



Instituto Politécnico Nacional

**Unidad Profesional Interdisciplinaria de
Ingeniería y Ciencias Sociales y
Administrativas**



Aplicación de los Sistemas Evolutivos a la Composición Musical

TESIS

**Que para obtener el
Titulo de**

**Licenciado en Ciencias de la
Informática**

Presenta

Horacio Alberto García Salas

1999

Resumen

Summary

Introducción

.....1

Capítulo I.- Introducción a la Música por Computadora

.....4

1.1 Sonido

.....

.4

1.2 Sistemas y Ruidos

.....5

1.2.1 Sistemas Determinísticos, Probabilísticos y Ruido Browniano6

1.2.1.1 Sistemas Determinísticos o Tradicionales6

1.2.1.2 Sistemas Probabilísticos7

1.2.1.3 Ruido Browniano o Pardo9

1.2.2 Sistemas Desordenados y Ruido Blanco15

1.2.2.1 Sistemas Desordenados15

1.2.2.2 Ruido Blanco15

1.2.3 Sistemas Caóticos, Fractales y Ruido 1/f18

1.2.3.1 Sistemas Caóticos18

1.2.3.2 Fractales19

1.2.3.3 Ruido 1/f21

<u>Capítulo II .- Sistema Evolutivo Generador de Música</u>	27
2.1 <u>Introducción a los Sistemas Evolutivos</u>	27
2.2 <u>Sistema Compositor</u>	28
2.3 <u>Funcionamiento del Corazón del Sistema Evolutivo</u>	32
<u>Capítulo III .- Funcionamiento y Construcción del Sistema</u>	40
3.1 <u>Administrador de los Procesos</u>	43
<u>Conclusiones</u>	53
<u>Anexo A.- Descripción del Formato Musical</u>	56
<u>Bibliografía</u>	58

RESUMEN

Este trabajo es el resultado de la aplicación de la Informática a una de las bellas artes, la Música, que dan por resultado otra área diferente, que desafortunadamente no tiene la atención que merece, la Música Informática.

Con el objetivo de desarrollar un sistema compositor de música, hemos tenido que introducirnos un poco, en dos

disciplinas, que a nuestro criterio, son de gran relevancia para la humanidad.

Por un lado, está la música, que por su carácter de arte, se le ha tratado de mantener, un tanto cuanto, alejada de la ciencia y por el otro lado, está la informática que ha demostrado, en incontables ocasiones, su gran utilidad y versatilidad. Ambas disciplinas son muy extensas, razón por la cual, no pretendemos abarcar su totalidad en unas cuantas páginas.

Trataremos de explicar, de la manera más sencilla, algunos conceptos necesarios para entender el funcionamiento de nuestro sistema.

Empezaremos por platicar, brevemente, acerca de los sistemas y la manera como se diferencian con respecto a su comportamiento, haciendo referencia a los sistemas tradicionales o deterministas, caóticos y desordenados.

Veremos como la evolución de la ciencia ha encontrado formas distintas de hacer modelos, buscando, no intrincados métodos matemáticos para hacerlo, sino la manera más

natural, tratando de que, no sólo sean más sencillos, sino más armoniosos.

Hablaremos de distintos tipos de ruidos y su aplicación para generar diferentes tipos de música. Presentaremos algunas gráficas de estos ruidos, obtenidas experimentalmente, así como otra obtenida físicamente por medio de la fotografía, en donde podremos comparar el comportamiento del modelo con respecto al del sistema real.

Explicaremos algunos algoritmos para generar música a partir de estos ruidos, basados en la teoría descrita por algunos científicos que se han dedicado a este tipo de estudios, así como también algunos otros desarrollados por nosotros.

Hablaremos de los fractales, un tema que ha adquirido bastante importancia desde hace algunos años debido a su belleza para describir el mundo y veremos su aplicación en la generación de música.

Por medio de los fractales se pueden hacer modelos que simulan el carácter caótico del Universo y dar explicación a

distintos fenómenos que, al final de cuentas, gozan de características similares.

Como parte fundamental de este trabajo, hablaremos de los Sistemas Evolutivos y los aplicaremos junto con técnicas fractales a la creación de sistemas informáticos de generación de música, que se comportan como la misma Naturaleza, de forma caótica.

Esperamos que con nuestras humildes palabras se entienda la belleza de la Música y de la Informática y aún más satisfactorio sería que muchas personas se unan al desarrollo Informático-Musical.

SUMMARY

The present work is the result of the application of Informatics to one of the Fine Arts: Music. The combination of these two yields a new, different area, which unfortunately, has not received the attention it deserves. This area is the Informatic Music.

Aiming to develop a music-composing system, the author delved into two disciplines, which under his criteria, are of the utmost importance for humanity.

On one side is Music. Due to its classification as an art, it has been somewhat kept apart from science. On the other is Informatics, which has proved time and again its ample usefulness and versatility. Both disciplines are vast and rich. Bearing this in mind, the author stays away from the pretension of embracing them in entirety in a few pages.

A simple extrication of some necessary concepts for the understanding of the system comes upfront in this work.

A brief talk about systems follows and the differences in their behavior, referring to traditional, deterministic, chaotic and disordered systems.

The reader will find a brief description of how the evolution of science has found its way to the creation of models, searching not for complex mathematical methods to achieve them, but the most natural way and trying to yield not only the simplest, but the most harmonious.

The present work will explore along different kinds of noises and their application on the creation of different sorts of music. Some graphics of such noises —obtained experimentally— will be shown, together with another one obtained physically by means of photography. At this point, the reader will be able to compare the model against the real system.

The explanation of some algorithms to generate music on the basis of noises will come next. Some of these algorithms are based upon the theory described by some scientists who have devoted to this kind of studies, and some others were developed by the author.

This work also talks about fractals. A topic of particular relevance, acquired in recent years due to its beauty to portray the world. Its application in the field of music generation will also be reviewed.

Through fractals, researchers can make models which simulate the chaotic features of the universe and provide an explanation for seemingly different phenomena which in the bottom share similar characteristics.

A fundamental part of this work is the scope on Evolutionary Systems, their application in conjunction with fractal techniques to the creation of music-generating informatic systems which behave like Nature itself, in a chaotic pattern.

It's the author's hope to achieve through this humble contribution to spread an understanding of the beauty of Music and Informatics, and an even more satisfactory dream to have many people join the process of the Informatic-Musical development.

INTRODUCCIÓN

Modelar la naturaleza es uno de los principales objetivos de la ciencia ya que es una de las formas en las que el ser humano va adquiriendo el conocimiento. Lograr construir un buen modelo tiene ciertas ventajas, por ejemplo, se pueden hacer modificaciones a las distintas variables del sistema y observar como afecta al modelo, si con dichas modificaciones se aumenta la eficiencia del sistema, entonces se puede pensar en modificar al sistema real.

En este contexto la Informática es una ciencia que, a pesar de su relativa juventud, ha hecho, en muy poco tiempo, un cambio en la vida de todas las personas sobre la Tierra. La velocidad con la que ha ido evolucionando es tan alta que apenas si queda un poco de tiempo para respirar. Una de las herramientas principales que ha permitido que tal cambio se lleve a cabo de manera tan vertiginosa es la computadora. Gracias a las computadoras, todas las personas que tienen que ver con el manejo de información, han encontrado una herramienta ideal para plasmar sus

ideas, para poder modelar sus teorías, para solucionar problemas que hace no más de 50 años hubiera sido inimaginable pensar que podrían solucionarse.

Existen muchas disciplinas científicas y áreas ligadas íntimamente con la Informática, todas ellas muy interesantes, como lo es, por ejemplo, la inteligencia artificial, realidad virtual, robótica, multimedia, etc. Cada una de las cuales se subdivide en distintas ramas, que por sí mismas representan extensos caminos de conocimientos.

También las bellas artes han sido complementadas con el uso de tecnologías modernas. Ya no nos sorprende ver que un bailarín de ballet le dé vida y movimientos propios a una caricatura, permitiendo que los dibujos animados se muevan y comporten, prácticamente como seres humanos.

Encontramos dentro de las bellas artes, “la música”, que de acuerdo con historiadores, arqueólogos y sociólogos, ha acompañado al ser humano desde sus inicios.

La música representa otro amplio sendero del conocimiento humano y al igual que éste, ha ido evolucionando a través del tiempo. De hecho podríamos

pensar en caracterizar las diferentes etapas del desarrollo humano de acuerdo a su música y para no ir tan lejos, un claro ejemplo lo vemos en estos tiempos que nos han tocado vivir, en donde el caos y la vida agitada caracterizan a nuestras sociedades. Los músicos actuales se valen de computadoras y equipos tales como sintetizadores, tarjetas de sonido, secuenciadores, interfases musicales que permiten conectar instrumentos como guitarras y violines a las computadoras y una amplia gama de dispositivos electrónicos creados para la generación de sonido, metiendo en problemas en algunas ocasiones a los músicos con la informática y a los informáticos con la música.

De esta manera, tenemos por un lado a la informática, que ha revolucionado el mundo drásticamente y por otro lado la música, que es tan antigua como el mismo ser humano. Ambas áreas se han ido uniendo con el paso del tiempo, creando el marco dentro del cual fue desarrollado este trabajo, “La Música Informática”.

Dentro del área de desarrollo informático musical, se encuentran diferentes corrientes de desarrollo y más ahora

con el advenimiento de lo que conocemos como multimedia, todo software que se desarrolla requiere del manejo de sonido, de esta manera las aplicaciones impactan a dos de nuestros sentidos, el visual y el sonoro. Así pues encontramos a quienes se dedican a fabricar software de edición musical, reproductores de música, secuenciadores, etc.

Otra de las corrientes, hasta ahora más de tipo experimental, es la dedicada a la obtención de software para la composición musical automática, “el sueño” de la maquina con la capacidad de componer música por siempre exquisitamente diferente.

Con la aparición de los fractales y la aplicación de las teorías de caos, el problema de crear una máquina capaz de hacer composición musical automática, se enfrentó pensando en lo más Natural. Y no es para menos, los hermosos paisajes generados por el científico Benoit Mandelbrot, mostraron que la mejor manera de hacer modelos es imitar el comportamiento de la Naturaleza.

Aplicando teorías fractales, el científico Richard F. Voss, ha hecho desarrollo musical informático, obteniendo compositores verdaderamente hermosos. En su trabajo ha desarrollado generadores de ruido browniano, ruido blanco y resalta uno que genera música muy agradable, que se denomina ruido 1/f.

En este punto es donde la música compuesta automáticamente cambia radicalmente, ya que se empieza a modelar pensando en el comportamiento de la Naturaleza. En el caso del compositor de ruido 1/f, sugiere Richard F. Voss que se está modelando una característica de la Naturaleza. Agradable es escuchar el ruido 1/f y con toda razón es el más Natural.

Dentro de los trabajos de otro investigador llamado Fernando Galindo Soria, podemos encontrar una amplia gama de aplicaciones de una herramienta informática desarrollada por él mismo, denominada “Sistemas Evolutivos”.

Los Sistemas Evolutivos modelan la característica principal de la Naturaleza, la Evolución.

La Naturaleza nos ha enseñado desde hace largo tiempo, que los sistemas que evolucionan son los que tienen la capacidad de adaptarse al medio que los rodea. De tal forma que si lo que se pretende al hacer un modelo es imitar a la realidad, porqué no desarrollar modelos o sistemas con la capacidad de adaptarse al medio que los rodea.

Aplicando esta filosofía, Fernando Galindo Soria ha desarrollado impresionantes Sistemas Evolutivos, algunos de ellos aplicados a la generación de paisajes.

1. INTRODUCCIÓN A LA MÚSICA POR COMPUTADORA

La Música es un arte que ha evolucionado de manera sorprendente, al extremo de que ya se puede plantear la posibilidad de “enseñarles” a las computadoras a componer música, para lo cual, por medio de la Informática, que es la ciencia que se encarga de estudiar la información, se puede buscar la manera de modelar los fenómenos musicales en términos de información.

Ahora bien, para esto es muy importante tener una clara imagen de la información que se está manejando y como hablar de música es hablar de sonido, es bueno empezar a platicar del mundo sonoro: algunos ruidos y algunos procesos matemáticos con los que se pueden describir a estos ruidos.

1.1 SONIDO

Uno de los sentidos con los que estamos dotados, es el oído. Gracias a él, podemos formar una imagen de lo que es el mundo sonoro. El oído percibe lo que llamamos sonido, que es la energía de movimiento vibratorio y estímulo adecuado para el oír. Los sonidos se clasifican en tonos y ruidos. Los tonos son vibraciones periódicas regulares que al ser oídas pueden descomponerse en sus armónicas. Los ruidos son vibraciones irregulares, al azar, que al ser oídas no se pueden descomponer[1]. Cuando se habla de música, no existe un límite en el que se pueda distinguir claramente la diferencia entre un ruido y un tono, pues, en algunos casos, esta diferencia es un tanto cuanto subjetiva y prueba de ello es que existen diferentes tipos de música que para ciertas personas son, sin duda alguna tonos y para otras no son más que ruidos, es el caso de la música oriental, que para ciertos occidentales es sólo ruido, otro ejemplo, es el rock, que para muchas personas es sólo gritos y tamborazos.

El sonido es de relevante importancia en las relaciones humanas, ya que permite llevar cabo la conversación. Los sonidos que producimos al hablar, se generan por medio de un sistema muy ingenioso. Las cuerdas vocales se hacen vibrar, produciendo ciertos tonos que se pueden variar al modificar la tensión de las cuerdas, dichos tonos se amplifican en la boca y en las cavidades nasales, permitiendo que se escuchen, no tonos

puros como los que genera un diapasón, pero lo suficientemente claros como para permitir la conversación. De hecho, hay personas que a través de la práctica han ido “educando la voz” y tienen tanto control sobre los sonidos que producen, que realmente se les considera como instrumentos musicales dentro de una orquesta [1].

Por otro lado, cuando se habla de ruido, es importante comentar que existen diferentes tipos de ruidos: Ruido $1/f$, ruido de fotón o de radiación, ruido de temperatura, ruido de amplificador, ruido térmico, ruido atmosférico, ruido cósmico, etc. Modelar algunos de ellos resulta un excelente ejercicio como introducción a la música por computadora.

Por lo que a continuación y a partir de los conceptos de *sistema* y *correlación* se verán algunos tipos de ruidos y se mostrará cómo generarlos.

1.2 SISTEMAS Y RUIDOS

Un *sistema*, es un conjunto de elementos interrelacionados. Esto quiere decir, que prácticamente todo lo que conocemos es o pertenece a un sistema.

