contaminacion & " ; Longitud de la
cadena: " & napccmsjreal & " m "
textlonreal.ToolTipText = msjlon

Line2:
abajo:
End Sub

Private Sub textvolnom_Change()
voltajenom = Val(textvolnom.Text)

Select Case voltajenom

    Case 115
textvolmax.Text = 123

    Case 138
textvolmax.Text = 145

    Case 161
textvolmax.Text = 170

    Case 230
textvolmax.Text = 245

    Case 287
textvolmax.Text = 300

    Case 345
textvolmax.Text = 362

    Case 400

```



```
textvolmax.Text = 420
```

```
Case 500
```

```
textvolmax.Text = 525
```

```
Case 765
```

```
textvolmax.Text = 765
```

```
Case Else
```

```
voltajenom = textvolnom.Text
```

```
textvolmax.Text = 0
```

```
End Select
```

```
End Sub
```

ANGULO DE DESVIACION DE LA CADENA

DE AISLADORES

(frmangulo.frm)

```
Private Sub btnahg_Click()
Hide
frmblindaje.Show
End Sub
```

```
Private Sub btncalcular_Click()
Static pi
pi = 3.14159265358979
btnahg.Enabled = True
```

```
If chktn = Checked Then ' *** si es tangente ***
shptan.BorderStyle = 1
If chkang.Value = Unchecked Then
shpang.BorderStyle = 3
End If
```

```
ncf = Val(textncf.Text)
wc = (Val(textpesototal.Text)) * (Val(textvano.Text)) * ncf / 1000
wah = Val(textpesoah.Text)
fvc = Val(textfvc.Text)
fvah = Val(textfvah.Text)
i = Atn((0.5 * fvah + fvc) / (0.5 * wah + wc)) * 180 / pi
texttandesvia.Text = Format(i, "0.0000") & " °"
End If
```

```
If chkang = Checked Then ' *** si es de angulo ***
shpang.BorderStyle = 1
If chktn.Value = Unchecked Then
shptan.BorderStyle = 3
End If
```

```
lblangulo.Enabled = True
textangulo.Enabled = True
ncf = Val(textncf.Text)
tension = Val(texttension.Text)
Angulo = (Val(textangulo.Text)) * pi / 180
wc = (Val(textpesototal.Text)) * (Val(textvano.Text)) * (Val(textncf.Text)) /
1000
wah = Val(textpesoah.Text)
fvc = Val(textfvc.Text)
```

```

fvah = Val(textfvah.Text)
If optconviento.Value = True Then ' condición de viento
    ft = 2 * tension * Sin(Angulo / 2) * ncf
    fvi = fvc * Cos(Angulo / 2)
    fr = ft + fvi
    iang = Atn((ft + (fvi + 0.5 * fvah)) / (0.5 * wah + wc)) * 180 / pi
    textft.Text = Format(ft, "0.0000")
    textfv.Text = Format(fvi, "0.0000")
    textfr.Text = Format(fr, "0.0000")
    textangdesvia.Text = Format(iang, "0.0000") & " °"
End If

If optsinviento.Value = True Then
    ft = 2 * tension * Sin(Angulo / 2) * ncf
    fr = ft
    iang = Atn(ft / (0.5 * wah + wc)) * 180 / pi
    textft.Text = ft
    textfv.Text = 0
    textfr.Text = fr
    textangdesvia.Text = Format(iang, , "0.0000") & " °"
End If

End If

If chktan.Value = Unchecked And chkang.Value = Unchecked Then
    lblangulo.Enabled = False
    textangulo.Enabled = False
    defi = MsgBox("Establezca al menos una opción para proceder al cálculo del
ángulo de desviación", vbInformation, "Verificación de opciones (DELTA)")
    chktan.SetFocus
End If

If chktan.Value = Unchecked Then
    texttandesvia.Text = 0
End If

If chkang.Value = Unchecked Then
    lblangulo.Enabled = False
    textangulo.Enabled = False
    textft.Text = 0
    textfv.Text = 0
    textfr.Text = 0
    textangdesvia.Text = 0
End If

End Sub
Private Sub btnfrmcalcable_Click()

```

```
Hide
frmcalcable.Show
End Sub
```

```
Private Sub btnveraisla_Click()
Hide
frmaisla.Show
End Sub
```

```
Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub
```

```
Private Sub cboff_Click()
velocidad = (textvelocidad.Text) * 1000 / 3600 'convierte de km/hr a m/s
ff = Mid(cboff.Text, 1, 5)
q = cbopesovol.Text
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)
textpresion.Text = presion
End Sub
```

```
Private Sub cboff_Change()
velocidad = (Val(textvelocidad.Text)) * 1000 / 3600 'Convierte de km/hr a m/s
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)
textpresion.Text = presion
End Sub
```

```
Private Sub cbopesovol_Click()
velocidad = (textvelocidad.Text) * 1000 / 3600 'Convierte de km/hr a m/s
ff = Mid(cboff.Text, 1, 5)
q = cbopesovol.Text
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)
textpresion.Text = presion
End Sub
```

```
Private Sub cbopesovol_Change()
velocidad = (Val(textvelocidad.Text)) * 1000 / 3600 'Convierte de km/hr a m/s
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)
textpresion.Text = presion
End Sub
```

```
Private Sub chkang_Click()
```

```
lblang.Enabled = True
lblangulo.Enabled = True
textangulo.Enabled = True
textangulo.Text = 25
fraang.Enabled = True
optconviento.Enabled = True
optsinviento.Enabled = True
lblt.Enabled = True
texttension.Enabled = True
shpang.BorderStyle = 1
textft.Enabled = True
textfv.Enabled = True
textfr.Enabled = True
textangdesvia.Enabled = True
texttension.SetFocus

If chkang.Value = Unchecked Then
    lblang.Enabled = False
    lblangulo.Enabled = False
    textangulo.Enabled = False
    'textangulo.Text = ""
    fraang.Enabled = False
    optconviento.Enabled = False
    optsinviento.Enabled = False
    lblt.Enabled = False
    texttension.Enabled = False
    shpang.BorderStyle = 3
    textft.Text = 0
    textfv.Text = 0
    textfr.Text = 0
    textangdesvia.Text = 0
    textft.Enabled = False
    textfv.Enabled = False
    textfr.Enabled = False
    textangdesvia.Enabled = False
    btncalcular.SetFocus
End If
End Sub

Private Sub chktan_Click()
    lbltan.Enabled = True
    shptan.BorderStyle = 1
    texttandesvia.Enabled = True

    If chktan.Value = Unchecked Then
        lbltan.Enabled = False
        shptan.BorderStyle = 3
    End If
End Sub
```

```

    texttandesvia.Text = 0
    texttandesvia.Enabled = False
    btncalcular.SetFocus
End If
End Sub

Private Sub Form_Activate()
Rem Para cables:
texttipo.Text = frmcalcable!texttipo.Text
textmcm.Text = frmcalcable!textmcm.Text
textpesototal.Text = frmcalcable!textpeso.Text
textdiametroext.Text = frmcalcable!textdiame.Text
textncf.Text = frmcalcable!ncf.Text

Rem Para aisladores:
textlonreal.Text = frmaisla!textdiscorregida.Text
textancho.Text = Val(frmaisla!textancho.Text)

Rem Establece el peso total de los aisla/herra
pesoa = Val(frmaisla!textpeso.Text)
numais = Val(frmaisla!textnumais.Text)
pesoah = (pesoa * numais) + Val(pesot)
textpesoah1.Text = pesoah
End Sub

Private Sub Form_Load()
textvano.Text = frmdatos.textvano.Text
End Sub

Private Sub SpinButton1_SpinDown()
Dim dblrate
dblrate = textncadenas.Text

If dblrate > 1 Then
    textncadenas.Text = dblrate - 1
Else
    textncadenas.Text = 1
End If
textncadenas.Refresh
End Sub

Private Sub SpinButton1_SpinUp()
textncadenas.Text = (textncadenas.Text) + 1

If dblrate >= 6 Then

```

```

    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")

```

```
End If
```

```
textncadenas.Refresh
```

```
End Sub
```

```
Private Sub textancho_Click()
```

```
ncadenas = Val(textncadenas.Text)
```

```
lonreal = Val(textlonreal.Text)
```

```
ancho = Val(Val(textancho.Text) / 1000) ' Convierte de mm a m
```

```
presion = Val(textpresion.Text)
```

```
fvah = (presion * ancho * lonreal * ncadenas)
```

```
textfvah.Text = Format(fvah, "0.00000")
```

```
End Sub
```

```
Private Sub textancho_Change()
```

```
ncadenas = Val(textncadenas.Text)
```

```
lonreal = Val(textlonreal.Text)
```

```
ancho = Val(Val(textancho.Text) / 1000) ' Convierte de mm a m
```

```
presion = Val(textpresion.Text)
```

```
fvah = (presion * ancho * lonreal * ncadenas)
```

```
textfvah.Text = Format(fvah, "0.00000")
```

```
End Sub
```

```
Private Sub textdiametroext_Click()
```

```
diametroext = Val((Val(textdiametroext.Text)) / 1000)
```

```
ncf = Val(textncf.Text)
```

```
vano = Val(textvano.Text)
```

```
presion = Val(textpresion.Text)
```

```
fvc = (ncf * presion * diametroext * vano)
```

```
textfvc.Text = fvc
```

```
End Sub
```

```
Private Sub textdiametroext_Change()
```

```
diametroext = Val((Val(textdiametroext.Text)) / 1000)
```

```
ncf = Val(textncf.Text)
```

```
vano = Val(textvano.Text)
```

```
presion = Val(textpresion.Text)
```

```
fvc = (ncf * presion * diametroext * vano)
```

```
textfvc.Text = fvc
```

```
End Sub
```

```
Private Sub textlonreal_Click()
```

```
ncadenas = Val(textncadenas.Text)
```

```

lonreal = Val(textlonreal.Text)
ancho = Val(Val(textancho.Text) / 1000) 'Convierte de mm a m
presion = Val(textpresion.Text)
fvah = presion * ancho * lonreal * ncadenas
textfvah.Text = Format(fvah, "0.00000")

```

```
End Sub
```

```

Private Sub textlonreal_Change()
ncadenas = Val(textncadenas.Text)
lonreal = Val(textlonreal.Text)
ancho = Val(Val(textancho.Text) / 1000) 'Convierte de mm a m
presion = Val(textpresion.Text)
fvah = presion * ancho * lonreal * ncadenas
textfvah.Text = Format(fvah, "0.00000")
End Sub

```

```

Private Sub textncadenas_Click()
Rem Sugerencias
ncadenas = Int(textncadenas.Text)
textncadenas.Text = ncadenas

```

```

Select Case ncadenas
Case Is = 0
def = MsgBox("Valor inaceptable ", 48, "Valor negativo ")
Line1:
textncadenas.Text = 1
Case Is < 1
def = MsgBox("Esta de acuerdo en el siguiente valor: " &
Abs(Int(ncadenas)), 4, "Parámetro inaceptable")
If def = vbYes Then
textncadenas.Text = Abs(Int(textncadenas.Text))
Else
GoTo Line1
End If
End Select
Rem Fin de sugerencias

```

```

ncadenas = Val(textncadenas.Text)
lonreal = Val(textlonreal.Text)
ancho = (Val(textancho.Text)) / 1000 'Convierte de mm a m
presion = Val(textpresion.Text)

```

```

fvah = presion * ancho * lonreal * ncadenas
textfvah.Text = Format(fvah, "0.00000")

```

```
ncadenas = Val(textncadenas.Text)
```



```

If textncf.Text >= 2 Then
    pesoah = (textncf.Text) * ncadenas
End If
End Sub

Private Sub textncadenas_Change()
    Rem Sugerencias
    ncadenas = Int(Val(textncadenas.Text))
    textncadenas.Text = ncadenas

    Select Case ncadenas

        Case Is >= 3
            mensa = MsgBox("Se puede admitir hasta 3 cadenas de aisladores, sin embargo su uso no es muy común, por lo que se aceptará solo como manera de ejemplo para simular el ángulo de desviación. ", vbExclamation, "Número de cadena de aisladores")
            GoTo Line2
            Case Is = 0
                def = MsgBox("No existen Subconductores, por lo menos debe de existir un conductor ", 48, "Valor negativo ")
                Line1:
                    textncadenas.Text = 1
            Case Is < 1
                def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncadenas)), 4, "Parámetro inaceptable")
                If def = vbYes Then
                    textncadenas.Text = Abs(Int(textncadenas.Text))
                Line2:
                    Else
                        GoTo Line1
                    End If
            End Select
        Rem Fin de sugerencias

        ncadenas = Val(textncadenas.Text)
        lonreal = Val(textlonreal.Text)
        ancho = (Val(textancho.Text)) / 1000 'Convierte de mm a m
        presion = Val(textpresion.Text)

        fvah = presion * ancho * lonreal * ncadenas
        textfvah.Text = Format(fvah, "0.00000")

        Rem Establece el peso total para más de una cadena
        textpesoah.Text = Val(textpesoah1.Text) * ncadenas
    End Sub

```

```

Private Sub textncf_Click()
diametroext = Val((Val(textdiametroext.Text)) / 1000)
ncf = Val(textncf.Text)
vano = Val(textvano.Text)
presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = fvc
End Sub

```

```

Private Sub textncf_Change()
diametroext = Val((Val(textdiametroext.Text)) / 1000)
ncf = Val(textncf.Text)
vano = Val(textvano.Text)
presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = Format(fvc, "0.00000")

```

```

If ncf = 1 Then
    SpinButton1.Enabled = False
    lblncadenas.Enabled = False
    textncadenas.Enabled = False
Else
    SpinButton1.Enabled = True
    lblncadenas.Enabled = True
    textncadenas.Enabled = True
End If
End Sub

```

```

Private Sub textpesoah1_Change()
textpesoah.Text = textpesoah1.Text
End Sub

```

```

Private Sub textpresion_Click()
Rem Cálculo de velocidad
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
presion = Val(textpresion.Text)
velocidad = (3600 / 1000) * Sqr(2 * 9.81 * presion / (0.55 * ff * q)) 'Pasa de m/s a
km/hr
textvelocidad.Text = velocidad

```

```

Rem Fuerza en los cables
diametroext = Val((Val(textdiametroext.Text)) / 1000)
ncf = Val(textncf.Text)
vano = Val(textvano.Text)

```

```

presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = Format(fvc, "0.00000")

```

```

Rem Fuerza en la cadena de aisladores
ncadenas = Val(textncadenas.Text)
lonreal = Val(textlonreal.Text)
ancho = Val(Val(textancho.Text) / 1000) 'Convierte de mm a m
presion = Val(textpresion.Text)
fvah = presion * ancho * lonreal * ncadenas
textfvah.Text = Format(fvah, "0.00000")
End Sub

```

```

Private Sub textpresion_Change()
Rem Fuerza en los cables
diametroext = Val((Val(textdiametroext.Text)) / 1000)
ncf = Val(textncf.Text)
vano = Val(textvano.Text)
presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = Format(fvc, "0.00000")

```

```

Rem Fuerza en la cadena de aisladores
ncadenas = Val(textncadenas.Text)
lonreal = Val(textlonreal.Text)
ancho = Val(Val(textancho.Text) / 1000) ' Convierte de mm a m
presion = Val(textpresion.Text)
fvah = presion * ancho * lonreal * ncadenas
textfvah.Text = Format(fvah, "0.00000")
End Sub

```

```

Private Sub textvano_Click()
diametroext = Val((Val(textdiametroext.Text)) / 1000)
ncf = Val(textncf.Text)
vano = Val(textvano.Text)
presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = Format(fvc, "0.00000")
End Sub

```

```

Private Sub textvano_Change()
diametroext = Val((Val(textdiametroext.Text)) / 1000) 'Diámetro en m
ncf = Val(textncf.Text)
vano = Val(textvano.Text)
presion = Val(textpresion.Text)
fvc = (ncf * presion * diametroext * vano)
textfvc.Text = Format(fvc, "0.00000")

```

End Sub

```
Private Sub textvelocidad_Click()  
velocidad = Val((Val(textvelocidad.Text)) * 1000 / 3600) 'Convierte de km/hr a  
m/s  
ff = Val(Mid(cboff.Text, 1, 5))  
q = Val(cbopesovol.Text)  
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)  
textpresion.Text = presion  
End Sub
```

```
Private Sub textvelocidad_Change()  
velocidad = Val((Val(textvelocidad.Text)) * 1000 / 3600) 'Pasa de km/hr a m/s  
ff = Val(Mid(cboff.Text, 1, 5))  
q = Val(cbopesovol.Text)  
presion = 0.55 * velocidad ^ 2 * ff * q / (2 * 9.81)  
textpresion.Text = presion  
End Sub
```

CALCULO DE LA ALTURA DEL CABLE DE GUARDA Y

ANGULO CRITICO DE BLINDAJE

(frmblindaje.frm)

```

Private Sub btnang_Click()
Static pi
pi = 3.14159265358979

Rem Lectura de Datos
BIL = Val(textbil.Text)
zc = Val(textzc.Text)
If zc = 0 Then
    ndd = MsgBox("Introduzca el dato de la impedancia característica Zc",
vbCritical, "Zc no definida ")
    textzc.SetFocus
    GoTo fin
End If
If BIL = 0 Then
    ndd = MsgBox("Introduzca el dato del BIL ", vbCritical, "BIL no definido ")
    textbil.SetFocus
    GoTo fin
End If

If itera = 0 And dif = 0 Then
    itera = Val(frmitera!textitera.Text) 'lee el valor en frmitera
    dif = Val(frmitera!texterr.Text) 'lee el valor en frmitera
End If

B = Val(texthgc.Text)
Y = Val(texthcon.Text)

If Y = 0 Then
    ndd = MsgBox("Introduzca el dato de la altura del conductor más elevado ",
vbCritical, "Altura de conductor no definido ")
    texthcon.SetFocus
    GoTo fin
End If

Dim beta(5000), teta(5000), c(5000)
c(0) = 1 + B
amp = 2 * BIL / zc
rsc = 2 * amp + 30 * (1 - Exp(-amp / 6.8))
textamp.Text = amp

```

```

texttrsc.Text = rsc
For i = 1 To itera Step 1
  X = c(i - 1) / (2 * rsc)
  beta(i) = Atn((X / Sqr(-X * X + 1)))
  r = ((Y / rsc) - 1)
  teta(i) = -beta(i) - Atn((r / Sqr(-r * r + 1)))
  c(i) = B / Cos(teta(i))
  If Abs(c(i - 1) - c(i)) < dif Then
    GoTo finalizar
  End If

  If i = itera Then
    GoTo finalizar
  End If

finalizar:
A = c(i) * Sin(teta(i))
texttetasc.Text = teta(i) * 180 / pi
textc.Text = c(i)
texta.Text = A

msjang = "Angulo Crítico de Blindaje: "
msjang = msjang & Format(teta(i) * 180 / pi, "0.0000") & " ° "
lblang.ToolTipText = msjang

msjc = "Distancia directa óptima entre los cables: "
msjc = msjc & Format(c(i), "0.0000") & " m "
lblc.ToolTipText = msjc

msja = "Distancia horizontal óptima: "
msja = msja & Format(A, "0.0000") & " m "
lbla.ToolTipText = msja

Next i
fin:
End Sub

Private Sub btn calcularhg_Click()
btnvervistazo.Enabled = True
btnang.Enabled = True
btnverfrmitera.Enabled = True
imga.Visible = True
imga.Picture = LoadPicture("C:\Delta\Graficas\torre17cortada.wmf")

Static pi
pi = 3.14159265358979

```

Rem Datos

$hmin = Val(texthmin.Text)$

$flecha = Val(textflecha.Text)$

$discorregida = Val(textdiscorregida.Text)$

$angdesvia = (Val(textangdesvia.Text) * \pi / 180)$

$disminleroy = Val(textdisminleroy.Text)$

$angblin = Val(textangblin.Text) * \pi / 180$

Rem Corrección de la distancia horizontal

$disrealleroy = disminleroy + discorregida * \sin(angdesvia)$

$textdisreal.Text = disrealleroy$

Rem Altura del cable de guarda

$hgc = (disrealleroy / \tan(angblin)) - discorregida$ 'Altura respecto al brazo

Select Case tipestruct 'Variable que se registra en el "CLICK" de cada imagen de frmestruc

Case 1

$acma = disminleroy + hmin + flecha + discorregida$ 'Altura del con. más elevado

Case 2, 5

$acma = hmin + flecha$ 'Altura del con. más elevado

Case 3, 4

$acma = 2 * disminleroy + hmin + flecha + 2 * discorregida$ 'Altura del con. más elevado

Case Else

$acma = hmin + flecha$

End Select

$hcon = acma$ 'Altura del conductor/sin ángulo de desviación

$hg = hgc + hcon + discorregida$ 'Altura del conductor respecto al suelo

$hconang = hcon + discorregida * (1 - \cos(angdesvia))$ 'altura del conductor/con ángulo de desviación

$texthconang.Text = hconang$

$texthcon.Text = hcon$

$texthgc.Text = hgc$

$texthg.Text = hg$

Rem Determinación si se usará cadenas de aisladores en "V"

$X = 1 / 3$

```

angv = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi
If angv <= angdesvia * 180 / pi Then
    MsgBox "Se sugiere utilizar cadenas de aisladores en 'V' "
End If

```

```

msjca = " Cedena de aisladores y herrajes: "
msjca = msjca & discorregida & " m "
lblaisherr.ToolTipText = msjca

```

```

msjbraso = " Distancia del cable de guarda al barazo "
msjbraso = msjbraso & Format(hgc, "0.00000") & " m "
lblbraso.ToolTipText = msjbraso
End Sub

```

```

Private Sub btnverangulo_Click()
Hide
frmangulo.Show
End Sub

```

```

Private Sub btnverfrmitera_Click()
rastreo2 = "blindaje" 'Parámetro para regresar a esta forma despues
            'de cerrar frmitera
frmitera.Show vbModal
End Sub

```

```

Private Sub btnverprincipiol_Click()
Unload Me
Principio.Show
End Sub

```

```

Private Sub btnvervistazo_Click()
Hide
frmvistazo.Show
End Sub

```

```

Private Sub Form_Activate()
texthmin.Text = Val(frmaisla!texthmin.Text)
textflecha.Text = Val(flecha) 'Flecha es una variable pública
textdiscorregida.Text = Val(frmaisla!textdiscorregida.Text)
textangdesvia.Text = Val(frmangulo!texttandesvia.Text)
textdisminleroy.Text = frmaisla!textdisminleroy.Text

```

```

textlonreal.Text = Val(frmaisla!textlonreal.Text) 'Longitud real de la cadena

```

```

textbil.Text = Val(frmaisla!cboBIL.Text)
textzc.Text = Val(alfa3)

```



```
If Val(alfa3) = 0 Then
  vol = Val(frmcalcable!cbovoltaje.Text)
  Select Case vol

    Case Is < 115
      textzc.Text = 550

    Case 115
      textzc.Text = 500

    Case 138
      textzc.Text = 450

    Case 161
      textzc.Text = 400

    Case 230
      textzc.Text = 367

    Case 345
      textzc.Text = 300

    Case 500
      textzc.Text = 285

    Case 765
      textzc.Text = 280

    Case Is >= 1200
      textzc.Text = 250

    Case Else
      textzc.Text = 300
  End Select
End If
End Sub
```

FIGURA FINAL

(frmvistazo.frm)

```
Private Sub btnverfrmangcri_Click()
Unload Me
frmblindaje.Show
End Sub
```

```
Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub
```

```
Private Sub btnvervistazo_Click()
Hide
frmcorona.Show
End Sub
Private Sub Form_Load()
Rem Cables
texttipo.Text = frmcalcable!texttipo.Text
textmcm.Text = frmcalcable!textmcm.Text
textncf.Text = frmcalcable!ncf.Text
```

```
textvoltaje.Text = frmcalcable!cbovoltaje.Text 'Voltaje de operación
```

```
Rem Aislamiento
textnumais.Text = frmaisla!textnumais.Text 'Núm. de aisladores
textlonreal.Text = frmaisla!textlonreal.Text 'Longitud real de la cadena
textncadenas.Text = frmangulo!textncadenas.Text 'Núm. de cadenas
textespesor.Text = frmaisla!textespesor.Text
textancho.Text = frmaisla!textancho.Text
```

```
Rem Distancias
textdisreal.Text = frmblindaje!textdisreal.Text 'Longitud horizontal
disho = Val(textdisreal.Text)
```

```
textdiscorregida.Text = frmaisla!textdiscorregida.Text 'Longitud vertical
disver = Val(textdiscorregida.Text)
```

```
texthmin.Text = frmaisla!texthmin.Text 'Altura mínima
hmin = Val(texthmin.Text)
```

```
disleroy = Val(frmaisla!textdisminleroy.Text) 'No se presenta como dato en esta
forma
```

hcg = Val(frdblindaje!texthg.Text) 'Altura del cable de guarda/suelo no se presenta como dato

hcgb = Val(frdblindaje!texthgc.Text) 'Altura del cable de guarda/brazo no se presenta como dato

Select Case tipestruct 'variable que se registra en el "CLICK" de cada imagen de frmeestruc

Case 1

imga.Picture = LoadPicture("\delta\Graficas\torre12.bmp")

hca = hmin + flecha 'fase más baja

lblhca1.Visible = True

lblhca2.Visible = False

texthca.Visible = True

textdisff.Visible = True

texthca.Text = hca

hcc = ((disleroy + disver) / 2) + hca 'fase intermedia

lblhcc.Visible = True

texthcc.Visible = True

texthcc.Text = hcc

hcb = hca + disver + disleroy ' fase más alta

lblhcb.Visible = True

texthcb.Visible = True

texthcb.Text = hcb

lblankoven.Visible = False

textankoven.Visible = False

dcab = disleroy + disver 'distancia de fase a fase vertical

lbdcab.Visible = True

textdcab.Visible = True

textdcab.Text = dcab

hcg = hcgb + disver + hcb 'Altura del cable de guarda/suelo

texthcg.Text = hcg

texthcgb.Text = hcgb

lbdisff1.Visible = True

lbdisff2.Visible = False

lbdca.Visible = False

disff = disho * 2 'Distancia de fase a fase horizontal

textdisff.Text = disff

```

lblisder.Visible = False
lblddb.Visible = False
textdisder.Visible = False

```

```

lblisextre.Visible = False
lbldac.Visible = False
textdisextre.Visible = False

```

Case 2

```

imga.Picture = LoadPicture("\delta\Graficas\torre8.bmp")

```

```

hca = hmin + flecha      'todas las fases tienen la misma altura
lblhca1.Visible = False
lblhca2.Visible = True
texthca.Visible = True
textdisff.Visible = True
texthca.Text = hca

```

```

lblhcc.Visible = False
texthcc.Visible = False

```

```

lblhcb.Visible = False
texthcb.Visible = False

```

```

blanchoven.Visible = True
textanchoven.Visible = True
anchoven = 2 * disho      'ancho de la ventana
textanchoven.Text = anchoven

```

```

lbldcab.Visible = False
textdcab.Visible = False

```

```

hcg = hcgb + disver + hca      'Altura del cable de guarda/suelo
texthcg.Text = hcg
texthcgb.Text = hcgb

```

```

lbldisff1.Visible = False
lbldisff2.Visible = True
lbldca.Visible = False
disff = disho * 2      'Distancia de fase a fase lado izquierdo
textdisff.Text = disff

```

```

lblisder.Visible = True
lblddb.Visible = False
textdisder.Visible = True
disder = disho * 2      'Distancia de fase a fase lado derecho

```

textdisder.Text = disder

lbldisextre.Visible = True

lbldac.Visible = False

textdisextre.Visible = True

disextre = 4 * disho 'Distancia entre fases extremas

textdisextre.Text = disextre

Case 3

imga.Picture = LoadPicture("\delta\Graficas\torre13.bmp")

hca = hmin + flecha 'fase más baja

hcb = hca + disver + disleroy 'fase intermedia

hcc = hcb + disver + disleroy 'fase más alta

hcg = hcgb + disver + hcc 'Altura del cable de guarda/suelo

dcab = disleroy + disver 'distancia de fase a fase vertical

texthcg.Text = hcg 'Altura del cable de guarda/suelo

texthcgb.Text = hcgb 'Altura del cable de guarda/brazo

lbhca1.Visible = True

lbhca2.Visible = False

texthca.Visible = True

textdisff.Visible = True

texthca.Text = hca 'fase más baja

lbhcc.Visible = True

texthcc.Visible = True

texthcc.Text = hcb 'fase más alta

lbhcb.Visible = True

texthcb.Visible = True

texthcb.Text = hcc 'fase intermedia

blanchoven.Visible = False

textanchoven.Visible = False

lbldcab.Visible = True

textdcab.Visible = True

textdcab.Text = dcab 'distancia de fase a fase vertical

lbldisff1.Visible = False

lbldisff2.Visible = False

lbldca.Visible = True

disff = disho * 2 'distancia de fase a fase horizontal

textdisff.Text = disff

lbldisder.Visible = False

```

lblddb.Visible = True
textdisder.Visible = True
disder = disho * 2
textdisder.Text = disder

```

```

lbldisextre.Visible = False
lbldac.Visible = True
textdisextre.Visible = True
disextre = 2 * disho
textdisextre.Text = disextre

```

Case 4

```

imga.Picture = LoadPicture("\delta\Graficas\torre5.bmp")
hca = hmin + flecha      'fase más baja
hcb = hca + disver + disleroy 'fase intermedia
hcc = hcb + disver + disleroy 'fase más alta
hcg = hcgb + disver + hcc  'Altura del cable de guarda/suelo
dcab = disleroy + disver  'distancia de fase a fase vertical

```

```

texthcg.Text = hcg
texthcgb.Text = hcgb

```

```

lblhca1.Visible = True
lblhca2.Visible = False
texthca.Visible = True
textdisff.Visible = True
texthca.Text = hca

```

```

lblhcc.Visible = True
texthcc.Visible = True
texthcc.Text = hcb

```

```

lblhcb.Visible = True
texthcb.Visible = True
texthcb.Text = hcc

```

```

blanchoven.Visible = False
textanchoven.Visible = False

```

```

lbldcab.Visible = True
textdcab.Visible = True
textdcab.Text = dcab

```

```

lbldisff1.Visible = False
lbldisff2.Visible = False
lbldca.Visible = True
disff = disho * 2

```

textdisff.Text = disff

lbldisder.Visible = False
 lbldb.b.Visible = True
 textdisder.Visible = True
 disder = disho * 2
 textdisder.Text = disder

lbldisextre.Visible = False
 lbldac.Visible = True
 textdisextre.Visible = True
 disextre = 2 * disho
 textdisextre.Text = disextre

Case 5

imga.Picture = LoadPicture("\delta\Graficas\marco2.bmp")

hca = hmin + flecha 'todas las fases tienen la misma altura
 lbhca1.Visible = False
 lbhca2.Visible = True
 texthca.Visible = True
 textdisff.Visible = True
 texthca.Text = hca

lbhcc.Visible = False
 texthcc.Visible = False

lbhcb.Visible = False
 texthcb.Visible = False

blanchoven.Visible = True
 textanchoven.Visible = True
 anchoven = 2 * disho 'ancho de la ventana
 textanchoven.Text = anchoven

lbldcab.Visible = False
 textdcab.Visible = False

hcg = hcgb + disver + hca 'Altura del cable de guarda/suelo
 texthcg.Text = hcg
 texthcgb.Text = hcgb

lbldisff1.Visible = False
 lbldisff2.Visible = True
 lbldca.Visible = False
 disff = disho * 2
 textdisff.Text = disff

```
lbldisder.Visible = True
lblddb.Visible = False
textdisder.Visible = True
disder = disho * 2
textdisder.Text = disder

lbldisextre.Visible = True
lbldac.Visible = False
textdisextre.Visible = True
disextre = 4 * disho
textdisextre.Text = disextre
End Select
End Sub
```


PERDIDAS POR EFECTO CORONA

(frmcorona.frm)

```
Private Sub btncalcorona_Click()
```

```
Static pi
```

```
pi = 3.14159265358979
```

```
Select Case tipestruct 'variable que se registra en el "CLICK" de cada imagen
de frmestruc
```

```
Case 1
```

```
dab = Val(textdab.Text)
```

```
dbc = Val(textdbc.Text)
```

```
dca = Val(textdca.Text)
```

```
dmg = (dab * dbc * dca) ^ (1 / 3)
```

```
Case 2, 5
```

```
dab = textdab.Text
```

```
dbc = textdbc.Text
```

```
dca = textdca.Text
```

```
dmg = (dab * dbc * dca) ^ (1 / 3)
```

```
Case 3, 4
```

```
dac2 = Val(textdab.Text)
```

```
dbb2 = Val(textdbc.Text)
```

```
dca2 = Val(textdca.Text)
```

```
dhab = Val(texthab.Text)
```

```
dhbc = Val(texthbc.Text)
```

```
dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
```

```
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
```

```
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5
```

```
dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
```

```
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
```

```
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5
```

```
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
```

```
12)
```

```
End Select
```

```
textdmg.Text = dmg
```

```

Rem ***** Radio equivalente *****
'Solo para un ckto
radio = Val(textradio.Text) / 10 'Datos radio en cm
dis = Val(textdis.Text) 'Datos en cm
ncfa = Val(ncf.Text) 'Datos # de c/fase

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line3
Else
    dis = Val(textdis.Text) 'Distancia entre subconductores en cm
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
    GoTo Line3
End If
Line3:

Rem ***** Correcciones por altura y temperatura *****
altura = Val(textaltura.Text) 'Datos
tempe = Val(texttempe.Text) 'Datos

presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

Rem ***** Cálculo de las pérdidas por corona *****

frec = Val(cbofre.Text)
coefrugo = Val(cbocoefrugo.Text)
vmax = Val(textvolmax.Text)
vnom = Val(textvoltaje.Text)
dmga = Val(textdmg.Text)

If dmga = 0 Then
    pelco = MsgBox("La distancia media geométrica (DMG) no puede ser cero",
vbCritical, "DMG no definida")
    textdmg.SetFocus
    GoTo fin
End If

If chkts.Value = Checked Then ' *** Para tiempo seco ***
    lblts.Enabled = True
    lblvcts.Enabled = True
    lblpcts.Enabled = True
    textvcts.Enabled = True

```

```

textpcts.Enabled = True
shpts.BorderStyle = 1

coefmetets = cbots.Text

vcts = 84 * fcc * coefrugo * coefmetets * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
'Logn(x) = Log(x) / Log(n)
pcts = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcts) ^ 2
* 10 ^ -5

If vnom < vcts Then
    lblmtsse.Visible = True
    lblmtsce.Visible = False

    textvcts.Text = vcts
    textpcts.Text = 0
Else
    lblmtsse.Visible = False
    lblmtsce.Visible = True

    textvcts.Text = vcts
    textpcts.Text = pcts
End If

If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lblmthse.Visible = False
    lblmthce.Visible = False
    lblvcth.Enabled = False
    lblpcth.Enabled = False
    textvcth.Text = 0
    textpcth.Text = 0
    textvcth.Enabled = False
    textpcth.Enabled = False
    shpth.BorderStyle = 3
End If
End If

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***
    lblth.Enabled = True
    lblvcth.Enabled = True
    lblpcth.Enabled = True
    textvcth.Enabled = True
    textpcth.Enabled = True
    shpth.BorderStyle = 1

```

```

coefmeteth = cboth.Text

vcth = 84 * fcc * coefrugo * coefmeteth * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
'Logn(x) = Log(x) / Log(n)
pcth = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcth) ^ 2
* 10 ^ -5

If vnom < vcth Then
    lblmthse.Visible = True
    lblmthce.Visible = False

    textvcth.Text = vcth
    textpcth.Text = 0
Else
    lblmthse.Visible = False
    lblmthce.Visible = True

    textvcth.Text = vcth
    textpcth.Text = pcth
End If

If chkts.Value = Unchecked Then
    lblts.Enabled = False
    lblmtsse.Visible = False
    lblmtsce.Visible = False
    lblvcts.Enabled = False
    lblpcts.Enabled = False
    textvcts.Text = 0
    textpcts.Text = 0
    textvcts.Enabled = False
    textpcts.Enabled = False
    shpts.BorderStyle = 3
End If
End If

If chkts.Value = Unchecked And chkth.Value = Unchecked Then
    msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos
uno: " & Chr(10) & Chr(13)
    msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "
    MsgBox msjcoro
    chkts.SetFocus
End If
fin:
End Sub

Private Sub btncorreco_Click()

```

```

rastreo = "corona"
frmcorrecorona.Show vbModal
End Sub

```

```

Private Sub btnmsjinf_Click()
msj = "El coeficiente de rugosidad del conductor viene dado por: " & Chr(10) &
Chr(13)
msj = msj & "Para superficies lisas m = 1" & Chr(10) & Chr(13)
msj = msj & "Para cables oxidados o rugosos: m = 0.93 a 0.98" & Chr(10) &
Chr(13)
msj = msj & "Para cables en general: m = 0.83 a 0.87"
MsgBox msj
End Sub

```

```

Private Sub btnverdista_Click()
If uckto.Value = True Then
dat1ckto!optasime.Value = True 'Obliga a optasime de dat1ckto a poner datos
asimétricos
dat1ckto.Show vbModal
Else
dat2ckto.Show vbModal
End If
End Sub

```

```

Private Sub btnverfrmangcri_Click()
Unload Me
frmvistazo.Show
End Sub

```

```

Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub

```

```

Private Sub btnverruido_Click()
Hide
frmruido.Show
End Sub

```

```

Private Sub chkth_Click()
Rem *** Para tiempo húmedo ***
lblth.Enabled = True
lblvcth.Enabled = True
lblpcth.Enabled = True
textvcth.Enabled = True
textpcth.Enabled = True
shpth.BorderStyle = 1

```

```
If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lblmthse.Visible = False
    lblmthce.Visible = False
    lblvcth.Enabled = False
    lblpcth.Enabled = False
    textvcth.Text = 0
    textpcth.Text = 0
    textvcth.Enabled = False
    textpcth.Enabled = False
    shpth.BorderStyle = 3
End If
End Sub
```

```
Private Sub chkts_Click()
    lblts.Enabled = True
    lblvcts.Enabled = True
    lblpcts.Enabled = True
    textvcts.Enabled = True
    textpcts.Enabled = True
    shpts.BorderStyle = 1
```

```
If chkts.Value = Unchecked Then
    lblts.Enabled = False
    lblmtsse.Visible = False
    lblmtsce.Visible = False
    lblvcts.Enabled = False
    lblpcts.Enabled = False
    textvcts.Text = 0
    textpcts.Text = 0
    textvcts.Enabled = False
    textpcts.Enabled = False
    shpts.BorderStyle = 3
End If
End Sub
```

```
Private Sub dckto_Click()
    If dckto.Value = True Then
        cktos = 2
        lblckto.Enabled = True
        lbluckto.Enabled = False
    Else
        cktos = 1
        lbluckto.Enabled = True
        lblckto.Enabled = False
    End If
End Sub
```

End Sub

Private Sub Form_Load()

If frmcalcable!uckto.Value = True Then

 uckto.Value = True

Else

 dckto.Value = True

End If

If frmcalcable!dckto.Value = True Then

 dckto.Value = True

Else

 uckto.Value = True

End If

texttipo.Text = frmcalcable!texttipo.Text

textmcm.Text = frmcalcable!textmcm.Text

textradio.Text = frmcalcable!textradio.Text

textvoltaje.Text = Val(frmcalcable!cbovoltaje.Text)

textvolmax.Text = Val(frmmaisla!textvolmax.Text)

If textvolmax.Text = 0 Then

 voltajenom = Val(textvoltaje.Text)

 Select Case voltajenom

 Case 115

 textvolmax.Text = 123

 Case 138

 textvolmax.Text = 145

 Case 161

 textvolmax.Text = 170

 Case 230

 textvolmax.Text = 245

 Case 287

 textvolmax.Text = 300

 Case 345

 textvolmax.Text = 362

 Case 400

 textvolmax.Text = 420

 Case 500

 textvolmax.Text = 525

 Case 765

 textvolmax.Text = 765

 Case Else

 voltajenom = textvoltaje.Text

 textvolmax.Text = 0

 End Select

End If

```

Rem *****
numcon = Val(frmcalcable!ncf.Text)      '# de conductores
textaltura.Text = frmaisla!textaltura.Text 'MSNM
texttempe.Text = frmaisla!texttempe.Text 'Temperatura

Rem *****
Select Case tipestruct 'variable que se registra en el "CLICK" de cada imagen
de frmestruc
  Case 1
    dcab = frmvistazo!textdcab.Text 'distancia de fase a fase vertical
    disff = frmvistazo!textdisff.Text 'Distancia de fase a fase horizontal
    dab = dcab
    dbc = Sqr((dcab / 2) ^ 2 + disff ^ 2)
    dca = dbc

    textdab.Text = dab
    textdbc.Text = dbc
    textdca.Text = dca

  Case 2, 5
    disff = frmvistazo!textdisff.Text 'Distancia de fase a fase lado izquierdo
    disder = frmvistazo!textdisder.Text 'Distancia de fase a fase lado
derecho
    disextre = frmvistazo!textdisextre.Text 'Distancia entre fases extremas
    dab = disff
    dbc = disder
    dca = disextre

    textdab.Text = dab
    textdbc.Text = dbc
    textdca.Text = dca

  Case 3, 4
    dcab = frmvistazo!textdcab.Text 'distancia de fase a fase vertical
    disff = frmvistazo!textdisff.Text 'distancia de fase a fase horizontal
    dac2 = disff
    dbb2 = disff
    dca2 = disff
    dhab = dcab
    dhbc = dcab

    textdab.Text = dca2
    textdbc.Text = dbb2
    textdca.Text = dca2

    texthab.Text = dhab

```



```

        texthbc.Text = dhbc
    End Select
    ncf.Text = numcon
End Sub

Private Sub ncf_Click()
    ncfa = Int(ncf.Text)
    ncf.Text = ncfa

    Select Case ncfa
        Case Is = 0
            def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
            ncf.Text = 1

        Case Is < 1
            def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
            If def = vbYes Then
                ncf.Text = Abs(Int(ncf.Text))
                frec.SetFocus
            Else
                GoTo Line1
            End If
        End Select
    End Sub

Private Sub ncf_Change()
    ncfa = Int(Val(ncf.Text))
    ncf.Text = ncfa

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If

    Select Case ncfa

        Case Is >= 6
            mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
            GoTo Line2

        Case Is = 0
            def = MsgBox("Por lo menos debe de existir un conductor ", 48, "Definición
de subconductores ")

```

```

Line1:
    ncf.Text = 1

    Case Is < 1
        def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
        If def = vbYes Then
            ncf.Text = Abs(Int(ncf.Text))
Line2:
            Else
                GoTo Line1
            End If
        End Select

    If ncfa >= 2 Then
        IblDis.Enabled = True
        textDis.Enabled = True
        textDis.Text = 40
        'textDis.SetFocus
    Else
        IblDis.Enabled = False
        textDis.Enabled = False
    End If
    Rem *****
    Rem ***** CALCULO DE PREDIDAS POR CORONA *****

    Static pi
    pi = 3.14159265358979

    Select Case tipestruct 'variable que se registra en el "CLICK" de cada imagen
de frmestruc
        Case 1
            dab = Val(textdab.Text)
            dbc = Val(textdbc.Text)
            dca = Val(textdca.Text)

            dmg = (dab * dbc * dca) ^ (1 / 3)

        Case 2, 5
            dab = textdab.Text
            dbc = textdbc.Text
            dca = textdca.Text

            dmg = (dab * dbc * dca) ^ (1 / 3)

        Case 3, 4
            dac2 = Val(textdab.Text)

```

```

dbb2 = Val(textdbc.Text)
dca2 = Val(textdca.Text)

dhab = Val(texthab.Text)
dhbc = Val(texthbc.Text)

dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5

dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)
End Select

textdmg.Text = dmg

Rem ***** Radio equivalente *****
'Solo para un ckto
radio = Val(textradio.Text) / 10 'Datos radio en cm
dis = Val(textdis.Text) 'Datos en cm
ncfa = Val(ncf.Text) 'Datos # de c/fase

If textdis.Text = "" Then
  rmgg = radio
  requi = 0
  GoTo Line3
Else
  dis = Val(textdis.Text) 'Distancia entre subconductores en cm
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
  GoTo Line3
End If
Line3:

Rem ***** Correcciones por altura y temperatura *****
altura = Val(textaltura.Text) 'Datos
tempe = Val(texttempe.Text) 'Datos

presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

Rem ***** Cálculo de las pérdidas por corona *****

```

```

frec = Val(cbofre.Text)
coefrugo = Val(cbocoefrugo.Text)
vmax = Val(textvolmax.Text)
vnom = Val(textvoltaje.Text)
dmga = Val(textdmg.Text)

If chkts.Value = Checked Then ' *** Para tiempo seco ***
    lblts.Enabled = True
    lblvcts.Enabled = True
    lblpcts.Enabled = True
    textvcts.Enabled = True
    textpcts.Enabled = True
    shpts.BorderStyle = 1

    coefmetets = cbots.Text

    vcts = 84 * fcc * coefrugo * coefmetets * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
    'Logn(x) = Log(x) / Log(n)
    pcts = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcts) ^ 2
* 10 ^ -5

    If vnom < vcts Then
        lblmtsse.Visible = True
        lblmtsce.Visible = False

        textvcts.Text = vcts
        textpcts.Text = 0
    Else
        lblmtsse.Visible = False
        lblmtsce.Visible = True

        textvcts.Text = vcts
        textpcts.Text = pcts
    End If

    If chkth.Value = Unchecked Then
        lblth.Enabled = False
        lblmthse.Visible = False
        lblmthce.Visible = False
        lblvcth.Enabled = False
        lblpcth.Enabled = False
        textvcth.Text = 0
        textpcth.Text = 0
        textvcth.Enabled = False
        textpcth.Enabled = False

```

```

        shpth.BorderStyle = 3
    End If
End If

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***
    lblth.Enabled = True
    lblvcth.Enabled = True
    lblpcth.Enabled = True
    textvcth.Enabled = True
    textpcth.Enabled = True
    shpth.BorderStyle = 1

    coefmeteth = cboth.Text

    vcth = 84 * fcc * coefrugo * coefmeteth * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
    'Logn(x) = Log(x) / Log(n)
    pcth = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcth) ^ 2
* 10 ^ -5

    If vnom < vcth Then
        lblmthse.Visible = True
        lblmthce.Visible = False

        textvcth.Text = vcth
        textpcth.Text = 0
    Else
        lblmthse.Visible = False
        lblmthce.Visible = True

        textvcth.Text = vcth
        textpcth.Text = pcth
    End If

    If chkts.Value = Unchecked Then
        lblts.Enabled = False
        lblmtsse.Visible = False
        lblmtsce.Visible = False
        lblvcts.Enabled = False
        lblpcts.Enabled = False
        textvcts.Text = 0
        textpcts.Text = 0
        textvcts.Enabled = False
        textpcts.Enabled = False
        shpts.BorderStyle = 3
    End If
End If

```

```

If chkts.Value = Unchecked And chkth.Value = Unchecked Then
    msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos
uno: " & Chr(10) & Chr(13)
    msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "
    MsgBox msjcoro
    chkts.SetFocus
End If
End Sub

```

```

Private Sub spinnu_SpinDown()
Dim dblrate
dblrate = Val(ncf.Text)
If dblrate > 1 Then
    ncf.Text = dblrate - 1
Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub

```

```

Private Sub spinnu_SpinUp()
ncf.Text = (Val(ncf.Text)) + 1
If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh
End Sub

```

```

Private Sub textdis_Change()
Rem *****
Rem ***** CALCULO DE PREDIDAS POR CORONA *****

```

