



**UNIVERSIDAD
NACIONAL
DE LOJA**



Área de la Energía las Industrias y los Recursos Naturales No Renovables

**CARRERA DE INGENIERÍA EN
SISTEMAS**

**“Software inteligente para el procesamiento
musical de guitarra melódica aplicando una red
neuronal supervisada”**

*Tesis previa a la obtención del
título de Ingeniero en Sistemas.*

Autores:

- Rodrigo Cornelio – Granda Veintimilla
- Juan Guillermo – Pinzón González

Director:

- Ing. Mario Andrés Palma Jaramillo, Mg. Sc.

Certificación del Director

Ing. Mario Andrés Palma Jaramillo, Mg. Sc.

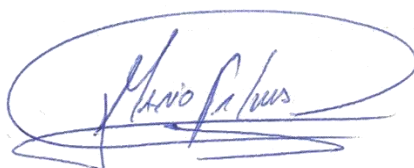
DIRECTOR DEL TRABAJO DE TITULACIÓN

**DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA
UNIVERSIDAD NACIONAL DE LOJA**

CERTIFICA:

Que los Egresados **Juan Guillermo Pinzón González y Rodrigo Cornelio Granda Veintimilla** desarrollaron cabalmente el trabajo de titulación denominado **“SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL DE GUITARRA MELÓDICA APLICANDO UNA RED NEURONAL SUPERVISADA”**, dicho proyecto cumple con los requisitos establecidos en las normas generales tanto en el aspecto de forma como de contenido bajo las sugerencias de mi dirección, supervisión y asesoramiento.

Loja, 29 de Octubre de 2015.



Ing. Mario Andrés Palma Jaramillo, Mg. Sc.

DIRECTOR DE TESIS

Autoría

Nosotros: **JUAN GUILLERMO PINZÓN GONZÁLEZ** y **RODRIGO CORNELIO GRANDA VEINTIMILLA** declaramos ser autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el repositorio institucional – Biblioteca Virtual.

Firma: 

Cédula: 1105148942

Fecha: 14 de diciembre de 2015

Firma: 

Cédula: 1104766348

Fecha: 14 de diciembre de 2015

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Nosotros: **JUAN GUILLERMO PINZÓN GONZÁLEZ y RORIGO CORNELIO GRANDA VEINTIMILLA**, declaramos ser autores de la tesis titulada: **SOFTWARE INTELIGENTE PARA EL PROCESAMIENTO MUSICAL DE GUITARRA MELÓDICA APLICANDO UNA RED NEURONAL SUPERVISADA**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios puedan consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los catorce días del mes de diciembre del dos mil quince.

Firma:



Autor: Juan Guillermo Pinzón González

Cédula: 1105148942

Dirección: Loja, José de San Martín y Salvador Allende.

Correo electrónico: jgpinzong@unl.edu.ec

Teléfono: (07) 2586904

Celular: 0989644512

Firma:



Autor: Rodrigo Cornelio Granda Veintimilla

Cédula: 1104766348

Dirección: Loja, Av. Eugenio Espejo e Integración Barrial.

Correo electrónico: rcgrandav@unl.edu.ec

Teléfono:

Celular: 0980753694

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Mario Andrés Palma Jaramillo, Mg. Sc.

Tribunal de Grado: Ing. Hernán Leonardo Torres Carrión, Mg. Sc.

Ing. Edwin René Guamán Quinche, Mg. Sc.

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Agradecimiento

A la Universidad Nacional de Loja que nos acogió para dar forma a nuestro futuro como profesionales y a la Carrera de Ingeniería en Sistemas, que moldeó nuestras capacidades dentro de sus aulas, nos brindó el conocimiento intelectual y humano de cada uno de los docentes, el mismo que nos sirvió para concluir con éxito nuestro Trabajo de Titulación.

A nuestro director de Tesis Ing. Mario Palma, por brindarnos el tiempo necesario y darnos sus aportes, críticas, comentarios y sugerencias durante el desarrollo de este proyecto.

Finalmente un agradecimiento a todos los amigos, que supieron darnos ideas para llegar a cumplir con todas las fases que contempla este Trabajo de Titulación.

Dedicatoria

Dedico este trabajo de titulación principalmente a Dios, por darme salud bienestar y fuerza para seguir siempre adelante y no desfallecer en el intento, así como también darme la oportunidad de pertenecer a la Carrera de Ingeniería en Sistemas y conocer grandes amigos con los que quedarán las anécdotas de toda esta etapa estudiantil.

A mis padres: Francisco y Alba, quienes me dieron la vida, educación, apoyo y consejos para poder llegar a esta instancia de mis estudios, sin su perseverancia este logro no hubiera sido posible, las palabras quedan cortas para agradecerles, sin embargo gracias por todo !!!

A mi hermano Víctor ya que ha estado pendiente de mi desenvolvimiento en la vida universitaria, y con su apoyo se ha logrado pasar algunas adversidades en toda esta etapa estudiantil.

A mi compañero en este Trabajo de Titulación: Rodrigo, ya que gracias a su apoyo se ha plasmado de mejor manera las ideas que surgieron dentro del proyecto, y esto a su vez nos permitió alcanzar con éxito el objetivo planteado de culminar nuestra carrera universitaria.

Juan Guillermo

Dedico este trabajo de titulación principalmente a Dios, por la vida, la salud y la oportunidad de llegar a este momento tan importante de la vida.

A mis padres: Juliano y Melania, quienes confiaron en mí y me han apoyado desde el inicio del camino hasta ahora, han estado conmigo en las buenas y malas siempre dispuestos a tenderme la mano cuando los he necesitado.

A mi compañero en este Trabajo de Titulación: Juan, promotor principal de la idea en torno a la cual gira el proyecto, gracias a su visión y perseverancia se ha podido culminar con éxito el presente trabajo y por ende nuestra carrera universitaria.

Rodrigo Cornelio

Cesión de Derechos

Nosotros: **Juan Guillermo Pinzón González y Rodrigo Cornelio Granda Veintimilla**, declaramos ser autores intelectuales del presente Trabajo de Titulación y autorizamos a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales No Renovables y por ende a la Carrera de Ingeniería en Sistemas a hacer uso del mismo en lo que estime sea conveniente.

Para constancia firmamos a continuación.



Juan Guillermo Pinzón González

1105148942



Rodrigo Cornelio Granda Veintimilla

1104766348

a. Título

Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada

b. Resumen

En los últimos años, la computación ha apuntado a desarrollar la Inteligencia Artificial (IA) tal como se la enunció en sus inicios, máquinas capaces de pensar como el ser humano. Un amplio campo de aplicación es la música debido a las herramientas, tanto de hardware como de software, que facilitan la captura y análisis de audio.

La finalidad del presente trabajo es ayudar a la composición musical en guitarra melódica por medio de una aplicación que captura el audio de la melodía tocada y luego de realizar el respectivo análisis de frecuencias por medio de una red neuronal supervisada, da al músico una respuesta acerca de la tonalidad de la pieza y la posible estructura que debe seguir para que dicha pieza musical sea agradable al oído, desde el punto de vista de la música occidental.

En el desarrollo del software se aplicó la metodología XP adaptándola a las características propias del proyecto, específicamente el desarrollo de la red neuronal, donde se siguió el flujo de trabajo propuesto por los creadores del Neural Network Toolbox de Matlab.

La aplicación se codificó utilizando la herramienta Matlab, debido a los toolboxes que ofrece tanto para el procesamiento de señales como para las redes neuronales, que facilitan la implementación de funciones de transferencia y algoritmos para el entrenamiento y aprendizaje de la red.

La topología de la red neuronal es la de un Multilayer Perceptron y el dataset para el entrenamiento se lo elaboró con 180 piezas de música clásica.

Summary

In the last years, the computer has pointed to develop Artificial Intelligence (AI) as it was enunciated in its beginnings, machines that can think like humans. A wide field of application is the music, because the hardware and software tools facilitate the capture and audio analysis.

The purpose of this work is to help the melodic guitar musical composition through an application that captures audio from the played melody, and, after making the respective frequency analysis through a supervised neural network, gives the musician an answer about the tone of the piece and the possible structure to follow, so this piece of music is pleasing to the ear, from the point of view of Western music.

In the software development XP methodology was applied, adapting it to the characteristics of the project, specifically the development of the neural network, where the flow of work proposed by the creators of the Neural Network Toolbox of Matlab was followed.

The application was coded using Matlab, because it's offered toolboxes for signal processing and for the creation of neural networks, facilitate the implementation of transfer functions and algorithms for training and learning of the network.

The architecture of the neural network is a Multilayer Perceptron and the dataset for training was developed with 180 pieces of classical music.

Índice de contenidos

Índice general

Certificación del Director	I
Autoría	II
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.	III
a. Título	VII
b. Resumen	VIII
Summary	IX
Índice de contenidos	X
Índice general	X
Índice de Figuras	XIII
Índice de Tablas	XIV
c. Introducción	15
d. Revisión Literaria	17
1. Teoría musical	17
1.1 Sonido	17
1.2 Elementos de la Música	19
1.2.1 Nota Musical	20
1.3 Escalas Musicales	21
1.3.1 Escala Mayor	22
1.3.2 Escala Menor	24
1.3.3 Frecuencia de las Notas	25
1.4 Tono y Semitono	29
2. Inteligencia Artificial	31
2.1 Redes neuronales	31
2.1.1 Fundamentos Biológicos	31

X

2.1.2	Neurona Artificial.....	32
2.1.3	Redes Neuronales Artificiales.....	33
2.1.3.1	Topología de la RNA.....	34
2.1.3.2	Aprendizaje	34
2.1.4	Perceptrón Multicapa (MLP)	36
3.	Música por Ordenador	37
3.1	Procesamiento digital de señales.....	37
3.1.1	Transformada de Fourier	38
3.2	Formato WAV	39
e.	Materiales y Métodos.	41
f.	Resultados	43
Fase 1: Analizar las tecnologías existentes para el desarrollo de la aplicación de tal manera que se satisfagan los requerimientos existentes.....		43
1.	Analizar casos de éxito en el desarrollo de aplicaciones inteligentes para composición musical.....	43
2.	Comparar herramientas para la creación de redes neuronales y elegir la que mejor se adapte al proyecto.	46
3.	Seleccionar las alternativas tecnológicas para la captura de sonido en tiempo real que se adapten de mejor manera a los requerimientos del sistema.....	47
Fase 2: Desarrollar un módulo para la captura y procesamiento de sonido en tiempo real que sirva como entrada de datos a la red neural.		48
1.	Definir una metodología de desarrollo, de acuerdo a los objetivos planteados y al tiempo establecido para la presentación de resultados.....	48
2.	Historias de usuario.....	49
3.	Prototipado del sistema	55
4.	Implementación del módulo en un lenguaje de programación que permita el procesamiento de sonido en tiempo real.	56
4.1.	Grabación de audio con Matlab.....	57
4.2.	Procesamiento Digital de la Señal (DSP)	58
Fase 3. Implementar una red neuronal de acuerdo a un modelo de aprendizaje supervisado para obtener melodías convincentes y realizar las pruebas del software desarrollado.		63
1.	Recolectar datos - Definir el tipo de red neuronal que satisfaga los requerimientos del proyecto a implementar.....	64

2.	Obtención del Dataset - Obtener las entradas de la red neuronal mediante el módulo de captura de sonido.....	65
3.	Crear la red Neuronal.....	70
3.1.	Arquitectura de la Red Neuronal.....	70
3.2.	Funciones de Transferencia (Transfer Functions).....	71
3.3.	Configurar la Red Neuronal	72
3.3.1.	Configurar Entradas y Salidas.....	72
3.3.2.	Inicializar Pesos y Bias	73
4.	Entrenar la Red Neuronal.....	73
5.	Validar la Red Neuronal.....	75
6.	Usar la Red Neuronal - Evaluar los resultados de la red neuronal.....	79
Fase 4.	Realizar las pruebas del software desarrollado para comprobar el correcto funcionamiento del mismo.	82
1.	Pruebas de interfaz	82
2.	Pruebas Unitarias	85
g.	Discusión.....	87
1.	Desarrollo de la propuesta alternativa.....	87
2.	Valoración técnica, económica, ambiental	88
h.	Conclusiones.....	93
i.	Recomendaciones.....	94
j.	Bibliografía.....	96
k.	Anexos.....	98
Anexo 1.	Dataset para el entrenamiento de la red neuronal.....	98
Anexo 2.	Certificación de funcionalidad de la aplicación.	105
Anexo 3:	Certificación Revisión Literaria	115
Anexo 4:	Licencia Creative Commons.....	117

Índice de Figuras

Figura 1. Onda de ruido y onda de sonido musical	19
Figura 2. Escala Representada en el Pentagrama	21
Figura 3. Estructura de la neurona biológica	32
Figura 4. Neurona Artificial	32
Figura 5. Esquema del aprendizaje supervisado en RNA	35
Figura 6. Perceptrón Multicapa	36
Figura 7. Prototipo de Interfaz de Captura de Audio	56
Figura 8. Prototipo Interfaz de Análisis de Audio	56
Figura 9. Multi-Layered Perceptron	70
Figura 10. Diseño de la Red Neuronal	71
Figura 11. Función LOGSIG	71
Figura 12. Función PURELIN	72
Figura 13. Performance de la red	77
Figura 14. Plot de regresión	78
Figura 15. Matriz de confusión	79
Figura 16. Interfaz principal	82
Figura 17. Interfaz de grabación de audio	83
Figura 18. Interfaz de análisis de sonido	84
Figura 19. Ingreso al menú Ayuda	85
Figura 20. Respuesta al testeo unitario	86

Índice de Tablas

TABLA I. NOMENCLATURA DE LAS NOTAS EN LA MÚSICA OCCIDENTAL.....	21
TABLA II. ESCALA CROMÁTICA GENERAL	22
TABLA III. SISTEMA MUSICAL INGLÉS	22
TABLA IV. ESTRUCTURA BÁSICA DE LA ESCALA DIATÓNICA	22
TABLA V. ESCALA C MAYOR	23
TABLA VI. ESCALA D MAYOR	23
TABLA VII. ESCALAS MAYORES	23
TABLA VIII. ESTRUCTURA BÁSICA DE LA ESCALA MENOR	24
TABLA IX. ESCALA A MENOR	24
TABLA X. ESCALA B MENOR	24
TABLA XI. ESCALAS MENORES	25
TABLA XII. ESCALA TEMPERADA A 440 Hz.	26
TABLA XIII. CASOS DE ÉXITO DE APLICACIONES DE IA EN LA MÚSICA.....	44
TABLA XIV. COMPARATIVA DE HERRAMIENTAS PARA CREACIÓN DE RNAS.....	46
TABLA XV. COMPARATIVA DE HERRAMIENTAS PARA CAPTURAR AUDIO	47
TABLA XVI. HISTORIA DE USUARIO 1	50
TABLA XVII. HISTORIA DE USUARIO 2	51
TABLA XVIII. HISTORIA DE USUARIO 3.....	52
TABLA XIX. HISTORIA DE USUARIO 4	53
TABLA XX. HISTORIA DE USUARIO 5	54
TABLA XXI. HISTORIA DE USUARIO 6	55
TABLA XXII. FRECUENCIA EN UN INSTANTE DADO	63
TABLA XXIII. FRECUENCIAS DE ENTRADA PARA LA RNA	65
TABLA XXIV. PERFIL DE CLAVE. ESCALA MAYOR	65
TABLA XXV. ESCALA C MAYOR	66
TABLA XXVI. PERFIL DE CLAVE. ESCALA MENOR.....	66
TABLA XXVII. PERFIL DE CLAVE. ESCALA C MENOR.....	66
TABLA XXVIII. PERFILES DE LA ESCALA MAYOR.....	68
TABLA XXIX. PERFILES DE CLAVE. ESCALA MENOR	69
TABLA XXX. ALGORITMOS DE APRENDIZAJE BACKPROPAGATION.....	74
TABLA XXXI. RESULTADOS OBTENIDOS.....	75
TABLA XXXII. RECURSOS HUMANOS.	89
TABLA XXXIII. RECURSOS MATERIALES	90
TABLA XXXIV. RECURSOS HARDWARE.....	90
TABLA XXXV. RECUSROS SOFTWARE.....	91
TABLA XXXVI. RECURSOS TÉCNICOS Y TECNOLÓGICOS	91
TABLA XXXVII. APROXIMACIÓN DEL COSTO REAL DEL TT.....	92

c. Introducción

El nuevo reto de la ciencia y tecnología es replicar la inteligencia humana en las máquinas, lo que se conoce como inteligencia artificial (IA). Un amplio campo de aplicación es el arte, específicamente la música, ya sea la manera de captarla, entenderla o hasta crearla.

La evolución del hardware y software ha facilitado en gran manera tareas como la captura y digitalización de audio, además se pueden extraer del archivo capturado sus principales características (frecuencia, amplitud, altura), que permiten analizarlo como el fenómeno físico al que representa mas no es posible definir su musicalidad. Para un músico con nivel de conocimiento medio o bajo, esto es un inconveniente porque lo importante para él es conocer la estructura musical de una pieza tocada a través de su tonalidad o escala.

La historia musical ha dejado para nosotros un amplio y variado repertorio de hermosas melodías, por lo que se cuenta con suficiente información para aplicar las técnicas de IA, que permitan al computador responder al usuario si una melodía tiene o no, una correcta estructura musical.

En relación a lo descrito es que se ha realizado este Trabajo de Titulación que tiene como objetivo principal el desarrollo de un software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada, para facilitar el proceso de composición musical en guitarra.

En cuanto a la conformación del documento, está dividido en los siguientes apartados:

El RESUMEN que contiene un extracto del contenido general del Trabajo de Titulación, ÍNDICE que describe la ubicación de los temas tratados, así como también la ubicación de las figuras y tablas presentes en este documento, INTRODUCCIÓN que relata lo relevante que es el tema y su aplicabilidad en la investigación científica, METODOLOGÍA donde se realiza una descripción de los principales materiales y

métodos de investigación empleados, así como también las técnicas aplicadas en el desarrollo del proyecto.

En la REVISIÓN DE LA LITERATURA, se exponen los aspectos teóricos relevantes para el desarrollo de la aplicación. En el orden descrito se revisa primero la teoría musical, centrándose sus elementos básicos y la forma en que oído humano la percibe, luego se tiene aspectos fundamentales de las redes neuronales, como sus tipos, formas de entrenamiento y formas de aprendizaje; y, finalmente se revisan varios casos de éxito de la aplicación de la I.A. en la música.

Los RESULTADOS contemplan el seguimiento de la metodología para cumplir con los objetivos planteados, por esto, como primer punto se identifican los requisitos del sistema, luego se realiza una revisión de las herramientas elegidas para la captura de audio y finalmente se explica el desarrollo del módulo de captura de audio y la red neuronal, en la DISCUSIÓN se explica el uso de los métodos y técnicas utilizadas, las CONCLUSIONES detallan las ideas a las que se llegó con los resultados obtenidos; el proyecto finaliza con las RECOMENDACIONES, BIBLIOGRAFÍA y ANEXOS.

d. Revisión Literaria

1. Teoría musical

En [1] se define la música como “el arte de ordenar los sonidos con el fin de crear una emoción en el oyente”. Aquí se destacan los términos **sonido** y **arte**, el primero haciendo referencia a la estructura de las piezas musicales y el segundo a las sensaciones que motiva en el oyente.

