



**BENEMÉRITA  
UNIVERSIDAD AUTÓNOMA DE  
PUEBLA**

---

---

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**ELECTROENCEFALÓGRAFO CON  
COMUNICACIÓN VÍA BLUETOOTH USANDO  
MICROCONTROLADOR DE 32 BITS**

**TESIS**

**PARA OBTENER EL TÍTULO DE:**

**INGENIERO EN CIENCIAS DE LA  
COMPUTACIÓN**

**PRESENTA:  
RICAÑO ZÚÑIGA RODOLFO**

**ASESOR:  
QUIROZ HERNÁNDEZ NICOLÁS**

**PUEBLA, PUE.**

**2013**

## AGRADECIMIENTOS

Le agradezco primeramente a Dios por haberme acompañado y guiado a lo largo de mi carrera, y fortalecerme en mis momentos de debilidad, sin El nada fuera posible.

Le doy gracias a mis padres Rodolfo y Ana por estar siempre ahí pendientes de mis necesidades apoyándome con amor y paciencia

Quiero agradecerles a mis hermanas Nayeli y Tere por animarme a terminar todo, a pesar de las circunstancias y a los compañeros y amigos que me acompañaron en estos años.

Agradezco al Maestro Nicolás Quiroz, mi asesor de tesis por su grande apoyo en la elaboración de la misma. Reconozco que ha sido un trabajo arduo, maestro gracias.

También agradezco a todos aquellos maestros que de una u otra forma estuvieron conmigo compartiendo sus conocimientos en estos 5 años de mi vida y que me han formado como profesionista.

Por ello agradezco a la universidad por abrirme sus puertas y regalarme en cada maestro un cúmulo de conocimiento.

# ÍNDICE

---

---

Contenido	Página
Capítulo 1. Fundamentos Teóricos .....	1
1.1 PLANTEAMIENTO DEL PROBLEMA .....	2
1.2 OBJETIVOS.....	3
1.2.1 OBJETIVO GENERAL .....	3
1.2.2 OBJETIVO ESPECIFICO .....	4
1.3 MARCO TEÓRICO .....	4
1.4 ESTADO DEL ARTE.....	10
1.4.1 ÁREAS DE ESTUDIO .....	12
1.5 CARACTERÍSTICAS Y RESTRICCIONES DEL TRABAJO.....	19
1.6 APORTACIONES DEL TRABAJO .....	19
1.7 DESCRIPCIÓN DEL DOCUMENTO.....	20
Capítulo 2. Tarjeta de Adquisición de Datos .....	21
2.1 REQUERIMIENTOS DE HARDWARE.....	22
2.1.1 AMPLIFICADOR DE INSTRUMENTACIÓN .....	23
2.1.2 FILTRO BUTTERWORTH.....	27
2.1.3 CONVERSIÓN DE NIVELES DE VOLTAJE (OFFSET).....	32
2.1.4 DIGITALIZACIÓN.....	35
2.1.5 TRANSMISIÓN DE DATOS.....	40
Capítulo 3. Desarrollo de Software.....	52
3.1 REQUERIMIENTOS DEL SOFTWARE .....	53
3.2 MÓDULO BLUETOOTH .....	53
3.2.1 CLASE PARA CONTROL DEL MÓDULO BLUETOOTH.....	56
3.3 VISUALIZACIÓN DE DATOS ENVIADOS POR LA PLACA DE DESARROLLO. ....	69
3.3.1 CLASE QUE GRAFICA EL ARREGLO DE DATOS .....	70
3.3.2 CLASE QUE SE ENCARGA DEL CONTROL DE DATOS.....	72

3.4 INTERFAZ DE USUARIO.....	74
3.5 CLASES COMPLEX Y FFT (TRANSFORMADA RÁPIDA DE FOURIER)...	76
3.5.1 CLASE COMPLEX.....	77
3.5.2 CLASE FFT (TRANSFORMADA RÁPIDA DE FOURIER) .....	78
3.6 MANEJO DE DATOS.....	81
3.6.1 MÉTODO QUE SE ENCARGA DE GUARDAR LOS DATOS EN UN ARCHIVO. ....	81
3.6.2 LECTURA DE DATOS.....	83
Capítulo 4. Pruebas y Resultados .....	87
4.1 APLICACIÓN .....	88
4.2 HARDWARE.....	91
4.3 SOFTWARE .....	94
4.4 FUNCIONAMIENTO COMPLETO DEL SISTEMA. ....	98
Capítulo 5. Conclusiones y Trabajo a Futuro. ....	108
Capítulo 6. Bibliografía.....	110

## ÍNDICE FIGURAS

---

Figura 1.3.1. Diferentes tipos de ondas electroencefalograficas normales. ....	5
Figura 1.4.1 Figura Comparativa.....	12
Figura 1.4.1.2.1 Diagrama a bloques de un microcontrolador .....	16
Figura 2.1.1 Diagrama General del Sistema .....	22
Figura 2.1.1 Etapa de acoplamiento.....	23
Figura 2.1.1.1 INA114.....	25
Figura 2.1.1.2 Esquema INA114 .....	26
Figura 2.1.2.1 Respuesta de filtros de Butterworth de varios órdenes.....	28
Figura 2.1.2.1.1 Configuración del filtro pasa bajo Butterworth .....	32
Figura 2.1.3.1 Acondicionador de señal .....	33
Figura 2.1.4.1 Diagrama general interno de un microcontrolador .....	36
Figura 2.1.4.2 Diagrama de flujo para el control del PIC .....	38
Figura 2.1.4.3 Simulación en Proteus .....	39
Figura 1.3.2.3.3.1 Pila de protocolos del Bluetooth .....	45
Figura 2.1.5.1 Diagrama de pines y módulo Bluetooth HC-05.....	48
Figura 2.1.5.2 Configuración básica para el módulo Bluetooth.....	50
Figura 2.1.5.3 Diseño Final del sistema .....	51
Figura 3.2 Diagrama de flujo general para el sistema completo .....	55
Figura 3.2.1.1 Diagrama de flujo general para el funcionamiento de la recepción y captura de datos .....	56
Figura 3.2.1.1.1 Código para la localización de dispositivos .....	58
Figura 3.2.1.2.1 Código para la localización de los servicios que ofrece cada dispositivo. ....	61
Figura 3.2.1.2.2 Código llamado cuando se completa la búsqueda de dispositivos .....	62
Figura 3.2.1.3.1. Diagrama de flujo que indica cómo se obtienen los datos .....	65
Figura 3.2.1.3.2. Código que se encarga de llamar el método que se encarga de buscar los servicios.....	66

Figura 3.2.1.3.3. Inicialización de los flujos de datos para la comunicación Bluetooth.....	66
Figura 3.2.1.3.4. Inicialización del hilo.....	67
Figura 3.2.1.3.5. Método que se manda a llamar con el hilo .....	67
Figura 3.2.1.3.6. Método que se encarga de la adquisición de datos .....	68
Figura 3.3.1. Diagrama de flujo del método que se encarga de representación de datos.....	69
Figura 3.3.1.1. Código que dibuja los arreglos de datos en el lienzo.....	71
Figura 3.3.2.1. Código que se encarga de dibujar línea e inicializar las variable necesarias para la FFT. ....	72
Figura 3.3.2.2. Continuación de 3.3.2.1 que se encarga ahora de ajustar la FFT para su representación. ....	73
Figura 3.3.2.2. Manejo de los valores para poder llenar el arreglo .....	74
Figura 3.4.1 GUI de la aplicación .....	75
Figura 3.4.2. Acción realizada por el radiobutton Fourier .....	76
Figura 3.4.3. Control de datos .....	76
Figura 3.5.1.1. Constructor que vuelve de real a complejo.....	78
Figura 3.5.1.2. Funciones que regresan tanto parte real como imaginaria .....	78
Figura 3.5.2.1. Función de la transformada rápida de Fourier .....	80
Figura 3.6.1.1 Método que se encarga de almacenamiento de datos .....	81
Figura 3.6.1.2 Método que se encarga agregar los datos del paciente y generar los respectivos archivos de cada canal .....	82
Figura 3.6.1.3 Método que se encarga de crear el archivo.....	83
Figura 3.6.1.4 Método que se encarga de guardar datos en el archivo .....	83
Figura 3.6.2.1 Método que se encarga de la lectura de datos .....	84
Figura 3.6.2.2 Método que se encarga de leer datos del archivo .....	84
Figura 3.6.2.3 Evento del Ratón.....	86
Figura 4.1.1 Diagrama que muestra la forma de conectar los electrodos en base al sistema internacional 10-20 .....	89
Figura 4.1.2 Sujeto de pruebas con electrodos ya implantados .....	90
Figura 4.1.3 Algunos sujetos que se usaron para las pruebas .....	91
Figura 4.2.1 Primeras pruebas del electroencefalógrafo .....	92

