ANEXOS

LISTADO DE LOS PROGRAMAS.-

1.- MEYER – PETER – MULLER

```
'DIMENSIONANDO VARIABLES COMO REAL DOBLE PRECISION (8BYTES)
'DATOS DE ENTRADA Y SALIDA:(V = VELOCIDAD=M/S, D = PROFUNDIDAD
MEDIA = M. W = ANCHO DE LA SECCION = M , S = PENDIENTE HIDRAULICA
=M/M
'D90= D90=MM, D50 =DIAMETRO MEDIO = MM)
"DATOS DE SALIDA.(QS = TN/DIA= DESCARGA DE ARENA)
Dim V As Double, d As Double, w As Double, S As Double, D50 As Double, d90 As
Double
Dim D50M As Double, D90M As Double, A As Double, STEMP As Double, XKS As
Double, XKR As Double, GAMMASPP As Double
Dim GAMMA As Double, temp As Double, GSPP As Double, QS As Double, G As
Double, GR As Double
Private Sub Command1_Click()
G = 2.65: GR = 9.81: GAMMA = 1 'CONSTANTES
'INTRODUCIENDO DATOS, ANALIZANDO VALORES DONDE V,D,W,S,D50,D90
NO DEBEN SER MENORES QUE CERO CASO
'CONTRARIO EL PROGRAMA PEDIRA NUEVO VALOR DONDE ESTE SEA
MENOR QUE CERO
Do
V = InputBox("M/S", "VELOCIDAD MEDIA")
If V < 0 Then
MsgBox "LA VELOCIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While V < 0
Do
d = InputBox("M", "PROFUNDIDAD MEDIA")
If d < 0 Then
MsgBox "LA PROFUNDIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While d < 0
Do
w = InputBox("M", "ANCHO DE LA SECCION")
If w < 0 Then
MsgBox "EL ANCHO DE LA SECCION NO PUEDE SER NEGATIVO", 0 + 16,
"ERROR"
End If
Loop While w < 0
Do
```

```
S = InputBox("M/M", "PENDIENTE HIDRAULICA")
If S < 0 Then
MsgBox "LA PENDIENTE NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While S < 0
Do
D50 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS 50%")
If D50 < 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While D50 < 0
Do
d90 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS 90%")
If d90 < 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While d90 < 0
'COMPARANDO EL VALOR DE D50 CON EL D90 DONDE D50 DEBE SER MENOR
QUE D90,
'CASO CONTRARIOPEDIRA UN NUEVO VALOR DE DE 50 Y DE D90, Y TAMBIEN
DEBEN SER MAYOR DE 0
Do While D50 > d90
MsgBox "EL DIAMETRO MAYOR NO PEDE SER MENOR QUE EL DIAMETRO
PROMEDIO", 0 + 16, "ERROR"
MsgBox "SE NECESITA NUVEO DIAMETRO PROMEDIO Y UN NUEVO
DIAMETRO PARA EL CUAL 90% ES MAS FINO", 0 + 64, "CORRIGIENDO"
D50 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS")
   Do While D50 < 0
   MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
   D50 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS")
   Loop
d90 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS PARA EL CUAL EL 90%
ES MAS FINO")
   Do While d90 < 0
   MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
   d90 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS PARA EL CUAL EL
90% ES MAS FINO")
   Loop
Loop
'MOSTRANDO DATOS DE ENTRADA
Text1.Text = Str(V) + "M/S"
Text2.Text = Str(d) + "M"
Text3.Text = Str(w) + "M"
Text4.Text = Str(S) + "M/M"
Text6.Text = Str(D50) + "MM"
Text8.Text = Str(d90) + "MM"
'OPERANDO
```

```
D50M = 0.001 * D50
D90M = 0.001 * d90
If S = 0 Then
A = 1
STEMP = S
S = 0.001
Else
A = 0
End If
XKS = V / (d ^ 0.66666667 * (S ^ 0.5))
XKR = 26 / D90M ^ 0.1666667
GAMMASPP = GAMMA * (G - 1)
temp = GAMMA * (XKS / XKR) ^ 1.5 * d * S - 0.047 * GAMMASPP * D50M
If temp <= 0 Then
GSPP = 0
Else
GSPP = (temp / (0.25 * (GAMMA / GR) ^ 0.33333)) ^ 1.5
End If
QS = 86400 * w * GSPP * (G / (G - 1))
If A = 1 Then
S = STEMP
End If
Text7.Text = Str(QS) + "TN/DIA"
End Sub
Private Sub Command2_Click()
'BORRANDO TODO
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
End Sub
Private Sub Command3_Click()
'SALIR DE TODO EL PROGRAMA
End
End Sub
Private Sub Command4_Click()
'SALTAR AL FORMULARIO PRINCIPAL
Form1.Show
End Sub
'NO DAR IMPORTANCIA
Private Sub Reloj_Timer()
   Static NúmFigura As Integer
   NúmFigura = NúmFigura + 1
   Load Figura(NúmFigura)
```

```
   Figura(NúmFigura).Top = Rnd * Height
   Figura(NúmFigura).Left = Rnd * Width
   Figura(NúmFigura).Height = Rnd * (Height - Figura(NúmFigura).Top)
   Figura(NúmFigura).Width = Rnd * (Width - Figura(NúmFigura).Left)
   Figura(NúmFigura).Shape = Rnd * 5
   Figura(NúmFigura).FillColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
   Figura(NúmFigura).FillStyle = Rnd * 7
   Figura(NúmFigura).Visible = True
   If NúmFigura > 10 Then Unload Figura(NúmFigura - 10)
End Sub
'NO DAR IMOPRTANCIA
Private Sub Form_MouseDown(Button As Integer, _
                           Shift As Integer, _
                           X As Single, Y As Single)
   'Esta función será llamada de forma automática cuando
   'pulsemos con el ratón sobre el formulario
   If Button = 1 Then
   Else
      'y si pulsamos botón derecho, pintamos rectángulo
      Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End If
End Sub
'NO DAR IMPORTANCIA
Private Sub Form_MouseMove(Button As Integer, _
                           Shift As Integer, _
                           X As Single, Y As Single)
   Select Case Button
   Case 2
         'Si nos movemos mientras pulsamos el botón derecho
         'dejamos un rastro de cuadrados rojos
         Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End Select
End Sub
' NO DAR IMPORTANCIA
Private Sub TEXT7_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyF1 ' Si pulsamos F1
         Text7.BackColor = vbGreen 'ponemos el fondo verde
End Select
End Sub
'O DAR IMPORTANCIA
Private Sub TEXT7_KeyUp(KeyCode As Integer, Shift As Integer)
   'Esta función es invocada automáticamete cada vez que
   'soltemos una tecla
   Text7.BackColor = vbYellow  ' El color que tenía al principio
End Sub
```

2.- ENGELUND - HANSEN

```