1.1 Sonido

Desde el punto de vista de la física el sonido es cualquier fenómeno que involucre la propagación en forma de ondas elásticas a través de un fluido u otro medio elástico (puede ser el aire) que esté generando el movimiento vibratorio de un cuerpo. En dicho movimiento se distinguen tres características [2].

- **Frecuencia (f):** número de ondas (vibraciones que producen las moléculas) que hay en un segundo, la unidad de medida de frecuencia es el Hertzio Hz.
- **Amplitud (A):** nivel de potencia a la que se han producido las oscilaciones, cuanto mayor sea el sonido será más fuerte, cuanto menos, más débil.
- **Período (T):** tiempo que tarda en finalizar una onda u oscilación, se mide en unidades de tiempo.

Para completar la definición se debe tener en cuenta también el fenómeno psicoacústico¹; entonces, para que exista sonido deben existir tres elementos [3]:

- La fuente, el objeto que vibra.

¹ Psicoacústica: Ciencia que estudia cómo escuchamos los seres humanos, se encarga de los fenómenos físicos que ocurren dentro del oído humano y los fenómenos sociológicos que afectan nuestra manera de escuchar [27]

- La propagación en ondas longitudinales
- El receptor, que puede ser el oído humano.

El receptor puede distinguir entre un sonido y otro por lo siguiente:

- **Altura o Tono:** Resultado de la frecuencia que produce un cuerpo sonoro. De acuerdo con esto se pueden definir los sonidos como "graves" y "agudos". Cuanto mayor sea la frecuencia, más agudo (o alto) será el sonido [4].
- **Duración.**- Corresponde al tiempo que duran las vibraciones que producen un sonido, Está relacionada con el ritmo en la música y viene representada en la onda por los segundos que ésta contenga, o sea el período (T).
- **Intensidad:** Fuerza con la que se produce un sonido; está asociada con el nivel de presión en el medio acústico y suele expresarse en dB referidos a 0.0002 μ bar (20 μ Pa). Sin embargo, el oído no percibe por igual las variaciones de presión a diferentes frecuencias. Así, la intensidad sonora, como la percibe un oyente promedio, es en realidad una intensidad aparente. [3].
- **Timbre:** Permite al receptor identificar las características de la fuente, por ejemplo la voz de una cierta persona o de un determinado instrumento musical. Tiene que ver con la frecuencia fundamental de la fuente, pero también con los armónicos de ésta [4] .

De acuerdo a estas características el oído humano puede definir si un sonido es agradable o no. Los sonidos desagradables son denominados ruido y forman ondas desordenadas, confusas y caóticas, mientras que los sonidos musicales forman ondas regulares, modeladas y ordenadas (Figura 1) [5] .

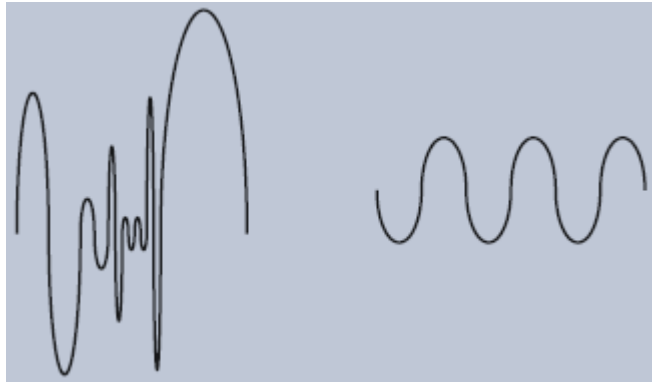


Figura 1. Onda de Ruido y onda de Sonido Musical

La ausencia de sonido perceptible se denomina **silencio**.

1.2 Elementos de la Música

Para que un conjunto de sonidos y silencios pueda ser reconocido como música deben existir los siguientes elementos:

- a) **Melodía:** Es el arreglo significativo y coherente de una serie de notas, en la música tonal se realiza según la tonalidad en la cual se diseña la melodía [4].
- b) **Armonía:** Es la superposición de sonidos que se producen simultáneamente. Es ciencia porque enseña a combinar los sonidos de acuerdo a las reglas inmutables con el fin de construir acordes, y arte porque de la habilidad y el buen gusto de la conducción de las voces armónicas resultará el trabajo realizado más o menos musical [4].
- c) **Ritmo:** Es la distribución de las duraciones sonoras en el tiempo y en el espacio en divisiones regulares de tiempo. El ritmo está relacionado con cualquier movimiento que se repite con regularidad en el tiempo, en la música se lo divide por medio de la combinación de sonidos y silencios de distinta duración [4].

d) Matices: Es la intención, el color o dinámica que se da a la música. Las diferentes gradaciones que se puede dar a un sonido o frase musical. Son las dinámicas que se aplican para enriquecer el hecho musical [6].

Los matices pueden ser de dos clases:

- *Dinámicos*, que tienen que ver con la intensidad de los sonidos; y,
- *Agógicos*, relacionados con las duraciones o el tempo de los sonidos.

Las características anteriores se refieren más que nada al modo en que el oyente siente la música, por lo que no son susceptibles a medidas físicas, Pero cada una de ellas se construye a partir de elementos que han sido creados como una manera para la escritura y lectura de la música.

1.2.1 Nota Musical

La nota es el elemento básico del sonido y la música, considerada el átomo de toda composición musical debido a que cada nota es particular e indivisible, y con la unión de ellas se pueden armar sonidos más complejos.

Las ondas que componen las notas musicales están espaciadas de modo uniforme, es decir son regulares [6]. En la música occidental se utilizan doce sonidos (notas), siete naturales y cinco alterados y cada repetición de los doce sonidos es llamada octava [6].

Las alteraciones son recursos para alterar la altura de las notas [4], las más importantes son:

- Bemol (b): baja la nota medio tono.
- Sostenido (#): sube la nota medio tono.
- Becuadro (♮): anula cualquier alteración y vuelve la nota a su altura original.

Tras varias modificaciones a través de los años ahora existen tres sistemas de notación musical que le da el nombre a las notas, como se ve en la tabla I.

1.3 Escalas Musicales

En la tradición musical occidental se divide equitativamente la octava en doce sonidos o notas. Una escala es un conjunto de notas escogidas de entre estas doce. Cada nota de una escala es llamada grado. Cada grado tiene su nombre, aunque también se acostumbra identificarlos usando números romanos [6] (Figura 2).

TABLA I. NOMENCLATURA DE LAS NOTAS EN LA MÚSICA OCCIDENTAL

Sistema latino	Sistema inglés	Sistema alemán
Do	C	C
Re	D	D
Mi	E	E
Fa	F	F
Sol	G	G
La	A	A
Si	B	H

Las dos características que distinguen una escala de otra son:

- El número de notas que tiene.
- La distancia entre sus grados.

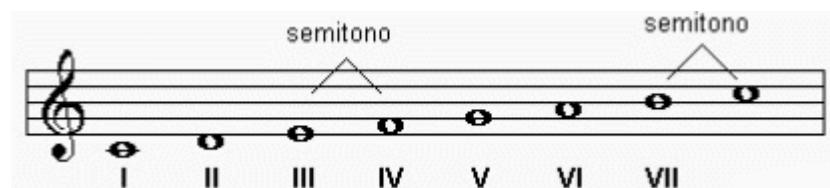


Figura 2. Escala Representada en el Pentagrama

Es posible construir una escala comenzando en cualquier nota usando alteraciones para mantener el orden de tonos y semitonos, por tanto existe gran cantidad de escalas musicales. Por ahora solamente se han tomado en cuenta las escalas más usadas en la música occidental clásica, ya que estas composiciones tienen estructuras teóricamente correctas. Cada una de ellas tiene 12 notas distintas que vienen dadas por la escala cromática general (Tabla II).

TABLA II. ESCALA CROMÁTICA GENERAL

Posición	1	2	3	4	5	6	7	8	9	10	11	12
Nota	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si

Para simplificar el uso de las notas musicales se ha hecho uso del sistema de notación musical inglés o cifrado americano, que asigna una letra a cada nota de la escala (Tabla III).

TABLA III. SISTEMA MUSICAL INGLÉS

Posición	1	2	3	4	5	6	7	8	9	10	11	12
Nota	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si
Nota cifrada	C	C#	D	D#	E	F	F#	G	G#	A	A#	B

1.3.1 Escala Mayor

La escala mayor es una escala diatónica, es decir está compuesta por 8 sonidos, que componen una octava. De esta manera se tiene una estructura básica para formar la escala (tabla IV).

TABLA IV. ESTRUCTURA BÁSICA DE LA ESCALA DIATÓNICA

Nota #	1	2	3	4	5	6	7	8
Intervalo		T	T	ST	T	T	T	ST

Donde:

T = Tono, y

ST = Semitono

Poniendo en práctica lo anterior podemos crear la escala de:

- Do (C) Mayor (Tabla V):

TABLA V. ESCALA C MAYOR

Nota #	C	D	E	F	G	A	B	C
Intervalo		T	T	ST	T	T	T	ST

- Re (D) Mayor (Tabla VI)

TABLA VI. ESCALA D MAYOR

Nota #	D	E	F#	G	A	B	C#	D
Intervalo		T	T	ST	T	T	T	ST

Y de la misma manera con las demás notas de la escala cromática (Tabla VII).

TABLA VII. ESCALAS MAYORES

Nombre Escala	Nota Número						
	1	2	3	4	5	6	7
C Mayor	C	D	E	F	G	A	B
C# Mayor	C#	D#	F	F#	G#	A#	C
D Mayor	D	E	F#	G	A	B	C#
D# Mayor	D#	F	G	G#	A#	C	D
E Mayor	E	F#	G#	A	B	C#	D#
F Mayor	F	G	A	A#	C	D	E
F# Mayor	F#	G#	A#	B	C#	D#	F
G Mayor	G	A	B	C	D	E	F#
G# Mayor	G#	A#	C	C#	D#	F	G
A Mayor	A	B	C#	D	E	F#	G#
A# Mayor	A#	C	D	D#	F	G	A
B Mayor	B	C#	D#	E	F#	G#	A#

1.3.2 Escala Menor

La escala menor al igual que la mayor es una escala diatónica, que está compuesta por 8 sonidos, que componen una octava. La estructura básica para formar la escala se muestra en la tabla VIII.

TABLA VIII. ESTRUCTURA BÁSICA DE LA ESCALA MENOR

Nota #	1	2	3	4	5	6	7	8
Intervalo		T	ST	T	T	ST	T	T

De esta manera podemos construir una escala ejemplo como:

- La (A) Menor (Tabla IX)

TABLA IX. ESCALA A MENOR

Nota #	A	B	C	D	E	F	G	A
Intervalo		T	ST	T	T	ST	T	T

- Si(B) menor (Tabla X)

TABLA X. ESCALA B MENOR

Nota #	B	C#	D	E	F#	G	A	B
Intervalo		T	ST	T	T	ST	T	T

En la tabla XI se muestran todas las escalas menores. Si se eliminan los números de las notas en las escalas anteriormente vistas, se puede notar que existe una relación entre ellas. Por ejemplo:

C Mayor = A Menor

Aunque ambas escalas tienen distinta estructura comparten las mismas notas, en música a esto se le llama escalas relativas, se hace énfasis en esta característica debido a que ayuda a disminuir las salidas en una red neuronal.

TABLA XI. ESCALAS MENORES

Nombre Escala	Nota Número						
	1	2	3	4	5	6	7
C Menor	C	D	D#	F	G	G#	A#
C# Menor	C#	D#	E	F#	G#	A	B
D Menor	D	E	F	G	A	A#	C
D# Menor	D#	E#	F#	G#	A#	B	C#
E Menor	E	F#	G	A	B	C	D
F Menor	F	G	G#	A#	C	C#	D#
F# Menor	F#	G#	A	B	C#	D	E
G Menor	G	A	A#	C	D	D#	F
G# Menor	G#	A#	B	C#	D#	E	F#
A Menor	A	B	C	D	E	F	G
A# Menor	A#	C	C#	D#	F	F#	G#
B Menor	B	C#	D	E	F#	G	A

1.3.3 Frecuencia de las Notas

La escala cromática está formada por 12 notas musicales y cada una de estas posee una frecuencia que la diferencia de las demás, es así que la frecuencia de cada una de esas 12 notas, sigue una representación logarítmica. Se puede determinar la frecuencia de cualquier nota si conocemos la distancia d en semitonos que la separa de otra frecuencia:

$$F2 = F1 * r^d = F1 * 1.059463^d$$

Ejemplo:

Calcular la frecuencia de Mi 5, a una quinta justa por encima de A4

$$\text{Mi5} = \text{A4} * r^d$$

$$\begin{aligned} \text{Mi5} &= 440\text{hz} * 1.059463^7 \\ \text{Mi5} &= 659,255\text{hz} \end{aligned}$$

Además, al tratarse de composición musical las frecuencias se delimitan al espectro audible del ser humano que en un oído joven y sano está comprendido en el rango de 20Hz – 20kHz.

En la tabla XII se pueden ver las frecuencias en el espectro audible de la escala temperada a 440 Hz [7].

TABLA XII. ESCALA TEMPERADA A 440 Hz.

Nota	Frecuencia (Hz)	Longitud de Onda (cm)
C₀	16.35	2109.89
C[#]₀/D^b₀	17.32	1991.47
D₀	18.35	1879.69
D[#]₀/E^b₀	19.45	1774.20
E₀	20.60	1674.62
F₀	21.83	1580.63
F[#]₀/G^b₀	23.12	1491.91
G₀	24.50	1408.18
G[#]₀/A^b₀	25.96	1329.14
A₀	27.50	1254.55
A[#]₀/B^b₀	29.14	1184.13
B₀	30.87	1117.67
C₁	32.70	1054.94
C[#]₁/D^b₁	34.65	995.73
D₁	36.71	939.85
D[#]₁/E^b₁	38.89	887.10
E₁	41.20	837.31
F₁	43.65	790.31
F[#]₁/G^b₁	46.25	745.96
G₁	49.00	704.09
G[#]₁/A^b₁	51.91	664.57

A₁	55.00	627.27
A[#]₁/B^b₁	58.27	592.07
B₁	61.74	558.84
C₂	65.41	527.47
C[#]₂/D^b₂	69.30	497.87
D₂	73.42	469.92
D[#]₂/E^b₂	77.78	443.55
E₂	82.41	418.65
F₂	87.31	395.16
F[#]₂/G^b₂	92.50	372.98
G₂	98.00	352.04
G[#]₂/A^b₂	103.83	332.29
A₂	110.00	313.64
A[#]₂/B^b₂	116.54	296.03
B₂	123.47	279.42
C₃	130.81	263.74
C[#]₃/D^b₃	138.59	248.93
D₃	146.83	234.96
D[#]₃/E^b₃	155.56	221.77
E₃	164.81	209.33
F₃	174.61	197.58
F[#]₃/G^b₃	185.00	186.49
G₃	196.00	176.02
G[#]₃/A^b₃	207.65	166.14
A₃	220.00	156.82
A[#]₃/B^b₃	233.08	148.02
B₃	246.94	139.71
C₄	261.63	131.87
C[#]₄/D^b₄	277.18	124.47
D₄	293.66	117.48
D[#]₄/E^b₄	311.13	110.89
E₄	329.63	104.66
F₄	349.23	98.79

F[#]₄/G^b₄	369.99	93.24
G₄	392.00	88.01
G[#]₄/A^b₄	415.30	83.07
A₄	440.00	78.41
A[#]₄/B^b₄	466.16	74.01
B₄	493.88	69.85
C₅	523.25	65.93
C[#]₅/D^b₅	554.37	62.23
D₅	587.33	58.74
D[#]₅/E^b₅	622.25	55.44
E₅	659.25	52.33
F₅	698.46	49.39
F[#]₅/G^b₅	739.99	46.62
G₅	783.99	44.01
G[#]₅/A^b₅	830.61	41.54
A₅	880.00	39.20
A[#]₅/B^b₅	932.33	37.00
B₅	987.77	34.93
C₆	1046.50	32.97
C[#]₆/D^b₆	1108.73	31.12
D₆	1174.66	29.37
D[#]₆/E^b₆	1244.51	27.72
E₆	1318.51	26.17
F₆	1396.91	24.70
F[#]₆/G^b₆	1479.98	23.31
G₆	1567.98	22.00
G[#]₆/A^b₆	1661.22	20.77
A₆	1760.00	19.60
A[#]₆/B^b₆	1864.66	18.50
B₆	1975.53	17.46
C₇	2093.00	16.48
C[#]₇/D^b₇	2217.46	15.56
D₇	2349.32	14.69

$D^{\#}_7/E^b_7$	2489.02	13.86
E_7	2637.02	13.08
F_7	2793.83	12.35
$F^{\#}_7/G^b_7$	2959.96	11.66
G_7	3135.96	11.00
$G^{\#}_7/A^b_7$	3322.44	10.38
A_7	3520.00	9.80
$A^{\#}_7/B^b_7$	3729.31	9.25
B_7	3951.07	8.73
C_8	4186.01	8.24
$C^{\#}_8/D^b_8$	4434.92	7.78
D_8	4698.63	7.34
$D^{\#}_8/E^b_8$	4978.03	6.93
E_8	5274.04	6.54
F_8	5587.65	6.17
$F^{\#}_8/G^b_8$	5919.91	5.83
G_8	6271.93	5.50
$G^{\#}_8/A^b_8$	6644.88	5.19
A_8	7040.00	4.90
$A^{\#}_8/B^b_8$	7458.62	4.63
B_8	7902.13	4.37

1.4 Tono y Semitono

El tono puede referirse a una característica acústica como se vio en [4], pero también se refiere al intervalo musical que en el sistema temperado es igual a un sexto de octava, se divide en dos partes iguales llamadas semitonos [1].

1.4.1 Relación entre frecuencias de notas en la música occidental

Se debe tomar en cuenta que lo que los seres humanos perciben como una diferencia en el tono es en realidad una relación de frecuencias: es decir, si $\frac{f_1}{f_2} = \frac{f_3}{f_4}$, vamos a escuchar el mismo intervalo entre f_1 y f_2 como lo hacemos entre f_3 y f_4 .

Partiendo de esta relación vista en [8] se calcula el número de semitonos necesarios para cambiar de escala musical.

Si S es la relación de paso que representa un semitono, y sus dos frecuencias son f_1 y f_2 , n es el número de semitonos distantes de la otra frecuencia:

$$\frac{f_1}{f_2} = S^n$$

Se puede encontrar n explícitamente en términos de f_1 , f_2 y S a través de logaritmos:

$$\log \frac{f_1}{f_2} = \log(S^n)$$

$$\log \frac{f_1}{f_2} = n \log(S)$$

$$n = \frac{\log(f_1/f_2)}{\log(S)}$$

Se supone que sabemos f_1 y f_2 . La pregunta que surge es ¿Qué hay de S ?

Un semitono es la doceava parte de una octava y una octava representa una proporción de **2: 1**.

Así

$$S^{12} = 2$$

que es,

$$S = \sqrt[12]{2}$$

Como mejora en logaritmo base 2, esto se convierte en:

$$n = \frac{\log_2 f_1/f_2}{\log_2 \sqrt[12]{2}}$$

$$n = 12 \log \frac{f_1}{f_2}$$

2. Inteligencia Artificial.

La Inteligencia Artificial (IA) es la rama de las Ciencias de la Computación que estudia el software y hardware necesarios para simular el comportamiento y comprensión humanos. Uno de los problemas más difíciles es la simulación de la conciencia, cualidad humana que hace que nos demos cuenta de nuestra propia existencia [9].

