

APÉNDICE D: SUBRUTINA PRINCIPAL

```
Sub Principal()  
  Randomize()  
  
  Dim Formulario3 As New Resultados  
  ReDim X1((Ge * mo) - 1)      'vector de soluciones modificadas  
  
  For Contador = 0 To (Ge * mo) - 1  
    ReDim X1(Contador).Pesos(Num_Activos - 1)  
  Next  
  
  ReDim Elitistas(epsilon - 1)  'vector de soluciones elitistas  
  
  For Contador = 0 To epsilon - 1  
    ReDim Elitistas(Contador).Pesos(Num_Activos - 1)  
  Next  
  
  ReDim Prodigas(phi + epsilon - 1)  'vector de soluciones pródigas  
  
  For Contador = 0 To phi + epsilon - 1  
    ReDim Prodigas(Contador).Pesos(Num_Activos - 1)  
  Next  
  
  ReDim PortInicial(Num_Activos - 1)  'pesos del portafolio inicial  
  ReDim Arreglo(1, Nu - 1)  
  ReDim Temporal.Pesos(Num_Activos - 1)  
  
  'Actualizar Barra de Progreso  
  ContadorPorcentaje = 1  
  PorcentajeEnt = (ContadorPorcentaje / (NumIteraciones * (Ge + Ge + w))) * 100  
  ActualizaProgresoAut(PorcentajeEnt)  
  
  GeneraSolucionesIniciales()  
  
  For Contador2 = 0 To Ge - 1  
    EstandarizaPesosCalculaRendimiento()  
    CalculaRiesgoCalidad()  
    CalculaPenalizaciones()  
  Next  
  
  'Generar primera lista de soluciones elitistas  
  For Contador = 0 To epsilon - 1  
    Elitistas(Contador) = X0(Contador)  
  Next
```

'REPETIR POR CADA ITERACION

For Contador = 0 To NumIteraciones - 1

For Contador2 = 0 To Ge - 1

X0(Contador2).Ordenamiento = 0

For Contador3 = 0 To mo - 1

X1(Contador2 * Contador3) = X0(Contador2)

Next

Next

For Contador2 = 0 To Ge - 1

'MODIFICACIÓN

For Contador3 = 0 To mo - 1

d = Int(Rnd() * (Nu - 1)) 'número de activos a modificar en cada cartera

For Contador4 = 1 To d

i = Int(Rnd() * (Num_Activos - 1)) 'posición del activo a modificar

z = 2 * Uo * Rnd() - Uo

Contador5 = i

Do Until X0(Contador2).Pesos(Contador5) > 0

'encuentra el activo ya incluido

If Contador5 = Num_Activos - 1 Then

Contador5 = 0

Else

Contador5 = Contador5 + 1

End If

Loop

X1(Contador2 * Contador3).Pesos(Contador5) =

Max(X0(Contador2).Pesos(Contador5) + z, 0)

If X1(Contador2 * Contador3).Pesos(Contador5) = 0 Then

p = Rnd()

If p < pr Then

'Localiza el activo j no incluido

j = 0

Do Until X0(Contador2).Pesos(j) = 0

j = j + 1

Loop

```

        X1(Contador2 * Contador3).Pesos(j) = 2 * Uo * Rnd()
    End If

End If

Next

X1(Contador2 * Contador3).Rendimiento = 0
X1(Contador2 * Contador3).Riesgo = 0
X1(Contador2 * Contador3).Suma = 0
X1(Contador2 * Contador3).ActivosIncluidos = 0

'Calcula la suma de los activos
For Contador4 = 0 To Num_Activos - 1

    If X1(Contador2 * Contador3).Pesos(Contador4) > 0 Then
        X1(Contador2 * Contador3).Suma = X1(Contador2 * Contador3).Suma +
            X1(Contador2 * Contador3).Pesos(Contador4)
    End If

Next

'Estandariza los pesos y calcula el rendimiento de la cartera
For Contador4 = 0 To Num_Activos - 1

    If X1(Contador2 * Contador3).Pesos(Contador4) > 0 Then
        X1(Contador2 * Contador3).Pesos(Contador4) = X1(Contador2 *
            Contador3).Pesos(Contador4) / X1(Contador2 * Contador3).Suma
        X1(Contador2 * Contador3).ActivosIncluidos = X1(Contador2 *
            Contador3).ActivosIncluidos + 1
    End If

    X1(Contador2 * Contador3).Rendimiento = X1(Contador2 *
        Contador3).Rendimiento + (X1(Contador2 * Contador3).Pesos(Contador4)
        * Rendimientos(Contador4))
    Parcial(Contador4) = 0
Next

'Calcula el riesgo y la calidad inicial

For Contador4 = 0 To Num_Activos - 1
    For Contador5 = 0 To Num_Activos - 1

        Parcial(Contador4) = Parcial(Contador4) + (X1(Contador2 *
            Contador3).Pesos(Contador5) * Covarianza(Contador5, Contador4))
    Next
Next