Figura 4.2.3. Encefalograma de sujeto femenino de 21 años, estudiante con migraña ligera .....	93
Figura 4.2.4. Diseño de placa donde se localizan acondicionamiento y digitalización .....	94
Figura 4.3.1 Señal donde se demuestra que la frecuencia corresponde a la producida por el generador de funciones .....	95
Figura 4.3.2. Comparación entre la señal mostrada en el osciloscopio y la señal mostrada en el software de las frecuencias sinusoidales de 30 y 100 Hz .....	96
Figura 4.3.3 FFT de una señal sinusoidal de 30 Hz .....	98
Figura 4.2.3.1 Señal mostrada tanto en ELVISII como en software de Java .....	100
Figura 4.2.3.2 Comparación entre las señales obtenidas por el software y un electroencefalograma obtenido de una revista de neuro-psiquiatría.....	101
Figura 4.2.3.3. Imagen normal y aplicando zoom x3 sobre el eje y, donde con los círculos se muestra los puntos donde más claramente se muestra esto .....	102
Figura 4.2.3.4. Imagen donde se muestra que campos llenar, para luego introducir el botón agregar .....	103
Figura 4.2.3.5. Imagen donde se muestra los archivos generados después de agregar los campos.....	104
Figura 4.2.3.6. Localización de los botones de guardar y guardar como .....	105
Figura 4.2.3.7. Lista datos del paciente .....	106
Figura 4.2.3.8. Electroencefalogramas obtenidos de archivos previamente guardados .....	107

## ÍNDICE TABLAS

---

---

Tabla 1.3.1 Tipos de onda cerebral .....	9
Tabla 1.4.1 Características Electroencefalógrafo .....	11
Tabla 2.1.1.1 Relación de ganancias con su respectiva resistencia .....	27
Tabla 2.1.5.2 Ancho de Banda de Bluetooth .....	41
Tabla 2.1.5.1 Pines del Módulo HC-05.....	49
Tabla 3.2.1.2.1. UUIDs comúnmente utilizadas.....	63
Tabla 4.1.1 Sujetos de pruebas que se usaron para las pruebas .....	90
Tabla 4.3.1 Pruebas que se hicieron con el programa para obtener datos del generador de funciones .....	97
Tabla 4.3.2 Pruebas de la transformada de Fourier .....	98

# Capítulo 1. Fundamentos Teóricos

## Introducción

Un electroencefalógrafo es un equipo que mide las señales bioeléctricas por medio de electrodos que van adheridos al cuero cabelludo, dichas señales se procesan y posteriormente se muestran en algún monitor o equipo de cómputo para ser estudiadas por un neurólogo. En este capítulo se dará un panorama general del presente trabajo donde se describen los objetivos generales, específicos, el estado actual del arte y el contenido de cada uno de los siguientes capítulos.

## 1.1 PLANTEAMIENTO DEL PROBLEMA

La electroencefalografía es una técnica de exploración neurofisiológica que registra la actividad bioeléctrica cerebral. Un encefalograma (EEG) es el registro obtenido mediante esta técnica. Con el EEG se pueden observar los distintos tipos de ondas cerebrales en estado de vigilia, durante el sueño, en reposo o bajo ciertas condiciones de activación [1].

Un electroencefalograma sirve como ayuda al diagnóstico cuando un paciente sufre convulsiones, un traumatismo craneoencefálico, un tumor, o incluso una enfermedad degenerativa del sistema nervioso. También se utiliza para evaluar los trastornos del sueño. Esto se realiza midiendo las señales bioeléctricas del cerebro. También se puede utilizar para formar un banco de EEG donde especialistas puedan identificar las principales características de los trastornos relacionado con el cerebro, para autenticación biométrica de individuos [2], negatividad relacionada a la retroalimentación de error (*feedback*) [3]; así como estudios concretos sobre el consumo de algunos estimulantes tales como el alcohol [4], drogas [5] [6], diagnóstico de distintos tipos de epilepsia [7], localización de tumores cerebrales, otras lesiones cerebrales voluminosas, etc. A pesar de que estas son básicamente las funciones importantes del electroencefalógrafo, por ahora únicamente se utilizara para analizar la actividad eléctrica de una persona con características normales (sin alguna tipo de enfermedad o trastorno en el cerebro), enfocándonos más en la comunicación inalámbrica.

En base a lo anterior, se requiere un sistema hardware-software, para la adquisición y acondicionamiento de las bioseñales; digitalización, comunicación y representación de las mismas.

Se requiere revisar circuitos propuestos de bioseñales (acondicionamiento, filtración, etc) para saber cómo se tratarán las señales. También revisar la

digitalización que se usa para poder enviar la información por un canal de comunicación digital que sea flexible, para su posterior procesamiento y visualización gráfica de dichas señales. Se explicará en el resto del trabajo como se lleva esto a cabo.

En los últimos años, los dispositivos móviles han tenido gran impacto en la sociedad esto gracias a la movilidad. Día tras día el hombre busca poder llegar a aplicar esto a todos los ámbitos cotidianos, siendo el sector salud no una excepción.

La predisposición en las comunicaciones actuales es de tipo inalámbrico, ya que los dispositivos que tienen esta característica no utilizan ningún tipo de cable, permitiendo tener una mejor versatilidad a la hora de poder comunicar varios dispositivos a las vez. En este caso, se busca adaptar el sistema a esas nuevas tendencias para así poder facilitar la tarea de hacer un análisis sin estar en el mismo lugar que el paciente. Esto se hace con el objetivo de poder hacer un chequeo constante de la persona. También se desea que a largo plazo esto permita tener una mayor compatibilidad con cualquier tipo de dispositivo portátil para una mayor comodidad.

## 1.2 OBJETIVOS

### 1.2.1 OBJETIVO GENERAL

Construir un electroencefalógrafo para ayudar a realizar un diagnóstico de algunos desórdenes neurológicos tales como los diferentes tipos de epilepsia, utilizando un módulo inalámbrico (Bluetooth) que envíe la señal a una computadora para su interpretación por un especialista.

### 1.2.2 OBJETIVO ESPECIFICO

Se implementaran las siguientes etapas

1. Diseñar una etapa de acondicionamiento para las señales cerebrales que incluirá una etapa de filtraje para eliminar ruido.
2. Digitalizar las señales cerebrales utilizando un ADC con resolución de 10 bits por medio del microcontrolador.
3. Diseñar una tarjeta controladora del proceso de digitalización y del módulo Bluetooth a través del microcontrolador de 32 bits.
4. Implementar un programa para el procesamiento de la señal bioeléctrica y visualización de los electroencefalogramas, dicho programa también maneja la comunicación con la tarjeta controladora.

### 1.3 MARCO TEÓRICO

El cerebro es un órgano del sistema nervioso, abundante en neuronas con funciones especializadas, constituyendo la parte más grande del encéfalo. Se divide en dos hemisferios o mitades llamadas hemisferios cerebrales. Las áreas dentro del cerebro controlan las funciones musculares y también controlan el habla, el pensamiento, las emociones, la lectura, la redacción y el aprendizaje.

Las células cerebrales (neuronas) se comunican produciendo diminutos impulsos eléctricos. La intensidad y calidad de esta actividad eléctrica dependen en gran parte de la excitación global del encéfalo. La intensidad y la calidad de esta actividad eléctrica dependen en gran parte de la excitación global del

encéfalo resultante de funciones del sistema reticular central. Las ondulaciones de los trazos eléctricos mencionados, mostradas en la figura 1.3.1, se llaman ondas cerebrales