Entre las técnicas más utilizadas dentro de la inteligencia artificial están:

- Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.
- Algoritmos genéticos (análogo al proceso de evolución de las cadenas de ADN).
- Redes neuronales artificiales (análogo al funcionamiento físico del cerebro de animales y humanos).
- Razonamiento mediante una lógica formal análogo al pensamiento abstracto humano.

2.1 Redes neuronales

2.1.1 Fundamentos Biológicos

- **Neuronas Biológicas:** Son las células básicas del sistema nervioso que se encargan de la recepción de estímulos y conducción del impulso nervioso (en forma de potencial de acción) entre ellas o con otros tipos celulares [10].

Las neuronas presentan características morfológicas típicas que sustentan sus funciones: un cuerpo celular, llamado soma o «pericarion» central; una o varias prolongaciones cortas que generalmente transmiten impulsos hacia el soma celular, denominadas dendritas; y una prolongación larga, denominada axón o «cilindroeje», que conduce los impulsos desde el soma hacia otra neurona u órgano (Figura 3) [10].

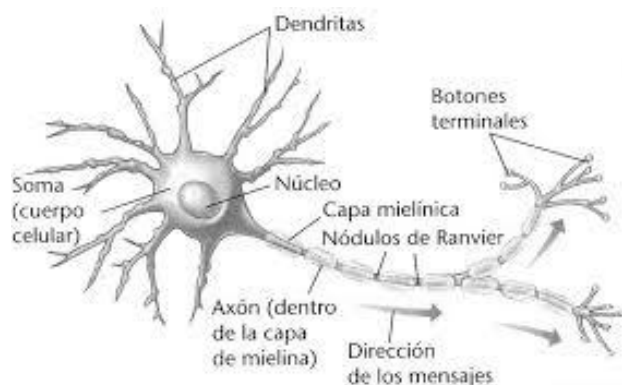


Figura 3. Estructura de la Neurona Biológica

2.1.2 Neurona Artificial

La neurona artificial busca emular las características más importantes de las neuronas biológicas, para esto cuenta con las siguientes características [11]:

- **Valor o Estado de Activación** que en un tiempo t es un valor numérico que se representa por un vector $A_i(t)$ y puede ser continuo o discreto, limitado o ilimitado. Lo más simple es considerarlo “1” y “0”, como on/off o activo/pasivo.
- **Conjunto de entradas** o vector de entradas x , de n componentes
- **Conjunto de pesos sinápticos** w_{ij} . Representan la interacción entre la neurona presináptica j y la postsináptica i .
- **Regla de propagación** $d(w_{ij}, x_j(t))$: proporciona el potencial postsináptico, $h_i(t)$.
- **Función de activación** $a_i(t) = f(a_i(t-1), h_i(t))$: proporciona el estado de activación de la neurona en función del estado anterior y del valor postsináptico.
- **Función de salida** $F_i(t)$: proporciona la salida $y_i(t)$, en función del estado de activación. En la figura 4 se pueden apreciar estas características

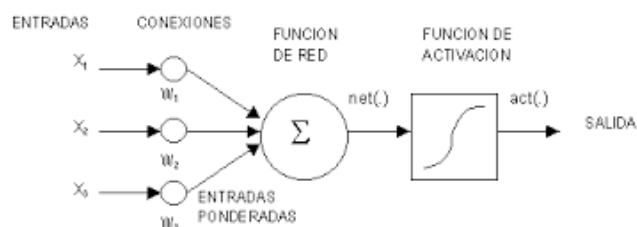


Figura 4. Neurona Artificial

2.1.3 Redes Neuronales Artificiales

Las RNAs² tratan de emular el comportamiento del cerebro humano caracterizado por el aprendizaje a través de la experiencia y la extracción de un conocimiento genérico a través de un conjunto de datos. La principal diferencia entre los sistemas computacionales y el cerebro biológico es la manera de procesar la información, dado que la arquitectura en los primeros es de tipo Von Neumann, por lo que realizan los procesos mediante una serie de tareas mientras que en el segundo, la red neuronal se encarga de procesar la información en paralelo [10].

Una red neuronal es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Esquemáticamente cada elemento de procesamiento (neuronas) de la red se representa como un nodo. Estas conexiones establecen una estructura jerárquica que tratando de emular la fisiología del cerebro busca nuevos modelos de procesamiento para solucionar problemas concretos del mundo real [10].

La posibilidad de resolver problemas difíciles es dable gracias a los principios de las redes neuronales. Estos son enunciados a continuación:

- **Aprendizaje adaptativo:** Las RNA pueden comportarse en función de un entrenamiento con una serie de ejemplos ilustrativos. De esta forma, no es necesario elaborar un modelo a priori, ni establecer funciones probabilísticas. Puede modificarse constantemente con el fin de adaptarse a nuevas condiciones de trabajo [11].
- **Autoorganización:** Consiste en la modificación de la red completa con el fin de llevar a cabo un objetivo específico. También significa generalización, de esta forma una red puede responder a datos o situaciones que no ha experimentado antes, pero que puede inferir sobre la base de su entrenamiento. Esta característica es muy útil sobre todo cuando la información de entrada es poco clara o se encuentra incompleta [11].

² RNA: Red Neuronal Artificial

- **Tolerancia a fallos:** Pueden reconocer patrones de información con ruido, distorsión o incompletos, pero que, además, pueden seguir trabajando aunque se destruya parte de la red (con cierta degradación). Mientras la computación tradicional almacena la información en espacios únicos, localizados y direccionables, las redes neuronales lo hacen de forma distribuida y con un alto grado de redundancia. [11].
- **Operación en tiempo real:** Trabajan en paralelo actualizando todas sus instancias simultáneamente. Es importante destacar que esta característica solo se aprecia cuando se implementan redes con hardware especialmente diseñados para el procesamiento paralelo [11].
- **Fácil inserción en la tecnología existente:** Es relativamente sencillo obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilita la integración modular en los sistemas existentes.

2.1.3.1 Topología de la RNA

Una primera clasificación de las redes de neuronas artificiales que se suele hacer es en función del patrón de conexiones que presenta. Así se definen tres tipos básicos de redes [10] :

- Monocapa. Ejemplos: perceptrón, Adaline.
- Multicapa. Ejemplos: perceptrón multicapa.
- Las redes recurrentes que presentan al menos un ciclo cerrado de activación neuronal. Ejemplos: Elman, Hopfield, máquina de Boltzmann.

2.1.3.2 Aprendizaje

Una segunda clasificación que se suele hacer es en función del tipo de aprendizaje de que es capaz (si necesita o no un conjunto de entrenamiento supervisado). Para cada tipo de aprendizaje encontramos varios modelos [11]:

- **Aprendizaje supervisado:** Es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. La salida de la función puede ser un valor numérico (como en los problemas de regresión) o una etiqueta de clase (como en los de clasificación). El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente [10].

Las redes de entrenamiento supervisado han sido los modelos de redes más desarrolladas desde inicios de estos diseños. Los datos para el entrenamiento están constituidos por varios pares de patrones de entrenamiento de entrada y de salida. El hecho de conocer la salida implica que el entrenamiento se beneficia de la supervisión de un maestro (Figura 5).

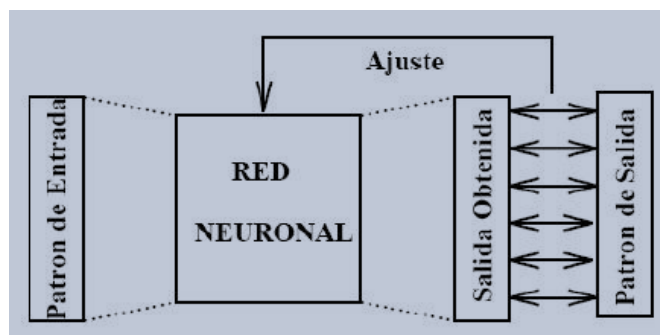


Figura 5. Esquema del Aprendizaje Supervisado en RNA

- **Aprendizaje no supervisado o autoorganizado:** no necesitan de tal conjunto previo. Ejemplos de este tipo de redes son: las memorias asociativas, las redes de Hopfield, la máquina de Boltzmann y la máquina de Cauchy, las redes de aprendizaje competitivo, las redes de Kohonen o mapas autoorganizados y las redes de resonancia adaptativa (ART).

2.1.4 Perceptrón Multicapa (MLP)

Se hace énfasis en esta arquitectura debido a que es simple y proporciona el soporte para problemas no lineales como lo es el musical.

El perceptrón multicapa (Multilayer Perceptron) es una generalización del perceptrón simple y nació como alternativa ante la imposibilidad del simple para resolver problemas no lineales. Es una red unidireccional (feedforward) que se caracteriza porque tiene neuronas agrupadas en capas sigmoidales de diferentes niveles, se distinguen tres tipos de capas: de entrada, las capas ocultas y la de salida. Las entradas a la red inciden sobre las unidades que constituyen la capa de entrada, estos sensores propagan las señales a una serie de nodos que forman la primera capa oculta, de esta capa parten conexiones hacia la siguiente, y de ésta a la siguiente, hasta llegar a la capa de salida, cuya salida es la red total (figura 6) [11].

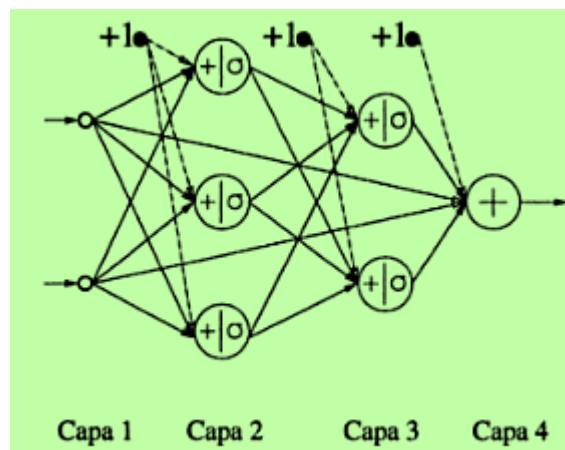


Figura 6. Perceptrón Multicapa

El número de entradas depende de los datos recolectados y las características que se desea clasificar [11].

El número de capas ocultas depende de la complejidad del problema a resolver, aunque en la práctica, no existe razón para usar más de dos capas ocultas debido a que cada capa oculta adicional da más poder a la red neuronal, sin embargo el costo computacional se incrementa de la misma manera, para problemas prácticos lo

recomendable es usar una capa oculta, si el uso de varias neuronas en una sola capa oculta no resuelve el problema, es necesario usar otra capa oculta.

El número de neuronas en la capa oculta viene dado de acuerdo a la manera prueba-error, donde se va probando cantidades de neuronas distintas, hasta que se satisfaga el problema, si se usa pocas neuronas se puede producir **underfitting** es decir la red no soluciona el problema; si se usan demasiadas neuronas en cambio se puede producir **overfitting** (sobre entrenamiento, lo que ocasiona particularización).

El número de neuronas en la capa de salida viene dado por las salidas que se desea obtener.

3. Música por Ordenador

3.1 Procesamiento digital de señales

El procesamiento de señales trata de la representación, transformación y manipulación de señales y de la importancia que contienen. Cuando se refiere al procesado digital de señales (DSP por sus siglas en inglés), se refiere a la representación mediante secuencias de números de precisión finita utilizando un computador digital.

Para definir las tareas del DSP se requiere primero precisar los conceptos de señal y sistema.

La señal es considerada como aquella observación de una magnitud física en función de variables independientes de tiempo y espacio, realizada de tal modo que contenga información de los procesos observados [12].

En general, toda señal contiene información que se desea extraer o modificar de acuerdo a los requisitos de cada aplicación particular. Las señales son representadas por funciones matemáticas de una o más variables y dichas funciones pueden ser escalares o vectoriales.

El término sistema denota a una colección o conjunto de elementos interrelacionados que conforman un todo unificado. Un sistema puede formar parte de otro sistema de mayor nivel, en cuyo caso al primero se le denomina subsistema del segundo. Los diferentes subsistemas intercambian por lo general información, materia o energía para lograr algún objetivo [12].

Los términos señales de entrada o de salida, se utilizan entonces para abstraer ese flujo de información, materia o energía en el concepto matemático de funciones. El sistema entonces puede interpretarse como un conjunto de subsistemas que logran transformar una señal en otra.

3.1.1 Transformada de Fourier

La Transformada de Fourier es una de las herramientas principales de análisis con que se cuenta hoy en los medios de ciencia y tecnología. Su poder radica en la posibilidad de establecer relaciones entre puntos de vista muy diferentes relativos a un mismo problema [12].

Así, al visualizar un fenómeno tanto en términos de la función asociada, como de su transformada de Fourier, se tiene frecuentemente un procedimiento de análisis útil para resolver un problema determinado.

a. Transformada discreta de Fourier (DFT)

Transforma una función matemática en otra, obteniendo una representación en el dominio de la frecuencia, siendo la función original una función en el dominio del tiempo. Pero la DFT requiere que la función de entrada sea una secuencia discreta y de duración finita. Dichas secuencias se suelen generar a partir del muestreo de una función continua [13]. Es por esto que la mayoría de las características importantes que se utilizan para analizar el contenido de audio se definen en el dominio de la frecuencia.

La Transformada Discreta de Fourier de una señal $x[n]$ definida en el rango $0 \leq n \leq N-1$ se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \cdot \frac{2\pi}{N} k \cdot n} = \sum_{n=0}^{N-1} x[n] \cdot W_N^{k \cdot n} ; 0 \leq k \leq N-1$$

b. Transformada rápida de Fourier

La Transformada Rápida de Fourier (FFT) no es una nueva transformada sino que se trata de un algoritmo para el cálculo de la Transformada Discreta de Fourier. Su importancia radica en el hecho que elimina una gran parte de los cálculos repetitivos a que está sometida la DFT, por lo tanto se logra un cálculo más rápido. Además, la FFT generalmente permite una mayor precisión en el cálculo de la DFT disminuyendo los errores de redondeo [14].

La implementación del algoritmo de la FFT puede realizarse de dos formas distintas:

- 1.- Mediante un programa que pueda ejecutarse tanto en un PC como en una tarjeta que posea un microprocesador específico para este tipo de operaciones (DSP).
- 2.- Mediante el desarrollo de una tarjeta (HARDWARE) en la cual se emplean circuitos integrados específicos. Tal es el caso de los modernos analizadores de espectro.

3.2 Formato WAV

Es importante tomar en cuenta las características del audio que se requieren al momento de realizar un análisis exhaustivo de su composición.

WAV (o WAVE), apócope de WAVE form audio file format, es un formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft y de IBM que se utiliza para almacenar sonidos en el PC, admite archivos mono y

estéreo a diversas resoluciones y velocidades de muestreo, su extensión es **.wav**. A pesar de que el formato WAV es compatible con casi cualquier códec de audio, se utiliza principalmente con el formato PCM (no comprimido) y, al no tener pérdida de calidad, es adecuado para uso profesional. Para tener calidad CD de audio se necesita que el sonido se grabe a 44100 Hz y a 16 bits. Por cada minuto de grabación de sonido se consumen unos 10 megabytes de espacio en disco. Una de sus grandes limitaciones es que solo se pueden grabar archivos de 4 gigabytes como máximo, lo cual equivale aproximadamente a 6,6 horas en calidad de CD de audio. Es una limitación propia del formato, independientemente de que el sistema operativo donde se utilice sea MS Windows u otro distinto, y se debe a que en la cabecera del fichero se indica la longitud del mismo con un número entero de 32 bits, lo que limita el tamaño del fichero a un máximo de 4294967295 bytes (O 4 gigabytes) [15].

Para el análisis de una melodía este formato es muy conveniente, porque a pesar de que es pesado al almacenar conserva todas las características originales.

e. Materiales y Métodos.

Durante el proceso de desarrollo del Trabajo de Titulación, se recurre a diferentes técnicas de recolección de información, métodos científicos y procedimientos que la investigación científica ofrece y que son de mucha utilidad. Entre los principales métodos de investigación científica que se manejaron tenemos:

- **Observación activa:**

Con este método se logró involucrar en el problema de investigación, sobre cómo utilizan técnicas de I. A. para resolver problemas inherentes a la creatividad del ser humano y los recursos necesarios para llevarlo a cabo.

- **Estudio de casos:**

La utilización de este método permitió realizar una investigación basada en casos de éxito de I. A. en el ámbito musical. Como resultado se pudo conocer el nivel científico de este campo de aplicación tanto a nivel nacional como internacional. Así mismo se pudo determinar la factibilidad del tema para su desarrollo en determinado plazo de tiempo y con las herramientas disponibles.

- **Experimentación:**

Con la utilización de esta técnica se realizó la implementación de la aplicación prevista, haciendo las respectivas pruebas tanto a nivel personal como a nivel profesional, para determinar la certeza de los resultados obtenidos.

- **Técnicas de recolección de información:**

Permitió la recolección de información de las principales temáticas que comprenden la teoría musical y la aplicación de I. A. en este campo, constituyéndose en la base teórica del proyecto, ver (*Sección Revisión de Literatura*).

La revisión bibliográfica se realizó en bases de datos científicas, entre la documentación revisada tenemos: libros, revistas tesis y artículos científicos.

Metodología del Trabajo de Titulación

Para el desarrollo del software que contempla el proyecto se tomó en cuenta la metodología XP, adaptándola a las necesidades del tema debido a que dicha metodología se orienta a la programación de objetos y no a la creación de redes neuronales. Las fases contempladas son las siguientes:

a. Planificación del proyecto

Contempla la redacción de historias de usuario, que pueden ir cambiando o desapareciendo en cada nueva iteración según las necesidades del cliente, al final solo quedan las historias que representan los verdaderos requisitos del sistema.

b. Diseño

Una vez definidos los requisitos, se realiza el prototipo de la aplicación, aplicando lo que indica la metodología: hacer lo justo. Por esto se ha tomado en cuenta solamente los aspectos más relevantes del sistema que luego serán codificados. En lo referente a la red neuronal, en este paso se definió las entradas, salidas y arquitectura de red.

c. Codificación

En esta fase se realizó la creación del código del módulo de captura de audio, la creación y entrenamiento de la red neuronal, siempre bajo la atenta observación del usuario final.

d. Pruebas

Finalmente, se realizaron las pruebas del sistema, de modo que se pueda comprobar que cumple con todos los requisitos planteados en las historias de usuario.

f. Resultados

Los resultados describen la manera en que se aplicó la metodología, definida en el apartado anterior, para la consecución de los objetivos planteados al inicio del proyecto.

Para cumplir con los objetivos, se definieron fases por cada uno de ellos y a su vez cada fase está dividida en tareas.

Fase 1: Analizar las tecnologías existentes para el desarrollo de la aplicación de tal manera que se satisfagan los requerimientos existentes.

1. Analizar casos de éxito en el desarrollo de aplicaciones inteligentes para composición musical.

Para presentar un caso de aplicación de la IA en la música y definirlo cómo exitoso se tomó en cuenta características como la actualidad del tema, por lo que se han elegido estudios desde el 2010 hasta la fecha. Otra característica importante es el aval que garantice la autenticidad, funcionalidad y el aporte al campo de estudio, por esto es que los casos están tomados de artículos científicos publicados en revistas indexadas. La tabla XIII muestra los casos de éxito y un abstracto de su contenido.

