CAPÍTULO 5. DESARROLLO DE LA INVESTIGACIÓN Y RESULTADOS

El propósito de este capítulo es el presentar el desarrollo de la propuesta de la tesis, y su principal objetivo: modelar la volatilidad del IPC por medio de multifractales, que extiendan los algoritmos existentes a generadores más complejos, con cuatro segmentos rectilíneos.

Inicialmente presentaremos la extensión de los algoritmos hacia generadores más complejos, para posteriormente introducir el algoritmo desarrollado e implementado con el programa *Mathematica 6*, que nos permitió realizar el "barajeo" aleatorio en cada una de las partes de la interpolación y así poder general el atractor multifractal que modela la gráfica del IPC.

Finalmente en base al atractor multifractal generado, calcularemos la volatilidad a la Mandelbrot, es decir bajo la propuesta del mismo autor, que consiste en analizar las variaciones porcentuales del atractor, para con ello lograr el objetivo de modelar la volatilidad del IPC.

5.1 Generadores Multifractales más Complejos para el IPC

En el capítulo anterior observamos cómo mediante la técnica de interpolación fractal podemos obtener atractores unifractales muy parecidos a las gráficas originales, en este caso la gráfica de la serie de tiempo de los cierres mensuales del IPC de 1994 a 2008. Sin embargo la interpolación fractal no permite modelar un comportamiento más agresivo o salvaje como lo describió Mandelbrot, y es por ello que para modelar este tipo de comportamientos que permitan un mayor grado de aleatoriedad a los sistemas, el mismo

Mandelbrot sugirió es su artículo: "*A Multifractal Walk Down Wall Street*" publicado en la revista *Scientific American* en 1999, que cada etapa de interpolación debería ser precedida por un "barajeo" aleatorio, en el que se permita interpolar con el generador, o con partes del mismo. Es decir que se reproduzca la firma, y partes de ella, de manera aleatoria en cada etapa de la interpolación.

Los algoritmos anteriores desarrollados en el trabajo teórico de Isabel Treviño (2006), lograron, por medio de los algoritmos generados, aumentar la aleatoriedad en el proceso de interpolación al introducir nuevas transformaciones al proceso. Por lo que posteriormente en el trabajo de Villazcan y Torres (2007), encontramos la aplicación de esta misma idea a las series de tiempo del IPC y de Telmex, con resultados satisfactorios.

Es por ello que ahora presentaremos la extensión de estas metodologías y de sus respectivos algoritmos, con el fin de aumentar la aleatoriedad, introduciendo un generador más complejo, el cual a su vez es dividido en partes para ser interpolado aleatoriamente en cada etapa, generando con ello una buena y mejor modelación de la volatilidad del índice, logrando la extensión de los algoritmos. Dicha extensión se realizó conjuntamente con María de Lourdes Alavez Estévez, en su trabajo de tesis (Alavez, 2008).

La idea central del algoritmo de interpolación multifractal, es el de no sólo replicar la firma M (la unión de los cuatro segmentos rectilíneos que forman al generador) en cada una de las regiones correspondientes de la gráfica, sino el de replicar partes de la firma en cada una de las regiones, eligiéndolas de manera aleatoria y produciendo así una versión del "barajeo" aleatorio que propone Mandelbrot para la modelación.

Por lo anterior se trabajará con la firma M que modela a la gráfica:

Figura 5.2 *Firma M*
Fuente: Elaboración propia

Así como con las firmas correspondientes a las partes del generador:

Figura 5.3 *Firma S*
Fuente: Elaboración propia

Figura 5.4 *Firma T*
Fuente: Elaboración propia

Figura 5.5 *Firma U*
Fuente: Elaboración propia

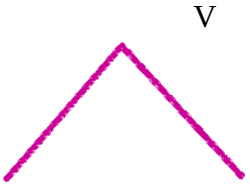

Figura 5.6 *Firma V*
Fuente: Elaboración propia

La firma M representa, a diferencia de los modelos anteriores, una consideración más, que es la caída del índice nuevamente. Es decir, con la firma lo que se trata de representar es el comportamiento del índice, por lo tanto en un primer segmento el índice sube, posteriormente baja y luego vuelve a subir, sin embargo ahora con cuatro segmentos rectilíneos la firma se vuelve más compleja, pues contempla que el índice vuelva a bajar, por lo que la firma de cierto modo se equilibra. Este comportamiento además de extender los algoritmos, también extiende las ideas detrás de ellos. Y es así como posteriormente al partir la firma y generar distintas, también se generan escenarios parecidos al descrito anteriormente, donde el índice viene bajando, sube y vuelve a bajar como en la firma T, o al contrario como en la firma S. Lo cual al permitir que el algoritmo elija aleatoriamente entre más firmas, nos genera el "barajeo" necesario para modelar la volatilidad del IPC de una mejor manera.