'DIMENSIONANDO LA VARIABLES COMO DOBLE PRECISION
'DATOS DE ENTRADA Y SALIDA:( V = VELOCIDAD=M/S,D = PROFUNDIDAD
MEDIA = M, W = ANCHO DE LA SECCION == M ,
'S = PENDIENTE HIDRAULICA = M/M, D50 DIAMETRO MEDIO DE LA
PARTICULA= MM)
' DATOS DE SALIDA ( QS = DESCARGA DE LA ARENA = TN/DIA)
Dim V As Double, d As Double, w As Double, S As Double, D50 As Double, QS As
Double
Dim G As Double, GR As Double, GAMMA As Double, CT As Double, G1 As Double
Dim VSS As Double, FPS As Double, RO As Double, D50M As Double
Private Sub Command1_Click()
G = 2.65: GR = 9.81: GAMMA = 1: CT = 0.05 ' CONSTANTE
' ANALIZANDO QUE LA VARIABLE METIDAS AL PROGRAMA NO SEAN
NEGATIVOS
'CASO CONTRARIO SE PEDIRA UN NUEVO VALOR AL INSTANTE DE LA
VARIABLE INCORRECTA
Do
V = InputBox("M/S", "VELOCIDAD MEDIA")
If V < 0 Then
MsgBox "LA VELOCIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While V < 0
Do
d = InputBox("M", "PROFUNDIDAD MEDIA")
If d < 0 Then
MsgBox "LA PROFUNDIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While d < 0
Do
w = InputBox("M", "ANCHO DE LA SECCION")
If w < 0 Then
MsgBox "EL ANCHO DE LA SECCION NO PUEDE SER NEGATIVO", 0 + 16,
"ERROR"
End If
Loop While w < 0
Do
S = InputBox("M/M", "PENDIENTE HIDRAULICA")
If S < 0 Then
MsgBox "LA PENDIENTE NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While S < 0
Do
D50 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS")
If D50 < 0 Then
```

```
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While D50 < 0
' MOSTRANDO LOS DATOS DE ENTRADA
Text6.Text = Str(D50) + "MM"
Text1.Text = Str(V) + "M/S"
Text2.Text = Str(d) + "M"
Text3.Text = Str(w) + "M"
Text4.Text = Str(S) + "M/M"
'OPERANDO
D50M = 0.001 * D50
G1 = G - 1
VSS = GR * d * S
FPS = V ^ 2 / (GR * G1 * D50M)
RO = G * GAMMA / (GR * G1)
QS = 86400 * w * CT * RO * FPS * VSS ^ 1.5
Text7.Text = Str(QS) + "TN/DIA"
End Sub
Private Sub Command2_Click()
'DEJANDO EN BLANCO TODO
Text6.Text = ""
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text7.Text = ""
End Sub
Private Sub Command3_Click()
'SALIENDO DEL PROGRAMA COMPLETO
End
End Sub
Private Sub Command4_Click()
'SALÑTANDO AL FORMULARIO PRINCIPAL
Form1.Show
End Sub
'NO DAR IMPORTANCIA
Private Sub Form_MouseDown(Button As Integer, _
                    Shift As Integer, _
                    X As Single, Y As Single)
   'Esta función será llamada de forma automática cuando
   'pulsemos con el ratón sobre el formulario
   If Button = 1 Then
   Else
      'y si pulsamos botón derecho, pintamos rectángulo
      Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End If
End Sub
```

```
'NO DAR IMPORTANCIA
Private Sub Form_MouseMove(Button As Integer, _
                    Shift As Integer, _
                    X As Single, Y As Single)
   Select Case Button
   Case 2
        'Si nos movemos mientras pulsamos el botón derecho
        'dejamos un rastro de cuadrados rojos
        Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End Select
End Sub
'NO DAR IMPORTANCIA
Private Sub TEXT7_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyF1 ' Si pulsamos F1
        Text7.BackColor = vbGreen 'ponemos el fondo verde
End Select
End Sub
'NO DAR IMPORTANCIA
Private Sub TEXT7_KeyUp(KeyCode As Integer, Shift As Integer)
   'Esta función es invocada automáticamete cada vez que
   'soltemos una tecla
   Text7.BackColor = vbYellow  ' El color que tenía al principio
End Sub
```

3.- ACKERS-WHITE.-

```
' DIMENSIONANDOLAS VARIABLES, AQUI SE TRABAJARA CON TODOS LOS
DECIMALES,DONDE DOUBLE = REAL DOBLE PRECISION(8 BYTES)

'DATOS DE ENTRADA Y SALIDA:(V = VELOCIDAD=M/S, D = PROFUNDIDAD
MEDIA = M. W = ANCHO DE LA SECCION = M , S = PENDIENTE HIDRAULICA
=M/M
'TEMP = TEMPERATURA = ºC, D50 DIAMETRO MEDIO = MM)

'DATOS DE SALIDA.(QS = TN/DIA= DESCARGA DE ARENA, B() = VECTOR DE
TEMPERATURAS, C() VECTOR DE VISCOCIDADES)

Dim V As Double, d As Double, w As Double, S As Double, temp As Double, D50 As
Double, I As Double
Dim G As Double, GR As Double, SR32 As Double, ALPHA As Double
Dim B() As Double, C() As Double
Dim D50M As Double, VISCOSIDAD As Double, DGR As Double, YN As Double, A As
Double, YM As Double, Z As Double, VS As Double
Dim FGR As Double, TEMPO As Double, GGR As Double, conc As Double, QW As
Double, QS As Double, X As Double, n As Double, ñ As Double
```

```
Private Sub Command1_Click()
'INTRODUCIENDO V,D,S,TEMP,D50,W, DONDE NINGUNO DEBE SER MENOR
QUE CERO, CASO CONTRARIO EL PROGRAMA PEDIRA UN NUEVO VALOR
'AL INSTANTE
X = 0
Do
V = InputBox("M/S", "VELOCIDAD MEDIA")
If V < 0 Then
MsgBox "LA VELOCIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While V < 0
Do
d = InputBox("M", "PROFUNDIDAD MEDIA")
If d < 0 Then
MsgBox "LA PROFUNDIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While d < 0
Do
w = InputBox("M", "ANCHO DE LA SECCION")
If w < 0 Then
MsgBox "EL ANCHO DE LA SECCION NO PUEDE SER NEGATIVO", 0 + 16,
"ERROR"
End If
Loop While w < 0
Do
S = InputBox("M/M", "PENDIENTE HIDRAULICA")
If S < 0 Then
MsgBox "LA PENDIENTE NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While S < 0
Do
temp = InputBox("ºC", "TEMPERATURA DEL AGUA")
If temp < 0 Then
MsgBox "LA TEMPERATURA NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While temp < 0
Do
D50 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS 50%")
If D50 < 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While D50 < 0
'MOSTRANDO LOS VALORES METIDOS AL PROGRAMA
Text1.Text = Str(V) + "M/S"
Text2.Text = Str(d) + "M"
Text3.Text = Str(w) + "M"
Text4.Text = Str(S) + "M/M"
```

```
