UNIVERSIDAD DE LAS AMÉRICAS
Laureate International Universities®

FACULTAD DE INGENIERIA Y CIENCIAS AGROPECUARIAS

SOFTWARE DE ESTIMACIÓN DE PÉRDIDA AUDITIVA

Trabajo de Titulación presentado en conformidad a los requisitos establecidos
para optar por el título de
Ingeniero en Sonido y Acústica

Profesor Guía
Ing. Felipe Mardones D.

Autor
Franklin Mauricio Robalino

Año
2011

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

……………………….

Felipe Mardones D.
Ing. Acústico U.A.Ch.
172068234-1

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

……………………….

Franklin Robalino
171353029-1

AGRADECIMIENTO

Aprovecho para agradecer a todas las personas que me han apoyado durante mis estudios, empezando por mi familia que siempre estuvo a mi lado, mis profesores que me supieron guiar, y un agradecimiento especial al Dr. Wilson Paz y Miño que me ayudó a entender más a fondo los procesos de audiometría y me facilitó acceso al consultorio para la medición.

DEDICATORIA

Dedico este trabajo a las personas que están siempre en busca de nuevos conocimientos y respuestas, aunque estas vayan en contra de todo lo ya aprendido durante el transcurso de la historia o de qué tan ilógicas sean estas para el resto, pues como dijo Albert Einstein “El misterio es lo mejor que podemos experimentar. Es la fuente de todo arte y ciencia verdaderos”.

RESUMEN

El Software de Estimación de Pérdida Auditiva fue creado con el objetivo de desarrollar un programa utilitario y de fácil aplicación para la evaluación de pérdida auditiva. Para lograr esto se estudió el funcionamiento de las máquinas de audiometría análogas, se realizó un diagrama de flujo para el software, luego se calibró el software como se calibra cualquier otro audiómetro análogo, y se validó el programa a través de comparaciones realizadas con el audiómetro AD229 del Centro Médico Meditrópoli.

Para poder programar el código del software, se utilizó el programa *LabVIEW* de la compañía *National Instruments*, el cual es un software de lenguaje de programación con interface gráfica, lo cual facilita la programación.

Los resultados que se obtuvieron con el software de Estimación de Pérdida Auditiva fueron muy similares a los resultados del audiómetro análogo AD229 del Centro Médico Meditrópoli. Pues la diferencia fue muy mínima en algunas frecuencias, y esto fue a causa de ruidos de fondo, mas no por errores del software ni del audiómetro análogo.

ABSTRACT

The Hearing Loss Estimation Software was created with the objective of developing a useful program that will be easy to apply in the evaluation of hearing loss. In order to do this, it was necessary to study the way the analog audiometers worked, a flow chart was made for the software, then the software was calibrated as any other analog audiometer would be, and then the program was validated through comparisons with the AD229 audiometer from the Meditrópoli Medical Center.

The program LabVIEW from National Instruments was used in order to program the software code, which is a programming language that uses a graphical interface, which makes programming easier.

The results obtained with the Hearing Loss Estimation Software were very similar to the ones obtained with the analog audiometer AD229 from the Meditrópoli Medical Center. The difference was very small in some frequencies, and this was due to external noises, and not caused by software errors or errors in the analog audiometer.

ÍNDICE

Introducción 1

1. Introducción 1
    1.1 Antecedentes 1
        1.1.1 Información en relación al tema 1
        1.1.2 Lugar de aplicación 2
        1.1.3 Historia de la temática 2
    1.2 Formulación del problema 2
    1.3 Objetivos 3
        1.3.1 Objetivo general 3
        1.3.2 Objetivos específicos 3
    1.4 Metodología y estructura del trabajo 4

2. Marco teórico 5
    2.1 El sonido 5
    2.2 Ondas 6
    2.3 Nivel de presión sonora 8
    2.4 Funcionamiento del oído 13
        2.4.1 El oído 13
        2.4.2 El oído externo 14
        2.4.3 El oído medio 16
        2.4.4 El oído interno 24
    2.5 Fisiología de la audición 30
    2.6 Psicoacústica (concepto) 33
        2.6.1 Audición binaural 33
        2.6.2 Altura 34
        2.6.3 Sonoridad 35

2.6.4 Timbre 36

2.6.5 Enmascaramiento 36

2.6.6 Bandas críticas 36

2.7 Audiometría 37

2.7.1 Proceso de medición 38

2.8 Qué es *LabVIEW* 42

3. Programando en *LabVIEW* 47

3.1 La pantalla principal 47

3.2 Ventanas de panel y de diagrama 48

3.2.1 Barra de herramientas del panel frontal 49

3.2.2 Barra de herramientas del diagrama de bloques 52

3.3 Menús de atajos (*shortcuts*) 53

3.4 Paletas 60

3.5 Cargando y grabando Vis 63

3.6 Opciones de ayuda en *labVIEW* 64

3.7 Instrumentos virtuales 67

3.8 El panel frontal 70

3.9 Diagrama de bloques 74

3.10 Programación con “*Data Flow*” 80

3.11 Editar y eliminar errores de un VI 82

3.12 *While loop* 100

3.12.1 *Shift registers* 104

3.13 *Case structures* 107

3.13.1 Añadiendo y borrando casillas 110

3.13.2 Conectando *inputs* y *outputs* 112

3.14 Gráficos XY 113

4. Desarrollo práctico del problema 127
4.1 Planes 127
4.2 Metodología 129
4.3 Calibración 151
4.4 Prototipos 154
4.5 Manual de usuario 155
4.6 Comprobación de resultados 157

5. Conclusiones 161

Bibliografía 163

Anexos 165

CAPÍTULO 1

INTRODUCCIÓN

1.1. ANTECEDENTES

1.1.1. INFORMACIÓN EN RELACIÓN AL TEMA

Desde hace mucho tiempo, los fenómenos acústicos han formado parte del ambiente de los seres humanos, esto se debe a que la mayor parte de los seres vivos producen sonidos, y también responden a dichos sonidos, debido a esto es de suponerse que el hombre primitivo ya tenía alguna idea de la acústica, al refugiarse en distintos lugares y notar diferentes efectos acústicos como el eco en las cavernas, y después debido a su necesidad de comunicación con otros seres vivos de su misma especie, empezará a producir diferentes sonidos guturales (con la garganta), y darse cuenta de lo importante que es el oído para su sobrevivencia, ya que el oído les permitía distinguir ruidos de depredadores y comunicarse entre ellos para advertirse de cualquier peligro, así como también para ubicar presas para cazar y comerlas.

El oído es uno de los órganos más importantes, sensibles y complejos del ser humano, y está dividido en 3 partes: el oído externo, el oído medio y el oído interno. Si hubiesen problemas en cualquiera de estas 3 secciones, estos puediesen causar complicaciones de audición como otitis catarral, sinusitis, tinitus permanente (se desplaza el umbral de audición), por tales causas y para evitar muchos daños auditivos se debe realizar una audiometría.

Los sonidos están divididos en tres grupos de acuerdo al rango de frecuencias: Infrasonidos (0 Hz < f < 20 Hz), Sonidos audibles (20 Hz < f < 20 kHz), Ultrasonidos (f > 20 kHz). La audiometría se encarga de los sonidos audibles, y evalúa este rango para encontrar problemas y buscar soluciones.

El software de estimación de pérdida auditiva que será diseñado se basará en el funcionamiento de audiómetros, para realizar evaluaciones del funcionamiento del oído humano. Este software de estimación de pérdida auditiva funcionará igual que un audiómetro normal análogo con la diferencia que será automatizado para ejecutar el programa sin necesidad de un operador (médico) que emita las frecuencias y niveles de presión sonora, y también graficará los resultados de ambos oídos automáticamente, facilitando al usuario el proceso de evaluación auditiva.

1.1.2. LUGAR DE APLICACIÓN.-

El software de estimación de pérdida auditiva que será diseñado tiene como lugar de aplicación principalmente en el área médica y secundario en las compañías y personas que trabajan en áreas con un nivel de presión sonora alto el cual incrementa la posibilidad de un problema con el sistema auditivo, de esta manera se pueden hacer chequeos más seguidos y al detectarse problemas acudir a un otorrinolaringólogo para poder solucionarlo.

1.1.3. HISTORIA DE LA TEMÁTICA.-

El audiómetro tiene su origen en el estudio de líneas telefónicas. Fue inventado por George von Békésy (nacido en 1899), un físico Húngaro-Americano ganador del premio Nóbel en 1961. Su máquina era un instrumento operado por el paciente que fue lanzado en 1946. Békésy estudió la transmisión del sonido en una compañía telefónica húngara. Las pruebas de las líneas telefónicas eran realizadas rutinariamente y hecha a menudo con tonos puros. Békésy experimentó con esto y estudió la audición con gran detalle y así llegó a su diseño del audiómetro.

1.2. FORMULACIÓN DEL PROBLEMA

Actualmente en el Ecuador, los exámenes de audiometría se realizan con audiómetros análogos, usando marcas tales como: Interacoustics AD 229 en el del Centro Médico Meditrópoli y Maico audiometer MA40 en la Fundación Mosquera.

El problema en este caso es que personas que trabajan en el área de sonido, o en fábricas, aeropuertos, etc, están expuestos a altos niveles de presión sonora (NPS) con más frecuencia que otras personas y por ende son más susceptibles a sufrir daños en el sistema auditivo.

Para evitar daños severos se recurre a exámenes médicos cada cierto tiempo para realizar los debidos chequeos. Por supuesto a nadie le gusta estar horas en una sala de espera para realizar un simple chequeo rutinario y peor aun si en los resultados no hay ningún problema serio.

Otro problema es que en los hospitales el procedimiento médico es un poco largo debido a que el médico tiene que estar haciendo todas las mediciones manualmente y después graficar la respuesta de cada oído, esto toma tiempo y la gráfica no es tan exacta.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL.-

- Desarrollar un programa utilitario y de fácil aplicación para la evaluación de pérdida auditiva.

1.3.2. OBJETIVOS ESPECÍFICOS.-

- Estudiar el funcionamiento de la máquina de audiometría

- Realizar un diagrama de flujo para el software.

- Programar usando el software original “Lab View” y adaptar el programa para trabajar con una tarjeta de sonido predeterminada para mejores resultados.

- Validar el programa a través de comparaciones realizadas con el audiómetro normal.

1.4. METODOLOGÍA Y ESTRUCTURA DEL TRABAJO

Hipótesis.- al realizar los exámenes de audiometría con el software se obtendrán resultados más rápidos y más exactos que al realizarlos con la máquina de audiometría usada hoy en día.

Alcance:

Se desarrollará un software que permita realizar evaluación de pérdida auditiva por medio de tonos puros en intervalos de Octava.

Justificación:

La solución a este problema es crear un programa para audiometría que se pueda instalar en cualquier computadora, el cual indicará la respuesta de ambos oídos por separado (monoaural) para ver en qué frecuencias existen alteraciones.

De esta manera se podrá evitar horas de espera y realizar el chequeo el paciente mismo ya sea en casa o en cualquier lugar con una laptop.

De esta manera sólo cuando se vea que hay principios de problemas en la audición (mediante los resultados del programa), recurrir al médico para poder prevenirlos.

Para el uso en hospitales también facilita en gran medida a los médicos ahorrándoles el trabajo de hacer todo manualmente, y poder guardar los resultados en la computadora.

Metodología a utilizar:

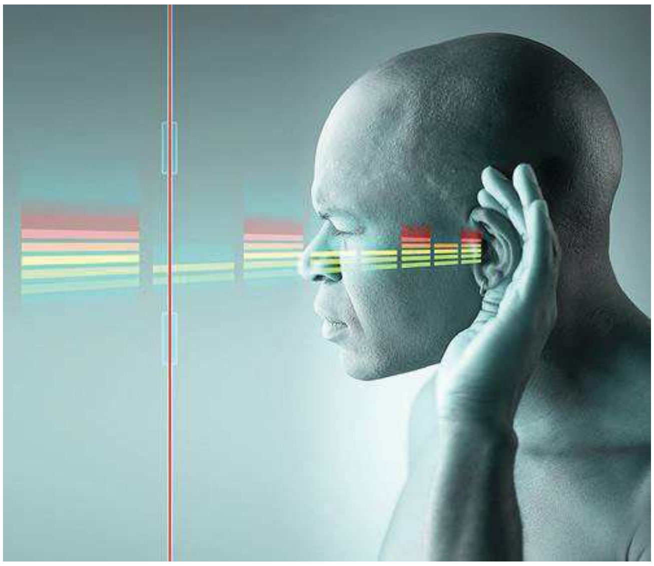
Para el diseño del software de estimación de pérdida auditiva se usará un método inductivo, observando y analizando la forma de operación de los audiómetros existentes para poder diseñar el programa y automatizarlo basándose en su funcionamiento.

CAPÍTULO 2

MARCO TEÓRICO

2.1. EL SONIDO

Figura 2.0.- El Sonido



Fuente: tnrtb.wordpress.com

El sonido es una sensación que se produce en el oído a causa de la vibración de partículas que se desplazan en forma de onda sonora a través de un medio elástico que las propaga ya sea este medio sólido, líquido o gaseoso, dichas ondas sonoras están dentro del rango audible (20-20kHz), y son capaces de ser detectadas por el oído humano. Cuando el sonido está ausente, la presión atmosférica se encontrará en un estado de equilibrio, el cual es constante en tiempo y espacio, y al aparecer una perturbación (vibración de un objeto), se producirá una variación en la presión atmosférica que se producirá a lo largo del tiempo y se propagará a través del espacio con una cierta velocidad.

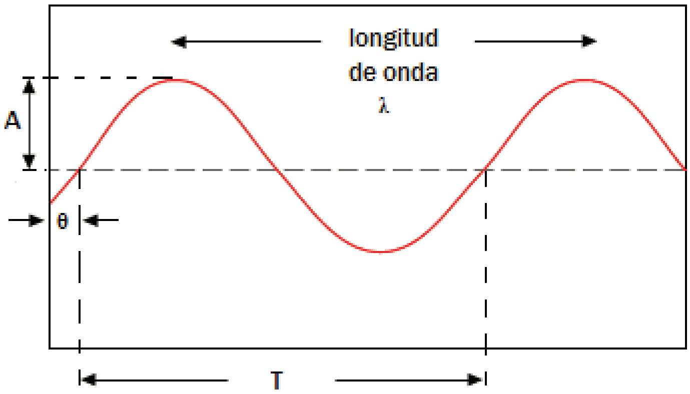
La velocidad del sonido depende del medio de propagación, su elasticidad, densidad y temperatura. Por ejemplo a una temperatura de 32 °F (0 °C), la velocidad del sonido (C) será de 331 metros (1,086 pies) por segundo.

2.2. ONDAS

Una onda sonora es generada por un objeto vibrante, cuya vibración causa movimiento en las partículas del medio elástico. Dichas partículas oscilan en la dirección de propagación de la onda.

Una onda posee las siguientes características:

Figura 2.1.- Características de una onda



Fuente: Franklin Robalino

Amplitud máxima o pico (A): es el alto de la onda desde un nivel de referencia 0 (cero), o qué tan fuerte es su nivel de presión NPS en dB (decibel).

Frecuencia (fr o f): es el número de veces que una función periódica se repite durante una unidad de variación con una variable independiente, es decir para el sonido es el número ciclos completos por segundo en [Hz] (ciclos por segundo o Hertz).

Período (T): es el tiempo en segundos (seg) que está ligado con la frecuencia y determina cuánto tiempo se toma un ciclo en completarse. Mientras más rápido la onda se mueva, el periodo será más pequeño.

Longitud de onda (λ): es la distancia en metros (m) entre dos picos consecutivos en la onda.

Fase (θ): se define como la posición angular de una onda con respecto al tiempo. Un ciclo completo se define como 360˚ de fase.

La ecuación de onda está definida como:

$$v(t) = Vmax \times sen(\omega t + \theta)$$

Donde ω = 2πf

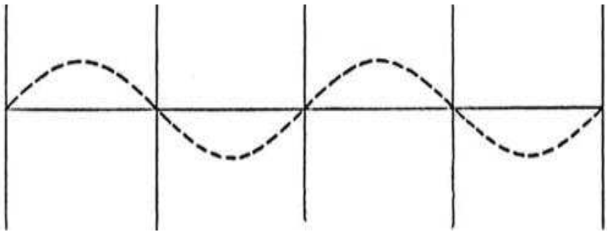
y Vmax= amplitud

La longitud de onda (λ), la frecuencia (fr o f) y la velocidad de la onda (v) están relacionadas de la siguiente manera:

$$fr = \frac{v}{\lambda}$$

Tono puro.- el tono puro es la oscilación en una sola frecuencia

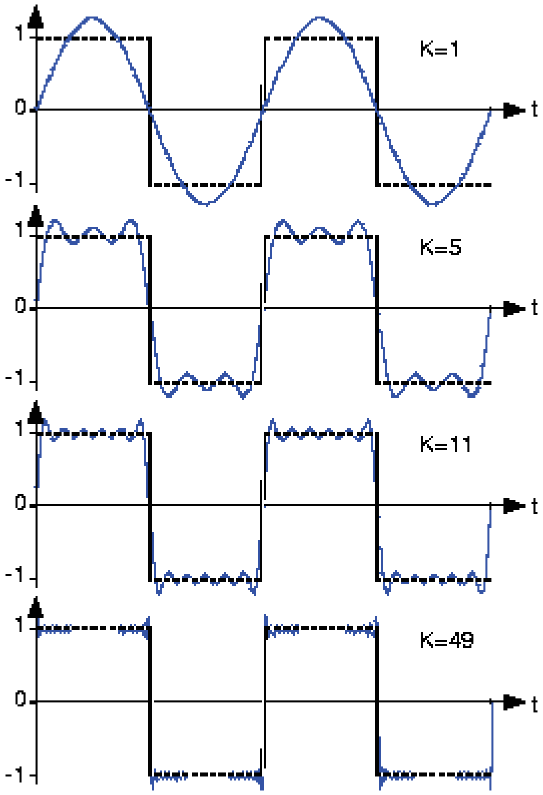
Figura 2.2.- Tono puro



Fuente: Fanklin Robalino

Tonos Complejos.- Los tonos complejos son el producto de la combinación de varias oscilaciones (tonos puros) donde como resultado obtenemos 1 frecuencia fundamental y varios armónicos o sobre tonos.

Figura 2.3.- Tono complejo



Fuente: cnx.org

2.3. NIVEL DE PRESIÓN SONORA

El nivel de presión sonora NPS es una comparación logarítmica entre la presión acústica RMS y la presión de referencia y varía entre 0 dB (umbral de audición) y 140 dB (umbral de dolor).

Para medir el nivel de presión sonora se utiliza el dB y no el Pascal, por el amplio margen que hay entre la sonoridad más intensa y la más débil (entre 20 Pa y 20 microPa).

Normalmente se adopta una escala logarítmica y se utiliza como unidad al decibelio. Como el decibelio es adimensional y relativo, para medir valores absolutos se necesita especificar a qué unidades está referida. En el caso del nivel de presión sonora, el dB SPL toma como unidad de referencia 1 microbar.

Las siglas las SLP hacen referencia al nivel de presión sonora en ingles (*Sound Pressure Level*).

Para medir el nivel de presión sonora se utiliza la fórmula:

$$NPS = 10\log\left(\frac{P1}{P0}\right)^2 = dB_{SPL}.$$

Donde: P1 es la presión sonora instantánea RMS.

P0 es la presión de referencia y se toma como referencia la presión sonora en el umbral de audición, que son 20 µPa. ($2x10^{-5}$Pa)

log es un logaritmo decimal (en base 10)

Potencia acústica.-

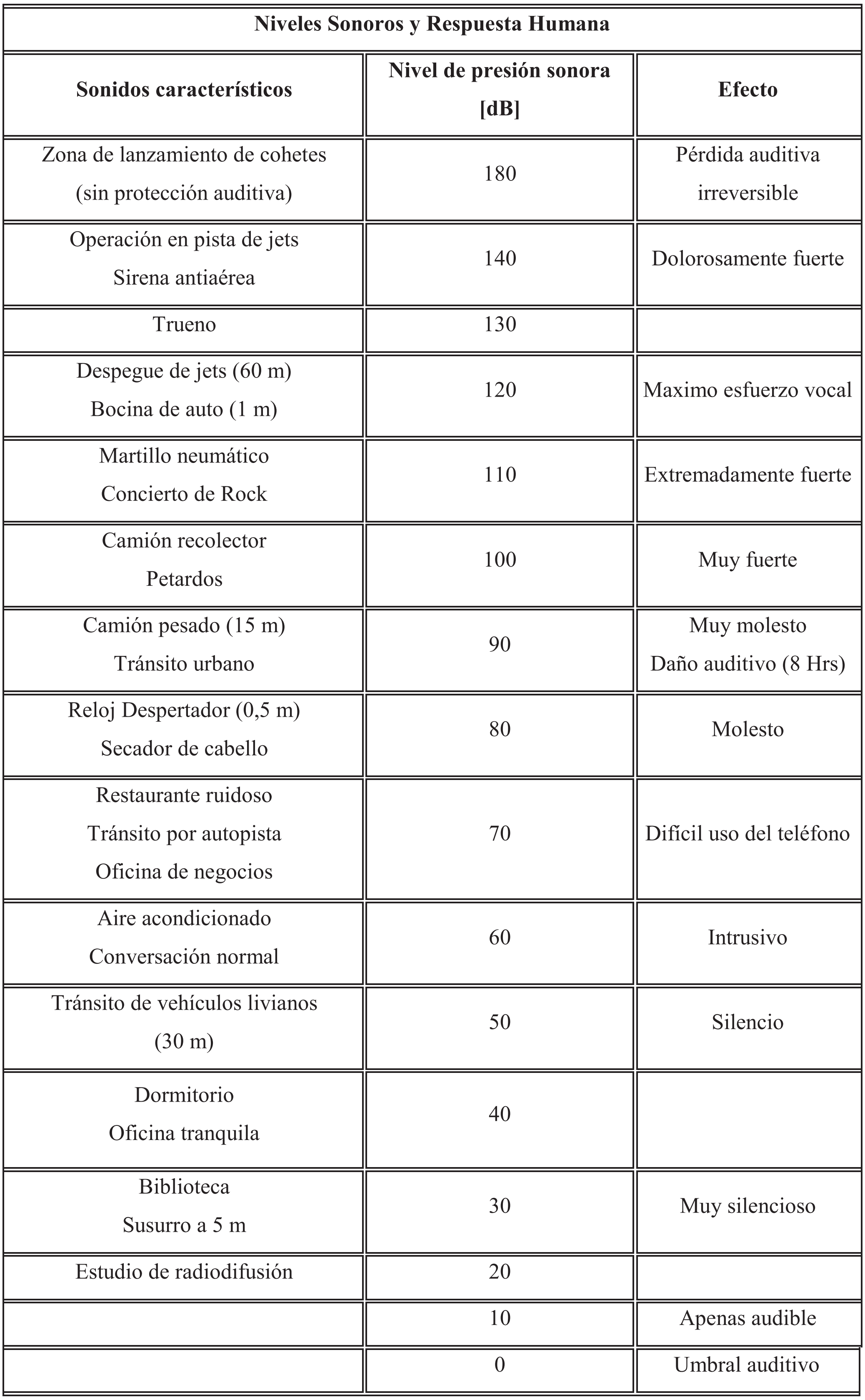
La potencia acústica es la cantidad de energía (potencia) radiada por una fuente determinada en forma de ondas por unidad de tiempo. Viene determinada por la propia amplitud de la onda, pues cuanto mayor sea la amplitud de la onda, mayor es la cantidad de energía (potencia acústica) que genera.

La potencia acústica es un valor intrínseco de la fuente y no depende del lugar donde se halle, el valor no varía por estar en un lugar reverberante, en uno seco o por estar al aire libre.

La medición de la potencia puede hacerse o en la fuente o a cierta distancia de la fuente, midiendo la presión que las ondas inducen en el medio de propagación. En cada caso respectivo se utilizaría la unidad de potencia acústica (que en el Sistema Internacional de unidades SI es el vatio acústico, W acústico) o la unidad de presión (que en el SI es el pascal, Pa).

La percepción que tiene el hombre de esa potencia acústica es lo que conocemos como “volumen” (que tan fuerte suena un sonido).

Figura 2.4.- Niveles sonoros y respuesta humana



Niveles Sonoros y Respuesta Humana		
Sonidos característicos	**Nivel de presión sonora [dB]**	**Efecto**
Zona de lanzamiento de cohetes (sin protección auditiva)	180	Pérdida auditiva irreversible
Operación en pista de jets Sirena antiaérea	140	Dolorosamente fuerte
Trueno	130	
Despegue de jets (60 m) Bocina de auto (1 m)	120	Maximo esfuerzo vocal
Martillo neumático Concierto de Rock	110	Extremadamente fuerte
Camión recolector Petardos	100	Muy fuerte
Camión pesado (15 m) Tránsito urbano	90	Muy molesto Daño auditivo (8 Hrs)
Reloj Despertador (0,5 m) Secador de cabello	80	Molesto
Restaurante ruidoso Tránsito por autopista Oficina de negocios	70	Difícil uso del teléfono
Aire acondicionado Conversación normal	60	Intrusivo
Tránsito de vehículos livianos (30 m)	50	Silencio
Dormitorio Oficina tranquila	40	
Biblioteca Susurro a 5 m	30	Muy silencioso
Estudio de radiodifusión	20	
	10	Apenas audible
	0	Umbral auditivo

Fuente: avmiramadrid.es

Suma de NPS: Para la suma de NPS existe un método matemático y un método empírico, a continuación se dará un ejemplo de cada uno para entenderlos mejor.

Ejemplo de suma de dB de NPS

Si NPS1=94dB y NPS2=94dB

Método matemático: para usar el método matemático se debe remplazar los valores de NPS en la fórmula del NPS total.

$$NPSt = 10\log\left(\frac{\sum P_n^{\ 2}}{\left(2*10^{-5}\right)^2}\right)$$

$$P^2 = \left(2*10^{-5}*10^{\frac{NPS}{10}}\right)$$

$$NPSt = 10\log\sum 10^{\frac{NPS}{10}}$$

Entonces:

$$NPSt = 10\log\left(10^{\frac{94}{10}} \oplus 10^{\frac{94}{10}}\right)$$

Suma NPSt=97.01dB

Método Empírico: este método consiste en restar los niveles P1-P2 y buscar el resultado de la resta en la tabla de la figura 2.5 para sumar el valor que indique la tabla al NPS mayor.

Figura 2.5.- Tabla para el método empírico

P1-P2	sumar al mayor
0 - 1	3 dB
2 - 3	2 dB
4 - 8	1 dB
>9	0 dB

Fuente: Franklin Robalino

Donde: P1 y P2 son las presiones asociadas al NPS1 y NPS2

P1-P2= 94-94= 0

Entonces se suma 3 dB al mayor (94)

Por lo tanto: NPSt =94+3=97 dB

Hay que tener en cuenta que se suma 3 si la fuente es lineal, en caso de que la fuente sea puntual se sumara 6.

Otros dB y sus usos

$$dBu = 10\log\left(\frac{voltaje}{0.775V}\right)^2 \quad --> procesadores, mixers, amplificadores$$

$$dBm = 10\log\left(\frac{potencia}{1mW}\right) \quad --> Eléctrica$$

$$dBv = 10\log\left(\frac{voltaje}{1V}\right)^2 \quad --> Potencias, parlantes$$

dB HL o dB hearing level.- es usado en audiometrías, en donde el nivel de referencia varía con la frecuencia.

NPS Equivalente

El NPS equivalente también llamado Leq, sirve para evaluar ruidos que son fluctuantes en el tiempo con un valor que tiene la misma energía acústica pero es constante, y se representa con la siguiente fórmula:

$$Leq = 10\log\left(\frac{\sum_{i=1}^{n} 10^{(\frac{Li}{10})}}{n}\right)$$

Donde

n = numero de mediciones

L= niveles de presión sonora

2.4. FUNCIONAMIENTO DEL OÍDO

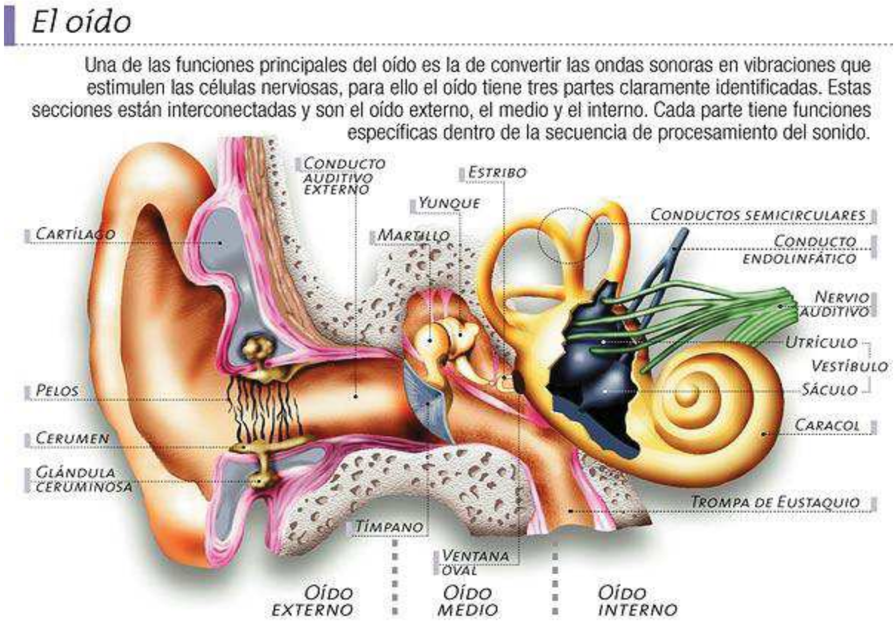
2.4.1. EL OIDO.-

El oído humano es un órgano sensorial muy complejo, que tiene dos funciones muy importantes que son la audición y el equilibrio. La audición permite al ser humano apreciar los sonidos tanto de la naturaleza como los creados por el hombre (ej. la música), también permite tener una idea de su entorno por medio de las reflexiones del sonido. Todo esto es muy importante en la audición ya que nos permite relacionarnos con el mundo que nos rodea, y de ella también depende mucho nuestro aprendizaje, como por ejemplo el aprender a hablar. El equilibrio por otro lado es una función del oído que no tiene relación con la audición, pero será mencionada debido a que tiene componentes dentro del oído.

La mayor parte del sistema auditivo está protegido por el hueso craneal, que protege las partes más sensibles del oído. El oído humano está formado por diferentes porciones especializadas según su función y propósito, dichas partes son de alta complejidad y por esta razón se explicará detenidamente cada uno de los componentes por separado.

El estudio del funcionamiento del oído se lo hará como lo hacen todos los textos y fuentes que fueron consultadas durante la investigación, es decir dividiendo al oído en tres partes: el oído externo, el oído medio y el oído interno, como se muestra en la figura 2.6. El funcionamiento de cada una de estas partes y sus componentes serán descritos con detalle más adelante.

Figura 2.6.- Anatomía del oído



El oído

Una de las funciones principales del oído es la de convertir las ondas sonoras en vibraciones que estimulen las células nerviosas, para ello el oído tiene tres partes claramente identificadas. Estas secciones están interconectadas y son el oído externo, el medio y el interno. Cada parte tiene funciones específicas dentro de la secuencia de procesamiento del sonido.

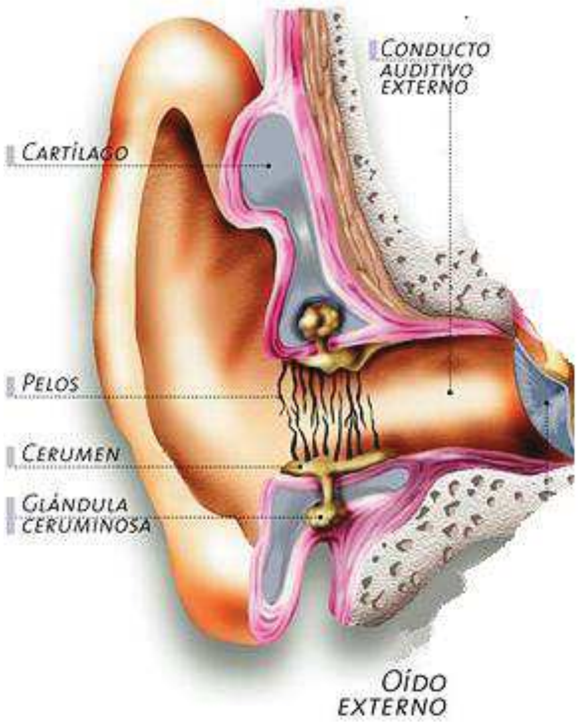
Conducto auditivo externo
Estribo
Yunque
Martillo
Cartílago
Conductos semicirculares
Conducto endolinfático
Nervio auditivo
Utrículo
Vestíbulo
Sáculo
Pelos
Caracol
Cerumen
Glándula ceruminosa
Trompa de Eustaquio
Tímpano
Ventana oval
Oído externo
Oído medio
Oído interno

Fuente: connect.in.com

2.4.2. EL OIDO EXTERNO.-

El oído externo está compuesto por la Oreja o pabellón auricular y el canal auditivo. La cabeza, hombros, y el dorso también son considerados ya que estos funcionan para llevar las frecuencias hacia el canal auditivo. El oído del ser humano es capaz de percibir sonidos en un rango de frecuencias que va desde los 20Hz hasta los 20kHz (vibraciones o ciclos por segundo). Dichas vibraciones acústicas son recogidas por la oreja.

Figura 2.7.- Oído Externo



Fuente: connect.in.com

El pabellón auricular u oreja es un cartílago plano elástico que tiene forma cóncava y está cubierto por piel gruesa: compuesta por hélix o borde exterior replegado, antihélix o eminencia central del pabellón que termina en una elevación llamada antítrago, concha o parte central y lóbulo, que es la parte inferior. El lóbulo es la única zona del pabellón que no posee cartílago.

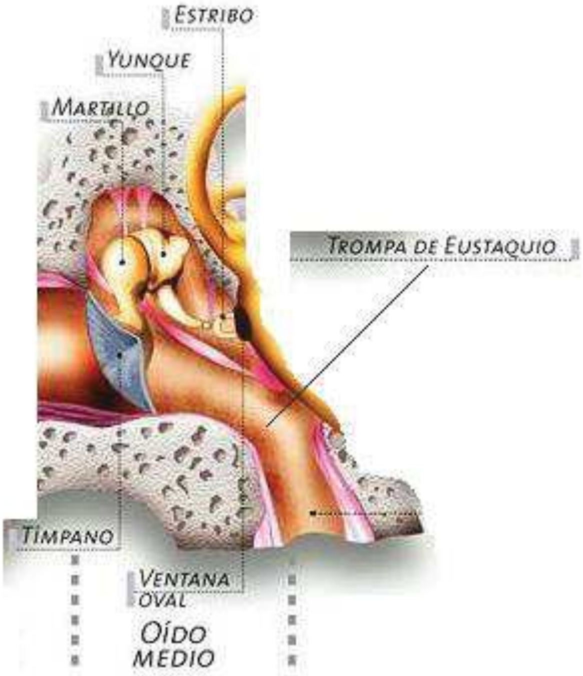
La oreja o pabellón auricular capta el sonido de una fuente y concentra la energía hacia el canal auditivo y actúa como un adaptador de impedancia acústica entre la impedancia baja del aire fuera del oído y la impedancia alta dentro de este, y su función más importante es la de ubicar las fuentes sonoras en el espacio. Al analizar bien la oreja o pabellón auricular se puede notar que su estructura es bastante compleja, dicho diseño complejo es lo que le ayuda para poder realizar su función, y si se compara dicho funcionamiento y complejidad con algún instrumento que haya sido creado por el hombre, se puede decir que es muy parecido a una antena parabólica, ya que esta también sirve para ubicar una fuente externa con el objetivo de captar sonidos o señales y llevarlas a un solo punto de captación sin que estas sufran cambios, al igual que lo hace la oreja en el ser humano.

El canal auditivo “meatus” está formado por una parte ósea y una cartilaginosa, este canal tiene una forma de un tubo entre 2,5 a 3,5 cm de largo y entre 0,65 y 1,2 cm de diámetro (resonancia del oído f_0 entre 2k a 4k). Dentro de este canal existen pelillos (folículos pilosos), y glándulas sebáceas, las cuales producen la cera en el oído y glándulas de ovillo, las cuales dan el color a la cera. La cera y los pelillos están ahí para la protección del oído medio, con la función de que las partículas del exterior que ingresan se peguen ahí y sean expulsadas progresivamente ya que partículas externas llegaran a pegarse al tímpano, el tímpano cambiaria su peso y por ende sus características. El canal auditivo también tiene una curvatura para evitar que objetos grandes llegasen a tocar el tímpano y dañarlo.

2.4.3. EL OIDO MEDIO.-

El oído medio está compuesto por el tímpano, el Martillo, Yunque, Estribo, trompas de Eustaquio (las cuales se conectan con la boca), la ventana coclear y la ventana oval. El oído medio también es conocido como cavidad timpánica o caja del tímpano.

Figura 2.8.- Oído Medio

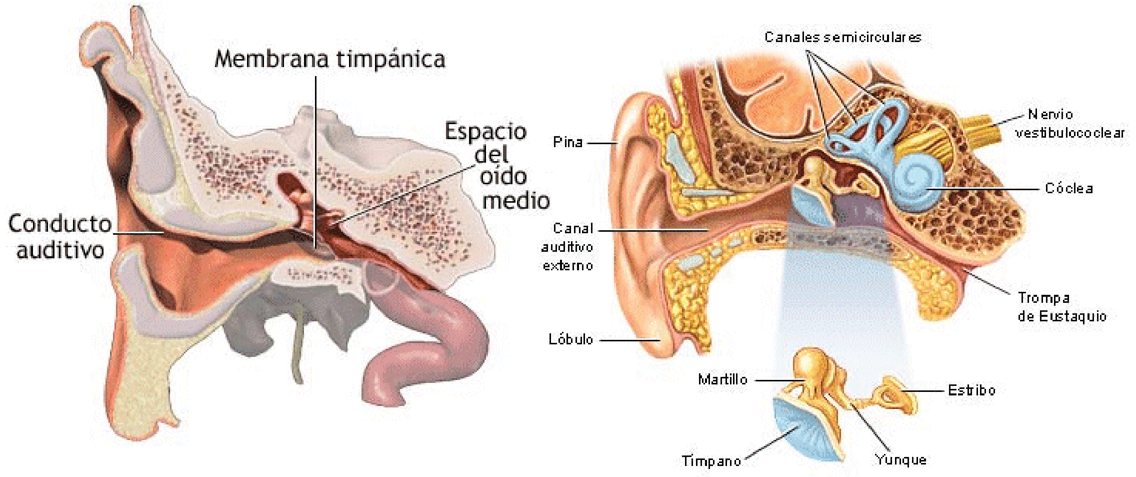


Fuente: connect.in.com

El oído medio es una cavidad de aproximadamente 2 cm^3 que existe en el hueso Temporal del cráneo. Esta cavidad se encuentra revestida de tejido epitelial y está llena de aire. Como ya se mencionó anteriormente el oído medio está delimitado por la membrana timpánica, y por una lámina ósea muy delgada, en la cual se encuentran dos pequeñas aberturas que tienen funciones determinadas. La ventana vestibular, también llamada ventana oval, y la ventana redonda o coclear. Esta porción del oído, debido a su temperatura, humedad y las conexiones que tiene con otros órganos, es altamente propensa a infecciones.

En la parte posterior de la cavidad se encuentran células que recubren el hueso Temporal, estas células son llamadas células mastoideas. Entre la cavidad y las células existe una cámara llamada antro mastoideo o timpánico. Mediante ésta cámara se esparcen muchas de las infecciones del oído medio hacia el hueso Temporal, como por ejemplo la infección conocida como mastioditis**,** entre otras que pueden llegar incluso al cerebro y causar daños mayores.

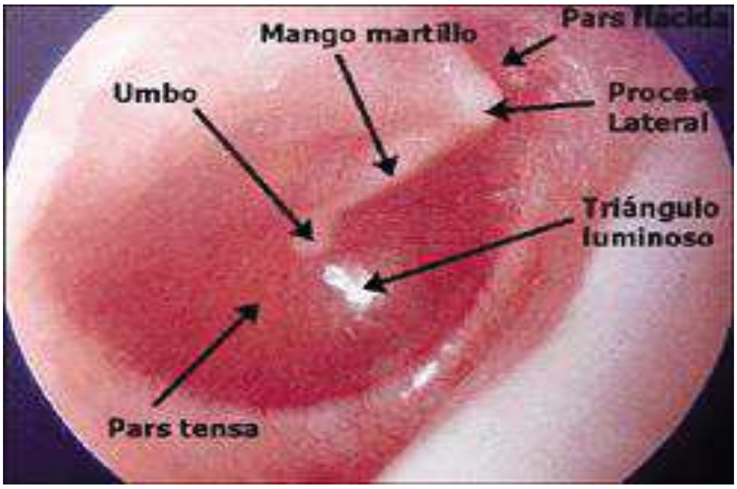
Figura 2.9.- Anatomía del Oído Medio


Membrana timpánica
Espacio del oído medio
Conducto auditivo
Canales semicirculares
Pina
Nervio vestibulococlear
Cóclea
Canal auditivo externo
Trompa de Eustaquio
Lóbulo
Martillo
Estribo
Tímpano
Yunque

Fuente: homeopatía-remedios.com y umm.edu

La membrana timpánica o tímpano se encarga de transformar las vibraciones acústicas en vibraciones mecánicas. El tímpano es una porción muy fina de tejido conectivo fibroso, el cual es semitransparente, mide aproximadamente entre 8 y 10 mm de diámetro, y está cubierta por piel delgada en su cara externa, por mucosa del oído medio en su cara interna, y es de forma cóncava hacia el canal auditivo. "Tiene una superficie aproximada de 0,055m2, que significa un área timpánica efectiva de 0.55 cm2 o 55 mm2. Es muy elástica debido a su estructura delgada y esto favorece a la función que debe desempeñar, que es la de vibrar de manera proporcional a las compresiones y descompresiones que se transmiten a través del aire como producto de un objeto vibratorio, de un sonido, para transmitir ésta vibración a las partes que siguen en la cadena del sistema auditivo. Para sonidos leves, la membrana timpánica presentará vibraciones leves, y para sonidos de altas intensidades, las vibraciones de la membrana timpánica serán de alta intensidad. Por este motivo, y por su estructura, la membrana timpánica es susceptible a los daños que le pueden ocasionar los sonidos muy intensos, pudiendo averiarse gravemente y hasta romperse, causando daños irreparables al sistema auditivo humano. Por esta razón no es aconsejable la exposición prolongada a sonidos de alta intensidad ni a sonidos de impacto de gran intensidad, como disparos, explosiones y otros similares.” [1]

Figura 2.10.- Foto amplificada de un tímpano



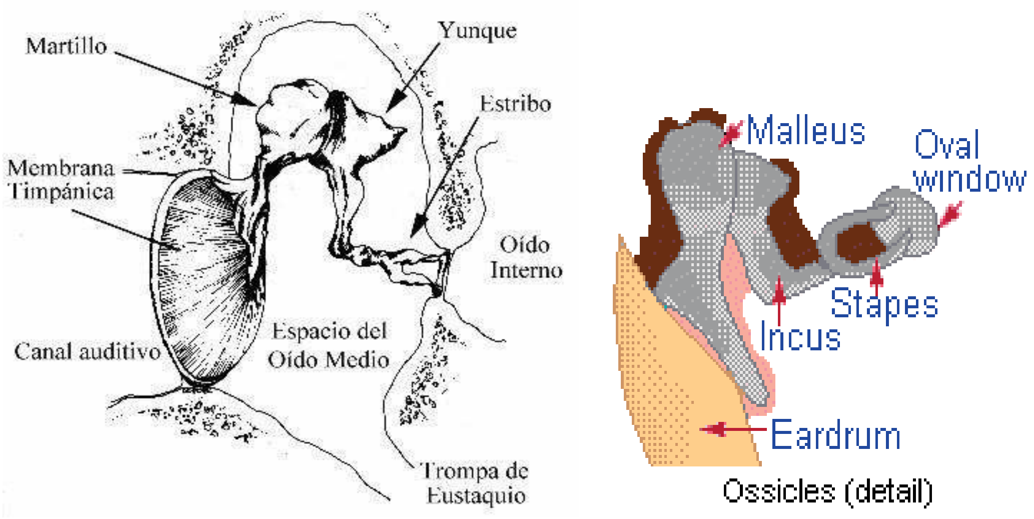
Fuente: Guía del sistema auditivo humano

Luego que el tímpano transforma las vibraciones acústicas en vibraciones mecánicas, dichas vibraciones mecánicas son transmitidas a través de los tres huesos (llamados Oscículos**)** más pequeños que hay en el ser humano, estos son nombrados según su forma: el martillo, yunque y estribo. Los huesos están conectados entre sí mediante unas articulaciones lubricadas por un humor viscoso llamado sinovia, que les da el nombre de articulaciones sinoviales. La parte interna del tímpano está conectada al mango del martillo, y la cabeza del martillo se une con el cuerpo del yunque, el cual a su vez se conecta con el estribo. Mientras que el estribo se une con la ventana oval o vestibular.

---

[1] Guía del sistema auditivo humano

Figura 2.11.- Martillo, Yunque y Estribo


Martillo
Yunque
Estribo
Membrana Timpánica
Oído Interno
Espacio del Oído Medio
Canal auditivo
Trompa de Eustaquio
Malleus
Oval window
Stapes
Incus
Eardrum
Ossicles (detail)

Fuente: Guía del sistema auditivo humano

Estos huesos tienen la función de transferir la energía del oído externo al interno (entre tímpano y ventana oval hay diferencia de 25:1 o 20:1 según el libro que se consulte y de acuerdo a sus dimensiones de huesecillos para formar un sistema de palanca), en los huesecillos existe una amplificación 22:1, se suele disminuir la amplitud y subir la intensidad (debido a que el medio de propagación cambia) y la frecuencia se mantiene. Este sistema de huesecillos puede actuar como un amplificador o a la vez como un limitador para proteger al oído de sonidos muy fuertes en NPS. La trompa de Eustaquio tiene una frecuencia de resonancia f0 alrededor de 1,2kHz en la cámara de aire y tiene un volumen de alrededor de unos 2 cm^3. Su función principal es nivelar presiones entre el oído medio y el oído externo de esta manera manteniendo al tímpano en su posición de reposo (otitis catarral). La ventana oval es la entrada al oído interno, está llena de líquidos endolinfáticos y perilinfáticos (dentro del caracol).

A continuación se discutirá a fondo los componentes del oído medio con sus respectivas funciones, para poder entender más detalladamente la forma de operación de este sistema.

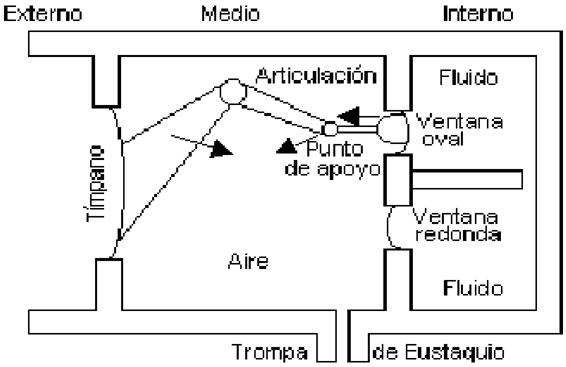
Los músculos que están conectados a los tres huesecillos ejercen una tensión, esta tensión sirve para protección de los huesecillos en casos de alta intensidad sonora y se denomina reflejo acústico. La limitación del reflejo acústico es que es necesario un tiempo de 500 ms, por lo tanto este reflejo es efectivo para sonidos crecientes, pero no lo es para sonidos impulsivos como en el caso de explosiones, disparos, etc. El reflejo acústico es efectivo con sonidos que se encuentren entre los 50 y 100 dB por encima del umbral auditivo de la persona.

El martillo está sometido a una tensión con respecto a la membrana timpánica, y esta tensión es relativa a uno de los dos músculos que se insertan en los huesecillos. El músculo tensor del tímpano (también conocido como músculo del martillo) tiene la función de evitar daños que puedan sufrir estos delicados componentes cuando sean sometidos a oscilaciones de gran amplitud (niveles altos de presión sonora) que pueden ser producidas por sonidos de gran intensidad. Otro músculo que también desempeña funciones de protección a estos tres huesecillos está ubicado en el estribo y se lo llama músculo estapedio o músculo del estribo. Este es el músculo estriado más pequeño que se puede encontrar en el ser humano y tiene la función de prevenir los movimientos excesivos que pueda sufrir el estribo, los cuales afectarían al sistema auditivo.

La disposición de los huesecillos tiene una relación de áreas de aproximadamente 30: 1 entre la membrana timpánica y la ventana oval, lo que quiere decir que si la superficie de la membrana timpánica es X y la de la ventana oval es Y, la relación entre ellas sería expresada como $X = 30Y$. Es decir estos tres huesecillos (que forman una palanca) son una especie de transformador mecánico, el cual prepara los sonidos recibidos para que puedan ser transmitidos a través de la ventana oval sin que se deteriore el funcionamiento de los órganos que se encuentran en el oído interno.