Discusión: Los casos de éxito revisados (TABLA XIII) muestran que es factible la utilización de IA en el ámbito musical, dadas las características de la computación actual. Se utilizan diversas técnicas de IA dependiendo del fin de la investigación. Se puede notar que los estudios que utilizan redes neuronales se enfocan en el reconocimiento de características que hacen distinguible al oído humano una pieza musical de otra. En estas características está el ritmo, la melodía, la estructura y la tonalidad cómo es el caso del presente trabajo.

TABLA XIII. CASOS DE ÉXITO DE APLICACIONES DE IA EN LA MÚSICA.

Caso de Éxito	Características
Composición Musical usando Algoritmos Genéticos [16]	Este artículo presenta un modelo de composición musical discreto, basado en las técnicas de escritura musical propuestas por Johann Sebastian Bach. El modelo fue implementado mediante el uso de la estrategia metaheurística: algoritmos genéticos implantados sobre el Framework JGap, el cual arroja resultados de composición con una sintaxis que cumple con las técnicas de escritura y composición musical.
Síntesis de audio musical con AUTOENCODING NEURAL NETS [17]	Propone la utilización de redes neuronales que optimicen su arquitectura por medio de la auto codificación para aplicaciones musicales creativas, en este caso como sintetizadores de audio.
Modelos computacionales para composición musical algorítmica [18].	El artículo presenta un modelo basado en la matriz de transición de eventos musicales (patrones musicales descritos por notas, medida y duración) y la clasificación de las partes instrumentales que aparecen en la obra musical de entrada. Un ejemplo de implementación del modelo se describe utilizando la caja de herramientas MIDI implementado en Matlab.
Artificial Neural Networks in Musical Performance [19]	Este artículo presenta un modelo de composición musical basado en el uso de redes neuronales de alimentación directa (feed forward network). La idea es construir una pieza de software (plugin) para supercollider que es un entorno y lenguaje de programación para la síntesis de audio en tiempo real y la composición algorítmica. De tal manera que esta herramienta soporte la composición musical a través de redes neuronales

Inmamusys: Sistema musical inteligente multiagente [20] .	Sistema multiagente que compone música por su cuenta mediante el uso de la tecnología de los Sistemas Expertos y en base sus principios. Utiliza reglas emitidas a partir del conocimiento musical y como entradas las emociones de usuarios. Su objetivo principal es la investigación de las relaciones entre ordenadores y emociones y cómo este último segundo se puede representar a través de los primeros.
The Wekinator: a system for real-time, interactive machine learning in music [21].	Este software ofrece una amplia gama de algoritmos de aprendizaje automático (otros aparte de las redes neuronales), con el fin de centrarse en la capacidad de un artista para crear un mapeo altamente personalizado entre gestos (rastreados por cualquier tipo de controlador que puede interactuar con un ordenador) y parámetros de síntesis de sonido.

2. Comparar herramientas para la creación de redes neuronales y elegir la que mejor se adapte al proyecto.

TABLA XIV. COMPARATIVA DE HERRAMIENTAS PARA CREACIÓN DE RNAS

CARACTERÍSTICAS	MATLAB [22]	JAVA [23]
API	Neural Network Toolbox	Weka
Orientación	Redes neuronales	Minería de Datos
Arquitecturas soportadas	<ul style="list-style-type: none"> - Feedforward - Base Radial - Dinámicas - Aprendizaje Cuantificación vectorial. 	<ul style="list-style-type: none"> - Feedforward - Base Radial
Funciones de aprendizaje	<ul style="list-style-type: none"> - Several gradient descent methods - Conjugate gradient methods - The Levenberg-Marquardt algorithm (LM) - The resilient backpropagation algorithm (Rprop). 	<ul style="list-style-type: none"> - Several gradient descent methods - Conjugate gradient methods
Parallel Computing Toolbox	X	
Interfaz Gráfica	X	X

Discusión: Luego de la comparativa se concluyó que la herramienta que mejor se adapta a las necesidades del proyecto es Matlab, por sus características en lo que se refiere a la documentación y el soporte para la creación de redes. El Neural Network Toolbox de MatLab está orientado específicamente a las redes neuronales por lo que cuenta con herramientas específicas para ello, no así Weka que está más orientado a la minería de datos y a pesar de contar con soporte para redes neuronales, solo se enfoca en lo básico.

3. Seleccionar las alternativas tecnológicas para la captura de sonido en tiempo real que se adapten de mejor manera a los requerimientos del sistema.

TABLA XV. COMPARATIVA DE HERRAMIENTAS PARA CAPTURAR AUDIO

CARACTERÍSTICA	MATLAB [22]	JAVA [24]
Librería o API	DSP Señal Processing Toolbox	Java Sound
Técnica de codificación	PCM	ALAW, PCM_SIGNED, PCM_UNSIGNED y ULAW
Rangos de frecuencia	8 Khz hasta 48 Khz	8 Khz hasta 48 Khz
Tipo de canal	Mono y estéreo	Mono y estéreo
Frecuencia de muestreo	8000, 11025, 22050 y 44100	
Formato	MP3, MP4, M4A (H.264 / audio AAC), WAVE, FLAC, OGG	WAVE, AIFF, AU
Número de bits	8, 16, 24 , 32	8, 16, 24 , 32
Interfaz Gráfica	Disponible	Programable
Gráficas de las características del espectro	Completo	Básico
Algoritmos DSP	Incluidos	

Discusión: Tras realizar la comparación de herramientas se prefirió Matlab, debido al toolbox de reconocimiento de señales que posee, dicho toolbox facilita el trabajo del programador al momento de generar gráficas de las características del sonido capturado, además integra varios algoritmos y funciones de transferencia propios de DSP por lo que minimiza el tiempo de codificación.

Fase 2: Desarrollar un módulo para la captura y procesamiento de sonido en tiempo real que sirva como entrada de datos a la red neural.

1. Definir una metodología de desarrollo, de acuerdo a los objetivos planteados y al tiempo establecido para la presentación de resultados.

Para el desarrollo del software se tomó en cuenta la metodología XP porque es un proceso ágil de desarrollo, ideal para proyectos con plazos cortos de tiempo. Además es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto (adaptabilidad), es iterativo, recomienda la programación en parejas y promueve la idea de simplicidad en el código, enfocarse en lo esencial. Las fases y presentables que pide la metodología se pueden revisar en el apartado de Materiales y Métodos del presente documento.

Hoy en día no existen metodologías para el desarrollo de redes neuronales que sean conocidas y/o aplicadas en nuestro medio, por lo que se adaptó en lo posible la XP al desarrollo de la red, aun así se siguió el workflow propuesto por los creadores del Neural Network Toolbox de Matlab [22] que consta de las siguientes actividades:

- Recolectar Datos
- Obtención del Dataset
- Crear la red Neuronal
- Configurar la Red Neuronal
- Entrenar la Red Neuronal
- Validar la Red Neuronal
- Usar la Red Neuronal

Cada una de estas actividades y sus resultados se describen en la fase 3 que corresponde a la implementación de la red neuronal.

2. Historias de usuario.

Las historias de usuario se contemplan en la metodología XP y se realizaron para cumplir con dos de las tareas planificadas en el anteproyecto:

- Realizar el análisis y especificación de requisitos del sistema a implementar, con base en características de aplicaciones ya existentes en el mismo campo y en busca de solucionar el problema planteado en el presente.
- Definir los casos de uso del sistema para satisfacer los requerimientos del sistema

La redacción inicial de las historias fue cambiando a lo largo del desarrollo debido a las conversaciones continuas con el cliente. Algunas de estas fueron eliminadas o cambiadas a lo largo del proyecto, a medida que cambiaban los requisitos del cliente o se tenía una concepción más clara del proyecto. Aquí se presentan las historias de usuario finales (Tablas: XVI - XXI).

TABLA XVI. HISTORIA DE USUARIO 1

Historia de Usuario	
Número: 1	Usuario: Usuario final.
Nombre historia: Captura de audio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Juan Pinzón – Rodrigo Granda	
Descripción: <p>El usuario selecciona del menú principal la opción para capturar audio. Se muestra una nueva interfaz donde se puede capturar audio ya sea externo o de un instrumento conectado a la entrada de audio del computador.</p> <p>La captura empieza al presionar START.</p>	
Observaciones:	

TABLA XVII. HISTORIA DE USUARIO 2

Historia de Usuario	
Número: 2	Usuario: Usuario final.
Nombre historia: Presentación de características de la captura	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Juan Pinzón – Rodrigo Granda	
<p>Descripción:</p> <p>La interfaz de captura de audio presenta un plano cartesiano bidimensional, donde el eje x representa la duración en segundos de la captura y en el eje y presenta las notas musicales o el rango de frecuencias en que se puede tocar la guitarra melódica. Al capturar audio se representa en el plano el tiempo y la nota o frecuencia representativa de la captura.</p> <p>La captura se detiene al presionar STOP.</p>	
Observaciones: El usuario puede elegir notas o frecuencia para representar la captura.	

TABLA XVIII. HISTORIA DE USUARIO 3

Historia de Usuario	
Número: 3	Usuario: Usuario final.
Nombre historia: Guardar Audio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 2
Programador responsable: Juan Pinzón – Rodrigo Granda	
<p>Descripción:</p> <p>Luego de capturar audio, el usuario puede archivar dicha captura, para esto presiona SAVE FILE en la interfaz de captura y el sistema se direcciona al directorio donde se encuentra la aplicación y le da al archivo un nombre predeterminado, que puede ser modificado por el usuario.</p> <p>Al no guardar el archivo, este se descarta y se puede iniciar una nueva captura.</p>	
<p>Observaciones: El archivo a guardar será de tipo wav, debido a que conserva mejor sus características y es posible un mejor análisis.</p>	

TABLA XIX. HISTORIA DE USUARIO 4

Historia de Usuario	
Número: 4	Usuario: Usuario final.
Nombre historia: Cargar archivo de audio para análisis	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 3
Programador responsable: Juan Pinzón – Rodrigo Granda	
<p>Descripción:</p> <p>El usuario puede realizar el análisis musical a un archivo de audio almacenado previamente. Puede ingresar a la interfaz de análisis ya sea desde la interfaz principal o desde la interfaz de captura.</p> <p>Una vez en la interfaz de análisis al seleccionar CARGAR, se direcciona por defecto al directorio donde se encuentra la aplicación para que el usuario elija el archivo a cargar.</p>	
Observaciones: Solamente se reconocen archivos wav para cargar al sistema.	

TABLA XX. HISTORIA DE USUARIO 5

Historia de Usuario	
Número: 5	Usuario: Usuario final.
Nombre historia: Reproducir Audio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: Ninguna
Programador responsable: Juan Pinzón – Rodrigo Granda	
Descripción: Luego de cargar un archivo el usuario puede reproducirlo de modo que si se trata de una captura realizada, pueda escuchar el resultado de su composición. Le es posible también pausar y detener la reproducción el cualquier momento.	
Observaciones:	

TABLA XXI. HISTORIA DE USUARIO 6

Historia de Usuario	
Número: 6	Usuario: Usuario final.
Nombre historia: Realizar el análisis del archivo de audio.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 3
Programador responsable: Juan Pinzón – Rodrigo Granda	
Descripción: <p>Luego de cargar un archivo se realiza el análisis al presionar ANALIZAR SONIDO. Se muestra una gráfica con el efecto de frecuencia que genera el sonido, y según la gráfica el sistema presenta al usuario el tono en que se encuentra la el audio cargado y las posibles notas a tocar para dar armonía a la melodía creada.</p>	
Observaciones:	

3. Prototipado del sistema

Los prototipos se realizaron para cumplir otra tarea definida en el anteproyecto:

- Realizar el diseño lógico del módulo a implementar, de acuerdo al análisis realizado anteriormente.

Una vez definidos los requisitos se procedió a realizar los prototipos de las interfaces de la aplicación que muestran los aspectos relevantes para el usuario, como son la disposición de los elementos dentro de la interfaz y las funcionalidades (Figuras 7- 8).

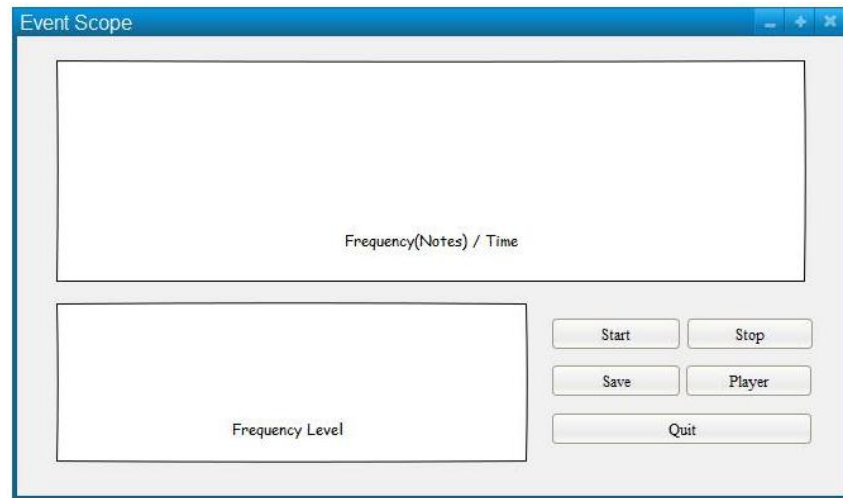


Figura 7. Prototipo de Interfaz de Captura de Audio

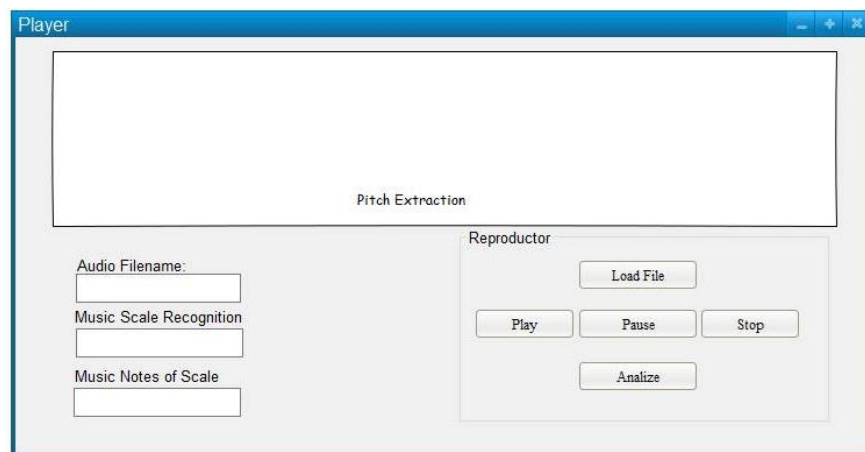


Figura 8. Prototipo Interfaz de Análisis de Audio

4. Implementación del módulo en un lenguaje de programación que permita el procesamiento de sonido en tiempo real.

Para la captura de audio se han tomado algunas temáticas del procesamiento digital de señales (DSP), debido que el objetivo es básicamente la representación, transformación y manipulación de la señal proveniente de la tarjeta de audio del pc, de tal manera que se pueda utilizar como entrada para la red neuronal.

4.1. Grabación de audio con Matlab

La digitalización de una señal analógica requiere dos procesos. Primero la toma de muestras la amplitud de la señal (muestreo) y luego la asignación a las muestras de valores proporcionales (cuantificación).

Obtener la señal de audio con Matlab se puede hacer de dos formas: una es mediante el Data Acquisition Toolbox y otra es mediante el core de Matlab a través de la función **audiorecorder**. Para el desarrollo del proyecto se ha optado por el core de Matlab ya que presenta mayor compatibilidad con las tarjetas de audio existentes.

La función **audiorecorder** crea un objeto para la grabación de audio a través de una entrada de audio del pc como puede ser un micrófono a través de la tarjeta de audio donde se ejecuta esta función.

Existen 3 maneras de llamar a esta función:

```
recorder = audiorecorder
recorder = audiorecorder(Fs,nBits,nChannels)
recorder = audiorecorder(Fs,nBits,nChannels,ID)
```

Donde:

- **Fs** es la tasa de muestras in Hz, los valores válidos dependen del hardware de sonido específico instalado en la máquina. Valores típicos pueden ser 8000, 11025, 22050, 44100, 48000, and 96000 Hz. Por default se usa 8000, pero en nuestro caso se tomó un valor de 44100 por el teorema de Nyquist.
- **nBits** son los bits por muestra. Valores válidos dependen del hardware instalado pudiendo ser 8, 16, o 24.
- **nChannels** es el número de canales para grabar 1(mono) o 2(estéreo), por default se usa 1, mismo valor que se utilizó en la aplicación debido al instrumento que en este caso es la guitarra melódica.
- **ID** es el identificador de dispositivo, para obtener el ID de un dispositivo se usa la función *audiodevinfo*, el valor por default que usa el dispositivo de audio del computador es -1.

A continuación un ejemplo para grabar 5 segundos de audio.

```
% Grabar sonido por 5 segundos.
recObj = audiorecorder;
disp('Inicia grabacion.')
recordblocking(recObj, 5);
disp('Termina Grabación. ');
% Reproducir grabacion.
play(recObj);
% Almacenar datos en un arreglo de doble precisión.
myRecording = getaudiodata(recObj);
% Plotear la forma de onda.
plot(myRecording);
```

Una dato importante a tomar en cuenta son los armónicos: cuando una cuerda de 36 cm se rasga, no sólo se produce una onda de 36 cm, sino que además se forman dos ondas de 18, cuatro de 9 y así sucesivamente cada una de estas vibraciones, producen una secuencia de armónicos, 1/2, 1/3, 1/4... de la longitud de la cuerda. Los sonidos son más agudos y mucho más suaves que el sonido de la cuerda completa y no se perciben de forma clara precisa y consciente pero son los que hacen que los instrumentos musicales suenen diferentes entre sí.

4.2. Procesamiento Digital de la Señal (DSP)

Una vez capturado el audio se usaron técnicas de DSP para poder transformarlo en datos entendibles para la red neuronal.

El teorema de muestreo que se utilizó es el de Nyquist [25], que indica que para poder digitalizar correctamente todo el ancho de banda que puede percibir el oído humano se deben tomar muestras del doble de frecuencia (44.1 Hz). También nos afirma que en sistemas analógicos, parte de la señal es información útil y parte de ella representa ruido. La relación entre las potencias de ambas se denomina Relación Señal/Ruido (SNR):

$$SNR = 10\log \frac{V_{signal}^2}{V_{ruido}^2} = 20\log \frac{V_{signal}}{V_{ruido}}$$

Se obtuvieron las ocurrencias de cada frecuencia en la muestra de sonido (Pitch Extraction), para esto, primero se obtuvieron los coeficientes de la transformada discreta de Fourier, es decir los frames en la muestra de audio. A continuación, el código utilizado para implementar la transformada discreta de Fourier (DFT).

```
function [FFT, Freq] = getDFT(signal, Fs, PLOT)
%
% function [FFT, Freq] = getDFT(signal, Fs, PLOT)
N = length(signal); % length of signal
% compute the magnitude of the spectrum
% (and normalize by the number of samples):
FFT = abs(fft(signal)) / N;

if nargin==2 % return the first half of the spectrum:
    FFT = FFT(1:ceil(N/2));
    Freq = (Fs/2) * (1:ceil(N/2)) / ceil(N/2); % define the
frequency axis
else
    if (nargin==3)
        % ... or return the whole spectrum
        % (in the range -fs/2 to fs/2)
        FFT = fftshift(FFT);
        if mod(N,2)==0 % define the
frequency axis:
            Freq = -N/2:N/2-1; % if N is even
        else
            Freq = -(N-1)/2:(N-1)/2; % if N is odd
        end
        Freq = (Fs/2) * Freq ./ ceil(N/2);
    end
end
```

Usando este código obtenemos el rango de frecuencias o coeficientes que ocurren en la muestra de sonido.