```

```
X1(Contador2 * Contador3).Riesgo = X1(Contador2 * Contador3).Riesgo +  
(Parcial(Contador4) * X1(Contador2 * Contador3).Pesos(Contador4))  
Next
```

```
X1(Contador2 * Contador3).Calidad = X1(Contador2 * Contador3).Riesgo
```

```
'Calcula la calidad final
```

```
For Contador4 = 0 To Num_Activos - 1
```

```
'Restricción de límite inferior
```

```
If X1(Contador2 * Contador3).Pesos(Contador4) < xmin Then  
    violacion = xmin - X1(Contador2 * Contador3).Pesos(Contador4)  
    X1(Contador2 * Contador3).Calidad = X1(Contador2 *  
    Contador3).Calidad + Penalizacion(a, b, violacion)
```

```
Else
```

```
'Restricción de límite superior
```

```
If X1(Contador2 * Contador3).Pesos(Contador4) > x_max Then  
    violacion = X1(Contador2 * Contador3).Pesos(Contador4) - x_max  
    X1(Contador2 * Contador3).Calidad = X1(Contador2 *  
    Contador3).Calidad + Penalizacion(a, b, violacion)
```

```
End If
```

```
End If
```

```
Qcompra = X1(Contador2 * Contador3).Pesos(Contador4) -  
PortInicial(Contador4)
```

```
Qventa = PortInicial(Contador4) - X1(Contador2 *  
Contador3).Pesos(Contador4)
```

```
'Restricción de compra mínima
```

```
If Qcompra > 0 Then
```

```
    If Qcompra < Bmin Then
```

```
        If Qcompra < Bmin / 2 Then  
            violacion = Qcompra
```

```
        Else
```

```
            violacion = Bmin - Qcompra
```

```
        End If
```

```
    End If
```

```
Else
```

```
'Restricción de venta mínima
```

```
If Qventa < Smin Then
```

```
    If Qventa < Smin / 2 Then  
        violacion = Qventa
```

```
    Else
```

```

                violacion = Smin - Qventa
            End If
        End If

    End If

    X1(Contador2 * Contador3).Calidad = X1(Contador2 * Contador3).Calidad
    + Penalizacion(a, b, violacion)
Next

'Restricción de rendimiento esperado
If X1(Contador2 * Contador3).Rendimiento <> Rend Then
    violacion = X1(Contador2 * Contador3).Rendimiento - Rend

    If violacion < 0 Then
        violacion = -violacion
    End If

    X1(Contador2 * Contador3).Calidad = X1(Contador2 * Contador3).Calidad
    + Penalizacion(4 * a, b, violacion)
End If

'Restricción de núm. máximo de activos incluidos
If X1(Contador2 * Contador3).ActivosIncluidos > Nu Then
    violacion = X1(Contador2 * Contador3).ActivosIncluidos - Nu
    X1(Contador2 * Contador3).Calidad = X1(Contador2 * Contador3).Calidad
    + Penalizacion(a, b, violacion)

End If

Next

For Contador3 = 0 To mo - 1

    'Templado Simulado
    If X0(Contador2).Calidad > X1(Contador2 * Contador3).Calidad Then
        X0(Contador2) = X1(Contador2 * Contador3)
    Else
        p = Rnd()
        pt = Exp((X0(Contador2).Calidad - X1(Contador2 * Contador3).Calidad) /
        T)
        If p < pt Then
            X0(Contador2) = X1(Contador2 * Contador3)
        End If
    End If

Next

```

```
'Actualizar Barra de Progreso
ContadorPorcentaje = ContadorPorcentaje + 1
PorcentajeEnt = (ContadorPorcentaje / (NumIteraciones * (Ge + Ge + w))) * 100
ActualizaProgresoAut(PorcentajeEnt)
```

Next

'EVALUACIÓN Y ORDENAMIENTO

```
For Contador2 = 1 To Ge - 1
  Temporal = X0(Contador2)
  i = Contador2
```

Do While i > 1

```
  If X0(i - 1).Calidad > Temporal.Calidad Then
    X0(i) = X0(i - 1)
  Else
    GoTo SalidaWhile
  End If
```

 i = i - 1

Loop

SalidaWhile: X0(i) = Temporal

Next

```
'Factor de Ordenamiento de soluciones pródigas
For Contador2 = 0 To phi - 1
  X0(Contador2).Ordenamiento = phi - Contador2 + 1
Next
```

'Ordenamiento de lista de soluciones elitistas
ReDim Preserve Elitistas(epsilon + Ge - 1)

```
For Contador2 = 0 To Ge - 1
  Elitistas(epsilon + Contador2) = X0(Contador2)
Next
```

```
For Contador2 = 1 To epsilon + Ge - 1
  Temporal = Elitistas(Contador2)
  i = Contador2
```

Do While i > 1

```
  If Elitistas(i - 1).Calidad > Temporal.Calidad Then
    Elitistas(i) = Elitistas(i - 1)
```

```
Else
  GoTo SalidaWhile2
End If
```

```
  i = i - 1
Loop
```

