

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO DE SONIDO.
2. **TITULO:** Desarrollo de un sistema de obtención de datos de ejecución fingerstyle y pulsación de traste de un bajo eléctrico con el fin de entregar un reporte de interpretación al usuario.
3. **AUTORES:** David Mateo Orellana Reyes y David Alejandro Sánchez Rodríguez.
4. **LUGAR:** Bogotá D.C
5. **FECHA:** Diciembre de 2016.
6. **PALABRAS CLAVES:** Protocolo de comunicación, Transmisión de datos, Fingerstyle, Temperado, Bluetooth, Micro controlador.
7. **DESCRIPCIÓN DEL TRABAJO:** Desarrollar un sistema de obtención de datos de ejecución fingerstyle y pulsación de traste en un bajo eléctrico, comparando los datos recibidos con los preestablecidos, que por medio de una aplicación para dispositivos móviles permita la visualización de los resultados de ejecución.
8. **LÍNEAS DE INVESTIGACIÓN:** Procesamiento de señales.
Núcleo Problemático: Audio y Diseño de sistemas.
9. **METODOLOGÍA:** Tipología de proyecto aplicado experimental.
Enfoque de investigación mixto. (Cuantitativo y Cualitativo)
10. **CONCLUSIONES:**
 - Por medio de la inserción de 3 sensores FSR cuadrados en los primeros 3 trastes y abarcando las cuatro cuerdas del instrumento, se logró reducir el porcentaje de error a la hora del reconocimiento de nota y posición de los dedos, de modo que se tuvo que crear una segunda condición en el algoritmo, para así reconocer la pulsación de traste y también un complemento en la programación de un aspecto que no se tenía en cuenta; el reconocimiento de octava de la nota ejecutada, parámetro que causaba algunos problemas en la identificación de la nota en términos de traste y cuerda.
 - Se logró de forma exitosa el desarrollo de un sistema concreto conformado por un conjunto de subsistemas acoplados entre si, estructurado con sistemas abstractos

(Aplicación, Programación de micro controlador y algoritmos) y sistemas físicos (electrónica, sensórica y herramientas de uso cotidiano(Dispositivo móvil)).

- Respecto a la implementación de herramientas como: Arreglo de sensores, programación de un analizador de frecuencia, Arduino y la aplicación de dispositivo móvil la cual se encarga de generar un reporte de aciertos y errores al usuario, se puede llegar a concluir que este sistema físico concreto puede sustituir a un tutor, generando una guía interactiva y una retroalimentación en tiempo real.
- En cuanto a la transmisión de datos se pudo concluir lo siguiente: el sistema tiene un blindaje importante en cuestión de envío y recepción de datos, es decir que no habrá ningún tipo de interferencia en la comunicación entre el micro controlador y el dispositivo móvil, así mismo se pudo concluir que cuando el paquete de datos es ligeramente pesado, se generarán retardos en el envío y recepción de la información.
- Precizando que ya se tenía como limitante el retardo en la transmisión de datos debido al uso del protocolo Bluetooth, fue oportuno implementar un arreglo de leds y un visualizador de 7 segmentos, los cuales ayudaron a mejorar el rendimiento del sistema en cuanto a tiempo, ya que envía una visualización de la información requerida por el usuario, con mayor velocidad a la que aparece visualizada en el dispositivo móvil.
- Se realizó una prueba subjetiva con el fin de determinar el punto de vista de la población seleccionada, la cual estaba constituida por estudiantes de música de últimos semestres (9 y 10) y estratificada por estudiantes que sabían ejecutar el bajo eléctrico y los que no. Con dicha prueba se pudo concluir que: La mayoría de la muestra seleccionada manifestó que el Bassmaster es una herramienta muy útil a la hora de instruir y retroalimentar al usuario. A su vez, la mayoría de encuestados concluyeron que la falencia del sistema era

la limitación de no ser capaz de identificar otras técnicas aparte de fingerstyle y pulsación de traste.

**Desarrollo de un sistema de obtención de datos de ejecución fingerstyle y ^{iv}
pulsación de traste de un bajo eléctrico con el fin de entregar un reporte de
interpretación al usuario.**

Proyecto de Grado Presentada Para Obtener El Título De
Ingeniero de Sonido
Universidad De San Buenaventura, Bogotá

David Mateo Orellana Reyes & David Alejandro Sánchez Rodríguez.
Noviembre 2016.

Desarrollo de un sistema de obtención de datos de ejecución fingerstyle y pulsación de traste de un bajo eléctrico con el fin de entregar un reporte de interpretación al usuario.

Proyecto de Grado Presentada Para Obtener El Título De
Ingeniero de Sonido
Universidad De San Buenaventura, Bogotá

David Mateo Orellana Reyes & David Alejandro Sánchez Rodríguez.
Noviembre 2016.

Nota de Aceptación

Firma del Jurado

Firma del Jurado

Ciudad y fecha (dd/mes/año) _____, ____ / ____ / _____

Agradecimientos

vii

Dedicamos este proyecto y toda la carrera universitaria a Dios, porque ha sido un soporte en nuestras vidas para seguir adelante. Una mención especial a nuestros padres, por brindarnos su apoyo, cariño y confianza y por darnos la oportunidad de educarnos.
A todos los docentes y maestros que estuvieron y están al tanto del desarrollo de este proyecto.

Tabla de Contenidos

1.	INTRODUCCIÓN	2
2.	PROBLEMA DE INVESTIGACIÓN	5
2.1.	Planteamiento del Problema.	5
2.2.	Pregunta de investigación	6
3.	OBJETIVOS.	7
3.1.	Objetivo General.	7
3.2.	Objetivos Específicos.	7
4.	Alcances y Limitaciones.	8
4.1.	Alcances.	8
4.2.	Limitaciones.	8
5.	Antecedentes del problema.	9
6.	Justificación.	14
7.	MARCO DE REFERENCIA.	16
7.1.	Marco conceptual.	16
7.1.1.	Instrumento.	16
7.1.2.	Nivel de presión sonora.	17
7.1.3.	Micrófono.	17
7.1.4.	Niveles de señal.	19
7.1.5.	Cables.	20
7.1.6.	Frecuencias Fundamentales en el espectro de las notas.	22
7.1.7.	Hardware.	27
7.1.8.	Sensores.	27
7.1.9.	Preamplificadores.	33
7.1.10.	Sistemas embebidos.	37
7.1.11.	Interacción.	47
7.1.12.	Tipos de Interacción.	47
7.1.13.	Estilos de Interacción.	47
7.1.14.	Interfaz.	48
7.1.15.	Interfaz de usuario.	48
7.1.16.	Tipos de Interfaz de Usuario.	49
7.1.17.	Programación orientada a objetos.	53
7.1.18.	Protocolo de comunicación.	55
7.1.20.	Comunicación serial.	58
7.1.21.	Modos de transmisión.	58
7.1.22.	Sistemas operativos y aplicaciones.	59
7.1.23.	Desarrollo de Aplicaciones:	62
8.	DISEÑO METODOLÓGICO.	66
8.1.	Metodología y enfoque de la investigación.	66
9.	Categorías de análisis.	67
9.1.1.	Variables Dependientes.	67
9.1.2.	Variables Independientes.	67
10.	Hipótesis.	68

11.	DESARROLLO INGENIERIL	ix
11.1.	Selección de dispositivos y herramientas	69
11.1.1.	Instrumento temperado de cuerda pulsada (BAJO).....	73
11.1.2.	Implementación del bajo eléctrico Tom Grasso.	73
11.1.3.	Sensores.	74
11.1.4.	Diseño y desarrollo de pre amplificador.....	78
11.1.5.	Placa con micro controlador.	83
11.1.6.	Protocolo de comunicación.....	98
11.1.7.	Sistema Operativo.....	117
11.1.8.	Sistema Operativo.....	123
11.1.9.	Dispositivo móvil.....	124
11.1.10.	Entorno de programación.....	125
11.1.10.	Unión de subsistemas y consolidación del sistema físico.....	139
12.	Representación y análisis de los resultados.	140
12.1.	Medición subjetiva.....	140
13.	CONCLUSIONES	156
14.	RECOMENDACIONES.....	158
15.	BIBLIOGRAFIA	159

Lista de tablas

x

Tabla 1 Sensores artificiales y biológicos. [46].....	28
Tabla 2 Componentes del Seeduino [4].....	39
Tabla 3 Componentes del Arduino Uno.[4].....	41
Tabla 4 Comandos AT [27].	46
Tabla 5 Medios de transmisión no guiados. [90][48]	56
Tabla 6 de frecuencias fundamentales	78
Tabla 7 Frecuencias fundamentales (Nivel Tutorial Easy).....	106
Tabla 8. Frecuencias fundamentales (Nivel Tutorial Medium).....	107
Tabla 9. Frecuencias fundamentales (Nivel Tutorial Hard).....	109
Tabla 10 Comparación de los protocolos no guiados [97]	117
Tabla 11 Ficha técnica, dispositivo móvil implementado.	125
Tabla 12. Matriz en App inventor y nombre de las imágenes.	130
Tabla 13 Tabulación Grupo 1 y 2.	142
Tabla 14. Pruebas valor de voltaje de señal del instrumento.....	148
Tabla 15 Porcentaje de error en amplitud de cada nota.	149
Tabla 16 Porcentaje de error en amplitud de Vpp Out de cada etapa.....	152
Tabla 17. Comunicación entre Modulo Hc 06 y dispositivo móvil.....	154
Tabla 18. Porcentaje de error en la transmisión de datos de Arduino a la Aplicación.	155

Lista de ecuaciones

Ecuación 2 Expresión del nivel de presión sonora en dB [49].....	17
Ecuación 3 Expresión para hallar frecuencia fundamental [41].	23
Ecuación 4 Expresión para hallar frecuencia fundamental [41].	23
Ecuación 5 Expresión para hallar frecuencia fundamental [41].	24
Ecuación 6 Impedancia de entrada de etapa JFET.....	84
Ecuación 7 Corriente de drenaje.....	84
Ecuación 8 Resistencia en source.	85
Ecuación 9 Valor de gmo a partir de la corriente de drenaje a fuente (I_{DSS}) y el voltaje de pinch off(V_p).....	85
Ecuación 10 Trans-conductancia.....	85
Ecuación 11 Ganancia de voltaje etapa JFET.....	85
Ecuación 12 Resistencia en el drain.....	85
Ecuación 13 Voltaje gate source y voltaje en R_s	86
Ecuación 14 Voltaje en gate.	86
Ecuación 15 Resistencia R_2 Etapa JFET.....	86
Ecuación 16 Resistencia R_1 Etapa JFET.....	86
Ecuación 17 Valor de la resistencia de carga.....	87
Ecuación 18 Ganancia de voltaje etapa bipolar.	88
Ecuación 19 paralelo entre R_C y R_L	88
Ecuación 20 Resistencia en el emisor.....	88
Ecuación 21 Resistencia en DC.....	88
Ecuación 22 Resistencia en AC.....	89
Ecuación 23 Voltaje colector emisor de polarización.....	89

Ecuación 24 Corriente de polarización en el colector.	89
Ecuación 25 Resistencia de Thevenin.....	89
Ecuación 26 Ecuación de voltaje en la resistencia 2.....	90
Ecuación 27 Voltaje en la resistencia 2.	90
Ecuación 28 Resistencia 1 etapa bipolar.....	90
Ecuación 29 Resistencia 2 etapa bipolar.....	90
Ecuación 30 Fórmula muestral cualitativa conociendo el número de población.....	141
Ecuación 31 Porcentaje de error de la señal de entrada.....	148

Lista de figuras

xii

Figura 1. Especificaciones del bajo [8].....	16
Figura 2. Combinación de cables balanceados y no balanceados. [68].....	21
Figura 3 Notas fundamentales octava 0. [41].	25
Figura 4 Notas fundamentales octava 1. [41].	25
Figura 5 Notas fundamentales octava 2. [41].	26
Figura 6 Notas fundamentales octava 3. [41].	26
Figura 7 Notas fundamentales octava 4. [41].	27
Figura 8 Sensor de proximidad infrarrojo. [48].	29
Figura 9 Sensor de proximidad ultrasónico. [48].	30
Figura 10 Sensor de proximidad magnético. [48].	30
Figura 11 Sensor de fuerza y componentes. [48].	31
Figura 12 Sensor de fuerza resistivo FSR redondo [48].	31
Figura 13 Sensor de fuerza resistivo FSR Cuadrado. [48].	32
Figura 14 Convención de signos. [76].	34
Figura 15 Estructura transistor BJT npn.[99]	34
Figura 16 Estructura transistor BJT pnp.[99]	34
Figura 17 Circuito amplificador BJT[99]	36
Figura 18 Circuito amplificador MOSFET [99]	36
Figura 19 Arquitectura de un sistema embebido. [78].	38
Figura 20 Prototipo de Arduino. [24].	40
Figura 21 Frontal y reverso de una placa de Arduino uno. [4].	41
Figura 22 Elementos de la placa de Arduino Uno. [4].	42
Figura 23 Placa de Arduino Mega 2560. [24].	45
Figura 24 Modulo Bluetooth [27].	46
Figura 25 Ejemplo de CLI. [24].	50
Figura 26 Ejemplo GUI Fuente Internet. Interfaz gráfica tridimensional Compiz en GNU/Linux [85].	51
Figura 27 Hiroshi Ishii. [84].	51
Figura 28 Tangible Media Group, MIT TUI [85].	52
Figura 29 MUSIC BOTTLES [18].	53
Figura 30 Arquitectura de Android [94].	60
Figura 31 Arquitectura de iOS [94].	62
Figura 32 Diagrama de flujo de una Activity (oficial) [94].	65
Figura 33 Bajo Eléctrico.	74
Figura 34 Señal del Bajo Eléctrico.	75
Figura 35 Diagrama de flujo algoritmo Matlab.	76
Figura 36 Algoritmo implementado en Matlab.	77
Figura 37 Fundamental F en la cuerda E.	77
Figura 38 Sensor de fuerza resistivo FSR Cuadrado. [48].	79
Figura 39 Montaje de prueba del sensor	80
Figura 40 Medición de resistencia del sensor (valor mínimo).	80
Figura 41 Medición de resistencia del sensor (valor máximo)	81
Figura 42 SKETCH Arduino para prueba de sensor.	81
Figura 43 Arreglo de sensores en el mástil del bajo eléctrico.	82

Figura 44 Etapa JFET	xiii
Figura 45 Etapa Bipolar 1	84
Figura 46 Elevador de nivel DC (offset).....	87
Figura 47 Simulación de preamplificador en NI Multisim.....	91
Figura 48 Simulación multímetro en la salida del pre amplificador en NI Multisim.....	92
Figura 49 Preamplificador montado en protoboard.....	93
Figura 50 Señal de entrada.....	93
Figura 51 Señal de la primera etapa JFET.....	94
Figura 52 Señal de la segunda etapa BJT.....	95
Figura 53 Señal amplificada con elevador de voltaje.....	95
Figura 54 PCB del preamplificador.....	96
Figura 55 preamplificador montado en baquela.....	96
Figura 56 Entrada y salida del preamplificador mostrada por el osciloscopio.....	97
Figura 57 Arduino Mega. [24].....	97
Figura 58 Prueba comunicación Arduino Mega-Dispositivo móvil.....	98
Figura 59 Arduino Nano.....	100
Figura 60 Diagrama general de algoritmo en Arduino.....	100
Figura 61 Diagrama de conversión análoga-digital del valor de los sensores en el arduino nano.....	101
.....	102
Figura 62 Subprograma lectura de sensores (ingresa el valor análogo de entrada del sensor para envío de estado).....	103
Figura 63 Diagrama de sub-programa (Valor de entrada de puerto serial).....	104
Figura 64 Número de comparaciones en cada nivel.....	104
Figura 65 Diagrama para sub-programa Tutorial Easy.....	105
Figura 66. Diagrama para sub-programa Tutorial Medium.....	107
Figura 67. Diagrama para sub-programa Tutorial Hard.....	108
Figura 68. Número de comparaciones en cada nivel.....	109
Figura 69. Diagrama de sub-programa Challenge Easy.....	110
Figura 70. Diagrama de sub-programa Challenge Medium.....	111
Figura 71. Diagrama de sub-programa Challenge Hard.....	112
Figura 72. Diagrama de sub-programa (Audio IN).....	113
Figura 73. Instrucción, ejecución de nota F en aplicación.....	114
Figura 74. Arreglo de leds, instrucción de ejecución de nota F.....	114
Figura 75 Conexión del led en Arduino.....	115
Figura 76. Entradas empleadas para la conexión de los leds.....	115
Figura 77. Visualizador led de 7 segmentos con punto decimal.....	116
Figura 78 conexión visualizador led de 7 segmentos con punto decimal.....	116
Figura 79. Nota Fa sostenido en visualizador.....	116
Figura 80. RSSI Bluetooth, estabilidad de red, medición 1.....	120
Figura 81 RSSI Bluetooth, estabilidad de red, medición 2.....	120
Figura 82. Conexión módulo Hc 06 a ARDUINO. [24] [27].....	121
Figura 83. Algoritmo App inventor.....	122
Figura 84 Algoritmo SKETCH Arduino.....	123
Figura 85. Dispositivo móvil implementado.....	124
Figura 86 Algoritmo del sub programa Activity 1.....	126
Figura 87 Activity 1 vista en el dispositivo móvil.....	127

Figura 88 Algoritmo activity 2 y 3.	xiv 128
Figura 89 Activity 2 y 3 vistas en el dispositivo móvil.	128
Figura 90 Activity 4. Diagrama algoritmo Tutorial Easy.	129
Figura 91. Activity 4. Visualización en la aplicación de Tutorial Easy Nota 1 Fa.	131
Figura 92 Activity 5. Diagrama algoritmo Tutorial Medium.	132
Figura 93. Activity 6. Diagrama algoritmo Tutorial Hard.	133
Figura 94. Activity 7. Diagrama algoritmo Challenge Easy.	134
Figura 95. Activity 8. Diagrama algoritmo Challenge Medium.	135
Figura 96. Activity 9. Diagrama algoritmo Challenge Hard.	136
Figura 97. Ejemplo de selección de etapa challenge, nivel easy.	137
Figura 98. Resultados BassMaster.	138
Figura 99. Activity 10. Reporte de interpretación.	139
Figura 100. Gráfica pregunta número 1.	143
Figura 101. Gráficas pregunta número 2.	144
Figura 102. Gráfica pregunta número 3.	145
Figura 103. Gráfica pregunta número 4.	146
Figura 104. Gráfica pregunta número 5.	147
Figura 105. Prueba de arreglo de sensores.	150
Figura 106 Puerto serial Arduino. Reconocimiento de frecuencia fundamental.	152
Figura 107. Sesión en Pro Tools con los audios grabados.	165
Figura 108. Fundamental F en la cuerda E.	166
Figura 109. Fundamental F# en la cuerda E.	166
Figura 110. Fundamental G en la cuerda E.	167
Figura 111. Fundamental G# en la cuerda E.	167
Figura 112. Fundamental A# en la cuerda A.	168
Figura 113. Fundamental B en la cuerda A.	168
Figura 114. Fundamental C en la cuerda A.	169
Figura 115. Fundamental C# en la cuerda A.	169
Figura 116. Fundamental D# en la cuerda D.	170
I Figura 117. Fundamental E en la cuerda D.	171
Figura 118 Fundamental F en la cuerda D.	171
Figura 119. Fundamental F# en la cuerda D.	172
Figura 120. Fundamental G# en la cuerda G.	172
Figura 121. Fundamental A en la cuerda G.	173
Figura 122. Fundamental A# en la cuerda G.	173
Figura 123. Fundamental B en la cuerda G.	174

1. INTRODUCCIÓN

En el campo de la ingeniería se ha intentado incorporar herramientas hardware y software con el fin de desarrollar sistemas físicos (concretos) más completos que a su vez produzcan una relación hombre maquina funcional e interactiva.

Pensando en el avance y la proyección de la tecnología en los dispositivos portátiles y el fácil acceso de los usuarios a estas herramientas, se quiere implementar una aplicación interactiva con el fin de facilitar la forma en la que el usuario pretenda ejecutar un instrumento temperado de cuerda pulsada, dando así una serie de parámetros y ayudas sin la necesidad de un instructor. Basado en lo anterior, el proyecto que se presenta, está catalogado en la tipología de proyecto aplicado con un enfoque de investigación mixto; La aplicación práctica de éste trabajo, se basa en la realización de una herramienta que se desarrolla de la siguiente manera:

La primera parte consta de una aplicación para dispositivo móvil que cuenta con dos etapas primordiales en su estructura: la “Etapa Tutorial” y la “Etapa Challenge”. Cada una de estas etapas cuenta con tres niveles: Easy, Medium y Hard.

Estos 3 niveles se repiten en cada una de las etapas y para cada nivel se programó su respectiva sucesión de eventos o notas que deben ser ejecutadas por el usuario en el instrumento temperado de cuerda pulsada (bajo eléctrico). La diferencia está en que en la etapa Tutorial se brindan dos ayudas al usuario, la primera ayuda será el arreglo de leds con forma de bajo eléctrico que se implementó en un dispositivo hardware, en la cual se visualiza la nota o evento que el usuario debe tomar como guía para ejecutar según el

nivel seleccionado y la segunda ayuda será una imagen en la pantalla del dispositivo móvil la cual muestra un bajo eléctrico y una mano que indica mediante un código de colores cual será el dedo a usar y la nota que se debe ejecutar según la sucesión. Para obtener los datos de ejecución fingerstyle y pulsación de traste ejecutados por el usuario se desarrolló un “proceso de recepción de datos” en el dispositivo hardware que consiste en un pre amplificador elevador, este procesa y adapta la señal para que un micro controlador la reconozca, en el micro controlador se registra la pulsación de traste mediante el arreglo de sensores (con esto también se conoce la octava) que permite conocer como primer parámetro, que traste se está pulsando, además se reconoce la frecuencia fundamental de la señal eléctrica entrante que se analiza y procesa digitalmente para realizar una comparación con una matriz de datos pre establecidos (que son el conjunto de constantes previamente extraídas por medio de la transformada de Fourier o frecuencias fundamentales de la secuencia de eventos en cada nivel) dando así un segundo parámetro. En esta etapa el usuario no tiene opción de equivocarse ya que el programa está diseñado para validar los dos parámetros y solo si acierta podrá continuar con la sucesión.

En la etapa Challenge el usuario puede pulsar cualquier nota con la posibilidad de equivocarse dándole como ayuda únicamente la notación (Cifrado americano), que se visualizará en un display de 7 segmentos. Se realiza el “proceso de recepción de datos” y se obtiene tanto el parámetro del arreglo de sensores como el parámetro de la validación de la frecuencia fundamental, con estos datos se halla y se guarda una matriz de aciertos y errores.

Por último, en el dispositivo hardware se realiza la codificación de la matriz de aciertos y errores con el fin de generar la comunicación y enviar los datos al dispositivo móvil, haciendo uso de un módulo inalámbrico (protocolo L2CAP (BLUETOOTH)). Dada la comunicación se decodifican los datos en el dispositivo móvil para brindar la visualización del reporte de interpretación del usuario.

El hardware cuenta con su respectiva programación basada en Arduino y el dispositivo móvil con su aplicación desarrollada en App inventor.

2. PROBLEMA DE INVESTIGACIÓN

2.1. Planteamiento del Problema.

La aptitud de poder analizar e interpretar música, está ligada a un concepto humano que es la técnica de la escucha y las habilidades motoras (Finas), la cual permite generar significados a los sonidos y estímulos capturados. Gracias a esta aptitud el ser humano está en completa capacidad de crear instrumentos musicales e interactuar con ellos, para desarrollar destreza motriz ejecutándolos habitualmente es necesaria la asistencia de un tutor. Por lo general son usadas las clases presenciales para el desarrollo de estas habilidades, pero el intento de suplir este acompañamiento ha llevado a que las personas usen métodos como: Guías escritas, digitales, o como comúnmente ocurre por medio de video tutoriales; este intento ha traído grandes ventajas, pero a su vez presenta falencias al momento de obtener una retroalimentación de lo ejecutado en tiempo real. Con este proyecto se busca hacer uso de los dispositivos electrónicos (los cuales reconocen, almacenan, analizan y procesan la información en términos musicales), que permiten su aplicación para facilitar la interacción que refuerza la naturaleza básica de la musicalidad humana [98] y posibilitan herramientas para una retroalimentación en tiempo real.

Con el paso del tiempo, el tema musical ha sido foco de estudio para investigadores e ingenieros; incluso hay proyectos y tesis de grado como el Guitarduino Self Teaching Electronic Guitar [29], que hace uso de dispositivos electrónicos, guiando al usuario en la interacción con el instrumento, esta herramienta cumple parte del rol de un tutor pero no genera una retroalimentación en tiempo real, para desarrollar este proceso se hace

necesario recibir la información de dicha interacción, es decir la señal eléctrica generada por el usuario. El proyecto " Desarrollo de un algoritmo para la detección eficiente de acordes musicales"[71] resuelve parte de este problema logrando detectar la frecuencia fundamental de los acordes en diferentes muestras de audio. Sin embargo, dada la detección de frecuencia también es relevante conocer al instante, más información de la ejecución del usuario, para poder procesar, comparar y validar los datos con el fin de generar un reporte de interpretación que brinde una retroalimentación en tiempo real.

2.2. Pregunta de investigación

Una vez identificadas las necesidades, se plantea la siguiente pregunta: ¿Cómo mejorar la interacción entre usuario y bajo eléctrico, tomando conceptos desarrollados en el sistema de guía interactiva "Guitarduino" y en el algoritmo para la detección eficiente de acordes musicales, buscando generar retroalimentación en tiempo real de la ejecución fingerstyle y la pulsación de traste del usuario en el instrumento?

3. OBJETIVOS.

3.1. Objetivo General.

Desarrollar un sistema de obtención de datos de ejecución fingerstyle y pulsación de traste en un bajo eléctrico, comparando los datos recibidos con los preestablecidos, que por medio de una aplicación para dispositivos móviles permita la visualización de los resultados de ejecución.

3.2. Objetivos Específicos.

- Implementar un arreglo de sensores y un analizador de frecuencia en el Bajo eléctrico con el fin de obtener datos de la interpretación del usuario.
- Diseñar y realizar la aplicación para dispositivos móviles con el fin de recibir datos de la ejecución en el bajo eléctrico y generar un reporte de aciertos y errores para el usuario.
- Realizar una prueba objetiva para comprobar la efectividad del sistema.
- Realizar una prueba subjetiva a partir de una evaluación estadística en la población definida. (pruebas de usabilidad)

4. ALCANCES Y LIMITACIONES.

4.1. Alcances.

- Con este proyecto se pretende crear una herramienta que permita la obtención de datos de ejecución fingerstyle y pulsación de traste en un bajo eléctrico donde se comparan los datos obtenidos de la ejecución con los datos preestablecidos en un software para posteriormente validar la información correlacionada, en un sistema que cuenta con un micro-controlador comunicado con una aplicación para dispositivo móvil.
- Pensando en el avance y la proyección de la tecnología en los dispositivos portátiles y el fácil acceso de los usuarios a estas herramientas, se quiere implementar una aplicación interactiva con el fin de facilitar la forma en la que el usuario pretenda ejecutar un instrumento temperado de cuerda pulsada, dando así una serie de parámetros y ayudas sin la necesidad de un instructor.

4.2. Limitaciones.

- Cabe destacar que en el desarrollo del sistema no se podrán procesar acordes ni técnicas específicas tales como palm mute, slap, tapping etc.
- De igual forma es posible que se generen algunos retardos en el envío y recepción de los datos.

5. ANTECEDENTES DEL PROBLEMA.

En las últimas décadas se han diseñado y desarrollado diversas investigaciones, artículos y trabajos de grado que han sido elegidos de manera cuidadosa y teniendo en cuenta que se requiere y puede ser de utilidad en el presente proyecto. En el 2012 se diseñó y desarrolla un proyecto de grado llamado Fiammetta 2.0 interfaz sonora de carácter digital [58], desarrollado en la Universidad Politécnica de Valencia por Carlos Yenes, expone el desarrollo de una caja de ritmos digital y el estrecho lazo que hay entre hombre y máquina, por medio de un sistema que contenía diferentes elementos. En la elaboración de este proyecto de grado analizaron diferentes temas que se requerían para realización de este prototipo, los más relevantes son: Programación en Arduino, posicionamiento y programación de LDR foto resistores (sensores de luz), video tracking y por último la relación entre hombre y máquina. El desarrollo de este proyecto es de gran utilidad, ya que no solo es un sistema basado en el posicionamiento de sensores, manejo de Arduino en un sistema, sino que también abarca un tema muy importante para nosotros como es la relación hombre-máquina dentro de un sistema.

En el 2013 se desarrolla la tesis “Gestión De Mecanismos Arduino Controlados Inalámbricamente Por Dispositivos Android” [3] guiada por la ingeniera MARTA PRIM SABRIÀ y desarrollada en la Universidad Autónoma de Barcelona por el ingeniero ADRIÁN FERNÁNDEZ, es un sistema que se basa en la activación de una alarma a distancia, por medio de transmisión inalámbrica, utilizando Bluetooth y manipulado por una aplicación en un dispositivo móvil cuya plataforma es ANDROID. La aplicación se desarrolló con la ayuda de un software llamado Androide software development kit

(Android SDK) cuyo lenguaje de programación es java, complementado por el sistema de programación gratuito App Inventor de Google Labs y cuyo algoritmo se trabaja en lenguaje visual de bloques, su manipulación es completamente gratuita y permite la compresión a formato APK.

Dentro del sistema desarrollado por el Ingeniero ADRIÁN, se hace uso de Arduino y de iTeed un módulo Bluetooth compatible con Arduino, posteriormente complementó el sistema con la transmisión de datos que enviaba a la alarma a la aplicación mediante un pequeño programa o sketch que desarrolló en Arduino. A partir de lo anterior se puede observar un sistema el cual puede ser manipulado inalámbricamente y lo más importante, desde un dispositivo móvil dando confort y versatilidad.

En el 2009 Luis Carrión, Darío Ochoa y Andrés Valverde redactan un trabajo describiendo el funcionamiento del sensor de fuerza resistivo [6], dando información exacta de las características de trabajo de éste y expone los siguientes modelos: FSR400, FSR402, FSR406 y FSR408. A partir de estas ideas, se describen tablas de información como es el caso de fuerza vs la resistencia, fuerza vs la conductancia y se estudian comportamientos como, el rango de sensibilidad de la fuerza, rango de sensibilidad de la presión, resistencia sin aplicar fuerza y resistencia de máxima corriente, dando así una guía importante para el uso de dichos sensores en el presente proyecto de grado, ya que se implementará un arreglo de sensores de fuerza resistivo.

En el 2015 se desarrolla una investigación “Desarrollo de un algoritmo para la detección eficiente de acordes musicales” [71] por el ingeniero Óscar Arturo Ramírez, siendo este uno de nuestros antecedentes principales, con el que busca obtener una maestría en

sistemas computacionales. Se basa en el planteamiento de desarrollo de un algoritmo, con el fin de conseguir la detección de acordes musicales en una muestra de audio, de igual forma el ingeniero Ramírez quiere que el algoritmo sea implementado en un marco de trabajo en el cual facilitará la portabilidad de este, y que pueda ser utilizado como base de proyectos posteriores que requieran la detección de tonos y acordes.

Dentro del desarrollo de este trabajo hay diferentes temas que pueden servir para el presente proyecto, como son los siguientes: La arquitectura de este sistema, detección monofónica y polifónica, señales musicales, obtención de FFT, obtención de componentes de una nota y de un acorde, y detección de un acorde. Estos temas son importantes ya que dentro del sistema planteado en este proyecto es necesario la obtención de nota pulsada en el instrumento temperado.

En el año 2011 Andrew Garza en la UIC (University of Illinois at Chicago College of Engineering), muestra un procedimiento que llevó a cabo para el diseño y desarrollo de un proyecto que comenzó como una idea DIY (Do It Yourself) y que con el paso del tiempo modificó y complementó para entregar lo que se conoce como Guitarduino Self Teaching Electronic Guitar [29]. Guitarduino básicamente es un sistema basado en Arduino, que enseña a los usuarios los acordes por medio de un conjunto de leds incrustados en el brazo de la guitarra y una pantalla LCD, mostrando el acorde y la posición de los dedos en el instrumento.

Este proyecto toca temas importantes para el presente proyecto como son: Manejo y especificaciones de Arduino, Programación en Arduino, Circuitos electrónicos para el

diseño de filtros, conexiones a la pantalla LCD, conexiones a los 100 leds y conexiones de los pulsadores para crear un sub menú, y por supuesto la interacción hombre-máquina. En el año 2014 el artículo Democraztizing computing with app inventor [17] ,expuesto en la revista “GETMOBILE” fue escrito por tres ingenieros reconocidos en el ámbito de la programación y diseñadores de App inventor; David Wolber de la Universidad de San Francisco, Harlod Abelson del MIT (Instituto Tecnológico de Massachusetts) y Mark Friedman Programador de Google Inc.

Básicamente exponen las ventajas de App inventor, las características que tiene la programación de aplicaciones, también hablan de la historia y como App inventor fue evolucionando con el tiempo.

Este artículo es una guía importante para el proyecto que se desarrollará, ya que explican a fondo todo lo que contiene App inventor, exponiendo las facilidades de uso, la ventaja de ser gratuito, como manejar el entorno de programación por medio de bloques, y emular las aplicaciones sin la necesidad de tener a la mano un dispositivo móvil.

En 2012 “Desarrollo De Un Chat Para Dispositivos Móviles Android Basados En El Protocolo de Comunicación Bluetooth” [86] el proyecto de grado diseñado y desarrollado por Nicolás Martínez y Omar Rocha, en la universidad EAN de Bogotá en la facultad de ingeniería, fue guiado por la Ingeniera Luz Amparo Acosta, directora del programa de ingeniería de sistemas. El proyecto tuvo un enfoque en la tecnología de transmisión de datos de corto alcance en los dispositivos móviles, se usó el protocolo Bluetooth, con el fin de evidenciar que aunque en el mercado de aplicaciones no se utiliza mucho este

protocolo, tiene muchas ventajas y beneficios que el usuario puede aprovechar dentro de una aplicación para dispositivos móviles, en este caso una aplicación de chating.

Dentro de la investigación de este proyecto nos arroja datos importantes como: Estudios del uso actual de las aplicaciones para dispositivos móviles que utilicen Bluetooth para la transferencia de datos, características de las versiones del sistema operativo Android, funcionamiento del protocolo de comunicación Bluetooth en dispositivos móviles y la utilidad en el futuro con otras aplicaciones.

6. JUSTIFICACIÓN.

Las investigaciones en cuanto a temas relacionados con el campo musical han venido creciendo, uno de los focos más importantes es el proceso mental y comportamiento de los humanos frente a la música; estudios aseguran que practicar la ejecución de algún instrumento puede tener impacto en la estructura mental como en sus áreas funcionales. En los últimos años se han realizado estudios a músicos profesionales, para demostrar cuán importante es el aprendizaje de un instrumento en el cerebro humano; Lutz Jäncke del Instituto de Tecnología de Zúrich, asegura que al aprender a tocar un instrumento musical se mejora la fisiología cerebral. Al interactuar con un instrumento musical se fortalecen aspectos como: Aumento de intelecto, hábitos de disciplina, manejo de estrés, mejora notable de la coordinación en personas con problemas de origen neurológico (dislexia, digrafía, dislalia entre otras). [99]

Por lo tanto se puede considerar de suma importancia el desarrollo de herramientas interactivas ligadas a la música que sirvan no solo para obtener datos de una nota musical por medio de análisis, sino que también sirvan para generar retroalimentación en tiempo real en cuanto a la ejecución del instrumento, de este modo se pueden aprovechar los diferentes campos como la electrónica y la programación, además de herramientas de uso diario como dispositivos móviles para desarrollar sistemas concretos eficaces.

Para poder formular los objetivos es necesario responder a las siguientes preguntas.

- ¿El proyecto se materializará de forma práctica? Si, la base primordial de este proyecto es el desarrollo de un sistema concreto o físico, para posteriormente probarlo y documentar el proceso del desarrollo.
- ¿Es un prototipo y que tipo de prototipo es? Si, el sistema a desarrollar es algo experimental y no definitivo. Es un sistema concreto mixto que contiene una parte hardware y una parte software, lo que se considera como sistema electrónico digital.
- ¿Qué tipo de funciones realiza? Tiene diferentes funciones pero la principal es encargarse de obtener e identificar las notas ejecutadas en un instrumento musical (bajo eléctrico) y retroalimentar al usuario en tiempo real, de lo que deriva como resultado la interacción humano-máquina y el desarrollo de destreza motriz en la ejecución del instrumento.
- ¿Cuál es la población objetivo? El proyecto está dirigido a cualquier persona que desee interactuar de forma básica con el bajo eléctrico.
- ¿Qué tipo y estilo de interacción tendrá la herramienta? Este será de tipo externo y su estilo será de manipulación externa y selección de menús.

7. MARCO DE REFERENCIA.

7.1. Marco conceptual

7.1.1. Instrumento.

El bajo eléctrico, conocido comúnmente como bajo es un instrumento temperado de cuerda pulsada, principalmente ejecutado con los dedos o pick; hay diferentes técnicas para su ejecución como: plucking, slapping, popping, tapping, o thumping. Este instrumento es familia de los cordófonos, contiene similitudes en su apariencia y construcción a la guitarra eléctrica, pero con mayores dimensiones en su cuerpo, mástil y escala. Normalmente posee cuatro cuerdas y su afinación estándar es la misma del contrabajo, no obstante existen otros tipos, con mayor número de cuerdas conocidos usualmente como “bajos de rango extendido” [8]. A continuación en la *Figura 1*, se expone los detalles de las especificaciones de un bajo eléctrico.



Bass guitar	
 <p>A Rickenbacker 4001</p>	
String instrument	
Other names	electric bass guitar, electric bass, bass
Classification	String instrument (plucked or picked; rarely strummed)
Hornbostel-Sachs classification	321.322 (Composite chordophone)
Inventor(s)	Paul Tutmarc
Developed	1930s
Playing range	
 <p>(a standard tuned 4 string bass guitar)</p>	

Figura 1. Especificaciones del bajo [8].

7.1.2. Nivel de presión sonora.

Para comprender que es nivel de presión sonora; vamos hablar principalmente de que significa presión sonora. La “Presión Sonora” hace referencia a la desviación que hay en la presión del aire respecto a la presión atmosférica, y este fenómeno es causado por una onda sonora. Después de entender que es la presión sonora, nos enfocaremos en el nivel de presión sonora; El Nivel de Presión Sonora es la medición logarítmica del valor promedio de la presión sonora respecto a un nivel de referencia; este nivel de referencia es la presión sonora del nivel más débil audible que es 20 micro pascales.

La presión sonora se mide en pascales, mientras que el nivel de presión sonora se mide en decibeles, y su abreviación es “dB SPL”, estas siglas vienen de “Sound Pressure Level” que son sus siglas en inglés. Entre más decibeles SPL tenga un sonido más fuerte será la percepción en el oído humano. [49] [69]. El nivel de presión sonora está definido por la *Ecuación 1*. [49]

$$Spl = 20 \log \left(\frac{P}{P_0} \right)$$

Ecuación 1 Expresión del nivel de presión sonora en dB [49].

P es el valor cuadrático medio de la presión sonora y Po la presión de referencia [49].

7.1.3. Micrófono

Un micrófono básicamente se puede definir como: un elemento usado para la transformación de ondas sonoras en energía eléctrica y viceversa. Los micrófonos son usados para diferentes fines en muchos campos como: la salud, la ciencia, la ingeniería,

los medios audiovisuales entre otros; hay diferentes tipos de micrófonos, pero los mas sobresalientes son los siguientes:

Micrófonos dinámicos: Los micrófonos de bobina móvil o coloquialmente conocidos como micrófonos dinámicos son los micrófonos mas utilizados en el planeta. Tienen muchas ventajas ya que no necesitan energía externa, son duraderos, resistentes y dependiendo la marca y referencia son de excelente calidad. El funcionamiento de este tipo de micrófonos se basa en la vibración del diafragma por medio de las ondas sonoras, y al mismo tiempo produce vibración en la bobina suspendida en un campo magnético, obteniendo corriente eléctrica de muy pequeña magnitud provocada por la fricción. La fluctuación de esta corriente es similar en forma, amplitud y frecuencia a la onda sonora que la generó [66] [67].

Micrófonos Capacitivos: Micrófonos capacitivos o bien llamados micrófonos de condensador, son transductores los cuales basan su funcionamiento en dos placas metálicas separadas, en el momento que este se conecta a una fuente de energía las dos placas de condensador se cargan, con lo cual se aplica un voltaje a las placas del capacitor. Por lo general una de sus placas es móvil y la otra fija. Estas placas son separadas por un material aislante por lo que se puede usar una de estas como diafragma directamente, además que se puede variar el voltaje que es el objetivo del convertir impulsos mecánicos en impulsos eléctricos. Entre más cerca estén las placas mayor será el voltaje y viceversa [66] [67].

Micrófonos piezoeléctricos: Los materiales piezoeléctricos producen un desplazamiento de cargas cuando son deformados mecánicamente. Su funcionamiento depende

exclusivamente de hacer vibrar su diafragma, este movimiento busca mover su material interno generando fricción en sus partículas internas y produciendo tensión eléctrica en la superficie. Con el paso del tiempo se han usado diferentes materiales para su fabricación; los ejemplos más relevantes son: cuarzo, sal de Seignette o también conocida como sal de Rochelle y carbón entre otros [70].

