

## CAPÍTULO 4

### DESARROLLO DE LA APLICACIÓN

El presente capítulo da una descripción completa del algoritmo usado para resolver el problema propuesto en el Capítulo 3 de esta tesis. En la sección 4.1 se detalla la estructura de la solución así como sus principales características. El algoritmo de solución es desarrollado paso a paso en la sección 4.2, para finalmente explicar la arquitectura y el funcionamiento del software de solución al problema de optimización de carteras en la sección 4.3.

#### 4.1 Representación de la Solución del Problema

La estructura de la solución fue codificada en un vector  $x$  de  $n$  dimensiones, donde cada variable  $x_i$  representa la proporción del activo  $i$  en la cartera. Si  $x_i = 0$ , el activo  $i$  no está incluido en el portafolio. La calidad de la solución es medida por la varianza del portafolio  $x^T C x$ .

#### 4.2 Algoritmo de Solución del Problema

La serie de pasos precisos definidos por el algoritmo son representados a continuación en forma gráfica mediante diagramas de flujo para su mejor comprensión. El primer

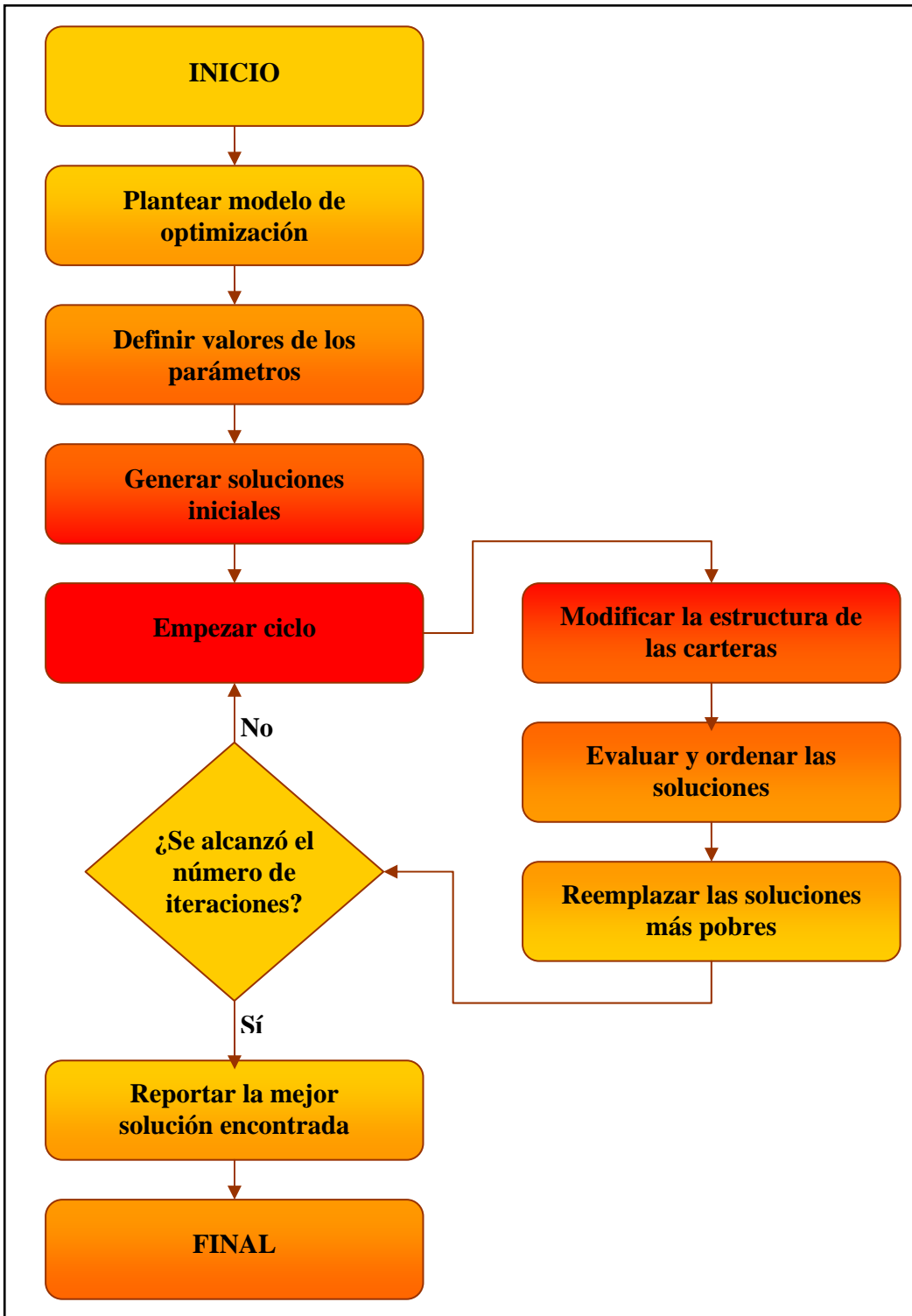
diagrama (Figura 4.1) muestra los cálculos previos necesarios para resolver el problema de optimización de una cartera de inversión, siempre y cuando se vaya a trabajar con el Modelo de Media–Varianza de Markowitz.

