

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



GENERACIÓN DE FRAGMENTOS MUSICALES POR MEDIO DE  
UN ALGORITMO MATEMÁTICO

Reporte Técnico de Investigación presentado por:

Ángel Rubén Hernández Quezada 93476

Requisito para la obtención del título de

INGENIERO EN SISTEMAS COMPUTACIONALES

Profesor Responsable: Saúl González Campos

Noviembre de 2012



## **Autorización de Impresión**

Los abajo firmantes, miembros del comité evaluador autorizamos la impresión del proyecto de titulación

### **GENERACIÓN DE FRAGMENTOS MUSICALES POR MEDIO DE UN ALGORITMO MATEMÁTICO**

Elaborado por el alumno:

Ángel Rubén Hernández Quezada 93476

Ivonne Haydee Robledo Portillo

Profesor de la Materia

Saúl González Campos

Asesor Técnico

## **Declaración de Originalidad**

Yo Ángel Rubén Hernández Quezada declaro que el material contenido en esta publicación fue generado con la revisión de los documentos que se mencionan en la sección de Referencias y que el Programa de Cómputo (Software) desarrollado es original y no ha sido copiado de ninguna otra fuente, ni ha sido usado para obtener otro título o reconocimiento en otra Institución de Educación Superior.

---

Ángel Rubén Hernández Quezada

## **Dedicatoria**

Este proyecto de titulación lo dedico a todas las personas que me han brindado su apoyo tanto técnico como moralmente, en especial a mis padres que me han permitido continuar con mis estudios en esta ciudad a pesar de la gran distancia.

A mis familiares cercanos por motivarme a seguir adelante siempre, a mi novia que me invitaba a darnos el tiempo para trabajar durante las tardes juntas en los pendientes de la escuela.

Por último también me gustaría dedicar este proyecto a quienes tienen interés en las áreas artísticas y que de alguna manera su área de estudio parece distanciarlos un poco de dichas áreas.

## **Agradecimientos**

Agradezco a mi maestra por guiarme en la elaboración de este proyecto, que al principio me costó encaminarme pero siempre buscó la manera de aclarar mis dudas.

Agradezco también a mi asesor de proyecto por leerme en varias ocasiones y hacerme saber su punto de vista, de esa manera me hizo ver caminos diferentes para abordar los problemas que pudieran presentarse.

# Índice

Autorización de Impresión.....	iii
Declaración de Originalidad .....	iv
Dedicatoria .....	v
Agradecimientos .....	vi
Lista de Figuras .....	ix
Lista de Tablas .....	x
Introducción .....	1
Capítulo 1. Planteamiento del problema.....	2
1.1 Antecedentes .....	2
1.2 Definición del problema .....	4
1.3 Objetivos de la investigación .....	4
1.4 Preguntas de investigación.....	4
1.5 Justificación de la investigación .....	4
1.6 Limitaciones y delimitaciones de la investigación .....	5
1.6.1 Limitaciones.....	5
1.6.2 Delimitaciones .....	5
Capítulo 2. Marco Teórico .....	6
2.1 Conceptos musicales de interés .....	6
2.1.1 Nota.....	6
2.1.2 Tiempo .....	7
2.1.3 El sonido y la música .....	7
2.1.4 Melodía .....	8
2.1.5 Armonía .....	8
2.2 ¿Qué es composición algorítmica? .....	8
2.2.1 Introducción .....	8
2.2.2 Historia de la composición algorítmica .....	9
2.3 Algoritmos de generación de música.....	14
2.3.1 Autómata celular .....	14
2.3.2 Fractales .....	20
2.3.3 Redes neuronales .....	21
2.3.4 Cadenas de Markov.....	22
2.3.5 Sistemas caóticos .....	22

2.3.6 Gramáticas formales .....	23
Capítulo 3. Materiales y Métodos .....	24
3.1 Descripción del área de estudio .....	24
3.2 Materiales.....	24
3.3 Métodos.....	26
3.3.1 Tipo de investigación.....	26
3.3.2 Metodología .....	27
3.3.3 Implementación.....	29
4.1 Presentación de resultados .....	34
4.2 Análisis e interpretación de resultados .....	35
4.1.1 Ejecución 1.....	36
4.1.2 Ejecución 2.....	37
4.1.3 Ejecución 3.....	38
Capítulo 5. Discusiones, conclusiones y recomendaciones .....	39
5.1 Con respecto a las preguntas de investigación.....	39
5.2 Con respecto al objetivo de la investigación.....	40
5.3 Recomendaciones para futuras investigaciones.....	41
Referencias.....	42
Apéndices.....	46



## Lista de Figuras

Figura 1. Representación grafica de la proporción aurea. ....	10
Figura 2. Autómata celular unidimensional con un radio de vecindad $r=1$ y una condición de frontera periódico .....	17
Figura 3. Primeras seis transiciones de un autómata celular unidimensional de 10 celdas (células).....	18
Figura 4. Comportamiento típico de un autómata celular bidimensional.....	20
Figura 5. Tabla de probabilidad de transición entre estados utilizando cadenas de Markov .....	22
Figura 6. Estructura general de generación y almacenamiento de sonido utilizando un autómata celular .....	32
Figura 7. Interfaz gráfica del usuario .....	34
Figura 8. Captura de pantalla con valores elegidos arbitrariamente No.1 .....	36
Figura 9. Cadena de ceros y unos generada durante la ejecución 1 .....	36
Figura 10. Captura de pantalla con valores elegidos arbitrariamente No.2 .....	37
Figura 11. Captura de pantalla con valores elegidos arbitrariamente No.3 .....	38

## **Lista de Tablas**

Tabla 1. Rango de notas MIDI y su relación con la escala musical .....	32
--	----

## Introducción

Este proyecto que está bajo el nombre de: “Generación de fragmentos musicales mediante un algoritmo matemático”, constituye una investigación y la elaboración de un programa que permita a un usuario generar una sucesión de notas musicales que serán creadas por una estructura matemática llamada “autómata celular”.

A lo largo del proyecto primeramente se plantea el problema, pasando por antecedentes, se define el problema y los objetivos del actual proyecto. Se define lo que se busca desarrollar y el porqué, tomando como pilar elemental la motivación de experimentar técnicas nuevas e ideas poco exploradas.

Sobre la parte teórica se investiga sobre dos áreas de conocimiento humano: La música y las matemáticas. Esto debido a que se trabaja con ambas áreas y es preciso tener al menos un conocimiento mínimo de música para poder incluir y comprender conceptos sobre la estructura elemental de los sonidos generados.

Por otro lado se investiga sobre la composición algorítmica y los modelos matemáticos que sean compatibles para implementarse en el programa final, estos modelos matemáticos son ampliamente conocidos en las áreas de estudio de algoritmos, pasando por: fractales, cadenas de Markov, sistemas caóticos y desde luego autómatas celulares.

Con respecto a los autómatas, se eligió el algoritmo del autómata celular unidimensional para el desarrollo del proyecto que aquí se presenta.

Por último se obtuvieron los resultados generados por el programa, se experimentó ampliamente en varias las condiciones iniciales de la aplicación para observar el comportamiento de la salida del mismo.

# Capítulo 1. Planteamiento del problema

A lo largo de este apartado se describe la inquietud que ha llevado a muchas personas a intentar hacer música por métodos matemáticos. Se plantea el objetivo que este proyecto tiene como fin, también se cuenta con preguntas de investigación que nos guiarán para la estructuración y orientación del proyecto, otro aspecto que se considera en posteriores apartados son las limitaciones y delimitaciones, ya que por motivos diversos nos enfocaremos en el estudio de un modelo matemático en particular que será ampliamente descrito y posteriormente implementado en nuestro programa para obtener un producto final.

## 1.1 Antecedentes

La composición de música es un proceso que requiere tanto de creatividad como de la imaginación y habilidad. Puesto que los algoritmos matemáticos están naturalmente relacionados con el determinismo es complicado establecer un juicio formal o estándar para cuantificar la calidad y la creatividad de un fragmento musical generado artificialmente. [1]

Hay quienes afirman que existe una estrecha relación entre las matemáticas y la música. El hecho de que las notas musicales no tengan un sonido aleatorio, sino que lo hagan en una unión pactada, esto ha llevado a realizar búsquedas con bases científicas que por supuesto podrían utilizarse como fuentes de creatividad. [2]

La composición algorítmica es un área interesante para la investigación dentro del campo de las matemáticas y la inteligencia artificial, esto es porque la música trata de dos grandes posibilidades hasta ahora exclusivamente humanas: La intelectual y la emocional. [3,4]

Si se remonta a la era del famoso filósofo Pitágoras, el concepto de matemáticas, englobaba lo que se conoce como el “Quatrivium”, es decir, que la astronomía, la música, aritmética y geometría constituían la base del saber exacto. Los seguidores de Pitágoras consideraban que los números eran una parte esencial en la música e intentaron explicar la

armonía en función de una serie de razones numéricas universales. El idealismo geométrico de los pitagóricos sobrevive hoy en día en el uso de técnicas de composición basadas en la teoría del caos y la gramática fractal, por ejemplo. [4]

Gracias a la computación moderna se pueden generar procesos más complejos para la composición de música algorítmica. Uno de los más elementales es conocido como “mapping”, este consiste en crear relaciones directas entre el *output* de un algoritmo (que por lo general genera un *output* numérico) y parámetros musicales. El *mapping* de un modo generalizado puede ser concebido como la traducción del algoritmo en música. [5,1]

Las ventajas de poder de procesamiento también han construido ventajas dentro de este campo, uno de los *frameworks* más ampliamente utilizados en la actualidad para la composición de música es jmusic, este se basa en el lenguaje de programación java, lo cual proporciona una gran ventaja en el ámbito de la compatibilidad ya que cualquier persona que cuente con una máquina virtual de java tiene potencialmente la capacidad de compilar y ejecutar sus propias creaciones musicales. [6]

Existe *Musical Generator*, que es un programa capaz de crear melodías a partir de fractales, teoría caótica, mapas complejos y textos. Sin embargo es un sistema carente en algunos aspectos como el hecho de que solamente aproxima los valores de las cadenas generadas por los otros sistemas y los acomoda según la cercanía de una nota musical en la escala determinada. [5]

Otro de estos lenguajes orientados a la producción de música es ChucK desarrollado por Ge Wang a partir del año 2003. [8] Este es un lenguaje de programación que permite generar y sintetizar sonidos en tiempo real.

Cabe mencionar que la composición algorítmica es también lógicamente parecida a hablar un lenguaje, es decir, tiene que tener un sentido para el receptor, de lo contrario el producto consistiría únicamente en palabras sin sentido. [8,9]

## **1.2 Definición del problema**

Evaluar si es factible utilizar un algoritmo matemático orientado a la generación de fragmentos musicales.

## **1.3 Objetivo de la investigación**

Desarrollar un prototipo basado en un algoritmo matemático que genere una sucesión de notas musicales.

## **1.4 Preguntas de investigación**

¿Qué es composición automatizada?

¿En dónde radica la pertinencia para considerar un algoritmo más apropiado que otro para la generación de sonidos?