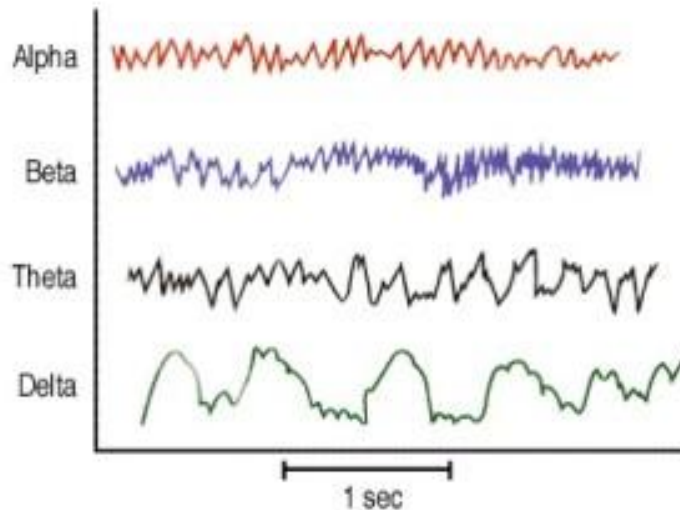


Figura 1.3.1 Diferentes tipos de ondas electroencefalográficas normales

La descarga de una neurona o fibra nerviosa única en el cerebro no puede registrarse desde el cuero cabelludo. En cambio, para que un potencial eléctrico pueda registrarse, desde la superficie del cráneo, es preciso que grandes cantidades de tejido nervioso produzcan simultáneamente corriente eléctrica. Esto puede ocurrir de dos maneras. En primer lugar, un enorme número de fibras nerviosas puede descargar sincrónicamente, dando lugar a corrientes eléctricas muy intensas. En segundo lugar, muchas neuronas pueden descargarse. Estas neuronas parcialmente descargadas pueden producir corrientes duraderas que fluctúan lentamente al cambiar el grado de excitabilidad de estas. Las mediciones eléctricas simultáneas en el encéfalo, mientras se registran ondas cerebrales desde el cuero cabelludo, indican que la causa habitual de las ondas cerebrales es el segundo de estos mecanismos.

La localización del origen de estas ondas es la superficie del cuero cabelludo, y se logra con la colocación de 16 electrodos distintos sobre el cuero

cabelludo, con lo cual se obtienen las ondas cerebrales. Para fines prácticos y de prueba, se usara un electroencefalógrafo que solo usara 4 canales, los cuales medirán la frecuencia del módulo frontal izquierdo y derecho, esto es suficiente para las pruebas que se harán. Más adelante se podrá usar este proyecto como base para hacer un electroencefalógrafo más completo, hasta llegar al de 16 canales.

Las intensidades de las ondas cerebrales en la superficie del cuero cabelludo varían de 0 a cerca de 300 microvoltios [8], sus frecuencias, de una en varios segundos hasta 500 más por segundo. Las características de estas ondas dependen estrechamente del grado de actividad de la corteza cerebral, y las ondas difieren considerablemente según haya vigilia o sueño.

La mayor parte del tiempo las ondas cerebrales son asincrónicas, y no se pueden distinguir en el encefalograma ningún patrón generalizado. Sin embargo, otras veces aparecen patrones característicos. Algunos de ellos se asocian con anomalías específicas del cerebro como la epilepsia. Otras pueden presentarse en individuos normales, que se pueden clasificar, como se muestra en la figura 1.3.1, en: Alfa, Beta, Theta, Delta

Ondas Beta: Se producen cuando el cerebro está despierto e implicado en actividades mentales. Son ondas amplias y las más rápidas de las cuatro. Su frecuencia oscila entre 15 y 40 ciclos por segundo. Denotan una actividad mental intensa. Cuando una persona está dando un discurso, estudiando, realizando un problema de matemáticas, etc. su cerebro se encuentra emitiendo este tipo de ondas. Sin embargo, la relación entre la actividad mental y la actividad eléctrica no es, ni con mucho, tan sencilla. Por ejemplo, si le pedimos que hagan un cálculo sencillo, como 143 menos 67, muchas personas reflejarán un aumento de nivel beta. Pero si les solicitamos que hagan por cálculo mental la raíz cuadrada de un número, esas mismas personas tenderán a generar más señales alfa.

Una posible explicación de ello es que todos sabemos restar, habiendo olvidado en cambio como se saca una raíz cuadrada. Al rebuscar en nuestra memoria lo que aprendimos al respecto en la escuela, la corteza cerebral queda en relativa inactividad y ello permite la aparición del nivel alfa.

Alrededor de un 10% de la población produce casi continuamente ondas beta y muy pocas alfa, ni siquiera con los ojos cerrados. Esto puede ser debido a un grado de ansiedad muy alto, pero también a una mente sumamente alerta y activa [9].

**Ondas Alfa:** Alfa representa un estado de no actividad y relajación. Son más lentas y de mayor amplitud que las beta. Su frecuencia oscila entre 9 y 14 ciclos por segundo. Una persona que ha terminado una tarea y se sienta a descansar, se encuentra a menudo en un estado alfa, así como la persona que está dando un paseo, disfrutando del paisaje. Se ha demostrado también que las personas que poseen ondas alfa bien definidas tienen más aptitud para la meditación y muestran facultades extrasensoriales. El finado C. Maxwell Cade, eminente investigador británico, identificó lo que llamó “un alfa zahorí”: ráfagas breves de baja tensión, que aparecen aproximadamente medio segundo antes de que se produzca la reacción del sensitivo [9].

**Ondas Theta:** Son ondas de mayor amplitud y menor frecuencia (entre 5 y 8 ciclos por segundo) que aparecen principalmente en las regiones parietal y temporal del cerebro (esto es, aproximadamente a la mitad del recorrido entre la frente y la nuca), y son más intensas en el lado izquierdo. Se trata de un estado de meditación profunda. La persona que está fantaseando o, como suele decirse, soñando despierta, se encuentra en este estado, así como la persona que tras conducir un rato, de repente se da cuenta de que no recuerda los últimos kilómetros. Es el estado más inspirador. Se cree también que van asociadas con los estados emotivos, porque las generan

los niños, sobre todo cuando ríen o lloran. En los adultos también aumenta su intensidad con motivo de alguna actividad agradable.

A menudo, una persona tiene las mejores ideas cuando predominan este tipo de ondas. Se trata de un estado en el que las tareas que realizas se han vuelto tan automáticas, que no necesitas tener un control consciente de su realización y puedes distanciarte de ellas mentalmente. Es decir, tu mente está en “otro sitio”. Las ideas que surgen en la mente en este estado fluyen con libertad y sin censura o culpa [9].

Ondas Delta: Son las ondas de mayor amplitud y menor frecuencia (entre 1,5 y 4 ciclos por segundo). Nunca llegan a cero, pues eso significaría la muerte cerebral. Es un estado de sueño profundo. Durante la actividad delta, las neuronas afectadas no pueden realizar ningún trabajo útil. A veces las ondas delta son de tanta duración que se sospecha que paralizan por electrocución la corteza, por decirlo así, y se puede especular acerca de si no será ésa su función específica en determinadas condiciones, exactamente como a veces la función del dolor estriba en la inmovilización de un miembro herido.

Un efecto similar se presenta en muchos animales cuando duermen, con el resultado de establecer una “desconexión” entre la mente y el cuerpo, lo mismo que el freno “de hombre muerto” desconecta los motores del tren si el maquinista se duerme o sufre un desmayo.

Al evitar que los músculos obedezcan a los impulsos muy poderosos del cerebro en sueños, se impide que los animales y los humanos se dañen a sí mismos, o a otros, durante el descanso [9].

En la Tabla 1.3.1 [10] se muestra los intervalos de frecuencia y situación mental correspondiente, además del voltaje que se alcanza en cada tipo de onda

TIPO DE ONDA	FRECUENCIA (Hz)	VOLTAJE ( $\mu$ V)	SITUACION MENTAL RELATIVA A LA QUE CORRESPONDE
DELTA	0.2 a 3.5	10-50	Estado hipnótico, hemisferio cerebral derecho en plena actividad, sueño profundo , meditación
THETA	3.5 a 7.5	50-100	Estado de vigilia, equilibrio entre los hemisferios izquierdo y derecho, plenitud, armonía,
ALFA	7.5 a 13	100-150	Relajación, tranquilidad, creatividad inicio de actividad plena del hemisferio izquierdo y desconexión del hemisferio derecho.
BETA	13 a 28	150-200	Estado de alerta máxima, vigilante, miedo, es la situación normal cuando estamos despiertos, conduciendo, o trabajando en donde estamos en estado de alerta, ansiedad.
RAM-ALTA	+ de 28 Hz	+200	Estado de stress y confusión.

Tabla 1.3.1 Tipos de Onda Cerebral

Los niveles de voltaje son muy pequeños (tabla 1.3.1) y son comparables con el ruido eléctrico, por lo tanto es importante utilizar no solamente un amplificador operacional sino un amplificador de instrumentación ya que, por su funcionamiento, realiza una resta de 2 señales que entran restando el ruido y dejando una señal nítida.

El electroencefalograma es el registro de la actividad cerebral en forma espontánea, a partir de la colocación de electrodos colocados sobre la cabeza o sea el cuero cabelludo, el cual informa sobre el funcionamiento eléctrico de las

neuronas en la corteza cerebral y sobre su mecanismo debajo de la corteza, es decir subcortical.

