[ANEXO 13]

Descripción de la estructura del programa “diseño”

En este anexo se presenta la estructura del código del programa “diseño” con los comentarios y descripción de la funcionalidad de cada una de las rutinas desarrolladas.

En la primera parte se describe la estructura del módulo genético y en la segunda el módulo de verificación.

1. Módulo Genético

```
Option Compare Database                              ‘encabezado del programa
Option Explicit
Type tIndvi
tsensor As Byte: tlogics As Byte: tvalvula As Byte
tsensor2 As Byte: tvalvula2 As Byte
asensor As Byte: alogics As Byte: avalvula As Byte
fit As Single: pfds As Single
End Type
Type tDevice
la As Single: las As Single: lad As Single: ladd As Single: ladu As Single
lasd As Single: lasu As Single: laddn As Single: laddc As Single: ladun As Single
laduc As Single: lasdn As Single: lasdc As Single: lasun As Single: lasuc As Single
MTTR  As Single: Ti As Single: SD As Single: sff As Single: type As String
End Type
Const n_indv = 100, nm_indv = n_indv * 0.2, fre_muta = 10
Const fitmax = 1, t = 0.01
Dim n_sen, n_sen2, n_log, n_val, n_val2
Dim nb_sen, nb_sen2, nb_log, nb_val, nb_val2, n_bits
Dim cantd_sen, cantd_val, cantd_sen2, cantd_val2
Dim tiposen, tiposen2, tipoval, tipoval2
Dim db, sil_tg, caso As Byte
'-------------------------------------------------------
Public Function algoritmo_gen(caso_tr As Byte) As Boolean
Dim j As Byte, RRF_tg
Dim pob(1 To n_indv) As tIndvi
Dim guard(1 To n_indv) As tIndvi
Dim pobla, sif, ind, truc, comi
Set db = CurrentDb()
Set truc = db.Openrecordset("truco")
comi = truc!Comilla             'se emplea para poner las comillas
Set pobla = db.Openrecordset("poblacion")
Set sif = db.Openrecordset("genetico")

sil_tg = sif!SIL                'se obtiene los parametros para el analisis
Select Case sil_tg
    Case 1
    RRF_tg = 100
    Case 2
    RRF_tg = 1000
    Case 3
    RRF_tg = 10000
End Select

With sif
tiposen = !tipo_sensor1
cantd_sen = !n_sensor1
tiposen2 = !tipo_sensor2
cantd_sen2 = !n_sensor2
tipoval = !tipo_valvula1
```

```
cantd_val = !n_valvula1
tipoval2 = !tipo_valvula2
cantd_val2 = !n_valvula2
End With

If IsNull(tiposen2) Then                'determina en que caso se esta
    If IsNull(tipoval2) Then            'trabajando a saber:
        caso = 1                        ' caso 1: 1 sensor y 1 válvula
    Else                                ' caso 2: 1 sensor y 2 válvulas
        caso = 2
    End If
Else
    If IsNull(tipoval2) Then
        caso = 3                        'caso 3: 2 sensores y 1 válvula
    Else                                'caso 4: 2 sensores y 2 válvulas
        caso = 4
    End If
End If
caso_tr = caso
                                        'se obtiene la cantidad de equipos
n_sen = elementos(tiposen)              'existentes en la base de datos
If comprobacion(n_sen, tiposen, "sensor/transmisor") Then 'comprueba existencia
    nb_sen = n_bitele(n_sen)            'determina el nº de bits necesarios
Else: GoTo mensaj                       'para trabajar
End If

If caso = 3 Or caso = 4 Then
    n_sen2 = elementos(tiposen2)
    If comprobacion(n_sen2, tiposen2, "sensor/transmisor") Then
        nb_sen2 = n_bitele(n_sen2)
    Else: GoTo mensaj
    End If
End If

n_log = elementos("Logicsolvers")
nb_log = n_bitele(n_log)

n_val = elementos(tipoval)
If comprobacion(n_val, tipoval, "final de control") Then
    nb_val = n_bitele(n_val)
Else: GoTo mensaj
End If

If caso = 2 Or caso = 4 Then
    n_val2 = elementos(tipoval2)
    If comprobacion(n_val2, tipoval2, "final de control") Then
        nb_val2 = n_bitele(n_val2)
    Else: GoTo mensaj
    End If
End If

n_bits = nb_sen + nb_sen2 + nb_log + nb_val + n_val2 + 9
```

```
ind = 0
poblacion_ini pob    'poblacion inicial
Do                   'proceso genético
mutacion pob
fitness pob, n_indv, RRF_tg
burbuja pob
cruce pob, guard, RRF_tg
For j = 1 To n_indv
pob(j) = guard(j)
Next
burbuja pob
ind = ind + 1
Loop Until pob(1).fit >= fitmax * 0.95 Or ind = 100
Agregar pobla, pob, RRF_tg
MsgBox "Se evolucionó " & ind & " veces", vbInformation, "Evoluciones"
algoritmo_gen = True
Exit Function
mensaj:
MsgBox "El proceso de determinación se ha detenido!" _
        , vbInformation, "SIS: Diseño y verificación"
algoritmo_gen = False
End Function
'-------------------------------------------------------
Private Function elementos(sql) As Byte    'determina el nº de elementos en
Dim n_elem, elem, a                        'la base de datos
n_elem = 0

Set elem = db.Openrecordset(sql)
Do Until elem.EOF
n_elem = n_elem + 1
a = elem.GetRows
Loop
elementos = n_elem
End Function
'-------------------------------------------------------
Private Function n_bitele(n_elemen) As Byte   'determina el n de bits
Dim j As Byte, flag As Boolean                'necesarios para tarbajar
flag = False
j = 0
Do
    j = j + 1
    If n_elemen < 2 ^ j And n_elemen > (2 ^ (j - 1) - 1) Then
        n_bitele = j
        flag = True
    End If
Loop Until flag = True
End Function
'-------------------------------------------------------
Private Function comprobacion(n_elem, tipoelem, name_device As String) As Boolean
If n_elem = 0 Then
comprobacion = False
```

```
MsgBox "En la actual base de datos no existe ningún " & Chr(13) & _
       "elemento " & name_device & " de/tipo " & tipoelem & "." & Chr(13) & _
       "", vbCritical, "SIS: Diseño y verificación"
Else: comprobacion = True
End If
End Function
'-------------------------------------------------------
Private Sub poblacion_ini(pob() As tIndvi)
Dim j As Byte
Randomize
For j = 1 To n_indv
    pob(j).tsensor = Int((Rnd * n_sen) + 1) 'asigna valor de 1 a n_sensor
    If caso = 3 Or caso = 4 Then
        pob(j).tsensor2 = Int((Rnd * n_sen2) + 1)
    End If
    pob(j).tlogics = Int((Rnd * n_log) + 1)
    pob(j).tvalvula = Int((Rnd * n_val) + 1)
    If caso = 2 Or caso = 4 Then
        pob(j).tvalvula2 = Int((Rnd * n_val2) + 1)
    End If
    pob(j).asensor = Int((Rnd * 7) + 1)      'asigna valor de 1 a 7
    pob(j).alogics = Int((Rnd * 7) + 1)
    pob(j).avalvula = Int((Rnd * 7) + 1)
    pob(j).fit = 0
Next
End Sub
'-------------------------------------------------------
Private Sub burbuja(pobl() As tIndvi)
Dim temp As tIndvi
Dim i#, j#, flag As Byte
Do
For i = 1 To n_indv
    flag = 0
    For j = 1 To n_indv - i
        If pobl(j + 1).fit > pobl(j).fit Then
        temp = pobl(j)
        pobl(j) = pobl(j + 1)
        pobl(j + 1) = temp
        flag = 1
        End If
    Next
Next
Loop While flag = 1
End Sub
'-------------------------------------------------------
    ' Agrega 15 registros nuevos al Recordset utilizando
    ' datos transferidos del procedimiento que llama.
    ' El registro nuevo pasa a ser el registro actual.