¿Qué ventajas encontramos en el algoritmo seleccionado?

Dependiendo de los resultados, ¿qué uso podría darse al prototipo generado?

¿Qué limitantes encontramos en nuestro algoritmo empleado?

## **1.5 Justificación de la investigación**

Es una gran ventaja contar con una herramienta automatizada para la generación de música, debido a que se podría apoyar una parte del trabajo musical en una secuencia generada por un modelo matemático. Es por esto que existe una inquietud tecnológica por explorar la naturaleza de la música e intentar simularla lo más acertadamente posible.

## **1.6 Limitaciones y delimitaciones de la investigación**

### **1.6.1 Limitaciones**

Este proyecto se limita a presentar un prototipo capaz de generar una sucesión de notas musicales mediante un método seleccionado. Debido a la complejidad musical y matemática de modelar un fragmento armónico, no existe una garantía de por medio para considerar al producto final como un motor comparable con un individuo que dedica su intelecto en crear composiciones.

### **1.6.2 Delimitaciones**

En cuanto a las delimitaciones del proyecto se encuentra el funcionamiento del algoritmo que aunque tiene cierto factor de comportamiento estocástico a su vez queda limitado por cuestiones matemáticas que no simulan situaciones de la psique humana. Otro factor a considerar es el tiempo de desarrollo del prototipo, siendo un área de estudio poco explorada y potencialmente muy vasta en un periodo de tiempo como este no se puede cubrir en totalidad.

## Capítulo 2. Marco Teórico

A lo largo de este capítulo se abordan conceptos diversos que son útiles para la elaboración y estructuración de los fragmentos musicales. Primeramente se repasa la definición de nota, tiempo, sonido, música, melodía y armonía. Estos conceptos musicales otorgan un mejor criterio a la hora de distinguir cualitativamente el producto calificando la “calidad” del output final. También se define lo que es composición algorítmica, un concepto que sin duda es elemental y de sumo interés, así como los que son los procesos formales, las máquinas de composición que han sido utilizadas a lo largo de la historia para generar sucesiones de sonidos y la nomenclatura de la composición algorítmica. Sobre todo, un aspecto relevante que explica este capítulo son los algoritmos existentes para ser utilizados como motor de una composición, estos algoritmos pueden ser determinísticos o estocásticos y dependiendo de su naturaleza algunos tienden a generar sucesiones más rítmicas que otros.

### 2.1 Conceptos musicales de interés

#### 2.1.1 Nota

Se denomina nota en el ámbito musical a un sonido determinado por una vibración y cuya frecuencia es constante. Guido D’Arezzo quien es considerado el padre de la notación musical elaboro una aproximación al sistema actual de notas (con excepción de la séptima nota) asignando un nombre a cada nota. [7] Las notas musicales son siete:

- Do
- Re
- Mi
- Fa
- Sol
- La
- Si



### 2.1.2 Tiempo

El tiempo es la velocidad con que debe ejecutarse una pieza de música. A menudo la palabra *tempo* está acompañada por la indicación metronómica. Este es una expresión que indica la velocidad más adecuada para una pieza de música indicando cuantas figuras de un determinado valor deben tocarse en un minuto. [7]

### 2.1.3 El sonido y la música

Se le denomina sonido al fenómeno físico que da lugar a la música, que es a su vez la manipulación intencional del sonido para generar estructuras audibles que sean estéticamente agradables. Por definición sabemos que el sonido es la sensación percibida por el oído al decodificar las vibraciones exteriores y asimilarlas. [1,7]

El sonido tiene los siguientes parámetros:

**Altura:** Es la cantidad de ciclos de las vibraciones por segundo de un cuerpo sonoro, es decir, hercios (Hz) que se emiten. De acuerdo con esto se pueden definir los sonidos como "graves" y "agudos". Cuanto mayor sea la frecuencia, más agudo será el sonido. La longitud de onda es la distancia medida en la dirección de propagación de la onda, entre dos puntos cuyo estado de movimiento es idéntico; es decir, que alcanzan sus máximos y mínimos en el mismo instante. [7,9]

**Duración:** Corresponde al intervalo de tiempo de las vibraciones que producen un sonido.

**Intensidad:** Es la fuerza con la que se produce un sonido; depende de la energía. La intensidad viene representada en una onda por la amplitud.

**Timbre:** Es la cualidad que permite distinguir los diferentes instrumentos o voces a pesar de que estén produciendo sonidos con la misma altura, duración e intensidad.

Los sonidos que escuchamos son complejos, es decir son el resultado de la suma de diferentes frecuencias pero que se perciben como si fuera un solo. El timbre depende de la

cantidad de armónicos o la forma de la onda que tenga un sonido y de la intensidad de cada uno de ellos a la cual se denomina espectro.

**El ruido:** En cuanto al ruido, es un sonido simple o complejo pero no armónico y de muy alta intensidad, generando intolerancia o dolor al oído y una sensación de displacer al individuo, es decir, es la sensación auditiva inarticulada generalmente desagradable.

En el medio ambiente, se define como todo lo molesto para el oído. Desde ese punto de vista, la más excelsa música puede ser calificada como ruido por aquella persona que en cierto momento no desee oírla. [7,3]

**El silencio:** es la ausencia de sonido o al menos su aproximación más cercana.

#### **2.1.4 Melodía**

Por definición una melodía es la composición sobre la cual se desarrolla una idea musical, sea simple o compuesta independientemente de su acompañamiento. En otras palabras se le puede resumir también como la sucesión coherente y agradable de sonidos y silencios. [7,3]

#### **2.1.5 Armonía**

Es básicamente la organización de los acordes. Un acorde consiste en tres o más notas distintas que suenen simultáneamente, puede ser reconocida también como un ajuste, acuerdo o concordancia entre dichos acordes.

## **2.2 ¿Qué es composición algorítmica?**

### **2.2.1 Introducción**

Los compositores han descubierto a través del tiempo que algunos procesos musicales pueden ser formalizados por una representación simbólica, un algoritmo de composición es el motor de esta creación. [10]

Ada Lovelace (1818-1852), quien describió la máquina analítica de Charles Babbage, es considerada la primera programadora. Recordada actualmente por deducir y prever la capacidad de las computadoras para ir más allá de los simples cálculos numéricos.

Mencionó acerca de la música:

*“Supposing for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.”* [3]

Estas ideas dieron las condiciones propicias para el nacimiento de una nueva rama dentro de las ciencias de la computación que es llamada composición algorítmica, que a su vez se complementa con otra gran área llamada sintetización musical. [10,11] Esta última es de gran utilidad en la ingeniería de sonido gracias a la gran variedad de ventajas que se encuentran al poder manipular los sonidos.

La composición algorítmica consiste en la implementación de modelos algorítmicos en la generación de música, un algoritmo debe contar con lo siguiente para ser considerado como tal: [1,12]

- Un número finito de acciones
- La secuencia de las acciones tiene una única acción inicial
- Cada acción en la secuencia tiene un único sucesor
- La secuencia finaliza con cualquier solución al problema o en su defecto con un acuerdo comprobable que indique la imposibilidad de resolver el problema.

### **2.2.2 Historia de la composición algorítmica**

Esta se divide en 5 grandes etapas

- Pitagórica: Lo más destacable de esta etapa se basaba en lo siguiente: todas las cosas son números o todas las cosas se componen con números o todas las cosas se comportan como números. [7,13]

- Edad media: El ejemplo más temprano conocido de un método algorítmico aplicado a la composición musical y el primero documentado en el siglo XI fue desarrollado por el músico Guido D'Arezzo. Su método consistió en crear una correspondencia entre bonches de texto para asignar notas. [14,4]
- Periodo clásico: Otro ejemplo de los procesos matemáticos aplicados a la música antes de la era de las computadoras lo podemos encontrar en Mozart. La idea fue la de cortar y pegar medidas pre-escritas de música para crear un minueto and trio. Más tarde, en la era barroca se convirtió en una práctica regular para los compositores, incluso para aquellos que tenían un gran prestigio. Esta obra de Mozart ha sido ampliamente estudiada tanto por los compositores de música como por matemáticos y aficionados a la geometría musical, es conocida comúnmente como “Juego de dados musical de Mozart” [7, 15,16]
- Proporción aurea: Otro modelo matemático que ha sido utilizado en las artes a través de la historia es la proporción aurea. Esta puede ser expresada por el símbolo matemático en griego *phi*. Está definido como el punto que divide un segmento en dos partes, por ende la porción entre la porción más grande y la más chica es equivalente a la proporción entre el segmento entero y la sección más grande. Matemáticamente, si se toma un segmento de longitud 1 la proporción aurea es el punto  $x$  sobre el segmento que satisface la siguiente ecuación:

$$1/x = x/(1-x) \quad | \quad x^2 + 2 - 1 = 0$$

Gráficamente:

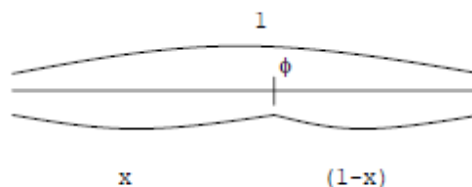


Figura 1. Representación grafica de la proporción aurea.

Esta proporción ha fascinado a muchos compositores a través del tiempo, algunos la han usado como instrumento de estructura principal. Si se considera la duración total de una pieza, el número áureo, es aproximadamente el 61.8 de su duración total.

Los compositores comúnmente reservan este momento para algún arreglo especial algo así como el “Clímax” de la canción o el momento más dramático. [7,17]

- Siglo XX: Los avances en el campo de las matemáticas y la ciencia en general desde mediados del siglo XIX han permitido a los compositores incorporar ideas y procedimientos dentro de su música que en otras épocas parecerían imposibles. En la actualidad la idea de la música como una arte-ciencia postulada por los antiguos griegos ha sido comprobada.

En los años 20's Joseph Schillinger, un compositor de música ucraniano desarrolló un sistema de composición musical, basado en principios científicos.

Sin embargo sus esfuerzos fueron eludidos, ya que la creatividad humana ha buscado alejarse de cualquier tipo de formalización. [7,18]

### **2.2.1.1 Procesos formales**

El uso de procesos formales para la composición de música siempre ha atraído compositores. En el siglo XI Gullaume Dufay usó un calor numérico para derivar parámetros musicales. [7] Algunos de sus trabajos están inspirados en las proporciones de la catedral de florentino, este valor numérico es conocido como “Numero áureo” Además Dufay anticipó técnicas que siglos después formaron parte de la conocida “Música serial” al aplicar procesamientos como la inversión y la transposición de secuencias de tonos.

Otra técnica formal comúnmente utilizada y probablemente la más famosa dentro de este campo sea la ya anteriormente mencionada “Juego de dados musical de Mozart”. [16, 19]

### **2.2.1.2 Máquinas de composición**

Estas son máquinas que han sido diseñadas con el propósito de facilitar los procesos dentro de los campos de síntesis y generación de sonidos mediante técnicas digitales y

computacionales. Algunas de estas técnicas formales han existido ya a través de los siglos como es el caso del “juego de dados musical” [16, 20]. Sin embargo para aplicar procesos automatizados en un contexto más avanzado es necesaria una máquina de procesamiento.