7.1.4. Niveles de señal

Señales de Baja impedancia Low-Z (Señal microfónica).

Las señales de baja impedancia o señal de micrófono son básicamente como lo dice su nombre, son señales enviadas que tienen una resistencia muy pequeña y la mayoría de casos que envían este tipo de señal son los micrófonos; estos suelen tener una impedancia de 150Ω , pero los equipos que reciben esta señal poseen una impedancia de entrada mucho más grande, algo así como $1.5\text{ K}\Omega$. Lo cual se concluye que los pocos mili voltios que posee un micrófono, debe ser amplificado hasta el nivel adecuado para ser operado [64].

Señales de Alta impedancia Hi-Z (Señal de instrumento).

La señal de instrumento, es la señal que proporcional algunos instrumentos como las guitarras, bajos eléctricos y algunos teclados antiguos. Esta señal suele ser demasiado alta ya que puede poseer una impedancia de $15\text{ K}\Omega$ y aunque posee un voltaje muy parecido al voltaje de línea, no son compatibles por lo anteriormente hablado. En pocas palabras este tiempo de señal no es apropiada para la entrada ni de micrófono ni de línea [64].

Señales de línea.

La señal de línea se divide en dos tipos de señales; la señal de línea profesional y la señal de línea consumidor. Estos tipos de señales de línea se puede diferenciar por sus voltajes nominales los cuales son: El voltaje de señal de línea profesional es de +4dBu y tiene una tensión aplicada a una impedancia de más o menos 600Ω . Ahora la señal de línea consumidor tiene un voltaje nominal de -10 dBV pero su impedancia es de $1k\Omega$, este tipo de señal es desbalanceada. [64].

Señales de amplificación de cascos.

Señales de amplificación de cascos o de auriculares; para poder obtener una señal en los cascos, es necesario recibir una señal amplificada. Lo ideal, sería tener una impedancia de salida baja; la razón es tener una mayor intensidad de corriente cuando se conecte la resistencia que producirá los auriculares, que oscilan entre 8 a 300Ω , en estas condiciones la amplificación generará una baja distorsión y poca o ninguna cantidad de ruido. [64].

7.1.5. Cables.

Los cables básicamente tienen como características el transporte de una señal, los cables usualmente se fabrican con cobre ya que es un material conductor. Los cables también tienen como función la transmisión de luz como es el caso de cables ópticos, de electricidad, de datos y esfuerzo mecánico.

Existen dos formas en el transporte de señal eléctrica de audio por medio de cables, estas formas son conocidas como balanceada y no balanceada.[68].

No-balanceada: El transporte de la señal es por medio de dos cables conductores. Los conectores de señal no-balanceada tienen dos pines, como el RCA (Radio Corporation of America); la señal no balanceada es conocida también como señal no equilibrada.

Balanceada: El proceso que se realiza en el transporte de señal balanceada se lleva dos veces, en la cual una de ellas invierte la polaridad. A esto se lo conoce como el balanceado de una señal. En este caso se necesitan conectores de tres pines con un cable de tres líneas conductoras. En la *Figura 2* se exponen los tipos de cables existentes.

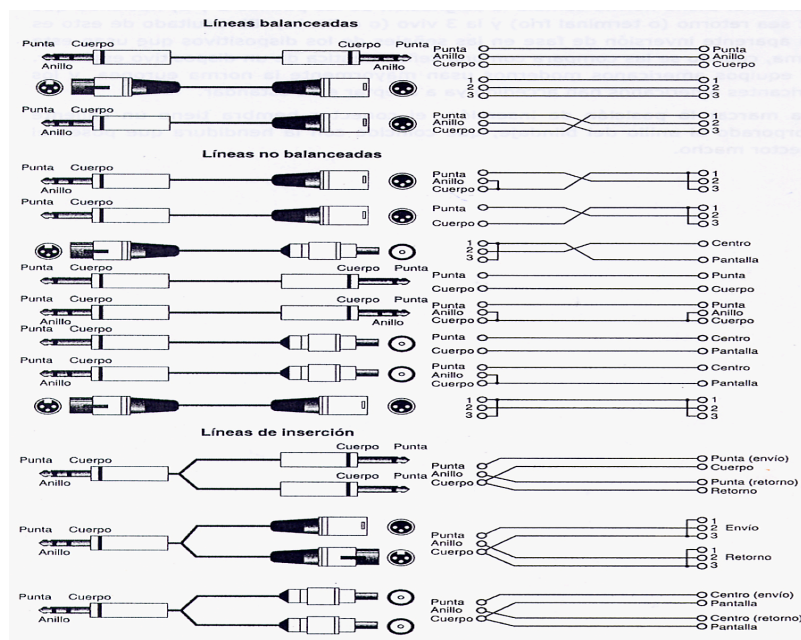


Figura 2. Combinación de cables balanceados y no balanceados. [68].

La diferencia que hay entre estos dos tipos de cables es simple, todo se basa en la línea extra que tienen los cables balanceados, ya que este cable tiene la función de retorno, mientras que los no balanceados solo tienen dos líneas, los cuales tienen como función pasar la señal activa, también conocida como: positiva o hot; por uno de los hilos, y en el otro hilo viaja la señal en contrafase que es la señal invertida, negativa o cold. [68].

7.1.6. Frecuencias Fundamentales en el espectro de las notas.

El sonido, es la sensación que siente el oído humano a una vibración o un conjunto de estas, que se propaga en un medio elástico. El tono que tiene un sonido es exclusivamente producido por la frecuencia a la cual vibra ese medio elástico, como ejemplo el aire. Hay también otros tipos de vibraciones como lo son las frecuencias concretas, uno de los ejemplos más claros es la vibración sinusoidal en estas frecuencias, lo cual genera tonos puros que al oído humano es semejante a un pitido.

En el sistema musical occidental, se ha acordado utilizar sólo unas frecuencias concretas, a las cuales llamamos notas. [41]

Para obtener la frecuencia de una nota por medio de una expresión matemática se tiene como base, una frecuencia de referencia, esta frecuencia por lo general suele ser La en la octava cuarta, a 440 Hertzios, esto se multiplica por la raíz duodécima de dos, la cual esta elevada al número de semitonos que separa nuestra frecuencia de referencia. Vamos a realizar un ejemplo: necesitamos hallar la frecuencia del Mi₅, los semitonos que separan esta nota de nuestra nota de referencia es de 7 semitonos, entonces para hallar la frecuencia se expresaría de la siguiente forma: $((440) \cdot (\sqrt[12]{2})^7)$ recordemos que la raíz duodécima de dos es igual a 1,0594630943....

Ahora, vamos hacer un ejemplo cuando se necesita hallar la frecuencia de una nota por debajo de la nota de referencia; supongamos que necesitamos hallar Do₄, los semitonos que los separan son 9, pero como están por debajo de la nota de referencia se toma como 9 negativo (-9); por lo que la expresión quedara así: $((440) \cdot (\sqrt[12]{2})^{-9})$.

Para poder entregar la expresión general, vamos a tener en cuenta que n es igual a la nota que queremos hallar, esta va de $n=1$ para Do; hasta $n=12$ para Sí, y debemos también tener en cuenta que o es la octava de la nota que queremos hallar, y tiene un rango de 0 a 10. Entonces la expresión quedara de esta manera (*Ecuación 2*): [41]

$$f(n, o) = 440 \cdot \left(\sqrt[12]{2} \right)^{(o-4) \cdot 12 + (n-10)}$$

Ecuación 2 Expresión para hallar frecuencia fundamental [41].

La expresión anterior puede tener algunos problemas a la hora de ser codificada en algunos lenguajes de programación, la razón puede ser que el lenguaje tenga carentes algunas funciones matemáticas, por ejemplo para hallar la raíz duodécima de dos, pero matemáticamente esa expresión se puede transformar como $2^{1/12}$, así la expresión queda (*Ecuación 3*):

$$f(n, o) = 440 \cdot 2^{(o-4) + \frac{(n-10)}{12}}$$

Ecuación 3 Expresión para hallar frecuencia fundamental [41].

Después de lo anterior, puede haber una posibilidad de que continúe sin poder codificarse, ya que algunos lenguajes no poseen de una función que permita potenciar cualquier tipo de exponente. Pero esto sigue sin ser un problema, porque todos los lenguajes disponen al menos de una función para hallar logaritmos y otra para hallar potencias de e (la base de los logaritmos naturales).

Una potencia cualquiera, por ejemplo x^y puede ser calculada con potencias de e y logaritmos naturales (en realidad, podría hacerse con cualquier base). Si tenemos una potencia $k=x^y$ podemos tomar logaritmos a ambos lados de la expresión y manipularla un poco. [41] Partimos de ésta expresión:

$\ln(k)=\ln(x^y)$ y la manipulamos un poco aplicando las reglas de manipulación de los logaritmos

$$\ln(k)=\ln(x)^y$$

$\ln(k)=y \cdot \ln(x)$ finalmente, tomamos potencias de base e en ambos lados de la ecuación

$e^{\ln(k)}=e^{y \cdot \ln(x)}$ Así que podemos concluir que

$$k=e^{y \cdot \ln(x)}$$

Utilizando esta expresión en nuestra fórmula de la frecuencia para quitarnos de en medio la potencia, finalmente queda de ésta manera (*Ecuación 4*):

$$f(n, o) = 440 \cdot e^{\left((o-4) + \frac{(n-10)}{12} \right) \cdot \ln 2}$$

Ecuación 4 Expresión para hallar frecuencia fundamental [41].

La tabla expuesta en la *Figura 2*, muestra el resultado de la *Ecuación 4*, la cual da la frecuencia fundamental o primer armónico, de las notas de la octava 0, en la afinación estándar.

Octava 0

Nota	Frecuencia
Do ₀	16,351598 Hz
Do# ₀	17,323914 Hz
Re ₀	18,354048 Hz
Re# ₀	19,445436 Hz
Mi ₀	20,601722 Hz
Fa ₀	21,826764 Hz
Fa# ₀	23,124651 Hz
Sol ₀	24,499715 Hz
Sol# ₀	25,956544 Hz
La ₀	27,500000 Hz
La# ₀	29,135235 Hz
Si ₀	30,867706 Hz

Figura 3 Notas fundamentales octava 0. [41].

En la *Figura 4* se muestran los resultados de la *Ecuación 4*, expone la frecuencia fundamental o primer armónico, de las notas de la octava 1, en la afinación estándar.

Octava 1

Nota	Frecuencia
Do ₁	32,703196 Hz
Do# ₁	34,647829 Hz
Re ₁	36,708096 Hz
Re# ₁	38,890873 Hz
Mi ₁	41,203445 Hz
Fa ₁	43,653529 Hz
Fa# ₁	46,249303 Hz
Sol ₁	48,999429 Hz
Sol# ₁	51,913087 Hz
La ₁	55,000000 Hz
La# ₁	58,270470 Hz
Si ₁	61,735413 Hz

Figura 4 Notas fundamentales octava 1. [41].

En la *Figura 5* se muestran los resultados de la *Ecuación 4*, expone la frecuencia fundamental o primer armónico, de las notas de la octava 2, en la afinación estándar.

Octava 2

Nota	Frecuencia
Do ₂	65,406391 Hz
Do# ₂	69,295658 Hz
Re ₂	73,416192 Hz
Re# ₂	77,781746 Hz
Mi ₂	82,406889 Hz
Fa ₂	87,307058 Hz
Fa# ₂	92,498606 Hz
Sol ₂	97,998859 Hz
Sol# ₂	103,826174 Hz
La ₂	110,000000 Hz
La# ₂	116,540940 Hz
Si ₂	123,470825 Hz

Figura 5 Notas fundamentales octava 2. [41].

En la *Figura 6* se muestran los resultados de la *Ecuación 4*, expone la frecuencia fundamental o primer armónico, de las notas de la octava 3, en la afinación estándar.

Octava 3

Nota	Frecuencia
Do ₃	130,812783 Hz
Do# ₃	138,591315 Hz
Re ₃	146,832384 Hz
Re# ₃	155,563492 Hz
Mi ₃	164,813778 Hz
Fa ₃	174,614116 Hz
Fa# ₃	184,997211 Hz
Sol ₃	195,997718 Hz
Sol# ₃	207,652349 Hz
La ₃	220,000000 Hz
La# ₃	233,081881 Hz
Si ₃	246,941651 Hz

Figura 6 Notas fundamentales octava 3. [41].

En la *Figura 7* se muestran los resultados de la *Ecuación 4*, expone la frecuencia fundamental o primer armónico, de las notas de la octava 4, en la afinación estándar.

Octava 4

Nota	Frecuencia
Do ₄	261,625565 Hz
Do# ₄	277,182631 Hz
Re ₄	293,664768 Hz
Re# ₄	311,126984 Hz
Mi ₄	329,627557 Hz
Fa ₄	349,228231 Hz
Fa# ₄	369,994423 Hz
Sol ₄	391,995436 Hz
Sol# ₄	415,304698 Hz
La ₄	440,000000 Hz
La# ₄	466,163762 Hz
Si ₄	493,883301 Hz

Figura 7 *Notas fundamentales octava 4. [41].*

7.1.7. Hardware.

Es o son los conjuntos de elementos físicos o materiales que constituye un sistema [50].

7.1.8. Sensores.

Hay diferentes tipos de sensores, estos se basan en su comportamiento físico, los sensores táctiles artificiales se clasifican de la siguiente manera:

Sensores piezoeléctricos, capacitivos, resistivos, celdas de carga, galgas extensiométricas, de Efecto Magnético, acelerómetros, biopotenciales, sensores basados en análisis de color, entre otros.

En la siguiente *Tabla 1* se muestra que tipo de sensores biológicos y artificiales miden cada uno de los estímulos característicos [46].

Estímulos y características	Sensor biológico	Sensor artificial
Fuerza: Medida de presión aplicada por una zona del dispositivo de agarre. Puede ser medida con algún dispositivo externo al robot.	Terminaciones nerviosas, Merkel, Ruffini, Paccini.	Galgas, Piezoeléctricos, Celdas de Carga.
Tacto: Medida continúa de las fuerzas ejercidas en un array. Proporciona información muy difusa sobre el contacto.	Terminaciones nerviosas, Merkel, Ruffini.	Resistivos, Capacitivos
Contacto: Medida de simples contactos, debida a las fuerzas ejercidas en pocos puntos de la superficie. Permite determinar distribución de las fuerzas de contacto e identificar características del objeto.	Meissner, Órgano terminal del pelo, Paccini.	Resistivos, Capacitivos, Magnetoresistor.
Deslizamiento: Medida de la mínima fuerza que hay que ejercer para agarrar un objeto produciendo la suficiente fricción para detenerlo entre las superficies de contacto.	Meissner, Órgano terminal del pelo.	Acelerómetros, Micrófonos, Rodadores.
Posición: Medida de la situación de los puntos más característicos del objeto en las superficies táctiles.	Meissner, Merkel.	Resistivos, Capacitivos, Ópticos.

Tabla 1 Sensores artificiales y biológicos. [46]

Sensor de proximidad: El sensor de proximidad es un transductor que tiene como fin detectar señales, objetos o alguna barrera que este cerca de este tipo de sensor; existen varias versiones de este sensor de proximidad todo depende del principio físico de cada sensor. Entre los más representativos son los siguientes: Sensor capacitivo, Sensor inductivo, Sensor infrarrojo, Sensor ultrasónico, Sensor magnético [71][48].

Sensor infrarrojo: Los sensores suelen tener un emisor que es el encargado de enviar la

señal infrarroja, mientras que el receptor recibe esta señal; los receptores mayormente son fototransistores o fotodiodos. El circuito de salida utiliza la señal del receptor para amplificarla y adaptarla a una salida que el sistema pueda entender. La señal enviada por el emisor se codifica para poder diferenciar esta señal de otras señales, esto quiere decir que se puede diferenciar o identificar varios sensores a la vez, esto es muy utilizado en la robótica en casos en que se necesita tener más de un emisor infrarrojo y solo se requiere tener un receptor [72][48].

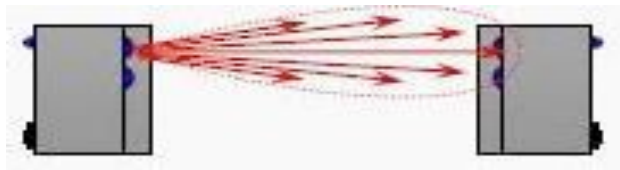


Figura 8 Sensor de proximidad infrarrojo. [48].

Estos sensores por lo general se utilizan como detectores de posición. Su principio es sencillo, el emisor genera un haz de luz que es recibido por una foto receptora o también puede ser recibido por un dispositivo reflejante. Al interrumpir el paso del haz de luz o al reflejarlo, provoca el cambio de estado de la salida de la fotocélula. Hay tres tipos de detección estos son: Sistema de detección de “barrera”, Sistema de detección “réflex”, Sistema de detección “auto réflex” [73].

Sensor ultrasónico: Los sensores ultrasónicos, expuestos en la *Figura 9*, tienen como función principal la detección de objetos a través de la emisión y reflexión de ondas acústicas. Su funcionamiento es simple, este emite o genera un pulso ultrasónico contra el objeto que se quiere detectar, y en el mismo momento que el objeto se detecta el

contador interno, que se generó al emitir el pulso se para. Este tiempo refiere a la distancia y también a los parámetros que se eligieron a la respuesta, este manda una señal eléctrica ya sea digital o análoga. [48].

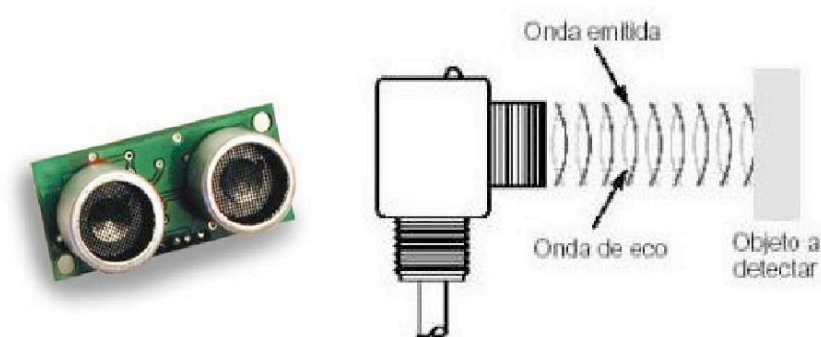


Figura 9 Sensor de proximidad ultrasónico. [48].

Sensores magnéticos: Los sensores de proximidad magnéticos que se muestran en la *Figura 10*, usualmente se caracterizan por tener una vía de comunicación entre su emisor y su receptor en distancias muy largas, su principio es la detección de ondas magnéticas o corrientes eléctricas en los objetos, uno de los principales es el interruptor de reed, básicamente este se usa como un sensor de detección de un elemento móvil sin la necesidad de entrar en la carrera de este elemento, es decir lo puede detectar sin ningún contacto directo.[48].



Figura 10 Sensor de proximidad magnético. [48].

Sensores de fuerza o presión: Los sensores de fuerza, que se muestran en la *Figura 11* son sensores el cual su principio es medir la carga de tracción y compresión estática y dinámica, sin ningún desplazamiento. Usualmente su utilización son para la medición de fuerza en bancos de ensayos y en tecnologías de automatización.[48].

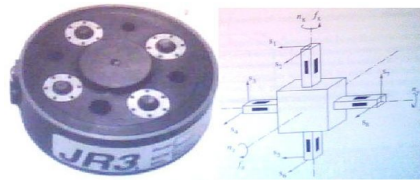


Figura 11 Sensor de fuerza y componentes. [48].

Sensor de fuerza resistivo (fsr): En la *Figura 12* se observa un sensor de fuerza resistivo FSR este sensor esta hecho por una película de polímero la cual se encarga de que la resistencia disminuya cada vez que una fuerza es aplicada a la superficie activa de este sensor. Su sensibilidad a la fuerza está normalizada para el toque humano en dispositivos electrónicos. Como se explicaba anteriormente la resistencia varia dependiendo de la presión que se ejerza en su área sensible, posee un área redonda de 0.5" de diámetro, pero en el momento que no exista presión la resistencia del sensor será mayor a $1M\Omega$. Este FSR puede sentir la fuerza aplicada dentro de un rango de 100 g y 10 kg [48].



Figura 12 Sensor de fuerza resistivo FSR redondo [48].

Sensores piezoeléctricos: Los sensores piezoeléctricos están constituidos por materiales cerámicos o cristales iónicos que generan una pequeña cantidad de energía eléctrica cuando son deformados. Estos sensores son utilizados para mediciones de fuerzas y presiones aplicadas en un corto período de tiempo. [48][72].

Flexiforce cuadrado: Este sensor básicamente es una resistencia sensible a la fuerza aplicada en su superficie, su superficie es cuadrada con unas medidas de 1,75 x 1.5 pulgadas, El FSR expuesto en la *Figura 13* varía su resistencia siempre y cuando esté recibiendo alguna presión en su área de detección. Cuanto más es la fuerza, más baja es la resistencia. Este FSR puede detectar la fuerza aplicada entre 100 g a 10 kg [48].

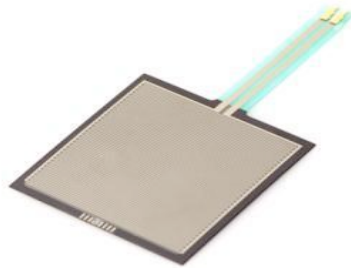


Figura 13 Sensor de fuerza resistivo FSR Cuadrado. [48].

Flexiforce rectangular: Como lo indica su nombre, este sensor se encarga de medir la flexión aplicada en él mismo, este sensor aumenta o disminuye su resistencia según sea la fuerza aplicada, la resistencia aumenta cuando la flexión es mayor y los dientes de metal están en la parte externa, lo podemos asociar de la siguiente manera: [48]. Resistencia recto (sin flexionar): ~9000 Ohm, Resistencia doblado en 90 grados: ~14000 Ohm, Resistencia doblado en 180 grados: ~22000 Ohm.

7.1.9. Preamplificadores.

Es un tipo de amplificador electrónico dentro de la cadena de audio, enfocado en realzar o aumentar el voltaje, lo cual difiere a un amplificador de potencia. El preamp, al ser un amplificador su principal característica es actuar sobre la tensión de la señal entrante, dando como resultado que la salida tendrá nivel de línea y estará estandarizado a los 0 dB. Cuando el preamplificador es usado en un micrófono no solo eleva su voltaje que va en el orden de los mil voltios a un voltio aproximadamente, sino que también trabaja en la respuesta en frecuencia, lo que quiere decir que la ganancia varía con la frecuencia, y este da un sonido característico al dispositivo que lo use. El preamplificador de audio también es el encargado de acoplar la impedancia entre el dispositivo de entrada y el dispositivo de salida. [75]

FET: Los transistores de efecto de campo (FET), Figura 14 contienen 3 puntos o terminales estas son conocidas como: Fuente (Source), Drenaje(Drain) y Puerta (Gate). Sus características son el trabajo de control de la corriente entre la fuente y el drenaje en el campo eléctrico, por medio del voltaje que se aplica en la puerta. La puerta no maneja corriente exceptuando una corriente de fuga. El FET tiene como característica principal el gran nivel de impedancia que tiene en la entrada, esto puede llegar hasta los $10\text{ M}\Omega$, como consecuencia se puede aplicar en muchas cosas entre ellos están: llaves analógicas, amplificadores de muy alta impedancia de entrada, también suelen ser usados como resistencias controladas por voltaje y para fuentes de corriente.[76].

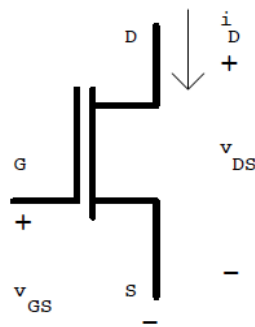


Figura 14 Convención de signos. [76].

BJT:BJT son las siglas que provienen de Bipolar Junction transistor. El término bipolar se refiere a que los dos tipos de portadores los cuales son: Electrones y huecos intervienen en la conducción de la corriente. El termino junction (unión) hace referencia a la estructura del dispositivo. [99]

El transistor posee tres capas, las cuales se pueden dividir de la siguiente manera: una zona con material tipo n en la mitad de dos zonas tipo p, este transistor tiene el nombre de transistor pnp, el cual podemos observar en la (Figura 15), también puede tener una zona con material tipo p entre dos zonas tipo n, en el cual es llamado transistor npn, como se puede observar en la (Figura 16). [99]

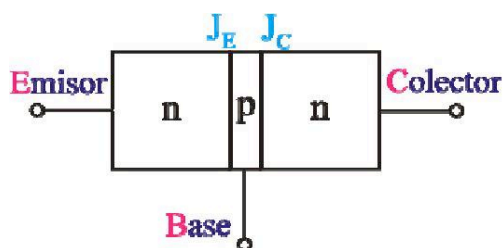


Figura 15 Estructura transistor BJT npn.[99]

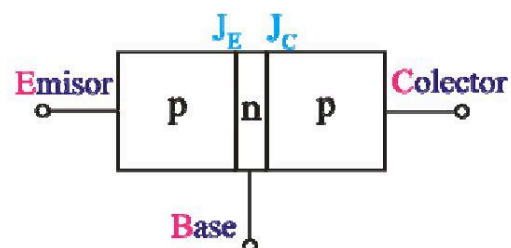


Figura 16 Estructura transistor BJT pnp.[99]

La zona central se denomina base, y las laterales emisor y colector. Cada una de las zonas consta de un terminal por donde extraer las corrientes. Estos terminales se representan por la inicial del nombre de la zona respectiva: E (emitter), B (base) y C (colector). [99]

La zona donde está el emisor es la zona más dopada de las tres zonas anteriormente explicadas, ya que esta es la encargada de emitir portadores mayoritarios directo a la base, ya sean huecos al transistor pnp o electrones en el caso del transistor npn.

La base tiene un nivel de dopado muy inferior al de la zona del emisor, esta posee una capa muy pequeña al ser comparada con las del exterior, ya que básicamente su función es dejar pasar la cantidad de portadores posibles de la zona del emisor a la zona del colector. La zona del colector como su nombre lo muestra, es la zona encargada de colectar o recoger los portadores que envía el emisor y son capaces de pasar por la zona de base. Es la zona con un nivel de dopado inferior de las tres. [99]

El transistor como amplificador: Los transistores como lo explicamos anteriormente pueden servir como un amplificador en un caso intermedio entre corte y saturación, esto puede ser ocasionado cuando en la base la corriente no es tan pequeña para cortar la corriente en los otros terminales, pero a su vez no es lo suficientemente grande para permitirle pasarla completamente. En ese caso el transistor funciona como un amplificador que proporciona entre el colector y el emisor un múltiplo de la corriente, que pasa por la base. Cuando el transistor se comporta como un amplificador y conduce parcialmente decimos que trabaja en la zona activa.[99].

Para poder entender los transistores como amplificadores, en la *Figura 17* observamos un

circuito amplificador BJT, con sus respectivas características las cuales son: La ecuación característica, La zona de amplificación, zonas de conmutación las cuales (zona de corte y zona de saturación) y punto Q.

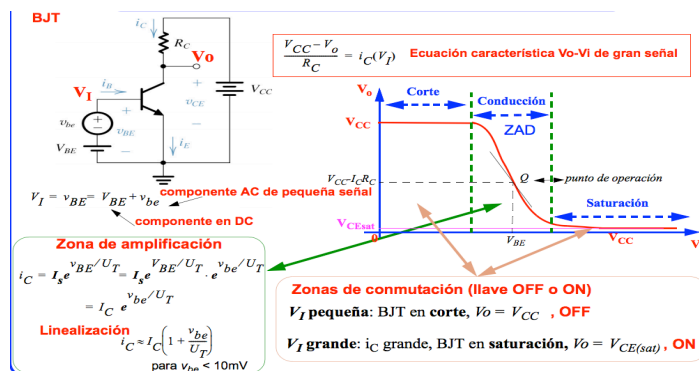


Figura 17 Circuito amplificador BJT[99]

Como en el anterior ejemplo BJT, también podemos observar en la Figura 18 un circuito amplificado MOSFET con sus características: Ecuación característica, Zona de amplificación, punto Q (punto de operación).

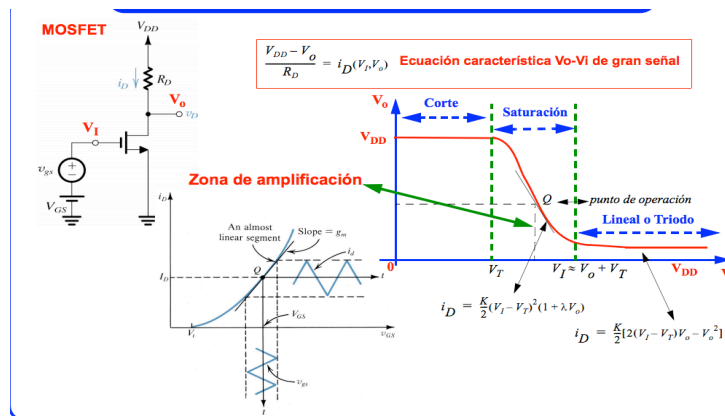


Figura 18 Circuito amplificador MOSFET [99]

7.1.10. Sistemas embebidos.

Los sistemas embebidos básicamente se usan como un sistema de electrónica programable, y usualmente son empleados para desarrollar aplicaciones específicas en tiempo real. Su nombre viene de la palabra empotrado, eso quiere decir que hace parte de un sistema más grande, como puede ser una maquina o procesos.

Encontramos sistemas embebidos en productos tan diversos como electrodomésticos, todos tipo de dispositivos para comunicaciones y cómputo, automotores, etc.

Los sistemas embebidos integran recursos de hardware y software.

El hardware está constituido por: Un núcleo que consiste en una o más CPUs (Central Processing Unit) en alguno de los siguientes formatos: Microprocesador, Micro controlador, DSP de punto fijo o flotante y/o algún dispositivo "custom" como FPGA.

La electrónica de acondicionamiento de señal que permite la interacción entre la CPU y el entorno.

A la hora de programar el software hay una gran variedad de formas para desarrollar este; ya que puede ser por medio del lenguaje implementado por el ensamblador de la CPU del sistema, o también puede ser un lenguaje de programación de mayor nivel usado con un compilador apropiado. Hay distintos entornos para el desarrollo de compilación, carga, edición y depuración del software. Se puede decir que en los sistemas embebidos hay una pequeña brecha entre el hardware y el software, lo que conlleva a desarrollar diferentes procesos de muy alta calidad, los cuales son basados en el proceso de señales

sin la necesidad de tocar el circuito electrónico, sin embargo al tener un conocimiento de este, puede dar mayor eficiencia. [78].

En la *Figura 19* se exponen los componentes que deben estar presentes en un sistema embebido.

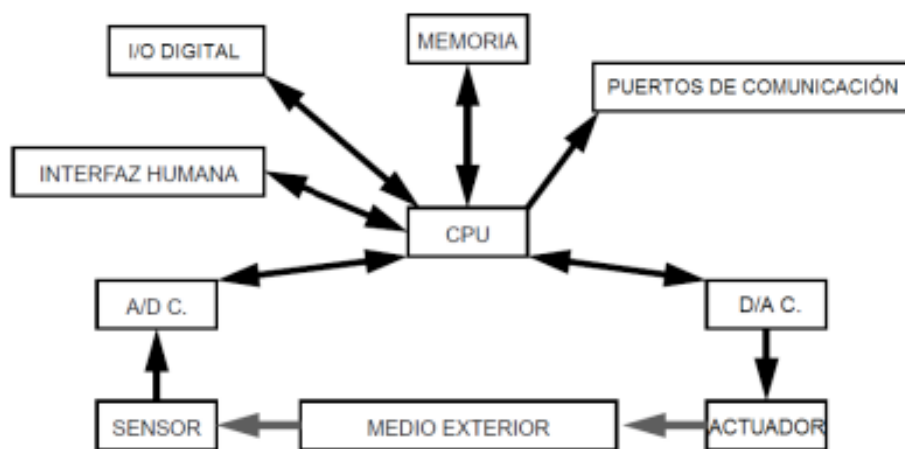


Figura 19 Arquitectura de un sistema embebido. [78].

Seeeduino.

Seeeduino es un diseño por la empresa llamada seeed. El esquema de Seeeduino es basado con el de Arduino Diecimilia, pero con ciertas características diferentes, ya que han añadido y cambiado algunos elementos para mejorar su rendimiento. Los fabricantes de esta placa dicen que todos los elementos compatibles para Arduino también servirán para Seeeduino, ya sea hardware o software. La medida de esta placa es de 71 x 53 mm, y

su procesamiento depende de un micro controlador ATmega168. Al igual que Arduino puede ser alimentada externamente o por USB, tienes pines analógicos y digitales. [4]

En la (Tabla 2) se puede observar todas las especificaciones.

Microcontrolador	ATmega168
Voltaje operativo	5V/3.3V
Voltaje de entrada(recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	8
Corriente continua para pines E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash	16 KB (ATmega168) de los cuales 2 KB son para el bootloader
SRAM	1 KB (ATmega168)
EEPROM	512 bytes (ATmega168)
Velocidad del reloj	16 MHz

Tabla 2 Componentes del Seeduino [4].

Arduino.

Arduino es una plataforma electrónica de código abierto basada en hardware y software, esta plataforma se diferencia a otras por su fácil y flexible forma de usar.

El micro controlador que posee la placa es programable en lenguaje (Arduino Programming Language), el cual es basado en wiring; mientras que el entorno de desarrollo de Arduino se basa en processing.

Hay distintos tipos de placas de Arduino, y vienen para ensamblar a mano o ya pre ensambladas; El software se puede conseguir por medio de la página y su descarga es

gratuita. El primer prototipo, mostrado en la *Figura 20*, fue desarrollado en el instituto IVRAE pero aún no se llamaba Arduino.[24]

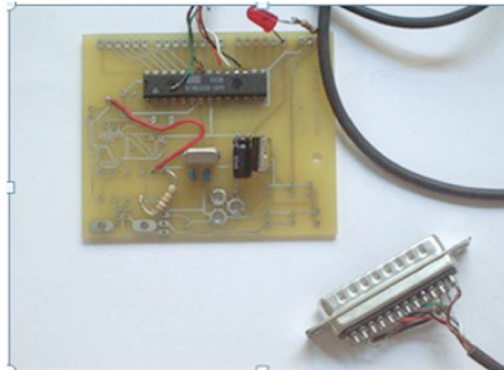


Figura 20 *Prototipo de Arduino.* [24].

Las placas Arduino tienen muchas ventajas comparadas con otros micro controladores, una de ellas es que son económicas. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino pre ensamblados cuestan menos de 50 dólares. Otra de las ventajas es que Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los sistemas micro controladores están limitados a Windows.

El software Arduino como anteriormente lo dijimos es una herramienta de código abierto. El lenguaje puede ser expandido mediante librerías de C++ y de igual forma puede hacer el salto de programación de Arduino al lenguaje de AVR-C, se pueden añadir códigos AVR-C directamente en los sketch de Arduino.

En la *Figura 21*, podemos observar una de las placas de Arduino, esta es conocida como Arduino Uno, es constituida por un micro controlador ATMEGA328 de Atmel; y sus planos están publicados bajo la licencia de Creative Commons, lo que da la posibilidad de crear nuestra propia versión del módulo. [31].[32]

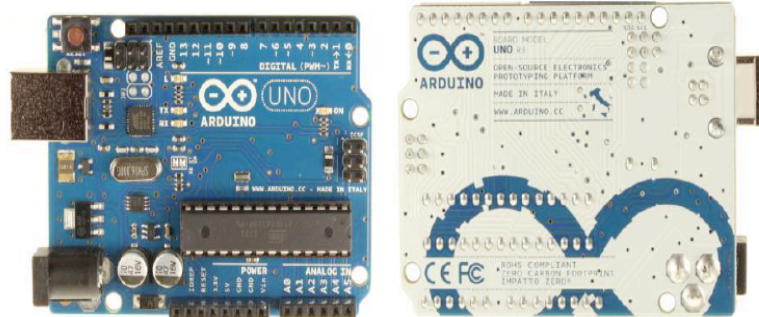


Figura 21 Frontal y reverso de una placa de Arduino uno. [4].

En la *Tabla 3*, podemos observar las especificaciones de la placa Arduino Uno. Su tamaño es de 75 x 53 mm, puede ser alimentada mediante USB o alimentación externa y contiene pines tanto analógicos como digitales. [4].

Microcontrolador	ATmega328
Voltaje operativo	5V
Voltaje de entrada(recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente continua para pines E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son para el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz

Tabla 3 Componentes del Arduino Uno.[4]

A continuación se muestra la *Figura 22*, en la cual podemos observar cada uno de los elementos que constituye la placa de Arduino Uno.

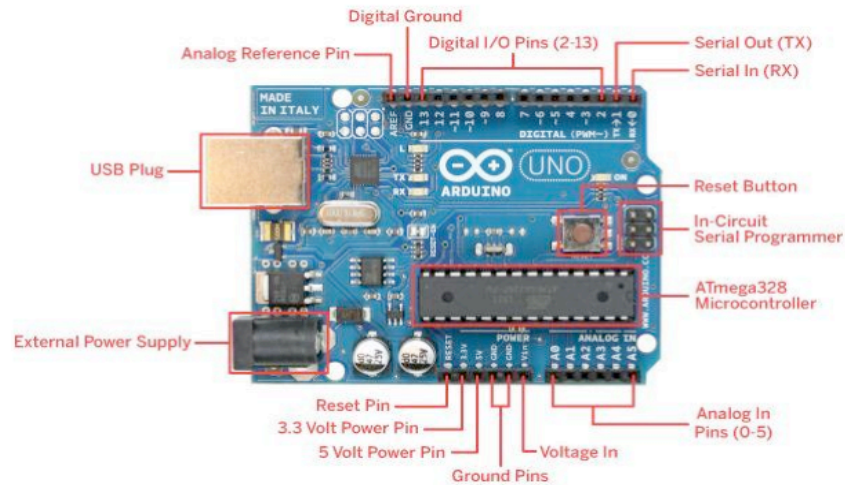


Figura 22 Elementos de la placa de Arduino Uno. [4].

Referencia para pines analógicos (AREF): Tensión de referencia para entradas analógicas.

Pines de tierra (GND): Masa del circuito para pines, es decir es la tensión de referencia de 0V.

Pines digitales de entrada y salida: Hay 14 pines digitales que pueden utilizarse como entrada o salida. Estos pines tienen la capacidad de trabajar a 5 V, y cada pin envía y recibe como máximo 40mA.

Serie: 0(RX) y 1(TX). Utilizados para recibir (RX) y transmitir (TX) datos serie. Están directamente conectados a los pines serie del micro controlador. Utilizando estos pines podremos conectarnos con otras placas.

Interrupciones externas: 2 y 3. Estos pines pueden ser configurados para activar interrupciones.

PWM: 3, 5, 6, 9, 10 y 11. Proporcionan una salida de 8 bits en modo PWM.

SPI: 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI.

LED: 13. Este pin está conectado con un led de la placa. Cuando se le asigne un valor HIGH se encenderá, en cambio si lo dejamos en LOW estará apagado.

Conector USB: Existen varios tipos de conectores USB, en concreto esta placa utiliza el tipo B hembra.

Botón Reset: Utilizando este botón podremos reiniciar la ejecución del código del micro controlador.

Micro controlador ATmega328: El micro-controlador es el elemento con mayor importancia de todos, ya que es en donde se instala y se ejecuta el código. 5v es su voltaje operativo, pero se recomienda que la entrada oscile entre 7 a 12 voltios y como límite establecido es de 20V. Contiene 14 pines digitales de entrada y salida, Dispone de 32KB de memoria flash (de los cuales 512 bytes son utilizados por el bootloader). En El bootloader será el encargado de preparar el micro-controlador para que pueda ejecutar nuestro programa. Dispone de un reloj de 16 MHz y 2KB de memoria RAM.

Fuente de alimentación externa: La placa puede ser alimentada mediante corriente DC suministrada por el conector jack que podrá recibir entre 7 y 12V.

Pin de Reset: Podemos imitar el funcionamiento del botón reset suministrando un valor LOW (0V) para reiniciar el micro-controlador.

Pin de 3.3V: Desde aquí podremos suministrar 3.3 V a los dispositivos que lo necesiten

con una corriente máxima de 50 mA. Es generada gracias al chip FTDI integrado en la placa.

Pin de 5V: Este pin saca una tensión de 5 V del regulador de la placa. El regulador es necesario puesto que puede ser alimentada con distintos voltajes.

Pin de Vin: Es el voltaje de entrada cuando se usa una fuente de alimentación externa (no tiene en cuenta la conexión USB). Se puede proporcionar voltaje a la placa a través de este pin, o en caso de que se esté utilizando una fuente de alimentación externa tomar el valor que está siendo suministrado.

Pines analógicos: Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales pero su uso requiere de una lógica levemente mayor. Más adelante se comentará el uso de un termistor analógico.[41]

Arduino Mega 2560: Esta placa como base tiene un micro controlador ATMega2560, también contiene 54 pines digitales, 16 entradas análogas, 4 puertos seriales, un cristal oscilador de 16 MHz, conexión USB, alimentación externa, un cabezal ICSP y el botón de reset. Otras de las tantas características de esta placa son las siguientes: Voltaje de entrada que oscila entre 7 a 12 voltios, tiene una memoria flash de 265KB y una velocidad de reloj de 16 MHz.[24].