```

Static pi
pi = 3.14159265358979

```

```

Select Case tipestruct 'variable que se registra en el "CLICK" de cada imagen
de frmestruc
Case 1
    dab = Val(textdab.Text)
    dbc = Val(textdbc.Text)
    dca = Val(textdca.Text)

    dmg = (dab * dbc * dca) ^ (1 / 3)

```

Case 2, 5

```
dab = textdab.Text
dbc = textdbc.Text
dca = textdca.Text
```

$$dmg = (dab * dbc * dca)^{(1/3)}$$

Case 3, 4

```
dac2 = Val(textdab.Text)
dbb2 = Val(textdbc.Text)
dca2 = Val(textdca.Text)
```

```
dhab = Val(texthab.Text)
dhbc = Val(texthbc.Text)
```

$$dab = (\text{Abs}(0.25 * (dbb2 - dac2)^2 + dhab^2))^{0.5}$$

$$dbc = (\text{Abs}(0.25 * (dca2 - dbb2)^2 + dhbc^2))^{0.5}$$

$$dac = (\text{Abs}(0.25 * (dca2 - dac2)^2 + (dhab + dhbc)^2))^{0.5}$$

$$dab2 = (\text{Abs}(0.25 * (dbb2 + dac2)^2 + dhab^2))^{0.5}$$

$$dba2 = (\text{Abs}(0.25 * (dca2 + dbb2)^2 + dhbc^2))^{0.5}$$

$$daa2 = (\text{Abs}(0.25 * (dca2 + dac2)^2 + (dhab + dhbc)^2))^{0.5}$$

$$dmg = (dab^2 * dac^2 * dab2^2 * dba2^2 * dbc^2 * dac2 * dca2)^{(1/12)}$$

12)

End Select

```
textdmg.Text = dmg
```

Rem ***** Radio equivalente *****

'Solo para un ckto

```
radio = Val(textradio.Text) / 10 'Datos radio en cm
```

```
dis = Val(textdis.Text) 'Datos en cm
```

```
ncfa = Val(ncf.Text) 'Datos # de c/fase
```

```
If textdis.Text = "" Then
```

```
  rmgg = radio
```

```
  requi = 0
```

```
  GoTo Line3
```

```
Else
```

```
dis = Val(textdis.Text) 'Distancia entre subconductores en cm
```

```
If dis = 0 Then
```

```
  rmgg = radio
```

```
  GoTo Line3
```

```
End If
```

```
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
```

```

    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line3
  End If

```

Line3:

```

  Rem ***** Correcciones por altura y temperatura *****
  altura = Val(textaltura.Text) 'Datos
  tempe = Val(texttempe.Text) 'Datos

  presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
  fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

  Rem ***** Cálculo de las pérdidas por corona *****

  frec = Val(cbofre.Text)
  coefrugo = Val(cbocoefrugo.Text)
  vmax = Val(textvolmax.Text)
  vnom = Val(textvoltaje.Text)
  dmga = Val(textdmg.Text)

  If chkts.Value = Checked Then ' *** Para tiempo seco ***
    lblts.Enabled = True
    lblvcts.Enabled = True
    lblpcts.Enabled = True
    textvcts.Enabled = True
    textpcts.Enabled = True
    shpts.BorderStyle = 1

    coefmetets = cbots.Text

    vcts = 84 * fcc * coefrugo * coefmetets * rmgg * (Log(dmga / (rmgg / 100)) /
  Log(10)) 'Cálculo de voltaje crítico
    'Logn(x) = Log(x) / Log(n)
    pcts = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcts) ^ 2
  * 10 ^ -5

    If vnom < vcts Then
      lblmtsse.Visible = True
      lblmtsce.Visible = False

      textvcts.Text = vcts
      textpcts.Text = 0
    Else
      lblmtsse.Visible = False
      lblmtsce.Visible = True
    End If
  End If

```



```

    textvcts.Text = vcts
    textpcts.Text = pcts
End If

```

```

If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lblmthse.Visible = False
    lblmthce.Visible = False
    lblvcth.Enabled = False
    lblpcth.Enabled = False
    textvcth.Text = 0
    textpcth.Text = 0
    textvcth.Enabled = False
    textpcth.Enabled = False
    shpth.BorderStyle = 3
End If
End If

```

```

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***
    lblth.Enabled = True
    lblvcth.Enabled = True
    lblpcth.Enabled = True
    textvcth.Enabled = True
    textpcth.Enabled = True
    shpth.BorderStyle = 1

    coefmeteth = cboth.Text

    vcth = 84 * fcc * coefrugo * coefmeteth * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
    'Logn(x) = Log(x) / Log(n)
    pcth = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcth) ^ 2
* 10 ^ -5

    If vnom < vcth Then
        lblmthse.Visible = True
        lblmthce.Visible = False

        textvcth.Text = vcth
        textpcth.Text = 0
    Else
        lblmthse.Visible = False
        lblmthce.Visible = True

        textvcth.Text = vcth
        textpcth.Text = pcth
    End If

```

```

If chkts.Value = Unchecked Then
  lblts.Enabled = False
  lblmtsse.Visible = False
  lblmtsce.Visible = False
  lblvcts.Enabled = False
  lblpcts.Enabled = False
  textvcts.Text = 0
  textpcts.Text = 0
  textvcts.Enabled = False
  textpcts.Enabled = False
  shpts.BorderStyle = 3
End If
End If

If chkts.Value = Unchecked And chkth.Value = Unchecked Then
  msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos
uno: " & Chr(10) & Chr(13)
  msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "
  MsgBox msjcoro
  chkts.SetFocus
End If

End Sub

Private Sub textdmg_Change()
Static pi
pi = 3.14159265358979

Rem *****
Rem ***** CALCULO DE PREDIDAS POR CORONA *****

dmg = Val(textdmg.Text)

Rem ***** Radio equivalente *****
'Solo para un ckto
radio = Val(textradio.Text) / 10 'Datos radio en cm
dis = Val(textdis.Text) 'Datos en cm
ncfa = Val(ncf.Text) 'Datos # de c/fase

If textdis.Text = "" Then
  rmgg = radio
  requi = 0
  GoTo Line3
Else
  dis = Val(textdis.Text) 'Distancia entre subconductores en cm
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente

```

```

    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase

```

```

    GoTo Line3

```

```

End If

```

```

Line3:

```

```

Rem ***** Correcciones por altura y temperatura *****

```

```

altura = Val(textaltura.Text) 'Datos

```

```

tempe = Val(texttempe.Text) 'Datos

```

```

presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b

```

```

fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

```

```

Rem ***** Cálculo de las pérdidas por corona *****

```

```

frec = Val(cbofre.Text)

```

```

coefrugo = Val(cbocoefrugo.Text)

```

```

vmax = Val(textvolmax.Text)

```

```

vnom = Val(textvoltaje.Text)

```

```

dmga = Val(textdmg.Text)

```

```

If dmga = 0 Then

```

```

    pelco = MsgBox("La distancia media geométrica (DMG) no puede ser cero",
vbCritical, "DMG no definida")

```

```

    textdmg.SetFocus

```

```

    GoTo fin

```

```

End If

```

```

If chkts.Value = Checked Then ' *** Para tiempo seco ***

```

```

    lblts.Enabled = True

```

```

    lblvcts.Enabled = True

```

```

    lblpcts.Enabled = True

```

```

    textvcts.Enabled = True

```

```

    textpcts.Enabled = True

```

```

    shpts.BorderStyle = 1

```

```

    coefmetets = cbots.Text

```

```

    vcts = 84 * fcc * coefrugo * coefmetets * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico

```

```

    'Logn(x) = Log(x) / Log(n)

```

```

    pcts = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcts) ^ 2
* 10 ^ -5

```

```

If vnom < vcts Then

```

```

    lblmtsse.Visible = True

```

```

    lblmtsce.Visible = False

```

```

textvcts.Text = vcts
textpcts.Text = 0
Else
lblmtsse.Visible = False
lblmtsce.Visible = True

textvcts.Text = vcts
textpcts.Text = pcts
End If

If chkth.Value = Unchecked Then
lblth.Enabled = False
lblmthse.Visible = False
lblmthce.Visible = False
lblvcth.Enabled = False
lblpcth.Enabled = False
textvcth.Text = 0
textpcth.Text = 0
textvcth.Enabled = False
textpcth.Enabled = False
shpth.BorderStyle = 3
End If
End If

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***
lblth.Enabled = True
lblvcth.Enabled = True
lblpcth.Enabled = True
textvcth.Enabled = True
textpcth.Enabled = True
shpth.BorderStyle = 1

coefmeteth = cboth.Text

vcth = 84 * fcc * coefrugo * coefmeteth * rmgg * (Log(dmga / (rmgg / 100)) /
Log(10)) 'Cálculo de voltaje crítico
'Logn(x) = Log(x) / Log(n)
pcth = (80.33 / fcc) * (frec + 25) * Sqr((rmgg / 100) / dmga) * (vmax - vcth) ^ 2
* 10 ^ -5

If vnom < vcth Then
lblmthse.Visible = True
lblmthce.Visible = False

textvcth.Text = vcth
textpcth.Text = 0
Else

```

```

    lblmthse.Visible = False
    lblmthce.Visible = True

    textvcth.Text = vcth
    textpcth.Text = pcth
End If

If chkts.Value = Unchecked Then
    lblts.Enabled = False
    lblmtsse.Visible = False
    lblmtsce.Visible = False
    lblvcts.Enabled = False
    lblpcts.Enabled = False
    textvcts.Text = 0
    textpcts.Text = 0
    textvcts.Enabled = False
    textpcts.Enabled = False
    shpts.BorderStyle = 3
End If
End If

If chkts.Value = Unchecked And chkth.Value = Unchecked Then
    msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos
uno: " & Chr(10) & Chr(13)
    msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "
    MsgBox msjcoro
    chkts.SetFocus
End If
fin:
End Sub

Private Sub uckto_Click()
If uckto.Value = True Then
    cktos = 1
    lbluckto.Enabled = True
    lbldckto.Enabled = False
Else
    cktos = 2
    lbldckto.Enabled = True
    lbluckto.Enabled = False
End If

End Sub

```

RUIDO AUDIBLE Y EFECTO CORONA

(frmruído.frm)

```

Private Sub btncalruído_Click()
Static pi
pi = 3.14159265358979

Rem ***** Radio equivalente *****
'Solo para un ckto
radio = Val(textradio.Text) / 10 'Datos radio en cm
dis = Val(textdis.Text) 'Datos en cm
ncfa = Val(ncf.Text) 'Datos # de c/fase

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line3
Else
    dis = Val(textdis.Text) 'Distancia entre subconductores en cm
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
    GoTo Line3
End If
Line3:

Rem ***** Correcciones por altura y temperatura *****
altura = Val(textaltura.Text) 'Datos
tempe = Val(texttempe.Text) 'Datos

presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

hfase = Val(texthfase.Text) 'Altura de la fase más baja
If hfase = 0 Then
    pelru = MsgBox("La altura de la fase más baja no puede ser cero", vbCritical,
"Altura de conductor no definido (DELTA)")
    texthfase.SetFocus
    GoTo fin
End If

Rem Otros datos
diametro = Val(textdiame.Text) / 10 'Diámetro en cm
freri = Val(cbofreri.Text)

```

```

coefrugo = Val(cbocoefrugo.Text)

If chkts.Value = Checked Then ' *** Para tiempo seco ***
  lblts.Enabled = True
  lblrats.Enabled = True
  lblrits.Enabled = True
  textrats.Enabled = True
  textrits.Enabled = True
  shpts.BorderStyle = 1

  coefmetets = cbots.Text

  gcs = 30 * (fcc) ^ (2 / 3) * coefrugo * coefmetets * (1 - 0.07 * radio) * (1 - (ncfa
- 1) * radio / rmgg) 'Cálculo del gradiente crítico
  Select Case ncfa
    Case 1
      ANo = 75.2
      cte1 = 7.5
      cte2 = 0
    Case 2
      ANo = 75.2
      cte1 = 2.6
      cte2 = 0
    Case Is >= 3
      ANo = 67.9
      cte1 = 0
      cte2 = 22.9 * (ncfa - 1) * diametro / (rmgg * 2)
  End Select

  Rem Cálculo de la distancia radial
  dx = Abs(Val(textx.Text))
  dy = Abs(Val(texty.Text))
  dr = Sqr((hfase - dy) ^ 2 + (dx) ^ 2)

  Rem Cálculo del nivel de ruido audible; Logn(x) = Log(x) / Log(n)
  AN5ts = (-665 / gcs) + 20 * Log(ncfa) / Log(10) + 44 * Log(diametro) / Log(10)
- 10 * Log(dr) / Log(10) - 0.02 * dr + ANo + cte1 + cte2

  Rem Cálculo del nivel de radio interferencia
  rits = -150.4 + 120 * Log(gcs) / Log(10) + 40 * Log(diametro * 10) / Log(10) +
20 * Log(hfase / (dr) ^ 2) / Log(10) + 10 * (1 - (Log(10 * freri) / Log(10)) ^ 2)
  textrats.Text = AN5ts
  textrits.Text = rits

  If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lblrath.Enabled = False

```

```

lbrith.Enabled = False
textrath.Enabled = False
textrith.Enabled = False
shpth.BorderStyle = 3
textrath.Text = 0
textrith.Text = 0
End If
End If

```

```

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***

```

```

  lblth.Enabled = True
  lblrath.Enabled = True
  lbrith.Enabled = True
  textrath.Enabled = True
  textrith.Enabled = True
  shpth.BorderStyle = 1

```

```

  coefmeteth = cboth.Text

```

```

  gch = 30 * (fcc) ^ (2 / 3) * coefrugo * coefmeteth * (1 - 0.07 * radio) * (1 - (ncfa
- 1) * radio / rmgg) 'Cálculo del gradiente crítico

```

```

  Select Case ncfa

```

```

    Case 1

```

```

      ANo = 75.2

```

```

      cte1 = 7.5

```

```

      cte2 = 0

```

```

    Case 2

```

```

      ANo = 75.2

```

```

      cte1 = 2.6

```

```

      cte2 = 0

```

```

    Case Is >= 3

```

```

      ANo = 67.9

```

```

      cte1 = 0

```

```

      cte2 = 22.9 * (ncfa - 1) * diametro / (rmgg * 2)

```

```

  End Select

```

```

  Rem Cálculo de la distancia radial

```

```

  dx = Abs(Val(textx.Text))

```

```

  dy = Abs(Val(texty.Text))

```

```

  dr = Sqr((hfase - dy) ^ 2 + (dx) ^ 2)

```

```

  Rem Cálculo del nivel de ruido audible 'Logn(x) = Log(x) / Log(n)

```

```

  AN5th = (-665 / gch) + 20 * Log(ncfa) / Log(10) + 44 * Log(diametro) / Log(10)
- 10 * Log(dr) / Log(10) - 0.02 * dr + ANo + cte1 + cte2

```

```

  Rem Cálculo del nivel de radio interferencia

```



```

    rith = -150.4 + 120 * Log(gch) / Log(10) + 40 * Log(diametro * 10) / Log(10) +
20 * Log(hfase / (dr) ^ 2) / Log(10) + 10 * (1 - (Log(10 * freri) / Log(10)) ^ 2)
    textrath.Text = AN5th
    textrith.Text = rith

```

```

If chkts.Value = Unchecked Then
    lblts.Enabled = False
    lblrats.Enabled = False
    lblrits.Enabled = False
    textrats.Enabled = False
    textrits.Enabled = False
    shpts.BorderStyle = 3
    textrats.Text = 0
    textrits.Text = 0
End If
End If

```

```

If chkts.Value = Unchecked And chkth.Value = Unchecked Then
    msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos
uno: " & Chr(10) & Chr(13)
    msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "
    MsgBox msjcoro
    chkts.SetFocus
End If

```

```

fin:
End Sub
Private Sub btncorreoro_Click()
    rastreo = "ruido"
    frmcorreorona.Show vbModal
End Sub

```

```

Private Sub btnmsjinf_Click()
    msj = "El coeficiente de rugosidad del conductor viene dado por: " & Chr(10) &
Chr(13)
    msj = msj & "Para superficies lisas m = 1" & Chr(10) & Chr(13)
    msj = msj & "Para cables oxidados o rugosos: m = 0.93 a 0.98" & Chr(10) &
Chr(13)
    msj = msj & "Para cables en general: m = 0.83 a 0.87"
    MsgBox msj
End Sub

```

```

Private Sub btnvercorona_Click()
    Hide
    frmcorona.Show
End Sub

```

```

Private Sub btnverprincipio_Click()

```

```
Unload Me
Principio.Show
End Sub
```

```
Private Sub chkth_Click()
Rem *** Para tiempo húmedo ***
lblth.Enabled = True
lbrath.Enabled = True
lbrith.Enabled = True
textrath.Enabled = True
textrith.Enabled = True
shpth.BorderStyle = 1
```

```
If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lbrath.Enabled = False
    lbrith.Enabled = False
    textrath.Enabled = False
    textrith.Enabled = False
    shpth.BorderStyle = 3
    textrath.Text = 0
    textrith.Text = 0
End If
End Sub
```

```
Private Sub chkts_Click()
Rem *** Para tiempo seco ***
lblts.Enabled = True
lbrats.Enabled = True
lbrits.Enabled = True
textrats.Enabled = True
textrits.Enabled = True
shpts.BorderStyle = 1
```

```
If chkts.Value = Unchecked Then
    lblts.Enabled = False
    lbrats.Enabled = False
    lbrits.Enabled = False
    textrats.Enabled = False
    textrits.Enabled = False
    shpts.BorderStyle = 3
    textrats.Text = 0
    textrits.Text = 0
End If
End Sub
```

```
Private Sub Form_Load()
```

```

numcom = frmcorona!ncf.Text          '# de cond./fase
textdis.Text = frmcorona!textdis.Text  'Distancia entre el haz

textaltura.Text = frmcorona!textaltura.Text 'MSNM
texttempe.Text = frmcorona!texttempe.Text  'Temperatura
cbots.Text = frmcorona!cbots.Text
cboth.Text = frmcorona!cboth.Text
cbocoefrugo.Text = frmcorona.cbocoefrugo
texttipo.Text = frmcorona!texttipo.Text
textmcm.Text = frmcorona!textmcm.Text
textradio.Text = frmcorona!textradio.Text
textdiame.Text = Val(textradio.Text) * 2

If frmcorona.chkth = Checked Then
    chkth.Value = Checked
Else
    chkth.Value = Unchecked
End If

If frmcorona.chkts = Checked Then
    chkts.Value = Checked
Else
    chkts.Value = Unchecked
End If

ncf.Text = numcom

hfase = Val(frmvistazo.textthca)
If hfase = 0 Then
    GoTo fin
Else
    texthfase.Text = hfase
End If
fin:
End Sub
Private Sub ncf_Click()
    ncfa = Int(ncf.Text)
    ncf.Text = ncfa

    Select Case ncfa
        Case Is = 0
            def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
            ncf.Text = 1

        Case Is < 1

```

```

        def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
        If def = vbYes Then
            ncf.Text = Abs(Int(ncf.Text))
            frec.SetFocus
        Else
            GoTo Line1
        End If
    End Select

End Sub

Private Sub ncf_Change()
    ncfa = Int(Val(ncf.Text))
    ncf.Text = ncfa

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If

    Select Case ncfa

        Case Is >= 6
            mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
            GoTo Line2

        Case Is = 0
            def = MsgBox("Por lo menos debe de existir un conductor ", 48, "Definición
de subconductores ")
Line1:
            ncf.Text = 1

        Case Is < 1
            def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
            If def = vbYes Then
                ncf.Text = Abs(Int(ncf.Text))
Line2:
            Else
                GoTo Line1
            End If
        End Select

    If ncfa >= 2 Then
        lbldis.Enabled = True

```

```

    textdis.Enabled = True
    Else
    lbldis.Enabled = False
    textdis.Enabled = False
End If
End Sub

```

```

Private Sub spinnu_SpinDown()
Dim dblrate
dblrate = Val(ncf.Text)

If dblrate > 1 Then
    ncf.Text = dblrate - 1
    Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub

```

```

Private Sub spinnu_SpinUp()
ncf.Text = (Val(ncf.Text)) + 1

If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh

End Sub

```

```

Private Sub textdiame_Change()
diametro = Val(textdiame.Text)
textradio.Text = diametro / 2
End Sub

```

```

Private Sub texthfase_Change()
Static pi
pi = 3.14159265358979

```

```

Rem ***** Radio equivalente *****
'Solo para un ckto
radio = Val(textradio.Text) / 10 'Datos radio en cm
dis = Val(textdis.Text) 'Datos en cm
ncfa = Val(ncf.Text) 'Datos # de c/fase

```

```

If textdis.Text = "" Then

```

```

    rmgg = radio
    requi = 0
    GoTo Line3
Else
    dis = Val(textdis.Text) 'Distancia entre subconductores en cm
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
    GoTo Line3
End If
Line3:

Rem ***** Correcciones por altura y temperatura *****
altura = Val(textaltura.Text) 'Datos
tempe = Val(texttempe.Text) 'Datos

presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
fcc = 3.921 * presion / (273 + tempe) 'factor de corrección

hfase = Val(texthfase.Text) 'Altura de la fase más baja

If hfase = 0 Then
    pelru = MsgBox("La altura de la fase más baja no puede ser cero", vbCritical,
"Altura de conductor no definido (DELTA)")
    texthfase.SetFocus
    GoTo fin
End If
diametro = Val(textdiame.Text) / 10 'Diámetro en cm
freri = Val(cbofreri.Text)
coefrugo = Val(cbocoefrugo.Text)

If chkts.Value = Checked Then ' *** Para tiempo seco ***
    lblts.Enabled = True
    lblrats.Enabled = True
    lblrits.Enabled = True
    textrats.Enabled = True
    textrits.Enabled = True
    shpts.BorderStyle = 1

    coefmetets = cbots.Text

    gcs = 30 * (fcc) ^ (2 / 3) * coefrugo * coefmetets * (1 - 0.07 * radio) * (1 - (ncfa
- 1) * radio / rmgg) 'Cálculo del gradiente crítico
    Select Case ncfa
        Case 1
            ANo = 75.2
            cte1 = 7.5

```

```

    cte2 = 0
Case 2
    ANo = 75.2
    cte1 = 2.6
    cte2 = 0
Case Is >= 3
    ANo = 67.9
    cte1 = 0
    cte2 = 22.9 * (ncfa - 1) * diametro / (rmgg * 2)
End Select

Rem Cálculo de la distancia radial
dx = Abs(Val(textx.Text))
dy = Abs(Val(texty.Text))
dr = Sqr((hfase - dy) ^ 2 + (dx) ^ 2)

Rem Cálculo del nivel de ruido audible 'Logn(x) = Log(x) / Log(n)
AN5ts = (-665 / gcs) + 20 * Log(ncfa) / Log(10) + 44 * Log(diametro) / Log(10)
- 10 * Log(dr) / Log(10) - 0.02 * dr + ANo + cte1 + cte2

Rem Cálculo del nivel de radio interferencia
rits = -150.4 + 120 * Log(gcs) / Log(10) + 40 * Log(diametro * 10) / Log(10) +
20 * Log(hfase / (dr) ^ 2) / Log(10) + 10 * (1 - (Log(10 * freri) / Log(10)) ^ 2)
textrats.Text = AN5ts
textrits.Text = rits

If chkth.Value = Unchecked Then
    lblth.Enabled = False
    lblrath.Enabled = False
    lblrith.Enabled = False
    textrath.Enabled = False
    textrith.Enabled = False
    shpth.BorderStyle = 3
    textrath.Text = 0
    textrith.Text = 0
End If
End If

If chkth.Value = Checked Then ' *** Para tiempo húmedo ***
    lblth.Enabled = True
    lblrath.Enabled = True
    lblrith.Enabled = True
    textrath.Enabled = True
    textrith.Enabled = True
    shpth.BorderStyle = 1

    coefmeteth = cboth.Text

```

```

gch = 30 * (fcc) ^ (2 / 3) * coefrugo * coefmeteth * (1 - 0.07 * radio) * (1 - (ncfa
- 1) * radio / rmgg) 'Cálculo del gradiente crítico

```

```

Select Case ncfa

```

```

  Case 1

```

```

    ANo = 75.2

```

```

    cte1 = 7.5

```

```

    cte2 = 0

```

```

  Case 2

```

```

    ANo = 75.2

```

```

    cte1 = 2.6

```

```

    cte2 = 0

```

```

  Case Is >= 3

```

```

    ANo = 67.9

```

```

    cte1 = 0

```

```

    cte2 = 22.9 * (ncfa - 1) * diametro / (rmgg * 2)

```

```

End Select

```

```

Rem Cálculo de la distancia radial

```

```

dx = Abs(Val(textx.Text))

```

```

dy = Abs(Val(texty.Text))

```

```

dr = Sqr((hfase - dy) ^ 2 + (dx) ^ 2)

```

```

Rem Cálculo del nivel de ruido audible 'Logn(x) = Log(x) / Log(n)

```

```

AN5th = (-665 / gch) + 20 * Log(ncfa) / Log(10) + 44 * Log(diametro) / Log(10)
- 10 * Log(dr) / Log(10) - 0.02 * dr + ANo + cte1 + cte2

```

```

Rem Cálculo del nivel de radio interferencia

```

```

rith = -150.4 + 120 * Log(gch) / Log(10) + 40 * Log(diametro * 10) / Log(10) +
20 * Log(hfase / (dr) ^ 2) / Log(10) + 10 * (1 - (Log(10 * freri) / Log(10)) ^ 2)

```

```

textrath.Text = AN5th

```

```

textrith.Text = rith

```

```

If chkts.Value = Unchecked Then

```

```

  lblts.Enabled = False

```

```

  lblrats.Enabled = False

```

```

  lblrits.Enabled = False

```

```

  textrats.Enabled = False

```

```

  textrits.Enabled = False

```

```

  shpts.BorderStyle = 3

```

```

  textrats.Text = 0

```

```

  textrits.Text = 0

```

```

End If

```

```

End If

```

```

If chkts.Value = Unchecked And chkth.Value = Unchecked Then

```



```
    msjcoro = "No ha definido ningún parámetro a calcular defina por lo menos  
uno: " & Chr(10) & Chr(13)  
    msjcoro = msjcoro & "Tiempo seco o Tiempo húmedo "  
    MsgBox msjcoro  
    chkts.SetFocus  
End If  
fin:  
End Sub  
Private Sub textradio_Change()  
    radio = Val(textradio.Text)  
    textdiame.Text = radio * 2  
End Sub
```

CALCULO DE TENSIONES Y FLECHAS

PLANTILLA DE FLECHAS (CURVAS)

(frmtensiones.frm)

Dim tiempo As String
 Dim tiempomin As String
 Dim tiempomax As String

Rem Variables que se visualizan con msgbox en la Ec principal (flecha normal)

Dim wvn 'Peso debido al viento kgf/m
 Dim whn 'Peso debido al hielo kgf/m
 Dim wavhn 'Peso aparente debido a viento y hielo Kgf/m
 'Dim fsn 'factor de sobrecarga
 Dim angf 'Angulo de la flecha
 Dim tdiseñonpmm 'Tensión de diseño Kgf/mm²
 Dim pesocpmm 'peso del cable Kgf/(m x mm²)
 Dim finclhip 'Flecha inclinada catenaria
 Dim finclipara 'Flecha inclinada parábola
 Dim ke 'Constante K de la cúbica:
 Dim tempn 'Temperatura inicial para flecha normal

Rem Variables que se visualizan con msgbox en la Ec de flecha mínima

Dim wvmin 'Peso debido al viento kgf/m
 Dim whmin 'Peso debido al hielo kgf/m
 Dim wavhmin 'Peso aparente debido a viento y hielo Kgf/m
 'Dim fsmin 'factor de sobrecarga
 Dim angfmin 'Angulo de la flecha
 Dim finclhipmin 'Flecha inclinada catenaria
 Dim fincliparamin 'Flecha inclinada parábola
 Dim nume1
 Dim nume2
 Dim tempmin 'Temperatura final para flecha mínima
 Dim tenfmin(3) 'Tensión final para flecha mínima (slns.)
 'Dim tffpmm 'Tensión final de diseño

Rem Variables que se visualizan con msgbox en la Ec de flecha máxima

Dim wvmax 'Peso debido al viento kgf/m
 Dim wavhmax 'Peso aparente debido a viento y hielo Kgf/m
 'Dim fsmax 'factor de sobrecarga
 Dim angfmax 'Angulo de la flecha
 Dim finclhipmax 'Flecha inclinada catenaria
 Dim fincliparamax 'Flecha inclinada parábola
 Dim nume12

```

Dim nume21
Dim tempmax      'Temperatura final para flecha máxima
Dim tenfmax(3)  'Tensión final para flecha máxima (slns.)
'Dim tffpmmx    'Tensión final de diseño flecha máxima

'Dim tffpmmx    'Tensión final de diseño
Private Sub btncaltenmin_Click()
btninfmin.Enabled = True
Rem ***** CALCULO DE TENSIONES PARA FLECHA MINIMA *****
Static pi
pi = 3.14159265358979

trupn = Val(textruptura.Text)      'Tensión de ruptura (kgf)
tempn = Val(texttempn.Text)        'Temperatura inicial
tempmin = Val(texttempmin.Text)    'Temperatura final (flecha mín.)
presionmin = Val(textpresionmin.Text)
areac = Val(textarea.Text)         'Area del conductor (mm²)
diametro = Val(textdiametro.Text) / 1000 'Diámetro del cable (m)
pesoc = Val(textpeso.Text) / 1000  'peso del cable (kgf/m)
vano = Val(textvano.Text)          'Vano (m)
elasti = Val(textelasticidad.Text) * 10 ^ 3 'Modulo de elasticidad
dilata = Val(textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación 1/°C

nume1 = elasti * dilata * (tempmin - tempn) - ke

If chkhmin.Value = Unchecked Then
    hielo = "no"
Else
    hielo = "si"
End If

If presionmin = 0 And hielo = "no" Then 'Sin viento y sin hielo
    GoTo svshmin
End If

If presionmin > 0 And hielo = "no" Then 'Con viento y sin hielo
    GoTo cvshmin
End If

If presionmin = 0 And hielo = "si" Then 'Sin viento y con hielo
    GoTo svchmin
End If

If presionmin > 0 And hielo = "si" Then 'Con viento y con hielo
    GoTo cvchmin
End If

```

```

Rem *****
Rem ***** DEFINICION DE CASOS *****

svshmin:   'Sin viento y sin hielo *****
           tiempomin = "SIN SOBRECARGA DE VIENTO Y HIELO"

           wvmin = 0      'Peso debido al viento kgf/m
           whmin = 0      'Peso debido al hielo kgf/m
           wavhmin = pesoc 'Peso aparente debido a viento y hielo Kgf/m
           fsmin = 1      'factor de sobrecarga
           angfmin = 0    'Angulo de la flecha

           'Ecuación cúbica:  $tf^3 + (nume1) tf^2 - (nume2) = 0$ 
           nume2 = elasti * (vano * pesocpmm * fsmin) ^ 2 / 24

           Rem resolver la ec. cúbica
           tenfmin(0) = cubica(-nume2, 0, nume1, 1)
           tenfmin(1) = slns(1)
           tenfmin(2) = slns(2)
           tenfmin(3) = slns(3)

           Rem Definición del valor ha tomar
           tffpmm = 0 'Reinicia la variable
           If Format(tenfmin(0), "0.000000") > 0 And tenfmin(0) < tdiseñonpmm Then
               'MsgBox "Se ha escojido " & tenfmin(0)
               tffpmm = tenfmin(0)
           End If
           If Format(tenfmin(1), "0.000000") > 0 And tenfmin(1) < tdiseñonpmm Then
               'MsgBox "Se ha escojido " & tenfmin(1)
               tffpmm = tenfmin(1)
           End If
           If Format(tenfmin(2), "0.000000") > 0 And tenfmin(2) < tdiseñonpmm Then
               'MsgBox "Se ha escojido " & tenfmin(2)
               tffpmm = tenfmin(2)
           End If
           If Format(tenfmin(3), "0.000000") > 0 And tenfmin(3) < tdiseñonpmm Then
               'MsgBox "Se ha escojido " & tenfmin(3)
               tffpmm = tenfmin(3)
           End If
           If tffpmm = 0 Then
               cnd = "Para este tipo de condiciones no resulta conveniente este tipo de
cable." & Chr(10) & Chr(13)
               cnd = cnd & "Verifique los siguientes parámetros:" & Chr(10) & Chr(13)
               cnd = cnd & "® Cable." & Chr(10) & Chr(13)
               cnd = cnd & "® Vano." & Chr(10) & Chr(13)
               cnd = cnd & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
               cnd = cnd & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)

```

```

otravez1:
    intentar = MsgBox(cnd, vbRetryCancel, tiempomin)
    If intentar = 4 Then
        cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
        cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
        cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
        cnd2 = cnd2 & "t1 = " & Format(slns(0), "0.00000") & " Kgf/mm2" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t2 = " & Format(slns(1), "0.00000") & " Kgf/mm2" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t3 = " & Format(slns(2), "0.00000") & " Kgf/mm2" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t4 = " & Format(slns(3), "0.00000") & " Kgf/mm2" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm2" & Chr(10) &
Chr(13)
        intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
        GoTo otravez1
    Else
        texttempmin.SelStart = 0
        texttempmin.SelLength = Len(texttempmin)
        texttempmin.SetFocus
    End If
End If
tfmin = tffpmm * areac 'tffpmm = solución
texttmin.Text = tfmin

Rem Comprobación del Factor de seguridad del cable
If tfmin = 0 Then GoTo fin
fscmin = trupn / tfmin
textfscamin.Text = fscmin

Rem **** Cálculo de la flecha ****
ctemmin = tfmin / pesoc 'Por la catenaria
flehipmin = fhip(ctemmin, vano, fsmin)
finclihipmin = 0 'Flecha inclinada
textfmin.Text = flehipmin 'Flecha vertical
'Flecha vertical

fleparamin = fpara(vano, pesoc, tfmin, fsmin) 'por la parábola
fincliparamin = 0 'Flecha inclinada
flepara12 = fleparamin 'Flecha vertical
'Flecha vertical

```

```
tipsmin = " Flecha (por la parábola): " & Format(flepara12, "0.000000") & " m "
textfmin.ToolTipText = tipsmin
GoTo fin
```

```
cvshmin:      'Con viento y sin hielo *****
tiempomin = "CON SOBRECARGA DE VIENTO Y SIN SOBRECARGA DE
HIELO"

wvmin = presionmin * diametro      'Peso debido al viento kgf/m
whmin = 0                          'Peso debido al hielo kgf/m
wavhmin = Sqr(pesoc ^ 2 + wvmin ^ 2) 'Peso aparente debido a viento y hielo
Kgf/m
fsmin = wavhmin / pesoc            'factor de sobrecarga
'Arccos(X) = Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)
X = pesoc / Sqr(pesoc ^ 2 + wvmin ^ 2)
angfmin = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi 'Angulo de la flecha

'Ecuación cúbica: tf3 + (nume1) tf2 - (nume2) = 0
nume2 = elasti * (vano * pesocpmm * fsmin) ^ 2 / 24

Rem resolver la ec. cúbica
tenfmin(0) = cubica(-nume2, 0, nume1, 1)
tenfmin(1) = slns(1)
tenfmin(2) = slns(2)
tenfmin(3) = slns(3)

Rem Definición del valor ha tomar
tffpmm = 0 'Reinicia la variable
If Format(tenfmin(0), "0.000000") > 0 And tenfmin(0) < tdiseñonpmm Then
  'MsgBox "Se ha escojido " & tenfmin(0)
  tffpmm = tenfmin(0)
End If
If Format(tenfmin(1), "0.000000") > 0 And tenfmin(1) < tdiseñonpmm Then
  'MsgBox "Se ha escojido " & tenfmin(1)
  tffpmm = tenfmin(1)
End If
If Format(tenfmin(2), "0.000000") > 0 And tenfmin(2) < tdiseñonpmm Then
  'MsgBox "Se ha escojido " & tenfmin(2)
  tffpmm = tenfmin(2)
End If
If Format(tenfmin(3), "0.000000") > 0 And tenfmin(3) < tdiseñonpmm Then
  'MsgBox "Se ha escojido " & tenfmin(3)
  tffpmm = tenfmin(3)
End If
If tffpmm = 0 Then
  cnd = "Para este tipo de condiciones no resulta conveniente este tipo de
cable." & Chr(10) & Chr(13)
```

```

cnd = cnd & "Verifique los siguientes parámetros:" & Chr(10) & Chr(13)
cnd = cnd & "® Cable." & Chr(10) & Chr(13)
cnd = cnd & "® Vano." & Chr(10) & Chr(13)
cnd = cnd & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
cnd = cnd & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)
otravez2:
    intentar = MsgBox(cnd, vbRetryCancel, tiempomin)
    If intentar = 4 Then
        cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
        cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
        cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
        cnd2 = cnd2 & "t1 = " & Format(slms(0), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t2 = " & Format(slms(1), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t3 = " & Format(slms(2), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t4 = " & Format(slms(3), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm²" & Chr(10) &
Chr(13)
        intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
        GoTo otravez2
    Else
        texttempmin.SelStart = 0
        texttempmin.SelLength = Len(texttempmin)
        texttempmin.SetFocus
    End If
End If
tfmin = tffpmm * areac 'tffpmm = solución
texttmin.Text = tfmin

Rem Comprobación del Factor de seguridad del cable
If tfmin = 0 Then GoTo fin
fscmin = trupn / tfmin
textfscamin.Text = fscmin

Rem **** Cálculo de la flecha ****
ctemmin = tfmin / pesoc 'Por la catenaria
flehipmin = fhip(ctemmin, vano, fsmin)
finclhipmin = flehipmin 'Flecha inclinada
textfmin.Text = flehipmin * Sin(angfmin * pi / 180) 'Flecha vertical
'Flecha vertical

```

```

fleparamin = fpara(vano, pesoc, tfmin, fsmin) 'por la parábola
fincliparamin = fleparamin 'Flecha inclinada
flepara12 = fleparamin * Sin(angfmin * pi / 180) 'Flecha vertical
'Flecha vertical
tipsmin = " Flecha (por la parábola): " & Format(flepara12, "0.000000") & " m "
textfmin.ToolTipText = tipsmin
GoTo fin

```

```

svchmin: 'Sin viento y con hielo *****
fchmin = Val(cbofchmin.Text)
tiempomin = "SIN SOBRECARGA DE VIENTO Y CON SOBRECARGA DE
HIELO"

```

```

wvmin = 0 'Peso debido al viento kgf/m
whmin = fchmin * Sqr(diametro * 1000) 'Peso debido al hielo kgf/m
wavhmin = pesoc + whmin 'Peso aparente debido a viento y hielo
Kgf/m
fsmin = wavhmin / pesoc 'factor de sobrecarga
angfmin = 0 'Angulo de la flecha

```

```

'Ecuación cúbica:  $tf^3 + (\text{nume1}) tf^2 - (\text{nume2}) = 0$ 
nume2 = elasti * (vano * pesocpmm * fsmin) ^ 2 / 24

```

```

Rem resolver la ec. cúbica
tenfmin(0) = cubica(-nume2, 0, nume1, 1)
tenfmin(1) = slns(1)
tenfmin(2) = slns(2)
tenfmin(3) = slns(3)

```

```

Rem Definición del valor ha tomar
tffpmm = 0 'Reinicia la variable
If Format(tenfmin(0), "0.000000") > 0 And tenfmin(0) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(0)
  tffpmm = tenfmin(0)
End If
If Format(tenfmin(1), "0.000000") > 0 And tenfmin(1) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(1)
  tffpmm = tenfmin(1)
End If
If Format(tenfmin(2), "0.000000") > 0 And tenfmin(2) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(2)
  tffpmm = tenfmin(2)
End If
If Format(tenfmin(3), "0.000000") > 0 And tenfmin(3) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(3)
  tffpmm = tenfmin(3)

```



```

End If
If tffpmm = 0 Then
  cnd = "Para este tipo de condiciones no resulta conveniente este tipo de
cable." & Chr(10) & Chr(13)
  cnd = cnd & "Verifique los siguientes parámetros:" & Chr(10) & Chr(13)
  cnd = cnd & "Ⓜ Cable." & Chr(10) & Chr(13)
  cnd = cnd & "Ⓜ Vano." & Chr(10) & Chr(13)
  cnd = cnd & "Ⓜ Condiciones de Sobrecarga." & Chr(10) & Chr(13)
  cnd = cnd & "Ⓜ Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)
otravez3:
  intentar = MsgBox(cnd, vbRetryCancel, tiempomin)
  If intentar = 4 Then
    cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
    cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
    cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
    cnd2 = cnd2 & "t1 = " & Format(slms(0), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t2 = " & Format(slms(1), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t3 = " & Format(slms(2), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t4 = " & Format(slms(3), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm²" & Chr(10) &
Chr(13)
    intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
    GoTo otravez3
  Else
    texttempmin.SelStart = 0
    texttempmin.SelLength = Len(texttempmin)
    texttempmin.SetFocus
  End If
End If

tfmin = tffpmm * areac 'tffpmm = solución
texttmin.Text = tfmin

Rem Comprobación del Factor de seguridad del cable
If tfmin = 0 Then GoTo fin
fscmin = trupn / tfmin
textfscamin.Text = fscmin

Rem **** Cálculo de la flecha ****

```

```

ctemmin = tfmin / pesoc          'Por la catenaria
flehipmin = fhip(ctemmin, vano, fsmin)
finclhipmin = 0                 'Flecha inclinada
textfmin.Text = flehipmin       'Flecha vertical
'Flecha vertical

fleparamin = fpara(vano, pesoc, tfmin, fsmin) 'por la parábola
fincliparamin = 0               'Flecha inclinada
flepara12 = fleparamin         'Flecha vertical
'Flecha vertical
tipsmin = " Flecha (por la parábola): " & Format(flepara12, "0.000000") & " m "
textfmin.ToolTipText = tipsmin
GoTo fin

cvchmin:      'Con viento y con hielo *****
fchmin = Val(cbofchmin.Text)
diahmin = Val(cbodiahmin.Text)
tiempomin = "CON SOBRECARGA DE VIENTO y HIELO"

wvmin = presionmin * (diametro + 2 * diahmin / 1000) 'Peso debido al viento
kgf/m
whmin = fchmin * Sqr(diametro * 1000)                'Peso debido al hielo kgf/m
wavhmin = Sqr((pesoc + whmin) ^ 2 + wvmin ^ 2)        'Peso aparente debido
a viento y hielo Kgf/m
fsmin = wavhmin / pesoc                              'factor de sobrecarga
'Arccos(X) = Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)
X = (pesoc + whmin) / Sqr((pesoc + whmin) ^ 2 + wvmin ^ 2)
angfmin = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi 'Angulo de la
flecha

'Ecuación cúbica: tf3 + (nume1) tf2 - (nume2) = 0
nume2 = elasti * (vano * pesocpmm * fsmin) ^ 2 / 24

Rem resolver la ec. cúbica
tenfmin(0) = cubica(-nume2, 0, nume1, 1) 'Llamar a la función cúbica
tenfmin(1) = slns(1)
tenfmin(2) = slns(2)
tenfmin(3) = slns(3)

Rem Definición del valor ha tomar
tffpmm = 0 'Reinicia la variable
If Format(tenfmin(0), "0.000000") > 0 And tenfmin(0) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(0)
  tffpmm = tenfmin(0)
End If
If Format(tenfmin(1), "0.000000") > 0 And tenfmin(1) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmin(1)

```

```

    tffpmm = tenfmin(1)
End If
If Format(tenfmin(2), "0.000000") > 0 And tenfmin(2) < tdiseñonpmm Then
    'MsgBox "Se ha escojido " & tenfmin(2)
    tffpmm = tenfmin(2)
End If
If Format(tenfmin(3), "0.000000") > 0 And tenfmin(3) < tdiseñonpmm Then
    'MsgBox "Se ha escojido " & tenfmin(3)
    tffpmm = tenfmin(3)
End If
If tffpmm = 0 Then
    cnd = "Para este tipo de condiciones no resulta conveniente este tipo de
cable." & Chr(10) & Chr(13)
    cnd = cnd & "Verifique los siguientes parámetros:" & Chr(10) & Chr(13)
    cnd = cnd & "® Cable." & Chr(10) & Chr(13)
    cnd = cnd & "® Vano." & Chr(10) & Chr(13)
    cnd = cnd & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
    cnd = cnd & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)
otravez4:
    intentar = MsgBox(cnd, vbRetryCancel, tiempomin)
    If intentar = 4 Then
        cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
        cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
        cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
        cnd2 = cnd2 & "t1 = " & Format(slms(0), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t2 = " & Format(slms(1), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t3 = " & Format(slms(2), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "t4 = " & Format(slms(3), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
        cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm²" & Chr(10) &
Chr(13)
        intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
        GoTo otravez4
    Else
        texttempmin.SelStart = 0
        texttempmin.SelLength = Len(texttempmin)
        texttempmin.SetFocus
    End If
End If
GoTo fin

```

```
tfmin = tffpmm * areac 'tffpmm = solución
texttmin.Text = tfmin
```

```
Rem Comprobación del Factor de seguridad del cable
If tfmin = 0 Then GoTo fin
fscmin = trupn / tfmin
textfscamin.Text = fscmin
```

```
Rem **** Cálculo de la flecha ****
ctemmin = tfmin / pesoc 'Por la catenaria
flehipmin = fhip(ctemmin, vano, fsmin)
finclhipmin = flehipmin 'Flecha inclinada
textfmin.Text = flehipmin * Sin(angfmin * pi / 180) 'Flecha vertical
'Flecha vertical
```

```
fleparamin = fpara(vano, pesoc, tfmin, fsmin) 'por la parábola
fincliparamin = fleparamin 'Flecha inclinada
flepara12 = fleparamin * Sin(angfmin * pi / 180) 'Flecha vertical
'Flecha vertical
tipsmin = " Flecha (por la parábola): " & Format(flepara12, "0.000000") & " m "
textfmin.ToolTipText = tipsmin
GoTo fin
```

```
fin:
End Sub
```

```
Private Sub btncaltenmmax_Click()
btninfmax.Enabled = True
```

```
Rem ***** CALCULO DE TENSIONES PARA FLECHA MAXIMA *****
Static pi
pi = 3.14159265358979
```

```
trupn = Val(textruptura.Text) 'Tensión de ruptura (kgf)
tempn = Val(texttempn.Text) 'Temperatura inicial
tempmax = Val(texttempmax.Text) 'Temperatura final (flecha máx.)
presionmax = Val(textpresionmax.Text)
areac = Val(textarea.Text) 'Area del conductor (mm2)
diametro = Val(textdiametro.Text) / 1000 'Diámetro del cable (m)
pesoc = Val(textpeso.Text) / 1000 'peso del cable (kgf/m)
vano = Val(textvano.Text) 'Vano (m)
elasti = Val(textelasticidad.Text) * 10 ^ 3 'Modulo de elasticidad
dilata = Val(textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación 1/°C
```

```
nume12 = elasti * dilata * (tempmax - tempn) - ke
```

```
If presionmax = 0 Then 'Sin viento
```

```

    GoTo svmax
End If

If presionmax > 0 Then 'Con viento
    GoTo cvmax
End If

Rem *****
Rem ***** DEFINICION DE CASOS *****

svmax:    'Sin viento *****
    tiempomax = "SIN SOBRECARGA DE VIENTO"

    wvmax = 0      'Peso debido al viento kgf/m
    wavhmax = pesoc 'Peso aparente debido a viento y hielo Kgf/m
    fsmax = 1      'factor de sobrecarga
    angfmax = 0    'Angulo de la flecha

    'Ecuación cúbica:  $tf^3 + (nume12) tf^2 - (nume21) = 0$ 
    nume21 = elasti * (vano * pesocpmm * fsmax) ^ 2 / 24

    Rem resolver la ec. cúbica
    tenfmax(0) = cubica(-nume21, 0, nume12, 1)
    tenfmax(1) = slns(1)
    tenfmax(2) = slns(2)
    tenfmax(3) = slns(3)