El diagrama 4.2 describe los pasos del algoritmo desarrollado en esta tesis, es decir, se siguen los pasos del primer diagrama, se definen los valores de los parámetros de las restricciones y de los algoritmos Evolutivos y de Templado Simulado, se generan soluciones iniciales, se fija el número de iteraciones que se quieren realizar y por cada una de ellas se efectúa un ciclo. El ciclo se realiza en tres pasos: se modifica la estructura de estas carteras, se evalúa y se ordenan todas las soluciones y se reemplazan aquellas que tengan la función objetivo más elevada. Si ya se terminaron las iteraciones deseadas, se reporta la mejor solución encontrada, es decir, la cartera que mejor cumple las condiciones establecidas previamente.



**Figura 4.1** Diagrama de Flujo de los Pasos Previos a la Solución del Algoritmo

Fuente: Elaboración Propia



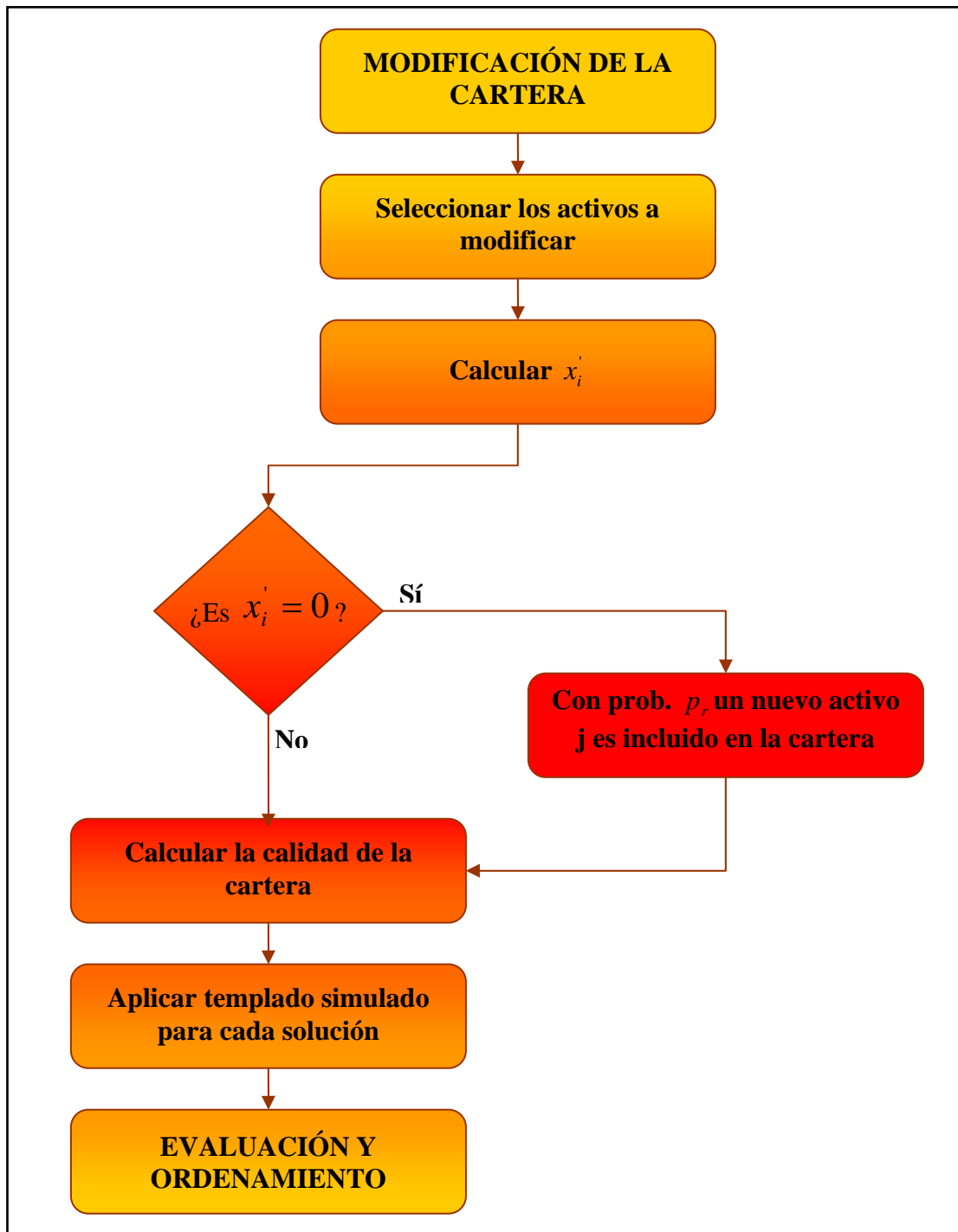
**Figura 4.2** Diagrama de Flujo del Algoritmo

Fuente: Elaboración Propia

Cada uno de los tres pasos que se realizan en cada iteración vienen desglosados en las siguientes tres figuras. La figura 4.3 describe la secuencia de la modificación de la estructura de las carteras. Primero se seleccionan los activos de cada cartera que se van a modificar y se calcula el nuevo peso del activo. Si el peso calculado es igual a 0, se incluye a la cartera un nuevo activo según cierta probabilidad, sino se continúa el proceso. Para cada solución se calcula el rendimiento, el riesgo y la función objetivo completa y finalmente se aplica el criterio de Templado Simulado para saber si la nueva solución reemplazará a la solución inicial.