En la *Figura 23*, se puede observar como es visualmente la placa de Arduino mega 2560.



Figura 23 Placa de Arduino Mega 2560. [24].

Software Arduino.

Arduino, como lo hemos hablado anteriormente contiene una plataforma de lenguaje propio que se basa en C y C++, aunque también existen diferentes tipos de lenguajes que se usan al programar en esta plataforma como : Java, Python, Processing, Matlab, Visual Basic entre otros. Lo anterior es posible, ya que Arduino posee una comunicación de transmisión de datos en serie, al igual que los lenguajes previamente mencionados. También existe la posibilidad de trabajar con lenguajes que no soportan el formato en serie, por medio de software intermediario, que traduce los mensajes y permite la comunicación. [24][31].

El módulo Bluetooth jy-mcu:

Este módulo (*Figura 24*) trabaja por medio de comunicación serial, contiene 4 pines para conectarse con el Arduino; estos pines son los siguientes: VCC (5V), GND (0V), TX (Tx Serial), RX (Rx Serial). Básicamente este módulo trabaja como un puente transparente entre el Arduino y el dispositivo móvil conectado a este módulo, así la información que

reciba del móvil la reenviara al Arduino mediante su conexión Serial, de igual forma la información que reciba del Arduino (mediante su Puerto Serial) será enviada al Móvil [27].



Figura 24 Modulo Bluetooth [27].

Este módulo soporta los siguientes comandos AT con los cuales se le puede dar algunas configuraciones de ser requerido, en la siguiente *Tabla 4* a se muestra el resumen de los comandos soportados por este módulo:

Comando	Descripción	Ejemplo
AT+VERSION	Retorna la versión del Modulo	
AT+BAUDx	Configura la velocidad de trasmisión del módulo según el valor de x, si x=1 =>1200, x=2 => 2400, x=3 => 4800, x=4 => 9600, x=5 =>19200.	AT+BAUD4 Configura la velocidad a 9600 baud rate
AT+NAMEx	Configura el nombre con el que se visualizara el modulo, soporta hasta 20 caracteres	AT+NAMECreatividad Configura el nombre del módulo a Creatividad
AT+PINxxxx	Configura el Pin de Acceso al modulo (pairing)	AT+PIN1234 Configura el password a 1234, el cual es el default

Tabla 4 Comandos AT [27].

7.1.11. Interacción.

Básicamente la interacción es el intercambio de información entre dos integrantes activos o más, dependiendo el caso. Joshua noble dice que la interacción ocurre a través de señales que son enviadas por los diferentes sistemas participantes en la interacción [18]. Para entender el significado de interacción, debemos también entender que significa un sistema interactivo y qué lo comprende.

Sistema Interactivo: Sistema que interrelaciona al hombre con la máquina, el cual el receptor puede ser también el emisor y donde el inter actuador posee una gran cantidad de responsabilidad a la hora de la manipulación del contenido audiovisual o de otra área.

7.1.12. Tipos de Interacción

Interacción Externa: Consiste en la interfaz humano-maquina, así como en las formas ofrecidas por el entorno, cuyos datos son procesados por ordenador mediante diferentes interfaces. [58]

7.1.13. Estilos de Interacción.

Los estilos de interacción son diferentes modos de interactuar, hay 4 tipos que se pueden definir como los primordiales y son los siguientes:

Manipulación Directa: La Interacción de manipulación directa básicamente es un estilo de interacción hombre – máquina que se caracteriza por la representación y la visualización de objetos y acciones sobre estos de manera directa y en gran parte muy realista, permitiendo la manipulación y simulación.

Selección de Menús: Esta interacción básicamente se caracteriza como un conjunto de opciones que el usuario tiene para elegir en una lista y también se pueden seleccionar,

dependiendo cual es el más adecuado. Para desarrollar y diseñar menús deben existir algunos parámetros estos son: No ser muy largos, debe tener al lado del texto una clave para selección rápida. [80]

Lenguaje de Órdenes: Los lenguajes de órdenes se caracterizan por utilizar un conjunto de órdenes básica, que le dan funciones a los sistemas operativos, cada conjunto es específico dependiendo de los sistemas que operan. El lenguaje de órdenes es utilizado por medio de botones o interpretación de gestos.

Lenguaje Natural: Estilo de interacción donde la máquina y el humano se interrelacionan de una manera natural, interactuando entre ellos sin obstáculos de relación. [81]

7.1.14. Interfaz.

La interfaz se refiere a la comunicación e interacción que hay entre entidades o individuos, hay diferentes tipos de interfaces ya sean tangibles o intangibles. La Interfaz básicamente hace el trabajo de traductor entre hombre y máquina, ya que los dos sistemas hablan distintos lenguajes, pero hace que sea más fácil la comunicación que puede haber entre sistemas diferentes. [82].

7.1.15. Interfaz de usuario.

Cuando la palabra interfaz está junto a la palabra usuario, es un concepto que nos da a entender que la interfaz está diseñada para que sea usada siempre por el hombre, esto conlleva que la relación hombre-máquina es más estrecha y por ende más eficiente.

Mientras la tecnología iba en aumento, el hombre y la maquina tenía un lenguaje que se volvía más complejo con el tiempo, lo que llevó a buscar que este tipo de comunicación

mejorara, y se logró aplicando interfaces de lenguajes que utilizaban ambos bandos, esto fue un logro de las interfaces de usuario dando como resultado una comunicación óptima entre estas dos partes. Es importante entender que la interfaz de usuario es un puente entre los lenguajes que buscan manipular lo que está entre la tarea a desarrollar y el usuario. [83]

7.1.16. Tipos de Interfaz de Usuario.

Hay diferentes tipos de interfaces de usuarios creadas con el paso del tiempo, las más representativas son las siguientes:

CLI: Son las siglas de “Command line interface” que en español traduce interfaz de línea de comandos. Esto básicamente es un método que permite dar instrucciones por medio de una línea de texto simple a un programa. Este estilo de interfaz se emplea interactivamente, donde se escriben instrucciones en una entrada de texto o de manera automatizada, como leyendo comandos desde un archivo de scripts. Existe desde los comienzos de la computación, y se utilizan para diferentes programas para sistemas operativos y hardware. Su uso es fácil ya que se escribe el comando dando la orden y al ejecutar pasa a la siguiente línea, utilizando la tecla enter, un ejemplo claro se muestra en la *Figura 25*. [83]

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\diseno>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 50F6-31D7

Directorio de C:\Documents and Settings\diseno

11/12/2007  09:23    <DIR>          .
11/12/2007  09:23    <DIR>          ..
24/04/2007  09:38    <DIR>          Application Data
29/11/2006  18:37    <DIR>          Bluetooth Software
16/01/2007  22:37    <DIR>          Configuracin local
03/07/2007  19:36                4.368 ErrorLog.txt
30/12/2007  19:40    <DIR>          Escritorio
30/12/2007  10:24    <DIR>          Favoritos
24/07/2007  17:23                0 ICUBEX.CFG
01/09/2007  02:07                12 intlname.ols
13/10/2006  20:04    <DIR>          Menú Inicio
29/09/2007  10:54    <DIR>          Mis documentos
17/12/2006  19:19    <DIR>          Phone Browser
18/01/2007  00:13    <DIR>          WINDOWS
                3 archivos      4.380 bytes
                11 dirs 17.960.566.784 bytes libres

C:\Documents and Settings\diseno>

```

Figura 25 Ejemplo de CLI. [24].

GUI: Son las siglas en ingles de “Graphical user interface” o en español Interfaz gráfica de usuario. Básicamente su manera de uso es simple, en el cual el usuario interactúan por medio de un conjunto de gráficos e imágenes llamado “Entorno Gráfico” en un programa informático; estos conjuntos de imágenes so la representación de la información y acciones que hay en la interfaz. La idea de esta interface es dar al usuario un entorno agradable y sencillo entre la comunicación que hay con la máquina. En la *Figura 26* podemos observar un ejemplo de GUI implementada en Linux. [84].



Figura 26 Ejemplo GUI Fuente Internet. Interfaz gráfica tridimensional Compiz en GNU/Linux [85].

TUI: Tangible User Interface o en español Interfaz de usuario Tangible. Es una interface en el cual el usuario interactúa con la información digital a través de un ambiente físico palpable [84]. Hiroshi ishii *Figura 27* es el pionero en TUI.

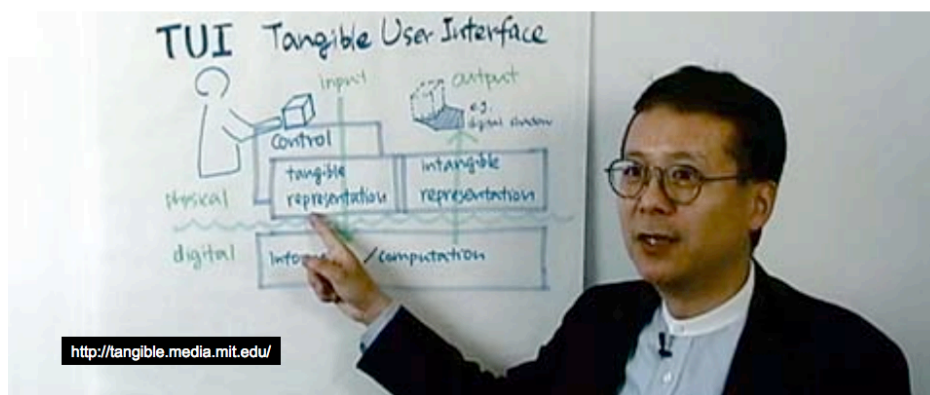


Figura 27 Hiroshi Ishii. [84].

En la *Figura 28* podemos observar una interfaz TUI llamado MIT TUI, de tangible Media Group.



Figura 28 Tangible Media Group, MIT TUI [85].

NUI: Natural User Interface o en español Interfaz de Usuario Natural. Básicamente es una interfaz que no necesita de mandos externos como si lo necesita el GUI; ya sea con un ratón, un teclado o un touchpad, etc. En su lugar, se hace uso de movimientos gestuales ya sean con las manos o el cuerpo, Cuando se usan pantallas capacitivas multi-táctiles su manejo es por medio de la yema de los dedos en uno o varios contactos, también hay control de sistemas operativos por medio de voz humana y control cercano a la pantalla, pero sin la necesidad de tocarla. En la *Figura 29* podemos ver la muestra de un NUI, llamado music bottles.[84]

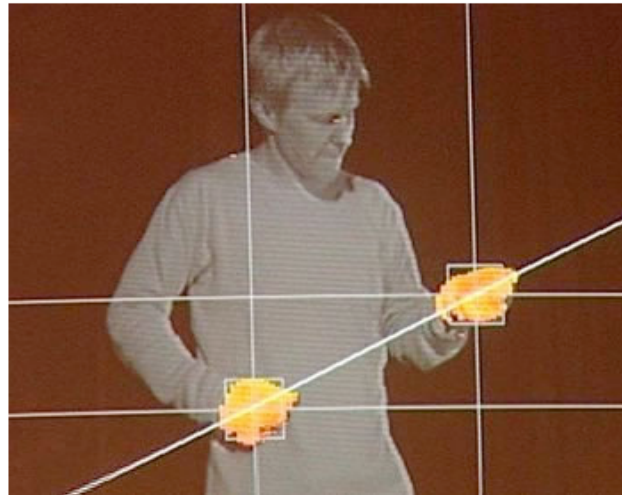


Figura 29 *MUSIC BOTTLES* [18].

7.1.17. Programación orientada a objetos.

Más conocido como POO, es un paradigma de la programación, es otra forma de pensar, que busca la innovación en su modo de obtención de resultados. La finalidad de esto es la manipulación de datos de entrada por los objetos, para una posterior entrega de datos de específicos de salida. [48].

Lenguajes de programación.

Este lenguaje fue diseñado y llevado a cabo para determinar procesos que serán realizados por máquinas un ejemplo claro es la computadora; las cuales se pueden emplear para la creación de programas que dominen y controlen el comportamiento lógico como el físico de la máquina, para plasmar conceptos, ideas y modelos por medio de algoritmos[86].

Basic: Basic posee diversas posibilidades, las más destacadas son las siguientes: NS Basic/Palm, con esta herramienta es posible desarrollar todo tipo de aplicaciones para Palm dentro de un entorno de desarrollo visual. Mobile Visual Basic, permite extender la funcionalidad de Visual Basic, ya que al instalarlo se añade una opción para poder generar proyectos del tipo AppForge, que incorpora además de los controles estándares una serie de elementos específicos para el desarrollo de aplicaciones para Palm [1].

PRC-Tools: Al igual que el anterior es un lenguaje para Palm, como característica importante es un producto de código abierto que consiste en un conjunto de herramientas que hacen uso de una versión SDK, permite el desarrollo de aplicaciones empleando el compilador GCC, ya sea usando C o C++.[1].

Visual Studio .NET: Visual es un entorno el cual dispone de una tecnología muy parecida a Java Virtual Machine (JVM), esta tecnología se conoce como Framework. El framework también está disponible en una versión más ligera llamada .NET Compact Framework, y es exclusiva para los dispositivos móviles. [1].

App Inventor: Es una aplicación originalmente desarrollada por Google y mantenida ahora por el Instituto de Tecnología de Massachusetts. Tiene como función principal la programación de aplicaciones para Android, pero no necesita tener un conocimiento previo de programación con SDK de Android. Usa una interfaz gráfica amigable que trabaja por medio de bloque de código [15].

Windows Phone: Su código se programa en C#, un lenguaje de programación orientado a objetos desarrollado por Microsoft. Su lenguaje es llamado XAML, el cual se basa en XML y se analiza por Framework. [18]

Eclipse: Eclipse es un programa desarrollado por la empresa IBM, pero en este momento es dirigido por Eclipse Foundation, la cual es una entidad sin ánimo de lucro, es gratuito y de código abierto. Se programa por medio de lenguaje JAVA. Este lenguaje orientado a objetos está pensado para poder ser compilado una vez y ejecutado en una gran variedad de dispositivos en los que debe estar instalada la denominada "Máquina virtual Java", y para el aspecto visual Eclipse utiliza el lenguaje de marcado XML [18].

7.1.18. Protocolo de comunicación.

El protocolo, como su nombre lo indica es un conjunto de reglas que dan la posibilidad a diferentes elementos participantes en un sistema que se comuniquen entre ellos, ya sea enviando datos, recibiendo datos o las dos a la vez. [88]

Los protocolos de comunicación tiene como función dar los criterios necesarios para determinar la sintaxis y semántica que se empleará en el proceso comunicativo, incluso habrán pautas puestas que permitirán la recuperación de los datos perdidos en el intercambio de datos. [48].

Guiado.

Los medios guiados son conocidos por ser aquellos que utilizan medios palpables para la transmisión de los datos, un ejemplo claro son los cables ya que transmiten datos, impulsos eléctricos, lumínicos etc. En la *Tabla 5* podemos observar diferentes tipos de medios guiados y sus características.

Medio de Transmisión	Razón de datos total	Ancho de Banda	Separación entre repetidores
Par Trenzado	4 Mbps	3 Mhz	2 a 10 km
Cable Coaxial	500 Mbps	350MHz	1 a 10 km
Fibra Óptica	2Gbps	2GHz	10 a 100 km

Tabla 5 Medios de transmisión no guiados. [90][48]

No Guiado.

Los medios no guiados tienen como característica, la transmisión de datos o de información sin necesidad de cables o de medios físicos, en cambio hacen transmisión de la información por ondas electromagnéticas. Hay dos tipos de medios no guiados los cuales son: Direccionales y Omnidireccionales. [48]

Direccional: Básicamente la información se envía por medio de un haz de luz en la cual concentra toda la energía que se emite en una sola dirección, por consecuencia los emisores y receptores deben estar alineados entre sí.[48]

Omnidireccional: La energía es enviada en múltiples direcciones, por lo que pueden recibir los datos dos o más antenas a la vez. [48]

En los medios no guiados direccionados se suele usar microondas, que se caracteriza por su alta frecuencia, mientras que para enlaces con varios receptores se utilizan las ondas de radio, que se caracterizan por su baja frecuencia. [48]

7.1.19. Protocolo de comunicación inalámbrica.

Este es un medio que se caracteriza por enviar datos sin necesidad de cableado o medios físicos externos, la transmisión de datos usa puertos, al igual que su recepción. Tiene

ventajas importantes ya que elimina la necesidad de los cables y sus costos bajan por la misma razón; pero una de sus desventajas es su seguridad ya que puede haber ajenos filtrándose a la señal [48].

Wi-Fi.

Wi-Fi viene de la sigla Wireless Fidelity, que en español significa fidelidad inalámbrica. Al principio se creó para obtener acceso a redes locales, pero con el paso del tiempo se usan para establecer conexiones a internet. Wi-fi fue creada por la compañía Wi-fi Alliance la cual se hace cargo de certificar que los equipos cumplan con la normativa vigente, (IEE 802.11). Este protocolo fue creado con el fin de establecer conexiones inalámbricas compatibles entre diferentes dispositivos. [92][48].

Bluetooth

Bluetooth tiene como características el envío y recepción de datos de corto alcance, se desarrolló para evitar el cableado entre dispositivos. La tecnología de Bluetooth permite la conexión entre dos o más dispositivos en tiempo real, evita interferencia y provee seguridad en el envío de datos. El radio Bluetooth es un microchip que opera en una banda de frecuencia disponible mundialmente. Pueden realizarse conexiones punto a punto y punto multipunto [91].

Infrarrojo

Al igual que los anteriores protocolos tiene como fin el envío y recepción de datos, pero con el paso del tiempo su uso ha disminuido ya que resulta ser un protocolo con muchas falencias; estas son las representativas: Los dos dispositivos deben estar frente uno al otro para poder establecer la comunicación, otra falencia es su falta de potencia ya que no

puede atravesar obstáculos para establecer la comunicación y no se consiguen elevadas velocidades de transmisión [93].

7.1.20. Comunicación serial

La comunicación serial es un protocolo usual para comunicación entre dispositivos, lo que se lleva a decir que cualquier computadora actualmente incluye este tipo de comunicación. Básicamente el puerto serial envía y recibe bytes de información, un bit a la vez; esto puede ser un proceso más demorado que la comunicación en paralelo, pero es un método más sencillo que a su vez puede alcanzar distancias más grandes. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales [48].

7.1.21. Modos de transmisión

Hay dos tipos de modos de transmisión y son los siguientes:

Transmisiones asíncronas: La transmisión asíncrona básicamente es la relación establecida entre la estación que envía y la que recibe, la cual no tienen ninguna relación temporal. Esto quiere decir que la información al receptor no tiene un ritmo que coincida con la del emisor. [48].

Transmisiones síncronas: Este tipo de transmisión consiste en el envío de un conjunto de datos o caracteres el cual configura la información, que comienza como un conjunto de bits sincrónicos y finaliza con un conjunto de final de bloque, con lo anterior se puede decir que el conjunto de datos sincrónicos tienen como características sincronizar el reloj

de emisor y receptor, dando como resultado el control de la duración de los bits y caracteres. [48].

7.1.22. Sistemas operativos y aplicaciones

Plataformas móvil

Las plataformas móviles son la base para cualquier dispositivo móvil. Estas plataformas ofrecen el kit de desarrollo de software o SDK, las diferentes herramientas y el sistema operativo que da la posibilidad de diseñar y desarrollar aplicaciones para la plataforma.

Sistemas operativos para dispositivos móviles

Hay diferentes tipos de sistemas operativos, pero los principales son los siguientes:

Palm OS: Palm os o también conocido como Garnet os, es un sistema operativo que fue creado por la empresa Palm Source, como característica principal es su simplicidad de manejo, por lo que lo hace uno de los sistemas operativos más reconocidos en la historia. Este sistema operativo es el que corre en los PDAs de la marca Palm, aunque también hay otras marcas que lo usan como por ejemplo: Sony, Samsung, IBM y kyocera entre otros. En la actualidad Palm tiene dos entornos los cuales son: Palm OS Garnet (Versión 5) y Palm OS Cobalt (Versión 6). [1].

Android: Android es un sistema operativo que se basó en los núcleos de Linux, tiene como característica principal operar dispositivos como: Tablets, Smartphones, PDAs, Televisores, relojes entre otros; desarrollado por la empresa Android inc que fue respaldado por Google y que posteriormente compraría a Android inc, con el uso de la filosofía de código abierto, lo que conlleva a que cualquier persona puede descargar el

código fuente, modificarlo dependiendo las necesidades y compartir los cambios con la comunidad. [9].

Características Android: Como anteriormente mencionamos una de las principales características de Android es su composición por varias capas, basadas en el uso del núcleo de Linux llamado Kernel Linux, así como lo muestra la *Figura 30.* , tiene la característica de adaptarse a pantallas de mayor tamaño, tiene una base de gráficos 2D al igual que uno 3D, para su almacenamiento usa SQLite, Android soporta las diferentes conectividades las cuales son: GSM/EDGE, IDEN, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, NFC entre otras, también soporta los siguientes formatos multimedia: WebM, H.263, Mp4, Mp3, MIDI, WAV, JPEG, PNG, GIF y BMP, soporta pantallas táctiles, cámaras de fotos, video, acelerómetros, giroscopios entre muchos otros. Una de sus características más especiales es que las aplicaciones se pueden conseguir por medio de Play Store, pero también es posible conseguir aplicaciones por medio de la navegación por internet. [94].

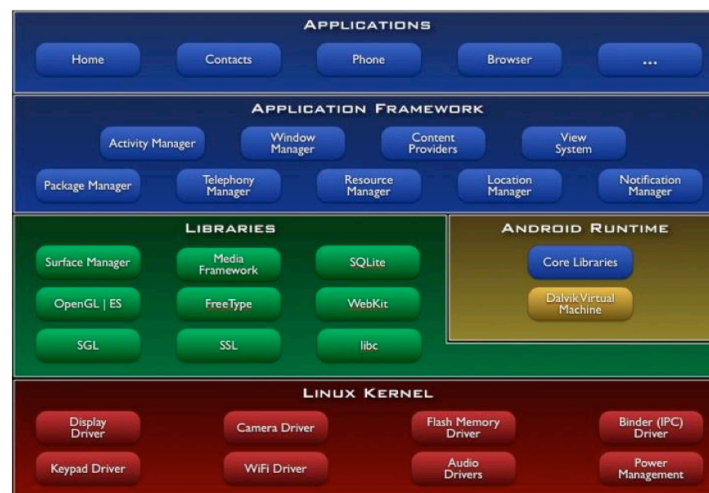


Figura 30 *Arquitectura de Android* [94].

Windows Mobile: Es una plataforma creada por la empresa Microsoft para herramientas y dispositivos móviles, al principio Windows Mobile era conocido como Pocket PC. Esta plataforma se basó en el núcleo de Windows CE. Sus principales versiones son las siguientes: Pocket Pc 2000, Pocket PC 2002, Windows Mobile 2003, Windows Mobile 5.0, Windows Mobile 6.0, Windows Mobile 6.1 y Windows Mobile 6.5 [1].

iOS: iOS es una plataforma para dispositivos móviles, la cual fue creada por la empresa Apple inc. Desarrollado principalmente para el Iphone, después también fue incluido en dispositivos como el Ipod Touch y el Ipad. Esta plataforma solo puede ser usada por dispositivos Apple y no por terceros. iOS tiene una arquitectura de capas de software que varía dependiendo de la líneas de código que se necesita para obtener las aplicaciones, en pocas palabras cuanto mayor sea el nivel, menos esfuerzo requerirá la creación de la aplicación móvil. Las capas se dividen en cuatro partes como lo muestra la *Figura 31*; y son las siguientes: Cocoa Touch es la capa superior de la arquitectura, esta capa proporciona servicios como notificaciones, entre otros. La siguiente capa es la capa Media, esta capa que proporciona la habilidad de reproducción de audio y video, también capacidades de graficación para las animaciones. La tercera capa es llamada Core service que junto con la cuarta capa llamada Core Os se encargan de la parte de variedades de servicios, accesos a los accesorios externos y servicios del sistema, tales como la gestión de memoria, gestión de sistemas[14].

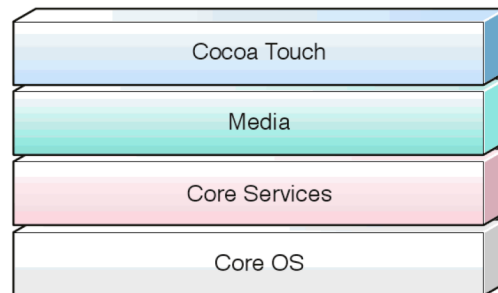


Figura 31 *Arquitectura de iOS [94]*.

Xcode: Es el entorno de trabajo de Apple, permite desarrollar aplicaciones para Macintosh, iPhone y iPad. Solamente se puede trabajar con Xcode teniendo un ordenador de Apple [11]. Para distribuir las aplicaciones para la plataforma iOS se debe hacer únicamente por la aplicación Appstore de Apple. Las aplicaciones para que salgan al mercado por medio de Appstore primero debe pasar por un protocolo el cual consiste en: debe presentar la aplicación para una inspección al centro de desarrollo de iOS. Una vez que se ha completado la revisión y la aplicación cumple los requisitos obligatorios establecidos por Apple, se publica la aplicación en la AppStore[14].

7.1.23. Desarrollo de Aplicaciones:

En la actualidad, hay diferentes lenguajes de programación para dispositivos móviles en el mercado, donde van desde C, C++, Java hasta Basic. Estos programas pueden ser desarrollados en cualquier ordenador, para después emularlos e instalarlos en los dispositivos reales. [1].

Android tiene un marco de trabajo de aplicaciones la cual permite la reutilización de componentes, debido a que cualquiera aplicación puede publicar sus capacidades y cualquiera otra aplicación puede luego hacer uso de esas capacidades. [9].

Kernel Linux: Como anteriormente se explicaba Kernel es el núcleo del sistema operativo Android. El programador no accede al kernel directamente, sino que debe utilizar librerías disponibles en capas superiores. [2].

Librerías: Están conformadas por un conjunto de bibliotecas C/C++, administrador de superficies, medios de framework, SQLite, Open GLjES, FreeType, Webkit, SGL, SSL, LBC. Runtime, conformado por librerías que se basan en Java y Dalvik la cual es una máquina virtual que delega tareas relacionadas con la gestión de hilos y memoria de bajo nivel, permitiendo que cada aplicación Android corra su propio proceso con su propia instancia Kernel de Linux, la cual da los siguientes servicios de controladores de: pantalla, cámara, Bluetooth, memoria flash, carpetas, USB, teclados, tarjetas inalámbricas, audio y de encendido. [9].

Framework de aplicaciones: Anteriormente ya habíamos mencionado que la estructura se divide en diferentes capas, estas capas son compuestas por librerías de Java y son las siguientes:

Activity Manager: Se encarga de administrar la pila de actividades y su ciclo de vida de la aplicación.

Windows Manager: Organiza lo que se mostrará en pantalla.

Content Provider: Esta librerías se encarga del encapsulamiento de datos para compartir entre las aplicaciones.

Views: Se encarga de los elementos básicos como los botones, cuadros de texto, entre otros.

Notification Manager: Su característica principal es la notificación al usuario de lo que se requiera.

Package Manager: Esto se encarga de mostrar al usuario la información de los paquetes que se han instalado anteriormente.

Resource Manager: Permite la gestión de todos los elementos que están incluidos en la aplicación pero fuera del código.

Location Manager: Con esto permite dar la posición geográfica del dispositivo.

Multimedia: Permite reproducir y visualizar archivos.

Aplicaciones: Es la capa de más alto nivel. En ella se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz gráfica de usuario como las que no.[2].

Entre la programación de aplicaciones para dispositivos Android, se encuentra una serie de programaciones llamadas “ACTIVITIES”, como su nombre lo indica es una serie de actividad que el usuario puede hacer en el dispositivo móvil y que se visualizan en la aplicación. Como se puede observar en la *Figura 32* en el sistema Android una actividad es un conjunto de acciones como por ejemplo tocar la pantalla, para apretar un botón, para escribir con el teclado, etc. que son una iteración directa con el usuario y que afectan a una parte de la aplicación. [94]

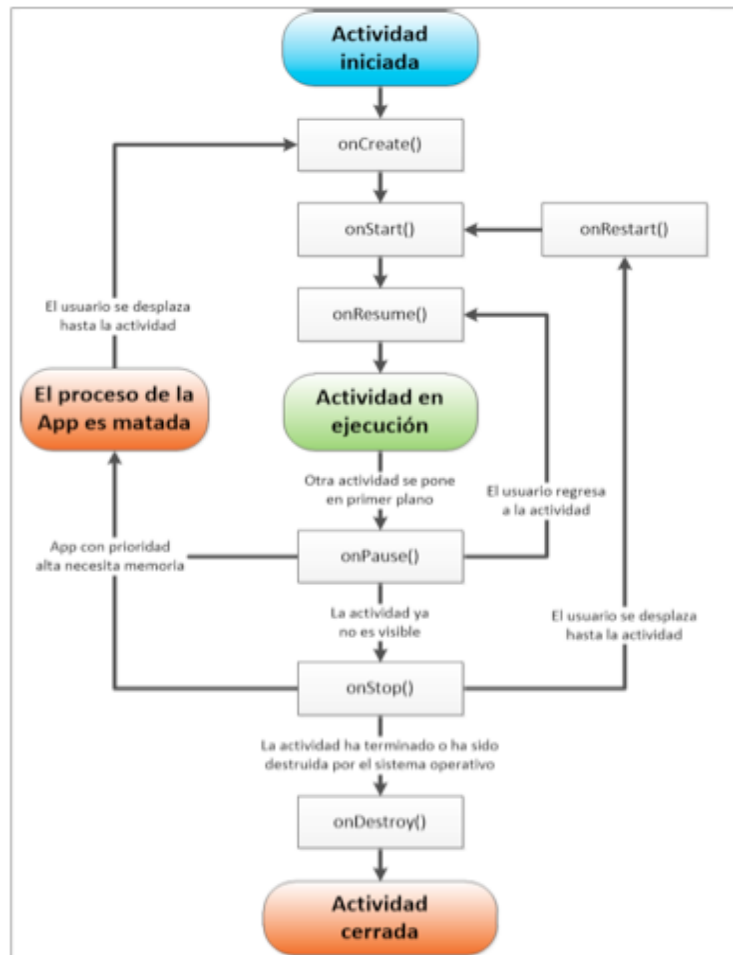


Figura 32 Diagrama de flujo de una Activity (oficial) [94]

8. DISEÑO METODOLÓGICO.

8.1. Metodología y enfoque de la investigación.

Haciendo referencia a la metodología, este proyecto se cataloga en la tipología de proyecto aplicado experimental, cuya base primordial es el desarrollo de un sistema concreto o físico, para posteriormente probarlo y documentar el proceso del desarrollo. Se optó por usar una estructura práctica en el orden de los contenidos. Lo que conlleva a dividirlo en dos partes:

- Presentación e introducción de los contenidos teóricos que hacen referencia a los elementos y conceptos esenciales que se aplicarán en el proyecto, esto se denominará “marco conceptual”, buscando que el contenido esté fundamentado a nivel conceptual, técnico y referencial.
- Desarrollo práctico: En este punto se diseñará el proyecto, se implementarán las técnicas y procesos de ingeniería necesarios para poder desarrollar la parte electrónica como también la programación del micro controlador (Arduino) comunicado con el dispositivo móvil (aplicación en Android desarrollada en App Inventor), se realizará la respectiva medición buscando que cada subsistema encaje a la perfección en el sistema completo y se evaluará su funcionamiento.
- Desarrollo de prueba subjetiva: Para la medición subjetiva se diseñó una encuesta que cuenta con 5 preguntas la cual se aplicó a 105 personas y luego se realizó el respectivo análisis estadístico.

Como se explicó anteriormente el tipo de investigación que se utilizará en este documento será de enfoque experimental, esto se debe a que se manipulará más de una

variable, buscando que la variable dependiente o consecuencia sea modificada gracias a los cambios que se generarán en las variables independientes.

9. CATEGORÍAS DE ANÁLISIS.

9.1.1. Variables Dependientes.

- Obtención de datos.
- Instrucción en los leds y en la aplicación, de las secuencias programadas.

9.1.2. Variables Independientes.

- Posicionamiento de sensores.
- Identificación de contenido armónico y frecuencia fundamental.
- Voltaje generado por el instrumento musical. (bajo eléctrico).
- Transmisión de datos de forma inalámbrica.

9.1.3. Línea de investigación

Procesamiento de señales.

Núcleo Problemático: Audio y Diseño de sistemas.

10. HIPÓTESIS.

Es posible diseñar e implementar una herramienta que se basa en un sistema físico, el cual es un conjunto de subsistemas acoplados entre sí, donde se utilizan técnicas computacionales, electrónica básica, sensórica y herramientas de uso cotidiano como son los dispositivos móviles y aplicaciones, con el fin de crear una herramienta interactiva capaz de retroalimentar al usuario en tiempo real; sustituyendo la asistencia de un tutor, y la identificación de notas ejecutadas en un instrumento musical, en este caso el bajo eléctrico. Para dicha identificación de nota del instrumento se utilizan dos técnicas de captura complementándose entre sí, la primera se basa en el contenido armónico de la nota ejecutada, y la segunda se basa en una serie de sensores puestos en el mástil del bajo eléctrico, obteniendo datos de pulsación de traste y ejecución fingerstyle. Toda esta información se procesa por medio de un micro controlador que genera comunicación a través de transmisión Bluetooth con un dispositivo móvil que posteriormente genera un reporte de ejecución del instrumento al usuario por medio de una aplicación, dando paso a la retroalimentación del usuario en tiempo real.

11. DESARROLLO INGENIERIL.

Antes de empezar a describir cada procedimiento realizado a lo largo del desarrollo ingenieril para la ejecución de nuestro sistema físico, se tuvo que definir tres fases esenciales, las cuales fueron:

- Captura de señal.
- Envío y recepción de datos.
- Visualización de datos en una aplicación para Android.

Después de definir las fases, fue necesario plantear unos requerimientos específicos, basados en los conceptos investigados en el marco conceptual, para así hacer uso e implementación de los elementos apropiados, buscando cumplir con los objetivos propuestos en este proyecto de grado.

- **Instrumento temperado de cuerda pulsada.**

Requerimientos

- Se requiere que el instrumento genere señales electromagnéticas por medio de la vibración de las cuerdas.
- Se requiere que el instrumento este revisado, calibrado y octavado por un lutier, esto con el fin de tener una correcta afinación del instrumento.
- Se requiere que el sistema eléctrico del instrumento no genere ningún tipo de interferencia, ni ruido.

- **Sensores de fuerza resistivos.**

Requerimientos

- Se requiere que sean sensibles al tacto y resistan una gran cantidad de presión.
- Se requiere que manejen un voltaje hasta de 5 voltios.
- Se requiere que el tamaño del sensor se ajuste al espacio existente entre los trastes del bajo.
- Se requiere que el grosor del sensor sea tal, que logre entrar entre el mástil y las cuerdas sin afectar la ejecución del intérprete.
- Debe tener compatibilidad con Arduino.

- **Preamplificador.**

Requerimientos.

- Se requiere que el pre amplificador tenga una ganancia de voltaje en la señal de entrada de línea de bajo a 5V pico a pico. Ganancia de 30.
- El pre amplificador no debe presentar ningún tipo de interferencia o ruido que pueda modificar la señal de entrada.
- Se requiere una señal de 0 a 5 Voltios y que su origen este en 2.5 V.
- Se requiere que la señal no llegue distorsionada ni con efecto peine por culpa de los armónicos.

- **Placa con micro-controlador.**

Requerimientos

Es necesario seleccionar una plataforma y un lenguaje de programación que pueda ejecutar la detección de frecuencia y procesos análogos digitales tales como: la detección de estado de los sensores, las ayudas al usuario (Arreglo de leds y display 7 segmentos) y la comunicación serial inalámbrica.

- Se requiere una plataforma abierta para realizar trabajos y prototipos que se basen en hardware y software, flexibles y fáciles de usar.
- Se requiere que la plataforma de la placa con micro-controlador sea funcional en sistemas operativos como: Windows, OSX y Linux.
- Se requiere que la placa tenga como función alimentar circuitos externos ya que es necesario alimentar, con una fuente Dc, al preamplificador.
- Se requiere que la placa con micro-controlador permita el anexo de un módulo Bluetooth.
- Se requiere que la placa con micro-controlador tenga la virtud de ser alimentada por USB o por baterías externas.
- Es necesario que las entradas sean suficientes para el sistema.
- Como último requerimiento es preciso que se cuente con librerías y código que faciliten la manipulación de audio.
- Se requiere un Arduino extra para la conversión análogo-Digital de la señal proveniente del arreglo sensores.

- **Protocolo de comunicación.**

Requerimientos

- Se requiere que el protocolo de comunicación sea, no guiado con base a lo especificado en los objetivos del proyecto. Los dispositivos se deben comunicar de manera inalámbrica.
- La transmisión y la recepción de datos se realizará a través de puerto.
- Se requiere un módulo hardware compatible con Arduino para realizar la comunicación serial con el dispositivo móvil.
- La alimentación del módulo debe ser de 5v.

- **Sistema operativo.**

Requerimientos

- Se requiere que permita el desarrollo de aplicaciones con interfaz gráfica, acceso a modulo Bluetooth para envío y recepción de datos.
- Se requiere que tenga un entorno propio de programación.

- **Dispositivo móvil.**

Requerimientos

- Se requiere que el dispositivo sea compatible con el lenguaje de programación empleado en Android.
- Debe manejar protocolo de comunicación inalámbrico L2CAP (Bluetooth).

- Se requiere que el dispositivo tenga un mínimo de espacio de 8 MB de memoria para la instalación del archivo .APK.

- **Entorno de programación.**

Requerimientos

- Se requiere que tenga compatibilidad con el sistema operativo Android.
- Es necesario tener herramientas para la manipulación del módulo Bluetooth.
- Se requiere que el entorno de programación genere aplicaciones ejecutables en el formato APK, compatibles en el dispositivo móvil.
- Se requiere que el entorno de programación dé la opción de generar GUI(Interfaz gráfica de usuario), capaz de manipular elementos en formatos JPEG, PNG.

11.1. Selección de dispositivos y herramientas

Una vez se plantea lo requerido para desarrollar el proyecto, se realizarán una serie de comparaciones para la selección de los dispositivos y herramientas pretendidas, con el fin de realizar sus respectivas pruebas antes de la unión de todos los subsistemas, consolidando el sistema completo.

11.1.1. Instrumento temperado de cuerda pulsada (BAJO)

De acuerdo al planteo de requerimientos al principio del desarrollo ingenieril, se puede llegar a la conclusión de que: El bajo eléctrico obtenido es el adecuado para llevar acabo el desarrollo de este proyecto.

Implementación del bajo eléctrico Tom Grasso.

Este instrumento representado en la *Figura 33* cuenta con una afinación estándar de 440 Hz, salida de línea, potenciómetros de volumen y tono, por ser un bajo eléctrico de 4 cuerdas estándar su afinación es de la siguiente forma:

- Sol= 98 Hz
- Re= 73,42 Hz
- La= 55 Hz
- Mi= 41,20 Hz



Figura 33 Bajo Eléctrico.

- Dentro del análisis que se pudo hacer del bajo, se ratificó que su sistema electrónico y su afinación están dentro del rango que se requería para poder ser implementado en este sistema.
- En la siguiente *Figura 34* se visualiza, mediante un osciloscopio, la señal que envía el instrumento. Esto fue posible gracias a las señales electromagnéticas conducidas

por el transductor (pick up), haciendo las veces de micrófono en la vibración de las cuerdas y el sistema electrónico de line out.

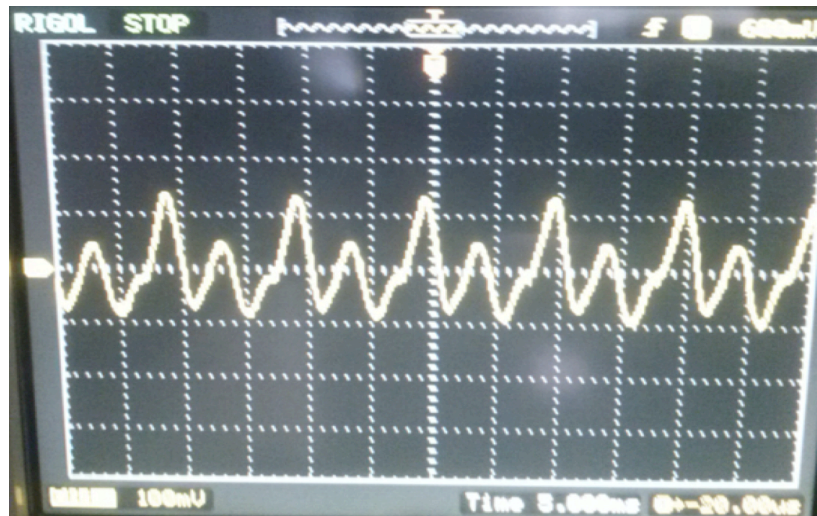


Figura 34 Señal del Bajo Eléctrico.

- Mediante un programa en “Matlab” se identificó la afinación o la frecuencia fundamental de cada nota a utilizar, así se pudo saber que el instrumento estaba en condiciones ideales para la implementación en el sistema. El algoritmo de este programa lo podemos visualizar en la *Figura 35*.

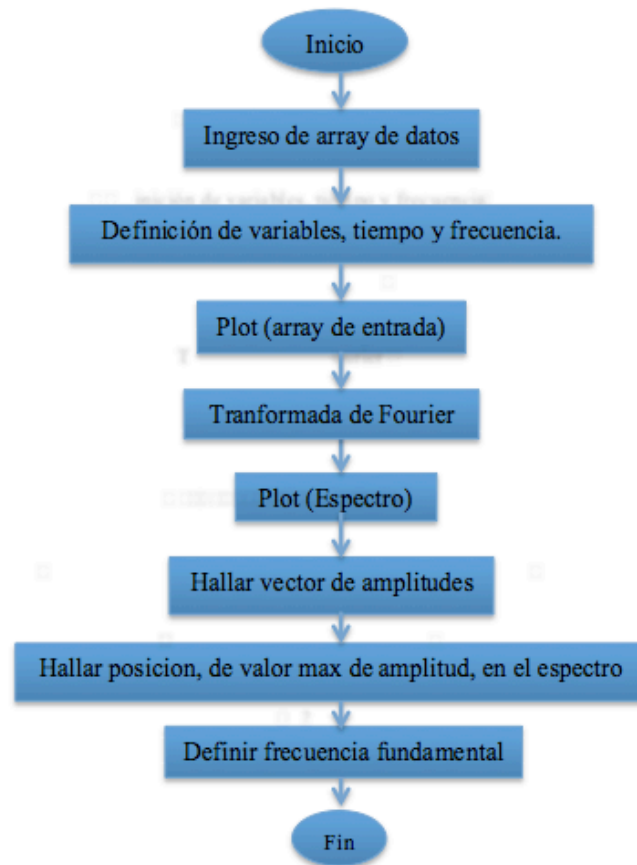


Figura 35 Diagrama de flujo algoritmo Matlab.

Nota: Guiados por la gráfica.; dependiendo el número del armónico con amplitud máxima se realizó la operación respectiva para hallar la posición y frecuencia en los vectores, obteniendo así la frecuencia fundamental (Primer armónico). La *Figura 36* muestra el código que se realizó en Matlab para implementar el diagrama de la *Figura 35*.