"Lo que hace es captar energía acústica en una gran superficie, que es la del tímpano, 55 mm^2, y concentrarla en la pequeña superficie de la ventana oval, 3.2 mm^2, lo que le profiere un aumento en el nivel de presión sonora de alrededor de 14 dB. La acción de los Huesecillos aumenta 1.3 veces la presión. La multiplicación de estos factores determina la "relación de transformación del oído medio", pasa con un incremento de aproximadamente unas 18 veces el valor inicial, una intensidad de 25 dB, lo que equipara la pérdida que se da naturalmente al pasar del medio aéreo al medio líquido, que, como se explica a continuación, es de aproximadamente unos 30 dB, y la "ineficiencia" de oído medio sería únicamente de unos 5 dB, y es la adaptación requerida por las partes del oído interno que son susceptibles a daños en caso de recibir altas intensidades sonoras sin la amortiguación del oído medio." [2]

Figura 2.12.- Estructura del oído medio



Fuente: Guía del sistema auditivo humano

Para resumir el funcionamiento del oído medio en otras palabras, se puede decir que la clave en el funcionamiento de esta parte del sistema auditivo es la transferencia de las vibraciones revividas que provienen de un medio aéreo (desde el oído externo), hacia un medio líquido (en el oído interno).

---

[2] Guía del sistema auditivo humano

Existen algunas constantes que intervienen en dicho proceso, como por ejemplo la impedancia característica del medio Zs, que viene a ser una oposición del medio por el cual el sonido va a ser transmitido. Para poder entender mejor a la impedancia acústica sin necesidad de irnos a la parte matemática, podemos emplear un ejemplo de un caso que está relacionado con el oído medio, lo cual ayudará a entender mejor lo que es la impedancia (Zs).

Al estar dentro de una habitación (no muy ruidosa) el sonido de una conversación viajaría a través de la habitación sin ningún problema, pasando desde el emisor (persona que emite el sonido), por el medio de transmisión por el cual se propaga el sonido (aire) hasta el receptor (persona que recibe el sonido). Por otro lado si cambiamos el medio de transmisión de aire a agua, es muy difícil trasmitir el mismo sonido hacia el receptor, debido a que este medio no es favorable para la transmisión de los sonidos que son emitidos por el ser humano. En este medio además de que el ser humano tiene que emplear mucha energía para poder emitir un sonido, se pierden muchas de las frecuencias que emite, lo cual dificulta el entendimiento (inteligibilidad).

Del ejemplo anterior se puede deducir que el agua presenta una impedancia mucho mayor a la que tiene el aire, debido a que tiene una mayor oposición al sonido que es transmitido por este medio.

“La Impedancia varía dependiendo de las características del medio, cuando una onda choca con una superficie líquida, un 99.9 % de la energía es reflejada. En otras palabras, si el sonido no es emitido dentro del líquido sino que es emitido en el aire y luego pasa al medio líquido, lo que logra pasar es un 0.1 % del total emitido. La atenuación que sufre en términos de intensidad sonora es de alrededor de 30 dB.” [3]

Los sonidos externos no necesariamente tienen que entrar en la oreja y viajar por el canal auditivo, al tímpano y hacia los tres huesecillos para ingresar luego al oído interno, se pueden también transmitir al oído interno, mediante los huesos del cráneo.

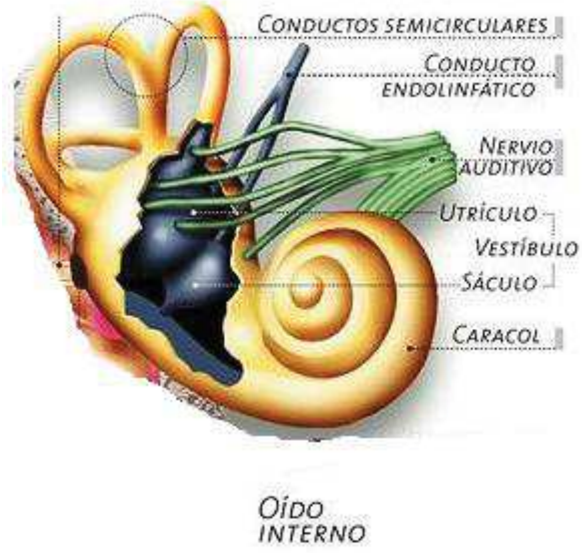
---

[3] Guía del sistema auditivo humano

2.4.4 EL OIDO INTERNO.-

El oído interno está compuesto principalmente por la Cóclea o caracol (dentro de este se encuentra el órgano de Corti y las células Ciliadas), canales semicirculares.

Figura 2.13.- Oído Interno


CONDUCTOS SEMICIRCULARES
CONDUCTO ENDOLINFÁTICO
NERVIO AUDITIVO
UTRÍCULO
VESTÍBULO
SÁCULO
CARACOL
OÍDO INTERNO

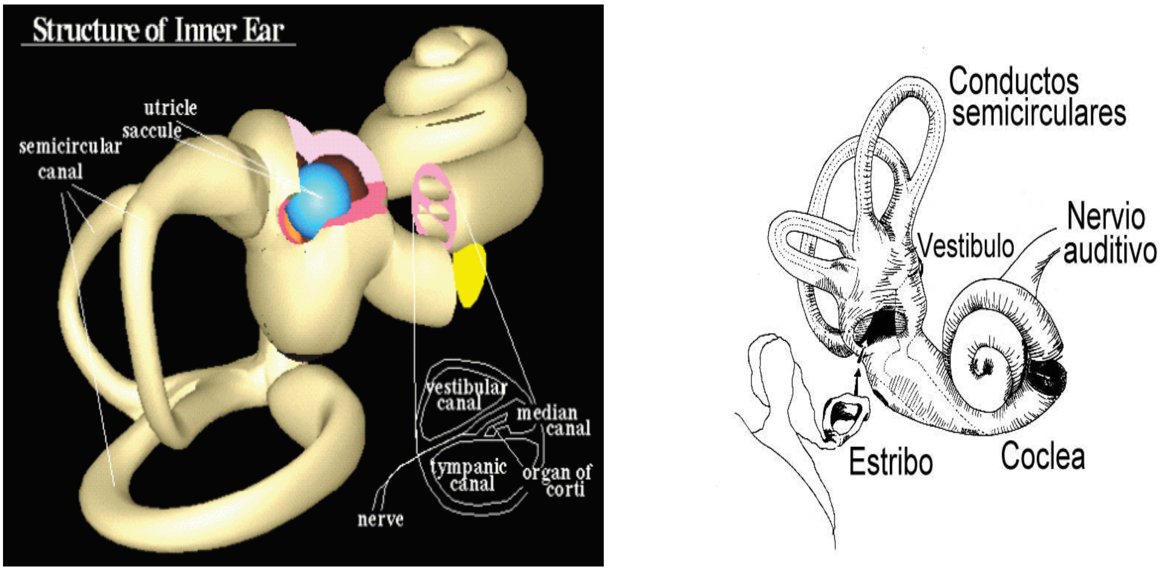
Fuente: connect.in.com

El oído interno también conocido como laberinto, se divide morfológicamente en laberinto óseo y en laberinto membranoso. El laberinto óseo es una cápsula ósea que rodea al laberinto membranoso que consiste en un sistema hueco el cual contiene la endolinfa. Entre laberinto óseo y laberinto membranoso se encuentra la perilinfa, que es en parte un filtrado de la sangre y en parte una propagación de líquido cefalorraquídeo. Este laberinto óseo consiste en unas cavidades llamadas también petrosa del hueso temporal del cráneo. Y estas cavidades a su vez se dividen en tres partes que han sido nombradas según su forma: el vestíbulo, la cóclea o caracol y los canales semicirculares óseos, también conocidos como canales semicirculares.

La porción central del laberinto óseo llamada vestíbulo, tiene una forma oval y contiene una porción del laberinto membranoso que consiste en dos sacos conectados entre sí por un pequeño conducto, estos dos sacos se llaman utrículo y sáculo. El utrículo está en comunicación con los canales semicirculares, y el sáculo se comunica con el caracol.

Los canales o conductos semicirculares nos ayudan con el equilibrio y la orientación espacial o localización perceptual. Básicamente son 3 canales, para determinar tres coordenadas (x,y,z). Los canales semicirculares son tres tubos de forma semicircular. Dos de ellos están en una posición vertical y el tercero en una posición horizontal (anterior, posterior y lateral).

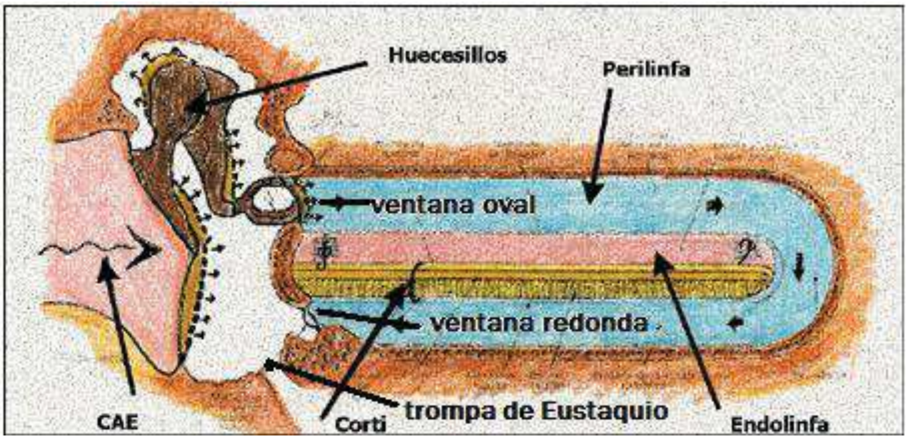
Figura 2.14.- Anatomía del Oído Interno


Structure of Inner Ear
utricle
saccule
semicircular canal
vestibular canal
median canal
tympanic canal
organ of corti
nerve
Conductos semicirculares
Vestibulo
Nervio auditivo
Estribo
Coclea

Fuente: Guía del sistema auditivo humano

La cóclea o caracol se encuentra al frente del vestíbulo, y su nombre, caracol, es debido a la forma que tiene, la cual es muy parecida a la concha de un caracol. Este consiste en un pilar óseo central llamado modíolo, al rededor de este se envuelve un canal óseo espiral, que alcanza a envolverse más o menos tres vueltas. En una persona promedio, se envuelve dos vueltas y tres cuartos alrededor del modíolo. En el caracol entra presión por la ventana oval y desfoga presión por la ventana redonda (ver figura 2.15) que está ubicada justo al inferior de la ventana oval, estas ventanas sirven para generar diferencia de presión para que se mueva la membrana basilar.

Figura 2.15.- Vista dentro del caracol (si colocamos al caracol horizontalmente)



Fuente: Guía del sistema auditivo humano

Las vibraciones mecánicas del oído medio al pasar al oído interno se transforman en ondas, las cuales son transmitidas a través de un medio líquido (líquidos endolinfáticos y perilinfáticos dentro del caracol). La membrana basilar se encuentra dentro del caracol y tiene células ciliadas o pelillos que conforman al órgano de Corti, dichas células son selectivas a la frecuencia. En el momento en que la onda que representa el sonido pasa por las células ciliadas, estas transformaran la información recibida a impulsos eléctricos, los cuales pasaran a través del sistema nervioso hasta llegar al cerebro, en donde se decodificarán los impulsos eléctricos para que el receptor (el ser humano) sepa lo que está captando el oído.

Para entender mejor la estructura dentro de la cóclea, se puede realizar un corte transversal, el cual mostrará que la cóclea está internamente dividida en tres canales o conductos.

Figura 2.16.- Vista dentro del caracol (corte transversal) y foto amplificada

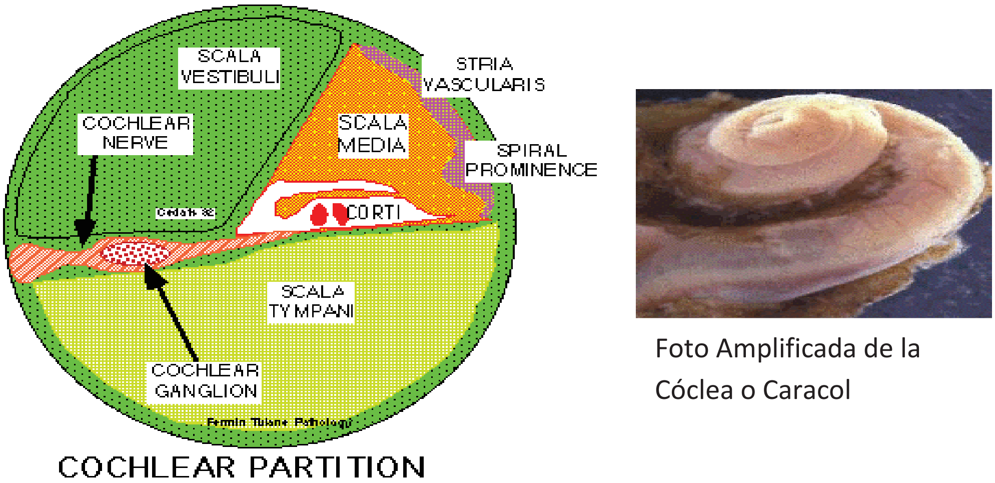

SCALA VESTIBULI
STRIA VASCULARIS
COCHLEAR NERVE
SCALA MEDIA
SPIRAL PROMINENCE
CORTI
SCALA TYMPANI
COCHLEAR GANGLION
Fermin Tulane Pathology
COCHLEAR PARTITION

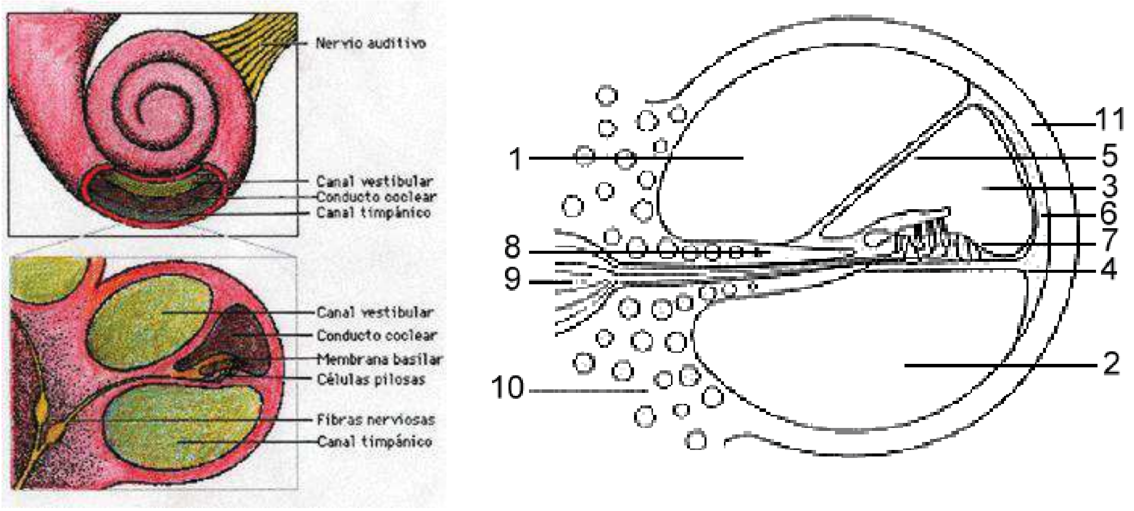
Foto Amplificada de la Cóclea o Caracol

Fuente: Guía del sistema auditivo humano

Las divisiones de éstos canales, al ser observados en el corte transversal, tienen la forma de una letra Y. La parte vertical de la Y está compuesta por un tabique óseo, mientras que las porciones diagonales de la Y son extensiones del laberinto membranoso. El conducto o canal que se encuentra por encima del tabique óseo es conocido como rampa o escala vestibular (SCALA VESTIBULI), mientras que el conducto que se encuentra por debajo es conocido como rampa o escala timpánica (SCALA TYMPANI). La perilinfa del vestíbulo y escala vestibular están conectadas entre sí. La rampa timpánica se comunica con la vestibular en el vértice de la "concha del caracol", que es conocido como el helicotrema.

El tercer canal o conducto, que se encuentra ubicado entre las secciones diagonales de la Y, es decir en el laberinto membranoso, recibe el nombre de conducto coclear o SCALA MEDIA. Entre el laberinto membranoso y la escala o rampa vestibular existe una finísima membrana llamada pared vestibular del conducto coclear, esta también es conocida como membrana vestibular de Reissner o simplemente membrana de Reissner, esta membrana es muy delgada, pues apenas tiene dos células de ancho. El laberinto membranoso está también separado de la rampa o escala timpánica por una fina membrana, la cual no es tan delgada como la anterior, conocida como lámina o membrana basilar.

Figura 2.17.- Sección de una espira de la cóclea. 1. Rampa vestibular (perilinfa); 2. Rampa timpánica (perilinfa); 3.rampa media (endolinfa); 4. Membrana basilar; 5. Membrana de Reissner; 6. Ligamento espiral; 7. Órgano de Corti; 8. Lámina espiral; 9. Ganglio de Corti; 10. Colúmena; 11. Lámina de los contornos.

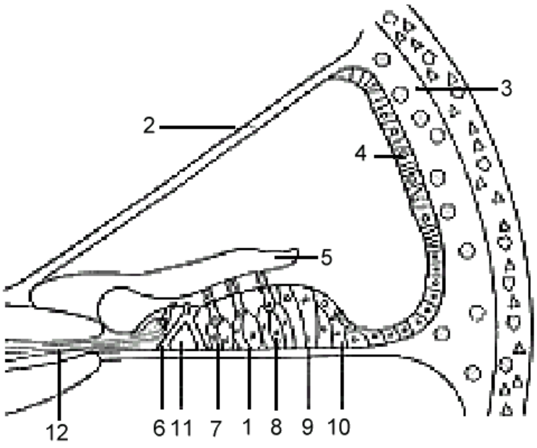

Nervio auditivo
Canal vestibular
Conducto coclear
Canal timpánico
Canal vestibular
Conducto coclear
Membrana basilar
Células pilosas
Fibras nerviosas
Canal timpánico
11
1
5
3
6
8
7
4
9
2
10

Fuente: Guía del sistema auditivo humano

Al interior de la cóclea se encuentra el denominado órgano espiral, también conocido como órgano de Corti, que ya fue mencionado anteriormente. El órgano de Corti es el órgano de la audición, y está conformado por una cantidad de células de epitelio que se encuentran en la cara interna de la membrana basilar o lámina basilar, junto a otras células conocidas como células de sostén o de soporte y otras llamadas células pilosas, las cuales están ubicadas sobre la lámina o membrana basilar y en una cantidad limitada.

“Algunos autores aseveran que hay cuatro hileras de células pilosas y que suman un total de treinta mil células, otros dicen que se encuentran en una sola hilera. Lo cierto es que hay células pilosas internas distribuidas en sentido interno o medial en una sola hilera, dispuestas a lo largo de toda la cóclea o caracol, y células pilosas externas dispuestas así mismo en toda la cóclea, pero en varias hileras.” [4]

Figura 2.18.- Organo de Corti. 1. Membrana basilar; 2. Membrana de Reissner; 3. Ligamento espiral; 4. Estría vascular; 5. Membrana tectoria; 6. Células ciliadas internas (CCI); 7. Células ciliadas externas (CCE); 8. Células de Deiters; 9. Células de Hensen; 10. Células de Claudius; 11. Canal de Corti; 12. Fibras de la primera neurona del VIII par craneal.



Fuente: Guía del sistema auditivo humano

---

[4] Guía del sistema auditivo humano

Estas células cuentan con prolongaciones que tienen forma de pelillos, y son conocidas como prolongaciones filiformes. Estos pelillos están en contacto con el líquido que se encuentra en el interior del canal o conducto coclear (endolinfa), mientras que sus extremos contrarios que se encuentran adosados al epitelio, también conocidos como extremos basales, se encuentran en contacto con las células del vestibulo-coclear, conocido también como nervio vestibular.

Las células pilosas son las partes más importantes del Sistema Auditivo, ya que estas son los receptores de las sensaciones auditivas. Estas células pueden sufrir daños debido a la exposición a ruidos intensos, y en caso de recibir daños estas no tienen la capacidad de regenerarse.

Para resumir el funcionamiento del oído interno, las vibraciones recibidas en la ventana oval se transmiten por medio de los líquidos hacia las células ciliadas del órgano de Corti, transmitiéndose por último esta señal al nervio acústico y de él al cerebro. Es decir el oído interno se encarga de transformar la energía mecánica producida por las ondas sonoras en energía eléctrica a través del órgano de Corti.

2.5.- FISIOLOGÍA DE LA AUDICIÓN

A continuación se detallará paso a paso el proceso de la audición, desde el momento en que el sonido es captado por el oído del receptor (ser humano) hasta el momento en que la información es transportada vía nervios auditivos hacia el cerebro, para dar la idea de qué es lo que se ha oído.

Primero las ondas sonoras producidas por una fuente externa llegan hasta la oreja o pabellón auditivo, donde son re direccionadas hacia el canal auditivo externo o meato auditivo sin que las ondas sufran alteraciones de ningún tipo, es decir, las mismas ondas que recibe la oreja son exactamente las mismas al pasar dentro del canal auditivo. Una vez dentro del canal auditivo el sonido comprime y descomprime las moléculas de aire que están a su paso, hasta llegar a la membrana timpánica o tímpano.

La fina membrana timpánica, como respuesta a este estímulo, se mueve en proporción a la velocidad en que vibran las partículas. Es decir, para vibraciones en bajas frecuencias (sonidos graves), la vibración de la membrana timpánica será lenta, y para vibraciones en altas frecuencias (sonidos agudos), la vibración de la membrana timpánica será más rápida. En cualquiera de los casos, el desplazamiento de la membrana timpánica será de una distancia muy pequeña.

Dicha vibración de la membrana timpánica producirá una vibración proporcional en los huesecillos del oído medio. Primero la vibración de la membrana timpánica generará un movimiento del mango del martillo, el cual está conectado a la cara interna del tímpano. El martillo después provocará la vibración del yunque el cual hará vibrar al estribo y así el sonido o la vibración será transmitida a través de los tres huesecillos.

La vibración transmitida a través del estribo llegará hasta la ventana oval o vestibular, puerta de entrada hacia el oído interno. Esta vibración causará que la perilinfa que se encuentra en la escala vestibular sufra desplazamientos ondulatorios proporcionales a las vibraciones de la ventana oval. Es decir que cuando la ventana oval se desplaza hacia adelante empujará a la perilinfa de la escala vestibular o rampa.

La presión generada en la perilinfa moverá la membrana vestibular y esto causará un incremento de presión en la endolinfa que se encuentra al interior del conducto coclear. La membrana basilar o lámina es luego empujada y se mueve como consecuencia de dicha presión. El movimiento de la membrana basilar ocasionara que las células pilosas que se encuentran en el órgano espiral o de Corti vibren, y como estas están en contacto con la membrana tectoria, causará que las células pilosas se muevan, lo que genera la transducción de vibraciones en impulsos nerviosos (impulsos eléctricos).

Las distintas frecuencias que pueden ser transmitidas por medio de vibraciones en el oído interno serán recibidas en distintas zonas de la membrana basilar para su transducción a impulsos nerviosos. La membrana basilar es menos susceptible al movimiento (más rígida) cerca de la base de la cóclea y esta rigidez disminuye conforme se aleja de la base y se acerca a lo que sería el ápice de la cóclea. También cerca de la base (zona más rígida) es donde se transducen las altas frecuencias (sonidos agudos), mientras que en la parte en que la membrana es más flexible y móvil (cerca del ápice de la cóclea), es donde transducen las bajas frecuencias (sonidos graves).

El helicotrema se encuentra en el vértice de la cóclea y es una especie de membrana que tiene una importante función. Debido a que el líquido dentro del oído interno (la perilinfa) no puede ser comprimido, cuando una onda desplaza las partículas del líquido, éstas vibraciones deben desfogarse de alguna manera (para que no se deterioren las partes internas que se encuentran en el medio líquido). En este caso lo que sucede es que el helicotrema se deflecta, o se flexiona, como consecuencia de la presión interna con el propósito de evitar la compresión en la cóclea. Algo muy similar ocurre con la ventana redonda, pues cuando la presión del líquido aumenta en la escala o rampa timpánica, la ventana redonda es desplazada hacia el oído medio. De estas dos maneras la presión que se origina en el interior de la cóclea es controlada para que no se deterioren los delicados componentes del oído interno.

Como último paso los impulsos nerviosos producidos por la transducción de las vibraciones de la membrana basilar viajarán hasta el área auditiva del cerebro. Después se evaluarán los datos de dichos impulsos nerviosos por medio de los centros nerviosos destinados a esta labor.

Dependiendo de los datos de los impulsos nerviosos, se generará una reacción de los mismos, en caso de tratarse de un sonido agradable o desagradable, conveniente o perjudicial, el cerebro enviará las órdenes adecuadas, según el tipo de estímulo que esté recibiendo el centro nervioso. Estas reacciones por ejemplo ordenarán a la persona que se tape los oídos o se aleje de la fuente en caso de que el sonido sea intenso y perjudicial para el receptor, o a su vez si el sonido es muy bajo en nivel, se aumentará la concentración.

2.6. PSICOACÚSTICA (CONCEPTO).-

La psicoacústica es el estudio de la percepción subjetiva del ser humano hacia los sonidos. También puede ser definida como el estudio psicológico de los parámetros físicos de la acústica.

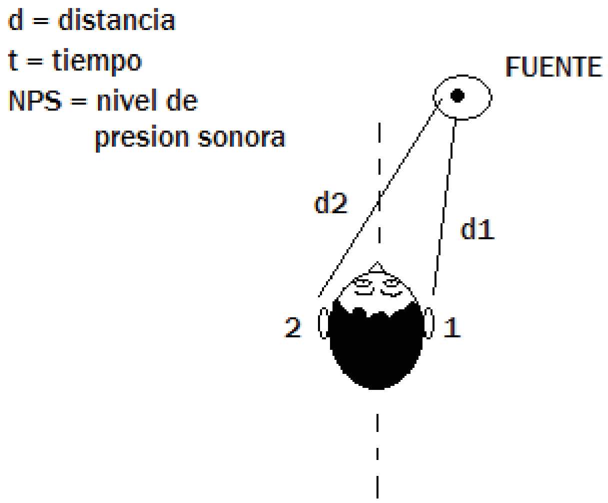
El estudio de la psicoacústica incluye el cómo escuchamos, las respuestas psicológicas, y el impacto psicológico de los sonidos en el sistema nervioso humano.

La psicoacústica explica la respuesta subjetiva de todo lo que escuchamos. Entre otras cosas, estudia la relación entre señales de sonido y su percepción acústica, la fisiología del sistema auditivo, y la percepción psicológica del sonido, para poder explicar la respuesta y el comportamiento del ser humano, las habilidades y limitaciones del oído humano, y el complejo proceso auditivo que ocurre dentro del cerebro. El escuchar involucra una respuesta de comportamiento a los atributos físicos del sonido incluyendo intensidad, frecuencia, y otras características basadas en el tiempo que permiten al sistema auditivo determinar distancia, dirección, sonoridad, altura, y tonos de varios sonidos individuales que ocurren simultáneamente.

2.6.1. AUDICIÓN BINAURAL.-

Está relacionado con el hecho de que nuestros oídos están separados por una cierta distancia permitiendo localizar la fuente del sonido por medio de una pequeña diferencia en el tiempo, fase, e intensidad del sonido, que llega a cada oído. El oído es capaz de detectar una diferencia de tiempo hasta de 30 ms entre dos sonidos. La comparación entre el tiempo de los sonidos recibidos entre el oído izquierdo y el oído derecho y la evaluación de intensidad entre los dos oídos permiten automáticamente, sin la actuación de la conciencia, identificar una aproximación de la ubicación de la fuente de sonido. Ejemplo si d1 < d2 => t1 < t2 => NPS1 > NPS2 (ver figura 2.19)

Figura 2.19.- (d1 < d2 => t1 < t2 => NPS1 > NPS2)



Fuente: Fanklin Robalino

2.6.2. ALTURA.-

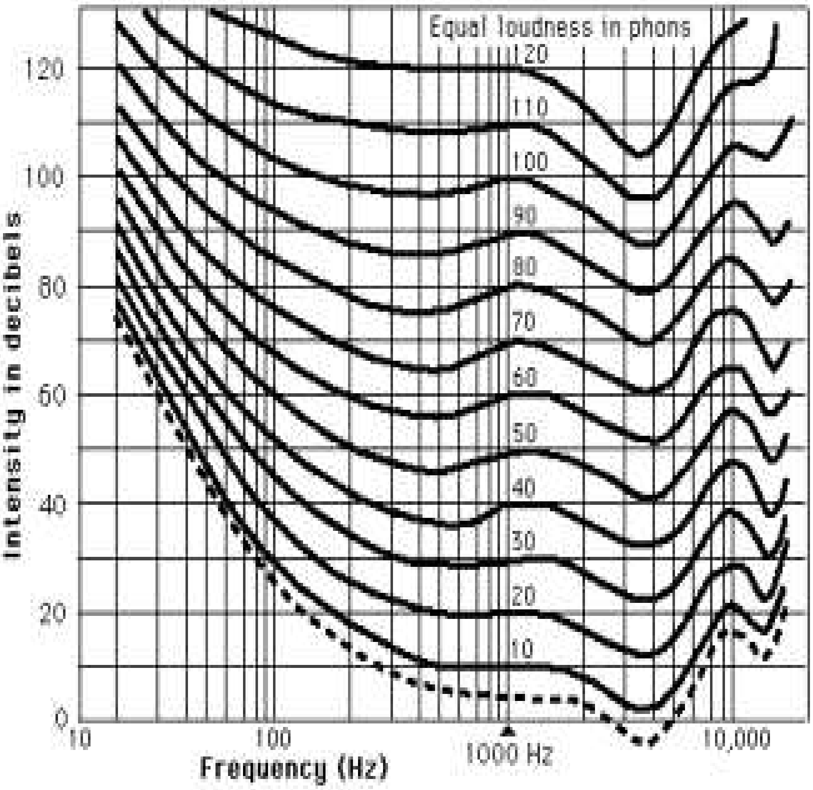
En la psicoacústica, el término altura es considerado una percepción psicológica de la frecuencia de un sonido. Se han realizado diversos estudios para poder encontrar la relación entre altura y frecuencia. En música, la altura se define como la posición de un sonido en el rango de frecuencias. Es la característica de este (el sonido) la cual permite al oyente organizarlos en una escala desde graves hacia agudos. Son graves o agudos en altura dependiendo de la frecuencia de vibración de las ondas sonoras que producen dicho sonido.

2.6.3. SONORIDAD.-

Es la percepción subjetiva de la intensidad de un sonido, que permite ordenar a los sonidos en una escala desde silenciosos a bulliciosos. La intensidad es definida como la potencia del sonido por unidad de área.

En la figura 2.3 se muestra la curva que describe los estudios realizados por Fletcher & Munson en los años 30´s, recalculadas luego por Robinson & Dadson, denominadas Curvas Isofónicas o de igual Nivel de Sonoridad. Cada curva describe un rango de frecuencias que son percibidas como igual Sonoridad. Las curvas están en unidades de fonos (*phons*), midiendo el NPS (nivel de presión sonora) de una curva en 1kHz. Estas curvas muestran que el oído humano es menos sensible a frecuencias bajas, y además que la máxima región de sensibilidad para el oído humano está entre 1kHz y 5kHz. La curva con línea punteada representa el *threshold* o umbral de escucha del oído.

Figura 2.20: Curvas de Fletcher & Munson



Fuente: img.lx.it.pt

2.6.4. TIMBRE.-

El sonido puede ser generalmente caracterizado por Altura, Sonoridad y por la calidad del sonido o Timbre. El Timbre es el atributo de una sensación auditiva en términos que el receptor puede distinguir o identificar de qué fuente provienen dos sonidos similares que tengan la misma Altura y Sonoridad. El timbre es mayormente determinado por su contenido armónico y las características dinámicas del sonido tales como vibrato, y su envolvente: Ataque (*Attack*), Sostenimiento (*Sustain*), Relajamiento (*Release*) y Decaimiento (*Decay*). El timbre es la característica del sonido la cual nos permite discriminar sonidos producidos por diferentes instrumentos que suenan al mismo tiempo y con la misma altura.

2.6.5. ENMASCARAMIENTO.-

Es una propiedad del sistema auditivo humano donde algunos sonidos desaparecen en presencia de sonidos más poderosos (altos en sonoridad). El sonido más fuerte es llamado el enmascarante y el sonido débil es llamado el enmascarado. Este aspecto del oído humano tiene implicaciones importantes para el diseño de codificadores perceptuales de audio como el mp3.

2.6.6. BANDAS CRÍTICAS.-

Una banda crítica es la banda más pequeña de frecuencias que activan la misma zona de la membrana basilar. El oído puede distinguir tonos de pocos hertz de diferencia en frecuencias bajas y los tonos deben diferenciarse por cientos de hertz en frecuencias altas para poder ser distinguidos. En cualquiera de los casos, las células ciliadas (hair cells) responden a la estimulación más potente en su región o banda crítica. El concepto de banda crítica fue introducido por Fletcher en 1940. Experimentos han demostrado que las bandas críticas son más estrechas en frecuencias altas que en frecuencias bajas.

Las bandas críticas tan bien pueden ser explicadas de otro modo, cuando tenemos dos sonidos separados de igual Sonoridad y con un altura cercana, su combinación de Sonoridad cuando se juntan será solamente un poco mayor en Sonoridad que la de las fuentes por separado. Se puede decir que se encuentran en la misma banda crítica en donde están compitiendo por el mismo nervio en la membrana basilar del oído interno. Si en cambio estos dos sonidos fuesen mucho más separados en altura, el nivel de Sonoridad percibido de la combinación de estos tonos será considerada mayor debido a que no se sobreponen en la membrana basilar y por lo tanto no compiten por las mismas células ciliadas. Si es que los tonos estuviesen más separados por frecuencia, no estarían dentro de una misma banda crítica, y la combinación puede ser percibida como dos veces mayor en Sonoridad que una sola por separado. La teoría de bandas críticas es muy importante para el concepto de audición debido a que esto nos muestra que el oído discrimina entre energías dentro de la banda, y energías fuera de la banda, esto promueve al efecto de enmascaramiento.

2.7. AUDIOMETRÍA

Existen varios tipos de exámenes para comprobar la capacidad auditiva del ser humano, y la audiometría es la más usada.

La audiometría es parte de la audiología (estudio del oído, desde el punto de vista acústico) que se ocupa de la medida de la capacidad auditiva y sus alteraciones. Esta se divide en audiometría objetiva y subjetiva.

La audiometría objetiva es una audiometría de respuesta electrónica y está dividida en emisiones otoacústicas (OAE), timpanometría, y respuestas auditivas tronco encefálicas (ABR). Todas estas son realizadas sin necesidad de que el paciente tenga una participación activa durante el proceso, pues las señales se generan, miden y registran los datos necesarios. Estas audiometrías son muy útiles en especial para medir la audición de recién nacidos.

La audiometría subjetiva, depende mucho de la participación activa del paciente, y está dividida en audiometría refleja, audiometría por refuerzo visual, audiometría convencional de tonos puros, y audiometría vocal.

La audiometría convencional de tonos puros, proporciona una medición precisa de la audición. Para evaluar la conducción aérea, se colocan unos audífonos que van conectados al audiómetro. Los tonos puros producidos por el audiómetro tienen una intensidad y frecuencia controlada y son transmitidos generalmente a un oído a la vez. Se solicita levantar la mano, presionar un botón o indicar de otro modo cuando se escuche un sonido.

Luego, se grafica la intensidad (volumen) mínima requerida para escuchar cada tono. Se coloca un accesorio, llamado oscilador óseo, contra el hueso por detrás de cada oído (hueso mastoideo) para evaluar la conducción ósea.

Este sistema es muy preciso y permite determinar el grado y tipo de sordera en un individuo, e incluso su origen en algunos casos mediante el análisis de la gráfica. Este examen no causa ninguna molestia a los pacientes y tiene un gran valor diagnóstico.

Actualmente en el Ecuador, los exámenes de audiometría se realizan con audiómetros análogos, usando marcas tales como Interacoustics AD 229 en el Centro Médico Meditrópoli y Maico audiometer MA40 en la fundación Mosquera.

2.7.1. PROCESO DE MEDICIÓN

La audiometría (tonal) se realiza mediante el uso de un aparato electrónico llamado audiómetro, el cual es operado por un técnico entrenado llamado Audiólogo. Este examen médico con el equipo de audiometría es simple y sin dolor, y es efectuado con un simple proceso que se explicará a continuación.

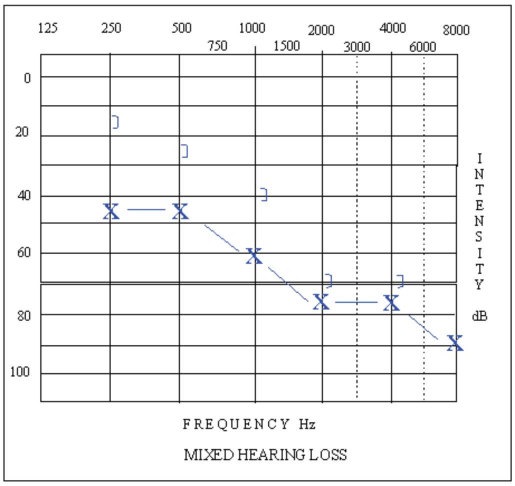
El Audiólogo entrenado (especialista) usa un audiómetro para realizar el examen de audiometría. Este equipo emite sonidos (tonos puros), en varias frecuencias y en diferentes niveles de presión sonora (NPS). Este examen se lo hace por lo general en un cuarto de evaluación que sea a prueba de ruido, para que de este modo el ruido externo no afecte la medición.

La persona que está siendo evaluada usa unos audífonos que bloquearan sonidos provenientes de fuera de los audífonos (ruido de fondo) y estos transmitirán los tonos puros con diferentes niveles de presión sonora a cada uno de los oídos por separado.

Cuando el paciente escuche un tono, levantará el brazo para indicar que el sonido ha sido detectado. El Audiólogo bajará el volumen y repetirá el mismo tono hasta que el paciente ya no lo escuche. Este proceso es repetido con diferentes tonos (frecuencias) para el un oído y luego para el otro. Por lo general los niveles de sensibilidad son diferentes entre los dos oídos.

Los resultados de la audiometría son tabulados en una gráfica llamada audiograma (el oído derecho con color rojo y el izquierdo con color azul). Este gráfico va generalmente de frecuencias bajas hacia frecuencias altas ubicadas de izquierda a derecha como se muestra a continuación (figura 2.21). En la figura también se encuentran los niveles de presión sonora de los tonos ubicando a los sonidos más bajos en la parte superior de la gráfica y los más altos en nivel en la parte inferior.

Figura 2.21.- Audiograma



Fuente: illinoissoundbeginnings.com

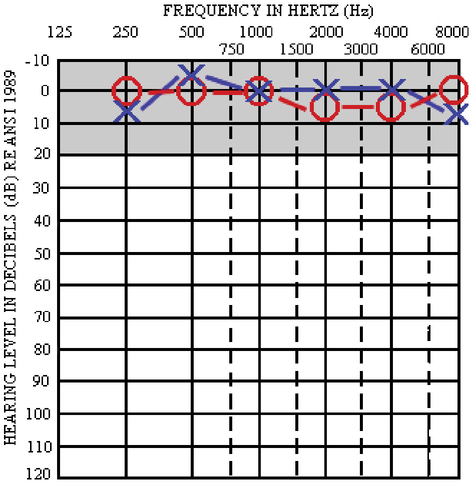
Para medir el nivel de presión sonora se usa la unidad de decibeles HL que se explicó anteriormente (página 12). La mayoría de los sonidos asociados con los parámetros de la voz están generalmente en el rango de 20-50 decibeles.

Antes del examen de audiometría se pueden examinar cada uno de los oídos con un otoscopio, para determinar si es que existe algún tipo de obstrucción en el canal auditivo ya sea por cera o algún otro material.

Los resultados normales de una audiometría para una persona adulta se obtienen cuando la persona puede reconocer todos los tonos generados con una presión sonora entre 0 a 20 dB en ambos oídos. En cambio los resultados anormales de una audiometría son considerados cuando existe una gran diferencia entre los niveles de presión sonora entre ambos oídos, o si es que la persona no puede escuchar el rango de frecuencias y niveles de presión normales. Los patrones de respuesta de los oídos que son graficados en el audiograma, pueden ser usados por el Audiólogo para poder identificar si es que existe una pérdida auditiva significante y si el paciente pudiese necesitar un aparato auditivo (hearing aid) o una cirugía para corregir la audición.

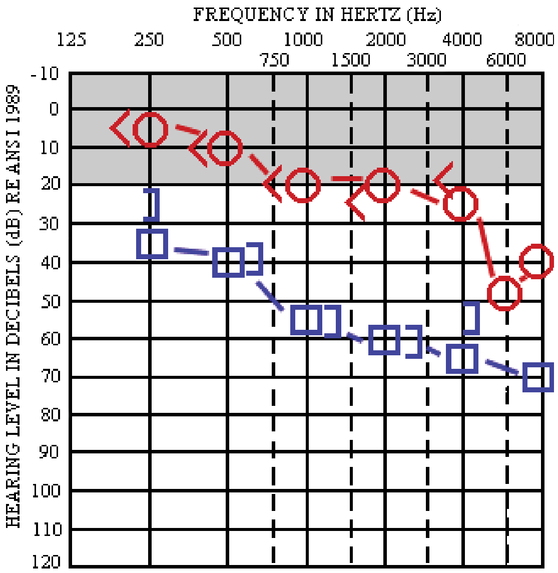
A continuación se mostrarán algunos gráficos de audiogramas para referencia.

Figura 2.22.- Audiograma con resultados normales



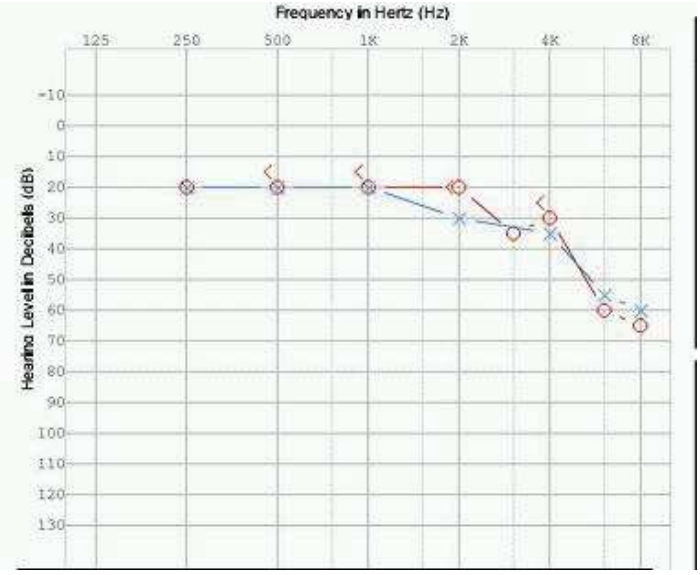
Fuente: dizziness-and-balance.com

Figura 2.23.- Audiograma con resultados anormales



Fuente: dizziness-and-balance.com

Figura 2.24.- Audiograma típico de pérdida auditiva en una persona adulta



Fuente: dizziness-and-balance.com

2.8. QUE ES LAB VIEW

Lab VIEW (*Laboratory Virtual Instrument Engineering Workbench*) es un software de lenguaje de programación gráfico para el diseño de instrumentación, control, análisis y adquisición de datos; este es muy poderoso y flexible, y fue creado por la compañía *National Instrumens*, la cual diseña productos tanto de hardware como de software, los cuales ayudan a ingenieros y científicos para realizar mediciones, procesos de control, análisis y guardar datos.

La compañía *National Instruments* fue fundada hace más de 25 años en Austin, Texas por Jeffrey Kodosky, William Nowlin y James Truchard, también conocido como Dr. T. En esa época los tres estaban trabajando en aplicaciones de sonar para la marina de Estados Unidos (*US Navy*) dentro de los laboratorios llamados "*Applied Research Laboratories*". Mientras buscaban una manera de conectar el equipo de prueba al DEC PDP-11, Dr. T decidió realizar una interface. Dr. T reclutó a Jeff y Bill para que lo ayudara, y juntos realizaron exitosamente Lab VIEW y la noción de un "instrumento virtual".

Lab VIEW permite diseñar interfaces mediante una consola interactiva gráfica, con este programa software se puede diseñar un diagrama de bloques. Este programa es a la vez compatible con otras herramientas de programación similares como por ejemplo Matlab. Lab VIEW también permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos, incluyendo imágenes.

Ingenieros y científicos en producción, investigación, desarrollo, industrias de servicio, industria automotriz, electrónica, química, telecomunicaciones, farmacéuticos, entre otros han usado Lab VIEW en sus trabajos. Este software es muy usado en áreas de medición, pruebas, automatización, y análisis de datos. Por ejemplo, los científicos del laboratorio de "*Jet Propulsion*" en la NASA usan Lab VIEW para analizar datos del *Mars Pathfinder Sojourner rover*, incluyendo la posición y temperatura del *rover*, cuanta potencia queda en las baterías del *rover*, y para monitorear el estado del *rover*.

Los programas dentro de Lab VIEW se llaman *virtual instruments* o Vis. Lab VIEW es diferente a otro tipos de lenguajes de programación como Fortran y C, debido a que Lab VIEW usa una interface gráfica conocida como *G programing*, para crear programas con símbolos gráficos que describen las acciones del programa. Lab View usa una terminología familiar para ingenieros y científicos, y los íconos usados para construir los programas son fácilmente identificados por inspección visual. Debido a estas características el programa es muy fácil de usar inclusive si no se tiene experiencia en programación.

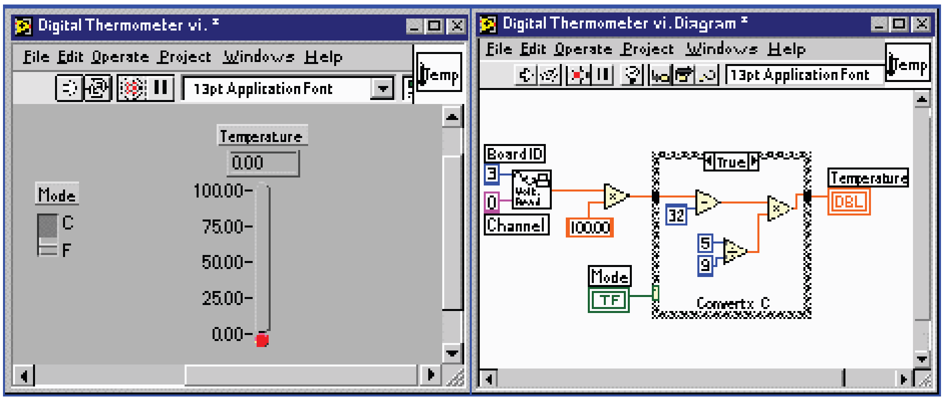
Lab VIEW contiene una librería extensa de instrumentos virtuales y funciones para facilitar la programación. También contiene librerías de aplicaciones específicas para adquisición de datos, *input y outputs* de archivos, control de instrumentación serial, y análisis de datos. Incluye herramientas para detección de errores convencionales de programación las cuales crean puntos de análisis paso a paso en el programa que incluso pueden ser animados durante la ejecución del programa para observar cómo fluyen los datos dentro de este.

Guía rápida de Lab VIEW

Una vez que Lab VIEW este instalado en la computadora, al hacer click en el ícono del programa se abrirán dos pantallas como se muestra en la figura 2.25, la pantalla de la izquierda es el panel frontal y la de la derecha es el panel de programación o también conocido como diagrama de bloques. El panel frontal se utiliza para diseñar la interfaz con el usuario y se ubican los controles e indicadores es decir esta pantalla es la que el usuario final va a operar, mientras que en el panel de programación o diagrama de bloques se encontraran los elementos que fueron utilizados para realizar dicho programa, es decir aquí esta toda la parte de la programación.

Cada uno de los elementos que se utiliza en la interfaz tendrá una representación gráfica en el panel de programación, y además todos los controles e indicadores que sean necesarios para adquirir o entregar información al usuario final serán representados en el panel de programación y en el panel frontal. Los controles pueden ser de tipo booleanos (tipo de dato que establece uno de dos estados, ejemplo verdadero o falso, prendido o apagado, cero o uno), numéricos, *strings* (oraciones o palabras), un arreglo matricial o una combinación de estos. Mientras que los indicadores además de los tipos anteriores se pueden incluir tablas, gráficas 2D o 3D, entre otros.

Figura 2.25 pantallas principales


Digital Thermometer vi. *
File Edit Operate Project Windows Help
13pt Application Font
Temp
Temperature
0.00
100.00
75.00
50.00
25.00
0.00
Mode
C
F
Digital Thermometer vi.Diagram *
File Edit Operate Project Windows Help
13pt Application Font
Temp
Board ID
3
0
Channel
Volt Read
100.00
True
32
5
9
Mode
TF
Convert to C
Temperature
DBL

Fuente: Franklin Robalino

Al iniciar un nuevo programa estas pantallas estarán vacías, y en el panel de programación se podrá comenzar a programar el diseño de una manera gráfica como si fuese un diagrama de bloques. Esta programación gráfica facilita la interface con el usuario debido a que se basa en operaciones representadas por íconos (Vis) que son fácilmente reconocidos por el usuario. Mediante estos íconos y con unas simples conexiones para enlazarlos se determina la operación o funcionamiento del ícono.