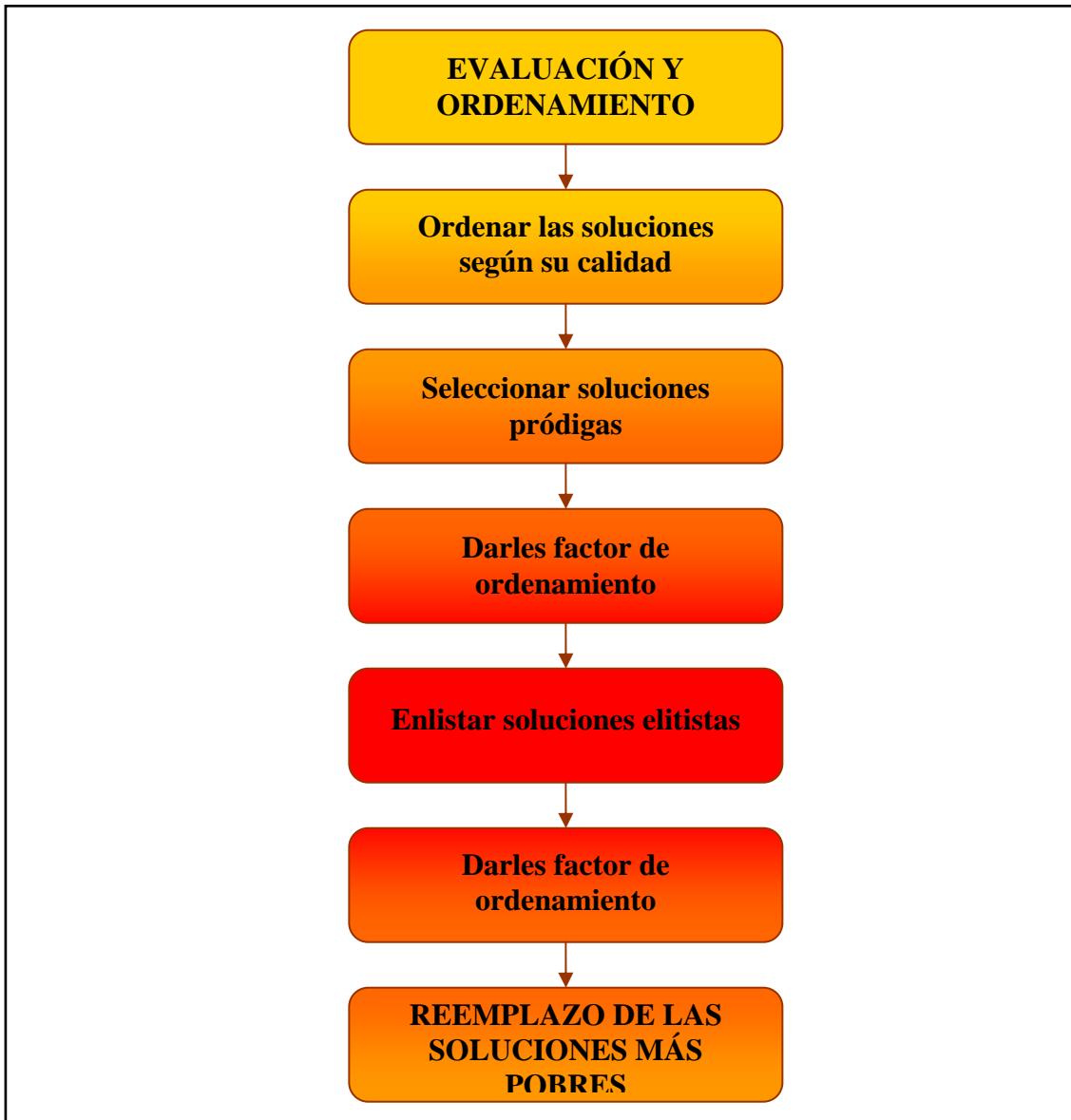
En la figura 4.4 se plantea la evaluación y el ordenamiento de las soluciones. De acuerdo al problema de minimizar el riesgo, se ordenan las carteras de menor a mayor según la función objetivo, se seleccionan las mejores soluciones como soluciones pródigas y se les da un factor de ordenamiento que se utilizará más tarde. También se enlistan las soluciones elitistas, es decir, las mejores soluciones encontradas hasta el momento, y se les da su factor de ordenamiento.

Para el reemplazo de las soluciones más pobres en calidad se deben realizar los pasos planteados en la figura 4.5. Primero se deben seleccionar las soluciones a reemplazar, es decir, las soluciones con mayor valor en la función objetivo. Se seleccionan las carteras padres de la lista de soluciones pródigas para generar la siguiente generación usando sus factores de ordenamiento. Con cierta probabilidad  $p_c$  el método de reemplazo será la clonación, y con probabilidad  $1 - p_c$  será el de "Averaged Idol" (ver Capítulo 3, sección 3.1.1).



