

CAPITULO 4

Diseño e Implementación

En el actual capítulo se explicará y dará a conocer el análisis, diseño e implementación que se llevó a cabo para la realización de este proyecto. Cada uno de estos puntos es de suma importancia, debido a que es parte del proyecto, pues ahorra tiempo y apoya a la organización del mismo.

4.1 Análisis

4.1.1 Sistema a Desarrollar

El sistema a desarrollar tiene como objetivo principal la generación y ejecución de nuevas obras musicales. Esta generación parte de los patrones dados por el compositor musical Wolfgang Amadeus Mozart, quien en su obra “Juego de Dados Musical” deja plasmado, esta obra es explicada en el capítulo 3 de este documento.

El propósito principal de este proyecto consiste en construir un prototipo que permita la automatización de generación y ejecución de composiciones musicales, utilizando las herramientas necesarias para su ejecución. Por otro lado, se pretende que el usuario pueda ingresar los compases a utilizar para la generación de estas obras musicales. En adición a lo anterior, se busca que este prototipo funcione para la creación de compases y combinaciones de autoría propia, ya que las herramientas se extenderán para dar oportunidad a esta opción.

A continuación se presenta un modelado, el cual, para el análisis de este proyecto es importante, pues muestra el flujo y transformación de la información que se busca para el proyecto en sí.

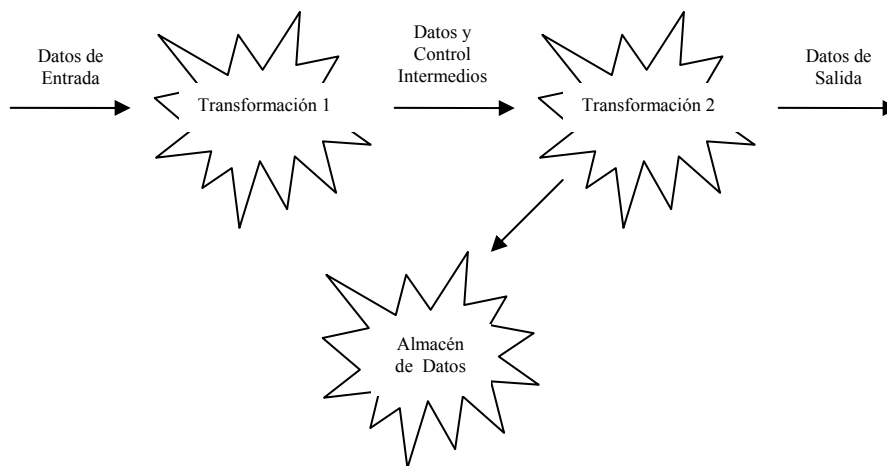


Figura 4.1 Diagrama de Flujo y Transformación de Información

En la figura 4.1 se muestra el flujo de información que se pretende implementar para este sistema, donde:

- Datos de Entrada: estos pueden ser archivos o instrucciones ingresadas por el usuario.
- Transformación 1: los datos de entrada por usuario son tratados para una ejecución intermedia.
- Datos y Control Intermedios: los datos tratados en la transformación 1 son almacenados temporalmente en estructuras de conveniencia.
- Transformación 2: se genera la lista de compases a ejecutar.
- Almacén de Datos: Si los datos fueron ingresados por usuario y/o modificados de algún archivo, estos estarán disponibles para ser salvados con las nuevas modificaciones o bien, como nuevos archivos.
- Datos de salida: los datos están listos para ser ejecutados en orden de generación.

4.1.2 Lenguaje de Programación

El lenguaje de programación a utilizar para el desarrollo de este proyecto es Object Pascal. La utilización de este lenguaje se debe a que el sistema debe ser ejecutado en tiempo real, sin retrasos, característica que Java, el lenguaje de actualidad, no proporciona, pues cuenta con un recolector de basura (*Garbage Collection*) el cuál va liberando recursos al sistema cada vez que un objeto ya no va a ser usado; lo que provoca un retraso (*delay*) pequeño, evento que para este proyecto no puede ser utilizado, es por tanto que Java no fue una opción para la implementación de este proyecto. Básicamente la ejecución en tiempo real limita a que la ejecución tome lugar en un periodo determinado de tiempo, sin retrasos ni adelantos, pues ésto provocaría una falta de ritmo en la pieza musical.

El uso de Delphi se hace a partir de que es un ambiente de programación gráfico, herramienta de apoyo para el desarrollo de interfaces amigables y de relativo fácil uso.

4.2 Diseño

4.2.1 Diseño del Sistema

En esta sección se presentarán el diseño del sistema que será implementado para la generación y ejecución de composiciones musicales. Se incluyen los diagramas correspondientes a cada una de las partes que conforman el diseño de un sistema.

A continuación se muestra un diagrama de flujo de información del sistema para la construcción final. Cabe mencionar que este diagrama de flujo de información es muy general, ya que sólo se quiere asentar la idea de seguimiento para éste.