Cuando se diseña un programa en Lab VIEW de forma gráfica, el programador puede visualizar el flujo de datos dentro del programa y para facilitarlo más, cada conexión tiene un color diferente dependiendo del tipo de datos que circule por la conexión, por ejemplo los datos de tipo booleano se caracterizarán por tener una conexión de color verde, y así cada tipo de datos tendrá un color diferente para ayuda visual dentro de Lab VIEW.

Es necesario entender que por lo general al hacer una conexión de un ícono a otro, esta conexión va a tener un tipo de dato específico y por lo tanto la salida de un ícono tiene que tener el mismo tipo de datos que la entrada del otro ícono al cual se va a enlazar, aunque esto no es cierto en todos los casos como por ejemplo en un "*cluster*" que sí permite distintos tipos de variables, sin embargo para la mayoría de los Vis esto sí se cumple, como se podrá ver más adelante.

Otra cosa que hay que tener en cuenta al programar es que el flujo de los datos va de izquierda a derecha dentro del panel de programación. Esto es fácil de observar en cada parte del programa al realizar una ejecución del programa paso a paso (ícono).

Los íconos (Vis) pueden ser tanto prediseñados por el fabricante o también pueden ser creados por el programador (usuario del programa) y estos pueden ser reutilizados dentro de cualquier otro programa. Los Vis constan tanto de entradas y salidas al igual que cualquier lenguaje de programación estándar y pueden estar conformados de otros sub Vis y así sucesivamente representando un árbol genealógico donde un VI se relaciona o depende de otro o varios sub Vis.

El programa Lab VIEW contiene varios Vis tanto de adquisición de datos e imágenes, de comunicaciones, de procesamiento digital de señales, de funciones matemáticas simples, hasta funciones que utilizan otros programas como Matlab, también permite funciones más complejas como “nodos de fórmula” que son utilizados para la edición directa de ecuaciones como se hace en lenguajes de programación tradicionales.

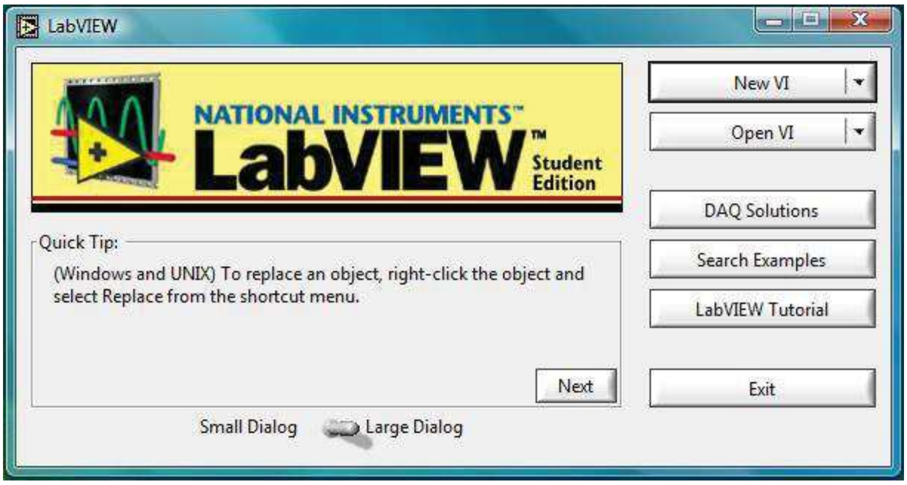
CAPÍTULO 3

PROGRAMANDO EN LABVIEW

3.1. LA PANTALLA PRINCIPAL

Para abrir el programa Lab VIEW, se hace doble click en el ícono y se abrirá la pantalla principal (ver la figura3.1).

Figura 3.1.- Pantalla principal de LabVIEW


LabVIEW
NATIONAL INSTRUMENTS™
LabVIEW™ Student Edition
New VI
Open VI
DAQ Solutions
Search Examples
LabVIEW Tutorial
Exit
Next
Small Dialog
Large Dialog

Quick Tip:
(Windows and UNIX) To replace an object, right-click the object and select Replace from the shortcut menu.

Fuente: Franklin Robalino

Al comienzo de cada sesión de Lab VIEW, la pantalla principal contendrá un cuadro pequeño en la parte inferior izquierda, el cual se puede seleccionar si es que se desea comenzar el programa directamente sin que aparezca la pantalla principal. Si se escoge esta opción, Lab VIEW siempre abrirá un nuevo VI llamado "*Untitled1.vi*" en vez de abrir la pantalla principal.

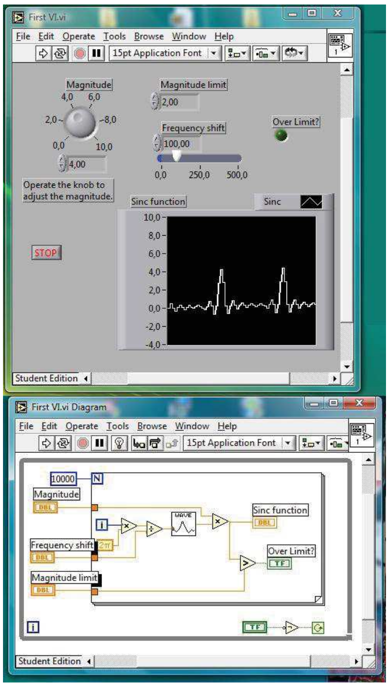
La pantalla principal incluirá un nuevo "*Quick Tip*" cada vez que se abra una sesión. Se puede ver más *tips* al oprimir el botón "*Next*". Una vez que la sesión de Lab VIEW comience, es decir después de haber escogido una opción en la pantalla principal, un *switch* aparecerá en la parte inferior de la pantalla principal que permitirá escoger ver una versión grande de la pantalla con más opciones o una versión pequeña con las opciones básicas.

3.2. VENTANAS DE PANEL Y DE DIAGRAMA

Al seleccionar “New VI” en la pantalla principal, un panel frontal (*front panel*) aparecerá. Este panel frontal es la interface al código VI y es una de las dos ventanas de Lab VIEW que conforman un instrumento virtual. La otra ventana es conocida como la ventana de diagramas de bloque, la cual contiene códigos de programación que existen en forma gráfica, como íconos, cables, etc.

Tanto el panel frontal como el diagrama de bloques consisten de objetos gráficos que son los elementos de programación G (*G programing*). El panel frontal contiene varios tipos de control e indicadores, es decir inputs y outputs respectivamente. Los diagramas de bloque contienen terminales correspondientes a los controles e indicadores del panel frontal, y también constantes, funciones, sub Vis, estructuras, y cables que transportan datos desde un objeto a otro. Las estructuras son elementos de control del programa, como por ejemplo el *While Loop*, el cual será explicado más adelante.

Figura 3.2.- muestra un panel frontal con su respectivo diagrama de bloques.

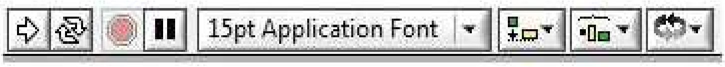

First VI.vi
File Edit Operate Tools Browse Window Help
15pt Application Font
Magnitude
4,00
Magnitude limit
2,00
Frequency shift
100,00
Over Limit?
Operate the knob to adjust the magnitude.
STOP
Sinc function
Sinc
Student Edition
First VI.vi Diagram
10000
N
Magnitude
Frequency shift
Magnitude limit
Sinc function
Over Limit?
i
TF

Fuente: Franklin Robalino

3.2.1. BARRA DE HERRAMIENTAS DEL PANEL FRONTAL

Una barra de herramientas con botones de comando e indicadores de estado que son usados para controlar a los instrumentos virtuales, está ubicada tanto en la ventana del panel frontal como en la ventana del diagrama de bloques. La barra de herramientas del panel frontal y la barra de herramientas del diagrama de bloques son diferentes, a pesar que ambas contienen algunos de los mismos botones e indicadores. La barra de herramientas que aparece en la parte superior del panel frontal se muestra en la figura 3.3.

Figura 3.3.- Barra de herramientas del panel frontal.



Fuente: Franklin Robalino

Mientras el VI está siendo ejecutado, el botón **“*Abort Execution*”** aparece, a pesar que al presionar el botón de aborto causa que se termine la ejecución del VI, como regla general hay que evadir terminar la ejecución del programa de esta manera y dejar que el VI se ejecute hasta el final o incorporar por medio de programación un control de ejecución, es decir un *switch* o botón de prendido y apagado para terminar la ejecución del VI en el panel frontal.

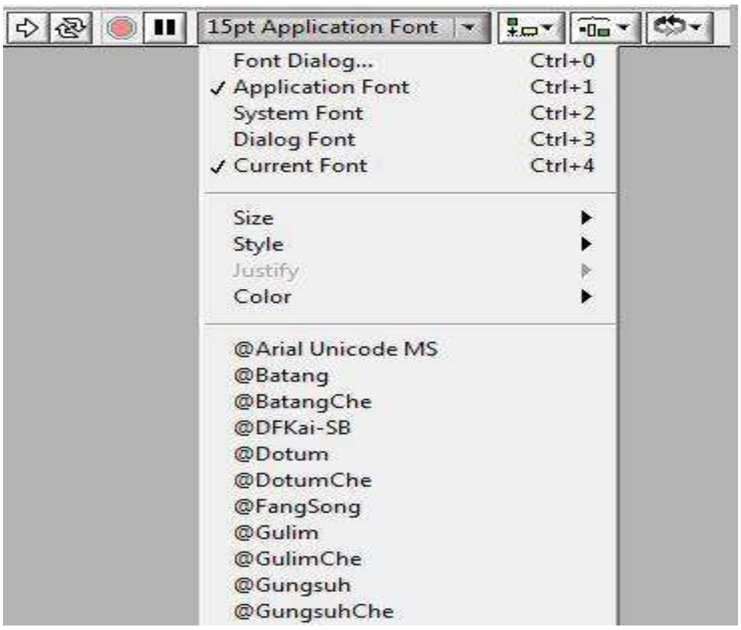
El botón **“*Broken Run*”** reemplaza al botón “*Run*” cuando el VI no puede ejecutarse debido a errores de código. Si se llegase a encontrar un problema al ejecutar el VI, sólo hay que hacer click en el botón “*Broken Run*”, y una ventana aparecerá automáticamente en la pantalla, la cual listará todos los errores detectados dentro del programa. Y después al hacer doble click en uno de los errores dentro de la lista, el programa se dirigirá automáticamente al lugar específico del código en el diagrama de bloques en donde está el error.

Al hacer click en el botón **“*Continuous Run*”** provoca que el VI se ejecute de manera continua. Al oprimir nuevamente este botón deshabilita el modo de ejecución continua del VI, es decir el VI se detendrá cuando se complete la ejecución.

El botón **“*Pause/Continue*”** detiene temporalmente la ejecución del VI. Para continuar con la ejecución del programa después, sólo hay que aplastar el botón nuevamente, y el VI resumirá su ejecución.

El “*Font Ring*” sirve para seleccionar distintos tipos de porte, color, y estilos de Font, para los textos que se ingresen al programa, como se muestra en la figura3.4.

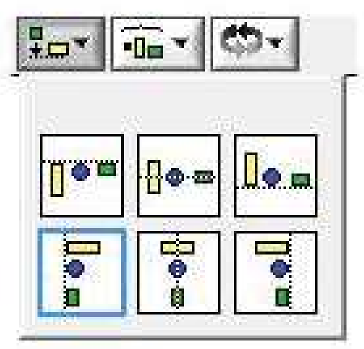
Figura 3.4.- Font Ring en el panel frontal


15pt Application Font
Font Dialog... Ctrl+0
✓ Application Font Ctrl+1
System Font Ctrl+2
Dialog Font Ctrl+3
✓ Current Font Ctrl+4
Size
Style
Justify
Color
@Arial Unicode MS
@Batang
@BatangChe
@DFKai-SB
@Dotum
@DotumChe
@FangSong
@Gulim
@GulimChe
@Gungsuh
@GungsuhChe

Fuente: Franklin Robalino

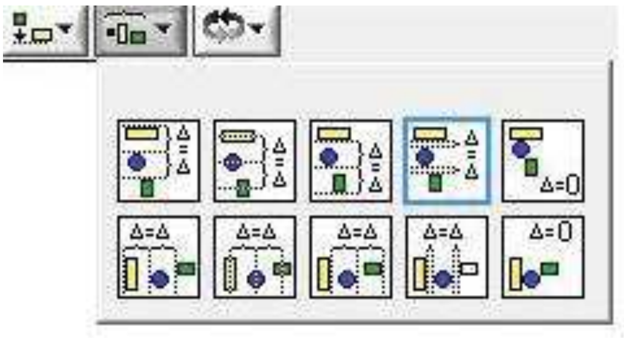
El “*Alignment ring*” configura el alineamiento preferido de varios objetos ya sea que se encuentren en el panel frontal o en el diagrama de bloques. Una vez seleccionados los objetos que van a ser alineados, se puede elegir el alineamiento preferido para dos o más objetos. Se puede elegir alinear los objetos por su filo izquierdo, filo derecho, centro horizontal, centro vertical, filo inferior o filo superior. Las opciones están ilustradas en el menú “*Alignment ring*”, como se muestra en la figura 3.5. Alinear objetos es muy útil para organizar los objetos tanto en el panel frontal como en el de diagrama de bloques.

Figura 3.5.- Alignment ring.



Fuente: Franklin Robalino

Figura 3.6.- Distribution ring.

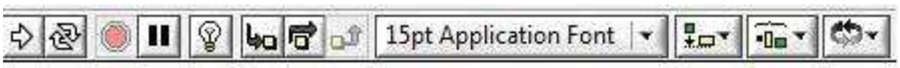


Fuente: Franklin Robalino

El “*Distribution ring*” configura la opción preferida de distribución de dos o más objetos, esto se muestra en la figura 3.6.

3.2.2. BARRA DE HERRAMIENTAS DEL DIAGRAMA DE BLOQUES

Figura 3.7.- Barra de herramientas del diagrama de bloques.



Fuente: Franklin Robalino

La barra de herramientas del diagrama de bloques (ver figura 3.7) contiene muchos de los botones que están dentro de la barra de herramientas del panel frontal. Cuatro botones adicionales para detectar errores son encontrados en la barra de herramientas del diagrama de bloques:

Al activar el botón **“*Highlight Execution*”** se habilita el resaltador de ejecución. Esto causará que mientras el programa se ejecute se resalte los datos, permitiendo al programador observar el flujo de datos dentro del código en el diagrama de bloques. Esto sirve de gran ayuda para detectar errores y verificación de ejecución apropiada. En el modo de ejecución **“*Highlight Execution*”**, el botón cambiará a un foco prendido.

Las capacidades de Lab VIEW para detectar errores permiten al programador ir paso a paso por el VI revisando nodo por nodo. Por el momento un nodo puede ser relacionado con un elemento de ejecución (una sección del código), como un *While Loop* o un sub VI. Una mejor definición de los diferentes tipos de nodos se dará más adelante.

El botón **“*Step Over*”** se salta “sobre” un nodo. Permitiendo que se ejecute el nodo sin hacer un proceso paso a paso dentro del nodo.

El botón **“*Step Into*”** permite ingresar a un nodo. Una vez que se ingrese al nodo se podrá realizar el paso a paso dentro del nodo.

El botón **“*Step Out*”** permite salir de un nodo. Una vez que se sale de un nodo se completa el proceso de paso a paso dentro de un nodo y se continua al siguiente nodo.

El indicador de **“*Warning*”** aparece sólo cuando existe un error potencial dentro del diagrama de bloques. Al aparecer el ícono de “*warning*”, esto no impide que se ejecute el VI. Se puede habilitar el indicador de “*warning*” dentro de “Tools>>Options>>debugging menú”.

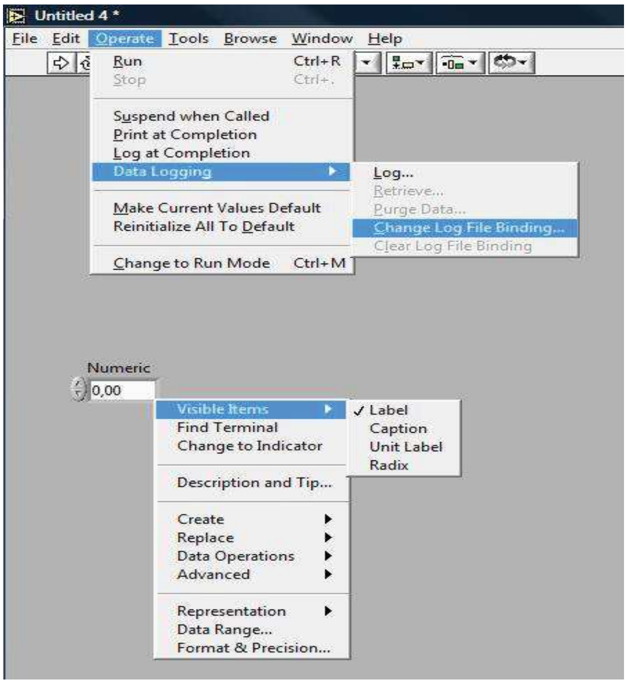
3.3. MENÚS DE ATAJOS (*SHORT CUTS*)

El programa Lab VIEW tiene dos tipos de menús: los menús “*pull down*” y los menús “*short cut*” (menú de atajos).

Para ingresar a un menú “*short cut*”, se debe posicionar el cursor en el objeto deseado ya sea en el panel frontal o en el diagrama de bloques, y hacer un click derecho (estas instrucciones son para PC). En la mayoría de casos aparecerá un menú de atajos, esto es debido a que no todos los objetos en Lab VIEW poseen menús de atajos con opciones y comandos. Este proceso es llamado “*poping up*” y por esta razón los menús de atajo también son llamados menús “*pop up*”. También se puede hacer un “*pop up*” en cualquier espacio vacío del panel frontal o del diagrama de bloques, lo cual dará acceso a las paletas de control y funciones (*Control Palette, Function Palette*), entre otras paletas importantes. Las opciones que aparezcan dentro de los menús de atajos dependerán del objeto seleccionado. Al hacer un “*pop up*” en un control numérico abrirá un menú de atajos distinto al que se abrirá haciendo un “*pop up*” en un *While Loop*.

Muchos menús tanto de "*short cut*" y "*pull down*" contienen sub menús, como se muestra en la figura 3.8. Los ítems de menús que se expanden en submenús se llaman menús jerárquicos (*hierarchical*) y se los identifica por que tienen una flecha hacia la derecha en el menú. Los menús jerárquicos se presentaran con diferentes tipos de opciones y comandos. Hay que tener en cuenta que al hacer un "*pop up*" en diferentes áreas de un mismo objeto puede producir que aparezca un menú de atajos distinto.

Figura 3.8.- Menús de "short cut" y "pull down".


Untitled 4 *
File
Edit
Operate
Tools
Browse
Window
Help
Run Ctrl+R
Stop Ctrl+.
Suspend when Called
Print at Completion
Log at Completion
Data Logging
Make Current Values Default
Reinitialize All To Default
Change to Run Mode Ctrl+M
Log...
Retrieve...
Purge Data...
Change Log File Binding...
Clear Log File Binding
Numeric
0,00
Visible Items
Find Terminal
Change to Indicator
Description and Tip...
Create
Replace
Data Operations
Advanced
Representation
Data Range...
Format & Precision...
✓ Label
Caption
Unit Label
Radix

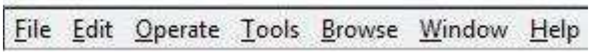
Fuente: Franklin Robalino

Otros tipos de ítems en los menús abren cuadros de diálogo que contienen opciones para modificar y configurar los elementos del programa. Los ítems de los menús que llevan a cuadros de diálogo se los identifican porque tienen tres puntos suspensivos a la derecha del menú (…). Ítems de menú que no tengan ni puntos suspensivos ni flechas al lado derecho son generalmente comandos que se ejecutan inmediatamente una vez seleccionados.

MENUS DE “*PULL DOWN*”

La barra de menú en la parte superior de la pantalla de Lab VIEW (ver figura 3.9), contiene los menús “*pull down*” importantes: *File, Edit, Operate, Tools, Browse, Window,* y *Help*.

Figura 3.9.- Menús “pull down”

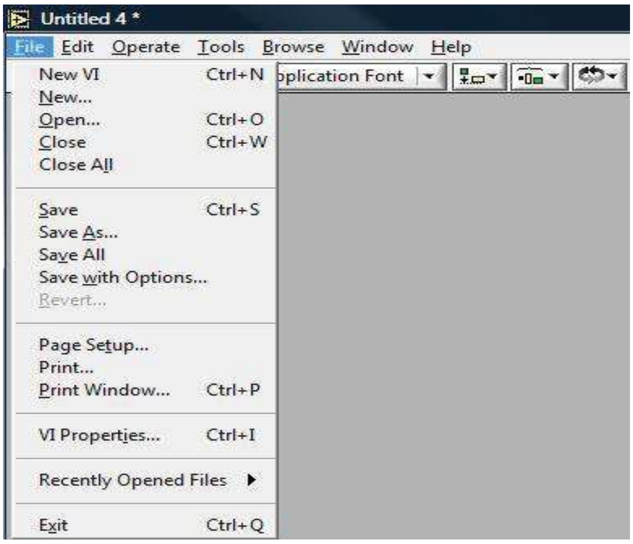

File Edit Operate Tools Browse Window Help

Fuente: Franklin Robalino

Menú “*File*”

El menú *File* (archivo), que se muestra en la figura 3.10, contiene comandos asociados con la manipulación de archivos. Se puede crear nuevos Vis o abrir Vis ya existentes desde el menú “*File*”. Las opciones del menú “*File*” son principalmente para abrir, cerrar, grabar, imprimir, etc. Como se puede observar, en cada menú “*pull down*” los comandos y opciones tienen sus respectivos atajos escritos a su lado derecho. Estos atajos son secuencias del teclado que pueden ser usadas para escoger la opción deseada sin necesidad de entrar al menú.

Figura 3.10.- Menú “File”

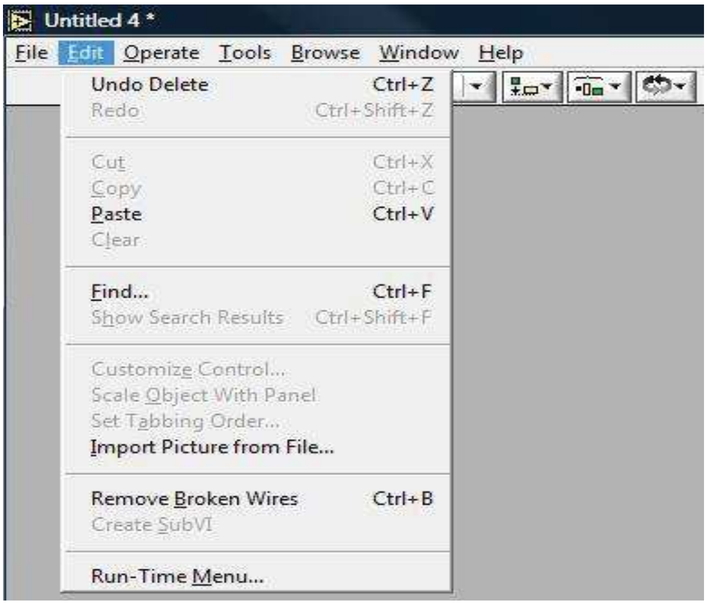

Untitled 4 *
File Edit Operate Tools Browse Window Help
New VI Ctrl+N
New...
Open... Ctrl+O
Close Ctrl+W
Close All
Save Ctrl+S
Save As...
Save All
Save with Options...
Revert...
Page Setup...
Print...
Print Window... Ctrl+P
VI Properties... Ctrl+I
Recently Opened Files
Exit Ctrl+Q
pplication Font

Fuente: Franklin Robalino

Menú “*Edit*”

El menú “*Edit*” (editar), que se muestra en la figura 3.11, es usado para modificar los objetos de un VI tanto en el panel frontal como en el diagrama de bloques. Las opciones “*Undo*” y “*Redo*” (deshacer y rehacer) son muy útiles para editar debido a que permiten deshacer una acción luego de ser ejecutada, y una vez desecha, la acción se puede rehacer. Por default, el número máximo de “*undo*” por VI es de 8, pero se puede cambiar el número máximo si se desea.

Figura 3.11.- Menú “Edit”

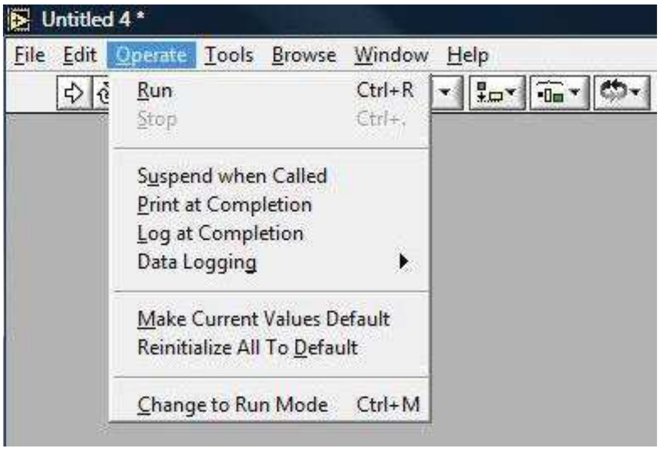

Untitled 4 *
File
Edit
Operate
Tools
Browse
Window
Help
Undo Delete Ctrl+Z
Redo Ctrl+Shift+Z
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Clear
Find... Ctrl+F
Show Search Results Ctrl+Shift+F
Customize Control...
Scale Object With Panel
Set Tabbing Order...
Import Picture from File...
Remove Broken Wires Ctrl+B
Create SubVI
Run-Time Menu...

Fuente: Franklin Robalino

Menú “*Operate*”

El menú “*Operate*” (operar), que se muestra en la figura 3.12, es usado para correr o parar la ejecución de un VI, cambiar los valores default del VI, cambiar entre el modo de ejecución y el modo de edición, etc.

Figura 3.12.- Menú "Operate"

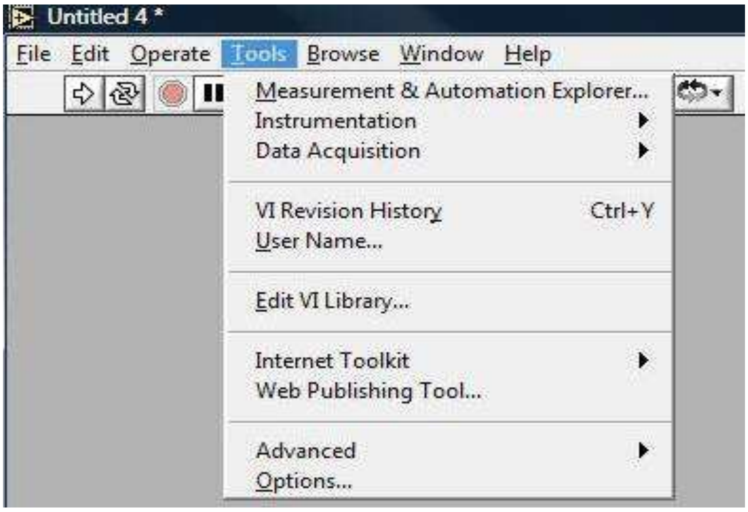

Untitled 4 *
File Edit Operate Tools Browse Window Help
Run Ctrl+R
Stop Ctrl+.
Suspend when Called
Print at Completion
Log at Completion
Data Logging
Make Current Values Default
Reinitialize All To Default
Change to Run Mode Ctrl+M

Fuente: Franklin Robalino

Menú "*Tools*"

El menú "*Tools*" (herramientas), que se muestra en la figura 3.13, es usado para la comunicación entre instrumentos y circuitos de adquisición de datos, comparación de VIs, construcción de aplicaciones, habilitar el servidor web, etc.

Figura 3.13.- Menú "Tools"


Untitled 4 *
File Edit Operate Tools Browse Window Help
Measurement & Automation Explorer...
Instrumentation
Data Acquisition
VI Revision History Ctrl+Y
User Name...
Edit VI Library...
Internet Toolkit
Web Publishing Tool...
Advanced
Options...

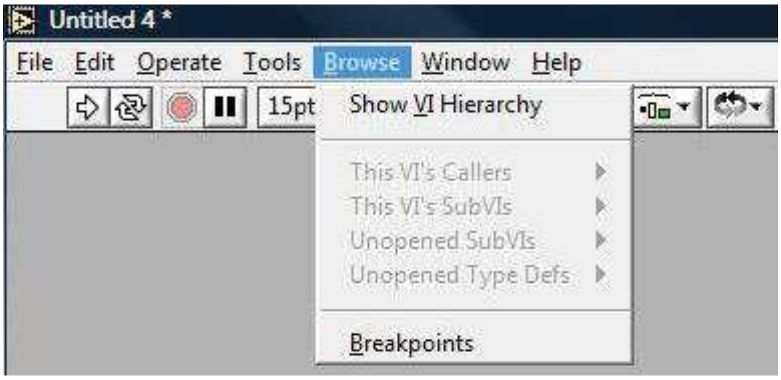
Fuente: Franklin Robalino

Un link importante en este menú es el de la página web de "*National Instruments*", en donde se puede obtener información acerca de los productos de la compañía. Si el programa Lab VIEW es utilizado para controlar instrumentos externos se debe instalar el "*Instrument Driver Network*" que se encuentra en la página web.

Menú "*Browse*"

El menú "*Browse*" (buscar), que se muestra en la figura 3.14, es usado para navegar a través de la jerarquía del instrumento virtual.

Figura 3.14.- Menú "Browse"


Untitled 4 *
File
Edit
Operate
Tools
Browse
Window
Help
15pt
Show VI Hierarchy
This VI's Callers
This VI's SubVIs
Unopened SubVIs
Unopened Type Defs
Breakpoints

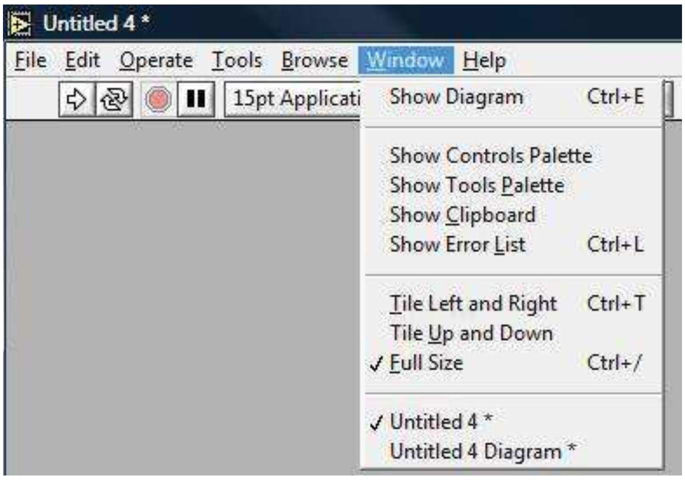
Fuente: Franklin Robalino

Menú "*Windows*"

El menú "*Windows*" (ventanas), que se muestra en la figura 3.15, es usado para una variedad de actividades, sirve para saltar entre el panel frontal y el diagrama de bloques, ver ambas pantallas (el panel frontal y el diagrama de bloques) simultáneamente, ya sea dividiendo la pantalla de forma vertical o horizontal.

Todos los instrumentos virtuales abiertos están listados dentro de este menú en la parte inferior, así mismo se puede saltar entre los Vis abiertos. Este menú también permite mostrar las paletas de trabajo en la pantalla (*Tools Palette* y *Functions Palette*).

Figura 3.15.- Menú “Windows”

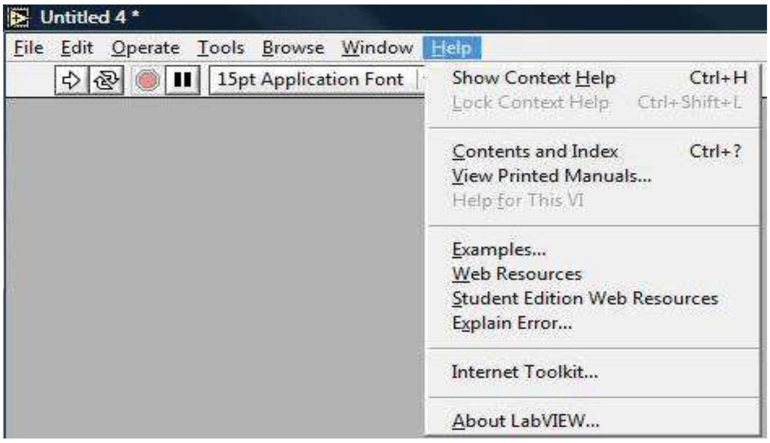

Untitled 4 *
File Edit Operate Tools Browse Window Help
15pt Applicati
Show Diagram Ctrl+E
Show Controls Palette
Show Tools Palette
Show Clipboard
Show Error List Ctrl+L
Tile Left and Right Ctrl+T
Tile Up and Down
✓ Full Size Ctrl+/
✓ Untitled 4 *
Untitled 4 Diagram *

Fuente: Franklin Robalino

Menú “*Help*”

El menú “*Help*” (ayuda), que se muestra en la figura 3.16, es usado para ingresar por medio del internet a la ayuda de Lab VIEW. Se puede ver información de los objetos en el panel frontal o en el diagrama de bloques, activar las referencias de la web, ver que versión de Lab VIEW está instalada, ver la memoria siendo usada en la computadora, etc.

Figura 3.16.- Menú “Help”


Untitled 4 *
File Edit Operate Tools Browse Window Help
15pt Application Font
Show Context Help Ctrl+H
Lock Context Help Ctrl+Shift+L
Contents and Index Ctrl+?
View Printed Manuals...
Help for This VI
Examples...
Web Resources
Student Edition Web Resources
Explain Error...
Internet Toolkit...
About LabVIEW...

Fuente: Franklin Robalino

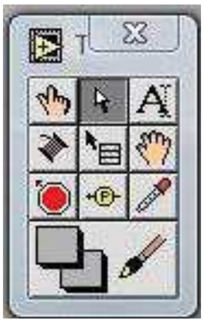
3.4. PALETAS

Las paletas (*palettes*) son paneles gráficos que contienen varias herramientas y objetos que son usados para crear y operar instrumentos virtuales. Las paletas se pueden mover a cualquier parte del escritorio que se desee, por lo general se las ubica a un lado de la pantalla para que no estorben la visualización de objetos dentro del panel frontal o del diagrama de bloques. Las tres paletas principales son la paleta de herramientas (*Tools Palette*), la paleta de controles (*Controls Palette*) y la paleta de funciones (*Functions Palette*).

La Paleta de Herramientas

Una herramienta es un modo especial de operación del cursor (mouse). Las herramientas son usadas para realizar funciones específicas de edición, al igual que se utilizan en un programa de dibujo estándar. Se puede crear, modificar, y eliminar errores de los VIs usando las herramientas que se encuentran en la paleta de herramientas (*Tools Palette*), que se muestra en la figura 3.17. Si por alguna razón la paleta de herramientas no está visible en la pantalla, se debe entrar al menú (*pull down*) "*Windows*" y escoger la opción "*Show Tool Palette*". Una vez seleccionada una de las herramientas de la paleta de herramientas, el cursor del mouse cambiará de figura (según la herramienta escogida).

Figura 3.17.- Paleta de Herramientas



Fuente: Franklin Robalino

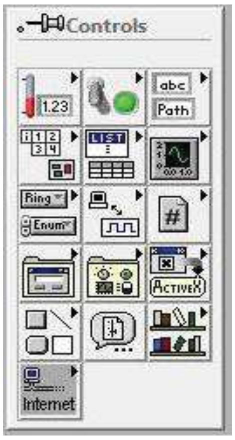
Otra manera de ingresar a la ayuda en línea es al ubicar cualquier herramienta de la paleta de herramientas sobre el objeto de interés dentro de la pantalla de diagrama de bloques. Si es que la ayuda en línea existe para ese objeto, la información aparecerá dentro de una pantalla de ayuda. Para realizar este proceso se requiere que la opción "*Show Context Help*" del menú "*Help*" esté seleccionada.

Un atajo para tener acceso a la paleta de herramientas es presionando "*Shift*" más el botón derecho del mouse, ya sea en el panel frontal o en el diagrama de bloques.

La Paleta de Controles

La paleta de controles (*Controls Palette*), que se muestra en la figura 3.18, consiste en íconos de niveles superiores que representan sub paletas, las cuales contienen un rango mayor de objetos que pueden ser usados en la creación del panel frontal.

Figura 3.18.- Paleta de Controles



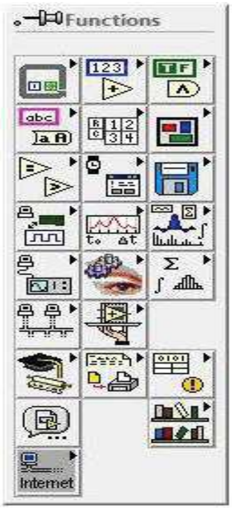
Fuente: Franklin Robalino

Para ingresar a las sub paletas se debe hacer click en el ícono de nivel superior deseado. Un ejemplo de ícono de nivel superior es el ícono "*Numeric*" (ver figura 3.18), al ingresar en la sub paleta de "*Numeric*" se mostrarán varios controles (*inputs*) numéricos e indicadores (*outputs*) que serán utilizados en el panel frontal como una manera de mover datos desde y hacia el código de programación (diagrama de bloques).

La Paleta de Funciones

La paleta de funciones (*Functions Palette*), que se muestra en la figura 3.19, trabaja de igual manera que la paleta de controles. Es decir consiste de íconos de nivel superior, los cuales representan sub paletas, que contienen un rango mayor de objetos que pueden ser usados en la creación del diagrama de bloques.

Figura 3.19.- Paleta de Funciones



Fuente: Franklin Robalino

Para ingresar a las sub paletas se debe hacer click en el ícono de nivel superior deseado. Muchos de los elementos de programación gráfica (*G programing*) se encuentran dentro de la paleta de funciones. Por ejemplo la sub paleta “*Structures*” (estructuras), (ver figura 3.19), contiene estructuras “*For Loops*”, “*While Loops*”, y “*Formula Nodes*”, los cuales son elementos típicos de un instrumento virtual.

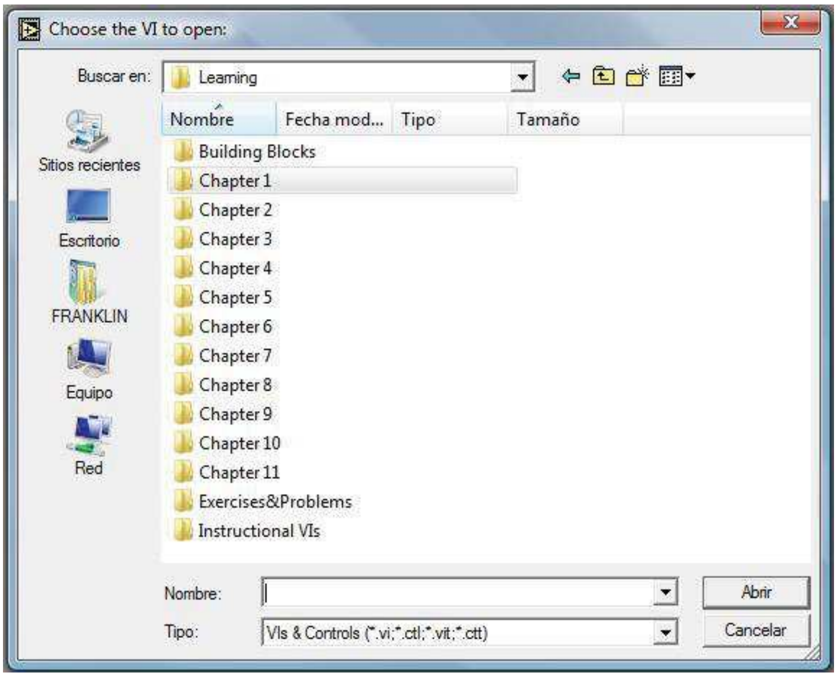
Si por alguna razón la paleta de funciones no está visible en la pantalla, se debe entrar al menú (*pull down*) “*Windows*” y escoger la opción “*Show Tool*

Palette”. También se puede entrar a la paleta de funciones haciendo un “*pop up*” en cualquier ares vacía del diagrama de bloques. Hay que tener en cuenta que la paleta de funciones solo podrá aparecer cuando la ventana de diagrama de bloques esté activa.

3.5. CARGANDO Y GRABANDO VIS

Para carga un instrumento virtual a la memoria, se abre el menú “*File*” y se selecciona la opción “*Open*”. Al escoger esta opción un cuadro de diálogo se abrirá, como se muestra en la figura 3.20. Dentro de este cuadro de diálogo, aparecen los directorios de los VIs y las librerías de los VIs. Las librerías de los VIs contienen múltiples VIs en un formato compreso.

Figura 3.20.- Cuadro de dialogo para abrir VIs



Buscar en: Learning

Tipo: VIs & Controls (*.vi;*.ctl;*.vit;*.ctt)

Choose the VI to open:
Nombre
Fecha mod...
Tipo
Tamaño
Sitios recientes
Escritorio
FRANKLIN
Equipo
Red
Building Blocks
Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Chapter 7
Chapter 8
Chapter 9
Chapter 10
Chapter 11
Exercises&Problems
Instructional VIs
Nombre:
Abrir
Cancelar

Fuente: Franklin Robalino

Para abrir una librería de VI o un directorio de VI se debe hacer click en el ícono y luego en “*Open*”, o también se puede hacer un doble click en el ícono. En el cuadro de diálogo se abren las librerías de los VIs como si fuesen directorios. Una vez que el directorio o librería se abre, se puede localizar el VI deseado y cargarlo en la memoria haciendo click en el ícono y luego en "*Ok*" o haciendo doble click en el ícono.

Mientras se carga el VI, aparecerá el cuadro de diálogo de estado (*status dialog box*). El sector de "*Loading*" lista los sub VIs que están siendo cargados en la memoria, y en la parte superior derecha donde dice xx (número) "*Loaded*", aparece el número de sub VIs que ya han sido cargados dentro de la memoria. Se puede cancelar la acción en cualquier momento al hacer click en el botón "*Stop…*".

Si por alguna razón Lab VIEW no puede localizar el sub VI inmediatamente de algún VI, comenzara a buscarlo dentro de todos los directorios que están especificados dentro de "*VI Search Path*" en "Tools>>Options>>Paths". Estos directorios o VIs se mostraran en el área de la ventana que dice "*Searching*" en el cuadro de diálogo de estado. Se puede también hacer que Lab VIEW ignore la búsqueda de dichos sub VIs, o se puede hacer click en el botón "*Browse*" para buscar los sub VIs que falten automáticamente con el cuadro de diálogo de estado.

Se puede gravar el VI en un directorio regular o en una librería de VIs al seleccionar "*Save, Save As…, Save All…, o Save with Options…*" desde el menú "*File*". También se pueden transferir los VIs desde una plataforma de trabajo a otra, por ejemplo desde Macintosh hacia Windows. Lab VIEW traduce automáticamente y recopila los VIs en la nueva plataforma. Debido a que los VIs son archivos, se puede usar cualquier tipo de método para transferir la información de los VIs entre las plataformas.

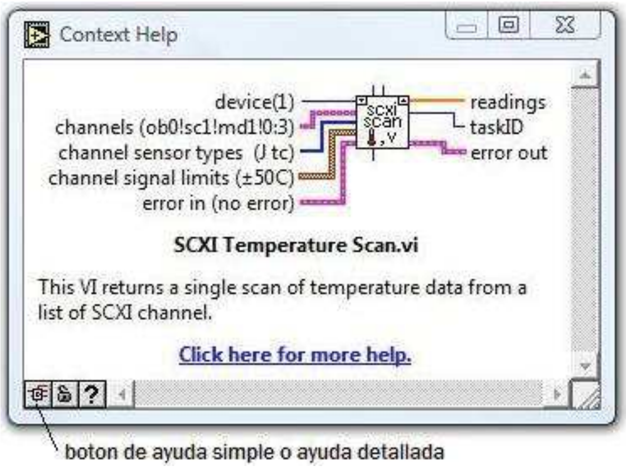
3.6. OPCIONES DE AYUDA EN LAB VIEW

Las dos opciones de ayuda más comunes usadas para aprender programación dentro de Lab VIEW son el "*Show Context Help*" y el "*Contents and Index*". Estas dos opciones de ayuda pueden obtenerse mediante el menú "*HELP*".

Ventana de “*Context Help*”

Para mostrar la ventana de ayuda, se escoge “*Show Context Help*” del menú “*Help*”. Si es que ya existen objetos en el panel frontal o diagrama de bloques, se puede averiguar más de esos objetos al poner alguna de las herramientas de la paleta de herramientas encima del objeto del panel frontal o del diagrama de bloques de el cual se requiera más información. Este proceso causa que aparezca la ventana de “*Context Help*”, la cual mostrará el ícono que está asociado con el objeto seleccionado y mostrara el tipo de conexión que requiere cada terminal. Hay que tener en cuenta que algunos terminales en ciertos íconos tienen que necesariamente estar conectados mientras otros terminales son opcionales. Para poder ubicar los terminales que requieren conexión, en la ventana de ayuda dichos terminales estarán puestos en negrillas, para poder ubicarlos con facilidad, mientras que los terminales recomendados estarán en texto regular (sin negrilla) y los terminales opcionales estarán de color plomo, como se muestran en la figura 3.21.

Figura 3.21.- Ventana de “Context Help”


Context Help
device(1)
channels (ob0!sc1!md1!0:3)
channel sensor types (J tc)
channel signal limits (±50C)
error in (no error)
SCXI scan
readings
taskID
error out

SCXI Temperature Scan.vi

This VI returns a single scan of temperature data from a list of SCXI channel.

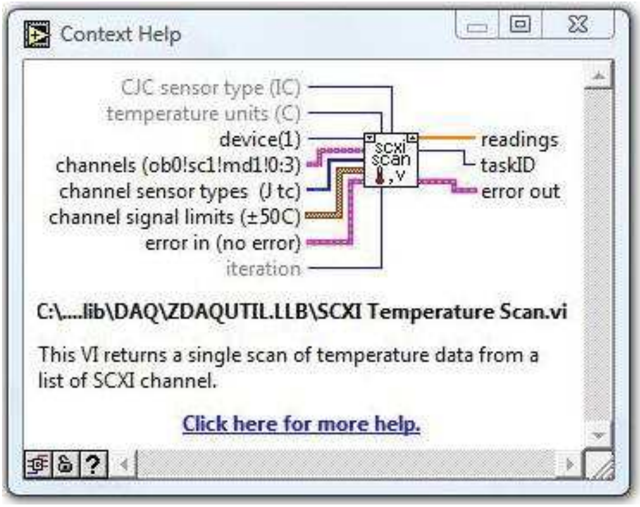
Click here for more help.

boton de ayuda simple o ayuda detallada

Fuente: Franklin Robalino

En la parte inferior izquierda de la ventana de ayuda hay un botón para escoger entre los modos de ayuda simple y de ayuda detallada. La ayuda simple enfatiza las conexiones importantes de los terminales, mientras que la ayuda detallada muestra todos los terminales, como se muestra en la figura 3.22.

Figura 3.22.- Ventana de “Context Help” con ayuda detallada


Context Help
CJC sensor type (IC)
temperature units (C)
device(1)
channels (ob0!sc1!md1!0:3)
channel sensor types (J tc)
channel signal limits (±50C)
error in (no error)
iteration
SCXI scan
readings
taskID
error out

C:\....lib\DAQ\ZDAQUTIL.LLB\SCXI Temperature Scan.vi

This VI returns a single scan of temperature data from a list of SCXI channel.

Click here for more help.

Fuente: Franklin Robalino

En la parte inferior izquierda de la ventana de ayuda hay un botón con un ícono de candado el cual permite asegurar el contenido de la ventana, es decir si se pasa por encima de otro objeto en el panel frontal o diagrama de bloques, esta acción no hará que cambie el contenido de la ventana de ayuda. Para regresar al modo de operación regular de la ventana de ayuda solamente hay que hacer click nuevamente en el ícono con el candado.

El ícono “*More Help*” tiene un signo de pregunta y también está ubicado en la parte inferior izquierda de la ventana de ayuda. Este botón es un link para entrar en la documentación referencial en línea donde se describe con más detalles el objeto seleccionado.