Es tal la cantidad de sistemas que se pueden encontrar en el Universo, que nos hace pensar que la Naturaleza tiene por gusto crear sistemas, un caso particular es nuestro propio planeta Tierra, que forma parte de lo que denominamos sistema solar, que está formado por una estrella llamada Sol y sus planetas. Este sistema forma parte de un conjunto que contiene miles de millones de estrellas, que muy probablemente tienen sistemas planetarios propios. Este conjunto, llamado Vía Láctea, es una galaxia en espiral, de dimensiones fuera de los límites de la comprensión humana, su diámetro es de aproximadamente 100,000 años luz. Ésta pertenece a un cúmulo de aproximadamente 20 galaxias y existe una gran cantidad de estos sistemas formando parte de este Universo esplendoroso.

Sin embargo, no sólo los cuerpos macroscópicos forman sistemas y subsistemas, pues de hecho, los grandes sistemas están compuestos por otros. Esto se ve en el caso de nuestra casa, la Tierra, en donde habitamos infinidad de seres animados e inanimados, todos formados por sistemas y subsistemas con propósitos específicos: por ejemplo el circulatorio, el respiratorio, el óseo, etc; tales sistemas y subsistemas contienen elementos que escalonadamente llegan a ser formados por células que son a su vez de gran variedad y

funcionamiento: como las neuronas del cerebro humano, por medio de las cuales se transmite información, que a su vez están formadas por moléculas y átomos, al igual que los seres inanimados. Las moléculas están constituidas por átomos con neutrones, electrones, protones, y una amplia gama de subpartículas.

Como existe tal cantidad de sistemas y la realidad que viven esos sistemas va cambiando a través del tiempo, también la forma de visualizar la realidad, así como la manera de representar los distintos fenómenos que suceden día con día van evolucionando.

Por tal motivo si se quiere hacer que un sistema “artificial” se comporte como algún sistema real, decimos que estamos *modelando* y siempre que se hace un modelo es bueno saber de que tipo de sistema se trata, por tal motivo comenzaremos por describir algunos tipos de sistemas con la finalidad de introducirnos a la música informática.

1.2.1 Sistemas Determinísticos y Ruido Browniano

1.2.1.1 Sistemas Determinísticos o Tradicionales

Se denominan sistemas determinísticos a todos aquellos en donde se pueden encontrar ecuaciones que describen de manera general al sistema, con la característica de que siempre que las variables del sistema adquieran valores similares, se obtendrán resultados similares.

Por ejemplo: supóngase que se trata de saber la distancia que recorre un vehículo que viaja a velocidad constante de 3 m/seg en 2 segundos, hacer un modelo de este sistema es, prácticamente, directo.

$$d = (v)(t) \text{ (distancia igual a velocidad por tiempo)}$$

$$\text{si } v=3 \text{ m/seg. y } t=2 \text{ seg} \quad d = (3\text{m/seg}) (2 \text{ seg}) = 6 \text{ m}$$

Supongamos que hubiera una pequeña variación en la velocidad y que en lugar de 3 m/seg.

Ahora recorra 4 m/seg. entonces

$$d = (4\text{m/seg}) (2 \text{ seg}) = 8 \text{ m.}$$

que es un resultado que cambia de manera directamente proporcional a la variación de velocidad.

Una característica de los sistemas determinísticos es que son bastante ordenados, lo que facilita su comprensión, el problema real es que no todos los sucesos de la naturaleza se pueden representar fácilmente en forma determinística. ¿Cómo modelar el crecimiento de un árbol, el comportamiento de las olas del mar, la formación de galaxias o el tránsito de una avenida? Por lo que como siguiente paso, veremos sistemas en los cuales los fenómenos analizados tienen un comportamiento probabilístico.

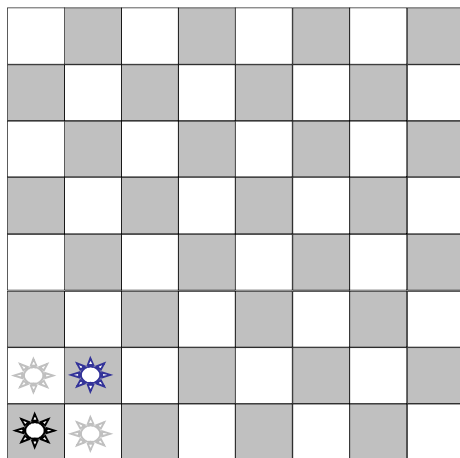
1.2.1 2 Sistemas Probabilísticos

Cuando se habla de *sistemas probabilísticos*, se está hablando de sistemas que están formados por eventos que tienen mayor o menor posibilidad de ocurrir. En general, siempre que se habla de probabilidades se piensa en los ejemplos clásicos de los juegos de azar, pero las probabilidades se pueden aplicar a diferentes áreas.

Por ejemplo, las personas que se dedican al estudio del clima, se encuentran, frecuentemente, en medio de procesos probabilísticos. Cuando se espera la formación de un huracán, es muy importante saber que probabilidad existe de que esto ocurra, ya que muchas veces las vidas de personas dependen de ello. Por otro lado, debido al carácter caótico del desplazamiento de los huracanes, se tiene que recurrir a los procesos probabilísticos para encontrar una posible ruta de desplazamiento del huracán. De esta forma, se intenta predecir si puede tocar tierra o no y si lo hace, que pueblos evacuar. De una buena información, obtenida a partir de un modelo probabilístico, se puede tomar una certera decisión de evacuar o no un pueblo, en caso de que el huracán se comporte según las predicciones permite que se salven las vidas de las personas, de lo contrario, se ponen en riesgo pueblos enteros o en el más benigno de los casos se evacuan pueblos que no lo necesitan, acción que cuesta mucho dinero.

Así como este ejemplo, podemos encontrar miles de aplicaciones de los procesos probabilísticos. En ciertos sistemas probabilísticos, los elementos se relacionan de tal forma, que la probabilidad de que el sistema se encuentre en cierto estado, depende de los estados o situaciones previas, el grado de relación o el grado de dependencia que guarden los estados entre sí, se mide basándose en la *correlación*.

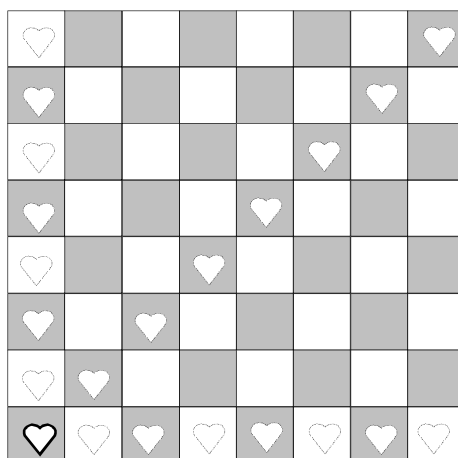
Por definición, correlación es la analogía o relación recíproca entre dos o más casos. Desde luego podemos encontrar valores autocorrelacionados, en cuyo caso, significa que



depende de sí mismo.

Por ejemplo, tomemos un tablero de ajedrez y un Rey, recuérdese que el Rey sólo puede avanzar un cuadro en cualquier dirección, esto implica que su posición depende, en gran medida, de su posición anterior, lo cual quiere decir que su posición es altamente autocorrelacionada. Supongamos que el Rey se encuentra en una esquina, en el siguiente movimiento se pueden escoger sólo 3 posibles casillas, así es que podemos predecir que después de la siguiente tirada, el Rey estará en cualquiera de estas 3 casillas.

Ahora utilicemos a la Reina en lugar del Rey y ubiquémosla en la misma esquina, la Reina se puede mover en cualquier dirección y el número de cuadros que quiera dentro del tablero. Predecir la posición de la Reina en el siguiente movimiento no es tan fácil, ya que puede escoger entre 7 cuadros en dirección vertical, 7 en dirección horizontal y otros 7 cuadros diagonalmente, por lo tanto la posición de la Reina es menos autocorrelacionada que la del Rey, pues aunque la posición de la Reina también depende de su posición anterior, esta dependencia o autocorrelación es mucho menor que en el caso del Rey.



De esta manera, los sistemas que tienen mayor correlación, son más determinísticos o por decirlo de otra forma, son más ordenados y aquellos en los que el grado de correlación es menor, tienden más al desorden.

Como ejemplo de un proceso probabilístico altamente correlacionado tenemos lo que se conoce como ruido browniano.

1.2.1.3 Ruido Browniano o Pardo

El botánico escocés Robert Brown (1773-1858) descubrió lo que ahora conocemos como *movimiento browniano* de las partículas, que es el movimiento oscilatorio de éstas, en suspensión en un líquido. Dicho movimiento, prueba la existencia de actividad al nivel molecular. El movimiento browniano, tiene la característica de ser altamente correlacionado.

El llamado ruido browniano, al igual que el movimiento con mismo nombre, presenta un comportamiento similar, también es altamente correlacionado. Hacer un modelo de este tipo de ruido es muy sencillo y se puede utilizar, por ejemplo, para generar música browniana o parda.

Una composición parda se puede generar de diferentes maneras, una sugerida por R. F. Voss [2] es la siguiente: Se escoge la primer nota al azar, supongamos la nota *La* (6, sobre el eje de las ordenadas en la gráfica de la fig. 1); a partir de esta nota se suman o se restan aleatoriamente 0, 1 ó 2 números: ejemplo (6 + 2) y se obtiene *Do*, ya se tiene la segunda nota de la melodía, la nota *Do*; se repite el procedimiento de la suma o resta de un número aleatorio entre 0, 1 y 2. A partir de *Do* se encuentra la tercer nota: ejemplo, (8 + 1)

= 9 = *Re* y así sucesivamente, ejemplo, $(9 - 2) = 7 = Si$, es la cuarta nota, hasta obtener una melodía de la duración que se quiera.

La figura 1, muestra una gráfica donde se ve el inicio de una melodía parda, el valor de las ordenadas se obtiene al sumar o restar de manera aleatoria 0, 1, ó 2 unidades (notas). El tiempo en este ejemplo se puede considerar constante.

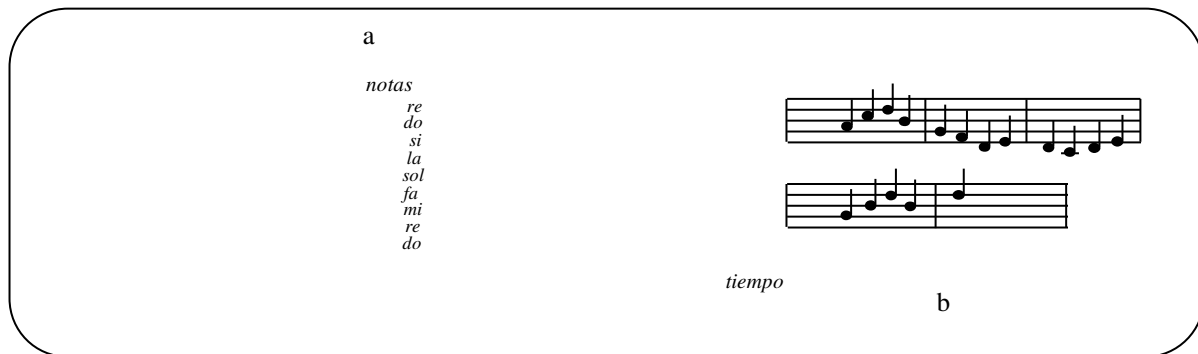


Fig. 1: (a) Gráfica en dos dimensiones de ruido pardo. (b) Partitura de esta melodía parda.

Escuchar la música compuesta de esta forma es bastante agradable, en algunos fragmentos es muy parecida a la música del vuelo del abejorro de Nikolai Rimsky Korsakov (1844-1908). Enseguida se presenta el pseudocódigo que desarrollamos basándonos en la teoría sugerida por R.F. Voss para generar música parda.

La figura 2, muestra la gráfica tomada de la pantalla de la computadora que se genera al ejecutar este algoritmo: En donde la longitud y ángulo de las líneas se determina dependiendo de la nota.

Fig.2 Gráfica que se genera con líneas que adquieren diferentes longitudes y ángulos de acuerdo a las notas de la melodía.

Otra manera de producir música parda o browniana, es mediante el siguiente algoritmo:

Tomemos siete dados de juego y lancémoslos. Súmense los valores de cada una de las caras superiores de los dados. Como se ve a continuación.

$$\text{Nota} = 5 + 6 + 1 + 6 + 2 + 1 + 3$$

El valor que resulte de dicha suma representa el número de la primer Nota de la melodía. Para saber que Nota es, basta con echar un vistazo al siguiente teclado.

No se puede mostrar la imagen en este momento.

2 4 7 9 11 14 16 19 21 23 26 28 31 33 35 38 40 43 45 47 50 52 55 57 59

1 3 5 6 8 10 12 13 15 17 18 20 22 24 25 27 29 30 32 34 36 37 39 41 42 44 46 48 49 51 53 54 56 58 60 61

Para obtener la segunda Nota, hay que escoger uno de los dados al azar y lanzarlo, el valor de la cara superior que resulte, deberá de sumarse junto con los valores que tienen los dados restantes de la tirada anterior y el resultado representa la segunda Nota de la melodía. Supóngase que es el dado que está en la cuarta posición de izquierda a derecha el que escogemos.



$$\text{Nota} = 5 + 6 + 1 + 4 + 2 + 1 + 3 = 22$$

De los dados anteriores, sabemos ahora que la segunda Nota es la número 22(La). La siguiente Nota se determina de igual manera; se toma uno de los siete dados al azar, se lanza y nuevamente se suman los valores de las caras superiores de todos los dados, habiendo cambiado tan sólo el valor de uno de ellos. Supóngase que en esta ocasión es el dado de la posición seis de izquierda a derecha el que se escoge.



$$\text{Nota} = 5 + 6 + 1 + 4 + 2 + 5 + 3$$

A continuación se presenta el pseudocódigo de este algoritmo para generar ruido browniano, con la peculiar característica de que, además de los valores que comúnmente se encuentran en cada una de las diferentes caras de los dados (1, 2, 3, 4, 5, 6), también se

puede obtener el valor de cero, es decir, como si se utilizaran dados de siete caras (0, 1, 2, 3, 4, 5, 6).

Prog. 2: Pseudocódigo para la generación de ruido browniano desarrollado por nosotros.

```
1 2 .... ... 60
notas {do, re, mi, fa, sol, la, si, do, re, mi, ..., la, si}
main()
{
  While( no ESC)
  {
    r=random(7);
    dado[r]=random(7);
    Nota=dado[0]+dado[1]+dado[2]+dado[3]+dado[4]+dado[5]+dado[6];
    sound( notas[Nota] ) ;
    delay(200); //en milisegundos
  }
  nosound();
}
```

La figura 3, es la gráfica tomada de la pantalla de la computadora, que se genera al ejecutar este algoritmo.