SalidaWhile2: Elitistas(i) = Temporal

```
Next
```

```
ReDim Preserve Elitistas(epsilon)
```

```
'Factor de Ordenamiento de soluciones elitistas
```

```
For Contador2 = 0 To epsilon - 1
```

```
  Elitistas(Contador2).Ordenamiento = epsilon - Contador2 + 1
```

```
Next
```

```
For Contador3 = 0 To phi - 1
```

```
  For Contador4 = 0 To Num_Activos - 1
```

```
    Prodigas(Contador3).Pesos(Contador4) = X0(Contador3).Pesos(Contador4) *
```

```
    X0(Contador3).Ordenamiento
```

```
  Next
```

```
  Prodigas(Contador3).Suma = 0
```

```
  For Contador4 = 0 To Num_Activos - 1
```

```
    Prodigas(Contador3).Suma = Prodigas(Contador3).Suma +
```

```
    Prodigas(Contador3).Pesos(Contador4)
```

```
  Next
```

```
  For Contador4 = 0 To Num_Activos - 1
```

```
    If Prodigas(Contador3).Pesos(Contador4) > 0 Then
```

```
      Prodigas(Contador3).Pesos(Contador4) =
```

```
      Prodigas(Contador3).Pesos(Contador4) / Prodigas(Contador3).Suma
```

```
    End If
```

```
  Next
```

```
  Prodigas(Contador3).Ordenamiento = X0(Contador3).Ordenamiento
```

```
Next
```

```
For Contador3 = 0 To epsilon - 1
```

```
  For Contador4 = 0 To Num_Activos - 1
```

```
    Prodigas(phi + Contador3).Pesos(Contador4) =
```

```
    Elitistas(Contador3).Pesos(Contador4) * Elitistas(Contador3).Ordenamiento
Next
```

```
Prodigas(phi + Contador3).Suma = 0
```

```
For Contador4 = 0 To Num_Activos - 1
    Prodigas(phi + Contador3).Suma = Prodigas(phi + Contador3).Suma +
    Prodigas(phi + Contador3).Pesos(Contador4)
Next
```

```
For Contador4 = 0 To Num_Activos - 1
```

```
    If Prodigas(phi + Contador3).Pesos(Contador4) > 0 Then
        Prodigas(phi + Contador3).Pesos(Contador4) = Prodigas(phi +
        Contador3).Pesos(Contador4) / Prodigas(phi + Contador3).Suma
    End If
```

```
Next
```

```
Prodigas(phi + Contador3).Ordenamiento = Elitistas(Contador3).Ordenamiento
```

```
'Actualizar Barra de Progreso
```

```
ContadorPorcentaje = ContadorPorcentaje + 1
```

```
PorcentajeEnt = (ContadorPorcentaje / (NumIteraciones * (Ge + Ge + w))) * 100
```

```
ActualizaProgresoAut(PorcentajeEnt)
```

```
Next
```

```
'REEMPLAZO DE LAS SOLUCIONES MÁS POBRES
```

```
For Contador2 = (Ge - w - 1) To Ge - 1
```

```
    p = Rnd()
```

```
    If p < pc Then
```

```
        'Clonación
```

```
        total = SumaTotal(phi)
```

```
        j = Int(Rnd() * (phi - 1))
```

```
        X0(Contador2) = X0(j)
```

```
        GoTo SalidaIf
```

```
    Else
```

```
        ""Averaged Idol"
```

```
    For Contador3 = 0 To Num_Activos - 1
```

```
        X0(Contador2).Pesos(Contador3) = 0
```

```
    Next
```

```
    For Contador3 = 0 To Nu - 1
```

```
        j = Rnd()
```



```

total = SumaTotal(phi + epsilon)

For Contador4 = 0 To phi + epsilon - 1
  If j > (Prodigas(Contador4).Ordenamiento / total) Then
    i = Int(Rnd() * (Num_Activos - 1))
    Contador5 = i

    Do Until Prodigas(Contador4).Pesos(Contador5) > 0
      If Contador5 = Num_Activos - 1 Then
        Contador5 = 0
      Else
        Contador5 = Contador5 + 1
      End If
    Loop

    X0(Contador2).Pesos(Contador5) =
    Prodigas(Contador4).Pesos(Contador5)
    GoTo Salida1

  End If
Next

```

Salida1:

```

Next

EstandarizaPesosCalculaRendimiento()
CalculaRiesgoCalidad()
CalculaPenalizaciones()

```

End If

SalidaIf:

```

'Actualizar Barra de Progreso
ContadorPorcentaje = ContadorPorcentaje + 1
PorcentajeEnt = (ContadorPorcentaje / (NumIteraciones * (Ge + Ge + w))) * 100
ActualizaProgresoAut(PorcentajeEnt)

```

Next

```

Uo = Uo * amplitud
T = T * temperatura

```

Next