Text5.Text = Str(temp) + "ºC"
Text6.Text = Str(D50) + "MM"
' MOSTRANDO VECTOR DE TEMPERATURA Y DE VISCOSIDADES
CARGARVECTOR1
CARGARVECTOR2
'OPERANDO
G = 2.65: GR = 9.81: SR32 = 5.656854: ALPHA = 10 ' CONSTANTES
D50M = 0.001 * D50
HALLANDOVISCOCIDAD
DGR = D50M * (GR * (G - 1) / VISCOSIDAD ^ 2) ^ 0.3333333
If DGR <= 1 Then
MsgBox "DEBIDO A QUE DGR <= 1 HAY ERROR", 0 + 16, "ERROR"
X = 8
End If
If DGR > 1 And DGR <= 60 Then
'''''''''''''''''
'''''''''''''''''
'''''''''''''''''
n = Log(DGR)
n = n * 0.434294 """""""convercion a log1o
'''''''''''''''''
'''''''''''''''''
'''''''''''''''''
YN = 1 - 0.56 * n
A = (0.23 / ((DGR) ^ 0.5)) + 0.14
YM = 9.66 / DGR + 1.34
Z = 10 ^ (2.86 * n - (n) ^ 2 - 3.53)
Else
YN = 0
A = 0.17
YM = 1.5
Z = 0.025
End If
VS = (GR * d * S) ^ 0.5
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
ñ = Log(ALPHA * d / D50M)
ñ = ñ * 0.434294 'convercion as log10
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
FGR = VS ^ YN / ((GR * D50M * (G - 1)) ^ 0.5) * (V / (SR32 * ñ)) ^ (1 - YN)
TEMPO = FGR / A
If TEMPO <= 1 Then
GGR = 0
Else
```

```
GGR = Z * (TEMPO - 1) ^ YM
End If
conc = (GGR * G * D50M / d) * (V / VS) ^ YN
QW = V * d * w
QS = 86400 * conc * QW
Text7.Text = Str(QS) + "TM/DIA"
If X = 8 Then
Text7.Text = "ERROR YA QUE DGR <= 1"
End If
End Sub
Public Sub Command2_Click()
' DEJANDO EN BLANCO LOS TEXTOS, Y LOS GRID
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Grid1.Rows = 1
Grid1.Cols = 1
Grid1.Text = ""
Grid2.Rows = 1
Grid2.Cols = 1
Grid2.Text = ""
Command1.SetFocus
End Sub
Private Sub Command3_Click()
'SALIR DEL PROGRAMA POR COMPLETO
End
End Sub
Sub CARGARVECTOR1()
' DIMENSIONANDO EL GRID DONDE SE TENDRAN CUADORS DE IGUAL
ALTURA Y ANCHO
ReDim B(1 To 15)
Grid1.Rows = 1
Grid1.Cols = 15
Grid1.RowHeight(0) = Grid1.Height
For I = 0 To 15 - 1
Grid1.ColWidth(I) = Grid1.Width / 15
Next I
'DEFICIENDO VALORES POR POSICION
B(1) = 0: B(6) = 25: B(11) = 60
B(2) = 5: B(7) = 30: B(12) = 70
B(3) = 10: B(8) = 35: B(13) = 80
B(4) = 15: B(9) = 40: B(14) = 90
B(5) = 20: B(10) = 50: B(15) = 100
```

```
'MOSTRANDO VALORES
For I = 1 To 15
Grid1.Col = I - 1
Grid1.Text = B(I)
Grid1.CellAlignment = 4
Next I
End Sub
Sub CARGARVECTOR2()
' DIMENSIONANDO EL GRID DONDE SE TENDRAN CUADORS DE IGUAL
ALTURA Y ANCHO
ReDim C(1 To 15)
Grid2.Rows = 1
Grid2.Cols = 15
Grid2.RowHeight(0) = Grid2.Height
For I = 0 To 15 - 1
Grid2.ColWidth(I) = Grid2.Width / 15
Next I
'DEFICIENDO VALORES POR POSICION
C(1) = 0.00000179: C(6) = 0.00000089: C(11) = 0.000000474
C(2) = 0.00000152: C(7) = 0.000000801: C(12) = 0.000000413
C(3) = 0.00000131: C(8) = 0.000000723: C(13) = 0.000000365
C(4) = 0.00000114: C(9) = 0.000000658: C(14) = 0.000000326
C(5) = 0.000001: C(10) = 0.000000554: C(15) = 0.000000294
'MOSTRANDO VALORES
For I = 1 To 15
Grid2.Col = I - 1
Grid2.Text = C(I)
Grid2.CellAlignment = 4
Next I
End Sub
Sub HALLANDOVISCOCIDAD()
'INTERPOLANDO LA TEMPERATURA PARA HALLAR LA VISCOSIDAD
CORRESPONDIENTE
If temp <= B(1) Then
VISCOSIDAD = C(2) - (C(2) - C(1)) * ((B(2) - temp) / (B(2) - B(1)))
End If
If temp >= B(15) Then
VISCOSIDAD = C(14) - (C(14) - C(15)) * ((B(14) - temp) / (B(14) - B(15)))
End If
If temp > B(1) And temp < B(15) Then
   For I = 2 To 14
      If temp <= B(I) Then
         VISCOSIDAD = ((C(I) - C(I - 1)) * ((B(I) - temp) / (B(I) - B(I - 1)))) + C(I)
         I = 15
      End If
   Next I
End If
```

```
End Sub
Private Sub Command4_Click()
' SALTO AL FORMULARIO PRIMERO
Form1.Show
End Sub
' NO DAR IMPORTANCIA
Private Sub Reloj_Timer()
   Static NúmFigura As Integer
   NúmFigura = NúmFigura + 1
   Load Figura(NúmFigura)
   Figura(NúmFigura).Top = Rnd * Height
   Figura(NúmFigura).Left = Rnd * Width
   Figura(NúmFigura).Height = Rnd * (Height - Figura(NúmFigura).Top)
   Figura(NúmFigura).Width = Rnd * (Width - Figura(NúmFigura).Left)
   Figura(NúmFigura).Shape = Rnd * 5
   Figura(NúmFigura).FillColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
   Figura(NúmFigura).FillStyle = Rnd * 7
   Figura(NúmFigura).Visible = True
   If NúmFigura > 10 Then Unload Figura(NúmFigura - 10)
End Sub
' NO DAR IMPORTANCIA
Private Sub Form_MouseDown(Button As Integer, _
                           Shift As Integer, _
                           X As Single, Y As Single)
   If Button = 1 Then
   Else
   Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End If
End Sub
'NO DAR IMPORTANCIA
Private Sub Form_MouseMove(Button As Integer, _
                           Shift As Integer, _
                           X As Single, Y As Single)
   Select Case Button
   Case 2

         Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End Select
End Sub
' NO DAR IMPORTANCIA
Private Sub TEXT7_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyF1
         Text7.BackColor = vbGreen
End Select
End Sub
' NO DAR IMPORTANCIA
```

```
Private Sub TEXT7_KeyUp(KeyCode As Integer, Shift As Integer)

   Text7.BackColor = vbYellow
End Sub
```

4.- TOFALETTI

```
'DIMENSIONANDO VARIABLES COMO DOBLE PRECISION (8 BYTES)
' DATOS DE ENTRADA Y SALIDA :(V = VELOCIDAD=M/S, R = RADIO MEDIO
HIDRAULICO=M, W = ANCHO DE LA SECCION = M, S = PENDIENTE DE
EQUILIBRIO
'TEMP = TEMPERATURA =ºC , D65= DIAMETRO PARA EL CUAL EL 65% ES MAS
FINO =MM, FB() FRACCION DE MATERIAL DE LECHO(VECTOR))
' DATOS DE SALIDA:(QS = DESCARGA DE LA ARENA=TN/DIA
' DIAMG()= DIAMETRO MEDIO GEOMETRICO DEL TAMAÑO DE LA PARTICULA
=MM,
'VS() = VELOCIDAD DE ASENTAMIENTO DE LA PARTICULA M/S,GT()=
DESCARGA DE LA ARENA=TN/DIA)
'VS(),GT(),DIAMG() SON: VECTORES