Fig 3 Gráfica que se genera con líneas que adquieren diferentes longitudes y ángulos de acuerdo a las notas de la melodía.

Un hecho que permitió darnos cuenta que los modelos que estábamos diseñando iban por buen camino, fue el descubrimiento de una fotografía del movimiento browniano

de una partícula de humo, que se encontró en un libro muy interesante llamado *Materia Tierra y Cielo* del renombrado científico, George Gamow(1904-1968)[3].

La figura 4 es una copia de dicha fotografía. El objetivo es que la compare con las gráficas obtenidas en la ejecución de los algoritmos para generar ruido browniano.

El ruido browniano es un ejemplo de sistemas donde a partir de eventos erráticos se obtienen resultados ordenados con un alto grado de correlación cuando la correlación es muy baja o no existe, se

Fig. 4 Posiciones sucesivas de una partícula de humo registradas en intervalos de 1 minuto.

1.2.2 Sistemas Desordenados y Ruido Blanco

1.2.2.1 Sistemas Desordenados

Cuando la correlación del sistema es muy baja o no existe, los sistemas tienden al desorden y en general no se pueden encontrar modelos para describirlos debido a que parecen no respetar ningún tipo de orden; son sistemas que por el número de variables en juego se complican tanto, que es verdaderamente imposible representarlos mediante modelos matemáticos. Tal es el caso del movimiento de las partículas de un fluido al aumentar la temperatura de éste. Al ir aumentando la temperatura llega un momento en que el movimiento de las partículas es tal, que resulta imposible predecir su posición.

Un ejemplo de procesos con un grado de correlación muy bajo es aquel que se obtiene al generar el que se denomina ruido blanco.

1.2.2.2 Ruido Blanco

El ruido blanco, es una sucesión de sonidos que no guardan ninguna correlación entre sí, cuando menos en teoría. Lo que esto significa es que todos los sonidos son completamente aleatorios, cada frecuencia de los sonidos emitidos es tomada al azar y además todas las frecuencias tienen la misma probabilidad de ser escogidas.

El ruido blanco se presenta frecuentemente en la radio y en la televisión, y se le denomina interferencia, de hecho se presenta prácticamente en todos los dispositivos de comunicación, por lo que existen múltiples investigaciones para tratar de evitar que este tipo de ruido ocasione problemas en las comunicaciones, como los filtros que se utilizan en la comunicación mediante modems.

Si se quiere hacer un modelo que produzca ruido blanco, las variables que se utilicen para generarlo tienen que ser completamente independientes, es decir, que ninguno de los valores de las variables debe guardar relación con los otros, sus valores tienen que ser completamente aleatorios, entonces, podemos decir que se habrá generado un ruido

blanco, que desafortunadamente no es puro, ya que este sólo se puede crear de un modo teórico, en la realidad no existe el ruido blanco puro.

En la figura 5, se muestra un ejemplo de ruido blanco en dos dimensiones: Se utilizan dos variables *notas* y *tiempo*, en donde *notas* representa las ordenadas y *tiempo* las abscisas. Se obtuvo escogiendo al azar entre las notas presentadas, el tiempo se ha considerado constante.

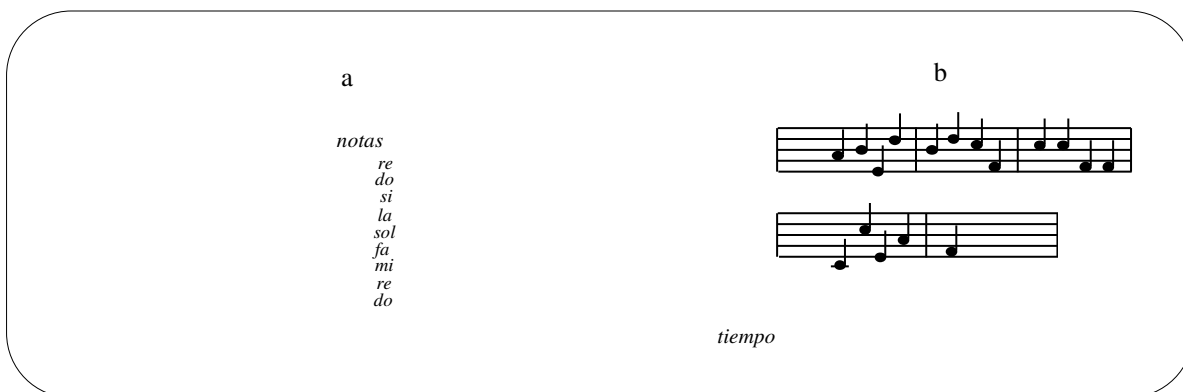


Fig. 5: (a) Gráfica en dos dimensiones de ruido blanco.
(b) partitura resultante de la melodía blanca.

Generar música blanca (ruido blanco) es muy sencillo, basta con tocar notas al azar [2].

Enseguida se establece el pseudocódigo para generar ruido blanco.

```

1 2 .... ... 60
notas {do, re, mi, fa, sol, la, si, do, re, mi,....,la,si}
main()
{
  While( no ESC)
  {
    sound( notas[ random(60) +1] );
    delay(200); /*en milisegundos
  }
  nosound();
}

```

Prog. 3: Pseudocódigo para la generación de ruido blanco.

El producto de una composición blanca es una secuencia acústica que no es muy agradable al oído humano, ya que carece de ritmo, color, melodía, etc, pues ninguna nota tiene relación con las otras y si los tiempos de duración de las notas también son totalmente aleatorios, es ruido que desagrada bastante.

Ya que se mencionó aquello de color, sería conveniente aclarar el porqué del nombre de *ruido blanco*. Todo lo que percibimos de este mundo, es debido a que está formado por ondas con diferentes frecuencias; dependiendo del rango de frecuencias, es el sentido que utilizamos para percibirlo, el oído es capaz de escuchar las frecuencias entre 20 y 20000 Hz [5] y el ojo humano percibe frecuencias entre $7.5 * 10^{14}$ Hz (violeta) y $4.2 * 10^{14}$ Hz (rojo) [10].

El mal llamado color negro, que realmente representa la ausencia de colores no origina ninguna frecuencia, el negro, no lo podemos ver. Vemos la diferencia de colores, debido a que cada uno de ellos se origina, transmite, refleja o refracta, con una frecuencia determinada, o en un momento dado, una mezcla de frecuencias.

A dicha mezcla de frecuencias o para llamarla propiamente, distribución de frecuencias, se le denomina *dimensión espectral*, y se representa mediante una función que muestra como la potencia total de la señal se distribuye entre las distintas frecuencias[6]. El color blanco, está formado por la suma de todos los colores, es decir por todas las frecuencias que podemos ver, esto significa que todas las frecuencias están presentes en el color blanco. Lo mismo ocurre en el caso del sonido, se denomina blanco porque está formado por todas las frecuencias sonoras y se le representa con una dimensión espectral = $1/f^0 = 1$, mientras que la dimensión espectral del ruido pardo o browniano es igual a $1/f^2$.

Los modelos del ruido blanco, así como el del browniano, son ejemplos que se encuentran en los extremos, en cuanto a correlación se trata, si los comparamos, podemos darnos cuenta de que son opuestos, el ruido blanco representa la ausencia de correlación y se caracteriza por ser desordenado y el ruido pardo es altamente correlacionado. Sin embargo, desde hace varios años, se han encontrado que existe otro tipo de sistemas que juegan de forma maravillosa con el orden, pues no son tan ordenados como el movimiento browniano, pero tampoco se puede decir que son desordenados. Para diferenciarlos de los unos y los otros, se les ha llamado *sistemas caóticos* y se ha descubierto que una gran cantidad de sistemas de la Naturaleza presentan un comportamiento caótico.

1.2.3 Sistemas Caóticos, Fractales y Ruido 1/f

1.2.3.1 Sistemas Caóticos

La manera que la física y la matemática nos han venido explicando los fenómenos naturales, es tratando de buscar su orden, su patrón de conducta y en general, se les ha tratado de dar explicaciones deterministas, tratándose de describir mediante ecuaciones, que aunque algunas veces son complicadas, están perfectamente definidas, cada vez que se encontraban eventos impredecibles, se les hacía a un lado y se marcaban como la excepción a la regla.

Hoy día, se descubre que son los eventos poco determinísticos, los que rigen el comportamiento de la Naturaleza y ahora que han dejado de ser la excepción para convertirse en la regla, se les ha llamado “caóticos”.

Se denomina *caos* al área encargada de estudiar a los sistemas que no respetan las reglas determinísticas, pero que tampoco se vuelven desordenados.

Los sistemas caóticos respetan un orden que a diferencia de los sistemas determinísticos no es tan evidente, un ejemplo de ello es el mezclado de pinturas; cuando se mezclan dos pinturas de diferente color, da por resultado un tercer color, este color no se podrá volver a obtener mezclando nuevamente las dos pinturas iniciales, a menos que la forma de revolverlas: el agitado, la temperatura, el tiempo, y muchas otras variables fueran exactamente iguales a las de la primera vez; como es prácticamente imposible tener las mismas condiciones, los resultados no se pueden predecir, convirtiéndose éste en un sistema caótico y, realmente, *es de admirarse el trabajo de los pintores cuando igualan colores, ya que les quedan bastante parecidos.*

En los sistemas caóticos basta una pequeña variación en las condiciones iniciales para obtener resultados completamente distintos, por ejemplo, dos árboles creciendo en la misma tierra, teniendo la misma cantidad de agua y teniendo en general los mismos cuidados, pueden ser completamente diferentes si uno de ellos recibe un poco más de sol al día. Esa pequeña diferencia basta para que mientras uno esté dando frutos, el otro apenas tenga la mitad de su tamaño, sin hablar de cuestiones subjetivas como la alegría que emana de uno comparada con la tristeza del otro.

Para modelar sistemas caóticos, regularmente se utilizan ecuaciones en las que intervienen factores de probabilidad, convirtiendo a los procesos del sistema en procesos

probabilísticos, interactuando dichos procesos dan por resultado que el comportamiento del modelo sea como el del sistema real, impredecible.

Por esta razón se piensa que modelar utilizando *caos* permite crear sistemas con la característica de comportarse como los sistemas Naturales.

La revolución conceptual que ha provocado el descubrimiento de los sistemas caóticos, empezó a finales del siglo pasado, con científicos como Hadamard Poincaré(1854-1912), pero no es hasta la llegada de las computadoras, cuando personas como Edward Lorenz pudieron plasmar sus ideas y llegar a conclusiones que desencadenaron una gran cantidad de investigaciones en múltiples áreas como la física, matemáticas, química, biología, economía, etc. dando por resultado la aparición de más caos en donde se creía que no existía[7][9].

Por otro lado, Benoit Mandelbrot se enfocó al desarrollo de una teoría geométrica a la que denominó *fractales*, aplicando sus descubrimientos a la generación de hermosos paisajes y vaya que fueron sorprendentes, pues realmente reflejaban el caos inherente a los sistemas diseñados [8].

1.2.3.2 Fractales

Entiéndase por *fractales*, los objetos que gozan de una propiedad llamada autosimilaridad, además de tener dimensiones que no responden exclusivamente al universo de los números naturales, sino que pueden ser dimensiones fraccionarias, de ahí su nombre *fractal* (*fracción*).

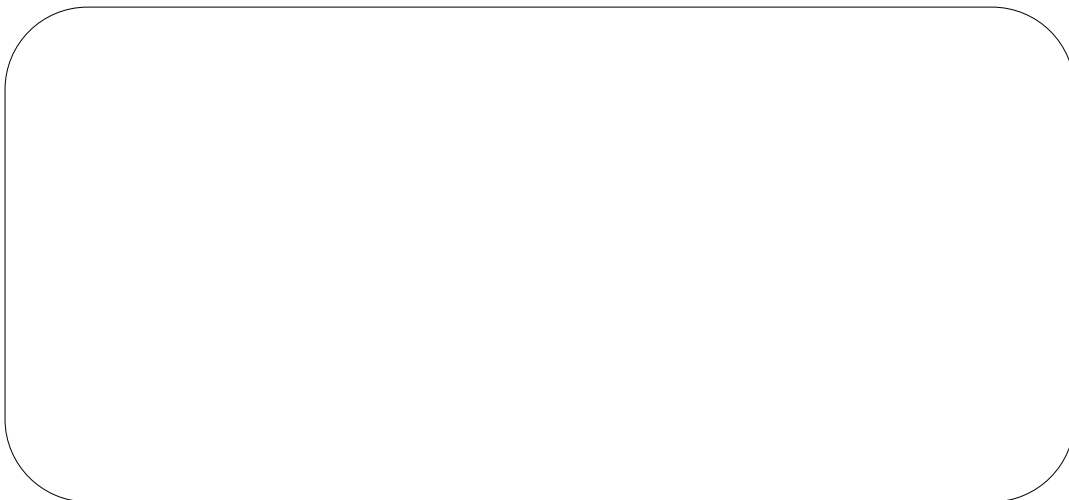
El que los fractales sean autosimilares indica que las partes que forman el fractal son similares al fractal en su conjunto; por ejemplo, si se parte una piedra dura y rasposa, sus fragmentos conservan las mismas propiedades, son rasposos y duros; lo mismo ocurre si volvemos a partir los trozos de piedra, las propiedades de piedra las van a conservar, incluso, si los fragmentos llegan a ser microscópicos, seguirán siendo piedras. Lo mismo ocurre en el sentido contrario, si juntamos dos piedras, vamos a tener una piedra más grande, si continuamos juntando piedras, obtendremos, un peñasco, un pequeño cerro, una montaña, luego una cordillera; de cualquier forma, el fractal piedra va a conservar propiedades similares independientemente de su tamaño.

Así como este ejemplo, podemos pensar en sistemas solares formando galaxias, pequeñas gotas de agua creando nubes, singulares casas construyendo ciudades o miles de otros ejemplos que podemos encontrar en la Naturaleza, ya que prácticamente todo tiene propiedades fractales.