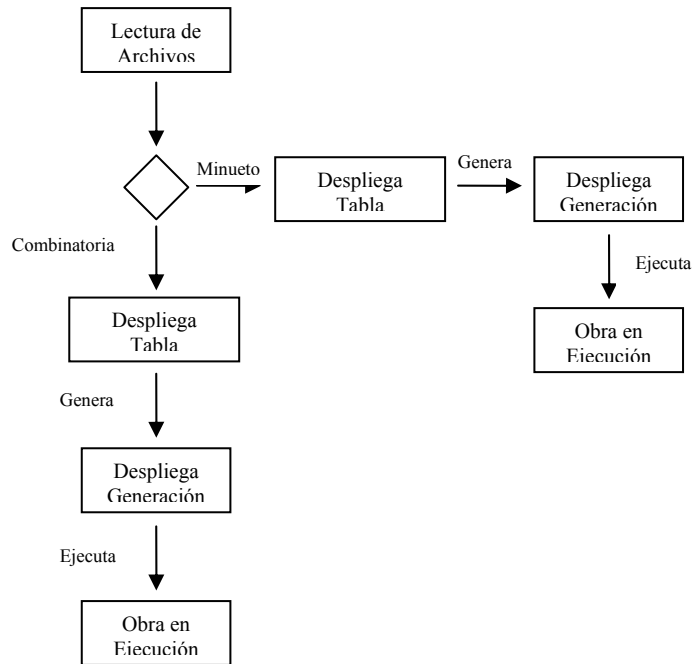


Figura 4.2 Diagrama de Flujo de Información

Por otra parte se pretende construir un prototipo, el cuál es más elaborado, debido que a partir de éste los compases a utilizar podrán ser almacenados. A continuación se muestra un diagrama de flujo de datos para este prototipo, el cuál muestra la estructura a seguir por el mismo.

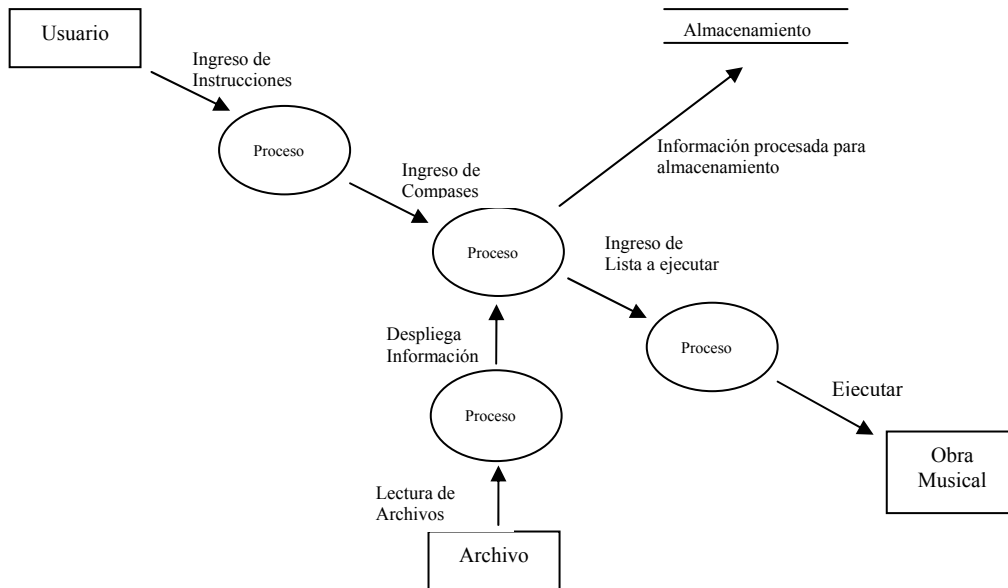


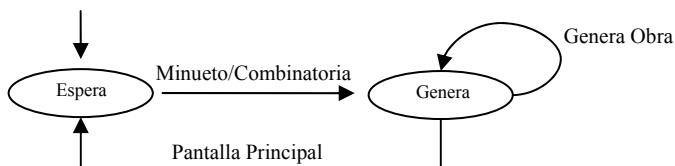
Figura 4.3 Diagrama de Flujo para Prototipo

4.2.1.1 Diagramas de Transición de Estados

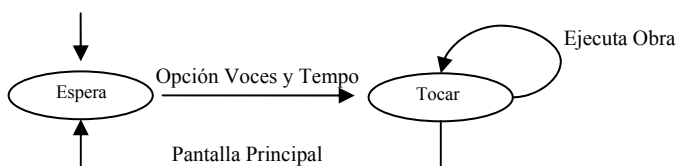
Como otro punto de seguimiento en la parte de diseño se encuentran los diagramas de transición, los cuales ayudarán a entender la forma en la cual el sistema se comporta a partir del manejo que le dará el usuario; de tal forma que se conozca la ruta a seguir en el sistema por parte del mismo. Se mostrarán los estados con más importancia, ya que existen algunos estados que está por demás mostrar, pues son triviales, con ello me refiero a los estados de guardar archivos, abrir archivos y estados que está por demás explicar.

Los estados en el que se encuentra el sistema serán marcados por óvalos y las transiciones por flechas.