Dim V As Double, R As Double, w As Double, S As Double, temp As Double, d65 As
Double
Dim fb() As Double, DIAMG() As Double, VS() As Double, GF() As Double, GT() As
Double, XTEMP() As Double, YNU() As Double
Dim YYA() As Double, XPAM() As Double, XFAC() As Double, YACORR() As Double
Dim GR As Double, G As Double, J As Double, FAC As Double, ACORR As Double
Dim CMTFT As Double, CMMFT As Double, CTNTM As Double, I As Double
Dim SFMC As Double, CL1 As Double, CL2 As Double, TPRIME As Double, T As
Double, YA As Double, YB As Double, CZ As Double
Dim SI As Double, U2 As Double, L As Double, XNU As Double, U3 As Double, ñ As
Double
Dim CEF As Double, CCC As Double, U3L As Double, U3R As Double, U3X As Double,
U1 As Double
Dim C1 As Double, C2 As Double, UPS As Double, AM As Double, P As Double, PAM
As Double, A As Double
Dim ZOM As Double, ZV As Double, CV As Double, ZV15 As Double, ZOL As Double,
ZOU As Double, F1 As Double, F4 As Double, F5 As Double
Dim F6 As Double, F21 As Double, F32 As Double, X As Double, C As Double, bl As
Double, DD As Double, UD As Double, UBL As Double
Private Sub Command1_Click()
GR = 32.17: G = 2.65: SFMC = 0.7 ' CONTANTES
CMTFT = 0.3048: CMMFT = 304.304: CTNTM = 0.9072 ' CONSTANTES
' ANALIZANDO QUE LA VARIABLE METIDAS AL PROGRAMA NO SEAN
NEGATIVOS
'CASO CONTRARIO SE PEDIRA UN NUEVO VALOR AL INSTANTE DE LA
VARIABLE INCORRECTA
Do
```

```
V = InputBox("M/S", "VELOCIDAD MEDIA")
If V <= 0 Then
MsgBox "LA VELOCIDAD NO PUEDE SER NEGATIVA O` = 0", 0 + 16, "ERROR"
End If
Loop While V <= 0
Do
R = InputBox("M", "RADIO MEDIO HIDRAULICO")
If R <= 0 Then
MsgBox "EL RADIO NO PEDE SER NEGATICO O` = 0", 0 + 16, "ERROR"
End If
Loop While R <= 0
Do
w = InputBox("M", "ANCHO DE LA SECCION")
If w <= 0 Then
MsgBox "EL ANCHO DE LA SECCION NO PUEDE SER NEGATIVO O` =0", 0 + 16,
"ERROR"
End If
Loop While w <= 0
Do
S = InputBox("M/M", "PENDIENTE DE EQUILIBRIO")
If S <= 0 Then
MsgBox "LA PENDIENTE NO PUEDE SER NEGATIVA O =0`", 0 + 16, "ERROR"
End If
Loop While S <= 0
Do
temp = InputBox("ºC", "TEMPERATURA DEL AGUA")
If temp <= 0 Then
MsgBox "LA TEMPERATURA NO PUEDE SER NEGATIVA O` =0", 0 + 16, "ERROR"
End If
Loop While temp <= 0
Do
d65 = InputBox("MM", "DIAMETRO DE LAS PARTICULAS")
If d65 <= 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO O` =0", 0 + 16, "ERROR"
End If
Loop While d65 <= 0
' MOSTRANDO VALORES INTRODUCIDOS AL COMPUTADOR MAS ALGUNA
OTRAS VARIABLES HALLADAS
Text1.Text = Str(V) + "M/S"
Text2.Text = Str(R) + "M"
Text3.Text = Str(w) + "M"
Text4.Text = Str(S) + "M/M"
Text5.Text = Str(temp) + "ºC"
Text6.Text = Str(d65) + "MM"
CARGARVECTOR1
CARGARVECTOR2
CARGARVECTOR3
```

```
CARGARVECTOR4
CARGARVECTOR5
CARGARVECTOR6
CARGARVECTOR7
CARGARVECTOR8
'INTRODUCIENDO VALORES AL VECTOR FB()
PREGUNTARVALORESVECTOR8
'ANALIZANDO QUE LA SUMATORIA DE LOS 8 VALORES DEN =1 CASO
CONTRARIO VOLVER A CARGAR VECTOR
ANALIZARVECTOR8
' SI PASA LA INSPECCION MOSTRAR LOS VALORES DEL VECTOR FB()
MOSTRARVECTOR8
'OPERANDO
V = V / CMTFT
R = R / CMTFT
temp = temp * 1.8 + 32
w = w / CMMFT
d65 = d65 / CMMFT
"""""""
"""""""
CARGARVECTOR7
SFM = SFMC
CL1 = G - 1
CL2 = SFM
TPRIME = GR * (0.0000028 * temp)
T = CL1 * CL2 * TPRIME
YA = R / 11.24
YB = R / 2.5
CZ = 260.67 - 0.667 * temp
SI = S * R * CZ
U2 = V / ((GR * d65 * S) ^ 0.5)
PAWLINMODIFICADO
U3 = V ^ 3 / (GR * XNU * S)
If U2 < 500 Then
CEF = 0.92
Else
CEF = 1
End If
"""""""""""""""""""""""""
"""""""""""""""""""
ñ = Log((U2 / 50) ^ CEF)
ñ = ñ * 0.434294
CCC = 21.9 + 9.7 * ñ
U3L = 2.16 + 0.23 * CC
U3R = 3.32 + 0.23 * CCC
U3X = Log(U3) * 0.434294
If U3X > U3R Then
```

```
U1 = CCC
Else
   If U3X < U3L Then
      U1 = (U3X - 2.17) / 0.23
   Else
      C1 = -((U3R - U3L) / 2) ^ 2
      C2 = (U3R + U3L) / 2
      U1 = (U3X - C2) ^ 2 / C1 + (1 + CCC)
   End If
End If
UPS = V / U1
AM = 10 * UPS
P = 100000 * XNU
PAM = P ^ 0.3333333 / AM
PAWLINLOGMODIFICADO
FAC = PAM * S * d65 * 100000
If FAC <= 0.25 Then
ACORR = 1
Else
PAWLINGMODIFICADO2
A = ACORR * A
End If
If A < 16 Then
A = 16
End If
QS = 0
''''''''''''''''''''''''''''''''''
' PREPARANDO EL GRID PARA EL VECTOR GF(),GT(),VS()
ReDim VS(1 To 8)
Grid9.Rows = 1
Grid9.Cols = 8
Grid9.RowHeight(0) = Grid9.Height
For I = 0 To 8 - 1
Grid9.ColWidth(I) = Grid9.Width / 8
Next I


ReDim GF(1 To 8)
Grid10.Rows = 1
Grid10.Cols = 8
Grid10.RowHeight(0) = Grid10.Height
For I = 0 To 8 - 1
Grid10.ColWidth(I) = Grid10.Width / 8
Next I

ReDim GT(1 To 8)
Grid11.Rows = 1
```

```
Grid11.Cols = 8
Grid11.RowHeight(0) = Grid11.Height
For I = 0 To 8 - 1
Grid11.ColWidth(I) = Grid11.Width / 8
Next I
' CONTINUANDO LA OPERACION
For J = 1 To 8
DD = 2 * DIAMG(J)
VS(J) = (((0.6666667 * GR * DIAMG(J) ^ 3 * CL1 + 36 * XNU ^ 2) ^ 0.5) - 6 * XNU) /
DIAMG(J)
ZOM = VS(J) * V / SI
ZV = 0.1198 + 0.00048 * temp
CV = 1 + ZV
ZV15 = 1.5 * ZV
If ZOM < ZV15 Then
ZOM = ZV15
End If
ZOL = 0.756 * ZOM
ZOU = 1.5 * ZOM
F1 = ZOL - ZV
F4 = 1 - ZOL + ZV
F5 = 1 - ZOM + ZV
F6 = 1 - ZOU + ZV
F21 = ZOM - ZOL
F32 = ZOU - ZOM
If J = 1 Then
GF(J) = 1.905 / ((T * A) / V ^ 2) ^ 1.66666667
Else
GF(J) = GF(J - 1) / 3.175
End If
X = F4 * GF(J) / (YA ^ F4 - DD ^ F4)
C = fb(J) * w * X
bl = C * DD ^ F4
UD = CV * V * (DD / R) ^ ZV
UBL = X / (43.2 * UD * DD ^ F1)
If UBL > 100 Then
bl = (bl / UBL) * 100
End If
GT(J) = GF(J) * fb(J) * w + bl + (C / 75) * YA ^ F21 * (YB ^ F5 - YA ^ F5) + (C / F6) *
YA ^ F21 * YB ^ F32 * (R ^ F6 - YB ^ F6)
DIAMG(J) = CMMFT * DIAMG(J)
VS(J) = VS(J) * CMTFT
GT(J) = CTNTM * GT(J)
QS = QS + GT(J)
Next J
' MOSTRANDO LOS VALORES DEL VECTOR VS(),GF(),GT()