Hablar de la dimensión fractal es hablar de cambiar, incluso, la manera en como conceptualizamos la realidad. La matemática clásica nos ha enseñado, durante mucho tiempo, que el mundo en el que vivimos posee ciertas dimensiones y hay objetos que pertenecen al mundo de una sola dimensión como es el caso de las líneas; también existen otros que viven en un mundo de dos dimensiones, tal es el caso de los cuadrados, los triángulos, rectángulos y muchas otras figuras bidimensionales. En el mundo de tres dimensiones, se encuentran objetos como esferas, cubos, poliedros, árboles, perros, leones y una lista que difícilmente acabaríamos de escribir. Hay demostraciones matemáticas que establecen la existencia de universos de n dimensiones, en donde pueden existir objetos de cuarta, quinta y un número infinito de dimensiones, e incluso puntos sin dimensión.

Sin embargo, a todos estos objetos se les ha considerado sólo de dimensiones enteras. No es hasta la aparición de los fractales que se muestra en forma normal o cotidiana objetos que poseen dimensiones fraccionarias. Los fractales poseen dimensiones fraccionarias y entonces se puede hablar de figuras con dimensiones de 1.75 o 2.34; un clásico ejemplo es la famosa figura fractal denominada *copo de nieve*.

Fig. 6: Primeros 5 niveles del fractal clásico, copo de nieve.



Aunque es cierto que estamos acostumbrados a vivir entre los fractales y realmente los percibimos de forma natural cuando vemos los árboles, las nubes o las olas del mar, no estamos acostumbrados a modelar la realidad en términos de fractales, ya que entender la propiedad fractal del Universo implica cambiar nuestra forma de pensar y la manera en la que describimos al mundo que nos rodea.

Por ejemplo, al observar los ruidos blanco y browniano desde el punto de vista fractal, nos damos cuenta que tienen un comportamiento fractal ya que si tomamos un fragmento de ruido blanco o browniano, seguiremos teniendo ruido blanco y browniano respectivamente, como se ve en las figuras 1, 2, 3, 4 y 5.

Existe otro tipo de ruido muy común en la Naturaleza que se conoce como ruido $1/f$, que también tiene la característica de presentar un comportamiento fractal.

1.2.3.3 Ruido $1/f$

El *ruido $1/f$* es una clase muy especial de ruido, para el cual no se tiene una definición concreta. Aunque no se sabe con precisión que lo produce o que representa, si es bien sabido que se encuentra en una gran cantidad de eventos del Universo.

Se caracteriza porque su densidad espectral o manera en la que se distribuyen sus frecuencias es igual a $1/f$ [6]. Haciendo una comparación de densidades espectrales, el ruido blanco tiene densidad espectral igual a $1/f^0$ y la del ruido pardo es igual a $1/f^2$, con estos valores, podemos pensar que el valor de la densidad espectral del ruido $1/f$ queda “en medio” de los valores del ruido blanco y pardo, esto también se puede interpretar como que el grado de correlación del ruido $1/f$ no es ni muy bajo ni muy alto, sino término medio, es decir, no es totalmente aleatorio, pero tampoco es altamente correlacionado.

Densidad Espectral		
$1/f^0$	$1/f$	$1/f^2$
Blanco		Pardo

Casualmente, de acuerdo a estudios realizados por Richard F. Voss[2], algunos tipos de música, como la clásica, el jazz y el rock presentan una densidad espectral alrededor de $1/f$ y un hecho, relativamente curioso, es que al observar el flujo del tránsito, el crecimiento de las ciudades, el crecimiento de los árboles, la aparición de las manchas solares, el

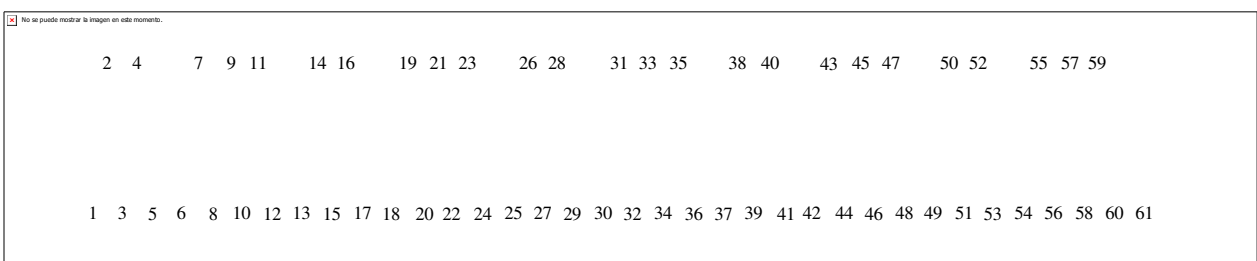
crecimiento de las montañas, el movimiento de las olas del mar, la formación de galaxias, de nubes y prácticamente todo a nuestro alrededor, presenta un comportamiento con densidad espectral alrededor de $1/f$. Parece entonces, que en vez de ser un suceso intrascendente, *el ruido $1/f$ es más bien una propiedad del Universo, una propiedad de los fractales*, pero además nos explica la belleza de los fractales, esa armonía con la que han sido creados los habitantes de este Universo. Se puede decir que todo lo que sucede no es ni muy correlacionado, ni completamente aleatorio sino $1/f$.

Es, sin duda alguna, el ruido $1/f$ un tema muy interesante. Se cree, que siendo una característica del Universo, quizá es a esta propiedad a la que imita la música, ya que hay que recordar que las bellas artes, en general imitan, ya sea al colorido mundo de la naturaleza como en el caso de la pintura o al quehacer humano como lo hace el teatro, porqué no pensar que la música imita la propiedad $1/f$ de la Naturaleza.

En caso de que esto no sea cierto, entonces la música sería, simplemente, un fenómeno más que presenta la propiedad $1/f$. De cualquier forma, al generar ruido con una densidad espectral $1/f$, es posible que sea agradable al oído humano, que sea música.

Existe una manera muy sencilla, para generar música $1/f$, sugerida por el científico Richard F. Voss[2], que consiste básicamente en lo siguiente:

Supóngase que se tienen 60 notas, numeradas en forma ascendente en un teclado, como en el que a continuación se presenta.

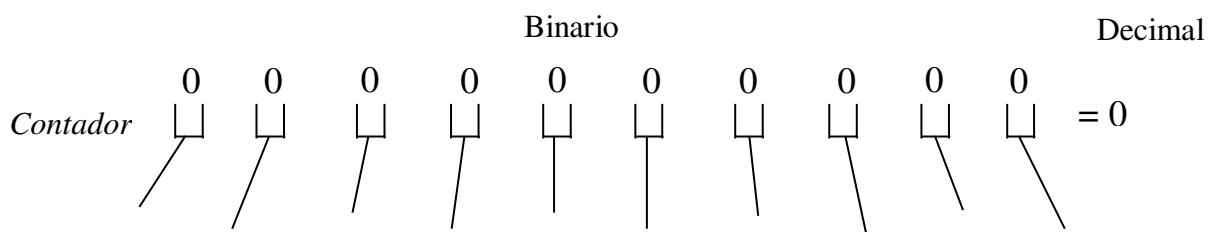


Se toma un número binario de 10 dígitos y se iguala a cero, a cada dígito se le asigna un dado. Se avientan los 10 dados, se suman y el resultado nos indica el número de la nota con la que empieza la melodía. Se incrementa el valor del número binario en la unidad. Será lanzado el dado de cada dígito que haya experimentado algún cambio de valor. Para obtener la siguiente nota de la melodía, basta sumar el valor de los dados, aquellos que

fueron lanzados probablemente cambiaron su valor, los que no, sumarán el valor que tenían a partir de la tirada anterior. Como se puede ver en el siguiente ejemplo:

Se tienen 10 dados de juego, un *contador* con incremento igual a 1, dicho *contador* será un número binario de 10 dígitos y deberá de ser inicializado a cero. Supóngase que los dados han sido lanzados, que cayeron y quedaron en la posición que se muestra a continuación, una variable, a la cual denominaremos *Nota*, irá tomando los valores de la suma de los puntos que muestran las caras superiores de cada dado.

Haciendo coincidir cada dado con cada una de las posiciones de los dígitos del

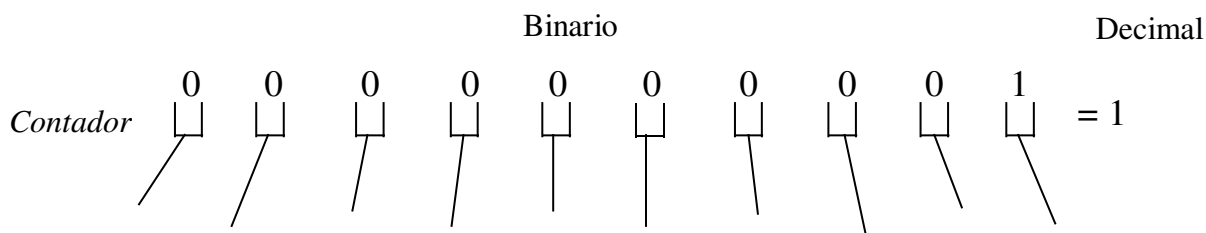


$$5 + 6 + 1 + 6 + 2 + 1 + 3 + 6 + 5 + 4$$

Nota = 39

contador. En este caso, se obtiene el valor de *Nota* = 39, que corresponde al #Re (re sostenido) que se toma como la primer nota de la melodía.

El siguiente paso, es obtener la siguiente Nota de la melodía, primeramente se incrementa el *contador*, pasando de 0 a 1, con lo que el 0 del bit menos significativo, cambia de estado a 1, siendo, en este paso, el único componente del *contador* que cambia de estado, es decir, el único de los 10 dígitos del contador, que manifestó un cambio, entonces el dado correspondiente al bit menos significativo, se lanza, mientras los dados restantes permanecen en reposo; se calcula el valor de nota, sumando el número de puntos que indique la cara superior de cada uno de los dados. Si por ejemplo, el número de puntos de la cara superior del dado lanzado fue igual a 6, entonces *nota* = 41 (Fa), siendo ésta la siguiente nota de la melodía.



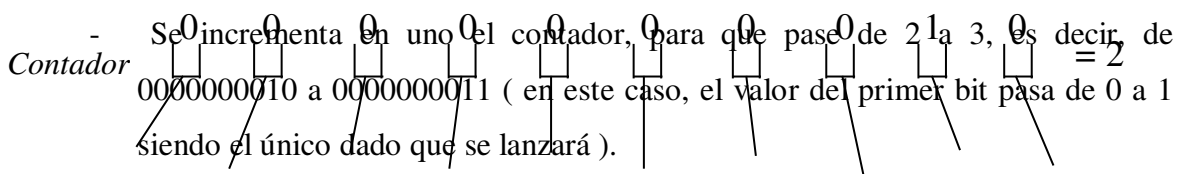
$$5 + 6 + 1 + 6 + 2 + 1 + 3 + 6 + 5 + \underline{6}$$

Nota = 41

El siguiente incremento del contador, hará que sus dígitos pasen de 000000001 a 000000010. Los dos bits menos significativos cambian de estado, el bit menos significativo pasa de 1 a 0 y el segundo bit menos significativo de 0 pasa a 1, luego los dados correspondientes a estos bits se toman y se lanzan, mientras los dados restantes permanecen sin cambio, supongamos que los puntos que muestran los dados ahora son 3 y 1, se repite el procedimiento para calcular el valor de *nota* que ahora es igual a 34, que será

la tercer nota de la melodía.

Para obtener la siguiente nota hay que seguir el mismo procedimiento: Decimal



$$5 + 6 + 1 + 6 + 2 + 1 + 3 + 6 + \underline{3} + \underline{1}$$

Nota = 34

- Se lanzan todos los dados cuyo dígito haya experimentado un cambio de estado.
- Se suman los valores de los 10 dados, y se obtiene la nueva *Nota*.

Y el proceso se repite indefinidamente. La figura 7 muestra algunas notas generadas de esta manera.

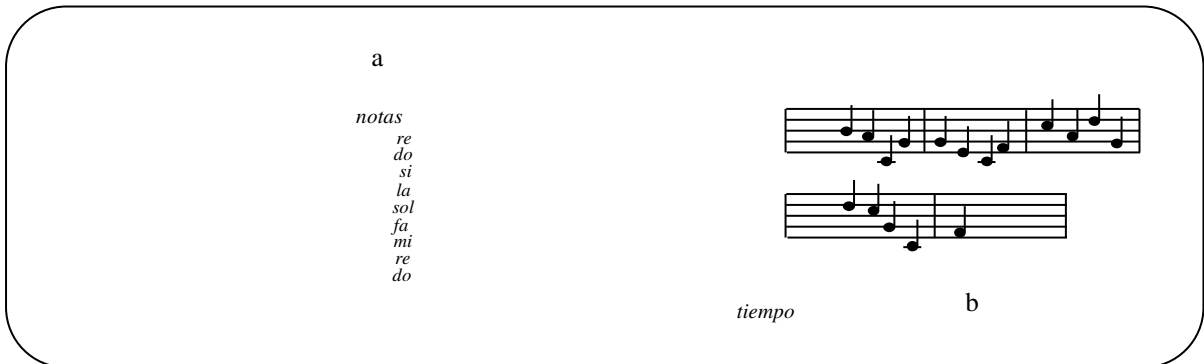


Fig. 7: (a) Gráfica en dos dimensiones de ruido 1/f. (b) Partitura de ruido 1/f.

Y enseguida se lista el pseudocódigo para generar música 1/f que desarrollamos basándonos en el algoritmo de Voss.

Prog. 4: Pseudocódigo para la generación de ruido 1/f.

```

1 2 .... ... 60
notas {do, re, mi, fa, sol, la, si, do, re, mi, ..., la, si}
avienta_dado [10]
numero_binario = 0000000000
numero_binario2 = 1111111111
main()
{
  While( No Esc )
  {
    for( nn=0; nn<=9; nn++ )
    {
      if( bit nn de numero_binario != bit nn de numero_binario2 );
      avienta_dado[nn] = random(7);
      Nota += avienta_dado[nn];
    }
    sound( notas[Nota]);
    delay(200); //en milisegundos
    numero_binario2 = numero_binario;
    numero_binario ++;
  }
  nosound();
}

```

2.0 SISTEMA EVOLUTIVO GENERADOR DE MÚSICA

2.1 INTRODUCCIÓN A LOS SISTEMAS EVOLUTIVOS

En las páginas anteriores, se ha descrito como generar música parda, blanca y 1/f. La música obtenida a partir de estos ruidos, es ya agradable, sobre todo la música 1/f, sin embargo, no basta con utilizar estas técnicas para desarrollar un verdadero sistema compositor. Supongamos que se desarrollan diferentes algoritmos que hagan composición musical, es posible, que en un momento dado, se encuentre alguno que genere música agradable, sin duda alguna, dicho algoritmo presentaría una densidad espectral alrededor de 1/f. Pero, un primer problema es que existen una cantidad enorme de algoritmos para generar música, de esta manera, encontrar uno que genere música “humana”, puede tomar algo de tiempo. Por otro lado, se van a tener que encontrar tantos algoritmos como diferentes géneros de música existen y ¿qué tal si quisiéramos que el sistema compusiera música parecida a las composiciones hechas por algún autor en especial?, esto, implica que cada autor debe de tener su propio algoritmo, lo cual, vuelve a ésta, una tarea titánica.