Un gran número de máquinas han sido utilizadas para propósitos de composición, éstas han creado y abierto nuevos caminos para la composición, síntesis y reproducción. Otro ejemplo destacable es el del inventor alemán Johann Maelzel, conocido como el inventor del metrónomo, él construyó el “Panharmonicon”, una orquesta de cuarenta y dos robots músicos. [4,21]

Algunas de estas máquinas fueron desarrolladas en las décadas anteriores, debido a ello cuentan con algoritmos menos sofisticados a los actuales, es decir no utilizaban autómatas celulares, redes neuronales, teoría caótica, etc. En su lugar utilizaban estructuras probabilísticas más rudimentarias. [22]

### **2.2.1.3 Nomenclatura**

De acuerdo a la naturaleza de los algoritmos existe un punto de separación entre sus dos principales arquitecturas:

#### **Música serial**

Mientras el desarrollo técnico estaba preparando terreno para la composición automatizada, nuevas tendencias se gestaban desde el mundo musical gracias a la inspiración producida por la composición algorítmica. El serialismo es una extensión de la música de 20 tonos y fue introducida por los compositores Anton Webern y Oliver Messiaen. Ellos generalizaron la técnica serial para cubrir otros parámetros también, tal como la duración y el timbre entre otros. [23]

#### **Música aleatoria**

En contraste con el *framework* y el estricto determinismo de la música serial, se encuentra del otro lado el modelo de la música aleatoria. Las reglas de composición desaparecen y los elementos de probabilidad juegan un rol importante. La aparición de la música aleatoria se

debe en gran parte al compositor John Cage, aunque podemos encontrar algunas aplicaciones anteriores. [24,25]

Aunque el serialismo y la música aleatoria se consideran modelos opuestos, ambas registran importantes tendencias después de la “guerra” de la historia de la música.

Ambas formaron un camino diferente para pensar en la música rompiendo un poco el esquema tradicional. Esas ideas han influenciado fuertemente la composición algorítmica y se reflejan en diversos métodos y técnicas que serán presentadas posteriormente en este trabajo. [25]

#### **2.2.1.4 Pioneros en la composición de música algorítmica**

- Edgard Varese: Jugó un rol importante en la afirmación de ver a la música como un arte-ciencia. El defendió la idea antigua de que el lugar de la música se sitúa junto con la geometría y las matemáticas. Se le considera también el padre de la composición electrónica. [1,19]
- Lejaren Hiller y Leonard Isaacson: En 1955 iniciaron sus trabajos de música algorítmica (Hiller e Isaacson, 1957,1958), basados en el modelo de generación y prueba. Crearon la que se considera la primera composición hecha completamente por una computadora. Fue publicado como “Illiac Suite”. Se basaron en cadenas de Markov para generar una nota pseudo-aleatoria dada una secuencia de notas previas que pasa una serie de pruebas heurísticas. Excluyen todo lo relacionado con la expresividad y el contenido emocional de la música.[16,19]
- Joseoh Schillinger: Experimentó desarrollar una teoría musical que pudiese ser útil con cerebros electrónicos.[18]
- Iannis Xenakis: Fue el primer compositor en adoptar un enfoque puramente matemático a la composición de música, no solo lo hizo como herramienta si no como filosofía de composición. Criticó los modelos determinísticos, argumentando que perdían el sentido después de un número de repeticiones, fue cuando introdujo el uso de la probabilidad en la composición dando lugar a los procesos de composición estocásticos.[19]

- Charles Ames: Propuso la utilización de las cadenas de Markov con diferentes tamaños de notas. [19]

## **2.3 Algoritmos de generación de música**

Los algoritmos empleados para la generación de música mediante sistemas computacionales son varios. Más adelante se explica a detalle cada uno de ellos, aunque sean algoritmos creados para un mismo fin es importante notar las grandes diferencias que tienen en su funcionamiento. [3] Es decir, algunos de ellos trabajan de forma determinística o no aleatoria cuyo comportamiento podría resultar predecible.

En el caso contrario están los no-determinísticos, estos actúan de manera aleatoria o pseudo-aleatoria, su comportamiento es ligeramente predecible en algunos casos debido a la probabilidad de que un evento ocurra, en otros casos es meramente caótico. [20,26]

### **2.3.1 Autómata celular**

Los autómatas celulares funcionan como redes de autómatas simples conectados localmente. Cada autómata simple produce una salida a partir de varias entradas, modificando en el proceso su estado según una función de transición. Por lo general, en un autómata celular, el estado de una célula en una generación determinada depende única y exclusivamente de los estados de las células vecinas y de su propio estado en la generación anterior. [20,27]

Este tipo de autómata es útil para modelar casi cualquier sistema en el universo. Pueden considerarse como una buena alternativa a las ecuaciones diferenciales y han sido utilizados para modelar sistemas físicos, como interacciones entre partículas, formación de galaxias, cinética de sistemas moleculares y crecimiento de cristales, así como diversos sistemas biológicos a nivel celular, multicelular y poblacional. [28]

#### **2.3.2.1 ¿Que es un autómata celular?**

Es una estructura matemática que evoluciona en función del tiempo, presenta una manera interesante para la composición de música. [20,21]



Este modelo es ideal para mostrar de manera gráfica la evolución de la música.

Un Autómata celular consiste en:

- Una matriz o arreglo de celdas que pueden estar en uno de un número finito de estados
- Una regla que define cómo los estados de las celdas son actualizados en función del tiempo

La matriz de celdas puede tener cualquier número de dimensiones. Teniendo en cuenta el estado de la celda y el estado de los vecinos en un tiempo  $t$ , la regla determina el estado de la celda en un tiempo  $t+1$ .

### **2.3.2.2 Historia de los autómatas**

La historia de los autómatas celulares a menudo se divide en 3 etapas, cada una de ellas corresponde al nombre de cada científico que hizo importantes aportes dentro de estos.

#### **Etapas de Von Neumann**

Este periodo está mayormente constituido por dos científicos. En la década de los 40's, Von Neumann buscó construir un autómata que se auto replicara, tomando como modelo inicial un robot que construyera otro robot (Modelo kinemático). Sin embargo este sistema mostró diversos problemas. Mientras Von Neumann lidiaba con estos inconvenientes su compañero Stanislaw Ulam se encargaba de investigar el fenómeno del crecimiento de los cristales mediante una red de rejillas. Él fue quien propuso a Neumann que utilizara un modelo matemático similar para la creación de sus autómatas. De esta manera Neumann construyó el primer autómata celular, basado en una rejilla de dos dimensiones, cada celda contenía hasta 29 estados y estos cambiaban según un algoritmo que establecía los parámetros en función de los vecinos colindantes de cada celda. Este sistema se llamó "Constructor universal de Von Neumann", con él se pudo demostrar que dado ciertos patrones estos se pueden reproducir de manera ilimitada. [7,22]

#### **Etapas de Martin Gardner**

Durante los años 70's Martin Gardner dio a conocer a gran escala los autómatas celulares mediante un artículo en *Scientific American* que mostraba un autómata realizado por John Conway conocido como "El juego de la vida". Este autómata está ampliamente explicado y goza de una gran popularidad en el campo de la computación. [20, 29]

### **Etapas de Stephen Wólfram**

Hacia el año 1983 este científico publicó una serie de artículos mostrando una gran cantidad de autómatas celulares que hasta el momento eran desconocidos, posteriormente se les llamó "Autómatas celulares elementales". A partir de estas investigaciones y al encontrar reglas relativamente sencillas que podían otorgar resultados complejos, Wólfram tuvo la idea de que la naturaleza funcionase de una manera similar.

Fue tan grande la convicción por el tema que se dedicó a estudiarlos de manera concienzuda, posteriormente escribió el libro *A new kind of science* donde muestra algunos de los descubrimientos que ha logrado mediante los autómatas celulares, y en el cual también postula su utilidad en todos los campos de la ciencia. Este libro causó revuelto en la comunidad científica, sin embargo no consiguió demostrar que mediante autómatas celulares se pueda crear un modelo físico elemental. [7,30]

### **2.2.3.3 Tipos de autómatas**

Los autómatas celulares pueden ser clasificados de diferentes maneras. Sin embargo cada una de ellas únicamente clasifica un aspecto en particular, esto da lugar a que exista más de una manera de clasificarlos, las principales son: [24]

- Serie o paralelo
- De acuerdo a su número de dimensiones

Los autómatas celulares son en teoría retículos compuestos de infinitos números enteros, lo cual es un problema a la hora de realizar aplicaciones prácticas de los autómatas celulares. Por ello, para dar cabida a los supuestos prácticos, debemos modificar la concepción teórica para dar lugar a retículos finitos. Esto nos obliga a concebir las condiciones de frontera: Las funciones de transición realizan modificaciones en una célula basándose en el estado de dicha célula y en las de su vecindad, pero si dicha célula se

encuentra en el borde, su vecindad queda reducida. ¿Qué hacer en estos casos? Se plantean varias opciones de actuación que la función de transición podrá considerar:

Se puede imaginar por ejemplo una frontera abierta, en la que se considera la existencia de células fuera de la cuadrícula con valores fijos (todas las celdas externas tienen el mismo estado); una segunda opción es usar una frontera reflectora, que usa los valores del borde de la matriz para las celdas exteriores; la tercera opción consiste en usar una frontera periódica, que se trata de considerar el sistema como una cuadrícula enrollada, haciendo que el conjunto de celdas sea continuo; y por último, existen la posibilidad de que el sistema no tenga frontera, haciendo que la cuadrícula crezca en función de las necesidades del autómata

### **Autómata celular unidimensional**

En una sucesión lógica, el autómata celular unidimensional es el primer paso para entender la dinámica de los autómatas celulares. Básicamente consiste en una sola fila o lista de células a las que se aplica un principio de vecindad básico de dos vecinos por célula y a los que igualmente se pueden aplicar las diversas reglas o condiciones de frontera que se han mencionado a lo largo de esta sección.

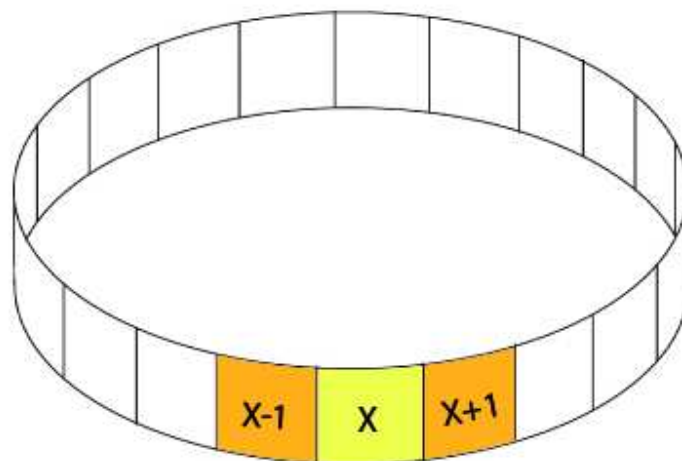


Figura 2. Autómata celular unidimensional con un radio de vecindad  $r=1$  y una condición de frontera periódico.