La diferencia entre la firma U y la firma V está en que la firma U está formada por los dos segmentos iniciales de la firma total y la firma V por los dos finales, y se tendrá una diferencia significativa al trabajar con la gráfica de la serie de tiempo modelada por la firma M, que en esta ocasión corresponde a la serie del IPC descrita anteriormente.

Las transformaciones que emplearemos son las siguientes:

- Las transformaciones afines $F_1, \dots, F_{23}$, que mapean toda la gráfica (modelada por la firma M), en cada una de las regiones correspondientes (similares a la gráfica total).
- Las transformaciones $FU_1, \dots, FU_{12}$, que mapean la parte de la gráfica correspondiente a la firma U, en cada una de las regiones comprendidas en esa

parte, de manera que de alternen los signos de sus parámetros d correspondientes.

- Las transformaciones $FS_1, \dots, FS_{21}$, que mapean la parte de la gráfica correspondiente a la firma S, en cada una de las regiones comprendidas en esa parte, de manera que de alternen los signos de sus parámetros d correspondientes.
- Las transformaciones $FT_1, \dots, FT_{14}$, que mapean la parte de la gráfica correspondiente a la firma T, en cada una de las regiones comprendidas en esa parte, de manera que de alternen los signos de sus parámetros d correspondientes.
- Las transformaciones $FV_1, \dots, FV_{11}$, que mapean la parte de la gráfica correspondiente a la firma V, en cada una de las regiones comprendidas en esa parte, de manera que de alternen los signos de sus parámetros d correspondientes.

El caso teórico de cuatro regiones se ejemplifica en la siguiente figura:

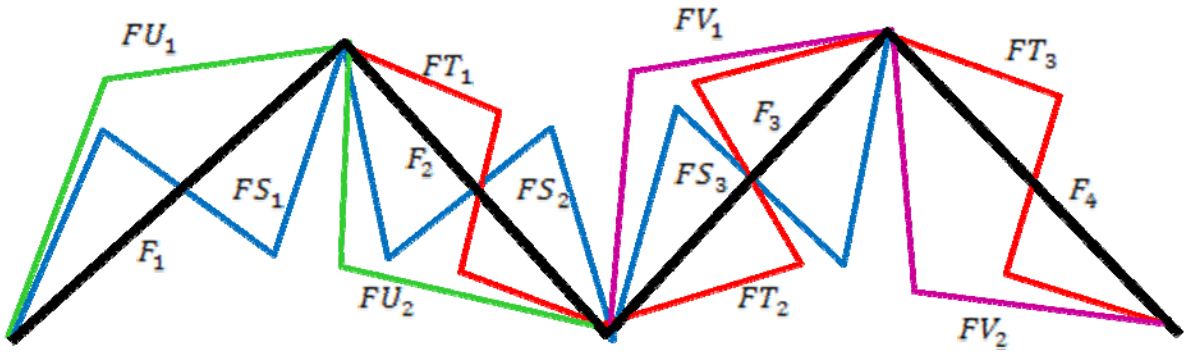

Figura 5.7 *Firma Generadora Compleja*
Fuente: Elaboración propia

El algoritmo descrito anteriormente, logra la extensión de los algoritmos existentes, presentados inicialmente en el trabajo teórico de Treviño (2006), y posteriormente en las aplicaciones prácticas de Villazcan y Torres a las series del IPC y de Telmex (2007), ambos algoritmos presentados en tesis dirigidas por el Dr. Guillermo Romero.

Para la extensión de estos métodos es importante recalcar la siguiente observación: un factor importante a considerar, es el punto de origen para iniciar la interpolación multifractal, éste debe de ser el punto de origen de la gráfica original, a diferencia de los algoritmos anteriores que comenzaban en (0,0). Lo anterior pone de manifiesto la alta sensibilidad a las condiciones iniciales del modelo. Y pone en evidencia su naturaleza caótica y compleja.