Private Function Agregar(rstTemp, pobl() As tIndvi, RRF_tg)
Dim j  As Byte
For j = 1 To 15
```

```
With rstTemp
    .AddNew
    !tipo_sensor = det_device(pobl(j).tsensor, tiposen)
    !tipo_sensor2 = det_device(pobl(j).tsensor2, tiposen2)
    !tipo_logics = det_device(pobl(j).tlogics, "Logicsolvers")
    !tipo_valvula = det_device(pobl(j).tvalvula, tipoval)
    !tipo_valvula2 = det_device(pobl(j).tvalvula2, tipoval2)
    !asensor = arq_det(pobl(j).asensor)
    !alogics = arq_det(pobl(j).alogics)
    !avalvula = arq_det(pobl(j).avalvula)
    !fit = pobl(j).fit
    !PFD = pobl(j).pfds
    !tiposen = tiposen
    !tiposen2 = tiposen2
    !tipoval = tipoval
    !tipoval2 = tipoval2
    !cantd_sen = cantd_sen
    !cantd_sen2 = cantd_sen2
    !cantd_val = cantd_val
    !cantd_val2 = cantd_val2
    If pobl(j).pfds = 0 Then
    !RRF = 0
    Else
    !RRF = 1 / pobl(j).pfds
    End If
    !target = RRF_tg
    .Update
End With
Next
End Function
'-------------------------------------------------------
Private Sub mutacion(pob() As tIndvi)
Dim j
j = n_indv * n_bits * fre_muta / 100
Randomize
Select Case caso
    Case 1: mutacion_caso1 pob, j
    Case 2: mutacion_caso2 pob, j
    Case 3: mutacion_caso3 pob, j
    Case 4: mutacion_caso4 pob, j
End Select
End Sub
'-------------------------------------------------------
Private Function mascara(n_bit) As Byte
Dim bit As Byte
Randomize
bit = Int((Rnd * n_bit) + 1)
mascara = (2 ^ (bit)) - 1
End Function
'-------------------------------------------------------
Private Function fit_arq(arq_device As Byte) As Single
Dim a
```

```
a = 1 / 7
Select Case arq_device
    Case 1  '1oo1
        fit_arq = 1
    Case 2  '1oo2
        fit_arq = 1 - a
    Case 3  '2oo2
        fit_arq = 1 - (4 * a)
    Case 4  '2oo3
        fit_arq = 1 - (6 * a)
    Case 5  '1oo1D
        fit_arq = 1 - (2 * a)
    Case 6  '2oo2D
        fit_arq = 1 - (5 * a)
    Case 7  '1oo2D
        fit_arq = 1 - (3 * a)
End Select
End Function
'-------------------------------------------------------
Private Function fit_PFD(PFD_calc As Single, RRF_tg) As Single
Dim RRF_calc As Single, error_calc As Single
RRF_calc = 1 / PFD_calc
error_calc = e_RRF(RRF_calc, RRF_tg)
fit_PFD = (1 / (1 + error_calc))
End Function
'-------------------------------------------------------
Private Function e_RRF(RRF_calc As Single, RRF_tg) As Single
e_RRF = (RRF_tg - RRF_calc) / 100
If e_RRF < 0 Then
e_RRF = 1000        'alto grado de error
End If
End Function
'-------------------------------------------------------
Private Sub fitness(pobl() As tIndvi, n_i, RRF_tg)
Dim j As Byte, a
If RRF_tg = 10000 Then
      a = 0.999
Else: a = 0.7
End If
Dim senso As tDevice, senso2 As tDevice, logsol As tDevice, _
    valvul As tDevice, valvul2 As tDevice
For j = 1 To n_i
   calc pobl(j), senso, senso2, logsol, valvul, valvul2
   pobl(j).pfds = calcpfd(pobl(j), senso, senso2, logsol, valvul, valvul2)
   pobl(j).fit = a * fit_PFD(pobl(j).pfds, RRF_tg) + ((1 - a) / 3) *
(fit_arq(pobl(j).asensor) + fit_arq(pobl(j).alogics) + fit_arq(pobl(j).avalvula))
Next
End Sub
'-------------------------------------------------------
Private Function cruce(pob() As tIndvi, guard() As tIndvi, RRF_tg)
Dim k#, ven As Single, masc#, masc_n As Byte, nbn
Dim hijo(1 To 2) As tIndvi
```

```
Dim padre1#, padre2 As Byte
Randomize
k = 0
Do
    ven = nm_indv + (n_indv - nm_indv) * (1 - Exp(-t * k))
    padre1 = Int((Rnd * ven) + 1)
    Do
        Randomize
        padre2 = Int((Rnd * ven) + 1)
    Loop While padre1 = padre2

    Do
        masc = mascara(nb_sen)
        nbn = (2 ^ (nb_sen)) - 1
        masc_n = masc Xor nbn

        hijo(1).tsensor = (pob(padre1).tsensor And masc) Or (pob(padre2).tsensor And
masc_n)
        hijo(2).tsensor = (pob(padre1).tsensor And masc_n) Or (pob(padre2).tsensor
And masc)
    Loop While hijo(1).tsensor = 0 Or hijo(2).tsensor = 0 Or hijo(1).tsensor > n_sen
Or hijo(2).tsensor > n_sen

    If caso = 3 Or caso = 4 Then
        Do
            masc = mascara(nb_sen2)
            nbn = (2 ^ (nb_sen2)) - 1
            masc_n = masc Xor nbn

            hijo(1).tsensor2 = (pob(padre1).tsensor2 And masc) Or
(pob(padre2).tsensor2 And masc_n)
            hijo(2).tsensor2 = (pob(padre1).tsensor2 And masc_n) Or
(pob(padre2).tsensor2 And masc)
        Loop While hijo(1).tsensor2 = 0 Or hijo(2).tsensor2 = 0 Or hijo(1).tsensor2 >
n_sen2 Or hijo(2).tsensor2 > n_sen2
    End If

    Do
        masc = mascara(nb_log)
        nbn = (2 ^ (nb_log)) - 1
        masc_n = masc Xor nbn

        hijo(1).tlogics = (pob(padre1).tlogics And masc) Or (pob(padre2).tlogics And
masc_n)
        hijo(2).tlogics = (pob(padre1).tlogics And masc_n) Or (pob(padre2).tlogics
And masc)
    Loop While hijo(1).tlogics = 0 Or hijo(2).tlogics = 0 Or hijo(1).tlogics > n_log
Or hijo(2).tlogics > n_log

    Do
        masc = mascara(nb_val)
        nbn = (2 ^ (nb_val)) - 1
```

```
        masc_n = masc Xor nbn

        hijo(1).tvalvula = (pob(padre1).tvalvula And masc) Or (pob(padre2).tvalvula
And masc_n)
        hijo(2).tvalvula = (pob(padre1).tvalvula And masc_n) Or (pob(padre2).tvalvula
And masc)
    Loop While hijo(1).tvalvula = 0 Or hijo(2).tvalvula = 0 Or hijo(1).tvalvula >
n_val Or hijo(2).tvalvula > n_val

    If caso = 2 Or caso = 4 Then
        Do
            masc = mascara(nb_val2)
            nbn = (2 ^ (nb_val2)) - 1
            masc_n = masc Xor nbn

            hijo(1).tvalvula2 = (pob(padre1).tvalvula2 And masc) Or
(pob(padre2).tvalvula2 And masc_n)
            hijo(2).tvalvula2 = (pob(padre1).tvalvula2 And masc_n) Or
(pob(padre2).tvalvula2 And masc)
        Loop While hijo(1).tvalvula2 = 0 Or hijo(2).tvalvula2 = 0 Or
hijo(1).tvalvula2 > n_val2 Or hijo(2).tvalvula2 > n_val2
    End If

    Do
        masc = mascara(3)
        masc_n = masc Xor 7

        hijo(1).asensor = (pob(padre1).asensor And masc) Or (pob(padre2).asensor And
masc_n)
        hijo(2).asensor = (pob(padre1).asensor And masc_n) Or (pob(padre2).asensor
And masc)
    Loop While hijo(1).asensor = 0 Or hijo(2).asensor = 0

    Do
        masc = mascara(3)
        masc_n = masc Xor 7

        hijo(1).alogics = (pob(padre1).alogics And masc) Or (pob(padre2).alogics And
masc_n)
        hijo(2).alogics = (pob(padre1).alogics And masc_n) Or (pob(padre2).alogics
And masc)
    Loop While hijo(1).alogics = 0 Or hijo(2).alogics = 0

    Do
        masc = mascara(3)
        masc_n = masc Xor 7

        hijo(1).avalvula = (pob(padre1).avalvula And masc) Or (pob(padre2).avalvula
And masc_n)
        hijo(2).avalvula = (pob(padre1).avalvula And masc_n) Or (pob(padre2).avalvula
And masc)
```

```
    Loop While hijo(1).avalvula = 0 Or hijo(2).avalvula = 0 Or hijo(1).avalvula > 7
Or hijo(2).avalvula > 7

    fitness hijo, 2, RRF_tg
    If hijo(2).fit > hijo(1).fit Then
        hijo(1) = hijo(2)
    End If
    k = k + 1
    guard(k) = hijo(1)

Loop While k < n_indv And guard(k).fit < fitmax

End Function
'-------------------------------------------------------
'Este procedimiento calcula todas las tasas necesarias para
'calcular las PFD, PFS, MTTFs
Private Sub CalcTasas(ladu, ladd, lasu, lasd, cantseg, cantpel, B, _
device As tDevice)

device.la = ladu + ladd + lasu + lasd
device.las = (lasu + lasd) * cantseg
device.lad = (ladu + ladd) * cantpel
device.ladd = ladd * cantpel
device.ladu = ladu * cantpel
device.lasd = lasd * cantseg
device.lasu = lasu * cantseg
device.laddn = (1 - B) * device.ladd
device.laddc = B * device.ladd
device.ladun = (1 - B) * device.ladu