```

Editor - C:\Users\211\Documents\MATLAB\Frecuenciafundamental.m
Frecuenciafundamental.m
1
2 [y, Fs] = audioread(' .wav');
3
4 Fs=44100;
5 T = 1/Fs; % Sample time
6 L = length(y); % Length of signal
7 t = (0:L-1)*T; % Time vector
8
9 subplot(2,1,1);
10 plot(t,y);
11 xlabel('time (milliseconds)');
12
13 NFFT = 2^nextpow2(L); % Next power of 2 from length of y
14 Y = fft(y,NFFT)/L;
15 f = Fs/2* linspace(0,1,NFFT/2+1);
16
17 % Plot single-sided amplitude spectrum.
18 subplot(2,1,2);
19 plot(f,2*abs(Y(1:NFFT/2+1)));
20 title('Single-Sided Amplitude Spectrum of y(t)')
21 xlabel('Frequency (Hz)')
22 ylabel('|Y(f)|')
23
24 a=2*abs(Y(1:NFFT/2+1));
25
26 d=find(a==max(a));
27

```

Figura 36 Algoritmo implementado en Matlab.

La Figura 37 muestra el sub-plot que arroja el programa en Matlab, en la parte superior se puede observar el array de entrada, previamente grabado en Pro Tools (Figura 107) y en la parte inferior el espectro o gráfica de la transformada de Fourier, la cual muestra tres armónicos de la señal, posicionando el cursor en el primer armónico $F= 43,4\text{Hz}$.

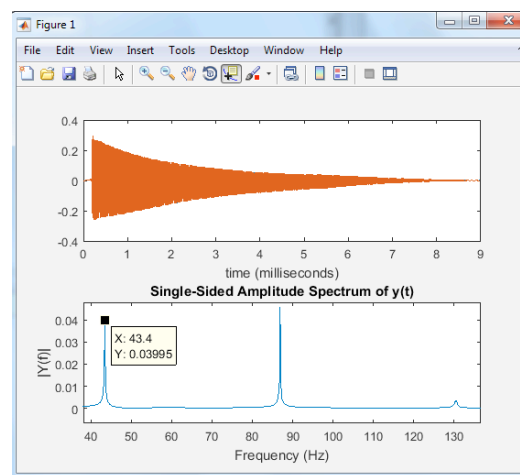


Figura 37 Fundamental F en la cuerda E.

Se realizó el proceso anterior para cada una de las notas a utilizar y en la *Tabla 6* se muestran los valores extraídos de Matlab. Las gráficas de cada nota se encuentran en el Anexo A.

Cuerda	Nota	Frecuencia(Hz)
E	F	43.4
E	F#	45.93
E	G	48.79
E	G#	51.73
A	A#	58.29
A	B	61.74
A	C	65.44
A	C#	69.31
D	D#	77.89
D	E	82.6
D	F	87.48
D	F#	92.86
G	G#	103.4
G	A	109.8
G	A#	116.3
G	B	123.1

Tabla 6 de frecuencias fundamentales.

11.1.2. Sensores.

Sensor de fuerza o presión

Los sensores que lograron cumplir con los requerimientos expuestos al inicio de este desarrollo ingenieril fueron los sensores de fuerza resistivos FSR 400 y FSR 406, debido al tamaño, grosor, compatibilidad con Arduino y rango de tensión en el funcionamiento, Los FSR400 no se usaron debido a que su superficie redonda era muy

pequeña. Finalmente se usó el FSR 406, que se puede observar en la *Figura 38*, ya que lograban cumplir con todo lo requerido.

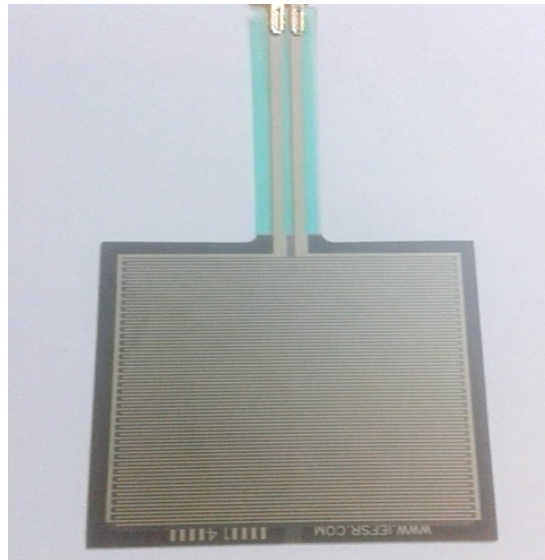


Figura 38 Sensor de fuerza resistivo FSR Cuadrado. [48].

11.1.2.1. Prueba de sensor de fuerza resistivo.

Para realizar una prueba del funcionamiento del sensor se procedió a elaborar un montaje como se puede ver en la *Figura 39* que consta de una resistencia de 3 KOhm conectada al polo negativo del sensor, con esto se podrá polarizar y obtener la salida de datos y mediante la plataforma Arduino recibir los valores del sensor (FSR 406).



Figura 39 Montaje de prueba del sensor

Se realizó la medición del sensor mediante el multímetro y se obtuvo el valor mínimo, el cual fue de 049.847 MOhm (*Figura 40*) y máximo de 071.399 MOhm (*Figura 41*) de resistencia del sensor.

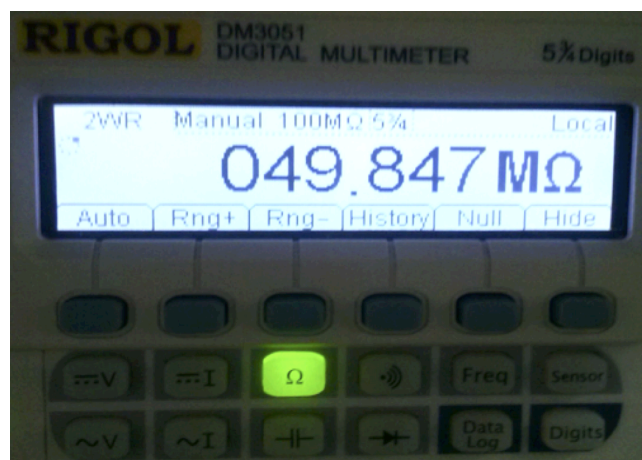


Figura 40 Medición de resistencia del sensor (valor mínimo).



Figura 41 Medición de resistencia del sensor (valor máximo)

11.1.2.2. Sketch para prueba de sensor

En la *Figura 42* se muestra un sketch básico en Arduino, donde se define la variable en la que se va a guardar el valor de la entrada análoga proveniente del sensor, en este caso la entrada A0. Luego se inicializa el puerto serial y en un ciclo de 100 valores se imprimen los valores que se captaron.

```

sensor
int pulsador=A0;
int sensorValue = 0;
int i=0;