For I = 1 To 8
```

```
Grid9.Col = I - 1
Grid9.Text = VS(I)
Grid9.CellAlignment = 4
Next I

For I = 1 To 8
Grid10.Col = I - 1
Grid10.Text = GF(I)
Grid10.CellAlignment = 4
Next I

For I = 1 To 8
Grid11.Col = I - 1
Grid11.Text = GT(I)
Grid11.CellAlignment = 4
Next I
Text7.Text = Str(QS) + "TM/DIA"
End Sub
Private Sub Command2_Click()
' DEJAR EN BLANCO TODO
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Grid1.Rows = 1
Grid1.Cols = 1
Grid1.Text = ""
Grid2.Rows = 1
Grid2.Cols = 1
Grid2.Text = ""
Grid3.Rows = 1
Grid3.Cols = 1
Grid3.Text = ""
Grid4.Rows = 1
Grid4.Cols = 1
Grid4.Text = ""
Grid5.Rows = 1
Grid5.Cols = 1
Grid5.Text = ""
Grid6.Rows = 1
Grid6.Cols = 1
Grid6.Text = ""
Grid7.Rows = 1
Grid7.Cols = 1
```

```
Grid7.Text = ""
Grid8.Rows = 1
Grid8.Cols = 1
Grid8.Text = ""
Grid9.Rows = 1
Grid9.Cols = 1
Grid9.Text = ""
Grid10.Rows = 1
Grid10.Cols = 1
Grid10.Text = ""
Grid11.Rows = 1
Grid11.Cols = 1
Grid11.Text = ""
Command1.SetFocus
End Sub
Private Sub Command3_Click()
' SALIR DEL PROGRAMA POR COMPLETO
End
End Sub
Private Sub Command4_Click()
' SALTAR AL PRINCIPAL FORMULARIO
Form1.Show
End Sub
Sub CARGARVECTOR1()
ReDim XTEMP(1 To 10)
Grid1.Rows = 1
Grid1.Cols = 10
Grid1.RowHeight(0) = Grid1.Height
For I = 0 To 10 - 1
Grid1.ColWidth(I) = Grid1.Width / 10
Next I
XTEMP(1) = 32: XTEMP(6) = 80
XTEMP(2) = 40: XTEMP(7) = 90
XTEMP(3) = 50: XTEMP(8) = 100
XTEMP(4) = 60: XTEMP(9) = 120
XTEMP(5) = 70: XTEMP(10) = 140
For I = 1 To 10
Grid1.Col = I - 1
Grid1.Text = XTEMP(I)
Grid1.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR2()
ReDim YNU(1 To 10)
Grid2.Rows = 1
Grid2.Cols = 10
```

```
Grid2.RowHeight(0) = Grid2.Height
For I = 0 To 10 - 1
Grid2.ColWidth(I) = Grid2.Width / 10
Next I
YNU(1) = 0.00001931: YNU(6) = 0.0000093
YNU(2) = 0.00001664: YNU(7) = 0.00000826
YNU(3) = 0.0000141: YNU(8) = 0.00000739
YNU(4) = 0.00001217: YNU(9) = 0.00000609
YNU(5) = 0.00001058: YNU(10) = 0.00000514
For I = 1 To 10
Grid2.Col = I - 1
Grid2.Text = YNU(I)
Grid2.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR3()
ReDim YYA(1 To 6)
Grid3.Rows = 1
Grid3.Cols = 6
Grid3.RowHeight(0) = Grid3.Height
For I = 0 To 6 - 1
Grid3.ColWidth(I) = Grid3.Width / 6
Next I
YYA(1) = 100: YYA(6) = 780
YYA(2) = 28
YYA(3) = 33
YYA(4) = 49
YYA(5) = 49
For I = 1 To 6
Grid3.Col = I - 1
Grid3.Text = YYA(I)
Grid3.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR4()
ReDim XPAM(1 To 6)
Grid4.Rows = 1
Grid4.Cols = 6
Grid4.RowHeight(0) = Grid4.Height
For I = 0 To 6 - 1
Grid4.ColWidth(I) = Grid4.Width / 6
Next I
XPAM(1) = 0.216: XPAM(6) = 35
XPAM(2) = 0.5
XPAM(3) = 0.665
```

```
XPAM(4) = 0.72
XPAM(5) = 1.27
For I = 1 To 6
Grid4.Col = I - 1
Grid4.Text = XPAM(I)
Grid4.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR5()
ReDim XFAC(1 To 4)
Grid5.Rows = 1
Grid5.Cols = 4
Grid5.RowHeight(0) = Grid5.Height
For I = 0 To 4 - 1
Grid5.ColWidth(I) = Grid5.Width / 4
Next I
XFAC(1) = 0.2
XFAC(2) = 0.25
XFAC(3) = 0.35
XFAC(4) = 1
For I = 1 To 4
Grid5.Col = I - 1
Grid5.Text = XFAC(I)
Grid5.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR6()
ReDim YACORR(1 To 4)
Grid6.Rows = 1
Grid6.Cols = 4
Grid6.RowHeight(0) = Grid6.Height
For I = 0 To 4 - 1
Grid6.ColWidth(I) = Grid6.Width / 4
Next I
YACORR(1) = 0.2
YACORR(2) = 0.25
YACORR(3) = 0.35
YACORR(4) = 1
For I = 1 To 4
Grid6.Col = I - 1
Grid6.Text = YACORR(I)
Grid6.CellAlignment = 4
Next I
End Sub
Sub CARGARVECTOR7()
```

```
ReDim DIAMG(1 To 8)
Grid7.Rows = 1
Grid7.Cols = 8
Grid7.RowHeight(0) = Grid7.Height
For I = 0 To 8 - 1
Grid7.ColWidth(I) = Grid7.Width / 8
Next I
DIAMG(1) = 0.00029
For J = 2 To 8
DIAMG(J) = 2 * DIAMG(J - 1)
Next J
For I = 1 To 8
Grid7.Col = I - 1
Grid7.Text = DIAMG(I)
Grid7.CellAlignment = 4
Next I
End Sub
Sub CARGARVECTOR8()
ReDim fb(1 To 8)
Grid8.Rows = 1
Grid8.Cols = 8
Grid8.RowHeight(0) = Grid8.Height
For I = 0 To 8 - 1
Grid8.ColWidth(I) = Grid8.Width / 8
Next I
End Sub
Sub PREGUNTARVALORESVECTOR8()
For J = 1 To 8
    Grid8.Col = J - 1
    Grid8.Text = InputBox(Str(J), "FRACCION DE MATERIAL DE LECHO EN
FRACCION")
    fb(J) = Val(Grid8.Text)
    Grid8.CellAlignment = 4
Next J
ANALIZARVECTOR8
End Sub
Sub ANALIZARVECTOR8()
For J = 1 To 8
FBMTOT = FBMTOT + fb(J)
Next J
If FBMTOT > 1.001 Or FBMTOT < 0.999 Then
MsgBox "LA SUMATORIA DE LA FRACCION DEL METRIAL " + Chr(13) + "DE
LECHO DEBE DAR 1", 0 + 16, "ERROR"
PREGUNTARVALORESVECTOR8
End If
End Sub
Sub MOSTRARVECTOR8()
```

```
For I = 1 To 8
Grid8.Col = I - 1
Grid8.Text = fb(I)
Grid8.CellAlignment = 4
Next I
End Sub
Sub PAWLINMODIFICADO()
If temp <= XTEMP(1) Then
XNU = YNU(2) - (YNU(2) - YNU(1)) * ((XTEMP(2) - temp) / (XTEMP(2) - XTEMP(1)))
End If
If temp >= XTEMP(10) Then
XNU = YNU(14) - (YNU(14) - YNU(15)) * ((XTEMP(14) - temp) / (XTEMP(14) -
XTEMP(15)))
End If
If temp > XTEMP(1) And temp < XTEMP(10) Then
   For I = 2 To 9
      If temp <= XTEMP(I) Then
         XNU = ((YNU(I) - YNU(I - 1)) * ((XTEMP(I) - temp) / (XTEMP(I) - XTEMP(I -
1)))) + YNU(I)
         I = 9
      End If
   Next I
End If
End Sub

Sub PAWLINLOGMODIFICADO()
If PAM < XPAM(1) Then
A = 10 ^ ((((Log(YYA(2)) - Log(YYA(1))) * Log(PAM) + (Log(YYA(1)) *
Log(XPAM(2)) - Log(YYA(2)) * Log(XPAM(1)))) / (Log(XPAM(2)) - Log(XPAM(1)))) *
(0.434294))
Else
   If PAM >= XPAM(6) Then
   A = 10 ^ ((((Log(YYA(6)) - Log(YYA(5))) * Log(PAM) + (Log(YYA(5)) *
Log(XPAM(6)) - Log(YYA(6)) * Log(XPAM(5)))) / (Log(XPAM(6)) - Log(XPAM(5)))) *
(0.434294))
   Else
      If PAM >= XPAM(1) And PAM < XPAM(6) Then
         For I = 2 To 14
            If PAM <= XPAM(I) Then