device.laduc = B * device.ladu
device.lasdn = (1 - B) * device.lasd
device.lasdc = B * device.lasd
device.lasun = (1 - B) * device.lasu
device.lasuc = B * device.lasu
device.sff = (1 - (device.ladu / (device.lad + device.las))) * 100
End Sub
'-------------------------------------------------------
'Calcula PFD
Private Function PFD(arq, device As tDevice) As Single

Select Case arq
    Case 0
    PFD = 1000
    Case 1  '1oo1
    PFD = device.ladd * device.MTTR + (device.ladu * device.Ti) / 2
    Case 2  '1oo2
    PFD = (device.laduc * device.Ti) / 2 + device.laddc * device.MTTR + (device.laddn
* device.MTTR) ^ 2 + ((device.laddn * device.MTTR * device.ladun * device.Ti) ^ 2) /
2 + ((device.ladun * device.Ti) ^ 2) / 3
    Case 3  '2oo2
    PFD = device.laddc * device.MTTR + (device.laduc * device.Ti) / 2 + 2 *
device.laddn * device.MTTR + device.ladun * device.Ti
```

```
    Case 4  '2oo3
    PFD = ((3 / 2) * device.laduc * device.Ti + 3 * device.laddc * device.MTTR + 3 *
(device.laddn * device.MTTR) ^ 2 + (3 / 2) * device.laddn * device.MTTR *
device.ladun * device.Ti + (device.ladun * device.Ti) ^ 2)
    Case 5  '1oo1D
    PFD = device.ladu * device.Ti / 2
    Case 6  '2oo2D
    PFD = device.laduc * device.Ti / 2 + device.ladun * device.Ti
    Case 7  '1oo2D
    PFD = device.laduc * device.Ti / 2 + ((device.ladun * device.Ti) ^ 2) / 3
End Select
End Function
'-------------------------------------------------------
'Calcula PFS
Private Function PFS(arq, device As tDevice) As Single

Select Case arq
    Case 1  '1oo1
    PFS = device.lasd * device.SD + device.lasu * device.SD
    Case 2  '1oo2
    PFS = (device.lasdc + device.lasuc + 2 * device.lasdn + 2 * device.lasun) *
device.SD
    Case 3  '2oo2
    PFS = device.lasdc * device.SD + device.lasuc * device.SD + 2 * (device.lasdn *
device.MTTR * device.lasdn * device.SD + device.lasun * device.Ti * device.lasdn *
device.SD + device.lasdn * device.MTTR * device.lasun * device.SD + device.lasun *
device.Ti * device.lasun * device.SD)
    Case 4  '2oo3
    PFS = 3 * device.lasuc * device.SD + 3 * device.lasdc * device.SD + 6 *
(device.lasdn * device.MTTR * device.lasdn * device.SD + device.lasun * device.Ti *
device.lasdn * device.SD + device.lasdn * device.MTTR * device.lasun * device.SD +
device.lasun * device.Ti * device.lasun * device.SD)
    Case 5  '1oo1D
    PFS = (device.lasd + device.lasu + device.ladd) * device.SD
    Case 6  '2oo2D
    PFS = (device.lasuc + device.lasdc + device.laddc) * device.SD + 2 *
((device.lasdn + device.laddn) * device.MTTR * (device.lasdn + device.laddn) *
device.SD + device.lasun * device.Ti * (device.lasdn + device.laddn) * device.SD +
(device.lasdn + device.laddn) * device.MTTR * device.lasun * device.SD + device.lasun
* device.Ti * device.lasun * device.SD)
    Case 7  '1oo2D
    PFS = (device.lasdc + device.lasuc + device.laddc) * device.SD + (device.lasdn *
device.MTTR + device.laddn * device.MTTR) ^ 2 + (device.lasun * device.Ti) ^ 2

End Select
End Function
'-------------------------------------------------------
'Calcula MTTFs
Private Function MTTFs(arq, device As tDevice) As Single

Select Case arq
    Case 1  '1oo1
```

```
    MTTFs = 1 / (device.lasd + device.lasu)
    Case 2  '1oo2
    MTTFs = 1 / (device.lasdc + device.lasuc + 2 * device.lasdn + 2 * device.lasun)
    Case 3  '2oo2
    MTTFs = 1 / (device.lasdc + device.lasuc + 2 * (device.lasdn * device.MTTR *
device.lasdn + device.lasun * device.Ti * device.lasdn + device.lasdn * device.MTTR *
device.lasun + device.lasun * device.Ti * device.lasun))
    Case 4  '2oo3
    MTTFs = 1 / (3 * device.lasuc + 3 * device.lasdc + 6 * (device.lasdn *
device.MTTR * device.lasdn + device.lasun * device.Ti * device.lasdn + device.lasdn *
device.MTTR * device.lasun + device.lasun * device.Ti * device.lasun))
    Case 5  '1oo1D
    MTTFs = 1 / ((device.lasd + device.lasu + device.ladd))
    Case 6  '2oo2D
    MTTFs = 1 / ((device.lasuc + device.lasdc + device.laddc) + 2 * ((device.lasdn +
device.laddn) * device.MTTR * (device.lasdn + device.laddn) + device.lasun *
device.Ti * (device.lasdn + device.laddn) + (device.lasdn + device.laddn) *
device.MTTR * device.lasun + device.lasun * device.Ti * device.lasun))
    Case 7  '1oo2D
    MTTFs = 1 / ((device.lasdc + device.lasuc + device.laddc) + (device.lasdn *
device.MTTR + device.laddn * device.MTTR) ^ 2 + (device.lasun * device.Ti) ^ 2)
 End Select
End Function
'-----------------------------------------------------------
Private Sub calc(indvi As tIndvi, senso As tDevice, senso2 As tDevice, _
        logsol As tDevice, valvul As tDevice, valvul2 As tDevice)

Dim cpu As tDevice, diin As tDevice, diout As tDevice, anin As tDevice
Dim sens, sens2, logcs, valv, valv2
Dim ladu, ladd, lasu, lasd, cantseg, cantpel, B, i

Set sens = db.Openrecordset(tiposen)
i = 1
Do While i < indvi.tsensor
sens.GetRows                'coloca el cursor en el registro adecuado
i = i + 1
Loop
With sens
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
senso.MTTR = ![MTTR]
senso.Ti = ![Ti (tiempo de inspección)]
senso.SD = ![SD (tiempo de parada)]
senso.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, senso
```

```
If caso = 3 Or caso = 4 Then
Set sens2 = db.Openrecordset(tiposen2)
i = 1
Do While i < indvi.tsensor2
sens2.GetRows                'coloca el cursor en el registro adecuado
i = i + 1
Loop
With sens2
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
senso2.MTTR = ![MTTR]
senso2.Ti = ![Ti (tiempo de inspección)]
senso2.SD = ![SD (tiempo de parada)]
senso2.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, senso2
End If

Set valv = db.Openrecordset(tipoval)
i = 1
Do While i < indvi.tvalvula
valv.GetRows
i = i + 1
Loop
With valv
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
valvul.MTTR = ![MTTR]
valvul.Ti = ![Ti (tiempo de inspección)]
valvul.SD = ![SD (tiempo de parada)]
valvul.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, valvul

If caso = 2 Or caso = 4 Then
Set valv2 = db.Openrecordset(tipoval2)
i = 1
Do While i < indvi.tvalvula2
valv2.GetRows
i = i + 1
Loop
With valv2
```

```
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
valvul2.MTTR = ![MTTR]
valvul2.Ti = ![Ti (tiempo de inspección)]
valvul2.SD = ![SD (tiempo de parada)]
valvul2.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, valvul2
End If

Set logcs = db.Openrecordset("Logicsolvers")
i = 1
Do While i < indvi.tlogics
logcs.GetRows
i = i + 1
Loop
With logcs
ladu = ![ladu_CPU]
ladd = ![ladd_CPU]
lasu = ![lasu_CPU]
lasd = ![lasd_CPU]
cantseg = 1
cantpel = 1
B = ![B  (factor causa comun)] / 100
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, cpu

With logcs
ladu = ![ladu_DI]
ladd = ![ladd_DI]
lasu = ![lasu_DI]
lasd = ![lasd_DI]
cantseg = 1
cantpel = 1
B = ![B  (factor causa comun)] / 100
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, diin

With logcs
ladu = ![ladu_DO]
ladd = ![ladd_DO]
lasu = ![lasu_DO]
lasd = ![lasd_DO]
cantseg = 1
cantpel = 1
B = ![B  (factor causa comun)] / 100
End With
```

```
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, diout

With logcs
ladu = ![ladu_AI]
ladd = ![ladd_AI]
lasu = ![lasu_AI]
lasd = ![lasd_AI]
cantseg = 1
cantpel = 1
B = ![B  (factor causa comun)] / 100
logsol.MTTR = ![MTTR]
logsol.Ti = ![Ti (tiempo de inspección)]
logsol.SD = ![SD (tiempo de parada)]
logsol.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, anin

logsol.la = cpu.la + diin.la + diout.la + anin.la