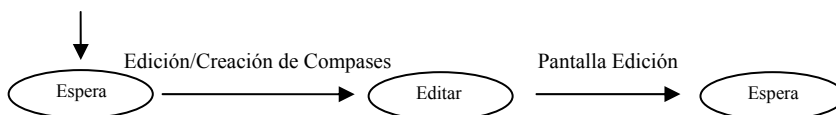
1. Genera minueto y/o combinatoria



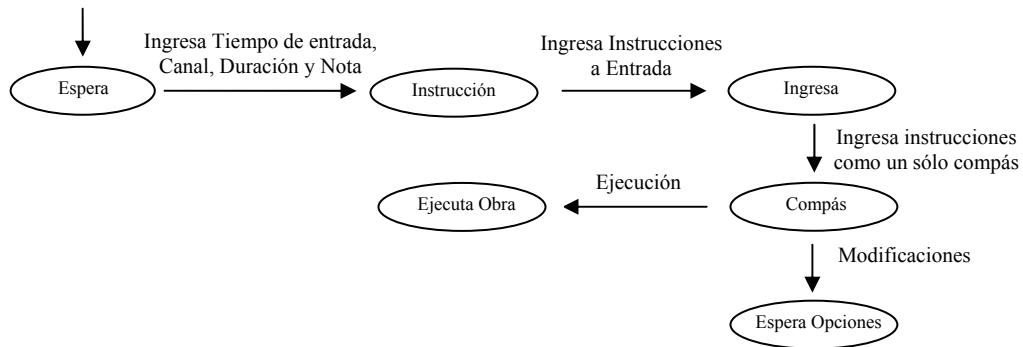
2. Ejecuta obra



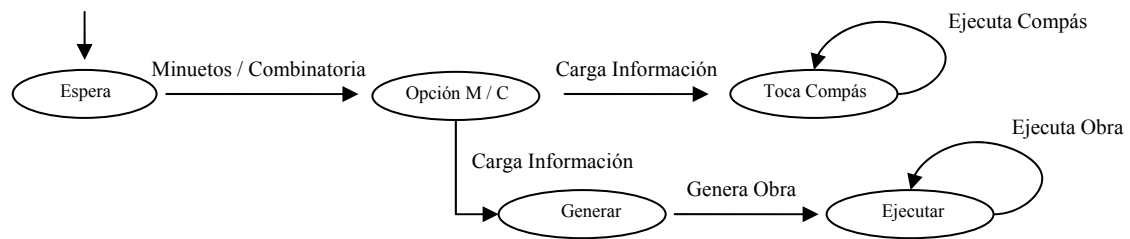
3. Crear nuevos compases – Ir a pantalla de edición



4. Ingreso datos – Pantalla de edición



5. Generación y ejecución – Pantalla de edición



4.2.1.2 Diagrama de Caso de Uso

La finalidad de la construcción del caso de uso es mostrar la interacción del usuario del sistema con las actividades de éste y las que corresponden a cada actor (tipo de usuario) y las afectaciones directas o indirectas que se tienen al mismo. En el caso de este proyecto solo tenemos un tipo de usuario (general) ya que no contamos con administrador del proyecto, y tampoco se hace diferencia entre otro tipo de usuario. A continuación se muestra el caso de uso para el sistema a construir.

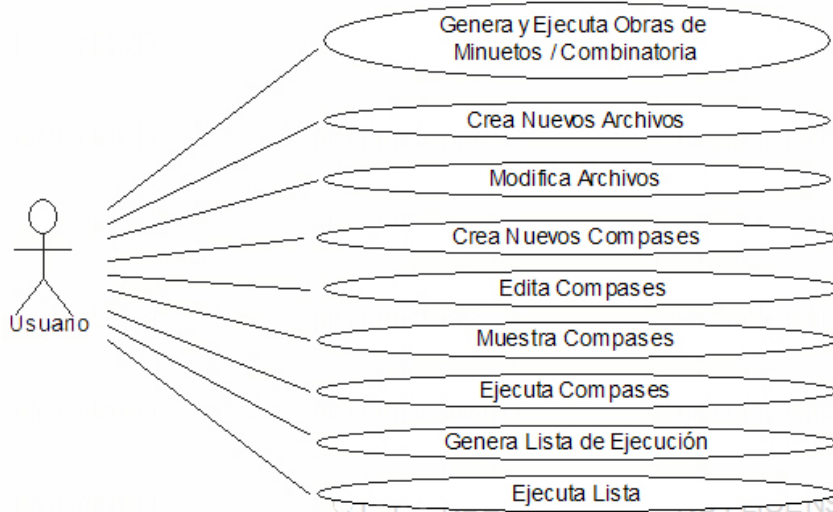


Figura 4.4 Diagrama de Uso

4.2.2 Diseño de Interfaz

El diseño de la interfaz de usuario es importante, ya que es el enlace de comunicación entre el hombre y la máquina. Este diseño se desarrolla a partir de la identificación de objetivos y acciones de la interfaz, y se crea un formato de pantalla base para el prototipo.

4.2.2.1 Pantalla Principal

El objetivo de esta pantalla es presentar la generación aleatoria de compases a ejecutar por parte del sistema y que estos puedan ser reproducidos por el mismo, mostrando al usuario cuales son los números de compases y el orden en el que se reproducen, además de presentar la ejecución en un teclado.

Es por lo anterior, que se pretende, que la pantalla cuente con un teclado que interactúe según las notas a activar de la obra en ejecución. Por otra parte se desplegarán las tablas correspondientes al tipo de obra a generar (minuetos o combinatoria). Se usará una

barra para marcar el Tempo al que se quiere ejecutar la obra y finalmente se contará con opción a ejecutar la obra en el instrumento que se desee. Habrá un menú, el cual permitirá guardar la combinación de minuetos y/o combinatoria para poder ser ejecutas mas tarde. En adición se tendrá una liga a la pantalla de edición que más adelante será presentada.