La precisión de los analizadores de Fourier se evalúa a través del número de líneas que pueden representar, siendo los valores más habituales los de 256, 400 y 800 líneas.

Cada línea corresponde a una banda de frecuencia de ancho constante y de valor el de la frecuencia más alta analizada dividido por el número de líneas calculadas. Así, por ejemplo, si obtenemos un espectro en el que la frecuencia más alta sea de 1,6 kHz y tenemos un analizador de 800 líneas, entonces el ancho de cada línea será de $1,6 \text{ kHz} / 800 \text{ líneas} = 2 \text{ Hz/línea}$.

Una de las grandes ventajas del análisis FFT que se utilizó en este caso es la posibilidad de efectuar un zoom de una zona concreta del espectro obtenido, donde el sentido de zoom es el mismo que en fotografía. Es decir, si tenemos un espectro de 800 líneas de una señal cualquiera, un zoom nos permite efectuar una ampliación de una parte concreta del espectro que nos interese, con lo que el grado de resolución es extraordinariamente elevado.

Una vez obtenidos los coeficientes pasamos a hacer la extracción de características del sonido (Feature Extraction) mediante la siguiente función:

```
function Features = stFeatureExtraction(signal, fs, win, step)
% if STEREO ...
if (size(signal,2)>1)
    signal = (sum(signal,2)/2); % convert to MONO
end

% convert window length and step from seconds to samples:
windowLength = round(win * fs);
step = round(step * fs);

curPos = 1;
L = length(signal);

% compute the total number of frames:
numOfFrames = floor((L-windowLength)/step) + 1;
% number of features to be computed:
```

```

numOfFeatures = 35;
Features = zeros(numOfFeatures, numOfFrames);
Ham = window(@hamming, windowLength);
mfccParams = feature_mfccs_init(windowLength, fs);

for i=1:numOfFrames % for each frame
    % get current frame:
    frame = signal(curPos:curPos+windowLength-1);
    frame = frame .* Ham;
    frameFFT = getDFT(frame, fs);

    if (sum(abs(frame))>eps)
        % compute time-domain features:
        Features(1,i) = feature_zcr(frame);
        Features(2,i) = feature_energy(frame);
        Features(3,i) = feature_energy_entropy(frame, 10);

        % compute freq-domain features:
        if (i==1) frameFFTPrev = frameFFT; end;
        [Features(4,i) Features(5,i)] = ...
            feature_spectral_centroid(frameFFT, fs);
        Features(6,i) = feature_spectral_entropy(frameFFT, 10);
        Features(7,i) = feature_spectral_flux(frameFFT, frameFFTPrev);
        Features(8,i) = feature_spectral_rolloff(frameFFT, 0.90);
        MFCCs = feature_mfccs(frameFFT, mfccParams);
        Features(9:21,i) = MFCCs;

        [HR, F0] = feature_harmonic(frame, fs);
        Features(22, i) = HR;
        Features(23, i) = F0;
        Features(23+1:23+12, i) = feature_chroma_vector(frame, fs);
    else
        Features(:,i) = zeros(numOfFeatures, 1);
    end
    curPos = curPos + step;
    frameFFTPrev = frameFFT;
end

```

```
Features(35, :) = medfilt1(Features(35, :), 3);
```

En este caso lo que nos interesa es obtener el tono real o frecuencia 0, en cada instante de la muestra de audio por lo que se usa la característica #23 F0.

Las funciones anteriormente presentadas anteriormente han sido obtenidas de [28].

Para usar estas funciones hemos implementado la siguiente función:

```
function [t,freq]= pitchExtraction(file)

% Abre el archivo de audio:
[x,fs] = audioread(file);

% feature extraction:
Features = stFeatureExtraction(x, fs, 0.10, 0.020);

% Saca el pitch (23 feature):
F = Features(23,:);

% Calcula el vector de tiempo:
timeFeature = 0.020:0.020:length(x)/fs;

% post-proceso vector de tiempo, vector de señales:
MIN1 = min([length(F);length(timeFeature)]);
timeFeature = timeFeature(1:MIN1);
t=timeFeature;
F = F(1:MIN1);
freq=F;

plot(timeFeature, F);
hold on
xlabel('Time (sec)');
ylabel('Pitch (Hz)');
legend({'Estimado'});
```

Lo que hace esta función es obtener las frecuencias en cada instante de la muestra de sonido obteniendo la salida que muestra la tabla XXII. Además hace un plot de las frecuencias en cada instante de tiempo.

TABLA XXII. FRECUENCIA EN UN INSTANTE DADO

Instante de muestra	Frecuencia
0.58050	298.09763
0.60372	599.19476
0.62694	1221.82190
0.65016	617.98389
0.67338	623.30933
0.69660	625.22308
0.71982	626.41998
0.74304	628.41095
0.76626	629.93896
0.78948	630.20062
0.81270	629.96625
0.83592	629.90228

Lo que se envía a la red neuronal son las frecuencias normalizadas con la probabilidad de ocurrencia de cada una, esto se detallara en el siguiente apartado.

Fase 3. Implementar una red neuronal de acuerdo a un modelo de aprendizaje supervisado para obtener melodías convincentes y realizar las pruebas del software desarrollado.

En esta fase vale recordar que se siguió el flujo de trabajo del Neural Network Toolbox, por lo que se indica la tarea del workflow y las tareas a las que corresponden en el anteproyecto.

1. Recolectar datos - Definir el tipo de red neuronal que satisfaga los requerimientos del proyecto a implementar.

Lo que se realizó es algo similar a *speech recognition* con redes neuronales, pero en este caso aplicado a guitarra en tiempo real; para lo cual se tomó como base la información recopilada de las escalas musicales (Ver apartado 1.3 de la Revisión Literaria).

Haciendo corresponder cada escala mayor (Ver 1.3.1 de la Revisión literaria) con su relativa escala menor (Ver 1.3.2 de la Revisión literaria) se tiene 12 clases que sirven como salidas o targets para la red neuronal, quedando de la siguiente manera:

1. Cmayor/Amenor
2. C#mayor/Bbmenor
3. Dmayor/Bmenor
4. Ebmayor/Cmenor
5. Emayor/C#menor
6. Fmayor/Dmenor
7. F#mayor/Ebmenor
8. Gmayor/Emenor
9. G#mayor/Fmenor
10. Amayor/F#menor
11. Bbmayor/Gmenor
12. Bmayor/G#menor

La máquina debe aprender la estructura musical de las escalas. Y ya que la red neuronal solo entiende valores numéricos como entradas de datos, se trabajó con los valores de las frecuencias en la escala temperada (Tabla XII)

El instrumento elegido para el trabajo es la guitarra, por tanto se toman como referencia las frecuencias iniciales de cada cuerda afinada de la manera estándar, es decir la afinación en E (tabla XXIII).

TABLA XXIII. FRECUENCIAS DE ENTRADA PARA LA RNA

Cuerda	Frecuencia
1ra (Mi)	329,63 [Hz]
2da (Si)	246,94 [Hz]
3ra (Sol)	196,00 [Hz]
4ta (Re)	146,83 [Hz]
5ta (La)	110,00 [Hz]
6ta (Mi)	82,41 [Hz]

2. Obtención del Dataset - Obtener las entradas de la red neuronal mediante el módulo de captura de sonido.

Esta es la parte más importante del proyecto así como también la más compleja ya que supone todo el aprendizaje de la red neuronal.

Para obtener el dataset se creó un vector con las características del sonido a analizar. Sabiendo que en teoría el sonido tiene varias características, nace la interrogante: ¿Cuál de ellas representa una determinada escala en la pieza de sonido?

Existen investigaciones acerca de este tema y se optó por una propuesta muy utilizada en el ámbito de percepción musical, el modelo Clave/Perfil de Carol Krumhansl y Mark Schmuckler.

El modelo postula que cada Clave/Escala tiene una clave de perfil: es decir un vector que representa la distribución óptima de tonos para esa Clave/Escala [26]. El perfil de clave para la escala mayor se muestra en la tabla XXIV.

TABLA XXIV. PERFIL DE CLAVE. ESCALA MAYOR

Nota	1	1#	2	2#	3	4	4#	5	5#	6	6#	7
Número												
Rango	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88

Una vez visto el perfil clave de la escala mayor se creó el perfil de clave para la escala de C Mayor (tabla XXV)

TABLA XXV. ESCALA C MAYOR

Nota Número	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Rango	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88

El perfil de clave para la escala menor se muestra en la tabla XXVI.

TABLA XXVI. PERFIL DE CLAVE. ESCALA MENOR

Nota Número	1	1#	2	2#	3	4	4#	5	5#	6	6#	7
Rango	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17

Una vez visto el perfil clave de la escala menor podemos crear el perfil de clave para la escala de C Menor (Tabla XXVII)

TABLA XXVII. PERFIL DE CLAVE. ESCALA C MENOR

Nota	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Rango	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17

El rango representa la importancia de cada nota en la escala.

Todos los perfiles de clave de la escala mayor se muestran en la tabla XXVIII y los de la escala menor en la tabla XXIX.

Cada clase dentro del dataset es un vector de 1x12 donde cada columna representa las 12 notas de la escala, y cada fila representa el perfil de clave en el que está el sonido. Cada valor numérico de este vector ha sido obtenido a través de una técnica de procesamiento digital de sonido llamada pitch-detection que calcula las frecuencias en cada período, sin embargo cabe destacar que la frecuencia obtenida no entra de forma bruta a la red neuronal es decir no entra una frecuencia como 55Hz, sino que entra normalizada de acuerdo a los perfiles de clave que se vio anteriormente.

Además en el análisis digital de sonido se extraen la ocurrencia de cada nota y la probabilidad de ocurrencia de cada frecuencia en el sonido analizado, lo que nos da un vector 1x12 de la siguiente forma:

**[0.3608 -0.2985 0.2124 0.3385 -0.7103 0.5578 -0.4265 0.3582
-0.1434 -0.2688 0.7062 -0.6865]**

Este vector es el que entra a la red neuronal y permite entrenar la red.

TABLA XXVIII. PERFILES DE LA ESCALA MAYOR

Notas												
Perfil de Clave	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88
C#	2.88	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29
D	2.29	2.88	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66
D#	3.66	2.29	2.88	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39
E	2.39	3.66	2.29	2.88	6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19
F	5.19	2.39	3.66	2.29	2.88	6.35	2.23	3.48	2.33	4.38	4.09	2.52
F#	2.52	5.19	2.39	3.66	2.29	2.88	6.35	2.23	3.48	2.33	4.38	4.09
G	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35	2.23	3.48	2.33	4.38
G#	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35	2.23	3.48	2.33
A	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35	2.23	3.48
A#	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35	2.23
B	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	6.35

TABLA XXIX. PERFILES DE CLAVE. ESCALA MENOR

Notas													
Perfil de Clave	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	
C	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17	
C#	3.17	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	
D	3.34	3.17	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	
D#	2.69	3.34	3.17	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	
E	3.98	2.69	3.34	3.17	6.33	2.68	3.52	5.38	2.6	3.53	2.54	4.75	
F	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	5.38	2.6	3.53	2.54	
F#	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	5.38	2.6	3.53	
G	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	5.38	2.6	
G#	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	5.38	
A	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	
A#	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	
B	2.68	3.52	5.38	2.6	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	

3. Crear la red Neuronal.

Este paso implica dos tareas definidas en el anteproyecto:

- Definir las salidas de la red neuronal, funciones de activación y funciones de aprendizaje.
- Implementar la red neuronal en la herramienta que mejor se adapte a las necesidades del proyecto.

3.1. Arquitectura de la Red Neuronal

Una vez identificado el problema como de reconocimiento o clasificación de patrones y después de la recopilación teórica sobre redes neuronales, Se concluyó que la arquitectura que mejor se adapta a este tipo de problemas es la del Perceptron Multicapa (Multi-Layered Perceptron) MLP por sus siglas en ingles. Al elegir esta arquitectura generalmente se tiene Entradas, Capa Oculta, Capa de Salida (Figura 9).

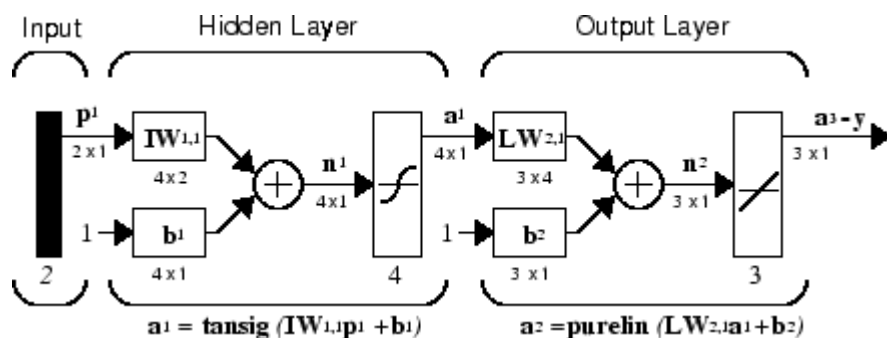


Figura 9. Multi-Layered Perceptron

De forma general la red cuenta con 1 capa oculta y 1 de salida. El vector de entrada consta de 12 elementos (inputs), cada input está asociada a la nota que representa en la escala cromática.

Después de realizar algunas pruebas se determinó que el número de neuronas óptimo en la capa oculta es 28 por resultados de aprendizaje de acuerdo al método prueba error, siendo 28 el número de neuronas que ofreció mejores resultados, el número de

neuronas de la capa de salida va de acuerdo a las entradas que se tiene, por lo que en este caso son 12. En la figura 10 se puede observar el diseño de la red neuronal.

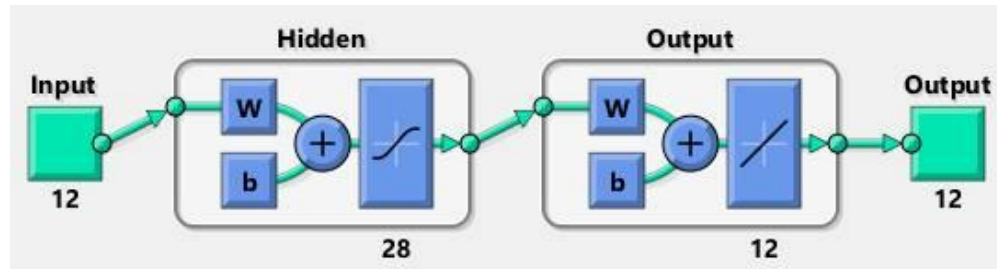


Figura 10. Diseño de la Red Neuronal

3.2. Funciones de Transferencia (Transfer Functions)

Una vez elegida la arquitectura de la red neuronal, es necesario elegir las funciones de transferencia que servirán para el aprendizaje de la red neuronal.

• Capa Oculta

Generalmente en la arquitectura MLP se usa en la capa oculta la función de transferencia logarítmica sigmoideal o logsig, aunque también es usada la función de transferencia tangencial sigmoideal o tansig.

Se prefirió la función Logsig (Figura 11), debido a que su saturación está comprendida entre 0 y 1, y al momento de obtener las salidas se trabaja con este rango.

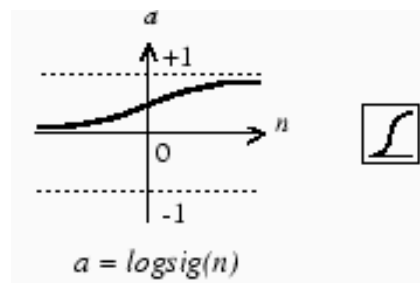


Figura 11. Función LOGSIG

- **Capa de Salida**

En la capa de salida generalmente se usa la función de transferencia Purelin (Figura 12), que igual es una función de reconocimiento de patrones.

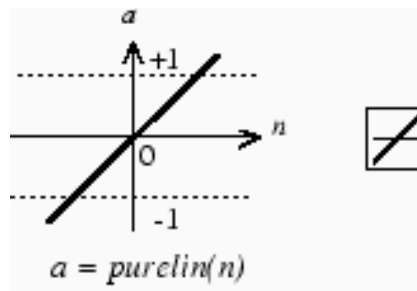


Figura 12. Función PURELIN

3.3. Configurar la Red Neuronal

3.3.1. Configurar Entradas y Salidas

Aquí es donde se centra la resolución del problema ya que una vez obtenido el dataset se debe generar las entradas de la red neuronal, y a su vez determinar las salidas que se desea obtener.

- **Entradas**

Las entradas de la red neuronal están explicadas en la obtención del dataset, pero de forma general el vector de entrada es el siguiente:

**[0.3608 -0.2985 0.2124 0.3385 -0.7103 0.5578 -0.4265 0.3582
-0.1434 -0.2688 0.7062 -0.6865]**

- **Salidas**

Las salidas que se desea obtener están dadas en formato 0 1, donde el 1 representa la escala en que se está tocando. Por ejemplo si se obtiene una salida como la siguiente, el sonido está en la escala de C mayor / A menor

[1 0 0 0 0 0 0 0 0 0]

3.3.2. Inicializar Pesos y Bias

Antes de entrenar la red es necesario inicializar los pesos y el bias de la red, esta sección es automática al crear la red en Matlab con el comando **configure**, sin embargo si se quiere entrenar de nuevo o con pesos nuevos, se lo puede hacer mediante el comando **init** del neural network toolbox de Matlab.

4. Entrenar la Red Neuronal.

Es la parte más delicada dentro del flujo de trabajo para la creación de la red neuronal debido a que los algoritmos de aprendizaje pueden generar diferentes resultados.

Una vez inicializados los pesos y el bias, la red neuronal esta lista para ser entrenada, una red neuronal MLP puede ser entrenada para funciones de aproximación (regresión no lineal) o para reconocimiento de patrones, como nuestro problema es de reconocimiento de patrones los algoritmos usados están orientados a esta tarea.

El entrenamiento requiere un dataset que contenga entradas (inputs) y las salidas deseadas (targets). Así mismo usa los pesos y el bias de la red para ajustar el rendimiento de la red que viene dado por la siguiente formula:

$$F = mse = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2$$

Esta es la fórmula del error cuadrático medio (mean square error), utilizada por default en Matlab para las redes MLP.

Para optimizar el aprendizaje de la red se usó el algoritmo **Backpropagation**, que consiste en la realización de cálculos hacia atrás a través de la red.

Existen numerosas variaciones del algoritmo Backpropagation y estas se encuentran en el Neural Network toolbox de Matlab (tabla XXX).