void setup()
{
  Serial.begin(9600);
  pinMode(pulsador, INPUT);
}

void loop()
{
  for(i=0; i <= 100; i++)
  {
    sensorValue = analogRead(pulsador);
    Serial.println(sensorValue);
  }
}

```

Figura 42 SKETCH Arduino para prueba de sensor.

11.1.2.3. Implementación de arreglo de sensores en el bajo eléctrico.

Para la implementación del arreglo de sensores fue necesario modificar la dimensión de algunos sensores para la inserción de estos a los trastes, este procedimiento se hizo con sumo cuidado para no dañar las celdas o dejar inservible el sensor. Se ubicaron en total 3 sensores en los primeros 3 trastes del instrumento. Se procedió a soldar las resistencias anteriormente mencionadas en el numeral 11.1.3.1 y pines a cada sensor con el fin de sacar una línea estilo ribbon al Arduino. Para ubicar y ordenar las conexiones internas se diseñó una estructura adherida al mástil del bajo, buscando no obstruir al usuario en el momento de ejecutar el instrumento. Esto se ilustra en la *Figura 43*.



Figura 43 Arreglo de sensores en el mástil del bajo eléctrico.

11.1.3. Diseño y desarrollo de pre amplificador.

Preamplificador de 2 etapas.

En este proyecto se requería un pre amplificador que al tomar la entrada de señal del instrumento pueda elevar esta señal para que sea reconocida por el Arduino ya que este requiere una señal de 0 a 5 Voltios y que su origen este en 2.5 V. Otra de las razones por las cuales se necesitó el diseño e implementación del pre amplificador, fue, que la señal de línea del bajo es pequeña, un valor de ± 83 mV pico (*Tabla 14*). Se realizó una copia de la señal de entrada esto con el fin de tener entrada al pre amplificador y una salida de línea. Se cuenta con dos etapas y es necesaria una fuente de alimentación dc de 15 voltios, la primera etapa es una etapa JFET (transistor de efecto de campo de juntura o unión) y la segunda etapa es una etapa BJT (transistor de unión bipolar). Por último se implementa un elevador de voltaje, para así enviar la señal de salida a la entrada analógica A0 del Arduino, para luego proceder en el análisis espectral en Arduino, para sacar la frecuencia fundamental (*Figura 72*).

11.1.3.1. Procedimiento de simulación del preamplificador de 2 etapas

Como primera medida se realizaron los diseños y cálculos de cada uno de los montajes. La *Figura 44* muestra la parte JFET del preamplificador y posteriormente su respectivo cálculo.

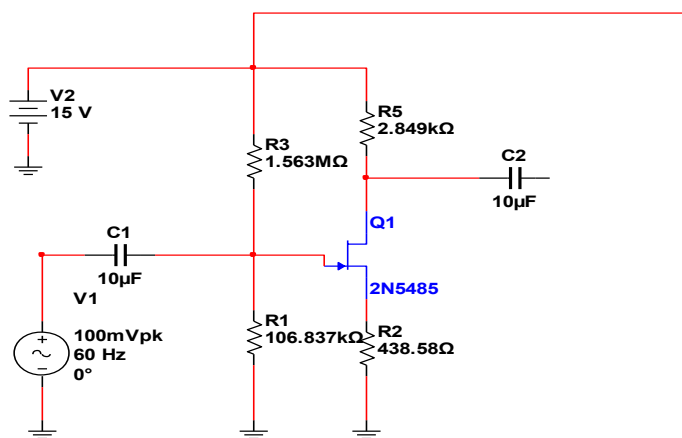


Figura 44 Etapa JFET

El transistor J-FET que se utilizara en este proyecto será el 2N5485, para el método de diseño los valores de entrada son la corriente de drenaje a fuente (I_{DSS}) y el voltaje de pinch off (V_p).

$$I_{DSS} = 6,84_{mA}$$

$$V_p = -1,8V$$

El voltaje de polarización o V_{DD} será de 15V. Por medio de los criterios de diseño se toma una impedancia de entrada muy grande, es decir:

$$Z_{in} = R_1 // R_2$$

Ecuación 5 Impedancia de entrada de etapa JFET.

La I de drenaje será:

$$I_D = I_{DSS} / 2 \quad I_D = 3,42_{mA}$$

Ecuación 6 Corriente de drenaje.

A partir de la siguiente formula hallaremos la resistencia en source :

$$R_S = (V_{DD}/ 10)/ I_d \qquad R_S = 438,59_{\text{Ohm}}$$

Ecuación 7 Resistencia en source.

Para hallar la trans-conductancia (gm) primero es necesario hallar gmo.

$$g_{m0} = (2 \cdot I_{DSS}) / (1 - V_p) \qquad g_{m0} = 7,6_{\text{mS}}$$

Ecuación 8 Valor de gmo a partir de la corriente de drenaje a fuente (I_{DSS}) y el voltaje de pinch off (V_p).

Entonces a partir de la *ecuación 8* hallamos gm.

$$g_m = 0,7 \cdot g_{m0} = 5,32_{\text{mS}}$$

Ecuación 9 Trans-conductancia.

La ganancia establecida para el diseño es de 4,5, con la fórmula de ganancia hallaremos la resistencia de drenaje con los valores obtenidos de R_S en la *ecuación 7* y gm de la *ecuación 9*.

$$A_v = 4,5$$

$$A_v = (- g_m \cdot R_D) / (1 + (g_m \cdot R_S))$$

Ecuación 10 Ganancia de voltaje etapa JFET.

Se despeja R_D y se halla su valor.

$$R_D = ((4,5) (1 + g_m \cdot R_S)) / g_m \qquad R_D = 2,849_{\text{Ohm}}$$

Ecuación 11 Resistencia en el drain.

Para hallar las resistencias R_1 y R_2 se debe conocer el valor del voltaje en el gate V_G a partir del valor del voltaje de pinch off V_P y del valor del voltaje de polarización V_{DD} .

$$V_{GS} = 0,3 * V_P = 0,3 * (-1,8V) = -0,54V \quad V_{RS} = V_{DD}/10 = 15V/10 = 1,5V$$

Ecuación 12 Voltaje gate source y voltaje en R_S .

$$V_G = V_{GS} + V_{RS}$$

$$V_G = -0,54V + 1,5V = 0,96V$$

Ecuación 13 *Voltaje en gate.*

A continuación hallaremos R_2 .

$$R_2 = Z_{in} / (1 - (V_G/V_{DD})) \quad R_2 = 100K\Omega / (1 - (0,96V/15V))$$

Ecuación 14 Resistencia R_2 Etapa JFET.

De la *ecuación 14* obtenemos:

$$R_2 = 106,837K$$

A continuación hallaremos R_1 .

$$R_1 = (z_{in} * V_{DD})/V_G \quad R_1 = (100K\Omega - 15V)/ 0,96V$$

Ecuación 15 Resistencia R_1 Etapa JFET.

Y así obtenemos la resistencia R_1 .

$$R_1 = 1,563\text{M}\Omega$$

Debido a las limitaciones con los valores del V_P y del I_{DSS} se tuvo complicaciones en la ganancia de esta etapa. En la *Figura 45* podemos ver como es el diseño de la parte bipolar número 1 del preamplificador y sus respectivos cálculos.

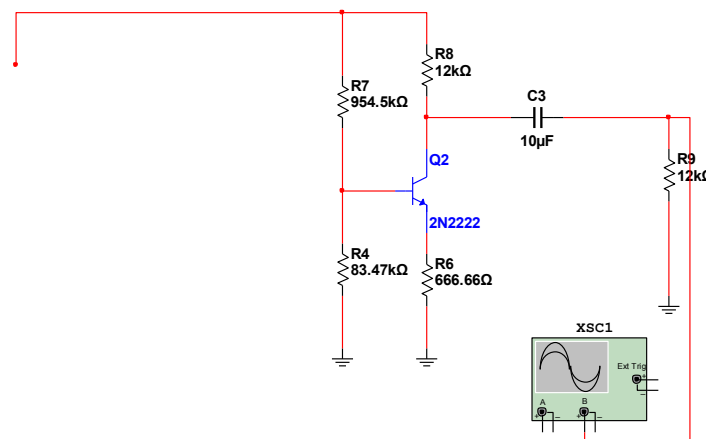


Figura 45 Etapa Bipolar 1.

La referencia del transistor que se utilizara en esta segunda etapa bipolar es 2n2222. Según el criterio de diseño para máxima transferencia de potencia la resistencia del colector debe ser igual a la resistencia de carga.

$$R_C = R_L = 12\text{K}_{\text{Ohm}}$$

Ecuación 16 Valor de la resistencia de carga.

La ganancia de voltaje requerida para esta etapa con el fin de obtener 5V de salida, es de:

$$A_v=9$$

A partir de la fórmula de ganancia de voltaje hallamos la resistencia en el emisor.

$$A_v=(R_C//R_L)/R_E$$

Ecuación 17 Ganancia de voltaje etapa bipolar.

El paralelo entre R_L y R_C tomando el valor de la *ecuación 16* será:

$$R_C//R_L=(R_C \cdot R_L)/(R_C+R_L)=6K_{Ohm}$$

Ecuación 18 paralelo entre R_C y R_L .

Se despeja la resistencia en el emisor de la *ecuación 17*.

$$R_E=(6K_{ohm}) / 9 \quad R_E= 666.6_{ohm}$$

Ecuación 19 Resistencia en el emisor.

El voltaje de polarización es $V_{cc}=15V$ y la resistencia en DC es :

$$R_{DC}=R_C+R_E$$

Ecuación 20 Resistencia en DC.

Entonces $R_{DC}=12.6K_{ohm}$

La resistencia en AC es:

$$R_{AC}=(R_C//R_L)+R_E$$

Ecuación 21 Resistencia en AC.

Entonces $R_{AC}=6.6K_{ohm}$.

De las *ecuaciones 20 y 21* extraemos el valor de R_{DC} y R_{AC} para así hallar el voltaje colector emisor de polarización a partir de la formula.

$$V_{CEQ}=V_{CC} / (1+(R_{DC}/R_{AC}))$$

Ecuación 22 Voltaje colector emisor de polarización.

Así $V_{CEQ}= 5.17V$ y la corriente de polarización en el colector será:

$$I_{CQ}=V_{CC}/(R_{AC}+R_{DC})$$

Ecuación 23 Corriente de polarización en el colector.

Entonces $I_{CQ}=775.86_{uA}$.

El beta del transistor es $B=115.5$ entonces la resistencia de Thevenin.

$$R_{TH} = B \cdot R_E = 115.5 \cdot 666.6_{ohm} = 77K_{ohm}$$

Ecuación 24 Resistencia de Thevenin.

El voltaje base emisor es $V_{BE}=0.7V$, tomamos los valores de I_{CQ} de la *ecuación 23* y R_E de la *ecuación 19* así el voltaje en la resistencia 2 es:

$$V_{R2}=(I_{CQ} * R_E)+V_{BE}$$

Ecuación 25 Ecuación de voltaje en la resistencia 2.

Entonces

$$V_{R2}=(775.86_{\mu A} * 666.6_{\text{ohm}})+0.7V \quad V_{R2}=1.21V$$

Ecuación 26 Voltaje en la resistencia 2.

Se hallan R_1 y R_2 con las siguientes ecuaciones, teniendo el valor de V_{cc} que es de 15 voltios, el valor de la resistencia de thevenin en la *ecuación 24* y el valor del voltaje en R_2 de la *ecuación 26*.

$$R_1= (V_{CC} * R_{TH})/V_{R2} \quad R_1=(15V * 77K_{\text{ohm}})/1.21V \quad R_1=954.5K_{\text{ohm}}$$

Ecuación 27 Resistencia 1 etapa bipolar.

$$R_2= (R_{TH} * R_1)/(R_1 - R_{TH}) \quad R_2=83756.34_{\text{ohm}}$$

Ecuación 28 Resistencia 2 etapa bipolar.

Para finalizar el preamplificador se requiere un elevador de nivel en Dc (offset) la *Figura 46* muestra cuáles son sus valores. Poniendo a R9 como la resistencia que permite el acople de impedancia y C4, C5, R10, R11 y la fuente DC de 5v permite el desplazamiento de la señal.

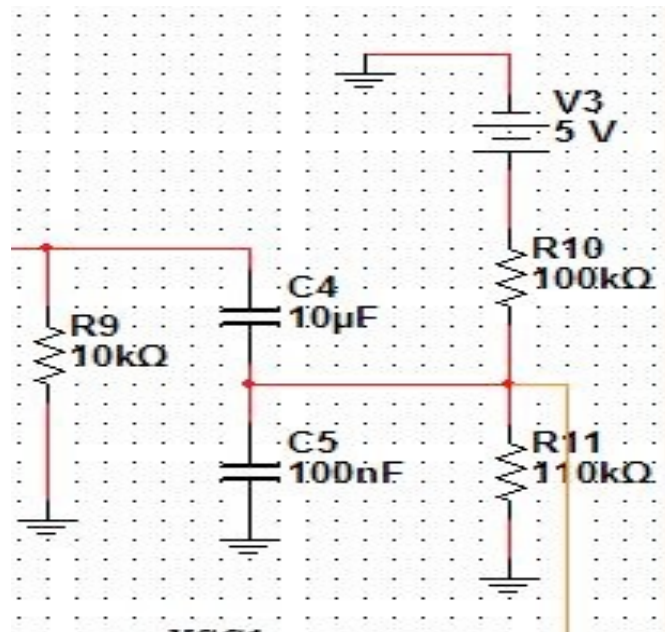


Figura 46 Elevador de nivel DC (offset).

11.1.3.2. Implementación del preamplificador de 2 etapas

Dados los cálculos anteriores se implementaron los materiales necesarios para la construcción del preamplificador de la siguiente forma:

- Primero. Se simularon los valores obtenidos por medio de los cálculos en el programa NI Multisim como lo muestra la *Figura 47*. Dando el resultado esperado.

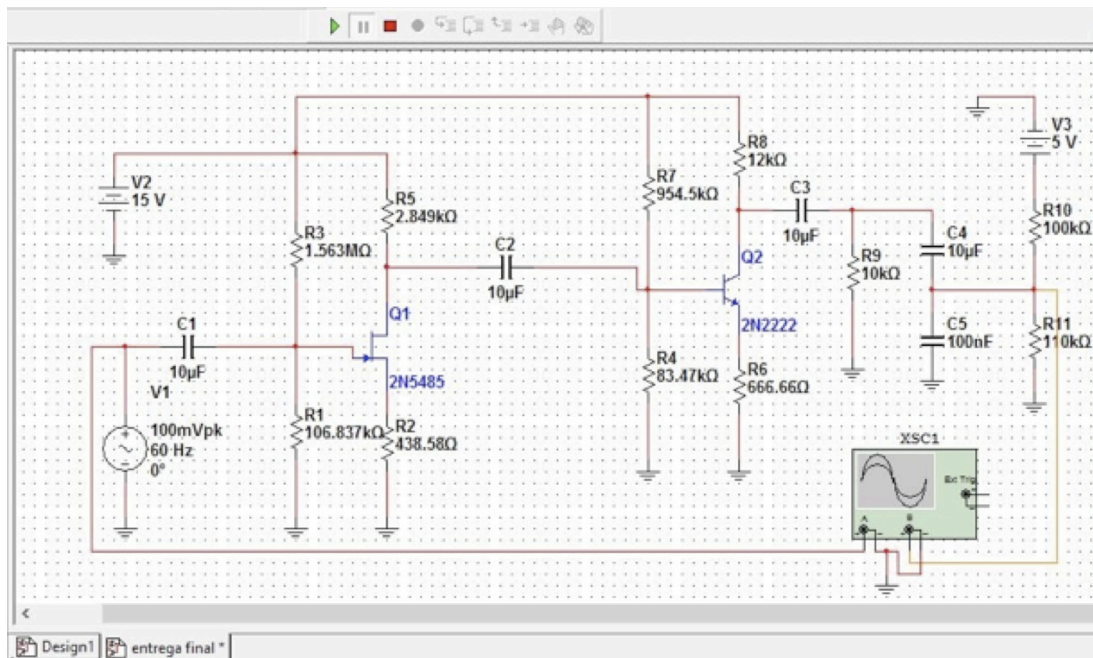


Figura 47 Simulación de preamplificador en NI Multisim.

Debido a que la señal de entrada con mayor amplitud (*Tabla 14*), es de 193mVpp la resistencia de carga se redujo de 12kOhms(*Figura 45*, R9) a 10kOhms, con el fin de evitar distorsión. La *Figura 48* muestra el osciloscopio digital, la señal de color rojo en escala de 500mV es la de entrada y la de color naranja en escala de 2V es la señal de salida.

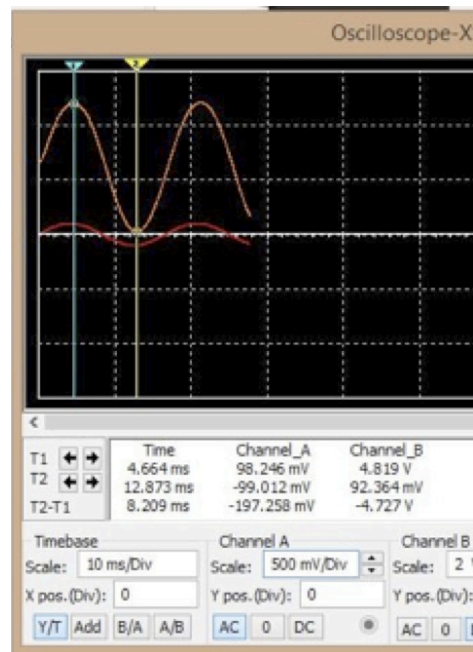


Figura 48 Simulación multímetro en la salida del pre amplificador en NI Multisim.

- Segundo. Después de simular los datos y obtener una respuesta positiva en la simulación se prosigue a la implementación de los materiales en una protoboard. El resultado se puede apreciar en la *Figura 49*.

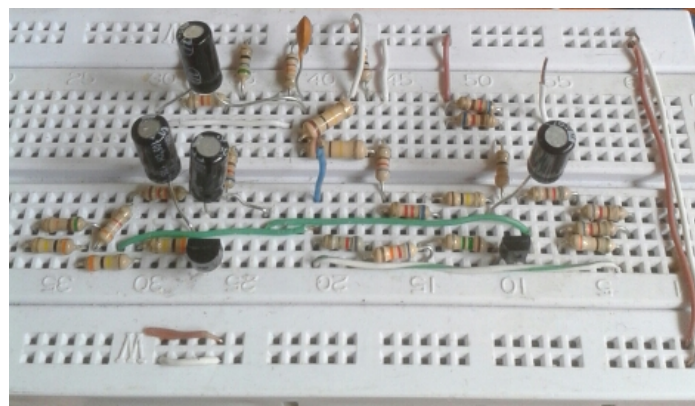


Figura 49 Preamplificador montado en protoboard.

11.1.3.3. Medición y montaje del preamplificador

Para la prueba de un preamplificador es necesario medir cada una de sus etapas y analizar si funcionan correctamente. Para esto fue necesario utilizar un osciloscopio, una fuente Dc y un generador. Con el osciloscopio se puede observar el comportamiento de la señal de entrada y su paso por cada etapa. La fuente Dc es necesaria para suplir de polarización al preamplificador, para este caso se necesitaron dos fuentes Dc, una de 5V para el elevador de tensión y una de 15 V para las dos etapas, como se puede observar en la simulación. El generador fue usado para suplir la señal del bajo eléctrico, dando la facilidad de tener una señal constante en el tiempo que permite verificar cada una de las etapas, sin necesidad de interpretar el instrumento. El primer paso para analizar el preamplificador es visualizar la señal de entrada como se ve en la *Figura 50*, es decir se envía la señal del generador al preamplificador y por medio del osciloscopio se observa cual es la señal que entra al preamplificador. Se puede ver que la escala está en 100mV de amplitud y la señal tiene 83mV_p de amplitud.

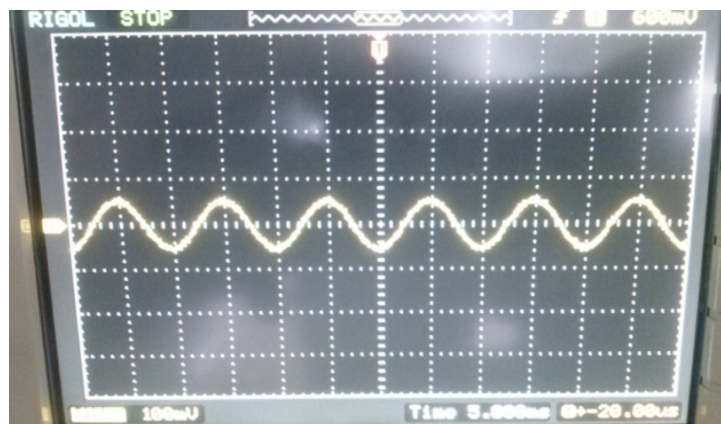


Figura 50 Señal de entrada.

El siguiente paso es el de medir la primera etapa del preamplificador, esta etapa es la JFET, se puede observar en la *Figura 51* la escala a 500mV y la señal que ha aumentado su nivel de ganancia a 3 veces su entrada.

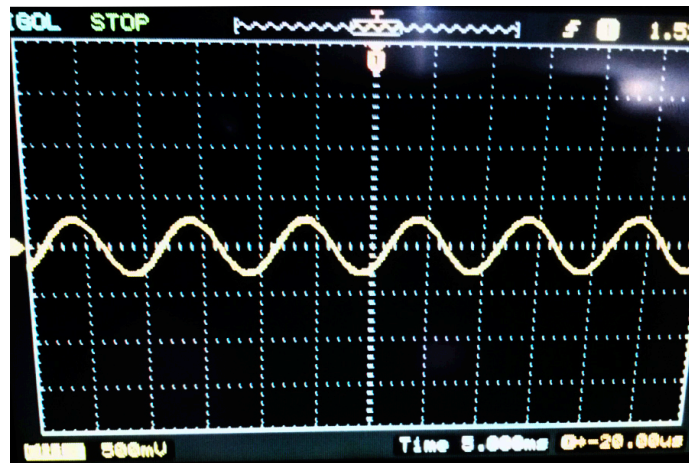


Figura 51 Señal de la primera etapa JFET.

Para el siguiente análisis del pre amplificador se mide su segunda etapa, esta es conocida como la etapa bipolar o BJT, en la *Figura 52* se puede observar con escala a 2V que su ganancia es de aproximadamente 10.

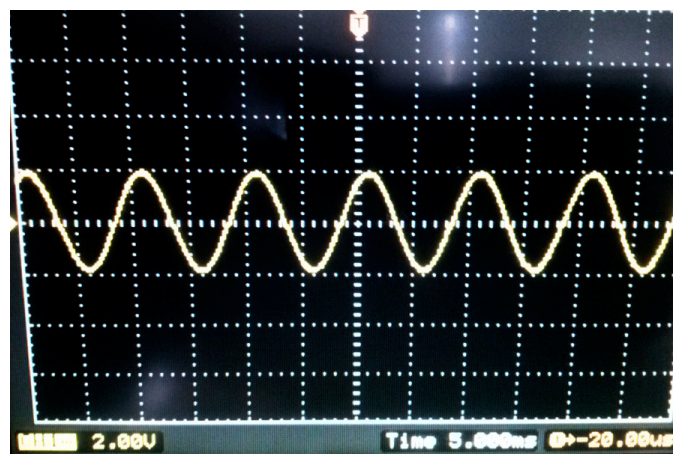


Figura 52 Señal de la segunda etapa BJT.

Para finalizar analizaremos la señal en la parte del elevador de nivel DC dando como resultado lo siguiente (Ver *Figura 53*):

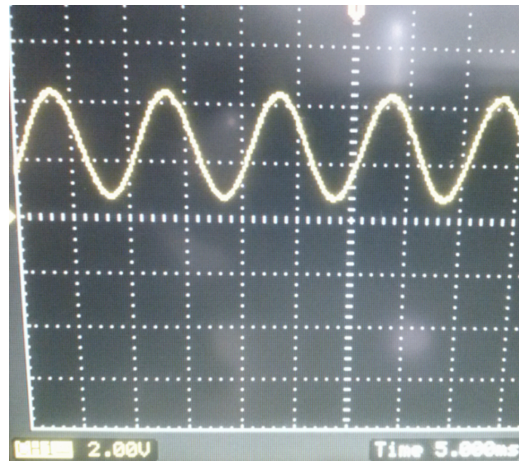


Figura 53 Señal amplificada con elevador de voltaje.

Después del anterior análisis y ver que el circuito del preamplificador funciona perfectamente cumpliendo con todos los requerimientos, el paso a seguir fue pasar el preamplificador de la protoboard a una vácueta con el fin de implementarlo en el sistema. En la *Figura 54* se encuentra el diseño del PCB con todos los elementos de la simulación y en la *Figura 55* se muestra el preamplificador quemado en baqueta.

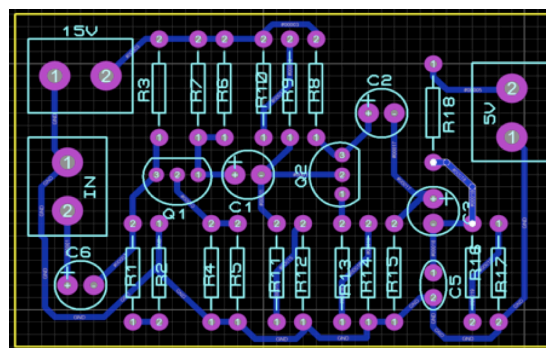


Figura 54 PCB del preamplificador.



Figura 55 preamplificador montado en baqueta.

Para la finalización de las pruebas y análisis del preamplificador, se medirá y se observará por medio del osciloscopio el comportamiento en conjunto, que consta de la entrada y la salida del preamplificador mostrando en la *Figura 56*, que la ganancia obtenida es la requerida para el sistema.



Figura 56 Entrada y salida del preamplificador mostrada por el osciloscopio.

11.1.4. Placa con micro controlador.

Arduino: Dadas las siguientes ventajas Arduino (Ver *Figura 57*) fue la placa con micro-controlador más idónea para el sistema ya que cumple con todos los requerimientos establecidos.

- Como se especifica en los requerimientos, esta placa puede ser compatible con los principales sistemas operativos para su programación, es el caso de Windows, OXs y Linux.
- El lenguaje de programación es sencillo y directo. Arduino se basa en un entorno de programación processing por lo cual el usuario, teniendo conocimientos básicos de programación, lograra familiarizarse.
- Arduino da la opción de publicar bajo licencia libre, además de proporcionar librerías de gran ayuda para procesar audio. Su lenguaje puede ser ampliado por comandos de C++ y modificado a través de AVR C que es su lenguaje nativo.
- Su hardware es ampliable, ya que hay diferentes tipos de Arduinos con micro-controladores distintos y sus planos están online para que personas con conocimientos en electrónica, puedan realizar su propia versión de Arduino.

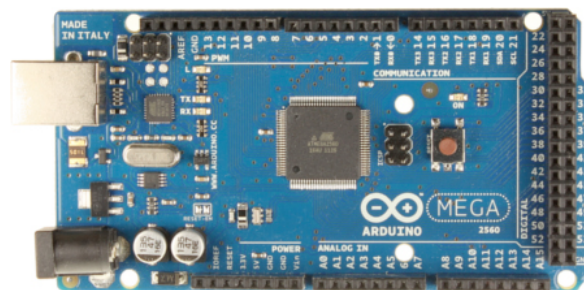


Figura 57 Arduino Mega. [24]

11.1.4.1. Implementación de Arduino mega

Después de hacer la comparación respectiva, se eligió la placa de Arduino y se procedió a ver distintos tipos de Arduinos, obteniendo ventajas y desventajas de estos. Arduino uno y nano: Dentro de las especificaciones de este Arduino encontramos que esta placa suple algunas o la mayorías de necesidades que se requieren para hacer parte del sistema, posee una plataforma abierta, es flexible y fácil de usar, es funcional con los sistemas operativos que se necesitan, tiene la función de alimentar circuitos externos con 5V DC, además es compatible con un módulo Bluetooth, ya que Arduino tiene sus propios módulos, tiene la ventaja de ser alimentada por medio de USB, adaptadores y baterías, pero posee una gran limitante y es que no posee las entradas suficientes para el sistema, ya que esta placa solo posee 6 salidas digitales y 6 entradas análogas. Mientras que la placa que les sigue, la placa de Arduino mega/2560 que se muestra en la *Figura 57*, suple todas las necesidades requeridas por el sistema, dando 54 pines digitales y 16 pines de entradas análogas siendo esta la placa con la capacidad de suplir todas las necesidades. En la *Figura 58* podemos observar una de las primeras pruebas de comunicación Bluetooth en el Arduino Mega, que consistía en prender un Led por medio de un dispositivo móvil.

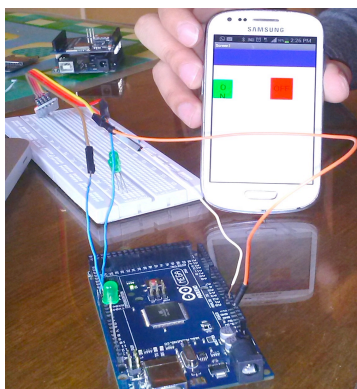


Figura 58 Prueba comunicación Arduino Mega-Dispositivo móvil.

11.1.4.2. Implementación de Arduino Nano

Aunque al principio del proyecto no era necesario un segundo Arduino, al momento de realizar las pruebas pertinentes en la detección de frecuencia, se concluyó que los puertos análogos del Arduino Mega quedaban inactivos. Por tal razón, para generar la conexión análoga del arreglo de sensores (Ver numeral 11.1.3.3) al Arduino Mega fue necesario utilizar un programa en un Arduino extra (Arduino Nano ver *Figura 59*) y con este generar una conversión análoga-digital. Aunque en este caso se podía seleccionar tanto el Arduino Nano como el Arduino Uno, se eligió el primero, puesto que presenta ventajas en cuanto a tamaño.

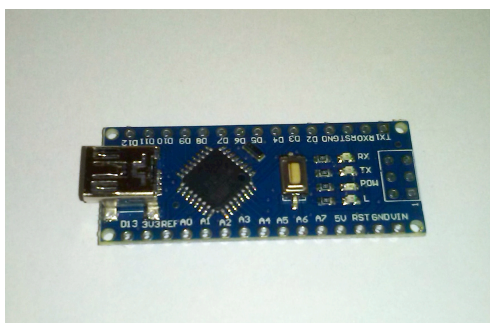


Figura 59 Arduino Nano.

NOTA 1: Para la implementación de los micro controladores (Arduinos) en el sistema completo, será necesario antes conocer los algoritmos y programas diseñados y desarrollados para estos; esto se verá a continuación.

NOTA 2: Es necesario mencionar que los subprogramas se representaran con color verde, además para todas las condiciones en los diagramas realizados en este proyecto, se entenderá que el lado derecho será Si y el lado Izquierdo será No.



11.1.4.3. Diseño del algoritmo para implementar en Arduino

Para la implementación de la placa con micro-controlador, se realizó un diagrama de bloques, el cual muestra el algoritmo general (*Figura 60*).

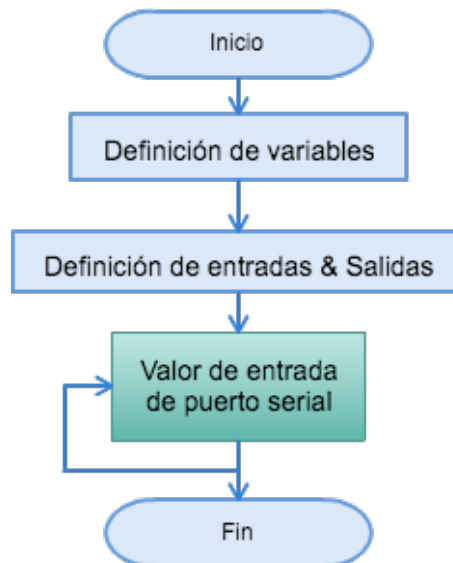


Figura 60 Diagrama general de algoritmo en Arduino.

Como se explicó anteriormente en el numeral 11.1.5.2, hubo un problema en los puertos analógicos del Arduino mega, por lo que se implementó un segundo Arduino. El cual contenía un programa que se encargaba de leer los valores análogos de los sensores (Arreglo de sensores ver numeral 11.1.3.2) y enviar una escritura digital que al pasar un determinado umbral indicaba el estado del sensor, mediante la salida digital se enviaba el estado del arreglo de sensores a las entradas digitales del Arduino Mega, teniendo en cuenta que habían puertos digitales disponibles. A continuación en las *Figura 61* , podemos observar el diagrama que explica el desarrollo del algoritmo.(En la *Figura 62* se muestra el subprograma el cual es utilizado en los 3 sensores implementados anteriormente.)

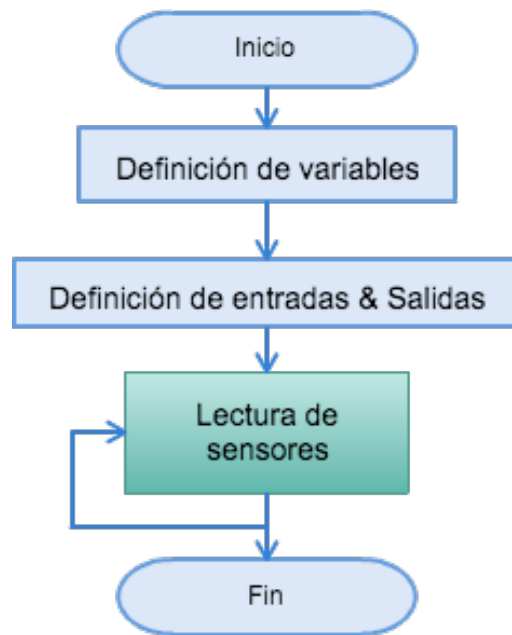


Figura 61 Diagrama de conversión análoga-digital del valor de los sensores en el Arduino nano.

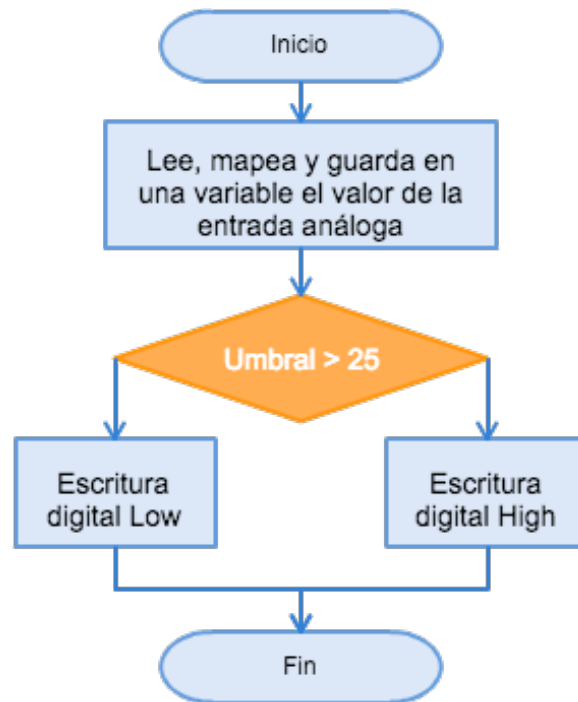


Figura 62 Subprograma lectura de sensores (ingresa el valor análogo de entrada del sensor para envío de estado).

En la *Figura 63* se puede observar el diagrama que muestra el sub programa, valor de entrada de puerto serial, el cual es un loop que pide como dato de entrada, el texto que proviene de la aplicación mediante Bluetooth por el puerto serial (Activity 4, 5, 6, 7, 8, 9, 10 (*Ver Numeral 11.1.9.3*)), dependiendo de ese valor se ingresa al respectivo sub programa, Tutorial Easy, Medium o Hard (*Figuras 65, 66 y 67*), a los sub programas Challenge Easy, Medium o Hard (*Figuras 69, 70 y 71*) o se envía la matriz de resultados a la aplicación, si esta lo pide desde el Activity 10 (*Figura 98*).

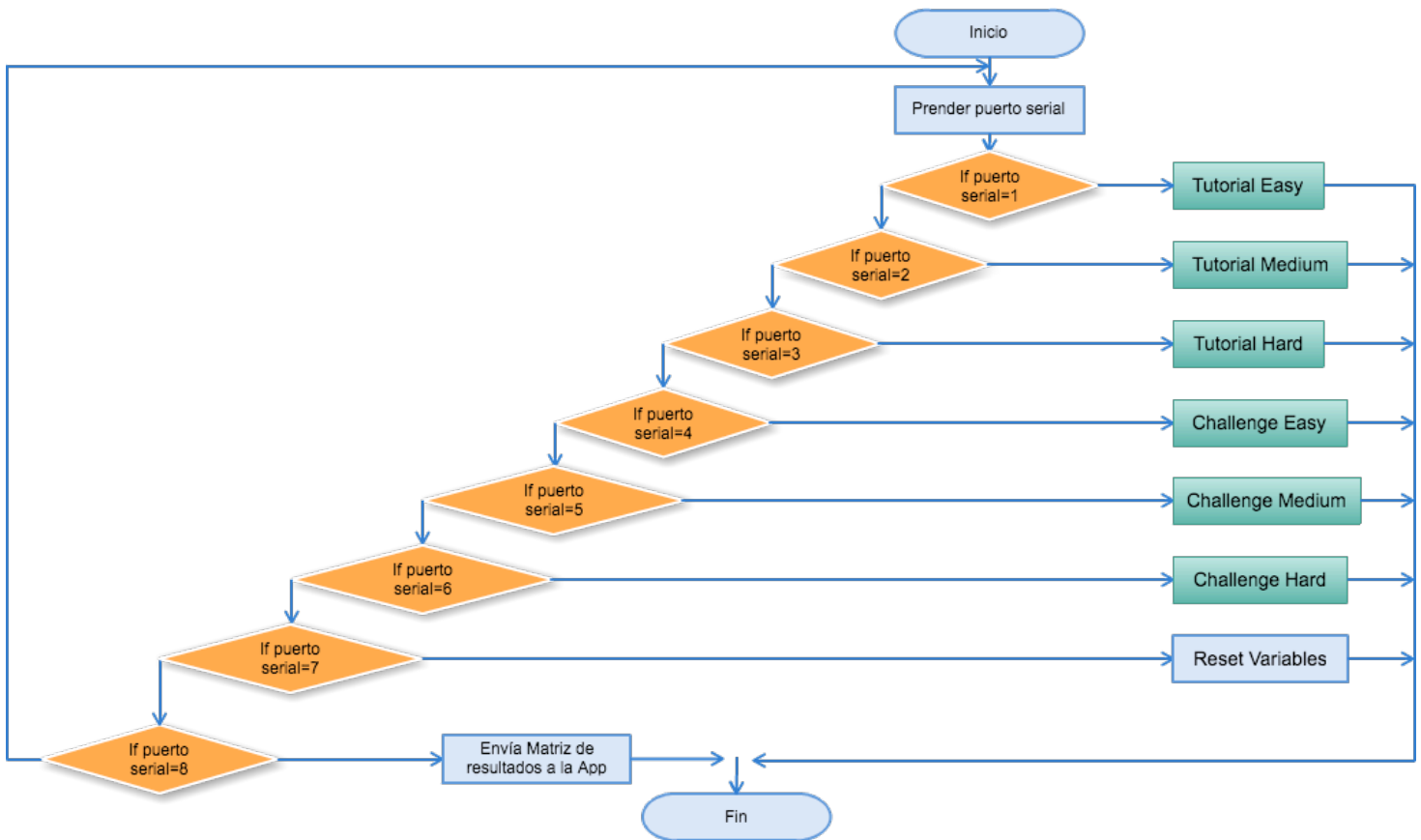


Figura 63 Diagrama de sub-programa (Valor de entrada de puerto serial).

La *Figura 64* muestra cada uno de los tres niveles en la etapa tutorial, cada nivel cuenta con diferente número de comparaciones (Frecuencia de entrada contra frecuencia pre establecida), es decir, notas interpretadas por el usuario traducidas como datos que en el sistema son frecuencias fundamentales extraídas de la entrada análoga.

Tutorial Easy	16 Comparaciones
Tutorial Medium	8 Comparaciones
Tutorial Hard	13 Comparaciones

Figura 64 Número de comparaciones en cada nivel.

En la *Figura 64* se aprecia el algoritmo cuyo funcionamiento aplica para el sub programa “Tutorial Easy”. Se envía el valor (i) SP de imagen (*Tabla 7*). Se obtiene el dato de entrada FF (frecuencia fundamental) (*Figura 72*) que genera el usuario a partir de la interpretación del instrumento y se realizarán las 16 comparaciones (*Tabla 7*) respectivas de forma ascendente, teniendo en cuenta que el programa solo deja pasar a la siguiente nota si el usuario acierta la nota pedida.

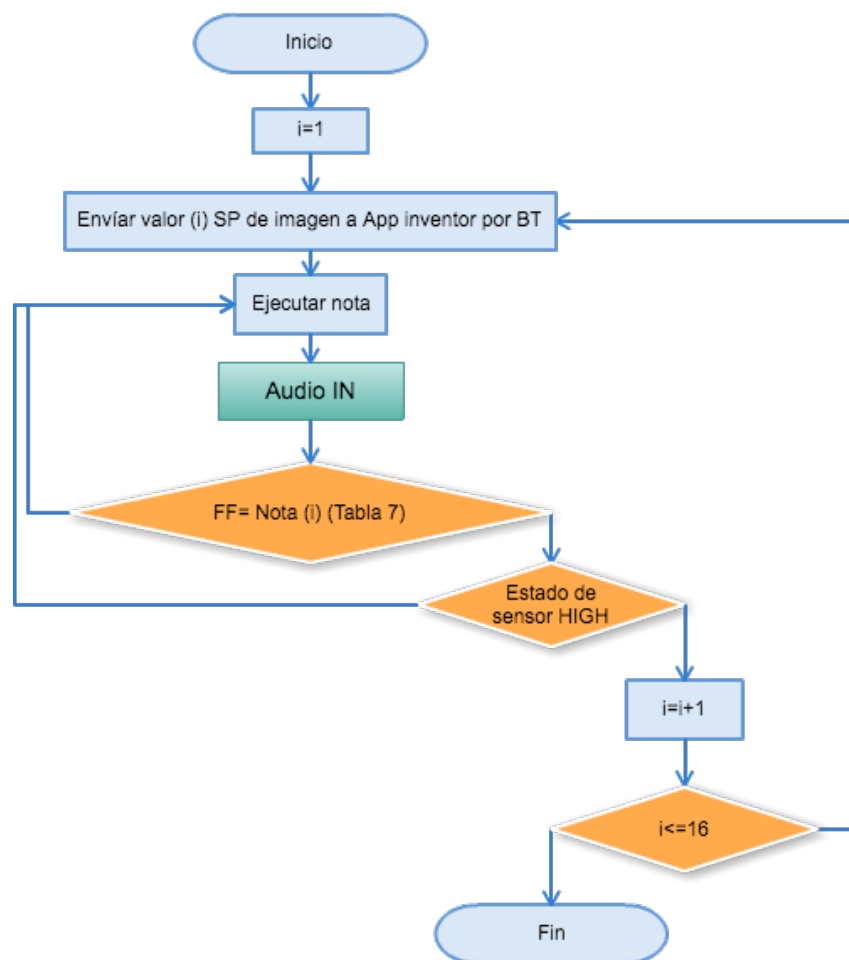


Figura 65 Diagrama para sub-programa Tutorial Easy.

En la siguiente *Tabla 7* se muestran los valores pre establecidos en orden de ejecución, para el nivel Tutorial Easy, que se compararán con los datos de entrada. Debido a que el

dato de entrada dependerá de la afinación del instrumento y del rango en frecuencia entre cada nota, se propone un espacio en el espectro para cada nota (*tabla 6*), o sea un margen de error de más o menos 1.2 Hz. ‘Valor (i) SP (serial print) de imagen’ es el carácter que se envía de Arduino al dispositivo móvil mediante Bluetooth.

Valor(i) SP de imagen	Nota (i)	Cuerda	Nota	Frecuencia(Hz)
1	1	E	F	42.7 a 44.5
2	2	E	F#	45 a 47
3	3	E	G	48.5 a 49.5
4	4	E	G#	51.5 a 52.5
5	5	A	A#	58 a 59
6	6	A	B	61.5 a 63
7	7	A	C	65 a 66.5
8	8	A	C#	68.5 a 70.5
9	9	D	D#	77 a 79
0	10	D	E	81.5 a 84
A	11	D	F	86 a 88.5
B	12	D	F#	90 a 94
C	13	G	G#	102 a 105
D	14	G	A	107 a 113
E	15	G	A#	115 a 118.5
F	16	G	B	122 a 125

Tabla 7 Frecuencias fundamentales (Nivel Tutorial Easy).

En la *Figura 66* se aprecia el algoritmo cuyo funcionamiento aplica para el sub programa “Tutorial Medium”. Se envía el valor (i) SP de imagen (*Tabla 8*). Se realizan las 8 comparaciones de forma ascendente (*Tabla 8*), teniendo en cuenta que el programa solo deja pasar a la siguiente nota si el usuario acierta.

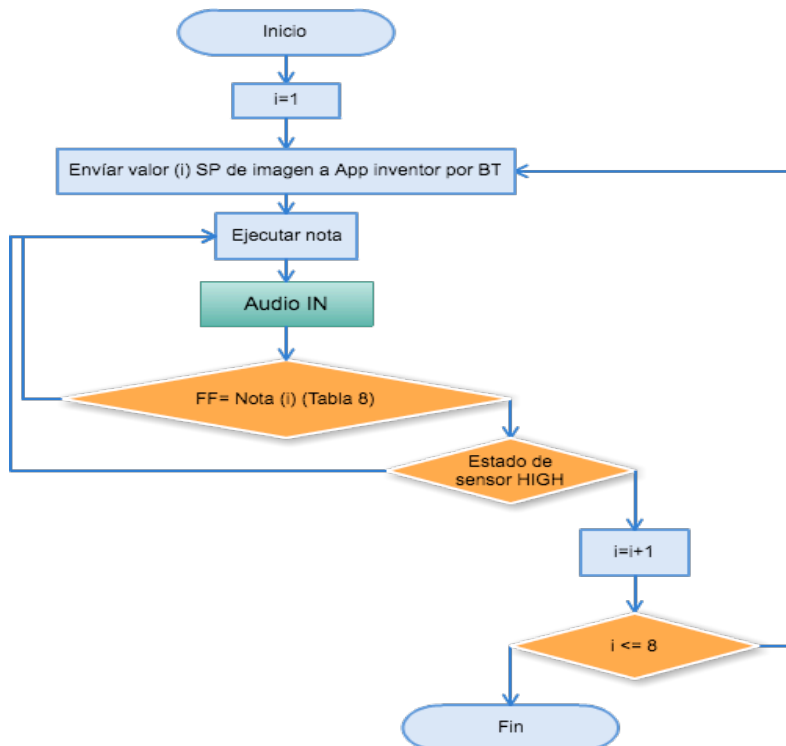


Figura 66. Diagrama para sub-programa Tutorial Medium.

En la siguiente *Tabla (8)* se muestran los valores pre establecidos, para el nivel Tutorial Medium, que se compararán con los datos de entrada. Debido a que el dato de entrada dependerá de la afinación del instrumento y del rango en frecuencia entre cada nota, se propone un espacio en el espectro para cada nota, o sea un margen de error de más o menos 1.2 Hz.

Valor(i) SP de imagen	Nota (i)	Cuerda	Nota	Frecuencia(Hz)
2	1	E	F#	45 a 47
5	2	A	A#	58 a 59
8	3	A	C#	68.5 a 70.5
9	4	D	D#	77 a 79
0	5	D	E	81.5 a 84
9	6	D	D#	77 a 79
8	7	A	C#	68.5 a 70.5
5	8	A	A#	58 a 59

Tabla 8. Frecuencias fundamentales (Nivel Tutorial Medium).

En la *Figura 67* se aprecia el algoritmo cuyo funcionamiento aplica para el sub programa “Tutorial Hard”. Se envía el valor (i) SP de imagen (*Tabla 9*). Se realizan las 13 comparaciones respectivas de forma ascendente (*Tabla 9*).

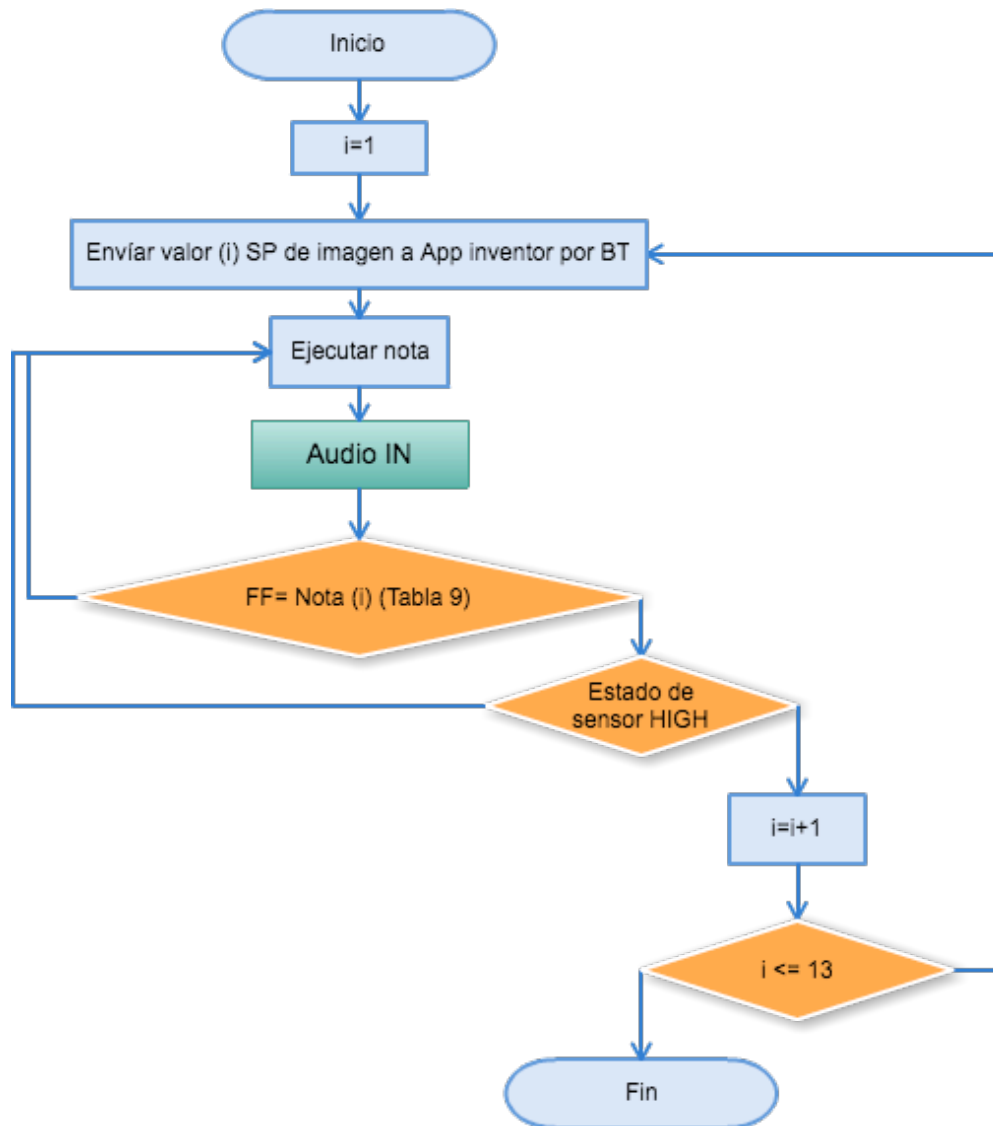


Figura 67. Diagrama para sub-programa Tutorial Hard.

En la siguiente *Tabla (9)* se muestran los valores pre establecidos, para el nivel Tutorial Hard, que se compararán con los datos de entrada. Se propone el margen de error mencionado, de más o menos 1.2 Hz.

Valor(i) SP de imagen	Nota (i)	Cuerda	Nota	Frecuencia(Hz)
5	1	A	A#	58 a 59
A	2	D	F	86 a 88.5
5	3	A	A#	58 a 59
8	4	A	C#	68.5 a 70.5
2	5	E	F#	45 a 47
A	6	D	F	86 a 88.5
2	7	E	F#	45 a 47
8	8	A	C#	68.5 a 70.5
5	9	A	A#	58 a 59
A	10	D	F	86 a 88.5
5	11	A	A#	58 a 59
8	12	A	C#	68.5 a 70.5
B	13	D	F#	90 a 94

Tabla 9. Frecuencias fundamentales (Nivel Tutorial Hard).

La *Figura 68* muestra cada uno de los tres niveles en la etapa Challenge, cada nivel cuenta con diferente número de comparaciones. Para los 3 casos el único cambio es el número de notas a ejecutar.

Challenge Easy	16 Comparaciones
Challenge Medium	8 Comparaciones
Challenge Hard	13 Comparaciones

Figura 68. Número de comparaciones en cada nivel.

En la *Figura 69* se aprecia el algoritmo para el sub programa “Challenge Easy”. Se obtendrá el dato de entrada FF (frecuencia fundamental)(*Figura 72*) que genera el usuario a partir de la interpretación del instrumento y se realizarán las 16 comparaciones(*Tabla 7*) respectivas de forma ascendente, el usuario podrá ejecutar cualquier nota y el programa lo dejará continuar, guardando su respuesta en una matriz.

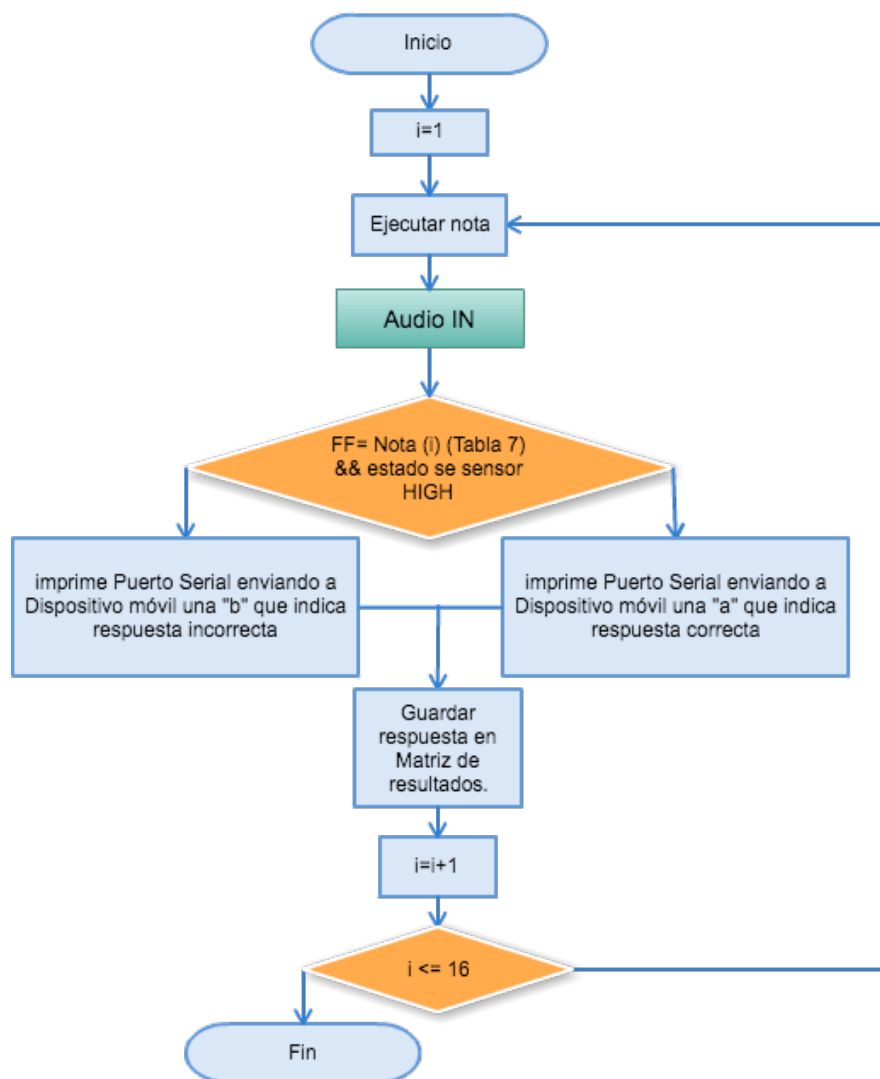


Figura 69. Diagrama de sub-programa Challenge Easy.

En la *Figura 70* se aprecia el algoritmo del sub programa “Challenge Medium”. Se realizarán las 8 comparaciones (*Tabla 8*) de forma ascendente, el usuario podrá ejecutar cualquier nota y el programa lo dejará continuar, guardando su respuesta en una matriz.

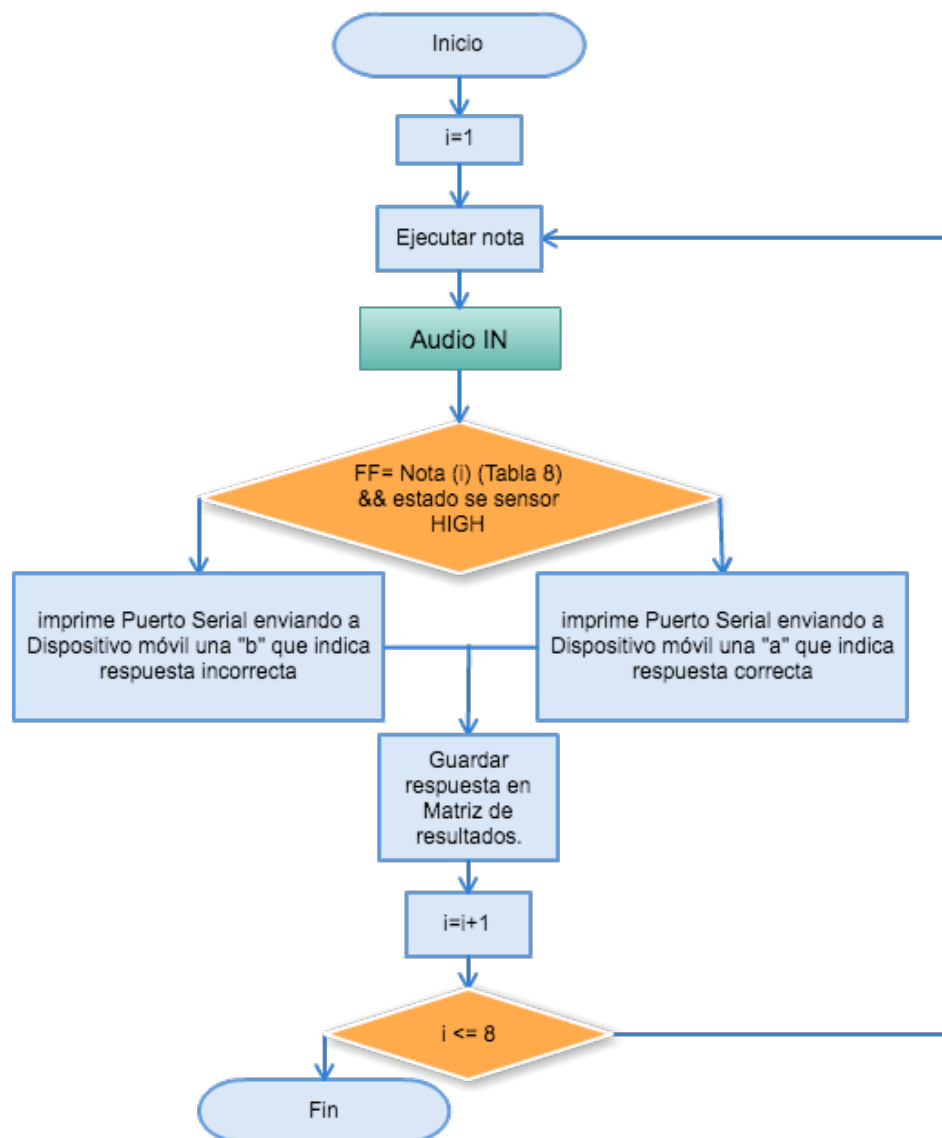


Figura 70. Diagrama de sub-programa Challenge Medium.

En la *Figura 71* se aprecia el algoritmo para el sub programa “Challenge Hard”. Se realizarán las 13 comparaciones (*Tabla 9*) respectivas de forma ascendente, el usuario podrá ejecutar cualquier nota y el programa lo dejará continuar, guardando su respuesta en una matriz.

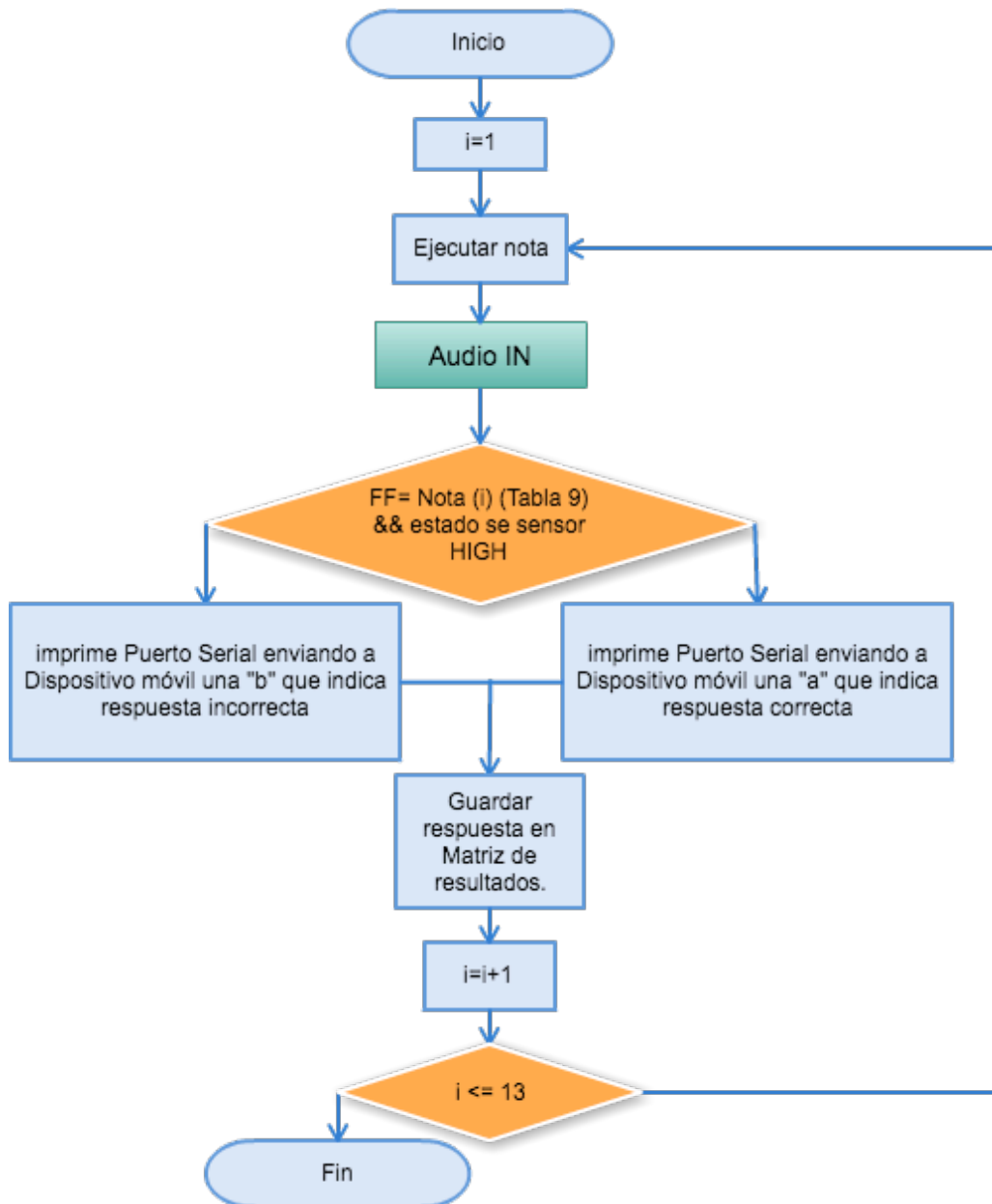


Figura 71. Diagrama de sub-programa Challenge Hard.

En la *Figura 72* se aprecia el sub programa Audio IN empleado en todos los niveles de las dos etapas. A partir de la conversión análoga digital y con una frecuencia de muestreo de 38.500 Hz, se obtendrá el valor FF que será una constante que indica la frecuencia fundamental de la señal que ingresa al sistema. En el anexo B se encuentra el código completo en Arduino.

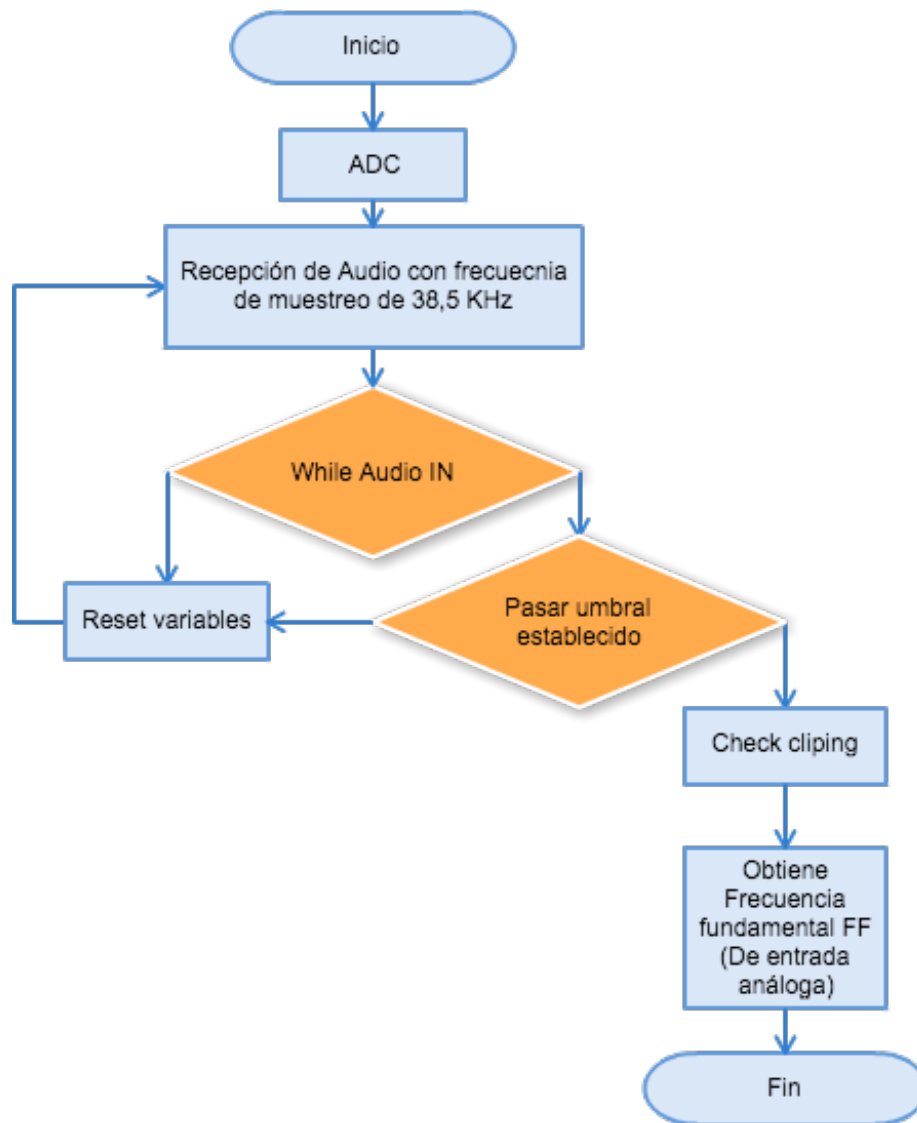


Figura 72. Diagrama de sub-programa (Audio IN).

11.1.4.4. Implementación del Arreglo de leds

Como complemento, además de la indicación que se aprecia en la *Figura 73*, dada en las Activities 4, 5 y 6 (*Ver numeral 11.1.9.3*) de la aplicación, en la etapa Tutorial se implementa un arreglo de leds en el hardware, el cual indicará de manera gráfica la nota a ejecutar (*Figura 74*).



Figura 73. Instrucción, ejecución de nota F en aplicación.

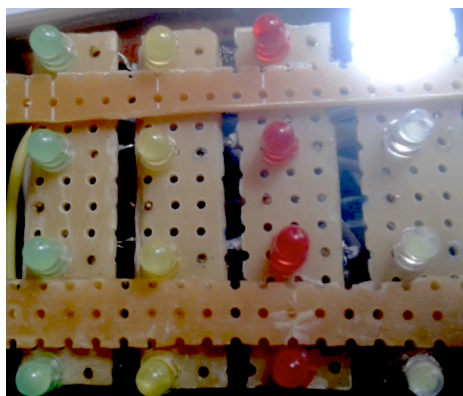


Figura 74. Arreglo de leds, instrucción de ejecución de nota F.

Se realizó la conexión de leds como se muestra en la *Figura 75* teniendo en cuenta las entradas usadas que se muestran en la *Figura 76*, donde el numero en el nombre de la variable quiere decir el número de cuerda y el número de traste, teniendo en cuenta que la

primera cuerda será MI, la segunda cuerda será La, la tercera cuerda será Re, y la cuarta cuerda será Sol.

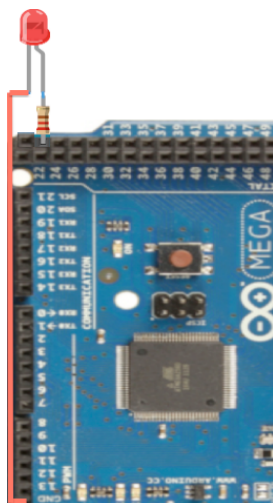


Figura 75 Conexión del led en Arduino.

Por ejemplo led11=2, quiere decir, led en la entrada 2 del micro controlador, que indica la posición del led en la cuerda 1(cuerda Mi), en el traste 1(Nota F).