    Rem Definición del valor ha tomar
    tffpmmmax = 0 'Reinicia la variable
    If Format(tenfmax(0), "0.000000") > 0 And tenfmax(0) < tdiseñonpmm Then
        'MsgBox "Se ha escogido " & tenfmax(0)
        tffpmmmax = tenfmax(0)
    End If
    If Format(tenfmax(1), "0.000000") > 0 And tenfmax(1) < tdiseñonpmm Then
        'MsgBox "Se ha escogido " & tenfmax(1)
        tffpmmmax = tenfmax(1)
    End If
    If Format(tenfmax(2), "0.000000") > 0 And tenfmax(2) < tdiseñonpmm Then
        'MsgBox "Se ha escogido " & tenfmax(2)
        tffpmmmax = tenfmax(2)
    End If
    If Format(tenfmax(3), "0.000000") > 0 And tenfmax(3) < tdiseñonpmm Then
        'MsgBox "Se ha escogido " & tenfmax(3)
        tffpmmmax = tenfmax(3)
    End If
    If tffpmmmax = 0 Then

```

```

        cndmax = "Para este tipo de condiciones no resulta conveniente este tipo
de cable." & Chr(10) & Chr(13)
        cndmax = cndmax & "Verifique los siguientes parámetros:" & Chr(10) &
Chr(13)
        cndmax = cndmax & "® Cable." & Chr(10) & Chr(13)
        cndmax = cndmax & "® Vano." & Chr(10) & Chr(13)
        cndmax = cndmax & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
        cndmax = cndmax & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) &
Chr(13)
otrazvezmax1:
        intentar = MsgBox(cndmax, vbRetryCancel, tiempomax)
        If intentar = 4 Then
                cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
                cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
                cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
                cnd2 = cnd2 & "t1 = " & Format(slns(0), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
                cnd2 = cnd2 & "t2 = " & Format(slns(1), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
                cnd2 = cnd2 & "t3 = " & Format(slns(2), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
                cnd2 = cnd2 & "t4 = " & Format(slns(3), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
                cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm²" & Chr(10) &
Chr(13)
                intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
                GoTo otravezmax1
        Else
                texttempmax.SelStart = 0
                texttempmax.SelLength = Len(texttempmin)
                texttempmax.SetFocus
        End If
    End If
    tfmax = tffpmmmax * areac 'tffpmmmax = solución
    texttmax.Text = tfmax
    Rem Comprobación del Factor de seguridad del cable
    fscmax = trupn / tfmax
    textfscamax.Text = fscmax

    Rem **** Cálculo de la flecha ****

    ctemmax = tfmax / pesoc           'Por la catenaria
    flehipmax = fhip(ctemmax, vano, fsmax)

```

```

    finclhipmax = 0                                'Flecha inclinada
    textfmax.Text = flehipmax                      'Flecha vertical
'Flecha vertical

    fleparamax = fpara(vano, pesoc, tfmax, fsmax) 'por la parábola
    fincliparamax = 0                             'Flecha inclinada
    flepara13 = fleparamax                        'Flecha vertical
'Flecha vertical
    tipsmax = " Flecha (por la parábola): " & Format(flepara13, "0.000000") & " m
"
    textfmax.ToolTipText = tipsmax
GoTo fin

cvmax:      'Con viento *****
    tiempomax = "CON SOBRECARGA DE VIENTO"

    wvmax = presionmax * diametro                 'Peso debido al viento kgf/m
    wavhmax = Sqr(pesoc ^ 2 + wvmax ^ 2)         'Peso aparente debido a viento y
hielo Kgf/m
    fsmax = wavhmax / pesoc                       'factor de sobrecarga
    'Arccos(X) = Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)
    X = pesoc / Sqr(pesoc ^ 2 + wvmax ^ 2)
    angfmax = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi 'Angulo de la
flecha

'Ecuación cúbica:  $tf^3 + (\text{nume12}) tf^2 - (\text{nume21}) = 0$ 
    nume21 = elasti * (vano * pesocpmm * fsmax) ^ 2 / 24

Rem resolver la ec. cúbica
tenfmax(0) = cubica(-nume21, 0, nume12, 1)
tenfmax(1) = slns(1)
tenfmax(2) = slns(2)
tenfmax(3) = slns(3)

Rem Definición del valor ha tomar
tffpmm = 0 'Reinicia la variable
If Format(tenfmax(0), "0.000000") > 0 And tenfmax(0) < tdiseñonpmm Then
    'MsgBox "Se ha escogido " & tenfmax(0)
    tffpmm = tenfmax(0)
End If
If Format(tenfmax(1), "0.000000") > 0 And tenfmax(1) < tdiseñonpmm Then
    'MsgBox "Se ha escogido " & tenfmax(1)
    tffpmm = tenfmax(1)
End If
If Format(tenfmax(2), "0.000000") > 0 And tenfmax(2) < tdiseñonpmm Then
    'MsgBox "Se ha escogido " & tenfmax(2)
    tffpmm = tenfmax(2)

```

```

End If
If Format(tenfmax(3), "0.000000") > 0 And tenfmax(3) < tdiseñonpmm Then
  'MsgBox "Se ha escogido " & tenfmax(3)
  tffpmm = tenfmax(3)
End If
If tffpmmmax = 0 Then
  cndmax = "Para este tipo de condiciones no resulta conveniente este tipo
de cable." & Chr(10) & Chr(13)
  cndmax = cndmax & "Verifique los siguientes parámetros:" & Chr(10) &
Chr(13)
  cndmax = cndmax & "® Cable." & Chr(10) & Chr(13)
  cndmax = cndmax & "® Vano." & Chr(10) & Chr(13)
  cndmax = cndmax & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
  cndmax = cndmax & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) &
Chr(13)
otravezmax2:
  intentar = MsgBox(cndmax, vbRetryCancel, tiempomax)
  If intentar = 4 Then
    cnd2 = "Se ha intentado evaluar las mismas condiciones, pero la
solución" & Chr(10) & Chr(13)
    cnd2 = cnd2 & "esta fuera del rango de la tensión de diseño: " & Chr(10)
& Chr(13)
    cnd2 = cnd2 & "Las tensiones calculadas son: " & Chr(10) & Chr(13)
    cnd2 = cnd2 & "t1 = " & Format(slms(0), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t2 = " & Format(slms(1), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t3 = " & Format(slms(2), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "t4 = " & Format(slms(3), "0.00000") & " Kgf/mm²" &
Chr(10) & Chr(13)
    cnd2 = cnd2 & "Las cuales superan a la tensión de diseño ( o son
'negativas') : " & Format(tdiseñonpmm, "0.00000") & " Kgf/mm²" & Chr(10) &
Chr(13)
    intentar2 = MsgBox(cnd2, vbInformation, "Condición de diseño fuera de
rango (DELTA)")
    GoTo otravezmax2
  Else
    texttempmax.SelStart = 0
    texttempmax.SelLength = Len(texttempmin)
    texttempmax.SetFocus
  End If
End If
tfmax = tffpmmmax * areac 'tffpmmmax = solución
texttmax.Text = tfmax
Rem Comprobación del Factor de seguridad del cable
fscmax = trupn / tfmax

```



```

textfscamax.Text = fscmax

Rem **** Cálculo de la flecha ****

ctemmax = tfmax / pesoc           'Por la catenaria
flehipmax = fhip(ctemmax, vano, fsmax)
finclihpmax = flehipmax          'Flecha inclinada
textfmax.Text = flehipmax * Sin(angfmax * pi / 180) 'Flecha vertical
'Flecha vertical

fleparamax = fpara(vano, pesoc, tfmax, fsmax) 'por la parábola
fincliparamax = fleparamax        'Flecha inclinada
flepara13 = fleparamax * Sin(angfmax * pi / 180) 'Flecha vertical
'Flecha vertical
tipsmax = " Flecha (por la parábola): " & Format(flepara13, "0.000000") & " m
"

textfmax.ToolTipText = tipsmax
GoTo fin

fin:
End Sub

Private Sub btncaltenn_Click()
btninf.Enabled = True
btncaltenmin.Enabled = True
btncaltenmmax.Enabled = True

Rem ***** CALCULO DE TENSIONES *****
Static pi
pi = 3.14159265358979

presionn = Val(textpresionn.Text)
trupn = Val(textruptura.Text)      'Tensión de ruptura (kgf)
fsc = Val(textfsc.Text)           'Factor de seguridad del cable
pesoc = Val(textpeso.Text) / 1000 'peso del cable (kgf/m)
areac = Val(textarea.Text)        'Area del conductor (mm²)
diametro = Val(textdiametro.Text) / 1000 'Diámetro del cable (m)
vano = Val(textvano.Text)         'Vano (m)
elasti = Val(textelasticidad.Text) * 10 ^ 3 'Modulo de elasticidad
dilata = Val(textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación 1/°C

tdiseño = trupn / fsc             'Tensión de diseño normal
texttn.Text = tdiseño
tdiseño pmm = tdiseño / areac     'Tensión de diseño (Kgf/mm²)
tffpmmn = tdiseño pmm
pesocpmm = pesoc / areac         'peso del cable (Kgf/m x mm²)

```

```
If chkhn.Value = Unchecked Then
  hielo = "no"
Else
  hielo = "si"
End If
```

```
If presionn = 0 And hielo = "no" Then 'Sin viento y sin hielo
  GoTo svsh
End If
```

```
If presionn > 0 And hielo = "no" Then 'Con viento y sin hielo
  GoTo cvsh
End If
```

```
If presionn = 0 And hielo = "si" Then 'Sin viento y con hielo
  GoTo svch
End If
```

```
If presionn > 0 And hielo = "si" Then 'Con viento y con hielo
  GoTo cvch
End If
```

```
Rem *****
Rem ***** DEFINICION DE CASOS *****
```

```
svsh: 'Sin viento y sin hielo *****
```

```
tiempo = "SIN SOBRECARGA DE VIENTO Y HIELO"
textfscan.Text = fsc
wvn = 0 'Peso debido al viento kgf/m
whn = 0 'Peso debido al hielo kgf/m
wavhn = pesoc 'Peso aparente debido a viento y hielo Kgf/m
fsn = 1 'factor de sobrecarga
angf = 0 'Angulo de la flecha
```

```
Rem **** Cálculo de la flecha ****
```

```
ctem = tdiseño / (pesoc) 'Por la catenaria
flehip = fhip(ctem, vano, fsn)
finclhip = 0 'Flecha inclinada
textfn.Text = flehip 'Flecha vertical
```

```
flepara = fpara(vano, pesoc, tdiseño, fsn) 'por la parábola
finclipara = 0 'Flecha inclinada
flepara1 = flepara 'Flecha vertical
tipsn = " Flecha (por la parábola): " & Format(flepara1, "0.000000") & " m "
```

```

textfn.ToolTipText = tipsn

'Constante ke de la cúbica:
ke = tdiseñonpmm - (elasti * (vano * pesocpmm * fsn) ^ 2 / (24 * tdiseñonpmm
^ 2))
'Ecuación cúbica:  $tf^3 + (num) tf^2 - (cte) = 0$ 
GoTo fin

cvsh:   'Con viento y sin hielo *****

        tiempo = "CON SOBRECARGA DE VIENTO Y SIN SOBRECARGA DE
HIELO"
textfscan.Text = fsc
wvn = presionn * diametro      'Peso debido al viento kgf/m
whn = 0                        'Peso debido al hielo kgf/m
wavhn = Sqr(pesoc ^ 2 + wvn ^ 2) 'Peso aparente debido a viento y hielo
Kgf/m
fsn = wavhn / pesoc           'factor de sobrecarga
'Arccos(X) =  $Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)$ 
X = pesoc / Sqr(pesoc ^ 2 + wvn ^ 2)
angf = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi 'Angulo de la flecha

Rem **** Cálculo de la flecha ****

ctem = tdiseñon / (pesoc)      'Por la catenaria
flehip = fhip(ctem, vano, fsn)
finclihip = flehip             'Flecha inclinada
textfn.Text = flehip * Sin(angf * pi / 180) 'Flecha vertical

flepara = fpara(vano, pesoc, tdiseñon, fsn) 'por la parábola
finclipara = flepara           'Flecha inclinada
flepara1 = flepara * Sin(angf * pi / 180) 'Flecha vertical
tipsn = " Flecha (por la parábola): " & Format(flepara1, "0.000000") & " m "
textfn.ToolTipText = tipsn

'Constante ke de la cúbica:
ke = tdiseñonpmm - (elasti * (vano * pesocpmm * fsn) ^ 2 / (24 * tdiseñonpmm
^ 2))
'Ecuación cúbica:  $tf^3 + (num) tf^2 - (cte) = 0$ 
GoTo fin

svch:   'Sin viento y con hielo *****

        fchn = Val(cbofchn.Text)
        tiempo = "SIN SOBRECARGA DE VIENTO Y CON SOBRECARGA DE
HIELO"
        textfscan.Text = fsc

```

```

wvn = 0                                'Peso debido al viento kgf/m
whn = fchn * Sqr(diametro * 1000)      'Peso debido al hielo kgf/m
wavhn = pesoc + whn                    'Peso aparente debido a viento y hielo Kgf/m
fsn = wavhn / pesoc                    'factor de sobrecarga
angf = 0                                'Angulo de la flecha

Rem **** Cálculo de la flecha ****

ctem = tdiseño / (pesoc)                'Por la catenaria
flehip = fhip(ctem, vano, fsn)
finclihip = 0                          'Flecha inclinada
textfn.Text = flehip                    'Flecha vertical

flepara = fpara(vano, pesoc, tdiseño, fsn) 'por la parábola
finclipara = 0                          'Flecha inclinada
flepara1 = flepara                      'Flecha vertical
tipsn = " Flecha (por la parábola): " & Format(flepara1, "0.000000") & " m "
textfn.ToolTipText = tipsn

'Constante ke de la cúbica:
ke = tdiseño * pmm - (elasti * (vano * pesoc * pmm * fsn) ^ 2 / (24 * tdiseño * pmm
^ 2))
'Ecuación cúbica: tf^3 + (num) tf^2 - (cte) = 0
GoTo fin

cvch:   'Con viento y con hielo *****

fchn = Val(cbofchn.Text)
diahn = Val(cbodiahn.Text)
tiempo = "CON SOBRECARGA DE VIENTO y HIELO"
textfscan.Text = fsc
wvn = presionn * (diametro + 2 * diahn / 1000) 'Peso debido al viento kgf/m
whn = fchn * Sqr(diametro * 1000)            'Peso debido al hielo kgf/m
wavhn = Sqr((pesoc + whn) ^ 2 + wvn ^ 2)     'Peso aparente debido a
viento y hielo Kgf/m
fsn = wavhn / pesoc                          'factor de sobrecarga
'Arccos(X) = Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)
X = (pesoc + whn) / Sqr((pesoc + whn) ^ 2 + wvn ^ 2)
angf = (Atn(-X / Sqr(-X * X + 1)) + 2 * Atn(1)) * 180 / pi 'Angulo de la flecha

Rem **** Cálculo de la flecha ****

ctem = tdiseño / (pesoc)                    'Por la catenaria
flehip = fhip(ctem, vano, fsn)
finclihip = flehip                          'Flecha inclinada
textfn.Text = flehip * Sin(angf * pi / 180) 'Flecha vertical

```

```

flepara = fpara(vano, pesoc, tdiseñon, fsn) 'por la parábola
finclipara = flepara 'Flecha inclinada
flepara1 = flepara * Sin(angf * pi / 180) 'Flecha vertical
tipsn = " Flecha (por la parábola): " & Format(flepara1, "0.000000") & " m "
textfn.ToolTipText = tipsn

'Constante ke de la cúbica:
ke = tdiseñonpmm - (elasti * (vano * pesocpmm * fsn) ^ 2 / (24 * tdiseñonpmm
^ 2))
'Ecuación cúbica:  $tf^3 + (\text{num}) tf^2 - (\text{cte}) = 0$ 
GoTo fin

fin:
End Sub

Private Sub btngrabar_Click()
Dim dato As exportar
Dim Linea As String
Dim NFile As Integer
Dim NArchivo As String

NFile = FreeFile()
Linea = InputBox("Titulo del Caso ?", "Membrete del caso estudiado (DELTA)")
If IsNull(Linea) Then Linea = "Ninguno"

dato.Titulo = Linea
dato.Nominal = Val(texttmin.Text)
dato.Caliente = Val(texttmax.Text)
dato.Fria = Val(texttn.Text)
dato.Peso = Val(textpeso.Text) / 1000

frmtensiones.CMD1.Filter = "Archivo de Tensiones (*.ten) |*.ten"
frmtensiones.CMD1.Flags = &H4
frmtensiones.CMD1.Action = 2
NArchivo = frmtensiones.CMD1.filename

If NArchivo <> "" Then
Open NArchivo For Random As #NFile
Put #NFile, , dato
Close #NFile
End If

'Dim RetVal As Double
'RetVal = Shell("C:\Archivos de programa\Microsoft
Office\Office\MSACCESS.EXE C:\Delta\bases\DataAisladores.mdb", 1)
Dim RetVal As Double

```

```
RetVal = Shell("C:\Archivos de programa\Microsoft Office\Office\Excel.exe
C:\Delta\Curvas", 1)
```

```
final:
```

```
End Sub
Private Sub btninf_Click()
msjn = tiempo & Chr(10) & Chr(13)
msjn = msjn & Chr(10) & Chr(13)
msjn = msjn & "Datos relacionados a esta condición: " & Chr(10) & Chr(13)
msjn = msjn & "Tensión de trabajo por unidad de area: to = " &
Format(tdiseñonpmm, "0.00000") & " Kgf / mm²" & Chr(10) & Chr(13)
msjn = msjn & "Peso del cable por unidad de longitud por area: w = " &
Format(pesocpmm, "0.0000000") & " Kgf / (m x mm²)" & Chr(10) & Chr(13)
msjn = msjn & "peso ocasionado por el viento: wv = " & Format(wvn, "0.00000")
& " Kgf / m" & Chr(10) & Chr(13)
msjn = msjn & "peso ocasionado por el hielo: wh = " & Format(whn, "0.00000")
& " Kgf / m" & Chr(10) & Chr(13)
msjn = msjn & "peso aparente: wa = " & Format(wavhn, "0.00000") & " Kgf / m"
& Chr(10) & Chr(13)
msjn = msjn & "Factor de sobrecarga: mo = " & Format(fsn, "0.00000") &
Chr(10) & Chr(13)
msjn = msjn & "Angulo de inclinación de la flecha: " & Format(angf, "0.00000") &
" °" & Chr(10) & Chr(13)
msjn = msjn & "Flecha inclinada (Catenaria) = " & Format(finclihip, "0.00000") &
" m" & Chr(10) & Chr(13)
msjn = msjn & "Flecha inclinada (Parábola) = " & Format(finclipara, "0.00000") &
" m" & Chr(10) & Chr(13)
msjn = msjn & Chr(10) & Chr(13)
msjn = msjn & "Factor K de la ecuación cúbica: " & Chr(10) & Chr(13)
msjn = msjn & "K = to - ( E / 24 ) ( vano w mo / to ) ^2" & " = " & Format(ke,
"0.00000") & Chr(10) & Chr(13)
msjn = msjn & Chr(10) & Chr(13)
Norm = MsgBox(msjn, vbExclamation, "ECUACION BASE")
End Sub
```

```
Private Sub btninfmax_Click()
msjmax = tiempomax & Chr(10) & Chr(13)
msjmax = msjmax & Chr(10) & Chr(13)
msjmax = msjmax & "Datos relacionados a esta condición: " & Chr(10) &
Chr(13)
msjmax = msjmax & "peso ocasionado por el viento: wv = " & Format(wvmax,
"0.00000") & " Kgf / m" & Chr(10) & Chr(13)
msjmax = msjmax & "peso aparente: wa = " & Format(wavhmax, "0.00000") & "
Kgf / m" & Chr(10) & Chr(13)
msjmax = msjmax & "Factor de sobrecarga: m = " & Format(fsmax, "0.00000") &
Chr(10) & Chr(13)
```

```

msjmax = msjmax & "Angulo de inclinación de la flecha: " & Format(angfmax,
"0.00000") & " °" & Chr(10) & Chr(13)
msjmax = msjmax & "Flecha inclinada (Catenaria) = " & Format(finclihipmax,
"0.00000") & " m" & Chr(10) & Chr(13)
msjmax = msjmax & "Flecha inclinada (Parábola) = " & Format(fincliparamax,
"0.00000") & " m" & Chr(10) & Chr(13)
msjmax = msjmax & "Temperatura inicial: Ti = " & tempn & " °C" & Chr(10) &
Chr(13)
msjmax = msjmax & "Temperatura final: Tf = " & tempmax & " °C" & Chr(10) &
Chr(13)
msjmax = msjmax & Chr(10) & Chr(13)
msjmax = msjmax & "Ecuación de cambio de estado:  $tf^3 + a tf^2 - b = 0$ " &
Chr(10) & Chr(13)
msjmax = msjmax & "a =  $E \alpha (Tf - Ti) - K$  = " & Format(ume12, "0.0000") &
Chr(10) & Chr(13)
msjmax = msjmax & "b =  $( E / 24 ) ( vano w mf ) ^2$  = " & Format(ume21,
"0.00000") & Chr(10) & Chr(13)
msjmax = msjmax & Chr(10) & Chr(13)
msjmax = msjmax & "Tensión de trabajo por unidad de area: tf = " &
Format(tffpmax, "0.00000") & " Kgf / mm2" & Chr(10) & Chr(13)
Max = MsgBox(msjmax, vbExclamation, "FLECHA MAXIMA")
End Sub

```

```

Private Sub btninfmin_Click()
msjmin = tiempomin & Chr(10) & Chr(13)
msjmin = msjmin & Chr(10) & Chr(13)
msjmin = msjmin & "Datos relacionados a esta condición: " & Chr(10) & Chr(13)
msjmin = msjmin & "peso ocasionado por el viento: wv = " & Format(wvmin,
"0.00000") & " Kgf / m" & Chr(10) & Chr(13)
msjmin = msjmin & "peso ocasionado por el hielo: wh = " & Format(whmin,
"0.00000") & " Kgf / m" & Chr(10) & Chr(13)
msjmin = msjmin & "peso aparente: wa = " & Format(wavhmin, "0.00000") & "
Kgf / m" & Chr(10) & Chr(13)
msjmin = msjmin & "Factor de sobrecarga: m = " & Format(fsmin, "0.00000") &
Chr(10) & Chr(13)
msjmin = msjmin & "Angulo de inclinación de la flecha: " & Format(angfmin,
"0.00000") & " °" & Chr(10) & Chr(13)
msjmin = msjmin & "Flecha inclinada (Catenaria) = " & Format(finclihipmin,
"0.00000") & " m" & Chr(10) & Chr(13)
msjmin = msjmin & "Flecha inclinada (Parábola) = " & Format(fincliparamin,
"0.00000") & " m" & Chr(10) & Chr(13)
msjmin = msjmin & "Temperatura inicial: Ti = " & tempn & " °C" & Chr(10) &
Chr(13)
msjmin = msjmin & "Temperatura final: Tf = " & tempmin & " °C" & Chr(10) &
Chr(13)
msjmin = msjmin & Chr(10) & Chr(13)

```

```

msjmin = msjmin & "Ecuación de cambio de estado:  $tf^3 + a tf^2 - b = 0$ " &
Chr(10) & Chr(13)
msjmin = msjmin & "a = E alfa (Tf - Ti) - K = " & Format(ume1, "0.0000") &
Chr(10) & Chr(13)
msjmin = msjmin & "b = ( E / 24 ) ( vano w mf ) ^2 = " & Format(ume2,
"0.00000") & Chr(10) & Chr(13)
msjmin = msjmin & Chr(10) & Chr(13)
msjmin = msjmin & "Tensión de trabajo por unidad de area: tf = " &
Format(tffpmm, "0.00000") & " Kg / mm2" & Chr(10) & Chr(13)
Min = MsgBox(msjmin, vbExclamation, "FLECHA MINIMA")
End Sub

```

```

Private Sub btnotrainf_Click()
frmLonvanocri.Show vbModal
End Sub

```

```

Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub

```

```

Private Sub btnvfrm cables_Click()
rastreo6 = "activo frm tensiones"
Hide
frmCables.Show
End Sub

```

```

Private Sub btnvfrmprevie_Click()
frmpacalpre.Show vbModal
End Sub

```

```

Private Sub btnvfrmvdes_Click()
Hide
frmvanosdes.Show
End Sub

```

```

Private Sub cbodiahmin_Click()
btninfmin.Enabled = False
End Sub

```

```

Private Sub cbodiahmin_Change()
btninfmin.Enabled = False
End Sub

```

```

Private Sub cbodiahn_Click()
btninf.Enabled = False
btncaltenmin.Enabled = False

```



```
btninfmin.Enabled = False
btncaltenmmax.Enabled = False
btninfmax.Enabled = False
End Sub
```

```
Private Sub cbodiahn_Change()
btninf.Enabled = False
btncaltenmin.Enabled = False
btninfmin.Enabled = False
btncaltenmmax.Enabled = False
btninfmax.Enabled = False
End Sub
```

```
Private Sub cbofchmin_Click()
btninfmin.Enabled = False
End Sub
```

```
Private Sub cbofchmin_Change()
btninfmin.Enabled = False
End Sub
```

```
Private Sub cbofchn_Click()
btninf.Enabled = False
btncaltenmin.Enabled = False
btninfmin.Enabled = False
btncaltenmmax.Enabled = False
btninfmax.Enabled = False
End Sub
```

```
Private Sub cbofchn_Change()
btninf.Enabled = False
btncaltenmin.Enabled = False
btninfmin.Enabled = False
btncaltenmmax.Enabled = False
btninfmax.Enabled = False
End Sub
```

```
Private Sub chkhmin_Click()
btninfmin.Enabled = False
```

```
lblfchmin.Enabled = True
cbofchmin.Enabled = True
cbofchmin.Text = 0.18
```

```
If Val(textvelocidadmin.Text) = 0 And Val(textpresionmin.Text) = 0 Then
    lbldiahmin.Enabled = False
    cbodiahmin.Enabled = False
```

```
    cbodiahmin.Text = ""
    Else
    lbldiahmin.Enabled = True
    cbodiahmin.Enabled = True
    cbodiahmin.Text = 20.32
End If

If chkhmin.Value = Unchecked Then
    lbfchmin.Enabled = False
    cbofchmin.Enabled = False
    lbldiahmin.Enabled = False
    cbodiahmin.Enabled = False

    cbofchmin.Text = ""
    cbodiahmin.Text = ""
End If
End Sub
```

```
Private Sub chkhn_Click()
    btninf.Enabled = False
    btncaltenmin.Enabled = False
    btninfmin.Enabled = False
    btncaltenmmax.Enabled = False
    btninfmax.Enabled = False
```

```
    lbfchn.Enabled = True
    cbofchn.Enabled = True
    cbofchn.Text = 0.18
```

```
    If Val(textvelocidadn.Text) = 0 And Val(textpresionn.Text) = 0 Then
        lbldiahn.Enabled = False
        cbodiahn.Enabled = False
        cbodiahn.Text = ""
    Else
        lbldiahn.Enabled = True
        cbodiahn.Enabled = True
        cbodiahn.Text = 20.32
    End If
```

```
    If chkhn.Value = Unchecked Then
        lbfchn.Enabled = False
        cbofchn.Enabled = False
        lbldiahn.Enabled = False
        cbodiahn.Enabled = False

        cbofchn.Text = ""
        cbodiahn.Text = ""
```

```
End If
```

```
End Sub
```

```
Private Sub dbctipo_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbctipo.SelectedItem
End Sub
```

```
Private Sub Form_Activate()
dbctipo.Text = frmcalcable.texttipo
End Sub
```

```
Private Sub Form_Load()
textvano.Text = frmdatos.textvano.Text
End Sub
```

```
Private Sub textfsc_Click()
textfsc.SelStart = 0
textfsc.SelLength = Len(textfsc)
End Sub
```

```
Private Sub textfscamax_Change()
fscmax = Val(textfscamax.Text)
fsc = Val(textfsc.Text)
signo = " > "
If fscmax < fsc Then
    signo = " < "
    crimax = "En estas condiciones, el cable no cumple con el factor de seguridad
de diseño." & Chr(10) & Chr(13)
    crimax = crimax & Chr(10) & Chr(13)
    crimax = crimax & "F.S. del cable: " & fsc & " > " & Format(fscmax, "0.0000")
& Chr(10) & Chr(13)
    crimax = crimax & Chr(10) & Chr(13)
    crimax = crimax & "Por lo tanto se recomienda verificar los siguientes
parámetros:" & Chr(10) & Chr(13)
    crimax = crimax & "® Cable." & Chr(10) & Chr(13)
    crimax = crimax & "® Vano." & Chr(10) & Chr(13)
    crimax = crimax & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
    crimax = crimax & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)
    cabfa = MsgBox(crimax, vbCritical, "FLECHA MAXIMA: Factor de seguridad no
permitido (DELTA)")
End If
deffsmax = " C.S. final del cable : " & Format(fscmax, "0.0000") & signo & fsc & "
C.S. de diseño del cable "
textfscamax.ToolTipText = deffsmax
End Sub
```

```

Private Sub textfscamin_Change()
fscmin = Val(textfscamin.Text)
fsc = Val(textfsc.Text)
signo = " > "
If fscmin < fsc Then
    signo = " < "
    cri = "En estas condiciones, el cable no cumple con el factor de seguridad de
diseño." & Chr(10) & Chr(13)
    cri = cri & Chr(10) & Chr(13)
    cri = cri & "F.S. del cable: " & fsc & " > " & Format(fscmin, "0.0000") & Chr(10)
& Chr(13)
    cri = cri & Chr(10) & Chr(13)
    cri = cri & "Por lo tanto se recomienda verificar los siguientes parámetros:" &
Chr(10) & Chr(13)
    cri = cri & "® Cable." & Chr(10) & Chr(13)
    cri = cri & "® Vano." & Chr(10) & Chr(13)
    cri = cri & "® Condiciones de Sobrecarga." & Chr(10) & Chr(13)
    cri = cri & "® Temperaturas (Máx. y Mín.), etc.." & Chr(10) & Chr(13)
    cabfa = MsgBox(cri, vbCritical, "FLECHA MINIMA: Factor de seguridad no
permitido (DELTA)")
End If
deffsmin = " C.S. final del cable : " & Format(fscmin, "0.0000") & signo & fsc & "
C.S. de diseño del cable "
textfscamin.ToolTipText = deffsmin
End Sub

```

```

Private Sub textpresionmax_Click()
Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frmpacalprelcbhoff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frmpacalprelcbopesovol.Text) 'lee el valor en frmitera
End If
presionmax = Val(textpresionmax.Text)
velocidadmax = Sqr(2 * 9.81 * presionmax / (ff * q * 0.55)) * 3600 / 1000
textvelocidadmax.Text = velocidadmax

textpresionmax.SelStart = 0
textpresionmax.SelLength = Len(textpresionmax)
End Sub

```

```

Private Sub textpresionmax_Change()
btninfmax.Enabled = False
End Sub

```

```

Private Sub textpresionmin_Click()
Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frmpacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frmpacalpre!cbopesovol.Text) 'lee el valor en frmitera
End If
presionmin = Val(textpresionmin.Text)
velocidadadmin = Sqr(2 * 9.81 * presionmin / (ff * q * 0.55)) * 3600 / 1000
textvelocidadadmin.Text = velocidadadmin

textpresionmin.SelStart = 0
textpresionmin.SelLength = Len(textpresionmin)
End Sub

Private Sub textpresionmin_Change()
btninfmin.Enabled = False

If chkhmin.Value = Checked And Val(textpresionmin.Text) = 0 And
Val(textvelocidadadmin.Text) = 0 Then
    lbldiahmin.Enabled = False
    cbodiahmin.Enabled = False
    cbodiahmin.Text = ""
Else
    lbldiahmin.Enabled = True
    cbodiahmin.Enabled = True
    cbodiahmin.Text = 20.32
End If

If chkhmin.Value = Unchecked Then
    lbfchmin.Enabled = False
    cbofchmin.Enabled = False
    lbldiahmin.Enabled = False
    cbodiahmin.Enabled = False

    cbofchmin.Text = ""
    cbodiahmin.Text = ""
End If
End Sub

Private Sub textpresionn_Click()
Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frmpacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

```

```

If q = 0 Then
    q = Val(frmpacalpre!cbopesovol.Text)    'lee el valor en frmitera
End If
presionn = Val(textpresionn.Text)
velocidadn = Sqr(2 * 9.81 * presionn / (ff * q * 0.55)) * 3600 / 1000
textvelocidadn.Text = velocidadn

textpresionn.SelStart = 0
textpresionn.SelLength = Len(textpresionn)
End Sub

Private Sub textpresionn_Change()
    btninf.Enabled = False
    btncaltenmin.Enabled = False
    btninfmin.Enabled = False
    btncaltenmmax.Enabled = False
    btninfmax.Enabled = False

    Rem Activación de las etiquetas y combos
    If chkhn.Value = Checked And Val(textpresionn.Text) = 0 And
Val(textvelocidadn.Text) = 0 Then
        lbldiahn.Enabled = False
        cbodiahn.Enabled = False
        cbodiahn.Text = ""
    Else
        lbldiahn.Enabled = True
        cbodiahn.Enabled = True
        cbodiahn.Text = 20.32
    End If

    If chkhn.Value = Unchecked Then
        lbfchn.Enabled = False
        cbofchn.Enabled = False
        lbldiahn.Enabled = False
        cbodiahn.Enabled = False

        cbofchn.Text = ""
        cbodiahn.Text = ""
    End If
End Sub

Private Sub texttempmax_Click()
    texttempmax.SelStart = 0
    texttempmax.SelLength = Len(texttempmax)
End Sub

```

```
Private Sub texttempmax_Change()
btninfmax.Enabled = False
End Sub
```

```
Private Sub texttempmin_Click()
texttempmin.SelStart = 0
texttempmin.SelLength = Len(texttempmin)
End Sub
```

```
Private Sub texttempmin_Change()
btninfmin.Enabled = False
End Sub
```

```
Private Sub texttempn_Click()
texttempn.SelStart = 0
texttempn.SelLength = Len(texttempn)
End Sub
```

```
Private Sub texttempn_Change()
btninfmin.Enabled = False
btninfmax.Enabled = False
End Sub
```

```
Private Sub textvano_Click()
textvano.SelStart = 0
textvano.SelLength = Len(textvano)
End Sub
```

```
Private Sub textvelocidadmax_Click()
textvelocidadmax.SelStart = 0
textvelocidadmax.SelLength = Len(textvelocidadmax)
```

```
Rem ***** Cálculo de velocidad *****
```

```
If ff = 0 Then
    ff = Val(Mid(frmpacalprelcboff.Text, 1, 5)) 'lee el valor en frmitera
End If
```

```
If q = 0 Then
    q = Val(frmpacalprelcbopesovol.Text) 'lee el valor en frmitera
End If
```

```
velocidadmax = Val(textvelocidadmax.Text) * 1000 / 3600
presionmax = (ff * q * 0.55) * velocidadmax ^ 2 / (2 * 9.81)
textpresionmax.Text = presionmax
End Sub
```

```
Private Sub textvelocidadmax_Change()
btninfmax.Enabled = False
```

```

Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frpmacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frpmacalpre!cbopesovol.Text)    'lee el valor en frmitera
End If
velocidadmax = Val(textvelocidadmax.Text) * 1000 / 3600
presionmax = (ff * q * 0.55) * velocidadmax ^ 2 / (2 * 9.81)
textpresionmax.Text = presionmax
End Sub

Private Sub textvelocidadadmin_Click()
textvelocidadadmin.SelStart = 0
textvelocidadadmin.SelLength = Len(textvelocidadadmin)

Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frpmacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frpmacalpre!cbopesovol.Text)    'lee el valor en frmitera
End If
velocidadadmin = Val(textvelocidadadmin.Text) * 1000 / 3600
presionmin = (ff * q * 0.55) * velocidadadmin ^ 2 / (2 * 9.81)
textpresionmin.Text = presionmin
End Sub

Private Sub textvelocidadadmin_Change()
btninfmin.Enabled = False

Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frpmacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frpmacalpre!cbopesovol.Text)    'lee el valor en frmitera
End If
velocidadadmin = Val(textvelocidadadmin.Text) * 1000 / 3600
presionmin = (ff * q * 0.55) * velocidadadmin ^ 2 / (2 * 9.81)
textpresionmin.Text = presionmin

```



```

If chkxmin.Value = Checked And Val(textpresionmin.Text) = 0 Or
Val(textvelocidadn.Text) = 0 Then
    lblidiahmin.Enabled = False
    cbodiahmin.Enabled = False
    cbodiahmin.Text = ""
Else
    lblidiahmin.Enabled = True
    cbodiahmin.Enabled = True
    cbodiahmin.Text = 20.32
End If

```

```

If chkxmin.Value = Unchecked Then
    lblfchmin.Enabled = False
    cbofchmin.Enabled = False
    lblidiahmin.Enabled = False
    cbodiahmin.Enabled = False

    cbofchmin.Text = ""
    cbodiahmin.Text = ""
End If
End Sub

```

```

Private Sub textvelocidadn_Click()
textvelocidadn.SelStart = 0
textvelocidadn.SelLength = Len(textvelocidadn)

```

```

Rem ***** Cálculo de velocidad *****
If ff = 0 Then
    ff = Val(Mid(frmpacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

```

```

If q = 0 Then
    q = Val(frmpacalpre!cbopesovol.Text) 'lee el valor en frmitera
End If

```

```

velocidadn = Val(textvelocidadn.Text) * 1000 / 3600
presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn
End Sub

```

```

Private Sub textvelocidadn_Change()
btninf.Enabled = False
btncaltenmin.Enabled = False
btninfmin.Enabled = False
btncaltenmmax.Enabled = False
btninfmax.Enabled = False

```

```

Rem ***** Cálculo de velocidad *****

If ff = 0 Then
    ff = Val(Mid(frmpacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
    q = Val(frmpacalpre!cbopesovol.Text) 'lee el valor en frmitera
End If

velocidadn = Val(textvelocidadn.Text) * 1000 / 3600
presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn

If chkhn.Value = Checked And Val(textpresionn.Text) = 0 Or
Val(textvelocidadn.Text) = 0 Then
    lbldiahn.Enabled = False
    cbodiahn.Enabled = False
    cbodiahn.Text = ""
Else
    lbldiahn.Enabled = True
    cbodiahn.Enabled = True
    cbodiahn.Text = 20.32

End If

If chkhn.Value = Unchecked Then
    lb!fchn.Enabled = False
    cbofchn.Enabled = False
    lbldiahn.Enabled = False
    cbodiahn.Enabled = False

    cbofchn.Text = ""
    cbodiahn.Text = ""
End If
End Sub

```

VANOS A DESNIVEL: CALCULO DE TENSIONES Y FLECHAS

PLANTILLA DE FLECHAS (CURVAS)

(frmvanosdes.frm)

Dim tiempo As String

Rem Variables que se visualizan con msgbox en la Ec principal (flecha normal)

Dim wvn 'Peso debido al viento kgf/m

Dim whn 'Peso debido al hielo kgf/m

Dim wavhn 'Peso aparente debido a viento y hielo Kgf/m

Dim fsn 'factor de sobrecarga

Dim tdiseñonpmm 'Tensión de diseño Kgf/mm²

Dim pesocpmm 'peso del cable Kgf/(m x mm²)

Dim loncab 'Longitud del cable inicial

Dim loncabfinal 'Longitud del cable condición final

Dim tempini 'Temperatura inicial

Dim tempfin 'Temperatura final

Dim g 'Número de iteraciones ocupadas

Dim vanovir 'vano virtual

Dim fxo 'punto x donde se da la mayor flecha

'Dim xinf 'abscisa del vértice respecto al apoyo inferior

Private Sub btncal_Click()

btninf.Enabled = True

btnvisual.Enabled = True

btninflong.Enabled = True

If itera = 0 And dif = 0 Then

 itera = Val(frmitera!textitera.Text) 'lee el valor en frmitera

 dif = Val(frmitera!texterr.Text) 'lee el valor en frmitera

End If

Rem ***** CALCULO DE TENSIONES *****

Static pi

pi = 3.14159265358979

presionn = Val(textpresionn.Text)

trupn = Val(textruptura.Text)

'Tensión de ruptura (kgf)

fsc = Val(textfsc.Text)

'Factor de seguridad del cable

pesoc = Val(textpeso.Text) / 1000

'peso del cable (kgf/m)

areac = Val(textarea.Text)

'Area del conductor (mm²)

diametro = Val(textdiametro.Text) / 1000

'Diámetro del cable (m)

vano = Val(textvano.Text)

'Vano (m)

```

elasti = Val(textelastidad.Text) * 10 ^ 3 'Modulo de elasticidad
dilata = Val(textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación 1/°C
tempini = Val(textttempini.Text) 'Temperatura inicial
tempfin = Val(textttempf.Text) 'Temperatura final
desnivel = Val(textdesnivel.Text) 'Desnivel entre los apoyos

tdiseño = trupn / fsc 'Tensión de diseño normal (Kgf)
texttn.Text = tdiseño
tdiseño/mm = tdiseño / areac 'Tensión de diseño (Kgf/mm²)
msjdise = "Tensión de diseño t = " & Format(tdiseño/mm, "0.00000") & "
Kgf/mm²"
texttn.ToolTipText = msjdise

pesoc/mm = pesoc / areac 'peso del cable (Kgf/m x mm²)

If chkhn.Value = Unchecked Then
    hielo = "no"
Else
    hielo = "si"
End If

If presionn = 0 And hielo = "no" Then 'Sin viento y sin hielo
    GoTo svsh
End If

If presionn > 0 And hielo = "no" Then 'Con viento y sin hielo
    GoTo cvsh
End If

If presionn = 0 And hielo = "si" Then 'Sin viento y con hielo
    GoTo svch
End If

If presionn > 0 And hielo = "si" Then 'Con viento y con hielo
    GoTo cvch
End If

Rem *****
Rem ***** DEFINICION DE CASOS *****

svsh: 'Sin viento y sin hielo *****

    tiempo = "SIN SOBRECARGA DE VIENTO Y HIELO"
    wvn = 0 'Peso debido al viento kgf/m
    whn = 0 'Peso debido al hielo kgf/m
    wavhn = pesoc 'Peso aparente debido a viento y hielo Kgf/m
    fsn = 1 'factor de sobrecarga

```

GoTo fin

cvsh: 'Con viento y sin hielo *****

```

tiempo = "CON SOBRECARGA DE VIENTO Y SIN SOBRECARGA DE
HIELO"
wvn = presionn * diametro      'Peso debido al viento kgf/m
whn = 0                        'Peso debido al hielo kgf/m
wavhn = Sqr(pesoc ^ 2 + wvn ^ 2) 'Peso aparente debido a viento y hielo
Kgf/m
fsn = wavhn / pesoc           'factor de sobrecarga
GoTo fin

```

svch: 'Sin viento y con hielo *****

fchn = Val(cbofchn.Text)

```

tiempo = "SIN SOBRECARGA DE VIENTO Y CON SOBRECARGA DE
HIELO"
wvn = 0                        'Peso debido al viento kgf/m
whn = fchn * Sqr(diametro * 1000) 'Peso debido al hielo kgf/m
wavhn = pesoc + whn           'Peso aparente debido a viento y hielo Kgf/m
fsn = wavhn / pesoc           'factor de sobrecarga
GoTo fin

```

cvch: 'Con viento y con hielo *****

fchn = Val(cbofchn.Text)

diahn = Val(cbodiahn.Text)

```

tiempo = "CON SOBRECARGA DE VIENTO y HIELO"
wvn = presionn * (diametro + 2 * diahn / 1000) 'Peso debido al viento kgf/m
whn = fchn * Sqr(diametro * 1000)           'Peso debido al hielo kgf/m
wavhn = Sqr((pesoc + whn) ^ 2 + wvn ^ 2)     'Peso aparente debido a
viento y hielo Kgf/m
fsn = wavhn / pesoc                         'factor de sobrecarga
GoTo fin

```

fin:

wapmm = wavhn / areac 'Peso aparente (Kgf/m x mm²)

Rem *****

Rem ***** Inicio del cálculo de xinf y Tensión en el apoyo superior *****

div = Val(cbodivisor.Text)

teninicio = tdiseño / div

textinicial.Text = teninicio

tv1 = tdiseño / div 'Primera estimación

```

tvini = tv1

msjini = "Tensión inicial en el vértice tv = " & Format(tvini, "0.00000") & "
Kgf/mm2"
textinicial.ToolTipText = msjini

Rem Inicio del Loop
For g = 1 To itera
  hv1 = tv1 / wapmm
  xinf = loninf(vano, desnivel, hv1) 'Función
  'Prueba: Tensión en el apoyo superior
  xsup = vano + xinf
  pa4 = xsup / hv1
  tas = tv1 * cosh(pa4)
  'Verifica el error
  If Abs(tas - tdiseñonpmm) <= dif Then
    GoTo SALIR
  Else
    tv1 = tdiseñonpmm * tv1 / tas 'Nueva aproximación
  End If
Next g

SALIR:
texttfinalv.Text = tv1 * areac

msjfin = "Tensión final en el vértice tv = " & Format(tv1, "0.00000") & " Kgf/mm2"
texttfinalv.ToolTipText = msjfin

textxinf.Text = xinf 'Visualiza la distancia x inferior hasta el vértice

Rem Tensión en el apoyo superior
xsup = vano + xinf
pa4 = xsup / hv1
tas = tv1 * cosh(pa4)
texttas.Text = tas * areac 'Visualiza la tensión en el apoyo superior
fstas = trupn / Val(texttas.Text) 'Cálculo del factor de seguridad
texttas.ToolTipText = "Factor de Seguridad: " & Format(fstas, "0.0000") & "
Temp.: " & tempini & " °C "

Rem Tensión en el apoyo inferior
pa5 = xinf / hv1
tai = tv1 * cosh(pa5)
texttai.Text = tai * areac 'Visualiza la tensión en el apoyo inferior
fstai = trupn / Val(texttai.Text) 'Cálculo del factor de seguridad
texttai.ToolTipText = "Factor de Seguridad: " & Format(fstai, "0.0000") & "
Temp.: " & tempini & " °C "

```

Rem **** Cálculo de la longitud del cable ****

$long1 = (vano + xinf) / hv1$

$long2 = xinf / hv1$

$loncab = hv1 * (senh(long1) - senh(long2))$ 'Longitud del cable (m)

Rem *****

Rem ***** Cambio de condiciones *****

$deltalonf = dilata * Abs(tempfin - tempini) * loncab$

$loncabfinal = deltalonf + loncab$

$Z = Sqr(-6 + ((6 / vano) * Sqr(loncabfinal ^ 2 - desnivel ^ 2)))$

$htempf = vano / (2 * Z)$

$tvtempf = htempf * pesocpmm$

$xinfff = loninf(vano, desnivel, htempf)$ 'Función

Rem Tensión en el apoyo superior por cambio de condiciones

$xsuptf = vano + xinfff$

$pa6 = xsuptf / htempf$

$tastf = tvtempf * cosh(pa6)$

$texttasf.Text = tastf * areac$ 'Visualiza la tensión en el apoyo superior

$fstastf = trupn / Val(texttasf.Text)$ 'Cálculo del factor de seguridad

$texttasf.ToolTipText = "Factor de Seguridad: " & Format(fstastf, "0.0000") & "$

$Temp.: " & tempfin & " °C "$

Rem Tensión en el apoyo inferior por cambio de condiciones

$xinfff = xinfff$

$pa7 = xinfff / htempf$

$tastf = tvtempf * cosh(pa7)$

$texttaif.Text = tastf * areac$ 'Visualiza la tensión en el apoyo inferior

$fstaitf = trupn / Val(texttaif.Text)$ 'Cálculo del factor de seguridad

$texttaif.ToolTipText = "Factor de Seguridad: " & Format(fstaitf, "0.0000") & "$

$Temp.: " & tempfin & " °C "$

Rem *****

Rem Cálculo de flechas para condiciones finales

$vanovir = 2 * (vano + xinfff)$

Rem Flecha vertical del apoyo superior

$fpa1 = vanovir / (2 * htempf)$

$fastf = htempf * (cosh(fpa1) - 1)$

$textfas.Text = fastf$ 'Visualiza la flecha

$textfas.ToolTipText = " Temp. final: " & tempfin & " °C "$

Rem Flecha vertical del apoyo inferior

$faitf = fastf - desnivel$

$textfai.Text = faitf$ 'Visualiza la flecha