Como ejemplo, se puede tomar un autómata celular unidimensional con un radio de vecindad  $r=1$ , dos estados (0 y 1) y una condición de frontera de tipo periódico, como el que se muestra en la imagen. Se utiliza para este caso un tamaño de diez células y unas funciones de transición basadas en lo siguiente:

- Si ambos vecinos de la célula tienen el mismo estado, el estado de la célula a la que se aplica la función cambiará.
- Si ambos vecinos de la célula tienen distinto estado, el estado de la célula a la que se aplica se mantendrá igual.

En la figura 3 se plantean los valores iniciales 0001010011.

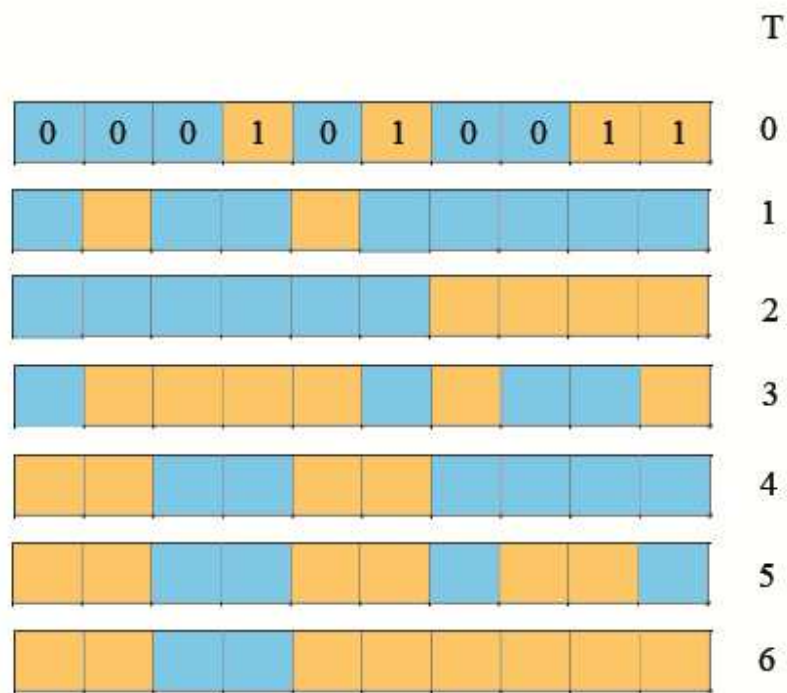


Figura 3. Primeras seis transiciones de un autómata celular unidimensional de 10 celdas (células).

Stephen Wólfraam clasificó el comportamiento de los autómatas celulares unidimensionales. Según Wólfraam, todo autómata celular pertenece a una de las siguientes clases [26]:

Clase I. La evolución lleva a una configuración estable y homogénea, es decir, todas las células terminan por llegar al mismo valor.

Clase II. La evolución lleva a un conjunto de estructuras simples que son estables o periódicas.

Clase III. La evolución lleva a un patrón caótico.

Clase IV. La evolución lleva a estructuras aisladas que muestran un comportamiento complejo (es decir, ni completamente caótico, ni completamente ordenado).

### **Autómatas celulares Bidimensionales**

Es el más frecuente de los autómatas celulares en cuanto a simulación y estudio. Consiste en una cuadrícula compuesta por células ( $z \times z$ ), implantada por Von Neumann por consejo y Stanislaw Ulam. Este tipo de autómata celular, por ser el más estudiado, ha sido variado en múltiples ocasiones, modificando las reglas de su evolución y por lo tanto su comportamiento. Ver figura 4.

Considerando el principio de vecindad de Von Neumann, que implicaba cuatro vecinos para cada célula, pero este no es el único. En 1970 se dio a conocer el juego de la vida de John Conway, que se basa en un principio de vecindad diferente. Este principio se conoce con el nombre de vecindad de Moore, y aplica ocho vecinos a cada célula, es decir, los cuatro que consideraba Von Neumann más las cuatro células de las diagonales respectivas. Además, se basaba en una condición de frontera de tipo periódico y dos estados posibles. Este juego de la vida consideraba tres reglas fundamentales para su funcionamiento [31]:

- Nacimiento: se reemplaza una celda muerta por una viva si dicha celda tiene exactamente 3 vecinos vivos.

- Muerte: se reemplaza una celda viva por una muerta si dicha celda no tiene más de 1 vecino vivo (muerte por aislamiento) o si tiene más de 3 vecinos vivos (muerte por sobrepoblación).
- Supervivencia: una celda viva permanecerá en ese estado si tiene 2 o 3 vecinos vivos

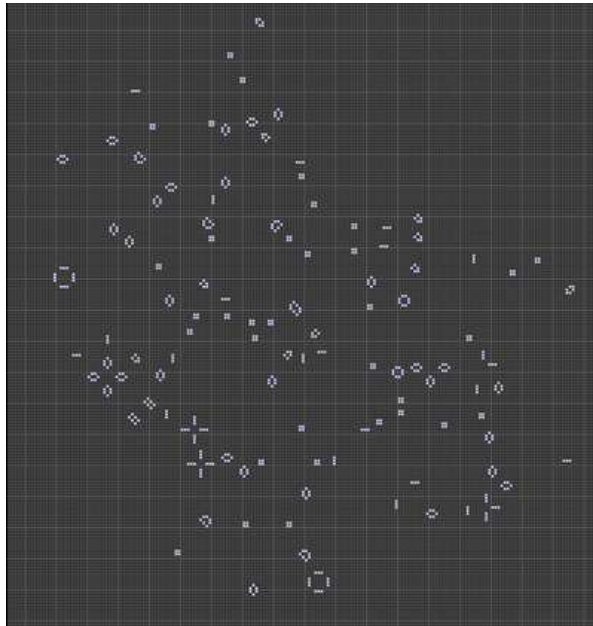


Figura 4. Comportamiento típico de un autómata celular bidimensional.

Producto del amplio estudio que han realizado múltiples expertos y estudiantes, se han descubierto y catalogado una serie de patrones característicos. Se diferencian entonces cuatro tipos de patrones: estáticos, patrones que no varían; recurrentes, patrones que varían de una posición a otra indefinidamente; patrones con movilidad, que se reproducen y se mueven por toda la cuadrícula; matusalenes, que son patrones que tardan muchos pasos en estabilizarse.

### 2.3.2 Fractales

A grandes rasgos un fractal es un objeto semi-geométrico en cuya estructura (fragmentada o irregular) se repite indefinidamente y a diferentes escalas a esto se le denomina auto referencia. También presentan el fenómeno de auto semejanza, esto implica que el objeto

fractal presenta la misma apariencia independientemente del grado de ampliación con el que sea visto. Desde los años 20's Schillinger venía considerando la música con cierta naturaleza caótica y recursiva, una imagen fractal [27]

### **Aplicación**

Técnicamente una imagen fractal está formada por infinitos puntos en un plano, si a cada punto de estos le asignamos una nota el resultado de esto será música fractal.

Por otro lado se ha demostrado que existen fractales dentro de la música clásica, un ejemplo de esto es el Scherzo de Beethoven. Esta pieza está construida sobre un compás de 3 tiempos y tiene un patrón claramente identificable. Dos corcheas y una negra como acorde desplegado en forma descendente, el principal motivo consiste en 2/8 (dos corcheas) agregadas en conjunto en 1/4 (una negra) como acorde. Este motivo ternario (porque está compuesto por 2 corcheas y una negra ( $2+1=3$ )), es repetido en todas las partes del scherzo [6,31]

### **2.3.3 Redes neuronales**

Las redes neuronales artificiales ofrecen una ventaja alternativa a la tradicional teoría de computación mayormente usada en computadoras basadas en el estándar de arquitectura Von Neumann. Rompen el esquema de procesamiento centralizado, sustituyéndolo por un sistema de procesamiento distribuido. La mayor ventaja de estos sistemas es la de aprovechar su capacidad para aprender a partir de ejemplos proporcionados. Por lo tanto no es necesario analizar el problema de manera explícita para formular funciones y subrutinas que produzcan los resultados deseados. [24]

Las redes neuronales están inspiradas en el cerebro humano que se forma de redes de millones de neuronas. Cada una de ellas es considerada una unidad de procesamiento independiente mientras que también existen diferentes tipos de neuronas [25]. Por lo tanto las redes neuronales artificiales utilizan el mismo principio, estas fueron introducidas en 1958 por Frank Rosenblatt. En su modelo había un número de conexiones entrantes que fungían como sinapsis de una neurona real. [29]

### 2.3.4 Cadenas de Markov

Anteriormente se mencionaron los procesos aleatorios de composición, estos sistemas no estaban condicionados probabilísticamente, por lo tanto los eventos pasados no influían en la sucesión generada. Pero con las cadenas de Markov entra en función el estado actual para continuar la secuencia de los eventos, básicamente se podría definir como la probabilidad de que un evento ocurra dependiendo del evento inmediatamente anterior. [7,32]

#### Matriz de transición

Los cambios de estados (en cadena) son conocidos como transiciones y una cadena de Markov puede ser expresada por una matriz de transiciones probables. Ver figura 5.

Source state	Destination state			
	C	D	F	G
C	0.1	0.1	0.4	0.4
D	0.7	0	0	0.3
F	0.3	0.2	0	0.5
G	1	0	0	0

Figura 5. Tabla de probabilidad de transición entre estados utilizando cadenas de Markov.

### 2.3.5 Sistemas caóticos

La teoría caótica aplica en diversos campos de la naturaleza. La idea elemental del caos es que una simple semilla puede florecer hasta formar una estructura compleja. Este modelo se caracteriza por un comportamiento no lineal, el término caótico es utilizado para describir un *output* generado bajo ciertas condiciones [24, 32].

Aunque a menudo se asocia el término caótico con una conducta aleatoria e indeterminada en este caso los sistemas caóticos son estrictamente determinísticos. La composición de música caótica comienza aproximadamente a finales de la década de los 80's. En dicha época empezaron a aparecer las primeras investigaciones relacionadas con el análisis de las estructuras fractales contenidas en la música clásica. [25,30]



En cuanto al funcionamiento, las melodías caóticas son definidas como “la producción a escala de una función iterada, siempre que no se escape hasta el infinito, o tienden hacia cero” en otras palabras, la iteración matemática es considerada una melodía caótica verdadera si eventualmente presenta un patrón repetitivo [1,9]. Ya que un sistema caótico puede ser descrito como un mapa iterado, una función interactiva en la cual el resultado de una iteración se convierte en el *input* para calcular la próxima iteración, tal como un flujo continuo parecido a un esquema de ecuaciones diferenciales.