En base a la metodología anterior y tomando en cuenta las consideraciones anteriores, se logró modelar un algoritmo para la interpolación multifractal de la serie de tiempo de los cierres mensuales del IPC para el período de 1994 a 2008.

Para el caso del IPC, las transformaciones F que se encontraron fueron un total de 23, y son las siguientes (ver página siguiente):

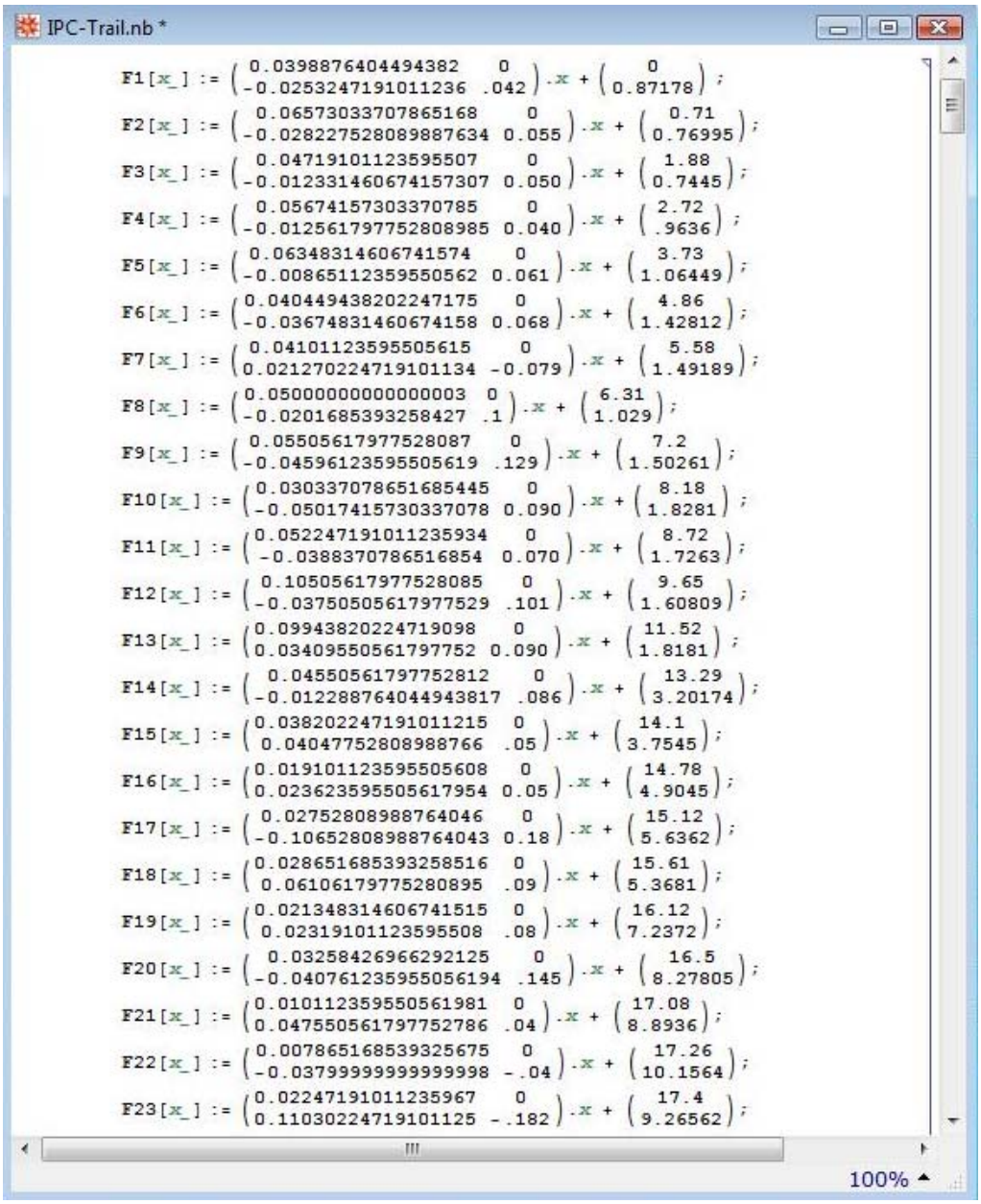

Figura 5.8 *Transformaciones F*
Fuente: Elaboración propia

Las transformaciones FS fueron un total de 21, y son las siguientes:

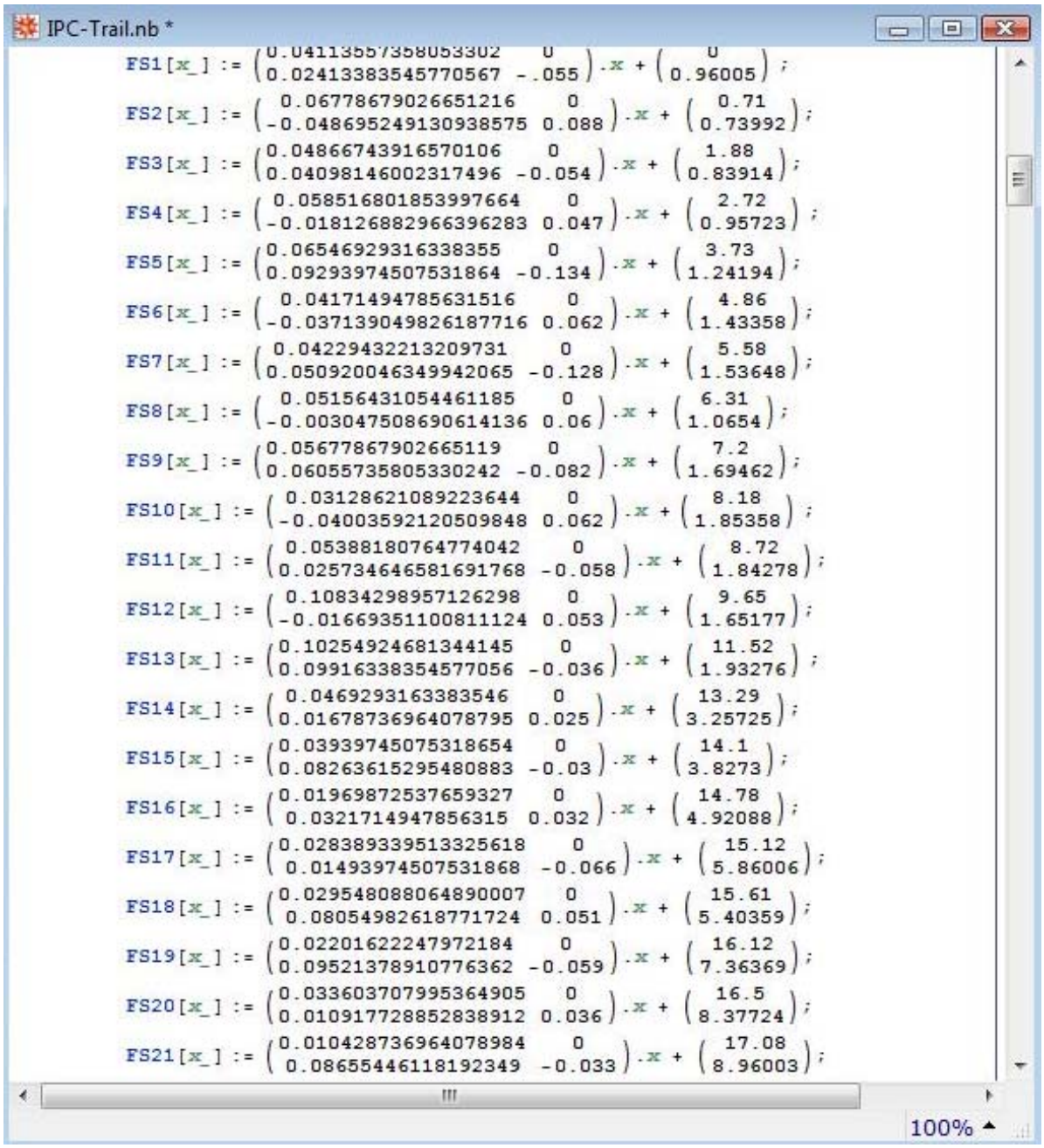

Figura 5.9 *Transformaciones FS*
Fuente: Elaboración propia

Las transformaciones FT fueron un total de 14, y son las siguientes:

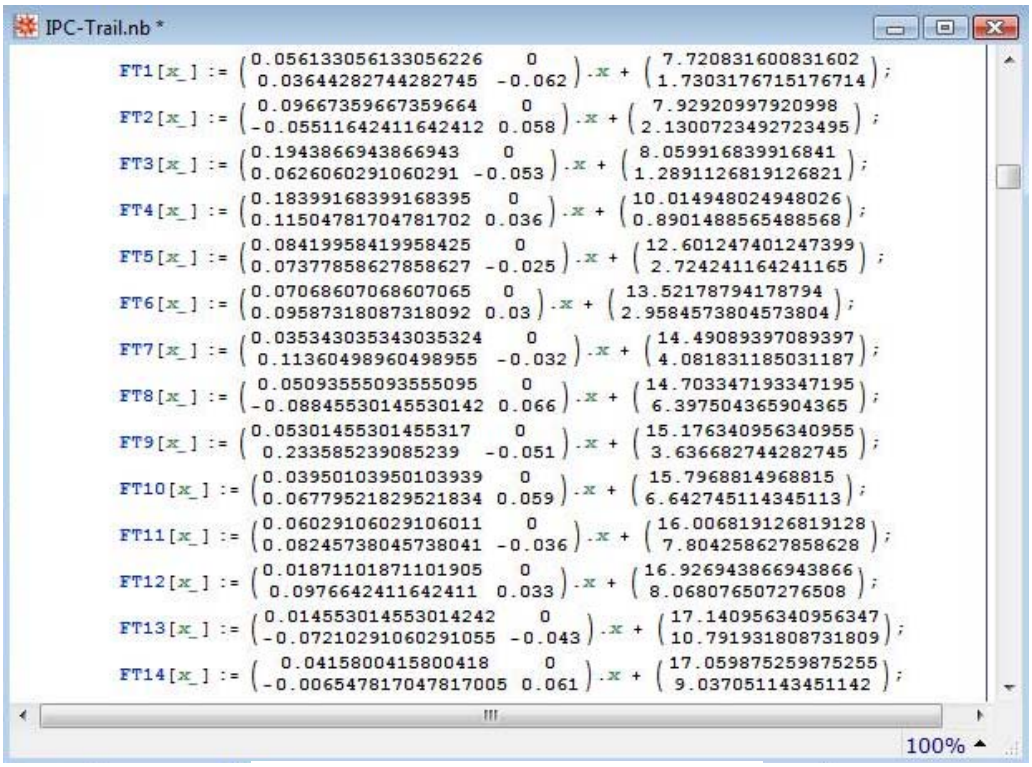

Figura 5.10 *Transformaciones FT*
Fuente: Elaboración propia

Las transformaciones FU fueron un total de 12, y son las siguientes:

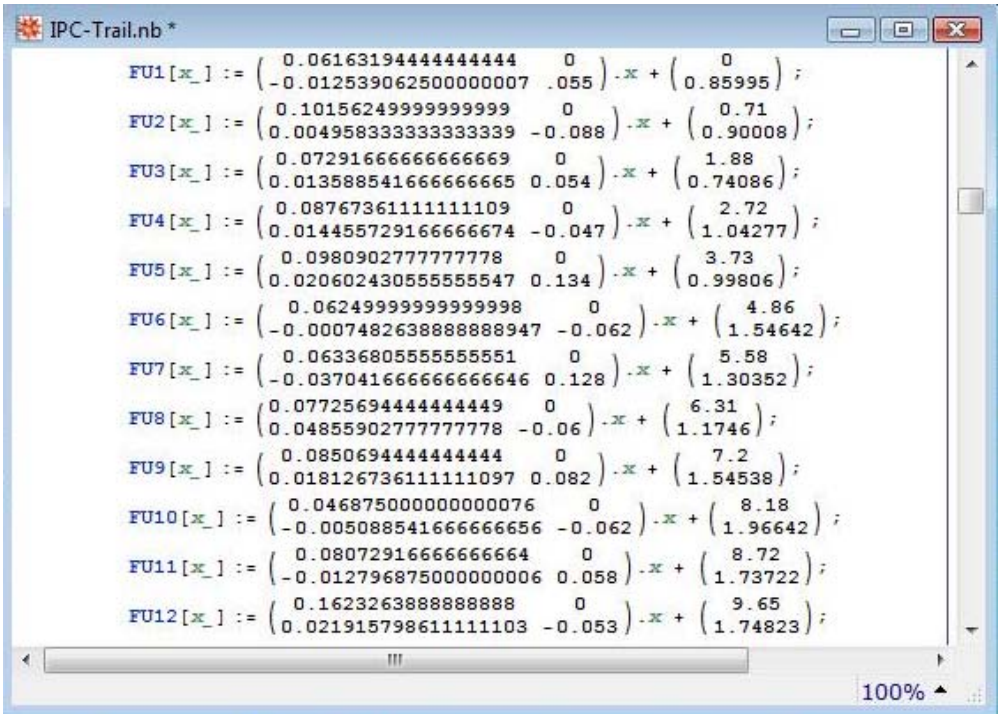

Figura 5.11 *Transformaciones FU*
Fuente: Elaboración propia

Y finalmente las transformaciones FV fueron un total de 11, y son las siguientes:

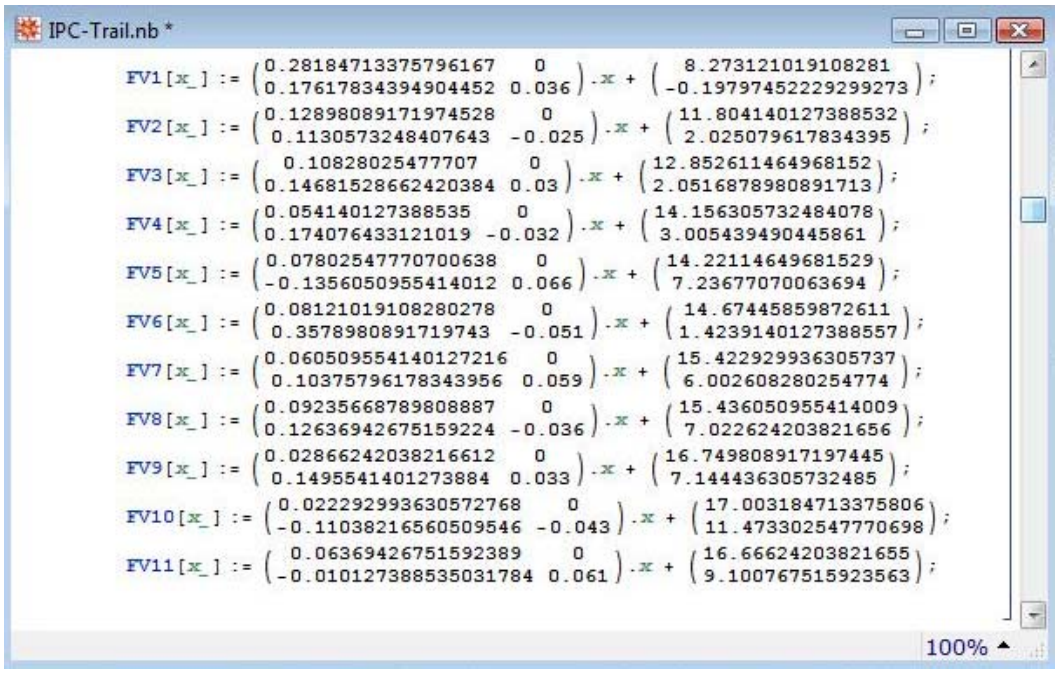

Figura 5.12 *Transformaciones FV*
Fuente: Elaboración propia

Después de encontrar las trasformaciones, el programa en *Mathematica 6* que genera el atractor multifractal que modela la gráfica del IPC, eligiendo de manera aleatoria entre cada transformación se presenta a continuación:

```
Clear[T];
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS1, FU1}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        0 ≤ x[[1, 1]] ≤ 0.71
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS2, FU2}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        0.71 < x[[1, 1]] < 1.88
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS3, FU3}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        1.88 ≤ x[[1, 1]] ≤ 2.72
```

```
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS4, FU4}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        2.72 < x[[1, 1]] < 3.73
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS5, FU5}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        3.73 ≤ x[[1, 1]] ≤ 4.86
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS6, FU6}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        4.86 < x[[1, 1]] < 5.58

T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS7, FU7}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
         5.58 ≤ x[[1, 1]] ≤ 6.31
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS8, FU8}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
         6.31 < x[[1, 1]] < 7.2
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS9, FU9}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
        7.2 ≤ x[[1, 1]] ≤ 8.18

  T[x_] :=
   Par[
     {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
       F17, F18, F19, F20, F21, F22, F23, FS10, FU10, FT1}[[
      Random[Integer, {1, 26}]]][x[[1]]]] /;
          8.18 < x[[1, 1]] < 8.72
  T[x_] :=
   Par[
     {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
       F17, F18, F19, F20, F21, F22, F23, FS11, FU11, FT2}[[
      Random[Integer, {1, 26}]]][x[[1]]]] /;
          8.72 ≤ x[[1, 1]] ≤ 9.65
  T[x_] :=
   Par[
     {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
       F17, F18, F19, F20, F21, F22, F23, FS12, FU12, FT3}[[
      Random[Integer, {1, 26}]]][x[[1]]]] /;
          9.65 < x[[1, 1]] < 11.52
  T[x_] :=
   Par[
     {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
       F17, F18, F19, F20, F21, F22, F23, FS13, FV1, FT4}[[
      Random[Integer, {1, 26}]]][x[[1]]]] /;
          11.52 ≤ x[[1, 1]] ≤ 13.29
```

```
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS14, FV2, FT5}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       13.29 < x[[1, 1]] < 14.10
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS15, FV3, FT6}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       14.10 ≤ x[[1, 1]] ≤ 14.78
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS16, FV4, FT7}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       14.78 < x[[1, 1]] < 15.12
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS17, FV5, FT8}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       15.12 ≤ x[[1, 1]] ≤ 15.61

T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS18, FV6, FT9}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       15.61 < x[[1, 1]] < 16.12
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS19, FV7, FT10}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       16.12 ≤ x[[1, 1]] ≤ 16.50
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS20, FV8, FT11}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       16.50 < x[[1, 1]] < 17.08
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FS21, FV9, FT12}[[
    Random[Integer, {1, 26}]]][x[[1]]] /;
       17.08 ≤ x[[1, 1]] ≤ 17.26
```

```
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FV10, FT13}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
       17.26 < x[[1, 1]] < 17.40
T[x_] :=
 Par[
   {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16,
     F17, F18, F19, F20, F21, F22, F23, FV11, FT14}[[Random[Integer, {1, 25}]]][
    x[[1]]]] /;
       17.40 ≤ x[[1, 1]] ≤ 17.8
```

Después de 200 iteraciones se obtiene la sigueinte gráfica, que es un atractor multifractal:

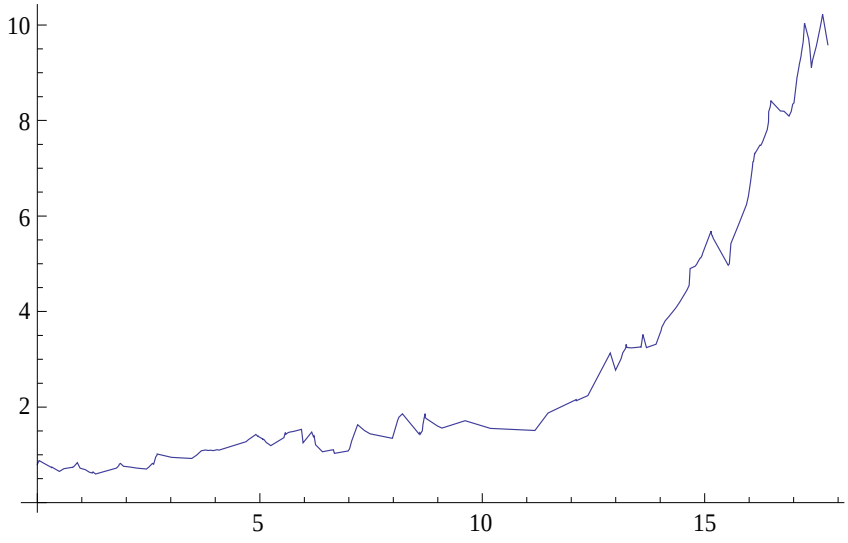

Figura 5.13 *Atractor Multifractal Generado IPC*
Fuente: Elaboración propia

Ahora nuestro objetivo principal está cerca, y para ello emplearemos la técnica de modelación de la volatilidad enunciada por Mandelbrot es su artículo: "*A Multifractal Walk Down Wall Street*" publicado en la revista *Scientific American* en 1999, y que presentaremos en la siguiente sección.