```

textfai.ToolTipText = " Temp. final: " & tempfin & " °C "

Rem Flecha de regulación
fxo = (vano + 2 * xinfft) / 2
fpa2 = fxo / htemf
fpa3 = vano / (2 * htemf)
fMn = htemf * cosh(fpa2) * (cosh(fpa3) - 1)
textfmm.Text = fMn      'Visualiza la flecha
textfmm.ToolTipText = " Temp. final: " & tempfin & " °C "
End Sub

Private Sub btninf_Click()
cotaxinf = Val(textxinf.Text)
If cotaxinf < 0 Then
    decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & ") < 0
: " & Chr(10) & Chr(13)
    decide = decide & "El vértice de la catenaria esta situado a la derecha del
apoyo inferior" & Chr(10) & Chr(13)
    decide = decide & "(Catenaria con vértice real entre los apoyos del vano)"
End If

If cotaxinf = 0 Then
    decide = "Abscisa X del apoyo inferior = 0 : " & Chr(10) & Chr(13)
    decide = decide & "El vértice de la catenaria coincide con el apoyo inferior" &
Chr(10) & Chr(13)
    decide = decide & "(Catenaria con vértice real en el apoyo inferior del vano)"
End If

If cotaxinf > 0 Then
    decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & ") > 0
: " & Chr(10) & Chr(13)
    decide = decide & "El vértice de la catenaria esta situado a la izquierda del
apoyo inferior" & Chr(10) & Chr(13)
    decide = decide & "(Catenaria con vértice virtual fuera del vano)"
End If

msjn = tiempo & Chr(10) & Chr(13)
msjn = msjn & Chr(10) & Chr(13)
msjn = msjn & decide & Chr(10) & Chr(13)
msjn = msjn & Chr(10) & Chr(13)
msjn = msjn & "Datos relacionados a esta condición: " & Chr(10) & Chr(13)
msjn = msjn & "Tensión de trabajo por unidad de area: to = " &
Format(tdiseñonpmm, "0.00000") & " Kgf / mm²" & Chr(10) & Chr(13)
msjn = msjn & "Peso del cable por unidad de longitud por area: w = " &
Format(pesocpmm, "0.0000000") & " Kgf / (m x mm²)" & Chr(10) & Chr(13)

```



```

    msjn = msjn & "peso ocasionado por el viento: wv = " & Format(wvn, "0.00000")
    & " Kg / m" & Chr(10) & Chr(13)
    msjn = msjn & "peso ocasionado por el hielo: wh = " & Format(whn, "0.00000")
    & " Kg / m" & Chr(10) & Chr(13)
    msjn = msjn & "peso aparente: wa = " & Format(wavhn, "0.00000") & " Kg / m"
    & Chr(10) & Chr(13)
    msjn = msjn & "Factor de sobrecarga: mo = " & Format(fsn, "0.00000") &
    Chr(10) & Chr(13)
    msjn = msjn & "vano virtual: a´ = " & Format(vanovir, "0.00000") & " m" &
    Chr(10) & Chr(13)
    msjn = msjn & "Abscisa 'x' donde se da la mayor flecha: Xm = " & Format(fxo,
    "0.00000") & " m" & Chr(10) & Chr(13)
    msjn = msjn & Chr(10) & Chr(13)
    msjn = msjn & "Máximo número de iteraciones: " & itera & Chr(10) & Chr(13)
    msjn = msjn & "Número de iteraciones ocupadas: " & g - 1
    N = MsgBox(msjn, vbExclamation, "VANOS CON APOYOS A DESNIVEL
    (DELTA)")
    End Sub

```

```

Private Sub btninflong_Click()
    msjlon = "Longitud inicial del cable: L = " & Format(loncab, "0.00000") & " m" & "
    (Temp.= " & tempini & " °C)" & Chr(10) & Chr(13)
    msjlon = msjlon & "Longitud final del cable: L = " & Format(loncabfinal,
    "0.00000") & " m" & " (Temp.= " & tempfin & " °C)"
    N = MsgBox(msjlon, vbExclamation, "VANOS CON APOYOS A DESNIVEL
    (DELTA)")
    End Sub

```

```

Private Sub btnotroscas_Click()
    Hide
    frmnunamorti.Show
    End Sub

```

```

Private Sub btnverprincipio_Click()
    Unload Me
    Principio.Show
    End Sub

```

```

Private Sub btnvfrm cables_Click()
    rastreo7 = "activo frmvanosdes"
    Hide
    frmCables.Show
    End Sub

```

```

Private Sub btnvfrmitera_Click()
    rastreo2 = "vanodesnivel" "Variable que permite retornar desde frmitera
    'al cerrarla

```

```
frmitera.Show vbModal  
End Sub
```

```
Private Sub btnvfrmprevie_Click()  
frmprevdes.Show vbModal  
End Sub
```

```
Private Sub btnvfrmten_Click()  
Hide  
frmtensiones.Show  
End Sub
```

```
Private Sub btnvinf_Click()  
msjvd = "El 'factor divisor (F.D.)' permite establecer una primera aproximación  
de la tensión en el vértice, con el cual se iniciarán las iteraciones," & Chr(10) &  
Chr(13)  
msjvd = msjvd & "para determinar la ubicación del mismo." & Chr(10) &  
Chr(13)  
msjvd = msjvd & Chr(10) & Chr(13)  
msjvd = msjvd & "Tensión en el vértice (Tv) < Tensión de diseño (Td)" & Chr(10)  
& Chr(13)  
msjvd = msjvd & "Primera aproximación:  $T_v = T_d / F.D.$ "  
infor = MsgBox(msjvd, vbInformation, "Información general (DELTA)")  
End Sub
```

```
Private Sub btnvisual_Click()  
frmvervanodes.Show vbModal  
End Sub
```

```
Private Sub cbodiahn_Click()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub cbodiahn_Change()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub cbodivisor_Click()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub cbodivisor_Change()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub cbofchn_Click()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub cbofchn_Change()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False  
End Sub
```

```
Private Sub chkhn_Click()  
btninf.Enabled = False  
btnvisual.Enabled = False  
btninflong.Enabled = False
```

```
lblfchn.Enabled = True  
cbofchn.Enabled = True  
cbofchn.Text = 0.18
```

```
If Val(textvelocidadn.Text) = 0 And Val(textpresionn.Text) = 0 Then  
    lbldiahn.Enabled = False  
    cbodiahn.Enabled = False  
    cbodiahn.Text = ""  
Else  
    lbldiahn.Enabled = True  
    cbodiahn.Enabled = True  
    cbodiahn.Text = 20.32  
End If
```

```
If chkhn.Value = Unchecked Then  
    lblfchn.Enabled = False  
    cbofchn.Enabled = False  
    lbldiahn.Enabled = False  
    cbodiahn.Enabled = False  
  
    cbofchn.Text = ""  
    cbodiahn.Text = ""  
End If  
End Sub
```

```
Private Sub dbctipo_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbctipo.SelectedItem
End Sub
```

```
Private Sub Form_Activate()
dbctipo.Text = frmcalcable.texttipo
End Sub
```

```
Private Sub Form_Load()
textvano.Text = frmtensiones.textvano.Text
textfsc.Text = frmtensiones.textfsc.Text
texttempn.Text = frmtensiones.texttempn.Text
texttempf.Text = frmtensiones.texttempmax.Text
textvelocidadn.Text = frmtensiones.textvelocidadmax.Text
End Sub
```

```
Private Sub textdesnivel_Change()
btninf.Enabled = False
btnvisual.Enabled = False
btninflong.Enabled = False
End Sub
```

```
Private Sub textpresionn_Click()
Rem ***** Cálculo de velocidad *****
If ff = 0 Then
ff = Val(Mid(frmpacalpre!cboff.Text, 1, 5)) 'lee el valor en frmitera
End If
```

```
If q = 0 Then
q = Val(frmpacalpre!cbopesovol.Text) 'lee el valor en frmitera
End If
presionn = Val(textpresionn.Text)
velocidadn = Sqr(2 * 9.81 * presionn / (ff * q * 0.55)) * 3600 / 1000
textvelocidadn.Text = velocidadn
End Sub
```

```
Private Sub textpresionn_Change()
btninf.Enabled = False
btnvisual.Enabled = False
btninflong.Enabled = False
```

```
Rem Activación de las etiquetas y combos
If chkhn.Value = Checked And Val(textpresionn.Text) = 0 And
Val(textvelocidadn.Text) = 0 Then
lbldiahn.Enabled = False
cbodiahn.Enabled = False
```

```

    cbodiahn.Text = ""
    Else
    lbldiahn.Enabled = True
    cbodiahn.Enabled = True
    cbodiahn.Text = 20.32
End If

If chkhn.Value = Unchecked Then
    lbifchn.Enabled = False
    cbofchn.Enabled = False
    lbldiahn.Enabled = False
    cbodiahn.Enabled = False

    cbofchn.Text = ""
    cbodiahn.Text = ""
End If
End Sub

Private Sub texttempf_Change()
    btninf.Enabled = False
    btnvisual.Enabled = False
    btninflong.Enabled = False
End Sub

Private Sub texttempi_Change()
    btninf.Enabled = False
    btnvisual.Enabled = False
    btninflong.Enabled = False
End Sub

Private Sub textvano_Change()
    btninf.Enabled = False
    btnvisual.Enabled = False
    btninflong.Enabled = False
End Sub

Private Sub textvelocidadn_Click()
    Rem ***** Cálculo de velocidad *****
    If ff = 0 Then
        ff = Val(Mid(frmpacalprelcboff.Text, 1, 5)) 'lee el valor en frmitera
    End If

    If q = 0 Then
        q = Val(frmpacalprelcbopesovol.Text) 'lee el valor en frmitera
    End If

    velocidadn = Val(textvelocidadn.Text) * 1000 / 3600

```

```

presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn
End Sub

Private Sub textvelocidadn_Change()
btninf.Enabled = False
btnvisual.Enabled = False
btninflong.Enabled = False

Rem ***** Cálculo de velocidad *****

If ff = 0 Then
ff = Val(Mid(frmpacalprelcboff.Text, 1, 5)) 'lee el valor en frmitera
End If

If q = 0 Then
q = Val(frmpacalprelcbopesovol.Text) 'lee el valor en frmitera
End If

velocidadn = Val(textvelocidadn.Text) * 1000 / 3600
presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn

If chkhn.Value = Checked And Val(textpresionn.Text) = 0 Or
Val(textvelocidadn.Text) = 0 Then
Ibldiahn.Enabled = False
cbodiahn.Enabled = False
cbodiahn.Text = ""
Else
Ibldiahn.Enabled = True
cbodiahn.Enabled = True
cbodiahn.Text = 20.32

End If

If chkhn.Value = Unchecked Then
Iblfchn.Enabled = False
cbofchn.Enabled = False
Ibldiahn.Enabled = False
cbodiahn.Enabled = False

cbofchn.Text = ""
cbodiahn.Text = ""
End If
End Sub

```

CALCULO DEL NUMERO DE ANTIVIBRADORES Y

DISTANCIAS

(frmnumamorti.frm)

```
Private Sub btnverinf_Click()
frmminfnumamorti.Show vbModal
End Sub
```

```
Private Sub btnverprincipio_Click()
Unload Me
Principio.Show
End Sub
```

```
Private Sub btnvfrmvanosdes_Click()
Hide
frmvanosdes.Show
End Sub
```

```
Private Sub Form_Activate()
If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6
```

```
textnum450.ToolTipText = " Número de amortiguadores/vano: " & nustoc450 &
" "
textnum600.ToolTipText = " Número de amortiguadores/vano: " & nustoc600 &
" "
textnum900.ToolTipText = " Número de amortiguadores/vano: " & nustoc950 &
" "
End Sub
```

```
Private Sub Form_Load()
Rem Lectura de datos
texttipo.Text = frm tensiones.dbctipo.Text           'Tipo de conductor
textpeso.Text = Val(frm tensiones.textpeso.Text)   'peso (kgf / Km)
textdiame.Text = Val(frm tensiones.textdiametro.Text) 'Diámetro (mm)
texttension.Text = Val(frm tensiones.texttmin.Text) 'Tensión (Kgf)
'textvano.Text = Val(frm tensiones.textvano.Text)   'vano (m)
textnumcables.Text = Val(frm datos.ncf.Text)       'Número de conductores
If frm datos.uckto.Value = True Then
textnumckto.Text = 1
Else
textnumckto.Text = 2
```

End If

If frmdatos.dckto.Value = True Then

 textnumckto.Text = 2

 Else

 textnumckto.Text = 1

End If

Pica.Picture = LoadPicture("C:\delta\Graficas\amortiguadores.wmf")

End Sub

Private Sub textdiame_Change()

Rem AMORTIGUADORES

If nustoc450 = 0 Then nustoc450 = 2

If nustoc600 = 0 Then nustoc600 = 4

If nustoc950 = 0 Then nustoc950 = 6

Rem datos

numckto = Val(textnumckto.Text)

numconfa = Val(textnumcables.Text) * 3

diame = Val(textdiame.Text)

Peso = Val(textpeso.Text)

tension = Val(texttension.Text)

Rem Cálculo PARA VANOS HASTA 450 M

numvano450 = Val(textnum450.Text)

dis1 = $13 * 10^{-4} * diame * Sqr(Abs(tension / (Peso / 1000)))$

total450 = numckto * numconfa * nustoc450 * numvano450

Rem resultados

textdis1.Text = Format(dis1, "0.0000")

texttotal450.Text = total450

Rem Cálculo PARA VANOS ENTRE 450 Y 950 M

numvano600 = Val(textnum600.Text)

dis1 = $22 * 10^{-4} * diame * Sqr(Abs(tension / (Peso / 1000)))$

total650 = numckto * numconfa * nustoc600 * numvano600

Rem resultados

textdis2.Text = Format(dis1, "0.0000")

texttotal600.Text = total650

Rem Cálculo VANOS MAYORES DE 950

numvano950 = Val(textnum900.Text)

dis1 = $27.2 * 10^{-4} * diame * Sqr(Abs(tension / (Peso / 1000)))$

total950 = numckto * numconfa * nustoc950 * numvano950

Rem resultados


```

textdis3.Text = Format(dis1, "0.0000")
texttotal900.Text = total950
End Sub

Private Sub textdis1_Change()
lbl1.ToolTipText = " Distancia del primer amortiguador: " & Val(textdis1.Text) &
" m "
End Sub

Private Sub textdis2_Change()
lbl2.ToolTipText = "Distancia del segundo amortiguador: " & textdis2.Text & "
m "
End Sub

Private Sub textdis3_Change()
lbl3.ToolTipText = "Distancia del tercer amortiguador: " & textdis3.Text & " m "
End Sub

Private Sub textnum450_Change()
'nustoc450 variables universales
'nustoc600
'nustoc950
If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6

Rem datos
numckto = Val(textnumckto.Text)
numconfa = Val(textnumcables.Text) * 3
diame = Val(textdiame.Text)
Peso = Val(textpeso.Text)
tension = Val(texttension.Text)

Rem Cálculo
numvano450 = Val(textnum450.Text)
dis1 = 13 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total450 = numckto * numconfa * nustoc450 * numvano450

Rem resultados
textdis1.Text = Format(dis1, "0.0000")
texttotal450.Text = total450
End Sub

Private Sub textnum600_Change()
'nustoc450 variables universales
'nustoc600
'nustoc950

```

```

If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6

```

```

Rem datos
numckto = Val(textnumckto.Text)
numconfa = Val(textnumcables.Text) * 3
diame = Val(textdiame.Text)
Peso = Val(textpeso.Text)
tension = Val(texttension.Text)

```

```

Rem Cálculo
numvano600 = Val(textnum600.Text)
dis1 = 22 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total650 = numckto * numconfa * nustoc600 * numvano600

```

```

Rem resultados
textdis2.Text = Format(dis1, "0.0000")
texttotal600.Text = total650
End Sub

```

```

Private Sub textnum900_Change()
'nustoc450 variables universales
'nustoc600
'nustoc950
If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6

```

```

Rem datos
numckto = Val(textnumckto.Text)
numconfa = Val(textnumcables.Text) * 3
diame = Val(textdiame.Text)
Peso = Val(textpeso.Text)
tension = Val(texttension.Text)

```

```

Rem Cálculo
numvano950 = Val(textnum900.Text)
dis1 = 27.2 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total950 = numckto * numconfa * nustoc950 * numvano950

```

```

Rem resultados
textdis3.Text = Format(dis1, "0.0000")
texttotal900.Text = total950
End Sub

```

```

Private Sub textnumcables_Change()

```

```

ncfa = Int(Val(textnumcables.Text))
textnumcables.Text = ncfa
Select Case ncfa
  Case Is >= 6
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
    GoTo Line2

  Case Is = 0
    def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
    textnumcables.Text = 1

  Case Is < 1
    def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncfa)), 4,
"Parámetro inaceptable")
    If def = vbYes Then
      textnumcables.Text = Abs(Int(textnumcables.Text))
Line2:
    Else
      GoTo Line1
    End If
End Select
End Select

Rem *****
Rem AMORTIGUADORES
If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6

Rem datos
numckto = Val(textnumckto.Text)
numconfa = Val(textnumcables.Text) * 3
diame = Val(textdiame.Text)
Peso = Val(textpeso.Text)
tension = Val(texttension.Text)

Rem Cálculo PARA VANOS HASTA 450 M
numvano450 = Val(textnum450.Text)
dis1 = 13 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total450 = numckto * numconfa * nustoc450 * numvano450

Rem resultados
textdis1.Text = Format(dis1, "0.0000")
texttotal450.Text = total450

```

Rem Cálculo PARA VANOS ENTRE 450 Y 950 M

```
numvano600 = Val(textnum600.Text)
dis1 = 22 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total650 = numckto * numconfa * nustoc600 * numvano600
```

Rem resultados

```
textdis2.Text = Format(dis1, "0.0000")
texttotal600.Text = total650
```

Rem Cálculo VANOS MAYORES DE 950

```
numvano950 = Val(textnum900.Text)
dis1 = 27.2 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total950 = numckto * numconfa * nustoc950 * numvano950
```

Rem resultados

```
textdis3.Text = Format(dis1, "0.0000")
texttotal900.Text = total950
End Sub
```

Private Sub textnumckto_Change()

```
numckto = Val(textnumckto.Text)
textnumckto.Text = Int(Val(textnumckto.Text))
```

If numckto >= 3 Then

```
MsgBox "El número de circuitos que se ha elegido es mayor que dos"
End If
```

Rem AMORTIGUADORES

```
If nustoc450 = 0 Then nustoc450 = 2
If nustoc600 = 0 Then nustoc600 = 4
If nustoc950 = 0 Then nustoc950 = 6
```

Rem datos

```
numckto = Val(textnumckto.Text)
numconfa = Val(textnumcables.Text) * 3
diame = Val(textdiame.Text)
Peso = Val(textpeso.Text)
tension = Val(texttension.Text)
```

Rem Cálculo PARA VANOS HASTA 450 M

```
numvano450 = Val(textnum450.Text)
dis1 = 13 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total450 = numckto * numconfa * nustoc450 * numvano450
```

Rem resultados

```
textdis1.Text = Format(dis1, "0.0000")
```

```
texttotal450.Text = total450
```

```
Rem Cálculo PARA VANOS ENTRE 450 Y 950 M
```

```
numvano600 = Val(textnum600.Text)
```

```
dis1 = 22 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
```

```
total650 = numckto * numconfa * nustoc600 * numvano600
```

```
Rem resultados
```

```
textdis2.Text = Format(dis1, "0.0000")
```

```
texttotal600.Text = total650
```

```
Rem Cálculo VANOS MAYORES DE 950
```

```
numvano950 = Val(textnum900.Text)
```

```
dis1 = 27.2 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
```

```
total950 = numckto * numconfa * nustoc950 * numvano950
```

```
Rem resultados
```

```
textdis3.Text = Format(dis1, "0.0000")
```

```
texttotal900.Text = total950
```

```
End Sub
```

```
Private Sub textpeso_Change()
```

```
Rem AMORTIGUADORES
```

```
If nustoc450 = 0 Then nustoc450 = 2
```

```
If nustoc600 = 0 Then nustoc600 = 4
```

```
If nustoc950 = 0 Then nustoc950 = 6
```

```
Rem datos
```

```
numckto = Val(textnumckto.Text)
```

```
numconfa = Val(textnumcables.Text) * 3
```

```
diame = Val(textdiame.Text)
```

```
Peso = Val(textpeso.Text)
```

```
tension = Val(texttension.Text)
```

```
Rem Cálculo PARA VANOS HASTA 450 M
```

```
numvano450 = Val(textnum450.Text)
```

```
dis1 = 13 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
```

```
total450 = numckto * numconfa * nustoc450 * numvano450
```

```
Rem resultados
```

```
textdis1.Text = Format(dis1, "0.0000")
```

```
texttotal450.Text = total450
```

```
Rem Cálculo PARA VANOS ENTRE 450 Y 950 M
```

```
numvano600 = Val(textnum600.Text)
```

```
dis1 = 22 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
```

```
total650 = numckto * numconfa * nustoc600 * numvano600
```

Rem resultados

textdis2.Text = Format(dis1, "0.0000")

texttotal600.Text = total650

Rem Cálculo VANOS MAYORES DE 950

numvano950 = Val(textnum900.Text)

dis1 = $27.2 * 10^{-4} * \text{diame} * \text{Sqr}(\text{Abs}(\text{tension} / (\text{Peso} / 1000)))$

total950 = numckto * numconfa * nustoc950 * numvano950

Rem resultados

textdis3.Text = Format(dis1, "0.0000")

texttotal900.Text = total950

End Sub

Private Sub texttension_Change()

Rem AMORTIGUADORES

If nustoc450 = 0 Then nustoc450 = 2

If nustoc600 = 0 Then nustoc600 = 4

If nustoc950 = 0 Then nustoc950 = 6

Rem datos

numckto = Val(textnumckto.Text)

numconfa = Val(textnumcables.Text) * 3

diame = Val(textdiame.Text)

Peso = Val(textpeso.Text)

tension = Val(texttension.Text)

Rem Cálculo PARA VANOS HASTA 450 M

numvano450 = Val(textnum450.Text)

dis1 = $13 * 10^{-4} * \text{diame} * \text{Sqr}(\text{Abs}(\text{tension} / (\text{Peso} / 1000)))$

total450 = numckto * numconfa * nustoc450 * numvano450

Rem resultados

textdis1.Text = Format(dis1, "0.0000")

texttotal450.Text = total450

Rem Cálculo PARA VANOS ENTRE 450 Y 950 M

numvano600 = Val(textnum600.Text)

dis1 = $22 * 10^{-4} * \text{diame} * \text{Sqr}(\text{Abs}(\text{tension} / (\text{Peso} / 1000)))$

total650 = numckto * numconfa * nustoc600 * numvano600

Rem resultados

textdis2.Text = Format(dis1, "0.0000")

texttotal600.Text = total650

Rem Cálculo VANOS MAYORES DE 950

```
numvano950 = Val(textnum900.Text)
dis1 = 27.2 * 10 ^ -4 * diame * Sqr(Abs(tension / (Peso / 1000)))
total950 = numckto * numconfa * nustoc950 * numvano950
```

```
Rem resultados
textdis3.Text = Format(dis1, "0.0000")
texttotal900.Text = total950
End Sub
```

```
Private Sub texttotal450_Change()
namo1 = Val(texttotal450.Text)
namo2 = Val(texttotal600.Text)
namo3 = Val(texttotal900.Text)
total = namo1 + namo2 + namo3
texttotal.Text = total
End Sub
```

```
Private Sub texttotal600_Change()
namo1 = Val(texttotal450.Text)
namo2 = Val(texttotal600.Text)
namo3 = Val(texttotal900.Text)
total = namo1 + namo2 + namo3
texttotal.Text = total
End Sub
```

```
Private Sub texttotal900_Change()
namo1 = Val(texttotal450.Text)
namo2 = Val(texttotal600.Text)
namo3 = Val(texttotal900.Text)
total = namo1 + namo2 + namo3
texttotal.Text = total
End Sub
```

TIPOS DE CABLES

(frmCables)

Private slastrecord As String

Private Sub ircontac_Click()

Unload Me

frmContactos.Show

End Sub

Private Sub btniracapac_Click()

Hide

Capac.Show

End Sub

Private Sub btniraindut_Click()

Hide

Induct.Show

End Sub

Private Sub btniraresi_Click()

Hide

resi.Show

End Sub

Private Sub btntabla_Click()

Hide

frmgridcables.Show

End Sub

Private Sub cmdadd_Click()

slastrecord = dtacables.Recordset.Bookmark

cmdadd.Enabled = False

cmdsave.Enabled = True

cmdcancel.Enabled = True

dtacables.Recordset.AddNew

dbccables.Enabled = True

Me!dbccables.SetFocus

End Sub

Private Sub cmdcancel_Click()

cmdcancel.Enabled = False


```

cmdsave.Enabled = False
cmdadd.Enabled = True
dtacables.Recordset.Bookmark = slastrecord
End Sub

```

```

Private Sub cmddelete_Click()
msg = "Seguro que desea borrar todos los datos presentados en pantalla y el
registro en la base de datos del cable " & Me!dbccables.Text & " ?"
If MsgBox(msg, vbQuestion + vbYesNo, "Borrar registro de base de datos") =
vbYes Then
dtacables.Recordset.Delete
dtacables.Recordset.MovePrevious
End If
End Sub

```

```

Private Sub cmdsave_Click()
dtacables.Recordset.Update
cmdsave.Enabled = False
cmdcancel.Enabled = False
cmdadd.Enabled = True
End Sub

```

```

Private Sub dbccables_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbccables.SelectedItem
End Sub

```

```

Private Sub Form_Activate()
dbccables.Text = dtacables.Recordset("TIPO")
End Sub

```

```

Private Sub Form_Load()
If rastreo5 = "activo Cab frmcalcable" Or rastreo6 = "activo frm tensiones" Or
rastreo7 = "activo frmvanosdes" Then
btniraresi.Enabled = False
btniraindut.Enabled = False
btniracapac.Enabled = False
End If
End Sub

```

```

Private Sub out_Click()
If rastreo5 = "activo Cab frmcalcable" Then
rastreo5 = ""
btniraresi.Enabled = True
btniraindut.Enabled = True
btniracapac.Enabled = True
Unload Me
frmcalcable.Show

```

```
    GoTo fin
End If

If rastreo6 = "activo frm tensiones" Then
    rastreo6 = ""
    btniraresi.Enabled = True
    btniraindut.Enabled = True
    btniracapac.Enabled = True
    Unload Me
    frm tensiones.Show
    GoTo fin
End If

If rastreo7 = "activo frm vanosdes" Then
    rastreo7 = ""
    btniraresi.Enabled = True
    btniraindut.Enabled = True
    btniracapac.Enabled = True
    Unload Me
    frm vanosdes.Show
    GoTo fin
End If
Unload Me
Param.Show
fin:
End Sub
```

TABLA DE TIPOS DE CABLES

(frmgridcables.frm)

```
Private Sub btniracables_Click()
Hide
frmCables.Show vbModal
End Sub

Private Sub dbctipos_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbctipos.SelectedItem
End Sub

Private Sub Form_Activate()
dbctipos.Text = dtacables.Recordset("TIPO")
End Sub

Private Sub salirfrmgridcables_Click()
If rastreo5 = "activo Cab frmcalcable" Then
    rastreo5 = ""
    Unload Me
    frmcalcable.Show
    GoTo fin
End If

If rastreo6 = "activo frm tensiones" Then
    rastreo6 = ""
    Unload Me
    frm tensiones.Show
    GoTo fin
End If

If rastreo7 = "activo frmvanosdes" Then
    rastreo7 = ""
    Unload Me
    frmvanosdes.Show
    GoTo fin
End If
Unload Me
Param.Show
fin:
End Sub
```

INTERACCIÓN DE LAS LÍNEAS DE TRANSMISIÓN CON LOS SISTEMAS ELÉCTRICOS DE POTENCIA: PARÁMETROS ELÉCTRICOS

RESISTENCIA

(Resi.frm)

```
Dim resi
Private Sub C_Click()
Unload Me
Capac.Show
End Sub
```

```
Private Sub L_Click()
Unload Me
Induct.Show
End Sub
```

```
Private Sub btnawg_Click()
areamm = InputBox("Los mm2 del conductor son: ", "AWG")
```

```
If areamm = "" Then
  GoTo fin
Else
  diame = Val(1973.5 * areamm / 1000)
End If
textdiame.Text = diame
```

```
coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugosi.Text, 1, 6))
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi
```

```
coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
```

```
fin:
End Sub
```

```
Private Sub btndiame_Click()
diame = (InputBox("El Area del cable a utilizar (KCMIL o MCM) :", "Cálculo de la  
resistencia del cable"))
```

```

If diame = "" Then
    GoTo fin
End If
textdiame.Text = diame

coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugo.Text, 1, 6))
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
fin:
End Sub

Private Sub btniracables_Click()
Hide
frmCables.Show
End Sub

Private Sub btnresi_Click()
resi = (InputBox("La resistencia en Ohms/Km :", "Resistencia del cable"))
textresi.Text = resi

coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugo.Text, 1, 6))
If resi = "" Then
    MsgBox "No se ha registrado dato de resistencia, presione el botón
nuevamente"
    GoTo coco
Else
diame = Val((coefresis * rugosi / resi) * (100 / 30.48)) 'formula
textdiame.Text = diame
coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End If
coco:
End Sub

Private Sub coefresi_Click()
coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugo.Text, 1, 6))

```

```

diame = Val(textdiame.Text)
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi

```

```

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub

```

```

Private Sub coefresi_Change()
coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugosi.Text, 1, 6))
diame = Val(textdiame.Text)
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi

```

```

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub

```

```

Private Sub coeftemp_Click()
resi = Val(textresi.Text)

```

```

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
texttemp.SetFocus
End Sub

```

```

Private Sub coeftemp_Change()
resi = Val(textresi.Text)

```

```

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
texttemp.SetFocus
End Sub

```

```

Private Sub Form_Activate()
texttipo.Text = frmCables!texttipo.Text
textmcm.Text = frmCables!textmcm.Text
textrdc20.Text = frmCables!textrdc20.Text

```

```
textrdc50.Text = frmCables!textrdc50.Text
textrac20.Text = frmCables!textrac20.Text
textrac50.Text = frmCables!textrac50.Text
textrac75.Text = frmCables!textrac75.Text
```

```
textresi.Text = Val(textrac75.Text)
```

```
textrac75nc.Text = Val(textresitemp.Text / ncfa)
End Sub
```

```
Private Sub Form_Deactivate()
rac = Val(textrac75nc.Text)
End Sub
```

```
Private Sub Form_Load()
If rastreo3 = "activo R frmcalcable" Then
    iracapac.Enabled = False
    irainduct.Enabled = False
    iractes.Enabled = False
    btniracables.Enabled = False
End If
```

```
textrac75nc.Text = Val(textresitemp.Text)
End Sub
```

```
Private Sub iracapac_Click()
Hide
Capac.Show
End Sub
```

```
Private Sub iractes_Click()
Hide
ctes.Show
End Sub
```

```
Private Sub irainduct_Click()
Hide
Induct.Show
End Sub
```

```
Private Sub ncf_Click()
ncfa = Int(Val(ncf.Text))
ncf.Text = ncfa
```

```
Select Case ncfa
    Case Is = 0
```

```

    def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")

```

```

Line1:

```

```

    ncf.Text = 1

```

```

    Case Is < 1

```

```

        def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")

```

```

        If def = vbYes Then

```

```

            ncf.Text = Abs(Int(ncf.Text))

```

```

        Else

```

```

            GoTo Line1

```

```

        End If

```

```

    End Select

```

```

Rem Cálculo de R equivalente para n conductores/fase

```

```

textrac75nc.Text = textresitemp.Text / ncfa

```

```

rac = textrac75nc.Text

```

```

End Sub

```

```

Private Sub ncf_Change()

```

```

    ncfa = Int(Val(ncf.Text))

```

```

    ncf.Text = ncfa

```

```

Select Case ncfa

```

```

    Case Is >= 6

```

```

        mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")

```

```

        GoTo Line2

```

```

    Case Is = 0

```

```

        def = MsgBox("No existen Subconductores, por lo menos debe de existir un
conductor ", 48, "Valor negativo ")

```

```

Line1:

```

```

    ncf.Text = 1

```

```

    Case Is < 1

```

```

        def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")

```

```

        If def = vbYes Then

```

```

            ncf.Text = Abs(Int(ncf.Text))

```

```

Line2:

```

```

        Else

```

```

            GoTo Line1

```



```
End If
End Select
```

```
Rem Cálculo de R equivalente para n conductores/fase
textrac75nc.Text = textresitemp.Text / ncfa
```

```
rac = textrac75nc.Text
End Sub
```

```
Private Sub rugo_Click()
rugosi = Val(Mid(rugo.Text, 1, 6))
coefresis = Val(Mid(coefresi.Text, 1, 6))
diame = Val(textdiame.Text)
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi
```

```
coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub
```

```
Private Sub rugo_Change()
rugosi = Val(Mid(rugo.Text, 1, 6))
coefresis = Val(Mid(coefresi.Text, 1, 6))
diame = Val(textdiame.Text)
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi
```

```
coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub
```

```
Private Sub salirresi_Click()
If rastreo3 = "activo R frmcalcable" Then
    rastreo3 = ""
    iracapac.Enabled = True
    irainduct.Enabled = True
    iractes.Enabled = True
    btniracables.Enabled = True
    Unload Me
    frmcalcable.Show
    GoTo fin
End If
Unload Me
```

```

Param.Show
fin:
End Sub

Private Sub spinnu_SpinDown()
Dim dblrate
dblrate = Val(ncf.Text)
If dblrate > 1 Then
    ncf.Text = dblrate - 1
Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub

Private Sub spinnu_SpinUp()
ncf.Text = (Val(ncf.Text)) + 1
If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh
End Sub

Private Sub spintemp_SpinDown()
Dim dblrate
dblrate = textemp.Text
textemp.Text = dblrate - CDec(Mid(cbopor.Text, 2))
textemp.Refresh
End Sub

Private Sub spintemp_SpinUp()
textemp.Text = (textemp.Text) + CDec(Mid(cbopor.Text, 2))
textemp.Refresh
End Sub

Private Sub textdiame_Click()
diame = Val(textdiame.Text)

coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugo.Text, 1, 6))
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(textemp.Text)

```

```

resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub

Private Sub textdiame_Change()
diame = Val(textdiame.Text)

coefresis = Val(Mid(coefresi.Text, 1, 6))
rugosi = Val(Mid(rugo.Text, 1, 6))
resi = Val((coefresis * rugosi / diame) * (100 / 30.48)) 'formula
textresi.Text = resi

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub

Private Sub textemp_Click()
resi = Val(textresi.Text)

coeftempe = Val(Mid(coeftemp.Text, 1, 6))
temp = Val(texttemp.Text)
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
textresitemp.Text = resitemp
End Sub

Private Sub textemp_Change()
resi = Val(textresi.Text)

If resi = "" Then
    GoTo mandarina
    MsgBox "No se ha registrado ningún valor de resistencia, presione el botón
nuevamente"
    textresi.SetFocus
Else
    coeftempe = Val(Mid(coeftemp.Text, 1, 6))
    temp = Val(texttemp.Text)
    resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula
    textresitemp.Text = resitemp
End If

mandarina:
End Sub

Private Sub textresi_Click()
resi = Val(textresi.Text)

```

```
coefresis = Val(Mid(coefresi.Text, 1, 6))  
rugosi = Val(Mid(rugo.Text, 1, 6))  
diame = Val((coefresis * rugosi / resi) * (100 / 30.48)) 'formula  
textdiame.Text = diame
```

```
coeftempe = Val(Mid(coeftemp.Text, 1, 6))  
temp = Val(textemp.Text)  
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula  
textresitemp.Text = resitemp  
End Sub
```

```
Private Sub textresi_Change()  
resi = Val(textresi.Text)
```

```
coefresis = Val(Mid(coefresi.Text, 1, 6))  
rugosi = Val(Mid(rugo.Text, 1, 6))  
diame = Val((coefresis * rugosi / resi) * (100 / 30.48)) 'formula  
textdiame.Text = diame
```

```
coeftempe = Val(Mid(coeftemp.Text, 1, 6))  
temp = Val(textemp.Text)  
resitemp = Val(resi * (coeftempe + temp) / (coeftempe + 20)) 'formula  
textresitemp.Text = resitemp  
End Sub
```

```
Private Sub textresitemp_Click()  
textrac75nc.Text = textresitemp.Text  
End Sub
```

```
Private Sub textresitemp_Change()  
If ncfa = 0 Then  
    ncfa = 1  
Else  
    textrac75nc.Text = Val(textresitemp.Text / ncfa)  
End If  
End Sub
```

INDUCTANCIA

(Induct.frm)

```
Private Sub Command1_Click()
frmverdis.Show
End Sub
```

```
Sub declaracion()
Static evita As Integer
End Sub
```

```
Private Sub btncálculo_Click()
Static pi
pi = 3.14159265358979
```

```
If uckto = True Then
```

```
    ***** inicio para un ckto *****
```

```
    fre = Val(Mid(textfrec.Text, 1, 4))
    rmg = (Val(textrmg.Text)) / 1000
```

```
    If ncf.Text = 1 Then
        textdis.Text = ""
    End If
```

```
    dab = (dat1ckto!textdab.Text)
```

```
    If dat1ckto!optsime.Value = True Then
        dmng = dab
    Else
        dab = (dat1ckto!textdab.Text)
        dbc = (dat1ckto!textdbc.Text)
        dca = (dat1ckto!textdca.Text)
        dmng = (dab * dbc * dca) ^ (1 / 3)
    End If
```

```
    If Val(textdis.Text) = 0 Then
```

```
        rmgg = rmg
        requi = 0
        GoTo Line3
```

```
    Else
        dis = (Val(textdis.Text)) / 100
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
```

```

    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    GoTo Line3
End If
Line3:
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
Else
    ***** inicio de dos ckto *****

    fre = Mid(textfreq.Text, 1, 4)
    rmg = (textrmg.Text) / 1000
    dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If

    If textdis.Text = "" Then
        rmgg = rmg
        requi = 0
        rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
        GoTo Line5
    Else
        dis = (textdis.Text) / 100
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
        'aumento por el caso de 2ckts:
        rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    End If
Line5:

    If rmgdckto = 0 Then
        End If

    L = 0.2 * Log(dmg / rmgdckto)
    XL = 2 * pi * fre * L * 10 ^ -3
    textdmg.Text = dmg
    textrmgg.Text = rmgdckto * 1000
    textrequi.Text = requi * 1000

```

```
    textL.Text = L
    textXL.Text = XL
End If
End Sub
```

```
Private Sub dbccables_Click(Area As Integer)
dtacables.Recordset.Bookmark = dbccables.SelectedItem
End Sub
```

```
Private Sub btniracables_Click()
Hide
frmCables.Show
End Sub
```

```
Private Sub btnverdis_Click()
frmverdis.Show
End Sub
```

```
Private Sub dckto_Click()
If evita = 4 Then
    GoTo final
End If
```

```
evita = 2
```

```
If dckto.Value = True Then
    btncálculo.Enabled = True
```

```
    Capacdckto.Value = True 'Establece la opt de dckto en la frm de capac
```

```
    lblckto.Visible = True
    lbluckto.Visible = False
    Capacdcktolab.Visible = True
    Capaclucktolab.Visible = False
```

```
    lblasime.Visible = False
    lblsime.Visible = False
    Capac!lblasime.Visible = False
    Capac!lblsime.Visible = False
```

```
    btnverdis.Enabled = True
    Capac!btnverdis.Enabled = True
```

```
    dat2ckto.Show vbModal 'Despliega la frm de datos
Else
    lblckto.Visible = False
    lbluckto.Visible = True
```

```

Capac!dcktolab.Visible = False
Capac!ucktolab.Visible = True

    btnverdis.Enabled = False
    Capac!btnverdis.Enabled = False
End If

If dckto.Value = False Then
    Capac!dckto.Value = False 'Desabilita la opt de uckto en la frm de capac
    lbdckto.Visible = False
    lbluckto.Visible = True
    Capac!dcktolab.Visible = False
    Capac!ucktolab.Visible = True

    btnverdis.Enabled = False
    Capac!btnverdis.Enabled = False
    Else
    lbdckto.Visible = True
    lbluckto.Visible = False
    Capac!dcktolab.Visible = True
    Capac!ucktolab.Visible = False

    btnverdis.Enabled = True
    Capac!btnverdis.Enabled = True
End If
final:
End Sub

Private Sub dckto_Db!Click()
    btncálculo.Enabled = True
    dat2ckto.Show vbModal
End Sub

Private Sub Form_Activate()
    texttipo.Text = frmCables!texttipo.Text
    textmcm.Text = frmCables!textmcm.Text
    texttrenzado.Text = frmCables!texttrenzado.Text
    textrmg.Text = frmCables!textrmg.Text

    evita = 1
End Sub

Private Sub Form_Deactivate()
    L = Val(textL.Text)
End Sub

Private Sub Form_Load()

```



```
If rastreo4 = "activo L frmcalcable" Then
    iracapac.Enabled = False
    irairesi.Enabled = False
    iractes.Enabled = False
    btniracables.Enabled = False
End If

End Sub

Private Sub iracapac_Click()
Hide
Capac.Show
End Sub

Private Sub iractes_Click()
Hide
ctes.Show
End Sub

Private Sub irairesi_Click()
Hide
resi.Show
End Sub

Private Sub ncf_Click()
ncfa = Int(ncf.Text)
ncf.Text = ncfa

Select Case ncfa

    Case Is = 0
        def = MsgBox("No existen Subconductores, por lo menos debe de existir
un conductor ", 48, "Valor negativo ")
Line1:
        ncf.Text = 1

    Case Is < 1
        def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
        If def = vbYes Then
            ncf.Text = Abs(Int(ncf.Text))
            frec.SetFocus
        Else
            GoTo Line1
        End If
    End Select
End Sub
```

```

Rem *****
Rem ***** Cálculo de la inductancia *****

Static pi
pi = 3.14159265358979

If uckto = True Then

    ***** inicio para un ckto *****

    fre = Val(Mid(textfreq.Text, 1, 4))
    rmg = (Val(textrmg.Text)) / 1000

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If
    dab = (dat1ckto!textdab.Text)

    If dat1ckto!optsime.Value = True Then
        dmg = dab
    Else
        dab = (dat1ckto!textdab.Text)
        dbc = (dat1ckto!textdbc.Text)
        dca = (dat1ckto!textdca.Text)
        dmg = (dab * dbc * dca) ^ (1 / 3)
    End If

    If textdis.Text = "" Then
        rmgg = rmg
        requi = 0
        GoTo Line3
    Else
        dis = (Val(textdis.Text)) / 100
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
        existe más de un conductor por fase
        GoTo Line3
    End If
Line3:
    L = 0.2 * Log(dmg / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textdmg.Text = dmg
    textrmgg.Text = rmgg * 1000
    textrequi.Text = requi * 1000
    textL.Text = L
    textXL.Text = XL
Else

```

```

***** inicio de dos ckto *****

fre = Mid(textfrec.Text, 1, 4)
rmg = (textrmg.Text) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

If textdis.Text = "" Then
    rmgg = rmg
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line5
Else
    dis = (textdis.Text) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
    existe más de un conductor por fase

    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If
Line5:

If rmgdckto = 0 Then
End If

If dmg = 0 Then
    msj = "Defina el número de circuitos por estructura. !"
    MsgBox msj
    uckto.SetFocus
    GoTo fin
End If

L = 0.2 * Log(dmg / rmgdckto)
XL = 2 * pi * fre * L * 10 ^ -3
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
End If
fin:
End Sub

```

```

Private Sub ncf_Change()
ncfa = Int(Val(ncf.Text))
ncf.Text = ncfa

If ncf.Text = 1 Then
    textdis.Text = ""
End If

Select Case ncfa

    Case Is >= 6
        mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
        GoTo Line2

    Case Is = 0
        def = MsgBox("Por lo menos debe de existir un conductor ", 48, "Definición
de subconductores ")
Line1:
        ncf.Text = 1

    Case Is < 1
        def = MsgBox("Esta de acuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
        If def = vbYes Then
            ncf.Text = Abs(Int(ncf.Text))
Line2:
            Else
                GoTo Line1
            End If
        End Select

If ncfa >= 2 Then
    lbldis.Enabled = True
    textdis.Enabled = True
    textdis.SetFocus
Else
    lbldis.Enabled = False
    textdis.Enabled = False
End If

Rem *****
Rem ***** Cálculo de la inductancia *****

Static pi
pi = 3.14159265358979

```

If uckto = True Then

***** inicio para un ckto *****

fre = Val(Mid(textfrec.Text, 1, 4))
 rmg = (Val(textrmg.Text)) / 1000

If ncf.Text = 1 Then
 textdis.Text = ""

End If

dab = (dat1ckto!textdab.Text)

If dat1ckto!optsime.Value = True Then

dmg = dab

Else

dab = (dat1ckto!textdab.Text)

dbc = (dat1ckto!textdbc.Text)

dca = (dat1ckto!textdca.Text)

dmg = (dab * dbc * dca) ^ (1 / 3)

End If

If textdis.Text = "" Then

rmgg = rmg

requi = 0

GoTo Line3

Else

dis = (Val(textdis.Text)) / 100

requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente

rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si

existe más de un conductor por fase

GoTo Line3

End If

Line3:

L = 0.2 * Log(dmg / rmgg)

XL = 2 * pi * fre * L * 10 ^ -3

textdmg.Text = dmg

textrmgg.Text = rmgg * 1000

textrequi.Text = requi * 1000

textL.Text = L

textXL.Text = XL

Else

***** inicio de dos ckto *****

fre = Mid(textfrec.Text, 1, 4)

rmg = (textrmg.Text) / 1000

12) $dmg = (dab^2 * dac^2 * dab2^2 * dba2^2 * dbc^2 * dac2 * dca2)^{1/6}$

```
If ncf.Text = 1 Then
    textdis.Text = ""
End If
```

```
If textdis.Text = "" Then
    rmgg = rmg
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line5
Else
    dis = (textdis.Text) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
    existe más de un conductor por fase
```

```
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If
```

Line5:

```
If rmgdckto = 0 Then
End If
```

```
If dmng = 0 Then
    msj = "Defina el número de circuitos por estructura. !"
    MsgBox msj
    uckto.SetFocus
    GoTo fin
End If
```

```
L = 0.2 * Log(dmng / rmgdckto)
XL = 2 * pi * fre * L * 10 ^ -3
textdmng.Text = dmng
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
```

```
End If
```

fin:

```
End Sub
```

```
Private Sub salirindut_Click()
    Hide
    Param.Show
End Sub
```

```

Private Sub salircapac_Click()
If rastreo4 = "activo L frmcalcable" Then
    rastreo4 = ""
    iracapac.Enabled = True
    irairesi.Enabled = True
    iractes.Enabled = True
    btniracables.Enabled = True
    Unload Me
    frmcalcable.Show
    GoTo fin
End If
Unload Me
Param.Show
fin:
End Sub

Private Sub spinnu_SpinDown()
Dim dblrate
dblrate = ncf.Text

If dblrate > 1 Then
    ncf.Text = dblrate - 1
Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub

Private Sub spinnu_SpinUp()
ncf.Text = (ncf.Text) + 1
If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh
End Sub