### **2.3.6 Gramáticas formales**

Una gramática formal es una estructura matemática con un conjunto de reglas de formación que definen las cadenas de caracteres admisibles en un determinado lenguaje formal o lengua natural. Las gramáticas formales aparecen en varios contextos diferentes: la lógica matemática, las ciencias de la computación y la lingüística teórica, frecuentemente con métodos e intereses divergentes. [7,33]

En un lenguaje formal, a las cadenas formadas según las reglas de la gramática formal se las llama fórmulas bien formadas, y el conjunto de todas las fórmulas bien formadas constituye un lenguaje formal. Una gramática formal no describe el significado de las fórmulas bien formadas, sino solamente su forma. La teoría de los lenguajes formales estudia las gramáticas formales y los lenguajes formales, y es una rama de la matemática aplicada. [7,9]

#### **Elementos**

Está compuesto por una serie de categorías sintácticas que se combinan entre sí por medio de unas reglas sintácticas que definen cómo se crea una categoría sintáctica por medio de otras o símbolos de la gramática. Existe una única categoría superior que denota cadenas completas y válidas.

## Capítulo 3. Materiales y Métodos

Este capítulo documenta el aspecto de la implementación del proyecto. Se analizan en forma general las herramientas disponibles y se describen en detalle las elegidas así como los procesos que facilitan la construcción del prototipo planteado a lo largo de los dos capítulos anteriores. Otro aspecto relevante de este capítulo es el desarrollo de un plan de acción que permite organizar mediante una secuencia de pasos las actividades necesarias para el desarrollo del prototipo. De la misma manera este plan nos ayudará, al término del trabajo, a encontrar las respuestas a las preguntas de investigación planteadas al inicio.

### 3.1 Descripción del área de estudio

Ya que el proyecto está dentro del área de la música y las matemáticas y al estar dirigido a usuarios que estén interesados en conocer la creatividad computacional, se considera un proyecto experimental y de innovación, ya que proveerá al usuario final una pista audible que haya sido compuesta completamente por un algoritmo matemático, es decir sin la intervención directa de una persona. [34]

Los pasos a seguir para el desarrollo del prototipo se resumen en los siguientes puntos básicos:

- Investigación acerca de los lenguajes de programación que se adapten a las actuales necesidades
- Conocer los modelos matemáticos disponibles y una vez elegido el lenguaje de programación se opta también el modelo matemático con el que se trabajará
- Realizar pruebas con el código utilizando las librerías de terceros para conocer su funcionamiento en la práctica
- Dependiendo de los resultados que hasta este punto se han obtenido se evalúa en base a la opinión de personas si es apto para su uso en la composición de música
- Describir las limitaciones existentes.

### 3.2 Materiales

Los materiales requeridos por la presente investigación han sido escogidos en base a una exploración previa, antes de describirlos en este capítulo se explica de manera breve y clara

el porqué de la selección de los presentes materiales considerando diversas posibilidades que existen en el campo de la programación orientada a satisfacer áreas especializadas, en el caso actual esta especialización es en el campo de la música mediante el poder de computo. [35]

En este capítulo también se discute el uso de un lenguaje de programación para el desarrollo del prototipo propuesto, esta elección es tomada tomando en cuenta los siguientes criterios:

- Compatibilidad
- Soporte
- Documentación
- Facilidad de uso

Durante la etapa de investigación se encontraron diversos lenguajes que cubrían los requerimientos básicos anteriores. Los lenguajes considerados fueron Cmusic, Java, Chuck y Common Lisp Music, entre otros.

La mayoría de estos lenguajes de programación fueron descartados en primera instancia debido a que están más enfocados a la síntesis de audio y a la programación en tiempo real, es decir, un lenguaje que está orientado para servir eventualmente como un instrumento de síntesis de audio e incluso acorde a los comentarios de sus autores estos pueden competir cómodamente con herramientas especializadas para la síntesis de sonidos.

Considerando lo anterior así como también los requerimientos básicos mencionados, se eligió el lenguaje de programación Java para desarrollar el prototipo propuesto en el objetivo de este documento. Este lenguaje de programación cuenta con librerías que facilitan el desarrollo, nos proporcionan herramientas bastante útiles como escribir un archivo MIDI y asociar un output numérico con una nota musical.

Dentro de esta sección se mencionan los requerimientos técnicos que son necesarios para el desarrollo del prototipo acústico, cuyo motor será un autómata celular:

- PC con sistema operativo compatible con Java
- Java Development Kit 6 (o posterior)
- Librerías “jmusic”
- Quicktime plugin (opcional)
- Tarjeta de audio o audio integrado
- Bocinas

### 3.3 Métodos

Esta sección se enfoca a explicar la implementación de la estructura matemática elegida para ser el motor del prototipo. Durante la etapa de investigación se encontró que había diversos métodos que pueden ser implementados para la generación de música. Un punto importante para aclarar por qué utilizar un algoritmo que genere un *output* de sucesiones musicales o rítmicas es sencillamente por lo siguiente: aunque en términos meramente técnicos se pueden generar sonidos bajo cualquier función simple tal como “random” estos sonidos carecerían en su totalidad de ritmo y sentido armónico.

#### 3.3.1 Tipo de investigación

El diseño de la investigación es de tipo experimental, ya que se investiga lo suficiente tanto de algoritmos como de música para poder plantear un argumento que los enlace dentro del área de la música computacional. Primeramente se describe en términos generales el problema con el fin de identificar el conocimiento que es necesario adquirir para atender el problema de estudio. Los resultados que posteriormente se obtienen son de carácter cualitativo, como el considerar opiniones de los usuarios finales acerca del *output* de sonidos del autómata celular. [36]

Se conoce acerca de este tipo de investigación que su objetivo es documentar ciertas experiencias, examinar temas o problemas poco estudiados o que no han sido abordados

antes. Por lo general se investigan tendencias, se identifican relaciones potenciales entre variables y se establece el “tono” de investigaciones posteriores más rigurosas. Estas investigaciones se efectúan por lo regular cuando el objetivo es examinar un tema o problema de investigación poco estudiado.

Esta clase de estudios es común regularmente en la investigación del comportamiento, sobre todo en situaciones donde hay poca información. Los estudios exploratorios en algunas ocasiones constituyen un fin en sí mismos. [37]

### **3.3.2 Metodología**

El autómata celular unidimensional

Por lo anterior se precisa de un algoritmo que simule hasta cierto punto la creatividad humana o natural tal como lo es un autómata celular, que en su origen buscaba modelar fenómenos físicos y biológicos.

Dicho modelo ya ha sido explicado a mayor detalle en el marco teórico, de manera que ahora que se ha hecho la elección lo siguiente consiste en implementarlo con el lenguaje de programación Java.

Un autómata celular o AC puede ser utilizado para generar melodías y ritmos, esto debido a que es posible utilizarlo en el campo del audio para dar lugar a sonidos que presenten un comportamiento específico.

#### **¿Cómo funciona el autómata celular unidimensional?**

A continuación se explica de manera sencilla el funcionamiento de un autómata celular aplicado a la generación de sonidos.

El Autómata se compone de un arreglo (de celdas), este arreglo es llenado con números aleatorios. Un método re-circulatorio escanea las celdas inmediatamente próximas y aplica reglas a los números encontrados. Una celda y sus dos celdas vecinas son utilizadas para escoger una regla en la tabla de búsqueda.

El código desarrollado asocia el *output* de nuestro autómeta a patrones rítmicos, es decir, cada celda en el arreglo representa una semi-corchea (1/16 de redonda), en el arreglo final de Java llamado “Score” las celdas que generan sonido son las celdas vivas mientras que los silencios son producidos por las celdas muertas.

Las reglas que son aplicadas para actualizar las celdas en cada iteración son la clave para que el estilo musical tenga lugar. Estas reglas son las de un autómeta celular tradicional basado en el número de celdas vecinas. Musicalmente esto se refiere a la densidad del ritmo si consideramos el siguiente fenómeno: entre más vecinos tenga una semicorchea, más densa es el área del arreglo lineal de celdas en dicha región. Por lo tanto es una buena práctica el variar las reglas para buscar generar patrones más rítmicos que otros.

En el programa propuesto por esta investigación, el autómeta celular unidimensional respeta las siguientes reglas dependiendo del número de vecinos activos la celda morirá o sobrevivirá a la siguiente generación, de acuerdo a los siguientes criterios:

- Si tiene 1 o 2 vecinos permanece igual
- Si tiene 3 vecinos y está viva cambia de estado
- Si tiene 4 vecinos muere por sobrepoblación.

Cabe destacar que un arreglo temporal es utilizado para almacenar los valores de la siguiente iteración durante el proceso, una vez completada la nueva generación estos valores son transferidos a un nuevo arreglo que se vuelve a su vez el estado actual de nuestro autómeta. Por lo tanto las reglas anteriores interactúan directamente con la metodología.

El método con el que se trabaja es lógicamente simple ya que dentro del *framework* de *jmusic* existen herramientas para el almacenamiento de nuestro *output*. El modelo es el siguiente: El autómeta genera notas, esas notas son agregadas a estructuras llamadas

“phrases” y estas últimas son agregadas a su vez a estructuras de mayor capacidad de almacenamiento llamadas “Parts”. De esta manera se estructura un arreglo de notas que puede ser tocado y almacenado en un archivo de audio.

### 3.3.3 Implementación

Una vez que se cuente con los materiales requeridos en el apartado anterior y se tenga bien identificado el método con el que se busca trabajar es tiempo de poner manos a la obra para poner en marcha todo lo que hasta este punto es meramente teórico.

En esta sección únicamente se explican partes clave del código desarrollado bajo Java. El código fuente entero, así como el archivo de audio resultado del mismo se incluirán en un CD-ROM.

Declaraciones:

```
public class MiAutomata implements JMC{  
    int longitudarreglo = 10;  
    int[] lon = new int[longitudarreglo];  
    int[] arreglotemporal = new int[arreglotemporal];  
    Part sonido1 = new Part("golpe", 0, 3);  
    Part sonido2 = new Part("bombo", 0, 4);  
    Part sonido3 = new Part("ritmo", 0, 4);  
    Score score = new Score("FinalScore", 90);
```

El código anterior es un código regular hecho en el lenguaje de programación Java en el que se declaran tres variables enteras, la primera es longitudarreglo a la cual se le asigna el valor de 10, las otras dos son arreglos de longitud 10. Hasta esta parte no hay nada nuevo. Nótese también que en el siguiente renglón se utilizaron dos declaraciones que no existen por default en Java. “Part” es un tipo de dato que almacena estructuras de sonido más pequeñas (phrases). “Score” funciona exactamente igual que “Part”, con la única diferencia de que esta puede almacenar en ella un gran número de sonidos.

Reglas del autómata:

```
private void nextBar() {
    int unos = 0;
        //Calcular numero de vecinos
    for(int i=0; i<longitudarreglo; i++) {
        for(int j=1; j<3; j++) {
            if (i-j > 0) unos += arreglo[i - j];
            if (i+j < longitudarreglo) unos += arreglo[i +j];
        }
        // Aqui se aplican las reglas
        if (unos == 0) {
            if (arreglo[i] == 1) arreglotemporal[i] = 0;
            else arreglotemporal[i] = 1;
        }
        if (unos == 1) arreglotemporal[i] = arreglo[i];
        if (unos == 2) arreglotemporal[i] = arreglo[i];
        if (unos == 3) {
            if (arreglo[i] == 1) arreglotemporal[i] = 0;
            else arreglotemporal[i] = 1;
        }
        if (unos == 4) arreglotempral[i] = 0;
        unos = 0;    }
    arreglo = arreglotempral;    }
```

Esta parte es sin duda un pilar muy importante para el programa. Dentro de este método las reglas son aplicadas. El número de celdas vivas, tres lugares a cada lado de la celda actual, son contabilizadas. Dependiendo del número de vecinos vivos cada celda nacerá, morirá o sobrevivirá a la siguiente generación.