Ventana de “*Contents and Index*”

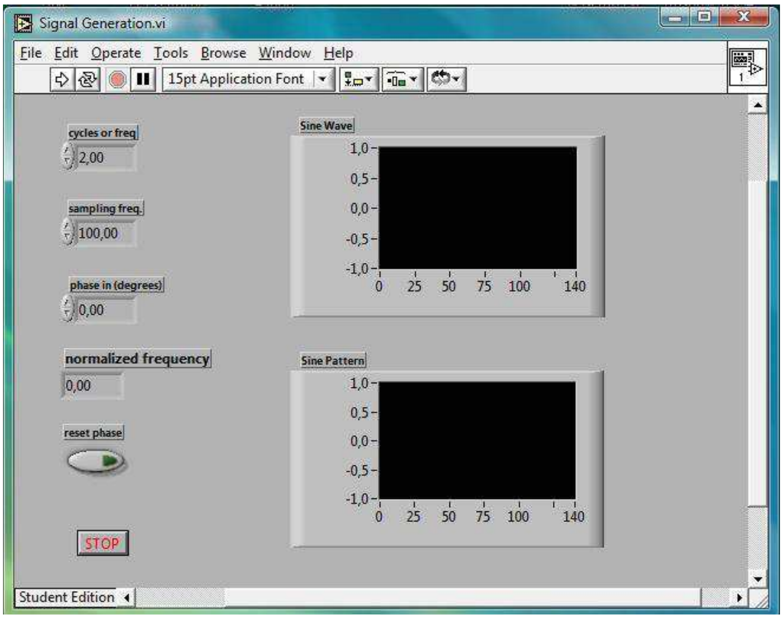
Las referencias en línea de Lab VIEW contienen una descripción detallada de la mayoría de objetos en el diagrama de bloques. Para ver esta información se lo puede hacer mediante el ícono de “*More Help*” dentro de la ventana de “*Context Help*”, como se explicó anteriormente. También se lo puede hacer escogiendo la opción “*Contents and Index*” del menú “*Help*” o haciendo click en la oración “*Click here for more help*” dentro de la ventana de “*Context Help*”.

3.7. INSTRUMENTOS VIRTUALES

Los instrumentos virtuales también conocidos como VIs, son los cimientos dentro de la programación de Lab VIEW. Los instrumentos virtuales tienen tres componentes principales: el panel frontal, el diagrama de bloques y los conectores del ícono.

Los programas de Lab VIEW son llamados instrumentos virtuales o VIs, por que se ven y se sienten como sistemas físicos de instrumentos reales. En la figura 3.23 se puede observar un ejemplo de un VI en el panel frontal. Un VI y sus componentes son lo que en lenguajes de programación textual se conocen como programas principales y subrutinas. Los VI tienen tanto una interface interactiva conocida como el panel frontal y un código fuente representado gráficamente en el diagrama de bloques. Lab VIEW contiene mecanismos que permiten a los datos pasar fácilmente entre el panel frontal y el diagrama de bloques.

Figura 3.23.- Panel frontal de un VI


Signal Generation.vi
File Edit Operate Tools Browse Window Help
15pt Application Font
Sine Wave
1,0
0,5
0,0
-0,5
-1,0
0 25 50 75 100 140
Sine Pattern
1,0
0,5
0,0
-0,5
-1,0
0 25 50 75 100 140
STOP
Student Edition

cycles or freq: 2,00

sampling freq.: 100,00

phase in (degrees): 0,00

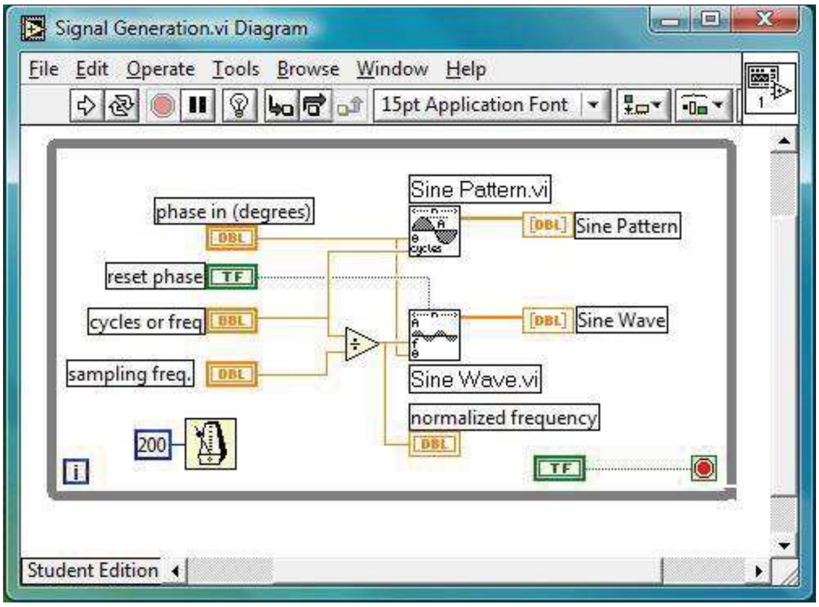
normalized frequency: 0,00

reset phase

Fuente: Franklin Robalino

El diagrama de bloques es una representación gráfica del código de programación. El diagrama de bloques que está asociado con el panel frontal de la figura 3.23 se muestra en la figura 3.24. El diagrama de bloques consiste en íconos ejecutables llamados nodos, los cuales están conectados unos con otros por medio de cables. Es importante tener en cuenta que en el lenguaje de programación grafica (*G programming*) el diagrama de bloques es el código fuente.

Figura 3.24.- Diagrama de bloques del VI de la figura 3.23.


Signal Generation.vi Diagram
File
Edit
Operate
Tools
Browse
Window
Help
15pt Application Font
Sine Pattern.vi
phase in (degrees)
DBL
Sine Pattern
reset phase
TF
cycles or freq
Sine Wave
sampling freq.
Sine Wave.vi
normalized frequency
200
Student Edition

Fuente: Franklin Robalino

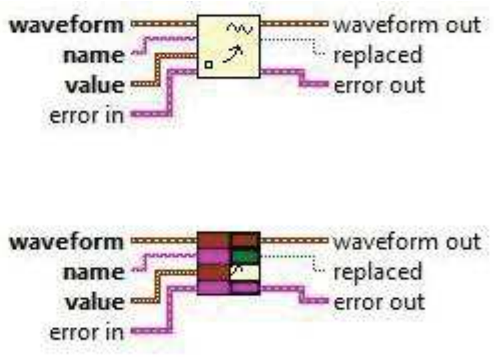
La programación gráfica es un arte. Después de dividir a un trabajo en una serie de subrutinas más simples, los cuales dentro de la programación gráfica son llamados sub Vis. Se construye un instrumento virtual el cual realizará cada una de las subrutinas. Estas subrutinas, llamados sub VIs son luego ensambladas en un diagrama de bloques para formar un programa completo. Cada sub VI se ejecuta independientemente, por lo tanto la verificación correcta del subprograma y la corrección de errores se facilitan, es más si es que el sub VI es un programa de propósito general, puede ser usado dentro de otro programa.

Los VIs y los sub Vis, como se mencionó anteriormente, tienen tres partes principales: el panel frontal, el diagrama de bloques y los conectores del ícono. El panel frontal es la interface interactiva del usuario con el VI, es una ventana por la cual el usuario interactúa con el código. Cuando se corre (ejecuta) un instrumento virtual, el panel frontal debe estar abierto para poder ingresar datos de entrada (*input*) al programa que está siendo ejecutado, para que este a la vez genere las salidas (*outputs*), como por ejemplo gráficos. El panel frontal es indispensable para poder observar los resultados del programa. También es posible mandar dichos outputs a un archivo para un analizarlos después, lo cual será útil para guardar los gráficos de las audiometrías. El panel frontal contiene perillas, botones, gráficos, y muchos otros tipos de controles (*inputs*) e indicadores (*outputs*).

El diagrama de bloques es el código fuente (principal) para el VI. El código fuente está representado en el lenguaje de programación gráfica por medio de íconos, cables, etc., en vez de líneas de códigos. El diagrama de bloques es realmente el código que se ejecuta. Los íconos del diagrama de bloques representan VIs de inferior nivel, funciones, y estructuras de control para el programa. Estos íconos están conectados entre sí para permitir el flujo de datos. Hay que tener en cuenta que en la ejecución de un programa gráfico importa mucho la manera en que fluyen los datos, ya que este tipo de programas no se rigen a una ejecución linear como en las líneas de código que se leen siempre de arriba hacia abajo. A este concepto se lo conoce como "*data flow programing*", porque depende del flujo de datos para ver en qué secuencia se ejecutarán las distintas partes del código.

Los íconos y los conectores especifican el camino o ruta por donde fluirán los datos desde o hacia un VI. El ícono es la representación gráfica de un VI en el diagrama de bloques, y el conector define los *inputs* y *outputs*. Todos los VIs tienen un ícono y al menos un conector. Los VIs son jerárquicos y modulares, como ya se menciono anteriormente, es decir se los puede usar como programas de nivel superior o como subprogramas (sub VIs) dentro de otros programas. Un ejemplo del ícono y sus conectores se muestra a continuación en la figura 3.25.

Figura 3.25.- Icono y sus conectores



Fuente: Franklin Robalino

3.8. EL PANEL FRONTAL

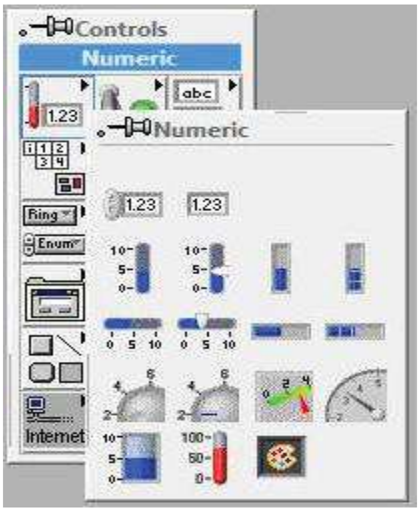
El panel frontal de un instrumento virtual es una combinación de controles e indicadores. Los controles simulan algunos tipos de dispositivos que se encuentran en instrumentos convencionales, tales como perillas, y interruptores, los cuales poseen un mecanismo para mover datos de *input* desde el panel frontal hacia el diagrama de bloques. Por otro lado los indicadores poseen un mecanismo para mostrar datos que se originan en el diagrama de bloques, para luego ser mostrados en el panel frontal. Dentro de los indicadores están varios tipos de gráficos, valores numéricos, Booleano (*Boolean*), e indicadores tipo *string*.

Para poner controles e indicadores en el panel frontal, se los selecciona y se los arrastra desde la paleta de controles (*Controls palette*) hacia el panel frontal. Una vez seleccionado el control o indicador de la paleta de controles y ubicado en el panel frontal, el cursor cambiará a un ícono en forma de mano, el cual es usado para mover el objeto a cualquier lugar deseado dentro del panel frontal, para soltarlo en la ubicación deseada se hace click con el botón del mouse y listo. Una vez que el objeto esté en el panel frontal, puede ser reajustado tanto en tamaño, forma o posición.

Controles e indicadores numéricos

Para ingresar a los controles numéricos e indicadores, se lo hace mediante la sub paleta numérica que está dentro de la paleta de control (*Controls palette*), como se muestra en la figura 3.26. En la figura podemos observar que existen un gran número de controles numéricos e indicadores disponibles. Los dos objetos numéricos más usados son los controles digitales y los indicadores digitales.

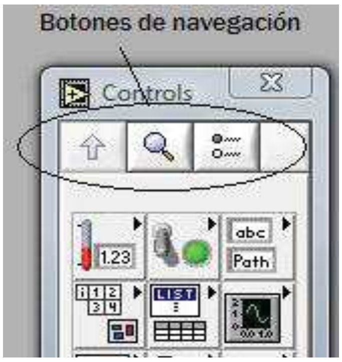
Figura 3.26.- Controles numéricos



Fuente: Franklin Robalino

Se puede usar los botones de navegación que se encuentran en las paletas tanto de controles como de funciones, como se muestra en la figura 3.27, para navegar y buscar controles, VIs y funciones. Al hacer un click izquierdo en un ícono de una sub paleta, la paleta entera cambiará a la sub paleta que fue seleccionada. En cambio al hacer un click derecho en un ícono de una sub paleta, la sub paleta aparecerá, pero la paleta de controles o funciones permanecerá a la vista en la pantalla. Un ejemplo de un ícono de una sub paleta en la paleta de controles es el ícono de la sub paleta numérica (*Numerics*). Tanto la paleta de controles (*Controls*) como la paleta de funciones (*Functions*) contienen tres botones de navegación, como se muestra en la figura 3.27.

Figura 3.27.- Botones de navegación



Fuenta: Franklin Robalino

El botón **“*Up*”** permite subir al nivel superior (jerárquicamente) en la paleta.

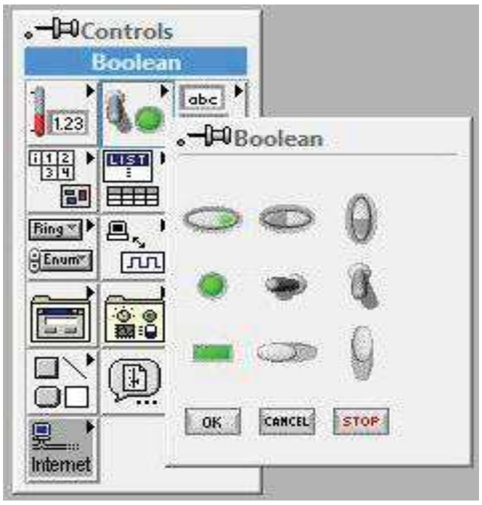
El botón **“*Search*”** permite cambiar la paleta al modo de búsqueda. En este modo se puede buscar controles, VIs o funciones dentro de la paleta, con tan solo escribir en el espacio de búsqueda, el ícono que se desee buscar.

El botón **“*Options*”** permite abrir un cuadro de diálogo llamado “*Function Browser Options*”, en el cual se puede configurar la apariencia de las paletas.

Controles e indicadores Booleanos

Para ingresar al los controles e indicadores Booleanos, se lo hace mediante la sub paleta “*Boolean*” que está dentro de la paleta de control (*Controls palette*), como se muestra en la figura 3.28. Al igual que con los controles e indicadores numéricos, existe una gran cantidad de controles e indicadores disponibles. Los controladores e indicadores Booleanos simulan a interruptores, botones, y LEDs, y son usados para ingresar y mostrar valores Booleanos (Verdadero y Falso). Por ejemplo, se puede usar un LED Booleano en un sistema de de monitoreo de sonido como una señal de aviso que se ha sobrepasado algún nivel.

Figura 3.28.- Controles e indicadores Booleanos

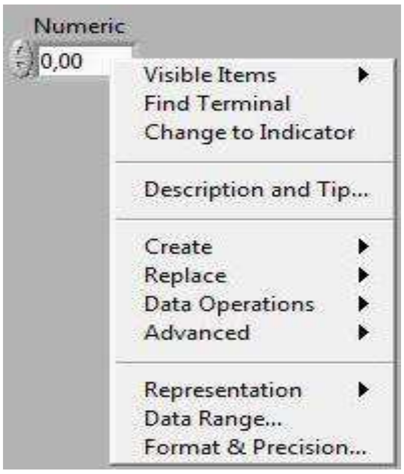


Fuente: Franklin Robalino

Configuración de controles e indicadores

Al hacer un "*pop up*" en un control digital, como se muestra en la figura 3.29. Se puede cambiar los valores default de los controles e indicadores mediante el uso de las opciones disponibles en el menú. Por ejemplo, dentro del sub menú "Representation" (*representación*) se podrá escoger entre 12 representaciones del control digital o del indicador, incluyendo precisiones de 32 bit (*single precision*), 64 bit (*double precision*), 8 bit (*signed integer*), entre otras. La representación indica cuánta memoria es usada para representar un número. Esto hay que tomarlo muy en cuenta en especial si se desea mostrar grandes cantidades de datos y conservar la memoria del computador. Otra opción importante dentro del menú es la capacidad de cambiar un control a un indicador usando la opción "*Change to indicator*" o a su vez si es un indicador y se lo desea cambiar a control, aparecerá la opción "*Change to Control*". En el capítulo 4 se explicará otras opciones de este y otros menús a manera que sean utilizadas dentro de la programación del software de medida de pérdida auditiva, aquí se explicarán sólo las opciones y herramientas usadas para la elaboración de la tesis, si se requiere más información, referirse a la bibliografía.

Figura 3.29.- “pop up” en un control digital


Numeric
0,00
Visible Items
Find Terminal
Change to Indicator
Description and Tip...
Create
Replace
Data Operations
Advanced
Representation
Data Range...
Format & Precision...

Fuente: Franklin Robalino

3.9. DIAGRAMA DE BLOQUES

Los objetos gráficos que se encuentran dentro del diagrama de bloques conforman lo que se llama el código fuente o código principal. El diagrama de bloques representa visualmente el diagrama de flujo del programa, y corresponde a las líneas de texto que se encuentran en cualquier lenguaje de programación que se basa en texto. Es decir, el diagrama de bloques es el código ejecutable del programa. El diagrama de bloques se lo construye al conectar varios objetos para que ejecuten una función específica.

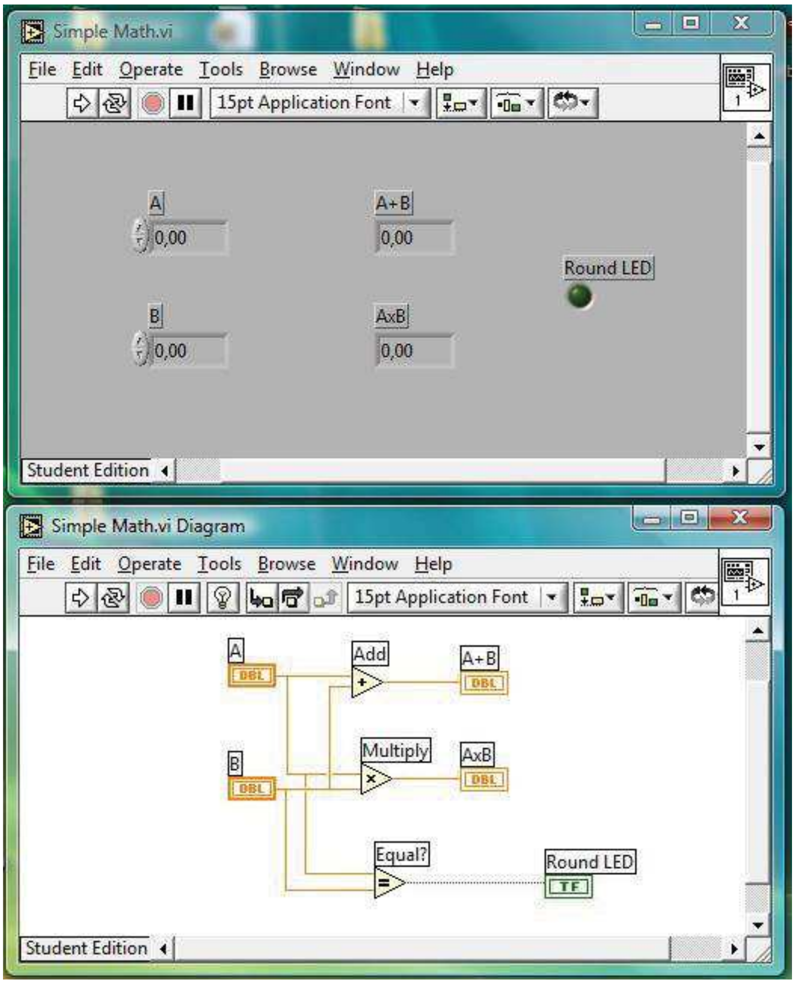
Los componentes de un diagrama de bloques, como se muestran en la figura 3.30, pertenecen a una de las tres clases de objetos:

“*Nodes*” (nodos).- son elementos ejecutables del programa.

“*Terminals*” (terminales).- son puertos por los cuales pasan los datos entre el diagrama de bloques y el panel frontal y también pasan los datos a otros nodos del diagrama de bloques.

“*Wires*” (cables).- son los caminos o rutas que tomaran los datos para cruzar entre los distintos terminales.

Figura 3.30.- Panel frontal y diagrama de bloques con componentes


Simple Math.vi
File Edit Operate Tools Browse Window Help
15pt Application Font
A
0,00
B
0,00
A+B
0,00
AxB
0,00
Round LED
Student Edition
Simple Math.vi Diagram
File Edit Operate Tools Browse Window Help
15pt Application Font
A
DBL
B
DBL
Add
A+B
DBL
Multiply
AxB
DBL
Equal?
Round LED
TF
Student Edition

Fuente: Franklin Robalino

Nodos

Los nodos son iguales que las funciones, subrutinas, o frases (*statements*) en un lenguaje de programación basado en texto. Existen tres tipos de nodos: funciones (*functions*), sub VIs, y estructuras (*structures*). Las funciones son los nodos que sirven para ejecutar operaciones elementales como sumar números. Las funciones "*Add*" (sumar) y "*Multiply*" (multiplicar) (ver figura 3.30) representan un tipo de nodo. Los nodos sub VIs son VIs diseñados por el programador. Las estructuras tales como los "*While Loops*" controlan cómo fluirá la información de datos dentro del programa.

Terminales

Los terminales son iguales a parámetros y constantes en un lenguaje de programación basado en texto. Existen algunos tipos de terminales: terminales de controles e indicadores, terminales de nodo, constantes, y terminales especiales que se encuentran en varias estructuras. En otras palabras un terminal es cualquier punto por el cual se puede conectar un cable para el paso de datos.

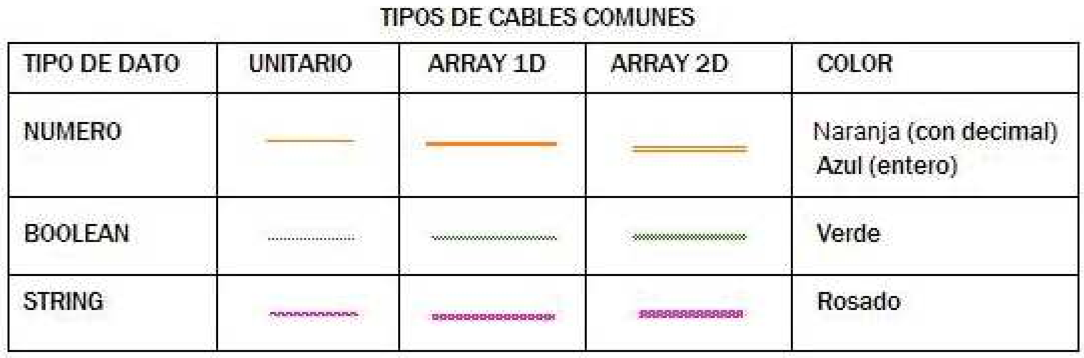
Por ejemplo en el caso de terminales de controles e indicadores, los datos numéricos que ingresan en un control numérico del panel frontal pasan al diagrama de bloques por medio de los terminales de control cuando se ejecuta el VI. Cuando el VI termina de ejecutarse, los datos del output numérico pasan desde el diagrama de bloques hacia el panel frontal mediante los terminales del indicador. Los datos fluyen en una sola dirección, desde un terminal de fuente (*source*) a un terminal de destino (*destination*). Los controles son terminales de fuente y los indicadores son terminales de destino. La dirección del flujo de datos es siempre desde el terminal de control hacia el terminal del indicador, y nunca de manera contraria.

Los terminales de los controles e indicadores pertenecen a los controles e indicadores del panel frontal, y son automáticamente creados o borrados al crear o borrar el control o indicador correspondiente del panel frontal. No se puede borrar un terminal en el diagrama de bloques que pertenezca a un control o indicador, el terminal solo desaparecerá al borrar el control o indicador correspondiente del panel frontal. El diagrama de bloques de la figura 3.30 muestra terminales que pertenecen a dos controles numéricos y dos indicadores numéricos del panel frontal. Las funciones sumar y multiplicar del diagrama de bloques también poseen terminales de nodo por los cuales los datos del input y output fluyen hacia adentro y hacia afuera de la función. Hay que tener en cuenta que los terminales de los controles tienen bordes más gruesos que los terminales de los indicadores. Distinguir entre los bordes gruesos de los controles y los bordes delgados de los indicadores es muy importante debido a que estos tienen dos maneras muy diferentes de funcionar.

Cables

Los cables son caminos o rutas de datos entre los distintos terminales. Los cables permiten representar diferentes tipos de datos que fluyen dentro del diagrama de bloques. Debido a que el diagrama de bloques consiste de objetos gráficos, es apropiado utilizar diferentes tipos de patrones para representar los distintos cables (figuras, estilos de líneas, color, etc.) que representaran el tipo de datos. Cada cable posee un patrón único el cual depende del tipo de datos (numérico, *Boolean*, *string*, etc.) que fluyen por el cable. Para determinar el tipo de datos que fluyen por un cable en particular, se debe comparar el color y el estilo del cable, con los tipos de cable de la figura 3.31.

Figura 3.31.- Tipos de cables



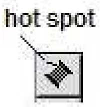
TIPOS DE CABLES COMUNES

TIPO DE DATO	UNITARIO	ARRAY 1D	ARRAY 2D	COLOR
NUMERO	—	━	═	Naranja (con decimal) Azul (entero)
BOOLEAN	┈	┈	┈	Verde
STRING	∼	∼	∼	Rosado

Fuente: Franklin Robalino

El punto clave (*hot spot*) de la herramienta "*Wiring*" es la punta suelta del segmento del cable que se muestra en el ícono (ver figura 3.32).

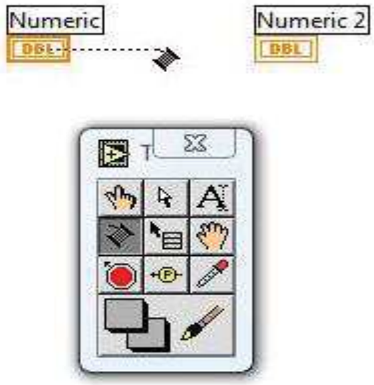
Figura 3.32.- Herramienta "Wiring"



Fuente: Franklin Robalino

Para conectar un terminal a otro con el cable, hay que hacer click con el “*hot spot*” de la herramienta “*Wiring*” en el terminal deseado, mover la herramienta al segundo terminal, y hacer click en el segundo terminal. No es necesario mantener presionado el botón del mouse mientras se mueve la herramienta “*Wiring*” desde un terminal hacia otro. El proceso para conectar los cables está ilustrado en la figura 3.33. Hay que tener en cuenta que al conectar dos terminales, el rato que se posiciona la herramienta “*Wiring*” encima de un terminal causara que el terminal se ponga intermitente (parpadee). Esto es una indicación que se hará una conexión correcta en ese terminal al hacer click con el botón del mouse.

Figura 3.33.- Proceso de conexión de cables



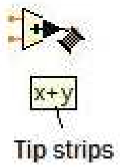
Fuente: Franklin Robalino

El VI que se muestra en la figura 3.33 es un ejemplo típico de conexión de cable. Si después de conectar el primer terminal (sin conectar al segundo) se desea borrar el cable en el proceso, sólo hay que hacer click con el botón derecho del mouse en cualquier lugar o conectarlo nuevamente en el terminal principal.

Al conectar dos terminales juntos, puede que se desee doblar el cable para evadir otros objetos que se encuentren en el paso. Esto se puede lograr durante el proceso de cableado al hacer click con el botón del mouse para anclar el cable en algún lugar y luego cambiarlo de dirección a donde se requiera enviarlos (siempre en dirección perpendicular). Otra manera de cambiar la dirección del cable mientras se cablea es aplastando la barra espaciadora mientras se mueve la herramienta de "*Wiring*".

Los "*Tip strips*" facilitan el cableado, identificando la función y terminal de nodo al cual se va a conectar. La mover la herramienta de "*Wiring*" por encima de un terminal, aparecerá un "*tip strip*", como se puede observar en la figura 3.34. Los "*tip strips*", son pequeñas ventanas de texto que muestran el nombre de cada terminal. Al poner la herramienta de "*Wiring*" sobre un nodo, se mostrará cada *input* y *output*, los terminales de inputs tendrán un punto como referencia (ver figura 3.34). Los "*tip strips*" son una ayuda para hacer las conexiones correctas.

Figura 3.34.- "Tip strips"

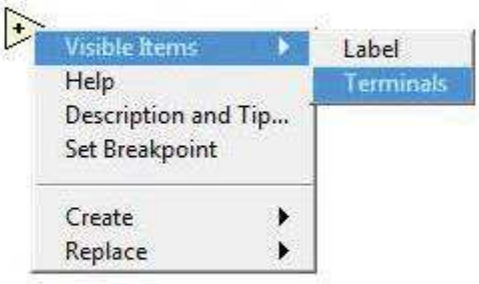


Fuente: Franklin Robalino

Es posible hacer que los objetos se conecten automáticamente al ser soltados en el diagrama de bloques. Una vez seleccionado el objeto de la paleta de funciones (*Functions palette*), y moviendo el nodo cerca de otro nodo al cual se desee conectar, entonces los terminales que contienen el mismo tipo de datos, se conectarán automáticamente. Se puede desactivar la conexión automática al aplastar la barra espaciadora. Para poder ajustar las conexiones automáticas se lo hace entrando a "Tools>>Options>>Block Diagram".

Debido a que es muy importante las conexiones correctas entre los terminales de las funciones, el programa Lab VIEW permite mostrar el conector del ícono, y de esta manera facilita la conexión. Para lograr esto, se necesita hacer un "*pop up*" en la función y escoger la opción "Visible Items>>Terminals" dentro del menú, como se observa en la figura 3.35. Si se desea ver puntos en los lugares donde existen junturas de cable en el diagrama de bloques, hay que habilitarlos entrando a “Options>>Block Diagram” dentro del menú “*Tools*” y haciendo click en el primer casillero que dice “*Show dots at wire junctions*”.

Figura 3.35.- Menu “pop up”



Fuente: Franklin Robalino

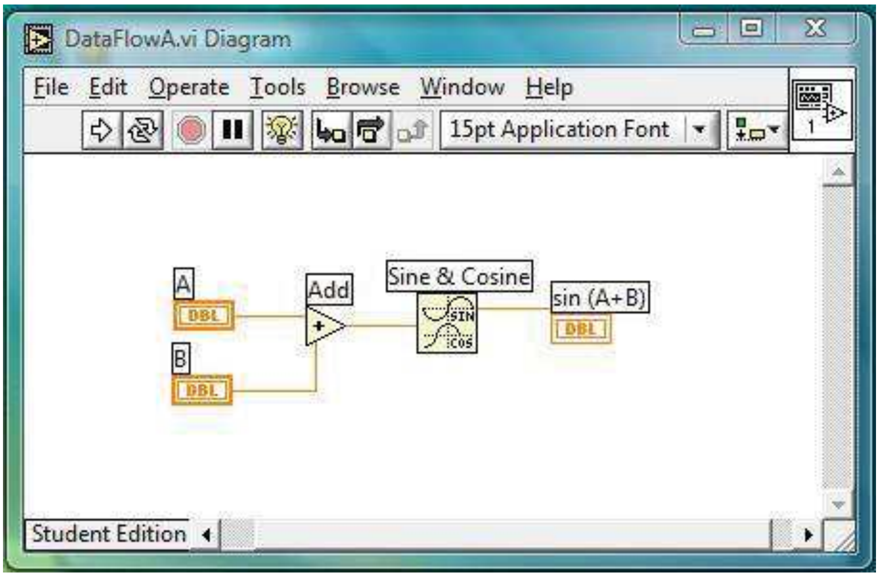
3.10. PROGRAMACIÓN CON “*DATA FLOW*”

El principio de ejecución de un VI se denomina “*data flow*” (flujo de datos). A diferencia de otros lenguajes de programación secuenciales, los elementos ejecutables de un VI son ejecutados solo cuando se haya recibido todos los datos de *input*, es decir los datos fluyen hacia afuera del elemento ejecutable solamente cuando el código se haya completado de ejecutar. La idea de un flujo de datos es contradictoria al método de flujo de un lenguaje de programación convencional, en donde las líneas de código se ejecutan secuencialmente en el orden especificado por el programador.

Para entender mejor el flujo de datos observemos el VI en el diagrama de bloques que se muestra en la figura 3.36. Este VI suma dos números y después saca el seno del resultado. En este caso, el diagrama de bloques se ejecutará de izquierda a derecha, esto no se debe a que los objetos están puestos en ese orden (de izquierda a derecha), sino porque uno de los *inputs* de la función “*Sine & Cosine*” (seno & coseno) no existe hasta que la función

“*Add*” (suma) haya sumado los dos números y transferido el resultado a la función “*Sine & Cosine*”. Hay que tener en cuenta que un nodo, como es en este caso la función “*Sine & Cosine*”, se ejecutará sólo cuando haya recibido todos los datos necesarios de input en sus terminales, y entregará datos por medio de sus *outputs* sólo cuando se haya completado la ejecución.

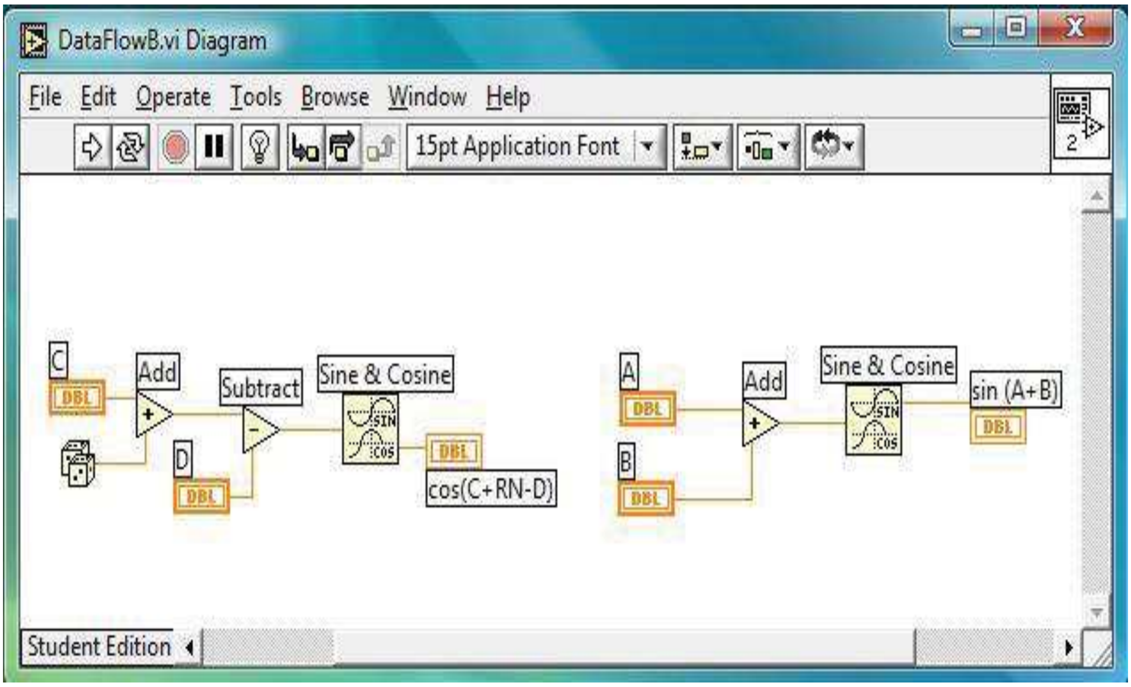
Figura 3.36.- Diagrama de bloques de un VI


DataFlowA.vi Diagram
File Edit Operate Tools Browse Window Help
15pt Application Font
1
A
DBL
B
DBL
Add
Sine & Cosine
SIN
COS
sin (A+B)
DBL
Student Edition

Fuente: Franklin Robalino

Otro caso típico para entender el flujo de datos se muestra en la figura 3.37. En este ejemplo la función coseno no necesariamente terminará primero de ejecutarse y luego la función seno, es decir no se ejecutará de izquierda a derecha. En este caso si se desea ejecutar una función antes que la otra, se necesita forzar el código mediante una estructura de secuencia, como por ejemplo un “*While Loop*”. Para poder ver el flujo de datos hay que hacer click en el botón llamado “*Execution Highlighting*” que tiene un ícono con un foco, como se muestra en la figura 3.36, este hará que se resalte el flujo de datos mientras se ejecuta el programa.

Figura 3.37.- Diagrama de bloques de un VI


DataFlowB.vi Diagram
File Edit Operate Tools Browse Window Help
15pt Application Font
C
DBL
Add
Subtract
Sine & Cosine
D
DBL
cos(C+RN-D)
A
B
sin (A+B)
Student Edition

Fuente: Franklin Robalino

3.11 EDITAR Y ELIMINAR ERRORES DE UN VI

Al igual que los lenguajes de programación basados en texto, los instrumentos virtuales o VIs son dinámicos. Los VIs cambian cada vez que su aplicación se evoluciona, es decir cada vez que el VI se hace más complejo. Por ejemplo, un VI que al principio sólo realiza operaciones de suma, puede después requerir una capacidad de multiplicación y por ende se vuelve más complejo. Al hacerse cada vez más complejo, el VI puede dejar de funcionar por cualquier error de codificación, es por esto que se necesitan herramientas de edición y de eliminación de errores (*debugging*), para poder verificar y probar los cambios que se realicen en el código del VI. Debido a que la programación en G (*G programming*) es de naturaleza gráfica, la edición y la eliminación de errores también es gráfica, con diferentes opciones tanto en los menús como en las paletas.

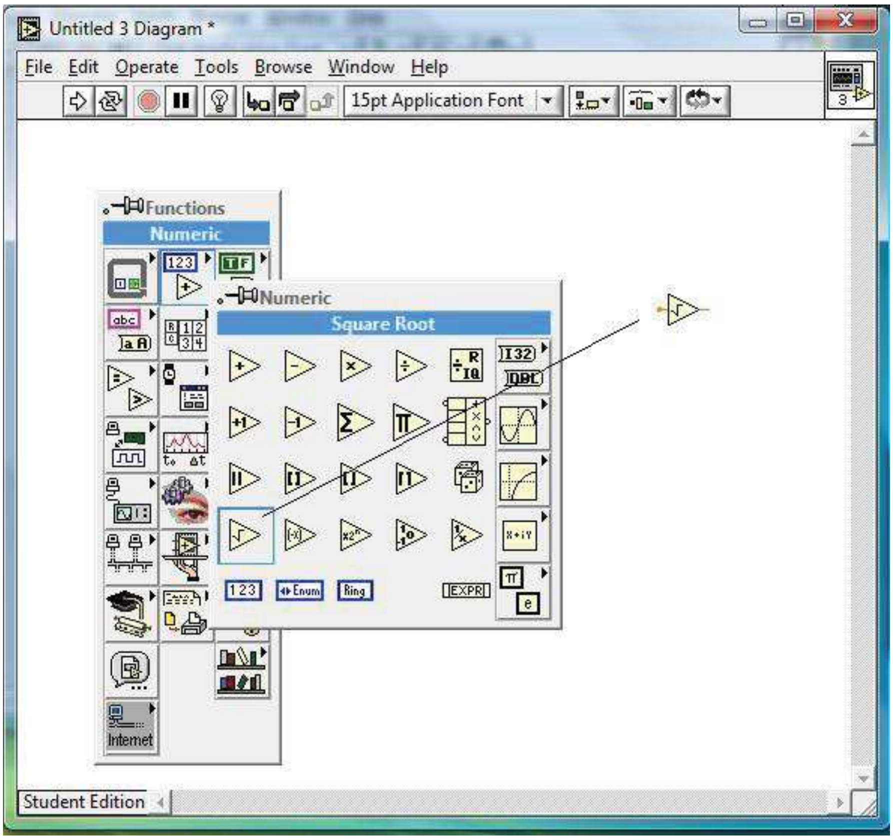
TÉCNICAS DE EDICIÓN

Como crear controles e indicadores en el diagrama de bloques

Al construir un instrumento virtual se puede crear controles e indicadores en el panel frontal y automáticamente aparecerán sus respectivos terminales en el diagrama de bloques. En el diagrama de bloques se puede realizar las conexiones entre los terminales de las funciones, sub VIs u otros objetos.

Existe también un método alternativo para crear y conectar los controles e indicadores en un solo paso dentro del diagrama de bloques. Para realizar este método debe haber un VI abierto con su respectivo diagrama de bloques en pantalla. Una vez que esté abierto el diagrama de bloques se colocará la función deseada de la paleta de funciones, como se muestra en la figura 3.38.

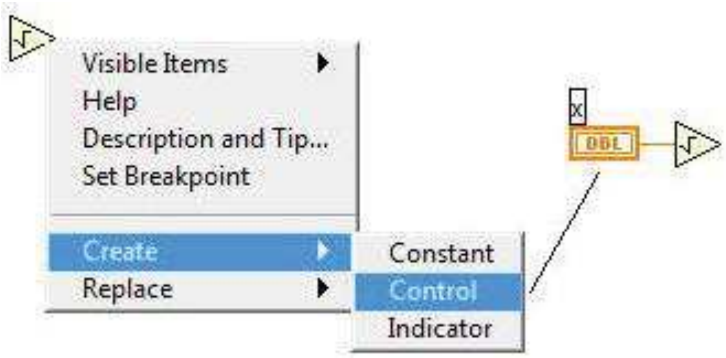
Figura 3.38.- Método para crear funciones


Untitled 3 Diagram *
File
Edit
Operate
Tools
Browse
Window
Help
15pt Application Font
Functions
Numeric
Numeric
Square Root
Internet
Student Edition

Fuente: Franklin Robalino

Después para poder añadir un cable y un control al terminal de la función, como se muestra en la figura 3.39, se hace un "pop up" en el lado que se encuentra el terminal (en este caso el lado izquierdo de la función), y se selecciona "Create>>Control" dentro del menú. Como resultado de este proceso un terminal de control será creado y automáticamente conectado a la función. Hay que tener en cuenta que al realizar este proceso aparecerá un control numérico también en el panel frontal, el cual corresponde al control recién creado en el diagrama de bloques.

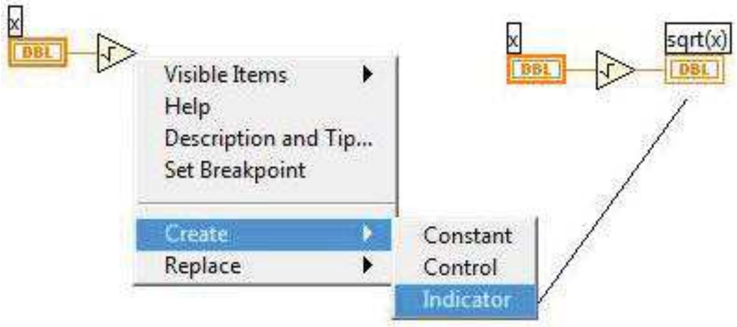
Figura 3.39.- Crear control



Fuente: Franklin Robalino

Se puede conectar indicadores y constantes al hacer un "pop up" en los terminales de cualquier nodo. Por ejemplo, al hace un "pop up" en el lado derecho de la función "*Square Root*" (raíz cuadrada), como se muestra en la figura 3.40, y seleccionando "Create>>Indicator" dentro del menú, esta acción creará y automáticamente conectará un indicador hacia el panel frontal. Este método es muy conveniente debido a que en la mayoría de ocasiones, se deseará crear y conectar automáticamente una constante a un terminal.

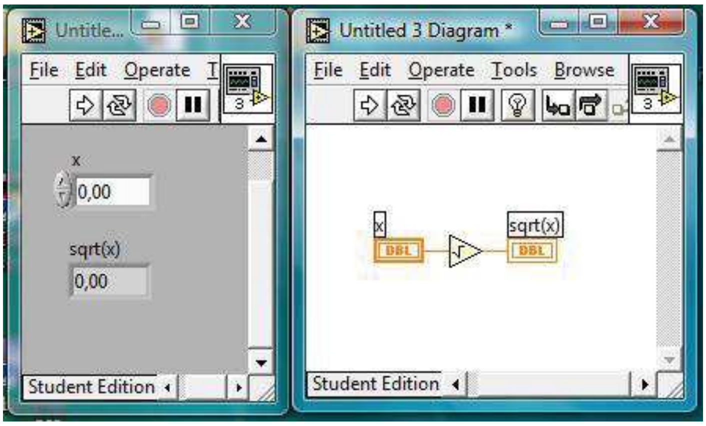
Figura 3.40.- Crear indicador


x
DBL
Visible Items
Help
Description and Tip...
Set Breakpoint
Create
Replace
Constant
Control
Indicator
x
DBL
sqrt(x)
DBL

Fuente: Franklin Robalino

En la figura 3.41 se muestra el resultado de este proceso, en donde se puede observar los controles e indicadores tanto en el panel frontal como en el diagrama de bloques.

Figura 3.41.- Panel frontal y diagrama de bloques


Untitle...
File Edit Operate T
x
0,00
sqrt(x)
0,00
Student Edition
Untitled 3 Diagram *
File Edit Operate Tools Browse
x
DBL
sqrt(x)
DBL
Student Edition

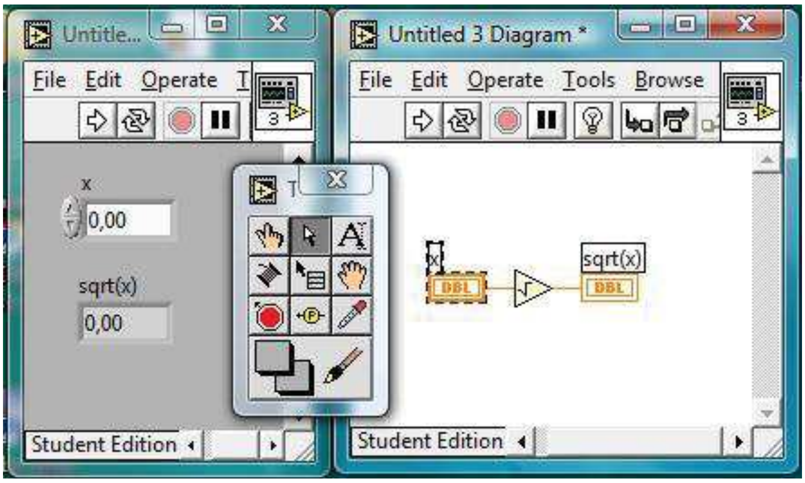
Fuente: Franklin Robalino

Se puede también cambiar un control o un indicador que ya esté creado y conectado, por otro que lo substituya, por ejemplo cambiar una perilla de control análoga por una perilla digital (*dial*). Para hacer esto hay que hacer un “pop up” en el objeto que se desee remplazar del panel frontal y seleccionar “*Replace*”. La paleta de controles (*controls palette*) aparecerá, para que se pueda seleccionar al objeto con el cual va ha ser reemplazado el objeto seleccionado del panel frontal.

Seleccionar objetos

Para poder seleccionar los objetos tanto en el panel frontal como en el diagrama de bloques se lo hace con la herramienta de "*Positioning*" (posicionamiento) que está dentro de la paleta de herramientas. Además de poder seleccionar objeto con esta herramienta de posicionamiento, también se los puede mover y reajustar de tamaño. Para seleccionar el objeto deseado hay que hacer click sobre el objeto con la herramienta de posicionamiento, esto hará que aparezca un borde con líneas segmentadas alrededor del objeto seleccionado, como se muestra en la figura 3.42.

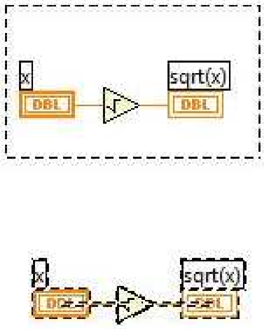
Figura 3.42.- Seleccionar objeto


Untitle...
File Edit Operate T
x
0,00
sqrt(x)
0,00
Student Edition
Untitled 3 Diagram *
File Edit Operate Tools Browse
x
DBL
sqrt(x)
DBL
Student Edition

Fuente: Franklin Robalino

Para poder seleccionar más de un objeto hay que presionar *Shift*-click en los objetos deseados, es decir mantener presionada la tecla *shift* y presionar el botón del mouse en cada objeto que se desee seleccionar. Otra manera de seleccionar varios objetos es haciendo click en un espacio vacío y arrastrando el mouse alrededor de los objetos deseados, al arrastrar el mouse se creará un rectángulo imaginario el cual sirve de referencia para asegurarse que los objetos que estén dentro de este estén seleccionados, como se muestra en la figura 3.43. Si por alguna circunstancia después de haber seleccionado varios objetos, se requiere deseleccionar uno, esto se lo hace al presionar *Shift*-click en el objeto a deseleccionar.

Figura 3.43.- Seleccionar objeto

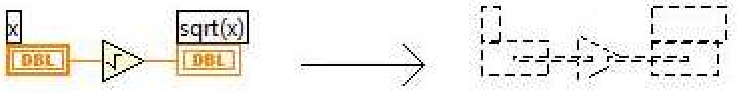


Fuente: Franklin Robalino

Moviendo Objetos

Para mover un objeto hay que hacer click en el objeto deseado con la herramienta "*Positioning*", y luego arrastrar al objeto al lugar deseado, como se muestra en la figura 3.44. Los objetos seleccionados también pueden moverse al presionar las flechas del teclado. Al presionar una de las flecha una sola vez, causará que el objeto se mueva una distancia de un pixel, mientas que si se mantiene presionada la tecla moverá el objeto repetidamente hasta que se deje de presionar dicha tecla, de esta manera se puede mover un objeto de una forma más precisa.