Private Sub textvedmg_Click()
textvedmg.Text = dmg
'Print dmg
End Sub

Private Sub textvedmg_Change()
textvedmg.Text = dmg
'Print dmg

```

```

End Sub

Private Sub textdis_Click()
Static pi
pi = 3.14159265358979

If uckto = True Then

    ***** inicio para un ckto *****

    fre = Val(Mid(textfrec.Text, 1, 4))
    rmg = (Val(textrmg.Text)) / 1000

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If
    dab = (dat1ckto!textdab.Text)

    If dat1ckto!optsime.Value = True Then
        dmng = dab
    Else
        dab = (dat1ckto!textdab.Text)
        dbc = (dat1ckto!textdbc.Text)
        dca = (dat1ckto!textdca.Text)
        dmng = (dab * dbc * dca) ^ (1 / 3)
    End If

    If Val(textdis.Text) = 0 Then
        rmgg = rmg
        requi = 0
        GoTo Line3
    Else
        dis = (Val(textdis.Text)) / 100
        requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
        rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
        existe más de un conductor por fase
        GoTo Line3
    End If
Line3:
    L = 0.2 * Log(dmng / rmgg)
    XL = 2 * pi * fre * L * 10 ^ -3
    textdmng.Text = dmng
    textrmgg.Text = rmgg * 1000
    textrequi.Text = requi * 1000
    textL.Text = L
    textXL.Text = XL
Else

```



```

***** inicio de dos ckto *****

fre = Mid(textfrec.Text, 1, 4)
rmg = (textrmg.Text) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

If textdis.Text = "" Then
    rmgg = rmg
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line5
Else
    dis = (textdis.Text) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
    existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If
Line5:

If rmgdckto = 0 Then
End If

L = 0.2 * Log(dmg / rmgdckto)
XL = 2 * pi * fre * L * 10 ^ -3
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
End If
End Sub

Private Sub textdis_Change()
Static pi
pi = 3.14159265358979

If uckto = True Then

***** inicio para un ckto *****

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
rmg = (Val(textrmg.Text)) / 1000

If ncf.Text = 1 Then
    textdis.Text = ""
End If
dab = (dat1ckto!textdab.Text)

If dat1ckto!optsime.Value = True Then
    dmg = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If Val(textdis.Text) = 0 Then
    rmgg = rmg
    requi = 0
    GoTo Line3
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
    existe más de un conductor por fase
    GoTo Line3
End If
Line3:
L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
Else
    ***** inicio de dos ckto *****

    fre = Mid(textfrec.Text, 1, 4)
    rmg = (textrmg.Text) / 1000
    dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 /
12)

    If ncf.Text = 1 Then
        textdis.Text = ""
    End If

```

```

If textdis.Text = "" Then
    rmgg = rmg
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line5
Else
    dis = (textdis.Text) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * rmg * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si
existe más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If
Line5:
If rmgdckto = 0 Then
End If

L = 0.2 * Log(dmg / rmgdckto)
XL = 2 * pi * fre * L * 10 ^ -3
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textL.Text = L
textXL.Text = XL
End If

End Sub

Private Sub textdmg_Click()
Static pi
pi = 3.14159265358979
fre = Val(Mid(textfrec.Text, 1, 4))

textrmgg.Text = textrmg.Text
dmg = Val(textdmg.Text)
rmgg = Val(textrmgg.Text) / 1000

If dmg = 0 Then
    textdmg.SetFocus
    GoTo fin
End If

L = 0.2 * Log(dmg / rmgg)
XL = 2 * pi * fre * L * 10 ^ -3
textL.Text = L
textXL.Text = XL

```

```

fin:
  End Sub

  Private Sub textfrec_Click()
  Static pi
  pi = 3.14159265358979
  fre = Val(Mid(textfrec.Text, 1, 4))
  L = Val(textL.Text)

  If L = 0 Then
    GoTo fin
  End If

  XL = 2 * pi * fre * L * 10 ^ -3
  textXL.Text = XL
fin:
  End Sub

  Private Sub textfrec_Change()
  Static pi
  pi = 3.14159265358979
  fre = Val(Mid(textfrec.Text, 1, 4))
  L = Val(textL.Text)

  If L = 0 Then
    GoTo fin
  End If

  XL = 2 * pi * fre * L * 10 ^ -3
  textXL.Text = XL
fin:
  End Sub

  Private Sub uckto_Click()
  If evita <> 1 Then
    GoTo final
  End If

  If dat1ckto!optsime.Value = True Then
    lblasime.Visible = False
    lblsime.Visible = True
    Capac!lblasime.Visible = False 'Para capac
    Capac!lblsime.Visible = True

    btnverdis.Enabled = False
    Capac!btnverdis.Enabled = False
  Else

```

```

Iblasime.Visible = True
Iblsime.Visible = False
Capac!Iblasime.Visible = True 'Para capac
Capac!Iblsime.Visible = False

    btnverdis.Enabled = True
    Capac!btnverdis.Enabled = True
End If

If uckto.Value = True Then
    btncálculo.Enabled = True

    Capac!uckto.Value = True 'Establece la opt de uckto en la frm de capac
    Ibluckto.Visible = True
    Iblckto.Visible = False
    Capac!ucktolab.Visible = True
    Capac!dcktolab.Visible = False

    dat1ckto.Show vbModal 'Despliega la frm de datos

    btnverdis.Enabled = False
    Capac!btnverdis.Enabled = False
Else
    Ibluckto.Visible = False
    Iblckto.Visible = True
    Capac!ucktolab.Visible = False
    Capac!dcktolab.Visible = True

    btnverdis.Enabled = True
    Capac!btnverdis.Enabled = True
End If

If uckto.Value = False Then
    Capac!uckto.Value = False 'Desabilita la opt de uckto en la frm de capac
    Ibluckto.Visible = False
    Iblckto.Visible = True
    Capac!ucktolab.Visible = False
    Capac!dcktolab.Visible = True

    btnverdis.Enabled = False
    Capac!btnverdis.Enabled = False
Else
    Ibluckto.Visible = True
    Iblckto.Visible = False
    Capac!ucktolab.Visible = True
    Capac!dcktolab.Visible = False

```

```
        btnverdis.Enabled = True
        Capaci!btnverdis.Enabled = True
    End If
final:
End Sub
```

```
Private Sub uckto_DblClick()
    btncálculo.Enabled = True
```

```
    dat1ckto.Show vbModal
End Sub
```

CAPACITANCIA

(Capac.frm)

```

Public efsuelo As Boolean

Private Sub btncalcapa_Click()
Static pi
pi = 3.14159265358979

If efsuelo = False Then
If uckto = True Then

Rem CALCULO DE LA CAPACITANCIA

Rem ***** SIN EFECTO SUELO *****

Rem ***** inicio para un ckto *****

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
dmg = dab
Else
dab = (dat1ckto!textdab.Text)
dbc = (dat1ckto!textdbc.Text)
dca = (dat1ckto!textdca.Text)
dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If Val(textdis.Text) = 0 Then
rmgg = radio
requi = 0
GoTo Line3
Else
dis = (Val(textdis.Text)) / 100
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase

```

```

GoTo Line3
End If

```

Line3:

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

```

```

textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1

```

```

Else
Rem ***** inicio de dos ckto *****

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

```

```

If ncf.Text = 1 Then
textdis.Text = ""
End If

```

```

If textdis.Text = "" Then
rmgg = radio
requi = 0
rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
GoTo Line6

```

```

Else
dis = (Val(textdis.Text)) / 100
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
'aumento por el caso de 2ckts:
rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

```

Line6:

```

If rmgdckto = 0 Then
End If

```

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgdckto)

```



```

XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
End If

Else 'SEPARA LA CONDICION DE EFECTO SUELO

If uckto = True Then

Rem ***** CON EFECTO SUELO *****

Rem ***** inicio para un ckto *****

If flecha = 0 Then
    obs = MsgBox("Si va a considerar el efecto del suelo (efecto espejo),
entonces introduzca el dato de la flecha (si no ha calculado la línea completa): ",
vbInformation, "Flecha")
    textflecha.SetFocus
    GoTo fin
End If

flecha = Val(textflecha.Text)

If dat1ckto!opthigual.Value = True Then
    ha = Val(dat1ckto!text1ha.Text)
    hb = ha
    hc = ha
Else
    ha = Val(dat1ckto!text1ha.Text)
    hb = Val(dat1ckto!text1hb.Text)
    hc = Val(dat1ckto!text1hc.Text)
End If

Rem CALCULO DE HMG
hma = Abs(ha - (2 / 3) * flecha)
hmb = Abs(hb - (2 / 3) * flecha)
hmc = Abs(hc - (2 / 3) * flecha)

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

fre = Val(Mid(textfrec.Text, 1, 4))

```

```

radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
    textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)

If dat1ckto!optsime.Value = True Then
    dmg = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line7
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line7
End If

Line7:

    c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^ 2)
    ^ (1 / 2)))
    XC = (2 * pi * fre * c) ^ -1
    Yc1 = 1 / XC
    textdmg.Text = dmg
    textrmgg.Text = rmgg * 1000
    textrequi.Text = requi * 1000
    textc.Text = c
    textXC.Text = XC
    textyc.Text = Yc1
Else

Rem ***** inicio de dos ckto *****

flecha = Val(textflecha.Text)

```

```

If dat2ckto!opthigual.Value = True Then
  ha = dat2ckto!text2ha.Text
  hb = ha
  hc = ha
Else
  ha = dat2ckto!text2ha.Text
  hb = dat2ckto!text2hb.Text
  hc = dat2ckto!text2hc.Text
End If

Rem CALCULO DE HMG
hma = ha - (2 / 3) * flecha
hmb = hb - (2 / 3) * flecha
hmc = hc - (2 / 3) * flecha

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
  textdis.Text = ""
End If

If textdis.Text = "" Then
  rmgg = radio
  requi = 0
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  GoTo Line8
Else
  dis = (Val(textdis.Text)) / 100
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
  más de un conductor por fase
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

Line8:

If rmgdckto = 0 Then
End If

c = (2 * pi * 8.85 * (10 ^ -3)) / Log((dmg * 2 * hmg) / (rmgdckto * (4 * (hmg ^ 2) +
(dmg ^ 2)) ^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1

```

```

Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
End If

End If
fin:
End Sub

Private Sub btniracables_Click()
Hide
frmCables.Show
End Sub

Private Sub btnverdis_Click()
frmverdis.Show
End Sub

Private Sub dckto_Click()
If evita = 2 Then
    GoTo final
End If

evita = 4

If dckto.Value = True Then
    btncalcapa.Enabled = True
    Induct!dckto.Value = True 'Establece la opt de dckto en la frm de induct
    dcktolab.Visible = True
    ucktolab.Visible = False
    Induct!!bldckto.Visible = True
    Induct!!bluckto.Visible = False
    Induct!luckto.Value = False
    lblasime.Visible = False
    llsime.Visible = False
    Induct!!lblasime.Visible = False
    Induct!!llsime.Visible = False
    btnverdis.Enabled = True
    Induct!btnverdis.Enabled = True
    btnverdis.TabStop = True
    dat2ckto.Show vbModal
Else
    dcktolab.Visible = False

```

```

    ucktolab.Visible = True
    Induct!lbdckto.Visible = False
    Induct!lbluckto.Visible = True
    btnverdis.Enabled = False
    Induct!btnverdis.Enabled = False
End If

If dckto.Value = False Then
    Induct!luckto.Value = True 'Establece la opt de uckto en la frm de induct
    dcktolab.Visible = False
    ucktolab.Visible = True
    Induct!lbdckto.Visible = False
    Induct!lbluckto.Visible = True
    indut!dckto.Value = False
    btnverdis.Enabled = False
    Induct!btnverdis.Enabled = False
Else
    dcktolab.Visible = True
    ucktolab.Visible = False
    Induct!lbdckto.Visible = True
    Induct!lbluckto.Visible = False

    btnverdis.Enabled = True
    Induct!btnverdis.Enabled = True
End If
final:
End Sub

Private Sub dckto_DbClick()
    btncalcapa.Enabled = True
    dat2ckto.Show vbModal
End Sub

Private Sub Form_Activate()
    texttipo.Text = frmCables!texttipo.Text
    textmcm.Text = frmCables!textmcm.Text
    texttrenzado.Text = frmCables!texttrenzado.Text
    textradio.Text = frmCables!textradio.Text
    textflecha.Text = flecha
    evita = 3
End Sub

Private Sub Form_Deactivate()
    c = Val(textc.Text)
End Sub

Private Sub iractes_Click()

```

```

Hide
ctes.Show
End Sub

Private Sub irainduct_Click()
Hide
Induct.Show
End Sub

Private Sub irairesi_Click()
Hide
resi.Show
End Sub

Private Sub ncf_Click()
ncfa = Int(ncf.Text)
ncf.Text = ncfa

Select Case ncfa
Case Is = 0
def = MsgBox("No existen Subconductores, por lo menos debe de existir un
conductor ", 48, "Valor negativo ")
Line1:
ncf.Text = 1
Case Is < 1
def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
If def = vbYes Then
ncf.Text = Abs(Int(ncf.Text))
frec.SetFocus
Else
GoTo Line1
End If
End Select

Rem *****
Rem ***** CALCULO DE LA CAPACITANCIA *****

Static pi
pi = 3.14159265358979

If efsuelo = False Then
If uckto = True Then

Rem ***** SIN EFECTO SUELO *****

Rem ***** inicio para un ckto *****

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
    textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
    dmg = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line3
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line3
End If

Line3:

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1

Else

Rem ***** inicio de dos ckto *****

```

```

fre = Val(Mid(textfreq.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line6
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

Line6:

If rmgdckto = 0 Then
End If

If dmg = 0 Then
    msj = "Defina el número de circuitos por estructura. !"
    MsgBox msj
    uckto.SetFocus
    GoTo fin
End If

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgdckto)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
End If

Else ' SEPARA LA CONDICION DE EFECTO SUELO

```



```

If uckto = True Then

Rem ***** CON EFECTO SUELO *****

Rem ***** inicio para un ckto *****

If flecha = 0 Then
    obs = MsgBox("Si va a considerar el efecto del suelo (efecto espejo),
entonces introduzca el dato de la flecha (si no ha calculado la línea completa): ",
vbInformation, "Flecha")
    textflecha.SetFocus
    GoTo fin
End If

flecha = Val(textflecha.Text)

If dat1ckto!opthigual.Value = True Then
    ha = Val(dat1ckto!text1ha.Text)
    hb = ha
    hc = ha
Else
    ha = Val(dat1ckto!text1ha.Text)
    hb = Val(dat1ckto!text1hb.Text)
    hc = Val(dat1ckto!text1hc.Text)
End If

Rem CALCULO DE HMG
hma = Abs(ha - (2 / 3) * flecha)
hmb = Abs(hb - (2 / 3) * flecha)
hmc = Abs(hc - (2 / 3) * flecha)

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
    textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
    dmng = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)

```

```

    dca = (dat1ckto!textdca.Text)
    dmng = (dab * dbc * dca) ^ (1 / 3)
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line7
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
    GoTo Line7
End If

Line7:

    c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmng * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmng ^ 2)
^ (1 / 2)))
    XC = (2 * pi * fre * c) ^ -1
    Yc1 = 1 / XC
    textdmng.Text = dmng
    textrmgg.Text = rmgg * 1000
    textrequi.Text = requi * 1000
    textc.Text = c
    textXC.Text = XC
    textyc.Text = Yc1
Else

Rem ***** inicio de dos ckto *****

flecha = Val(textflecha.Text)
If dat2ckto!opthigual.Value = True Then
    ha = dat2ckto!text2ha.Text
    hb = ha
    hc = ha
Else
    ha = dat2ckto!text2ha.Text
    hb = dat2ckto!text2hb.Text
    hc = dat2ckto!text2hc.Text
End If

Rem CALCULO DE HMG
hma = ha - (2 / 3) * flecha
hmb = hb - (2 / 3) * flecha
hmc = hc - (2 / 3) * flecha

```

```

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line8
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

Line8:

If rmgdckto = 0 Then
End If

c = (2 * pi * 8.85 * (10 ^ -3)) / Log((dmg * 2 * hmg) / (rmgdckto * (4 * (hmg ^ 2) +
(dmg ^ 2)) ^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
End If

End If
fin:
End Sub

Private Sub ncf_Change()

```

```

If ncf.Text = "" Then
    GoTo papaya
Else
    ncfa = Int(ncf.Text)
    ncf.Text = ncfa
End If

If ncf.Text = 1 Then
    textdis.Text = ""
End If

Select Case ncfa

    Case Is >= 6
        mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo se puede
aceptar. ", vbExclamation, "Número de subconductores (Haz de conductores)")
        GoTo Line2

    Case Is = 0
        def = MsgBox("Por lo menos debe de existir un conductor ", 48, "Definición de
subconductores ")
Line1:
        ncf.Text = 1

    Case Is < 1
        def = MsgBox("Esta deacuerdo en el siguiente valor: " & Abs(Int(ncf)), 4,
"Parámetro inaceptable")
        If def = vbYes Then
            ncf.Text = Abs(Int(ncf.Text))
Line2:
        Else
            GoTo Line1
        End If
    End Select

If ncfa >= 2 Then
    lbldis.Enabled = True
    textdis.Enabled = True
    textdis.SetFocus
Else
    lbldis.Enabled = False
    textdis.Enabled = False
End If

Rem *****
Rem ***** CALCULO DE LA CAPACITANCIA *****

```

```

Static pi
pi = 3.14159265358979

If efsuelo = False Then
If uckto = True Then

Rem ***** SIN EFECTO SUELO *****

Rem ***** inicio para un ckto *****

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
  textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
  dmng = dab
Else
  dab = (dat1ckto!textdab.Text)
  dbc = (dat1ckto!textdbc.Text)
  dca = (dat1ckto!textdca.Text)
  dmng = (dab * dbc * dca) ^ (1 / 3)
End If

If textdis.Text = "" Then
  rmngg = radio
  requi = 0
  GoTo Line3
Else
  dis = (Val(textdis.Text)) / 100
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmngg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
  más de un conductor por fase
  GoTo Line3
End If

Line3:

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmng / rmngg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textdmng.Text = dmng

```

```

textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1

```

```
Else
```

```
Rem ***** inicio de dos ckto *****
```

```

fre = Val(Mid(textfreq.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

```

```

If ncf.Text = 1 Then
  textdis.Text = ""
End If

```

```
If textdis.Text = "" Then
```

```

  rmgg = radio
  requi = 0
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  GoTo Line6

```

```
Else
```

```

  dis = (Val(textdis.Text)) / 100
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
  más de un conductor por fase
  'aumento por el caso de 2ckts:
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

```

```
Line6:
```

```

If rmgdckto = 0 Then
End If

```

```
If dmg = 0 Then
```

```
  If dmg = 0 Then
```

```
    msj = "Defina el número de circuitos por estructura. !"
```

```
    MsgBox msj
```

```
    uckto.SetFocus
```

```
    GoTo fin
```

```
  End If
```

```
  MsgBox msj
```

```
  textdmg.SetFocus
```

```
  GoTo fin
```

```
End If
```

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgdckto)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
    End If

Else ' SEPARA LA CONDICION DE EFECTO SUELO

If uckto = True Then

Rem ***** CON EFECTO SUELO *****

Rem ***** inicio para un ckto *****

If flecha = 0 Then
    obs = MsgBox("Si va a considerar el efecto del suelo (efecto espejo),
entonces introduzca el dato de la flecha (si no ha calculado la línea completa): ",
vbInformation, "Flecha")
    textflecha.SetFocus
    GoTo fin
End If

flecha = Val(textflecha.Text)

If dat1ckto!opthigual.Value = True Then
    ha = Val(dat1ckto!text1ha.Text)
    hb = ha
    hc = ha
Else
    ha = Val(dat1ckto!text1ha.Text)
    hb = Val(dat1ckto!text1hb.Text)
    hc = Val(dat1ckto!text1hc.Text)
End If

Rem CALCULO DE HMG
hma = Abs(ha - (2 / 3) * flecha)
hmb = Abs(hb - (2 / 3) * flecha)
hmc = Abs(hc - (2 / 3) * flecha)

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

If ncf.Text = 1 Then
    textdis.Text = ""
End If

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
    dmg = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line7
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line7
End If

Line7:

c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^ 2)
^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
Else

Rem ***** inicio de dos ckto *****

flecha = Val(textflecha.Text)

```



```

If dat2ckto!opthigual.Value = True Then
  ha = Val(dat2ckto!text2ha.Text)
  hb = ha
  hc = ha
Else
  ha = Val(dat2ckto!text2ha.Text)
  hb = Val(dat2ckto!text2hb.Text)
  hc = Val(dat2ckto!text2hc.Text)
End If

```

```

Rem CALCULO DE HMG
hma = ha - (2 / 3) * flecha
hmb = hb - (2 / 3) * flecha
hmc = hc - (2 / 3) * flecha

```

```

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

```

```

If ncf.Text = 1 Then
  textdis.Text = ""
End If

```

```

If textdis.Text = "" Then
  rmgg = radio
  requi = 0
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  GoTo Line8
Else
  dis = (Val(textdis.Text)) / 100
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
  más de un conductor por fase
  rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

```

Line8:

```

If rmgdckto = 0 Then
End If

```

```

c = (2 * pi * 8.85 * (10 ^ -3)) / Log((dmg * 2 * hmg) / (rmgdckto * (4 * (hmg ^ 2) +
(dmg ^ 2)) ^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1

```

```
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
    End If

End If
fin:
papaya:
    End Sub

Private Sub optnsuelo_Click()
If optnsuelo.Enabled = True Then
    Iblflecha.Enabled = False
    textflecha.Enabled = False
    efsuelo = False
    Iblhmg.Enabled = False
    textverhmg.Enabled = False
Else
    Iblflecha.Enabled = True
    textflecha.Enabled = True
    efsuelo = True
    Iblhmg.Enabled = True
    textverhmg.Enabled = True
End If
End Sub

Private Sub optsuelo_Click()
If optsuelo.Enabled = True Then
    Iblflecha.Enabled = True
    textflecha.Enabled = True
    textflecha.SetFocus
    efsuelo = True
    Iblhmg.Enabled = True
    textverhmg.Enabled = True
Else
    Iblflecha.Enabled = False
    textflecha.Enabled = False
    efsuelo = False
    Iblhmg.Enabled = False
    textverhmg.Enabled = False
End If
End Sub
```

```
Private Sub salircapac_Click()
Unload Me
Param.Show
End Sub
```

```
Private Sub spinnu_SpinDown()
Dim dblrate
dblrate = Val(ncf.Text)
If dblrate > 1 Then
    ncf.Text = dblrate - 1
Else
    ncf.Text = 1
End If
ncf.Refresh
End Sub
```

```
Private Sub spinnu_SpinUp()
ncf.Text = (Val(ncf.Text)) + 1
If dblrate >= 6 Then
    mensa = MsgBox("En una aplicación práctica no es adecuado utilizar
demasiados subconductores, sin embargo, a manera de ejemplo (o simulación de
cálculo) se puede aceptar ", vbExclamation, "Número de subconductores (Haz)")
End If
ncf.Refresh
End Sub
```

```
Private Sub textdis_Click()
Static pi
pi = 3.14159265358979
```

```
If efsuelo = False Then
    If uckto = True Then
```

```
Rem ***** SIN EFECTO SUELO *****
```

```
Rem ***** inicio para un ckto *****
```

```
fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
```

```
If ncf.Text = 1 Then
    textdis.Text = ""
End If
```

```
dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
    dmg = dab
```

```

Else
dab = (dat1ckto!textdab.Text)
dbc = (dat1ckto!textdbc.Text)
dca = (dat1ckto!textdca.Text)
dmg = (dab * dbc * dca) ^ (1 / 3)
End If

If Val(textdis.Text) = 0 Then
  rmgg = radio
  requi = 0
  GoTo Line3
Else
  dis = (Val(textdis.Text)) / 100
  requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
  rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
  más de un conductor por fase
  GoTo Line3
End If

Line3:

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1

Else

Rem ***** inicio de dos ckto *****

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
  textdis.Text = ""
End If

If textdis.Text = "" Then
  rmgg = radio
  requi = 0

```

```

    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line6
    Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    End If

```

Line6:

```

    If rmgdckto = 0 Then
    End If

    c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmng / rmgdckto)
    XC = (2 * pi * fre * c) ^ -1
    Yc1 = 1 / XC
    textdmng.Text = dmng
    textrmgg.Text = rmgdckto * 1000
    textrequi.Text = requi * 1000
    textc.Text = c
    textXC.Text = XC
    textyc.Text = Yc1
    End If

    Else ' SEPARA LA CONDICION DE EFECTO SUELO

    If uckto = True Then

    Rem ***** CON EFECTO SUELO *****

    Rem ***** inicio para un ckto *****

    If flecha = 0 Then
        obs = MsgBox("Si va a considerar el efecto del suelo (efecto espejo),
        entonces introduzca el dato de la flecha (si no ha calculado la línea completa): ",
        vbInformation, "Flecha")
        textflecha.SetFocus
        GoTo fin
    End If

    flecha = Val(textflecha.Text)

    If dat1ckto!opthigual.Value = True Then
        ha = Val(dat1ckto!text1ha.Text)

```

```

hb = ha
hc = ha
Else
ha = Val(dat1ckto!text1ha.Text)
hb = Val(dat1ckto!text1hb.Text)
hc = Val(dat1ckto!text1hc.Text)
End If

```

```

Rem CALCULO DE HMG
hma = Abs(ha - (2 / 3) * flecha)
hmb = Abs(hb - (2 / 3) * flecha)
hmc = Abs(hc - (2 / 3) * flecha)

```

```

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

```

```

If ncf.Text = 1 Then
textdis.Text = ""
End If

```

```

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
dmg = dab
Else
dab = (dat1ckto!textdab.Text)
dbc = (dat1ckto!textdbc.Text)
dca = (dat1ckto!textdca.Text)
dmg = (dab * dbc * dca) ^ (1 / 3)
End If

```

```

If textdis.Text = "" Then
rmgg = radio
requi = 0
GoTo Line7
Else
dis = (Val(textdis.Text)) / 100
requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
más de un conductor por fase
GoTo Line7
End If

```

Line7:

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^ 2)
^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
Else

Rem ***** inicio de dos ckto *****

flecha = Val(textflecha.Text)
If dat2ckto!opthigual.Value = True Then
    ha = dat2ckto!text2ha.Text
    hb = ha
    hc = ha
Else
    ha = dat2ckto!text2ha.Text
    hb = dat2ckto!text2hb.Text
    hc = dat2ckto!text2hc.Text
End If

Rem CALCULO DE HMG
hma = ha - (2 / 3) * flecha
hmb = hb - (2 / 3) * flecha
hmc = hc - (2 / 3) * flecha

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line8
Else

```

```

    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
  End If

```

Line8:

```

  If rmgdckto = 0 Then
  End If

  c = (2 * pi * 8.85 * (10 ^ -3)) / Log((dmg * 2 * hmg) / (rmgdckto * (4 * (hmg ^ 2) +
  (dmg ^ 2)) ^ (1 / 2)))
  XC = (2 * pi * fre * c) ^ -1
  Yc1 = 1 / XC
  textdmg.Text = dmg
  textrmgg.Text = rmgdckto * 1000
  textrequi.Text = requi * 1000
  textc.Text = c
  textXC.Text = XC
  textyc.Text = Yc1
  End If
  End If

```

fin:

```
End Sub
```

```
Private Sub textdis_Change()
```

```
Static pi
```

```
pi = 3.14159265358979
```

```
If efsuelo = False Then
```

```
If uckto = True Then
```

```
Rem ***** SIN EFECTO SUELO *****
```

```
Rem ***** inicio para un ckto *****
```

```
fre = Val(Mid(textfrec.Text, 1, 4))
```

```
radio = (Val(textradio.Text)) / 1000
```

```
If ncf.Text = 1 Then
```

```
  textdis.Text = ""
```

```
End If
```

```
dab = (dat1ckto!textdab.Text)
```

```
If dat1ckto!optsime.Value = True Then
```



```

    dmng = dab
    Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmng = (dab * dbc * dca) ^ (1 / 3)
End If

'If textdis.Text = "" Then
If Val(textdis.Text) = 0 Then
    rmgg = radio
    requi = 0
    GoTo Line3
Else
    dis = (Val(textdis.Text)) / 100

    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line3
End If

```

Line3:

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmng / rmgg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textdmng.Text = dmng
textrmgg.Text = rmgg * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1

Else

Rem ***** inicio de dos ckto *****

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000
dmng = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)

If ncf.Text = 1 Then
    textdis.Text = ""
End If

```

```

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line6
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    'aumento por el caso de 2ckts:
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
End If

```

Line6:

```

If rmgdckto = 0 Then
End If

```

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgdckto)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
textdmg.Text = dmg
textrmgg.Text = rmgdckto * 1000
textrequi.Text = requi * 1000
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
End If

```

```

Else ' SEPARA LA CONDICION DE EFECTO SUELO

```

```

    If uckto = True Then

```

```

        Rem ***** CON EFECTO SUELO *****

```

```

        Rem ***** inicio para un ckto *****

```

```

        If flecha = 0 Then

```

```

            obs = MsgBox("Si va a considerar el efecto del suelo (efecto espejo),
            entonces introduzca el dato de la flecha (si no ha calculado la línea completa): ",
            vbInformation, "Flecha")
            textflecha.SetFocus
            GoTo fin
        End If

```

```

        flecha = Val(textflecha.Text)

```

```

If dat1ckto!opthigual.Value = True Then
    ha = Val(dat1ckto!text1ha.Text)
    hb = ha
    hc = ha
Else
    ha = Val(dat1ckto!text1ha.Text)
    hb = Val(dat1ckto!text1hb.Text)
    hc = Val(dat1ckto!text1hc.Text)
End If

```

```

Rem CALCULO DE HMG
hma = Abs(ha - (2 / 3) * flecha)
hmb = Abs(hb - (2 / 3) * flecha)
hmc = Abs(hc - (2 / 3) * flecha)

```

```

hmg = (hma * hmb * hmc) ^ (1 / 3)
textverhmg = hmg

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
radio = (Val(textradio.Text)) / 1000

```

```

If ncf.Text = 1 Then
    textdis.Text = ""
End If

```

```

dab = (dat1ckto!textdab.Text)
If dat1ckto!optsime.Value = True Then
    dmng = dab
Else
    dab = (dat1ckto!textdab.Text)
    dbc = (dat1ckto!textdbc.Text)
    dca = (dat1ckto!textdca.Text)
    dmng = (dab * dbc * dca) ^ (1 / 3)
End If

```

```

If textdis.Text = "" Then
    rmgg = radio
    requi = 0
    GoTo Line7
Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    GoTo Line7
End If

```

Line7:

```
c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^ 2)
^ (1 / 2)))
```

```
XC = (2 * pi * fre * c) ^ -1
```

```
Yc1 = 1 / XC
```

```
textdmg.Text = dmg
```

```
textrmgg.Text = rmgg * 1000
```

```
textrequi.Text = requi * 1000
```

```
textc.Text = c
```

```
textXC.Text = XC
```

```
textyc.Text = Yc1
```

```
Else
```

```
Rem ***** inicio de dos ckto *****
```

```
flecha = Val(textflecha.Text)
```

```
If dat2ckto!opthigual.Value = True Then
```

```
    ha = dat2ckto!text2ha.Text
```

```
    hb = ha
```

```
    hc = ha
```

```
Else
```

```
    ha = dat2ckto!text2ha.Text
```

```
    hb = dat2ckto!text2hb.Text
```

```
    hc = dat2ckto!text2hc.Text
```

```
End If
```

```
Rem CALCULO DE HMG
```

```
hma = ha - (2 / 3) * flecha
```

```
hmb = hb - (2 / 3) * flecha
```

```
hmc = hc - (2 / 3) * flecha
```

```
hmg = (hma * hmb * hmc) ^ (1 / 3)
```

```
textverhmg = hmg
```

```
fre = Val(Mid(textfrec.Text, 1, 4))
```

```
radio = (Val(textradio.Text)) / 1000
```

```
dmg = (dab ^ 2 * dac ^ 2 * dab2 ^ 2 * dba2 ^ 2 * dbc ^ 2 * dac2 * dca2) ^ (1 / 12)
```

```
If ncf.Text = 1 Then
```

```
    textdis.Text = ""
```

```
End If
```

```
If textdis.Text = "" Then
```

```
    rmgg = radio
```

```
    requi = 0
```

```

    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    GoTo Line8
    Else
    dis = (Val(textdis.Text)) / 100
    requi = dis / (2 * Sin(pi / ncfa)) 'Radio equivalente
    rmgg = (ncfa * radio * (requi) ^ (ncfa - 1)) ^ (1 / ncfa) 'RMG del grupo si existe
    más de un conductor por fase
    rmgdckto = (rmgg ^ 3 * daa2 ^ 2 * dbb2) ^ (1 / 6)
    End If

```

Line8:

```

    If rmgdckto = 0 Then
    End If

    c = (2 * pi * 8.85 * (10 ^ -3)) / Log((dmg * 2 * hmg) / (rmgdckto * (4 * (hmg ^ 2) +
    (dmg ^ 2)) ^ (1 / 2)))
    XC = (2 * pi * fre * c) ^ -1
    Yc1 = 1 / XC
    textdmg.Text = dmng
    textrmgg.Text = rmgdckto * 1000
    textrequi.Text = requi * 1000
    textc.Text = c
    textXC.Text = XC
    textyc.Text = Yc1
    End If
    End If
fin:
    End Sub

```

```

Private Sub textdmg_Click()
    Static pi
    pi = 3.14159265358979
    fre = Val(Mid(textfrec.Text, 1, 4))

```

```

    If optnsuelo.Value = True Then

        Rem ***** Sin efecto del suelo *****
        textrmgg.Text = textradio.Text
        rmgg = Val(textrmgg.Text) / 1000
        dmng = Val(textdmg.Text)

        If dmng = 0 Then
            textdmg.SetFocus
            GoTo fin
        End If

```

```

c = 2 * pi * 8.85 * 10 ^ -3 / Log(dmg / rmgg)
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

```

```

textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
GoTo fin
End If

```

```

If optsuelo.Value = True Then

```

```

    Rem ***** Con efecto del suelo *****
    textrmgg.Text = textradio.Text
    rmgg = Val(textrmgg.Text) / 1000
    dmg = Val(textdmg.Text)
    hmg = Val(textverhmg.Text)

```

```

    If dmg = 0 Then
        textdmg.SetFocus
        GoTo fin
    End If

```

```

    If hmg = 0 Then
        textverhmg.SetFocus
        GoTo fin
    End If

```

```

    c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^
2) ^ (1 / 2)))
    XC = (2 * pi * fre * c) ^ -1
    Yc1 = 1 / XC

```

```

    textc.Text = c
    textXC.Text = XC
    textyc.Text = Yc1

```

```

End If

```

```

fin:

```

```

End Sub

```

```

Private Sub textflecha_Change()
flecha = Val(textflecha.Text)
End Sub

```

```

Private Sub textfrec_Click()
Static pi
pi = 3.14159265358979

```

```

fre = Val(Mid(textfrec.Text, 1, 4))
c = Val(textc.Text)

If c = 0 Then
    GoTo fin
End If

If fre = 0 Then
    textfrec.SetFocus
    GoTo fin
End If

XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textXC.Text = XC
textyc.Text = Yc1
fin:
End Sub

Private Sub textfrec_Change()
Static pi
pi = 3.14159265358979
fre = Val(Mid(textfrec.Text, 1, 4))
c = Val(textc.Text)

If c = 0 Then
    GoTo fin
End If

If fre = 0 Then
    textfrec.SetFocus
    GoTo fin
End If

XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC

textXC.Text = XC
textyc.Text = Yc1
fin:
End Sub

Private Sub textverhmg_Click()
Static pi
pi = 3.14159265358979
fre = Val(Mid(textfrec.Text, 1, 4))

```

Rem ***** Con efecto del suelo *****

```
textrmgg.Text = textradio.Text
rmgg = Val(textrmgg.Text) / 1000
dmg = Val(textdmg.Text)
hmg = Val(textverhmg.Text)
```

```
If dmg = 0 Then
    textdmg.SetFocus
    GoTo fin
End If
```

```
If hmg = 0 Then
    textverhmg.SetFocus
    GoTo fin
End If
```

```
c = 2 * pi * 8.85 * 10 ^ -3 / Log((dmg * 2 * hmg) / (rmgg * (4 * hmg ^ 2 + dmg ^ 2)
^ (1 / 2)))
XC = (2 * pi * fre * c) ^ -1
Yc1 = 1 / XC
```

```
textc.Text = c
textXC.Text = XC
textyc.Text = Yc1
```

```
fin:
End Sub
```

```
Private Sub uckto_Click()
If evita <> 3 Then
    GoTo final
End If
```

```
If dat1ckto!optsime.Value = True Then
    lblasime.Visible = False
    lblsime.Visible = True
    Induct!lblasime.Visible = False
    Induct!lblsime.Visible = True
    btnverdis.Enabled = False
    Induct!btnverdis.Enabled = False
Else
    lblasime.Visible = True
    lblsime.Visible = False
    Induct!lblasime.Visible = True
    Induct!lblsime.Visible = False
    btnverdis.Enabled = True
```



```
    Induct!btnverdis.Enabled = True  
End If
```

```
If uckto.Value = True Then
```

```
    Rem Establece la opt de uckto en la frm de induct
```

```
    btnalcapa.Enabled = True  
    Induct!uckto.Value = True  
    ucktolab.Visible = True  
    dcktolab.Visible = False  
    Induct!lbluckto.Visible = True  
    Induct!lbdckto.Visible = False  
    dat1ckto.Show vbModal  
    btnverdis.Enabled = False  
    Induct!btnverdis.Enabled = False  
Else  
    ucktolab.Visible = False  
    dcktolab.Visible = True  
    Induct!lbluckto.Visible = False  
    Induct!lbdckto.Visible = True  
    btnverdis.Enabled = True  
    Induct!btnverdis.Enabled = True
```

```
End If
```

```
If uckto.Value = False Then
```

```
    Rem Establece la opt de uckto en la frm de induct
```

```
    Induct!uckto.Value = False  
    ucktolab.Visible = False  
    dcktolab.Visible = True  
    Induct!lbluckto.Visible = False  
    Induct!lbdckto.Visible = True  
    btnverdis.Enabled = True  
    Induct!btnverdis.Enabled = True  
Else  
    ucktolab.Visible = True  
    dcktolab.Visible = False  
    Induct!lbluckto.Visible = True  
    Induct!lbdckto.Visible = False  
    btnverdis.Enabled = False  
    Induct!btnverdis.Enabled = False
```

```
End If
```

```
final:
```

```
End Sub
```

```
Private Sub uckto_Db1Click()
```

```
If dat1ckto!optsime.Value = True Then
    lblasime.Visible = False
    lblsime.Visible = True
Else
    lblasime.Visible = True
    lblsime.Visible = False
End If
```

```
If uckto.Value = True Then
    btnalcapa.Enabled = True
    Induct!uckto.Value = True 'Establece la opt de uckto en la frm de induct
    ucktolab.Visible = True
    dcktolab.Visible = False
    dat1ckto.Show vbModal
    btnverdis.Enabled = False
Else
    ucktolab.Visible = False
    dcktolab.Visible = True
End If
```

```
If uckto.Value = False Then
    Induct!uckto.Value = False 'Establece la opt de uckto en la frm de induct
    ucktolab.Visible = False
    dcktolab.Visible = True
Else
    ucktolab.Visible = True
    dcktolab.Visible = False
End If
```

```
final:
End Sub
```

CONSTANTES A, B, C Y D

(Ctes.frm)

```

Dim tipolinea As String

Private Sub btncalcular_Click()
Static pi
pi = 3.14159265358979
Dim CH(6), d(8), E(4), f(6)

km = Val(textkm.Text)
If km = 0 Then
    evit = MsgBox("Para calcular las constantes se necesita la longitud de la línea
(km), por favor entre el dato", vbCritical, "Longitud de la Línea")
    textkm.SetFocus
    GoTo final
Else
    km = Val(textkm.Text)
End If

rac = textresi.Text
L = textinduct.Text
c = textcapa.Text

fre = Mid(cbofrec.Text, 1, 4)

If rac = "" Then
    evita2 = MsgBox("Para calcular las constantes se necesita la Resistencia del
cable (Ohms/km), por favor entre el dato", vbCritical, "Resistencia del Cable")
    textresi.SetFocus
    GoTo final
Else
    rackm = rac * km
End If

XL = 2 * pi * fre * L * km * 10 ^ -3

tipolinea = cbotipolinea.Text 'Selecciona los tres casos
Select Case tipolinea
Case "Corta"
    GoTo Line1
Case "Media"
    GoTo Line2
Case "Larga"

```

```

    GoTo Line3
Case Else
    Item = MsgBox("No se ha definido el modelo: " & tipolinea, 48, "Modelo sin
definir")
    cbotipolinea.SetFocus
    GoTo Line4
End Select

```

```
Rem *****
```

```
Line1:  Rem ***** rutina de inicio: L. corta *****
```

```
    btndatos.Enabled = False
```

```
    Z = hip(rackm, XL)
```

```
    ang1 = ang(XL, Z)
```

```
    ang1 = 180 * ang1 / pi
```

```
    textA.Text = 1
```

```
    textD.Text = 1
```

```
    textC.Text = 0
```

```
    textB.Text = Format(Z, "0.00000") & ", β= " & Format(ang1, "0.00000") & " °"
```

```
    GoTo Line5      'Fin de rutina: L. corta
```

```
Rem *****
```

```
Line2:  Rem ***** rutina de inicio: L. media *****
```

```
    btndatos.Enabled = True
```

```
    u = textcapa.Text
```

```
    XC = (1 / (2 * pi * fre * u * 10 ^ -6)) / km
```

```
    Yc = 1 / XC
```

```
    If optpi.Value = True Then 'Define la dirección a tomar
```

```
        GoTo Line7      ' Si es "pi"
```

```
    Else
```

```
        GoTo Line8      ' Si es "te"
```

```
    End If
```

```
Line7:  Rem *** rutina para "pi" L / media ***
```

```
    Z = hip(rackm, XL)
```

```
    A = Z
```

```
    ang1 = ang(XL, Z)
```

```
    B = 180 * ang1 / pi
```

```
    v = Z: w = ang1
```

```
    T = Yc: g = pi / 2      'YC (pol)
```

```
    h = (v * T) / 2: i = w + g 'Fórmula Z*Yc/2 (pol)
```

```

J = h * Cos(i) + 1      'Fórmula Z*Yc/2 + 1 (rec)
K = h * Sin(i)         'Fórmula Z*Yc/2 + 1 (rec)
m = hip(J, K)
o = ang(K, m)
o = 180 * o / pi

```

```

'Para frmdatos presentación ***** pi/media *****
alfa = Sqr(h * 2): beta = i / 2 'Fórmula (Z*Yc)^0.5 (pol)
alfa1 = alfa * Cos(beta): beta1 = alfa * Sin(beta) '(rec)
londa = (2 * pi / beta1) * 1.609
velo = londa * fre

```

```

frmdatosabcd!textctepropa.Text = "μ=" & Format(alfa1, "0.00000") & " , β=" &
Format(beta1, "0.00000")
frmdatosabcd!textlonda.Text = Format(londa, "0.0000")
frmdatosabcd!textvelo.Text = Format(velo, "0.00000")

```

```

alfa2 = v / T: beta2 = w - g      'Fórmula Z/Yc (pol)
alfa3 = Sqr(alfa2): beta3 = beta2 / 2 'Fórmula (Z/Yc)^0.5 = Zc (pol)

```

```

voltaje = cbvoltaje.Text
sil = voltaje ^ 2 / alfa3

```

```

frmdatosabcd!textzc.Text = "Zc= " & Format(alfa3, "0.00000") & " , Ç=" &
Format(beta3 * 180 / pi, "0.00000") & " °"

```

```

frmdatosabcd!textsil.Text = Format(sil, "0.00000")
'Fin de presentación frmdatos ***** pi/media *****

```

```

h = (v * T) / 4: i = w + g 'Fórmula Z*Yc/4 (pol)
p = h * Cos(i) + 1      'Fórmula Z*Yc/4 + 1 (rec)
q = h * Sin(i)         'Fórmula Z*Yc/4 + 1 (rec)
r = hip(p, q)          'Fórmula Z*Yc/4 + 1 (pol)
s = ang(q, r)          'Fórmula Z*Yc/4 + 1 (pol)
r = r * T: s = s + g    'Fórmula (Z*Yc/4 + 1)*Yc (pol)
s = 180 * s / pi

```

```

textA.Text = Format(m, "0.00000") & " , Ø= " & Format(o, "0.00000") & " °"
textD.Text = Format(m, "0.00000") & " , Ç= " & Format(o, "0.00000") & " °"
textB.Text = Format(A, "0.00000") & " , β= " & Format(B, "0.00000") & " °"
textC.Text = Format(r, "0.00000") & " , μ= " & Format(s, "0.00000") & " °"

```

```

GoTo Line6 ' *** fin de rutina de "pi" L / media

```

```

Line8:      ' *** rutina para "te" L / media ***

```

```

Z = hip(rackm, XL)

```

```

A = Z
ang1 = ang(XL, Z)
B = 180 * ang1 / pi
v = Z: w = ang1
T = Yc: g = pi / 2      'YC (pol)
h = (v * T) / 2: i = w + g 'Fórmula Z*Yc/2 (pol)
J = h * Cos(i) + 1      'Fórmula Z*Yc/2 + 1 (rec)
K = h * Sin(i)          'Fórmula Z*Yc/2 + 1 (rec)
m = hip(J, K)
o = ang(K, m)
o = 180 * o / pi

```

```

'Para frmdatos presentación ***** T/media *****
alfa = Sqr(h * 2): beta = i / 2 'Fórmula (Z*Yc)^0.5 (pol)
alfa1 = alfa * Cos(beta): beta1 = alfa * Sin(beta) '(rec)
londa = (2 * pi / beta1) * 1.609
velo = londa * fre

```

```

frmdatosabcd!textctepropa.Text = "μ=" & Format(alfa1, "0.00000") & " , β=" &
Format(beta1, "0.00000")
frmdatosabcd!textlonda.Text = Format(londa, "0.00000")
frmdatosabcd!textvelo.Text = Format(velo, "0.00000")

```

```

alfa2 = v / T: beta2 = w - g      'Fórmula Z/Yc (pol)
alfa3 = Sqr(alfa2): beta3 = beta2 / 2 'Fórmula (Z/Yc)^0.5 = Zc (pol)

```

```

voltaje = cbvoltaje.Text
sil = voltaje ^ 2 / alfa3

```

```

frmdatosabcd!textzc.Text = "Zc=" & Format(alfa3, "0.00000") & " , Ç=" &
Format(beta3 * 180 / pi, "0.00000") & " °"

```

```

frmdatosabcd!textsil.Text = Format(sil, "0.00000")
'Fin de presentación frmdatos ***** T/media *****

```

```

h = (v * T) / 4: i = w + g 'Fórmula Z*Yc/4 (pol)
p = h * Cos(i) + 1      'Fórmula Z*Yc/4 + 1 (rec)
q = h * Sin(i)          'Fórmula Z*Yc/4 + 1 (rec)
r = hip(p, q)           'Fórmula Z*Yc/4 + 1 (pol)
s = ang(q, r)           'Fórmula Z*Yc/4 + 1 (pol)
r = r * A: s = s + ang1  'Fórmula (Z*Yc/4 + 1)*Z (pol)
s = 180 * s / pi
g = 180 * g / pi

```

```

textA.Text = Format(m, "0.00000") & " , Ø=" & Format(o, "0.00000") & " °"
textD.Text = Format(m, "0.00000") & " , Ç=" & Format(o, "0.00000") & " °"
textB.Text = Format(r, "0.00000") & " , β=" & Format(s, "0.00000") & " °"

```