¿Porqué no permitir que sea el mismo sistema el que encuentre las reglas de composición, el algoritmo a seguir?

Con lo cual ha llegado el momento, para hablar de un poco de magia, para hablar de los *Sistemas Evolutivos*.

Resulta que uno de los problemas que comúnmente se encuentran en el desarrollo de sistemas, es encontrar las reglas de operación, muchas veces, esta búsqueda no es tan trivial, por ejemplo ¿cómo encontrar las reglas de composición del maestro Nicolo Paganini? Los sistemas evolutivos están diseñados para encontrar las reglas del medio que los rodea y aplicarlas para resolver los problemas. Un sistema evolutivo tiene la capacidad de aprender cotidianamente; a un sistema tal se le pueden dar ejemplos de música de Paganini y encontrará las reglas con las cuales compuso dicho autor, una vez que se hayan encontrado, el sistema podrá componer música basándose en ellas, como el mismo Paganini lo haría.

La característica principal de un *sistema evolutivo* está en que es un sistema capaz de aprender permanentemente de su medio ambiente, por ejemplo, los seres humanos

somos sistemas evolutivos de lo más sofisticados que hay, un niño desde que nace empieza a conocer y a aprender del medio que lo rodea. Uno de los primeros elementos que aprende, es a reconocer quien es su madre y dado que la madre se viste cada día diferente, usa peinados distintos, se pinta de diferente manera y toda una serie de factores que cambian día con día, el niño va evolucionando; encuentra un valor medio de quien es su madre y aprende a reconocerla, se vista como se vista o se peine como se peine. Lo mismo ocurre cuando aprende a hablar, primero reconoce los diferentes tonos de voz y pese a que no sabe “Lenguaje Natural” reconoce perfectamente cuando su madre habla con tono de enojo y cuando está contenta; ya cuando domina ciertas palabras entonces todo lo que no conoce: lo inquiera, pregunta qué significa, permitiendo que la base de conocimientos aumente y entonces se encuentra en el camino en el que todos estamos, aprender a hablar, unos saben más y unos menos pero el proceso de aprendizaje es el mismo y permanente.

Esa es la forma en la que los sistemas evolutivos trabajan, aprenden continuamente las reglas del medio que los rodea y las aplican para resolver problemas.

2.2 SISTEMA COMPOSITOR

Para ver como se aplican los sistemas evolutivos en el desarrollo de un sistema compositor, *primero presentaremos el algoritmo base que usaremos para generar música* y luego veremos la componente evolutiva.

Primero: Utilizaremos sólo siete notas ordenadas, Do, Re, Mi, Fa, Sol, La, Si; no implica esto que se limite a siete notas, ya que pueden ser n escalas y sólo es con el fin de hacer más sencilla la explicación.

Segundo: Con estas siete notas, construiremos una matriz cuadrada, de columnas Do, Re, Mi, Fa, Sol, La, Si e igualmente renglones Do, Re, Mi, Fa, Sol, La, Si.

Tercero: Llenamos algunas casillas de la matriz con números generados en forma aleatoria, por lo que la llamamos *matriz evolutiva aleatoria*. Por ejemplo, los números diferentes de cero que tiene la siguiente matriz, han sido escogidos aleatoriamente.

Matriz Evolutiva Aleatoria

	Do	Re	Mi	Fa	Sol	La	Si
Do	0	0	90	0	30	40	0

Re	10	0	15	45	20	70	80
Mi	5	25	0	60	0	30	90
Fa	0	80	10	40	15	95	30
Sol	25	15	0	40	65	70	0
La	0	35	5	10	0	0	0
Si	20	30	0	0	60	0	70

Nota: Si en cada renglón de la matriz se divide el valor de cada casilla entre el Total del renglón, entonces lo que obtenemos es una matriz de probabilidades y lo que representa es la probabilidad de pasar de una nota a otra.

Por ejemplo, si tomamos el renglón del Do

	Do	Re	Mi	Fa	Sol	La	Si
Do	0	0	90	0	30	40	0

El total vale $90+30+40 = 160$. Si dividimos el renglón entre 160, queda:

	Do	Re	Mi	Fa	Sol	La	Si
Do	0	0	9/16	0	3/16	4/16	0

La probabilidad de pasar de Do a Fa es cero y de Do a Mi es 9/16. O sea que, este algoritmo se basa en una matriz de probabilidad, aunque para propósitos prácticos, seguiremos trabajando con los valores enteros.

Cuarto: Agregamos una columna llamada Total, que originalmente está llena de ceros, a la derecha en la matriz evolutiva aleatoria.

Quinto: En cada renglón vamos a hacer un recorrido de izquierda a derecha ignorando las casillas con valor cero.

El primer número (más a la izquierda) diferente de cero, se suma a Total y el resultado sustituye al primer número.

El segundo número diferente de cero se suma a Total y el resultado se sustituye en

el segundo número.

Así sucesivamente con todos los valores diferentes de cero, el *i*-ésimo número diferente de cero se suma al Total y el resultado sustituye al *i*-ésimo número. Agotados estos, la suma debe quedar almacenada en la columna Total del renglón. Por ejemplo, el renglón original es

	Do	Re	Mi	Fa	Sol	La	Si	Total
Do	0	0	90	0	30	40	0	0

Y al aplicar el algoritmo queda:

					90 +30	120+40	90+30+40	
	Do	Re	Mi	Fa	Sol	La	Si	Total
Do	0	0	90	0	120	160	0	160

En esta forma se obtiene la que llamaremos *matriz evolutiva de frecuencias acumuladas*

Matriz Evolutiva de Frecuencias Acumuladas

	Do	Re	Mi	Fa	Sol	La	Si	Total
Do	0	0	90	0	120	160	0	160
Re	10	0	25	70	90	160	240	240
Mi	5	30	0	90	0	120	210	210
Fa	0	80	90	130	145	240	270	270
Sol	25	40	0	80	145	215	0	215
La	0	35	40	50	0	0	0	50
Si	20	50	0	0	110	0	180	180

Sexto: Empecemos a componer la primer melodía; tómease al azar una nota, *Fa* por ejemplo, será la primer nota de la melodía; se toma el valor de la columna *Total* del renglón *Fa*, en este caso 270 (véase la matriz de frecuencias acumuladas) y se genera un número aleatorio entre cero y este número. El número aleatorio así obtenido, se compara en orden de izquierda a derecha, con los valores del renglón *Fa*; al llegar a un valor que sea mayor o igual al número aleatorio, se toma la nota de la columna como la nota segunda de la melodía. Por ejemplo, si se obtiene el número 157, como este es mayor que 145 (nota *Sol*) y menor que 240 (nota *La*), se toma la nota *La* como la segunda nota de la melodía, llevando *Fa, La,...*

Séptimo: Repetimos el procedimiento anterior: sólo que ahora, se toma el renglón de la nota anterior *La*, que tiene *Total* = 50, por lo tanto, se genera un número aleatorio entre 0 y 50; por ejemplo el 18, 35 es mayor que 18, luego *Re* es la tercer nota de la melodía y así va *Fa, La, Re,....* Para obtener la 5^a, 6^a y n-ésima nota, basta repetir el mismo procedimiento:

- Se genera un número aleatorio entre cero y el valor de la columna *Total* del renglón de la última nota obtenida.
- Se compara el número así obtenido con los valores de ese renglón, hasta que alguno sea mayor o igual. La columna en la que se encuentre este valor, representa la siguiente nota de la melodía.

En este ejemplo, se han utilizado matrices de sólo 7 notas, basta ampliarlas hasta el número de notas sobre las que se desee componer y efectuar el mismo procedimiento. Ésta, es una forma muy interesante de generar música, además de muy sencilla. Como se puede ver, el sistema compone basándose en ciertos valores que en el ejemplo, se generaron aleatoriamente, pero si queremos que refleje el estilo de algún compositor, el problema es encontrar los valores apropiados de la matriz evolutiva aleatoria, de manera tal que las melodías suenen agradables.

Encontrar dichos valores podría ser una tarea no muy sencilla, por lo tanto es conveniente que sea el mismo sistema el que averigüe las probabilidades apropiadas, aquí

es donde interviene la magia de los Sistemas Evolutivos. Por esta razón, a la matriz que se obtuvo mediante el proceso anteriormente descrito se le denomina *Matriz Evolutiva* y representa el corazón del *Sistema Evolutivo*.

2.3 FUNCIONAMIENTO DEL CORAZÓN DEL SISTEMA EVOLUTIVO

Hasta el momento, sólo hemos descrito como interpretar una melodía, compuesta con una matriz que fue llenada con números aleatorios y el resultado es una composición de ruido.

Supongamos que quisiéramos que el sistema componga música de un cierto tipo, más aún, que componga como algún autor en específico, digamos el maestro Niccoló Paganini (1782-1840).

Para lograr componer música como Niccoló Paganini, un ser humano tiene que dedicar “largas horas” estudiando la música de este autor, interpretando una y otra vez cada melodía. El momento llegará, en que el músico “sienta” la música y sea capaz de componer con influencia de Paganini. De igual manera podemos enseñar al sistema a componer. Démosle ejemplos de música de Niccoló Paganini, que analice las melodías y que encuentre las reglas con las que compuso, es decir que conozca su música para que después sea capaz de componer como el mismo Paganini lo haría.

Cada compositor, tiene un estilo muy personal de componer, esto lo refleja en la forma que acomoda las notas de las melodías, esto quiere decir, que a gusto propio uno encuentra agradable o no cierta sucesión de notas y mientras que a alguien le puede gustar escuchar Do, Mi, Sol, La, a alguien más le puede gustar Do, Sol, Mi, La. Esta característica, nos permite escuchar la diferencia de estilos y nos permite reconocer a tal o cual autor.

Si analizamos nota por nota las diferentes composiciones de cada autor, encontraremos que ciertas notas son tocadas de manera más o menos regular, por mejor decir, la probabilidad de que Mi aparezca después de Do varía entre un autor y otro, de esta manera, podemos hacer la lectura de diferentes melodías de algún autor y podemos contar cuantas veces utiliza Mi después de Do, Fa después de Re, Si después de Do y en fin, analizar todas las notas de manera que tengamos una lista en la que se encuentren todas las

notas que han sido utilizadas, además de saber que notas aparecen, con mayor regularidad, después de otras.

Para que el sistema pueda componer una melodía basándose en las reglas de Niccoló Paganini, significa que hay que encontrar los valores de la matriz evolutiva aleatoria adecuados, de manera que las composiciones hechas por el sistema reflejen características propias de dicho autor, en otras palabras, que se genere música similar a la que el mismo Paganini componía.

Por lo que a continuación le daremos al sistema el primer ejemplo de la música que quisiéramos que componga y que empiece a contar, sea en esta ocasión un breve ejercicio del maestro Niccoló Paganini.

Basándonos en el siguiente teclado,

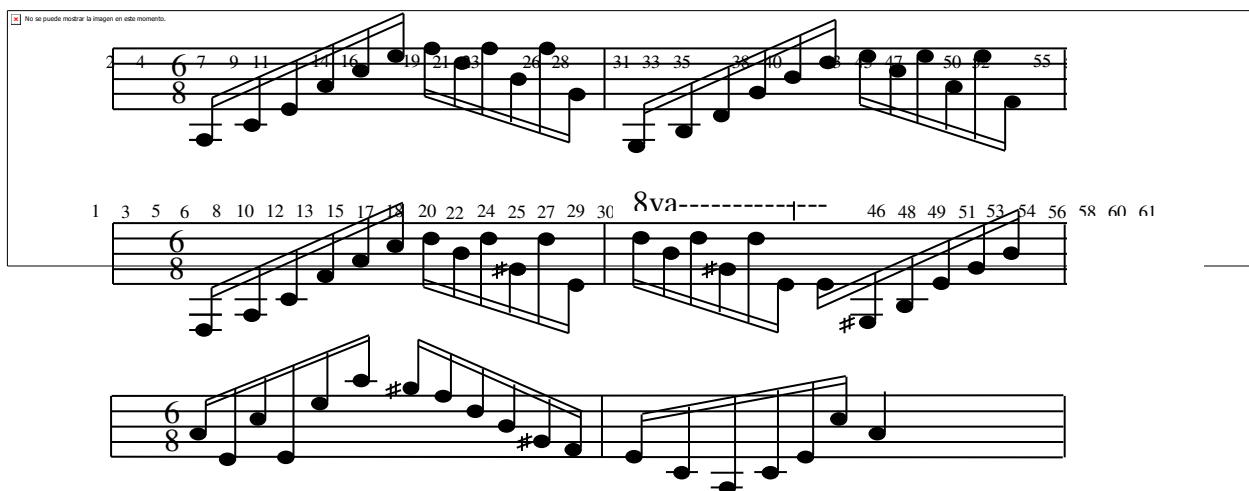


Fig. 8: Fragmento de la obra Los 24 Caprichos de Niccoló Paganini.

los números correspondientes a las notas de esta composición son los siguientes:

10(La),13(Do),17(Mi),22(La),25(Do),29(Mi),30(Fa),27(Re),30(Fa),24(Si),30(Fa),
20(Sol),8(Sol),12(Si),15(Re),20(Sol),24(Si),27(Re),29(Mi),25(Do),29(Mi),22(La),29(Mi),
18(Fa),6(Fa),10(La),13(Do),18(Fa),22(La),25(Do),27(Re),24(Si),27(Re),21(#Sol),27(Re),
17(Mi),15(Re),12(Si),15(Re),9(#Sol),15(Re),5(Mi),17(Mi),9(#Sol),12(Si),17(Mi),21(#Sol),
24(Si),22(La),17(Mi),25(Do),17(Mi),29(Mi),34(La),33(#Sol),30(Fa),27(Re),24(Si),20(Sol),
18(Fa),17(Mi),13(Do),9(#Sol),13(Do),17(Mi),25(Do),22(La).

A continuación, generaremos la matriz evolutiva aleatoria, para lo cual partiremos de una matriz completamente llena de ceros.