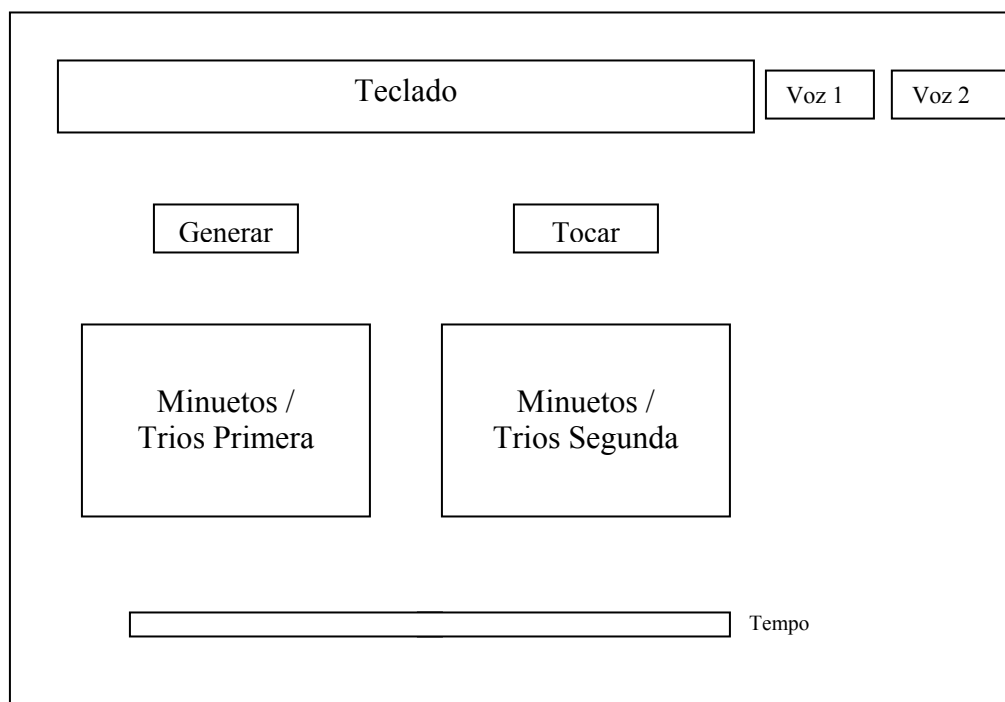


Figura 4.5 Prototipo de Pantalla Principal

4.2.2.2 Pantalla de Edición

Uno de los principales objetivos de esta pantalla es contar con las herramientas necesarias para el ingreso de nuevos compases, la modificación de estos, el almacenamiento de los mismos en archivo, lectura desde archivo y la interpretación de los datos procesados. Por otra parte la ejecución de compases y una lista generada de autoría propia.

El diseño propuesto en primera instancia se muestra a continuación.

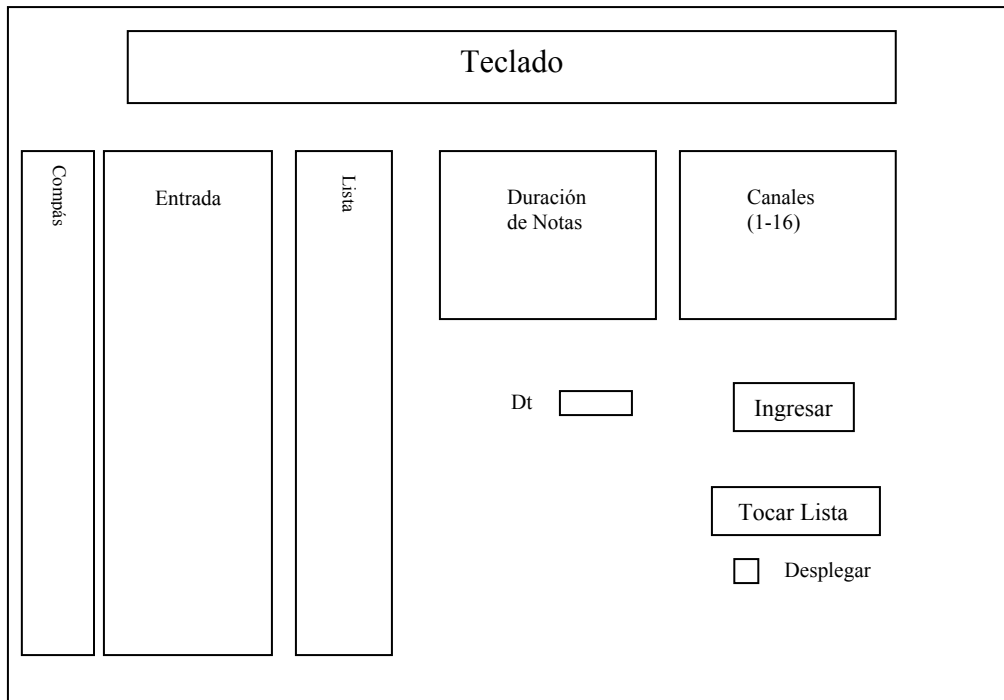


Figura 4.6 Prototipo de Pantalla Edición

4.3 Implementación

En este apartado se explicará como el sistema es dividido para el manejo y organización óptimo del mismo, qué estructuras de datos se utilizaron, cómo se llevó a cabo la manipulación de datos, cómo se hizo el manejo de tiempo real y finalmente se explicará cuales fueron los parámetros a partir de los cuales se implementó las interfaces de usuario. La implementación se hizo a partir del diseño del sistema, este fue de gran apoyo para la implementación; sin embargo, lo anterior no quiere decir que la implementación es exactamente como fue planeado el diseño del sistema, ya que en ésta se mejoraron algunas funciones y fueron aumentadas opciones a algunas interfaces.

4.3.1 Infraestructura

El sistema fue diseñado e implementado de forma tal que permita la generación y ejecución de obras musicales, en adición, que permita también ejecutar partes de estas obras; y por otro lado, permitir un manejo y creación de obras de autoría propia, apoyados por un editor que permite esta opción.

Las instrucciones que forman los compases para la generación de minuetos y tríos se encuentran almacenados en archivos, los cuales fueron creados con anterioridad a partir del editor desarrollado como parte de este software.

El funcionamiento del software es básicamente el siguiente: la información contenida en los archivos de minuetos y tríos es cargada a la forma y desplegadas las tablas a utilizar para la generación de la nueva obra musical. Para la parte de la edición de compases se cuenta con opciones que permiten crear nuevos compases y con ello nuevos archivos musicales, se permite borrar, editar, reproducir e ingresar compases. Para las instrucciones que componen un compás se permite ingresar nuevas y/o borrar existentes. Cada instrucción esta constituida por cuatro componentes los cuales son indispensables para la ejecución de la obra. Sumado a lo anterior, se tiene opción a generar obra de minuetos y combinatoria desde el editor para ver la interacción de la información por compás, cada uno de los que componen la obra completa. En el siguiente apartado se explica, de forma más detallada, las estructuras más importantes utilizadas para la implementación de este proyecto y sus componentes.