5.2 Modelando la Volatilidad del IPC a la Mandelbrot

Para la modelación de la volatilidad, de acuerdo con la propuesta de Mandelbrot, para cada período *t* se grafica el cociente de la diferencia entre el valor que toma la variable en el período *t+1* respecto de su valor en el período anterior *t*, y luego se divide esta diferencia por el valor absoluto de la variable en el período *t*, más uno. Lo anterior implementado en el programa *Mathematica 6*, queda como sigue:

```
NesT = NestList[T,
        {{0., 0.8}}, 200];
LT = Par[NesT];
SLT = Sort[LT];
ListPlot[SLT, PlotStyle -> PointSize[.005],
 Joined → True]
Clear[f0];
f0[x_, y_] := N[x / (1 + Abs[y])]
Clear[Lp];
Lp[L_] := Table[
  {i, f0[L[[i + 1, 2]] - L[[i, 2]],
    2 + Abs[L[[i, 2]]]]}, {i, 1, Length[L] - 1}]
ListPlot[Lp[SLT], PlotStyle -> PointSize[.005],
 Joined → True, PlotRange → {-.20, .20}]
```

Y finalmente después de 200 iteraciones, el resultado obtenido es el que sigue:

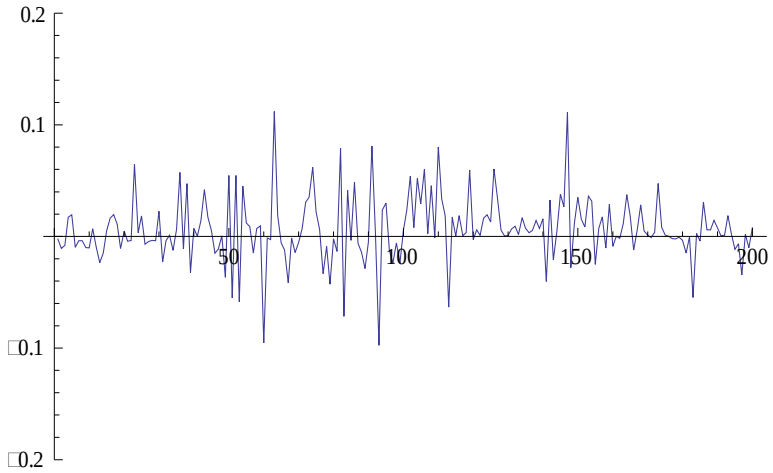

Figura 5.14 *Volatilidad Modelada del IPC*
Fuente: Elaboración propia

La figura 5.14 es la gráfica de la volatilidad generada por nuestro algoritmo, a partir del atractor multifractal construido en base al generador de cuatro segmentos rectilíneos, la gráfica real de la volatilidad del IPC se muestra a continuación:

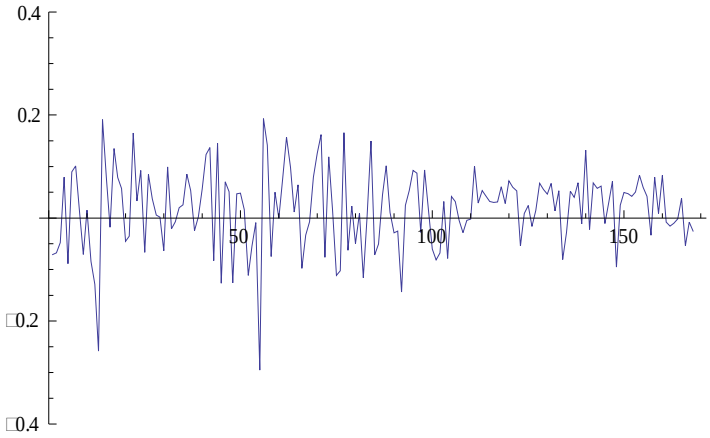

Figura 5.15 *Volatilidad Real del IPC*
Fuente: Elaboración propia

Como podemos observar, hay una gran similitud entre la gráfica original y la imagen modelada, lo cual completa la propuesta de esta tesis, y asegura su existencia.

Es importante señalar que éste es sólo uno de los tantos escenarios de volatilidad que son generados por nuestro algoritmo, pues cada caso es distinto, dada la naturaleza aleatoria al elegir las transformaciones. Lo que resulta muy interesante, porque tomando en cuenta las propiedad de los sistemas caóticos en cuanto a la sensibilidad que presentan ante los cambios, podemos observar que las gráficas de la volatilidad generadas, pudieran haber sido escenarios alternos a los ocurridos en dicho período, es decir, millones de caminos que colapsan en uno solo, en la realidad presente.