Figura 3.44.- Mover objeto



Fuente: Franklin Robalino

Borrando y Duplicando Objetos

Para borrar objetos hay que seleccionar el objeto u objetos que se deseen borrar, y seleccionar la opción "*Clear*" dentro del menú "*Edit*", también se lo puede hacer con el botón "*backspace*" del teclado. La mayoría de objetos pueden ser borrados de esta manera, ya que existen componentes de los controles e indicadores, que no pueden ser borrados, como el membrete del objeto o el *display* digital. Por esta razón se recure al método de esconder componentes, esto se hace con un "pop up" en el objeto y deseleccionando lo que se requiera esconder dentro del menú bajo "*Visible Items*".

La mayoría de objetos pueden ser duplicados, y existen tres maneras distintas para duplicarlos:

1.- copiar y pegar

2.- clonando el objeto

3.- arrastrando y soltando al objeto

Se puede duplicar objetos por medio del menú de edición, usando "Edit>>Copy" y "Edit>>Paste". Primero hay que ubicar el cursor con la herramienta "*Positioning*" sobre el objeto deseado y seleccionarlo, luego escoger "Edit>>Copy". Una vez que el objeto ya esté copiado, hay que hacer click en el lugar donde se desee pegar la copia y escoger "Edit>>Paste". Este proceso también se le puede hacer por medio del teclado con ctrl+c y ctrl+v. Hay que tener en cuenta que todos los procesos y combinaciones de teclas descritos en esta tesis son específicamente para una pc (Windows), para una mac será distinto.

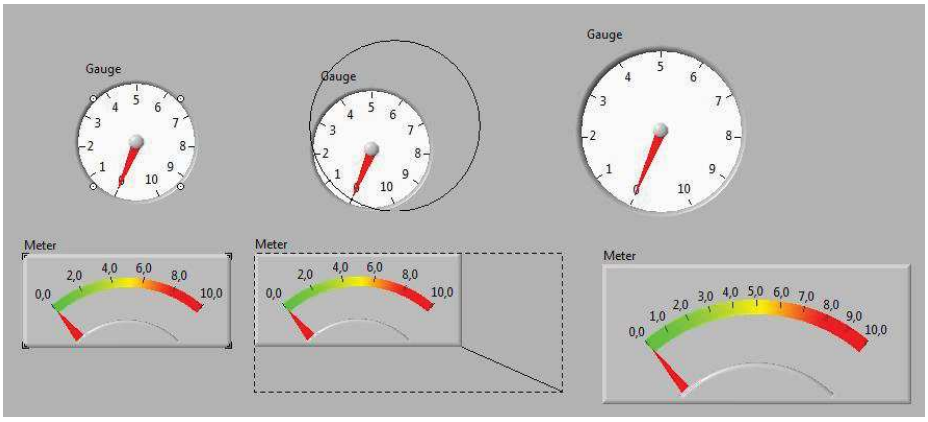
Para poder clonar un objeto, hay que hacer click sobre el objeto con la herramienta de "*Positioning*" mientras se presiona la tecla <Ctrl>, y arrastrar al objeto al lugar deseado. Después de arrastrar el objeto al lugar deseado y soltarlo, aparecerá una copia del objeto en el nuevo lugar, mientras que el original permanecerá en su lugar original. Al clonar o copiar objetos las copias son membretadas con el mismo nombre del objeto original y la palabra "*Copy*" junto al nombre.

El tercer método para duplicar objetos es arrastrando y soltando al objeto. Con este método se pueden copiar objetos entre diferentes VIs. Para hacer esto se selecciona los objetos con la herramienta "*Positioning*" y se los arrastra al panel frontal o diagrama de bloques del otro VI.

Reajustando el Tamaño de los Objetos

Se puede cambiar de tamaño fácilmente a la mayoría de objetos. Los "*Resizing handles*" (palancas de tamaño) aparecen al mover la herramienta "*Positioning*" sobre un objeto que tenga la capacidad de ser reajustado en tamaño, como se muestra en la figura 3.45. En los objetos con forma rectangular las palancas de tamaño aparecerán en las esquinas del objeto, mientras que en objetos circulares aparecerán las palancas de forma circular alrededor del objeto. Al pasar la herramienta de "*Positioning*" sobre una palanca de tamaño hará que cambie la herramienta a un cursor de reajuste de tamaño. Para agrandar o reducir el tamaño de un objeto, hay que posicionar la herramienta "*Positioning*" sobre la palanca de tamaño del objeto y luego hacer click y arrastrar el cursor de reajuste de tamaño hasta llegar al tamaño deseado. Al soltar el botón del mouse, el objeto reaparecerá con su nuevo tamaño. Este proceso está ilustrado en la figura 3.45.

Figura 3.45.- Reajustando el Tamaño de los objetos



Fuente: Franklin Robalino

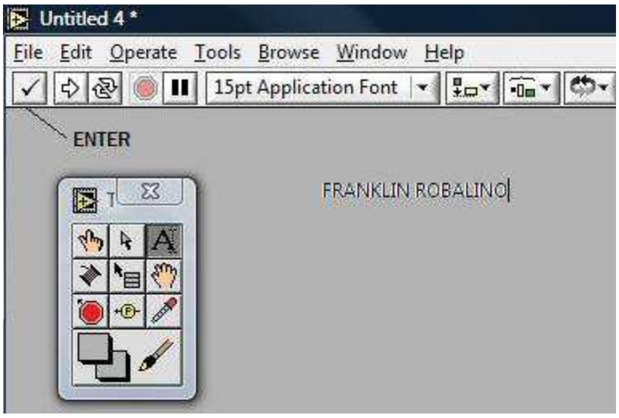
Algunos objetos sólo se reajustan de tamaño horizontalmente o verticalmente, o mantienen la misma proporción al reajustar su tamaño, como por ejemplo una perilla mantendrá su proporción circular al ser reajustada de tamaño.

Membretando Objetos

Los membretes son cuadros de texto de los componentes del panel frontal y diagrama de bloques. Existen dos tipos de membretes: "*free labels*" (membretes libres) y "*owned labels*" (membretes personales). Los membretes personales pertenecen y se mueven con un objeto en particular y describen a ese único objeto. Se pueden esconder estos membretes pero no pueden ser borrados o copiados independientemente de su objeto. Por otro lado los membretes libres no pertenecen a ningún objeto, y por lo tanto pueden ser creados, movidos, copiados o borrados independientemente. Estos pueden ser usados para hacer anotaciones tanto en el panel frontal como en el diagrama de bloques.

Se usa la herramienta "*Labeling*" de la paleta de herramientas (*tools palette*) para crear membretes libres, o para editar cualquier tipo de membretes. Para crear un membrete libre, se escoge la herramienta "*Labeling*", y se hace un click en cualquier lugar de la pantalla donde se desee ubicar el texto, al hacer esto aparecerá un cuadro en donde se puede escribir el texto, como se muestra en la figura 3.46.

Figura 3.46.- Membretando los objetos



Fuente: Franklin Robalino

Luego de escribir el texto, se hace click en el botón "*Enter*" ubicado en la barra de herramientas (ver figura 3.46), el cual aparecerá como recordatorio de que el texto aun esta activo. También se puede finalizar el texto presionando <*Enter*> en el teclado numérico. Hay que tener en cuenta que si no se escribe ningún texto en el membrete, este desaparecerá inmediatamente al hacer click en otro lugar de la pantalla.

Al añadir un control o un indicador en el panel frontal, un membrete personal aparecerá automáticamente. El membrete en este caso está ya listo para recibir texto. Si por alguna razón no se escribe texto, el membrete de default se mantendrá. Para crear un membrete personal para un objeto ya existente, se hace un "*pop up*" en el objeto deseado y se selecciona "Visible Items>>Label". Hay que tener en cuenta que si se ubica un membrete o algún otro objeto encima (cubriendo) un control o un indicador, hace que se vuelva lenta la pantalla y puede ocasionar que el control o indicador se ponga a operar de forma intermitente. Para que esto no ocurra hay que evitar sobreponer objetos o membretes en el panel frontal.

Se puede copiar el texto de un membrete al hacer doble click en el texto con la herramienta de "*Labeling*", o arrastrando la herramienta de "*Labeling*" a través del texto para resaltar el texto deseado. Una vez que el texto deseado este seleccionado, hay que escoger "Edit>>Copy". Una vez copiado el texto se puede resaltar el texto de otro membrete y seleccionar "Edit>>Paste>> para reemplazar el texto. Para crear un nuevo membrete con el texto que ya fue copiado, hay que hacer click en cualquier lugar deseado de la pantalla usando la herramienta "*Labeling*" y seleccionar "Edit>>Paste".

La técnica de reajuste de tamaño también funciona para los membretes. Se puede reajustar el tamaño de los membretes al igual que se reajustan los otros objetos con el uso del cursor de reajuste de tamaño. Los membretes por lo general se ajustan automáticamente de tamaño de tal manera que el texto escrito calce en él. Si por alguna razón no se desea que el membrete se reajuste automáticamente para que calce el texto, se debe hacer un "pop up" en el membrete y seleccionar el "*Size to text*" para que cambie a la posición de "*off*". Hay que tener en cuenta que el texto ingresado se mantendrá en una sola línea a menos que se ponga un <*Enter*>. Por default, la tecla <*Enter*> esta seteada para añadir una línea extra, pero se puede cambiar su funcionamiento dentro de "Tools>>Options>>Front Panel" para que al aplastar <*Enter*> se termine la opción de ingresar texto en el membrete.

Cambiando el Font, Estilo y Tamaño de Texto

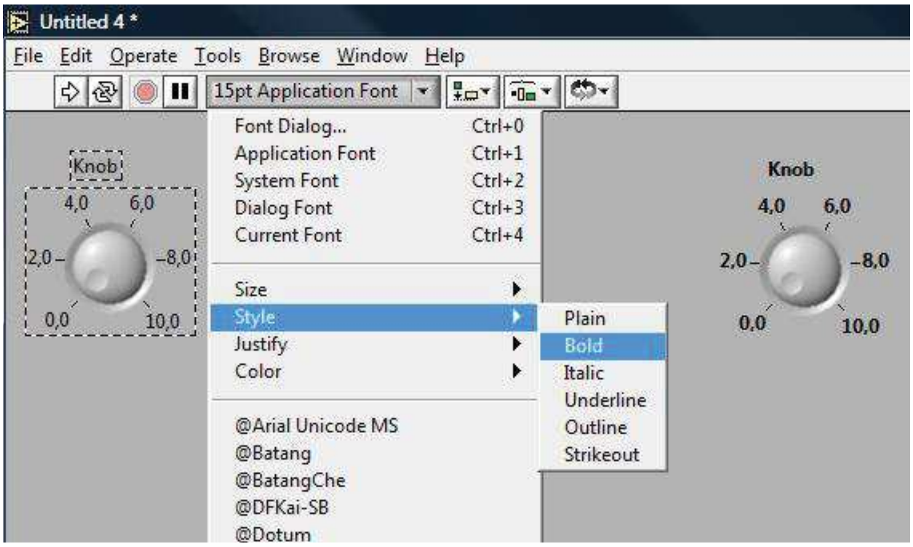
Al utilizar el "*Font Ring*" que está ubicado en la barra de herramientas, se puede cambiar el Font, estilo, tamaño y alineamiento de cualquier texto en un membrete o dentro de los *displays* de los controles e indicadores. Algunos controles e indicadores muestran el texto en diferentes lugares del objeto, por ejemplo, en los indicadores de gráficos, los marcadores de escalas de los ejes están compuestos de varios números, uno por cada marcador en el eje. Se tiene la posibilidad de modificar cada texto (número) independientemente.

Dentro del menú del "*Font Ring*" tenemos las opciones de "*Application*", "*System*", "*Dialog*", y "*Current*".

- El *Font* "*Application*" es la fuente de *default*, y es usada en las paletas de control y de funciones.
- El *Font* "*System*" es la fuente usada en los menús.
- El *Font* "*Dialog*" es la fuente usada en los textos de los cuadros de dialogo.
- El *Font* "*Current*" se refiere a la última fuente usada.

El "*Font ring*" tiene opciones de tamaño, estilo, justificación, y color. Estos se aplican simultáneamente en todos los objetos seleccionados. Por ejemplo, si se selecciona un nuevo *Font* mientras esta seleccionado el membrete y las escalas de una perilla, todas cambiarán al nuevo *Font*, como se muestra en la figura 3.47.

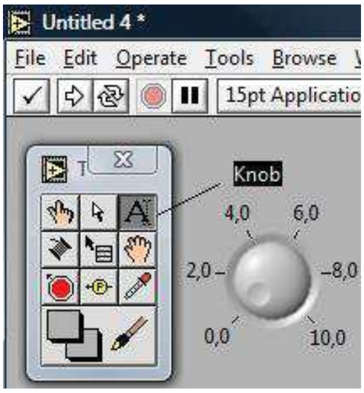
Figura 3.47.- Font


Untitled 4 *
File Edit Operate Tools Browse Window Help
15pt Application Font
Font Dialog... Ctrl+0
Application Font Ctrl+1
System Font Ctrl+2
Dialog Font Ctrl+3
Current Font Ctrl+4
Size
Style
Justify
Color
@Arial Unicode MS
@Batang
@BatangChe
@DFKai-SB
@Dotum
Plain
Bold
Italic
Underline
Outline
Strikeout
Knob
4,0 6,0
2,0 8,0
0,0 10,0
Knob
4,0 6,0
2,0 8,0
0,0 10,0

Fuente: Franklin Robalino

Si se selecciona cualquier objeto o texto y se hace una selección desde el menú “*Font ring*”, el cambio se aplicará a todo lo seleccionado. El proceso para seleccionar sólo el texto de un membrete personal (*owned label*), se muestra en la figura 3.48. Una vez que se haya seleccionado el texto deseado con la herramienta “*Labeling*”, se puede realizar cualquier cambio al texto por medio de las opciones del menú “*Font ring*”. Si por alguna razón ningún texto es seleccionado, los cambios en Font se realizan al Font de default, es decir que cualquier texto creado después tendrá el nuevo Font.

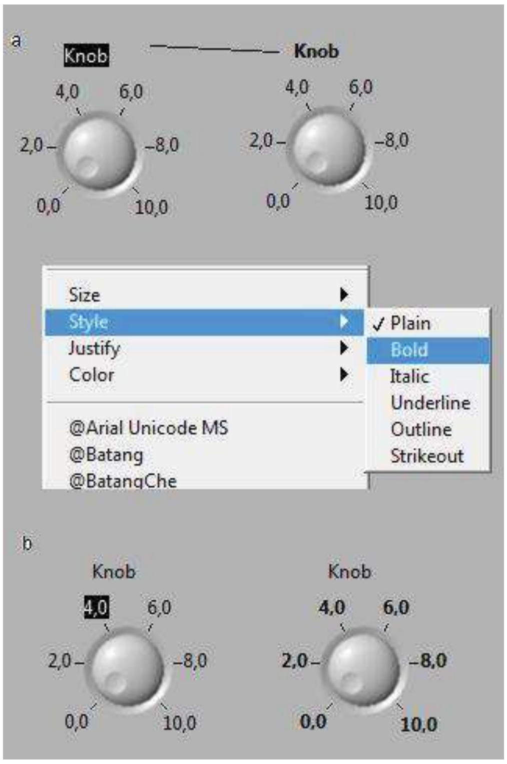
Figura 3.48.- Font


Untitled 4 *
File Edit Operate Tools Browse
15pt Applicatio
Knob
4,0 6,0
2,0 8,0
0,0 10,0

Fuente: Franklin Robalino

Cuando se trabaja con objetos que tienen bastantes piezas de texto hay que tener en cuenta que los cambios en el texto afectarán sólo a los textos seleccionados del objeto. Es decir si se selecciona todo el objeto y se cambia el *Font*, todo el texto dentro del objeto cambiará (ver figura 3.47). Mientras que si se selecciona sólo el membrete y luego se cambia el *Font*, sólo cambiará el membrete, como se puede observar en la figura 3.49a. De igual manera si se selecciona sólo el texto de la escala y luego se cambia el *Font*, sólo cambiará los atributos de la escala, como se puede observar en la figura 3.49b.

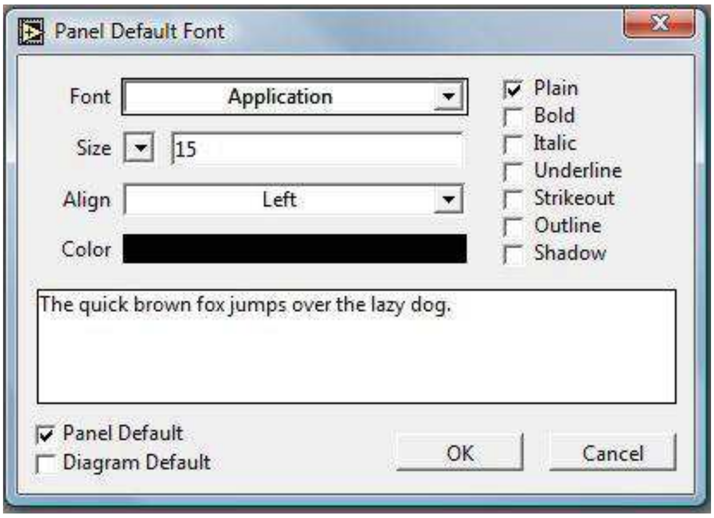
Figura 3.49 a y b.- Font


a
Knob
Knob
4,0 6,0
2,0 8,0
0,0 10,0
Size
Style
Justify
Color
@Arial Unicode MS
@Batang
@BatangChe
✓ Plain
Bold
Italic
Underline
Outline
Strikeout
b
Knob
Knob
4,0 6,0
2,0 8,0
0,0 10,0

Fuente: Franklin Robalino

Si se selecciona “*Font Dialog*…” dentro del menú “*Font ring*” cuando está activo el panel frontal, aparecerá un cuadro de diálogo, como se muestra en la figura 3.50. Dentro de este cuadro se puede seleccionar si los cambios se aplican al panel frontal con la opción "*Panel Default*", al diagrama de bloques con la opción "*Diagram Default*", o a ambos seleccionando los dos. Si el diagrama de bloques está activo aparecerá el mismo diálogo pero con la opción "*Diagram Default*" ya seleccionada. Esto es muy útil ya que se puede seleccionar diferentes tipos de *Font* para el panel frontal y para el diagrama de bloques, es decir se puede tener un Font grande para el panel frontal y un pequeño para el diagrama de bloques.

Figura 3.50.- Panel Default Font



Panel Default Font

Font: Application

Size: 15

Align: Left

Color: ■

☑ Plain
☐ Bold
☐ Italic
☐ Underline
☐ Strikeout
☐ Outline
☐ Shadow

The quick brown fox jumps over the lazy dog.

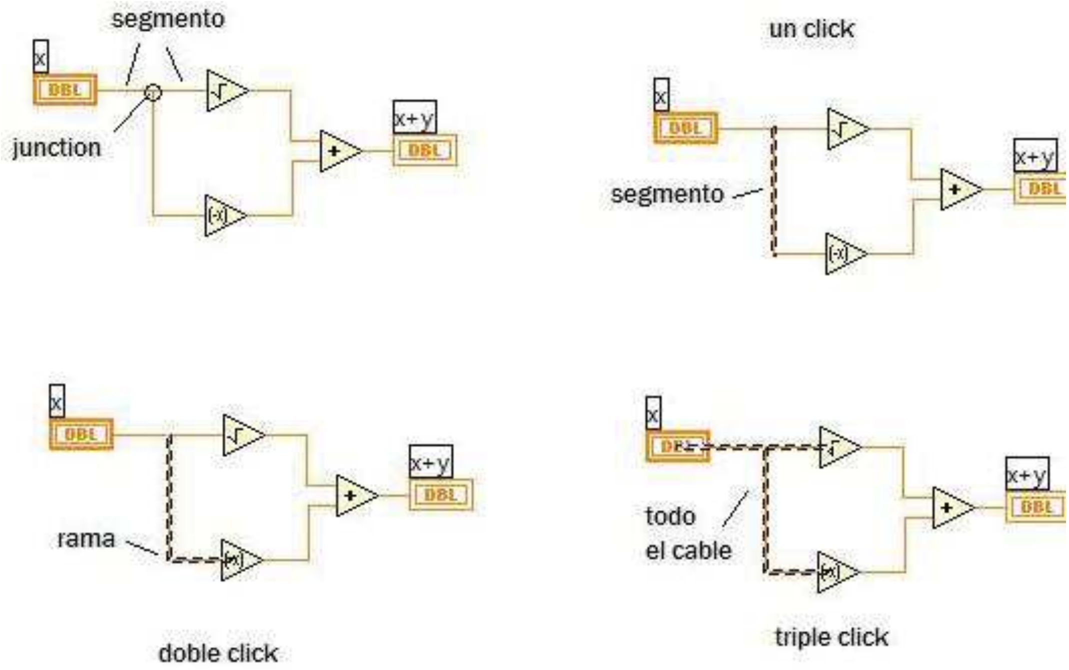
☑ Panel Default
☐ Diagram Default

Fuente: Franklin Robalino

Seleccionando y Borrando Cables

Un pedazo de cable ya sea de forma horizontal o vertical se llama "*wire segment*" (segmento de cable). El punto donde se juntan dos o más segmentos de cable se denomina "*junction*" (juntura). Un "*wire branch*" (rama de cable) contiene a todos los segmentos de cable entre dos junturas, entre una juntura y un terminal, o entre un terminal y otro si es que no existen junturas entre terminales. Para seleccionar un segmento de cable se hace un click en el segmento con la herramienta "*Positioning*", si se hace doble click se seleccionará una rama, y si se hace click tres veces se seleccionará todo el cable, como se puede observar en la figura 3.51.

Figura 3.51.- Seleccionando cables



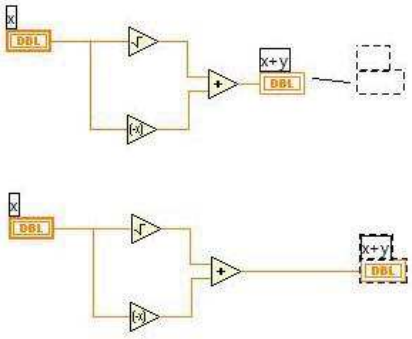
Fuente: Franklin Robalino

Alargando Cables y Cables Rotos

Los objetos que ya están conectados por medio de cables pueden ser reposicionados individualmente o en grupos al seleccionarlos y arrastrarlos hacia el lugar deseado con la herramienta "*Positioning*". Los cables que conectan a los objetos se alargarán automáticamente. Si se desea mover objetos de un diagrama de bloques a otro diagrama de bloques, los cables conectores no se moverán con el objeto, al menos que los cables también sean seleccionados.

Un ejemplo de cómo alargar un cable se muestra en la figura 3.52. En este ejemplo, el objeto seleccionado que en este caso es un indicador numérico, es seleccionado con la herramienta "*Positioning*" y arrastrado a un nuevo sitio, provocando que el cable se alargue.

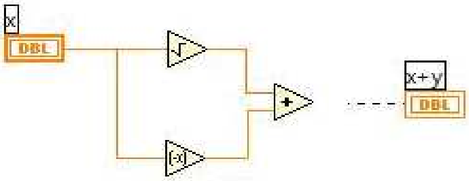
Figura 3.52.- Alargar cables



Fuente: Franklin Robalino

Algunas veces al querer alargar los cables, provoca que estos se desconecten, debido a que no se seleccionó todo el cable. El VI no se ejecutará si es que existen cables desconectados. Los cables desconectados también llamados cables rotos se muestran como líneas segmentadas, ver figura 3.53. Los cables desconectados pueden ser fácilmente removidos del diagrama de bloques seleccionando “Edit>>Remove Broken Wires”.

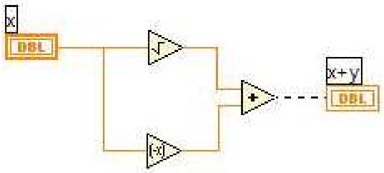
Figura 3.53.- Cable roto



Fuente: Franklin Robalino

Al hacer una mala conexión, un cable roto aparecerá, como se muestra en la figura 3.54. Existen muchas razones por las cuales se puede generar una mala conexión dentro de la programación gráfica (*G programming*). Un error común es conectar el terminal de un indicador al terminal de otro indicador, o al conectar un control a un indicador cuando sus tipos de datos son distintos, por ejemplo conectando a un indicador numérico un control Booleano. Para remover los cables rotos se los selecciona con la herramienta de "*Positioning*" y se presiona la tecla <*Delete*> del teclado para eliminarlos. Si se desea borrar todos los cables rotos que existan en el diagrama de bloques, se selecciona "Edit>>Remove Broken Wires".

Figura 3.54.- Cable roto

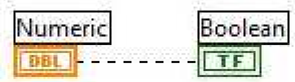


Franklin Robalino

Existen varios tipos de errores que causarán que aparezcan cables rotos, algunos ejemplos son:

Conflicto por el tipo de cable, dimensión, unidad, o elementos.- un conflicto con el tipo de cable ocurre cuando se intenta conectar dos objetos que manejan distintos tipos de datos, como se mencionó anteriormente en el caso de indicador numérico y control Booleano. Este caso se puede observar en la figura 3.55.

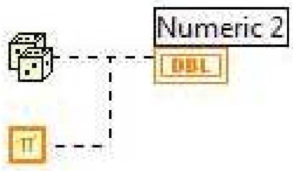
Figura 3.55.- Conflicto por el tipo de cable



Fuente: Franklin Robalino

Múltiples fuentes de cable.- se puede conectar un control a múltiples destinos de *output* (indicadores), pero no hay como conectar múltiples fuentes de datos (controles) a un solo destino (indicador). En la figura 3.56 se muestra la conexión de dos fuentes o controles (en este caso un valor aleatorio y una constante “π”) a un indicador. Esto produce que se genere un cable roto el cual debe ser compuesto al desconectar el valor aleatorio (representado por los dados) o desconectando la constante “π”. Otro ejemplo de “múltiples fuentes de cable” ocurre durante la construcción del panel frontal, cuando al poner un indicador numérico en el panel frontal por equivocación se coloca un control numérico. Al hacer esto y conectarlo a un valor de output, se generara un error de múltiples fuentes. La manera más fácil de arreglar esto es haciendo un “*pop up*” en el terminal del control y seleccionar “*Change to Indicator*”.

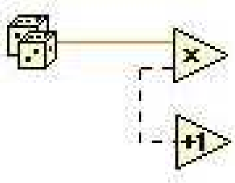
Figura 3.56.- Múltiples fuentes de cable



Fuente: Franklin Robalino

Cable sin fuente.- un ejemplo de un cable sin una fuente se muestra en la figura 3.57. Donde para la multiplicación existe un valor aleatorio en la una entrada y en la otra no se tiene un valor aceptable.

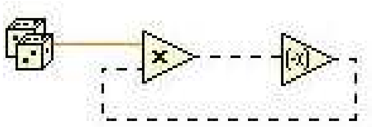
Figura 3.57.- Cable sin fuente



Fuente: Franklin Robalino

Ciclo de cables.- los cables no deben de producir ciclos cerrados entre íconos o estructuras, debido a que cada nodo no se ejecutará mientras no tenga toda la información necesaria, como se muestra en la figura 3.58.

Figura 3.58.- Ciclo de cables

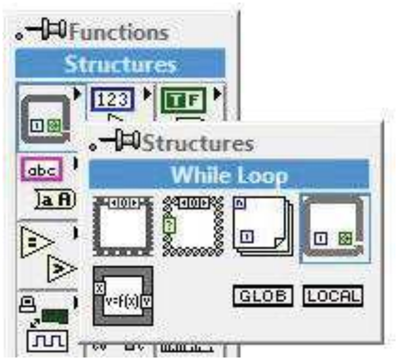


Fuente: Franklin Robalino

3.12 *WHILE LOOP*

El *While Loop* controla operaciones repetitivas en un VI mientras una condición específica sea verdadera, es decir que se detendrá cuando la condición sea falsa. También se puede cambiar para que se detenga sólo si la condición es verdadera y se repita mientras es falsa. El ícono del *While Loop* se encuentra dentro de "funciones-estructuras", como se muestra en la figura 3.59.

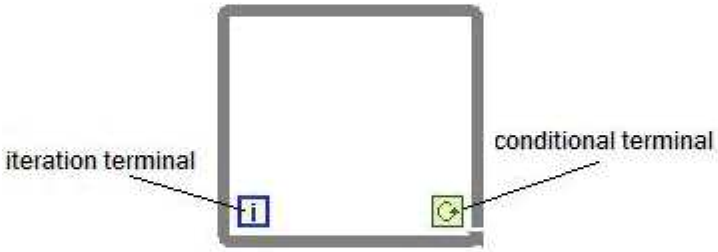
Figura 3.59.- While Loop



Fuente: Franklin Robalino

Como se mencionó anteriormente el *While Loop* es una estructura que repite una sección de código mientras que la condición no cambie. Este se puede comparar con un "*Do Loop*" o un "*Repeat Until Loop*" que son usados en lenguajes de programación tradicional. El *While Loop* ejecutará el código que se encuentre dentro de sus bordes mientras que una condición sea satisfecha. El *While Loop* posee dos terminales: el "*conditional terminal*" (terminal condicional) que es un input y el "*iteration terminal*" que es un output, como se muestra en la figura 3.60.

Figura 3.60.- Componentes de un While Loop



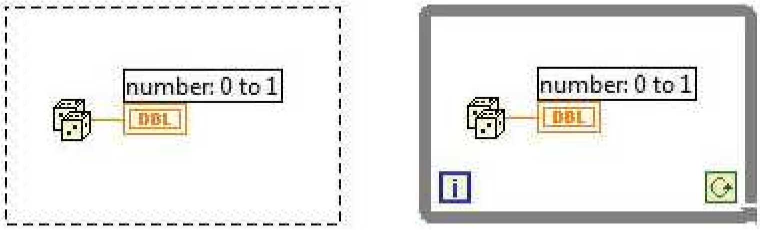
Fuente: Franklin Robalino

El "*iteration terminal*" es un terminal de output numérico el cual muestra el número de veces que el *loop* (ciclo) ha sido ejecutado (este comienza desde el valor cero, es decir 0, 1, 2, etc.). Mientras que por el "*conditional terminal*" ingresa una variable Booleana: verdadero o falso. El *While Loop* se ejecutará hasta que el valor Booleano que está conectado al terminal condicional sea falso.

El VI chequea el "*conditional terminal*" al finalizar el ciclo, por esta razón, el *While Loop* siempre se ejecutará al menos una vez. Si el valor en el "*conditional terminal*" es verdadero, se ejecutara nuevamente el ciclo, y si es falso se detendrá la ejecución del *While Loop*. Como valor default el "*conditional terminal*" será falso, por esta razón si se le deja desconectado al "*conditional terminal*" el ciclo se ejecutará una sola vez.

Para usar el *While Loop* se ingresa a “funciones-estructuras” en el panel frontal, al seleccionar el *While Loop* este no aparece en el diagrama de bloques sino que aparecerá un ícono que lo representa, este ícono permitirá darle un porte adecuado al *While Loop* y posicionarlo en el lugar deseado. Para hacer esto debemos primero hacer click en el *While Loop*, una vez que aparezca el ícono, se debe colocar el ícono en la parte superior izquierda del código que se desee ejecutar con el *Loop*, como se muestra en la figura 3.61. Luego mientras se mantiene presionado el botón del mouse, se dibuja un rectángulo de tal manera que los objetos deseados queden dentro de éste. Al soltar el botón del mouse el *While Loop* será creado en ese espacio. Otra manera es creando el *loop* primero y después insertar objetos.

Figura 3.61.- Insertando un While Loop

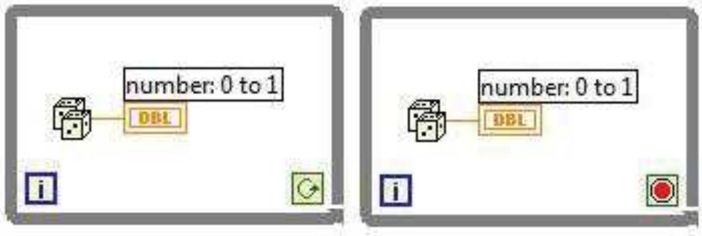


Fuente: Franklin Robalino

Una vez que el *While Loop* esté ubicado en el diagrama de bloques, puede ser reajustado de tamaño por medio de la herramienta “*positioning*” al agarrar una esquina del *loop* y reubicarla, ajustándola a una dimensión deseada. Elementos adicionales pueden ser añadidos al *While Loop* una vez que esté ya creado. Los bordes del *While Loop* serán resaltados cada vez que un nuevo objeto ingrese al *loop* para asegurar al usuario que está adentro y no fuera de este. Hay que tener en cuenta que una vez creado el *While Loop* este no puede ser arrastrado hacia otro objeto para que el objeto ingrese al *loop*, al hacer esto causará que el objeto quede bajo el *loop*, la manera correcta es la de arrastrar el objeto dentro del *While loop*.

Se puede cambiar la manera en que el "*conditional terminal*" funciona, al hacer click derecho en el *While Loop* y seleccionando "*Stop if True*" (parar si es verdadero), entonces en vez de que el *While Loop* continúe cuando es verdadero, se detendrá cuando el valor es verdadero. Al hacer esto el ícono del "*conditional terminal*" cambiará de símbolo, como se muestra en la figura 3.62.

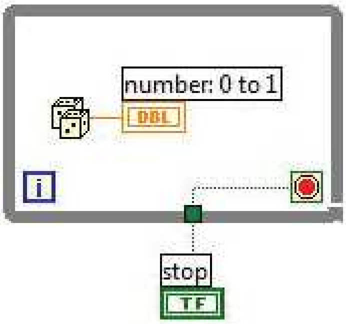
Figura 3.62.- Terminal condicional "Stop if True"



Fuente: Franklin Robalino

Hay que tener en cuenta que al colocar el control Booleano del "*conditional terminal*" fuera del *While Loop*, como se muestra en la figura 3.63, se crea ya sea un *loop* infinito, o un *loop* que se ejecutará una sola vez, dependiendo en el valor inicial. Esto sucede debido a que en la programación gráfica (*G programing*) el valor Booleano es leído antes que entre al *While Loop*, y no será leído dentro del *loop* una vez que este sea ejecutado. Si por alguna razón el control esta fuera del *loop*, en los bordes del *loop* aparecerá un rectángulo verde llamado túnel, este indica que hay una conexión fuera del *loop*. La única manera de detener un *loop* infinito es abortando la ejecución, mediante el botón "*Abort Execution*" en el panel frontal.

Figura 3.63.- Control condicional fuera del While Loop

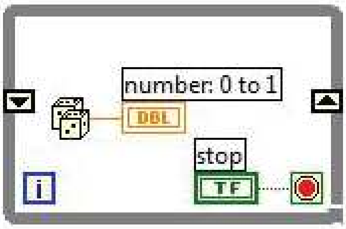


Fuente: Franklin Robalino

3.12.1. *SHIFT REGISTERS*

Los *Shift Registers* transfieren valores de un ciclo del *While Loop* al próximo ciclo. Los *Shift Registers* se ilustran como un par de terminales directamente opuestos el uno al otro en los lados verticales del borde del *While Loop*, como se muestra en la figura 3.64. Para crear los *Shift Registers* se hace click izquierdo en el borde derecho o izquierdo del *loop* y se selecciona "*Add Shift Register*" del menú.

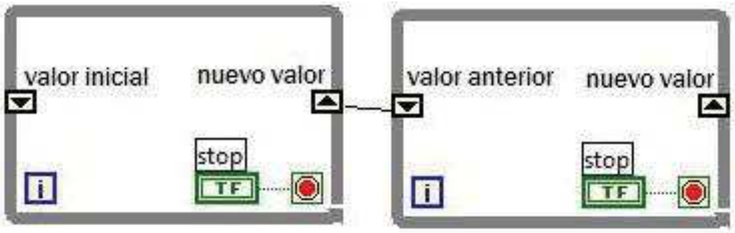
Figura 3.64.- Shift registers



Fuente: Franklin Robalino

El terminal de la derecha, es decir el rectángulo con la flecha para arriba, guarda los datos cada vez que el ciclo termina. Los datos guardados del ciclo anterior son transferidos al terminal izquierdo, es decir al rectángulo con la flecha para abajo, y estos datos aparecen cuando comienza el siguiente ciclo, como se muestra en la figura 3.65. Un *shift register* puede retener cualquier tipo de datos, incluyendo numéricos, *Boolean, strings*, y *arrays*, pero los datos conectados a los terminales de cada *shift register* deben ser del mismo tipo para que no ocurra un error.

Figura 3.65.- Pasando datos con Shift registers

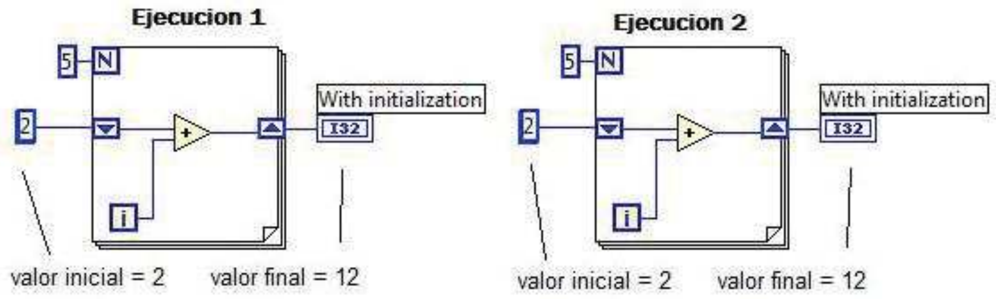


Fuente: Franklin Robalino

INICIALIZANDO *SHIFT REGISTERS*

Los *Shift Registers* son inicializados al conectar desde afuera una constante o control en el terminal izquierdo del *shift register*. La inicialización de un *shift register* se muestra en la figura 3.66. Como se puede observar, en la primera ejecución el valor final del *shift register* es 12, debido a que el valor final es una suma de los ciclos ejecutados (0+1+2+3+4) más el valor inicial que es igual a 2. En la segunda ejecución del código que contiene el valor de inicio del *shift register*, el resultado es exactamente el mismo que el de la primera ejecución. De hecho, los resultados en todas las ejecuciones después serán los mismos, es decir darán un resultado de 12. Esto quiere decir que el valor final del *shift register* en la primera ejecución no toma importancia en la segunda vez que se ejecute el programa. Hay que tomar en cuenta que en el ejemplo se está usando un *For loop*, el cual es similar al *While Loop* pero este se repite solo un número de veces preestablecido en el terminal con el símbolo “N”. Es decir en este caso el *loop* se repite cinco veces, y no depende de un terminal condicional.

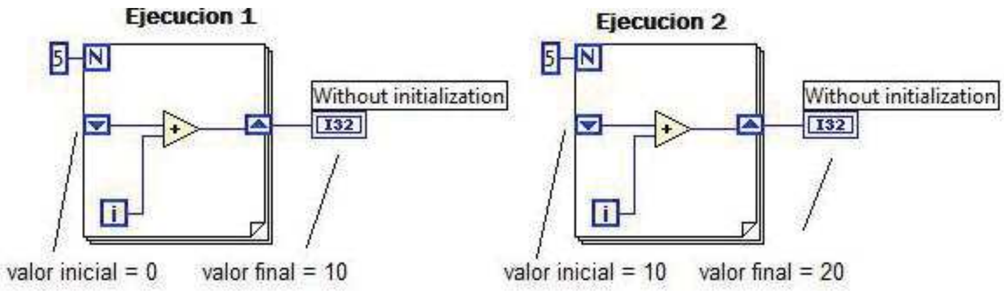
Figura 3.66.- Inicilizando Shift registers



Fuente: Franklin Robalino

En otro caso similar, si el *shift register* no fuera inicializado, la primera vez que el VI sea ejecutado lo hará con el valor default del tipo de dato que se encuentre dentro del *shift register*, es decir si el tipo de dato del *shift register* es booleano, el valor inicial será “falso”, o si el tipo de dato es numérico, el valor inicial será el número “cero”. La figura 3.67 ilustra lo que pasaría si se ejecuta el código del *loop* dos veces sin inicializar los *shift registers*. En la primera ejecución, el valor final del *shift register* es 10, debido a que el valor inicial de default en el *shift register* es cero. Al ejecutar el código por segunda vez sin cerrar el VI, el resultado final del *shift register* será un valor de 20. Este resultado se debe al hecho que en la segunda ejecución del código, el valor inicial del *shift register* ya no es cero, sino que ahora es igual a 10, que fue el valor final de la ejecución previa, y este ahora es sumado a la segunda ejecución. Cuando el *shift register* no es inicializado, el valor final será acumulado para la siguiente ejecución, por esta razón es preferible inicializar los *shift register* para no obtener resultados diferentes cada vez que se corra el mismo código.

Figura 3.67.- Shift registers sin inicialización



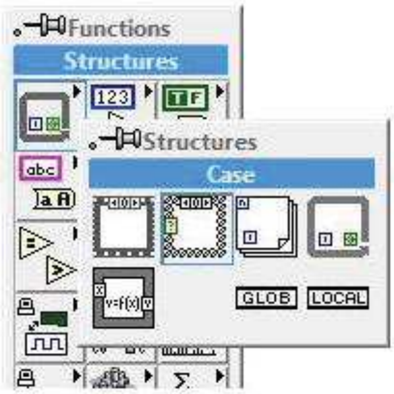
Fuente: Franklin Robalino

Hay que tener en cuenta que los valores de los datos guardados en los *shift registers* son guardados sólo hasta que el programa sea cerrado y luego los valores serán borrados de la memoria, esto quiere decir que si se tiene un VI con los *shift registers* sin inicializar, el valor inicial será primero el de default y luego acumulativo, mientras no se cierre el programa. Una vez cerrado el programa la próxima vez que se abra los *shift register* volverán al valor default (si es que no están inicializados).

3.13. *CASE STRUCTURES*

El *Case Structure* (estructuras tipo casilla) es un método para ejecutar texto condicional, y es parecido al *If…Then…Else* de los lenguajes de programación convencionales basados en texto. Para usar el *Case Structure*, se debe abrir desde "funciones-estructuras", como se muestra en la figura 3.68. Se coloca el *Case Structure* en el diagrama de bloques, y al igual que el *While Loop* que fue mencionado anteriormente, este no aparece en el diagrama de bloques, sino que aparece un ícono que lo representa, el cual debe ser reajustado de tamaño de tal manera que los objetos deseados queden dentro de los bordes del *Case Structure* (al igual que con el *While Loop*), o a su vez se puede colocar primero el *Case Structure* en el diagrama de bloques, luego reajustar su tamaño por medio de la herramienta "*positioning*" al agarrar una esquina del *loop* y reubicarla, ajustándola a una dimensión deseada, y luego se puede insertar los objetos deseados dentro del *Case Structure*.

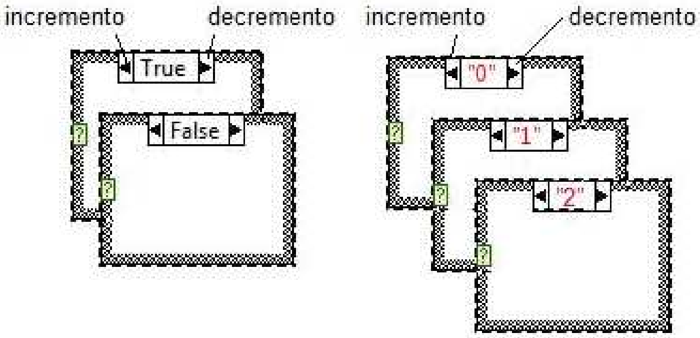
Figura 3.68.- Case Structures



Fuente: Franklin Robalino

El *Case Structure* (estructuras tipo casilla) puede tener múltiples subdiagramas de códigos dentro de su estructura, dichos subdiagramas están configurados uno sobre otro de tal manera que sólo uno puede ser visible a la vez (cada subdiagrama dentro de una casilla). En la parte superior del *Case Structure* dentro del borde, se encuentra el identificador de diagramas (*selector label*), este identificador de diagramas puede ser numérico, Booleano, *string*, o un tipo de control, y son seleccionados mediante los botones de incremento y descenso (flechas a la izquierda y derecha) como se muestra en la figura 3.69. El identificador de diagramas muestra los valores requeridos que causaran la ejecución del subdiagrama correspondiente. Para ver los subdiagramas que se encuentran dentro del *Case Structure*, se lo hace mediante los botones de incremento y descenso (flechas a la izquierda y derecha) que se encuentran en el borde superior del *Case Structure*. Mediante estas flechas podemos navegar entre los distintos subdiagramas que se encuentran dentro de las casillas del *Case Structure*, es decir al aplastar el botón de decremento observaremos el subdiagrama anterior y al aplastar el botón de incremento observaremos el subdiagrama siguiente.

Figura 3.69.- incremento y decremento



Fuente: Franklin Robalino

Para poder seleccionar el subdiagrama de código que va ha ser ejecutado, el *Case Structure* utiliza un control llamado "*selector terminal*" (terminal de selección). Dicho terminal como su nombre lo indica, seleccionará la casilla con el subdiagrama a ser ejecutado. El "*selector terminal*" puede ubicarse en cualquier lugar del borde del *Case Stucture*. La nomenclatura del selector automáticamente se ajusta al tipo de datos que son ingresados (*input data*). Por ejemplo, si se cambia el valor que está conectado al selector, cambiándolo de un valor numérico a un valor Booleano, las casillas numeradas "0" y "1" cambiarán a "falso" y "verdadero", respectivamente. Al cambiar el tipo de dato conectado al "*selector terminal*" hay que tomar en cuenta, que si tenemos un *Case Structure* que originalmente recibía un input numérico y con un numero de casilleros *n*, los valores 0, 1, 2,…, *n* pueden existir en el código, y al cambiar el input del "*selector terminal*" de tipo numérico a Booleano, los casilleros "0" y "1" cambiarán automáticamente a "falso" y "verdadero", pero los casilleros 2, 3,…, *n* no cambiarán, por lo tanto deben ser borrados para que el programa pueda ejecutarse correctamente.

También se puede escribir y editar los valores directamente en el identificador de diagramas (*selector label*) usando el "*labeling tool*". Los valores del selector pueden ser de un único valor, como una lista, o un rango de valores. Una lista de valores es separada por comas, como por ejemplo -1, 0, 1, 2, 3. Un rango se escribiría de la siguiente manera 10..20, lo cual indica todos los números enteros del 10 al 20. También se puede usar rangos "*open-ended*", por ejemplo, todos los números menores o iguales que 0 son representados como ..0. similarmente 100.. representa todo número mayor o igual que 100. El case selector también puede usar valores *string* que se mostrarán entre comillas, como por ejemplo "rojo" y "azul". En este caso no es necesario escribir los valores entre comillas.

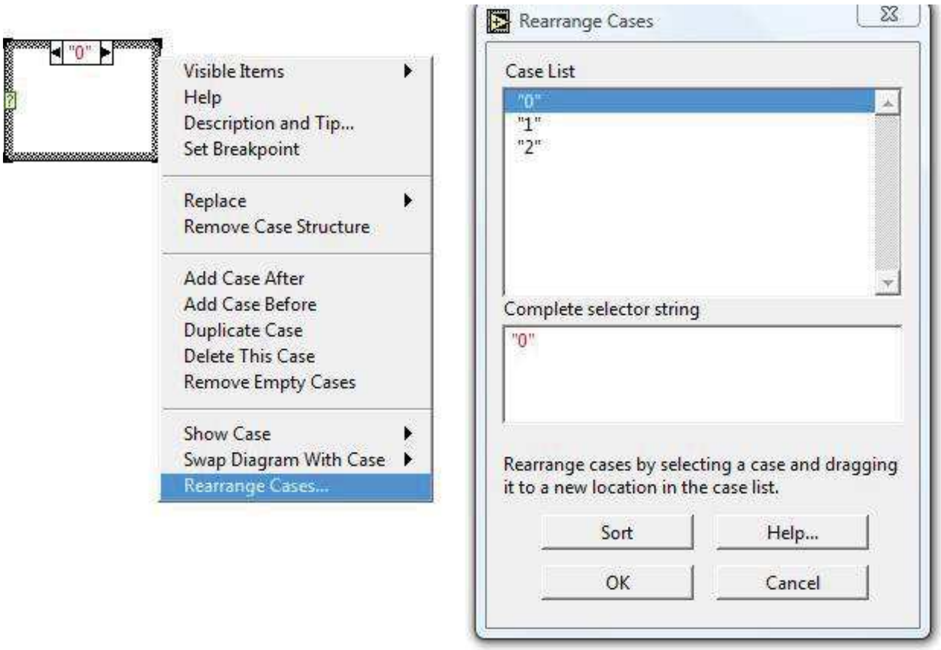
Si se ingresa en el selector un valor que no es del mismo tipo que el objeto que está conectado al selector terminal, entonces el valor que se mostrará en el selector estará de color rojo, lo que significa que el VI tiene un error de conexión.

3.13.1. AÑADIENDO Y BORRANDO CASILLAS

Para añadir o borrar casillas en el *Case Structure*, se debe hacer un click derecho en el borde del *Case Structure*, lo cual nos desplegará un menú de opciones como se muestra en la figura 3.70. En este menú se puede seleccionar "*Add Case After*" para añadir un casillero después del casillero que está visible al momento. "*Add Case Before*" para añadir un casillero antes del casillero que está visible al momento. "*Delete This Case*" para borrar el casillero que está visible al momento. "*Duplicate Case*" para duplicar el casillero que esta visible al momento. Cuando se añade o borra casilleros (subdiagramas) en un *case structure*, los identificadores de diagramas (*selector label*) son automáticamente cambiados para reflejar el contenido insertado o borrado del casillero.