4.3.1.1 Estructuras

Una de las principales estructuras de almacenamiento temporal es la utilizada para la retención de las instrucciones principales que componen el archivo. Esta estructura es de

tipo record, el cual representa un conjunto de elementos heterogéneos, donde cada elemento es un campo y esta estructura puede ser visualizada como una matriz. Éstas son las instrucciones que son usadas para la manipulación de datos MIDI, la cual será explicada en el apartado 4.3.2 de este capítulo.

Otra estructura de apoyo a la anterior y al mismo tiempo de apoyo para la ejecución de la obra es un arreglo el cuál contiene (por cada índice) un número de compás y su numero de instrucción de inicio correspondiente.

Para la ejecución en tiempo real de las obras musicales son utilizadas dos estructuras de apoyo, la principal es una lista ligada con varios campos, la cuál contiene información de duración, nota y canal; la secuencia de instrucciones va implícita en la construcción de la tabla. La manipulación y manejo de ésta será especificado en el apartado 4.3.3. La segunda estructura es una matriz que contiene información de secuencia de cada uno de los compases a ejecutar, y un apuntador a su inicio y final de instrucción por compás. Lo anterior con el objetivo de tener un recorrido lineal para la activación de las notas musicales. La construcción de esta estructura se hace a partir de la lista que se genera por medio de lanzamientos de dados, o bien, por asignación de ésta por parte del usuario.

En términos generales la estructura tipo record es la que contiene la secuencia de instrucciones a ejecutar, la lista ligada es la que lleva el control de ejecución y las demás estructuras son de almacenamiento temporal para obtener la información que se necesita y llevar a cabo esta función.

4.3.2 Manipulación de datos MIDI

Este apartado pretende, por objetivo, explicar y denotar la diferencia entre la forma de datos en representación musical y la representación de datos MIDI, cómo éstos deben ser manejados y cuál su representación computacional.

Para llegar al objetivo deseado, ejecutar una obra musical, es necesario contar con información MIDI, representación computacional, ésta con obtención a partir de una representación netamente musical, lo que se quiere decir con “netamente musical” es que se hace una traducción a partir de partituras, es decir la definición de la música en representación gráfica de notas rítmicas y melódicas.

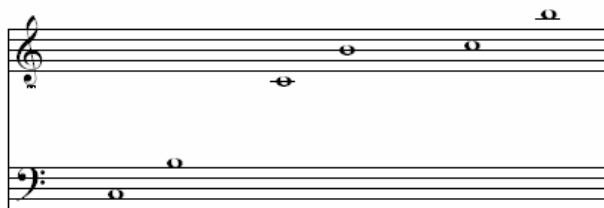
A continuación se presenta un ejemplo simple asíncrono en forma netamente musical (representativa musical).

The image displays two musical staves. The top staff, labeled '1: Track' and 'Primer Pentagrama', is in treble clef and contains a melodic line with notes on the 2nd, 4th, 5th, 6th, 7th, and 8th lines. The bottom staff, labeled '2: Track' and 'Segundo Pentagrama', is in bass clef and contains a bass line with notes on the 1st, 2nd, 3rd, 4th, 5th, and 6th lines. A horizontal timeline at the bottom is numbered from 1 to 12, with vertical lines indicating the time positions of the notes.

Figura 4.7 Representación musical

Esta es una representación gráfica, es decir es como tal, parte de una partitura en la cual solo se está representando la forma rítmica y melódica de un compás, esta será utilizada para hacer notar las diferencias que existen entre la representación musical y la representación de datos MIDI (representación computacional) la cual se generará para poder manejar la información y procesarla. Para ello, antes se necesitan definir algunos parámetros, entre los cuales se encuentra la nota a la que se hará referencia. Las notas que

se encuentran entre Do en clave de Fa y Si en la misma clave se llamarán por su nombre mas un identificador el cual corresponde al número 3. Un ejemplo de lo anterior se presenta a continuación:



Do₃ – Si₃ Do₄ – Si₄ Do₅ – Si₅

Figura 4.8 Numeración de notas por octava

Una breve explicación de la conversión de tipos de información se describe en los siguientes párrafos (ver Figura 4.7):

- 1) Se ejecuta al mismo tiempo, en forma de acorde un sol₃ y un sol₄. El sol₃ tardará la mitad del tiempo de todo el compás, mientras se tocan las demás notas. Por otro lado el sol₄ tardará la mitad del tiempo de ejecución del sol₃.
- 2) Antes de terminar el sol₄, una mitad exactamente antes, se activará la nota mi₄, nótese en la posición (2).
- 3) La nota mi₄ y sol₄ son desactivadas, es entonces cuando se activan la nota do₄ y la₄ como acorde en la posición (3).
- 4) La nota do₄ se desactiva y activa el mi₄ sin afectar a las notas activadas en el primer pentagrama.
- 5) Fracciones de segundo después de (4) la nota la₄ se desactiva, activándose la nota si₄ por muy poco tiempo. Las notas del segundo pentagrama no son afectadas.
- 6) La nota mi₄ en (4), la nota si₄ mencionada en (5) y sol₃, el cual fue activado en (1) son desactivadas. Y se activan sol₃, y la nota do₅ en la posición (6).