El registro de las ondas cerebrales en el electroencefalograma es útil en pacientes que presenten epilepsia o convulsiones, trastornos del sueño, en enfermedades neurológicas y enfermedades psiquiátricas, e inclusive en aquellos pacientes que presenten alteraciones del movimiento que pudiera confundirse con epilepsia.

Para lograr sacar este registro se usa un equipo médico denominado electroencefalógrafo. Para hacer esta lectura el electroencefalógrafo utilizan electrodos que se colocan en el cuero cabelludo. Estos electrodos transmiten la señal eléctrica analógica a un convertidor analógico digital que las filtra, amplifica y vuelve a filtrar para que sean desplegadas en una pantalla.

## 1.4 ESTADO DEL ARTE

El electroencefalógrafo es un equipo electrónico utilizado para el registro de la actividad bioeléctrica cerebral [11]. Actualmente los electroencefalogramas comerciales son de uso en investigación o médico, llegando a ser difícil su obtención. Entre la búsqueda que se realizó esta un electroencefalograma modelo N\_EC01 de alrededor de 300,000 pesos mexicanos, el cual tienes las siguientes características [12], que se muestran en la tabla 1.4.1.

1 Electroencefalógrafo Portátil de 19 canales	N_E7.5
1 Estimulador a Flash para Electroencefalógrafo Portátil	N_E7.5-31
1 Aplicación para Registro de EEG	N_E-SW-1.1-v
1 Aplicación para Edición de EEG	N_E-SW-1.2-v
Electrodos de Superficies Ag / AgCl (10 Ux3bolsa)	E_DSCS152600
Pasta Abrasiva (1 frasco)	E_EVERI160SPE
Pasta Conductora (1 frasco)	E_AC CREAM250V

Tabla 1.4.1 Características Electroencefalógrafo.

También se encontró varios modelos en una página de complementos médicos, los cuales son la mayor parte usados y/o no se encuentran en el país. Algunos de estos modelos se citan a continuación.

Electroencefalógrafo - Siemens - Mingograf EEG 10 con costo de 900 USD., el cual es usado y ha estado en funcionamiento durante 5 años, ignorando el estado actual, además de que su ubicación está en la ciudad de Falco, Venezuela [13].

Electroencefalógrafo - Nihon Kohden - EEG-9200 con costo de 12,582.65 USD y es de 32 Canales. No se menciona si el equipo es nuevo o usado, solo que se dan 24 meses de garantía y que su proveedor está ubicado en Porto, Portugal [14].

Electroencefalógrafo - Nihon Kohden - Neuropack MEB-9200 de costo de 8500.00 USD, el cual solo indica que está en perfectas condiciones y se ubica en California, Estados Unidos [15].

En cuanto al ámbito de investigación, se encuentran varios prototipos por ejemplo, un electroencefalógrafo que utiliza un INA114, 2 filtros activos y funciona

a base de baterías con interfaz bajo Visual Basic [16]. Otro proyecto que su diseño se ha miniaturizado hasta el punto de ser del tamaño de un centavo (figura 1.4.1) y está hecho en una placa de metal que es usado como sensor y utiliza un INA116 junto con otro amplificador para obtener una ganancia de 1000 además de un convertidor digital analógico de 18 bits [17]. También un prototipo que llega a manejar hasta cuatro canales y que contiene un amplificador INA114, un filtro de 60 Hz y aislamiento eléctrico por razones de seguridad [18], también un diseño que usa un amplificador de instrumentación hecho a base de amplificadores operacionales comerciales, con 2 etapas de filtrado y un convertidor analógico digital (microcontrolador) además de usar puerto serial y con interfaz hecha en MATLAB [19]. Y por último un electroencefalógrafo que funciona en tiempo real por medio de FPGA y un puerto serial con interfaz bajo MATLAB [20].

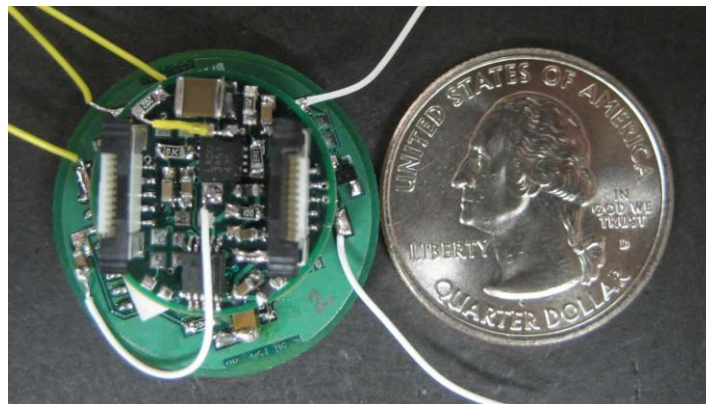


Figura 1.4.1 Imagen Comparativa

#### 1.4.1 ÁREAS DE ESTUDIO

Durante la indagación es necesario definir el campo de búsqueda en el que se ubica el trabajo, todo esto con la finalidad de delimitar las áreas de estudio para centrarse en puntos específicos referentes y que representa el problema en sí. La investigación nos lleva a tratar como puntos principales los siguientes temas:

- Acondicionamiento de la Señal

- Microcontrolador
- Modulo Bluetooth
- Procesamiento de la información

#### 1.4.1.1 ACONDICIONAMIENTO DE LA SEÑAL

Los impulsos eléctricos que tienen lugar en el cerebro son transmitidos hasta la superficie corporal. Basándonos en esta premisa, la colocación de una serie de sensores o electrodos en la piel permite detectar estas señales eléctricas y transformarlas en una representación gráfica. Esta representación consiste en una línea con ondulaciones, ángulos e inflexiones que representan las ondas del cerebro.

Para poder representar esta señal se dividirá en tres etapas que se explicaran a continuación

##### 1.4.1.1.1 PRIMERA ETAPA: DISPOSITIVO PARA AMPLIFICAR LA AMPLITUD, O POTENCIA, DE UNA SEÑAL ELÉCTRICA.

Se utiliza para ampliar la señal eléctrica débil captada por los electrodos, también es conocido como bioamplificadores. En esta aplicación, el amplificador de instrumentación construido con operacionales TI084 opera a una ganancia diferencial de 7642 [19] que amplifica a las ondas eléctricas cerebrales de ambos hemisferios que se encuentran en el orden de 10uV a 100uV [8] y que son captadas por 4 electrodos ubicados en zonas de la corteza cerebral: frontal, central y occipital, de acuerdo a lo que se requiera analizar. En esta etapa de amplificación accionan dos filtros pasivos pasa altos cuya frecuencia límite inferior cubre el objetivo central de eliminar los niveles de voltaje DC presentes y característicos del cuerpo humano, en este caso presentes en el cerebro.

#### 1.4.1.1.2 SEGUNDA ETAPA: FILTRACIÓN DE LAS SEÑALES BIOELÉCTRICAS.

La etapa siguiente comprende a los filtros activos pasa bajos en este caso los Butterworth de tercer orden cuyo límite superior de la frecuencia es de 30 Hz, en nuestro caso el rango de frecuencia de trabajo es de 0.5 a 30 Hz para tener una atenuación de 60 dB por década. La función central es eliminar las señales de impurezas o contaminantes de las señales que aparezcan, así como para atenuar la interferencia de la línea de transmisión a 50/60 Hz y sus armónicos que representan el mayor problema en la medición de biopotenciales, entre ellas existen:

- Interferencia: Contaminación por señales extrañas, generalmente artificiales y de forma similar a las de la señal.
- Ruido: Señales aleatorias e impredecibles de tipo eléctrico originadas en forma natural dentro o fuera del sistema. No puede ser eliminable completamente.
  - Extrínseco: Artificial atmosférico, galáctico o solar.
  - Intrínseco: Térmico, de disparo.
    - Ruido térmico: Existe debido a las variaciones (vibraciones) de moléculas y átomos en conductores y componentes electrónicos.
    - De disparo: Se debe al paso no uniforme de electrones o huecos en componentes electrónicos.
  - Distorsión: Alteración de la señal debido a la respuesta imperfecta del sistema, a diferencia del ruido y la interferencia, ésta desaparece cuando la señal deja de aplicarse.

#### 1.4.1.1.3 TERCERA ETAPA: ACONDICIONAMIENTO DE LA SEÑAL BIOELÉCTRICA.

Esta permite llevar la señal amplificada de las ondas cerebrales a niveles TTL, condición necesaria para que las señales EEG ingresen a los dos canales analógicos del microcontrolador.