Una vez que se tenga todo el código fuente del programa, lo siguiente es la compilación. Un punto importante a señalar aquí es el hecho de que para este proyecto no se hace uso de software de compilación para java (tales como netbeans o eclipse) esto debido a que se utilizan librerías externas que no pertenecen por naturaleza a Java. Por lo tanto el código fuente se compila bajo la máquina virtual de Java que puede ser invocada bajo la terminal de cualquier sistema operativo compatible con JDK.

Antes de acceder a cualquier sentencia de ejecución o compilación se recomienda incluir las siguientes variables dentro del *classpath* de nuestro sistema operativo. Debido a



que el proyecto es desarrollado bajo Windows 7, aquí se especifican las instrucciones para poder compilar bajo la línea de comandos de dicho sistema operativo.

1. Click derecho en “mi pc”, y acceder a propiedades.
2. Una vez en las propiedades del sistema buscar configuración avanzada y hacer click en esa opción.
3. En la ventana emergente localizamos el botón “Variables de entorno”.
4. Se hace click en el botón “agregar” en la sección variables de entorno.
5. Se asigna a la variable el nombre de “CLASSPATH” y el valor de “C:\jmusic;C:\jmusic\jmusic.jar”
6. Se hace click en aceptar

Finalmente han sido creadas las variables necesarias para trabajar en el directorio C:\jmusic utilizando la librería de jmusic.

Para compilar cualquier programa de Java en la terminal las indicaciones son:

- Abrir la línea de comandos
- Dirigirse al directorio C:\jmusic
- Utilizar el comando “Javac”, por ejemplo si se quiere compilar un archivo miprograma.java la sintaxis es la siguiente:
  - javac miprograma.java

Como resultado de lo anterior se genera un archivo miprograma.class, este archivo es un código compilado de java, en otras palabras es escrito en lenguaje maquina por lo que no puede ser más que ejecutado con la sintaxis siguiente:

```
Java -classpath “C:\jmusic;C:\jmusic\jmusic.jar” miprograma
```

Con la anterior sentencia se ejecuta el archivo compilado, una vez ejecutado correrá bajo la línea de comandos.

En la figura 6 se presenta la estructura general para dar lugar a una generación mediante un autómata celular, lo cual es el propósito general de la presente investigación.

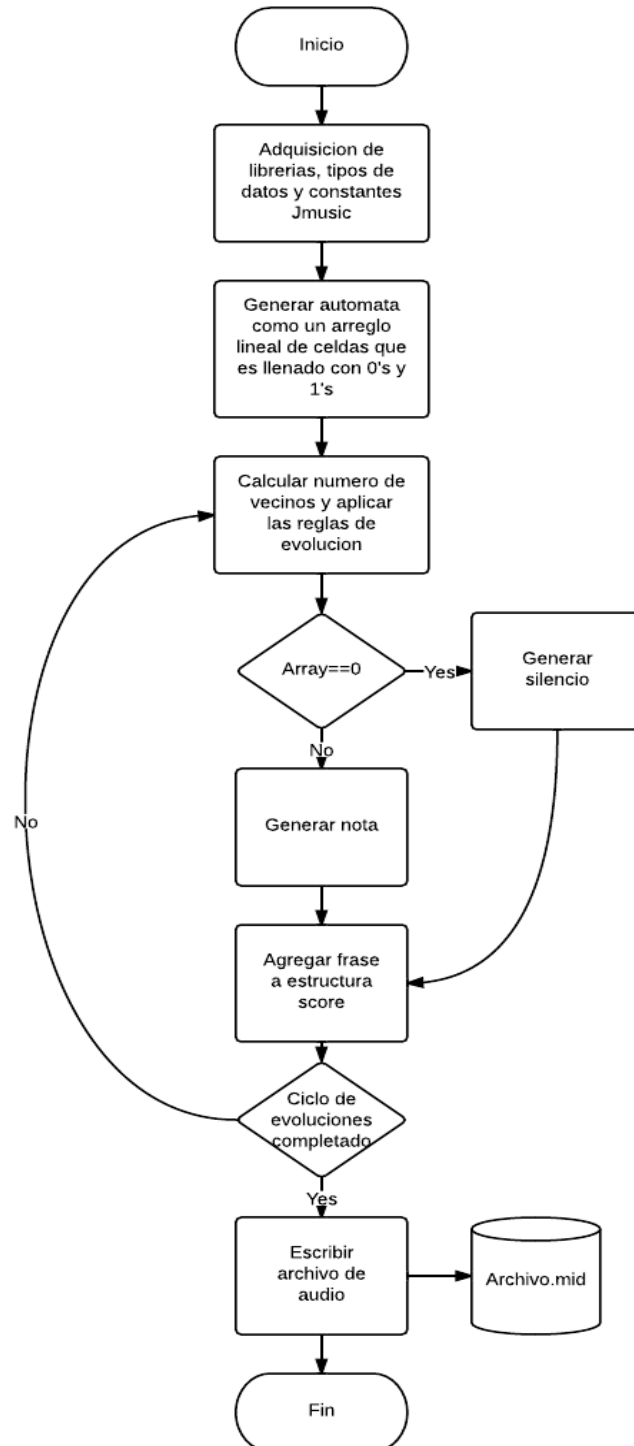


Figura 6. Estructura general de generación y almacenamiento de sonido utilizando un autómata celular.

Siguiendo la estructura propuesta por la figura 6 es posible armar estructuras matemáticas, hasta este punto no es muy diferente de cualquier otro lenguaje de programación salvo que al utilizar las librerías propias del framework utilizado es posible hacer una conversión de un output numérico a nota. Siguiendo un principio tan sencillo como asignar silencios cuando se encuentre un 0 en la cadena o en caso contrario cada que exista un 1 en la cadena asignamos un sonido.

Finalmente se escribe el archivo de audio en un formato reproducible en cualquier computadora, este último es el producto final de la aplicación desarrollada. Aquí infieren parámetros variables como la longitud, bajo diferentes condiciones se pretende obtener diferentes resultados.

En la tabla 1 se muestran las notas que pueden ser reproducidas por el algoritmo de composición musical desarrollado, estas van desde 0 a 127, siendo 0 equivalente a DO en su tono más grave y 127 sol en su tono más agudo. De esta manera el audio es estandarizado para relacionar valores numéricos con frecuencias musicales.

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

Tabla 1. Rango de notas MIDI y su relación con la escala musical

## Capítulo 4. Resultados de la investigación

Este apartado explica a detalle los resultados que se obtuvieron a partir del prototipo de composición musical desarrollado, se analizan estos resultados y se atienden las preguntas de investigación propuestas y que han servido como guía en el actual proyecto. Auxiliando y clarificando no solo los resultados sino también el funcionamiento real de los métodos aplicados, de esta manera se observa si la teoría dio exactamente los resultados esperados en la práctica. A lo largo del capítulo también se analizan los fragmentos. Se revisan los valores a ingresar y que pasaran a través del algoritmo para hacer un análisis del comportamiento melódico y exponerlo para buscar una mejor comprensión.

### 4.1 Presentación de resultados

Los resultados obtenidos después de una serie de pruebas son concluyentes, se consiguió generar numerosos archivos de audio, cada uno de ellos presenta condiciones iniciales diferentes y por lo tanto el resultado final es variable pero cabe resaltar que el motor utilizado es siempre el mismo. A continuación son explicados algunos de los archivos generados.

Los archivos de audio se anexan también en un CD-ROM junto con la aplicación principal, las respectivas librerías que son requeridas para ejecutar y compilar el código en JAVA.

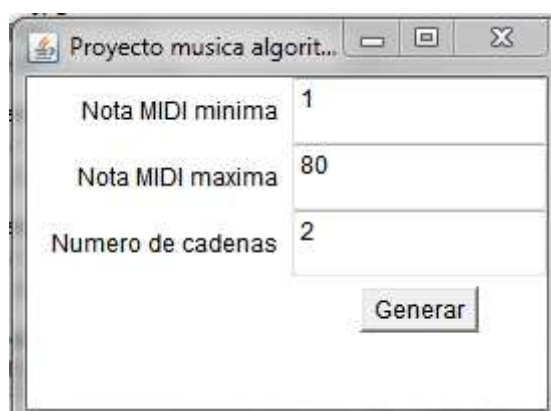


Figura 7. Interfaz gráfica del usuario

En la figura 7 se presenta la interfaz con la que se trabaja para utilizar el generador de fragmentos musicales. Como puede observarse tiene 3 campos de texto en los cuales se pueden ingresar valores numéricos para variar las condiciones iniciales del generador y por tanto contar también resultados variables y auditivamente diferentes.

El primer campo tiene la etiqueta de “Nota MIDI mínima” es decir el valor numérico que se ingresa en este campo será la nota más baja o de más baja frecuencia que se permite tocar en el fragmento musical.

El segundo campo cuenta con la etiqueta “Nota MIDI máxima” tiene una función muy parecida al anterior campo solo que este almacena el valor numérico de la nota más alta o de mayor frecuencia que se permite agregar en la generación del fragmento.

El último campo lleva por etiqueta “Número de cadenas” en esta campo se asigna un valor numérico para establecer el número de cadenas a ser tocadas, cada cadena tiene una longitud de 16 tonos, tomando en cuenta que un tiempo se compone de 4 semicorcheas es necesario incluir también 4 tiempos para completar un compás de 4/4.

Por último se tiene el botón “generar” al pulsar este botón los valores numéricos que estén asignados en los campos de texto anteriores pasan al algoritmo de generación de fragmentos musicales desarrollado para finalmente escribir un archivo MIDI con las condiciones preestablecidas.

## **4.2 Análisis e interpretación de resultados**

A continuación se presentan capturas de pantalla de diferentes ejecuciones que se realizaron en el algoritmo de composición musical desarrollado, posteriormente se da una breve explicación del resultado obtenido y se incluye el archivo de audio en un CD-ROM en el directorio D:\Archivos de audio\

## 4.2.1 Ejecución 1

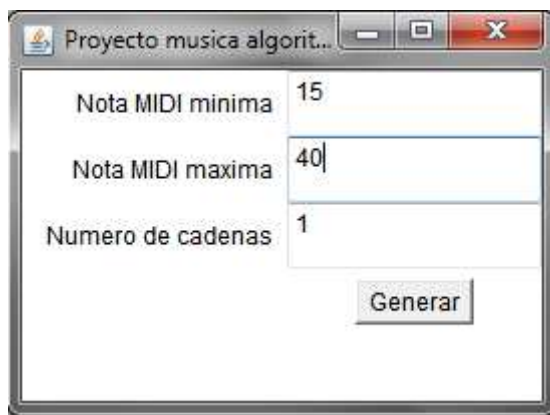


Figura 8. Captura de pantalla con valores elegidos arbitrariamente No.1

El resultado obtenido al ejecutar el algoritmo de composición musical con los valores mostrados en la figura 8 da por resultado un archivo MIDI compuesto por 12 sonidos audibles. Esto se explica debido a que la cadena generada en primera instancia tiene el mismo número de ceros y de unos, al ser transformados los ceros en silencios y los 1 en notas obtenemos el resultado actual que es puesto en la carpeta mencionada al principio del apartado bajo el nombre de "Prueba-4-2a.mid". Ver figura 9.