```
led11=2, led12=6, led13=34, led14=38;
led21=3, led22=7, led23=35, led24=39;
led31=4, led32=8, led33=36, led34=40;
led41=5, led42=9, led43=37, led44=41;
```

Figura 76. Entradas empleadas para la conexión de los leds.

11.1.4.5. Implementación de visualizador de 7 segmentos

En la etapa Challenge el usuario tendrá como ayuda un visualizador de leds de 7 segmentos, el cual mostrará que nota se deberá ejecutar, la notación estará en cifrado americano. En la siguiente *Figura(77)* está el visualizador de cátodo común con punto decimal utilizado.



Figura 77. Visualizador led de 7 segmentos con punto decimal.

Como su nombre lo indica, es necesario que la tierra o ground sea general y para encender cada segmento se debe polarizar con 5 voltios el pin correspondiente. Con el fin de limitar la corriente se usa una resistencia entre el ánodo de cada diodo y el potencial positivo. En la *Figura 76* se observa la conexión.

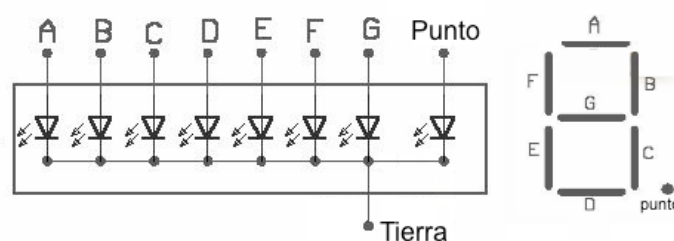


Figura 78 conexión visualizador led de 7 segmentos con punto decimal.

Un ejemplo claro puede ser observado en la *Figura 79*, donde el usuario tiene que ejecutar la nota Fa#, se tendría que enviar potencial positivo a los terminales A, E, F, G, con esto se visualizaría en notación, la letra F que representa el Fa en el cifrado americano y para expresar el sostenido se tendrá como referencia el punto decimal.



Figura 79. Nota Fa sostenido en visualizador.

11.1.5. Protocolo de comunicación

Teniendo en cuenta que la adquisición y transmisión de los datos se realiza de manera no guiada, se propone evidenciar cual es la transmisión indicada para la implementación del proyecto. En la *tabla 10* se presenta esta comparación.

IEEE	Velocidad de transferencia de datos	Estabilidad de red	Portabilidad de servicios	Distancia de cobertura
WIFI 802.11.b [41]	54 Mb / seg	-39 dBm	No	100 mts ideales
GSM IMT-2000 [43]	114 KB / seg	-71 dBm	Si	Depende cobertura de red
BLUETOOTH 802.15. [40]	720 / seg	-50 dBm	No	20 mts ideales

Tabla 10 Comparación de los protocolos no guiados [97]

- **Velocidad de transferencia de datos:** Refiere a los datos enviados por bytes/segundo.
- **Estabilidad de red:** Que tan fácil se pierde la conexión ente el servidor y el receptor las zonas libres de red, estructuras, perdidas por muro.

Interpretación de los valores.

En una escala de 0 a -80 RSSI (dBm):

- 0: señal ideal, difícil de lograr en la práctica.
 - 35 a -60: señal idónea con tasas de transferencia estables.
 - 60: enlace bueno; ajustando TX y Basic rates se puede lograr una conexión estable al 80%.
 - 70: enlace normal -bajo; es una señal medianamente buena, aunque se pueden sufrir problemas con lluvia y viento.
 - 80: es la señal mínima aceptable para establecer la conexión; puede ocurrir caídas, que se traducen en corte de comunicación (pérdida de llamada, pérdida de datos, mensajes (sms) corruptos (ilegibles). [98]
- Portabilidad de red: protocolo no dispone de tercero como empresas privadas para la prestación de cobertura.
- Distancia de cobertura: referente al rango de cobertura de la red del emisor hacia el receptor.

11.1.5.1. Implementación de protocolo de comunicación

Una excelente ventaja que tiene la tecnología Bluetooth es el salto de frecuencia. Esto quiere decir que Bluetooth posee una técnica que evita la interferencia ya que en el entorno donde se maneja Bluetooth es ISM y esta es una banda libre, por lo que la tecnología divide la banda de frecuencia en diferentes canales de salto. Esto permite que los transceptores salte de un canal a otro pero estos lo hacen de manera aleatoria, y estos

paquetes se encuentran seguros por un sistema interno llamado ARQ(Automatic retransmission request) es decir si en algún momento se pierde alguno de los paquetes de envió este se retransmite de nuevo. El protocolo que se usará será Bluetooth, debido a que el protocolo WI-FI cubre los requerimientos pero presenta gastos extras, ya que es necesario adquirir un host y una ip fija, o sea un dominio.

La transmisión de datos vía Bluetooth es eficiente para la implementación del proyecto, ya que tiene señal idónea con tasas de transferencia estables [98], además de la compatibilidad que presenta la mayoría de dispositivos disponibles en el mercado, incluyendo los ordenadores, cubriendo los requerimientos nombrados.

11.1.5.2. Medición de intensidad de la señal, protocolo de comunicación

Bluetooth

Para la medición del RSSI de los diferentes protocolos no guiados, se hace uso de aplicaciones para dispositivos móviles los cuales muestran la intensidad de señal, y así se obtiene el valor en dBm y se determinan los parámetros de selección para el protocolo de comunicación [49]. Para realizar la medición del protocolo no guiado de comunicación Bluetooth, cumpliendo los parámetros establecidos, se procedió utilizando la aplicación “Bluetooth signal” descargada del Play Store para Android. Donde se estableció comunicación entre un dispositivo móvil y un computador portátil, tomando dos valores de RSSI con distancias diferentes, la primera medición (*Figura 80*) se realizó tomando el dispositivo móvil justo en frente del computador y la segunda (*Figura 81*) a

un metro de distancia entre los dispositivos. Obteniendo valores de intensidad de señal, los cuales se muestran a continuación.

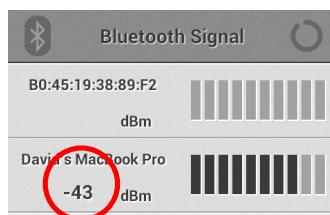


Figura 80. RSSI Bluetooth, estabilidad de red, medición 1.

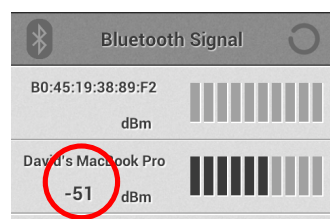


Figura 81 RSSI Bluetooth, estabilidad de red, medición 2.

11.1.5.3. Implementación de modulo Bluetooth.

Para la conexión entre un Smartphone o dispositivo móvil y una placa con micro-controlador de Arduino, es necesario tener en cuenta factores como:

- Que el Smartphone posea Bluetooth.
- El Arduino debe tener un módulo Bluetooth para el envío y recepción de información, que va del dispositivo móvil a Arduino y viceversa.

Para la implementación de esta conexión se utilizó el modulo Bluetooth de Arduino Hc 06 (*Numeral 7.1.10*) y un Smartphone (*Figura 85*) el cual debe poseer modulo Bluetooth.

11.1.5.4. Prueba de modulo Bluetooth

Como ya se había hablado anteriormente (*Numeral 7.1.10*) el módulo Hc 06 posee 4 pines diferentes que son: Vcc que hace referencia a 5v, GND o tierra que posee un voltaje de 0v, Tx y Rx que son los pines seriales, esto quiere decir que toda la información que reciba será pasada al Arduino de forma serial.

En la *Figura 82* se aprecia la conexión, la cual va de la siguiente forma:

- GND o la tierra del módulo va directo a GND del Arduino
- Vcc va a los 5v de alimentación que ofrece arduino.
- Tx del módulo ira directo a la entra 0 que es Rx0 en el Arduino
- Rx del módulo ira a la entrada1 que corresponde a Tx0 en el Arduino.

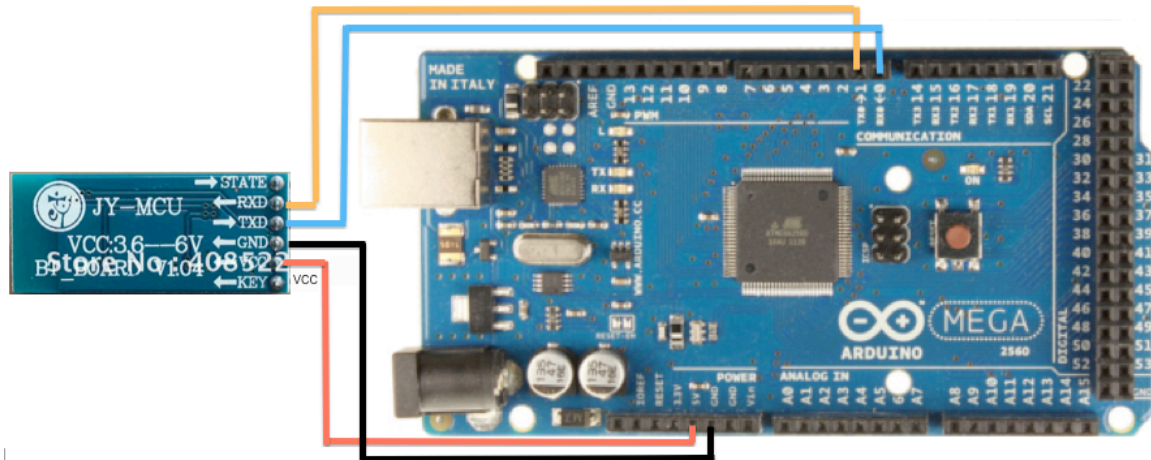


Figura 82. Conexión módulo Hc 06 a ARDUINO. [24] [27]

Para la conexión del dispositivo móvil a Arduino se hicieron dos programas uno en un Sketch de Arduino (*Figura 84*) y otro en App inventor (*Figura 83*).

El programa de prueba que se hizo en App Inventor (*Figura 83*) se diseñó con el fin de probar si exista una verdadera conexión entre el dispositivo móvil y el módulo Hc 06 de Arduino. Lo que se hace básicamente es enviar un texto por el puerto serial, teniendo como opción el número 1 y el número 2.

App Inventor es un software online gratuito que permite el desarrollo y diseño de aplicaciones. La aplicación que se hizo y se probó en el dispositivo móvil.

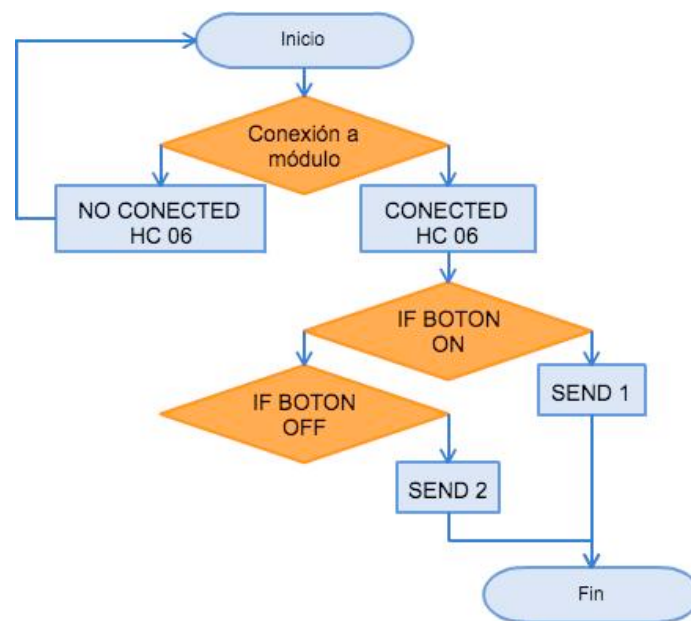


Figura 83. Algoritmo App inventor.

El segundo programa que se desarrolló fue el de Arduino esto se hizo con el fin de que los datos enviados desde el dispositivo móvil sean recibidos por el módulo Hc 06 y este transmita estos datos a la placa de Arduino, que poseía en su conexión un Led, que dependiendo a la orden que se daba en el dispositivo móvil prendía o apagaba.

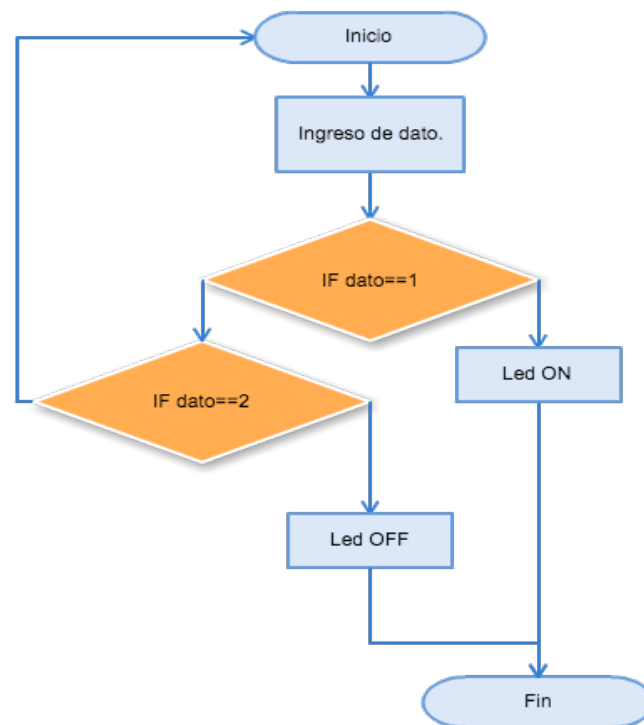


Figura 84 Algoritmo SKETCH Arduino.

11.1.6. Sistema Operativo.

11.1.6.2. Selección del sistema operativo

Al momento de hacer la implementación del sistema operativo seleccionado se hace el análisis diferencial con los sistemas iOS y Windows Phone, dando como resultado las siguientes ventajas:

- El acceso a diferentes software de programación legal, que pueden ser pagos y gratuitos tal es el caso de (App Inventor). Dando así una infinidad de opciones de programar de acuerdo a lo que el usuario necesite.
- Da acceso a programación online por medio de lenguaje de bloques teniendo opciones básicas para programar.

- Permite el desarrollo y distribución de la aplicación al Play Store totalmente gratis, sin tener que esperar el tiempo de revisión para ser aprobada.
- La compatibilidad entre diferentes versiones de sistemas Android. Sus programas podrán ejecutarse en cualquier dispositivo con Android.
- La gran mayoría de funciones, drivers y demás se encuentran fácilmente, donde pueden ser gratuitos o pagos de excelente calidad.
- Las librerías y documentación actualizadas y disponibles para el programador.

11.1.7. Dispositivo móvil

Se adquiere un dispositivo móvil (Smartphone) (*Figura 85*) el cual es compatible y maneja el mismo sistema Android. Este cuenta con lo requerido puede recibir y enviar datos por medio de Bluetooth.



Figura 85. Dispositivo móvil implementado.

11.1.7.2. Selección de dispositivo móvil

Ficha Técnica

ATRIBUTOS	DETALLES
Modelo	Galaxy S3 mini (Samsung)
Sistema operacional	Android 4.1 Jelly Bean
Velocidad de procesador	1 GHz
Memoria interna	16 GB
Memoria RAM	1 GB
Tipo de Pantalla	480 x 800
Táctil	Pantalla táctil
Conexión 3G	No
Wi-Fi	Si
Bluetooth	Si
Altavoz incorporado	Si
Cámara web	Si
Micrófono incorporado	SI
Puerto para audio	Si

Tabla 11 Ficha técnica, dispositivo móvil implementado.

Para la implementación de un Smartphone al sistema no se tuvo que hacer una búsqueda exhaustiva, ya que la mayoría de Smartphone o Dispositivos portátiles táctiles poseen Android y tienen un mínimo de memoria interna o externa, pueden ser utilizados en este sistema sin tener complicación alguna.

11.1.8. Entorno de programación

11.1.8.2. Diseño de algoritmo de programación

App inventor es una plataforma de Google Labs para realizar aplicaciones para el sistema operativo Android, está diseñado para desarrolladores que deseen implementar y ejecutar

aplicaciones para el sistema operativo Android, a partir de una serie de herramientas el usuario puede realizar diagramas de bloques para realizar la aplicación. El sistema es gratuito y se puede utilizar en modo online. Para el desarrollo de este proyecto se utilizará App inventor, ya que suple con las necesidades y requerimientos descritos anteriormente.

11.1.8.3. Diseño de algoritmo para programar en App Inventor

La aplicación desarrollada en App Inventor tiene el nombre de Bass Master. Para el diseño se utilizaron 10 activities. A continuación se expone cada activity y su visualización en el dispositivo móvil.

Activity 1: Se muestra en la *Figura 86*. Esta activity se diseñó con el fin de presentar un menú de la aplicación, este menú consta de 2 botones que al seleccionarlos nos llevan a al menú de tutorial o al menú de prueba.

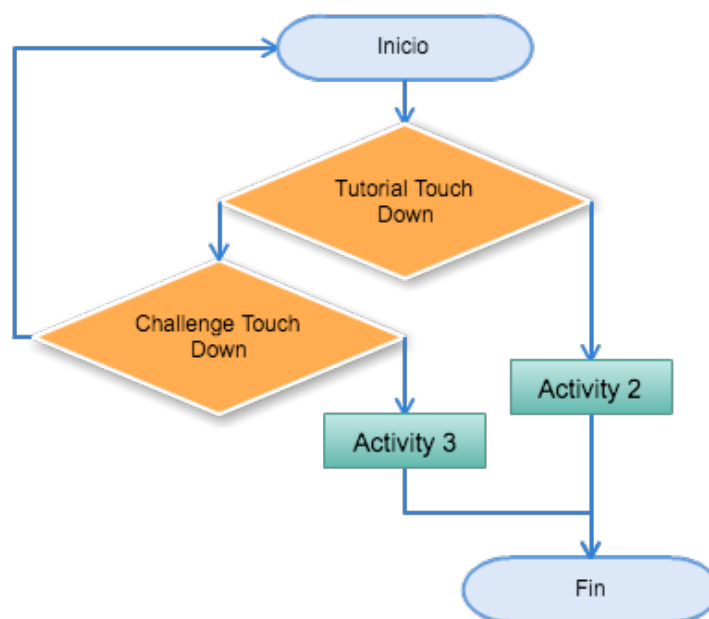


Figura 86 Algoritmo del sub programa Activity 1.

En la *Figura 87* se aprecia la activity 1 como interfaz gráfica de usuario implementada en el dispositivo móvil. Es una captura de pantalla, donde se encuentran las piezas diseñadas entre ellas los dos botones mencionados.

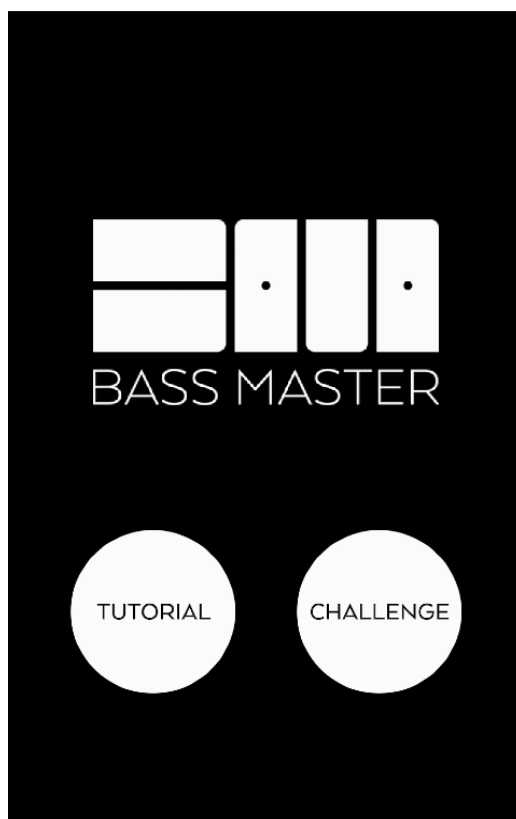


Figura 87 Activity 1 vista en el dispositivo móvil.

La figura 88 muestra el diagrama de sub menú de selección de nivel, que es igual tanto para la parte Tutorial (Activity 2) como para la parte challenge (Activity 3). Esto debido a que cuentan con los mismos niveles, es decir, hay un menú de selección de nivel, EASY, MEDIUM y HARD para cada etapa (Tutorial y Challenge), además está el botón de regreso al menú principal o activity 1.

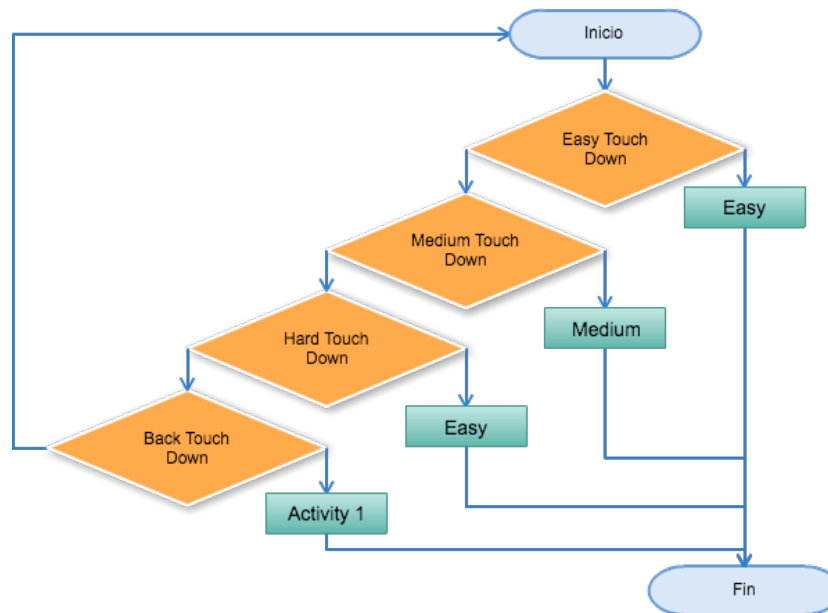


Figura 88 Algoritmo actividad 2 y 3.

En la *Figura 89* se aprecian las Activities 2 y 3, como interfaz gráfica de usuario, implementadas en el dispositivo móvil. Se visualiza una captura de pantalla, donde se encuentran las piezas diseñadas, entre ellas los 4 botones mencionados.

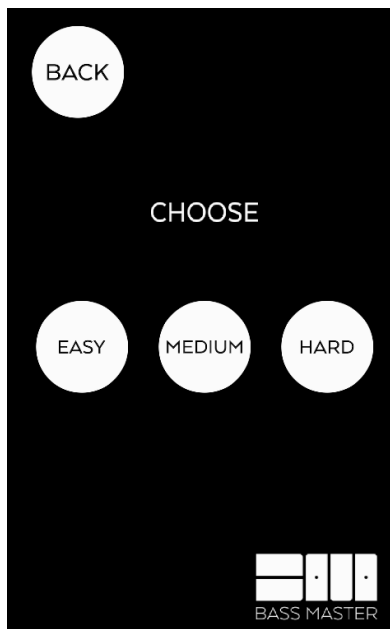


Figura 89 Activity 2 y 3 vistas en el dispositivo móvil.

Activity 4: La *Figura 90*, es la activity selección de EASY de la etapa tutorial, en esta se puede observar que se encuentran tres botones, uno que es llamado Bluetooth, otro Start y Back.

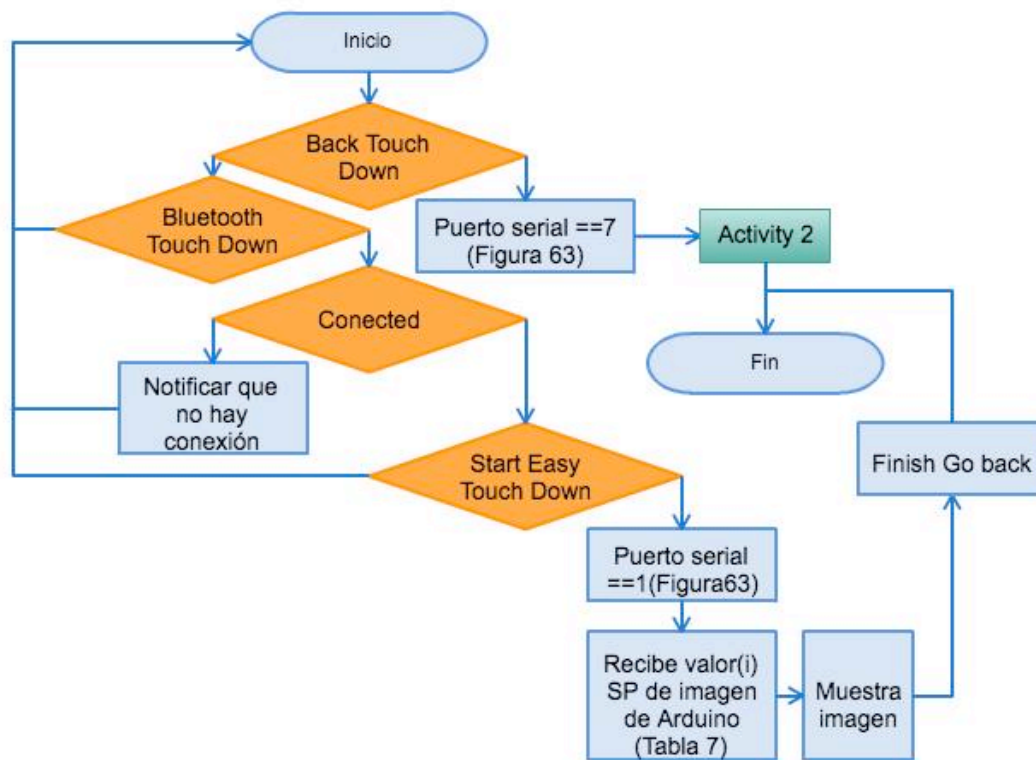


Figura 90 Activity 4. Diagrama algoritmo Tutorial Easy.

Bluetooth es un botón que nos da la posibilidad de conectar la App al módulo Bluetooth implementado en el sistema con el fin de recibir datos desde Arduino por medio de tecnología Bluetooth. El botón de Start Easy es el inicio de una serie sucesiva de explicaciones al usuario, es decir al tocar el botón start, se envía el valor '1', y se recibe 'valor (i) SP de imagen' (*Tabla 7*), mediante Bluetooth, para seleccionar (de una matriz en la que están guardadas las imágenes, *Tabla 12*) la respectiva gráfica y así empezar una

secuencia de instrucciones, en total 16, que el usuario debe imitar en el instrumento.

Si la ejecución de la explicación es correcta la imagen cambia y de este modo se recibe un nuevo dato desde el micro controlador (valor (i) SP de imagen, *Tabla 7*) dando al usuario la siguiente imagen o instrucción.

A continuación una tabla que muestra los valores en la matriz de app inventor que llaman a la respectiva imagen.

Valor SP(serial print) de imagen	Nombre de la imagen
1	Nota 1 Fa
2	Nota 2 Fa#
3	Nota 3 G
4	Nota 4 G#
5	Nota 5 A#
6	Nota 6 B
7	Nota 7 C
8	Nota 8 C#
9	Nota 9 D#
0	Nota 10 E
A	Nota 11 F
B	Nota 12 F#
C	Nota 13 G#
D	Nota 14 A
E	Nota 15 A#
F	Nota 16 B

Tabla 12. Matriz en App inventor y nombre de las imágenes.

En el centro de la actividad 4 (*Figura 91*), como ejemplo se muestra la imagen de una mano con sus dedos de diferente color, esto representa la mano del usuario y que dedo debe usar para pulsar la respectiva nota, el nombre de la imagen es 'Nota 1 Fa'.

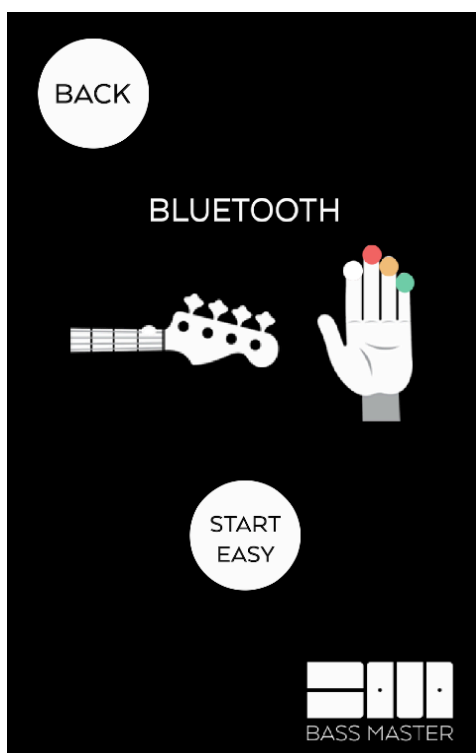


Figura 91. Activity 4. Visualización en la aplicación de Tutorial Easy Nota 1 Fa.

Esta actividad *Figura 91* tiene un botón de Back que en este caso, lleva al usuario al Sub menú de la parte tutorial (activity 2, *Figura 89*).

Activity 5: (*Figura 92*) posee las mismas funciones que la actividad 4, pero en lo que respecta a cambio de posiciones (pulsación de traste) en el instrumento y número de notas o imágenes para visualizar, varían y tendrán mayor grado de dificultad dependiendo del nivel seleccionado.

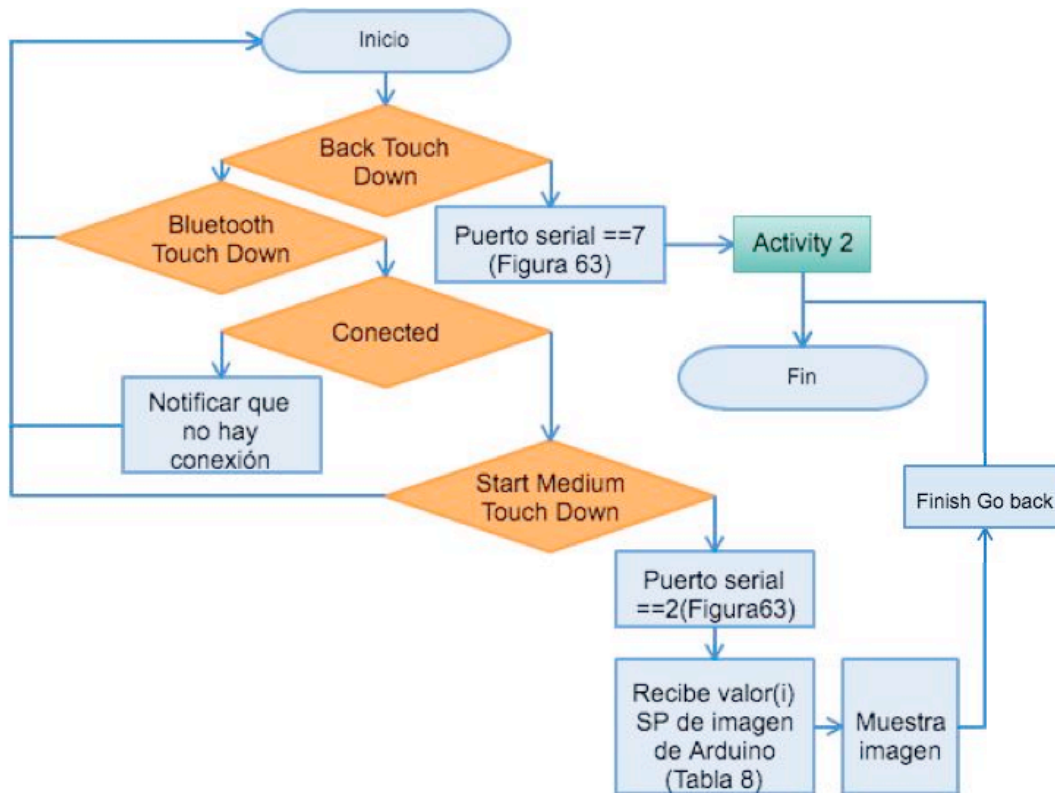


Figura 92 Activity 5. Diagrama algoritmo Tutorial Medium.

El botón de Start Medium es el inicio de una serie sucesiva de explicaciones al usuario, es decir al tocar el botón start medium, se envía el valor '2', y se recibe 'valor(i) SP de imagen'(tabla 8), mediante Bluetooth, para seleccionar (de una matriz en la que están guardadas las imágenes, *Tabla 12*) la respectiva gráfica y así empezar una secuencia de instrucciones, en total 8, (Se realizan en el sub programa de Arduino llamado Tutorial Medium (*Figura 66*)) que el usuario debe imitar en el instrumento. Si la ejecución de la explicación es correcta la imagen cambia y de este modo se recibe un nuevo dato desde el micro controlador (valor (i) SP de imagen, *Tabla 8*) dando al usuario la siguiente imagen o instrucción.

Activity 6: (Figura 93) posee las mismas funciones que la actividad 4, pero en cuanto se presiona el botón de Start Hard da inicio a una serie sucesiva de explicaciones al usuario, es decir al tocar el botón Start Hard, se envía el valor '3', y se recibe 'valor(i) SP de imagen' (Tabla 9), mediante Bluetooth, para seleccionar (de una matriz en la que están guardadas las imágenes, Tabla 12) la respectiva gráfica y así empezar una secuencia de instrucciones, en total 13, (Se realizan en el sub programa de Arduino llamado Tutorial Hard (Figura 67)) que el usuario debe imitar en el instrumento. Si la ejecución de la explicación es correcta la imagen cambia y de este modo se recibe un nuevo dato desde el micro controlador (valor (i) SP de imagen, Tabla 9) dando al usuario la siguiente imagen o instrucción.

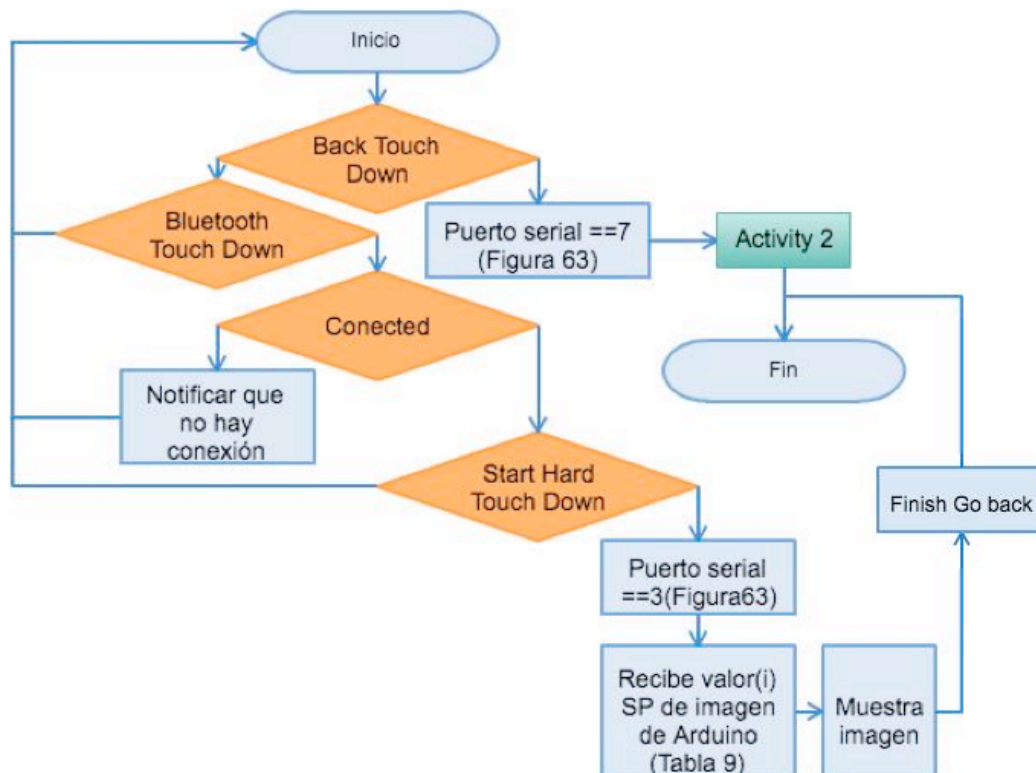


Figura 93. Activity 6. Diagrama algoritmo Tutorial Hard.

Activity 7: Como se muestra en la Figura 94, esta activity es la selección del nivel EASY de la etapa challenge, se puede observar que se encuentran tres botones uno que es llamado Bluetooth, Start y Back; Bluetooth es un botón que nos da la posibilidad de conectar el dispositivo móvil al módulo Bluetooth implementado en el sistema, con el fin de recibir datos del Arduino por medio de tecnología Bluetooth. El botón de Start Easy es el comienzo de probar lo explicado en el etapa tutorial, es decir al tocar el botón start easy el usuario debe tocar en el instrumento la misma secuencia ascendente realizada en la etapa tutorial easy. Si la ejecución es correcta se recibirá un carácter 'a' (Figura 65) desde el micro controlador indicando que la respuesta es correcta y así aparecerá una imagen verde en la pantalla de dispositivo móvil, que quiere decir que la ejecución de la nota fue correcta pero si se recibe un carácter 'b'(Figura 65) la imagen será de color rojo quiere decir que la ejecución de esa nota fue errónea.

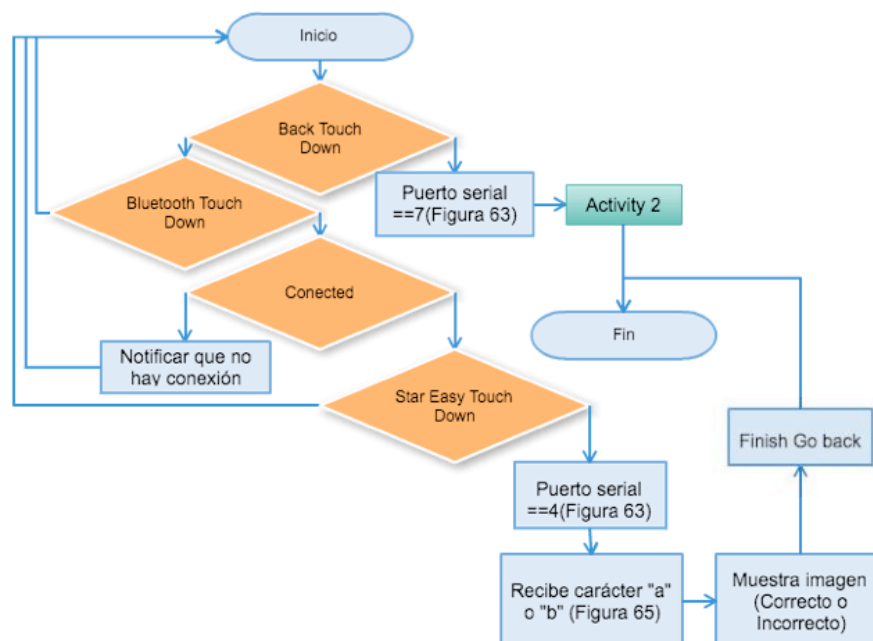


Figura 94. Activity 7. Diagrama algoritmo Challenge Easy.

Activity 8: Esta activity es la selección del nivel MEDIUM de la etapa challenge.

Tiene la misma estructura de la activity 7. El botón de Start Medium (Figura 95) es el

comienzo de probar lo explicado en el etapa tutorial, es decir al tocar el botón start

Medium el usuario debe tocar en el instrumento la misma secuencia ascendente realizada

en la etapa tutorial medium. Si la ejecución es correcta se recibirá un carácter 'a' (Figura

66) desde el micro controlador indicando que la respuesta es correcta y así aparecerá una

imagen verde en la pantalla de dispositivo móvil, que quiere decir que la ejecución de la

nota fue correcta si se recibe un carácter 'b' (Figura 66) la imagen será de color rojo

quiere decir que la ejecución de esa nota fue errónea.

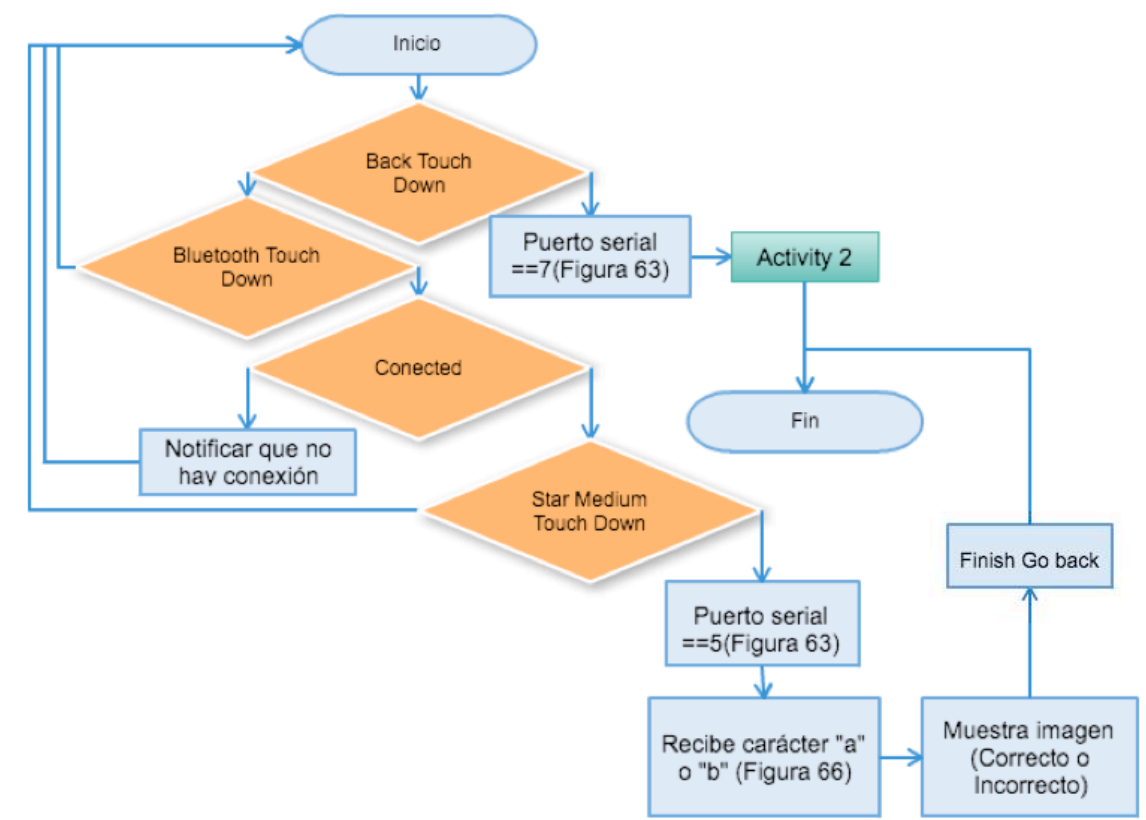


Figura 95. Activity 8. Diagrama algoritmo Challenge Medium.