Debido a las características de la melodía, tendríamos que utilizar una matriz de 60 columnas por 60 renglones, sin embargo, ya que el espacio en la página no es suficiente, sólo aparecerán las notas con las que estemos tratando en ese momento. De esta forma, podemos empezar por leer la primer nota de la partitura.

Iremos modificando la matriz evolutiva aleatoria de la siguiente manera:

Se fija un valor de incremento cualquiera, sea 1 en este caso.

Se lee la primer nota de la partitura, esta marca el renglón en el que empezaremos a llenar la matriz, en este caso es La (10), que se encuentra en ceros. Se lee la segunda nota, Do(13); apuntada en la columna 13 y en encuentro con el renglón 10, la casilla adquiere el valor igual al incremento (1), esto quiere decir que se ha aumentado la posibilidad de que la nota Do(13) sea tocada después de la nota inicial La (10).

	#La	Si	Do	#Do	Re
	11	12	13	14	15
Sol 8	0	0	0	0	0
#Sol 9	0	0	0	0	0
La 10	0	0	1	0	0
#La 11	0	0	0	0	0

Si 12	0	0	0	0	0
Do 13	0	0	0	0	0
⋮					

Ahora que se ha aumentado la posibilidad de que la nota Do (13) sea tocada después de la nota La (10), se utiliza la nota Do como el número del renglón que se ha de modificar. Es decir, en el renglón de Do (13), la segunda nota leída, se incrementará el valor de la columna que corresponde a la tercer nota leída Mi (17).

	Re	#Re	Mi	Fa	#Fa
	15	16	17	18	19
⋮					
#La 11	0	0	0	0	0
Si 12	0	0	0	0	0
Do 13	0	0	1	0	0
#Do 14	0	0	0	0	0
Re 15	0	0	0	0	0
#Re 16	0	0	0	0	0
⋮					

Se lee la cuarta nota y se aumenta el valor correspondiente a la casilla que se encuentra entre el renglón de la nota anterior y la columna de esta nota. Este proceso se repite hasta que termine la melodía.

	Sol	#Sol	La	#La	Si
	20	21	22	23	24
⋮					

Re 15	0	0	0	0	0
#Re 16	0	0	0	0	0
Mi 17	0	0	1	0	0
Fa 18	0	0	0	0	0
#Fa 19	0	0	0	0	0
Sol 20	0	0	0	0	0
⋮					

Después de haber leído toda la partitura, se tiene la matriz lista para calcular la matriz de frecuencias acumuladas, totales y finalmente permitir que el sistema produzca su primera composición. La matriz que presentamos en la figura 9, muestra cómo quedan los valores de las diferentes notas después de haber aprendido el fragmento de Paganini. Debido a que la matriz que utiliza el sistema es de 60 renglones por 60 columnas, no fue posible incluir toda la información contenida en la matriz, sin embargo sólo hemos quitado algunos renglones llenos de ceros (casillas vacías). De esta manera, la matriz se va modificando conforme se le vayan presentando ejemplos.

En la figura 10 se presenta la partitura de una composición producida por el sistema, después de haberle enseñado 4 ejemplos de Paganini.

En este capítulo, hemos descrito el proceso de aprendizaje que implica la generación y modificación de las matrices evolutivas. Finalmente se obtuvo la partitura de una composición musical generada por el sistema. Sin embargo, la evolución es la capacidad de adaptación al medio, es decir, la capacidad que tiene un sistema para modificarse de acuerdo a la interrelación que guarda con el medio que lo rodea. Desde luego, al estar interrelacionados los sistemas en algún medio ambiente, provocarán procesos de coevolución [9], así mismo el medio en el que están inmersos sufrirá los efectos de la evolución, convirtiendo al proceso de evolución en un proceso Universal.

La manera en la que un sistema se adapta al medio que lo rodea es a través de dos procesos, uno de aprendizaje y otro de aplicación. Mediante el proceso de aprendizaje, un sistema evolutivo aprende a través de ejemplos. Utilizando el proceso de aplicación el sistema se modifica a sí mismo de acuerdo a lo aprendido, es decir se enseña a sí mismo.

Ambos procesos interactúan cotidianamente, lo cual significa que el sistema aprende y aplica de manera permanente, en otras palabras, siempre está evolucionando, adaptándose al medio que lo rodea[12].

Aplicando lo dicho anteriormente, hemos construido un Sistema Evolutivo Compositor de Música, que permite que se le den ejemplos musicales, de los cuales aprende las reglas con los que se compusieron y las emplea para componer. El sistema aplica y aprende tanto del exterior como a partir de lo que aplica indefinidamente.

En el siguiente capítulo describiremos la manera en la que este sistema funciona, profundizando en los detalles técnicos de los procesos de aprendizaje y de aplicación.

3. FUNCIONAMIENTO Y CONSTRUCCIÓN DEL SISTEMA

El sistema que se ha diseñado sobre la base de lo expuesto anteriormente, tiene la estructura que se muestra en el *diagrama de flujo de datos* de la figura 11.

Los procesos que forman parte del sistema son los algoritmos que generan los diferentes ruidos de los que se habló en el Capítulo I Introducción a la Música por Computadora, el algoritmo generador de música que utiliza una matriz evolutiva, expuesto en el capítulo II Sistema Evolutivo Generador de Música y el proceso mediante el cual el sistema puede aprender a través de ejemplos. A continuación se describirá el funcionamiento de este Sistema.

a. Rutina Piano

Primeramente se construye un arreglo que contenga los valores de las frecuencias de las notas musicales, esto se logra mediante una rutina que se ha denominado Piano (). Básicamente esta rutina hace lo siguiente:

Define un arreglo, al que hemos llamado Notas[], en donde inicialmente se almacenan los valores de las frecuencias de una escala cromática Do (65), #Do (69), Re (73), #Re (77), Mi (82), Fa (87), #Fa (92), Sol (98), #Sol (103), La (110), #La (116) y Si (123), (los números que aparecen después de las notas son sus frecuencias en ciclos/seg).

Notas[]={65, 69 73, 77, 82, 87, 92, 98, 103, 110, 116, 123}

Las frecuencias de las notas tienen la propiedad de que el doble de su valor, representa la misma nota, sólo que una octava o escala más arriba, por ejemplo, *Do* tiene una frecuencia de 65 ciclos/seg. Si multiplicamos este valor por 2 obtenemos el valor de la frecuencia a 130 c/s, esto es un *Do* una escala más agudo, en la tabla 1 se pueden ver los valores de las frecuencias de 5 escalas cromáticas:

Tabla 1 Frecuencias de las notas musicales en ciclos/seg.

Do	#Do	Re	#Re	Mi	Fa	#Fa	Sol	#Sol	La	#La	Si
65	69	73	77	82	87	92	98	103	110	116	123
130	138	146	154	164	174	184	196	206	220	232	246
260	276	292	308	328	348	368	392	412	440	464	492
520	552	584	616	656	696	736	784	824	880	928	984
1040	1104	1168	1232	1312	1392	1472	1568	1648	1760	1856	1968

De esta forma, la rutina Piano (), solamente necesita ir duplicando los valores iniciales del arreglo Notas [], hasta obtener 5 escalas; desde luego, se pueden obtener tantas escalas como se desee, pero por razones prácticas sólo se generan 5 escalas, ya que frecuencias más altas difícilmente se escuchan mediante la bocina de la PC.

Una vez que la rutina Piano() ha obtenido las 5 escalas deseadas, se ha creado un

Prog. 5 : Pseudocódigo de la rutina Piano(), que genera, en el arreglo Notas[], las frecuencias de cinco escalas cromáticas.

```

do, #do, re, #re, mi, fa, #fa sol, #sol, la, #la, si
notas[]={65, 69, 73, 77, 82, 87, 92, 98, 103, 110, 116, 123}

Piano()
{
  x = 1;
  z = 1;
  while( x <= 32 )
  {
    for( i=0; i<12; i++ )
    {
      notas[z] = notas[i] * x;
      z++;
    }
    x*=2;
  }
}

```

piano virtual, en donde ya se puede interpretar una melodía.

b. Lee Matriz Evolutiva Aleatoria (Rutina InArchivosMatriz)

El paso siguiente que da el sistema es abrir un archivo, que representa la memoria a largo plazo del sistema y vacía su información en un arreglo bidimensional llamado Bethoven1 [70][70], que es la Matriz Evolutiva Aleatoria del sistema y representa su memoria a corto plazo. Este proceso se lleva a cabo mediante una rutina llamada InArchivosMatriz().

Prog. 6 : Pseudocódigo de la rutina InArchivosMatriz(), que copia en el arreglo Bethoven1[70][70], la Matriz Evolutiva Aleatoria, almacenada en el archivo matriz.evl.

```
InArchivosMatriz()
{
    /* Abre el archivo matriz.evl, en donde se almacena la
       Matriz Evolutiva Aleatoria */
    in = fopen( matriz.evl, "rb+");

    /*Copia la Matriz Evolutiva Aleatoria al arreglo Bethoven1[70][70]
       de 70 columnas por 70 renglones */
    fread(Bethoven1,70 * 70 * sizeof(unsigned long),1,in);
}
```

Una vez ejecutada esta rutina, el sistema entra en un ciclo, del cual sólo podrá salir hasta que el usuario presione la tecla Esc.

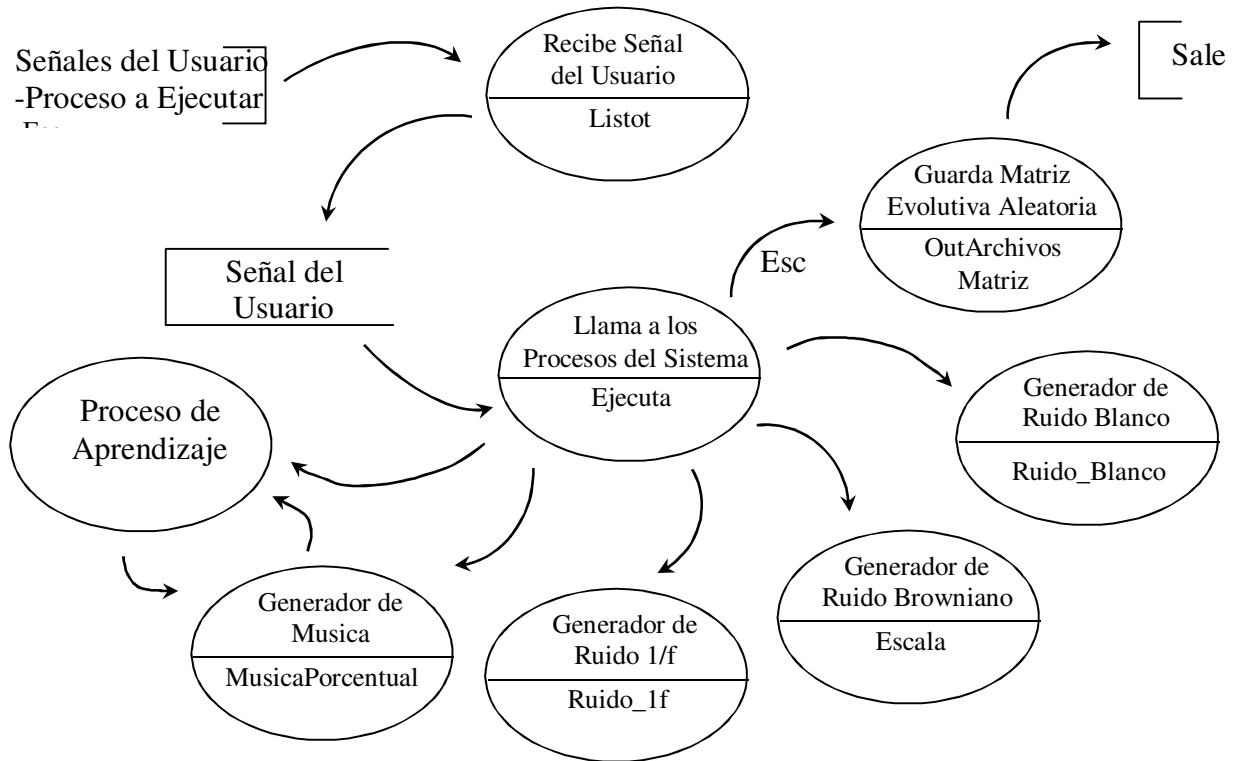
Este ciclo está compuesto básicamente de dos rutinas y representa lo que hemos denominado el Administrador de los Procesos.

3.1 ADMINISTRADOR DE LOS PROCESOS

El Administrador de los Procesos de este sistema, se compone principalmente de dos rutinas, una que se llama Listot() que se encarga de estar pendiente de lo que el usuario indique a través del teclado de la computadora y la rutina Ejecuta() que se encarga de ejecutar tal o cual proceso basándose en la instrucción recibida.

Los procesos que la rutina Ejecuta() puede llamar son 6, uno para salir del sistema, 3 generadores de ruido, uno de blanco, uno de browniano, y otro de 1/f, un generador de música y un proceso de aprendizaje que está íntimamente ligado con el proceso de generación de música, que *modifica en tiempo real la memoria a corto plazo del sistema, característica que permite que el sistema sea inteligente, pues no sólo aprende sino además evoluciona.*

A continuación se muestra el *diagrama de flujo de datos* del Administrador de los Procesos y los procesos del sistema (nótese, que las rutinas Listot() y Ejecuta() se manejan en forma concurrente).



c. Rutina Listot

El sistema debe estar atento, en todo momento, a las señales de entrada del usuario, por esta razón a esta rutina la hemos denominado Listot(). Será la encargada de recibir las señales y almacenarlas, ésta es la única función que le corresponde y tiene que ser una rutina extremadamente pequeña. Listot() deberá de estar en todas partes del programa, esto es, independientemente de las instrucciones que se estén ejecutando en el momento, por lo que se instala de tal forma que *se dispara mediante interrupciones al sistema*, si hay una señal de entrada, Listot() debe de ser capaz de recibir la señal.

```

Listot()
{
  Si usuario presiona tecla      /* Si alguna tecla es presionada,
  caracter = tecla                /* su valor es almacenado en la
}                                  /* variable carácter.
  