```
textC.Text = Format(T, "0.00000") & ", μ=" & Format(g, "0.00000") & " °"
```

```
GoTo Line6      ' fin de rutina "T" L / media
```

```
Rem *****
```

```
Line3:  Rem ***** rutina de inicio: L. larga *****
```

```
btndatos.Enabled = True
```

```
u = textcapa.Text
```

```
XC = (1 / (2 * pi * fre * u * 10 ^ -6)) / km
```

```
Yc = 1 / XC
```

```
Z = hip(rackm, XL)
```

```
ang1 = ang(XL, Z)
```

```
v = Z: w = ang1      'Z (pol)
```

```
T = Yc: g = pi / 2   'YC (pol)
```

```
h = v * T: i = w + g 'Fórmula Z*Yc (pol)
```

```
A = Sqr(h)          'Fórmula sqrt(Z*Yc) = r (pol)
```

```
B = i / 2          'Fórmula sqrt(Z*Yc) = r (pol)
```

```
J = A * Cos(B)     'Fórmula sqrt(Z*Yc) = r (rec)
```

```
K = A * Sin(B)     'Fórmula sqrt(Z*Yc) = r (rec)
```

```
CH(1) = 0.5 * (Exp(J) * Cos(K) - Exp(-J) * Cos(-K)) 'senh(r) (rec)
```

```
CH(2) = 0.5 * (Exp(J) * Sin(K) - Exp(-J) * Sin(-K)) 'senh(r) (rec)
```

```
d(1) = hip(CH(1), CH(2)) 'senh(r) (pol)
```

```
d(2) = ang(CH(2), d(1)) 'senh(r) (pol)
```

```
CH(3) = 0.5 * (Exp(J / 2) * Cos(K / 2) - Exp(-J / 2) * Cos(-K / 2)) 'senh(r/2) (rec)
```

```
CH(4) = 0.5 * (Exp(J / 2) * Sin(K / 2) - Exp(-J / 2) * Sin(-K / 2)) 'senh(r/2) (rec)
```

```
d(3) = hip(CH(3), CH(4)) 'senh(r/2) (pol)
```

```
d(4) = ang(CH(4), d(3)) 'senh(r/2) (pol)
```

```
CH(5) = 0.5 * (Exp(J / 2) * Cos(K / 2) + Exp(-J / 2) * Cos(-K / 2)) 'cosh(r/2) (rec)
```

```
CH(6) = 0.5 * (Exp(J / 2) * Sin(K / 2) + Exp(-J / 2) * Sin(-K / 2)) 'cosh(r/2) (rec)
```

```
d(5) = hip(CH(5), CH(6)) 'cosh(r/2) (pol)
```

```
d(6) = ang(CH(6), d(5)) 'cosh(r/2) (pol)
```

```
d(7) = d(1) / A     'senh(r) / r (pol)
```

```
d(8) = d(2) - B     'senh(r) / r (pol)
```

```
E(1) = d(3) / d(5) 'senh(r/2) / cosh(r/2)= tanh(r/2) (pol)
```

$$E(2) = d(4) - d(6) \cdot \tanh(r/2) \text{ (pol)}$$

$$E(3) = E(1) \cdot 2 / A \cdot \tanh(r/2) / (r/2) \text{ (pol)}$$

$$E(4) = E(2) - B \cdot \tanh(r/2) / (r/2) \text{ (pol)}$$

```
If optpi.Value = True Then 'Define la dirección a tomar
  GoTo Line9           ' Si es "pi"
Else
  GoTo Line10         ' Si es "te"
End If
```

Line9: ' *** rutina para "pi" L / larga ***

$$f(1) = d(7) \cdot v \text{ 'definición de: Z' (pol)}$$

$$f(2) = d(8) + w \text{ 'definición de: Z' (pol)}$$

$$f(3) = E(3) \cdot T \text{ 'definición de Y' (pol)}$$

$$f(4) = E(4) + g \text{ 'definición de Y' (pol)}$$

$$Z = f(1)$$

$$A = Z$$

$$\text{ang1} = f(2)$$

$$B = 180 \cdot \text{ang1} / \pi$$

$$v = Z: w = \text{ang1}$$

$$T = f(3): g = f(4)$$

$$h = (v \cdot T) / 2: i = w + g \text{ 'Fórmula } Z \cdot Y_c / 2 \text{ (pol)}$$

$$J = h \cdot \cos(i) + 1 \text{ 'Fórmula } Z \cdot Y_c / 2 + 1 \text{ (rec)}$$

$$K = h \cdot \sin(i) \text{ 'Fórmula } Z \cdot Y_c / 2 + 1 \text{ (rec)}$$

$$m = \text{hip}(J, K)$$

$$o = \text{ang}(K, m)$$

$$o = 180 \cdot o / \pi$$

'Para frmdatos presentación ***** pi / larga *****

$$\text{alfa} = \text{Sqr}(h \cdot 2): \text{beta} = i / 2 \text{ 'Fórmula } (Z \cdot Y_c)^{0.5} \text{ (pol)}$$

$$\text{alfa1} = \text{alfa} \cdot \cos(\text{beta}): \text{beta1} = \text{alfa} \cdot \sin(\text{beta}) \text{ '(rec)}$$

$$\text{londa} = (2 \cdot \pi / \text{beta1}) \cdot 1.609$$

$$\text{velo} = \text{londa} \cdot \text{fre}$$

```
frmdatosabcd\textctepropa.Text = "μ=" & Format(alfa1, "0.00000") & " , β=" &
Format(beta1, "0.00000")
```

```
frmdatosabcd\textlonda.Text = Format(londa, "0.00000")
```

```
frmdatosabcd\textvelo.Text = Format(velo, "0.00000")
```

$$\text{alfa2} = v / T: \text{beta2} = w - g \text{ 'Fórmula } Z / Y_c \text{ (pol)}$$

$$\text{alfa3} = \text{Sqr}(\text{alfa2}): \text{beta3} = \text{beta2} / 2 \text{ 'Fórmula } (Z / Y_c)^{0.5} = Z_c \text{ (pol)}$$


```
voltaje = cbovoltaje.Text
sil = voltaje ^ 2 / alfa3
```

```
frmdatosabcd!textzc.Text = "Zc= " & Format(alfa3, "0.00000") & " , Ç=" &
Format(beta3 * 180 / pi, "0.00000") & "°"
```

```
frmdatosabcd!textsil.Text = Format(sil, "0.00000")
'fin de presentación frmdatos ***** pi / larga *****
```

```
h = (v * T) / 4: i = w + g 'Fórmula Z*Yc/4 (pol)
p = h * Cos(i) + 1 'Fórmula Z*Yc/4 + 1 (rec)
q = h * Sin(i) 'Fórmula Z*Yc/4 + 1 (rec)
r = hip(p, q) 'Fórmula Z*Yc/4 + 1 (pol)
s = ang(q, r) 'Fórmula Z*Yc/4 + 1 (pol)
r = r * T: s = s + g 'Fórmula (Z*Yc/4 + 1)*Yc (pol)
s = 180 * s / pi
```

```
textA.Text = Format(m, "0.00000") & " , Ø= " & Format(o, "0.00000") & " °"
textD.Text = Format(m, "0.00000") & " , Ç= " & Format(o, "0.00000") & " °"
textB.Text = Format(A, "0.00000") & " , β= " & Format(B, "0.00000") & " °"
textC.Text = Format(r, "0.00000") & " , μ= " & Format(s, "0.00000") & " °"
```

```
GoTo Line6 ' *** fin de rutina de "pi" L / Larga
```

```
Line10: ' *** rutina para "te" L / larga ***
```

```
f(1) = E(3) * v 'definición de: Z' (pol)
f(2) = E(4) + w 'definición de: Z' (pol)
```

```
f(3) = d(7) * T 'definición de Y' (pol)
f(4) = d(8) + g 'definición de Y' (pol)
```

```
Z = f(1)
A = Z
ang1 = f(2)
B = 180 * ang1 / pi
v = Z: w = ang1
```

```
T = f(3): g = f(4)
```

```
h = (v * T) / 2: i = w + g 'Fórmula Z*Yc/2 (pol)
J = h * Cos(i) + 1 'Fórmula Z*Yc/2 + 1 (rec)
K = h * Sin(i) 'Fórmula Z*Yc/2 + 1 (rec)
m = hip(J, K)
o = ang(K, m)
o = 180 * o / pi
```

```
'Para frmdatos presentación ***** T / larga *****
alfa = Sqr(h * 2): beta = i / 2 'Fórmula (Z*Yc)^0.5 (pol)
alfa1 = alfa * Cos(beta): beta1 = alfa * Sin(beta) '(rec)
londa = (2 * pi / beta1) * 1.609
velo = londa * fre
```

```
frmdatosabcd!textctepropa.Text = "μ=" & Format(alfa1, "0.00000") & " , β=" &
Format(beta1, "0.00000")
frmdatosabcd!textlonda.Text = Format(londa, "0.00000")
frmdatosabcd!textvelo.Text = Format(velo, "0.00000")
```

```
alfa2 = v / T: beta2 = w - g 'Fórmula Z/Yc (pol)
alfa3 = Sqr(alfa2): beta3 = beta2 / 2 'Fórmula (Z/Yc)^0.5 = Zc (pol)
```

```
voltaje = cbvoltaje.Text
sil = voltaje ^ 2 / alfa3
```

```
frmdatosabcd!textzc.Text = "Zc= " & Format(alfa3, "0.00000") & " , Ç=" &
Format(beta3 * 180 / pi, "0.00000") & " °"
```

```
frmdatosabcd!textsil.Text = Format(sil, "0.00000")
'fin de presentación frmdatos ***** T / larga *****
```

```
h = (v * T) / 4: i = w + g 'Fórmula Z*Yc/4 (pol)
p = h * Cos(i) + 1 'Fórmula Z*Yc/4 + 1 (rec)
q = h * Sin(i) 'Fórmula Z*Yc/4 + 1 (rec)
r = hip(p, q) 'Fórmula Z*Yc/4 + 1 (pol)
s = ang(q, r) 'Fórmula Z*Yc/4 + 1 (pol)
r = r * A: s = s + ang1 'Fórmula (Z*Yc/4 + 1)*Z (pol)
s = 180 * s / pi
g = 180 * g / pi
```

```
textA.Text = Format(m, "0.00000") & " , Ø= " & Format(o, "0.00000") & " °"
textD.Text = Format(m, "0.00000") & " , Ç= " & Format(o, "0.00000") & " °"
textB.Text = Format(r, "0.00000") & " , β= " & Format(s, "0.00000") & " °"
textC.Text = Format(T, "0.00000") & " , μ= " & Format(g, "0.00000") & " °"
```

Line4:

Line5:

Line6:

final:

End Sub

Private Sub btndatos_Click()

frmdatosabcd.Show modal

End Sub

```
Private Sub btniracables_Click()  
Hide  
frmCables.Show  
End Sub
```

```
Private Sub cbotipolinea_Click()  
tipolinea = cbotipolinea.Text
```

```
If tipolinea = "Corta" Then  
    optpi.Enabled = False  
    optte.Enabled = False  
    btndatos.Enabled = False  
Else  
    optpi.Enabled = True  
    optte.Enabled = True  
End If  
End Sub
```

```
Private Sub Form_Activate()  
texttipo.Text = frmCables!texttipo.Text  
textinduct.Text = L  
textcapa.Text = c  
textresi.Text = rac  
End Sub
```

```
Private Sub Form_Load()  
texttipo.Text = frmCables!texttipo.Text  
textinduct.Text = L  
textcapa.Text = c  
textresi.Text = resi!textrac75nc.Text / resi!ncf.Text  
textresi.Text = rac  
End Sub
```

```
Private Sub iracapac_Click()  
Hide  
Capac.Show  
End Sub
```

```
Private Sub irainduct_Click()  
Hide  
Induct.Show  
End Sub
```

```
Private Sub irairesi_Click()  
Hide  
resi.Show
```

```
End Sub

Private Sub salirCtes_Click()
Unload Me
Param.Show
End Sub

Private Sub textcapa_Change()
If Val(textresi.Text) = 0 Or Val(textinduct.Text) = 0 Or Val(textcapa.Text) = 0
Then
    btncalcular.Enabled = False
    Else
    btncalcular.Enabled = True
End If

End Sub

Private Sub textinduct_Change()
If Val(textresi.Text) = 0 Or Val(textinduct.Text) = 0 Or Val(textcapa.Text) = 0
Then
    btncalcular.Enabled = False
    Else
    btncalcular.Enabled = True
End If
End Sub

Private Sub textkm_Change()
km = textkm.Text
End Sub

Private Sub textresi_Change()
rac = textresi.Text

If Val(textresi.Text) = 0 Or Val(textinduct.Text) = 0 Or Val(textcapa.Text) = 0
Then
    btncalcular.Enabled = False
    Else
    btncalcular.Enabled = True
End If
End Sub
```

BASE DE DATOS DE AISLADORES: FORMAS

(frmdataaisladores.frm; frmmasaisladores.frm; frmmasaisladores2.frm;
frmmasaisladores3.frm y frmmasaisladores4.frm)

FRMDATAAISLADORES.FRM¹

```
Private Sub btnactual_Click()  
frmactualizar.Show vbModal  
End Sub
```

```
Private Sub btnverfrmaisla_Click()  
frmaisla.textespesor.Text = textespesor.Text  
frmaisla.textancho.Text = textancho.Text  
frmaisla.textpeso.Text = textpeso.Text  
frmaisla.textfuga.Text = textdisfuga.Text
```

```
Unload Me  
frmaisla.Show  
End Sub
```

```
Private Sub btnverfrmmasaisladores_Click()  
Unload Me  
frmmasaisladores.Show  
End Sub
```

```
Private Sub btnverherrajes_Click()  
Unload Me  
frmherrajes.Show  
End Sub
```

¹ El código se repite para las otras formas, la diferencia es el nombre que se le asigna a la forma ("Name")

```

Private Sub cbonivel_Click()
Rem ***** Cálculo de la cadena de aisladores por contaminación *****
contaminacion = cbonivel.Text 'Selecciona los 4 casos
Select Case contaminacion
    Case "Ligera"
        'kfe = 16
        kfe = frmactualizar.textligera.Text
        GoTo Line1
    Case "Media"
        'kfe = 20
        kfe = frmactualizar.textmedia.Text
        GoTo Line1
    Case "Alta"
        'kfe = 25
        kfe = frmactualizar.textalta.Text
        GoTo Line1
    Case "Muy Alta"
        'kfe = 31
        kfe = frmactualizar.textmalta.Text
        GoTo Line1
    Case Else
        Item = MsgBox("No se ha definido el nivel de contaminación: " &
contaminacion, 48, "Nivel de contaminación sin definir")
        GoTo Line2
End Select

```

Line1:

```

espesor = (Val(textespesor.Text)) / 1000 'Lee espesor
disfuga = Val(textdisfuga.Text)
volmax = Val(textvolmax.Text)
If volmax = 0 Then

```

```

    pel = MsgBox("Defina el voltaje máximo en KV ! ", vbDefaultButton1, "Voltaje
no definido (DELTA)")

```

```

    textvolmax.SetFocus

```

```

    GoTo abajo

```

```

End If

```

```

If disfuga = 0 Then

```

```

    pel = MsgBox("La distancia de fuga no puede ser cero, defina el dato ! ",
vbCritical, "Distancia de fuga no definida (DELTA)")

```

```

    textdisfuga.SetFocus

```

```

    GoTo abajo

```

```

End If

```

```

ver = "Nivel de cotaminación: " & contaminacion & " ; Kfe = " & kfe & " mm/KV "
cbonivel.ToolTipText = ver

```

```

numais = kfe * volmax / disfuga 'Número de aisladores

```

```

textnumais.Text = Int(numais) + 1

```

```

lonaisreal = (textnumais.Text) * espesor 'Longitud real

```

```

textlonreal.Text = lonaisreal

```

```

relon = frmaisla!cborelon.Text 'Lee la relación de longitud

```

```

discorregida = relon * lonaisreal 'Distancia torre-conductor

```

```

textdiscorregida.Text = discorregida

```

```

Line2:

```

```

    cbonivel.SetFocus

```

```

abajo:

```

End Sub

Private Sub Form_Activate()

textvolmax.Text = (frmaisla!textvolmax.Text)

textespesor.Text = frmaisla!textespesor.Text

textancho.Text = frmaisla!textancho.Text

textpeso.Text = frmaisla!textpeso.Text

textdisfuga.Text = frmaisla!textfuga.Text

End Sub

Private Sub Form_Load()

If contaminacion = "" Then

 contaminacion = "Ligera"

Else

 contaminacion = contaminacion

End If

textvolmax.Text = (frmaisla!textvolmax.Text)

textespesor.Text = frmaisla!textespesor.Text

textancho.Text = frmaisla!textancho.Text

textpeso.Text = frmaisla!textpeso.Text

textdisfuga.Text = frmaisla!textfuga.Text

imga.Picture = LoadPicture("C:\Delta\Graficas\aislador2.bmp")

imgb.Picture = LoadPicture("C:\Delta\Graficas\aislador3.bmp")

End Sub

Private Sub imga_Click()

Rem Información del accesorio

nombre = " Aislador de suspensión"


```
fabricante = " A. B. CHANCE Co."
nc = " 66250"
maxtension = "6803.89 Kgf (15000 Lbs)"
diame = 254
alto = 146.05 'Espesor = Altura
fuga = 279.4
Peso = 5.13
```

```
msj = "Tipo de Aislador: " & nombre & Chr$(10) & Chr$(13)
msj = msj & "Fabricante: " & fabricante & Chr$(10) & Chr$(13)
msj = msj & "Catálogo Número: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Máxima Tensión Eléctrica y Mecánica: " & maxtension & Chr$(10)
& Chr$(13)
msj = msj & "Diámetro: " & diame & " mm" & Chr$(10) & Chr$(13)
msj = msj & "Altura: " & alto & " mm" & Chr$(10) & Chr$(13)
msj = msj & "Distancia de Fuga: " & fuga & " mm" & Chr$(10) & Chr$(13)
msj = msj & "Peso: " & Peso & " Kgf" & Chr$(10) & Chr$(13)
```

```
MsgBox msj
```

```
textespesor.Text = alto 'Espesor = Altura
textancho.Text = diame
textdisfuga.Text = fuga
textpeso.Text = Peso
End Sub
```

```
Private Sub imgb_Click()
Rem Información del accesorio
nombre = " Aislador de suspensión"
fabricante = " A. B. CHANCE Co."
nc = " 66255"
```

maxtension = "6803.89 Kgf (15000 Lbs)"

diame = 254

alto = 146.05 'Espesor = Altura

fuga = 292.1

Peso = 5.22

msj = "Tipo de Aislador: " & nombre & Chr\$(10) & Chr\$(13)

msj = msj & "Fabricante: " & fabricante & Chr\$(10) & Chr\$(13)

msj = msj & "Catálogo Número: " & nc & Chr\$(10) & Chr\$(13)

msj = msj & "Máxima Tensión Eléctrica y Mecánica: " & maxtension & Chr\$(10)
& Chr\$(13)

msj = msj & "Diámetro: " & diame & " mm" & Chr\$(10) & Chr\$(13)

msj = msj & "Altura: " & alto & " mm" & Chr\$(10) & Chr\$(13)

msj = msj & "Distancia de Fuga: " & fuga & " mm" & Chr\$(10) & Chr\$(13)

msj = msj & "Peso: " & Peso & " Kgf" & Chr\$(10) & Chr\$(13)

MsgBox msj

textespesor.Text = alto 'Espesor = Altura

textancho.Text = diame

textdisfuga.Text = fuga

textpeso.Text = Peso

End Sub

Private Sub textespesor_Change()

If Val(textespesor.Text) = 0 And Val(textancho.Text) = 0 Then

 btnverfrmaisla.Enabled = False

 Else

 btnverfrmaisla.Enabled = True

End If

End Sub

BASE DE DATOS DE HERRAJES: FORMAS

(frmherrajes.frm; frmmasherrajes1.frm; frmmasherrajes2.frm; frmmasherrajes3.frm)

FRMHERRAJES.FRM¹

```
Private Sub btnverfrmaisla_Click()
```

```
Unload Me
```

```
frmaisla.Show
```

```
End Sub
```

```
Private Sub btnverfrmdataaisladores_Click()
```

```
Unload Me
```

```
frmdataaisladores.Show
```

```
End Sub
```

```
Private Sub btnverfrmmasherrajes1_Click()
```

```
Unload Me
```

```
frmmasherrajes1.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
pesot = 0
```

```
largo = 0
```

```
imga.Picture = LoadPicture("C:\Delta\Graficas\herrajes1.bmp")
```

```
imgb.Picture = LoadPicture("C:\Delta\Graficas\herrajes3.bmp")
```

```
imgc.Picture = LoadPicture("C:\Delta\Graficas\herrajes4.bmp")
```

```
imgd.Picture = LoadPicture("C:\Delta\Graficas\herrajes5.bmp")
```

```
imge.Picture = LoadPicture("C:\Delta\Graficas\herrajes7.bmp")
```

```
imgf.Picture = LoadPicture("C:\Delta\Graficas\herrajes8.bmp")
```

¹ El código se repite para las otras formas, la diferencia es el nombre que se le asigna a la forma ("Name")

```
imgg.Picture = LoadPicture("C:\Delta\Graficas\herrajes36.bmp")
imggh.Picture = LoadPicture("C:\Delta\Graficas\herrajes37.bmp")
imgi.Picture = LoadPicture("C:\Delta\Graficas\herrajes9.bmp")
imgj.Picture = LoadPicture("C:\Delta\Graficas\herrajes10.bmp")
imgk.Picture = LoadPicture("C:\Delta\Graficas\herrajes9.bmp")
imgl.Picture = LoadPicture("C:\Delta\Graficas\herrajes11.bmp")
End Sub
```

```
Private Sub imga_Click()
```

```
imgb.Visible = False
```

```
imgc.Visible = False
```

```
imgd.Visible = False
```

```
imge.Visible = False
```

```
imgf.Visible = False
```

```
imgg.Visible = False
```

```
imggh.Visible = False
```

```
imgi.Visible = False
```

```
imgj.Visible = False
```

```
imgk.Visible = False
```

```
imgl.Visible = False
```

```
imga.Height = 5175
```

```
imga.Width = 5175
```

```
imga.Left = (Screen.Width - 5175) / 2
```

```
imga.Top = (Screen.Height - 5175) / 2
```

```
End Sub
```

```
Private Sub imga_DbClick()
```

```
imgb.Visible = True
```

```
imgc.Visible = True
```

```
imgd.Visible = True  
imge.Visible = True  
imgf.Visible = True  
imgg.Visible = True  
imgh.Visible = True  
imgi.Visible = True  
imgj.Visible = True  
imgk.Visible = True  
imgl.Visible = True
```

```
imga.Height = 1695  
imga.Width = 1695  
imga.Left = 240  
imga.Top = 240  
End Sub
```

```
Private Sub imgb_Click()  
imga.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgf.Visible = False  
imgg.Visible = False  
imgh.Visible = False  
imgi.Visible = False  
imgj.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgb.Height = 3500  
imgb.Width = 6495
```

```

imgb.Left = (Screen.Width - 6495) / 2
imgb.Top = (Screen.Height - 3500) / 2

```

Rem Información del accesorio

```

nombre = " Mordaza"
tipo = " Aluminio HAS"
nc = " HAS-62-S"
accesorios = " Socket (SA-04)"
esfuerzo = 7711
Peso = 1.45
longitud = 57.94

```

```

msj = "Nombre del herraje: " & nombre & Chr$(10) & Chr$(13)
msj = msj & "Tipo: " & tipo & Chr$(10) & Chr$(13)
msj = msj & "Catálogo Número: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Accesorios: " & accesorios & Chr$(10) & Chr$(13)
msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr$(10) & Chr$(13)
msj = msj & "Peso: " & Peso & " Kgf" & Chr$(10) & Chr$(13)
msj = msj & "Longitud vertical (H) : " & longitud & " mm"
MsgBox msj

```

```

pesoh = Peso
lon = longitud
largo = largo + lon
pesot = pesot + pesoh

```

```

msje = "La longitud y el peso total de los herrajes seleccionados son: " &
Chr$(10) & Chr$(13)
msje = msje & Chr$(10) & Chr$(13)
msje = msje & "Longitud: " & largo & " mm" & Chr$(10) & Chr$(13)
msje = msje & Chr$(10) & Chr$(13)

```

```
msje = msje & "Peso: " & pesot & " Kgf"
```

```
MsgBox msje
```

```
End Sub
```

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

```
imga.Visible = True
```

```
imgc.Visible = True
```

```
imgd.Visible = True
```

```
imge.Visible = True
```

```
imgf.Visible = True
```

```
imgg.Visible = True
```

```
imgh.Visible = True
```

```
imgi.Visible = True
```

```
imgj.Visible = True
```

```
imgk.Visible = True
```

```
imgl.Visible = True
```

```
imgb.Height = 1695
```

```
imgb.Width = 2295
```

```
imgb.Left = 2040
```

```
imgb.Top = 240
```

```
End Sub
```

```
Private Sub imgc_Click()
```

```
imgb.Visible = False
```

```
imga.Visible = False
```

```
imgd.Visible = False
```

```
imge.Visible = False
```

```
imgf.Visible = False
```

```
imgg.Visible = False
```

```
imgb.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgf.Visible = False
```

```
imgc.Height = 5175  
imgc.Width = 5175  
imgc.Left = (Screen.Width - 5175) / 2  
imgc.Top = (Screen.Height - 5175) / 2  
End Sub
```

```
Private Sub imgc_DblClick()  
imgb.Visible = True  
imga.Visible = True  
imgd.Visible = True  
imge.Visible = True  
imgf.Visible = True  
imgg.Visible = True  
imgb.Visible = True  
imgi.Visible = True  
imgj.Visible = True  
imgk.Visible = True  
imgl.Visible = True
```

```
imgc.Height = 1695  
imgc.Width = 1695  
imgc.Left = 4440  
imgc.Top = 240  
End Sub
```



```
Private Sub imgd_Click()
```

```
imga.Visible = False
```

```
imgb.Visible = False
```

```
imgc.Visible = False
```

```
imge.Visible = False
```

```
imgf.Visible = False
```

```
imgg.Visible = False
```

```
imgh.Visible = False
```

```
imgi.Visible = False
```

```
imgj.Visible = False
```

```
imgk.Visible = False
```

```
imgl.Visible = False
```

```
imgd.Height = 3500
```

```
imgd.Width = 6495
```

```
imgd.Left = (Screen.Width - 6495) / 2
```

```
imgd.Top = (Screen.Height - 3500) / 2
```

```
Rem Información del accesorio
```

```
nombre = " Mordaza"
```

```
tipo = " Aluminio HAC"
```

```
nc = " HAS-118-C"
```

```
accesorios = " Clevis (CA-10)"
```

```
esfuerzo = 11340
```

```
Peso = 2.4
```

```
longitud = 69.85
```

```
msj = "Nombre del herraje: " & nombre & Chr$(10) & Chr$(13)
```

```
msj = msj & "Tipo: " & tipo & Chr$(10) & Chr$(13)
```

```
msj = msj & "Catálogo Número: " & nc & Chr$(10) & Chr$(13)
```

```
msj = msj & "Accesorios: " & accesorios & Chr$(10) & Chr$(13)
```

```
msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr$(10) & Chr$(13)
```

```
msj = msj & "Peso: " & Peso & " Kgf" & Chr$(10) & Chr$(13)
```

```
msj = msj & "Longitud vertical (H) : " & longitud & " mm"
```

```
MsgBox msj
```

```
pesoh = Peso
```

```
lon = longitud
```

```
largo = largo + lon
```

```
pesot = pesot + pesoh
```

```
msje = "La longitud y el peso total de los herrajes seleccionados son: " &  
Chr$(10) & Chr$(13)
```

```
msje = msje & Chr$(10) & Chr$(13)
```

```
msje = msje & "Longitud: " & largo & " mm" & Chr$(10) & Chr$(13)
```

```
msje = msje & Chr$(10) & Chr$(13)
```

```
msje = msje & "Peso: " & pesot & " Kgf"
```

```
MsgBox msje
```

```
End Sub
```

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

```
imga.Visible = True
```

```
imgb.Visible = True
```

```
imgc.Visible = True
```

```
imge.Visible = True
```

```
imgf.Visible = True
```

```
imgg.Visible = True
```

```
imgh.Visible = True
```

```
imgi.Visible = True
```

```
imgj.Visible = True
```

```
imgk.Visible = True
```

```
imgl.Visible = True
```

```
imgd.Height = 1695
```

```
imgd.Width = 2295
```

```
imgd.Left = 6240
```

```
imgd.Top = 240
```

```
End Sub
```

```
Private Sub imge_Click()
```

```
imga.Visible = False
```

```
imgb.Visible = False
```

```
imgc.Visible = False
```

```
imgd.Visible = False
```

```
imgf.Visible = False
```

```
imgg.Visible = False
```

```
imgh.Visible = False
```

```
imgi.Visible = False
```

```
imgj.Visible = False
```

```
imgk.Visible = False
```

```
imgl.Visible = False
```

```
imge.Height = 5175
```

```
imge.Width = 5175
```

```
imge.Left = (Screen.Width - 5175) / 2
```

```
imge.Top = (Screen.Height - 5175) / 2
```

```
End Sub
```

```
Private Sub imge_DbClick()
```

```
imga.Visible = True
```

```
imgb.Visible = True
```

```
imgc.Visible = True
```

```
imgd.Visible = True  
imgf.Visible = True  
imgg.Visible = True  
imgh.Visible = True  
imgi.Visible = True  
imgj.Visible = True  
imgk.Visible = True  
imgl.Visible = True
```

```
imge.Height = 1695  
imge.Width = 1695  
imge.Left = 240  
imge.Top = 2040  
End Sub
```

```
Private Sub imgf_Click()
```

```
imga.Visible = False  
imgb.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgg.Visible = False  
imgh.Visible = False  
imgi.Visible = False  
imgj.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgf.Height = 3500  
imgf.Width = 6495  
imgf.Left = (Screen.Width - 6495) / 2
```

imgf.Top = (Screen.Height - 3500) / 2

Rem Información del accesorio

nombre = " Clevis eye"

tipo = " Acero ductil CA"

nc = " CA-10"

esfuerzo = 11340

Peso = 0.77

longitud = 84.14

msj = "Nombre del herraje: " & nombre & Chr\$(10) & Chr\$(13)

msj = msj & "Tipo: " & tipo & Chr\$(10) & Chr\$(13)

msj = msj & "Catálogo Número: " & nc & Chr\$(10) & Chr\$(13)

msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr\$(10) & Chr\$(13)

msj = msj & "Peso: " & Peso & " Kgf" & Chr\$(10) & Chr\$(13)

msj = msj & "Longitud vertical (L+R) : " & longitud & " mm"

MsgBox msj

pesoh = Peso

lon = longitud

largo = largo + lon

pesot = pesot + pesoh

msje = "La longitud y el peso total de los herrajes seleccionados son: " &
Chr\$(10) & Chr\$(13)

msje = msje & Chr\$(10) & Chr\$(13)

msje = msje & "Longitud: " & largo & " mm" & Chr\$(10) & Chr\$(13)

msje = msje & Chr\$(10) & Chr\$(13)

msje = msje & "Peso: " & pesot & " Kgf"

MsgBox msje

End Sub

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

imga.Visible = True

imgb.Visible = True

imgc.Visible = True

imgd.Visible = True

imge.Visible = True

imgg.Visible = True

imgh.Visible = True

imgi.Visible = True

imgj.Visible = True

imgk.Visible = True

imgl.Visible = True

imgf.Height = 1695

imgf.Width = 2295

imgf.Left = 2040

imgf.Top = 2040

End Sub

Private Sub imgg_Click()

imga.Visible = False

imgb.Visible = False

imgc.Visible = False

imgd.Visible = False

imge.Visible = False

imgf.Visible = False

imgg.Visible = False

imgi.Visible = False

```
imgj.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgg.Height = 5175  
imgg.Width = 5175  
imgg.Left = (Screen.Width - 5175) / 2  
imgg.Top = (Screen.Height - 5175) / 2  
End Sub
```

```
Private Sub imgg_DblClick()
```

```
imga.Visible = True  
imgb.Visible = True  
imgc.Visible = True  
imgd.Visible = True  
imge.Visible = True  
imgf.Visible = True  
imgg.Visible = True  
imgh.Visible = True  
imgi.Visible = True  
imgj.Visible = True  
imgk.Visible = True  
imgl.Visible = True
```

```
imgg.Height = 1695  
imgg.Width = 1695  
imgg.Left = 4440  
imgg.Top = 2040  
End Sub
```

```
Private Sub imgh_Click()
```

```
imga.Visible = False
```

```
imgb.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgf.Visible = False  
imgg.Visible = False  
imgi.Visible = False  
imgj.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgh.Height = 3500  
imgh.Width = 6495  
imgh.Left = (Screen.Width - 6495) / 2  
imgh.Top = (Screen.Height - 3500) / 2
```

Rem Información del accesorio

```
nombre = " Socket eye"  
tipo = " Acero ductil SA"  
nc = " SA-04"  
esfuerzo = 8165  
Peso = 0.57  
longitud = 73.03
```

```
msj = "Nombre del herraje: " & nombre & Chr$(10) & Chr$(13)  
msj = msj & "Tipo: " & tipo & Chr$(10) & Chr$(13)  
msj = msj & "Catálogo Número: " & nc & Chr$(10) & Chr$(13)  
msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr$(10) & Chr$(13)  
msj = msj & "Peso: " & Peso & " Kgf" & Chr$(10) & Chr$(13)  
msj = msj & "Longitud vertical (L+R) : " & longitud & " mm"  
MsgBox msj
```



```
pesoh = Peso
lon = longitud
largo = largo + lon
pesot = pesot + pesoh

msje = "La longitud y el peso total de los herrajes seleccionados son: " &
Chr$(10) & Chr$(13)
msje = msje & Chr$(10) & Chr$(13)
msje = msje & "Longitud: " & largo & " mm" & Chr$(10) & Chr$(13)
msje = msje & Chr$(10) & Chr$(13)
msje = msje & "Peso: " & pesot & " Kgf"
MsgBox msje

End Sub

Private Sub imgh_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
imga.Visible = True
imgb.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True
imgl.Visible = True

imgh.Height = 1695
```

```
imgb.Width = 2295  
imgb.Left = 6240  
imgb.Top = 2040  
End Sub
```

```
Private Sub imgc_Click()  
imga.Visible = False  
imgb.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgf.Visible = False  
imgg.Visible = False  
imgb.Visible = False  
imgj.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgi.Height = 5175  
imgi.Width = 5175  
imgi.Left = (Screen.Width - 5175) / 2  
imgi.Top = (Screen.Height - 5175) / 2  
End Sub
```

```
Private Sub imgc_DblClick()  
imga.Visible = True  
imgb.Visible = True  
imgc.Visible = True  
imgd.Visible = True  
imge.Visible = True  
imgf.Visible = True
```

```
imgg.Visible = True  
imgh.Visible = True  
imgj.Visible = True  
imgk.Visible = True  
imgl.Visible = True
```

```
imgi.Height = 1695  
imgi.Width = 1695  
imgi.Left = 240  
imgi.Top = 3840  
End Sub
```

```
Private Sub imgj_Click()  
imga.Visible = False  
imgb.Visible = False  
imgc.Visible = False  
imgd.Visible = False  
imge.Visible = False  
imgf.Visible = False  
imgg.Visible = False  
imgh.Visible = False  
imgi.Visible = False  
imgk.Visible = False  
imgl.Visible = False
```

```
imgj.Height = 5000  
imgj.Width = 5500  
imgj.Left = (Screen.Width - 5500) / 2  
imgj.Top = (Screen.Height - 5000) / 2
```

Rem Información del accesorio

nombre = " Y-Clevis eye"

tipo = " Acero ductil YCS"

nc = " YCS-07"

fig = " 1-YCS"

esfuerzo = 13608

Peso = 0.86

longitud = 92.08

msj = "Nombre del herraje: " & nombre & Chr\$(10) & Chr\$(13)

msj = msj & "Tipo: " & tipo & Chr\$(10) & Chr\$(13)

msj = msj & "Catálogo Número: " & nc & Chr\$(10) & Chr\$(13)

msj = msj & "Figura: " & fig & Chr\$(10) & Chr\$(13)

msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr\$(10) & Chr\$(13)

msj = msj & "Peso: " & Peso & " Kgf" & Chr\$(10) & Chr\$(13)

msj = msj & "Longitud vertical (L+R) : " & longitud & " mm"

MsgBox msj

pesoh = Peso

lon = longitud

largo = largo + lon

pesot = pesot + pesoh

msje = "La longitud y el peso total de los herrajes seleccionados son: " &
Chr\$(10) & Chr\$(13)

msje = msje & Chr\$(10) & Chr\$(13)

msje = msje & "Longitud: " & largo & " mm" & Chr\$(10) & Chr\$(13)

msje = msje & Chr\$(10) & Chr\$(13)

msje = msje & "Peso: " & pesot & " Kgf"

MsgBox msje

End Sub

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

```
    imga.Visible = True
```

```
    imgb.Visible = True
```

```
    imgc.Visible = True
```

```
    imgd.Visible = True
```

```
    imge.Visible = True
```

```
    imgf.Visible = True
```

```
    imgg.Visible = True
```

```
    imgh.Visible = True
```

```
    imgi.Visible = True
```

```
    imgk.Visible = True
```

```
    imgl.Visible = True
```

```
    imgj.Height = 1695
```

```
    imgj.Width = 2295
```

```
    imgj.Left = 2040
```

```
    imgj.Top = 3840
```

```
End Sub
```

```
Private Sub imgk_Click()
```

```
    imga.Visible = False
```

```
    imgb.Visible = False
```

```
    imgc.Visible = False
```

```
    imgd.Visible = False
```

```
    imge.Visible = False
```

```
    imgf.Visible = False
```

```
    imgg.Visible = False
```

```
    imgh.Visible = False
```

```
    imgi.Visible = False
```

```
imgj.Visible = False  
imgl.Visible = False
```

```
imgk.Height = 5175  
imgk.Width = 5175  
imgk.Left = (Screen.Width - 5175) / 2  
imgk.Top = (Screen.Height - 5175) / 2  
End Sub
```

```
Private Sub imgk_DblClick()  
imga.Visible = True  
imgb.Visible = True  
imgc.Visible = True  
imgd.Visible = True  
imge.Visible = True  
imgf.Visible = True  
imgg.Visible = True  
imgh.Visible = True  
imgi.Visible = True  
imgj.Visible = True  
imgl.Visible = True  
imgk.Height = 1695  
imgk.Width = 1695  
imgk.Left = 4440  
imgk.Top = 3840  
End Sub
```

```
Private Sub imgl_Click()  
imga.Visible = False  
imgb.Visible = False  
imgc.Visible = False
```

```
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False
imgl.Height = 5000
imgl.Width = 5500
imgl.Left = (Screen.Width - 5500) / 2
imgl.Top = (Screen.Height - 5000) / 2
```

Rem Información del accesorio

```
nombre = " Y-Clevis eye"
```

```
tipo = " Acero ductil YCS"
```

```
nc = " YCS-16-90-40"
```

```
fig = " 2-YCS-90"
```

```
esfuerzo = 18144
```

```
Peso = 1.13
```

```
longitud = 80.96
```

```
msj = "Nombre del herraje: " & nombre & Chr$(10) & Chr$(13)
```

```
msj = msj & "Tipo: " & tipo & Chr$(10) & Chr$(13)
```

```
msj = msj & "Catálogo Número: " & nc & Chr$(10) & Chr$(13)
```

```
msj = msj & "Figura: " & fig & Chr$(10) & Chr$(13)
```

```
msj = msj & "Ultimo esfuerzo: " & esfuerzo & " Kgf" & Chr$(10) & Chr$(13)
```

```
msj = msj & "Peso: " & Peso & " Kgf" & Chr$(10) & Chr$(13)
```

```
msj = msj & "Longitud vertical (L+R) : " & longitud & " mm"
```

```
MsgBox msj
```

```
pesoh = Peso
```

```
lon = longitud
largo = largo + lon
pesot = pesot + pesoh
```

```
msje = "La longitud y el peso total de los herrajes seleccionados son: " &
Chr$(10) & Chr$(13)
```

```
msje = msje & Chr$(10) & Chr$(13)
```

```
msje = msje & "Longitud: " & largo & " mm" & Chr$(10) & Chr$(13)
```

```
msje = msje & Chr$(10) & Chr$(13)
```

```
msje = msje & "Peso: " & pesot & " Kgf"
```

```
MsgBox msje
```

```
End Sub
```

```
Private Sub img1_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
imga.Visible = True
```

```
imgb.Visible = True
```

```
imgc.Visible = True
```

```
imgd.Visible = True
```

```
imge.Visible = True
```

```
imgf.Visible = True
```

```
imgg.Visible = True
```

```
imgh.Visible = True
```

```
imgi.Visible = True
```

```
imgj.Visible = True
```

```
imgk.Visible = True
```

```
imgl.Height = 1695
```

```
imgl.Width = 2295
```

```
imgl.Left = 6240
```

```
imgl.Top = 3840
```

```
End Sub
```


BASE DE DATOS DE ESTRUCTURAS: FORMAS

(frmestruc.frm)

```
Private Sub btnverfrmaisal_Click()  
Unload Me  
frmaisal.Show  
End Sub
```

```
Private Sub img138b_Click()  
estruc = "Torre"  
volop = 138  
nc = 2  
ncg = 2  
van = 230
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)  
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)  
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)  
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)  
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj  
End Sub
```

```
Private Sub img161b_Click()  
estruc = "Torre"  
volop = 161  
nc = 2  
ncg = 1  
van = 274
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)  
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)  
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)  
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)  
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj  
End Sub
```

```
Private Sub img230b_Click()  
estruc = "Torre"  
volop = 230  
nc = 2
```

```
ncg = 2
van = 274
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj
End Sub
```

```
Private Sub img287b_Click()
estruc = "Torre"
volop = 287
nc = 2
ncg = 2
van = 260
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj
End Sub
```

```
Private Sub Form_Load()
Rem Lee el botón de opción de la frmcalcable
If frmcalcable.luckto.Value = True Then
```

```
    imga.Picture = LoadPicture("\delta\Graficas\torre12.bmp")
    imgb.Picture = LoadPicture("\delta\Graficas\torre14.bmp")
    imgc.Picture = LoadPicture("\delta\Graficas\torre4.bmp")
    imgd.Picture = LoadPicture("\delta\Graficas\torre6.bmp")
    imge.Picture = LoadPicture("\delta\Graficas\torre8.bmp")
    imgf.Picture = LoadPicture("\delta\Graficas\marco1.bmp")
    imgg.Picture = LoadPicture("\delta\Graficas\marco3.bmp")
    imgh.Picture = LoadPicture("\delta\Graficas\torre18.bmp")
    imgi.Picture = LoadPicture("\delta\Graficas\torre17.bmp")
    imgj.Picture = LoadPicture("\delta\Graficas\torre19.bmp")
    imgk.Picture = LoadPicture("\delta\Graficas\torre16.bmp")
```

```
    lblimga.Caption = "Torre 115 KV"
    lblimgb.Caption = "Torre 138 KV"
    lblimgc.Caption = "Torre 161 KV"
    lblimgd.Caption = "Torre 230 KV"
```

```

Iblimge.Caption = "Torre 287 KV"
Iblimgf.Caption = "Marco 115 KV"
Iblimgg.Caption = "Marco 161 KV"
Iblimgh.Caption = "Marco 345 KV"
Iblimgi.Caption = "Torre 345 KV"
Iblimgj.Caption = "Torre 230 KV"
Iblimgk.Caption = "Torre 500 KV"

```

```
Elseif frmcalcable!luckto.Value = False Then
```

```

imga.Picture = LoadPicture("\delta\Graficas\torre13.bmp")
imgb.Picture = LoadPicture("\delta\Graficas\torre15.bmp")
imgc.Picture = LoadPicture("\delta\Graficas\torre5.bmp")
imgd.Picture = LoadPicture("\delta\Graficas\torre7.bmp")
imge.Picture = LoadPicture("\delta\Graficas\torre9.bmp")

```

```

Iblimga.Caption = "Torre 115 KV"
Iblimgb.Caption = "Torre 138 KV"
Iblimgc.Caption = "Torre 161 KV"
Iblimgd.Caption = "Torre 230 KV"
Iblimge.Caption = "Torre 287 KV"

```

```
End If
```

```
Rem Lee el botón de opción de la frmcalcable
```

```
If frmcalcable!dckto.Value = False Then
```

```

imga.Picture = LoadPicture("\delta\Graficas\torre12.bmp")
imgb.Picture = LoadPicture("\delta\Graficas\torre14.bmp")
imgc.Picture = LoadPicture("\delta\Graficas\torre4.bmp")
imgd.Picture = LoadPicture("\delta\Graficas\torre6.bmp")
imge.Picture = LoadPicture("\delta\Graficas\torre8.bmp")
imgf.Picture = LoadPicture("\delta\Graficas\marco1.bmp")
imgg.Picture = LoadPicture("\delta\Graficas\marco3.bmp")
imgh.Picture = LoadPicture("\delta\Graficas\torre18.bmp")
imgi.Picture = LoadPicture("\delta\Graficas\torre17.bmp")
imgj.Picture = LoadPicture("\delta\Graficas\torre19.bmp")
imgk.Picture = LoadPicture("\delta\Graficas\torre16.bmp")

```

```

Iblimga.Caption = "Torre 115 KV"
Iblimgb.Caption = "Torre 138 KV"
Iblimgc.Caption = "Torre 161 KV"
Iblimgd.Caption = "Torre 230 KV"
Iblimge.Caption = "Torre 287 KV"
Iblimgf.Caption = "Marco 115 KV"
Iblimgg.Caption = "Poste 161 KV"
Iblimgh.Caption = "Marco 345 KV"
Iblimgi.Caption = "Torre 345 KV"
Iblimgj.Caption = "Torre 230 KV"

```

```
lblimgk.Caption = "Torre 500 KV"
```

```
ElseIf frmcalcable!dckto.Value = True Then
```

```
imga.Picture = LoadPicture("\delta\Graficas\torre13.bmp")
imgb.Picture = LoadPicture("\delta\Graficas\torre15.bmp")
imgc.Picture = LoadPicture("\delta\Graficas\torre5.bmp")
imgd.Picture = LoadPicture("\delta\Graficas\torre7.bmp")
imge.Picture = LoadPicture("\delta\Graficas\torre9.bmp")
```

```
lblimga.Caption = "Torre 115 KV"
lblimgb.Caption = "Torre 138 KV"
lblimgc.Caption = "Torre 161 KV"
lblimgd.Caption = "Torre 230 KV"
lblimge.Caption = "Torre 287 KV"
```

```
End If
```

```
End Sub
```

```
Private Sub imga_Click()
```

```
imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False
```

```
imga.Height = 5000
imga.Width = 5500
imga.Left = (Screen.Width - 5500) / 2
imga.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
```

```
tipestruct = 1 'Define el caso de selección
estruc = "Torre autosoportada"
volop = 115
nc = 1
nCG = 1
van = 213
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
```

```

msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"

```

```

MsgBox msj

```

```

Else

```

```

tipestruct = 3 'Define el caso de selección
estruc = "Torre autosoportada"
volop = 115
nc = 2
ncg = 1
van = 213

```

```

msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"

```

```

MsgBox msj

```

```

End If

```

```

End Sub

```

```

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

```

```

imgb.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgh.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True

```

```

imga.Height = 1455
imga.Width = 1575
imga.Left = 2280
imga.Top = 360
End Sub

```

```

Private Sub imgb_Click()

```

```

imga.Visible = False
imgc.Visible = False
imgd.Visible = False

```

```

imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False

```

```

imgb.Height = 5000
imgb.Width = 5500
imgb.Left = (Screen.Width - 5500) / 2
imgb.Top = (Screen.Height - 5000) / 2

```

```

If frmcalcableluckto.Value = True Then
    tipestruct = 2 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 138
    nc = 1
    ncg = 2
    van = 275

```