Activity 9: Esta activity es la selección del nivel Hard de la etapa challenge. Tiene la misma estructura de la activity 7. Al tocar el botón start Hard el usuario debe tocar en el instrumento la misma secuencia ascendente realizada en la etapa tutorial Hard. Si la ejecución es correcta se recibirá un carácter 'a' (Figura 67) desde el micro controlador indicando que la respuesta es correcta y así aparecerá una imagen verde en la pantalla de dispositivo móvil, que quiere decir que la ejecución de la nota fue correcta pero si se recibe un carácter 'b' (Figura 67) la imagen será de color rojo quiere decir que la ejecución de esa nota fue errónea.

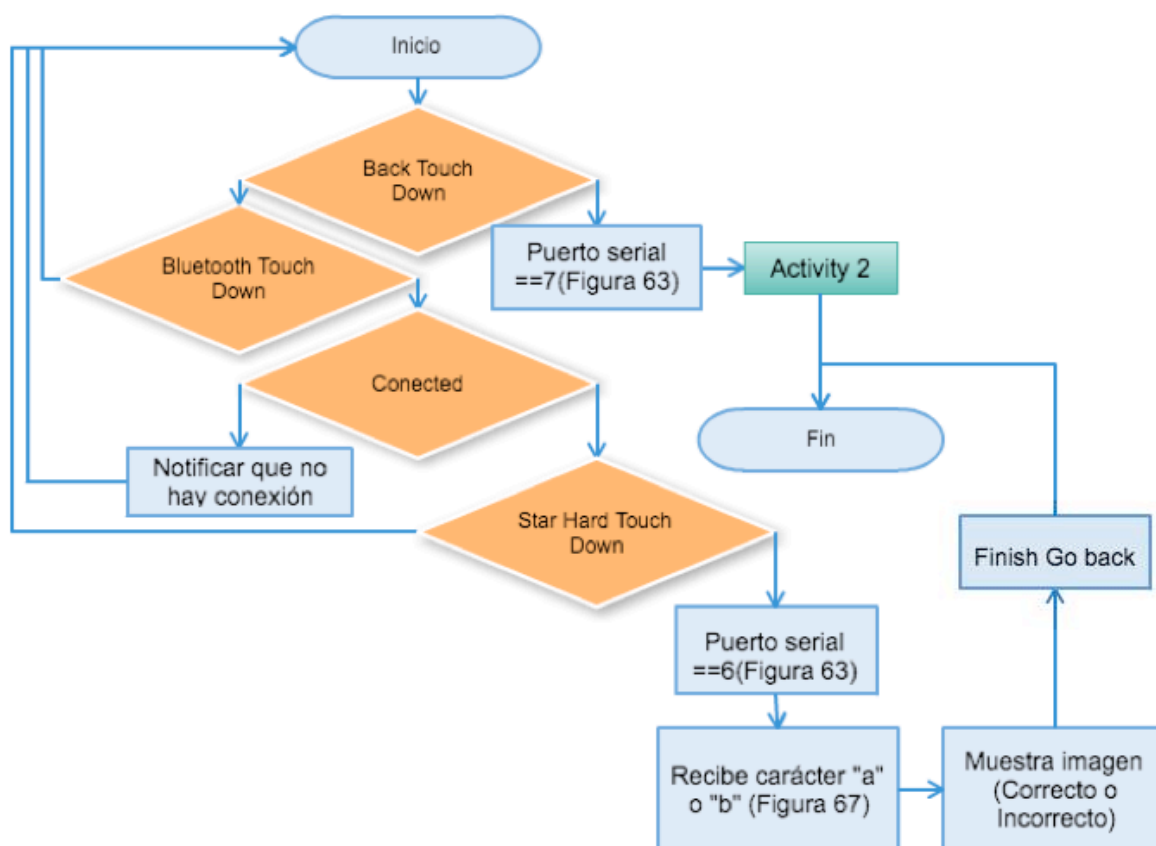


Figura 96. Activity 9. Diagrama algoritmo Challenge Hard.

A continuación la *Figura 97* muestra un ejemplo, de cómo se visualiza en el dispositivo móvil la activity 7. Es una ejecución de nota correcta, en la etapa challenge en el nivel easy.

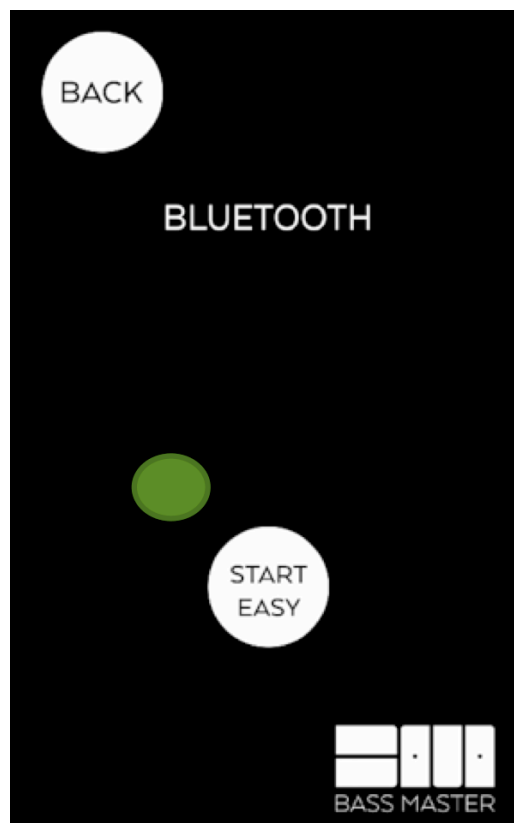


Figura 97. Ejemplo de selección de etapa challenge, nivel easy.

Activity 10: Esta activity aparece siempre y cuando se despliegue un mensaje warning al final de cualquier nivel del modo challenge, al dar “ok” aparecerá una activity como se muestra en la Figura 98, esta posee dos botones, un botón de back para regresar a la Activity 2 o 3 Figura(89), dependiendo de la activity donde se esté ubicado, es decir si se está en la etapa tutorial en el nivel Easy al oprimir back se abrirá la activity 2.

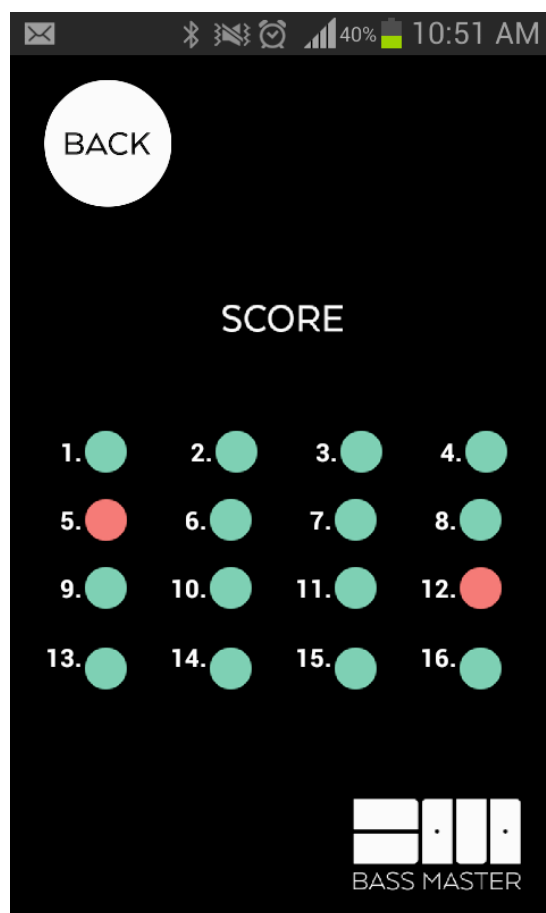


Figura 98. Resultados BassMaster.

En la *Figura 99* se muestra el diagrama del algoritmo de la actividad 10. Que además cuenta con un botón de score, el cual al ser pulsado enviará el carácter ‘8’ desde el dispositivo móvil hasta el micro controlador (*Figura 63*) por el puerto serial, mediante Bluetooth y de este modo se generara la recepción de la matriz de resultados desde el micro controlador mostrando el reporte de interpretación que podrá ser observado por el usuario, donde encontrará la cantidad de errores y aciertos que tuvo. *Figura 1* Activity 10. Reporte de interpretación (aciertos y errores de notas ejecutadas).

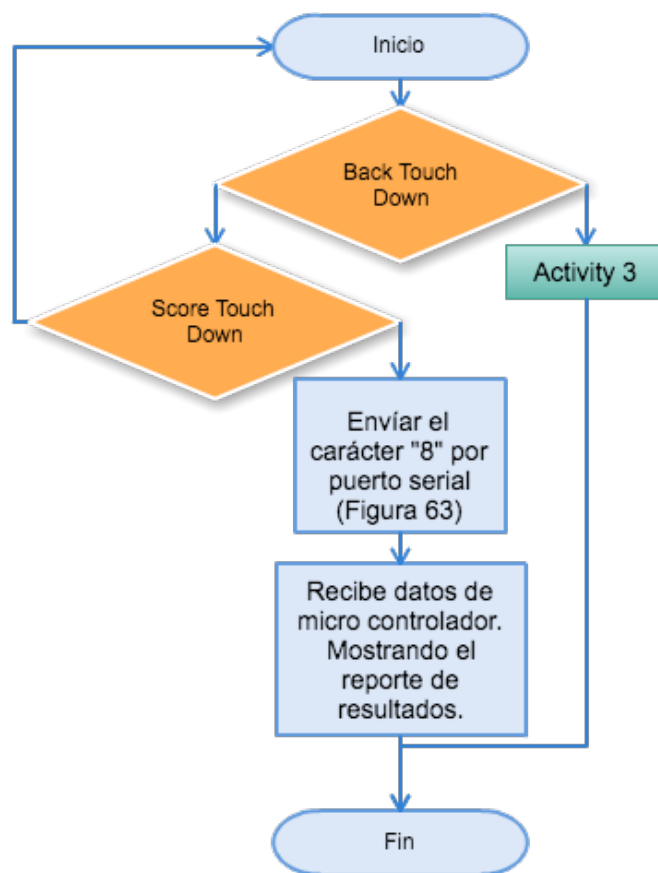


Figura 99. Activity 10. Reporte de interpretación.

11.1.9. Unión de subsistemas y consolidación del sistema físico.

Mediante el protocolo Bluetooth, se realizó la comunicación de dispositivos entre la aplicación y el Arduino, dada la comunicación, se podrán generar en Arduino las ayudas que constan de: Una visualización Led ó arreglo de leds en modo “tutorial” el cual se explicó en el *Numeral 11.1.5.4* y en el modo “challenge” el Display de 7 segmentos (*Numeral 11.1.5.5*).

Para la captura de la señal del bajo eléctrico y para el análisis de resultados se tomaron dos parámetros, con el fin de ser mas acertados en la obtención de datos. El primer parámetro consta del arreglo de sensores insertados en el bajo eléctrico (*Numerales 11.1.3.3*), en donde se obtiene el valor de estado (ON / OFF) de cada uno de los sensores,

con esto se sabrá cual traste se pulsa. El segundo parámetro será un detector que indica la frecuencia de la nota ejecutada por el usuario, para su funcionamiento se hizo necesaria la implementación de un pre amplificador de dos etapas (*Numeral 11.1.4.3*) y un elevador de nivel DC (*Figura 46*) A partir del estado y la comparación de las frecuencias obtenidas mediante el detector (*Figura 72*), se podrá realizar el proceso en Arduino para generar la matriz de aciertos y errores y visualizar un reporte de resultados (*Figura 98*) en el dispositivo móvil.

Nota: en el anexo B se encuentra un video de lo explicado en este *Numeral (11.1.10.)*

12. REPRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS.

12.1. Medición subjetiva

La medición subjetiva que se realizó es un estudio de tipo descriptivo y cualitativo. La población de estudio estuvo constituido por 105 estudiantes de últimos semestres (9-10 semestre) de licenciatura en música, en la universidad de Nariño sede Pasto. Estos estudiantes cumplen el criterio requerido para este tipo de encuesta (encuesta de satisfacción), ya que son personas con “conocimientos en teoría musical”. La muestra obtenida fue probabilística, la selección de la muestra por cada grupo se realizó mediante el método de estratificación ya que la población se dividió en 2 tipos de estudiantes, los cuales fueron: Grupo 1. Estudiantes que saben ejecutar el bajo eléctrico y Grupo 2. Estudiantes que no saben ejecutar el bajo eléctrico. Se realizó una operación con ayuda de una de las fórmula de tamaño muestral *Ecuación 29*, con variable cualitativa a un nivel de confianza del 95% y un nivel de error del 5%; se hace de esta forma para evitar pérdidas de tiempo y dinero; la fórmula elegida fue la siguiente:

$$\eta = \frac{Z\alpha^2 Npq}{e^2(N-1) + Z\alpha^2 pq}$$

Ecuación 29 Fórmula muestral cualitativa conociendo el número de población.

De la población total, la muestra que se seleccionó es de 83 estudiantes donde el 69,88 % eran pertenecientes al Grupo 1. Y el 30,12 % pertenecía al Grupo 2; es decir 58 personas eran pertenecientes al Grupo 1 y 25 personas al Grupo 2.

Análisis de datos tomados a 83 estudiantes de licenciatura en música interactuando con el sistema Bass Master: Para este análisis se tomó como base una encuesta de 5 preguntas las cuales se plantearon para dar su apreciación acerca del funcionamiento y el punto de vista (análisis subjetivo) de dos tipos de muestreo (Grupo 1. Y Grupo 2.) con respecto al sistema implementado en este proyecto, el formato de la encuesta se encuentra en el anexo C.

Como anteriormente se hizo énfasis, el tamaño de muestreo es de 83 personas y estratificadas en dos grupos donde el Grupo 1. Tiene 58 personas y grupo 2. Tiene 25. Después de analizar el resultado de las encuestas obtuvimos las siguientes gráficas y apreciaciones:

La siguiente *Tabla 13* es la tabulación de respuestas en la encuesta hechas por el “Grupo 1 y 2”.

83 Personas	<i>Grupo 1 y 2</i>				
<i>Clasificación</i>	<i>Pregunta 1</i>	<i>Pregunta 2</i>	<i>Pregunta 3</i>	<i>Pregunta 4</i>	<i>Pregunta 5</i>
5	54	40	57	38	72
4	19	23	23	18	8
3	4	10	1	8	2
2	4	5	0	11	1
1	2	5	2	8	0
Total	83	83	83	83	83

Tabla 13 *Tabulación Grupo 1 y 2.*

1. ¿Cree que el Bass Master es una herramienta dirigida a cualquier tipo de persona que desee interactuar de forma básica con el bajo eléctrico?

Después de Obtener la respuesta del Grupo 1 y 2 (Muestra). Que en total eran 83 personas, podemos observar en la *Figura 100* que nos arroja datos de la siguiente manera: 65% cree que es una excelente herramienta dirigida a cualquier tipo de persona que desee interactuar de forma básica con el bajo eléctrico, el 23% cree que es buena para esta función, 5% cree que es aceptable, 5% cree que es regular y el 2% cree que es mala para esa función. En conclusión, se puede afirmar que de acuerdo a la pregunta numero 1 para el Grupo 1 y 2 (Muestra), se cree que es una herramienta dirigida a cualquier tipo de persona.

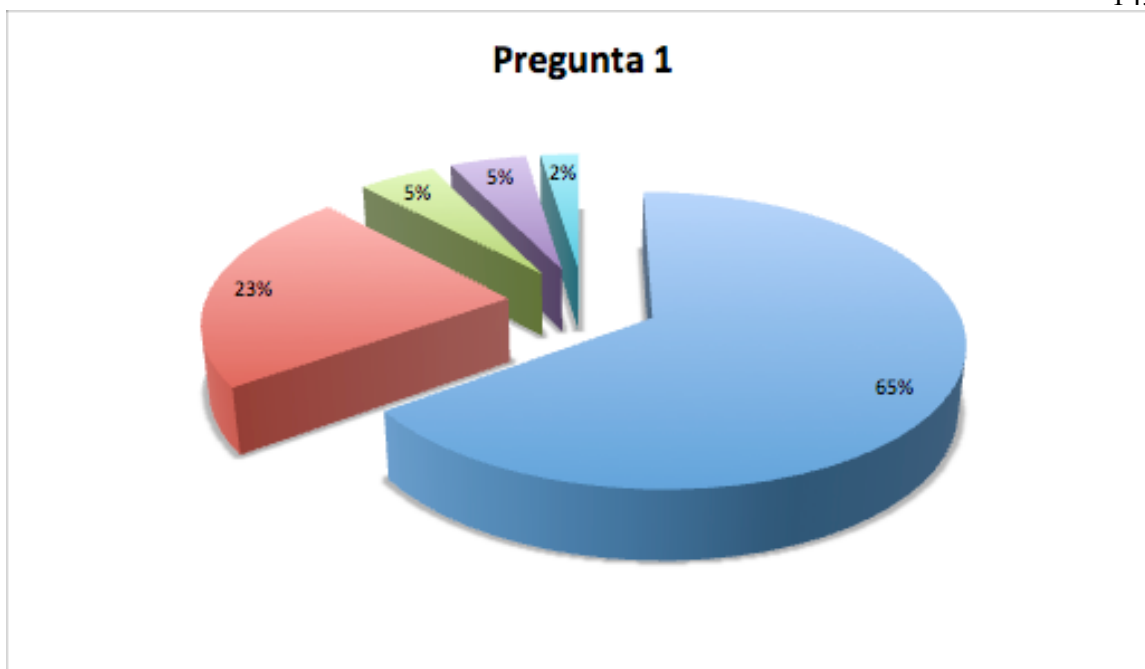


Figura 100. Gráfica pregunta número 1.

2. ¿Cree que el Bass Master es una herramienta útil para facilitar la comprensión, en cuanto a la ejecución del instrumento?

Después de Obtener la respuesta del Grupo 1 y 2 (Muestra), podemos observar en la *Figura 101* que nos arroja datos de la siguiente manera: El 50% cree que Bass master es una excelente herramienta útil, que facilita la comprensión en la ejecución del bajo eléctrico, el 29% cree que es buena para esa función, 12% cree que es aceptable, 6% cree que es regular y el 3% cree que es mala para esa función. La conclusión en la pregunta 2 para el Grupo 1 y 2 (Muestra). Es que se cree que es una herramienta útil para facilitar la comprensión, en cuanto a la ejecución del instrumento.

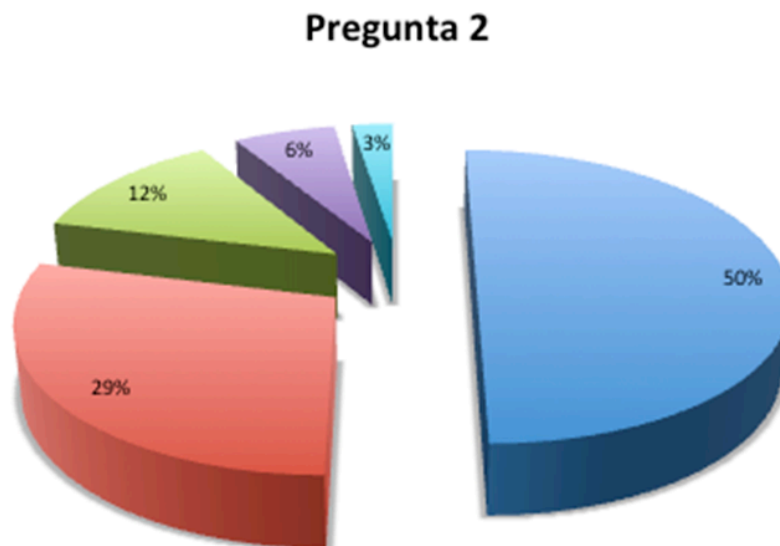


Figura 101. Gráficas pregunta número 2.

3. ¿Cree que Bass Master cumple con el objetivo de obtención e identificación de nota dentro de un instrumento temperado de cuerda pulsada, en este caso un Bajo Eléctrico?

Después de Obtener la respuesta del Grupo 1 y 2 (Muestra) en la pregunta número 3, podemos observar en la *Figura 102* que nos arroja datos de la siguiente manera: 69% cree que Bass master cumple con el objetivo de obtención e identificación de nota dentro del bajo eléctrico, el 28% cree que es buena cumpliendo el objetivo, 1% cree que es aceptable, 0% cree que es regular y tan solo el 2% cree que es mala en su objetivo. Después de observar esta grafica se puede concluir en la pregunta 3 que para la muestra seleccionada, cumple a cabalidad, la obtención e identificación de nota en el instrumento temperado de cuerda pulsada.

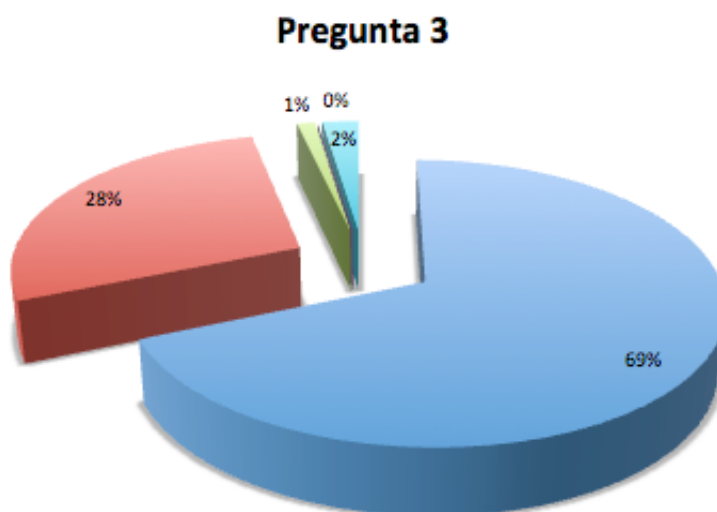


Figura 102. Gráfica pregunta número 3.

4. ¿ Es una limitante que el Bass Master no procese técnicas específicas como palm mute, slap, tapping y acordes?

Después de Obtener la respuesta de la muestra total en la pregunta 4 y dándoles a entender que en este caso: 5 era la peor calificación y 1 la menos mala, podemos observar en la *Figura 103* que nos arroja datos de la siguiente manera: 46% de la muestra total cree que Bass master tiene una limitación grande al no procesar técnicas específicas en la ejecución del bajo, el 22% cree que tiene una limitación mediana, 9% cree que es aceptable esa limitante, 13% cree que no es tan importante y el 10% cree que no es importante. En esta gráfica se pueden concluir en la pregunta 4, que gran parte de la muestra total cree que Bass Master tiene una limitante grande e importante al no procesar técnicas específicas del bajo eléctrico, dándonos a entender que se podría mejorar este proyecto en un futuro.

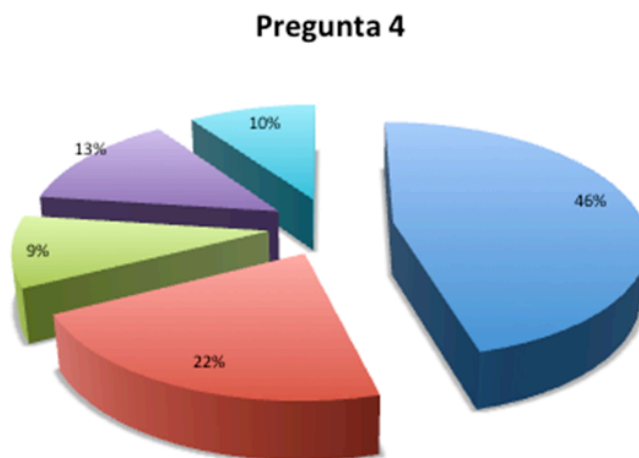


Figura 103. Gráfica pregunta número 4.

5. ¿El Bass Master cumple con la función de instruir, para posteriormente retroalimentar al usuario?

Después de Obtener la respuesta del Grupo 1 y 2 (Muestra), podemos observar en la *Figura 104* que nos arroja datos de la siguiente manera: 87% cree que Bass master cumple excelentemente la función de instruir y después tomar la lección, el 10% cree que es buena haciendo esa función, 2% cree que es aceptable, 1% cree que es regular y el 0% cree que es mala haciendo esa función. La conclusión en la pregunta 5 para el Grupo 1 y 2 (muestra), es que se cree que es una excelente herramienta cumpliendo la función de instruir y posteriormente tomar la lección al usuario.

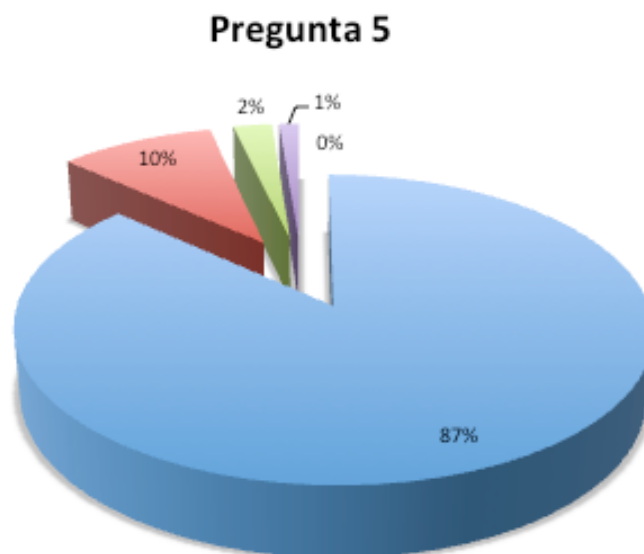


Figura 104. Gráfica pregunta número 5.

12.2 Medición objetiva del sistema

Para la medición objetiva es necesario tener en cuenta los puntos mencionados a continuación y algunos de los procedimientos realizados en el desarrollo ingenieril.

Señal del instrumento de cuerda pulsada.

La Figura 34 muestra mediante un osciloscopio, la señal que envía el instrumento (Bajo eléctrico), la cual es de aproximadamente 193 mVpp(Nota 1=Fa en cuerda 1=Mi), se realizaron pruebas con 8 notas diferentes, tocándolas 3 veces cada una, las primeras 4 notas son de la cuerda Mi, la más grave y las otras 4 notas de la cuerda más aguda, la cuerda Sol y el voltaje de salida variaba dando como promedio 166 mVpp (*Tabla 14*), esto debido a que la amplitud de la señal depende de la interpretación del usuario. Tomando el resultado obtenido por el diseño del pre amplificador realizado (*Numeral*

11.1.4.1) e implementado en la simulación llevada a cabo en el software NI Multisim (Numeral 11.1.4.2), en el cual la señal de entrada ideal era de 166 mVpp, es necesario tomar ciertas medidas, entre ellas, como se mencionó anteriormente, que la resistencia de carga sea menor a la que se encuentra en el diseño (en el diseño 12 kohms, en el real 10 kohms), esto ayudara a que la amplitud de la señal de salida sea reducida para evitar distorsión.

	Intentos cuerda 1			Intentos cuerda 2			
	1(mVpp)	2(mVpp)	3(mVpp)	1(mVpp)	2(mVpp)	3(mVpp)	
prueba nota 1	182	165	193	prueba nota 1	196	150	184
prueba nota 2	168	148	139	prueba nota 2	151	159	144
prueba nota 3	171	162	180	prueba nota 3	187	170	164
prueba nota 4	160	164	166	prueba nota 4	159	156	173

Tabla 14. Pruebas valor de voltaje de señal del instrumento.

Se realizó la operación para sacar el porcentaje de error (Ecuación 30) en cada una de las notas ejecutadas en esta etapa del sistema.

$$\frac{|\text{valor aproximado} - \text{valor exacto}|}{|\text{valor exacto}|} \times 100\%$$

Ecuación 30 Porcentaje de error de la señal de entrada.

El valor aproximado es de 2.5Vp, que tendría que ser el valor ideal de señal de entrada al Arduino. El valor exacto para la nota 1 es el promedio de las 3 pruebas que es de 2.71 Vp, el cual fue el valor de señal de salida del pre amplificador, realizando la operación de la Ecuación 30 dio como resultado un error de 7.4% en la Tabla 15 se muestra el porcentaje de error para las 8 notas tomadas en consideración.

Promedio de pruebas de notas(mVpp)	Valor de salida del pre amp (Vp)	Valor aproximado(Vp)	% de error
180	2.7	2.5	7.4
151.6	2.27	2.5	10
171.3	2.56	2.5	2.3
163.3	2.44	2.5	2.4
176.6	2.65	2.5	5.6
151.3	2.26	2.5	10.6
173.6	2.6	2.5	3.8
162.6	2.44	2.5	2.4

Tabla 15 Porcentaje de error en amplitud de cada nota.

Sensores de fuerza resistivos.

En el *Modulo 11.1.3.3* se muestra la medición que se hizo para verificar el funcionamiento de los sensores, con el fin de evaluar si el sensor tenía respuesta con la presión superficial ejercida por un dedo humano, se realizó un pequeño sketch en Arduino mostrado en la *Figura 42*, donde se pudo concluir que el sensor tiene buena sensibilidad, es decir, una buena respuesta en cuanto a pequeños cambios de presión superficial, además se midió la resistividad del sensor (*Figura 40 y 41*), dando un rango adecuado de 20 MOhms para recibir datos por la entrada análoga del Arduino. Se realizaron 30 pruebas de recepción de datos por la entrada análoga, para estar seguros de que el programa pudiera seguir un curso normal sin mayor porcentaje de error. Todas las pruebas realizadas tuvieron éxito.

El arreglo de sensores se conectó a las entradas análogas del Arduino Nano, la información se leyó y se mapeó, se estableció un umbral que al ser superado enviaba un estado HIGH (On) y de este modo se envió la señal digital al Arduino mega. Para comprobar su funcionamiento se conectaron 3 leds con resistencias de 3 KOhms a las

salidas digitales, cada led representaba un sensor y al presionar cada sensor se prendía el respectivo LED asignado, Este nuevo proceso se realizó 10 veces por cada sensor, dando como resultado 100% de efectividad al igual que en la prueba anterior.

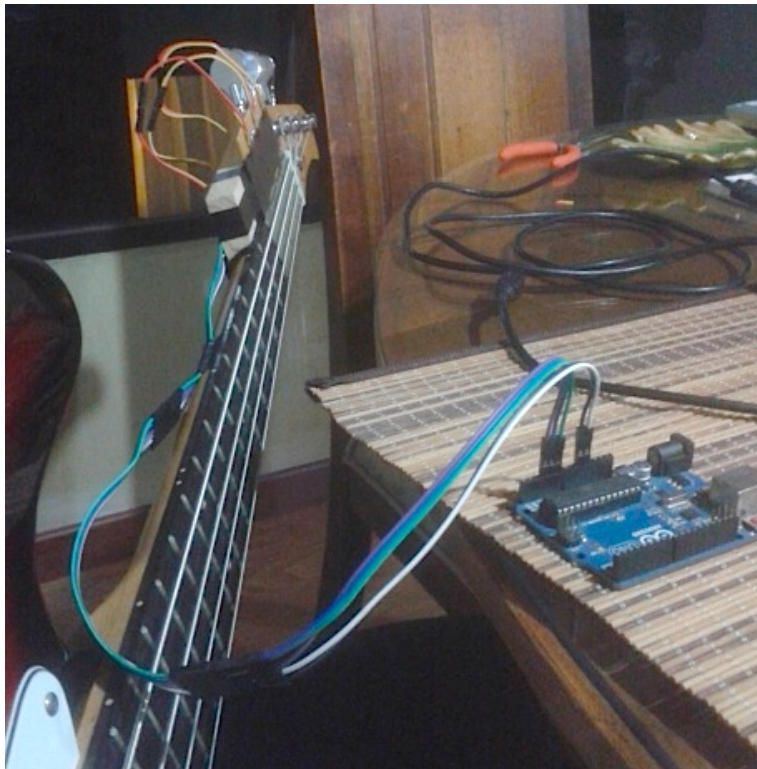


Figura 105. Prueba de arreglo de sensores.

Preamplificador.

Los valores obtenidos en la simulación fueron los esperados según el diseño, con una señal de entrada de 197.2 mVpp y con una señal de salida de 4.72Vpp (*Figura 48*), con resistencia de carga de 10 kOhms, si la resistencia de carga fuera la del diseño $R_L=12\text{KOhms}$ se tendrían 5.7 Vpp, es decir, casi la ganancia deseada $A=30$ y con el elevador dc funcionando con su origen en 2.9V. se procedió a sacar el porcentaje de error. Tomando 5.7Vpp como valor aproximado y 5.9 Vpp como valor exacto por lo que el error es de 3.3%.

La medición del pre amplificador diseñado se realizó por etapas, se toma como nivel de señal de entrada 83 mVp de una señal seno (*Figura 50*), luego se mide la señal en la salida de la etapa 1(JFET), justo después del capacitor 2 (*Figura 44*), teniendo en cuenta la ganancia de esta etapa, que es igual a 3, se evalúa la señal de salida. La señal medida fue de 260 mVp respecto a los 240 mVp que deberían ser, se obtiene un porcentaje de error de 7.6%. Se procede a medir la señal de salida de la segunda etapa (Bipolar BJT), el valor pico de salida de esta etapa medido en el capacitor 3(*Figura 45*) dio un voltaje de 2.4Vp teniendo la resistencia de carga $R_L=12\text{KOhms}$ y con la $R_L=10\text{KOhms}$ se obtienen 2Vp (*Figura 52*). Realizando el porcentaje de error según lo diseñado, se tienen 2.49Vp como valor ideal y 2.4 como valor medido. El porcentaje de error es de 3.7%. Con la resistencia de carga del diseño, se midió la señal de salida del elevador de nivel dc (*Figura 46*) entre las resistencias R10 y R11, el resultado fue 5.3Vp, pero era necesario realizar una resta ya que el pico mínimo llegaba hasta 103mVp, de este modo se obtuvo 5.19Vp como valor medido y 4.98Vp como valor ideal, dando un 4% de error.

Voltaje pico a pico	% de error
Vpp de salida en simulación	3.3
Vpp medido en salida de etapa 1	7.6
Vpp medido en salida de etapa 2	3.7
Vpp medido en salida de elevador DC	4

Tabla 16 Porcentaje de error en amplitud de Vpp Out de cada etapa.

Placa con micro-controlador.

Para comprobar el correcto funcionamiento de esta parte del proyecto se realizó un sketch en Arduino en el cual se comprueba la entrada de audio al Arduino, se conecta una señal seno de 179Hz del generador a la entrada A0 de Arduino y así se reconoce la frecuencia fundamental (*Figura 106*).

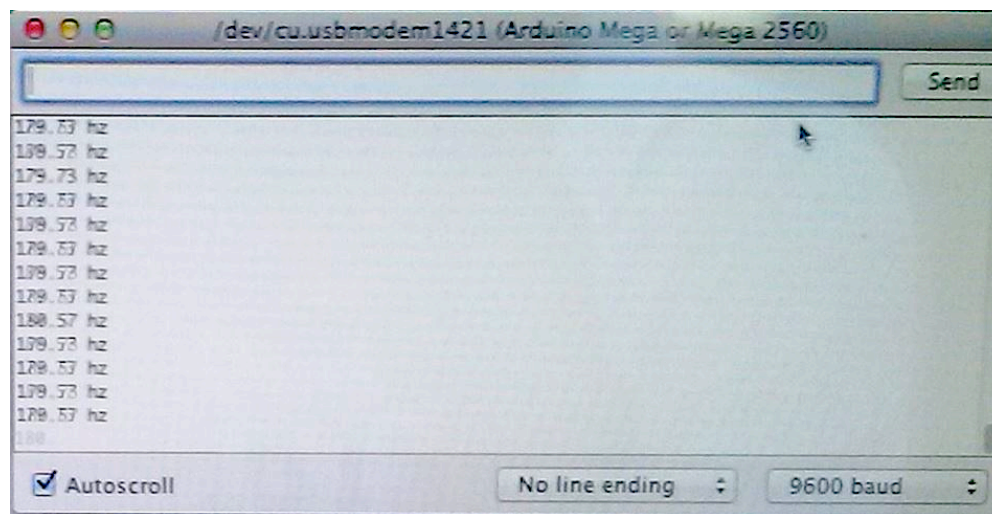


Figura 106 Puerto serial Arduino. Reconocimiento de frecuencia fundamental.

Protocolo de comunicación.

Del mismo modo para la conexión del dispositivo móvil a Arduino se hicieron dos programas uno en un Sketch de Arduino (*Figura 84*) y otro en App inventor (*Figura 83*). El programa de prueba que se hizo en App Inventor (*Figura 83*) se diseñó con el fin de probar si exista una verdadera conexión entre el dispositivo móvil y el módulo Hc 06 de Arduino. Básicamente contaba con 2 botones, el primero tenía como etiqueta 'ON' y el segundo tenía como etiqueta 'OFF', si el primer botón se presionaba, el programa desarrollado en Arduino recibía el dato 1 por puerto serial mediante el módulo Hc 06 y de este modo se encendía un led. Si el segundo botón se presionaba, el programa desarrollado en Arduino recibía el dato 2 por puerto serial mediante el módulo Hc 06 y de este modo se apagaba el led mencionado.

Se realizaron 40 pruebas, divididas en 4 partes, la primera tomando el dispositivo móvil justo en frente del módulo Hc 06, la segunda con 1 metro de distancia entre los dispositivos, la tercera prueba a 2 metros y la cuarta prueba a 3 metros. La *Tabla 17* muestra las 40 mediciones, en las cuales se indica con una R si recibió el dato y con una N si no se recibió el dato. El porcentaje de error fue de 8.1% ya que de las 40 pruebas solo en 3 fue imposible recibir el dato.

Prueba 1	Prueba 2	Prueba 3	Prueba 4
R	R	R	R
R	R	R	R
R	R	N	N
R	R	R	R
R	R	R	R
R	R	R	R
R	R	R	N
R	R	R	R
R	R	R	R
R	R	R	R

Tabla 17. Comunicación entre Modulo Hc 06 y dispositivo móvil.

Entorno de programación.

La programación en App inventor se desarrolló sin ningún problema generando el archivo en formato APK y también la sesión en formato .aia expuestos en el anexo B. Para entregar el reporte de resultados de interpretación al usuario, fue necesario hacer unas pruebas para analizar el porcentaje de error, a la hora de realizar la transmisión de la matriz de datos () desde el módulo Hc 06, comunicado con Arduino, hasta el dispositivo móvil. Fueron un total de 40 pruebas mostradas en la tabla a continuación. Donde el resultado arrojado fue un porcentaje de error igual a 5.15%.