- 7) Se activa la nota mi_4 y espera.
- 8) Se desactiva la nota do_5 , se activa la nota re_5 y espera. No se afecta al otro pentagrama.
- 9) Se desactivan las notas mi_4 y re_5 , las cuales fueron activadas en (7) y (8) respectivamente, para activar en forma de acorde las notas do_4 y mi_5 las cuales entran en tiempo de espera.
- 10) Se desactiva la nota do_4 activada en (9) y activo la nota mi_4 .
- 11) Desactivo mi_5 activada anteriormente en (9) y activo la nota fa_5 sin modificar las demás notas del segundo pentagrama.
- 12) Desactivo la nota sol_3 activada en la posición (6), la nota mi_4 activada en (10) y la nota fa_5 activada en (11). Es decir realmente en el punto 12 ya no hay sonido alguno.

Existen sistemas síncronos y asíncronos, los cuales difieren en el ritmo de un compás, los síncronos son relativamente fáciles de manejar, sin embargo existen problemas en el manejo de sistemas asíncronos. Como se puede observar todas las notas terminan y empiezan a “destiempo” es decir no empiezan y terminan iguales o son de valores equivalentes de tal forma que existan momentos en los cuales se activen o desactiven siempre al mismo tiempo. Más adelante se verá la representación en datos MIDI de este compás en específico.

Lo más accesible para el manejo de datos MIDI es tomar solo el momento de inicio y su duración; generando a partir de esto los eventos de *note on* (prendido) y *note off* (apagado) apropiados de cada nota e intercalarlos para que la sincronización y/o asincronización sean correctas.

Lo que es necesario, como datos, para llevar a cabo el proceso es la siguiente información:

- información MIDI a ejecutar
- ejecución de tiempos a esperar

La computadora, también necesita de eventos para tocar música como tal. Estos eventos son llamados eventos MIDI, los cuales constan de las siguientes instrucciones:

- prender una nota (*note on*),
- apagar una nota (*note off*), y
- la duración de cada nota, la cual está implícita en el tiempo que tarda la nota en prender y apagar.

Al tener la información descrita anteriormente su procesamiento no es complicado. La programación MIDI es relativamente simple cuando se cuenta con una descripción dividida en bloques a ejecutar y tiempos de espera.

Para ello ocupamos una tabla de eventos MIDI, la cual es difícil de conceptualizar por el usuario o músico. Lo que llevó a desarrollar una interfaz amigable al usuario para la entrada de datos.

De esta forma el sistema es quien se encarga de traducir esta información a información MIDI, la cual es representada a continuación.

Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
\$91	55	\$90	69	\$80	71	\$81	64	\$80	76
\$90	66	\$F1	4	\$81	64	\$80	72	\$90	77
\$F1	4	\$81	60	\$81	55	\$91	60	\$F1	2
\$91	64	\$91	64	\$91	55	\$90	76	\$80	77
\$F1	4	\$F1	2	\$90	72	\$F1	4	\$81	64
\$81	64	\$80	69	\$F1	2	\$81	60	\$81	55

\$80	66	\$90	71	\$91	64	\$91	64	
\$91	60	\$F1	2	\$F1	4	\$F1	2	
Col 1		Col 2		Col 3		Col 4		Col 5

Tabla 4.1 Representación computacional (información MIDI).

Conceptualización de la tabla:

La tabla contiene por cada registro un comando y un dato. Los datos son los números que aparecen en notación decimal, los cuales son las traducciones en números de las notas musicales, en sus octavas correspondientes, respectivos a la tabla de información MIDI. Los comandos son representados por numeración hexadecimal, los cuales se describen a continuación:

\$90 *note on* de pentagrama superior

\$91 *note on* de pentagrama inferior

\$80 *note off* de pentagrama superior

\$81 *note off* de pentagrama inferior

\$F1 es un delay

Esta tabla representa los datos MIDI que serán enviados para obtener la melodía, es decir, la ejecución de la obra musical.

Estos datos estarán concentrados en la estructura de Instrucción, la cual proporciona información del tiempo en donde una nota se activa, la nota a activar, el canal por el cual se manda y su duración. Sin embargo la manipulación de la duración, los *note on* y *note off* son manejados en la estructura llamada Tabla, la cual manipula más directamente los datos en notación MIDI.

4.3.3 Programación en Tiempo Real

Esta sección se dedica a la explicación de la ejecución en tiempo real que se hace de las obras musicales generadas.

Teóricamente se expuso, en el capítulo 2, qué son los sistemas de tiempo real, y para recordar un poco son los sistemas que ejecutan procedimientos sin interrupción de tiempo, donde las acciones llevan un orden y un tiempo de ejecución. Para el desarrollo de este proyecto si la ejecución cae en tiempo de retraso provocará una pérdida en tiempo y si por el contrario, las acciones se ejecutan en tiempo adelantado se tendrán consecuencias a las acciones siguientes, ejecutándose las instrucciones antes de tiempo y provocando que notas anteriores no sean escuchadas por el tiempo que deberían ser apreciadas.

Sin embargo existen rangos de error manejados para este tipo de sistemas, cada uno define su rango de error. En el caso de este proyecto, en específico, no se puede manejar un rango de error, puesto que una obra musical, cualquiera que ésta fuese, así como una canción, deben ser ejecutadas de forma continua, sin interrupción y respetando cada una de las secuencias en tiempo y forma.

La implementación de la ejecución en tiempo real se controla por medio de un reloj (*timer*), el cual cada cierto intervalo de tiempo, equivalente al tempo, ejecuta las instrucciones que en él se encuentran.

Para lograr que no existiera interrupción alguna se implementó un método, el cuál básicamente hace lo siguiente:

- a) Trata de apagar las notas que estén activas, y se utiliza la expresión “trata de hacerlo” debido a que si la nota no ha llegado a su tiempo de duración de cero, es decir, que la duración ha terminado, entonces solo disminuye ésta.