            A = 10 ^ ((((Log(YYA(I)) - Log(YYA(I - 1))) * Log(PAM) + (Log(YYA(I - 1))
* Log(XPAM(I)) - Log(YYA(I)) * Log(XPAM(I - 1)))) / (Log(XPAM(I)) - Log(XPAM(I -
1)))) * (0.434294))
            I = 15
            End If
         Next I
      End If
   End If
```

```
End If
End Sub
Sub PAWLINGMODIFICADO2()
If CAF < XFAC(1) Then
ACORR = 10 ^ ((((Log(YACORR(2)) - Log(YACORR(1))) * Log(PAM) +
(Log(YACORR(1)) * Log(XFAC(2)) - Log(YACORR(2)) * Log(XFAC(1)))) /
(Log(XFAC(2)) - Log(XFAC(1)))) * (0.434294))
Else
   If CAF >= XFAC(4) Then
   K = n - 1
   ACORR = 10 ^ ((((Log(YACORR(4)) - Log(YACORR(3))) * Log(PAM) +
(Log(YACORR(3)) * Log(XFAC(4)) - Log(YACORR(4)) * Log(XFAC(3)))) /
(Log(XFAC(4)) - Log(XFAC(3)))) * (0.434294))
   Else
      If CAF >= XFAC(1) And CAF < XFAC(4) Then
         For I = 2 To 3
            If PAM <= XFAC(I) Then
            ACORR = 10 ^ ((((Log(YACORR(I)) - Log(YACORR(I - 1))) * Log(PAM) +
(Log(YACORR(I - 1)) * Log(XFAC(I)) - Log(YACORR(I)) * Log(XFAC(I - 1)))) /
(Log(XFAC(I)) - Log(XFAC(I - 1)))) * (0.434294))
            I = 15
            End If
         Next I
      End If
   End If
End If
End Sub
'NO DAR IMPORTANCIA
Private Sub Form_MouseDown(Button As Integer, _
                     Shift As Integer, _
                     X As Single, Y As Single)
   'Esta función será llamada de forma automática cuando
   'pulsemos con el ratón sobre el formulario
   If Button = 1 Then
   Else
      'y si pulsamos botón derecho, pintamos rectángulo
      Line (X, Y)-(X + 250, Y + 250), vbRed, BF
   End If
End Sub
'NO DAR IMPORTANCIA
Private Sub Form_MouseMove(Button As Integer, _
                     Shift As Integer, _
                     X As Single, Y As Single)
   Select Case Button
   Case 2
         'Si nos movemos mientras pulsamos el botón derecho
         'dejamos un rastro de cuadrados rojos
```

```
        Line (X, Y)-(X + 250, Y + 250), vbRed, BF
    End Select
End Sub
'NO DAR IMPORTANCIA
Private Sub TEXT7_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyF1 ' Si pulsamos F1
        Text7.BackColor = vbGreen 'ponemos el fondo verde
End Select
End Sub
'NO DAR IMPORTANCIA
Private Sub TEXT7_KeyUp(KeyCode As Integer, Shift As Integer)
    'Esta función es invocada automáticamete cada vez que
    'soltemos una tecla
    Text7.BackColor = vbYellow  ' El color que tenía al principio
End Sub
```

5.- EINSTEIN PARA ARENAS.-

EINSTEIN PARA ARENAS

```
Dim uave As Double, depth As Double, w As Double, temp As Double, d65 As Double, I
As Double, J As Double
Dim d35 As Double, conc As Double, dn As Double, ds As Double, nd As Long, XNU As
Double, VS() As Double
Dim drl() As Double, dru() As Double, fb() As Double, fs() As Double, d() As Double,
xibqd() As Double
Dim fsl() As Double, fql() As Double, tbl As Double, tsl As Double, twl As Double, CORR
As Double
Dim tql As Double, CSIUS As Double, CTM As Double, DRB As Double, SCORR As
Double, DCORR As Double
Dim XTEMP() As Double, YKIVI() As Double, XCONC() As Double, YCORR() As
Double, AREA As Double, DISCH As Double
Dim X As Double, TOL As Double, XKS As Double, SRRS As Double, USHP As Double,
DEL As Double, DELKS As Double
Dim X2 As Double, XDKS As Double, RS As Double, A As Double, YDS As Double,
YDN As Double, PFS As Double
Dim PSI() As Double, XPSI As Double, YPSI As Double, XYPSI As Double, QSPT As
Double
Dim XX As Double, YY As Double, Y As Double, PHISH() As Double, XIBQB() As
Double, QSP() As Double
Dim DXKS As Double, P As Double, AP As Double, APP() As Double, N1 As Double,
NK As Double
Private Sub Command1_Click()
```

```
Do
uave = InputBox("M/S", "VELOCIDAD MEDIA")
If uave < 0 Then
MsgBox "LA VELOCIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While uave < 0

Do
depth = InputBox("M", "PROFUNDIDAD MEDIA")
If depth < 0 Then
MsgBox "LA PROFUNDIDAD NO PUEDE SER NEGATIVA", 0 + 16, "ERROR"
End If
Loop While depth < 0

Do
w = InputBox("M", "ANCHO DE LA SECCION")
If w < 0 Then
MsgBox "EL ANCHO DE LA SECCION NO PUEDE SER NEGATIVO", 0 + 16,
"ERROR"
End If
Loop While w < 0

Do
temp = InputBox("ºC", "TEMPERATURA DEL AGUA")
If temp < 0 Or temp > 100 Then
MsgBox "LA TEMPERATURA ESTA FUERA DE RANGO", 0 + 16, "ERROR"
End If
Loop While temp < 0 Or temp > 100

Do
d65 = InputBox("MM", "DIAMETRO 65% ES MAS FINO")
If d65 < 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While d65 < 0

Do
d35 = InputBox("MM", "DIAMETRO 35% ES MAS FINO")
If d35 < 0 Then
MsgBox "EL DIAMETRO NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While d35 < 0

Do
conc = InputBox("SUSPENDIDO,FONDO Y LAVADO =PPM", "CONCENTRACION
DE TODO EL MATERIAL")
If conc < 0 Then
```

```
MsgBox "LA CONCENTRACION NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While conc < 0

Do
dn = InputBox("M", "PORCION NO MUESTREADA PROFUDIDAD")
If dn < 0 Then
MsgBox "LA PORCION NO MUETREADA" + Chr(13) + "NO PUEDE SER
NEGATIVO", 0 + 16, "ERROR"
End If
Loop While dn < 0

Do
ds = InputBox("EN VERTICALES DE MUESTREO =M", "PROFUNDIDAD MEDIA")
If ds < 0 Then
MsgBox "LA PROFUNDIDAD MEDIA EN VERTICALES" + Chr(13) + "NO PUEDE
DAR NEGATIVO", 0 + 16, "ERROR"
End If
Loop While ds < 0

Do
nd = InputBox("ESTE VALOR DEBE SER ENTERO", "# DE FRACCIONES")
If nd < 0 Then
MsgBox "# DE FRACCIONES NO PUEDE SER NEGATIVO", 0 + 16, "ERROR"
End If
Loop While nd < 0

Do While dn > ds
MsgBox "LA PORCION NO NUESTREADA DE LA" + Chr(13) + " PROFUNDIDAD,
NO PUEDE SER MAYOR QUE" + Chr(13) + "LA PROFUNDIDAD MEDIA EN
VERTICALES DE" + Chr(13) + "MUESTREO", 0 + 16, "ERROR"