Hay ocasiones en las cuales el orden de los casilleros listados necesita ser reordenado. Por ejemplo, en lugar de tener las casillas listadas 1, 2, 3, 4 se podría necesitar listarlas de la siguiente forma 1, 3, 2, 4 (primero impares y luego pares). Al reordenar los casilleros dentro del *case structure* en el diagrama de bloques no afecta en el tiempo que se demora en ejecutar el *case structure*, sólo es cuestión de preferencias del programador. Para cambiar el orden en el cual están listados los casilleros del *case structure*, se debe hacer un click derecho en el borde del *Case Structure*, lo cual nos desplegará un menú de opciones como se muestra en la figura 3.70. Dentro de este menú se selecionara “*Rearrange Cases*”, lo cual abrirá un cuadro de diálogo. Mediante este cuadro de diálogo se podrá manipular el orden de los casilleros. El botón “*Sort*” clasificara los casilleros basándose en el primer casillero dentro del “*case list*” en el cuadro de diálogo. Para cambiar el puesto del casillero, se selecciona el casillero que se desee mover dentro del “*case list*” (el casillero se resaltara al seleccionarlo) y se lo mueve con el mouse al lugar deseado dentro de la lista.

Figura 3.70.- Cuadro de dialogo


"0"
Visible Items
Help
Description and Tip...
Set Breakpoint
Replace
Remove Case Structure
Add Case After
Add Case Before
Duplicate Case
Delete This Case
Remove Empty Cases
Show Case
Swap Diagram With Case
Rearrange Cases...
Rearrange Cases
Case List
"0"
"1"
"2"
Complete selector string
"0"
Sort
Help...
OK
Cancel

Rearrange cases by selecting a case and dragging it to a new location in the case list.

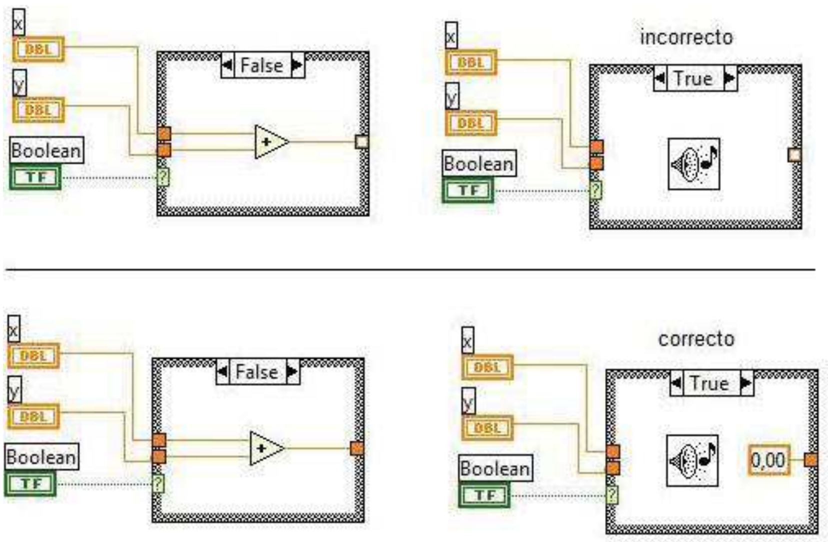
Fuente: Franklin Robalino

La opción “*Make This Default Case*” dentro del menú de opciones que se despliega al hacer click derecho en el borde del *case structure*, sirve para especificar el casillero que debe ser ejecutado si es que el valor que aparece en selector terminal no está listado como una posible opción dentro del *selector label* de los casilleros. Esto quiere decir que si en el *case structure* tenemos casilleros con *selector labels* 0, 1, 2, 3 y en el selector terminal aparece un valor de 5, el casillero que se ejecutará es el de *default*. Los cases en otros lenguajes de programación generalmente no ejecutan ningún casillero si es que el casillero esta fuera de rango, pero en el programa Lab View se necesita incluir un default case para ejecutarse cuando el valor este fuera de rango o listar toda opción posible que pueda aparecer como valor en el input, de esta manera se evitará errores en el programa. Para el casillero default que se escoja, la palabra "*default*" aparecerá en el *selector label* del casillero correspondiente.

3.13.2. CONECTANDO *INPUTS Y OUTPUTS*

Los datos en todos los terminales de *input* (tuneles) están disponibles para todas las casillas del *case structure*. Los casilleros no requieren datos de *input* ni *output*, pero si por alguna razón un casillero requiere un *output*, todos los casilleros deberán tener un *output*. Olvidar esto puede conducir a errores de codificación. Situaciones correctas e incorrectas de conexión se muestran en la figura 3.71. En donde el casillero falso "*FALSE*" tiene dos inputs numéricos añadidos y enviados por el túnel hacia afuera de la estructura, y para el casillero verdadero "*TRUE*" el sistema de la computadora generará un *beep*.

Figura 3.71.- Túneles



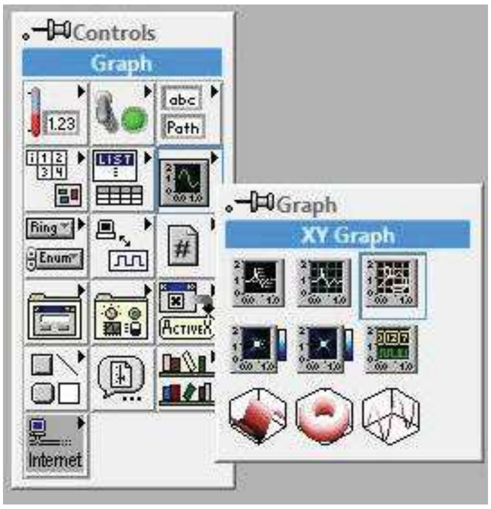
Fuente: Franklin Robalino

Al crear un túnel de *output* en una casilla, los túneles aparecen en la misma posición en los bordes de la estructura en las otras casillas. Se debe definir el túnel de *output* para cada casillero. Los túneles que están desconectados se mostraran como cuadrados blancos, y cuando esto ocurre, aparecerá el botón "*Broken Run*" el cual no dejará que el programa se ejecute, para evitar esto hay que asegurarse que todos los *outputs* estén conectados. Se puede conectar constantes o controles a los casilleros desconectados, al hacer click derecho en el cuadrado blanco y seleccionando "*Constant*" o "*Control*" del menú "*Create*". Cuando todas las casillas envíen datos al túnel, este tomará un color sólido, el color dependerá del tipo de datos que pasen por este, y el botón "*Run*" aparecerá, permitiendo ejecutar el programa.

3.14. GRÁFICOS XY

Los XY *Graphs* son planos cartesianos, usados para mostrar los datos de manera gráfica, y están ubicados dentro de "*controls-Graph-XY Graphs*", como se muestra en la figura 3.72.

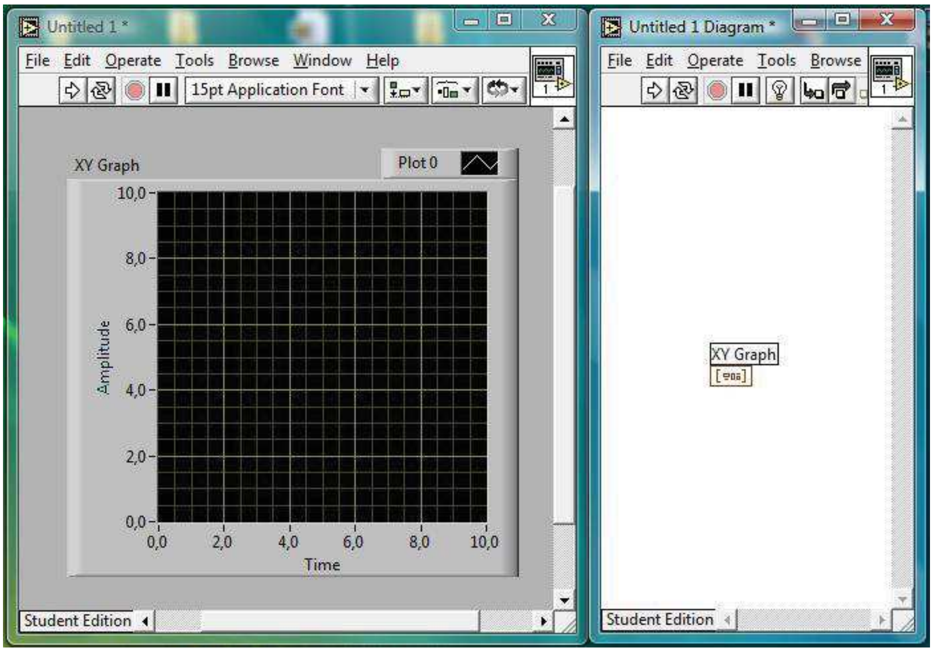
Figura 3.72.- Gráficos XY



Fuente: Franklin Robalino

Este tipo de gráfico es muy útil si es que se quiere graficar datos de intervalos irregulares, cuando los datos son de dos variables dependientes, o para especificar puntos por medio de coordenadas. El *XY Graph* y su correspondiente diagrama de bloques se muestran a continuación en la figura 3.73.

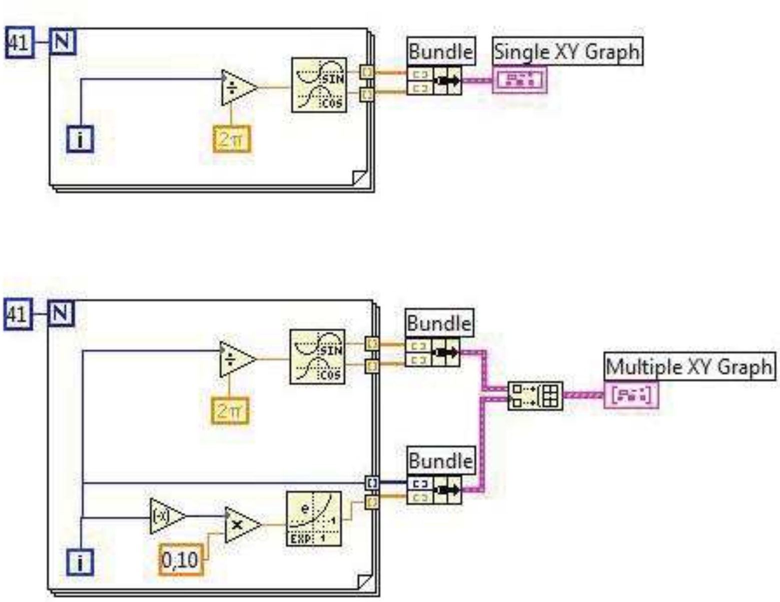
Figura 3.73.- Gráficos XY y diagrama de bloques


Untitled 1 *
File Edit Operate Tools Browse Window Help
15pt Application Font
XY Graph
Plot 0
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0
2,0
4,0
6,0
8,0
10,0
Time
Student Edition
Untitled 1 Diagram *
File Edit Operate Tools Browse
XY Graph
Student Edition

Fuente: Franklin Robalino

La función "*Bundle*" es utilizada para combinar los datos del *array X* con el *array Y* para mandarlos a un "*cluster*" el cual será conectado al *XY Graph*. Para el *XY Graph*, los componentes son un *bundled X array* (el *input* de ariba) y un *bundled Y array* (el *input* de abajo). El terminal del *XY graph* aparecerá en el diagrama de bloques como un indicador de "*cluster*". Para un gráfico múltiple de XY, se necesitará construir un *array* de *clusters* para los valores de X y Y, como se muestra en la figura 3.74.

Figura 3.74.- Gráficos XY y diagrama de bloques



Fuente: Franklin Robalino

PERZONALIZANDO LOS GRAFICOS

Los gráficos tienen componentes de edición que permiten al programador personalizar el gráfico. A continuación se explicará como configurar algunos de los componentes más importantes. Dentro de estos tenemos:

Automatización de escalas en los ejes X y Y del gráfico (*Autoscaling*).

Usando la leyenda del gráfico (*Plot Legend*).

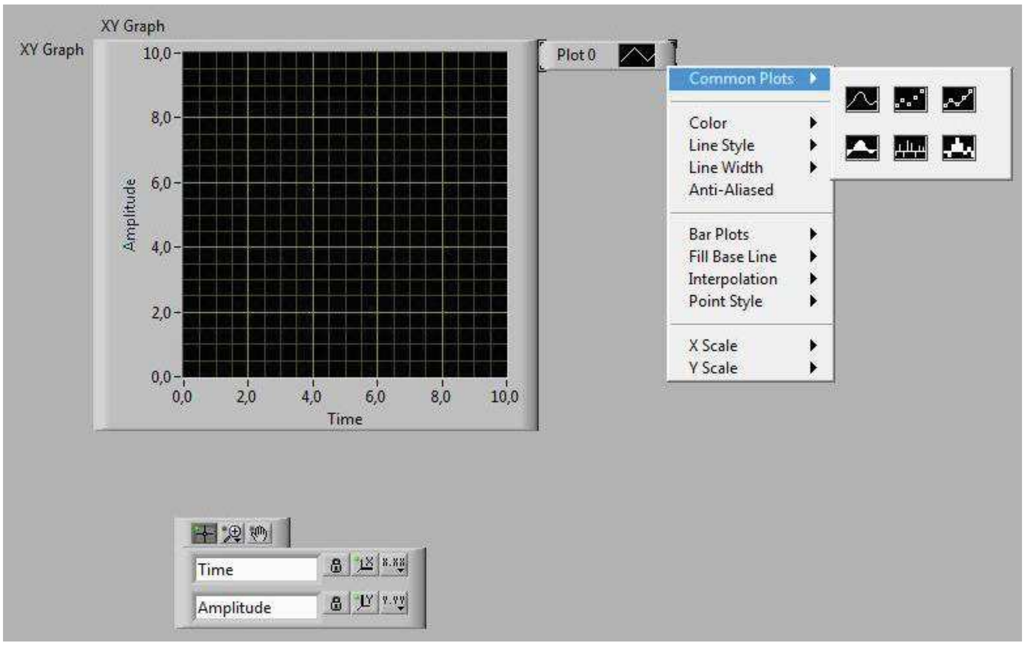
Usando la paleta del gráfico (*Graph Palette*).

Usando la leyenda de la escala (*Scale Legend*).

Componentes de gráfico, incluyendo el “*scrollbar*” y el *display* digital.

Por default, los gráficos tienen el “*Plot Legend*” (la leyenda del gráfico) visible, cuando se les coloca por primera vez en el panel frontal. Usando la herramienta “*Positioning*” (posicionamiento), se puede reubicar y cambiar la escala, la leyenda y sus componentes a cualquier lugar dentro del gráfico. La figura 3.75 muestra algunos de los componentes más importantes del gráfico que pueden ser personalizados.

Figura 3.75.- Personalizar gráficos

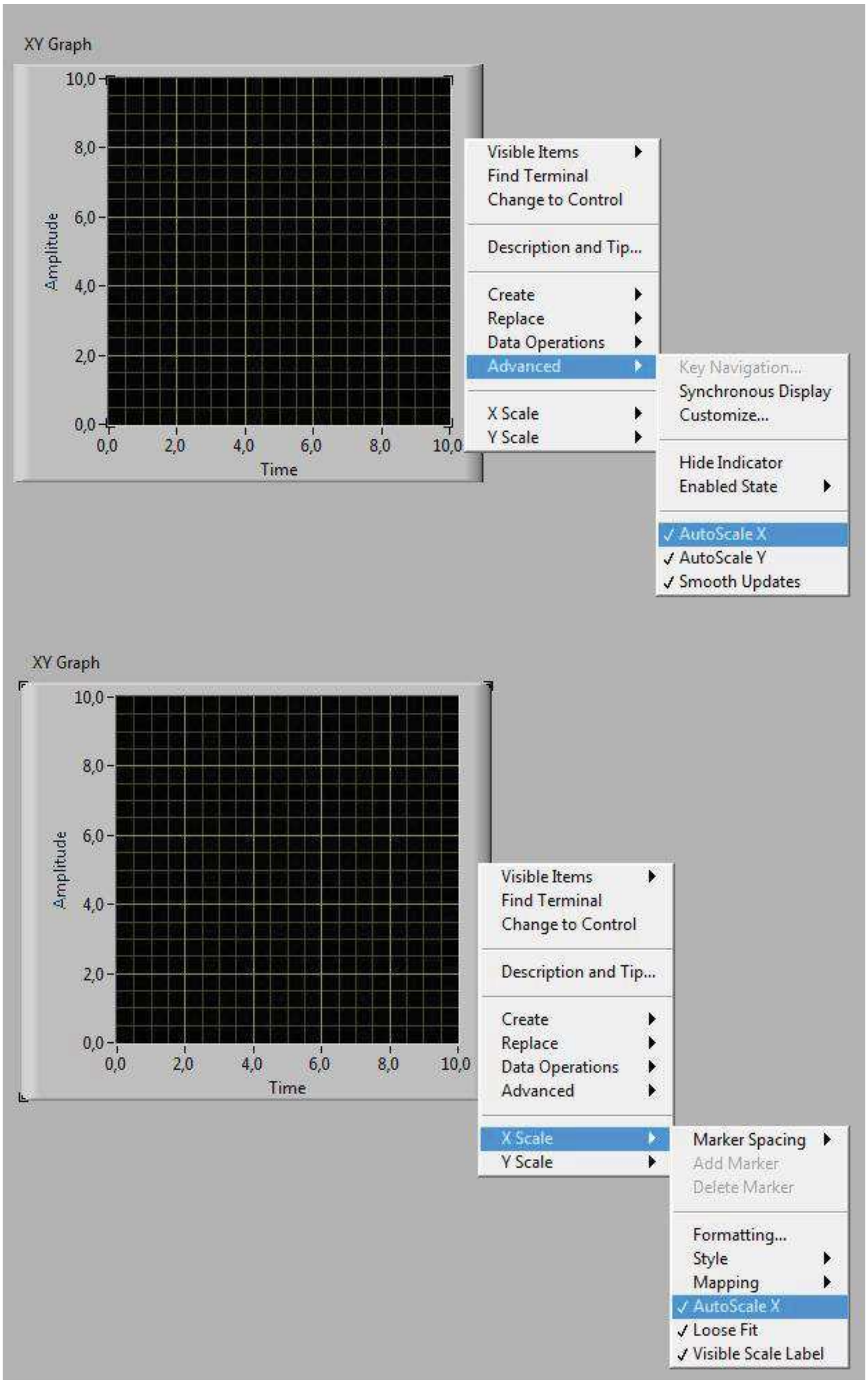

XY Graph
XY Graph
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0
2,0
4,0
6,0
8,0
10,0
Time
Plot 0
Common Plots
Color
Line Style
Line Width
Anti-Aliased
Bar Plots
Fill Base Line
Interpolation
Point Style
X Scale
Y Scale
Time
Amplitude

Fuente: Franklin Robalino

Escala de Ejes

Los ejes X y Y dentro de los gráficos pueden ser ajustados automáticamente a una escala que refleje el rango de los datos. La función “*Autoscaling*” puede ser activada o desactivada usando las opciones “*Autoscale X*” y “*AutoScale Y*” desde el sub menú “*Advanced*”, o por medio de los submenús “*X Scale*” o “*Y Scale*”. Como se muestra en la figura 3.76. Se debe tomar en cuenta que al automatizar la escala, el gráfico puede responder más lento al correr el programa.

Figura 3.76.- Escala Ejes

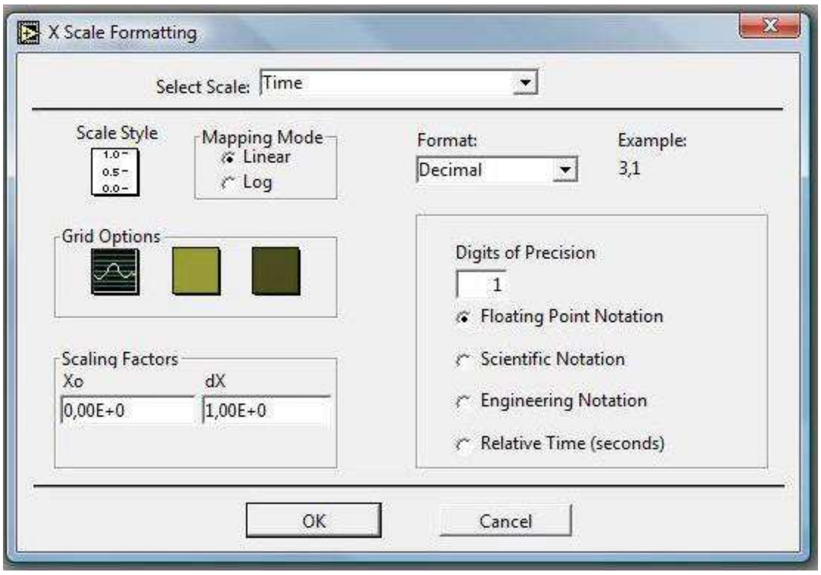

XY Graph
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0
2,0
4,0
6,0
8,0
10,0
Time
Visible Items
Find Terminal
Change to Control
Description and Tip...
Create
Replace
Data Operations
Advanced
X Scale
Y Scale
Key Navigation...
Synchronous Display
Customize...
Hide Indicator
Enabled State
✓ AutoScale X
✓ AutoScale Y
✓ Smooth Updates
XY Graph
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0
2,0
4,0
6,0
8,0
10,0
Time
Visible Items
Find Terminal
Change to Control
Description and Tip...
Create
Replace
Data Operations
Advanced
X Scale
Y Scale
Marker Spacing
Add Marker
Delete Marker
Formatting...
Style
Mapping
✓ AutoScale X
✓ Loose Fit
✓ Visible Scale Label

Fuente: Franklin Robalino

Las escalas de los ejes X y Y pueden ser variadas manualmente usando las herramientas de "*Operating*" o "*Labeling*", esto en caso se requiera hacerlo individualmente, para tener más control de la escala. Por ejemplo, si en el gráfico el último valor del eje X es 10, se puede utilizar la herramienta "*Operating*" para cambiar dicho valor a otro valor deseado. El espacio entre los valores, que se encuentran en el eje X cambiará de acuerdo al nuevo valor máximo que ha sido ingresado. En situaciones típicas, las escalas están seteadas para hacer calzar el rango exacto de datos dentro del espacio del gráfico. Se puede usar la opción "*Loose Fit*" (ver figura 3.76) cuando se requiera redondear la escala a valores múltiplos del incremento de la escala, es decir, usando el "*loose fit*" los valores de la escala son redondeados a un múltiplo del incremento usado por la escala. Por ejemplo, si los valores de la escala están en incrementos de 5, entonces con un "*loose fit*", los valores mayores y menores de la escala serán seteados a múltiplos de 5.

Tanto en el submenú "*X Scale*" como en "*Y Scale*", se encontrara una opción "*Formatting*.." (para formatear). Al escoger esta opción se abrirá un cuadro de diálogo como se muestra en la figura 3.77, que permite configurar muchos componentes del gráfico.

Figura 3.77.- Cuadro de dialogo


X Scale Formatting

Select Scale: Time

Mapping Mode: ● Linear ○ Log

Format: Decimal

Example: 3,1

Digits of Precision: 1

● Floating Point Notation

○ Scientific Notation

○ Engineering Notation

○ Relative Time (seconds)

Scaling Factors: Xo 0,00E+0 dX 1,00E+0

Fuente: Franklin Robalino

Se puede modificar individualmente varias de las características de los ejes X y Y, entre estas están las siguientes:

Scale Style.

Mapping Mode.

Grid Options.

Scaling Factors.

Precision.

El menú “*Scale Style*” permite seleccionar mayores o menores marcas dentro de la escala, o ninguna. Una marca mayor corresponde a un “*scale label*”, mientras una marca menor denota un punto interior entre los “*scale labels*”. Al hacer click en el icono de “*Scale Style*” se podrá ver las diferentes opciones disponibles. Dentro de este menú también hay la opción de hacer todos los marcadores del eje invisibles.

El “*Mapping Mode*” permite seleccionar entre una escala lineal y una escala logarítmica.

El “*Grid Options*” permite controlar las líneas de *grid* (líneas guias o cuadriculas): sin líneas, con líneas mayores (solo en “*scale label*”) o líneas en ambas mayores y menores. Esta opción también permite cambiar el color de las líneas.

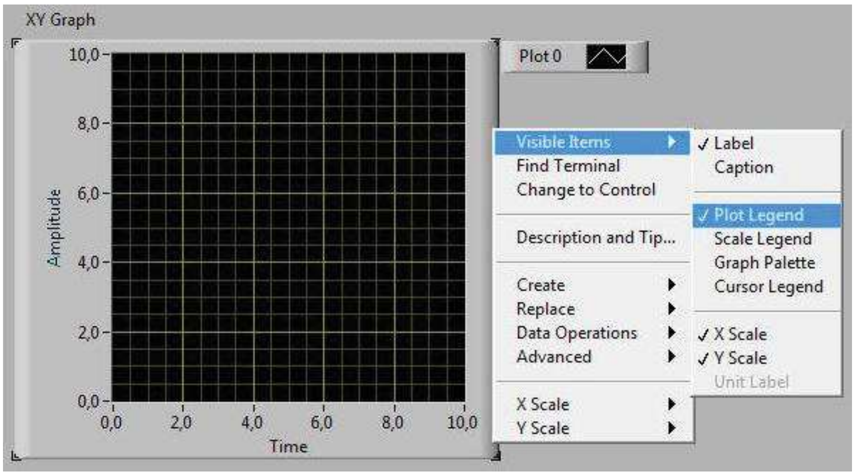
El “*Scaling Factors*” permite setear el punto de origen de los ejes X_0 (o Y_0) y también los incrementos de los ejes ΔX (o ΔY).

El área de “Precision” permite ajustar el número de dígitos de precisión para los números marcados en la escala del eje, al igual que el punto flotante, modo científico o modo de tiempo. Con el programa Lab VIEW se puede especificar el formato de la fecha y la hora para ser usados en los gráficos, como se ilustra en la figura 3.77. Dentro del menú formatear se puede elegir entre Decimal, Hexadecimal, Octal, Binario, o fecha y hora. Las opciones dentro de "*Scaling Factors*" y "*Precision*" dependen del tipo de formato seleccionado.

La Leyenda

La leyenda provee una manera de personalizar los gráficos. Aquí es donde se selecciona el tipo de línea para los datos, el estilo de puntos, espesor y color, y otras características de la apariencia del gráfico. Por ejemplo, si tenemos varios gráficos en un mismo plano cartesiano, se desearía diferenciarlos fácilmente entre ellos, así que se graficaría uno con línea solida y otro con línea punteada o cada una con distinto color, un ejemplo de leyenda se muestra en la figura 3.78. Al mover el gráfico dentro del panel frontal, la leyenda se moverá con este. Se puede cambiar la posición de leyenda en relación al gráfico al arrastrar la leyenda con la herramienta "*Positioning*" a cualquier lugar deseado.

Figura 3.78.- La leyenda


XY Graph
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0
2,0
4,0
6,0
8,0
10,0
Time
Plot 0
Visible Items
Find Terminal
Change to Control
Description and Tip...
Create
Replace
Data Operations
Advanced
X Scale
Y Scale
✓ Label
Caption
✓ Plot Legend
Scale Legend
Graph Palette
Cursor Legend
✓ X Scale
✓ Y Scale
Unit Label

Fuente: Franklin Robalino

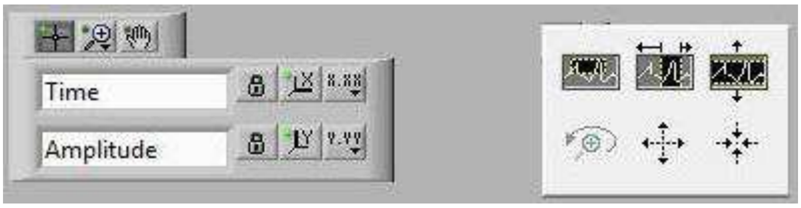
La leyenda puede estar visible o escondida, esta opción se encuentra dentro del submenú "*Visible Items*" (ver figura 3.78). Después de personalizar las características, el gráfico retiene esos datos sin importar si la leyenda es visible o no. Si el gráfico recibe más puntos (que no estén definidos dentro de la leyenda) para ser graficados, entonces estos últimos lo harán con las características de default.

Cuando la leyenda aparece por primera vez, esta tiene un porte para acomodar a un solo gráfico llamado “*Plot 0*”, por default. Si se tiene un gráfico múltiple se necesitará mostrar más membretes, esto se lo hace arrastrando la esquina del membrete de la leyenda con el cursor “*Resizing*”, para acomodar el número total de curvas. Cada curva en un gráfico múltiple es membretado como *Plot 0, Plot 1, Plot 2*, etc. Se puede también modificar el membrete default al asignar un nombre a cada una de las curvas con la herramienta “*Labeling*”. Escoger nombres que reflejen el valor físico de los datos graficados es una buena idea, por ejemplo si la curva representa una frecuencia de resonancia, se escogería el nombre de “frecuencia de resonancia en Hz” para dicha curva.

La paleta gráfica y la leyenda de la escala.

La paleta gráfica y la leyenda de la escala permiten acceso a varios aspectos de la apariencia del gráfico mientras el VI está siendo ejecutado. Se pueden auto escalar los ejes y controlar el formato de *display* de las escalas de los ejes. Para ayudar en el análisis de los datos presentados en el gráfico, botones permiten el paneo y el *zoom in* y *out*. La paleta gráfica y la leyenda de escala se muestran en la figura 3.79. Estas pueden ser mostradas en pantalla usando el submenú “*Visible Items*” desde el menú grafico.

Figura 3.79.- Paleta grafica y leyenda de escala



Fuente: Franklin Robalino

Al lado izquierdo de la leyenda de escala se encuentran tres botones que controlan el auto escala de los ejes. Al hacer click en el botón de auto escala "*Autoscale*" (ya sea en el eje X o Y), el eje seleccionado se auto escalará.

Los membretes de los ejes X y Y también pueden ser modificados con la herramienta "*Labeling*", se ingresa el nombre deseado dentro del membrete en el área de texto que se encuentra al lado izquierdo de la leyenda de escala.

Los botones para formatear la escala, se encuentran en el lado derecho de la leyenda de escala, como se muestra en figura 3.79. Se puede controlar el formato de los ejes, la precisión de los números marcadores, es decir el número de dígitos después del punto y el modo de mapeo (*mapping mode*) que fue explicado anteriormente. Hay que tomar en cuenta que en el modo de edición, se puede controlar el formato, precisión y el modo de mapeo mediante menús dentro del gráfico.

Los tres botones en el "*graph palette*" (paleta de grafico) permiten controlar el modo de operación del grafico. Bajo circunstancias normales este modo estará en modo estándar, donde el botón "*cross-hair*" (el botón del lado izquierdo de la paleta) estará seleccionado. Esto indica que se puede hacer click en el área del gráfico para poder mover a los cursores. Otros modos operacionales incluyen paneo y *zooming*. El botón de paneo (ver la figura 3.79), permite navegar por los datos del gráfico al hacer click y arrastrar las diferentes secciones del gráfico con el cursor "*pan*". El botón zoom dará acceso a un menú que permite escoger entre diferentes métodos de zoom.

Como se puede observar en la figura 3.79 existen varias opciones de zoom:

- *Zoom by Rectangle.-* (zoom rectangular) con este modo se puede usar el cursor para dibujar un rectángulo alrededor del área que se desea hacer un acercamiento, y al soltar el botón del mouse, los ejes cambiaran de escala para acercarse al área deseada.

- *Zoom by Rectangle 2*.- este es un zoom rectangular restringido a los datos del eje X. este es muy similar en su funcionamiento al modo "*Zoom by Rectangle*" anterior con la diferencia que el eje Y se mantiene sin cambios.

- *Zoom by Rectangle 3*.- este es un zoom rectangular restringido a los datos del eje Y. este es muy similar en su funcionamiento al modo "*Zoom by Rectangle*" anterior con la diferencia que el eje X se mantiene sin cambios.

- *Undo Last Zoom*.- (deshace el último zoom) este resetea las escalas de los ejes a su seteo anterior.

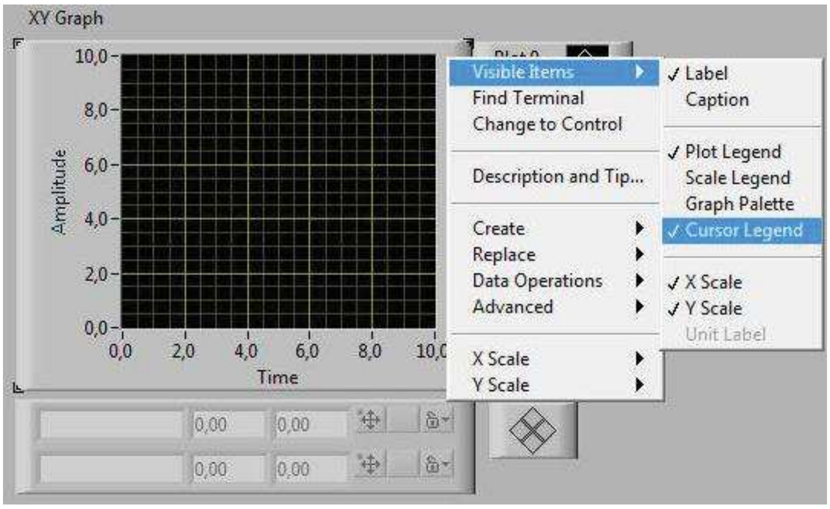
- *Zoom In about a Point*.- con esta opción, al mantener presionado el botón del mouse en un punto del gráfico, este aumentara de tamaño hasta que el botón sea soltado.

- *Zoom Out about a Point*.- con esta opción, al mantener presionado el botón del mouse en un punto del gráfico, este disminuirá de tamaño hasta que el botón sea soltado.

Propiedades especiales de personalización grafica: leyenda del cursor

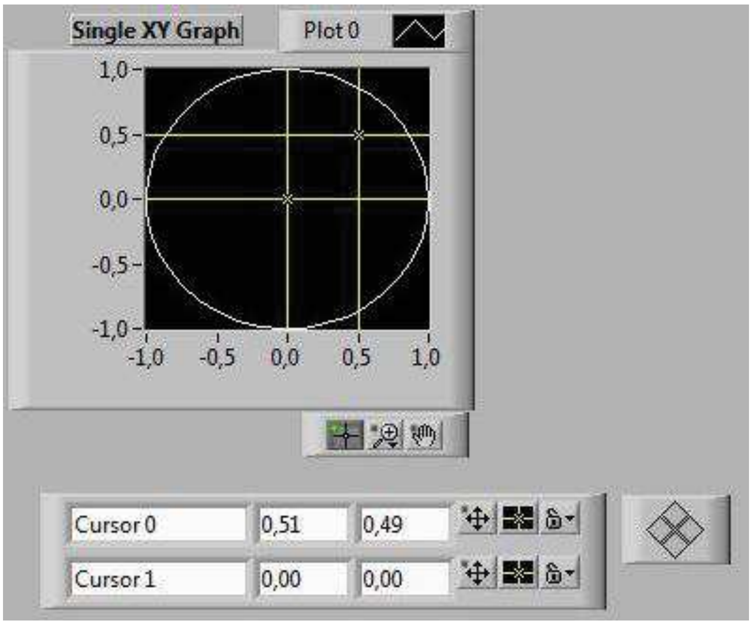
Para añadir cursores a un gráfico se debe primero tener en el panel frontal el "*Cursor Legend*", esto se lo hace mediante un click en el grafico y seleccionando "Visible Items>>Cursor Legend" dentro del menú, y después haciendo click en cualquier lugar de la leyenda del cursor para activar un cursor, como se muestra en la figura 3.80. Se puede añadir múltiples cursores en el gráfico, y dentro de la leyenda de cursores se encontrarán los valores exactos de los puntos. La leyenda del cursor (*Cursor Legend*) se muestra en la figura muestra en la figura 3.81.

Figura 3.80.- Leyenda del cursor


XY Graph
Amplitude
10,0
8,0
6,0
4,0
2,0
0,0
0,0
2,0
4,0
6,0
8,0
10,0
Time
Visible Items
Find Terminal
Change to Control
Description and Tip...
Create
Replace
Data Operations
Advanced
X Scale
Y Scale
✓ Label
Caption
✓ Plot Legend
Scale Legend
Graph Palette
✓ Cursor Legend
✓ X Scale
✓ Y Scale
Unit Label
0,00
0,00
0,00
0,00

Fuente: Franklin Robalino

Figura 3.81.- Leyenda del cursor


Single XY Graph
Plot 0
1,0
0,5
0,0
-0,5
-1,0
-1,0
-0,5
0,0
0,5
1,0

Cursor 0 0,51 0,49
Cursor 1 0,00 0,00

Fuente: Franklin Robalino

Se pueden poner y etiquetar todos los cursores dentro de los gráficos. Para asignar un cursor a una curva en el gráfico se selecciona el icono "*lock*" (candado) que está asociado con el ícono de leyenda del cursor. Un gráfico puede tener un sinnúmero de cursores y múltiples cursores pueden moverse simultáneamente.

La herramienta "*Operating*" es usada para seleccionar cualquiera de los siguientes botones ubicados en la parte derecha de la leyenda de cursor, para poder personalizar al cursor:

***Cursor Movement Selector*.-** al presionar este botón, se puede mover el cursor usando el "*cursor mover*" (ver figura 3.81). Si se presiona el "*cursor mover*" para más de un cursor, los cursores se moverán en forma paralela.

***Formatting Ring*.-** este permite personalizar la apariencia de cada cursor.

***Lock Ring*.-** este permite personalizar la acción de cada cursor o asociar un cursor en particular a una curva en particular.

Dentro del "*Formatting Ring*" se puede elegir personalizar la apariencia del cursor. Las siguientes opciones se encuentran dentro de éste:

***Color*.-** permite escoger el color del cursor desde una paleta de colores.

***Cursor Style*.-** provee varios estilos de cursores.

***Point Style*.-** suministra diversos estilos de puntos para las intersecciones del cursor.

***Line style*.-** proporciona algunos tipos de líneas tanto solidas como punteadas.

***Line width*.-** facilita varias anchuras para las líneas.

***Show name*.-** muestra el nombre del cursor en el gráfico. Se puede usar la herramienta "*Positioning*" para mover al nombre con relación al cursor.

***Bring to Center*.-** este centra el cursor dentro del gráfico sin cambiar las escalas de los ejes X y Y.

***Go to Cursor*.-** este cambia las escalas de los ejes X y Y para poder mostrar al cursor en el centro del gráfico.

El “*lock Ring*” provee tres opciones para personalizar las acciones del cursor:

***Free*.-** con esta opción se hace click con el cursor dentro del gráfico o se digitan los valores en las coordenadas X y Y en la leyenda del cursor, para que el cursor se mueva.

***Snap to Point*.-** con esta opción el cursor se mueve al punto del gráfico más cercano. El cursor puede cambiarse a otra curva dentro del gráfico con este modo de operación.

***Lock to Plot*.-** con esta opción el cursor se moverá sólo dentro de una curva. El cursor no puede cambiarse a otra curva dentro del gráfico con este modo de operación. Si existe más de una curva dentro del gráfico, Lab VIEW lista las curvas en la parte de abajo del menú “*Lock Ring*”.

CAPÍTULO 4

DESARROLLO PRÁCTICO DEL PROBLEMA

4.1. PLANES

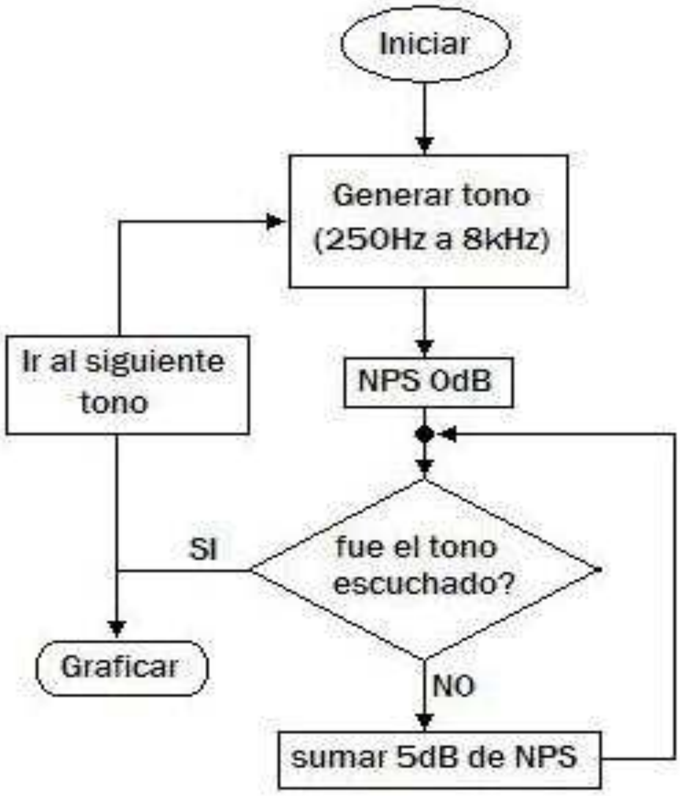
Para poder diseñar el “Software de Estimación de Pérdida Auditiva” se ha analizado el funcionamiento de los audiómetros analógicos existentes, con el propósito de usar dicha información para realizar la programación del software.

La idea es utilizar las herramientas del programa LabVIEW, que ya se describieron en el capítulo 3, para poder simular un audiómetro por medio de software y automatizar su operación.

El “Software de Estimación de Pérdida Auditiva” deberá funcionar de igual manera como lo hace un audiómetro normal, es decir, tendrá que reproducir tonos puros de 250Hz, 500Hz, 1kHz, 2kHz, 4kHz, y 8kHz, con diferentes valores de NPS, comenzando en 0dBs e incrementando su valor de NPS en intervalos de 5dBs, esto se realizará individualmente para cada oído (izquierdo y derecho). Dichos tonos saldrán por medio de audífonos hacia el usuario, el cual cada vez que escuche un tono, deberá oprimir un botón en la pantalla del computador. Una vez que el programa haya terminado la prueba, automáticamente graficará los resultados en la pantalla, mostrando un gráfico por cada oído (izquierdo y derecho).

Para facilitar la programación del “Software de Estimación de Pérdida Auditiva” se ha elaborado un diagrama de flujo (ver figura 4.1), con el propósito de tener una idea de la estructura básica que tendrá el programa.

Figura 4.1.- Diagrama de flujo


Iniciar
Generar tono (250Hz a 8kHz)
Ir al siguiente tono
NPS 0dB
SI
fue el tono escuchado?
Graficar
NO
sumar 5dB de NPS

Fuente: Franklin Robalino

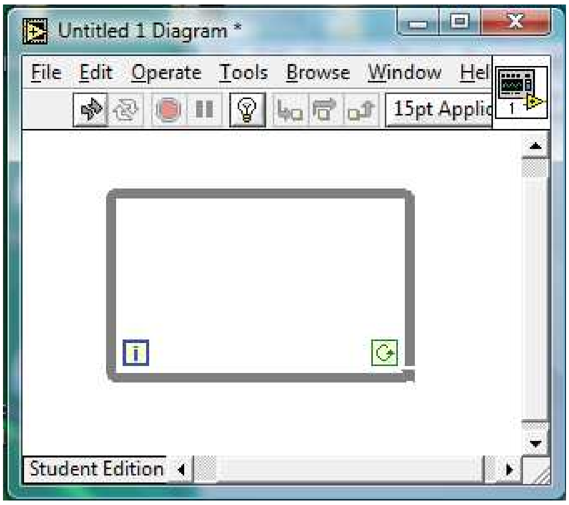
Al observar el diagrama de flujo de la figura 4.1, podemos ver que primero se deberá iniciar el programa, luego el programa generará el primer tono de 250Hz con un NPS de 0dBs. Si el tono fue escuchado, el programa se dirigirá por el lado izquierdo del diagrama de flujo para graficar el primer punto y generar el siguiente tono. Si el tono no fuese escuchado el programa irá hacia abajo en el diagrama de flujo y sumará 5dBs al tono hasta que este sea escuchado. Una vez que el programa termine se mostrarán los gráficos en la pantalla. Este diagrama de flujo es solamente el flujo básico de la información que estará dentro del programa, pues mientras que el programa vaya tomando su forma final en LabVIEW, puede que sea necesario hacerle algunos cambios.

4.2. METODOLOGÍA

Primero se programará en LabVIEW el “Software de Estimación de Pérdida Auditiva” solamente con dos tonos y con diferentes amplitudes, para un solo oído. La razón de hacer esto primero es para verificar que las herramientas del programa estén funcionando correctamente con los dos primeros tonos. De esta manera si se encuentra algún problema con la programación será más fácil de detectar el error y arreglarlo, antes de incluir a los demás tonos y los gráficos.

Se abrirá un nuevo archivo dentro de LabVIEW (ver capítulo 3 para procedimientos). Luego se creará un *while loop* dentro de la ventana de diagrama, como se muestra en la figura 4.2, con el propósito de controlar la ejecución del código en su interior con un botón de control (referirse al capítulo 3 para detalles del funcionamiento del *while loop*). De esta manera se podrá hacer que el código dentro del *loop* reproduzca un tono puro con distintos valores de NPS y una vez que el usuario escuche un tono, presionará el botón, deteniendo el *loop* e iniciando otro *loop* con otra frecuencia.

Figura 4.2.- Crear un While Loop

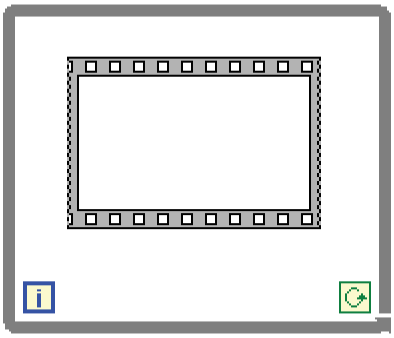


Fuente: Franklin Robalino

Dentro del *while loop* se creará a la estructura de secuencia, como se muestra en la figura 4.3, de este modo el *loop* tendrá la función de detener la secuencia al aplastar un botón en la pantalla (este botón es el que aplastará el usuario cuando escuche el tono).

La secuencia básicamente tendrá dentro de ella un código secuencial que generará el primer tono a un nivel de 0 dB (HL) e irá incrementando secuencialmente su amplitud en intervalos de 5 dB (HL).

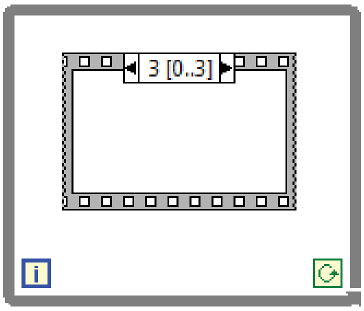
Figura 4.3.- Crear una estructura de secuencia



Fuente: Franklin Robalino

Se añadirán 3 *frames* en la estructura de secuencia, como se muestra en la figura 4.4, dándonos un total de 4 *frames*, cada uno de ellos será responsable de ejecutar un código con distintos valores de NPS, la razón por la que sólo se añadirán 3 *frames* es para verificar su funcionamiento antes de añadir el resto.

Figura 4.4.- Crear frames en la estructura de secuencia



Fuente: Franklin Robalino

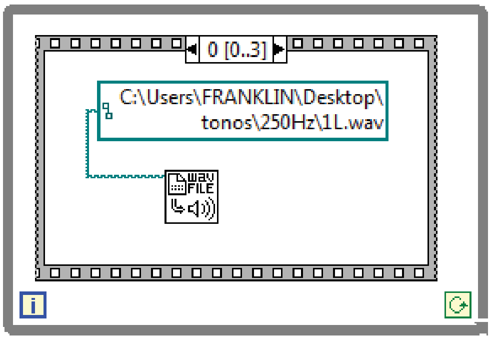
Luego se creará dentro de cada uno de los *frames* de la estructura de secuencia, un código, los cuales se encargarán de generar tonos de una frecuencia específica con distintos valores de NPS, comenzando en 0 dB (HL) e incrementándolos con un valor de 5dBs (HL) secuencialmente. Es decir el tono irá incrementando de nivel a medida que la secuencia avance.