```

    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"

```

```

    MsgBox msj

```

```

Else
    tipestruct = 4 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 138
    nc = 2
    ncg = 2
    van = 245

```

```

    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"

```

```

    MsgBox msj

```

```

End If
End Sub

```

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

```
imga.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgh.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True
```

```
imgb.Height = 1455
imgb.Width = 1575
imgb.Left = 4080
imgb.Top = 360
End Sub
```

```
Private Sub imgc_Click()
```

```
imga.Visible = False
imgb.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False
```

```
imgc.Height = 5000
imgc.Width = 5500
imgc.Left = (Screen.Width - 5500) / 2
imgc.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
  tipestruct = 2 'Define el caso de selección
  estruc = "Torre autosoportada"
  volop = 161
  nc = 1
  ncg = 2
  van = 305
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
```

```
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj
```

```
Else
```

```
tipestruct = 4 'Define el caso de selección
estruc = "Torre autosoportada"
volop = 161
nc = 2
ncg = 2
van = 275
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj
```

```
End If
```

```
End Sub
```

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

```
imga.Visible = True
imgb.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgh.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True
```

```
imgc.Height = 1455
imgc.Width = 1575
imgc.Left = 5880
imgc.Top = 360
End Sub
```

```
Private Sub imgd_Click()
```

```
imga.Visible = False
imgb.Visible = False
imgc.Visible = False
imge.Visible = False
imgf.Visible = False
```



```

imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False

```

```

imgd.Height = 5000
imgd.Width = 5500
imgd.Left = (Screen.Width - 5500) / 2
imgd.Top = (Screen.Height - 5000) / 2

```

```

If frmcalcable!luckto.Value = True Then
  tipestruct = 2 'Define el caso de selección
  estruc = "Torre autosoportada"
  volop = 230
  nc = 1
  ncg = 2
  van = 305

```

```

  msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
  msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
  msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
  msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
  msj = msj & "Vano promedio: " & van & " m"

```

```
  MsgBox msj
```

```
Else
```

```

  tipestruct = 4 'Define el caso de selección
  estruc = "Torre autosoportada"
  volop = 230
  nc = 2
  ncg = 2
  van = 275

```

```

  msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
  msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
  msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
  msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
  msj = msj & "Vano promedio: " & van & " m"

```

```
  MsgBox msj
```

```
End If
```

```
End Sub
```

```

Private Sub imgd_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
  imga.Visible = True

```

```
imgb.Visible = True
imgc.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgh.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True
```

```
imgd.Height = 1455
imgd.Width = 1575
imgd.Left = 7680
imgd.Top = 360
End Sub
```

```
Private Sub imge_Click()
imga.Visible = False
imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False
```

```
imge.Height = 5000
imge.Width = 5500
imge.Left = (Screen.Width - 5500) / 2
imge.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
    tipestruct = 2 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 287
    nc = 1
    ncg = 2
    van = 305
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```

    MsgBox msj
Else
    tipestruct = 4 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 287
    nc = 2
    ncg = 2
    van = 260

    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"

```

```

    MsgBox msj
End If
End Sub

```

```

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

```

```

    imga.Visible = True
    imgb.Visible = True
    imgc.Visible = True
    imgd.Visible = True
    imgf.Visible = True
    imgg.Visible = True
    imgh.Visible = True
    imgi.Visible = True
    imgj.Visible = True
    imgk.Visible = True

```

```

    imge.Height = 1455
    imge.Width = 1575
    imge.Left = 2280
    imge.Top = 2520
End Sub

```

```

Private Sub imgf_Click()
    imga.Visible = False
    imgb.Visible = False
    imgc.Visible = False
    imgd.Visible = False
    imge.Visible = False
    imgg.Visible = False
    imgh.Visible = False
    imgi.Visible = False

```

```
imgj.Visible = False
imgk.Visible = False
```

```
imgf.Height = 5000
imgf.Width = 5500
imgf.Left = (Screen.Width - 5500) / 2
imgf.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
    tipestruct = 5 'Define el caso de selección
    estruc = "Marco"
    volop = 115
    nc = 1
    ncg = 2
    van = 183
```

```
    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"
```

```
    MsgBox msj
End If
End Sub
```

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

```
    imga.Visible = True
    imgb.Visible = True
    imgc.Visible = True
    imgd.Visible = True
    imge.Visible = True
    imgg.Visible = True
    imgh.Visible = True
    imgi.Visible = True
    imgj.Visible = True
    imgk.Visible = True
```

```
    imgf.Height = 1455
    imgf.Width = 1575
    imgf.Left = 4080
    imgf.Top = 2520
End Sub
```

```
Private Sub imgg_Click()
    imga.Visible = False
```

```

imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False

```

```

imgg.Height = 5000
imgg.Width = 5500
imgg.Left = (Screen.Width - 5500) / 2
imgg.Top = (Screen.Height - 5000) / 2

```

```

If frmcalcable!luckto.Value = True Then
    tipestruct = 5 'Define el caso de selección
    estruc = "Marco"
    volop = 161
    nc = 1
    ncg = 2
    van = 183

```

```

    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"

```

```

    MsgBox msj
End If
End Sub

```

```

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

```

```

    imga.Visible = True
    imgb.Visible = True
    imgc.Visible = True
    imgd.Visible = True
    imge.Visible = True
    imgf.Visible = True
    imgg.Visible = True
    imgi.Visible = True
    imgj.Visible = True
    imgk.Visible = True

```

```

    imgg.Height = 1455

```

```

imgg.Width = 1575
imgg.Left = 5880
imgg.Top = 2520
End Sub

```

```

Private Sub imgh_Click()
imga.Visible = False
imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgi.Visible = False
imgj.Visible = False
imgk.Visible = False

```

```

imgh.Height = 5000
imgh.Width = 5500
imgh.Left = (Screen.Width - 5500) / 2
imgh.Top = (Screen.Height - 5000) / 2

```

```

If frmcalcable!luckto.Value = True Then
  tipestruct = 5 'Define el caso de selección
  estruc = "Marco"
  volop = 345
  nc = 1
  ncg = 2
  van = 450

```

```

  msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
  msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
  msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
  msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
  msj = msj & "Vano promedio: " & van & " m"

```

```

  MsgBox msj
End If
End Sub

```

```

Private Sub imgh_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
imga.Visible = True
imgb.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True

```

```
imgf.Visible = True
imgg.Visible = True
imgi.Visible = True
imgj.Visible = True
imgk.Visible = True
```

```
imgh.Height = 1455
imgh.Width = 1575
imgh.Left = 7680
imgh.Top = 2520
End Sub
```

```
Private Sub imgi_Click()
imga.Visible = False
imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgh.Visible = False
imgj.Visible = False
imgk.Visible = False
```

```
imgi.Height = 5000
imgi.Width = 5500
imgi.Left = (Screen.Width - 5500) / 2
imgi.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
    tipestruct = 2 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 345
    nc = 1
    ncg = 2
    van = 500
```

```
    msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
    msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
    msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
    msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
    msj = msj & "Vano promedio: " & van & " m"
```

```
    MsgBox msj
End If
End Sub
```

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

```
imga.Visible = True
imgb.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
imgg.Visible = True
```

```
imgi.Height = 1455
imgi.Width = 1575
imgi.Left = 2280
imgi.Top = 4680
End Sub
```

```
Private Sub imgj_Click()
```

```
imga.Visible = False
imgb.Visible = False
imgc.Visible = False
imgd.Visible = False
imge.Visible = False
imgf.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
imgg.Visible = False
```

```
imgj.Height = 5000
imgj.Width = 5500
imgj.Left = (Screen.Width - 5500) / 2
imgj.Top = (Screen.Height - 5000) / 2
```

```
If frmcalcable!luckto.Value = True Then
tipestruct = 2 'Define el caso de selección
estruc = "Torre autosoportada"
volop = 230
nc = 1
ncg = 2
van = 500
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
```



```

msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"

```

```

    MsgBox msj
End If
End Sub

```

```

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

```

```

    imga.Visible = True
    imgb.Visible = True
    imgc.Visible = True
    imgd.Visible = True
    imge.Visible = True
    imgf.Visible = True
    imgg.Visible = True
    imgh.Visible = True
    imgi.Visible = True
    imgk.Visible = True

```

```

    imgj.Height = 1455
    imgj.Width = 1575
    imgj.Left = 4080
    imgj.Top = 4680
End Sub

```

```

Private Sub imgk_Click()

```

```

    imga.Visible = False
    imgb.Visible = False
    imgc.Visible = False
    imgd.Visible = False
    imge.Visible = False
    imgf.Visible = False
    imgg.Visible = False
    imgh.Visible = False
    imgi.Visible = False
    imgj.Visible = False

```

```

    imgk.Height = 5000
    imgk.Width = 5500
    imgk.Left = (Screen.Width - 5500) / 2
    imgk.Top = (Screen.Height - 5000) / 2

```

```

If frmcalcable!luckto.Value = True Then
    tipestruct = 2 'Define el caso de selección
    estruc = "Torre autosoportada"
    volop = 500

```

```
nc = 1
ncg = 2
van = 600
```

```
msj = "Estructura seleccionada: " & estruc & Chr$(10) & Chr$(13)
msj = msj & "Voltaje de operación: " & volop & " KV" & Chr$(10) & Chr$(13)
msj = msj & "Número de circuitos: " & nc & Chr$(10) & Chr$(13)
msj = msj & "Número de cables de guarda: " & ncg & Chr$(10) & Chr$(13)
msj = msj & "Vano promedio: " & van & " m"
```

```
MsgBox msj
End If
End Sub
```

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

```
imga.Visible = True
imgb.Visible = True
imgc.Visible = True
imgd.Visible = True
imge.Visible = True
imgf.Visible = True
imgg.Visible = True
imgg.Visible = True
imgh.Visible = True
imgi.Visible = True
imgj.Visible = True
```

```
imgk.Height = 1455
imgk.Width = 1575
imgk.Left = 5880
imgk.Top = 4680
End Sub
```

```
Private Sub Label1_Click()
```

```
End Sub
```

DATOS PARA UN CIRCUITO

(dat1ckto.frm)

```
Private Sub btncerrar_Click()
frmcorona!textdab.Text = textdab.Text
frmcorona!textdbc.Text = textdbc.Text
frmcorona!textdca.Text = textdca.Text
Hide
End Sub
```

```
Private Sub Form_Load()
textdab.Text = frmcorona!textdab.Text
textdbc.Text = frmcorona!textdbc.Text
textdca.Text = frmcorona!textdca.Text
End Sub
```

```
Private Sub optasime_Click()
If optasime.Value = True Then
  picasime.Visible = True
  picsime.Visible = False
  lbldab.Visible = True
  lbldbc.Visible = True
  lbldca.Visible = True
  lbldab1.Visible = True
  lbldbc1.Visible = True
  lbldca1.Visible = True
  textdab.Visible = True
  textdbc.Visible = True
  textdca.Visible = True
  'para la etiqueta en la forma de induct y Capac indicador
  Induct!lblsime.Visible = False
  Induct!lblasime.Visible = True
  Capac!lblsime.Visible = False
  Capac!lblasime.Visible = True
Else
  picasime.Visible = False
  picsime.Visible = True
  lbldac.Visible = True
  lbldbc.Visible = False
  lbldca.Visible = False
  lbldac1.Visible = True
  lbldbc1.Visible = False
  lbldca1.Visible = False
  textdab.Visible = True
```

```
textdbc.Visible = False
textdca.Visible = False
'para la etiqueta en la forma de induct y Capac indicador
Induct!lblsime.Visible = True
Induct!lblasime.Visible = False
Capac!lblsime.Visible = True
Capac!lblasime.Visible = False
End If
End Sub
```

```
Private Sub opthdesigual_Click()
If opthdesigual.Value = True Then
    pichdesigual.Visible = True
    pichigual.Visible = False
    lblhc1.Visible = True
    lblhb1.Visible = True
    lblhc.Visible = True
    lblhb.Visible = True
    text1hc.Visible = True
    text1hb.Visible = True
Else
    pichigual.Visible = True
    pichdesigual.Visible = False
    lblhc.Visible = False
    lblhb.Visible = False
    lblhc1.Visible = False
    lblhb1.Visible = False
    text1hc.Visible = False
    text1hb.Visible = False
End If
End Sub
```

```
Private Sub opthigual_Click()
If opthigual.Value = True Then
    pichigual.Visible = True
    pichdesigual.Visible = False
    lblhc1.Visible = False
    lblhb1.Visible = False
    lblhc.Visible = False
    lblhb.Visible = False
    text1hc.Visible = False
    text1hb.Visible = False
Else
    pichigual.Visible = False
    pichdesigual.Visible = True
    lblhc.Visible = True
    lblhb.Visible = True
End If
End Sub
```

```
text1hc.Visible = True
text1hb.Visible = True
texthc.Visible = True
texthb.Visible = True
End If
End Sub

Private Sub optsime_Click()
textdbc.Text = 0
textdca.Text = 0

If optsime.Value = True Then
    lbldab.Visible = True
    lbldab1.Visible = True
    textdab.Visible = True
    picsime.Visible = True
    picasime.Visible = False
    lbldbc.Visible = False
    lbldca.Visible = False
    lbldbc1.Visible = False
    lbldca1.Visible = False
    textdbc.Visible = False
    textdca.Visible = False
    'para la etiqueta en la forma de induct y Capac indicador
    Induct!lblsime.Visible = True
    Induct!lblasime.Visible = False
    Capac!lblsime.Visible = True
    Capac!lblasime.Visible = False
    Else
    picsime.Visible = False
    picasime.Visible = True
    lbldac.Visible = True
    lbldbc.Visible = True
    lbldca.Visible = True
    lbldac1.Visible = True
    lbldbc1.Visible = True
    lbldca1.Visible = True
    textdab.Visible = True
    textdbc.Visible = True
    textdca.Visible = True
    'para la etiqueta en la forma de induct y Capac indicador
    Induct!lblsime.Visible = False
    Induct!lblasime.Visible = True
    Capac!lblsime.Visible = False
    Capac!lblasime.Visible = True
End If
End Sub
```

```
Private Sub textdab_Change()  
dab = Val(textdab.Text)  
If opttime.Value = True Then  
    dmg = dab  
Else  
    dbc = Val(textdbc.Text)  
    dca = Val(textdca.Text)  
    dmg = Val(dab * dbc * dca) ^ (1 / 3)  
  
End If  
End Sub
```

```
Private Sub textdbc_Change()  
dab = Val(textdab.Text)  
dbc = Val(textdbc.Text)  
dca = Val(textdca.Text)  
dmg = Val(dab * dbc * dca) ^ (1 / 3)  
End Sub
```

```
Private Sub textdca_Change()  
dab = Val(textdab.Text)  
dbc = Val(textdbc.Text)  
dca = Val(textdca.Text)  
dmg = (dab * dbc * dca) ^ (1 / 3)  
End Sub
```

DATOS PARA DOS CIRCUITOS

(dat2ckto.frm)

```

Private Sub btncerrar_Click()
Rem *****
frmcorona!textdab.Text = textdac2.Text
frmcorona!textdbc.Text = textdbb2.Text
frmcorona!textdca.Text = textdca2.Text
frmcorona!texthab.Text = texthab.Text
frmcorona!texthbc.Text = texthbc.Text

Rem *****
dac2 = Val(textdac2.Text)
dbb2 = Val(textdbb2.Text)
dca2 = Val(textdca2.Text)
dhab = Val(texthab.Text)
dhbc = Val(texthbc.Text)
dab = (Abs(0.25 * (dbb2 - dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dbc = (Abs(0.25 * (dca2 - dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
dac = (Abs(0.25 * (dca2 - dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5
dab2 = (Abs(0.25 * (dbb2 + dac2) ^ 2 + dhab ^ 2)) ^ 0.5
dba2 = (Abs(0.25 * (dca2 + dbb2) ^ 2 + dhbc ^ 2)) ^ 0.5
daa2 = (Abs(0.25 * (dca2 + dac2) ^ 2 + (dhab + dhbc) ^ 2)) ^ 0.5
Hide
End Sub

Private Sub optasime_Click()
If optasime.Value = True Then
    picasime.Visible = True
    picsime.Visible = False
    lbl dab.Enabled = True
    lbl dbc.Enabled = True
    lbl dca.Enabled = True
    textdab.Enabled = True
    textdbc.Enabled = True
    textdca.Enabled = True
Else
    picasime.Visible = False
    picsime.Visible = True
    lbl dac.Enabled = True
    lbl dbc.Enabled = False
    lbl dca.Enabled = False
    textdab.Enabled = True
    textdbc.Enabled = False

```

```
    textdca.Enabled = False
End If
End Sub

Private Sub opttime_Click()
textdbc.Text = 0
textdca.Text = 0

If opttime.Value = True Then
    lbldab.Enabled = True
    textdab.Enabled = True
    picsime.Visible = True
    picasime.Visible = False
    lbldbc.Enabled = False
    lbldca.Enabled = False
    textdbc.Enabled = False
    textdca.Enabled = False
Else
    picsime.Visible = False
    picasime.Visible = True
    lbldac.Enabled = True
    lbldbc.Enabled = True
    lbldca.Enabled = True
    textdab.Enabled = True
    textdbc.Enabled = True
    textdca.Enabled = True
End If

End Sub

Private Sub textdab_Change()
dab = (textdab.Text)
If opttime.Value = True Then
    dmj = dab
Else
    dbc = (textdbc.Text)
    dca = (textdca.Text)
    dmj = (dab * dbc * dca) ^ (1 / 3)
End If
End Sub

Private Sub textdbc_Change()
dab = (textdab.Text)
dbc = (textdbc.Text)
dca = (textdca.Text)
dmj = (dab * dbc * dca) ^ (1 / 3)
End Sub
```



```

Private Sub textdca_Change()
dab = (textdab.Text)
dbc = (textdbc.Text)
dca = (textdca.Text)
dmg = (dab * dbc * dca) ^ (1 / 3)
End Sub

```

```

Private Sub textac_Change()
dac = textdac.Text
dbb = textdbb.Text
dca = textdca.Text
hab = texthab.Text
hbc = texthbc.Text

```

```

disab = (Abs(0.25 * (dbb - dac) ^ 2 + hab ^ 2)) ^ 0.5
disbc = (Abs(0.25 * (dca - dbc) ^ 2 + hbc ^ 2)) ^ 0.5
disca = (Abs(0.25 * (dca - dac) ^ 2 + (hab + hbc) ^ 2)) ^ 0.5

```

```

disab2 = (Abs(0.25 * (dbb + dac) ^ 2 + hab ^ 2)) ^ 0.5
disba2 = (Abs(0.25 * (dca + dbc) ^ 2 + hbc ^ 2)) ^ 0.5
disaa2 = (Abs(0.25 * (dca + dac) ^ 2 + (hab + hbc) ^ 2)) ^ 0.5
End Sub

```

```

Private Sub Form_Activate()
textdac2.SetFocus
End Sub

```

```

Private Sub Form_Load()
textdac2.Text = frmcorona!textdab.Text
textdbb2.Text = frmcorona!textdbc.Text
textdca2.Text = frmcorona!textdca.Text
texthab.Text = frmcorona!texthab.Text
texthbc.Text = frmcorona!texthbc.Text
End Sub

```

```

Private Sub opthdesigual_Click()
If opthdesigual.Value = True Then
    pichdesigual.Visible = True
    pichigual.Visible = False
    lblhab.Visible = True
    lblhbc.Visible = True
    lblhb.Visible = True
    lblhc.Visible = True
    texthab.Visible = True
    texthbc.Visible = True
    text2hb.Visible = True
    text2hc.Visible = True

```

```
Else
pichdesigual.Visible = False
pichigual.Visible = True
lblhab.Visible = False
lblhbc.Visible = False
lblhb.Visible = False
lblhc.Visible = False
texthab.Visible = False
texthbc.Visible = False
text2hb.Visible = False
text2hc.Visible = False
texthab.Text = 0 'Reasigna un valor aunque no sea visible
texthbc.Text = 0 'Reasigna un valor aunque no sea visible
End If
End Sub
```

```
Private Sub opthigual_Click()
If opthigual.Value = True Then
    pichdesigual.Visible = False
    pichigual.Visible = True
    pichdesigual.Visible = False
    pichigual.Visible = True
    lblhab.Visible = False
    lblhbc.Visible = False
    lblhb.Visible = False
    lblhc.Visible = False
    texthab.Visible = False
    texthbc.Visible = False
    text2hb.Visible = False
    text2hc.Visible = False
    texthab.Text = 0 'Reasigna un valor aunque no sea visible
    texthbc.Text = 0 'Reasigna un valor aunque no sea visible
Else
    pichdesigual.Visible = True
    pichigual.Visible = False
    pichdesigual.Visible = True
    pichigual.Visible = False
    lblhab.Visible = True
    lblhbc.Visible = True
    lblhb.Visible = True
    lblhc.Visible = True
    texthab.Visible = True
    texthbc.Visible = True
    text2hb.Visible = True
    text2hc.Visible = True
End If
End Sub
```

NIVEL DE CONTAMINACIÓN

(frmactualizar.frm)

```
Private Sub btnrestaura_Click()
textligera.Text = 16
textmedia.Text = 20
textalta.Text = 25
textmalta.Text = 31
End Sub

Private Sub btnsalir_Click()
Unload Me
End Sub

Private Sub Form_Load()
If Val(kfeligera) = 0 Then
    GoTo final1
Else
    textligera.Text = Val(kfeligera)
End If

final1:
If Val(kfemedia) = 0 Then
    GoTo final2
Else
    textmedia.Text = Val(kfemedia)
End If

final2:
If Val(kfealta) = 0 Then
    GoTo final3
Else
    textalta.Text = Val(kfealta)
End If

final3:
If Val(kfemalta) = 0 Then
    GoTo final
Else
    textmalta.Text = Val(kfemalta)
End If

final:
End Sub
```

```
Private Sub textalta_Click()
textalta.SelStart = 0
textalta.SelLength = Len(textalta)
End Sub

Private Sub textalta_Change()
btnrestaura.Enabled = True
If Val(textligera.Text) = 16 And Val(textmedia.Text) = 20 And Val(textalta.Text) =
25 And Val(textmalta.Text) = 31 Then
    btnrestaura.Enabled = False
End If

'Public kfeligera, kfemedia, kfealta, kfemalta
kfealta = textalta.Text
End Sub

Private Sub textligera_Click()
textligera.SelStart = 0
textligera.SelLength = Len(textligera)
End Sub

Private Sub textligera_Change()
btnrestaura.Enabled = True
If Val(textligera.Text) = 16 And Val(textmedia.Text) = 20 And Val(textalta.Text) =
25 And Val(textmalta.Text) = 31 Then
    btnrestaura.Enabled = False
End If

'Public kfeligera, kfemedia, kfealta, kfemalta
kfeligera = textligera.Text
End Sub

Private Sub textmalta_Click()
textmalta.SelStart = 0
textmalta.SelLength = Len(textmalta)
End Sub

Private Sub textmalta_Change()
btnrestaura.Enabled = True
If Val(textligera.Text) = 16 And Val(textmedia.Text) = 20 And Val(textalta.Text) =
25 And Val(textmalta.Text) = 31 Then
    btnrestaura.Enabled = False
End If

'Public kfeligera, kfemedia, kfealta, kfemalta
kfemalta = textmalta.Text
```

```
End Sub
```

```
Private Sub textmedia_Click()  
textmedia.SelStart = 0  
textmedia.SelLength = Len(textmedia)  
End Sub
```

```
Private Sub textmedia_Change()  
btnrestaura.Enabled = True  
If Val(textligera.Text) = 16 And Val(textmedia.Text) = 20 And Val(textalta.Text) =  
25 And Val(textmalta.Text) = 31 Then  
    btnrestaura.Enabled = False  
End If
```

```
'Public kfeligera, kfemedia, kfealta, kfemalta  
kfemedia = textmedia.Text  
End Sub
```

CORRECCIONES POR ALTURA (PRESIÓN), TEMPERATURA Y HUMEDAD

(frmcorrecciones.frm)

```
Private Sub btnverfrmaisal_Click()
frmaisal!textaltura.Text = textaltura.Text
frmaisal!texttempe.Text = texttempe.Text
frmaisal!texthumedad.Text = texthumedad.Text
```

```
Unload Me
frmaisal.Show
End Sub
```

```
Private Sub Form_Activate()
Rem Lee datos de frmaisal
textaltura.Text = frmaisal!textaltura.Text
texttempe.Text = frmaisal!texttempe.Text
texthumedad.Text = frmaisal!texthumedad.Text
End Sub
```

```
Private Sub textaltura_Change()
altura = Val(textaltura.Text)
tempe = Val(texttempe.Text)
hume = Val(texthumedad.Text)
```

```
k3 = calhumedad(hume)
presión = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
k1 = presión / 76 'b/bo
k2 = (273 + 25) / (273 + tempe)
```

```
textpresion.Text = k1
textcoeftemp.Text = k2
textcoefhume.Text = k3
```

```
espin = Val((Mid(frmaisal!cboespin.Text, 1, 7))) 'Determina el F.
espinterométrico
fstorre = Val(frmaisal!cbofstorre.Text) 'Fact./Seg. torre
```

```
BIL = Val(frmaisal!cboBIL.Text) * 1000 'Determina el BIL
NBS = Val(frmaisal!cboNBS.Text) * 1000 'Determina el NBS
```

```
If frmaisal!optBIL.Value = True Then
GoTo Line1 'Diseño por Rayo
Else
GoTo Line2 'Diseño por maniobra
```

End If

Line1: ' ***** Inicio: diseño por Rayo *****

m = 1

w = 1

textprem.Text = m

texttempn.Text = w

texthumew.Text = w

GoTo Line3 ' ***** fin de Diseño por rayo *****

Line2: ' ***** Inicio: diseño por Maniobra *****

'Establece la primera aproximacion de D sin corrección

porcent = Val((frmaisla!cboNBSpor.Text)) / 100 'Determina "sigma"

vcf = (NBS * fstorre) / (1 - 1.3 * (porcent)) 'vcf= V(50%) fórmula

disminleroysc = 8 / ((3400 * espin / (vcf / 1000)) - 1)

'Determinación de los coeficientes m y w-n

m = calm(NBS / 1000)

w = calwn(disminleroysc)

textprem.Text = m

texttempn.Text = w

texthumew.Text = w

Line3:

End Sub

Private Sub texthumedad_Change()

altura = Val(textaltura.Text)

tempe = Val(texttempe.Text)

hume = Val(texthumedad.Text)

k3 = calhumedad(hume)

presión = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b

k1 = presión / 76 'b/bo

k2 = (273 + 25) / (273 + tempe)

textpresion.Text = k1

textcoeftemp.Text = k2

textcoefhume.Text = k3

espin = Val((Mid(frmaisla!cboespin.Text, 1, 7))) 'Determina el F.
espinterométrico

fstorre = Val(frmaisla!cbofstorre.Text) 'Fact./Seg. torre

BIL = Val(frmaisla!cboBIL.Text) * 1000 'Determina el BIL

NBS = Val(frmaisla!cboNBS.Text) * 1000 'Determina el NBS

```

If frmaisal!optBIL.Value = True Then
  GoTo Line1 'Diseño por Rayo
Else
  GoTo Line2 'Diseño por maniobra
End If

```

```

Line1:      ' ***** Inicio: diseño por Rayo *****
  m = 1
  w = 1
  textprem.Text = m
  texttempn.Text = w
  texthumew.Text = w
  GoTo Line3 ' ***** fin de Diseño por rayo *****

```

```

Line2:      Rem ***** Inicio: diseño por Maniobra *****

```

```

'Establece la primera aproximacion de D sin corrección
porcent = Val((frmaisal!cboNBSpor.Text)) / 100 'Determina "sigma"
vcf = (NBS * fstorre) / (1 - 1.3 * (porcent)) 'vcf= V(50%) fórmula
disminleroysc = 8 / ((3400 * espin / (vcf / 1000)) - 1)

```

```

'Determinación de m y w-n
m = calm(NBS / 1000)
w = calwn(disminleroysc)
textprem.Text = m
texttempn.Text = w
texthumew.Text = w

```

```

Line3:
  End Sub

```

```

Private Sub texttempe_Change()
  altura = Val(textaltura.Text)
  tempe = Val(texttempe.Text)
  hume = Val(texthumedad.Text)

```

```

k3 = calhumedad(hume)
presión = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
k1 = presión / 76 'b/bo
k2 = (273 + 25) / (273 + tempe)

```

```

textpresion.Text = k1
textcoeftemp.Text = k2
textcoefhume.Text = k3

```



```

    espin = Val((Mid(frmaisla!cboespin.Text, 1, 7))) 'Determina el F.
    espinterométrico

```

```

    fstorre = Val(frmaisla!cbofstorre.Text)      'Fact./Seg. torre

```

```

    BIL = Val(frmaisla!cboBIL.Text) * 1000      'Determina el BIL

```

```

    NBS = Val(frmaisla!cboNBS.Text) * 1000     'Determina el NBS

```

```

    If frmaisla!optBIL.Value = True Then

```

```

        GoTo Line1 'Diseño por Rayo

```

```

    Else

```

```

        GoTo Line2 'Diseño por maniobra

```

```

    End If

```

```

Line1:          ' ***** Inicio: diseño por Rayo *****

```

```

    m = 1

```

```

    w = 1

```

```

    textprem.Text = m

```

```

    texttempn.Text = w

```

```

    texthumew.Text = w

```

```

    GoTo Line3 ' ***** fin de Diseño por rayo *****

```

```

Line2:          Rem ***** Inicio: diseño por Maniobra *****

```

```

    'Establece la primera aproximacion de D sin corrección

```

```

    percent = Val((frmaisla!cboNBSpor.Text)) / 100 'Determina "sigma"

```

```

    vcf = (NBS * fstorre) / (1 - 1.3 * (percent)) 'vcf= V(50%) fórmula

```

```

    disminleroysc = 8 / ((3400 * espin / (vcf / 1000)) - 1)

```

```

    'Determinación de m y w-n

```

```

    m = calm(NBS / 1000)

```

```

    w = calwn(diminleroysc)

```

```

    textprem.Text = m

```

```

    texttempn.Text = w

```

```

    texthumew.Text = w

```

```

Line3:

```

```

    End Sub

```

CORRECCIONES POR ALTURA Y TEMPERATURA: EFECTO CORONA

(frmcorrecorona.frm)

```
Private Sub btnfrmcorona_Click()
```

```
  If rastreo = "corona" Then
    frmcorona!textaltura.Text = textaltura.Text
    frmcorona!texttempe.Text = texttempe.Text
  End If
```

```
  If rastreo = "ruido" Then
    frmruído.textaltura = textaltura.Text
    frmruído.texttempe = texttempe.Text
  End If
```

```
Unload Me
End Sub
```

```
Private Sub Form_Activate()
  Rem Lee datos de frmcorona
  If rastreo = "corona" Then
    textaltura.Text = frmcorona!textaltura.Text
    texttempe.Text = frmcorona!texttempe.Text
  End If
```

```
  Rem lee datos de frmruído
  If rastreo = "ruido" Then
    textaltura.Text = frmruído!textaltura.Text
    texttempe.Text = frmruído!texttempe.Text
  End If
```

```
  Rem ***** Correcciones por altura y temperatura *****
```

```
  altura = Val(textaltura.Text) 'Datos
  tempe = Val(texttempe.Text) 'Datos
```

```
  presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b
  fcc = 3.921 * presion / (273 + tempe) 'factor de corrección
```

```
  textpresion.Text = presion
  textdelta.Text = fcc
End Sub
```

```
Private Sub textaltura_Change()  
    altura = Val(textaltura.Text) 'Datos  
    tempe = Val(texttempe.Text) 'Datos  
  
    presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b  
    fcc = 3.921 * presion / (273 + tempe) 'factor de corrección  
  
    textpresion.Text = presion  
    textdelta.Text = fcc  
End Sub
```

```
Private Sub texttempe_Change()  
    altura = Val(textaltura.Text) 'Datos  
    tempe = Val(texttempe.Text) 'Datos  
  
    presion = 76 * 10 ^ (-altura / 18336) 'presión cmHg=b  
    fcc = 3.921 * presion / (273 + tempe) 'factor de corrección  
  
    textpresion.Text = presion  
    textdelta.Text = fcc  
End Sub
```

INFORMACIÓN SOBRE PARÁMETROS

(frmdatosabcd.frm)

```
Private Sub btnirctes_Click()  
Hide  
ctes.Show  
End Sub
```

```
Private Sub Form_Activate()  
btnirctes.SetFocus  
End Sub
```

ACTUALIZACIÓN DE NÚMEROS DE ANTIVIBRADORES POR VANO

(frminfnumamorti.frm)

```
Private Sub btncerrar_Click()  
'nustoc450 variables universales  
'nustoc600  
'nustoc950  
nustoc450 = Val(textvano450.Text)  
nustoc600 = Val(textvano600.Text)  
nustoc950 = Val(textvano900.Text)  
Unload Me  
End Sub
```

```
Private Sub Form_Load()  
If nustoc450 = 0 Then nustoc450 = 2  
If nustoc600 = 0 Then nustoc600 = 4  
If nustoc950 = 0 Then nustoc950 = 6
```

```
textvano450.Text = nustoc450  
textvano600.Text = nustoc600  
textvano900.Text = nustoc950  
End Sub
```

ITERACIONES REQUERIDAS

(frmitera.frm)

```

Private Sub btnverblindaje_Click()
itera = Val(textitera.Text) 'lee el valor
dif = Val(texterr.Text) 'lee el valor
Unload Me

If rastreo2 = "vanodesnivel" Then
    frmvanosdes.Show
Else
    frmblindaje.Show
End If
End Sub

Private Sub Form_Load()
If itera = 0 And dif = 0 Then
    GoTo fin
End If
texterr.Text = dif
textitera.Text = itera
fin:
End Sub

Private Sub spinerror_SpinDown()
Dim dblrate
dblrate = texterr.Text
varerror = Val(cboporerror.Text)
texterr.Text = dblrate - (varerror)
texterr.Refresh
End Sub

Private Sub spinerror_SpinUp()
varerror = Val(cboporerror.Text)
texterr.Text = (texterr.Text) + (varerror)
texterr.Refresh
End Sub

Private Sub spinitera_SpinDown()
Dim dblrango
dblrango = textitera.Text
entero = Val(Mid(cboporitera.Text, 2))
textitera.Text = dblrango - entero
textitera.Refresh
End Sub

```

```

Private Sub spinitera_SpinUp()
entero = Val(Mid(cboporitera.Text, 2))
textitera.Text = (textitera.Text) + entero
textitera.Refresh
End Sub

```

INFORMACIÓN DE LA LONGITUD DEL CABLE Y VANO CRÍTICO

(frmlonvanocri.frm)

```

Private Sub btncal_Click()
Rem Lectura de datos
areac = Val(frmtensiones.textarea.Text)           'Area del conductor (mm²)
pesoc = Val(frmtensiones.textpeso.Text) / 1000    'Peso del cable (Kgf/m)
dilata = Val(frmtensiones.textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación
1/°C

```

```

Rem ***** Cálculo de longitud del cable *****

```

```

vano = Val(textvano.Text) 'Vano (m)

```

```

Rem Flecha normal

```

```

tdn = Val(texttn.Text)

```

```

If tdn = 0 Then GoTo final

```

```

lchfn = longcah(vano, pesoc, fsn, tdn)

```

```

textln.Text = lchfn

```

```

lcpfn = longcap(vano, pesoc, fsn, tdn)

```

```

tipslnp = " Longitud (por la parábola): " & Format(lcpfn, "0.000000") & " m "

```

```

textln.ToolTipText = tipslnp

```

final:

```

Rem Flecha mínima

```

```

tdfmin = Val(texttmin.Text)

```

```

If tdfmin = 0 Then GoTo final1

```

```

lchfmin = longcah(vano, pesoc, fsmin, tdfmin)

```

```

textlmin.Text = lchfmin

```

```

lcpfmin = longcap(vano, pesoc, fsmin, tdfmin)

```

```

tipslminp = " Longitud (por la parábola): " & Format(lcpfmin, "0.000000") & " m "

```

```

textlmin.ToolTipText = tipslminp

```

final1:

```

Rem Flecha máxima

```

```

tdfmax = Val(texttmax.Text)

```

```

If tdfmax = 0 Then GoTo final2
lchfmax = longcah(vano, pesoc, fsmax, tdfmax)
textlmax.Text = lchfmax
lcpfmax = longcap(vano, pesoc, fsmax, tdfmax)
tipslmaxp = " Longitud (por la parábola): " & Format(lcpfmax, "0.000000") & " m "
textlmax.ToolTipText = tipslmaxp

Rem ***** Cálculo de vano crítico *****
tempn = Val(texttempn.Text)      'temperatura inicial
tempmin = Val(texttempmin.Text)  'temperatura final flecha mínima
tempmax = Val(texttempmax.Text)  'temperatura final flecha máxima
denomi = (Abs(fsmax ^ 2 - fsmin ^ 2))

If denomi = 0 Then
    textvciami.Text = vano
    GoTo final2
End If

vanocri1 = (tdn / pesoc) * Sqr(24 * dilata * Abs(tempmax - tempmin) / denomi)
textvciami.Text = vanocri1
final2:
End Sub

Private Sub btninf_Click()
    denomi = (Abs(fsmax ^ 2 - fsmin ^ 2))
    If denomi = 0 Then
        ver = "Parámetros de cálculo no definidos (DELTA)"
        msjnocal = "El vano crítico depende de varios factores, entre los cuales
tenemos el factor de sobrecarga, el cual se considera para los casos de" & Chr(10)
& Chr(13)
        msjnocal = msjnocal & "flecha máxima y flecha mínima. En este caso las
sobrecargas son iguales, por lo tanto no se puede hacer la estimación del cálculo."
& Chr(10) & Chr(13)
        msjnocal = msjnocal & Chr(10) & Chr(13)
        msjnocal = msjnocal & "          vano normal = vano crítico = " &
Val(textvciami.Text) & Chr(10) & Chr(13)
        msjnocal = msjnocal & " " & Chr(10) & Chr(13)
        msjnocal = msjnocal & "El vano crítico se calcula para dos condiciones de
sobrecarga diferentes: flecha máxima y flecha mínima." & Chr(10) & Chr(13)
        msjnocal = msjnocal & "Si el vano es superior al vano crítico, entonces en la
condición de flecha máxima se presenta la máxima tensión." & Chr(10) & Chr(13)
        msjnocal = msjnocal & "Si el vano es inferior al vano crítico, entonces en la
condición de flecha mínima se presenta la máxima tensión." & Chr(10) & Chr(13)
        msjnocal = msjnocal & "Si el vano es igual al vano crítico, entonces en las
dos condiciones (flecha mínima y flecha máxima) se presenta la máxima tensión."
& Chr(10) & Chr(13)
    Else

```

```

ver = "Información: Vano Crítico"
msjnocal = "El vano crítico se calcula para dos condiciones de sobrecarga
diferentes: flecha máxima y flecha mínima." & Chr(10) & Chr(13)
msjnocal = msjnocal & "Si el vano es superior al vano crítico ( > " &
Format(Val(textvcmami.Text), "0.0000") & " ) , entonces en la condición de flecha
máxima se presenta la máxima tensión." & Chr(10) & Chr(13)
msjnocal = msjnocal & "Si el vano es inferior al vano crítico ( < " &
Format(Val(textvcmami.Text), "0.0000") & " ) , entonces en la condición de flecha
mínima se presenta la máxima tensión." & Chr(10) & Chr(13)
msjnocal = msjnocal & "Si el vano es igual al vano crítico ( = " &
Format(Val(textvcmami.Text), "0.0000") & " ) , entonces en las dos condiciones
(flecha mínima y flecha máxima) se presenta la máxima tensión." & Chr(10) &
Chr(13)
End If
outvacri = MsgBox(msjnocal, vbExclamation, ver)
End Sub

Private Sub btnverfrm tension_Click()
Unload Me
frm tensiones.Show
End Sub

Private Sub Form_Load()
Rem Lectura de datos
areac = Val(frm tensiones.textarea.Text)           'Area del conductor (mm²)
pesoc = Val(frm tensiones.textpeso.Text) / 1000   'Peso del cable (Kgf/m)
textvano.Text = Val(frm tensiones.textvano.Text)  'Vano (m)
texttempn.Text = Val(frm tensiones.texttempn.Text) 'temperatura inicial
texttempmin.Text = Val(frm tensiones.texttempmin.Text) 'temperatura final
flecha mínima
texttempmax.Text = Val(frm tensiones.texttempmax.Text) 'temperatura final
flecha máxima
dilata = Val(frm tensiones.textdilata.Text) * 10 ^ -6 'Coeficiente de dilatación
1/°C

'tffpmmn Tensión de diseño (Kgf/mm²)
'tffpmm Tensión final de diseño flecha mínima (Kgf/mm²)
'tffpmmmax Tensión final de diseño flecha máxima (Kgf/mm²)
texttn.Text = Format(tffpmmn * areac, "0.0000") 'Tensión de diseño (Kgf)
texttmin.Text = Format(tffpmm * areac, "0.0000") 'Tensión final de diseño
flecha mínima (Kgf)
texttmax.Text = Format(tffpmmmax * areac, "0.0000") 'Tensión final de diseño
flecha máxima (Kgf)

' fsn factor de sobrecarga
' fsmin factor de sobrecarga flecha mínima
' fsmax factor de sobrecarga flecha máxima

```



```

Rem ***** Cálculo de longitud del cable *****
vano = Val(textvano.Text) 'Vano (m)

Rem Flecha normal
tdn = Val(texttn.Text)
If tdn = 0 Then GoTo final
lchfn = longcah(vano, pesoc, fsn, tdn)
textln.Text = lchfn
lcpfn = longcap(vano, pesoc, fsn, tdn)
tipslnp = " Longitud (por la parábola): " & Format(lcpfn, "0.000000") & " m "
textln.ToolTipText = tipslnp

final:
Rem Flecha mínima
tdfmin = Val(texttmin.Text)
If tdfmin = 0 Then GoTo final1
lchfmin = longcah(vano, pesoc, fsmin, tdfmin)
textlmin.Text = lchfmin
lcpfmin = longcap(vano, pesoc, fsmin, tdfmin)
tipslminp = " Longitud (por la parábola): " & Format(lcpfmin, "0.000000") & " m "
textlmin.ToolTipText = tipslminp

final1:
Rem Flecha máxima
tdfmax = Val(texttmax.Text)
If tdfmax = 0 Then GoTo final2
lchfmax = longcah(vano, pesoc, fsmax, tdfmax)
textlmax.Text = lchfmax
lcpfmax = longcap(vano, pesoc, fsmax, tdfmax)
tipslmaxp = " Longitud (por la parábola): " & Format(lcpfmax, "0.000000") & " m "
textlmax.ToolTipText = tipslmaxp

Rem ***** Cálculo de vano crítico *****
tempn = Val(texttempn.Text) 'temperatura inicial
tempmin = Val(texttempmin.Text) 'temperatura final flecha mínima
tempmax = Val(texttempmax.Text) 'temperatura final flecha máxima
denomi = (Abs(fsmax ^ 2 - fsmin ^ 2))
If denomi = 0 Then
    textvcmami.Text = vano
    GoTo final2
End If

vanocri1 = (tdn / pesoc) * Sqr(24 * dilata * Abs(tempmax - tempmin) / denomi)
textvcmami.Text = vanocri1
final2:
End Sub

```

```
Private Sub texttempmax_Click()  
texttempmax.SelStart = 0  
texttempmax.SelLength = Len(texttempmax)  
End Sub
```

```
Private Sub texttempmin_Click()  
texttempmin.SelStart = 0  
texttempmin.SelLength = Len(texttempmin)  
End Sub
```

```
Private Sub texttempn_Click()  
texttempn.SelStart = 0  
texttempn.SelLength = Len(texttempn)  
End Sub
```

```
Private Sub texttmax_Click()  
texttmax.SelStart = 0  
texttmax.SelLength = Len(texttmax)  
End Sub
```

```
Private Sub texttmin_Click()  
texttmin.SelStart = 0  
texttmin.SelLength = Len(texttmin)  
End Sub
```

```
Private Sub texttn_Click()  
texttn.SelStart = 0  
texttn.SelLength = Len(texttn)  
End Sub
```

```
Private Sub textvano_Click()  
textvano.SelStart = 0  
textvano.SelLength = Len(textvano)  
End Sub
```

AJUSTE DE LA VELOCIDAD DEL VIENTO

(frmpacalpre.frm)

```
Private Sub btnvfrmension_Click()
```

```
ff = Val(cboff.Text) 'lee el valor
```

```
q = Val(cbopesovol.Text) 'lee el valor
```

```
frmensiones.textvelocidadn.Text = textvelocidadn.Text
```

```
frmensiones.textvelocidadmin.Text = textvelocidadmin.Text
```

```
frmensiones.textvelocidadmax.Text = textvelocidadmax.Text
```

```
Unload Me
```

```
frmensiones.Show
```

```
End Sub
```

```
Private Sub cboff_Click()
```

```
ff = Val(Mid(cboff.Text, 1, 5))
```

```
q = Val(cbopesovol.Text)
```

```
Rem Condición normal
```

```
velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s
```

```
presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)
```

```
textpresionn.Text = presionn
```

```
Rem Condición fría
```

```
velocidadmin = (Val(textvelocidadmin.Text)) * 1000 / 3600 ' pasa de km/hr a m/s
```

```
presionmin = 0.55 * velocidadmin ^ 2 * ff * q / (2 * 9.81)
```

```
textpresionmin.Text = presionmin
```

```
Rem Condición Caliente
```

velocidadmax = (Val(textvelocidadmax.Text)) * 1000 / 3600 ' pasa de km/hr a
m/s

presionmax = $0.55 * velocidadmax^2 * ff * q / (2 * 9.81)$

textpresionmax.Text = presionmax

End Sub

Private Sub cboff_Change()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = $0.55 * velocidadn^2 * ff * q / (2 * 9.81)$

textpresionn.Text = presionn

Rem Condición fría

velocidadmin = (Val(textvelocidadmin.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionmin = $0.55 * velocidadmin^2 * ff * q / (2 * 9.81)$

textpresionmin.Text = presionmin

Rem Condición Caliente

velocidadmax = (Val(textvelocidadmax.Text)) * 1000 / 3600 ' pasa de km/hr a
m/s

presionmax = $0.55 * velocidadmax^2 * ff * q / (2 * 9.81)$

textpresionmax.Text = presionmax

End Sub

Private Sub cbopesovol_Click()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)

textpresionn.Text = presionn

Rem Condición fría

velocidadmin = (Val(textvelocidadmin.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionmin = 0.55 * velocidadmin ^ 2 * ff * q / (2 * 9.81)

textpresionmin.Text = presionmin

Rem Condición Caliente

velocidadmax = (Val(textvelocidadmax.Text)) * 1000 / 3600 ' pasa de km/hr a
m/s

presionmax = 0.55 * velocidadmax ^ 2 * ff * q / (2 * 9.81)

textpresionmax.Text = presionmax

End Sub

Private Sub cbopesovol_Change()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)

textpresionn.Text = presionn

Rem Condición fría

velocidadmin = (Val(textvelocidadmin.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionmin = 0.55 * velocidadmin ^ 2 * ff * q / (2 * 9.81)

textpresionmin.Text = presionmin

```

Rem Condición Caliente
velocidadmax = (Val(textvelocidadmax.Text)) * 1000 / 3600 ' pasa de km/hr a
m/s
presionmax = 0.55 * velocidadmax ^ 2 * ff * q / (2 * 9.81)
textpresionmax.Text = presionmax

```

```

End Sub

```

```

Private Sub Form_Load()
If ff = 0 Then
    GoTo fin
End If
fin:
If q = 0 Then
    GoTo abajo
End If

```