   Do
   dn = InputBox("M", "PORCION NO MUESTREADA PROFUDIDAD")
   If dn < 0 Then
   MsgBox "LA PORCION NO MUETREADA" + Chr(13) + "NO PUEDE SER
NEGATIVO", 0 + 16, "ERROR"
   End If
   Loop While dn < 0

   Do
   ds = InputBox("EN VERTICALES DE MUESTREO =M", "PROFUNDIDAD
MEDIA")
   If ds < 0 Then
   MsgBox "LA PROFUNDIDAD MEDIA EN VERTICALES" + Chr(13) + "NO PUEDE
DAR NEGATIVO", 0 + 16, "ERROR"
   End If
```

```
   Loop While ds < 0

Loop

Do While nd > 10
nd = InputBox("ESTE VALOR DEBE SER ENTERO", "# DE FRACCIONES")
If nd > 10 Then
MsgBox "# DE FRACCIONES NO PUEDE SER MAYOR QUE 10", 0 + 16, "ERROR"
End If
Loop

Text1.Text = Str(uave) + "M/S"
Text2.Text = Str(depth) + "M"
Text3.Text = Str(w) + "M"
Text4.Text = Str(temp) + "ºC"
Text5.Text = Str(d65) + "MM"
Text6.Text = Str(d35) + "MM"
Text7.Text = Str(conc) + "PPM"
Text8.Text = Str(dn) + "M"
Text9.Text = Str(ds) + "M"
Text10.Text = Str(nd)

CSIUS = 0.3048: CTM = 0.9072

CARGARVECTOR9
CARGARVECTOR10
CARGARVECTOR11

uave = uave / CSIUS
depth = depth / CSIUS
w = w / CSIUS
temp = 1.8 * temp + 32
d65 = d65 / (1000 * CSIUS)
d35 = d35 / (1000 * CSIUS)
dn = dn / CSIUS
ds = ds / CSIUS

If nd >= 9 Then

ReDim drl(1 To 11)
Grid7.Rows = 1
Grid7.Cols = 11
Grid7.RowHeight(0) = Grid7.Height
For I = 0 To 11 - 1
Grid7.ColWidth(I) = Grid7.Width / 11
Next I
```

```
ReDim dru(1 To 11)
Grid8.Rows = 1
Grid8.Cols = 11
Grid8.RowHeight(0) = Grid8.Height
For I = 0 To 11 - 1
Grid8.ColWidth(I) = Grid8.Width / 11
Next I
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''
DRB = 0.03125
drl(1) = 0.002
drl(2) = 0.0156
drl(3) = drl(1)
dru(1) = drl(2)
dru(2) = 2 * DRB
dru(3) = dru(2)

For J = 4 To 11
drl(J) = 2 ^ (J - 3) * DRB
dru(J) = 2 * 2 ^ (J - 3) * DRB
Next J
End If

For I = 1 To 11
Grid7.Col = I - 1
Grid7.Text = drl(I)
Grid7.CellAlignment = 4
Next I

For I = 1 To 11
Grid8.Col = I - 1
Grid8.Text = dru(I)
Grid8.CellAlignment = 4
Next I



SCORR = 0.98
DCORR = 0.02

ReDim YCORR(1 To 26)
Grid12.Rows = 1
Grid12.Cols = 26
Grid12.RowHeight(0) = Grid12.Height
```

```
For I = 0 To 26 - 1
Grid12.ColWidth(I) = Grid12.Width / 26
Next I

For J = 1 To 26
SCORR = SCORR + DCORR
YCORR(J) = SCORR
Next J

For I = 1 To 26
Grid12.Col = I - 1
Grid12.Text = YCORR(I)
Grid12.CellAlignment = 4
Next I

AREA = depth * w
DISCH = uave * AREA

HALLANDOCORR

QSM = 0.002696843 * CORR * DISCH * conc

HALLANDOXNU

ReDim VS(1 To 11)
Grid13.Rows = 1
Grid13.Cols = 11
Grid13.RowHeight(0) = Grid13.Height
For I = 0 To 11 - 1
Grid13.ColWidth(I) = Grid13.Width / 11
Next I

ReDim d(1 To 11)
Grid6.Rows = 1
Grid6.Cols = 11
Grid6.RowHeight(0) = Grid6.Height
For I = 0 To 11 - 1
Grid6.ColWidth(I) = Grid6.Width / 11
Next I

If nd >= 9 Then

For J = 1 To 11
d(J) = (drl(J) * dru(J)) ^ 0.5 / 304.8
VS(J) = ((2 / 3 * 32.17 * 1.65 * d(J) ^ 3 + 36 * XNU ^ 2) ^ 0.5 - 6 * XNU) / d(J)
Next J
```

```
   If nd = 10 Then
      For J = 3 To nd
      d(J) = d(J + 1)
      VS(J) = VS(J + 1)
      Next J
   Else
      For J = 1 To nd
      d(J) = d(J + 2)
      VS(J) = VS(J + 2)
      Next J
   End If
Else
   For J = 1 To nd
   d(J) = (dru(J) * drl(J)) ^ 0.5 / 304.8
   VS(J) = ((2 / 3 * 32.17 * 1.65 * d(J) ^ 3 + 36 * XNU ^ 2) ^ 0.5 - 6 * XNU) / d(J)
   Next J
End If

For I = 1 To 11
Grid13.Col = I - 1
Grid13.Text = VS(I)
Grid13.CellAlignment = 4
Next I

For I = 1 To 11
Grid6.Col = I - 1
Grid6.Text = d(I)
Grid6.CellAlignment = 4
Next I
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''
rscom
PLATE4
QSPT = QSM * PFS

ReDim PSI(1 To 10)
Grid14.Rows = 1
Grid14.Cols = 10
Grid14.RowHeight(0) = Grid14.Height
For I = 0 To 10 - 1
Grid14.ColWidth(I) = Grid14.Width / 10
Next I
```

```
For J = 1 To nd
XPSI = 1.65 * d35 / RS
YPSI = 0.66 * d(J) / RS
XYPSI = XPSI - YPSI
If XYPSI < 0 Then
PSI(J) = YPSI
Else
PSI(J) = XPSI
End If
Next J

For I = 1 To 10
Grid14.Col = I - 1
Grid14.Text = PSI(I)
Grid14.CellAlignment = 4
Next I


ReDim QSP(1 To 10)
Grid15.Rows = 1
Grid15.Cols = 10
Grid15.RowHeight(0) = Grid15.Height
For I = 0 To 10 - 1
Grid15.ColWidth(I) = Grid15.Width / 10
Next I


ReDim XIBQB(1 To 10)
Grid5.Rows = 1
Grid5.Cols = 10
Grid5.RowHeight(0) = Grid5.Height
For I = 0 To 10 - 1
Grid5.ColWidth(I) = Grid5.Width / 10
Next I


ReDim PHISH(1 To 10)
Grid16.Rows = 1
Grid16.Cols = 10
Grid16.RowHeight(0) = Grid16.Height
For I = 0 To 10 - 1
Grid16.ColWidth(I) = Grid16.Width / 10
Next I


For J = 1 To nd
```

```
XX = PSI(J)
PLATE5
PHISH(J) = YY
XIBQB(J) = 43.2 * w * 1200 * PHISH(J) / 2 * d(J) ^ 1.5 * fb(J)
QSP(J) = fs(J) * QSPT
Next J


For I = 1 To 10
Grid15.Col = I - 1
Grid15.Text = QSP(I)
Grid15.CellAlignment = 4
Next I

For I = 1 To 10
Grid16.Col = I - 1
Grid16.Text = PHISH(I)
Grid16.CellAlignment = 4
Next I

For I = 1 To 10
Grid5.Col = I - 1
Grid5.Text = XIBQB(I)
Grid5.CellAlignment = 4
Next I

ReDim APP(1 To 10)
Grid17.Rows = 1
Grid17.Cols = 10
Grid17.RowHeight(0) = Grid17.Height
For I = 0 To 10 - 1
Grid17.ColWidth(I) = Grid17.Width / 10
Next I

DXKS = 30 * 2 * X * depth / XKS
P = 2.303 * Log(DXKS) * 0.434294
AP = dn / ds
For J = 1 To nd
APP(J) = 2 * d(J) / depth
Next J


For I = 1 To 10
Grid17.Col = I - 1
Grid17.Text = APP(I)
Grid17.CellAlignment = 4
```

```
Next I
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''

'''''''''''''''
LLAMANDOSDR
N1 = n + 1
NK = n + K

End Sub

Private Sub Command2_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Text13.Text = ""
Text14.Text = ""
Text15.Text = ""

End Sub

Private Sub Command3_Click()
Form1.Show
End Sub
Private Sub Command4_Click()
End