logsol.las = cpu.las + diin.las + diout.las + anin.las
logsol.lad = cpu.lad + diin.lad + diout.lad + anin.lad
logsol.ladd = cpu.ladd + diin.ladd + diout.ladd + anin.ladd
logsol.ladu = cpu.ladu + diin.ladu + diout.ladu + anin.ladu
logsol.lasd = cpu.lasd + diin.lasd + diout.lasd + anin.lasd
logsol.lasu = cpu.lasu + diin.lasu + diout.lasu + anin.lasu
logsol.laddn = cpu.laddn + diin.laddn + diout.laddn + anin.laddn
logsol.laddc = cpu.laddc + diin.laddc + diout.laddc + anin.laddc
logsol.ladun = cpu.ladun + diin.ladun + diout.ladun + anin.ladun
logsol.laduc = cpu.laduc + diin.laduc + diout.laduc + anin.laduc
logsol.lasdn = cpu.lasdn + diin.lasdn + diout.lasdn + anin.lasdn
logsol.lasdc = cpu.lasdc + diin.lasdc + diout.lasdc + anin.lasdc
logsol.lasun = cpu.lasun + diin.lasun + diout.lasun + anin.lasun
logsol.lasuc = cpu.lasuc + diin.lasuc + diout.lasuc + anin.lasuc
logsol.sff = (1 - (logsol.ladu / (logsol.lad + logsol.las))) * 100
End Sub
'-------------------------------------------------------
Private Function calcpfd(individuo As tIndvi, senso As tDevice, _
        senso2 As tDevice, logsol As tDevice, valvul As tDevice, _
        valvul2 As tDevice) As Single

Dim arqs#, arql#, arqv As Byte, hft_s, hft_s2, hft_l, hft_v, hft_v2
Dim s_pfd, s2_pfd, l_pfd, v_pfd, v2_pfd

arqs = individuo.asensor
arql = individuo.alogics
arqv = individuo.avalvula

hft_s = hft(senso.type, senso.sff)
s_pfd = det_pfd(hft_s, senso, arqs)

If caso = 3 Or caso = 4 Then
    hft_s2 = hft(senso2.type, senso2.sff)
    s2_pfd = det_pfd(hft_s2, senso2, arqs)
```

```
Else: s2_pfd = 0
End If

hft_l = hft(logsol.type, logsol.sff)
l_pfd = det_pfd(hft_l, logsol, arql)

hft_v = hft(valvul.type, valvul.sff)
v_pfd = det_pfd(hft_v, valvul, arqv)

If caso = 2 Or caso = 4 Then
    hft_v2 = hft(valvul2.type, valvul2.sff)
    v2_pfd = det_pfd(hft_v2, valvul2, arqv)
Else: v2_pfd = 0
End If

calcpfd = cantd_sen * s_pfd + cantd_sen2 * s2_pfd + l_pfd + cantd_val * v_pfd +
cantd_val2 * v2_pfd

End Function
'-------------------------------------------------------
Private Function hft(Tipo As String, sff As Single)
Select Case Tipo
    Case "A"
        Select Case sff
            Case 0 To 60
                Select Case sil_tg
                    Case 1
                    hft = 0
                    Case 2
                    hft = 1
                    Case 3
                    hft = 2
                End Select
            Case 60 To 90
                Select Case sil_tg
                    Case 1
                    hft = 0
                    Case 2
                    hft = 0
                    Case 3
                    hft = 1
                End Select
            Case 90 To 100
                hft = 0
            Case Else
                hft = 3
        End Select
    Case "B"
        Select Case sff
            Case 0 To 60
                Select Case sil_tg
                    Case 1
```

```
                    hft = 1
                    Case 2
                    hft = 2
                    Case 3
                    hft = 3
                End Select
            Case 60 To 90
                Select Case sil_tg
                    Case 1
                    hft = 0
                    Case 2
                    hft = 1
                    Case 3
                    hft = 2
                End Select
            Case 90 To 99
                Select Case sil_tg
                    Case 1
                    hft = 0
                    Case 2
                    hft = 0
                    Case 3
                    hft = 1
                End Select
            Case 99 To 100
                hft = 0
            Case Else
                hft = 3
        End Select
End Select
End Function
'-------------------------------------------------------------
Private Function det_pfd(hft_device, device As tDevice, arq_device)
Select Case hft_device
    Case 0
        det_pfd = PFD(arq_device, device)
    Case 1
        If arq_device = 1 Or arq_device = 5 Then
            det_pfd = 1
        Else
            det_pfd = PFD(arq_device, device)
        End If
    Case 2
        If arq_device = 4 Then
            det_pfd = PFD(arq_device, device)
        Else
            det_pfd = 1
        End If
    Case 3
        det_pfd = 1
End Select
End Function
```

```
'-------------------------------------------------------------
Private Function arq_det(arq As Byte) As String
Select Case arq
    Case 1
    arq_det = "       1oo1"
    Case 2
    arq_det = "       1oo2"
    Case 3
    arq_det = "       2oo2"
    Case 4
    arq_det = "       2oo3"
    Case 5
    arq_det = "      1oo1D"
    Case 6
    arq_det = "      2oo2D"
    Case 7
    arq_det = "      1oo2D"
    Case Else
    arq_det = "           "
End Select
End Function
'-------------------------------------------------------------
Private Function det_device(indvi, tipo_device) As String
Dim device, i

If indvi = 0 Then
    det_device = "   ---   "
Else
Set device = db.Openrecordset(tipo_device)
i = 1
Do While i < indvi
    device.GetRows
    i = i + 1
Loop
det_device = device!Identificador
End If
End Function
'-------------------------------------------------------------
Public Sub Eliminar_p()
Dim del
Set del = db.Openrecordset("SELECT * FROM poblacion;")
With del
    .Delete
End With
End Sub
'-------------------------------------------------------------
Private Sub mutacion_caso1(pob() As tIndvi, j)
Dim i, mutante As Byte, gen As Byte, bit As Byte, bitt As Byte, a As Byte
For i = 1 To j
mutante = Int((Rnd * n_indv) + 1)   'asigna valor de 1 a n_indv
gen = Int((Rnd * 6) + 1)            'asigna valor de 1 a 6
```

```
Select Case gen
    Case 1
        bit = Int((Rnd * nb_sen) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor Xor bitt
        If a = 0 Or a > n_sen Then
            i = i - 1
        Else: pob(mutante).tsensor = a
        End If
    Case 2
        bit = Int((Rnd * nb_log) + 1) 'asigna valor de 1 a nb_logics
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tlogics Xor bitt
        If a = 0 Or a > n_log Then
            i = i - 1
        Else: pob(mutante).tlogics = a
        End If
    Case 3
        bit = Int((Rnd * nb_val) + 1)       'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula Xor bitt
        If a = 0 Or a > n_val Then
            i = i - 1
        Else: pob(mutante).tvalvula = a
        End If
    Case 4
        bit = Int((Rnd * 3) + 1)               'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).asensor Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).asensor = a
        End If
    Case 5
        bit = Int((Rnd * 3) + 1)               'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).alogics Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).alogics = a
        End If
    Case 6
        bit = Int((Rnd * 3) + 1)               'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).avalvula Xor bitt
        If a = 0 Or a > 7 Then
            i = i - 1
        Else: pob(mutante).avalvula = a
        End If
End Select
Next
End Sub
```

```
'-------------------------------------------------------------
Private Sub mutacion_caso2(pob() As tIndvi, j)
Dim i, mutante As Byte, gen As Byte, bit As Byte, bitt As Byte, a As Byte
For i = 1 To j
mutante = Int((Rnd * n_indv) + 1)               'asigna valor de 1 a n_indv
gen = Int((Rnd * 7) + 1)                        'asigna valor de 1 a 7
Select Case gen
    Case 1
        bit = Int((Rnd * nb_sen) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor Xor bitt