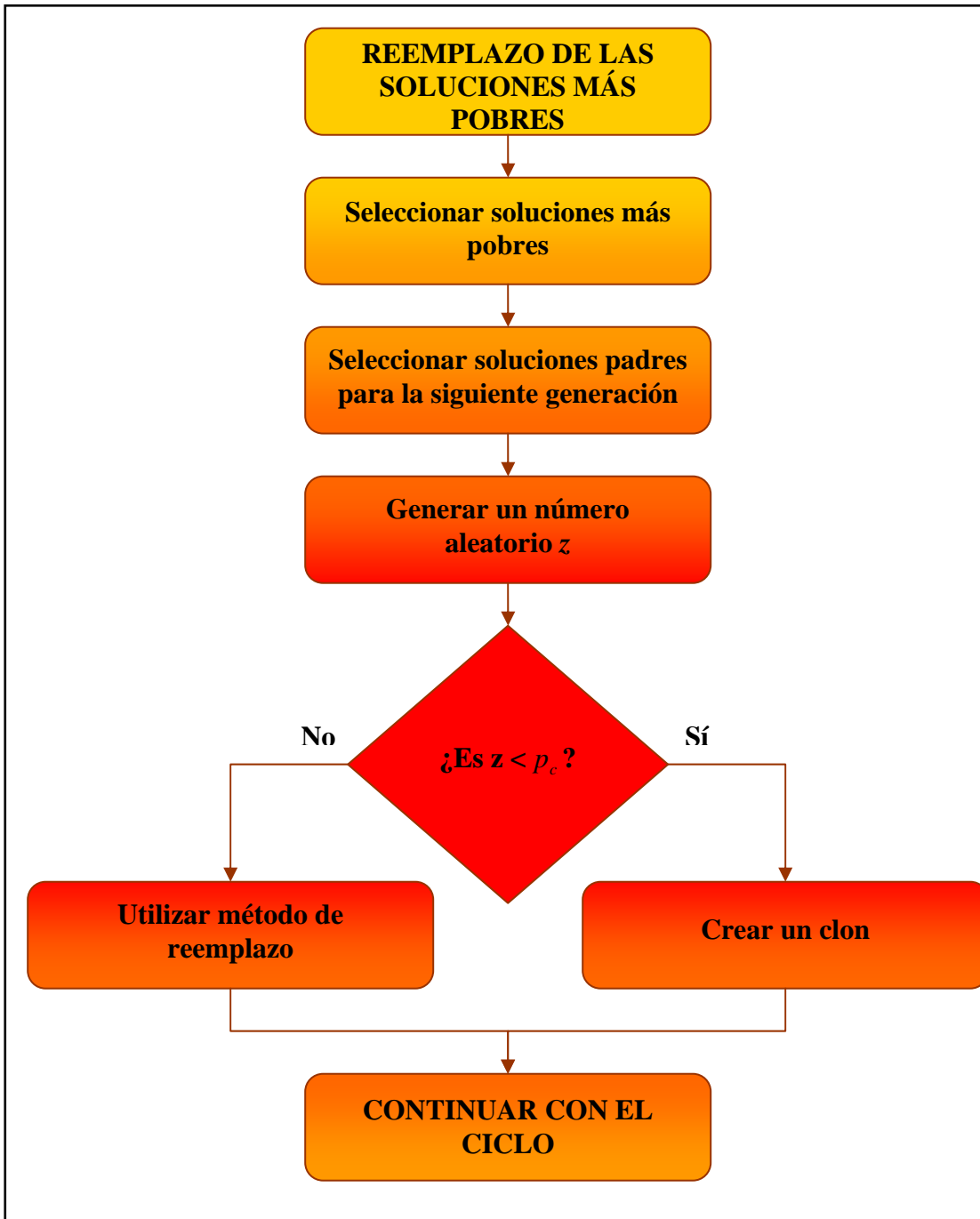
**Figura 4.3** Diagrama de Flujo de la Modificación de la Estructura de la Cartera

Fuente: Elaboración Propia



**Figura 4.4** Diagrama de Flujo de la Evaluación y Ordenamiento de las Soluciones

Fuente: Elaboración Propia



**Figura 4.5** Diagrama de Flujo del Reemplazo de las Soluciones Más Pobres

Fuente: Elaboración Propia



Para que el modelo permitiera trabajar con soluciones que estén fuera del conjunto de soluciones, se siguieron las conclusiones de Pirlot y Crama agregando a la función objetivo un término de penalidad para cada restricción violada. De acuerdo a dichas conclusiones se tomó el valor de la penalidad como  $a * |violación|^b$ , donde  $a$  y  $b$  son escalares y la violación es la diferencia del valor deseado de cada restricción menos el valor obtenido actualmente (Crama & Schyns, 2003; Pirlot, 1992).

