# Anexos

**A. Programación Modular**

**Variables conection y recordset**

Public conn As New ADODB.Connection

Public rst As New ADODB.Recordset

**Procedimiento para Abrir la Conexión de la BD**

Public Sub Abrir\_coneccion()

conn.Open "DSN=DSN\_ICE;uid=sa;pwd="

End Sub

**Procedimiento para Cerrar la Conexión de la BD**

Public Sub Cerrar\_Coneccion() ‘ Permite Cerrar la conexión conn.Close

Set conn = Nothing

End Sub

**Procedimiento para Llenar una Lista**

Public Sub LLenar(ByVal L1 As Lista, ByRef L2 As Lista)

Dim el As New ElementoLista

Set el = L1.First

While (el.Target <> 0)

L2.Add el.FechaDespacho, el.corrugadora, el.NumeroOrden, el.Cantidad, el.Test, \_

el.Flauta, el.Metros, el.LagoP, el.AnchoP, el.OrdenA, \_

el.FechaIActual, el.FechaFActual, el.OrdenP, \_

el.FechaIPropuesta, el.FechaFPropuesta, el.Tiempo

Set el = el.After

Wend

End Sub

**Procedimiento del Recocido Simulado**

Public Sub simulated\_annealing(ByVal Tinicio As Double, ByVal alfa As Double, ByVal iteracciones As Double, ByVal Tfinal As Double, ByVal l\_actual As Lista, ByVal capacidad As Integer, fecha As Date, hora As Date)

Dim T As Double

Dim delta As Double

Dim x As Long

Dim y As Long

Dim max As Long

Dim l\_candidata As New Lista

Dim boltzmann As Double

'Dim ePointer As New ElementoLista

max = l\_actual.GetLong

LLenar l\_actual, l\_candidata

T = Tinicio

frm\_programa\_orden.barra.Min = 0

frm\_programa\_orden.barra.max = iteracciones

While T >= Tfinal

For j = 1 To iteracciones

Randomize

x = CInt(Rnd \* (max - 1) + 1)

y = CInt(Rnd \* (max - 1) + 1)

l\_candidata.Permutar x, y

delta = costo\_solucion(fecha, hora, l\_candidata, capacidad) - costo\_solucion(fecha, hora, l\_actual, capacidad)

boltzmann = Exp(-(delta / T))

If (Rnd < boltzman) Or (delta < 0) Then

LLenar l\_candidata, l\_actual

End If

frm\_programa\_orden.barra.Value = j

Next j

Randomize

T = (Rnd \* (0.99 - 0.88) + 0.88) \* T

Wend

End Sub

**Procedimiento para Generar un Lista Doblemente Enlazada**

Public Sub genera\_lista(n As Integer)

Dim i As Integer

For i = n To 1 Step -1

solucion\_candidata(i, 1) = i

solucion\_candidata(i, 2) = i \* 15

solucion\_candidata(i, 3) = i \* 150

Next i

End Sub

**Function de Costo Solución**

Public Function costo\_solucion(fecha As Date, hora As Date, ByVal l As Lista, ByVal capacidad As Integer) As Double

Dim i As Integer

Dim max As Long

Dim suma As Double

Dim sol As Double

Dim sol1 As Double

Dim el1 As New ElementoLista

Dim mts As Double

Dim Test As Integer

Dim Flauta As String

Dim FechaInicio, FechaFin As Date

Dim TiempoCProg, TiempoProd, TiempoProd1 As Double

Set el1 = l.First

mts = el1.Metros

Test = el1.Test

Flauta = el1.Flauta

FechaDespacho = el1.FechaDespacho

FechaFin = hora

suma = 0

While (el1.Target <> 0)

If mts <> el1.Metros Or Test <> el1.Test Or Flauta <> el1.Flauta Then

TiempoCProg = 15

Else

TiempoCProg = 0

End If

If fecha >= FechaDespacho Then

' MsgBox "Penalidad por Fecha Despacho"

End If

FechaInicio = DateAdd("n", TiempoCProg, FechaFin)

TiempoProd1 = CDbl(el1.Cantidad) \* CDbl(el1.LagoP) / 1000

' MsgBox "No Orden = " & el1.NumeroOrden & " / Cantidad = " & CDbl(el1.Cantidad) & " / Largo = " & CDbl(el1.LagoP) / 1000

TiempoProd = TiempoProd1 / capacidad

If TiempoProd < 1 Then

seg = TiempoProd \* 60

FechaFin = DateAdd("s", seg, FechaInicio)

Else

Min = CInt(TiempoProd)

seg = TiempoProd - Min

FechaFin = DateAdd("n", Min, FechaInicio)

FechaFin = DateAdd("s", seg, FechaFin)

End If

el1.Tiempo = TiempoProd

el1.FechaIActual = FechaInicio

el1.FechaFActual = FechaFin

suma = suma + TiempoProd + TiempoCProg

mts = el1.Metros

Test = el1.Test

Flauta = el1.Flauta

Set el1 = el1.After

' MsgBox "Tiempo Cambio Programa = " & TiempoCProg & " / Fecha Incicio = " & FechaInicio & " / Tiempo Produccion = " & TiempoProd & " / Fecha Fin = " & FechaFin

Wend

costo\_solucion = suma

End Function

**Function de Factorial**

Public Function Factorial(n As Integer) As Double

Dim s As Double

Dim i As Integer

If n > 1 Then

s = 1

For i = 0 To n - 1

s = s \* (n - i)

Next i

Factorial = s

Else

If n = 0 Or n = 1 Then

Factorial = 1

End If

End If

End Function

**B. Programación Modular de Clases**

**Elementos de la Lista**

Option Explicit

Public FechaDespacho As Date

Public corrugadora As String

Public NumeroOrden As String

Public Cantidad As Long

Public Test As Long

Public Flauta As String

Public Metros As Double

Public LagoP As Double

Public AnchoP As Double

Public OrdenA As Long

Public FechaIActual As Date

Public FechaFActual As Date

Public OrdenP As Long

Public FechaIPropuesta As Date

Public FechaFPropuesta As Date

Public Tiempo As Double

**Procedimientos y Funciones de la Lista**

Option Explicit

Private Longitud As Long

Public First As New ElementoLista

Public Last As New ElementoLista

Private Pointer As ElementoLista 'TheApp

Private tPointer As New ElementoLista 'TheApp

Public Target As Long

Public Before As Object

Public After As Object

Public Target As Long

Public Before As Object

Public After As Object

**C. Creación de un Store Procedure**

CREATE proc programacion @maq int

as

select o.fechadespacho,c.nombre, o.no\_orden,o.cantidad, t.test, f.flauta,o.mts\_2,o.largo\_plancha,o.ancho\_plancha,o.indice,o.orden\_actual,

o.fechainicio\_actual, o.fechafin\_actual,o.orden\_propuesto,o.fechainicio\_propuesto, o.fechafin\_propuesto,o.tiempo

from ordenes o, tests t, flautas f, testsflautas tf,corrugadoras c

where tf.id\_flauta = f.id and tf.id\_test = t.id and tf.id = o.id\_testflauta and o.id\_corrugadora = @maq and O.estado= 'P' and o.id\_corrugadora=c.id

order by o.fechadespacho,o.fechainicio\_actual