        If a = 0 Or a > n_sen Then
            i = i - 1
        Else: pob(mutante).tsensor = a
        End If
    Case 2
        bit = Int((Rnd * nb_log) + 1)           'asigna valor de 1 a nb_logics
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tlogics Xor bitt
        If a = 0 Or a > n_log Then
            i = i - 1
        Else: pob(mutante).tlogics = a
        End If
    Case 3
        bit = Int((Rnd * nb_val) + 1)           'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula Xor bitt
        If a = 0 Or a > n_val Then
            i = i - 1
        Else: pob(mutante).tvalvula = a
        End If
    Case 4
        bit = Int((Rnd * nb_val2) + 1)          'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula2 Xor bitt
        If a = 0 Or a > n_val2 Then
            i = i - 1
        Else: pob(mutante).tvalvula2 = a
        End If
    Case 5
        bit = Int((Rnd * 3) + 1)                'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).asensor Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).asensor = a
        End If
    Case 6
        bit = Int((Rnd * 3) + 1)                'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).alogics Xor bitt
        If a = 0 Then
```

```
            i = i - 1
        Else: pob(mutante).alogics = a
        End If
    Case 7
        bit = Int((Rnd * 3) + 1)                'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).avalvula Xor bitt
        If a = 0 Or a > 7 Then
            i = i - 1
        Else: pob(mutante).avalvula = a
        End If
End Select
Next
End Sub
'-------------------------------------------------------------
Private Sub mutacion_caso3(pob() As tIndvi, j)
Dim i, mutante As Byte, gen As Byte, bit As Byte, bitt As Byte, a As Byte
For i = 1 To j
mutante = Int((Rnd * n_indv) + 1)               'asigna valor de 1 a n_indv
gen = Int((Rnd * 7) + 1)                        'asigna valor de 1 a 7
Select Case gen
    Case 1
        bit = Int((Rnd * nb_sen) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor Xor bitt
        If a = 0 Or a > n_sen Then
            i = i - 1
        Else: pob(mutante).tsensor = a
        End If
    Case 2
        bit = Int((Rnd * nb_sen2) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor2 Xor bitt
        If a = 0 Or a > n_sen2 Then
            i = i - 1
        Else: pob(mutante).tsensor2 = a
        End If
    Case 3
        bit = Int((Rnd * nb_log) + 1)       'asigna valor de 1 a nb_logics
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tlogics Xor bitt
        If a = 0 Or a > n_log Then
            i = i - 1
        Else: pob(mutante).tlogics = a
        End If
    Case 4
        bit = Int((Rnd * nb_val) + 1)       'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula Xor bitt
        If a = 0 Or a > n_val Then
            i = i - 1
        Else: pob(mutante).tvalvula = a
```

```
        End If
    Case 5
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).asensor Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).asensor = a
        End If
    Case 6
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).alogics Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).alogics = a
        End If
    Case 7
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).avalvula Xor bitt
        If a = 0 Or a > 7 Then
            i = i - 1
        Else: pob(mutante).avalvula = a
        End If
End Select
Next
End Sub
'-------------------------------------------------------------
Private Sub mutacion_caso4(pob() As tIndvi, j)
Dim i, mutante As Byte, gen As Byte, bit As Byte, bitt As Byte, a As Byte
For i = 1 To j
mutante = Int((Rnd * n_indv) + 1)           'asigna valor de 1 a n_indv
gen = Int((Rnd * 8) + 1)                    'asigna valor de 1 a 8
Select Case gen
    Case 1
        bit = Int((Rnd * nb_sen) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor Xor bitt
        If a = 0 Or a > n_sen Then
            i = i - 1
        Else: pob(mutante).tsensor = a
        End If
    Case 2
        bit = Int((Rnd * nb_sen2) + 1)
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tsensor2 Xor bitt
        If a = 0 Or a > n_sen2 Then
            i = i - 1
        Else: pob(mutante).tsensor2 = a
        End If
    Case 3
```

```
        bit = Int((Rnd * nb_log) + 1)       'asigna valor de 1 a nb_logics
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tlogics Xor bitt
        If a = 0 Or a > n_log Then
            i = i - 1
        Else: pob(mutante).tlogics = a
        End If
    Case 4
        bit = Int((Rnd * nb_val) + 1)       'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula Xor bitt
        If a = 0 Or a > n_val Then
            i = i - 1
        Else: pob(mutante).tvalvula = a
        End If
    Case 5
        bit = Int((Rnd * nb_val2) + 1)      'asigna valor de 1 a nb_valvula
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).tvalvula2 Xor bitt
        If a = 0 Or a > n_val2 Then
            i = i - 1
        Else: pob(mutante).tvalvula2 = a
        End If
    Case 6
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).asensor Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).asensor = a
        End If
    Case 7
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).alogics Xor bitt
        If a = 0 Then
            i = i - 1
        Else: pob(mutante).alogics = a
        End If
    Case 8
        bit = Int((Rnd * 3) + 1)            'asigna valor de 1 a 3
        bitt = 2 ^ (bit - 1)
        a = pob(mutante).avalvula Xor bitt
        If a = 0 Or a > 7 Then
            i = i - 1
        Else: pob(mutante).avalvula = a
        End If
End Select
Next
End Sub
'-------------------------------------------------------------
```

2. Módulo de verificación

```
Option Compare Database                     'encabezado del programa
Option Explicit
Type tDevice
la As Single: las As Single: lad As Single: ladd As Single: ladu As Single
lasd As Single: lasu As Single: laddn As Single: laddc As Single: ladun As Single
laduc As Single: lasdn As Single: lasdc As Single: lasun As Single: lasuc As Single
MTTR  As Single: Ti As Single: SD As Single: sff As Single: type As String
End Type
Type tSIS
p As Single: r As Single: arq_s As Byte: arq_l As Byte: arq_v As Byte
pf As Single: mt As Single
End Type
Const n_indv = 7 * 7 * 7
Dim db, sil_tg, caso As Byte, psen, plog, pval
Public Sub calculo(lazodes, jota, p_error As Boolean, caso_tr As Byte, graf As
Boolean)
Dim senso As tDevice, senso2 As tDevice, valvul As tDevice, valvul2 As tDevice,
logsol As tDevice
Dim ind, j, i, k As Byte, dis, Lazo, truc, graf_
Dim cantd_sen, cantd_log, cantd_val, cantd_sen2, cantd_val2
Dim arq_se As Byte, arq_lo As Byte, arq_va As Byte
Dim tiposen As String, sql As String, comi As String
Dim tipolog As String, tipoval As String, lazdef As String
Dim tiposen2 As String, tipoval2 As String
Dim p_sensor(1 To 7)
Dim p_sensor2(1 To 7)
Dim p_logsol(1 To 7)
Dim p_valvul(1 To 7)
Dim p_valvul2(1 To 7)
Dim sis(1 To n_indv) As tSIS
Dim elite(1 To n_indv) As tSIS
Dim RRF, delta, target, hft_s, hft_s2, hft_l, hft_v, hft_v2
Set db = CurrentDb()
Set dis = db.Openrecordset("Diseño")
Set truc = db.Openrecordset("truco")

p_error = False
comi = truc!Comilla          'se emplea para poner las comillas
lazdef = comi & lazodes & comi
sql = "SELECT * FROM Lazos WHERE Lazo = " & lazdef
Set Lazo = db.Openrecordset(sql)
On Error GoTo errores
sil_tg = Lazo!SIL

Select Case sil_tg
    Case "          1"
    target = 100