#### 1.4.1.2 DIGITALIZACIÓN.

Recibe el nombre de microcontrolador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el microcontrolador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de microcontrolador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y dispositivos de E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

El diagrama a bloques de un microcontrolador se muestra en la figura 1.4.1.2.1

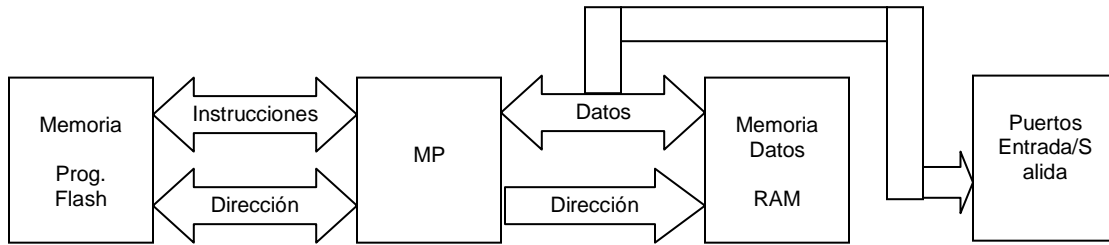


Figura 1.4.1.2.1 Diagrama a bloques de un microcontrolador

Un microcontrolador tiene 3 bloques principales que son:

Un microprocesador, puertos de entrada/salida y un tercer bloque que es la memoria que puede existir sólo una (Arquitectura Harvard) o dos memorias (Arquitectura Princeton).

El microprocesador realiza todo el control a través de las ordenes indicadas en el programa que se realiza para controlar algún proceso. El código del programa generalmente se almacena en la memoria Flash y los datos obtenidos en RAM.

Los dispositivos de entrada pueden ser un teclado, un interruptor, un sensor, etc.

Los dispositivos de salida pueden ser LED's, pequeños parlantes, zumbadores, interruptores de potencia (tiristores, optoacopladores), u otros dispositivos como relés, luces, un secador de pelo, en fin lo que quieras.

Para el uso de esta etapa se utilizará un microcontrolador, que cuenta con convertidos de analógico a digital, el cual se utilizará para lograr convertir los datos analógicos producidos por el cerebro a señales [21].

#### 1.4.1.3 Comunicación

La comunicación inalámbrica o sin cables es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. Hay varios tipos de comunicaciones que cumplen con este requisito tales como:

**Infrared Data Association (IRDA):** define un estándar físico en la forma de transmisión y recepción de datos por rayos infrarrojo.

**Wireless Fidelity (Wi-Fi):** Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: un ordenador personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso (o hotspot) tiene un alcance de unos 20 metros en interiores y al aire libre una distancia mayor. Pueden cubrir grandes áreas la superposición de múltiples puntos de acceso.

**Worldwide Interoperability for Microwave Access (Wi-MAX):** Es una norma de transmisión de datos que utiliza las ondas de radio en las frecuencias de 2,3 a 3,5 GHz.

**Global System for Mobile communications (GSM):** es un sistema estándar, libre de regalías, de telefonía móvil digital. Un cliente GSM puede conectarse a través de su teléfono con su computador y enviar y recibir mensajes por correo electrónico, faxes, navegar por Internet, acceder con seguridad a la red informática de una compañía (red local/Intranet), así como utilizar

otras funciones digitales de transmisión de datos, incluyendo el servicio de mensajes cortos (SMS) o mensajes de texto.

3G: es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles). Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz y datos (una llamada telefónica o una videollamada) y datos no-voz (como la descarga de programas, intercambio de correos electrónicos, y mensajería instantánea).

General Packet Radio Service (GPRS): Es una extensión del Sistema Global para Comunicaciones Móviles (Global System for Mobile Communications o GSM) para la transmisión de datos mediante conmutación de paquetes. Existe un servicio similar para los teléfonos móviles que es sistema IS-136. Permite velocidades de transferencia de 56 a 144 kbps.

Code Division Multiple Access (CDMA): Término genérico para varios métodos de multiplexación, o control de acceso al medio, basados en la tecnología de espectro expandido.

Bluetooth: Es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Esta frecuencia es libre, por lo tanto no se requiere ningún permiso para utilizarla.

Se optó por este último ya que es uno de los dispositivos utilizados tanto en computadoras como en dispositivos móviles de frecuencia libre, lo cual permite una mejor compatibilidad.

#### 1.4.1.4 Procesamiento de la Señal

Esta última fase se utiliza para poder representar los datos biomédicos enviados por el sistema para así poder hacer su posterior diagnóstico. Se procesan los datos que son enviados vía Bluetooth hacia un dispositivo, ya sea una computadora de escritorio, laptop o celular. Para hacer esto, se necesitara crear un programa en algún lenguaje de alto nivel. En este caso se usa Java, en la cual existe una API llamada bluecove que contiene las clases y funciones necesarias para que se pueda lograr la comunicación por medio del modulo Bluetooth de la computadora al módulo genérico que estará conectado con el microcontrolador. También se realizará todo lo necesario para graficar las señales bioeléctricas en tiempo y frecuencia.

### 1.5 CARACTERÍSTICAS Y RESTRICCIONES DEL TRABAJO

El electroencefalógrafo será de dos canales, el cual monitoreara la actividad de ambos lóbulos del cerebro. Consta de un microcontrolador que digitalizara la señal y la enviara a la computadora usando un modulo Bluetooth. Este dispositivo solo funcionara si el equipo al que se le envía la señal tiene el software, el cual graficara los datos para una posterior interpretación. Se planeo utilizar un microcontrolador de 32 bits, pero por problemas de diseño de la placa, al quedar los capacitores con diferente distancia de la que debían tener, además de cuestiones de tiempo, se decidió utilizar otro microcontrolador, en este caso un PIC18F4550.

### 1.6 APORTACIONES DEL TRABAJO

Implementar un sistema económico para observar electroencefalogramas en una computadora personal para ser interpretado por especialista, además de quitar la dependencia hacia los cables, para así poder tener una mayor flexibilidad

al momento de hacer un análisis del cerebro con la ayuda de pequeñas computadoras (microcontroladores) se puede digitalizar, procesar las señales, comunicarse con otras computadoras, etc., dando así robustez y control.

## 1.7 DESCRIPCIÓN DEL DOCUMENTO.

En el capítulo 2 se describe el diseño de la tarjeta de adquisición de datos, todo el hardware requerido para el electroencefalógrafo, en el capítulo 3 se describe todo lo relacionado con el software para adquirir los datos en la computadora mediante la comunicación Bluetooth, el procesamiento de los datos para graficar en tiempo y frecuencia, así como el diseño de la interfaz de usuario, en el capítulo 4 se mostrarán los resultados obtenidos del trabajo.

# Capítulo 2. Tarjeta de Adquisición de Datos

## Introducción

En este capítulo se describe el diseño del hardware para construir el electroencefalógrafo.

Una señal bioeléctrica es el resultado que se obtiene de obtener con electrodos los potenciales de acción, los cuales se producen como resultado de la actividad electroquímica de una cierta clase de células conocida como células excitables que son componentes del tejido celular, glandular o nervioso [22]. En este caso, las señales en la cual se enfocará el trabajo serán aquellas producidas por el cerebro.

Se recoge la señal por medio de electrodos conectados en la frente del paciente, para luego pasar a través de una etapa de acoplamiento, la cual amplifica, elimina el ruido, se ajusta el voltaje y se digitaliza con él microcontrolador, después es enviada al modulo Bluetooth por medio del puerto serial del microcontrolador, para después ser recogida por la computadora y luego ser procesada.

## 2.1 REQUERIMIENTOS DE HARDWARE

Los componentes básicos para el funcionamiento del sistema son:

- Acondicionamiento
- Digitalización
- Transmisión de datos
- Procesamiento y visualización de la señal

En la figura 2.1.1, se muestra, de forma general, el funcionamiento del sistema, en donde se explicará cada uno de sus bloques a detalle en los siguientes puntos.