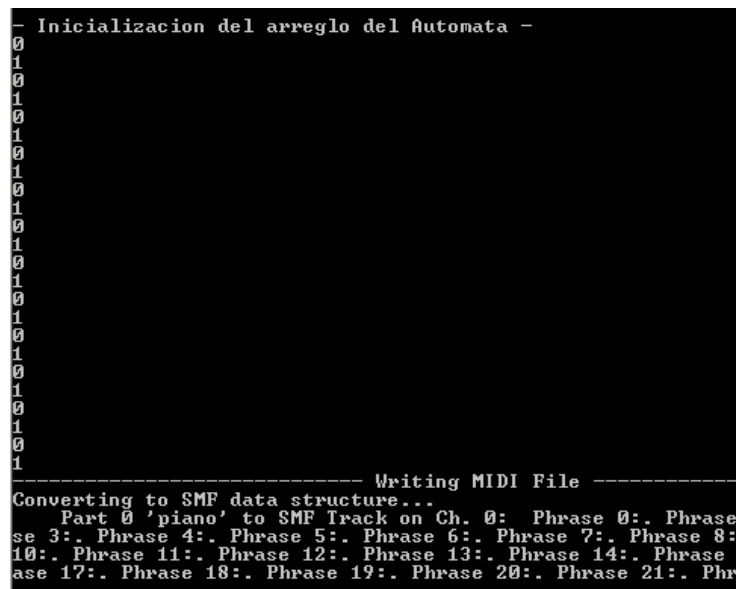


Figura 9. Cadena de ceros y unos generada durante la ejecución 1

Es importante resaltar que esta primera cadena no ha sido afectada por las reglas evolutivas propias del autómata celular unidimensional, ya que como se puede ver en la figura 8 el parámetro de número de cadenas tiene un valor de uno. Por lo tanto esa cadena no genera una nueva para que lo anterior suceda es necesario establecer un mínimo de dos cadenas.

#### 4.2.2 Ejecución 2

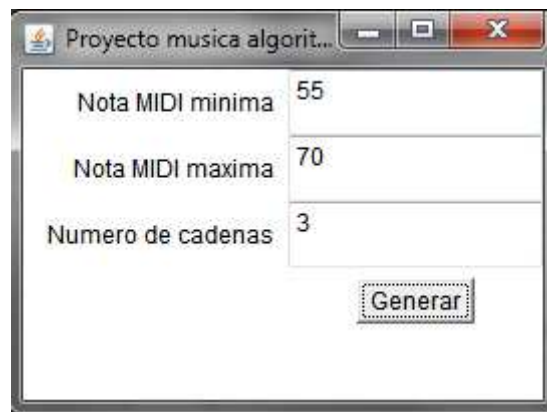


Figura 10. Captura de pantalla con valores elegidos arbitrariamente No.2

Como se puede observar en la figura 10 para ejecutar por segunda vez el algoritmo de composición musical se establecen como parámetros iniciales números distintos con respecto a la anterior ejecución, esto servirá para corroborar que dependiendo de las condiciones iniciales del algoritmo los resultados igualmente varían en más de un sentido.

El valor de 55 establecido en el campo de Nota MIDI mínima corresponde a la nota más grave que se permite utilizar en la ejecución del algoritmo. Por otro lado el valor de 70 establecido en el campo de texto Nota MIDI mínima corresponde a la nota más aguda de la cual el algoritmo puede hacer uso.

Finalmente se observa que el número de cadenas utilizadas para esta ejecución es de 3, es decir las reglas de evolución del autómata se aplicaran a partir de la primera tercera parte de la cadena, y por ende tendrá también una mayor duración el audio generado. El archivo de audio obtenido de esta ejecución se encuentra en el CD-ROM

bajo del nombre de “Prueba-4-2b.mid” y puede ser reproducido para su mejor entendimiento.

### 4.2.3 Ejecución 3

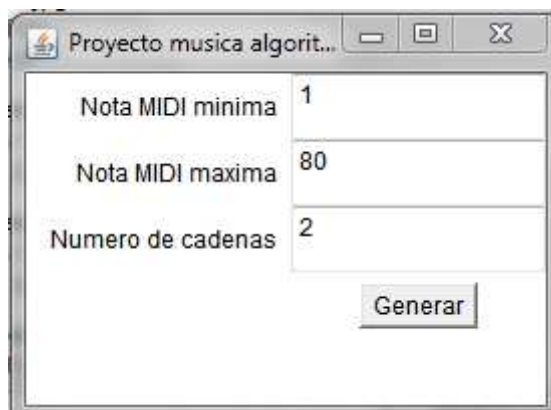


Figura 11. Captura de pantalla con valores elegidos arbitrariamente No.3

Como tercera prueba se establecen los valores mostrados en la figura 11 para hacer uso del algoritmo de composición musical diseñado, estos valores a diferencia de los establecidos en las anteriores ejecuciones permiten un espectro muy amplio de sonidos disponibles.

De manera similar a las ejecuciones pasadas los valores puestos en los campos de texto Nota MIDI mínima y Nota MIDI máxima establecen los parámetros de nota más aguda y más grave respectivamente. Así como también se observa que el número de cadenas utilizadas para esta ejecución es de 2, por lo tanto las reglas de evolución del autómata se ejecutan una vez. El archivo de audio obtenido de esta ejecución se encuentra en el CD-ROM bajo el nombre de “Prueba-4-2c.mid” y puede ser reproducido para su mejor entendimiento.

Hasta este punto el algoritmo de composición musical ha sido probado en distintos escenarios, mostrando como las condiciones iniciales que pueden ser establecidas repercuten directamente en la generación del fragmento musical.



## **Capítulo 5. Discusiones, conclusiones y recomendaciones**

En el presente capítulo se exponen las conclusiones generales obtenidas a partir del trabajo elaborado. También se hace mención del trabajo que se recomienda realizar eventualmente como trabajo a futuro.

### **5.1 Con respecto a las preguntas de investigación**

Las preguntas de investigación siempre sirvieron como una guía que indicó el camino apropiado a seguir en la presente investigación, finalmente en este apartado se responden bajo el contexto de la presente investigación.

Con respecto a la composición automatizada se define como una manera de composición musical que es generada o auxiliada por métodos desarrollados para este mismo fin. Estos van desde las máquinas de composición que han sido desarrolladas en distintas épocas, existe cierta ambigüedad en lo que refiere composición automatizada, cabe resaltar que aunque hay conceptos muy cercanos, como lo es la síntesis de música, que consta de herramientas para modular el audio e incluso transformarlo drásticamente este concepto no es lo mismo que la composición automatizada ya que carece de ser un motor de creatividad no humano.

Por otra parte existe la cuestión de considerar algún algoritmo más apropiado que otro para crear secuencias de sonido, es decir porque razón es más apropiado utilizar un autómata celular unidimensional que un algoritmo de modelo caótico. La respuesta es bastante subjetiva, los sonidos generados por cada modelo no divergen demasiado, carecen casi en su totalidad de armonía al ser acompañados por un segundo instrumento. En cuando a melodía también suelen generar sonidos que pueden o no ser considerados melódicos, dependen en un porcentaje considerable de la aleatoriedad, esto es en gran parte por la limitada capacidad de transformar una salida lógico-numérica en notas musicales.

El caso de los autómatas celulares tiene una particularidad muy importante, y esta es que toda salida de audio depende de una cuestión llamada “mapping” esta es básicamente la transformación a una nota y es también establecida por quien diseño el autómata o lo adapte a la generación de música, un método común es el de asociar a cada celda del autómata un tono, al cobrar vida dicha celda producirá el sonido

preestablecido. Otro modelo de mapeo puede ser también el de utilizar un contador, en cada iteración del autómata son contadas las celdas vivas y este número de celdas vivas es asociado a una nota también.

Por lo anterior se concluye que el algoritmo comparte importancia con la regla de mapeo, si se tienen conocimientos avanzados en teoría musical es posible crear un mapeo que genere melodías, y de esto se puede hablar con seguridad ya que ha sido descubierto por grandes compositores, el mismo Mozart describió un juego de dados que consistía en la composición de una obra musical a partir de la combinatoria de compases preestablecidos y enumerados del 1 al 176 y agrupados en 16 conjuntos de 11 compases cada uno y pueden ser tocados jugando con todas sus posibles combinaciones.

Con respecto a las ventajas principales encontradas en utilizar un autómata celular unidimensional es que estos han sido diseñados para simular sistemas biológicos y por lo tanto presenta comportamiento natural que puede diferir considerablemente de un patrón aleatorio. Además de que como se menciona más arriba puede ser implementado un mapeo mucho más complejo que el utilizado actualmente.

En relación a los resultados obtenidos se encuentra que un posible uso que puede ser dado al algoritmo generación musical diseñado es el de servir como estructura de creatividad no humana, incluso si es implementada una mejor técnica de mapeo puede ser una herramienta bastante útil para los compositores ya que tiene la capacidad de generar melodías bajo creatividad computacional.

En cuanto a las limitantes encontradas el autómata celular se encuentra desde luego el contar con un mejor mapeo, pero esto implica serias complicaciones al momento de programar ya que se puede volver bastante compleja la relación entre matemáticas y música, otra limitante a considerar es la tendencia de hacer del mapeo algo subjetivo al contexto cultural, es decir el sonido generado puede adoptar un patrón muy “occidental” dado que ha sido desarrollado por una persona de occidente y al ser escuchado por una persona que no comparta esas tendencias culturales puede sonar menos melódico.

## **5.2 Con respecto al objetivo de la investigación**

El objetivo original de la actual investigación es el de desarrollar un prototipo que basado en un algoritmo matemático genere una sucesión de notas musicales. Este

objetivo ha sido completado al poner en funcionamiento el algoritmo de composición desarrollado. Como conclusión con respecto a lo anterior se encuentra que generar una sucesión de notas musicales mediante un autómata celular puede volverse una tarea compleja y si se hace especial énfasis en la tarea de mapeo se puede encontrar toda una rama de investigación y producción musical.

### **5.3 Recomendaciones para futuras investigaciones**

Hay que considerar también que si se escribe un programa computacional que tenga la capacidad de componer una pieza en el estilo de Bach o de Chopin o de cualquier otro músico de gran prestigio, independientemente de ser un logro admirable dada la complejidad del problema y ser también grato poder escuchar otra pieza musical que este a la altura de estos compositores. Es entonces cuando se vuelve necesario analizar a lo que se busca llegar nuevamente. Es decir es muy probable que la composición algorítmica no tenga mayor éxito hasta el momento en que no solo componga música bella, sino que también tendría que componer música diferente a la ya hecha hasta ahora. Después de todo los grandes compositores no solo son reconocidos solo por su habilidad de componer buena música, si no que fueron reconocidos por su aportación dentro de la música en proporcionar una nueva forma de pensamiento armónico.