    Case "          2"
```

```
    target = 1000
    Case "          3"
    target = 10000
    Case Else
    target = 1
End Select

arq_se = det_arq(Lazo![Arq_sens])
arq_lo = det_arq(Lazo![Arq_logi])
arq_va = det_arq(Lazo![Arq_válv])
graf = False
If arq_se <> 0 And arq_lo <> 0 And arq_va <> 0 Then
    graf = True
Else
    If target = 1 Then
        MsgBox "Unicamente se puede verificar el SIL de una SIF" & _
               " si la arquitectura de todos" & Chr(13) & _
               " sus elementos (sensor, logic solver y válvula)" & _
               " han sido especificados.", vbInformation, "SIS: Diseño y
verificación"
        p_error = True
        Exit Sub
    End If
End If

tiposen = comi & Lazo!Sensor_ & comi
cantd_sen = Lazo!cantidad_s
tiposen2 = comi & Lazo![Sensor_2] & comi
cantd_sen2 = Lazo!cantidad_s2
tipolog = comi & Lazo!Logicsolver_ & comi
cantd_log = Lazo!cantidad_l
tipoval = comi & Lazo!Valvula_ & comi
cantd_val = Lazo!cantidad_v
tipoval2 = comi & Lazo!Valvula_2 & comi
cantd_val2 = Lazo!cantidad_v2

If tiposen2 = """""" Then          'determina en que caso se esta
    If tipoval2 = """""" Then      'trabajando a saber:
        caso = 1               ' caso 1: 1 sensor y 1 válvula
    Else                       ' caso 2: 1 sensor y 2 válvulas
        caso = 2
    End If
Else
    If tipoval2 = """""" Then
        caso = 3               'caso 3: 2 sensores y 1 válvula
    Else                       'caso 4: 2 sensores y 2 válvulas
        caso = 4
    End If
End If
caso_tr = caso
calc tiposen, cantd_sen, tiposen2, cantd_sen2, _
     tipolog, cantd_log, tipoval, cantd_val, tipoval2, cantd_val2, _
```

```
     senso, senso2, valvul, valvul2, logsol 'calcula las tasas de falla

hft_s = hft(senso.type, senso.sff)          'determina el hft minimo
det_pfd senso, hft_s, p_sensor, arq_se, "sensor/transmisor", p_error 'determina pfd
If p_error Then
GoTo salida
End If

If caso = 3 Or caso = 4 Then
    hft_s2 = hft(senso2.type, senso2.sff)
    det_pfd senso2, hft_s2, p_sensor2, arq_se, "2° sensor/transmisor", p_error
    If p_error Then
        GoTo salida
    End If
Else                        'se coloca pfd del sensor2 a cero
    For i = 1 To 7          'para que no contribuya a la pfd del lazo
        p_sensor2(i) = 0
    Next
End If

hft_l = hft(logsol.type, logsol.sff)
det_pfd logsol, hft_l, p_logsol, arq_lo, "logic solver", p_error
If p_error Then
    GoTo salida
End If

hft_v = hft(valvul.type, valvul.sff)
det_pfd valvul, hft_v, p_valvul, arq_va, "elemento final de control", p_error
If p_error Then
    GoTo salida
End If

If caso = 2 Or caso = 4 Then
    hft_v2 = hft(valvul2.type, valvul2.sff)
    det_pfd valvul2, hft_v2, p_valvul2, arq_va, "2° elemento final de control",
p_error
    If p_error Then
        GoTo salida
    End If
Else
    For k = 1 To 7
        p_valvul2(k) = 0
    Next
End If

ind = 0
For i = 1 To 7
For j = 1 To 7
For k = 1 To 7
ind = ind + 1
sis(ind).p = cantd_sen * p_sensor(i) + cantd_sen2 * p_sensor2(i) + p_logsol(j) +
cantd_val * p_valvul(k) + cantd_val2 * p_valvul2(k)
```

```
sis(ind).r = 1 / sis(ind).p
sis(ind).arq_s = i
sis(ind).arq_l = j
sis(ind).arq_v = k
Next
Next
Next

If graf Then
    Set graf_ = db.Openrecordset("graf")
    With graf_
    .AddNew
    !elemento = "sensor/transmisor"
    ![PFD] = cantd_sen * p_sensor(arq_se) + cantd_sen2 * p_sensor2(arq_se)
    .Update
    .AddNew
    !elemento = "logic solver"
    ![PFD] = p_logsol(arq_lo)
    .Update
    .AddNew
    !elemento = "final de control"
    ![PFD] = cantd_val * p_valvul(arq_va) + cantd_val2 * p_valvul2(arq_va)
    .Update
    End With
End If

If target = 1 Or graf Then      'si target = 1 se está en un
burbuja sis, n_indv             'proceso de verificación
Agregar_s dis, sis, 1, senso, senso2, logsol, valvul, valvul2, Lazo
j = 1
Else
RRF = 0.55 * target             'es el RRF requerido
delta = 0.45 * target           'es el delta para la determinación

j = 0

For i = 1 To n_indv
If sis(i).r <= RRF + delta And sis(i).r >= RRF - delta Then
j = j + 1
elite(j) = sis(i)
End If
Next

If j = 0 Then
    p_error = True
    For i = 1 To n_indv
        If sis(i).r > target Then
            j = j + 1
        End If
    Next
    If j = 0 Then
        MsgBox "Con los elementos seleccionados no" & Chr(13) & _
```

```
              "se puede alcanzar el SIL requerido!", vbCritical, "SIS: Diseño y
verificación"
    Else
        MsgBox "Con los elementos seleccionados se" & Chr(13) & _
               "sobrepasa el SIL requerido!", vbCritical, "SIS: Diseño y
verificación"
    End If
salida:
        MsgBox "Intente uno de los siguientes:" & Chr(13) & Chr(13) & _
               "1.- Cambie el SIL requerido" & Chr(13) & _
               "2.- Cambie los equipos empleados" & Chr(13) & _
               "3.- Cambie la arquitectura de los equipos", vbInformation, "SIS:
Diseño y verificación"
Else
    burbuja elite, j
    If j > 15 Then
        j = 15
    End If
    Agregar_s dis, elite, j, senso, senso2, logsol, valvul, valvul2, Lazo
End If
End If

jota = j    'es el numero de registros en la tabla Diseño
Exit Sub
errores:
MsgBox "Debe elegir una SIF de la lista!", vbCritical, "SIS: Diseño y verificación"
p_error = True
End Sub
'Este procedimiento calcula todas las tasas necesarias para
'calcular las PFD, PFS, MTTFs
Private Sub CalcTasas(ladu, ladd, lasu, lasd, cantseg, cantpel, B, _
device As tDevice)

device.la = ladu + ladd + lasu + lasd   '1 / (MTBF * 8760)
device.las = (lasu + lasd) * cantseg    'frs * device.la * cantseg
device.lad = (ladu + ladd) * cantpel    'frd * device.la * cantpel
device.ladd = ladd * cantpel            'c * device.lad
device.ladu = ladu * cantpel            '(1 - c) * device.lad
device.lasd = lasd * cantseg            'c * device.las
device.lasu = lasu * cantseg            '(1 - c) * device.las
device.laddn = (1 - B) * device.ladd
device.laddc = B * device.ladd
device.ladun = (1 - B) * device.ladu
device.laduc = B * device.ladu
device.lasdn = (1 - B) * device.lasd
device.lasdc = B * device.lasd
device.lasun = (1 - B) * device.lasu
device.lasuc = B * device.lasu
device.sff = (1 - (device.ladu / (device.lad + device.las))) * 100
End Sub
'determina el hardware failure tolerance minimo
Private Function hft(Tipo As String, sff As Single)
```

```
Select Case Tipo
    Case "A"
        Select Case sff
            Case 0 To 60
                Select Case sil_tg
                    Case "          1"
                    hft = 0
                    Case "          2"
                    hft = 1
                    Case "          3"
                    hft = 2
                End Select
            Case 60 To 90
                Select Case sil_tg
                    Case "          1"
                    hft = 0
                    Case "          2"
                    hft = 0
                    Case "          3"
                    hft = 1
                End Select
            Case 90 To 100
                hft = 0
        End Select
    Case "B"
        Select Case sff
            Case 0 To 60
                Select Case sil_tg
                    Case "          1"
                    hft = 1
                    Case "          2"
                    hft = 2
                    Case "          3"
                    hft = 3
                End Select
            Case 60 To 90
                Select Case sil_tg
                    Case "          1"
                    hft = 0
                    Case "          2"