End Sub

Sub CARGARVECTOR9()
ReDim XTEMP(1 To 14)
Grid9.Rows = 1
Grid9.Cols = 14
Grid9.RowHeight(0) = Grid9.Height
For I = 0 To 14 - 1
Grid9.ColWidth(I) = Grid9.Width / 14
Next I
```

```
XTEMP(1) = 32: XTEMP(6) = 80: XTEMP(11) = 160
XTEMP(2) = 40: XTEMP(7) = 90: XTEMP(12) = 180
XTEMP(3) = 50: XTEMP(8) = 100: XTEMP(13) = 200
XTEMP(4) = 60: XTEMP(9) = 120: XTEMP(14) = 212
XTEMP(5) = 70: XTEMP(10) = 140
For I = 1 To 14
Grid9.Col = I - 1
Grid9.Text = XTEMP(I)
Grid9.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR10()
ReDim YKIVI(1 To 14)
Grid10.Rows = 1
Grid10.Cols = 14
Grid10.RowHeight(0) = Grid10.Height
For I = 0 To 14 - 1
Grid10.ColWidth(I) = Grid10.Width / 14
Next I
YKIVI(1) = 1.931 * 10 ^ -5: YKIVI(6) = 0.93 * 10 ^ -5: YKIVI(11) = 0.442 * 10 ^ -5
YKIVI(2) = 1.664 * 10 ^ -5: YKIVI(7) = 0.826 * 10 ^ -5: YKIVI(12) = 0.386 * 10 ^ -5
YKIVI(3) = 1.41 * 10 ^ -5: YKIVI(8) = 0.739 * 10 ^ -5: YKIVI(13) = 0.341 * 10 ^ -5
YKIVI(4) = 1.217 * 10 ^ -5: YKIVI(9) = 0.609 * 10 ^ -5: YKIVI(14) = 0.319 * 10 ^ -5
YKIVI(5) = 1.058 * 10 ^ -5: YKIVI(10) = 0.514 * 10 ^ -5
For I = 1 To 14
Grid10.Col = I - 1
Grid10.Text = YKIVI(I)
Grid10.CellAlignment = 4
Next I
End Sub

Sub CARGARVECTOR11()
ReDim XCONC(1 To 26)
Grid11.Rows = 1
Grid11.Cols = 26
Grid11.RowHeight(0) = Grid11.Height
For I = 0 To 26 - 1
Grid11.ColWidth(I) = Grid11.Width / 26
Next I
XCONC(1) = 15.9 * 10 ^ 3: XCONC(6) = 159 * 10 ^ 3: XCONC(11) = 279 * 10 ^ 3:
XCONC(16) = 380 * 10 ^ 3: XCONC(21) = 467 * 10 ^ 3: XCONC(26) = 542 * 10 ^ 3
XCONC(2) = 46.9 * 10 ^ 3: XCONC(7) = 184 * 10 ^ 3: XCONC(12) = 300 * 10 ^ 3:
XCONC(17) = 398 * 10 ^ 3: XCONC(22) = 483 * 10 ^ 3
XCONC(3) = 76.9 * 10 ^ 3: XCONC(8) = 209 * 10 ^ 3: XCONC(13) = 321 * 10 ^ 3:
XCONC(18) = 416 * 10 ^ 3: XCONC(23) = 498 * 10 ^ 3
```

```
XCONC(4) = 105 * 10 ^ 3: XCONC(9) = 233 * 10 ^ 3: XCONC(14) = 341 * 10 ^ 3:
XCONC(19) = 434 * 10 ^ 3: XCONC(24) = 513 * 10 ^ 3
XCONC(5) = 132 * 10 ^ 3: XCONC(10) = 256 * 10 ^ 3: XCONC(15) = 361 * 10 ^ 3:
XCONC(20) = 451 * 10 ^ 3: XCONC(25) = 528 * 10 ^ 3
For I = 1 To 26
Grid11.Col = I - 1
Grid11.Text = XCONC(I)
Grid11.CellAlignment = 4
Next I
End Sub


Sub HALLANDOCORR()
'INTERPOLANDO PARA HALLAR CORR
If conc <= XCONC(1) Then
CORR = YCORR(2) - (YCORR(2) - YCORR(1)) * ((XCONC(2) - conc) / (XCONC(2) -
XCONC(1)))
End If
If conc >= XCONC(26) Then
CORR = YCORR(25) - (YCORR(25) - YCORR(26)) * ((XCONC(25) - conc) /
(XCONC(25) - XCONC(26)))
End If
If conc > XCONC(1) And conc < XCONC(26) Then
   For I = 2 To 25
      If conc <= XCONC(I) Then
         CORR = ((YCORR(I) - YCORR(I - 1)) * ((XCONC(I) - conc) / (XCONC(I) -
XCONC(I - 1)))) + YCORR(I)
         I = 26
      End If
   Next I
End If
End Sub

Sub HALLANDOXNU()
'INTERPOLANDO PARA HALLAR XNU
If temp <= XTEMP(1) Then
XNU = YKIVI(2) - (YKIVI(2) - YKIVI(1)) * ((XTEMP(2) - temp) / (XTEMP(2) -
XTEMP(1)))
End If
If temp >= XTEMP(14) Then
XNU = YKIVI(13) - (YKIVI(13) - YKIVI(14)) * ((XTEMP(13) - temp) / (XTEMP(13) -
XTEMP(14)))
End If
If temp > XTEMP(1) And temp < XTEMP(14) Then
   For I = 2 To 13
      If temp <= XTEMP(I) Then
```

```
        XNU = ((YKIVI(I) - YKIVI(I - 1)) * ((XTEMP(I) - temp) / (XTEMP(I) - XTEMP(I
- 1)))) + YKIVI(I)
        I = 26
      End If
   Next I
End If
End Sub
Sub rscom()
X = 1.6
TOL = 0.001
XKS = d65
dies
VEINTE
End If
End Sub
Sub dies()
XDKS = 12.27 * X * depth / XKS
SRRS = uave / (32.65 * (0.434294 * Log(XDKS)))
USHP = SRRS * 5.68
DEL = 11.6 * XNU / USHP
DELKS = XKS / DEL
plate3
delx = X - X2
   If Abs(delx) < TOL Then
   CONTINUE20
   X = X2
   dies
End Sub
Sub VEINTE()
   XDKS = 12.27 * X * depth / XKS
   SRRS = uave / (32.65 * Log(XDKS) * 0.434294)
   RS = SRRS ^ 2
End Sub

Sub plate3()
If DELKS <= 0.4 Then
X2 = 1.769 * 0.434294 * Log(DELKS / 0.8)
ElseIf DELKS <= 0.56 Then
X2 = 1.495 * 0.434294 * Log(DELKS / 0.059)
ElseIf DELKS <= 0.76 Then
X2 = 0.92 * Log(DELKS / 0.0145) * 0.434294
ElseIf DELKS <= 0.96 Then
X2 = 0.292 * Log(DELKS / 0.0000029) * 0.434294
ElseIf DELKS <= 1.35 Then
X2 = 0.277 * Log(632000 / DELKS) * 0.434294
ElseIf DELKS <= 3 Then
X2 = 1.115 * Log(34.4 / DELKS) * 0.434294
```

```
ElseIf DELKS <= 4 Then
X2 = 0.725 * Log(128 / DELKS) * 0.434294
ElseIf DELKS <= 6.7 Then
X2 = 0.399 * Log(2160 / DELKS) * 0.434294
Else
X2 = 1
End If
End Sub
Sub PLATE4()
XKS = d65
A = 30.2 * X / XKS
YDS = ds * Log(A * ds) - ds
YDN = dn * Log(A * dn) - dn
PFS = (YDS - YDN) / YDS
End Sub
Sub PLATE5()
If XX <= 0.77 Then
YY = (7.56 / XX) ^ 1.01
ElseIf XX <= 2.12 Then
YY = (5.35 / XX) ^ 1.19
ElseIf XX <= 4.1 Then
YY = (4.1 / XX) ^ 1.67
ElseIf XX <= 6.1 Then
YY = (4.1 / XX) ^ 2.3
ElseIf XX <= 11 Then
YY = (4.6 / XX) ^ 3.23
ElseIf XX <= 16.7 Then
YY = (5.66 / XX) ^ 4.26
ElseIf XX <= 22.5 Then
YY = (9.28 / XX) ^ 7.81
Else
YY = (13.1 / XX) ^ 12.66
End If

End Sub
Sub LLAMANDOSDR()
J = 0: K = 0: n = 0

ReDim fb(1 To 10)
Grid3.Rows = 1
Grid3.Cols = 10
Grid3.RowHeight(0) = Grid3.Height
For I = 0 To 10 - 1
Grid3.ColWidth(I) = Grid3.Width / 10
Next I

ReDim fs(1 To 10)
```

```
Grid4.Rows = 1
Grid4.Cols = 10
Grid4.RowHeight(0) = Grid4.Height
For I = 0 To 10 - 1
Grid4.ColWidth(I) = Grid4.Width / 10
Next I
CONTINUEDIES
CONTINUEVEINTE
CONTINUETREINTA



End Sub

Sub CONTINUEDIES()
If fb(J + 1) > 0 And fs(J + 1) > 0 Then
   CONTINUETREINTA
      If K <= 0 Then
      CONTINUEVEITE
      n = n + 1
End Sub
Sub CONTINUEVEINTE()
      J = J + 1
      If J = nd Then
      CONTINUEDIES
End Sub
Sub CONTINUETREINTA()
   K = K + 1
   J = J + 1
      If J = nd Then
   CONTINUEDIES
End If
End Sub
```