- b) Se define si se ha terminado con el compás en ejecución, si es así entonces se toman los nuevos índices de instrucciones a ejecutar siguientes, si no, continúa con la ejecución del compás activo.
- c) Activa las notas necesarias en los intervalos correspondientes.
- d) Se define si se ha terminado con la lista de ejecución o no, para de esta forma proceder a terminar el proceso de reloj (*timer*) o bien seguir ejecutando las iteraciones de éste.

El código que hace posible la ejecución continua de la obra musical se muestra en el apéndice A, apartado uno.

Se debe mencionar, debido a que es algo importante en este proyecto, que al activar y desactivar las notas se utiliza una lista ligada dinámica, ésta es modificada cada vez que el *timer* se ejecuta, pues se calcula el tiempo recorrido por cada nota, se agrega información a la lista cuando una nota es activada y se borra de la lista una instrucción cada vez que ésta llega a su fin.

Otro método de suma importancia es el que trata de apagar una nota, y se quiere dar a entender que “trata” de apagarla, debido a que del total de veces a ejecutarse este método no en todas apagará una nota, pues también calcula el tiempo de cada una de las notas activas para ver si ésta ha llegado al final de su ejecución o no. Básicamente este método es el que controla los *note off* de las notas activas. Éste es mostrado en el apartado número dos del apéndice A.

Otro de los métodos que son ejecutados dentro del *timer* es activar una nota, el cuál toma las instrucciones requeridas, las ingresa a la tabla de ejecución y finalmente manda los datos a MIDI para que sean convertidos en sonido. Este método se puede encontrar desplegado en el apéndice A del apartado tres.

La implementación de estos métodos no fue trivial, debido a que la lógica que se necesita para que se obtenga el resultado buscado era difícil de encontrar, en especial porque en el cambio de un compás a otro no debía existir retraso alguno, pues es una secuencia de sonidos con un ritmo definido el cuál debía ser respetado. Finalmente, a partir de la implementación de este método, la ejecución de la obra fue la esperada. La activación de las notas en tiempo definido de entrada, con las duraciones especificadas por el usuario, la nota y el canal son respetados tal cual el usuario hace el ingreso; y para el caso de combinatoria también lo es.

Una de las partes más interesantes de la implementación para este proyecto son los comandos de MIDI. Al crear la forma, los canales de MIDI se abren, para de esta forma estar listo el sistema para recibir información y dar la salida como sonido.

```
if MidiOutOpen(@h, DevID, 0, 0, CALLBACK_NULL) = NoError then
  if MidiOutPrepareHeader(h, @head, SizeOf(MidiHdr)) <> NoError then
    MessageDlg('Midi Failed!', mtInformation, [mbOK], 0);
```

Por otra parte los mensajes que se mandan a MIDI como información para obtener una representación de sonido es el siguiente:

```
MidiOutShortMsg(h, rgb($90 + canal, nota, 63));
```

Esta función manda un mensaje corto a MIDI para especificar la salida del dispositivo, donde:

h : es la variable de salida al dispositivo

rgb: es un procedimiento que lleva datos como el canal, la nota y el volumen (entre otros) para tener como resultado de salida el sonido correspondiente a esa información

canal: número de canal ingresado por usuario

nota: nota a activar ingresada por el usuario

4.3.4 Interfaz de Usuario

En el actual proyecto se desarrollaron solo dos interfaces de usuario. La principal, donde se muestra el resultado del objetivo específico de este proyecto; el cual era desarrollar un sistema de automatización para la generación y ejecución de obras musicales correspondientes a la obra Juego de Dados Musicales escrita por el compositor Wolfgang Amadeus Mozart. La segunda de ellas es una herramienta de apoyo a la primera, pues a partir de ésta los compases que son utilizados para la generación de Minuetos y Combinatoria fueron ingresados. Además de proporcionar una forma libre al usuario de crear sus propios compases y crear composiciones de autoría propia.

La interacción, de la primera interfaz nombrada en el párrafo anterior, con el usuario es estrictamente de generación de nuevas composiciones, su ejecución, el despliegue de la combinación que se generó, el tempo en el que se quiere ejecutar y la opción a ejecutar esa nueva obra en diferentes instrumentos. En realidad es una interfaz sencilla y de fácil entendimiento, pues como se mostró en los diagramas de transición de estados no se cuenta con opciones complicadas, solo de estricta comunicación a cinco opciones. De las cuales la ejecución de la obra ha sido explicada en el apartado anterior, y las demás serán explicadas a continuación.

La generación de obras musicales se hizo por medio de generaciones aleatorias de dos números entre 1 y 6, para la obtención de minuetos, y la generación de un solo número entre 1 y 6, para la obtención de tríos. Debido a que existen funciones predefinidas para la generación de números aleatorios, esta fue implementada. Aunque en realidad esta generación no es aleatoria en su totalidad, pues cada vez que el programa ejecuta esa

función los números serán los mismos, debido a que la semilla de la cual se parte siempre es la misma, en la implementación de esta función se modifica la semilla tomando los milisegundos y segundos del reloj de la maquina en donde se lleva a cabo la ejecución. De esta forma la generación de los números aleatorios cumplirá con la generación aleatoria de éstos, siempre serán diferentes.

Con respecto a la ejecución de diferentes voces, es decir, de diferentes instrumentos, es un dato que se le manda a MIDI donde el valor puede estar entre 0 y 127, cada número es una voz disponible. Al ser ejecutada la obra este dato es manipulado junto con la información requerida y ejecutada en el instrumento de preferencia. Cabe destacar que la obra de Juego de Dados Musical fue compuesta para piano, pero como un toque de detalle se implementó esta opción.