                    hft = 1
                    Case "          3"
                    hft = 2
                End Select
            Case 90 To 99
                Select Case sil_tg
                    Case "          1"
                    hft = 0
                    Case "          2"
                    hft = 0
                    Case "          3"
                    hft = 1
```

```
                End Select
            Case 99 To 100
                hft = 0
        End Select
End Select
End Function
Private Sub det_pfd(device As tDevice, hftm, p_device(), arq_device, name_device As
String, flag)
Dim j
Select Case hftm
    Case 0
        If arq_device = 0 Then          'arq_device = 0 significa que debe ser
determinada
            For j = 1 To 7              'calcula las PFD de acuerdo
                p_device(j) = PFD(j, device) 'a la arq del sensor
            Next
        Else
            For j = 1 To 7
                p_device(j) = 1
            Next
            p_device(arq_device) = PFD(arq_device, device)
        End If
    Case 1
        If arq_device = 0 Then          'arq_device = 0 significa que debe ser
determinada
            For j = 1 To 7              'calcula las PFD de acuerdo
                p_device(j) = PFD(j, device) 'a la arq del sensor
            Next
            p_device(1) = 1
            p_device(5) = 1
        Else
            If arq_device = 1 Or arq_device = 5 Then
                GoTo mensaje
            Else
                For j = 1 To 7
                    p_device(j) = 1
                Next
                p_device(arq_device) = PFD(arq_device, device)
            End If
        End If
    Case 2
        If arq_device = 0 Or arq_device = 4 Then          'arq_device = 0 significa
que debe ser determinada
            For j = 1 To 7
                p_device(j) = 1
            Next
            p_device(4) = PFD(4, device)               '2oo3
        Else
mensaje:
flag = True
MsgBox "La arquitectura seleccionada para el " & name_device & " escogido no cumple"
& Chr(13) & _
```

```
       " con los requerimientos de hardware failure tolerance del lazo" & Chr(13) & _
       "", vbCritical, "SIS: Diseño y verificación"
        End If
    Case 3
    flag = True
MsgBox "El " & name_device & " escogido no puede ser empleado en lazos " & _
       "con nivel SIL 3, debido a los requerimientos de hardware failure tolerance. "
& Chr(13) & _
       "", vbCritical, "SIS: Diseño y verificación"
End Select
End Sub
'Calcula PFD
Private Function PFD(arq, device As tDevice) As Single

Select Case arq
    Case 0
    PFD = 1000
    Case 1  '1oo1
    PFD = device.ladd * device.MTTR + (device.ladu * device.Ti) / 2
    Case 2  '1oo2
    PFD = (device.laduc * device.Ti) / 2 + device.laddc * device.MTTR + (device.laddn
* device.MTTR) ^ 2 + ((device.laddn * device.MTTR * device.ladun * device.Ti) ^ 2) /
2 + ((device.ladun * device.Ti) ^ 2) / 3
    Case 3  '2oo2
    PFD = device.laddc * device.MTTR + (device.laduc * device.Ti) / 2 + 2 *
device.laddn * device.MTTR + device.ladun * device.Ti
    Case 4  '2oo3
    PFD = ((3 / 2) * device.laduc * device.Ti + 3 * device.laddc * device.MTTR + 3 *
(device.laddn * device.MTTR) ^ 2 + (3 / 2) * device.laddn * device.MTTR *
device.ladun * device.Ti + (device.ladun * device.Ti) ^ 2)
    Case 5  '1oo1D
    PFD = device.ladu * device.Ti / 2
    Case 6  '2oo2D
    PFD = device.laduc * device.Ti / 2 + device.ladun * device.Ti
    Case 7  '1oo2D
    PFD = device.laduc * device.Ti / 2 + ((device.ladun * device.Ti) ^ 2) / 3
End Select
End Function
'Calcula PFS
Private Function PFS(arq, device As tDevice) As Single

Select Case arq
    Case 1  '1oo1
    PFS = device.lasd * device.SD + device.lasu * device.SD
    Case 2  '1oo2
    PFS = (device.lasdc + device.lasuc + 2 * device.lasdn + 2 * device.lasun) *
device.SD
    Case 3  '2oo2
    PFS = device.lasdc * device.SD + device.lasuc * device.SD + 2 * (device.lasdn *
device.MTTR * device.lasdn * device.SD + device.lasun * device.Ti * device.lasdn *
device.SD + device.lasdn * device.MTTR * device.lasun * device.SD + device.lasun *
device.Ti * device.lasun * device.SD)
```

```
    Case 4  '2oo3
    PFS = 3 * device.lasuc * device.SD + 3 * device.lasdc * device.SD + 6 *
(device.lasdn * device.MTTR * device.lasdn * device.SD + device.lasun * device.Ti *
device.lasdn * device.SD + device.lasdn * device.MTTR * device.lasun * device.SD +
device.lasun * device.Ti * device.lasun * device.SD)
    Case 5  '1oo1D
    PFS = (device.lasd + device.lasu + device.ladd) * device.SD
    Case 6  '2oo2D
    PFS = (device.lasuc + device.lasdc + device.laddc) * device.SD + 2 *
((device.lasdn + device.laddn) * device.MTTR * (device.lasdn + device.laddn) *
device.SD + device.lasun * device.Ti * (device.lasdn + device.laddn) * device.SD +
(device.lasdn + device.laddn) * device.MTTR * device.lasun * device.SD + device.lasun
* device.Ti * device.lasun * device.SD)
    Case 7  '1oo2D
    PFS = (device.lasdc + device.lasuc + device.laddc) * device.SD + (device.lasdn *
device.MTTR + device.laddn * device.MTTR) ^ 2 + (device.lasun * device.Ti) ^ 2

End Select
End Function
'Calcula MTTFs
Private Function MTTFs(arq, device As tDevice) As Single
If device.las = 0 Then
MTTFs = 1000
Exit Function
End If
Select Case arq
    Case 1  '1oo1
    MTTFs = 1 / ((device.lasd + device.lasu) * device.Ti)
    Case 2  '1oo2
    MTTFs = 1 / ((device.lasdc + device.lasuc + 2 * device.lasdn + 2 * device.lasun)
* device.Ti)
    Case 3  '2oo2
    MTTFs = 1 / ((device.lasdc + device.lasuc + 2 * (device.lasdn * device.MTTR *
device.lasdn + device.lasun * device.Ti * device.lasdn + device.lasdn * device.MTTR *
device.lasun + device.lasun * device.Ti * device.lasun)) * device.Ti)
    Case 4  '2oo3
    MTTFs = 1 / ((3 * device.lasuc + 3 * device.lasdc + 6 * (device.lasdn *
device.MTTR * device.lasdn + device.lasun * device.Ti * device.lasdn + device.lasdn *
device.MTTR * device.lasun + device.lasun * device.Ti * device.lasun)) * device.Ti)
    Case 5  '1oo1D
    MTTFs = 1 / ((device.lasd + device.lasu + device.ladd) * device.Ti)
    Case 6  '2oo2D
    MTTFs = 1 / (((device.lasuc + device.lasdc + device.laddc) + 2 * ((device.lasdn +
device.laddn) * device.MTTR * (device.lasdn + device.laddn) + device.lasun *
device.Ti * (device.lasdn + device.laddn) + (device.lasdn + device.laddn) *
device.MTTR * device.lasun + device.lasun * device.Ti * device.lasun)) * device.Ti)
    Case 7  '1oo2D
    MTTFs = 1 / (((device.lasdc + device.lasuc + device.laddc) + (device.lasdn *
device.MTTR + device.laddn * device.MTTR) ^ 2 + (device.lasun * device.Ti) ^ 2) *
device.Ti)
 End Select
End Function
```

```
Private Sub calc(sent As String, cantd_sen, sent2 As String, cantd_sen2, logt As
String, _
cantd_log, valt As String, cantd_val, valt2 As String, cantd_val2, senso As tDevice,
_
senso2 As tDevice, valvul As tDevice, valvul2 As tDevice, logsol As tDevice)

Dim cpu As tDevice, diin As tDevice, diout As tDevice, anin As tDevice
Dim sens, valv, sens2, valv2, logcs
Dim ladu, ladd, lasu, lasd, cantseg, cantpel, B
Dim sql As String


sql = "SELECT * FROM Sensores WHERE Identificador = " & sent
Set sens = db.Openrecordset(sql)
With sens
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
senso.MTTR = ![MTTR]
senso.Ti = ![Ti (tiempo de inspección)]
senso.SD = ![SD (tiempo de parada)]
senso.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, senso

If caso = 3 Or caso = 4 Then
sql = "SELECT * FROM Sensores WHERE Identificador = " & sent2