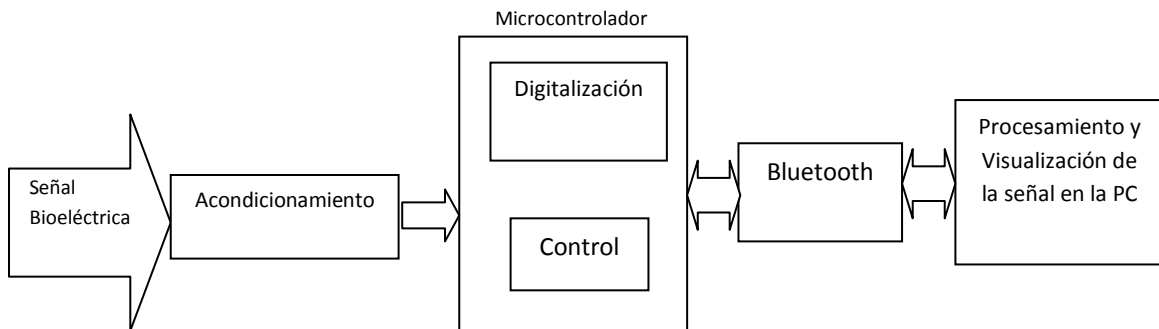


Figura 2.1.1 Diagrama general del sistema.

Después de revisar diferentes artículos y circuitos para la detección de señales cerebrales se optó por uno de los artículos ya mencionados en el capítulo 1 [23], solo que con algunas modificaciones tales como el uso de un integrado para el amplificador de instrumentación, así como el cambio que se hizo para la parte de comunicación, donde, aparte de modificar el programa del PIC, se uso comunicación por medio del módulo Bluetooth.

La etapa uno, la cual es el acondicionamiento, se encargara de amplificar la señal, filtrarla y ajustarla para su digitalización. En la figura 2.1.2 se muestra el diagrama completo que se refiere a esta parte. Está dividida en 3 secciones:

Amplificador de instrumentación, filtro Butterworth y un convertidor de niveles de voltaje de los cuales se tratarán en las siguientes secciones.

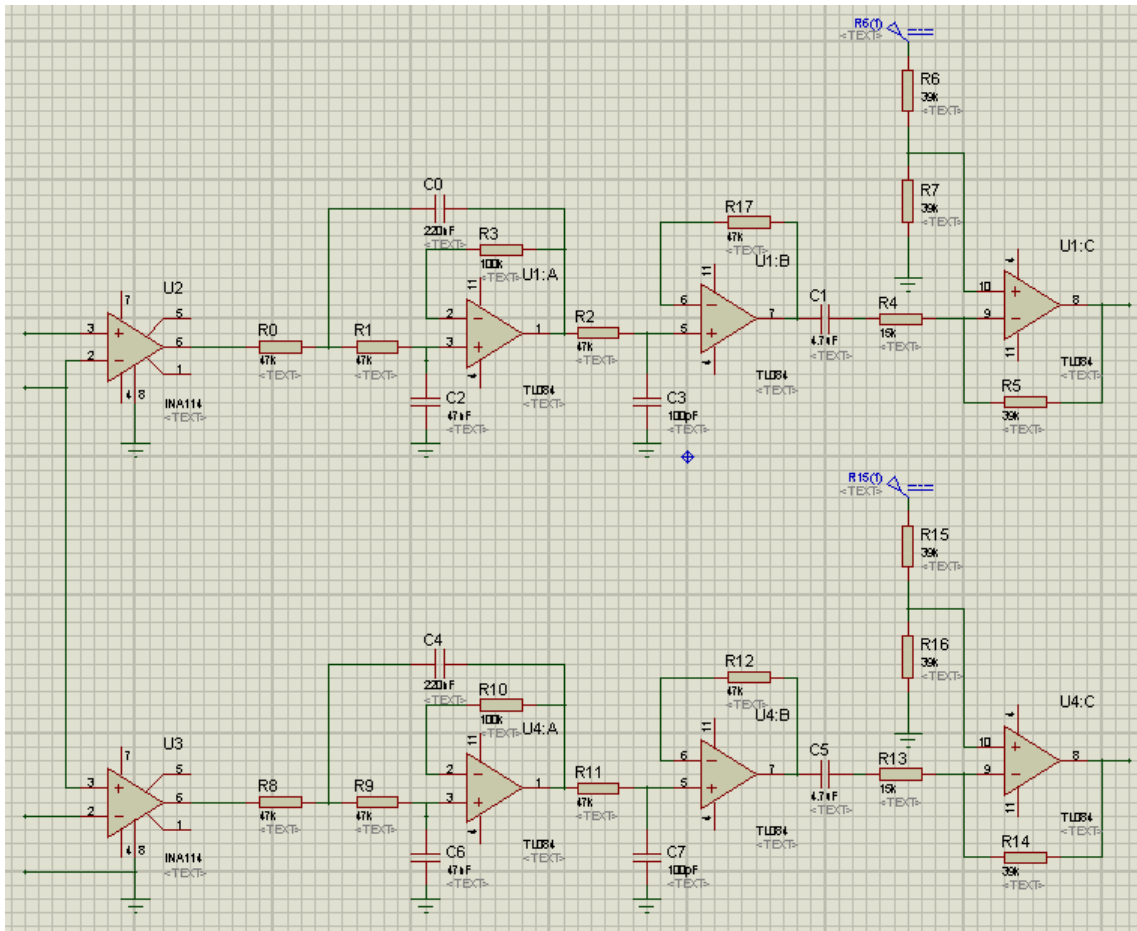


Figura 2.1.2 Etapa de Acoplamiento.

### 2.1.1 AMPLIFICADOR DE INSTRUMENTACIÓN

El amplificador de instrumentación es un amplificador diferencial tensión-tensión cuya ganancia puede establecerse de forma muy precisa y que ha sido optimizado para que opere de acuerdo a su propia especificación aún en un entorno hostil. Es un elemento esencial de los sistemas de medida, en los que se ensambla como un bloque funcional que ofrece características funcionales

propias e independientes de los restantes elementos con los que interacciona. Para ello, se le requiere:

a) Tengan unas características funcionales que sean precisas y estables.

b) Sus características no se modifiquen cuando se ensambla con otros elementos.

A los amplificadores de instrumentación se les requieren las siguientes características:

1) Son amplificadores diferenciales con una ganancia diferencial precisa y estable, generalmente en el rango de 1 a 1000.

2) Su ganancia diferencial se controlada mediante elementos analógicos (potenciómetro resistivo) o digital (conmutadores) lo que facilita su ajuste.

3) Su ganancia en modo común debe ser muy baja respecto de la ganancia diferencial, esto es, debe ofrecer un CMRR muy alto en todo el rango de frecuencia en que opera.

4) Una impedancia muy alta para que su ganancia no se vea afectada por la impedancia de la fuente de entrada.

5) Una impedancia de salida muy baja para que su ganancia no se vea afectada por la carga que se conecta a su salida.

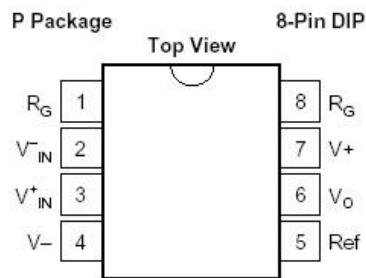
6) Bajo nivel de la tensión de nivel de DC (offset) del amplificador y baja deriva en el tiempo y con la temperatura, a fin de poder trabajar con señales de continua muy pequeñas.

7) Ancho de banda ajustable a las necesidades del diseño

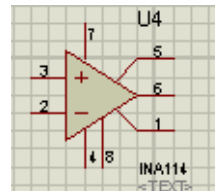
8) Un factor de ruido muy próximo a la unidad, Esto es, que no incremente el ruido.

9) Una razón de rechazo al rizado a la fuente de alimentación muy alto. [24]

El amplificador de instrumentación se encargara de convertir la señal de entrada (Onda Eléctrica Cerebral) de un intervalo de  $\mu\text{V}$  a  $\text{mV}$ . Para conseguir este objetivo, el rango del valor de ganancia debe estar comprendido a partir de 1000 para amplificar la señal y así poder visualizarla y tratarla de acuerdo a las necesidades del estudio.



a) Diagrama General



b) Componente Proteus

Figura 2.1.1.1 INA114

En este caso se usara como amplificador de instrumentación el integrado INA114 de la empresa Burr-Brown, que se puede observar en la imagen 2.1.1.1, ya que a pesar de haber otros amplificadores que presentan menor ruido, como el INA103 o el INA111, este amplificador funciona con un ruido lo suficientemente bajo para él sistema, además de ser comercial. En la imagen 2.1.1.2 obtenida de la hoja de datos se podrá ver su esquema interno.