El tempo sin embargo es una opción que se necesitaba implementar, ya que las obras no solo se ejecutan en un tempo determinado, si no que para cada una existe el tempo adecuado. Es por ello que el tempo fue definido de la siguiente manera.

- a) El intervalo del *timer* es constante a 125, lo cual se define como que cada 125 milisegundos se estará ejecutando una unidad de tiempo.
- b) 60 bpm (tonos por minuto) es igual a una negra (figura musical).
- c) 7500 es el número calculado a partir de la información en los incisos a y b. Éste tiene que dividirse entre 125 para obtener las 60 pulsaciones por minuto que se necesitan para ejecutar una negra.
- d) El manejo del tempo se hace por medio de una barra, la cual marcando la posición en la que se encuentra ésta es dividida entre 7500, lo que como resultado da el número de pulsaciones por minutos para que ésta pueda ser modificada al tempo que sea requerido por el usuario.

A continuación se muestran los diferentes tempos que pueden ser ejecutados.

Largo = 40 a 52 bmp
Adagio = 48 a 66 bmp
Andante = 60 a 88 bmp
Moderato = 84 a 108 bmp
Allegreto = 104 a 120 bmp
Allegro = 120 a 144 bmp
Vivace = 138 a 168 bmp
Presto = 160 a 200 bmp
Prestissimo = 200 a 208 bmp

Tabla 4.2 Tempos

Explicadas las opciones implementadas en la interfaz principal, se darán a conocer las principales de la pantalla de Edición.

Se buscó una notación, la más conveniente al usuario, en la cual éste sólo ingresa cuatro datos: a) la nota a ejecutar (do, re, mi, fa, sol, la, si, y alteraciones correspondientes); b) su equivalente en ritmo, es decir triple corchea, doble corchea, corchea, negra, blanca, redonda, con opción de puntillo para cada una de las anteriores y en adición se cuenta con al opción de trecillos; c) el canal para ejecución, donde se cuenta con 16 canales diferentes; y, d) el tiempo de espera entre el principio de compás y el *note on* de la nota.

De esta forma se busca que el usuario cuente con una herramienta de fácil entendimiento y uso para el ingreso de los compases, ya que en un principio se hablaba de que el usuario fuera quien tradujera esta información (partituras) a información MIDI, lo

cuál en un momento dado, en caso de que el usuario sea un músico no va a tener noción alguna de lo que necesita hacer, siendo que la mayoría de los usuarios puedan ser éstos últimos, pero por qué no, también gente no perteneciente a este grupo. Sin embargo, y dejando a un lado el tipo de usuario que haga utilización de este software, traducir la información de partituras a información MIDI suena absurda, pues es un trabajo que el usuario no debe realizar.

En forma general, la traducción de partituras a información MIDI no es inconcebible para que un usuario lo haga, por tanto el editor fue la respuesta al problema, contar con un teclado y ciertos parámetros que cualquier usuario pueda hacer entendibles en un tiempo relativamente corto, como lo es el tiempo de espera para la ejecución de las notas (*delta time*), su duración y el canal en el cuál deben de ejecutarse. Cada uno de estos será explicado más detalladamente en párrafos contiguos.

Como se había mencionado al principio de este capítulo, la interfaz de edición es más compleja, debido a que su utilización no es trivial para cualquier persona que no cuente con conocimientos base de música. Sin embargo, se trató de desarrollar una interfaz amigable y de fácil uso para el ingreso de compases, permitir su utilización, modificación y entre otras cosas, una interacción gráfica de lo ejecutado.

Una de las partes interesantes de esta interfaz es el teclado, el cuál tiene la funcionalidad de “botón” de entrada de datos. Cada vez que los campos de opción estén seleccionados éstos serán ingresados como instrucciones al tiempo que se toca la nota deseada. En otras palabras, para ingresar una instrucción es necesario poner el incremento de tiempo en el que ésta se ejecutará, es decir, después de que duración la nota debe ser activada, este incremento puede ser de 0 en caso de ser acorde y de alguna duración en caso de ser secuencia. La duración de la nota puede ser desde triple corchea, hasta una redonda,

y cada una puede ser usada con puntillo, el cuál marca una mitad más del valor normal de la duración de la nota seleccionada. Por último el canal por el cuál las voces son transmitidas, estos se extienden del 1 al 16; por cuestiones de organización el canal 1 se toma para el pentagrama superior y el canal 2 para el pentagrama inferior.

Cada una de las instrucciones se ve reflejada en una de las ventanas (Entrada, ver Figura 4.8), donde las instrucciones son almacenadas temporalmente, sin embargo al ingresar esas instrucciones como un compás, la información es resumida y guardada temporalmente en otra de las ventanas (Compás, ver Figura 4.8), donde sólo refleja el número de compás ingresado.

La opción de Tocar Compás fue implementada de forma tal que sólo toca el compás deseado y nada más, esto se desarrolló con el objetivo de que se pudiera escuchar cuantas veces se desee un trozo de la pieza a elaborar o elaborada.

La opción de Toca Lista es activada en cuanto se tiene una lista de compases a ejecutar, para la implementación de esta opción se validan los datos de entrada y solo los válidos son los ejecutados por el sistema de forma óptima.

En esta misma interfaz se cuenta con la opción de crear las obras de minuetos y combinatoria, pero a diferencia de la interfaz principal en este caso se puede ver qué compases son los que se están ejecutando, en que orden y cada una de las instrucciones que se ejecutan en su momento. Es una forma diferente de apreciar qué es lo que pasa mientras la ejecución está en curso.

Finalmente, en este capítulo se explicó cómo las estructuras son utilizadas y se dieron a conocer algunos de los parámetros para la implementación de este proyecto, los más importantes.