```

Prog. 7: Pseudocódigo de la rutina Listot(). Hay que resaltar que mientras menos funciones tenga esta rutina, mejor.

d. Rutina Ejecuta

La segunda rutina que forma parte del administrador de procesos es Ejecuta() y es la rutina encargada de ejecutar las acciones de las que es capaz el sistema. La única misión, o más bien la más importante de Ejecuta(), es asignar a cada señal, el proceso que le corresponde y ejecutarlo. Esto quiere decir, que cada vez que Listot() detecte una señal de entrada por el teclado, será la rutina Ejecuta() la que se encargue de seleccionar el proceso indicado a ejecutar. Cuando el usuario presione la tecla de Esc, el sistema saldrá, no importando que proceso se esté ejecutando. En caso de que la señal de entrada recibida sea un dato desconocido, causa que el sistema simplemente lo ignore. Es posible desarrollar un sistema evolutivo que aprenda lenguaje natural y que cada vez que encuentre una señal desconocida simplemente pregunte de qué se trata, pero bien podría ser éste, tema para otro trabajo.

```
Ejecuta()
{
  switch(caracter)
  {
    case '1': ruido browniano;      /* Compara la tecla presionada
    case '2': ruido 1/f;            /* con los distintos procesos del
    case '3': matriz evolutiva;    /* sistema, si es desconocida, la
    case '4': ruido blanco;        /* señal es ignorada.
    case 'Esc': Salir del Sistema;
    case 'a': Proceso de Aprendizaje;
  }
}
```

Prog. 8: Pseudocódigo de la rutina Ejecuta().

Debido a que los procesos para la generación de ruido que la rutina Ejecuta() puede llamar, ya se han descrito anteriormente, nos enfocaremos a describir con más detalle la música compuesta mediante la matriz evolutiva y el proceso que utiliza el sistema para aprender a partir de los ejemplos que el usuario proporcione codificados en el formato musical descrito en el Anexo A.

e. Rutina Aprende

El proceso de aprendizaje de este Sistema es una de sus principales características, es la puerta que lo pone en contacto con el medio ambiente y gracias a ella, *el Sistema puede aprender y evolucionar de manera permanente, aumentando su capacidad para adaptarse al medio que lo rodea.*

Este proceso de aprendizaje comienza cuando la rutina Ejecuta() llama a la rutina Aprende(). Esta, es la encargada de llenar la Matriz Evolutiva Aleatoria, incrementando la posibilidad de que cada nota leída del arreglo sea tocada, tal y como se hizo con el ejemplo del fragmento de Paganini, en el tema Funcionamiento del Corazón del Sistema Evolutivo, del Capítulo II Sistema Evolutivo Generador de Música.

La forma en la que la rutina Aprende() funciona, es registrando en el arreglo Bethoven1[70][70] de 70 renglones por 70 columnas, las notas que aparecen en cada melodía proporcionada como ejemplo. De esta manera, se modifica la Matriz Evolutiva Aleatoria, que originalmente fue cargada con los valores almacenados en el archivo matriz.evl, mismo en el que se almacenarán los cambios evolutivos que sufra la matriz, a lo largo del tiempo que el Sistema se esté ejecutando.

f. Genera Matriz Evolutiva de Frecuencias Acumuladas (Rutina EscalaPorcentual)

```

Aprende()
{
    Usuario selecciona el nombre de la melodía a aprender;           //Cuando la rutina ejecuta() llama a la
                                                                    //rutina aprende(), el sistema permite que
                                                                    //el usuario seleccione alguna melodía.

    renglón = primer nota de la melodía;                             //La primer nota de la melodía nos indica el
                                                                    //primer renglón que registrará un cambio.

    While(no se acabe la melodía)
    {
        nota = siguiente nota de la melodía;                       //El sistema leerá todas las notas hasta que se acabe la
        Bethoven1 [renglón] [nota] += 1;                           //melodía. Con cada nota leída, se incrementará la
        renglón = nota;                                             //probabilidad de volver a tocar esa nota, en la Matriz
                                                                    //Evolutiva Aleatoria.
    }
}

```

Prog. 9: Pseudocódigo de la rutina Aprende().

Una vez que la rutina Aprende() ha terminado de leer el ejemplo, es el turno de ejecutar otra rutina llamada EscalaPorcentual(), que genera la Matriz Evolutiva de Frecuencias Acumuladas a partir de los valores almacenados en la Matriz Evolutiva Aleatoria (arreglo bidimensional Bethoven1[70][70]).

Esta rutina, primeramente copia la información almacenada en el arreglo Bethoven1[70][70] al arreglo Bethoven[70][70], que también es de 70 columnas por 70 renglones y que representa la Matriz Evolutiva de Frecuencias Acumuladas.

De esta matriz sólo se utilizan 60 columnas y 60 renglones para almacenar el incremento de las probabilidades de cada nota y la columna 62 se utiliza para almacenar los Totales de cada renglón.

Prog. 10: Pseudocódigo de la rutina EscalaPorcentual().

```
EscalaPorcentual()
{
for( a1 = 1; a1 <= 60; a1++ )           //Copia el contenido de la matriz Bethoven1
  for( a2=1; a2 <= 60; a2++ )           //en la matriz Bethoven.
    Bethoven [a1] [a2] = Bethoven1 [a1] [a2];

for ( a1 = 1; a1 <= 60; a1++ )           //Se suman los porcentajes de cada renglón y la suma
  {                                       //se almacena en la columna 62 de cada renglón.
  zz = 0;
  for ( a2 = 1; a2 <= 60; a2++ )
  {
    if( Bethoven [a1] [a2] != 0 )       //zz es la variable que se utiliza
      {                                   //para ir sumando los porcentajes
      zz += Bethoven[a1][a2];
      Bethoven[a1][a2] = zz;
      }
    }
  Bethoven[a1][62] = zz;
}
}
```

Una vez que la rutina EscalaPorcentual() ha generado la Matriz Evolutiva de Frecuencias Acumuladas conforme a lo descrito en el tema Sistema Compositor, del Capítulo II Sistema Evolutivo Generador de Música, ya se han almacenado en la matriz Bethoven[70][70] los cambios aprendidos por el Sistema a través del ejemplo proporcionado y la matriz evolutiva contiene las características de los ejemplos que se proporcionen.

Utilizando la información contenida en la matriz obtenida de esta manera, el sistema ya es capaz de componer una melodía y esto lo hace por medio de un generador de música.

g. Generador de Música (Rutina MusicaPorcentual)

La rutina que se encarga de hacer las composiciones musicales se llama MusicaPorcentual(). Lo hace llevando a cabo el proceso mencionado en el tema Sistema Compositor, del Capítulo II Sistema Evolutivo Generador de Música. Esto es, escoge uno de los renglones en los que la columna 62, de la Matriz Evolutiva de Frecuencias Acumuladas, sea diferente de cero. Genera un número aleatorio entre cero y el valor del total almacenado en la casilla 62 de ese renglón y lo compara con todos los valores mayores de cero de ese renglón hasta encontrar el que sea mayor o igual, la columna donde está almacenado el valor encontrado representa la siguiente nota a ser tocada y el siguiente renglón a ser procesado de la misma forma.

Mediante la rutina MúsicaPorcentual() se pueden generar agradables melodías. Pero esta no es la única función de la rutina MúsicaPorcentual(). *Una función muy importante que realiza esta rutina es la de ir modificando las dos matrices al mismo tiempo que está generando la melodía. Este hecho es el que convierte a las matrices en evolutivas y por consecuencia al Sistema.*

Para explicar este proceso, al cual nos hemos referido anteriormente como proceso de aplicación, tomemos como ejemplo un ser humano (sistema evolutivo) que está

aprendiendo a tocar música. La manera en la que aprende a tocar, es repitiendo muchas veces ejercicios y melodías con el fin de adquirir la destreza para ejecutarlos perfectamente. Mientras más se toque una melodía, más se graba en la memoria del músico, llegando al punto en que es capaz de ejecutar la melodía sin tener que voltear a ver la nota que está tocando. Así pues, las notas que más se tocan son las que más se recuerdan y mientras menos se tocan tienden a olvidarse.

Similarmente, la rutina MusicaPorcentual() cada vez que genera una nota, incrementa su posibilidad de ser tocada nuevamente. Esto quiere decir, que las matrices están evolucionando constantemente, pues con cada nota que es tocada las matrices evolutivas sufren modificaciones.

Es importante que el Sistema cuente con un mecanismo de olvido, ya que mientras más se toca una nota, mayor es la posibilidad de que vuelva a ser tocada y si no existiera un mecanismo de olvido, después de un largo tiempo de ejecución, todas las notas tendrían probabilidades muy grandes de ser tocadas. Sin embargo, desarrollar un mecanismo de olvido no es una tarea sencilla y podría representar un trabajo completo por sí sólo. Uno muy sencillo es que cada vez que algún valor rebasa cierto límite, se les divide a todos los valores de la matriz entre 2, de tal forma que los valores menores a la unidad se igualan a 1.

Para que la melodía se escuche, una vez que se ha ejecutado alguno de los generadores de ruido o por el generador de música, el sistema llama a una rutina llamada

Prog. 11: Pseudocódigo de la rutina MusicaPorcentual().

```

MusicaPorcentual()
{
  a1 = nota_siguiente //Genera un número aleatorio entre el valor almacenado en el
  zz = random( Bethoven [a1] [62] )+1; // (renglón a1, columna 62) y cero.

  a2 = 1; //encuentra la columna a2 del renglón a1, que contiene
  while( Bethoven[a1][a2] < zz && a2<=60 ) a2++; //un valor mayor o igual al número aleatorio, el
  nota_siguiente = a2; // número de esa columna es la siguiente nota.

  Bethoven1[a1][a2] += 1; //Actualiza (evoluciona) la Matriz Evolutiva Aleatoria.

  if( Bethoven1[a1][a2] > 500 )
  {
    for( a1 = 1; a1 <= 60; a1++ ) //Mecanismo de olvido, cada vez que algún valor de la
    for( a2=1; a2 <= 60; a2++ ) //matriz llega a 500, a todos los valores se les divide
    if( Bethoven1[a1][a2] > 1 ) //entre 2 y aquellos menores a 1 se igualan a 1.
    Bethoven1[a1][a2] /= 2;
    else
    Bethoven1[a1][a2] = 1;
    EscalaPorcentual();
  }
  while( a2 <= 62 )
  {
    if( Bethoven[a1][a2] != 0 ) //Actualiza (evoluciona) la Matriz Evolutiva de
    { // Frecuencias Acumuladas
    Bethoven[a1][a2] += 1;
    }
    a2++;
  }
}

```

ControladorDeBocina(), que es la que se encarga de hacer que se escuche la melodía, por medio de la bocina de la PC.

h. Rutina ControladorDeBocina

Para decirle a la computadora que emita el sonido de alguna frecuencia en particular, necesitamos explicar el funcionamiento del *timer* de la PC. El *timer* está controlado mediante un “chip” denominado 8253, que está compuesto de 3 contadores de 16 bits y programables de manera independiente, mediante los puertos 40h – 43h[11].

El 8253 recibe una señal de un reloj a cierta frecuencia (en este caso son 1,193,180 ciclos por segundo) y emite señales por tres canales independientes que se controlan mediante los tres contadores.

El timer del puerto 40h genera una interrupción a 18.2 Mhz aproximadamente, la cual es utilizada por el sistema operativo. El contador del puerto 41h, se utiliza para actualizar la memoria del sistema DRAM, se recomienda no tocar este contador, ya que un error puede acarrear graves consecuencias. El puerto 42h se puede utilizar para controlar la bocina y el puerto 43h se utiliza como palabra de control.

Mediante la palabra de control se selecciona el contador que se va a programar, el modo de operación de dicho contador y el tipo de operación (escritura/lectura). La estructura de esta palabra se indica en la figura 12.

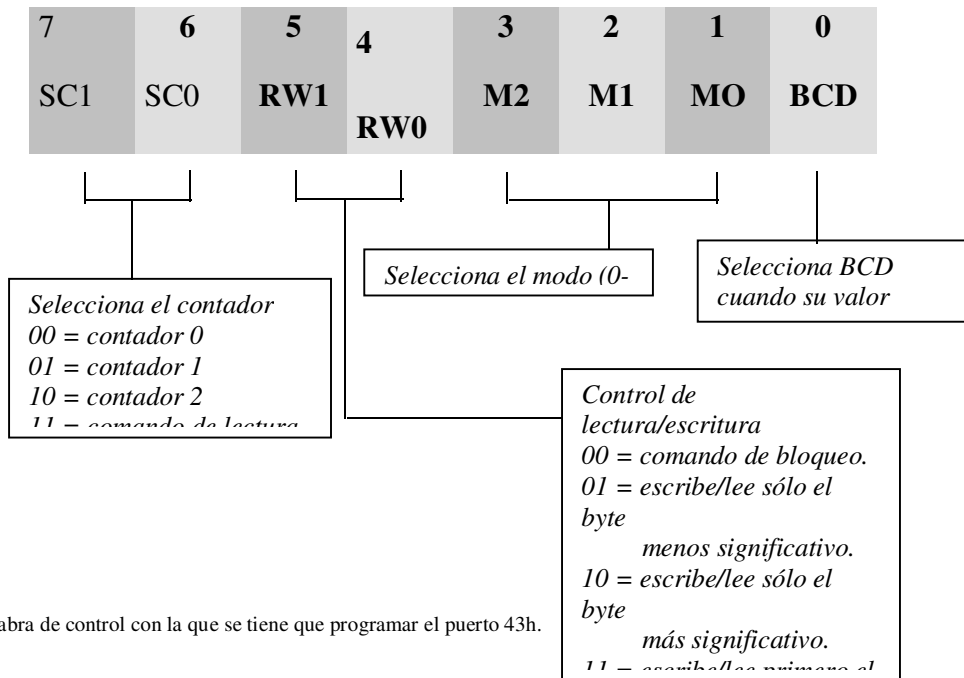


Fig. 12 Palabra de control con la que se tiene que programar el puerto 43h.

Como se puede deducir de la figura 12, cada contador se puede programar de 6 modos diferentes, se puede contar en binario o en BCD, se puede utilizar para leer o escribir, es decir tiene distintas maneras de programarse, pero debido a que no es el objetivo de este trabajo profundizar en este tema, sólo necesitamos saber, por el momento, que para controlar la bocina de la PC, basta con programar la palabra de control mandando el valor de 182 al puerto 43h y enviando la frecuencia a la que queremos que suene la bocina al puerto 42h.