Set sens2 = db.Openrecordset(sql)
With sens2
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1
cantpel = 1
B = ![B (Factor de causa común)] / 100  'esta en %
senso2.MTTR = ![MTTR]
senso2.Ti = ![Ti (tiempo de inspección)]
senso2.SD = ![SD (tiempo de parada)]
senso2.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, senso2
End If

sql = "SELECT * FROM Valvulas WHERE Identificador = " & valt
Set valv = db.Openrecordset(sql)
With valv
ladu = ![ladu]
```

```
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1 'cantd_val
cantpel = 1 'cantd_val
B = ![B (Factor de causa común)] / 100  'esta en %
valvul.MTTR = ![MTTR]
valvul.Ti = ![Ti (tiempo de inspección)]
valvul.SD = ![SD (tiempo de parada)]
valvul.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, valvul

If caso = 2 Or caso = 4 Then
sql = "SELECT * FROM Valvulas WHERE Identificador = " & valt2
Set valv2 = db.Openrecordset(sql)
With valv2
ladu = ![ladu]
ladd = ![ladd]
lasu = ![lasu]
lasd = ![lasd]
cantseg = 1 'cantd_val2
cantpel = 1 'cantd_val2
B = ![B (Factor de causa común)] / 100  'esta en %
valvul2.MTTR = ![MTTR]
valvul2.Ti = ![Ti (tiempo de inspección)]
valvul2.SD = ![SD (tiempo de parada)]
valvul2.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, valvul2
End If

sql = "SELECT * FROM Logicsolvers WHERE Identificador = " & logt
Set logcs = db.Openrecordset(sql)
With logcs
ladu = ![ladu_CPU]
ladd = ![ladd_CPU]
lasu = ![lasu_CPU]
lasd = ![lasd_CPU]
cantseg = 1 '![Cant_CPU]
cantpel = 1 '![Cant peligrosos CPU]
B = ![B  (factor causa comun)] / 100
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, cpu

With logcs
ladu = ![ladu_DI]
ladd = ![ladd_DI]
lasu = ![lasu_DI]
lasd = ![lasd_DI]
cantseg = 1 '![Cant_DI]
cantpel = 1 '![Cant peligrosos DI]
```

```
B = ![B  (factor causa comun)] / 100
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, diin

With logcs
ladu = ![ladu_DO]
ladd = ![ladd_DO]
lasu = ![lasu_DO]
lasd = ![lasd_DO]
cantseg = 1 '![Cant_DO]
cantpel = 1 '![Cant_DO]
B = ![B  (factor causa comun)] / 100
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, diout

With logcs
ladu = ![ladu_AI]
ladd = ![ladd_AI]
lasu = ![lasu_AI]
lasd = ![lasd_AI]
cantseg = 1 '![Cant_AI]
cantpel = 1 '![Cant peligrosos AI]
B = ![B  (factor causa comun)] / 100
logsol.MTTR = ![MTTR]
logsol.Ti = ![Ti (tiempo de inspección)]
logsol.SD = ![SD (tiempo de parada)]
logsol.type = ![Tipo_IEC]
End With
CalcTasas ladu, ladd, lasu, lasd, cantseg, cantpel, B, anin

logsol.la = cpu.la + diin.la + diout.la + anin.la
logsol.las = cpu.las + diin.las + diout.las + anin.las
logsol.lad = cpu.lad + diin.lad + diout.lad + anin.lad
logsol.ladd = cpu.ladd + diin.ladd + diout.ladd + anin.ladd
logsol.ladu = cpu.ladu + diin.ladu + diout.ladu + anin.ladu
logsol.lasd = cpu.lasd + diin.lasd + diout.lasd + anin.lasd
logsol.lasu = cpu.lasu + diin.lasu + diout.lasu + anin.lasu
logsol.laddn = cpu.laddn + diin.laddn + diout.laddn + anin.laddn
logsol.laddc = cpu.laddc + diin.laddc + diout.laddc + anin.laddc
logsol.ladun = cpu.ladun + diin.ladun + diout.ladun + anin.ladun
logsol.laduc = cpu.laduc + diin.laduc + diout.laduc + anin.laduc
logsol.lasdn = cpu.lasdn + diin.lasdn + diout.lasdn + anin.lasdn
logsol.lasdc = cpu.lasdc + diin.lasdc + diout.lasdc + anin.lasdc
logsol.lasun = cpu.lasun + diin.lasun + diout.lasun + anin.lasun
logsol.lasuc = cpu.lasuc + diin.lasuc + diout.lasuc + anin.lasuc
logsol.sff = (1 - (logsol.ladu / (logsol.lad + logsol.las))) * 100

End Sub
Private Function det_arq(arq As String) As Byte
Select Case arq
    Case "       1oo1"
    det_arq = 1
```

```
    Case "       1oo2"
    det_arq = 2
    Case "       2oo2"
    det_arq = 3
    Case "       2oo3"
    det_arq = 4
    Case "      1oo1D"
    det_arq = 5
    Case "      2oo2D"
    det_arq = 6
    Case "      1oo2D"
    det_arq = 7
    Case Else
    det_arq = 0
End Select

End Function
Private Function arq_det(arq As Byte) As String
Select Case arq
    Case 1
    arq_det = "       1oo1"
    Case 2
    arq_det = "       1oo2"
    Case 3
    arq_det = "       2oo2"
    Case 4
    arq_det = "       2oo3"
    Case 5
    arq_det = "      1oo1D"
    Case 6
    arq_det = "      2oo2D"
    Case 7
    arq_det = "      1oo2D"
    Case Else
    arq_det = "           "
End Select

End Function
Private Function det_sil(RRF_alc As Single) As String
Select Case RRF_alc
    Case Is < 10
    det_sil = "no alcanza SIL 1"
    Case 10 To 100
    det_sil = "           1"
    Case 100 To 1000
    det_sil = "           2"
    Case 1000 To 10000
    det_sil = "           3"
    Case Else
    det_sil = "supera SIL 3"
End Select
```

```
End Function

Private Sub Agregar_s(rstTemp, pobl() As tSIS, n, senso As tDevice, senso2 As tDevice
_
                           , logsol As tDevice, valvul As tDevice, valvul2 As
tDevice, Lazo)
Dim j, PFSc, MTTFsc
For j = 1 To n
Select Case caso
    Case 1
        PFSc = PFS(pobl(j).arq_s, senso) + PFS(pobl(j).arq_l, logsol) +
PFS(pobl(j).arq_v, valvul)
        MTTFsc = 1 / ((1 / MTTFs(pobl(j).arq_s, senso)) + (1 / MTTFs(pobl(j).arq_l,
logsol)) + (1 / MTTFs(pobl(j).arq_v, valvul)))
    Case 2
         PFSc = PFS(pobl(j).arq_s, senso) + PFS(pobl(j).arq_l, logsol) +
PFS(pobl(j).arq_v, valvul) + PFS(pobl(j).arq_v, valvul2)
         MTTFsc = 1 / ((1 / MTTFs(pobl(j).arq_s, senso)) + (1 / MTTFs(pobl(j).arq_l,
logsol)) + (1 / MTTFs(pobl(j).arq_v, valvul)) + (1 / MTTFs(pobl(j).arq_v, valvul2)))
    Case 3
         PFSc = PFS(pobl(j).arq_s, senso) + PFS(pobl(j).arq_s, senso2) +
PFS(pobl(j).arq_l, logsol) + PFS(pobl(j).arq_v, valvul)
         MTTFsc = 1 / ((1 / MTTFs(pobl(j).arq_s, senso)) + (1 / MTTFs(pobl(j).arq_s,
senso2)) + (1 / MTTFs(pobl(j).arq_l, logsol)) + (1 / MTTFs(pobl(j).arq_v, valvul)))
    Case 4
         PFSc = PFS(pobl(j).arq_s, senso) + PFS(pobl(j).arq_s, senso2) +
PFS(pobl(j).arq_l, logsol) + PFS(pobl(j).arq_v, valvul) + PFS(pobl(j).arq_v, valvul2)
         MTTFsc = 1 / ((1 / MTTFs(pobl(j).arq_s, senso)) + (1 / MTTFs(pobl(j).arq_s,
senso2)) + (1 / MTTFs(pobl(j).arq_l, logsol)) + (1 / MTTFs(pobl(j).arq_v, valvul)) +
(1 / MTTFs(pobl(j).arq_v, valvul2)))
End Select
With rstTemp
    .AddNew
    !PFD = pobl(j).p
    !RRF = pobl(j).r
    !SIL = det_sil(pobl(j).r)
    !sensor = arq_det(pobl(j).arq_s)
    !logicsolver = arq_det(pobl(j).arq_l)
    !valvula = arq_det(pobl(j).arq_v)
    !Lazo = Lazo!Lazo
    ![sif] = Lazo![Función Instrumentada de Seguridad]
    ![Sensor_e] = Lazo!Sensor_
    ![cantd_sen] = Lazo![cantidad_s]
    ![Sensor_e2] = Lazo!Sensor_2
    ![cantd_sen2] = Lazo![cantidad_s2]
    ![Logicsolver_e] = Lazo!Logicsolver_
    ![cantd_log] = Lazo![cantidad_l]
    ![Valvula_e] = Lazo!Valvula_
    ![cantd_val] = Lazo![cantidad_v]
    ![Valvula_e2] = Lazo!Valvula_2
    ![cantd_val2] = Lazo![cantidad_v2]
    !PFS = PFSc
```

```
    !MTTFs = MTTFsc
    .Update
End With
Next
End Sub

    ' Elimina un registro del Recordset utilizando
Public Sub Eliminar()
Dim del
Set del = db.Openrecordset("SELECT * FROM Diseño;")
With del
    .Delete
End With
End Sub
Public Sub Eliminar_graf()
Dim del
Set del = db.Openrecordset("SELECT * FROM graf;")
With del
    .Delete
End With
End Sub
Private Sub burbuja(pobl() As tSIS, n)
Dim temp As tSIS
Dim i#, j#, flag As Byte
Do
For i = 1 To n
    flag = 0
    For j = 1 To n - i
        If pobl(j + 1).r > pobl(j).r Then
        temp = pobl(j)
        pobl(j) = pobl(j + 1)
        pobl(j + 1) = temp
        flag = 1
        End If
    Next
Next
Loop While flag = 1
End Sub
```