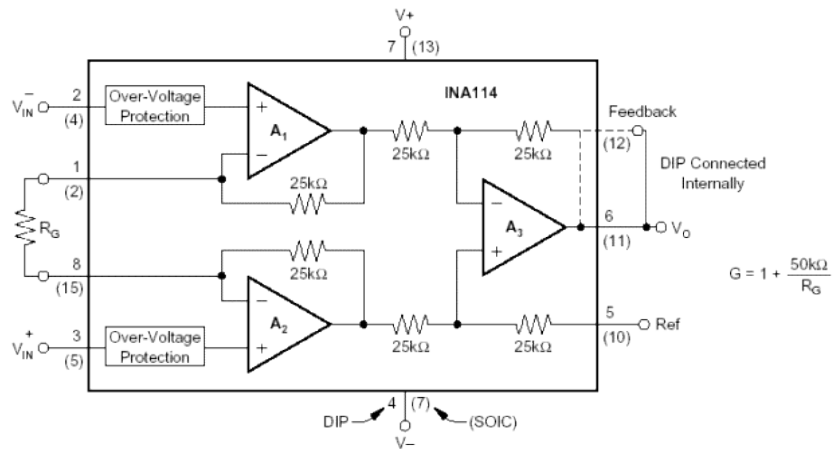


Figura 2.1.1.2 Esquema INA114

Para lograr la ganancia requerida es necesario poner una resistencia en el pin 1 y el 8. En este caso y de acuerdo a la tabla 2.1.1.1 [25], se utilizara una resistencia de 47  $\Omega$ , la cual se escogerá porque es de tipo comercial y es cercana al valor que se muestra en la tabla y además permite obtener la ganancia necesaria para que los datos puedan ser digitalizados correctamente que es calculada en relación de los niveles de las señales del cerebro y la señal que recibe el PIC, usando la relación

$$G = \frac{\text{Señal del cerebro}}{\text{nivel del PIC necesario}} \quad (1)$$

DESIRED GAIN	$R_G$ ( $\Omega$ )	NEAREST 1% $R_G$ ( $\Omega$ )
1	No Connection	No Connection
2	50.00k	49.9k
5	12.50k	12.4k
10	5.556k	5.62k
20	2.632k	2.61k
50	1.02k	1.02k
100	505.1	511
200	251.3	249
500	100.2	100
1000	50.05	49.9
2000	25.01	24.9
5000	10.00	10
10000	5.001	4.99

Tabla 2.1.1.1 Relación de ganancias con su respectiva resistencia

## 2.1.2 FILTRO BUTTERWORTH.

La siguiente parte de la etapa de acondicionamiento es un filtro de tipo Butterworth. El filtro de Butterworth es uno de los filtros electrónicos básicos, diseñado para producir la respuesta más plana que sea posible en la banda de paso hasta la frecuencia de corte. En otras palabras, la salida se mantiene constante casi hasta la frecuencia de corte, luego disminuye a razón de  $20n$  dB por década (ó  $\sim 6n$  dB por octava), donde  $n$  es el número de polos del filtro.

El filtro de Butterworth básico es el típico filtro pasa bajo de primer orden, que puede ser modificado a un filtro pasa alto o añadir en serie otros formando un filtro pasa banda o elimina banda y filtros de mayores órdenes como se muestra en la figura 2.1.2.1

Según lo mencionado antes, la respuesta en frecuencia del filtro es máximamente plana (con las mínimas ondulaciones) en la banda de paso. Visto en un diagrama de Bode con escala logarítmica, la respuesta decae linealmente

desde la frecuencia de corte hacia menos infinito. Para un filtro de primer orden son -20 dB por década (aprox. -6dB por octava).

El filtro de Butterworth es el único filtro que mantiene su respuesta plana para órdenes mayores (sólo con una caída de más pendiente a partir de la frecuencia de corte).

Este tipo de filtros necesita un mayor orden para los mismos requerimientos en comparación con otros, como los de Chebyshev o el elíptico.

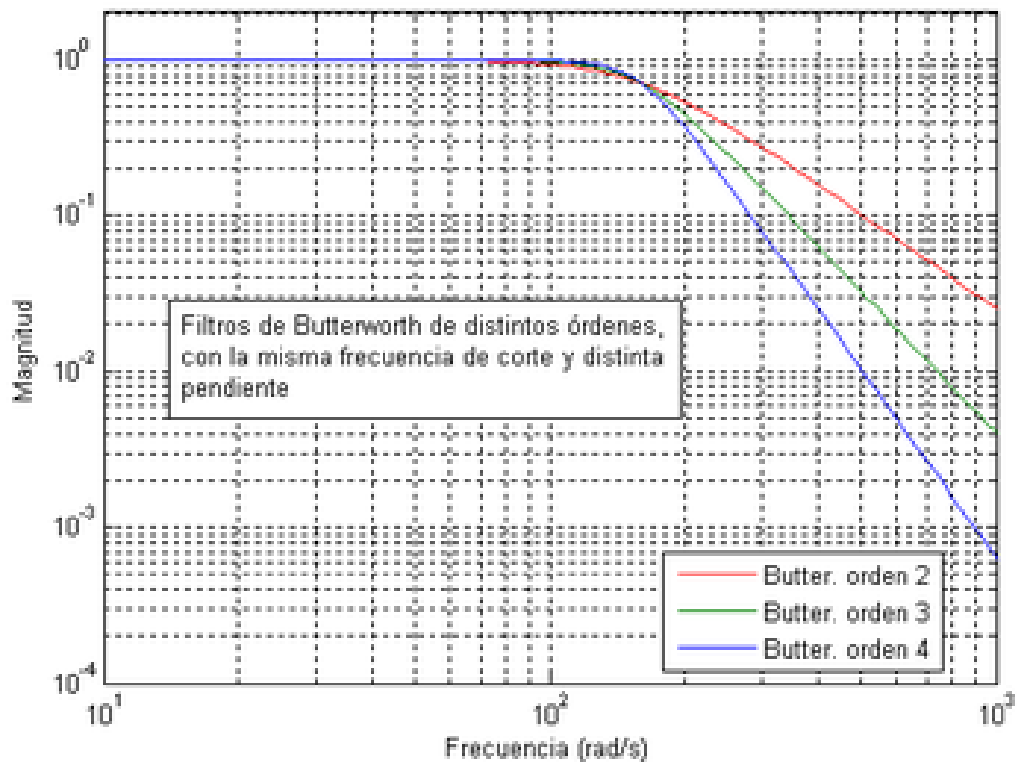


Figura 2.1.2.1 Respuesta de filtros de Butterworth de varios órdenes.

### 2.1.2.1 Diseño

Si llamamos H a la respuesta en frecuencia, se debe cumplir que las 2N-1 primeras derivadas de  $|H(\Omega)|^2$  sean cero para  $\Omega = 0$  y  $\Omega = \infty$ . Únicamente posee polos y la función de transferencia es la fórmula 1

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}} \quad (2)$$

donde N es el orden del filtro,  $\Omega_c$  es la frecuencia de corte (en la que la respuesta cae 3 dB por debajo de la banda pasante) y  $\Omega$  es la frecuencia analógica compleja ( $\Omega = j\omega$ ) [26].

Se escogió uno de 3er orden ya que así permitirá obtener una atenuación de -60db, línea verde en la gráfica de la figura 2.1.2.1.

1.- Se define la  $\omega_c$  o la frecuencia de corte; en este caso son 30 Hz.

2.- Se define el valor de  $C_3$  dentro del rango de 0.001 y 0.1 $\mu$ F.

Si

$$C_3 = 0.1 \mu F \quad (3)$$

3.- Definimos el valor de

$$C_2 = \frac{1}{2} C_3 \quad (4)$$

y

$$C_1 = 2C_3 \quad (5)$$

Se definen valores en (4)

$$C_2 = \frac{1}{2} * 0.1 \mu F = 0.05 \mu F = 5 \text{ nF} \quad (6)$$

Se utiliza valor estándar de 47 nF

Se definen valores en (5)

$$C_1 = 2 * 0.1 \mu F = 0.2 \mu F \quad (7)$$

Se escoge un valor estándar de 220 nF

4.- Se calcula el valor R

$$R = \frac{1}{\omega_c * C_3} \quad (8)$$

en la fórmula

$$R = \frac{1}{2\pi * f_c * C_3} = \frac{1}{2\pi * 30 \text{ Hz} * 0.1 * 10^{-6} \text{ F}} = 53.05 \text{ k}\Omega \quad (9)$$

Se escoge el valor estándar de 47 k $\Omega$  y recalculando la frecuencia de corte se tiene:

$$F_c = \frac{1}{2\pi * 47 * 10^3 \Omega * 0.1 * 10^{-6}} = 33.86 \text{ Hz} \quad (10)$$

5.- Se define

$$R_1 = R_2 = R_3 = R = 47 \text{ k}\Omega \quad (11)$$

6.- Se calcula

$$R_{f1} = 2R \quad (12)$$

y

$$R_{f2} = R \quad (13)$$

R debe estar dentro del rango de 10 y 100 KΩ; en el caso de que R esté fuera del rango, se debe escoger C<sub>3</sub> nuevamente .

$$R_{f1} = 2 \cdot 47 \text{ K}\Omega = 94 \text{ K}\Omega \quad (14)$$

estandarizado

$$R_{f1} = R1 = 100 \text{ K}\Omega \quad (15)$$

$$R_{f2} = R0 = R1 = R3 = R4 = 47 \text{ K}\Omega \quad (16)$$

En la figura 2.1.2.2.1 se muestra el la configuración de este filtro ya con valores incluidos de los componentes.