Prueba de Score Bass Master							
# Prueba	Resultado	# Prueba	Resultado	# Prueba	Resultado	# Prueba	Resultado
1	16	11	16	21	16	31	15
2	16	12	14	22	16	32	15
3	15	13	16	23	16	33	16
4	15	14	14	24	16	34	16
5	14	15	15	25	14	35	15
6	14	16	14	26	16	36	14
7	15	17	16	27	16	37	15
8	16	18	16	28	15	38	16
9	14	19	14	29	15	39	16
10	16	20	13	30	14	40	16
Total Resultado							607
Promedio							15,175
Promedio deseado							16
% Error							5,15625

Tabla 18. Porcentaje de error en la transmisión de datos de Arduino a la Aplicación.

13. CONCLUSIONES

Antes de empezar a nombrar las conclusiones, es necesario dejar en claro que el proyecto se desarrolló con el fin de crear un prototipo y fue necesario delimitar algunos parámetros que se nombrarán a continuación: El número de trastes que pueden ser identificados en el bajo eléctrico por el sistema son los primeros 4 y las técnicas reconocidas por el sistema se limitarán a la técnica de Fyngerstyle y pulsación de traste. Dejando claro lo anterior, buscando contestar nuestra pregunta problema y validando nuestra hipótesis, se concluye lo siguiente:

- Por medio de la inserción de 3 sensores FSR cuadrados en los primeros 3 trastes y abarcando las cuatro cuerdas del instrumento, se logró reducir el porcentaje de error a la hora del reconocimiento de nota y posición de los dedos, de modo que se tuvo que crear una segunda condición en el algoritmo, para así reconocer la pulsación de traste y también un complemento en la programación de un aspecto que no se tenía en cuenta; el reconocimiento de octava de la nota ejecutada, parámetro que causaba algunos problemas en la identificación de la nota en términos de traste y cuerda.
- Se logró de forma exitosa el desarrollo de un sistema concreto conformado por un conjunto de subsistemas acoplados entre si, estructurado con sistemas abstractos (Aplicación, Programación de micro controlador y algoritmos) y sistemas físicos (electrónica, sensórica y herramientas de uso cotidiano(Dispositivo móvil)).
- Respecto a la implementación de herramientas como: Arreglo de sensores, programación de un analizador de frecuencia, Arduino y la aplicación de dispositivo móvil la cual se encarga de generar un reporte de aciertos y errores al usuario, se

puede llegar a concluir que este sistema físico concreto puede sustituir a un tutor, generando una guía interactiva y una retroalimentación en tiempo real.

- En cuanto a la transmisión de datos se pudo concluir lo siguiente: el sistema tiene un blindaje importante en cuestión de envío y recepción de datos, es decir que no habrá ningún tipo de interferencia en la comunicación entre el micro controlador y el dispositivo móvil, así mismo se pudo concluir que cuando el paquete de datos es ligeramente pesado, se generarán retardos en el envío y recepción de la información.
- Precizando que ya se tenía como limitante el retardo en la transmisión de datos debido al uso del protocolo Bluetooth, fue oportuno implementar un arreglo de leds y un visualizador de 7 segmentos, los cuales ayudaron a mejorar el rendimiento del sistema en cuanto a tiempo, ya que envía una visualización de la información requerida por el usuario, con mayor velocidad a la que aparece visualizada en el dispositivo móvil.
- Se realizó una prueba subjetiva con el fin de determinar el punto de vista de la población seleccionada, la cual estaba constituida por estudiantes de música de últimos semestres (9 y 10) y estratificada por estudiantes que sabían ejecutar el bajo eléctrico y los que no. Con dicha prueba se pudo concluir que: La mayoría de la muestra seleccionada manifestó que el Bassmaster es una herramienta muy útil a la hora de instruir y retroalimentar al usuario. A su vez, la mayoría de encuestados concluyeron que la falencia del sistema era la limitación de no ser capaz de identificar otras técnicas aparte de fingerstyle y pulsación de traste.

14. RECOMENDACIONES

- Para garantizar la eficiencia del sistema es indispensable que al realizar la pulsación de traste, se genere la suficiente presión para activar el sensor FSR.
- Si la señal de audio de entrada al Arduino no tiene la correcta afinación, el sistema será incapaz de reconocer la frecuencia fundamental, por ende se generará un conflicto y el sistema será incapaz de continuar en funcionamiento.
- Si la distancia entre el dispositivo móvil y el modulo Bluetooth es superior a 3 metros, se obtendrá una señal medianamente buena, aunque se pueden sufrir problemas con lluvia y viento, como se describe en el análisis de resultados *Numeral 11.1.6.2. [98]* (medición de intensidad de la estabilidad de red Bluetooth).
- Debido al retardo que se genera en la transmisión de datos por puerto serial mediante Bluetooth, se recomienda al usuario tener en cuenta las ayudas o soportes aplicados al sistema, los cuales son: El arreglo de leds, y el visualizador de 7 segmentos, y de este modo optimizar en cuanto a tiempo, la interacción entre el usuario y el sistema.
- Debido a la sensibilidad con la que se realizó el programa, se debe tener en cuenta unas pautas a la hora de ejecutar el instrumento, estas son: Evitar hacer Bending en las cuerdas con la mano izquierda, evitar el trasteo de las cuerdas y la dinámica de ejecución debe ser “Piano”(Dinámica Suave).
- Para mejorar el tiempo de respuesta en la comunicación inalámbrica, se podrían presentar mejoras si se usa otro protocolo de comunicación como Wi-fi o Ethernet.
- Mediante la implementación de un arreglo de sensores en un guante, se tendrá en cuenta un aspecto importante en cuanto a la técnica de ejecución, para así saber el dedo usado por el usuario en la pulsación de traste .

15. BIBLIOGRAFIA

- [1] HERNÁNDEZ, N. Programación de aplicaciones para dispositivos móviles. (2006)
- [2] ZARAGOZA, J. Control automático de Bluetooth en dispositivos móviles con Android (2012).
- [3] FERNÁNDEZ, A. Gestión de mecanismos Arduino controlados inalámbricamente por dispositivos Android (2011).
- [4] LLEDÓ, E. Diseño de un sistema de control domótico basado en la plataforma Arduino. (2012).
- [5] VELASCO, R. Prototipos de prueba de acelerómetro y conexión Bluetooth para terminales Android. (2011).
- [6] CARRIÓN, L; OCHOA, D & VALVERDE, A. Análisis del funcionamiento del sensor de fuerza resistivo (FSR). (2009).
- [7] SUQUILANDIA, T & , IDROVO, M. Diseño y montaje de enlace inalámbrico para transmisión de datos utilizando tecnología Bluetooth. (2013)
- [8] SADIE, S. The new grove dictionary of music and musicians, edición 2, bass guitar. (2001)
- [9] MEDINA, F & ROBLES, J. Arduino y Android una pareja para aplicaciones de UBICIUDAD (2013).
- [10] SEN, D; ORTIZ, C & AGUILAR, G. Implementación de una aplicación Android orientada al control de un prototipo de silla de rueda. (2013).
- [11] OSORIO, D. Aplicación IOS para estudiantes de una academia de estudios. (2014).
- [12] ARTEAGA, J. Desarrollo de una aplicación móvil y una guía de turismo para la visualización y descripción de sitios turísticos el centro de la ciudad de Cartagena utilizando realidad aumentada (2014).
- [13] DIAZ, C; HOMMY, I; MONSALVE, P & BORIS, D. Desarrollo de un prototipo de casco inteligente para la prevención de lesiones y seguridad en vehículos motorizados en dos ruedas. (2014).
- [14] PÉREZ, A. Diseño de aplicación móvil para la comunicación inalámbrica de señales audiovisuales. (2013).

- [15] KLOSS, J. Android apps with App Inventor. (2012).
- [16] GOOGLE LABS. App inventor, programación de dispositivos móviles al alcance de todos.(2010).
- [17] WOLBER, D; ABELSON, H & FRIENDMAN, M. Democratizing Computing with App Inventor. (2014).
- [18] AJA, J. Aplicación móvil para el acceso a la información de la universidad de Cantabria. (2014).
- [19] GÓMEZ, A. Aplicación android para la empresa travelling-service. (2014).
- [20] Dr. MOLINA ESPINOSA, J. APP INVENTOR. (2010).
- [21] MIT CENTER FOR MOBILE LEARNING. App Inventor Beginner Tutorials.
- [22] LEGASA, X & GIL, M. Cuadricóptero arduino por control remoto android. (2012)
- [23] ABDERRAHIM, M. Quadcopter: Construcción, control de vuelo y navegación GPS. (2102).
- [24] FLORES, C & REDONDO, D. Desarrollo de un sistema embebido para el control luminotécnico con tecnología Led.(2013).
- [25] ANDRADE, A & PINZÓN, A. Implementación del sistema de domótica en el hogar. (2013)
- [26] MORALES, M. Implementación de un medidor de consumo eléctrico residencial para facturación vía internet. (2013).
- [27] CREATIVIDAD AHORA. Comunicación Arduino-Bluetooth JY-MCU.
- [28] MARÍN, E; MARÍN, C & VALDIVIESO, C. Diseño de controlador empleando sistema de Arduino para posicionamiento cabezal de impresión de tres dimensiones, ejemplo de aplicación utilizando prototipo de impresora 3D.
- [29] OPEN SOURCE ELECTRONIC GUITAR TUTOR. Guitarduino. (2010).
- [30] MARGOLIS, M. Arduino Cookbook.
- [31] TAPIA, C; MANZANO, H. Evaluación de la plataforma Arduino e implementación de un sistema de control e posición horizontal. (2013).
- [32] ENRÍQUEZ, R. Guía de usuario Arduino. (2009).

- [33] CASTRO, D & PEÑA, J. Diseño y construcción de un prototipo experimental de un vehículo electrónico auto balanceado unipersonal veau. (2011).
- [34] GONZALEZ, C. Tecnología Bluetooth.(2008).
- [35] LÓPEZ, P. Aplicaciones Bluetooth implementadas con Java.
- [36] MARTÍNEZ, S; OLIVO, M & RODRÍGUEZ, J. Desarrollo de aplicación domótica con comunicación inalámbrica Bluetooth. (2014).
- [37] MULLER, N. Bluetooth. (2009)
- [38] HP. Bluetooth wireless technology basics.
- [39] HAUGAN, A & RUDOLPH, L. Bluetooth for programmers.
- [40] ING. JOSÉ ROBERTO VIGNONI. BLUETOOTH. (2007).
- [41] << <http://www.instructables.com/id/Arduino-Frequency-Detection/>>>
- [42] MALAVÉ, P & SANGOI, A. Redes inalámbricas locales como herramientas de publicidad: Bluetooth y WIFI.(2012).
- [43] UNIVERSITY OF MADRAS; ANNA UNIVERSITY & JERUSALEM COLLEGE OF ENGINEERING. Bluetooth in wireless communication.(2009).
- [44] CAMACHO, R. Implementación de un sistema de control de fuerza para un prototipo e terminal de agarre.(2007).
- [45] AGUILAR, F & RUIZ, R. Diseño de construcción de un dedo de cuatro grados de libertad conformado por músculos neumáticos antagonistas.(2008).
- [46] DELGADO, M. Desarrollo de un prototipo de sensores táctiles de tres ejes.(2009).
- [47] PEDREÑO, J; GUERRERO,A & LÓPEZ, J. Estudios de sensores táctiles artificiales aplicados a la robótica de agarre.(2005).
- [48] ALBÁN, D & SALAZAR, F. Diseño e implementación de sensores de temperatura, presión y proximidad de un prototipo de piel sensible para robots compañeros.(2011).
- [49] MURCIA, O & FIERRO, V. Desarrollo de un sistema web para el procesamiento del nivel de presión sonora y variables climáticas en el análisis del impacto de ruido. (2014).

- [50] AES. ROSENBERG, C & MOSES, B. Future human interfaces to computer controlled sound systems.(1993).
- [51] RODRIGUEZ, O & MAYA, R. Implementación de una red inalámbrica Bluetooth.(2003).
- [52] LALAMA FERNÁNDEZ, L. Implementación de un robot móvil terrestre omnidireccional semiautónomo y telecontrolado a través de un teléfono inteligente Android.(2014).
- [53] COMUNICACIÓN SERIAL. Conceptos Generales.
- [54] MARTIN IBARRA ROMERO. Programación orientada a objetos. (2004)
- [55] MÉNDEZ ÁLVAREZ & YALITZA PASTORA. Seguridad al acceso de información n implantación de una red inalámbrica. 2010.
- [56] <<Metodologías de programación >> <http://www.unab.edu.co>.
- [57] MORÓN, J. Estudios el rendimiento de perfiles Bluetooth en redes de área personal. (2008).
- [58] YANEZ, C. FIAMETTA 2.0(Interfaz sonora de carácter digital).
- [59] ZUÑIGA, V. Redes de transmisión de datos.(2005).
- [60] FERRER, J. Sistemas inalámbricos ara comunicación con dispositivos móviles.(2012).
- [61] MORENO VELASCO, IGNACIO. Sensores resistivos y su condicionamiento.
- [62] GARCÍA, JUAN CARLOS. Implementación de sistemas de comunicación Bluetooth con java.(2013).
- [63] CANO, J; GARCÍA, T & SÁNCHEZ,P. Sistemas de acceso Bluetooth.(2009).
- [64] GRAUDIO PRODUCCIONES. <<WWW.GRAUDIO.ES>> Estructuras de la señal. (2005).
- [65] UNIVERSIDAD DE CANTABRIA. Sistemas de comunicación. (2006).
- [66] BIDONDO ALEJANDRO. Micrófonos. 2004.
- [67] ROLDAN ARANDA ANDRES. Los micrófonos en acústica. 2000 – 2001.
- [68] NICOLÁS SUÁREZ. << CABLES Y TIPOS DE CLABLES >>

- [69] ÁLVAREZ G, PICO J VICARIO, L AMUCHÁSTEGUI, AYUSO N. BENÍTEZ N. Desarrollo de un protocolo para comunicaciones inalámbricas e implementación en dispositivos de lógica configurable. Facultad de Informática. San Sebastián, España.
- [70] UNIVERSIDAD DE BUENOS AIRES (UBA). Micrófonos.
- [71] ING. ÓSCAR RAMÍREZ. Desarrollo de un algoritmo para la detección eficiente de acordes musicales.
- [72] <<<http://sensoresdeproximidad.galeon.com/index.html>. >>
- [73] <<García, E., Automatización de Procesos Industriales., pág. 142§ 2.>>
- [74] Andrés F. Pardo V. & Sebastián Lopera G. Métodos de análisis para preamplificadores de audio (2010).
- [75] << D. M. Thompson, "Understanding audio". Boston: Berklee Press. 360 p. 2005. >>
- [76] Schiavon, M. Transistores de efecto de campo.(1997).
- [77] Alejandro de La Plaza, Héctor Fabián Kirschenbaum. Elevador de tensión para circuitos cmos n-well.
- [78] <<sistemas embebidos. >>
<http://www.ib.cnea.gov.ar/~sisemb/apuntes/Modulo1.1.pdf>.
- [79] GIANNETTI, C. Estética digital: Sintopía del arte, la ciencia y la tecnología, Ed. Làngelot, Barcelona, 2002, p.119.
- [80] Fuente de internet <<<http://prof.usb.ve/meier/interf/estilos.html>>>
- [81] Fuente de internet
<<http://personales.upv.es/moimacar/master/download/interactivo_reactivo.pdf>>
- [82] << <http://buscon.rae.es/dpd/?key=interfaz&origen=REDPD> (Diccionario panhispánico de dudas ©2005Real Academia Española © Todos los derechos reservados) >>
- [83] Fuente de Internet "Interfaz De Línea De Comandos"
<<<http://funprogramacion.wikispaces.com/Consola+de+línea+de+comandos>>>.
- [84] Fuente internet
<<<http://personales.upv.es/moimacar/master/download/interfaces.pdf>>>

- [85] Ejemplo de GUI Fuente Internet. Interfaz gráfica tridimensional de Compiz en GNU/Linux
http://www.cecep.edu.co/index.php?option=com_content&view=article&id=640&Itemid=857
- [86] Omar Rocha, Nicolás Martínez. DESARROLLO DE UN CHAT PARA DISPOSITIVOS MÓVILES ANDROID BASADO EN EL PROTOCOLO DE COMUNICACIÓN BLUETOOTH. 2012.
- [87] ECURED. Lenguaje de programación. 2010.
- [88] <<REDES GUIADO>> www.fundamentosderedes.jimdo.com
- [89] ESTÁNDAR BLUETOOTH. IEEE 802.15.1.
- [90] REDES INALÁMBRICAS. Wireless Lan.
- [91] REDES LOCALES BÁSICAS, comunicación Guiada y no Guiada.
- [92] Ed Burnette. Hello, Android. The Pragmatic Programmers; Edición: 3rd revised edition. 3 de agosto de 2010.
- [93] Apple Inc. Cocoa Touch. [En línea]. Disponible en:
<<https://developer.apple.com/technologies/ios/cocoa-touch.html>>
- [94] <<ACER>>www.acer.com/iconia
- [95] redes inalámbricas. Protocolos para Comunicación de Datos en Aplicaciones de Instrumentación y Control.
- [96]<<http://cvb.ehu.es/open_course_ware/castellano/tecnicas/electro_gen/teoria/tema-4-teoria.pdf>>
- [97] <https://www.arduino.cc/en/uploads/Main/arduino-mega-schematic.pdf>
- [98] ROWE ROBERT, Machine Musicianship. 2001
- [99] Lutz Jäncke, Eckart Altenmüller, MachtMusik inteligente : nuevos hallazgos de la neurociencia y la psicología cognitiva. 2008

Anexo A

En la siguiente imagen se muestra la sesión en Pro Tools, donde se realizó la grabación de cada uno de los audios con sus respectivos nombres, de las 16 notas que se utilizarían en el proyecto.



Figura 107. Sesión en Pro Tools con los audios grabados.

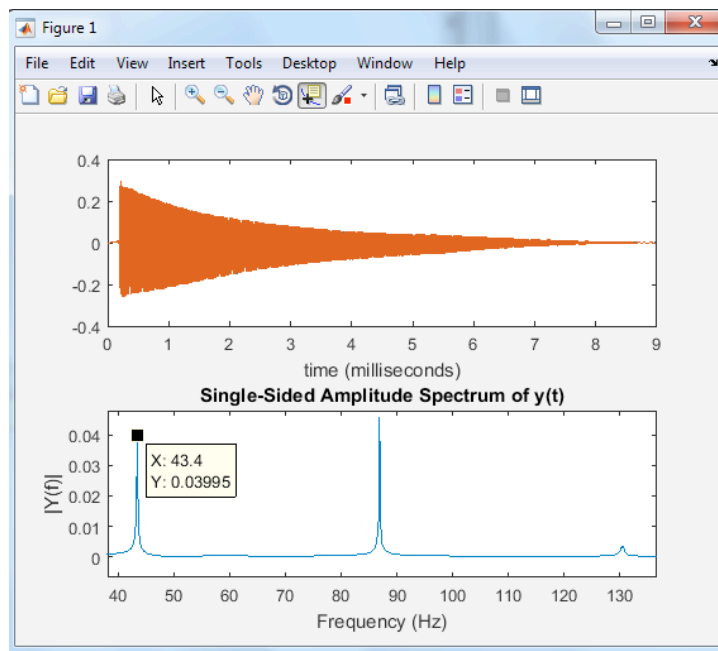


Figura 108.Fundamental F en la cuerda E.

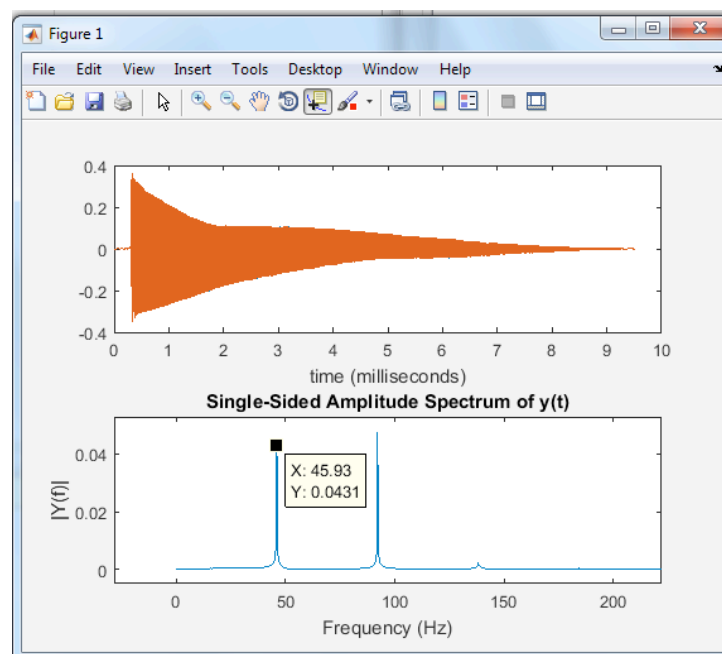


Figura 109.Fundamental F# en la cuerda E.

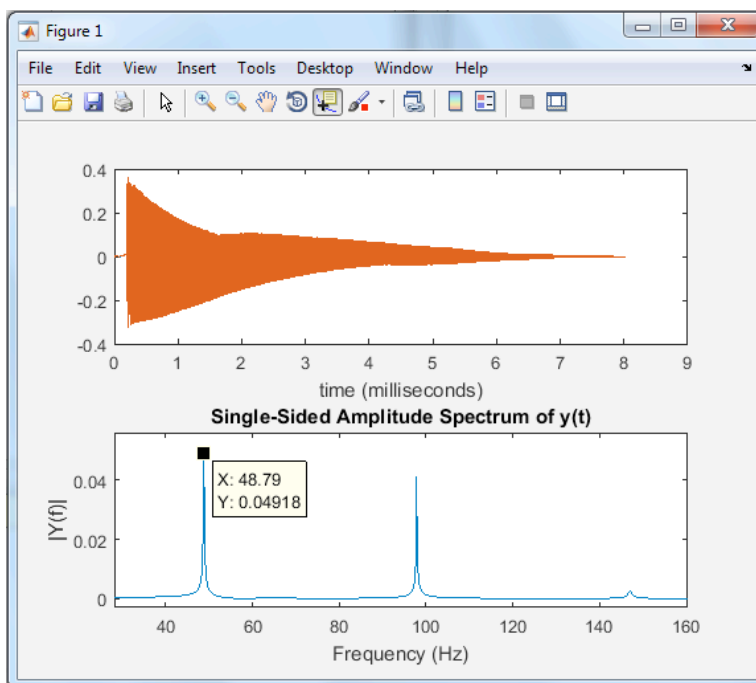


Figura 110. Fundamental G en la cuerda E.

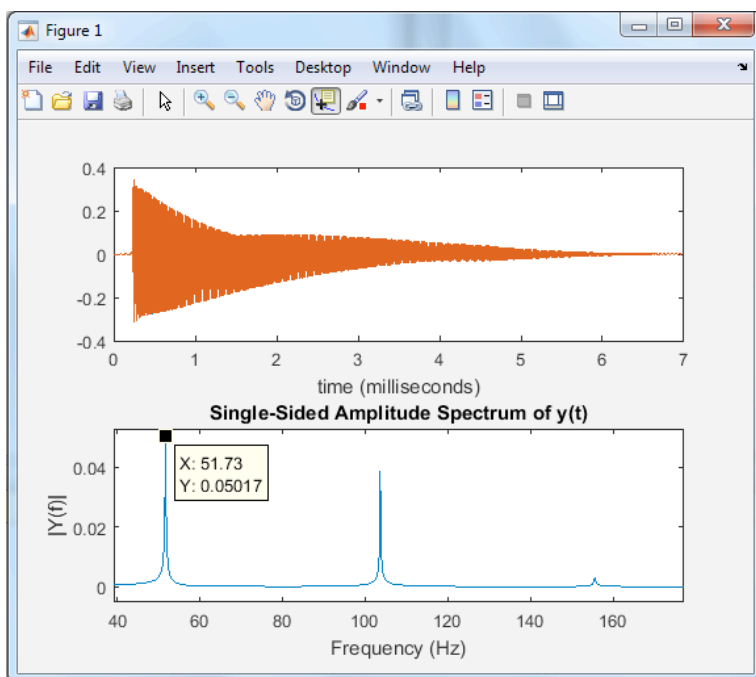


Figura 111. Fundamental G# en la cuerda E.

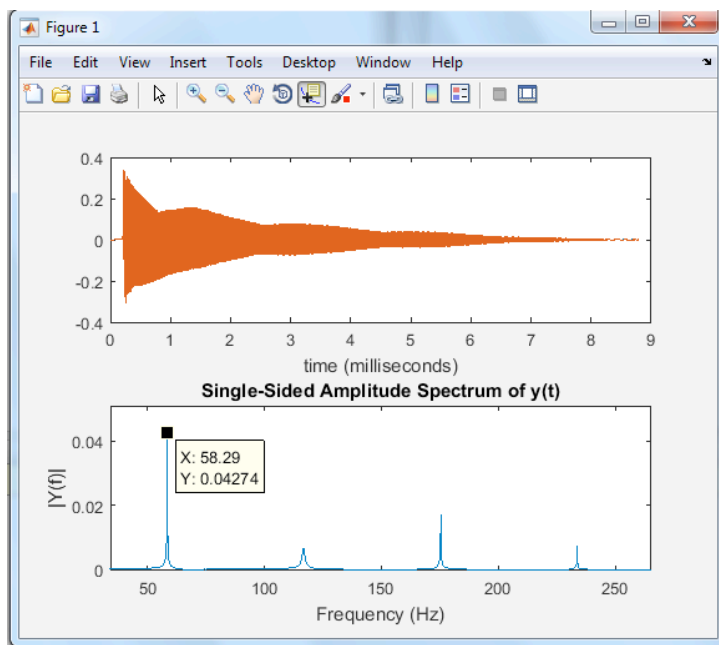


Figura 112. Fundamental A# en la cuerda A.

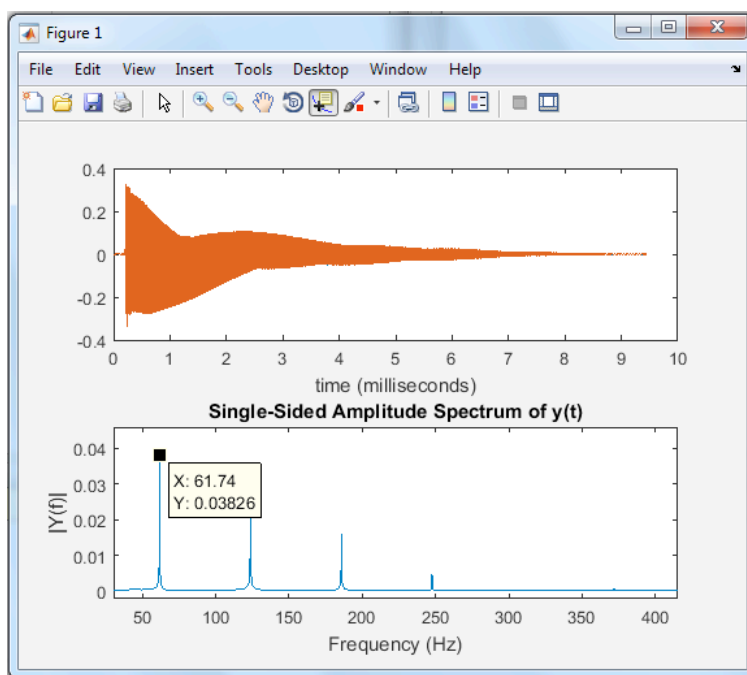


Figura 113. Fundamental B en la cuerda A.

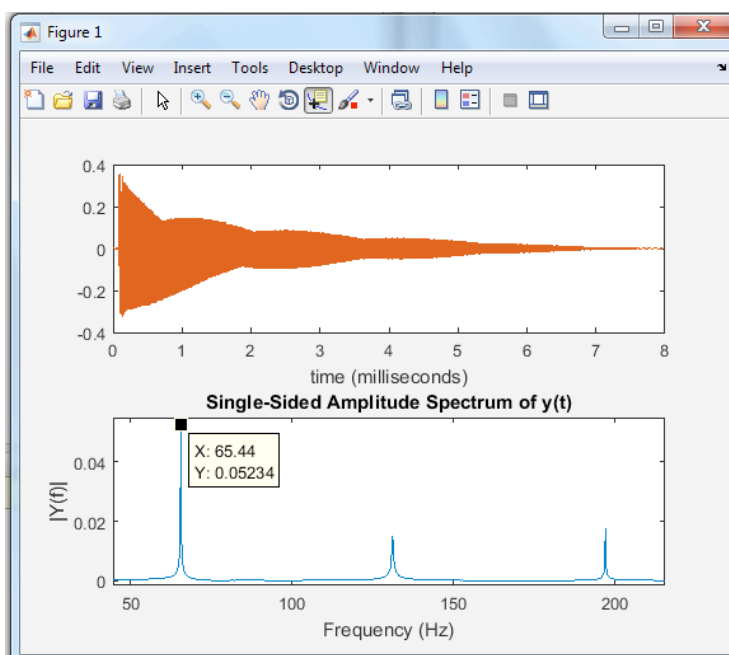


Figura 114. Fundamental C en la cuerda A.

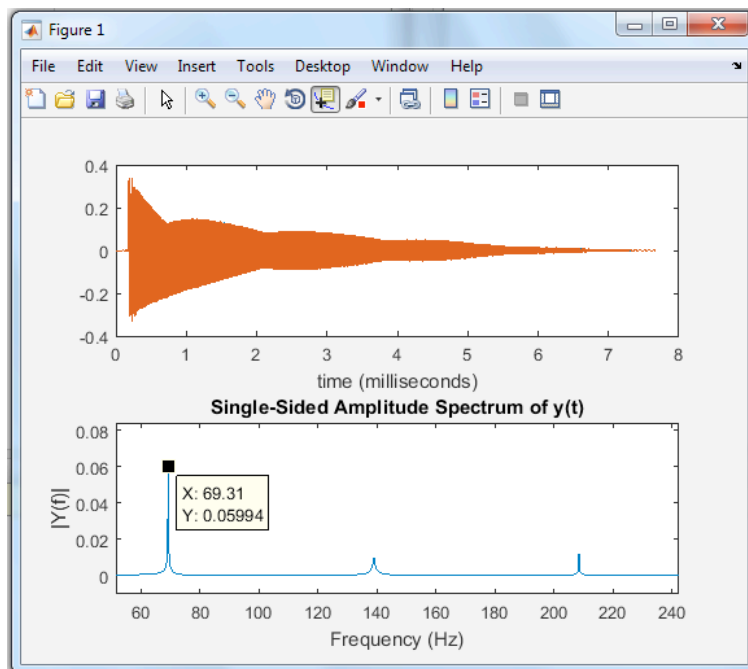


Figura 115. Fundamental C# en la cuerda A.

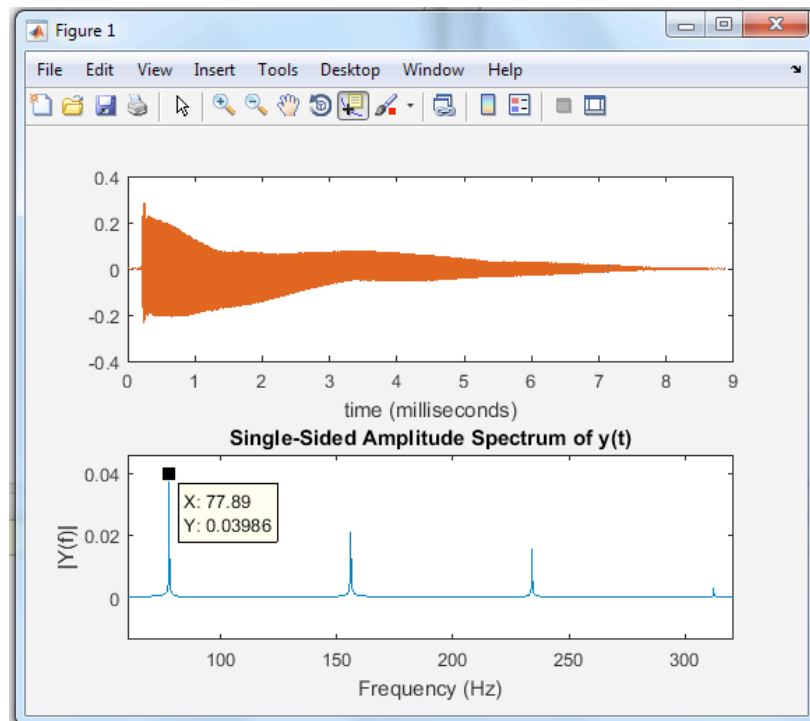


Figura 116. Fundamental D# en la cuerda D.

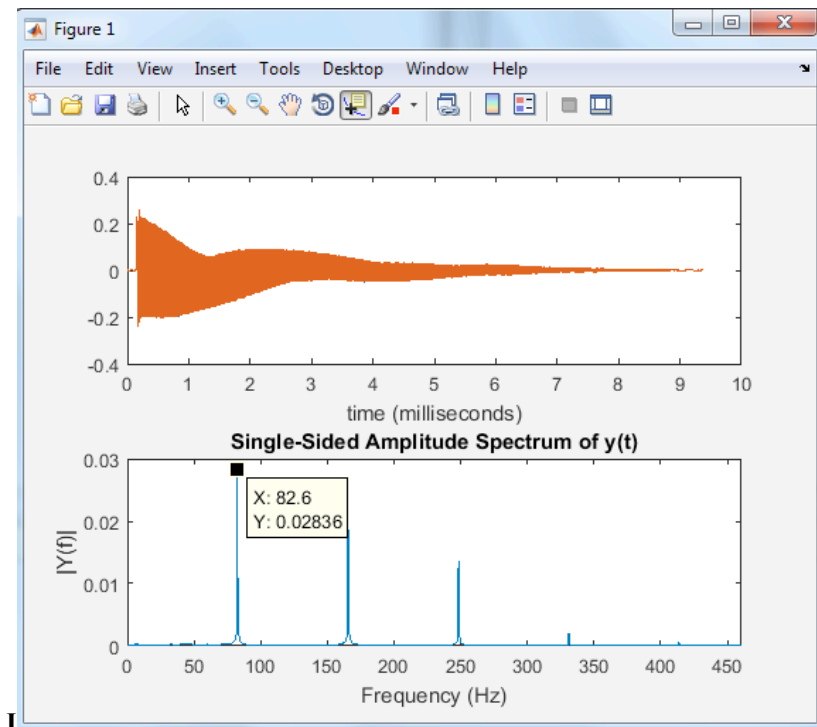


Figura 117. Fundamental E en la cuerda D.

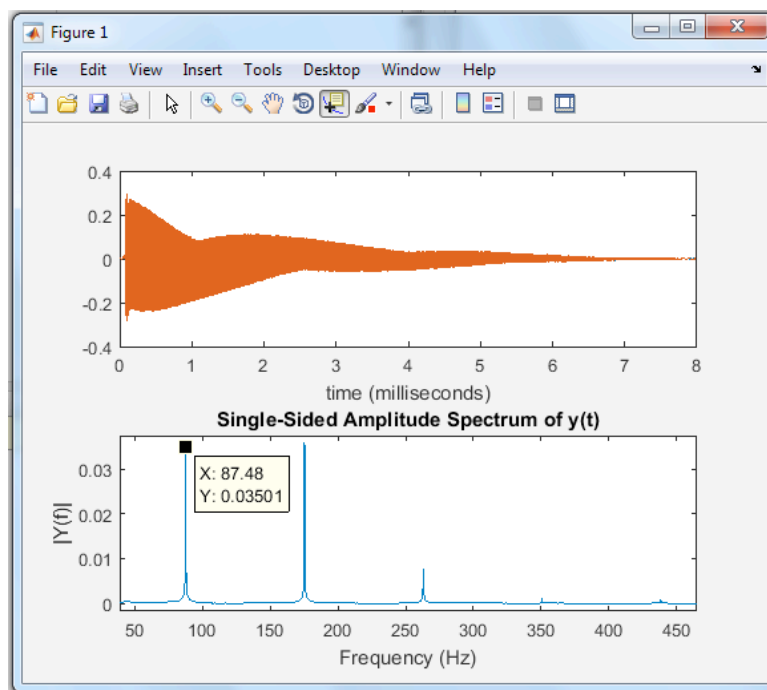


Figura 118 Fundamental F en la cuerda D.

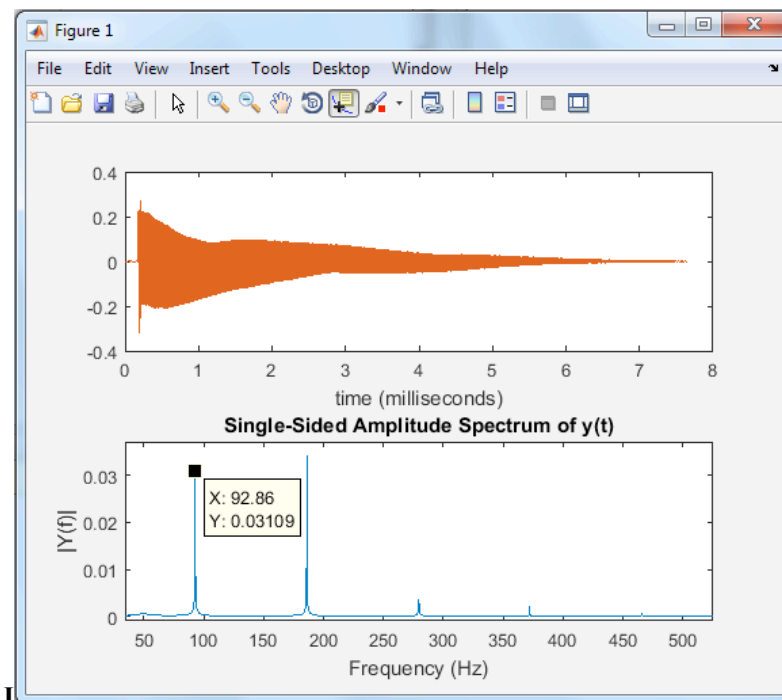


Figura 119. Fundamental F# en la cuerda D.

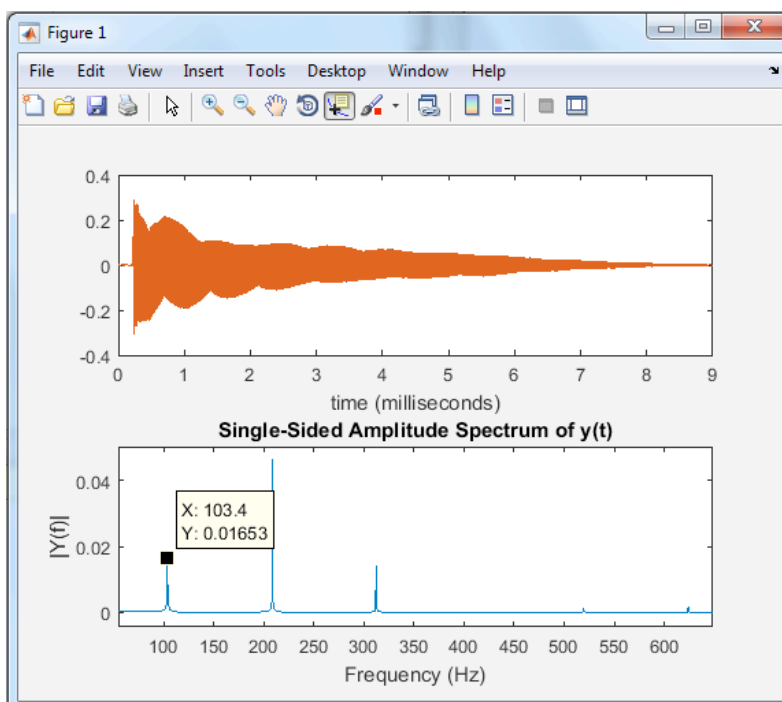


Figura 120. Fundamental G# en la cuerda G.

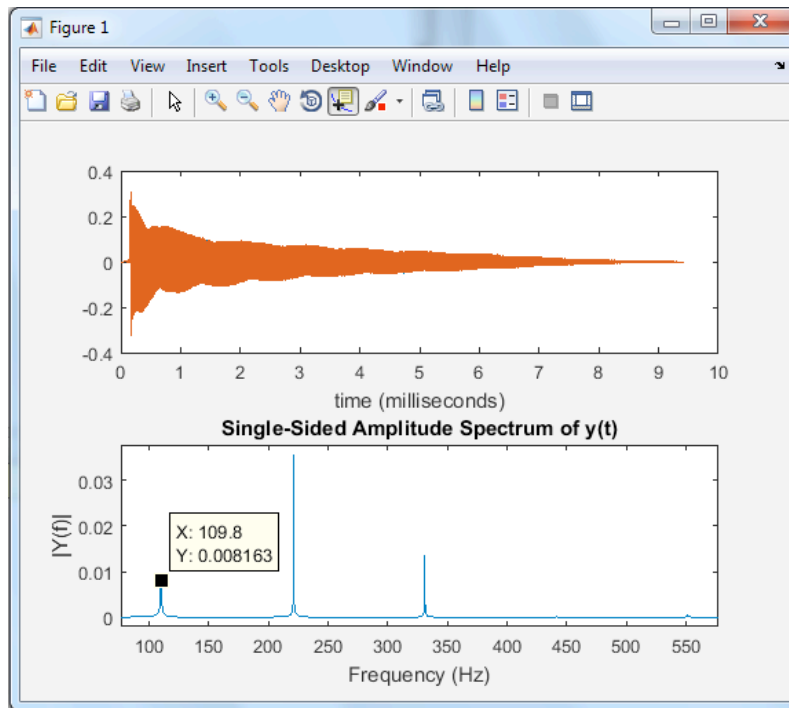


Figura 121. Fundamental A en la cuerda G.

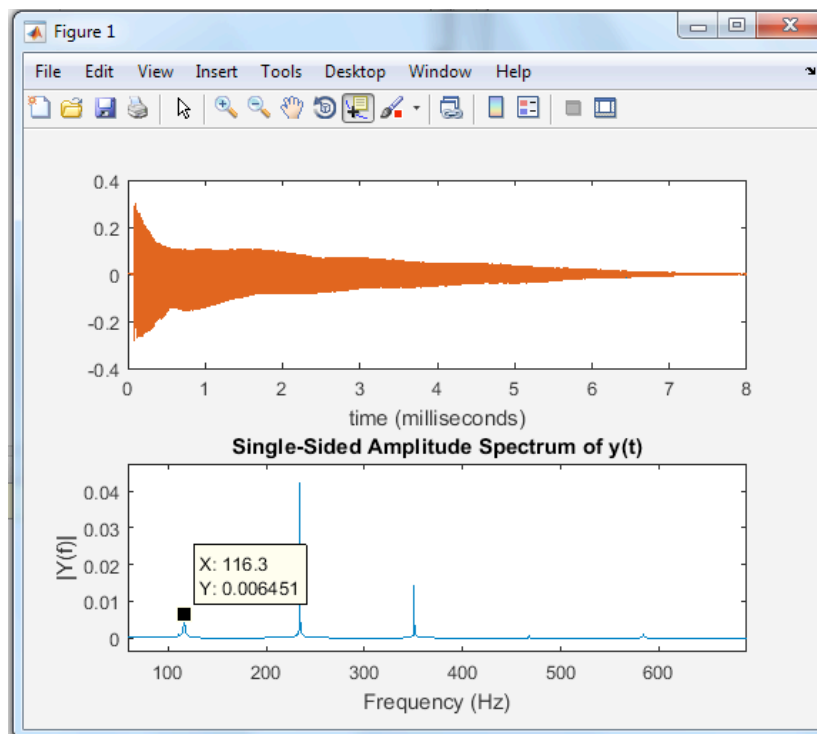


Figura 122. Fundamental A# en la cuerda G.

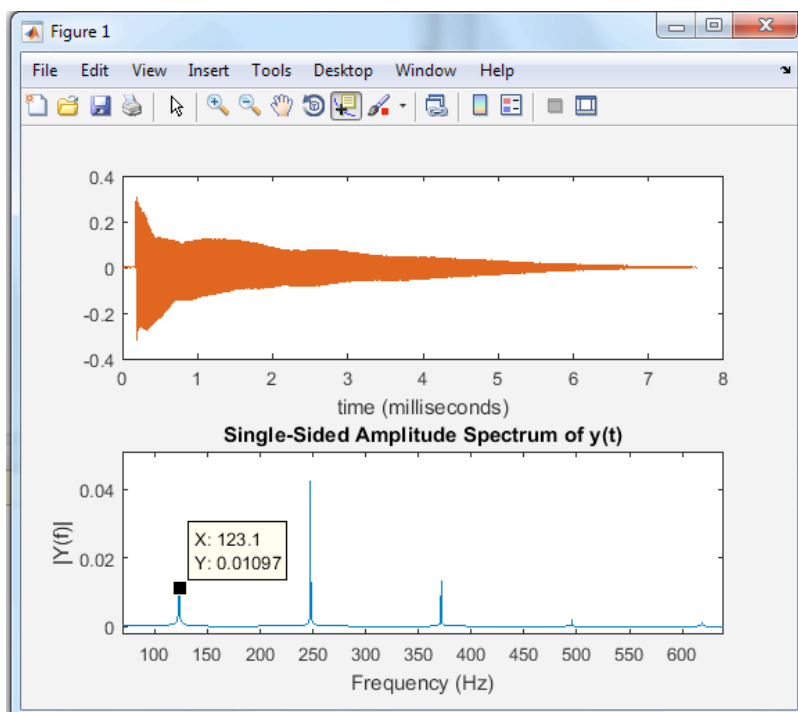


Figura 123.Fundamental B en la cuerda G.

Anexo B

Junto con este documento se hace entrega de un CD en donde se encuentran todos los programas realizados (Código y formatos) y un video del Sistema funcionando.

NOTA: Hay programas en formatos donde solo se pueden abrir por medio del software, en este caso el formato .aia se debe abrir desde App inventor, como es un desarrollador de aplicaciones Online se debe ingresar a “<http://ai2.appinventor.mit.edu>”. De igual forma, para poder abrir el formato .apk la aplicación debe ser instalada en un dispositivo móvil.

Anexo C

En este anexo se encuentra, la encuesta realizada en el presente proyecto

Encuesta para proyecto de Grado.
Ingeniería de sonido.

Nombre _____

Ejecuta el Bajo eléctrico? Si ___ / No ___

Después de utilizar el Bass master califique la siguiente encuesta de 1 a 5 dando 1 la calificación mas baja y 5 la mas alta.

1. ¿ Cree que el Bass Master es una herramienta dirigida a cualquier tipo de persona que desee interactuar de forma básica con el bajo eléctrico ?

1	2	3	4	5

2. ¿ Cree que el Bass Master es una herramienta útil para facilitar la comprensión, en cuanto a la ejecución del instrumento ?

1	2	3	4	5

3. ¿ Cree que Bass Master cumple con el objetivo de obtención e identificación de nota dentro de un instrumento temperado de cuerda pulsada, en este caso un Bajo Eléctrico?

1	2	3	4	5

4. ¿ Es una limitante que el Bass Master no procese técnicas específicas como palm mute, slap, tapping y acordes ? donde 5 es una limitante importante y 1 no es una limitante importante.

1	2	3	4	5

5. ¿ El Bass Master cumple con la función de instruir, para posteriormente retroalimentar al usuario?

1	2	3	4	5