Dependiendo de la naturaleza de cada restricción se asignaron las penalidades correspondientes. Para las restricciones de límites inferiores y superiores de cada activo, superiores de comercio y para el número máximo de activos incluidos en la cartera se agregaron las penalidades siguientes a la función objetivo del modelo.

- **Restricción de límite inferior:**

Si  $x_i < \underline{x}_i$ , entonces la penalidad =  $a (\underline{x}_i - x_i)^b$

- **Restricción de límite superior:**

Si  $x_i > \bar{x}_i$ , entonces la penalidad =  $a (x_i - \bar{x}_i)^b$

- **Restricción superior de compra:**

$$Q_{compra} = x_i - x_i^{(0)}$$

Si  $Q_{compra} \in (0, \underline{B}_i)$ , entonces

Si  $Q_{compra} < \frac{\underline{B}_i}{2}$ , entonces la penalidad =  $a(Q_{compra})^b$

Sino la penalidad =  $a(\underline{B}_i - Q_{compra})^b$

- **Restricción superior de venta:**

$$Q_{venta} = x_i^{(0)} - x_i$$

Si  $Q_{venta} \in (0, \underline{S}_i)$ , entonces

Si  $Q_{venta} < \frac{\underline{S}_i}{2}$ , entonces la penalidad =  $a(Q_{venta})^b$ .

Sino la penalidad =  $a(\underline{S}_i - Q_{venta})^b$ .

- **Restricción de número máximo de activos:**

Si  $|\{i \in \{1, \dots, n\} : x_i \neq 0\}| > N$ , entonces la penalidad =  $a(\text{violación})^b$ , donde la violación =  $|\{i \in \{1, \dots, n\} : x_i \neq 0\}| - N$ .

Inicialmente a la restricción de rendimiento no se le agregó ninguna penalidad, pero para obtener resultados con el rendimiento esperado exacto deseado al final del proceso iterativo, se decidió agregarle la penalidad correspondiente a la violación.

- **Restricción de rendimiento**

Si  $\sum_{i=1}^n r_i x_i < R_{esp}$ , entonces la penalidad =  $a\left(R_{esp} - \sum_{i=1}^n r_i x_i\right)^b$ .

Si  $\sum_{i=1}^n r_i x_i > R_{esp}$ , entonces la penalidad =  $a\left(\sum_{i=1}^n r_i x_i - R_{esp}\right)^b$ .

Aumentando estas restricciones al modelo, la formulación matemática para resolver el problema de optimizar la cartera de inversión queda como se plantea a continuación:

$$\text{Min. } \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j + \text{Penalizaciones} \quad (4.1)$$

$$\text{s.t. } \sum_{i=1}^n r_i x_i = R_{esp} \quad (4.2)$$

$$\sum_{i=1}^n x_i = 1 \quad (4.3)$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \quad (4.4)$$

$$\max (x_i - x_i^{(0)}, 0) \leq \overline{B}_i \quad (4.5)$$

$$\max (x_i^{(0)} - x_i, 0) \leq \overline{S}_i \quad (4.6)$$

$$x_i = x_i^{(0)} \text{ ó } x_i \geq (x_i^{(0)} + \underline{B}_i) \\ \text{ó } x_i \leq (x_i^{(0)} - \underline{S}_i) \quad (4.7)$$

$$|\{i \in \{1, \dots, n\} : x_i \neq 0\}| \leq N \quad (4.8)$$

$$\forall i \in \{1, \dots, n\}$$

### 4.3 Arquitectura del Software de Solución

Una vez establecido el modelo completo se continuó con la codificación del programa de optimización de carteras que permitió encontrar soluciones por medio de las técnicas heurísticas seleccionadas, a través del lenguaje de VBA.