TABLA XXX. ALGORITMOS DE APRENDIZAJE BACKPROPAGATION

Function	Algorithm
trainlm	Levenberg-Marquardt
trainbr	Bayesian Regularization
trainbfg	BFGS Quasi-Newton
trainrp	Resilient Backpropagation
trainscg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell/Beale Restarts
traingcf	Fletcher-Powell Conjugate Gradient
traingcp	Polak-Ribière Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Gradient Descent
traingdm	Gradient Descent with Momentum
traingd	Gradient Descent

De toda esta lista de funciones de aprendizaje se seleccionó el algoritmo **Variable Learning Rate Gradient Descent (traingdx)**, ya que permite controlar la tasa de aprendizaje (Learning Rate) y el Momentum de la red para optimizar el aprendizaje, además posee aprendizaje adaptativo que hace un poco autónoma la red, ajustando la tasa de aprendizaje de acuerdo al error. La configuración óptima para el aprendizaje en nuestro problema es la siguiente:

Algoritmo Backpropagation - Variable Learning Rate Gradient Descent

Learning Rate = 0.02

Momentum = 0.8

Los resultados obtenidos se muestran en la tabla XXXI.

TABLA XXXI. RESULTADOS OBTENIDOS

Learning Rate	Momentum	Training Classification Rate %	Testing Classification Rate %
0.02	0.8	79.4	20.6

5. Validar la Red Neuronal

Una vez terminada la fase de aprendizaje es necesario validar los resultados obtenidos para lo cual existen las siguientes etapas:

- **Revisar los datos del aprendizaje:**

Los datos que se obtuvieron de la red son los siguientes:

```
tr =
trainFcn: 'traingdx'
trainParam: [1x1 struct]
performFcn: 'mse'
performParam: [1x1 struct]
derivFcn: 'defaultderiv'
divideFcn: 'divideblock'
divideMode: 'sample'
divideParam: [1x1 struct]
trainInd: [1x126 double]
valInd: [1x27 double]
testInd: [1x27 double]
stop: 'Validation stop.'
num_epochs: 154
trainMask: {[12x180 double]}
valMask: {[12x180 double]}
```

testMask: {[12x180 double]}
best_epoch: 148
goal: 0
states: {'epoch' 'time' 'perf' 'vperf' 'tperf' 'gradient'
'val_fail' 'lr'}
epoch: [1x155 double]
time: [1x155 double]
perf: [1x155 double]
vperf: [1x155 double]
tperf: [1x155 double]
gradient: [1x155 double]
val_fail: [1x155 double]
lr: [1x155 double]
best_perf: 0.0245
best_vperf: 0.0573
best_tperf: 0.0778

- Validar el performance de la red:

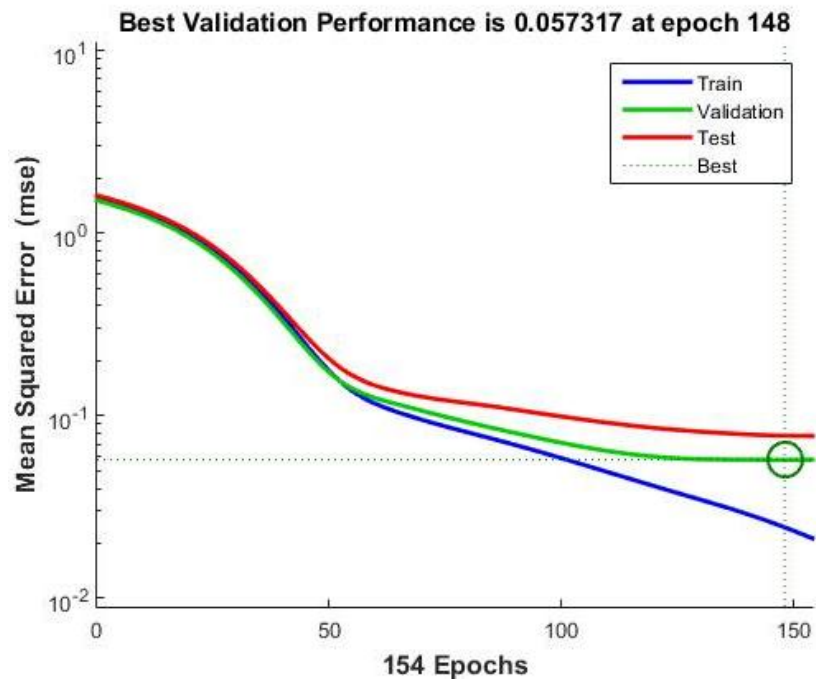


Figura 13. Performance de la Red

El grafico del performance de la red (Figura 13) nos muestra que no existe mayor problema con el entrenamiento ya que las curvas son similares, si la curva del test incrementara más pronto que la curva de validación eso significaría que existe overfitting, pero en este caso las curvas son similares por lo que el performance es bueno.

- Crear el plot de Regresion.

El plot (Figura 14) muestra la relación entre las salidas de la red y los targets, si el entrenamiento fue perfecto las salidas y los targets deberían ser iguales pero en la práctica nunca va a ser perfecto.

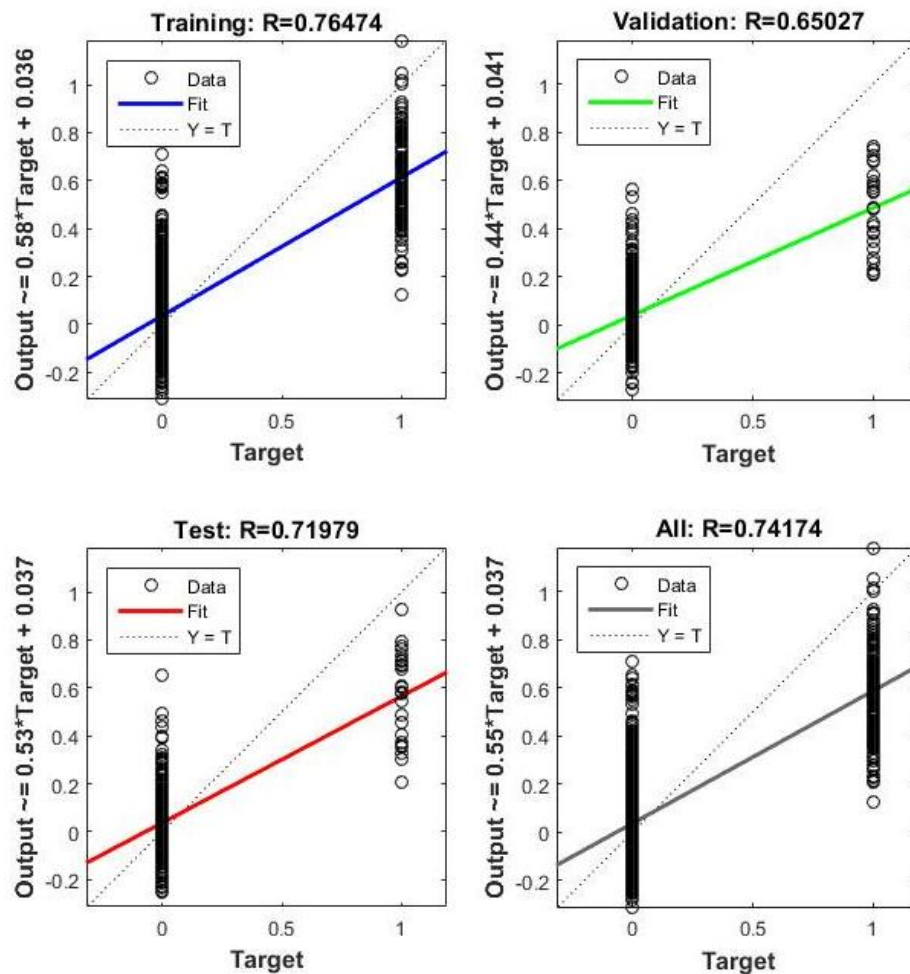


Figura 14. Plot de Regresión

El grafico nos muestra que no existe un buen ajuste (Fit) debido a que el problema a resolver no es de ajuste sino de reconocimiento de patrones. Por lo que podemos ver las clases delimitadas a las salidas que queremos, en este caso 0 1.

• Generar la matriz de confusión:

En la figura 15 podemos ver que cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Una ventaja de la matriz de confusión es que facilita ver si la red neuronal está confundiendo clases.

Confusion Matrix												
Output Class	1	2	3	4	5	6	7	8	9	10	11	12
	13 7.2%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	1 0.6%	0 0.0%	1 0.6%	1 0.6%	0 0.0%	0 0.0%	0 0.0%
	0 0.0%	13 7.2%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	2 1.1%	0 0.0%	1 0.6%	1 0.6%	0 0.0%	0 0.0%
	0 0.0%	0 0.0%	13 7.2%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	2 1.1%	0 0.0%	1 0.6%	1 0.6%	0 0.0%
	0 0.0%	0 0.0%	0 0.0%	12 6.7%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	1 0.6%	0 0.0%	1 0.6%	0 0.0%
	1 0.6%	0 0.0%	0 0.0%	0 0.0%	12 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 1.1%	0 0.0%	1 0.6%
	0 0.0%	1 0.6%	0 0.0%	0 0.0%	0 0.0%	12 6.7%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	1 0.6%	0 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 6.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.6%
	1 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	12 6.7%	0 0.0%	0 0.0%	1 0.6%	0 0.0%
	0 0.0%	1 0.6%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	0 0.0%	0 0.0%	11 6.1%	0 0.0%	0 0.0%	1 0.6%
	0 0.0%	0 0.0%	2 1.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 6.1%	0 0.0%	0 0.0%
	0 0.0%	0 0.0%	0 0.0%	2 1.1%	0 0.0%	1 0.6%	1 0.6%	0 0.0%	0 0.0%	0 0.0%	11 6.1%	0 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	12 6.7%
86.7%86.7%86.7%80.0%80.0%80.0%73.3%80.0%73.3%73.3%73.3%80.0%												
13.3%13.3%13.3%20.0%20.0%20.0%26.7%20.0%26.7%26.7%26.7%20.0%												
Target Class												
1	2	3	4	5	6	7	8	9	10	11	12	

$input = [0.3608 \quad -0.2985 \quad 0.2124 \quad 0.3385 \quad -0.7103 \quad 0.5578 \quad -0.4265 \quad 0.3582 \quad -0.1434 \quad -0.2688 \quad 0.7062 \quad -0.6865]$

Teniendo el vector podemos enviarlo a la red mediante el comando ***sim(input)*** o también directamente ***net(input)***, lo que nos arroja una matriz de información de 12x12 que representa cada una de las notas de la escala cromática, realmente es un poco confuso entender estos valores, sin embargo lo que se hace es buscar el mayor valor dentro de la matriz y ver en qué posición esta, en este caso el mayor valor es:

1.1223

Y se encuentra en la posición: [10] [11]

Extraemos la fila 10

[0.5316 -0.1453 0.2536 0.4875 -0.0501 0.8823 -0.1845 0.5265 -0.0316 -0.1259 1.1223 -0.0646]

Ya que cada posición representa la escala cromática hacemos el conteo para identificar la nota a la que corresponde y en este caso corresponde a A# (La#) y para la identificación de la escala se sigue el patrón: Escala Mayor/ Escala Menor, En este caso se obtiene que la escala es:

A#Mayor/GMenor, coincidiendo con la escala de la pieza musical.

MATRIZ DE INFORMACIÓN

ans = [

0.1723	-0.1420	0.0294	0.1531	-0.5281	0.2481	-0.2773	0.1702	-0.0782	-0.1213	0.2426	-0.5209
0.0921	0.2909	0.0170	0.0828	1.0339	0.0930	0.5439	0.0911	0.0968	0.2437	0.0395	1.0098
0.1900	0.5139	0.1728	0.1817	0.9369	0.3639	0.7203	0.1889	0.3335	0.4715	0.5752	0.9391
-0.2701	0.0012	-0.0534	-0.2417	0.2645	-0.3574	0.0322	-0.2670	0.0520	0.0059	-0.2931	0.2460
-0.3063	0.1239	-0.1650	-0.2880	0.2303	-0.4314	0.1004	-0.3042	0.1387	0.1311	-0.5009	0.2118
0.7477	-0.1514	0.4419	0.7087	-0.5266	0.8916	-0.2642	0.7434	-0.0268	-0.1269	0.8474	-0.5063
0.6044	0.0018	0.3648	0.5691	-0.2665	0.8515	-0.0857	0.6004	0.0822	0.0192	1.0005	-0.2550
-0.0535	-0.1504	-0.0017	-0.0498	-0.5534	-0.0038	-0.3243	-0.0532	0.0159	-0.1118	0.0966	-0.5454
0.7748	-0.0303	0.5610	0.7473	0.3721	0.9512	0.0587	0.7717	0.0176	-0.0350	1.0347	0.3517
0.5316	-0.1453	0.2536	0.4875	-0.0501	0.8823	-0.1845	0.5265	-0.0316	-0.1259	1.1223	-0.0646
0.1678	0.1286	0.1105	0.1625	0.5161	0.1487	0.2552	0.1673	0.0344	0.1044	0.0765	0.5001
0.2844	-0.0071	0.1339	0.2669	0.3748	0.3177	0.1175	0.2825	-0.0769	-0.0286	0.2519	0.3600

]

Fase 4. Realizar las pruebas del software desarrollado para comprobar el correcto funcionamiento del mismo.

Una vez terminadas las dos partes correspondientes a Captura de Audio y Red Neuronal se unen en el software final para lo cual se realizaron las pruebas de funcionalidad de interfaz y pruebas unitarias a través del Unit Test Framerwork de Matlab.

1. Pruebas de interfaz

• Interfaz principal

En esta interfaz tenemos el menú principal que consta de 2 submenús: *Archivo* y *Ayuda*.

El submenú archivo a su vez consta de 3 submenús: *Grabar Audio*, *Analizar Audio* y *Salir*, (figura 16). Este submenú contiene las partes principales del software.

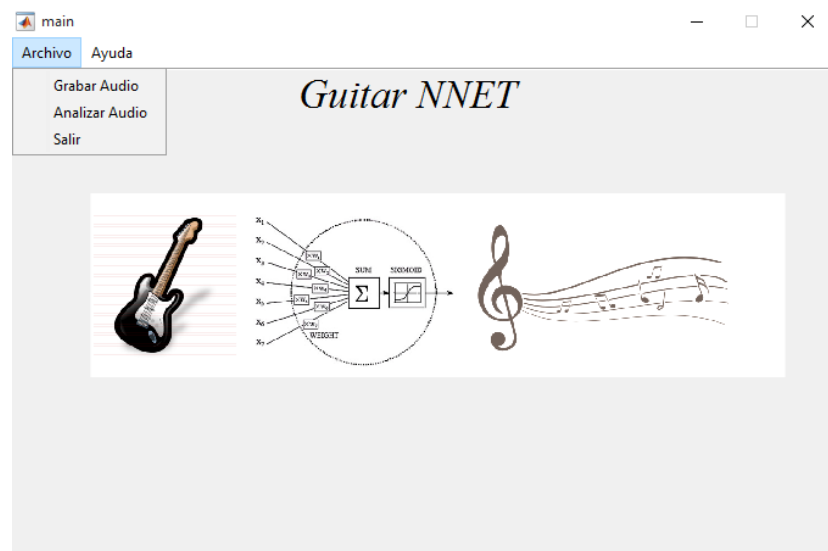


Figura 16. Interfaz Principal

- **Grabar Audio:** sirve para abrir la interfaz de captura de audio en tiempo real a través de la tarjeta de sonido del pc (figura 17) Se pueden observar 3 plots, el primero muestra el nivel del audio capturado y las notas que se están tocando identificadas de acuerdo al sistema americano o inglés, este plot se puede cambiar para ver las frecuencias de cada nota, para esto en el apartado **mode** seleccionamos **freq** y se cambiará a las frecuencias que se toca en tiempo real.

El siguiente plot nos muestra el nivel en decibeles así como la frecuencia en kHz de la entrada de sonido.

El siguiente plot nos muestra la gráfica del sonido en el dominio de **frecuencia / tiempo** donde el tiempo es igual a 1 segundo.

Los botones restantes son para: iniciar, detener, guardar y analizar el sonido capturado.



Figura 17. Interfaz de Grabación de Audio

- **Analizar Sonido:** Abre la interfaz que hace uso de la red neuronal para hacer el cálculo de la escala de la pieza de sonido grabada. En la figura 18 se puede

observar un plot general donde se hace el análisis de frecuencias obtenidas en el sonido capturado, este es una estimación de la clave o tono en el que se está tocando. Se puede observar los botones para cargar, iniciar, pausar, detener y analizar la pieza de sonido.

Una vez se da clic en analizar sonido se puede ver el tono en que se está tocando y las notas que están dentro de ese tono. En la parte inferior derecha se muestra una lista con las estimaciones de notas erróneas dentro de la escala definida.

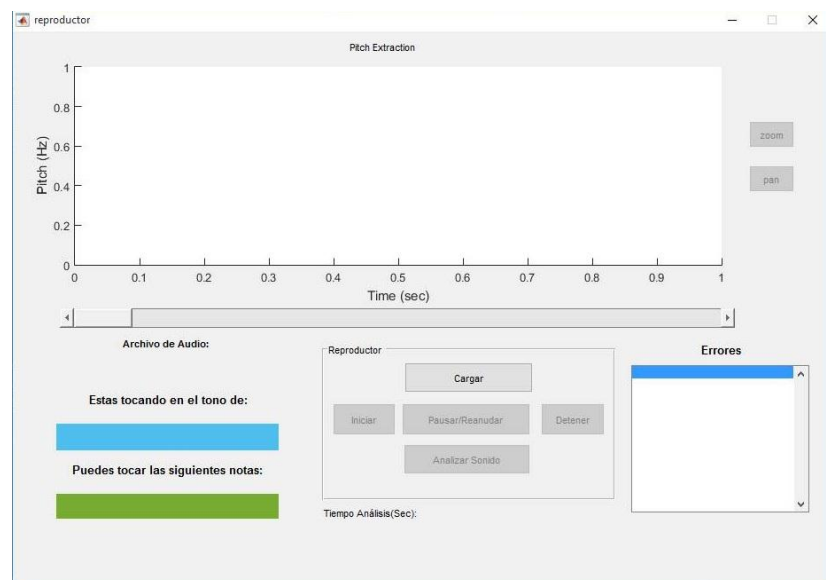


Figura 18. Interfaz de Análisis de Sonido

- El submenú **ayuda** contiene información general sobre el programa e información acerca del mismo (figura 19).