Para que el programa pueda reproducir los diferentes tonos, se usará el subVI llamado "*Snd play wave file*" (ver figura 4.5), este subVI permitirá leer archivos de sonidos que se encuentren dentro de cualquier carpeta del computador. Para que este subVI reproduzca el archivo de sonido deberá saber la dirección exacta del archivo dentro del computador. Por ejemplo para nuestro caso se ha creado una carpeta en el escritorio llamada "tonos" y dentro de ella una subcarpeta llamada "250Hz", la cual lleva dentro un archivo .wav llamado "1L". Este archivo es un tono puro de 250Hz, cuya dirección exacta dentro del computador será C:\Users\FRANKLIN\Desktop\tonos\250Hz\1L.wav

Figura 4.5.- Ícono del subVI "Snd play wave file"

Fuente: Franklin Robalino

Figura 4.6.- "Snd play wave file" dentro de la estructura

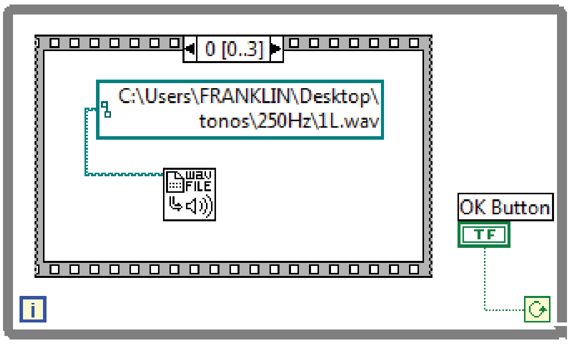


Fuente: Franklin Robalino

Una vez creado el ícono "*Snd play wave file*" con la respectiva dirección del archivo a leerse, se copiará el código del primer *frame* (0) de la secuencia, y se los pegará dentro de los *frames* 1, 2 y 3. Manteniendo la misma dirección (C:\Users\FRANKLIN\Desktop\tonos\250Hz\1L.wav), usaremos el mismo archivo wav solamente para probar la correcta operación del programa. Más adelante en este capítulo se explicará cómo se crean los distintos tonos a diferentes amplitudes. Hasta mientras, cualquier archivo wav de corta duración funcionará para probar el correcto funcionamiento del *loop*.

Dentro del *loop* se creará un botón de control, como se muestra en la figura 4.7, el cual será conectado al terminal condicional del *while loop*, este permitirá al usuario controlar el momento en que se detenga el *loop* como ya se mencionó anteriormente (en el momento que el usuario escuche el tono, aplastará el botón).

Figura 4.7.- conectando el botón de control al terminal condicional

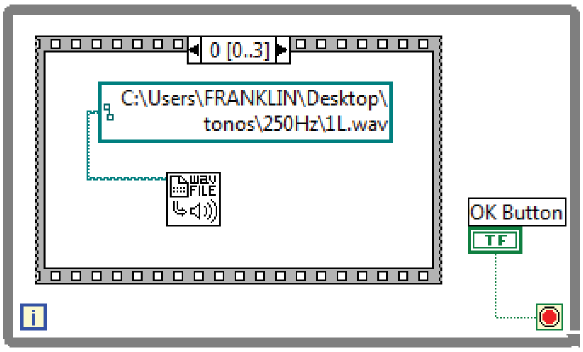


Fuente: Franklin Robalino

En este punto el botón "*Broken Run*", cambiará a "*Run*", esto es debido a que si se intenta ejecutar el programa antes de colocar el botón tendremos un error, debido a que no estaba conectado aun el terminal condicional del *loop*. Al cambiar de ícono el botón "*Broken Run*" a "*Run*" permite ejecutar el código.

Debido a que el terminal condicional está configurado a "*continue if true*" habrá que cambiarlo de "*continue if true*" a "*stop if true*" para que de esta manera, al aplastar el botón se detenga el *loop*, como se muestra en la figura 4.8.

Figura 4.8.- terminal condicional



Fuente: Franklin Robalino

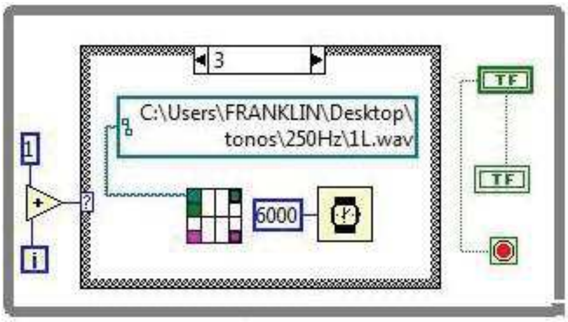
Al ejecutar el programa con el botón "*Run*", se encontrará un problema: que el *loop* no se detiene hasta que concluya la secuencia completa, es decir que si el usuario escuchó ya el tono y aplastó el botón, el *loop* seguirá incrementando la amplitud, y no se detendrá hasta acabar de reproducir todas las amplitudes de dicho tono . Otro problema es que si se presiona el botón en la primera secuencia, el programa no leerá el botón hasta comenzar el segundo *loop* y solamente al completarse el segundo este se detendrá, debido al funcionamiento del terminal condicional (referirse al capítulo 3).

Para solucionar esto reemplazaremos la estructura de secuencia por una estructura *case*, como se muestra en la figura 4.9. Con esta estructura se tendrá más control en la reproducción de los sonidos debido a que se puede elegir el case que va a ser ejecutado.

Para poder elegir el *case* que va a ser ejecutado se sumará una constante con un valor de "uno" al "*iteration terminal*" (terminal de iteración), como se muestra en la figura 4.9, de esta manera los case se empezarán a ejecutar desde el case número uno (debido a que el *iteration terminal* siempre comienza con un valor de cero) e irán incrementándose con un valor de uno. De esta manera cada *case* será responsable de reproducir una sola amplitud del tono y se detendrá en dicha amplitud cuando el usuario presione el botón.

También se añadirá una función de tiempo, para que cada *frame* espere un intervalo de tiempo prudente antes de pasar al siguiente *frame*, esto es para que el usuario tenga un tiempo suficiente para escuchar el tono y aplastar el botón antes de que el programa pase al siguiente valor de amplitud.

Figura 4.9.- remplazando la estructura de secuencia por la estructura case



Fuente: Franklin Robalino

Aquí también se añadirá un LED conectado al botón dentro del *loop* para que de esta manera el usuario pueda verificar visualmente la activación del botón desde la pantalla principal, y de esta manera se asegure que la información haya sido revivida por el programa al aplastar el botón.

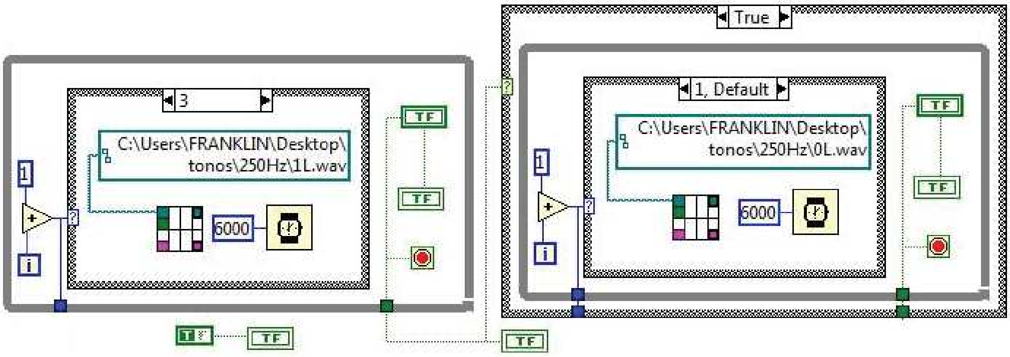
Una vez que se ejecute este programa, habrá otro problema que arreglar, pues debido al funcionamiento del *loop* como ya se mencionó anteriormente, si se presiona el botón en la primera estructura de un *case*, el programa no leerá el botón hasta comenzar la segunda estructura y solamente al completarse la segunda estructura este *loop* se detendrá. Es decir si el usuario escuchó el tono a un valor de 5 dB (HL) y aplastó el botón, el *loop* se detendrá una vez que haya reproducido el siguiente valor, es decir de 10 dB (HL), o si el botón fue aplastado en 20dB (HL), se detendrá en 25 dB (HL). Esto puede causar una serie de confusiones tanto para el usuario como para el programa, pues si el usuario escucha el primer tono y presiona el botón y enseguida escucha otra amplitud de la misma frecuencia, aplastará nuevamente el botón y el programa pensará que escuchó dos tonos en frecuencias diferentes, por lo tanto los resultados de los gráficos serán erróneos. Por otro lado si en las instrucciones de uso del programa se pone que el usuario debe oprimir el botón al escuchar un tono y después oprimir cuando escuche un tercer tono es decir sin aplastar cada vez que escuche un tono par o intermedio, sería muy complicado para el usuario. Debido a esto se hará un pequeño ajuste en los case, dejando vacío los case pares, para que de esta manera no reproduzcan ningún sonido y se facilite el uso del programa y funcione exactamente igual a un audiómetro análogo.

Una vez que esté esta parte del programa funcionando, se creará otro *loop* para la siguiente frecuencia y de esta manera poder comprobar si está funcionando correctamente el programa para luego añadir el resto de frecuencias.

Se copiará el *loop* ya existente y se lo pegará dentro de una estructura case con valores de verdadero y falso (*True* y *False*), como se muestra en la figura 4.10. Esta nueva estructura case se colocará en el programa para que al ejecutar el programa no reproduzcan todas las frecuencias simultáneamente sino que espere a que la anterior frecuencia sea escuchada y al aplastar el botón se detenga el *loop* anterior y active la siguiente frecuencia.

Para que esto funcione correctamente se necesitará de un control, el cual active el *case* que contiene al segundo *loop*, que para nuestro caso será el botón ya existente del primer *loop*. Debido a que el *loop* acepta un botón dentro del *loop* solamente y no fuera (referirse al capítulo 3) se crearán diferentes botones para cada frecuencia (*loop*) y un LED conectado al terminal del *case* que indique que está activo el *loop* dentro de la pantalla principal. También se creará un LED con una constante para el primer *loop* el cual se activará al iniciar el primer *loop*, esto permitirá al usuario saber qué botón está activo y cuáles no. (Referirse a sección 4.4 Manuales)

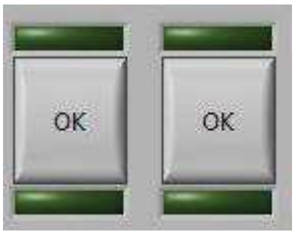
Figura 4.10.- generando la segunda estructura case


3
C:\Users\FRANKLIN\Desktop\tonos\250Hz\1L.wav
6000
1
i
TF
True
1, Default
C:\Users\FRANKLIN\Desktop\tonos\250Hz\0L.wav
6000
1
i
TF

Fuente: Franklin Robalino

Una vez que tengamos listos los dos *loops*, se comprobará el correcto funcionamiento del programa, pues al escuchar un tono del primer *loop* y aplastar el botón este *loop* se detendrá y se activará el segundo *loop* con la siguiente frecuencia. Ahora es cuestión de seguir duplicando los *loops* para poder generar todas las frecuencias requeridas para el primer oído.

Figura 4.11.- botones y LEDs en la pantalla principal hasta este punto

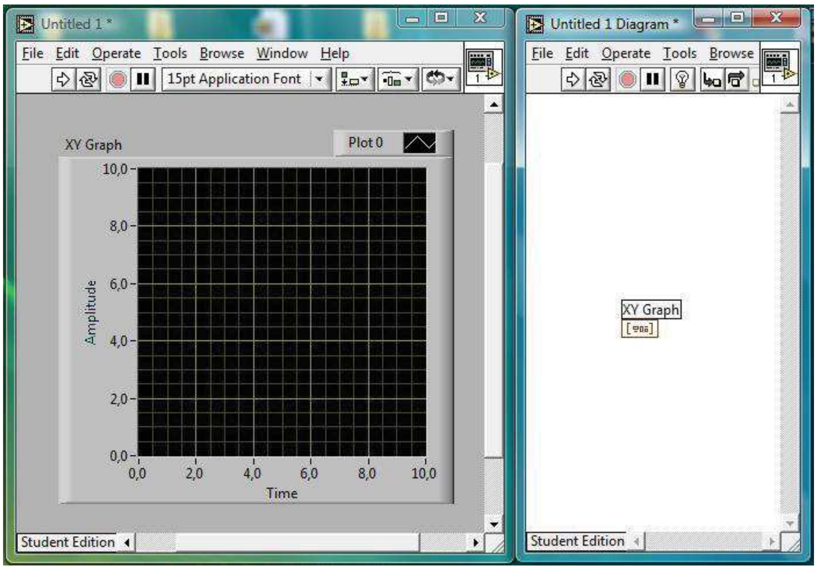


Fuente: Franklin Robalino

Luego se usarán los datos numéricos de la suma entre el "*iteration terminal*" y la constante con valor uno para poder obtener las coordenadas exactas y graficar la respuesta del un oído, para esto se hará una conexión desde la salida de la sumatoria hasta el borde del *loop* en el primer *loop* (creando un túnel) y hasta el borde del case en el segundo *loop* (creando dos túneles), como se muestra en la figura 4.10.

Para graficar los datos se creará el *XY Graph*, como se muestra en la figura 4.12, el cual permitirá graficar los puntos específicos por medio de coordenadas. Para obtener las coordenadas, se usarán los datos numéricos que se obtienen de la sumatoria del *loop* como se mencionó anteriormente. Hay que tener en cuenta que al crear el gráfico, éste automáticamente creará una escala del 0,0 al 10,0, la cual hay que rectificar editando los valores de la amplitud (eje Y) para que se grafiquen desde 50 a 0 dBs HL. El eje X que representará la frecuencia tendrá valores de 0 a 7 con intervalos de una unidad, esto es debido a que si se usa la escala logarítmica el gráfico no será igual al presentado en una audiometría normal, por que el papel usado para la audiometría no es logarítmico, y por eso para ser lo más parecido al real se usará una escala lineal, en nuestro caso el uno representará 250Hz, el 2 representara 500Hz, el 3 representara 1kHz, y así sucesivamente.

Figura 4.12.- XY graph con su diagrama de bloques


Untitled 1 *
File Edit Operate Tools Browse Window Help
15pt Application Font
XY Graph
Plot 0
10,0
8,0
6,0
4,0
2,0
0,0
Amplitude
0,0 2,0 4,0 6,0 8,0 10,0
Time
Student Edition
Untitled 1 Diagram *
File Edit Operate Tools Browse
XY Graph
Student Edition

Fuente: Franklin Robalino

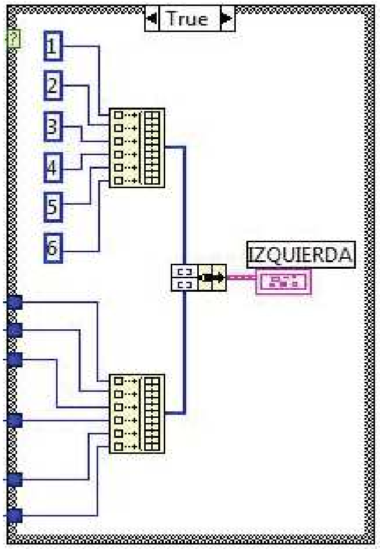
Para poder graficar los valores de las coordenadas se necesitará crear dos "*Build array*" que se encuentran en el menú *array* dentro de la paleta de funciones. Estos permitirán crear una especie de tabla, un "*Build array*" con los valores de X, mientras que el otro con los valores de Y, como se muestra en la figura 4.13.

Los “*Build array*” construirán un *array* con los elementos que estén conectados en sus *inputs*, donde cada elemento será considerado en orden para crear el *array*, es decir el elemento uno va a estar en el primer puesto en el *array*, el segundo elemento se encontrará en el segundo puesto y así sucesivamente.

La función “*Bundle*” (ver figura 4.13) será utilizada para combinar los datos del *array* X con el *array* Y, y de esta manera representarán las coordenadas, las cuales serán mandadas a ser graficadas en el *XY Graph*. El problema aquí es que no queremos que se grafiquen los puntos hasta que el programa haya completado de generar todos los *loops* (hasta que el usuario haya completado todo el examen de audición), entonces se creará un case de verdadero y falso, el cual permitirá controlar el momento en que el programa comience a graficar.

Para el *array* que representará el eje X los valores de los elementos serán constantes del 1 al 6, los cuales representarán las frecuencias en el gráfico de manera lineal como se mencionó anteriormente. Mientras que para los valores del eje Y se usarán los datos numéricos de la suma entre el “*iteration terminal*” y la constante con valor uno.

Figura 4.13.- XY graph con sus arrays y la función bundle



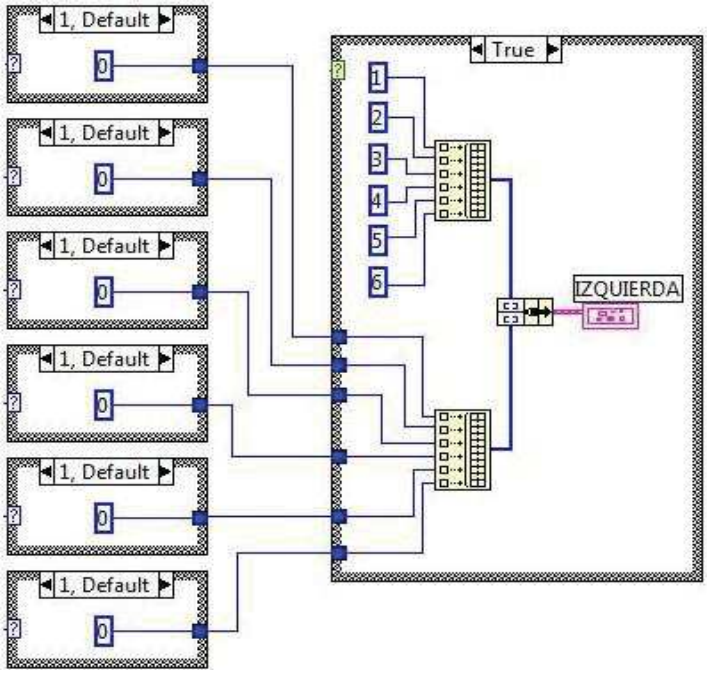
Fuente: Franklin Robalino

Una vez completado el código dentro del case para graficar el programa no podrá ser ejecutado pues aun faltan conectar el control *Boolean* del *case* y los inputs para crear el *array* del eje Y.

Debido a que los datos numéricos que salen de la sumatoria dentro del *loop* no representarán los valores de amplitud exactos en dB (HL), se tendrá que crear un código que lea el valor de la sumatoria y represente el valor interno del *loop* en dB (HL). Es decir si se detiene el *loop* en el *case* uno o dos, los valores numéricos de la sumatoria darán como resultado 1 ó 2, los cuales representan un valor de 0 dB (HL), mientras que si se detiene en los *cases* 3 ó 4, la sumatoria dará 3 ó 4 los cuales representarán a un valor de amplitud de 5 dB (HL). Esto puede sonar un poco complejo pero no lo es, pues con sólo crear un *case* numérico para interpretar cada valor del eje Y, se podrá solucionar esto.

Al crear los *cases* numéricos hay que tener en cuenta que cada uno de ellos va a realizar la misma operación así que se creará sólo uno y luego se duplicará el *case* para cada uno de los elementos del eje Y, como se muestra en la figura 4.14. El código del *case* básicamente lo que hará es recibir el valor de la sumatoria del *loop* por medio del *input* de la estructura *case* y según el valor recibido mandará a su salida (túnel) el valor que representará en la escala de dB (HL). De esta manera si los valores numéricos de la sumatoria dan como resultado 1 ó 2, el output en el túnel del *case* será un valor de 0 dB (HL), si la sumatoria da como resultado 3 ó 4, el *output* del *case* será un valor de 5 dB (HL) y así sucesivamente.

Figura 4.14.- estructuras case para cada elemento del eje Y



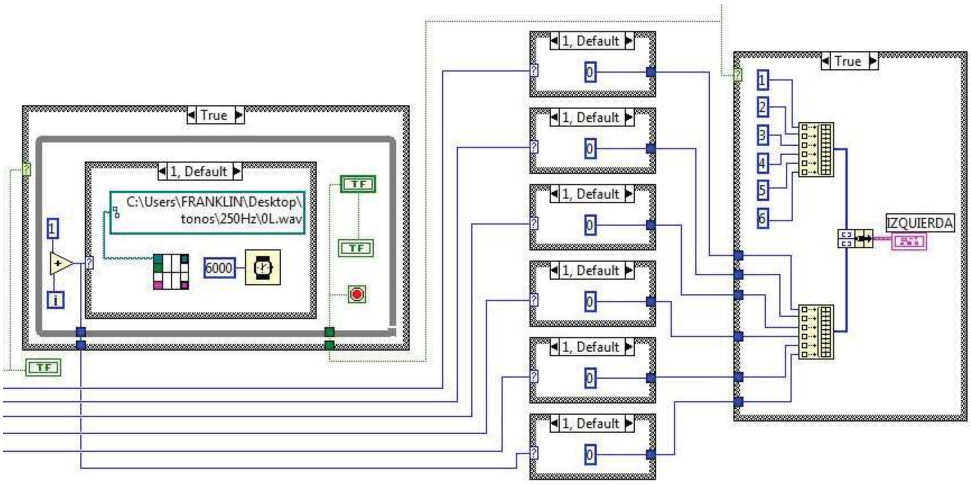
Fuente: Franklin Robalino

Al terminar la codificación de las estructuras numéricas *case*, se conectará sus túneles (*outputs*), a los túneles *inputs* de la estructura *case* verdadera y falsa que contiene el código para graficar, para que de esta manera los datos de los elementos ingresen al *array*, como se muestra en la figura 4.14.

El siguiente paso es conectar las salidas de las sumatorias de los *loops* a cada uno de los *case* numéricos para que los *cases* sepan que valores son los que el usuario escuchó, como se muestra en la figura 4.15. (Debido a que el código completo a estas alturas es muy grande, sólo se muestra el último *case* numérico conectado al último *loop*, y hay que tomar en cuenta que el resto de *cases* numéricos están conectados a los otros *loops*).

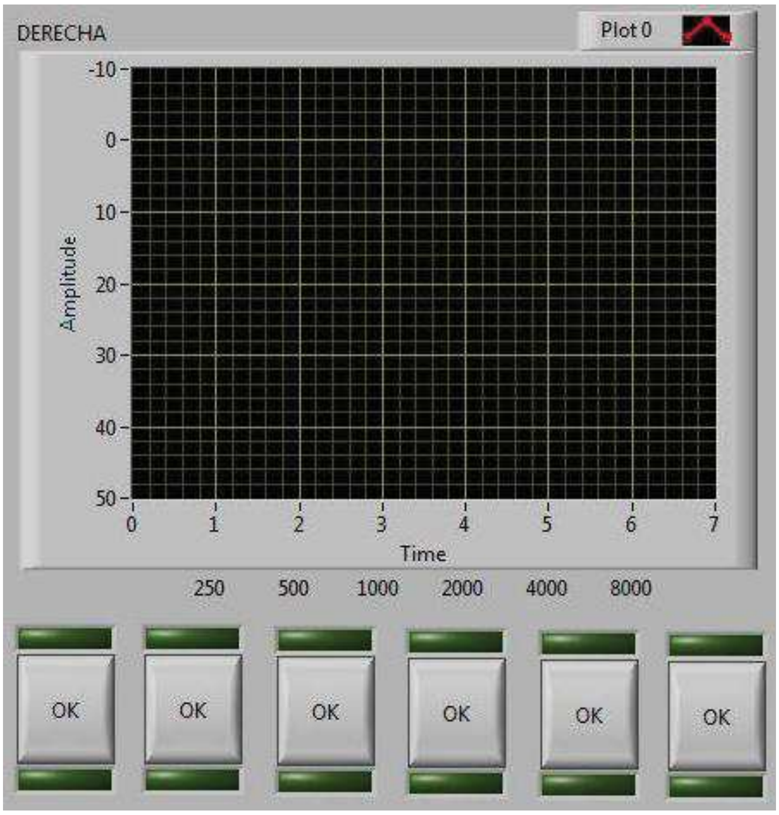
También el botón del último *loop* es el que activará al *case* del gráfico, para que de esta manera sólo una vez que el último *loop* haya sido reproducido (cuando el programa haya terminado el examen de audición) el gráfico pueda activarse y graficar los puntos en la pantalla principal del programa.

Figura 4.15.- elementos del eje Y


True
1, Default
C:\Users\FRANKLIN\Desktop\tonos\250Hz\0L.wav
6000
TF
TF
TF
1
i
1, Default
0
True
1
2
3
4
5
6
IZQUIERDA

Fuente: Franklin Robalino

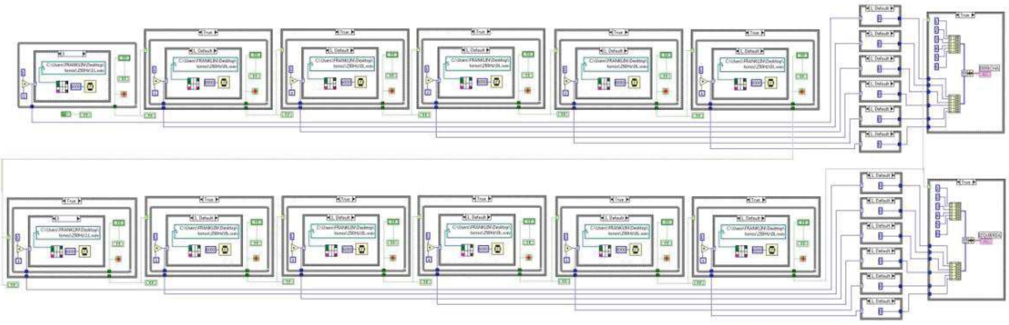
Figura 4.16.- vista de la pantalla principal hasta este punto


DERECHA
Plot 0
-10
0
10
20
30
40
50
Amplitude
0
1
2
3
4
5
6
7
Time
250
500
1000
2000
4000
8000
OK
OK
OK
OK
OK
OK

Fuente: Franklin Robalino

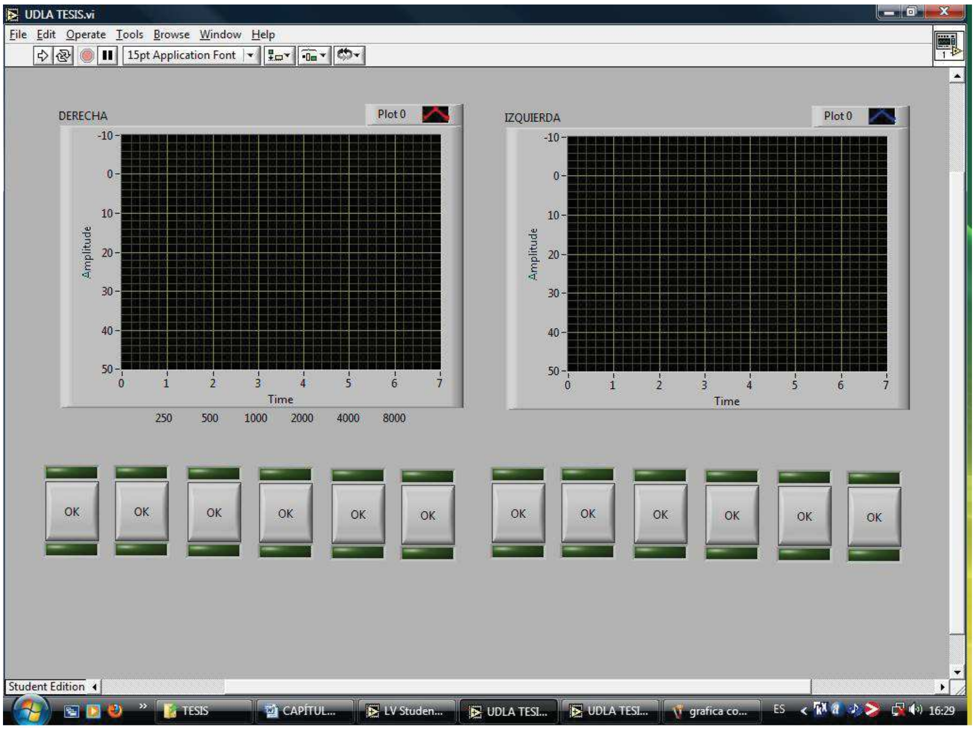
Al ejecutar el programa podemos verificar su correcto funcionamiento y ahora es cuestión de solamente duplicar el código de un oído para que realice la misma operación con el otro oído, como se muestra en la figura 4.17.

Figura 4.17.- código completo



Fuente: Franklin Robalino

Figura 4.18.- vista de la pantalla principal completa


UDLA TESIS.vi
File Edit Operate Tools Browse Window Help
15pt Application Font
DERECHA
Plot 0
-10
0
10
20
30
40
50
Amplitude
0 1 2 3 4 5 6 7
Time
250 500 1000 2000 4000 8000
IZQUIERDA
Plot 0
-10
0
10
20
30
40
50
Amplitude
0 1 2 3 4 5 6 7
Time
OK OK OK OK OK OK OK OK OK OK OK OK
Student Edition
TESIS
CAPÍTUL...
LV Studen...
UDLA TESI...
UDLA TESI...
grafica co...
ES
16:29

Fuente: Franklin Robalino

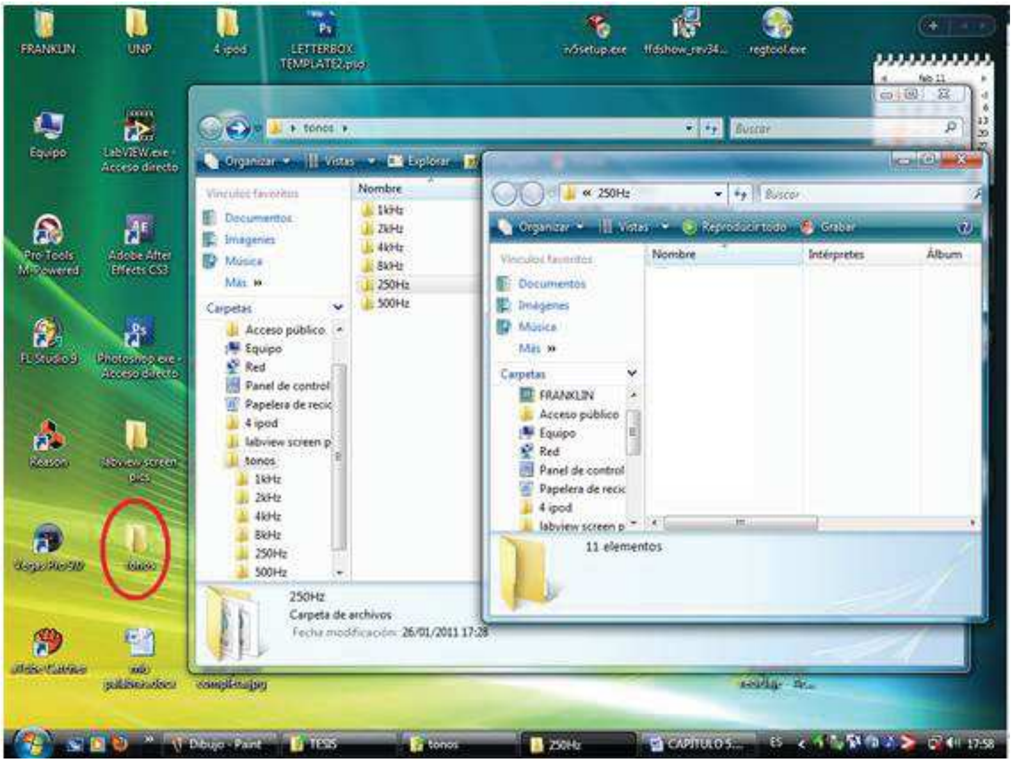
Para resumir, brevemente, lo que se hizo fue crear el programa con sólo dos tonos, y una vez que el programa funcionó con los dos primeros tonos para un solo oído se añadió los demás tonos para dicho oído y se usaron los índices de los *loops* para realizar la gráfica, luego se duplicó el código entero del un oído para que de esta manera ejecute el mismo código en el otro oído.

Ahora se tiene ya el código funcionando correctamente y lo único que falta es la generación de los diferentes tonos con diferentes amplitudes.

Se usará un programa de edición de audio para generar los tonos puros requeridos en una audiometría, es decir tonos con frecuencias de 250Hz, 500Hz, 1kHz, 2kHz, 4kHz, y 8kHz, con diferentes valores de NPS, comenzando en 0dBs e incrementando su valor de NPS en intervalos de 5dBs, estos tonos se generarán independientemente para el canal “L” (*left*) y para el canal “R” (*right*), para de esta manera poder chequear ambos oídos por separado.

Como ya se mencionó anteriormente, para que el programa hecho en LabVIEW pueda reproducir los diferentes tonos, se usará el subVI llamado “*Snd play wave file*”, este subVI permitirá leer archivos de sonidos que se encuentren dentro de cualquier carpeta del computador. Para que este subVI reproduzca el archivo de sonido deberá saber la dirección exacta del archivo dentro del computador. Para nuestro caso se ha creado una carpeta en el escritorio llamada “tonos” y dentro de ella habrá sub carpetas una para cada frecuencia es decir habrá 6 subcarpetas con los nombres de las frecuencias: 250Hz, 500Hz, 1kHz, 2kHz, 4kHz, y 8kHz, como se muestra en la figura 4.19. Y dentro de estas estarán los archivos wav de las respectivas frecuencias con distintos NPS tanto para el oído izquierdo como para el derecho.

Figura 4.19.- carpetas para archivar los tonos

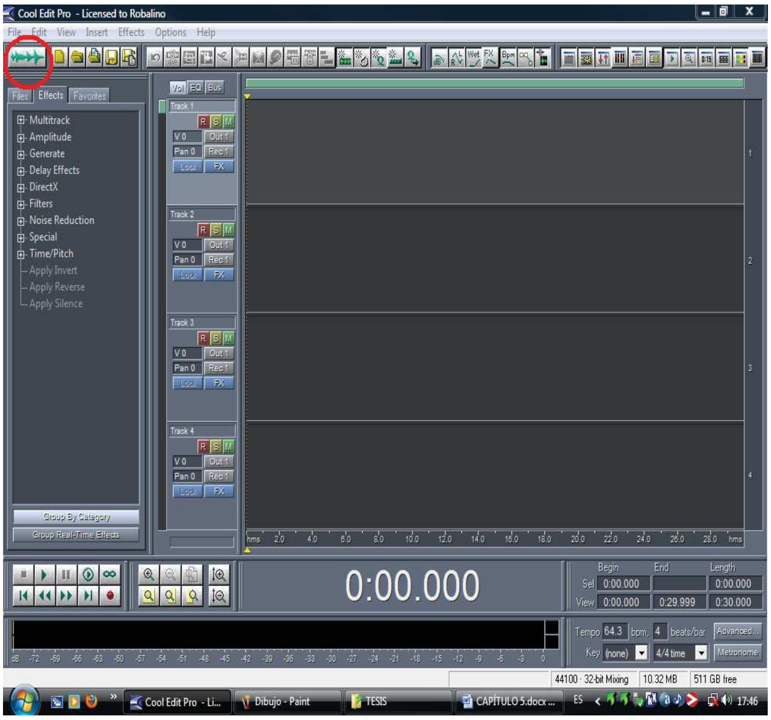

FRANKLIN
UNP
4 ipod
LETTERBOX TEMPLATE2.psd
Equipo
LabVIEW.exe - Acceso directo
Pro Tools M-Powered
Adobe After Effects CS3
FL Studio 9
Photoshop.exe - Acceso directo
Reason
labview screen pics
tonos
tonos
Buscar
Organizar
Vistas
Explorar
Vínculos favoritos
Documentos
Imágenes
Música
Más
Carpetas
Acceso público
Equipo
Red
Panel de control
Papelera de reciclaje
4 ipod
labview screen p
tonos
1kHz
2kHz
4kHz
8kHz
250Hz
500Hz
Nombre
250Hz
Carpeta de archivos
Fecha modificación: 26/01/2011 17:28
250Hz
Reproducir todo
Grabar
Nombre
Intérpretes
Álbum
FRANKLIN
11 elementos
Dibujo - Paint
TESIS
tonos
250Hz
CAPÍTULO 5...
17:58

Fuente: Franklin Robalino

Una vez creadas las carpetas, se usará el programa *Cool Edit Pro 2.0* (Este programa es una versión antigua de lo que ahora es *Adobe Audition*) para grabar los archivos de los diferentes tonos. Cualquier otro programa de edición de audio también podrá ser usado en vez de *Cool Edit Pro 2.0*, los únicos requisitos que debe tener el programa de edición de audio a usarse, debe ser que pueda producir tonos puros en estéreo y se pueda cambiar la amplitud, frecuencia y duración de los tonos generados.

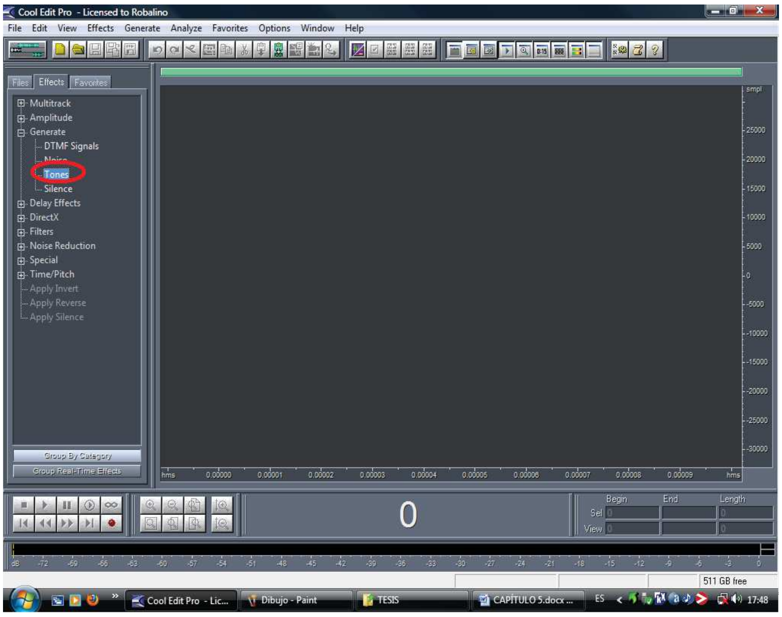
Al abrir el programa *Cool Edit Pro 2.0*, se abre la ventana *multitrack*, como se muestra en la figura 4.20. Después mediante el botón “*Switch to edit view*” (marcado con un círculo rojo en la figura 4.20) se cambiará a la pantalla de edición (ver figura 4.21).

Figura 4.20.- ventana multitrack


Cool Edit Pro - Licensed to Robalino
File Edit View Insert Effects Options Help
Files Effects Favorites
Multitrack
Amplitude
Generate
Delay Effects
DirectX
Filters
Noise Reduction
Special
Time/Pitch
Apply Invert
Apply Reverse
Apply Silence
Group By Category
Group Real-Time Effects
Track 1
Track 2
Track 3
Track 4
0:00.000
Begin End Length
Sel 0:00.000 0:00.000
View 0:00.000 0:29.999 0:30.000
Tempo 64.3 bpm 4 beats/bar Advanced...
Key (none) 4/4 time Metronome
44100 · 32-bit Mixing 10.32 MB 511 GB free
Cool Edit Pro - Li...
Dibujo - Paint
TESIS
CAPÍTULO 5.docx ...
17:46

Fuente: Franklin Robalino

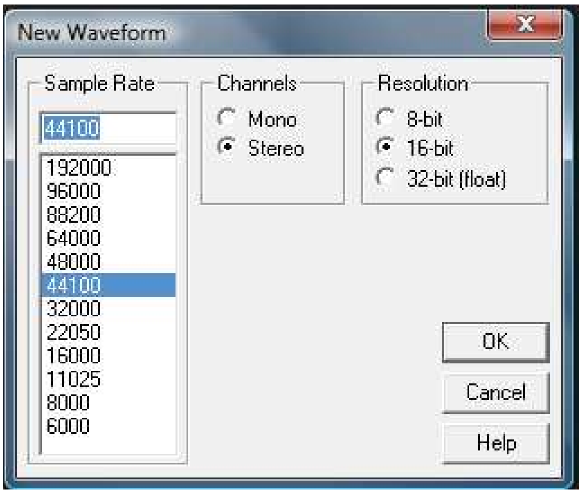
Figura 4.21.- ventana de edición


Cool Edit Pro - Licensed to Robalino
File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects Favorites
Multitrack
Amplitude
Generate
DTMF Signals
Noise
Tones
Silence
Delay Effects
DirectX
Filters
Noise Reduction
Special
Time/Pitch
Apply Invert
Apply Reverse
Apply Silence
Group By Category
Group Real-Time Effects
0
Begin End Length
Sel
View
511 GB free
Cool Edit Pro - Lic...
Dibujo - Paint
TESIS
CAPÍTULO 5.docx ...
17:48

Fuente: Franklin Robalino

En la pantalla de edición se entrará en la parte de *Effects-Generate-Tones*, como se muestra en la figura 4.21. Al hacer click en "*Tones*" se abrirá un cuadro de diálogo "*new waveform*", como se muestra en la figura 4.22, en este cuadro se podrá escoger entre varias opciones, las cuales son características principales del audio. Se escogerá un *sample rate* de 44100 con una resolución de 16 bit con canales en estéreo, la razón que se creará en estéreo es para poder controlar el volumen en el canal L y R por separado.

Figura 4.22.- cuadro de diálogo "new waveform"



Sample Rate
44100

Channels
○ Mono
● Stereo

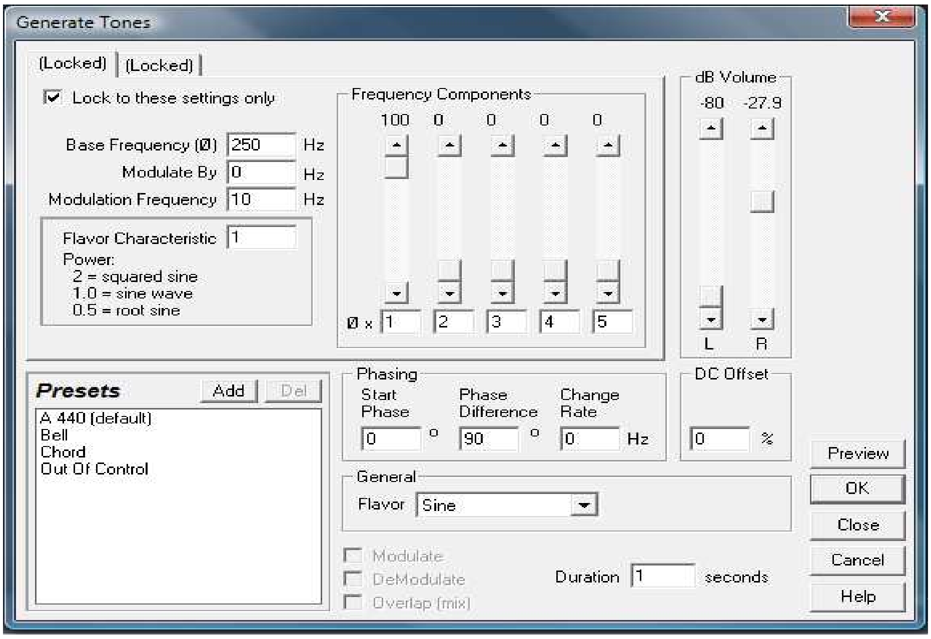
Resolution
○ 8-bit
● 16-bit
○ 32-bit (float)

Fuente: Franklin Robalino

Al aplastar el botón OK en el cuadro de diálogo "*new waveform*", se abrirá un segundo cuadro de diálogo llamado "*generate tones*", como se muestra en la figura 4.23. En este cuadro de diálogo se podrá controlar varios parámetros del tono que será generado, en este caso sólo se modificará la frecuencia, duración y amplitud en ambos canales.

La frecuencia se modifica en el casillero llamado "*Base Frequency*" que se encuentra en la parte superior izquierda de la pantalla, mientras que la amplitud se modifica con los sliders llamados "*dB Volume*" ubicados en la parte superior derecha, donde se tiene dos *sliders*: uno para el canal izquierdo (L) y otro para el canal derecho (R).

Figura 4.23.- cuadro de diálogo “generate tones”


Generate Tones
(Locked)
(Locked)
Frequency Components
Presets
Add
Del
Phasing
General
Preview
OK
Close
Cancel
Help

☑ Lock to these settings only

Base Frequency (Ø) 250 Hz

Modulate By 0 Hz

Modulation Frequency 10 Hz

Flavor Characteristic 1

Power:
2 = squared sine
1.0 = sine wave
0.5 = root sine

Frequency Components: 100 0 0 0 0

Ø x 1 2 3 4 5

dB Volume: -80 (L) -27.9 (R)

Presets:
A 440 (default)
Bell
Chord
Out Of Control

Start Phase 0 °

Phase Difference 90 °

Change Rate 0 Hz

DC Offset 0 %

Flavor Sine

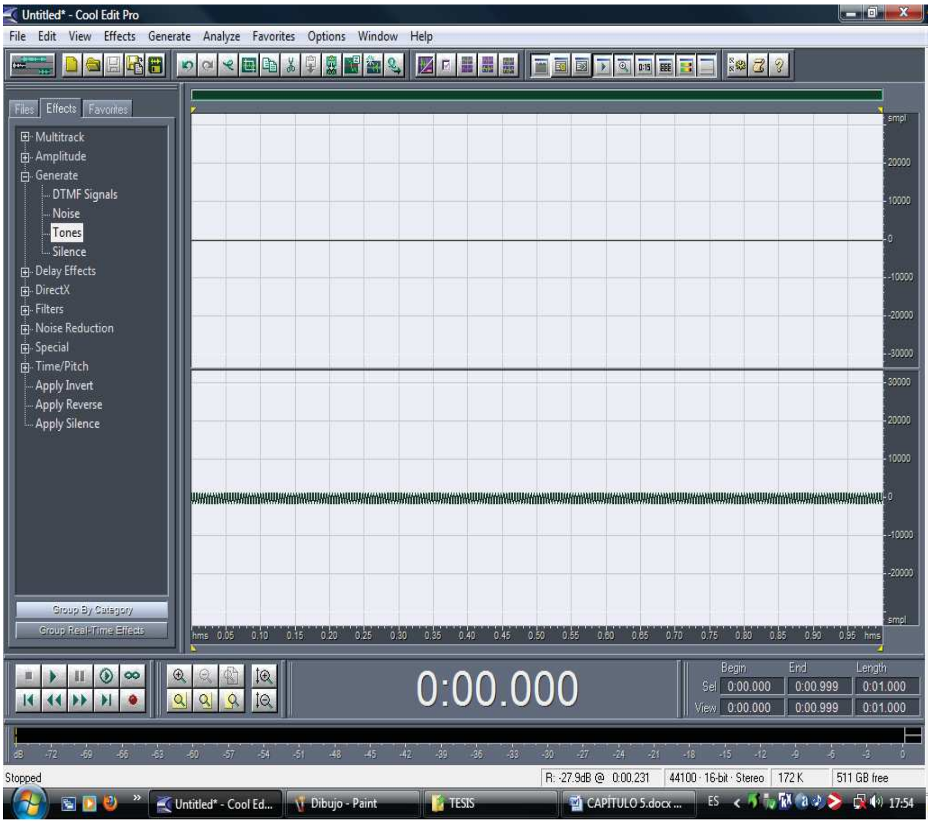
☐ Modulate
☐ DeModulate
☐ Overlap (mix)

Duration 1 seconds

Fuente: Franklin Robalino

Si se modifica la frecuencia a 250 Hz, la duración (en la parte inferior derecha del cuadro de diálogo) a 1 segundo, los sliders a -80 en L y -27.9 en R, como se muestra en la figura 4.23 y aplastamos el botón OK, se creará el tono en la pantalla de edición, como se puede observar en la figura 4.24.

Figura 4.24.- tono generado en la pantalla de edición

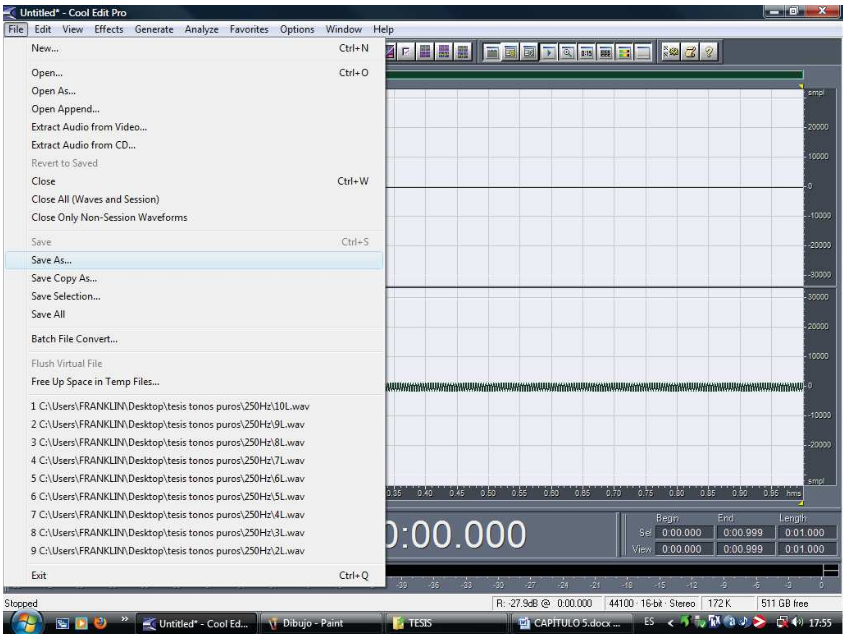

Untitled* - Cool Edit Pro
File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects Favorites
Multitrack
Amplitude
Generate
DTMF Signals
Noise
Tones
Silence
Delay Effects
DirectX
Filters
Noise Reduction
Special
Time/Pitch
Apply Invert
Apply Reverse
Apply Silence
Group By Category
Group Real-Time Effects
0:00.000
Begin End Length
Sel 0:00.000 0:00.999 0:01.000
View 0:00.000 0:00.999 0:01.000
Stopped
R: -27.9dB @ 0:00.231
44100 · 16-bit · Stereo
172 K
511 GB free

Fuente: Franklin Robalino

En la pantalla de edición se podrá observar que sólo se generó un tono en el canal R con la duración de 1 segundo y con una amplitud de -27.9. Una vez que el tono este en esta pantalla se podrá grabar dentro de los folders creados en el escritorio del computador.