El algoritmo ya programado consta de tres módulos. Cada módulo tiene una tarea específica y contiene un número determinado de procedimientos. En las tablas siguientes se describe el contenido de cada módulo.

Tabla 4.1 Módulo Algoritmo<sup>1</sup>

<b>Subrutina</b>	<b>Tarea</b>
<i>GeneraSolucionesIniciales</i>	Genera la población de soluciones iniciales.
<i>EstandarizaPesosCalculaRendimientos</i>	Estandariza los pesos de los activos de manera que la suma sea igual a 1 y calcula el rendimiento de la cartera.
<i>CalculaRiesgoCalidad</i>	Calcula el riesgo de la cartera y la calidad inicial.
<i>CalculaPenalidades</i>	Calcula las penalidades correspondientes a cada restricción y la calidad total de la cartera.
<i>Principal</i>	Hace los procesos de Modificación, Evaluación y Ordenamiento y Reemplazo de las Soluciones Pobres.
<i>MuestraResultados</i>	Muestra la ventana con los datos de la mejor solución encontrada.
<b>Función</b>	<b>Tarea</b>
<i>Penalidad</i>	Calcula la penalidad en la función objetivo de las soluciones de acuerdo al valor de la violación a las restricciones.
<i>SumaTotal</i>	Calcula la suma de la serie 1, 2, ..., y.

Fuente: Elaboración Propia

<sup>1</sup> El código del Módulo Algoritmo se encuentra en los Apéndices A, B y C.

Tabla 4.2 Módulo Validaciones

<b>Subrutina</b>	<b>Tarea</b>
<i>Validación</i>	Valida los datos de entrada de la ventana de parámetros.
<i>Validacion2</i>	Valida los valores de núm. de iteraciones y rendimiento esperado de la ventana inicial del algoritmo.

Fuente: Elaboración Propia

Tabla 4.3 Módulo BarraProgreso

<b>Subrutina</b>	<b>Tarea</b>
<i>MuestraProgreso</i>	Muestra la ventana con la barra de progreso mientras al algoritmo es ejecutado.
<i>ActualizaProgresoAut</i>	Actualiza la barra de progreso según el número de iteraciones que ya se hayan realizado.