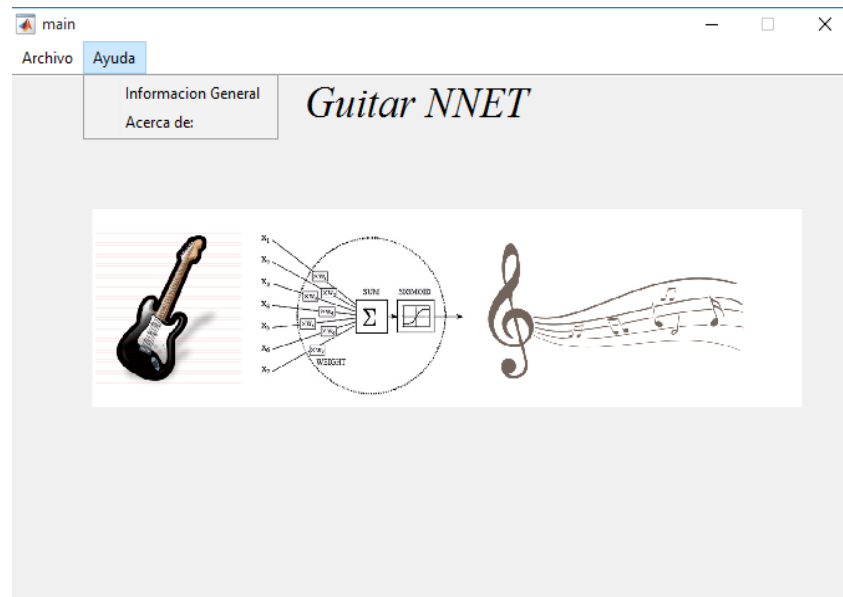


Figura 19. Ingreso al Menú Ayuda

2. Pruebas Unitarias

Las pruebas unitarias se realizaron mediante el Unit Test Framework de Matlab para comprobar el correcto funcionamiento de las funciones y la veracidad de los cálculos necesarios dentro de las interfaces, las funciones a testear son las siguientes:

```
function [t,freq]= pitchExtraction(file)
function [coeffs,t,fq,fqn] = bsefreqtry(file)
function [terror]=errorfreq(freq,error)
```

La función ***pitchExtraction*** obtiene las frecuencias reales dentro del archivo de sonido y devuelve el tiempo en que sucede cada frecuencia y la frecuencia real del sonido, esta a su vez es usada dentro de función ***bsefreqtry***.

La función ***bsefreqtry*** normaliza las frecuencias a la octava 1 dado un archivo de sonido y devuelve los coeficientes para la red neuronal, el tiempo en que sucede cada frecuencia, la frecuencia real, y la frecuencia normalizada.

La función ***errorfreq*** obtiene las frecuencias erróneas dentro de la pieza musical, recibe como parámetros las frecuencias normalizadas en la octava 1 y un arreglo con las frecuencias erróneas dentro de una escala musical, y devuelve la posición de las mismas en todo el arreglo de frecuencias.

Para realizar las pruebas se programó el siguiente código:

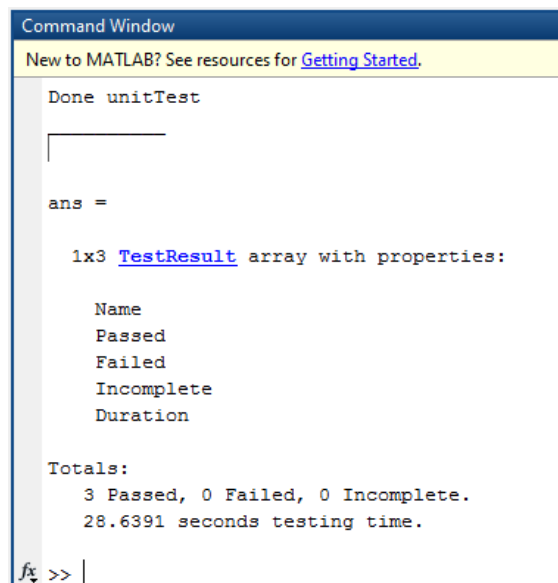
```
function tests = unitTest
    tests = functiontests(localfunctions);
end

function pitchExtractionTest
    pitchExtraction('Bethoven C#m.wav')
end

function bsefreqtryTest()
    bsefreqtry('Bethoven C#m.wav') end

function errorfreqTest()
    [entrada,t,freq,freqn] = bsefreqtry('Bethoven C#m.wav')
    errorf=[34.6500 38.8900 41.2000 46.2500 55.0000];
    errorfreq(freqn,errorf)
end
```

La respuesta obtenida una vez realizado el testeo unitario se ve en la figura 20.



```
Command Window
New to MATLAB? See resources for Getting Started.

Done unitTest

ans =

1x3 TestResult array with properties:
    Name
    Passed
    Failed
    Incomplete
    Duration

Totals:
    3 Passed, 0 Failed, 0 Incomplete.
    28.6391 seconds testing time.

fx >> |
```

Figura 20. Respuesta al Testeo Unitario

g. Discusión

1. Desarrollo de la propuesta alternativa

El desarrollo de la propuesta alternativa se basa en la realización y cumplimiento de los objetivos planteados.

a. Analizar las alternativas tecnológicas y modelos de aprendizaje supervisado en redes neuronales, para la implementación del proyecto de manera óptima.

Para el cumplimiento de este objetivo, se realizó la búsqueda, selección y análisis de casos de éxito en los 5 últimos años sobre la implementación de técnicas de inteligencia artificial para la composición musical (Ver Tabla XIII); la búsqueda se efectuó en universidades a nivel internacional y en centros de investigación científica.

Una vez concluido el análisis de los casos de éxito, se hizo un análisis exhaustivo para la elección de las herramientas de software que más se adapten a los requerimientos existentes, los puntos clave para la elección correcta de las herramientas son: arquitecturas de red neuronal soportadas, requisitos mínimos de hardware y software, soporte (Ver Tabla XX y Tabla XXVII).

b. Desarrollar un módulo para la captura y procesamiento de sonido en tiempo real que sirva como entrada de datos de la red neural.

Para el desarrollo de este objetivo se utilizó Matlab como lenguaje de implementación debido a las características que posee para procesar audio (Ver TABLA XX), así mismo para el desarrollo de la Interfaz Gráfica se usó el GUI de Matlab que facilita la creación de la misma.

Para obtener los datos de entrada de la red neuronal se decidió usar como formato de audio el formato wav, ya que permite la grabación en canales mono y

estéreo, además de aquí se obtiene los datos en bruto por lo que se puede sacar las frecuencias mucho más fácil.

c. Implementar una red neuronal de acuerdo a un modelo de aprendizaje supervisado para obtener melodías estructuralmente correctas.

Para dar cumplimiento a este objetivo se hizo un análisis de herramientas para la creación de redes neuronales (Ver TABLA XXVIII) siendo elegido Matlab para tal efecto, ya que presenta mejores características.

La implementación de la red neuronal se llevó a cabo a través del uso de técnicas de Procesamiento Digital de Señales (DSP) y Recuperación de Información Musical (MIR), haciendo uso de librerías para la extracción de la frecuencia fundamental.

d. Realizar las pruebas del software desarrollado para comprobar el correcto funcionamiento del mismo.

Para cumplir con este objetivo se hizo pruebas unitarias a través del Unit Test Framework de Matlab (Ver Sección Resultados), con lo que se comprobó la correcta ejecución del software desarrollado.

Para comprobar que el software cumple con el objetivo de reconocer la melodía que se está tocando, se recurrió a un docente de música para que comprobara los datos obtenidos a través del software, una vez hecho esto emitió una certificación del correcto funcionamiento del software (ver Anexo 4).

2. Valoración técnica, económica, ambiental

El Trabajo de Titulación denominado “Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada”, se considera desde el punto de vista técnico como un trabajo factible, puesto que servirá como alternativa para las futuras investigaciones académicas-científicas que se desarrollen en la CIS.

Tecnológicamente hablando el desarrollo y puesta en marcha del software no implica el uso de equipos excesivamente costosos, para el procesamiento de sonido se puede utilizar un equipo que posea una tarjeta de sonido con entradas de sonido, de tal manera que se pueda conectar un micrófono para grabar el sonido, o se pueda conectar directamente la guitarra al ordenador mediante un cable auxiliar o adaptadores adecuados para este caso.

La valoración económica del proyecto tiene su base en que el desarrollo del mismo se ajusta a los intereses de la investigación, puesto que el software utilizado tiene licencia privativa, mientras que el hardware utilizado son los equipos tecnológicos con los que cuentan los investigadores.

El talento humano que participó en el desarrollo del Trabajo de Titulación, está conformado principalmente por los investigadores quienes fuimos los encargados de desarrollar a cabalidad el proyecto, el docente de la materia de anteproyecto de tesis quien fue el guía para la elaboración del anteproyecto y la memoria final, y el director del trabajo de titulación. La Tabla XXXII detalla el tiempo y costo asignado a los investigadores, docente de anteproyecto de tesis, y director del proyecto, responsables de la culminación exitosa del mismo.

TABLA XXXII. RECURSOS HUMANOS.

DESCRIPCIÓN	CANTIDAD	COSTO/HORA	HORAS	COSTE TOTAL
Investigadores	2	\$ 4.00	400	\$ 1600.00
Docente Anteproyecto de Tesis	1	\$ 00.00	64	\$ 00.00
Director de Tesis	1	\$ 00.00	200	\$ 00.00
TOTAL				\$ 1600.00

La Tabla XXXIII presenta una descripción detallada de los recursos materiales que fueron necesarios para la presentación de los avances y el informe final del proyecto.

TABLA XXXIII. RECURSOS MATERIALES

DESCRIPCIÓN	CANTIDAD	V. UNITARIO	V. TOTAL
Resma de papel A4	1	\$ 6.00	\$ 6.00
Cartuchos de tinta	3	\$ 20.00	\$ 60.00
Empastado	3	\$ 10.00	\$ 30.00
Anillados	3	\$ 2.50	\$ 7.50
CD's	3	\$ 1.00	\$ 3.00
TOTAL			\$ 106.50

La Tabla XXXIV detalla los recursos hardware utilizados en el desarrollo del TT; que comprende dos computadores portátiles usados para la implementación virtual del Clúster de alto rendimiento, para luego implementarlo en el laboratorio de la CIS, la redacción de los avances e informe final del mismo.

TABLA XXXIV. RECURSOS HARDWARE

DESCRIPCIÓN	COSTO	DEPRECIACIÓN (3 años)	T. UTILIZACIÓN (horas)	TOTAL
Portátil TOSHIBA SATELLITE	\$ 850.00	\$ 161	400	\$ 214.67
Portátil DELL INSPIRION	\$ 900.00	\$ 214	400	\$ 285.33
Guitarra	\$ 250.00	\$59	400	\$ 116.50
Impresora	\$ 80.00	\$ 16	400	\$ 21.23
TOTAL				\$ 637.73

La Tabla XXXV muestra las herramientas de software utilizadas en la implementación del proyecto.

TABLA XXXV. RECUSROS SOFTWARE

DESCRIPCIÓN	VALOR
Windows 10	\$ 119.00
Matlab Student Version	\$ 99.00
TOTAL	\$ 218.00

En la Tabla XXXV se aprecia la suma parcial de los recursos técnicos y tecnológicos, usados luego en la suma total de recursos usados.

TABLA XXXVI. RECURSOS TÉCNICOS Y TECNOLÓGICOS

DESCRIPCIÓN	VALOR TOTAL
Recursos hardware	\$ 637.73
Recursos software	\$ 218.00
SUBTOTAL	\$ 855.73

La Tabla XXXVI Aproximación del costo real del TT), ilustra la suma total de todos los recursos: humanos, materiales, técnicos y tecnológicos usados en el TT, que nos brinda una aproximación real del coste del proyecto.

El Trabajo de Titulación se considera factible en el aspecto ambiental puesto que el desarrollo del software solo usa computadores portátiles por lo que el daño que se genera en el ambiente es mínimo

TABLA XXXVII. APROXIMACIÓN DEL COSTO REAL DEL TT.

DESCRIPCIÓN	VALOR TOTAL
Recursos humanos	\$ 1600.00
Recursos materiales	\$ 106.50
Recursos técnicos y tecnológicos	\$ 855.73
Subtotal	\$ 2562.23
Imprevistos (10%)	\$ 256.22
TOTAL	\$ 2818.45

h. Conclusiones

- El uso de redes neuronales para el reconocimiento de escalas musicales de tal manera que ayude al proceso creativo del músico es factible, ya que este es un problema de reconocimiento de patrones.
- La arquitectura de red neuronal que se adaptó mejor a la solución del problema es la arquitectura MLP (Multi Layered Perceptron) con aprendizaje backpropagation, ya que mediante esta arquitectura se puede controlar la tasa de aprendizaje y el momentum de la red para obtener mejores resultados.
- Matlab ofrece mejores herramientas para la solución de problemas que involucren inteligencia artificial y DSP, ya que tiene mayor cantidad de algoritmos tanto de DSP como de inteligencia artificial.
- Para el análisis de una melodía el formato wave es muy conveniente, gracias a que a pesar de que es pesado, al almacenar conserva todas las características originales del sonido, sin eliminar las frecuencias de sonido que el oído humano no percibe, como sucede en formatos como el mp3.
- Un algoritmo eficiente para calcular los coeficientes de la Transformada Discreta de Fourier (DFT) es la transformada rápida de Fourier (FFT) debido a que minimiza la cantidad de cálculos matemáticos y el número de ciclos de computación de la DFT.

i. Recomendaciones

- Para investigaciones de este tipo es necesario tener conocimientos previos sobre DSP (Digital Signal Processing) para poder trabajar con frecuencias y generar los dataset para el entrenamiento de la red, además para entender estos datasets también es conveniente tener conocimientos sobre MIR (Music Information Retrieval).
- Aunque es factible el uso de redes neuronales para resolver problemas de este tipo se podría realizar investigaciones aplicando otras técnicas como los modelo ocultos de Markov que podrían ofrecer mejores resultados.
- Para mejorar el rendimiento de la aplicación se podría trabajar con procesamiento en paralelo creando distintos hilos de ejecución para la ejecución y procesamiento del audio, ya que la captura de sonido en tiempo real consume bastantes recursos.
- Para mejorar el aprendizaje de la red neuronal se puede crear un dataset de mayores elementos debido a que las redes neuronales supervisadas trabajan o aprenden mejor de acuerdo a la cantidad de datos previos que se tengan para el aprendizaje.
- Al momento de programar GUI's con Matlab no se tiene mucho control sobre la interfaz, por lo que se podría probar con la ejecución de java dentro de Matlab para poder desarrollar GUI's visualmente más agradables.
- Se podría realizar estudios alternos sobre el manejo de señales a través de técnicas DSP, no solo en el ámbito musical sino en otras áreas de interés como: procesamiento de imágenes, medicina, telecomunicaciones, speech recognition, entre otras.
- Para complementar este trabajo se puede realizar un módulo adicional que permita graficar las notas y los errores en el pentagrama musical de tal manera

que los resultados sean más fácil de leerlos en especial para los estudiantes de música.

- Se puede implementar este software en el conservatorio Salvador Bustamante Celi de la ciudad de Loja ya que los docentes manifestaron interés en la aplicación del mismo en las aulas donde existan alumnos principiantes.

j. Bibliografía

- [1] E. Herrera, Teoría Musical y Armonía Moderna, vol. 43, Barcelona, España: Antoni Bosch, 1990.
- [2] E. Gómez y I. Cuenca, Manual Técnico de Sonido, Madrid, España.: Ediciones Paraninfo SA, 2011.
- [3] C. Giancoli, Física: Principios con Aplicaciones, México DF, México: Pearson Educación, 2006.
- [4] M. Castro, Música para todos: Una Introducción al Estudio de la Música, San José, Costa Rica: Editorial de la Universidad de Costa Rica, 2003.
- [5] M. Hewitt, Music Theory for Computer Musicians, Boston, USA: Course Technology, 2008.
- [6] C. Schmidt-Jones y R. Jones, Understanding Basic Music Theory, Houston, Texas: Rice University, 2007.
- [7] «Physics of Musical Notes, Frequencies for equal-tempered scale, A4 = 440 Hz,» [En línea]. Available: <http://www.phy.mtu.edu/~suits/notefreqs.html>. [Último acceso: 6 5 2015].
- [8] A. Patel, Music, Language and the Brain, Oxfordshire, Inglaterra: Oxford University Press, 2008.
- [9] A. García, Inteligencia Artificial. Fundamentos, práctica y aplicaciones, Madrid, España: RC Libros, 2012.
- [10] R. Flores y J. Fernandez, Las Redes Neuronales Artificiales: : Fundamentos Teóricos y Aplicaciones Prácticas, La Coruña, España: NetBiblo S.L., 2008.
- [11] P. Isasi y I. Galván, Redes De Neuronas Artificiales: Un Enfoque Práctico, Madrid, España: Pearson Educación, 2004.
- [12] J. Chitode, Signals and Systems, Pune, India: echnical Publications Pune, 1997.
- [13] J. Mariño, F. Vallverdú, J. Rodríguez y A. Moreno, Tratamiento Digital de una Señal: Una introducción experimental, Barcelona, España: Ediciones UPC, 1999.
- [14] A. N. Kani, Signals and Systems, New Delhi, India: Tata McGraw Hill Education Private Limited, 2010, pp. 9.1 - 9.65.
- [15] Wikipedia, «Wikipedia,» Wikipedia , [En línea]. Available: https://es.wikipedia.org/wiki/Waveform_Audio_Format. [Último acceso: 19 05 2015].
- [16] J. Carvajal y F. Giraldo, «Composición utilizando algoritmos genéticos,» Fundación Universitaria San Martín, Bogotá, Colombia, 2012.
- [17] A. Sarroff y . M. Casey, «Musical audio Synthesis using Autoencoding Neural Nets,» Computer Science Dartmouth College, Hannover, USA, 2014.
- [18] L. Mazurowshi, «Computer Models for Algorithmic Music Composition,» West Pomeranian University of Tecnology- Szczecin, Szczecin, Poland, 2012 .

- [19] J. Leben, «Artificial Neural Networks in Musical Performance,» Institute of Sonology Royal Conservatory, The Hague, 2012.
- [20] M. Delgado, W. Fajardo y M. Molina-Solana, «Inmamusys: Sistema Inteligente Multiagente,» Department of Computer Science and Artificial Intelligence, Universidad de Granada, Granada, España, 2008.
- [21] R. Fiebrink y P. Cook, «THE WEKINATOR: A SYSTEM FOR REAL-TIME, INTERACTIVE MACHINE LEARNING IN MUSIC,» Department of Computer Science, New Jersey, USA, 2011.
- [22] Matlab, «Matlab: Documentation,» Matlab, [En línea]. Available: <http://www.mathworks.com/help/matlab>. [Último acceso: 19 05 2015].
- [23] U. d. Waikato, «Weka Documentation,» Universidad de Waikato, [En línea]. Available: <http://www.cs.waikato.ac.nz/ml/weka/documentation.html>. [Último acceso: 19 05 2015].
- [24] Oracle, «The Java Tutorials,» Oracle, [En línea]. Available: <https://docs.oracle.com/javase/tutorial/sound/sampled-overview.html>. [Último acceso: 19 05 2015].
- [25] E. Soria, M. Martínez, J. Francés y G. Camps, Tratamiento Digital de Señales, Madrid, España: Pearson Educación S.A., 2003.
- [26] C. Krumhansl, Cognitive foundations of musical pitch, New York, USA: Oxford University Press, 1990.
- [27] A. Jaramillo, Acústica, la Ciencia del Sonido, Medellín Colombia: Fondo Editorial ITM, 2007.

k. Anexos

Anexo 1. Dataset para el entrenamiento de la red neuronal

DATASET PARA EL ENTRENAMIENTO DE LA RED NEURONAL

Para el entrenamiento de la red neuronal se tomaron 180 melodías de música clásica, debido a que dichas piezas poseen una estructura musical correcta de acuerdo a las reglas de la música occidental.