Lo anterior motiva a pensar que los resultados más contundentes podrían obtenerse no necesariamente de las técnicas de mapeo más complejas, sino de técnicas de mapeo simples pero altamente creativas, que tengan la capacidad de generar ideas y porque no, incluso estilos nuevos.

## Referencias

- [1] Gustavo Diaz Jerez, “Algorithmic music: using mathematical models in music composition” Ph.D Thesis, Musical Arts, The Manhattan school of music, Manhattan, NY, 2000
  
- [2] Adam Alpern, “Techniques for Algorithmic Composition of Music”, Report, Computer Science, Hampshire College, Amherst, Ma, year.
  
- [3] F. Richard Moore, “Composing” in Elements in Computer Music, 1st ed. New Jersey, Prentice Hall, 1990, pp. 398-453
  
- [4] Martin Supper, “A few remarks on algorithmic composition” in Computer Music Journal, Cambridge, MA:MIT Press, 2001, pp. 48-53
  
- [5] Anónimo (2011, Octubre a Diciembre). Matemáticas, música y algoritmia (1ra Edición) [En línea]. Disponible: <http://www.campusvirtual.unex.es>
  
- [6] Juan Antonio Perez Ortiz, “Composicion musical algoritmica” in Musica fractal: El sonido del caos (1ra edición). Stanford,Creative Commons, 2000, ch 6, pp. 29-35.
  
- [7] Rodrigo Gabino Ramírez Moreno, “Generación de música con gramáticas formales”, M.S. tesis, Ciencias de la computación, IPN, México D.F. , Distrito federal, 2011.
  
- [8] Ge Wang, “The chuck audio programming language” Ph.D. Thesis, Computer Science, Princeton University, Princeton, NJ, 2008
  
- [9] Daniel Aschauer. (2008, Enero 29). Algorithmic composition (1st ed) [Online]. Available: <http://wp1120301.msdata.at>
  
- [10] Anonimo. (2011, Octubre). Autómata celular de Divulgación del Saber (1st)[online].available: <http://campusvirtual.unex.es/cala/epistemowikia/index.php>

- [11] Farid Fleifel Tapia. (2010). Automatas Celulares (1st) [Online]. Available: [http://www.redcientifica.com/gaia/ac/auto\\_c.htm#in](http://www.redcientifica.com/gaia/ac/auto_c.htm#in)
- [12] Paul D. Reiners. (2004, May 18). Cellular automata and music [Online]. Available: <http://www.ibm.com/developerworks/java/library/j-camusic/>
- [13] Curtis Roads, “Algorithmic Composition Systems”, in The Computer Music Tutorial, 1th edition, Cambridge, MA:MIT Press, 1996, pp. 819
- [14] Mark Fell and Joe Gilmore. (2012). Composing with process: Perspectives on generative and systems music [Online]. Available: <http://rwm.macba.cat>
- [15] Jacob Peck, A study of cellular automata music: systems of composition and examination of several original works”, M.S. thesis, Dept. Computer science, State University of New York College at Oswego, New York, NY, 2010.
- [16] Gerhard Nierhaus, “Historical development of alghoritmik procedures” in Algorithmic composition, 1st ed. Morlenbach, Germany: SpringerWien, 2009, ch. 2, pp. 7–54.
- [17] John Fauvel, Raymond Flood and Robin Wilson “The science of musical sound” in Music and Mathematics, 1st ed. Oxford, Great Britain: Oxford university press, 2003, ch 2., pp. 47-61.
- [18] Hanna Järveläinen, “Algorithmic musical composition”, Telecommunications software and multimedia, Helsinki University of technology, Helsinki, Finland, 2000.
- [19] Dave Philips. (2010, June 01). Algorithmic music composition with linux (1st ed) [Online]. Available: <http://www.linuxjournal.com>

- [20] Eduardo Reck Miranda, "A cellular automata musical engine" in Composing with computers, 1st ed., Burlington, MA: Focal press, 2002, ch 3., pp. 127-129.
- [21] Chong Yu. (1996, May 28). Computer generated music composition 1st ed [Online]. Available:<http://home.comcast.net/~chtongyu/Thesis.html>
- [22] Phil Ball (2011, March 18). Music is all in the mind (1st) [Online]. Available:<http://www.scientificamerican.com/article.cfm?id=music-is-all-in-the-mind>
- [23] Alan Cane (2005, April 22), Mental ways to make music (1st), [Online]. Available: <http://cmr.soc.plymouth.ac.uk>
- [24] Tony Myatt (2005), Digital Music Research (1st), [Online]. Available: <http://music.york.ac.uk>
- [25] Dirk-Jan Povel, "Melody generator: A Device for algorithmic music construction" Centre for cognition Donders Institute of brain, Nijmegen, The Netherlands, Journal of Software Engineering and Applications, 2010
- [26] Dragan Matic, "A genetic algorithm for composing music", University of Banjaluka, Bosnia and Herzegovina, Yugoslav Journal of Operations Research, 2010
- [27] Bruce L Jacob (1996), Algorithmic composition as a model of creativity (1st), [Online]. Available: <http://www.ee.umd.edu>
- [28] John A Maurer (1999, March), A brief history of Algorithmic Composition (1st), [Online]. Available: <https://ccrma.stanford.edu/~blackrse/algorithm.html>
- [29] George Papadopoulos and Geraint Wiggins, "AI Methods for algorithmic composition: A survey, a critical view and future prospects", University of Edinburg, Scotland, School of artificial intelligence, 2010

- [30] Silas S Brown (1997), Algorithmic composition and reductionist analysis: Can a machine compose? (1st), [Online]. Available: <http://people.pwf.cam.ac.uk/ssb22/clara/article.htm>
- [31] Christopher Ariza, ““An open design for computer aided algorithmic music composition: AthenaCL”, Ph.D. dissertation, Departament of music, New York University, NY, New York,2005.
- [32] Matías Romero Costas, Emiliano Causa y Tarcisio Lucas Pirotta, (2009, June ). Automatas celulares interactivos (1sr ed.) [Online]. Available: <http://www.ceiarteuntref.edu.ar>
- [33] Christopher Ariza, “Navigating the landscape of computer aided algorithmic composition systems: A definition, seven descriptors, and a lexicon of systems and research”, School of arts and sciences, NY, New York, 2005
- [34] Jany Lozano (2012), Investigacion Exploratoria (1st), [Online]. Available: <http://janeth-investigacioniv.blogspot.mx/2008/11/investigacion-exploratoria.html>
- [35] J. Rodriguez Alvira. (2010, September 10). *Lectura musical* (1<sup>st</sup> ed) [Online]. Available: <http://www.teoria.com/aprendizaje/lectura/06-valores2.php>
- [36] Jesus R. Peinado, Jesus Manuel Garcia. (2008, April). *Automatas celulares* (1<sup>st</sup> ed) [Online]. Available: <http://www.enelnombredetux.com/project.php?project=autcel>
- [37] Andrew R. Brown, “Representing music,” in *Making music with Java, 1<sup>st</sup> ed.* South Bank, Australia: Queensland Conservatorium, 2009, Preface, pp. 12-14.

# Apéndices

Universidad Autónoma de Ciudad Juárez  
Instituto de Ingeniería y Tecnología



## Propuesta del tema para la materia de Seminario de Titulación I

Nombre: Ángel Rubén Hernández Quezada	Matrícula: 93476
Programa Académico: Ingeniería en sistemas computacionales	
Departamento: Eléctrica y computación	

**Título:**  
Generación de fragmentos musicales por medio de un algoritmo matemático

**Contextualización:**  
Es conocido que casi todo lo existente se puede representar por medio de modelos matemáticos, y la excepción no lo es la música. Incluso algunas piezas que se ha compuesto tienen una base algorítmica. Esto lleva a la implementación de técnicas que permitan desarrollar fragmentos musicales por medio de métodos algorítmicos entre los que encontramos una variedad de modelos distintos. Los cuales se pueden clasificar según la estructura y el procesamiento de los datos musicales en:

- Modelos matemáticos
- Modelos caóticos
- Modelos evolutivos
- Gramáticas
- Sistemas basados en conocimiento
- Sistemas de aprendizaje
- Sistemas híbridos

**Definición del Problema:**  
Evaluar si es factible utilizar un algoritmo matemático, orientado hacia la generación de fragmentos musicales

**Objetivo General:**  
Desarrollar un prototipo basado en un algoritmo matemático que genere una sucesión de notas musicales

**Preguntas de Investigación:**  
¿Qué limitante encontramos en un algoritmo matemático?  
¿En dónde radica la diferencia para considerar un algoritmo más apropiado que otro?  
¿Qué ventajas tuvo el algoritmo seleccionado?  
Dependiendo de los resultados, ¿qué uso podría darse al prototipo generado?



**Justificación:**

Es una gran ventaja contar con una herramienta automatizada para la generación de música, debido a que se podría apoyar una parte de un trabajo musical en una secuencia generada por un modelo matemático. Es por esto que existe una inquietud tecnológica por explorar la naturaleza de la música e intentar simularla lo más acertadamente.

**Solución Propuesta:**

El producto final consistirá en una aplicación que permita al usuario iniciar la simulación musical de la manera más sencilla posible, la manera de trabajar consistirá básicamente en la selección de un instrumento y la función de empezar simulación con parámetros como el tiempo y la nota inicial.

**Metodología Propuesta:**


- 1) Identificar las variantes existentes para la composición algorítmica
- 2) Desarrollar el prototipo propuesto
- 3) Pruebas de la aplicación
- 4) Evaluar los resultados obtenidos

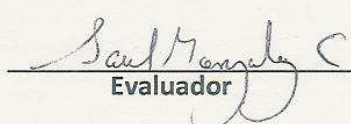
**Observaciones del evaluador:**

Aprobado       Aprobado con condición       Rechazado

**Fecha de terminación del proyecto:**

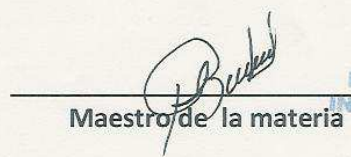
**Nombre del asesor responsable:**

  
\_\_\_\_\_  
Alumno

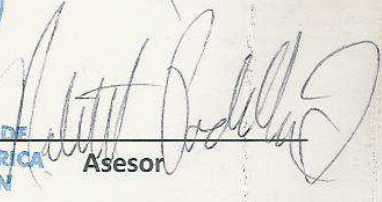
  
\_\_\_\_\_  
Evaluador



  
\_\_\_\_\_  
Evaluador

  
\_\_\_\_\_  
Maestro de la materia

DEPARTAMENTO DE  
INGENIERÍA ELÉCTRICA  
Y COMPUTACIÓN

  
\_\_\_\_\_  
Asesor