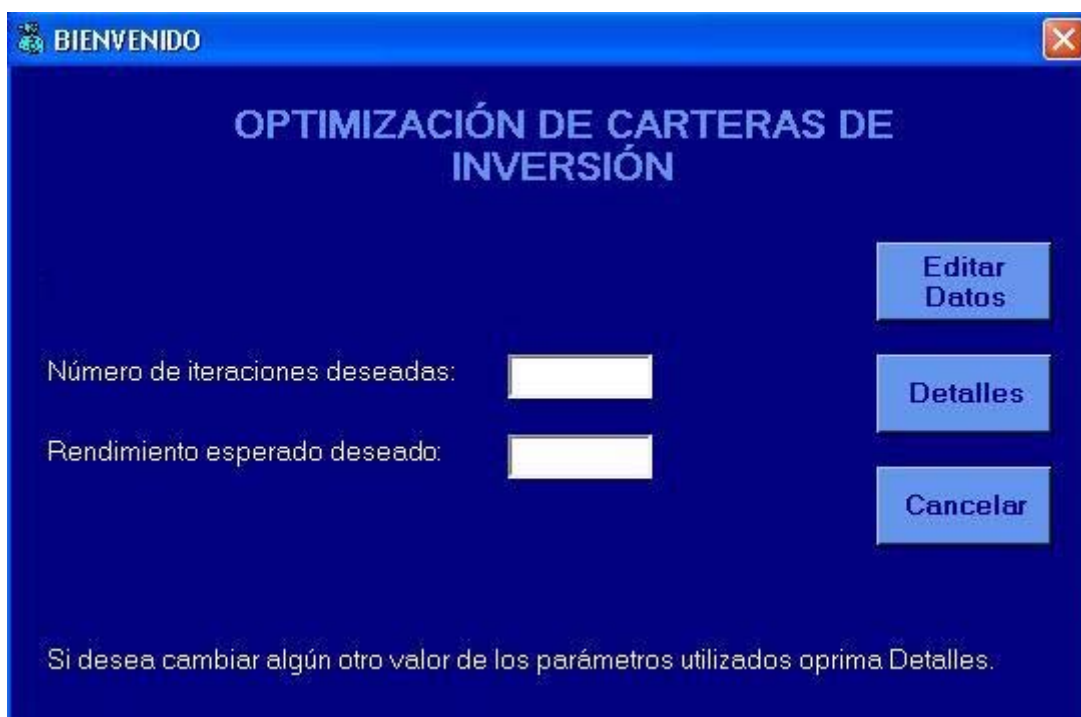
Fuente: Elaboración Propia

A continuación se da una descripción del programa que ejecuta el algoritmo. Al abrir el programa de Excel “Optimización de Carteras.xls” se muestran dos diferentes menús, “Algoritmo de Optimización” y “Resultados”. Inmediatamente se despliega una ventana de presentación (Figura 4.6).



**Figura 4.6** Presentación del Programa de Visual Basic

Fuente: Elaboración Propia



**Figura 4.7** Ventana de Ejecución del Algoritmo

Fuente: Elaboración Propia

En el primer menú, se tiene la opción de “Ejecutar Algoritmo”. Eligiendo esta opción, aparece una ventana (Figura 4.7), donde se debe introducir el número de iteraciones que se correrá el algoritmo y el rendimiento esperado anual para el portafolio de inversión. Para poder cambiar cualquier valor de los parámetros del modelo se necesita oprimir el botón de “Detalles”. Con esta opción aparecerá toda la lista de los parámetros utilizados (Figura 4.8). El usuario puede cambiar cualquiera de estos valores dentro del rango permitido de cada uno de ellos, de no ser así, el programa mostrará un mensaje de “Error” y no permitirá seguir con el proceso hasta que los datos sean modificados.

**DETALLES**

LISTA DE PARÁMETROS UTILIZADOS:

Límite inferior del activo $i$	$x_i =$ <input type="text" value="0"/>	Amplitud de movimiento inicial	$U_0 =$ <input type="text" value="0.3"/>