C major / A minor	C# major / A # minor
Beethoven - Symphony No 1 in C major, Op 21	Johann Sebastian Bach WTC1 No. 3 in C sharp major BWV 848
Mozart - Symphony No. 36 in C, K. 425	Johann Sebastian Bach Prelude and Fugue in C sharp major BWV 848
Mozart - Symphony No. 41 in C, K. 551	Joseph Haydn - Symphony No. 39
Beethoven- Für Elise	Wolfgang Amadeus Mozart - Symphony No. 25
Fryderyk Chopin - Mazurek	Wolfgang Amadeus Mozart - Symphony No. 40
Johannes Brahms - Sonata N° 1 in C Op. 1	Carl Nielsen - Symphony No.1
Fryderyk Chopin, Mazurek a-moll Op. 68 nr 2	Dmitri Shostakovich - Symphony No. 11
Joseph Haydn - Symphony No. 90 in C major	Johannes Brahms- Waltz No.6
Michael Haydn Symphony No.18	Beethoven- Waltz for Piano in C sharp major
Johann Michael Haydn - Symphony No. 39	Johann Christian Heinrich Rinck - Prelude no. 16 in A sharp minor
Sergei Prokofiev - Symphony no. 4, op.47	Charles H. V. Alkan Andante - Contrapunctus in C sharp major
Franz Schubert - Symphony No.6	Frederic Ayres - Moonlight in C-sharp major
Franz Schubert - Symphony No.9	Sergej Bortkiewicz - Elegie in C# Major
Franz Liszt Faust Symphony	Claude Debussy - L'ombre des arbres
Igor Stravinsky - Symphony in C major	Théodore Dubois - Heure héroïque

D major/B minor	D#major/ C minor
Beethoven - Symphony No. 2	Beethoven - Symphony No. 5
Johannes Brahms - Symphony No. 2	Johannes Brahms - Trío para piano
Dvořák: Symphony No.6	Anton Bruckner - Symphony No. 1
Joseph Haydn Symphony No. 13	Anton Bruckner - Symphony No. 2
Joseph Haydn Symphony No. 70	Anton Bruckner - Symphony No.8
Ralph Vaughan Williams Symphony No. 5	Gustav Mahler - Symphony No. 2
Joseph Haydn Symphony No. 93	Camille Saint-Saëns - Symphony No. 3
Joseph Haydn Symphony No. 96	Dmitri Shostakovich - Symphony No. 4
Wolfgang Amadeus Mozart Symphony No. 31	Dmitri Shostakovich - Symphony No. 8
Wolfgang Amadeus Mozart Symphony No. 35	Joseph Haydn - Symphony No. 52
Joseph Haydn Symphony No. 104	Joseph Haydn - Symphony No. 95
Jean Sibelius - Symphony No. 2	Johannes Brahms - Symphony No. 1
Sergei Prokofiev - Symphony No. 1	Dmitri Shostakovich - Symphony No. 8
Gustav Mahler - Symphony No. 9	Beethoven - Piano Concerto No 3
Gustav Mahler - Symphony No. 1	Frédéric Chopin - Rondo Op. 1

E major/C# minor	F major/ D minor
Vivaldi - «La primavera»	Beethoven - Symphony No. 9
Franz Liszt - Hungarian Rhapsody	Vivaldi- «L'autunno»
Franz Liszt - Hungarian Rhapsody No. 2	Mozart- Requiem
Beethoven - Moonlight Sonata	Johannes Brahms - Hungarian Dance No. 2
Arnold Bax - Symphony No. 5	Beethoven: Symphony No.6
Ernest Bloch - Symphony in C-sharp minor	Beethoven - Tempest Sonata No. 17, Opus 31 No. 2
Joseph Martin Kraus - Symphony in C-sharp minor	Johannes Brahms, Symphony No 3
Gustav Mahler - Symphony No. 5	Johannes Brahms - Violin Sonata No 3
Nikolai Myaskovsky - Symphony No. 2	Pyotr Tchaikovsky - Cherubim's Song
Sergei Prokofiev - Symphony No. 7	Franz Liszt - Cantico di San Francesco
Ture Rangström - Symphony No. 1	Johann Sebastian Bach - Wie schön leuchtet der Morgenstern
J. S. Bach - Liebster Gott, wenn werd ich sterben?	Antonín Dvořák - Symphony No. 5, Op. 76
Frédéric Chopin - Scherzo no. 3 opus 39	Mozart - Piano Concerto No. 01
Richard Wagner - Symphony E Major	Frédéric Chopin - Rondo à la mazur
Antonio Vivaldi - Concerto La Cetra No.11	Antonio Vivaldi - La tempesta di mare

F# major/ D# minor	G major/ E minor
Fryderyk Chopin - Barkarola	Fryderyk Chopin - Życzenie Op. 74 Nr 1
Beethoven - Sonata No. 24	Fryderyk Chopin - Koncert fortepianowy Nr 1
George Frederick Bristow - Symphony in F-sharp minor	Fryderyk Chopin - Prelude
Alexander Glazunov - Symphony No. 2	Piotr Ilich Tchaikowski - Symphony No.5
Joseph Haydn - Symphony No. 45	Beethoven - Sonata No. 25
Myaskovsky - Symphony No. 21	Antonín Dvořák - Symphony No. 8
Joseph Haydn - Symphony No. 88	Pyotr Tchaikovsky - Elegy for string orchestra
Joseph Haydn - Symphony No. 92	Joseph Haydn - Symphony No. 8
Joseph Haydn - Symphony No. 94	Mozart - Violin Sonata No. 21
Joseph Haydn - Symphony No. 100	Frédéric Chopin - Concerto for Piano and Orchestra No. 1
Gustav Mahler - Symphony No. 4	Franz Liszt - Ave maris stella
Ralph Vaughan Williams - Symphony No. 2	Mozart - Missa brevis in G major
Frédéric Chopin - Impromptu No. 2	J. S. Bach - Brandenburg Concerto No. 3
Frédéric Chopin - Allegretto in F# major	J. S. Bach - Cello Suite No. 1
Franz Liszt - Apparition No.1	Ralph Vaughan Williams - A London Symphony

G# major/ F minor	A major/ F# minor
Vivaldi- «L'inverno»	Johannes Brahms - violin sonata op. 100 no. 2
Fryderyk Chopin - Koncert fortepianowy Nr 2	Beethoven: Symphony No.7
Johannes Brahms - Hungarian Dance No. 4	Anton Bruckner - Symphony No. 6
Piotr Ilich Chaikovsky - Symphony No. 4	Joseph Haydn - Symphony No. 59
Ludwig van Beethoven - Symphony No. 6	Joseph Haydn - Symphony No. 64
Ludwig van Beethoven - Symphony No. 8	Felix Mendelssohn - Symphony No. 4
Anton Bruckner - Symphony in F minor	Wolfgang Amadeus Mozart - Symphony No. 29
Joseph Haydn - Symphony No. 49 "La Passione"	Dmitri Shostakovich - Symphony No. 15
Richard Strauss - Symphony No.2 in F minor	J. S. Bach - Ach Gott, wie manches Herzeleid
Pyotr Tchaikovsky - Hamlet fantasy overture	J. S. Bach - Wer Dank opfert, der preiset mich
Beethoven- Sonata No. 1 in F minor	J. S. Bach - Aus tiefer Not schrei ich zu dir
Frédéric Chopin - Fantaisie in F Minor	Robert Schumann - String Quartet op. 41 n. 3 in A major
Mozart - Fantasia in F minor	Alexander Glazunov - Symphony No. 2
Mozart - Così dunque tradisci	George Frederick Bristow - Symphony No 3
Franz Liszt - La Leggierezza	Pyotr Tchaikovsky - The Voyevoda

A# major/ G minor	B major/ G# minor
Vivaldi - «L'estate»	Franz Liszt - Gottes ist der Orient
W. A. Mozart - Symphony No. 25	Franz Liszt - La campanella
Fryderyk Chopin - Ballada	Franz Liszt - Les cloches de Geneve
Johannes Brahms - Hungarian Dance No 5	Frédéric Chopin - Etude op. 25 n° 6
Tomaso Albinoni - Adagio for organ and strings	Frédéric Chopin - Prelude Op.28 No.11
Piotr Tchaikovski - Symphony No. 1	Frédéric Chopin - Prelude op. 28 no. 12
John Williams - The Imperial March	Johannes Brahms - Liebe und Frühling
Joseph Haydn - Symphony No. 39	Johannes Brahms - Piano Trio No.1
Beethoven- Cello Sonata No.2 in G minor	Johannes Brahms - Ballade Op.10 No 4
Mozart - Symphony No. 40	Elliot Goldenthal - Symphony in G# minor
Carl Nielsen - symphony No 1	Georg Matthias Monn - Sinfonia in B major
J. S. Bach - Meine Seel erhebt den Herren	Erich Wolfgang Korngold - Sinfonietta op 5
J. S. Bach - Puer natus in Bethlehem	Dmitri Shostakovich - Symphony No. 2
Johannes Brahms - Piano quartet no.1 in G minor	Joseph Haydn - Symphony No. 46
Franz Schubert - Fantasy in G-Minor	Nikolai Myaskovsky - Symphony No. 17

Anexo 2. Certificación de funcionalidad de la aplicación.

La revisión de la aplicación para su certificación fue realizada por docentes del Conservatorio de música Salvador Bustamante Celi de la ciudad de Loja.

El día 01 de diciembre del 2015 se realizó una exposición del software a 5 docentes especializados en guitarra, además cada uno de ellos realizó las pruebas de funcionalidad personalmente.

Para conocer el nivel de aceptación por parte de los docentes, se realizó una encuesta y se tabularon las respuestas obtenidas.

TABULACIÓN DE DATOS DE LA ENCUESTA DE FUNCIONALIDAD
Software inteligente para el procesamiento musical de guitarra melódica
aplicando una red neuronal supervisada

La calificación de cumplimiento de cada una de las características del software estuvo dada desde 1 a 5, siendo:

- 1: No
- 2: Regular
- 3: Bueno
- 4: Satisfactorio
- 5: Excelente

Pregunta 1: ¿El Software es fácil de manejar?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	0	0%
Satisfactorio	1	20%
Excelente	4	80%
Total	5	100%

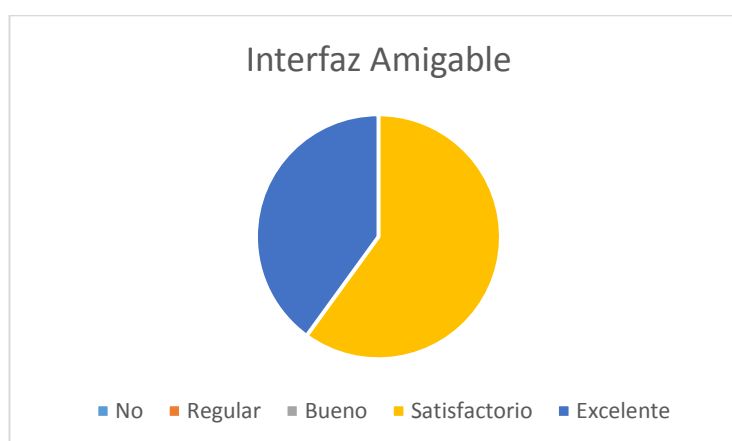


Interpretación: El 80% de los docentes encuestados respondieron que la aplicación ofrece una excelente facilidad de manejo mientras que el 20% respondieron que el nivel de facilidad de manejo es satisfactorio.

Análisis: La mayoría de docentes encuestados encuentra la aplicación muy fácil de manejar, mientras que un grupo menor considera que aún se puede mejorar en algunos puntos.

Pregunta 2: ¿La interfaz es amigable/intuitiva?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	0	0%
Satisfactorio	2	40%
Excelente	3	60%
Total	5	100%



Interpretación: El 60% de los docentes encuestados respondieron que el nivel de facilidad de interacción con la interfaz es satisfactorio mientras que el 40% respondieron que la aplicación ofrece un excelente nivel de interacción con el usuario.

Análisis: La mayoría de docentes encuestados piensa que, aunque la interfaz de la aplicación es bastante intuitiva, se puede utilizar elementos más explicativos en los menús.

Pregunta 3: ¿La información que presenta es clara y entendible?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	0	0%
Satisfactorio	2	40%
Excelente	3	60%
Total	5	100%



Interpretación: El 60% de los docentes encuestados respondieron que la información que se presenta en la aplicación es clara a nivel satisfactorio mientras que el 40% respondieron que la información que se presenta en la aplicación es clara a nivel excelente.

Análisis: La mayoría de docentes encuestados piensa que la información que presenta la aplicación en sus respuestas es bastante clara, no obstante se puede utilizar términos un poco menos técnicos para que la entienda todo público.

Pregunta 4: ¿Las funciones que presenta el sistema son útiles?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	0	0%

Satisfactorio	0	0%
Excelente	5	100%
Total	5	100%



Interpretación: El 100% de los docentes encuestados respondieron que las todas las funciones que presenta el software son útiles.

Análisis: Todos los docentes encuestados creen que las funciones con que cuenta el software son muy útiles para los usuarios que en este caso serían los aprendices de música con conocimientos a nivel básico.

Pregunta 5: ¿Las respuestas del software están en concordancia con la teoría musical?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	0	0%
Satisfactorio	2	40%
Excelente	3	60%
Total	5	100%



Interpretación: El 60% de los docentes encuestados respondieron que el nivel de veracidad de respuestas del software es excelente, mientras que el 40 % respondieron que el nivel es satisfactorio.

Análisis: La mayoría de docentes encuestados aprueba las respuestas que presenta el software porque van de acuerdo a la teoría musical, no obstante se pueden agregar más puntos que acompañen la enseñanza a ese nivel.

Pregunta 6: ¿El software ayuda a entender las reglas de composición musical?

RESPUESTA	Cantidad	Porcentaje
No	0	0%
Regular	0	0%
Bueno	1	20%
Satisfactorio	3	60%
Excelente	1	20%
Total	5	100%

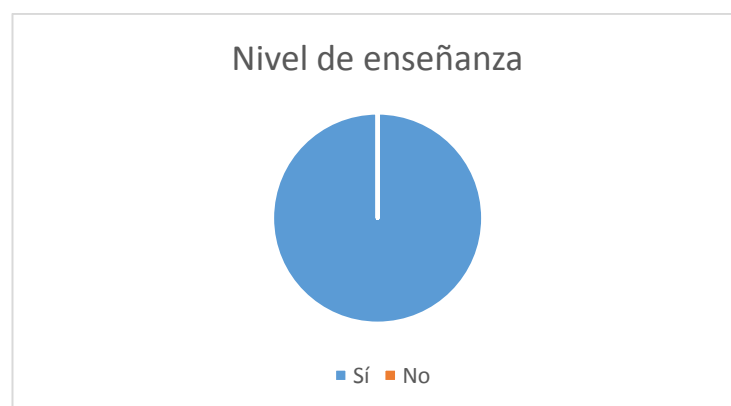


Interpretación: El 60% de los docentes encuestados respondieron que la el nivel de aplicabilidad en la enseñanza que pueda prestar el software es satisfactorio, el 20% respondieron que dicho nivel es bueno y el restante 20% respondieron que el nivel es excelente.

Análisis: La mayoría de docentes encuestados piensa es una herramienta de gran ayuda para los estudiantes de música a nivel básico, aun así se podría mejorar presentando las respuestas en un plano que represente el pentagrama musical para que sea más explicativo dado el nivel de los usuarios.

Pregunta 8: ¿Recomendaría el uso del software como alternativa de apoyo a la composición musical?

RESPUESTA	Cantidad	Porcentaje
Sí	5	100%
No	0	0%
Total	5	100%



Interpretación: El 100% de los docentes encuestados recomendarían el uso del software como alternativa de apoyo a la composición musical.

Análisis: Todos los docentes encuestados están de acuerdo en que la aplicación es una muy buena herramienta de apoyo para la educación tanto para ellos como para los estudiantes.


Loja, 14 de Octubre del 2015

Lic. Daniel Morocho
DOCENTE DE MÚSICA
COLEGIO BEATRIZ CUEVA DE AYORA

Certifica:

Que los egresados: Rodrigo Cornelio Granda Veintimilla y Juan Guillermo Pinzón González, autores del trabajo titulado: "Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada", una vez probado el software desarrollado, en el ámbito musical cumple con la funcionalidad deseada ya que sigue la teoría musical de manera correcta.

Es todo cuanto puedo decir en honor a la verdad, pudiendo los interesados hacer uso del presente en lo que estimen conveniente.


.....
Daniel Morocho
LICENCIADO EN CIENCIAS DE LA EDUCACIÓN
MENCIÓN EDUCACIÓN MUSICAL

Anexo 3: Certificación Revisión Literaria

Loja, 27 de Octubre del 2015

Dr.

Segundo Abel León Ojeda

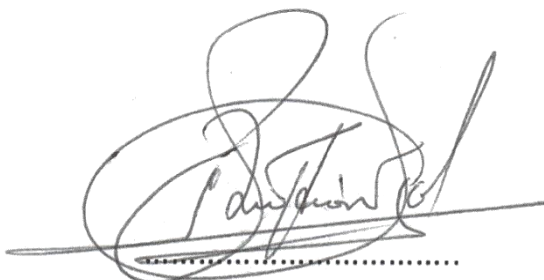
DOCENTE DE LENGUA Y LITERATURA

UNIDAD EDUCATIVA VICENTE ANDA AGUIRRE SECCIÓN DIURNA LA DOLOROSA

Certifica:

Que los egresados: Rodrigo Cornelio Granda Veintimilla y Juan Guillermo Pinzón González, autores del trabajo titulado: "Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada", en la redacción e interpretación del contenido de su trabajo, han cumplido con todas las normas y reglas gramaticales del idioma español, las cuales han sido revisadas minuciosamente para dar cumplimiento a dicho trabajo.

Es todo cuanto puedo decir en honor a la verdad, pudiendo los interesados hacer uso del presente en lo que estimen conveniente.

A handwritten signature in black ink, appearing to read 'Segundo Abel León Ojeda', is written over a horizontal dotted line.

Dr. Segundo Abel León Ojeda
LICENCIADO EN CIENCIAS DE LA EDUCACIÓN
ESPECIALIDAD LENGUA Y LITERATURA

Anexo 4: Licencia Creative Commons.



Software inteligente para el procesamiento musical de guitarra melódica
aplicando una red neuronal supervisada by Juan Guillermo Pinzón González and
Rodrigo Cornelio Granda Veintimilla
is licensed under a [Creative Commons
Reconocimiento-NoComercial 4.0 internacional License](#).

Certificación Summary

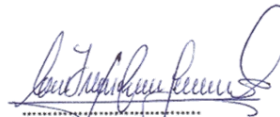
Loja, 26 de Octubre del 2015

Licda.
Carlota Ortiz Guaman
DOCENTE DE INGLÉS
COLEGIO LA INMACULADA

Certifica:

Que los egresados: Rodrigo Cornelio Granda Veintimilla y Juan Guillermo Pinzón González, autores del trabajo titulado: "Software inteligente para el procesamiento musical de guitarra melódica aplicando una red neuronal supervisada", en la traducción del resumen, han cumplido con todas las normas y reglas gramaticales del idioma inglés, las cuales han sido revisadas minuciosamente para dar cumplimiento con la sección summary de dicho trabajo.

Es todo cuanto puedo decir en honor a la verdad, pudiendo los interesados hacer uso del presente en lo que estimen conveniente.



Carlota Ortiz Guaman
LICENCIADA EN CIENCIAS DE LA EDUCACION
ESPECIALIDAD EN INGLES