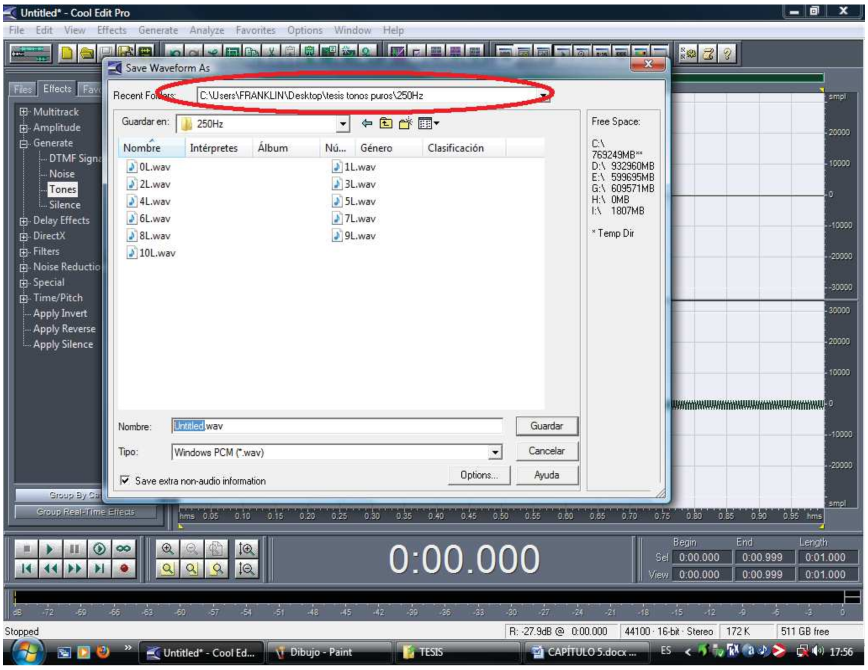
Para poder grabar los tonos se debe ingresar a *File-Save As…*, como se muestra en la figura 4.25. Al hacer esto se abrirá una ventana llamada "*save waveform as*", como se muestra en la figura 4.26, dentro de esta ventana se buscará la carpeta indicada donde será grabado el tono, por ejemplo en este caso deberá ir en la carpeta tonos, dentro de la carpeta 250 Hz y con el nombre según el valor de dB HL que tenga (para ver los valores de dB HL referirse a la parte 4.3 titulada calibración).

Figura 4.25.- grabando el tono


Untitled* - Cool Edit Pro
File Edit View Effects Generate Analyze Favorites Options Window Help
New... Ctrl+N
Open... Ctrl+O
Open As...
Open Append...
Extract Audio from Video...
Extract Audio from CD...
Revert to Saved
Close Ctrl+W
Close All (Waves and Session)
Close Only Non-Session Waveforms
Save Ctrl+S
Save As...
Save Copy As...
Save Selection...
Save All
Batch File Convert...
Flush Virtual File
Free Up Space in Temp Files...
1 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\10L.wav
2 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\9L.wav
3 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\8L.wav
4 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\7L.wav
5 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\6L.wav
6 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\5L.wav
7 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\4L.wav
8 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\3L.wav
9 C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz\2L.wav
Exit Ctrl+Q
0:00.000
Begin End Length
Sel 0:00.000 0:00.999 0:01.000
View 0:00.000 0:00.999 0:01.000
Stopped
R: -27.9dB @ 0:00.000
44100 · 16-bit · Stereo
172 K
511 GB free
Untitled* - Cool Ed...
Dibujo - Paint
TESIS
CAPÍTULO 5.docx ...
17:55

Fuente: Franklin Robalino

Figura 4.26.- ventana "save waveform as"


Untitled* - Cool Edit Pro
File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects
Multitrack
Amplitude
Generate
DTMF Signa
Noise
Tones
Silence
Delay Effects
DirectX
Filters
Noise Reductio
Special
Time/Pitch
Apply Invert
Apply Reverse
Apply Silence
Save Waveform As
Nombre Intérpretes Álbum Nú... Género Clasificación
0L.wav
2L.wav
4L.wav
6L.wav
8L.wav
10L.wav
1L.wav
3L.wav
5L.wav
7L.wav
9L.wav
Guardar
Cancelar
Options...
Ayuda
0:00.000
Begin End Length
Sel 0:00.000 0:00.999 0:01.000
View 0:00.000 0:00.999 0:01.000
Stopped
R: -27.9dB @ 0:00.000
44100 · 16-bit · Stereo
172 K
511 GB free
Untitled* - Cool Ed...
Dibujo - Paint
TESIS
CAPÍTULO 5.docx ...
17:56

Recent Folders: C:\Users\FRANKLIN\Desktop\tesis tonos puros\250Hz

Guardar en: 250Hz

Free Space:
C:\ 769249MB**
D:\ 932960MB
E:\ 599695MB
G:\ 609571MB
H:\ 0MB
I:\ 1807MB
* Temp Dir

Nombre: Untitled.wav

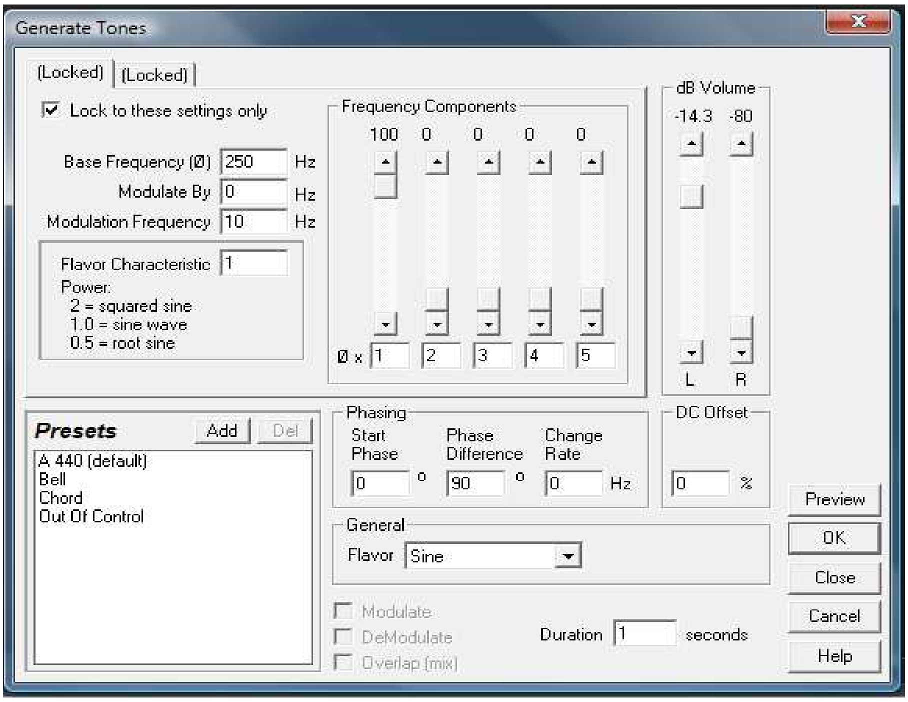
Tipo: Windows PCM (*.wav)

☑ Save extra non-audio information

Fuente: Franklin Robalino

Una vez grabado el primer tono se repetirá el proceso para las demas frecuencias y amplitudes para el oído derecho (R) y luego para el oído izquierdo (L). Para el oído izquierdo se deberá controlar la amplitud con el slider (L), mientras el slider (R) permanece al mínimo (-80). Por ejemplo en el cuadro de diálogo "generate tones" se ubicará el slider (L) a -14.3 manteniendo la misma frecuencia y duración anterior, es decir 250 Hz y un segundo, como se muestra en la figura 4.27.

Figura 4.27.- cuadro de diálogo



☑ Lock to these settings only

Base Frequency (Ø) 250 Hz

Modulate By 0 Hz

Modulation Frequency 10 Hz

Flavor Characteristic 1

Power:
2 = squared sine
1.0 = sine wave
0.5 = root sine

Frequency Components: 100 0 0 0 0

Ø x 1 2 3 4 5

dB Volume: -14.3 -80 (L R)

Phasing: Start Phase 0 °, Phase Difference 90 °, Change Rate 0 Hz

DC Offset 0 %

A 440 (default)
Bell
Chord
Out Of Control

General: Flavor Sine

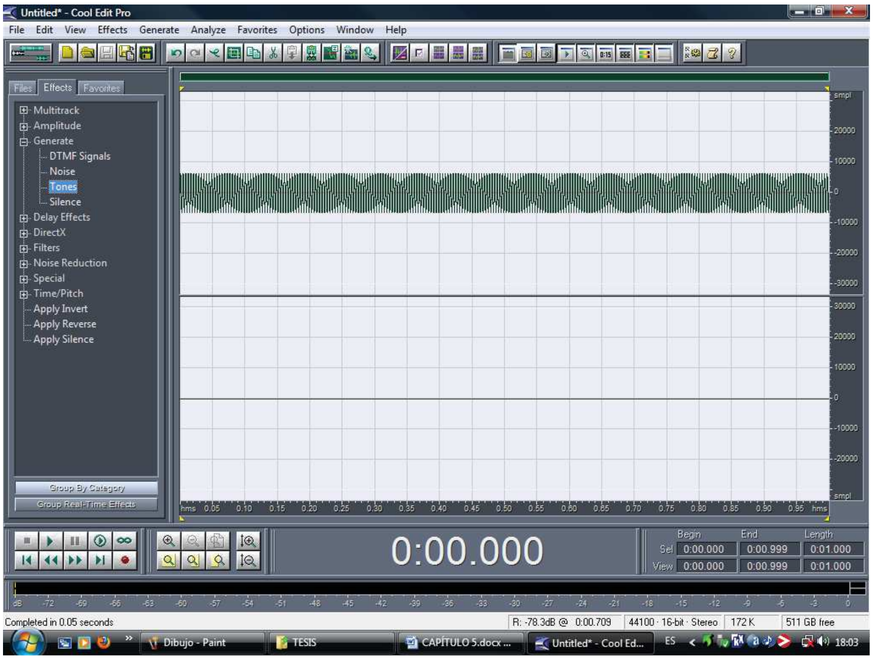
☐ Modulate
☐ DeModulate
☐ Overlap (mix)

Duration 1 seconds

Fuente: Franklin Robalino

Al aplastar el botón OK, se generará el tono en la ventana de edición, como se muestra en la figura 4.28. si se observa esta ventana, se notará que ahora el tono se generó sólo para el canal (L), y con una amplitud de -14.3, a simple vista se podrá notar que la amplitud es mayor si la comparamos con la de la figura 4.25, donde la amplitud era de -27.9.

Figura 4.28.- tono en la ventana de edición


Untitled* - Cool Edit Pro
File
Edit
View
Effects
Generate
Analyze
Favorites
Options
Window
Help
Files
Effects
Favorites
Multitrack
Amplitude
Generate
DTMF Signals
Noise
Tones
Silence
Delay Effects
DirectX
Filters
Noise Reduction
Special
Time/Pitch
Apply Invert
Apply Reverse
Apply Silence
Group By Category
Group Real-Time Effects
0:00.000
Begin
End
Length
Sel
View
0:00.000
0:00.999
0:01.000
R: -78.3dB @ 0:00.709
44100 · 16-bit · Stereo
172 K
511 GB free
Dibujo - Paint
TESIS
CAPÍTULO 5.docx ...
Untitled* - Cool Ed...
18:03

Completed in 0.05 seconds

Fuente: Franklin Robalino

Con esta explicación básica de cómo los tonos son generados y grabados en las carpetas que leerá el programa hecho en LabVIEW, se procederá a la calibración de dichos tonos, para tener las amplitudes exactas en dB HL.

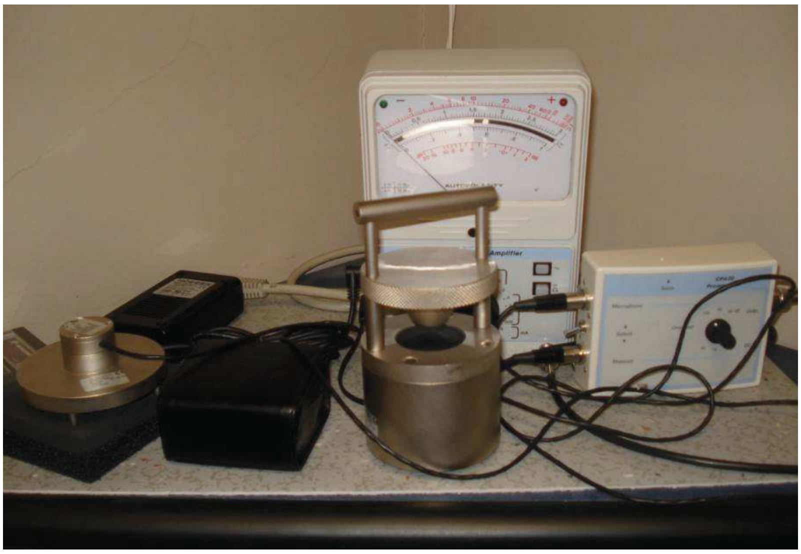
4.3. CALIBRACIÓN

Para realizar la calibración del programa al igual que todo audiómetro, éste debe ser calibrado según el estándar S3.6 de la norma ANSI (*American National Standards Institute*), con un equipo especializado.

En el Ecuador las calibraciones de los audiómetros se realizan en la compañía Pro Audio en donde el audiómetro, que en este caso será el programa diseñado en LabVIEW, fue llevado para su calibración. La compañía Pro Audio posee el instrumento de calibración para los audiómetros, el cual respeta la norma ANSI, pues es calibrado en Dinamarca para asegurar su correcto funcionamiento.

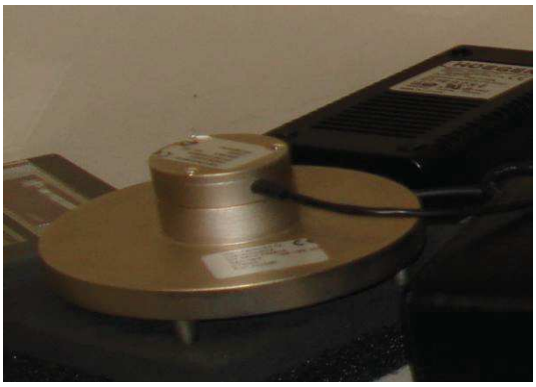
El equipo que ellos utilizan para su calibración se muestra en la figura 4.29, y consiste en unos audífonos de respuesta plana que se sellan herméticamente con el micrófono, el cual está conectado a un selector/preamplificador, que manda la señal al amplificador, donde la señal en dB HL será mostrada en el *display* análogo.

Figura 4.29.- equipo de calibración



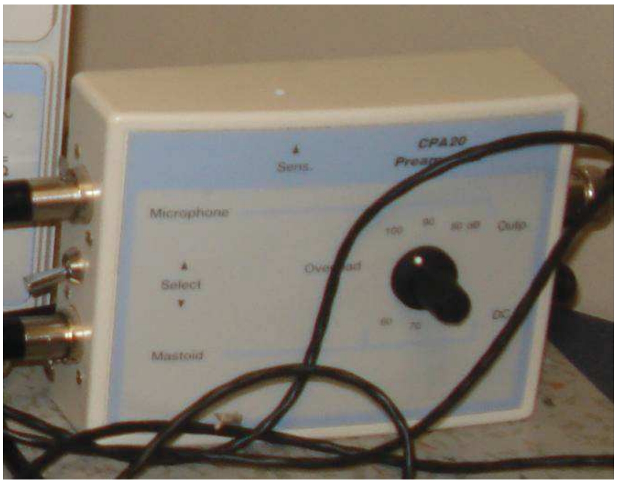
Fuente: Franklin Robalino

Figura 4.30.- micrófono para calibración



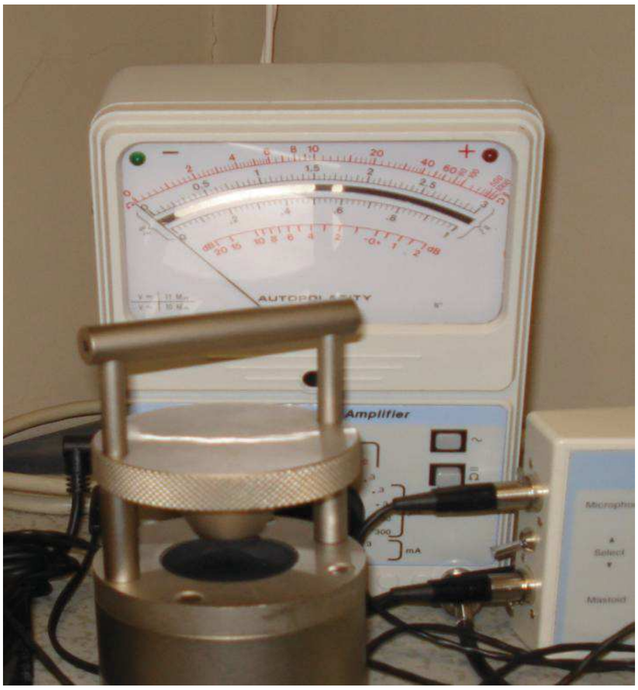
Fuente: Franklin Robalino

Figura 4.31.- selector/preamplificador


CPA20
Pream
Sens.
Microphone
Select
Mastoid
Overload
100
90
80 dB
Outp
60
70
DC

Fuente: Franklin Robalino

Figura 4.32.- amplificador con display análogo


AUTOPOLARITY
Amplifier
Microphone
Select
Mastoid

Fuente: Franklin Robalino

Se recomienda que los audiómetros sean calibrados por lo menos una vez al año. Todo audiómetro debe ser calibrado, el programa Software de Estimación de Pérdida Auditiva no es la excepción, pues cada computadora tendrá una tarjeta diferente de sonido, con diferentes características, tales como respuesta de frecuencia y volumen.

4.4. PROTOTIPOS

El primer prototipo realizado, fue programado para generar solamente dos tonos con diferentes amplitudes y para un solo oído. La razón de hacer esto primero era para verificar que las herramientas del programa estén funcionando correctamente con los dos primeros tonos. De esta manera si se encontraba algún problema con la programación sería más fácil de detectar el error y arreglarlo, antes de incluir a los demás tonos y los gráficos.

Una vez terminado el primer prototipo, se encontró que tenía problemas dentro del *loop* pues la secuencia tenía que terminar antes de detenerse por ende si se escuchaba la frecuencia y se aplastaba el botón el *loop* no se detenía hasta completar todas las amplitudes dentro de este *loop*.

En el segundo prototipo, se reemplazó la estructura de secuencia por un *case structure* para resolver el problema del *loop*, al correr el programa este tenía problemas debido a que una vez que se escuchaba la frecuencia y se aplastaba el botón el *loop* no se detenía hasta ejecutar el siguiente *case*.

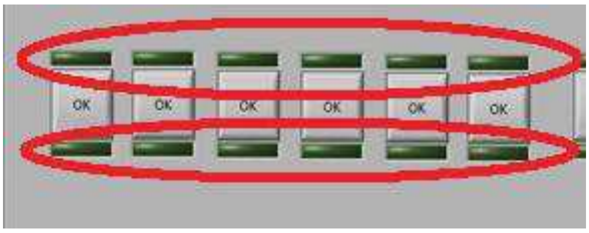
En el tercer prototipo se programó diferente el código dentro de cada *case* para que una vez que se escuchaba la frecuencia y se aplastaba el botón el *loop* no ejecutara otro tono más sino que iría a la siguiente frecuencia, de esta maneara funcionaría al igual que cualquier audiómetro.

En el cuarto prototipo se añadió la parte gráfica y se calibraron los tonos, para de esta manera dejar al Software de Estimación de Pérdida Auditiva funcionando correctamente y bajo norma.

4.5. MANUAL DE USUARIO

Una vez instalado el programa (con la carpeta de tonos) y calibrado los tonos en la computadora a ser usada se procederá al uso del programa mediante los siguientes pasos:

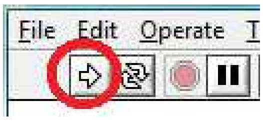
1.- Abrir el programa y conectar los audífonos (los audífonos deben ser de respuesta plana y herméticos). En este punto los focos LED en el programa estarán apagados.



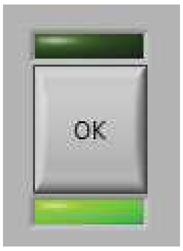
2.- Colocarse los audífonos bien ajustados.

3.- Asegurarse que no haya ruido externo que pueda afectar los resultados de la medición (ruido de: tráfico, voces, ventiladores, etc).

4.- Correr el programa (oprimir botón Run).

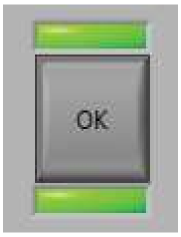


5.- El foco inferior (LED) del primer botón OK (en la parte inferior izquierda de la pantalla) se prenderá indicando al usuario que está activado.

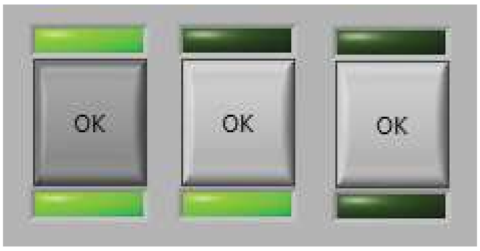


6.- Al escuchar el tono, oprimir el botón OK que está activado.

7.- Se prenderá el foco superior del botón OK para indicar que fue registrado el valor dentro del programa.

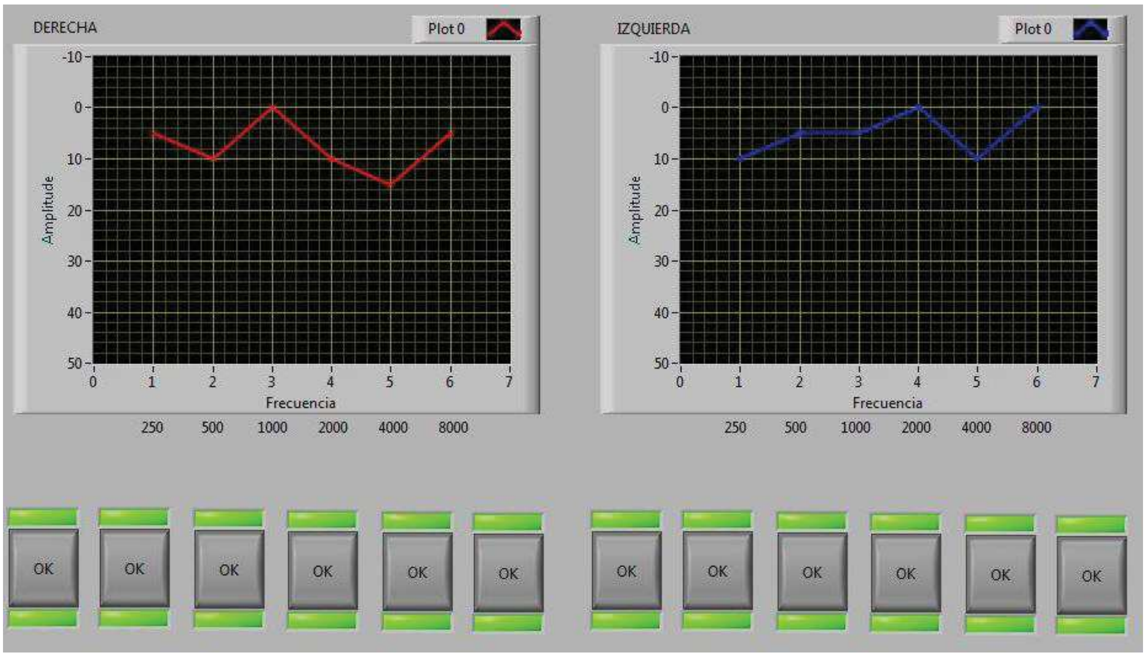


8.- Luego se activará el botón del lado derecho (el segundo botón).



9.- Esperar al siguiente tono y repetir los pasos del 6 al 8, hasta oprimir el último botón (los botones se activan secuencialmente de izquierda a derecha).

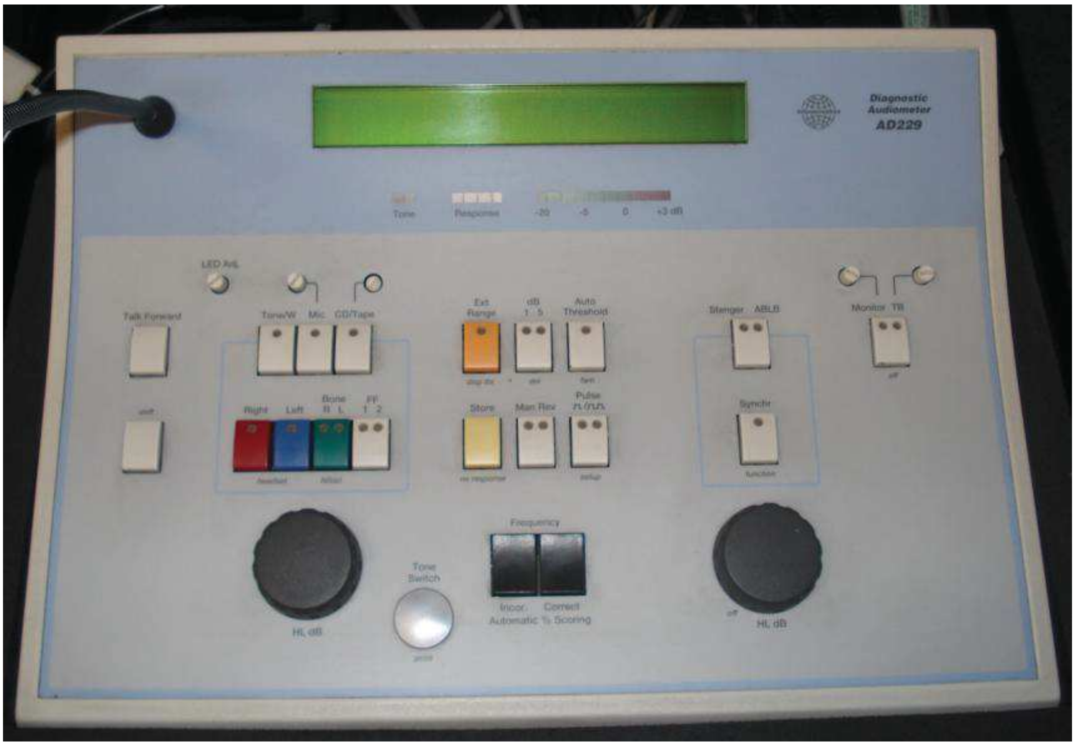
10.- una vez que el último botón sea oprimido, el programa graficará los resultados automáticamente en la pantalla.


DERECHA
Plot 0
Amplitude
-10
0
10
20
30
40
50
0
1
2
3
4
5
6
7
Frecuencia
250
500
1000
2000
4000
8000
IZQUIERDA
Plot 0
Amplitude
-10
0
10
20
30
40
50
0
1
2
3
4
5
6
7
Frecuencia
250
500
1000
2000
4000
8000
OK

4.6. COMPROBACIÓN DE RESULTADOS

En esta sección se comparan los resultados obtenidos del Software de Estimación de Pérdida Auditiva con los resultados obtenidos del audiómetro análogo *Diagnostic Audiometer AD229* del Centro Médico Meditrópoli (ver figura 3.33).

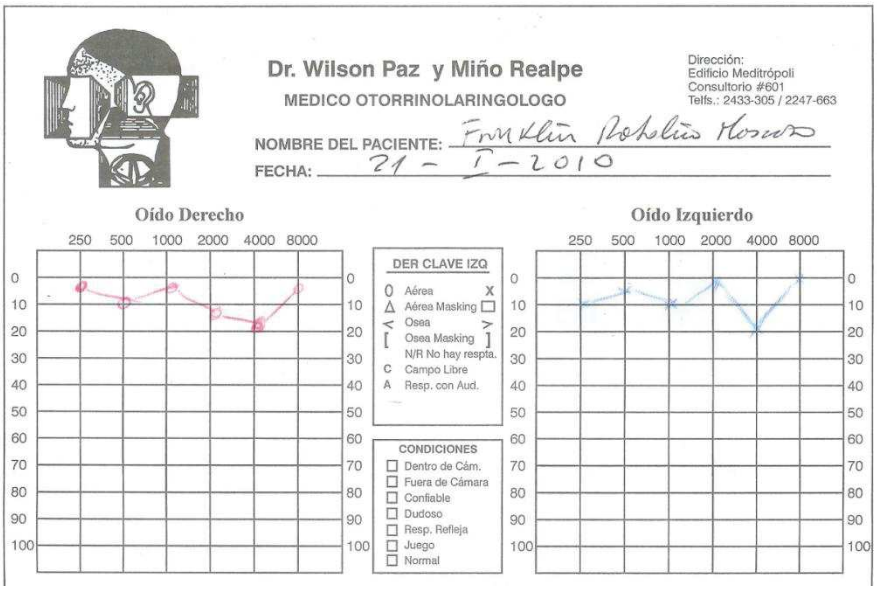
Figura 4.33.- Diagnostic Audiometer AD229 fabricado por Interacoustics


Diagnostic
Audiometer
AD229
Tone
Response
-20
-5
0
+3 dB
LED A/L
Talk Forward
Tone/W
Mic
CD/Tape
Ext Range
dB 1 5
Auto Threshold
Stenger
ABLB
Monitor
TB
Right
Left
Bone R L
FF 1 2
Store
Man Rev
Pulse
Synchr
Frequency
Tone Switch
HL dB
Incor.
Correct
Automatic
% Scoring
off
HL dB

Fuente: Franklin Robalino

La audiometría fue realizada a mi persona en el Centro Médico Meditrópoli por el Dr. Wilson Paz y Miño con el Diagnostic Audiometer AD229 y con audífonos PELTOR H7A, en donde los resultados fueron los siguientes.

Figura 4.34.- resultados de audiometría en el Centro Médico Meditrópoli



Dr. Wilson Paz y Miño Realpe
MEDICO OTORRINOLARINGOLOGO

Dirección:
Edificio Meditrópoli
Consultorio #601
Telfs.: 2433-305 / 2247-663

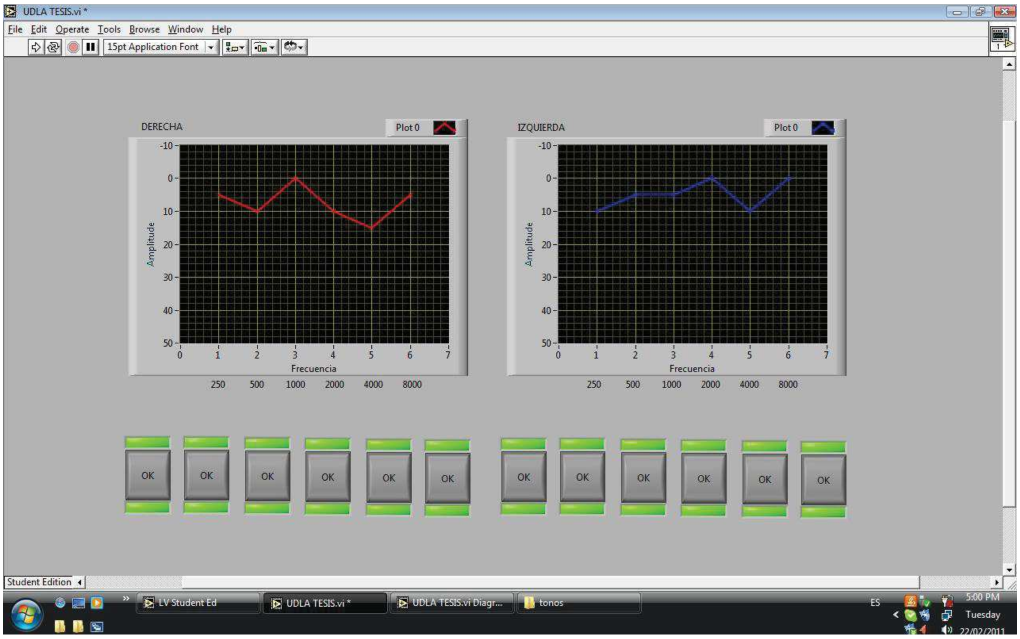
NOMBRE DEL PACIENTE: Franklin Robalino Moscoso
FECHA: 21 - I - 2010

CONDICIONES
☐ Dentro de Cám.
☐ Fuera de Cámara
☐ Confiable
☐ Dudoso
☐ Resp. Refleja
☐ Juego
☐ Normal

Fuente: Franklin Robalino

Después me realicé yo mismo la audiometría con el Software de Estimación de Pérdida Auditiva en mi apartamento con audífonos PHILIPS SHP2700, en donde los resultados fueron los siguientes.

Figura 4.35.- resultados con el software

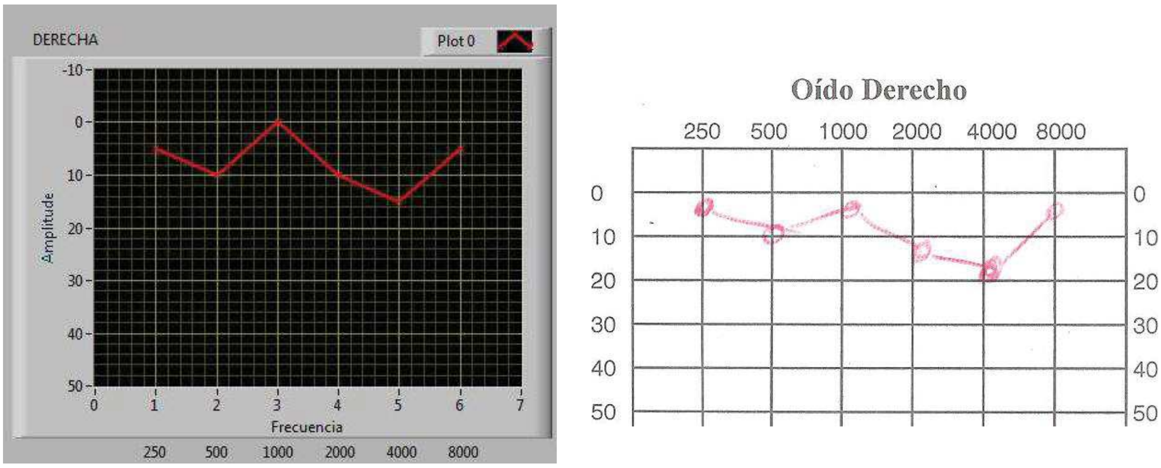


UDLA TESIS.vi *
File Edit Operate Tools Browse Window Help
15pt Application Font
DERECHA
Plot 0
IZQUIERDA
Plot 0
Amplitude
-10 0 10 20 30 40 50
0 1 2 3 4 5 6 7
Frecuencia
250 500 1000 2000 4000 8000
OK
Student Edition
LV Student Ed
UDLA TESIS.vi *
UDLA TESIS.vi Diagr...
tonos
ES
5:00 PM
Tuesday
22/02/2011

Fuente: Franklin Robalino

Al comparar los resultados del oído derecho, se podrá observar que los resultados son exactos con excepción de las frecuencias de 1000 Hz y 2000 Hz en donde existen diferencias de 5 dB HL.

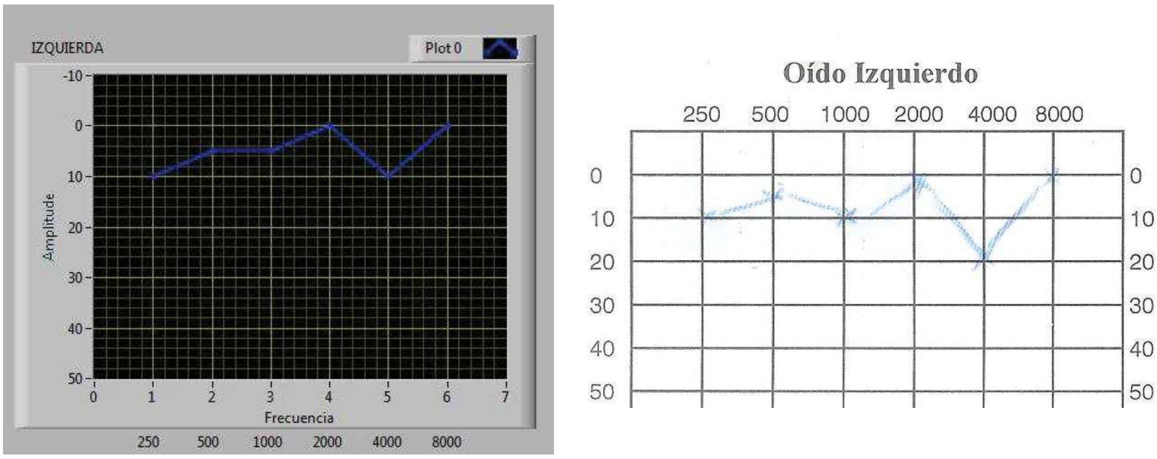
Figura 4.36.- resultados del oído derecho



Fuente: Franklin Robalino

Al comparar los resultados del oído izquierdo, se podrá observar que los resultados son también exactos con excepción de las frecuencias de 1000 Hz y 4000 Hz en donde existen diferencias de 5 dB HL y 10 dB HL respectivamente.

Figura 4.37.- resultados del oído izquierdo



Fuente: Franklin Robalino

Al analizar las diferencias mínimas en los resultados, se viene a la mente una posible causa, pues el ruido de fondo como se mencionó en el segundo capítulo influye en la medición, así que se optó por medir el ruido de fondo en los dos lugares donde ocurrieron las audiometrías, dando como resultado un nivel equivalente de NPS en el Centro Médico Meditrópoli de 62.4 dB(A) y en el apartamento un nivel equivalente de NPS de 54.7 dB(A). Estos niveles de ruido afectan a la medición, pues ruidos de carros o gente en el exterior pueden causar enmascaramiento (referirse al capítulo 2). Es por esto que se recomienda realizar las audiometrías en lugares que no sean ruidosos.

Figura 4.37.- sonómetro usado para medir ruido de fondo

Fuente: Franklin Robalino

CAPÍTULO 5

CONCLUSIONES

1. Los resultados que se obtuvieron con el software de Estimación de Pérdida Auditiva fueron muy similares a los resultados del audiómetro análogo AD229 del Centro Médico Meditrópoli.

2. Fue muy importante comparar el software de Estimación de Pérdida Auditiva con el audiómetro análogo AD229 del Centro Médico Meditrópoli pues de esta manera se pudo verificar el correcto funcionamiento del software.

3. Se pudo verificar que el software funcionará correctamente en cualquier computadora siempre que el programa en la computadora haya sido calibrado. Si el programa no se calibra los resultados pueden ser incorrectos, debido a que cada tarjeta de sonido funciona diferente, es por esto que la calibración es muy importante.

4. Las mínimas diferencias entre los resultados del software de Estimación de Pérdida Auditiva y del audiómetro análogo AD229 del Centro Médico Meditrópoli son causadas por ruido de fondo durante la medición, pues ruidos de carros o gente en el exterior causan enmascaramiento. Por esto se recomienda realizar las audiometrías en lugares que no sean ruidosos.

5. El software de Estimación de Pérdida Auditiva funcionará siempre con el programa LabVIEW, la razón por la que no se pudo grabar el programa como archivo ejecutable (.exe), es decir, para que funcione sin la presencia de labVIEW es porque para poder grabarlo como ejecutable se necesita la versión profesional del programa que cuesta alrededor de

$1,249.00 dólares, y por falta de dinero se trabajó con la versión más económica de LabVIEW, la versión *Student Edition*.

6. Hay que tener en cuenta también que el usuario tendrá que instalar los archivos con los tonos que leerá el programa de manera manual, esto es un grave inconveniente por el momento, debido a que si el usuario no los graba en el lugar correcto, el programa no los leerá y por ende no funcionará correctamente.

7. Para poder solucionar este problema en el software de Estimación de Pérdida Auditiva se debería grabar el programa para que sea ejecutable (con la versión profesional de LabVIEW). De esta manera la carpeta con los tonos se instalará automáticamente siempre en un solo lugar dentro de la computadora, asegurando el correcto funcionamiento del programa.

8. Para mejorar el prototipo se podría instalar una interface (hardware), que permita al usuario, usar solamente un control individual para presionar cada vez que escuche el tono, la razón por la que no se implementó aquí es porque sería mucho gasto para el usuario.

9. Los costos para elaborar este software fueron los siguientes:
   - HP *Pavilion* dv67000 *notebook* PC con un valor de $718.00

   - CD driver externo "*Portable Super Multidrive*" marca LG con un valor de $119.99

   - audífonos de respuesta plana *PHILIPS* SHP2700 con un valor de $40.00

   - programa LabVIEW *Student Edition* 6i con un valor de $178.49

   - programa *Cool Edit 2.0* para generar tonos *freeware*

   - la calibración del *software* tiene un valor de $250.00

BIBLIOGRAFÍA

[1] **ALBERNI**, DR. JUAN, Y OTROS, "Guía Medica", Vol. 9, Salvat, 1984.

[2] **ANDRADE**, DR. EDWIN, Otorrinolaringólogo, Entrevistas, 2011.

[3] **BISHOP**, ROBERT H., "LabVIEW *Student Edition 6i*", *Prentice Hall*, 2001.

[4] **BURKARD**, ROBERT, "Development *of ANSI Standards: An Audiologist's View*", asha.org, 2004, (15-03-11)

[5] **HAIN**, TIMOTHY C. MD, "*audiometry*", dizziness-and-balance.com, 2009, (15-03-11)

[6] **HAIN**, TIMOTHY C. MD, "*Hearing Testing*", dizziness-and-balance.com, 2009, (15-03-11)

[7] **HARRISON**, TINSLEY RNDOLF, "Principios de la Medicina Interna", 17ª Edición Vol.1, Mc Graw-Hill, 2008.

[8] **HUDSON**, JACK, **LUECKE**, JERRY, "*Basic Communications Electronics*", Master Publishing, 1999.

[9] **KINSLER**, LAWRENCE E., "Fundamentos de Acústica", Limusa, 1992.

[10] **LINDEBURG**, MICHAEL R., "*Engineer-in-Training Reference Manual*", *Eighth Edition, Professional Publications*, 2002.

[11] **MARDONES**, FELIPE, apuntes "Guía del Sistema Auditivo Humano".

[12] **MIMS III**, FORREST M., "*Electronic Formulas, Symbols and Circuits*", *First Printing*, 2000.

[13] **MIYARA**, FEDERICO, "Acústica y Sistemas de Sonido", URN, 2006.

[14] **MONREAL**, JOSÉ LUIS, "Diccionario Enciclopédico Océano Uno", Ediciones OCÉANO, 1991.

[15] PAZ Y MIÑO R., DR. WILSON, Otorrinolaringólogo, Entrevistas, 2010-2011.

[16] PAZ Y MIÑO R., DR. WILSON JR., Otorrinolaringólogo, Entrevistas, 2011.

[17] ROBALINO, FRANKLIN, apuntes de la UDLA y NVCC, 2000-2010.

[18] ROCAMORA, M., "Física del sonido", eumus.edu.uy, 1995, (15-03-11)

[19] VORVICK, LINDA J. MD, "Audiometría", nlm.nih.gov, 2011, (15-03-11)

"*audiometer*", discoveriesinmedicine.com, 2011, (15-03-11)

"*audiometry*", audioelectronicsinc.com, 2010, (15-03-11)

"*audiometers*", maico-diagnostics.com, 2011, (15-03-11)

"audiómetro", worldlingo.com, 2011, (15-03-11)

"*The Nobel Prize in Physiology or Medicine 1961*", nobelprize.org, 2011, (15-03-11)

"*audiometry*", hsmc.co.uk, 2010, (15-03-11)

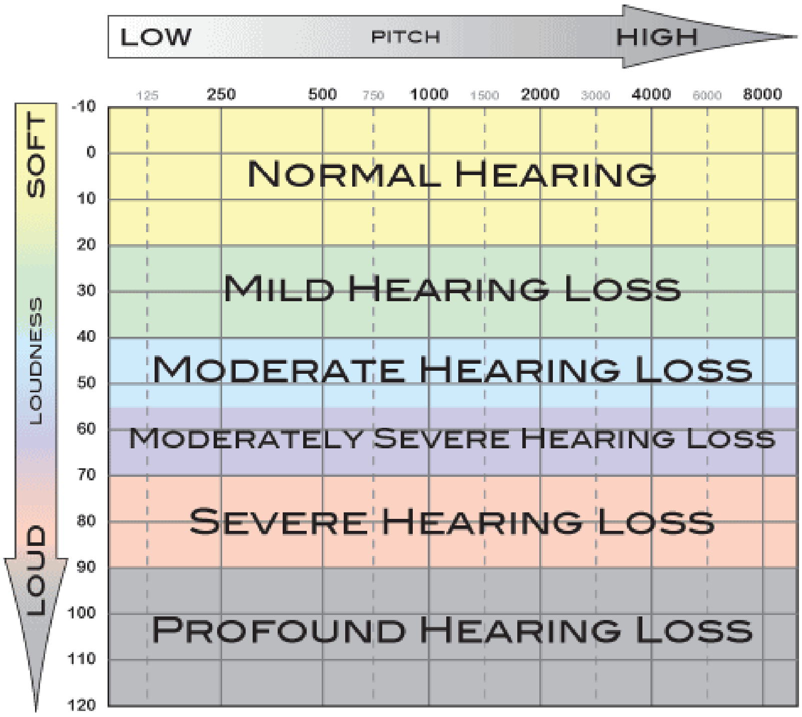
"*audiometry*", nlm.nih.gov, 2011, (15-03-11)

"audiometria subjetiva", hearing.siemens.com, 2011, (15-03-11)

"audiometria objetiva", hearing.siemens.com, 2011, (15-03-11)

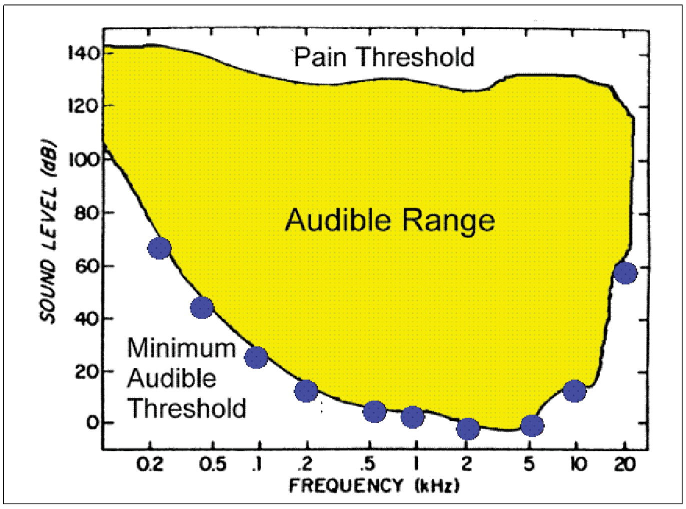
ANEXOS

Guía para los resultados de una audiometría



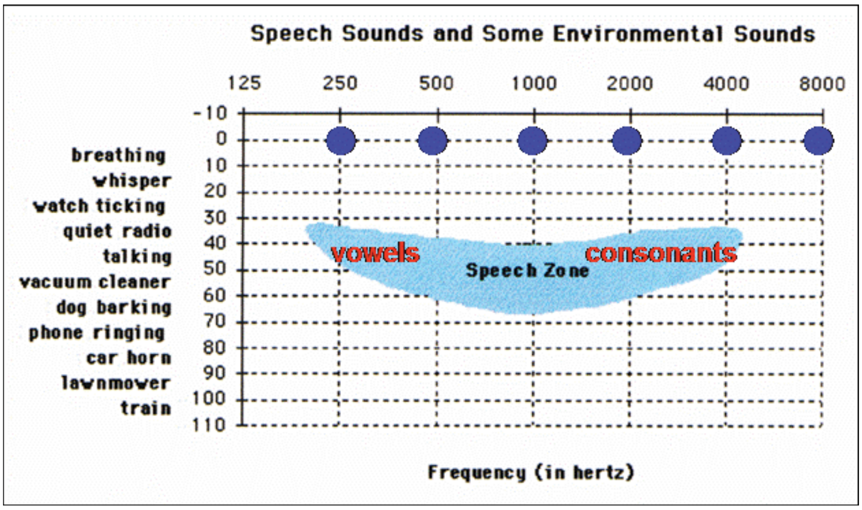
Fuente: pedsent.com

Gráfico del rango audible humano



Fuente: d.umn.edu

Gráfico del rango hablado humano



Fuente: d.umn.edu

Tabla de conversión de dB SPL a dB HL

Frequency (Hz)	dB SPL	dB HL
250	15.0	0.0
500	9.0	0.0
1000	3.0	0.0
2000	-3.0	0.0
4000	-4.0	0.0
8000	13.0	0.0

Fuente: tonmelster.ca