Límite superior del activo $i$	$\overline{x}_i =$ <input type="text" value="1"/>	Cambio de amplitud	$\gamma_U =$ <input type="text" value="0.985"/>
Compra mínima	$B_i =$ <input type="text" value="0.05"/>	Prob. de incluir el activo $j$	$p_r =$ <input type="text" value="0.4"/>
Venta mínima	$S_i =$ <input type="text" value="0.05"/>	Temperatura inicial	$T_0 =$ <input type="text" value="750"/>
Número máximo de activos	$N =$ <input type="text" value="20"/>	Cambio de temperatura	$\gamma_T =$ <input type="text" value="0.9"/>
Parámetros de penalización	$a =$ <input type="text" value="100"/>	Número de pródigos	$\pi =$ <input type="text" value="12"/>
	$b =$ <input type="text" value="2"/>	Número de elitistas	$\mathcal{E} =$ <input type="text" value="100"/>
Tamaño de generación	$G =$ <input type="text" value="100"/>	Soluciones reemplazadas	$\omega =$ <input type="text" value="10"/>
Número de modificaciones	$m =$ <input type="text" value="4"/>	Prob. de crear un clon	$p_c =$ <input type="text" value="0.3"/>

**Figura 4.8** Ventana del Valor de los Parámetros

Fuente: Elaboración Propia

En el menú de “Resultados” se tiene la opción “Lista de Resultados Obtenidos”, la cual sólo se puede ejecutar si ya se ha corrido el algoritmo anteriormente y se han grabado los resultados encontrados.