```

cboff.Text = ff
cbopesovol.Text = q
abajo:

```

```

textvelocidadn.Text = frm tensiones!textvelocidadn.Text
textvelocidadadmin.Text = frm tensiones!textvelocidadadmin.Text
textvelocidadmax.Text = frm tensiones!textvelocidadmax.Text

```

```

End Sub

```

```

Private Sub textpresionmax_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)

```

```

presionmax = Val(textpresionmax.Text)
velocidadmax = (3600 / 1000) * Sqr(presionmax * 2 * 9.81 / (ff * q * 0.55))
textvelocidadmax.Text = velocidadmax
End Sub

```

```

Private Sub textpresionmin_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
presionmin = Val(textpresionmin.Text)
velocidadmin = (3600 / 1000) * Sqr(presionmin * 2 * 9.81 / (ff * q * 0.55))
textvelocidadmin.Text = velocidadmin
End Sub

```

```

Private Sub textpresionn_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
presionn = Val(textpresionn.Text)
velocidadn = (3600 / 1000) * Sqr(presionn * 2 * 9.81 / (ff * q * 0.55))
textvelocidadn.Text = velocidadn
End Sub

```

```

Private Sub textvelocidadmax_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
velocidadmax = Val(textvelocidadmax.Text) * 1000 / 3600
presionmax = (ff * q * 0.55) * velocidadmax ^ 2 / (2 * 9.81)
textpresionmax.Text = presionmax
End Sub

```

```

Private Sub textvelocidadmax_Change()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
velocidadmax = Val(textvelocidadmax.Text) * 1000 / 3600
presionmax = (ff * q * 0.55) * velocidadmax ^ 2 / (2 * 9.81)
textpresionmax.Text = presionmax
End Sub

```

```

Private Sub textvelocidadadmin_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
velocidadadmin = Val(textvelocidadadmin.Text) * 1000 / 3600
presionmin = (ff * q * 0.55) * velocidadadmin ^ 2 / (2 * 9.81)
textpresionmin.Text = presionmin
End Sub

```

```

Private Sub textvelocidadadmin_Change()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
velocidadadmin = Val(textvelocidadadmin.Text) * 1000 / 3600
presionmin = (ff * q * 0.55) * velocidadadmin ^ 2 / (2 * 9.81)
textpresionmin.Text = presionmin
End Sub

```

```

Private Sub textvelocidadn_Click()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))

```



```

q = Val(cbopesovol.Text)
velocidadn = Val(textvelocidadn.Text) * 1000 / 3600
presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn
End Sub

```

```

Private Sub textvelocidadn_Change()
Rem ***** Cálculo de velocidad *****
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)
velocidadn = Val(textvelocidadn.Text) * 1000 / 3600
presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)
textpresionn.Text = presionn
End Sub

```

AJUSTE DE LA PRESIÓN DEL VIENTO

(frmprevdes.frm)

```

Private Sub btnvfrmension_Click()
ff = Val(cboff.Text) 'lee el valor
q = Val(cbopesovol.Text) 'lee el valor
frmvanosdes.textvelocidadn.Text = textvelocidadn.Text
Unload Me
frmvanosdes.Show
End Sub

```

```

Private Sub cboff_Click()
ff = Val(Mid(cboff.Text, 1, 5))
q = Val(cbopesovol.Text)

```

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)

textpresionn.Text = presionn

End Sub

Private Sub cboff_Change()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)

textpresionn.Text = presionn

End Sub

Private Sub cbopesovol_Click()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s

presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)

textpresionn.Text = presionn

End Sub

Private Sub cbopesovol_Change()

ff = Val(Mid(cboff.Text, 1, 5))

q = Val(cbopesovol.Text)

Rem Condición normal

```
velocidadn = (Val(textvelocidadn.Text)) * 1000 / 3600 ' pasa de km/hr a m/s
```

```
presionn = 0.55 * velocidadn ^ 2 * ff * q / (2 * 9.81)
```

```
textpresionn.Text = presionn
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    If ff = 0 Then
```

```
        GoTo fin
```

```
    End If
```

```
fin:
```

```
    If q = 0 Then
```

```
        GoTo abajo
```

```
    End If
```

```
    cboff.Text = ff
```

```
    cbopesovol.Text = q
```

```
abajo:
```

```
    textvelocidadn.Text = frmvanosdes!textvelocidadn.Text
```

```
End Sub
```

```
Private Sub textpresionn_Click()
```

```
    Rem ***** Cálculo de velocidad *****
```

```
    ff = Val(Mid(cboff.Text, 1, 5))
```

```
    q = Val(cbopesovol.Text)
```

```
    presionn = Val(textpresionn.Text)
```

```
    velocidadn = (3600 / 1000) * Sqr(presionn * 2 * 9.81 / (ff * q * 0.55))
```

```
    textvelocidadn.Text = velocidadn
```

```
End Sub
```

```
Private Sub textvelocidadn_Click()  
    Rem ***** Cálculo de velocidad *****  
    ff = Val(Mid(cboff.Text, 1, 5))  
    q = Val(cbopesovol.Text)  
    velocidadn = Val(textvelocidadn.Text) * 1000 / 3600  
    presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)  
    textpresionn.Text = presionn
```

```
End Sub
```

```
Private Sub textvelocidadn_Change()  
    Rem ***** Cálculo de velocidad *****  
    ff = Val(Mid(cboff.Text, 1, 5))  
    q = Val(cbopesovol.Text)  
    velocidadn = Val(textvelocidadn.Text) * 1000 / 3600  
    presionn = (ff * q * 0.55) * velocidadn ^ 2 / (2 * 9.81)  
    textpresionn.Text = presionn
```

```
End Sub
```

VISUALIZACIÓN DE DISTANCIAS DOS CIRCUITOS

(frmverdis.frm)

```
Private Sub btnsalir_Click()  
Hide  
End Sub
```

```
Private Sub Form_Activate()  
textverdab.Text = dab  
textverdbc.Text = dbc  
textverdac.Text = dac  
textverdab2.Text = dab2  
textverdba2.Text = dba2  
textverdaa2.Text = daa2  
End Sub
```

```
Private Sub Form_Load()  
textverdab.Text = dab  
textverdbc.Text = dbc  
textverdac.Text = dac  
textverdab2.Text = dab2  
textverdba2.Text = dba2  
textverdaa2.Text = daa2  
End Sub
```

VISUALIZACIÓN DE VANOS A DESNIVEL

(frmvervanodes.frm)

Dim verfig As Variant

Private Sub btnrestaurar_Click()

cotaxinf = Val(frmvanosdes.textxinf.Text)

Rem Condiciones de figura a presentar

Rem *****

If cotaxinf < 0 Then

 pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas.wmf")

 verfig = 1 'Asigna variable para control de figura en el btnvfigs

 For K = 0 To 12

 Label1(K).Visible = True

 Next K

 For K = 13 To 25

 Label1(K).Visible = False

 Next K

 For K = 26 To 38

 Label1(K).Visible = False

 Next K

 decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & ") < 0 : "
& Chr(10) & Chr(13)

 decide = decide & "El vértice de la catenaria esta situado a la derecha del apoyo inferior." & Chr(10) & Chr(13)

```

decide = decide & "(Catenaria con vértice real entre los apoyos del vano)"
lbltema.Caption = decide
GoTo final

End If

Rem *****
If cotaxinf = 0 Then
    pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas1.wmf")
    verfig = 2 'Asigna variable para control de figura en el btnvfigs
    For K = 0 To 12
        Label1(K).Visible = False
    Next K

    For K = 13 To 25
        Label1(K).Visible = True
    Next K

    For K = 26 To 38
        Label1(K).Visible = False
    Next K

    decide = "Abscisa X del apoyo inferior = 0 : " & Chr(10) & Chr(13)
    decide = decide & "El vértice de la catenaria coincide con el apoyo inferior." &
Chr(10) & Chr(13)
    decide = decide & "(Catenaria con vértice real en el apoyo inferior del vano)"
    lbltema.Caption = decide
    GoTo final

End If

Rem *****

```

```
If cotaxinf > 0 Then
    pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas2.wmf")
    verfig = 3 'Asigna variable para control de figura en el btnvfigs
    For K = 0 To 12
        Label1(K).Visible = False
    Next K

    For K = 13 To 25
        Label1(K).Visible = False
    Next K

    For K = 26 To 38
        Label1(K).Visible = True
    Next K

    decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & " ) > 0 : "
    & Chr(10) & Chr(13)
    decide = decide & "El vértice de la catenaria esta situado a la izquierda del
apoyo inferior." & Chr(10) & Chr(13)
    decide = decide & "(Catenaria con vértice virtual fuera del vano)"
    lbltema.Caption = decide
    GoTo final
End If

final:
If verfig = 3 Then
    verfig = 0
End If
btnrestaurar.Enabled = False
End Sub
```



```
Private Sub btnsalir_Click()
```

```
Unload Me
```

```
frmvanosdes.Show
```

```
End Sub
```

```
Private Sub btnvfigs_Click()
```

```
btnrestaurar.Enabled = True
```

```
If verfig = 0 Then
```

```
    verfig = verfig + 1
```

```
    GoTo Line1
```

```
End If
```

```
verfig = verfig + 1
```

```
Line1:
```

```
If verfig = 1 Then
```

```
    pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas.wmf")
```

```
    For K = 0 To 12
```

```
        Label1(K).Visible = True
```

```
    Next K
```

```
    For K = 13 To 25
```

```
        Label1(K).Visible = False
```

```
    Next K
```

```
    For K = 26 To 38
```

```
        Label1(K).Visible = False
```

```
    Next K
```

```
    decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & " ) < 0 : "  
& Chr(10) & Chr(13)
```

```
decide = decide & "El vértice de la catenaria esta situado a la derecha del apoyo inferior." & Chr(10) & Chr(13)
```

```
decide = decide & "(Catenaria con vértice real entre los apoyos del vano)"
```

```
lbltema.Caption = decide
```

```
End If
```

```
If verfig = 2 Then
```

```
pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas1.wmf")
```

```
For K = 0 To 12
```

```
Label1(K).Visible = False
```

```
Next K
```

```
For K = 13 To 25
```

```
Label1(K).Visible = True
```

```
Next K
```

```
For K = 26 To 38
```

```
Label1(K).Visible = False
```

```
Next K
```

```
decide = "Abscisa X del apoyo inferior = 0 : " & Chr(10) & Chr(13)
```

```
decide = decide & "El vértice de la catenaria coincide con el apoyo inferior." & Chr(10) & Chr(13)
```

```
decide = decide & "(Catenaria con vértice real en el apoyo inferior del vano)"
```

```
lbltema.Caption = decide
```

```
End If
```

```
If verfig = 3 Then
```

```
pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas2.wmf")
```

```
For K = 0 To 12
```

```
Label1(K).Visible = False
```

```
Next K
```

```

For K = 13 To 25
    Label1(K).Visible = False
Next K

For K = 26 To 38
    Label1(K).Visible = True
Next K
decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & " ) > 0 : "
& Chr(10) & Chr(13)
decide = decide & "El vértice de la catenaria esta situado a la izquierda del
apoyo inferior." & Chr(10) & Chr(13)
decide = decide & "(Catenaria con vértice virtual fuera del vano)"
lbltema.Caption = decide
End If

If verfig >= 3 Then
    verfig = 0
End If
End Sub
Private Sub Form_Load()
cotaxinf = Val(frmvanosdes.textxinf.Text)

Rem Condiciones de figura a presentar

Rem *****
If cotaxinf < 0 Then
    pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas.wmf")
    verfig = 1 'Asigna variable para control de figura en el btnvfigs
    For K = 0 To 12
        Label1(K).Visible = True
    
```

Next K

For K = 13 To 25

Label1(K).Visible = False

Next K

For K = 26 To 38

Label1(K).Visible = False

Next K

decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & ") < 0 : "
& Chr(10) & Chr(13)

decide = decide & "El vértice de la catenaria esta situado a la derecha del apoyo inferior." & Chr(10) & Chr(13)

decide = decide & "(Catenaria con vértice real entre los apoyos del vano)"

lbltema.Caption = decide

GoTo final

End If

Rem *****

If cotaxinf = 0 Then

pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas1.wmf")

verfig = 2 'Asigna variable para control de figura en el btnvfigs

For K = 0 To 12

Label1(K).Visible = False

Next K

For K = 13 To 25

Label1(K).Visible = True

Next K

```

For K = 26 To 38
    Label1(K).Visible = False
Next K

decide = "Abscisa X del apoyo inferior = 0 : " & Chr(10) & Chr(13)
decide = decide & "El vértice de la catenaria coincide con el apoyo inferior." &
Chr(10) & Chr(13)
decide = decide & "(Catenaria con vértice real en el apoyo inferior del vano)"
lbltema.Caption = decide
GoTo final
End If

Rem *****
If cotaxinf > 0 Then
    pic1.Picture = LoadPicture("C:\Delta\Graficas\flechas2.wmf")
    verfig = 3 'Asigna variable para control de figura en el btnvfigs
    For K = 0 To 12
        Label1(K).Visible = False
    Next K

    For K = 13 To 25
        Label1(K).Visible = False
    Next K

    For K = 26 To 38
        Label1(K).Visible = True
    Next K

    decide = "Abscisa X del apoyo inferior (" & Format(cotaxinf, "0.000") & " ) > 0 : "
& Chr(10) & Chr(13)

```

```
decide = decide & "El vértice de la catenaria esta situado a la izquierda del  
apoyo inferior." & Chr(10) & Chr(13)
```

```
decide = decide & "(Catenaria con vértice virtual fuera del vano)"
```

```
lblema.Caption = decide
```

```
GoTo final
```

```
End If
```

```
final:
```

```
If verfig = 3 Then
```

```
verfig = 0
```

```
End If
```

```
End Sub
```

FORMA PARA BARRA DE MENU: FORMA MDI (MDIfrm.frm)

```
Private Sub Aislamiento_Click()  
frmaisla.Show  
End Sub
```

```
Private Sub amortiguadores_Click()  
frmnunamorti.Show  
End Sub
```

```
Private Sub Angulo_Click()  
frmangulo.Show  
End Sub
```

```
Private Sub basedatos_Click()  
frmbasedatos.Show  
End Sub
```

```
Private Sub blindaje_Click()  
frmblindaje.Show  
End Sub
```

```
Private Sub Capacitancia_Click()  
Capac.Show  
End Sub
```

```
Private Sub Constantes_Click()  
ctes.Show  
End Sub
```

```
Private Sub corona_Click()  
frmcorona.Show  
End Sub
```

```
Private Sub desnivel_Click()  
frmvanosdes.Show  
End Sub
```

```
Private Sub figura_Click()  
frmvistazo.Show  
End Sub
```

```
Private Sub flechas_Click()
```

```
frmtensiones.Show  
End Sub
```

```
Private Sub general_Click()  
frmCables.Show  
End Sub
```

```
Private Sub Inductancia_Click()  
Induct.Show  
End Sub
```

```
Private Sub MDIForm_Load()
```

```
End Sub
```

```
Private Sub Momento_Click()  
frmcalcable.Show  
End Sub
```

```
Private Sub parametros_Click()  
Param.Show  
End Sub
```

```
Private Sub principal_Click()  
Principio.Show  
End Sub
```

```
Private Sub Resistencia_Click()  
resi.Show  
End Sub
```

```
Private Sub ruido_Click()  
frmruido.Show  
End Sub
```

```
Private Sub tabla_Click()  
frmgridcables.Show  
End Sub
```

```
Private Sub tutorial_Click()  
Tutor.Show  
End Sub
```


TUTORIAL

(Tutorial.frm)

```
Private Sub cmdadd_Click()  
datPrimaryRS.Recordset.AddNew  
End Sub
```

```
Private Sub cmddelete_Click()  
With datPrimaryRS.Recordset  
    .Delete  
    .MoveNext  
    If .EOF Then .MoveLast  
End With  
End Sub
```

```
Private Sub cmdRefresh_Click()  
'This is only needed for multi user apps  
datPrimaryRS.Refresh  
End Sub
```

```
Private Sub cmdUpdate_Click()  
datPrimaryRS.UpdateRecord  
datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified  
End Sub
```

```
Private Sub cmdClose_Click()  
Screen.MousePointer = vbDefault  
Unload Me  
Tutor.Show  
End Sub
```

```
Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)  
'This is where you would put error handling code  
'If you want to ignore errors, comment out the next line  
'If you want to trap them, add code here to handle them  
MsgBox "Data error event hit err:" & Error$(DataErr)  
Response = 0 'Throw away the error  
End Sub
```

```
Private Sub datPrimaryRS_Reposition()  
Screen.MousePointer = vbDefault  
On Error Resume Next  
'This will display the current record position for dynasets and snapshots  
datPrimaryRS.Caption = "Record: " &  
(datPrimaryRS.Recordset.AbsolutePosition + 1)
```

```
End Sub
```

```
Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
```

```
'This is where you put validation code
```

```
'This event gets called when the following actions occur
```

```
Select Case Action
```

```
    Case vbDataActionMoveFirst
```

```
    Case vbDataActionMovePrevious
```

```
    Case vbDataActionMoveNext
```

```
    Case vbDataActionMoveLast
```

```
    Case vbDataActionAddNew
```

```
    Case vbDataActionUpdate
```

```
    Case vbDataActionDelete
```

```
    Case vbDataActionFind
```

```
    Case vbDataActionBookmark
```

```
    Case vbDataActionClose
```

```
        Screen.MousePointer = vbDefault
```

```
End Select
```

```
Screen.MousePointer = vbHourglass
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Screen.MousePointer = vbDefault
```

```
End Sub
```

```
Private Sub oleFields_DbClick(Index As Integer)
```

```
'This is the way to get data into an empty ole control
```

```
' and have it saved back to the table
```

```
oleFields(Index).InsertObjDlg
```

```
End Sub
```

MENU DEL TUTORIAL (Tutor.frm)

```
Private Sub Form_Load()  
MDIfrm.conductor.Enabled = False  
MDIfrm.Parámetros.Enabled = False  
MDIfrm.DISEÑO.Enabled = False  
MDIfrm.tensiones.Enabled = False  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
MDIfrm.conductor.Enabled = True  
MDIfrm.Parámetros.Enabled = True  
MDIfrm.DISEÑO.Enabled = True  
MDIfrm.tensiones.Enabled = True  
End Sub
```

```
Private Sub Tutor1_Click()  
Unload Me  
Tutorial.Show  
End Sub
```

```
Private Sub Tutor4_Click()  
Unload Me  
Intro.Show  
End Sub
```

DECLARACION DE VARIABLES

(Variables.bas)

```
Public resi As Single
Public temp, coeftempe, resitemp, diame
Public coefresis, rugosi
Public rac
Public evita
```

```
Public dab As Single, dbc As Single, dac As Single
Public dab2 As Single, dba2 As Single
Public daa2 As Single, dca2 As Single
Public dac2 As Single, dbb2 As Single
Public hab As Single, hbc As Single
Public ha As Single, hb As Single, hc As Single
Public flecha As Single, hm As Single
```

```
Public dis As Single, ncfa As Single
Public dmg As Single
Public rmgg As Single, requi As Single, L As Single
Public XL As Single, rmgdckto As Single
Public c As Single, XC As Single, Yc1 As Single
```

```
Public adya As Single, adyb As Single
Public km As Single, Yc As Single
```

```
Public alfa3 As Single
```

```
Public Function ang(adyb, hip) As Double
B = adyb / hip
ang = Atn(B / Sqr(-B * B + 1))
End Function
```

```
Public Function hip(adya, adyb) As Single
hip = (adya ^ 2 + adyb ^ 2) ^ 0.5
End Function
```

VARIABLES DE DISEÑO

(Codediseño.bas)

```

Public pesot As Double, largo As Double
Public tipestruct As Double
Public dibujoload As Double
Public rastreo As String 'establece dirección de lectura de datos en
                          'frmcorona y frmruído para abrir frmcorrecorona

Public contaminacion As String
Public itera As Double, dif As Double
Public kfeligera, kfemedia, kfealta, kfemalta

Public rastreo2 As String 'establece dirección de lectura de datos en
                          'frmblindaje y frmvanodes para abrir frmitera

Public rastreo3 As String, rastreo4 As String, rastreo5 As String
Public rastreo6 As String, rastreo7 As String

Public ff As Double, q As Double

Public slns(8)          'Memoria de las soluciones de la cúbica

Public tffpmmn         'Tensión de diseño (Kgf/mm2)
Public tffpmm          'Tensión final de diseño flecha mínima (Kgf/mm2)
Public tffpmmmax       'Tensión final de diseño flecha máxima (Kgf/mm2)

Public fsn             'factor de sobrecarga
Public fsmin           'factor de sobrecarga flecha mínima
Public fsmax           'factor de sobrecarga flecha máxima

Public nustoc450 As Double
Public nustoc600 As Double
Public nustoc950 As Double

Type exportar
  Titulo As String
  Nominal As Single
  Caliente As Single
  Fria As Single
  Peso As Single
End Type

Public Function senh(numero) As Double

```

```

Rem Hyperbolic Sine HSin(X) = (Exp(X) - Exp(-X)) / 2
senh = (Exp(numero) - Exp(-numero)) / 2
End Function

```

```

Public Function cosh(numero) As Double
Rem Hyperbolic Cosine HCos(X) = (Exp(X) + Exp(-X)) / 2
cosh = (Exp(numero) + Exp(-numero)) / 2
End Function

```

```

Public Function calhumedad(humedadabs) As Double
calhumedad = 1 - 0.00818 * (humedadabs - 11)
End Function

```

```

Public Function calwn(distancia) As Double
Select Case distancia
Case Is < 1
calwn = 1
Case 1 To 6
calwn = -0.12 * distancia + 1.12
Case Is > 6
calwn = 0.4
End Select
End Function

```

```

Public Function longcah(vano, Peso, loadm, tension)
Rem LONGITUD DEL CABLE POR LA CATENARIA
Rem Hyperbolic Sine HSin(X) = (Exp(X) - Exp(-X)) / 2
ter1 = tension / (Peso * loadm)
ter2 = vano / (2 * ter1)
longcah = 2 * ter1 * senh(ter2)
End Function

```

```

Public Function longcap(vano, Peso, loadm, tension)
Rem LONGITUD DEL CABLE POR LA PARABOLA
ter3 = tension / (Peso * loadm)
longcap = vano + ((vano) ^ 3 / (24 * (ter3) ^ 2))
End Function

```

```

Public Function loninf(vano, desnivel, hv) As Double
Rem CALCULO DE LA DISTANCIA HORIZONTAL DEL VERTICE AL APOYO
INFERIOR
Rem EN VANOS A DESNIVEL
pa1 = vano / hv
pa2 = desnivel / hv
Rem Hyperbolic Cosine HCos(X) = (Exp(X) + Exp(-X)) / 2
nivel1 = ((Exp(pa1) + Exp(-pa1)) / 2) - 1

```

```
Rem Hyperbolic Sine   HSin(X) = (Exp(X) - Exp(-X)) / 2
nivel2 = (Exp(pa1) - Exp(-pa1)) / 2
```

```
Rem Inverse Hyperbolic
```

```
'Tangent           HArctan(X) = Log((1 + X) / (1 - X)) / 2
pa3 = nivel1 / nivel2
nivel3 = Log((1 + pa3) / (1 - pa3)) / 2
nivel4 = pa2 / Sqr((nivel2) ^ 2 - (nivel1) ^ 2)
```

```
Rem Inverse Hyperbolic
```

```
'Sine             HArctsin(X) = Log(X + Sqr(X * X + 1))
loninf = hv * (Log(nivel4 + Sqr(nivel4 * nivel4 + 1)) - nivel3)
End Function
```

```
Public Function calm(nbskv) As Double
```

```
calm = (0.00000004) * nbskv ^ 2 - 0.0005 * nbskv + 1.1302
End Function
```

```
Public Function fhip(ctehip, vanohip, fs) As Double
```

```
'HCos(X) = (Exp(X) + Exp(-X)) / 2
ctem = ctehip / fs
interna = vanohip / (2 * ctem)
fhip = ctem * (((Exp(interna) + Exp(-interna)) / 2) - 1)
End Function
```

```
Public Function fpara(vanopara, pesocpara, tensionpara, fs) As Double
```

```
fpara = vanopara ^ 2 * pesocpara * fs / (8 * tensionpara)
End Function
```

```
Public Function cubica(Coef0, Coef1, Coef2, Coef3) As Double
```

```
slns(0) = 0: slns(1) = 0: slns(2) = 0: slns(3) = 0
slns(4) = 0: slns(5) = 0: slns(6) = 0: slns(7) = 0
slns(8) = 0
```

```
5 'HOME
```

```
10 'Print "RAICES DE POLINOMIOS (METODO DE BAIRSTOWS)"
```

```
30 Dim coef(25) As Double, B(25) As Double, E(25) As Double
```

```
40 E(1) = 1 * 10 ^ -4
```

```
50 E(4) = 1 * 10 ^ -20
```

```
60 k1 = 100000 'número de iteraciones
```

```
70 'Print
```

```
80 'Print "GRADO DEL POLINOMIO";
```

```
90 'INPUT N
```

```
    N = 3 'Grado del polinomio
```

```
100 'PRINT
```

```
120 'PRINT "ENTRE LOS COEFICIENTES:"
```

```

130 'FOR I=0 TO N
140 'PRINT "A(";I;")";
150 'Rem LOS COEFICIENTES ESTAN STORED IN ARRAY A() EN ORDEN
INVERSO
160 'INPUT A(N-I+1)
170 'NEXT I

```

Rem entrada de datos según el grado de la ecuación

coef(4) = Coef0

coef(3) = Coef1

coef(2) = Coef2

coef(1) = Coef3

```

180 'Rem PRUEBA PARA DATOS INVALIDOS
190 'If Val(coef(1)) <> 0 Then GoTo 270
200 'Rem PROMPT FOR AN INVALID ENTRY
210 'PRINT "A(N) MUST BE NON-ZERO. RE-ENTER."
220 'PRINT "A(";N;")";
230 'INPUT A(1)

```

coef(1) = InputBox("A(N) MUST BE NON-ZERO. RE-ENTER. " & coef(N),
"DATO INVALIDO")

```

240 'Rem PRUEBA DEL NUEVO DATO
250 'GoTo 190
260 'Rem
270 'PRINT
280 'PRINT "RAICES:"
290 'REM BRANCH FOR SPECIAL TRATAMIENTO PARA EC. DE 1R Y 2O
GRADO
300 'If N <= 2 Then GoTo 1080
310 'coef(N + 2) = 0
320 'N1 = 2 * Int((N + 1) / 2)
330 'For M1 = 1 To N1 / 2
340 '    p = 1
350 '    qu = 1
360 '    For K = 1 To k1
370 '    For L = 1 To k1
380 '        'Rem STORE ALL COEFICIENTES IN ARRAY B
390 '        For i = 1 To N1 + 1
400 '            B(i) = coef(i)
410 '        Next i
420 '        For J = N1 - 2 To N1 - 4 Step -2
430 '        For i = 1 To J + 1
440 '            B(i + 1) = B(i + 1) - p * B(i)
450 '            B(i + 2) = B(i + 2) - qu * B(i)
460 '        Next i

```



```

470     Next J
480     R0 = B(N1 + 1)
490     R1 = B(N1)
500     S0 = B(N1 - 1)
510     S1 = B(N1 - 2)
520     V0 = -qu * S1
530     V1 = S0 - S1 * p
540     D0 = V1 * S0 - V0 * S1
550     If Abs(D0) >= E(4) Then GoTo 590
560     p = p + 5
570     qu = qu + 5
580     Next L
590     D1 = S0 * R1 - S1 * R0
600     D2 = R0 * V1 - V0 * R1
610     P1 = D1 / D0
620     qu1 = D2 / D0
630     p = p + P1
640     qu = qu + qu1
650     If Abs(R0) >= E(1) Then GoTo 690
660     If Abs(R1) >= E(1) Then GoTo 690
670     E(M1) = 1
680     GoTo 810
690     If Abs(P1) >= E(1) Then GoTo 730
700     If Abs(qu1) >= E(1) Then GoTo 730
710     E(M1) = 2
720     GoTo 810
730     If p = 0 Then GoTo 750
740     If Abs(P1 / p) >= E(1) Then GoTo 790
750     If qu = 0 Then GoTo 790
760     If Abs(qu1 / qu) >= E(1) Then GoTo 790
770     E(M1) = 3
780     GoTo 810
790     Next K
800     E(M1) = 4
810     r = -p / 2
820     T = (r) ^ 2 - qu
830     If T < 0 Then GoTo 890
840     T = Sqr(T)
850     'PRINT
860     'PRINT S+T
Rem aumento de linea captura de datos solo para el caso de la cúbica **
'MsgBox "1: " & Format(r + T, "0.0000")
If slns(0) = 0 Then
    slns(0) = r + T
    GoTo 870
Else
    GoTo para1

```

```

      End If
para1:
  If slns(2) = 0 Then
    slns(2) = r + T
  End If

870   'PRINT S-T
      'MsgBox "2: " & Format(r - T, "0.0000")
      If slns(1) = 0 Then
        slns(1) = r - T
        GoTo 880
      Else
        GoTo para2
      End If
para2:
  If slns(3) = 0 Then
    slns(3) = r - T
  End If
Rem fin de aumento de linea para captura de datos **

880   GoTo 930
890   T = Sqr(-T)
900   'PRINT
910   'PRINT S;" + I *";T
      'MsgBox "3: S= " & Format(r, "0.0000") & " + J " & Format(T, "0.0000")
      slns(4) = r
      slns(5) = T

920   'PRINT S;" - I *";T
      'MsgBox "4: S= " & Format(r, "0.0000") & " - J " & Format(T, "0.0000")
      slns(6) = r
      slns(7) = -T

930   If E(M1) = 4 Then GoTo 1180
940   For J = 1 To N1 - 1
950     coef(J + 1) = coef(J + 1) - p * coef(J)
960     coef(J + 2) = coef(J + 2) - qu * coef(J)
970   Next J
980   N1 = N1 - 2
990   If N1 > 1 Then GoTo 1010
1000  GoTo 1180
1010  If N1 >= 3 Then GoTo 1070
1020  M1 = M1 + 1
1030  E(M1) = 1
1040  p = coef(2) / coef(1)
1050  qu = coef(3) / coef(1)
1060  GoTo 810

```

```

1070 Next M1
1080 If N = 2 Then GoTo 1110
1090 'PRINT (-1)* COEF(2)/COEF(1)
      'MsgBox "5: " & Format(-1 * coef(2) / coef(1), "0.0000")
      slns(8) = -1 * coef(2) / coef(1)

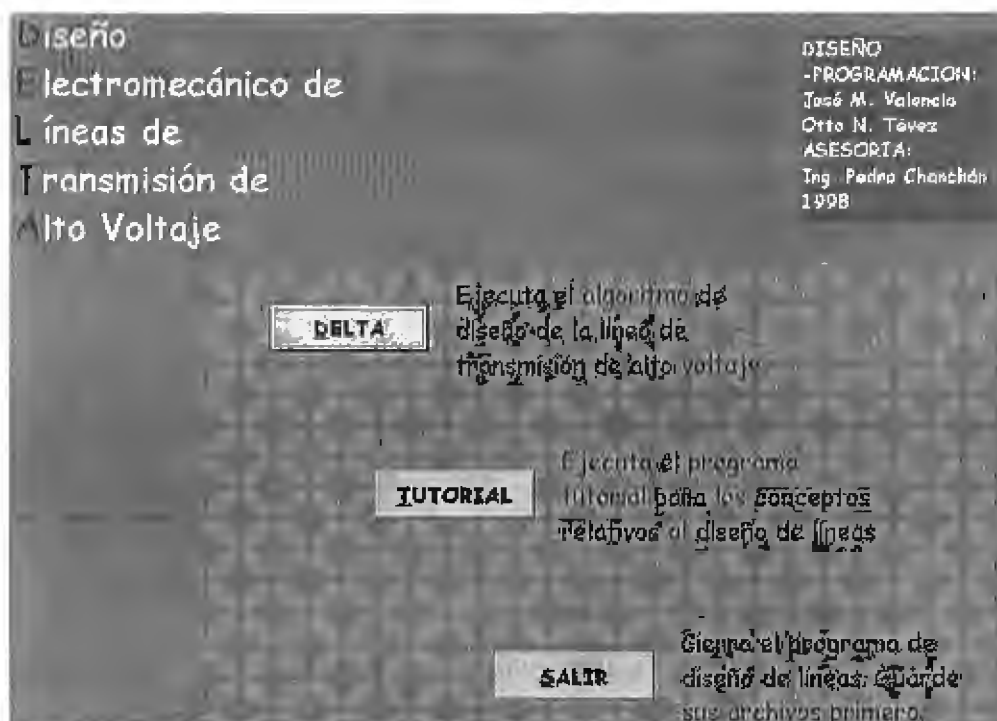
1100 GoTo 1180
1110 coef(3) = (coef(2)) ^ 2 - 4 * coef(1) * coef(3)
1120 r = -coef(2) / 2 / coef(1)
1130 T = Sqr(Abs(coef(3))) / 2 / coef(1)
1140 M1 = 4
1150 E(4) = 4
1160 If Sgn(coef(3)) < 0 Then GoTo 900
1170 GoTo 850
1180 'PRINT
1190 'PRINT "WOULD YOU LIKE TO RE-RUN THIS PROGRAM WITH NEW
DATA (Y/N)";
1200 'INPUT X$
1210 'IF X$ = "Y" THEN 70
1220 'IF X$ <> "N" THEN 1190
1230 'End

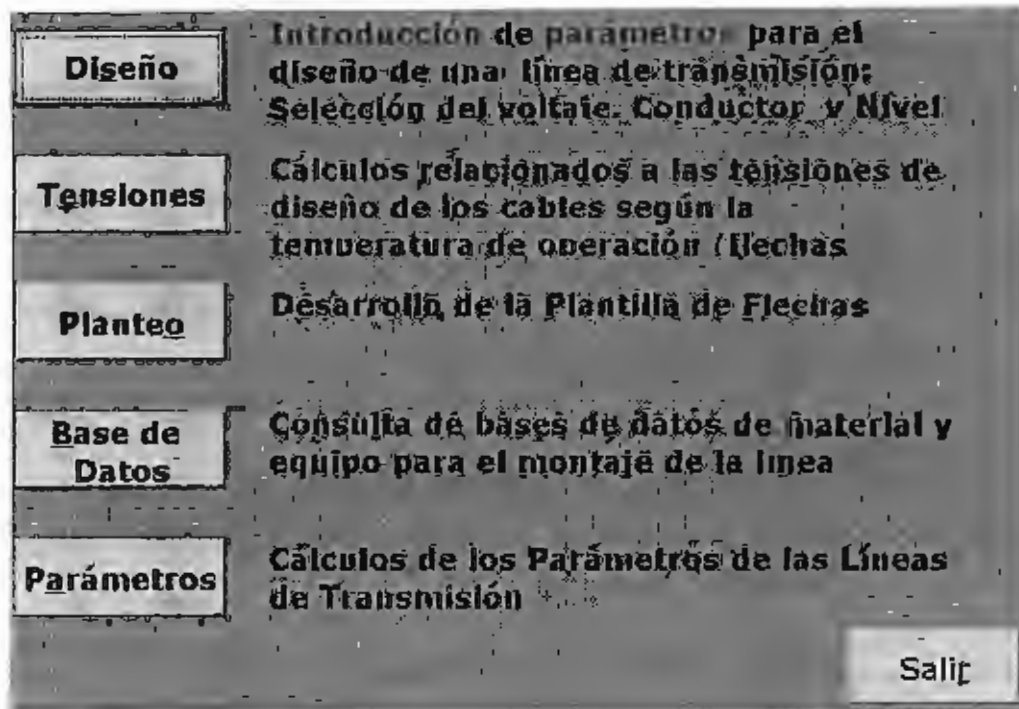
      Rem Para hacer pruebas
      'msj = "slns(0) = " & Format(slns(0), "0.0000") & " slns(1) = " &
Format(slns(1), "0.0000") & Chr(10) & Chr(13)
      'msj = msj & "slns(2) = " & Format(slns(2), "0.0000") & " slns(3) = " &
Format(slns(3), "0.0000") & Chr(10) & Chr(13)
      'msj = msj & "slns(4) = " & Format(slns(4), "0.0000") & " slns(5) = " &
Format(slns(5), "0.0000") & Chr(10) & Chr(13)
      'msj = msj & "slns(6) = " & Format(slns(6), "0.0000") & " slns(7) = " &
Format(slns(7), "0.0000") & Chr(10) & Chr(13)
      'msj = msj & "slns(8) = " & Format(slns(8), "0.0000")
      'MsgBox msj
      cubica = slns(0)
End Function

```

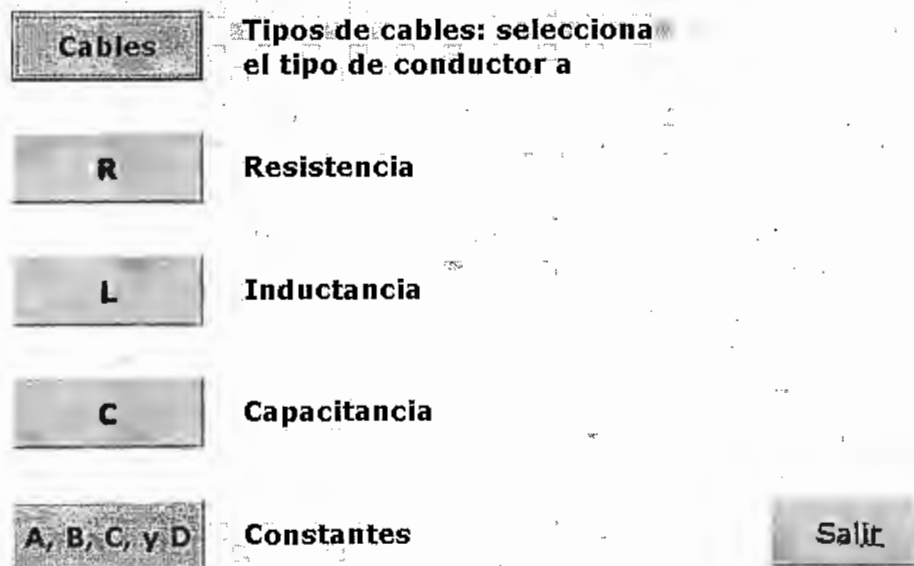
ANEXOS

PANTALLAS DEL PROGRAMA





CALCULO DE PARAMETROS DE LAS LINEAS DE TRANSMISION




Cables	Establace acceso a la base de datos de cables
Aisladores	Establace acceso a la base de datos de aisladores
Herrajes	Establace acceso a la base de datos de herrajes
Estructuras	Establace acceso a la base de datos de estructuras

Salir

Para iniciar el diseño preliminar de la línea de transmisión se necesita los siguientes datos referentes a la misma. En todo caso si no se disponen de todos los datos el programa permite introducirlos en el desarrollo de los pasos lógicos del diseño a realizarse.

Potencia y pérdidas					Longitudes		Ancho de vía
Pactual (MW)	F.P.	P (%)	V (%)	F.S.	L (KM)	Vano regla (m)	Ancho (m)
40	0.85	4	6	1.5	50	300	30 (115 KV)

Voltaje	Tipo de cable	Número de circuitos	Cables por fase
KV		<input checked="" type="radio"/> 1 Un circuito <input type="radio"/> 2 Dos circuitos	Número de cables
115	ACSR		 1

Correcciones por Altura, Temperatura y Humedad			Nivel de contaminación
Altura (M.S.N.M.)	Temperatura (°C)	Humedad absoluta (Grs/m³)	
0	25	11	Ligera Actualizar

Selección de **V**

Salir

Datos						Orden de compra		
Proceso (MW)	TF	L (MW)	CP	PK1	ES	Atm	Tasa	F (Unidad MW)
40	0.85	50	4	6	15	15		3543227
Número de circuitos		Cables por fase		Volaje superior		Comente		
Cables por fase		Humedad cable		V (V)		actual (A) (Módulo A)		
110		33		170		3543227 249.3073		
Cable Segundo								
Indice (Módulo A)	MDM	Temperatura (MW)	EMG (MW)	Peso (Kg/Km)	ES	ES	ES	ES
110	TURKEY	5	6/1	5.037	1.20091	537	2.684	Vari. cable
Valores establecidos						BLO NBS		
AC (Unidad / Km)	DC (Unidad / Km)	ES	V (V)					
2.584	0.025423	55	179643	45.448785				
						Datos		
						Salir		

Criterio de diseño		Selección del BLO del NBS			Flacha (m)		
Valor	Medida del tope	BLO	ES				
15	123	450					
Distancia crítica		Conexiones por Altura, Temperatura y Humedad					
Selección de Estabilidad	ES (Estructura)	Altura (MST/M)	Temperatura (C)	Humedad (MST/M)	Uso		
1.20	Condición venturo	1.15			Dirección		
Asaladores y trabajos			Factores mecánicos				
Selección de Asalador	Altura (m)	Angulo (m)	Fuerza (N/m)	Peso (N)	Factor de seguridad (coeficiente)		
					0.60 $\theta < 40^\circ$		
Resultados		Distancias críticas		Distancias		Selección de V	
Distancia crítica	Distancia crítica	Distancia crítica	Distancia crítica	Distancia crítica	Distancia crítica	Distancia crítica	Distancia crítica
Distancias críticas		Distancias críticas		Distancias críticas		Salir	

Cable					F. del viento: Aisladores y	
Tip. (Nomina)	MRM (AVG)	Peso (Kg/Km)	Diámetro (mm)	Número de conductores	Horquillas	Distancia vertical fase (m)
FAWKE	477	477.7	21.793			1.09538
Presión de viento: P					Peso de los Aisladores y Horquillas (Kg)	
Factor de diseño	Peso volumétrico del D (Kg / m ³)	Velocidad de viento (Km/h)	W (Kg / m ²)			Diámetro (mm)
1.45 Cables	1.225	80	13.891 37247		31.32	3.84825
Fuerza del viento: Cables		Tipo de Apoyo		Tipo de Anillo		
Vano (m)	F (Kg)	Tangente	De Angulo	Con Viento	Tan. Non glb	
300	90 42813	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Angulo de Desviación de la cadena de aisladores						
Cálculo del ángulo de desviación		Tangente	De Angulo	Fuerza	Fuerza	Fuerza
Angulo		Desviación (grados)	Fuerza (Kg)	Fuerza (Kg)	Fuerza (Kg)	Fuerza (Kg)

Angulo de blindaje (Grados)	Angulo de desviación (grados)	Flexa (m)	Hmin (m)	Distancia vertical real (m)	Longitud de la cadena (m)	Cálculo de la altura del cable de guarda
30	16.6416	6	6.06667	1.09538	0.8763	HG
Altura del cable de guarda (m)		Altura de la fase (m)		Distancia conductor - estructura (m)		
Respecto al piso	Respecto al brazo (b)	Sin ángulo	Con ángulo	Sin ajustar	Ajustado por ángulo de desviación	
14.716949189	1.5548991890	12.06667	12.112550	1.21644	1.530139403	
Angulo crítico de blindaje						
BIL (KV)	Zc (Ohms)	Opciones				
450	500					
Io (KA)	isc (m)	Angulo Desp				
1.8	10.577051					
a (m)	b (m)	θsc	Figura final			
-0.3422210	1.5921140	-12.412481610	Angulo			
			Salir			

	Cable Nombre: NMA 2 AWG # de cables: 477 # de cables: 115			
	# de subidas: 0 # de conductores: 1 Conductura local: 0 87530 Altura: 146.05 Área: 254			
	Distancia entre cables (m): Distancia: 1.09538 # de cables: 1 5301 39403			
Altura del cable de guarda (m): Distancia al cable: 1 554889183 Distancia al cable: 1 471584818 Altura mínima: 3 40867				
Altura del cable (m): 12.08667	Distancia de fase a fase (m): 3.060278805	Distancia de fase a neutro (m): 3 060278804	Distancia de fase a fase (m): 8 1205578128	Estado: Cables
Fecha de vigencia (m): 3 060278805				Estado: Cables
				Estado: Cables

Datos del sistema Voltaje: 115 Voltaje: 123 # de fases: 40			Condiciones Coeficiente de potencia: 0.83 Información: 0 Temperatura: 25		
Condición por fase # de subidas: 0			Coefficiente de medida de fase P Tiempo Seco: 1 P Tiempo Humedo: 0.8		Número de circuitos # de circuitos: 0
Conductor Tipo: NMA 2 AWG # de cables: 477 Distancia: 1.09538					
Propiedades Resistencia por cable: 3.85871		Tiempo seco: 193.63207 Resistencia: 0		Resistencia húmeda: 154.90666 Resistencia: 0	
Estado: Cables		Estado: Cables		Estado: Cables	
Estado: Cables		Estado: Cables		Estado: Cables	
Estado: Cables		Estado: Cables		Estado: Cables	

Conexiones

Velocidad	024	Velocidad	024	Velocidad	024	Velocidad	024
Información	75	Conexión		Tiempo Seco	00	Tiempo Húmedo	

Conductor

Tipo	MARK	AVG	477	Resultados	21.793
Marca	HAWK	AVG	477	Resultados	21.793

Conductores por fase

Fase	AVG	Resultados
1	53947	51.12584
2	45748	70.928047

Resultados

Velocidad	1.5	Resultado	12.0667
Velocidad	1.5	Resultado	12.0667

Tiempo "2y" desde se establece el nivel de ruido

Velocidad	1.5	Resultado	1.8
-----------	-----	-----------	-----

Datos del trabajo

Co	HAWK	AVG	477	Resultado (AVG)	28.77
Placa	977.7	Carro (mm)	21.793	Area (mm²)	281.03
Clase	8845	Velocidad	0	C. de planeación	18.9

Ficha Mínima

Velocidad del viento (m/h)	10	0.3842047	F de carga	18	20.22	Tiempo	15	Temperatura	24.8393	Flujo (m³)	3.731393	CS	5
----------------------------	----	-----------	------------	----	-------	--------	----	-------------	---------	------------	----------	----	---

Ficha Normal

Velocidad del viento (m/h)	10	0.3842047	F de carga	18	20.22	Tiempo	25	Temperatura	27.0328	Flujo (m³)	4.07356	CS	3.274
----------------------------	----	-----------	------------	----	-------	--------	----	-------------	---------	------------	---------	----	-------

Ficha Máxima

Velocidad del viento (m/h)	10	9.6051197	Temp. (°C)	20	Tiempo	17.942	Flujo (m³)	5.388813	CS	4.13
----------------------------	----	-----------	------------	----	--------	--------	------------	----------	----	------

Datos del cable

Tipos	AWG	477	26 / 7
Diámetro (mm)	377.7	Diámetro (mm)	21.793
Área (mm²)	140.45	Área (mm²)	261.03
Área (mm²)		Área (mm²)	18.9

Otros

Corriente: 100
 Distancia: 50

Parámetros de carga

Velocidad (m/s): 5
 Distancia (m): 9.6151197
 Presión: 15

Parámetros de inicio

Factor de corrección: 1.15
 Máx. Err: 10

Dirigir resultados

Clase
 Efectos
 Salir

Información general

Tipos de cables: AWG

Tipo de cable y tensión promedio

Nombre	Área (mm²)	Peso (kg/m)	Tensión (N)
AWG	21.793	377.7	

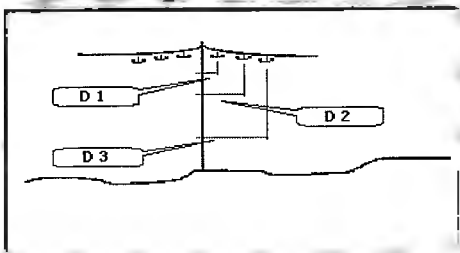
Número de vandas

Menos de 50 (m): 1
 Entre 50 y 900 (m): 1
 Más de 900 (m): 0

Número de amortiguadores

Vandas menos de 50 (m)	Vandas entre 50 y 900 (m)	Vandas mayores de 900 (m)
12	0	18

Distancia (m)	Distancia (m)	Distancia (m)
0.000	8.000	0.000



Continuar