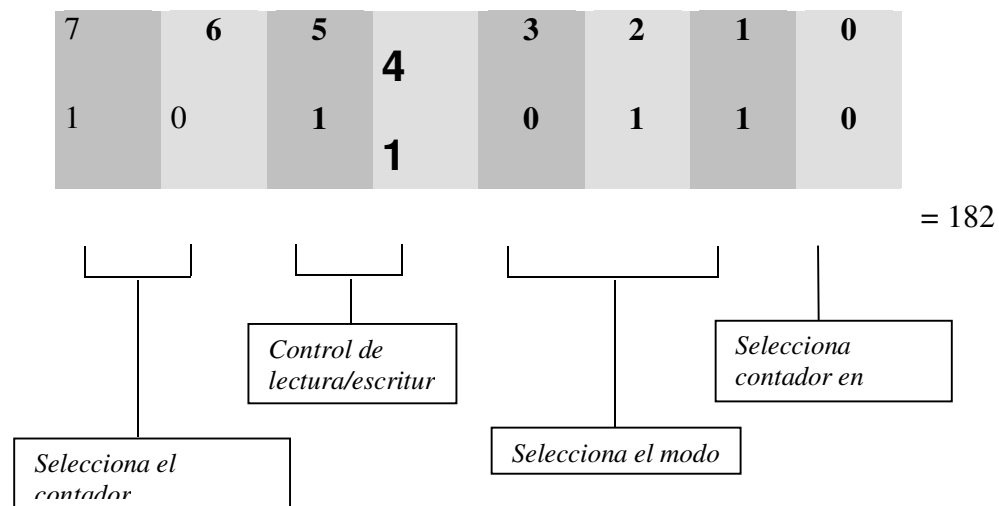


Fig. 13 Valor 182 como palabra de control

Ahora, sólo falta pedir que la bocina suene, esta operación se controla mediante el puerto 61h. Para que la bocina suene, es necesario que los 2 bits menos significativos de este puerto estén prendidos. Hay que tomar en cuenta que este es un puerto de 8 bits y que los 6 bits restantes son ocupados con otros propósitos, razón por la cual no deben de ser alterados. Una vez que los 2 bits menos significativos del puerto 61 estén en 1, se escuchará un sonido a la frecuencia programada en el puerto 42h, que no dejará de escucharse hasta que los dos bits menos significativos del puerto 61h vuelvan a cero.

```
ControladorDeBocina(frecuencia)
{
  contador = 1193180/frecuencia;
  outportb( 0x42, contador );
  soundon = ValorPuerto61 Or 3;      //SoundOn
  outportb( 0x61, soundon );
}
```

Prog. 12: Pseudocódigo para controlar el pulso del reloj.

Mediante el proceso anterior, lo que estamos haciendo es pedirle al 8253 que envíe un pulso, con la frecuencia programada en el puerto 42h, a la bocina, provocando la oscilación del cono de la bocina, emitiendo un sonido a la frecuencia programada.

De esta manera, podemos controlar en *tiempo real* el funcionamiento de la bocina.

I. Rutina ControladorTimer

Ahora bien, la duración del sonido puede ser controlada mediante una rutina de “delay” y se puede hacer utilizando, nuevamente, los puertos para programar los timers.

El puerto 40h, está programado, por el BIOS, para emitir un pulso en un poco más de un 18^{vo} de segundo, el sistema utiliza este timer para llamar automáticamente en cada “clock tick” a la interrupción 1Ch (system timer tick), conocida como la rutina del reloj.

Nosotros podemos colocar en lugar de la rutina del reloj, una rutina que tenga un contador, de esta manera, como esta rutina es llamada con cada pulso del timer del puerto 40h, le podemos pedir que con cada pulso haga un incremento en el contador.

Además, el puerto 40h lo podemos programar para que modifique la velocidad a la que envía pulsos y que en lugar de generar un pulso cada 18^{vo} envíe alrededor de 1,000,000 de pulsos por segundo, conque lo programemos a 10,000 c/s, tenemos una rutina de “delay” que nos permite tener control de la duración de las notas hasta en 1/10,000 de segundo[11].

De esta forma, ya podemos hacer interpretaciones musicales. He aquí el pseudocódigo de la rutina, denominada `ControladorTimer()`, mediante la cual, programamos el *timer* a una frecuencia de 10,000 c/s.

```
ControladorTimer()
{
    outportb(0x43, 54);           /*Envía al puerto 43h la
                                /*secuencia 00110110 para
                                /* programar el puerto 40h

    contador = 1193180/10,000;    /* 10,000 c/s
    outport( 0x40,contador );
}
```

Prog. 13: Pseudocódigo para controlar el timer.

Este es el funcionamiento del sistema y la manera en la que fue construido. La cantidad de procesos que el sistema realice puede llegar a ser tan grande como se quiera y pueden ser tan variados como nuestra imaginación lo permita. De manera similar ocurre con la matriz evolutiva, puede aprender por siempre y mientras más ejemplos se le proporcionen más se parecerá la música que se obtenga a la compuesta por los seres humanos.

CONCLUSIONES

La música ha acompañado, según los historiadores, al hombre en su recorrido sobre la tierra. Y le hemos dado un lugar especial en el mundo en que vivimos.

La música ha hecho surgir preguntas como: ¿Qué es la música? ¿Qué es lo que convierte a la música en un sonido tan especial, existiendo una gama tan diversa de sonidos que sería imposible describirlos a todos? ¿Porqué sólo ciertas frecuencias suenan tan bonito? ¿Acaso la música imita los sonidos de la naturaleza? ¿Será que describe las emociones? ¿La belleza? ¿O simplemente es un don con el cual los dioses dotaron al humano para su deleite? Este tipo de preguntas se las han hecho grandes filósofos, matemáticos, físicos, químicos, mecánicos, pescadores, campesinos, carpinteros, herreros y casi podríamos asegurar que todas las personas que han tenido la oportunidad de deleitar su espíritu con la música. Hasta la fecha no se llega a un acuerdo para poder ofrecer una respuesta satisfactoria a todas estas preguntas.

Afortunadamente existen miles de diferentes corrientes musicales, compleja o no, ruidosa o armoniosa, suave o fuerte, religiosa o satánica, rápida o lenta, folklórica o importada, que sirven para deleite del oído humano. Se ha de resaltar el hecho de que la subjetividad de la música, es una característica, que bien puede ser su más fuerte aliado, así como su peor enemigo. Dicha subjetividad es la que hace que una pieza se vuelva o no una obra maestra. Alguien podría argumentar que para que sea una obra maestra debe de tener cierta armonía, estar escrita sobre ciertas escalas o contener ciertas progresiones. Lo cierto es que lo único que cuenta es que le guste o no a quien escuche la melodía. Esto lo prueba el hecho de que la música llamada culta o clásica sólo la escucha una élite de personas que se autocalifican como cultas, mientras que composiciones coloquiales, que bien es cierto que algunas de ellas son bastante monótonas, son escuchadas por millones de personas alrededor del mundo. Así es de que si una melodía se convierte en obra maestra en base al tamaño del auditorio que la escucha, entonces un éxito popular puede superar a composiciones de Mozart, Bethoven o Bach, a pesar de la gran diferencia en la complejidad que existe entre estos diferentes tipos de música.

Algo maravilloso es que, no influyen factores como la edad para delimitar a quien escucha un tipo de música y quien escucha otro. Tampoco es del todo cierto que el medio en el cual nos desarrollamos influye de manera determinante. Es probable que aquella persona que creció escuchando música clásica, tenga cierta inclinación por escuchar este tipo de música o quien se desarrollo escuchando música popular tenga inclinaciones por este otro, sin embargo, esto no se podría asegurar, ya que las personas van evolucionando y quien escuchaba música clásica, es posible que ahora sea capaz de escuchar música popular e incluso deleitarse escuchándola y lo mismo ocurre en sentido opuesto, quien escuchaba música popular es capaz de disfrutar la clásica sin igual.

Ahora bien, nos preguntamos ¿porqué hacer un sistema compositor, si existen tantos compositores de tan excelente categoría?

Los cierto es, que entre algunos de los objetivos de esta herramienta se encuentra el de acercar a la mayor cantidad posible de personas, sean o no músicos, a la composición musical y a la informática.

Por otro lado, el simple hecho de poder escuchar música creada en ese instante y por siempre diferente, es fabuloso.

Además, si alguien se interesa en el desarrollo de música informática, encontrará que estudiar la música, que es tan sólo uno de los tantos miles de millones de fenómenos que ocurren en este Universo, exige buscar los intrincados caminos por los cuales fluye la información creando algo tan maravilloso para deleite del sentido auditivo de los seres humanos y no sólo escuchar las composiciones musicales es placentero, saber como crear un breve sonido, lo que implica ponerlo en sucesión con otros sonidos e ir escuchando como va adquiriendo complejidad hasta formar grandes sinfonías es verdaderamente agradable y provoca que la admiración que se tenía por los grandes maestros de la música crezca aún más, ya que es verdaderamente impresionante como esos genios se las arreglaron para poder hacer esas fabulosas composiciones.

Es pues entonces, que tanto la música como la informática, son disciplinas tan maravillosas que tan sólo por el simple hecho de existir han hecho cambios alrededor del mundo de manera trascendente.

Así es que, resulta bastante interesante conocer una de las bellas artes, como es la música, que como todas las disciplinas, nos permite echar un vistazo al interior de uno mismo y nos enseña que para tan sólo conocer una pequeña porción del largo y ancho sendero de la música, se necesita una gran dedicación, una buena disciplina y a pesar de todas las horas que uno entregue, “al final” siempre habrá más por aprender y siempre existirá el riesgo de equivocarse hasta en lo más trivial.

Por otro lado, la informática es otra disciplina, que nos permite navegar por mundos maravillosos, ya que es un área tan grande y con tantas aplicaciones, que prácticamente se puede utilizar para estudiar cualquier cosa. Debido a la idea de que el mundo en el que vivimos no está formado sólo de *materia* y *energía*, sino además de *información*, hay mucha informática por hacer, luego entonces no suena mal la idea de dedicarnos a revelar la información escondida en cada rincón de este Universo.

Una excelente manera de modelar la Naturaleza es utilizando los sistemas evolutivos. Probando su gran versatilidad, ahora le han dado vida a un sistema que puede hacer composiciones tal y como su propio creador, que finalmente cumplen con la característica de ser un conjunto de sonidos y silencios a través del tiempo.

Lo presentado hasta ahora, es sólo una parte de un proceso general de composición de música por computadora, ya que sólo se muestran melodías con notas de duración constante, además de utilizar como notas las frecuencias puras de las notas, sería interesante complementarlas con todos sus armónicos para escuchar un piano, un violín o una guitarra. Por lo tanto, queda abierto para un probable trabajo de grado, en el que se incluya el uso de matrices evolutivas de 3 o más dimensiones.

Disfrutemos pues del caos, de la música y dejemos que la informática haga presa de nosotros, al fin y al cabo lo que sobra es información.

Anexo A

Descripción del Formato Musical

Un formato, es la manera en la que se codifica cierta información. En la actualidad, existen miles de formatos para guardar y transmitir información, por ejemplo, entre algunos formatos gráficos ampliamente difundidos, encontramos los BMP, PCX, ICO y algunos musicales son, CMF, VOC, WAV y MIDI.

La manera en la que todos estos formatos almacenan la información es codificando, de una u otra forma, los bits que contienen dicha información, algunos de ellos utilizan códigos un poco complejos con el fin de que no puedan ser decodificados por cualquier persona.

Tratándose de formatos musicales, la información que comúnmente se necesita codificar son las notas musicales, los tiempos de duración de cada nota, los tipos de instrumentos que se usan, las intensidades de las notas, los distintos efectos que se pueden dar, que varían de pendiendo de los diferentes dispositivos que se utilicen, etc.

Para poder darle ejemplos de música a este sistema, hemos desarrollado un formato muy sencillo, el cual, dadas las necesidades del sistema sólo almacena los valores de las notas musicales, ya que las composiciones que se han dado como ejemplo utilizan notas con duración constante.

Recordemos que hemos creado un “piano virtual”, el cual contiene los valores de las frecuencias de 5 escalas cromáticas, dicho piano es un arreglo al que se le ha denominado Notas[.].

$$\text{Notas[.]} = \{0,65,69,73,77,82,87,92,98,103,\dots,1963\}$$

Por lo tanto, si queremos una nota específica, basta con referirnos a ella con el número de la posición que ocupa en el arreglo, de esta forma, Notas[1]=do, Notas[10]=La y Notas[60]=Si. Así pues, en el formato musical, tan sólo se utilizan los valores numéricos del 1 al 60.

Con un byte podemos representar los números del 0 al 255, entonces con definir un arreglo de tipo carácter, es suficiente para almacenar una melodía musical como las que hemos utilizado de ejemplos.

La siguiente, es una de las melodías que hemos proporcionado como ejemplo al sistema:

```
char paganini1[] =  
{0,15,34,32,30,29,27,26,23,22,20,18,17,15,14,15,18,22,27,30,34,39,42,41,39,44,41,38,34,  
32,29,42,39,34,30,29,27,26,34,29,26,22,17,14,10,9,10,12,14,15,34,32,30,29,27,26,23,22,  
20,18,17,15,14,15,18,22,27,30,34,39,42,41,39,35,32,29,35,34,32,30,29,27,34,22,26,27,26,  
27,22,21,20,19,18,17,20,18,17,15,99};
```

BIBLIOGRAFÍA

- [1] Sensación y Percepción Auditiva y de los Sentidos Menores Josef Cohen. Ed. Trillas
- [2] Música Blanca y Música Parda, Curvas Fractales y Fluctuaciones del tipo 1/f. Martin Gardner. Investigación y Ciencia, junio 1978.
- [3] Materia Tierra y Cielo. George Gamow. Compañía Editorial Continental S.A.
- [4] Técnicas de Simulación en Computadoras. Thomas H. Naylor, Joseph L. Balintfy, Donald S. Burdick y Kong Chu. Ed. Noriega Limusa.
- [5] Física General. Francis W. Sears y Mark W. Zemansky. Ed. Aguilar.
- [6] El Ruido 1/f. Edoardo Milotti. Investigación y Ciencia, noviembre 1996.
- [7] El Orden Caótico. Monique Dubois, Pierre Aften y Pierre Bergé. Mundo Científico No 68. Vol 7.
- [8] Del Azar Benigno al Azar Salvaje. Benoit Mandelbrot. Investigación y Ciencia, Diciembre 1996.
- [9] El Caos, la nueva Física, las nuevas Matemáticas y sus Aplicaciones a las Ciencias Sociales. Miguel José Yacamán. Conferencia dictada en el Colegio de México. Noviembre 1993.
- [10] Fundamentos de Física III Óptica. Francis W. Sears. Ed. Aguilar.
- [11] The Intel Microprocessors 8086/8088, 80186, 80286, 80386 and 80486. Architecture Programming and Interfacing. Barry B. Brey. Ed. Prentice Hall.
- [12] Evolución. Fernando Galindo Soria. Septiembre 1995.
- [13] Acerca del Continuo Dimensional: Un Universo Fractal. Fernando Galindo Soria.