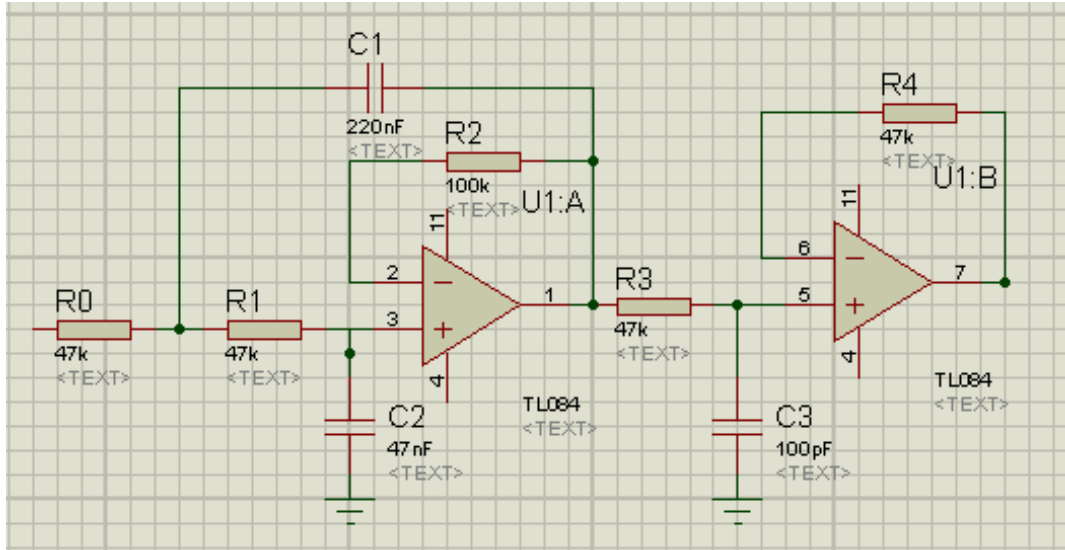


Figura 2.1.2.1.1 Configuración del filtro pasa bajo Butterworth con los valores de diseño

### 2.1.3 CONVERSIÓN DE NIVELES DE VOLTAJE (OFFSET)

La última etapa de los canales analógicos cumple la función de convertir las señales entre +12 V y -12 V a unas entre 0 y 5 V para que el convertidor A/D del microcontrolador pueda digitalizarlas, es así que a la señal de electroencefalografía de cada canal se añade un nivel de DC (offset) para desplazarla a un rango positivo entre 0 y 5 V, mediante un sumador no inversor implementado por medio del TL084 mencionado anteriormente.

Los problemas que se encuentra al diseñar esta etapa son principalmente los de saturación. La distorsión que introduce la deformación de la señal puede dar lugar a errores en la representación posterior, y lo que es más importante, la aparición de otras componentes de frecuencia que se vería reflejado en el análisis con la transformada de Fourier, lo que daría un electroencefalograma falseado. Para tratar de evitar lo anterior, es que se debe trabajar en el intervalo de +12 V y -12 V.

El problema de esta alimentación es: que si por mal funcionamiento de la parte de los canales analógicos el amplificador operacional se satura, la salida podría rondar los 12 V positivos o negativos, si sucede aquello con mucha seguridad esta tensión dañaría al microcontrolador que realiza la conversión A/D. Por ello es que el circuito que se utiliza es un no inversor con un divisor de tensión en el terminal positivo, con el que se fija a el nivel de DC como se muestra en la Figura 2.1.3.1.

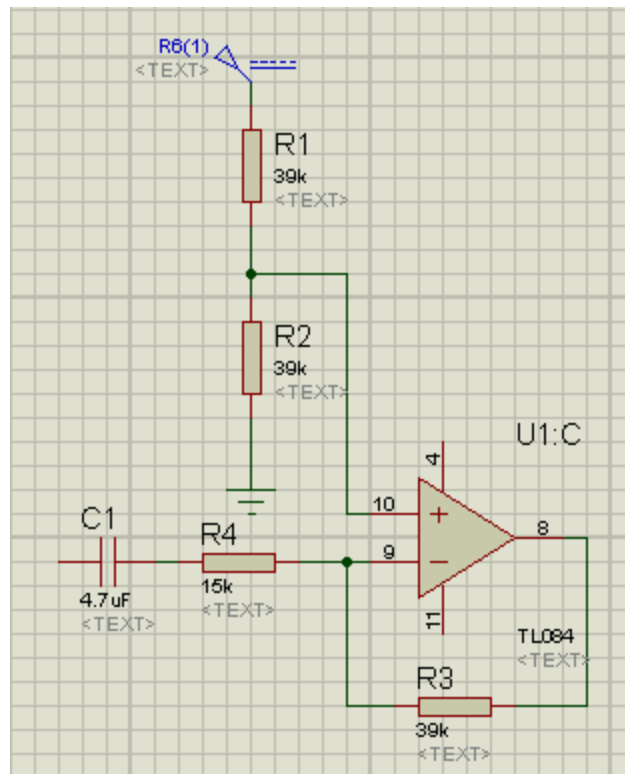


Figura 2.1.3.1 Acondicionador de señal con valores

Como la señal de electroencefalografía tiene una amplitud máxima de aproximadamente 500 mV, el diseño de la etapa con una ganancia de 2.5, y un nivel de DC de 2.5 V. El valor máximo de amplitud a la salida para una señal en la entrada de 500 mV es:

$$2.5 \times 0.5 \text{ V} + 2.5 \text{ V} = 3.75 \text{ V} \quad (17)$$

Con el capacitor  $C_1$  aislamos el nivel de DC de esta etapa inversora, de la anterior, y además se comporta como filtro paso alta.

Si llamamos  $Z$  a la impedancia formada por la asociación serie de  $C_1$  y  $R_4$  de la figura 2.1.3.1, obtenemos que la señal de partida depende de la formula 18.

$$V_0 = -\frac{R_3}{Z} * V_i + \left(\frac{R_3}{Z} + 1\right) \cdot V_{offset} \quad (18)$$

Como  $Z$  se hace infinito para una tensión DC, la fórmula 15 la podemos expresar como

$$V_0 = V_i + V_{offset} \quad (19)$$

El valor de nivel de DC se puede calcular con la expresión 17

$$V_{offset} = \frac{V_{cc}}{1 + \frac{R_1}{R_2}} = \frac{5V}{1 + \frac{39k\Omega}{39k\Omega}} = \frac{5V}{2} = 2.5V \quad (20)$$

Con los valores de las resistencias  $R_1$  y  $R_2$ , y sabiendo que  $V_{cc}$  vale 5 V, obtenemos un offset de 2.5 V.

La función de transferencia para la señal AC de esta etapa es

$$H = -\frac{R_3}{R_4} * \frac{S}{S + \frac{1}{R_4 * C_1}} \quad (21)$$

La frecuencia de corte de esta etapa pasa-alta es 19.

$$F_{corte} = \frac{1}{2\pi * R_4 * C_1} = \frac{1}{2\pi * 4.7 * 10^{-6} F * 15 * 10^3} = 2.25 Hz \quad (22)$$

La ganancia es

$$|H_{MAX}| = \frac{R_3}{R_4} = \frac{39k\Omega}{15k\Omega} = 2.6 \quad (23)$$

Con todos estos datos podemos dar un valor aproximado de la ganancia diferencial. El resultado es 21.

$$|G_d| = 2.6 * 7642.71 = 19871.05 \quad (24)$$

#### 2.1.4 DIGITALIZACIÓN

La conversión análogo digital de las señales del electroencefalograma (EEG) se realiza mediante el convertidor analógico-digital del microcontrolador.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

En la figura 2.1.4.1 se puede observar una representación en bloques del microcontrolador, en la cual se ven similitudes con un ordenador tales como su fuente de alimentación, un circuito de reloj y el chip microcontrolador. Dispone de su CPU, sus memorias, y por supuesto, sus puertos de comunicación listos para conectarse con dispositivos externos.

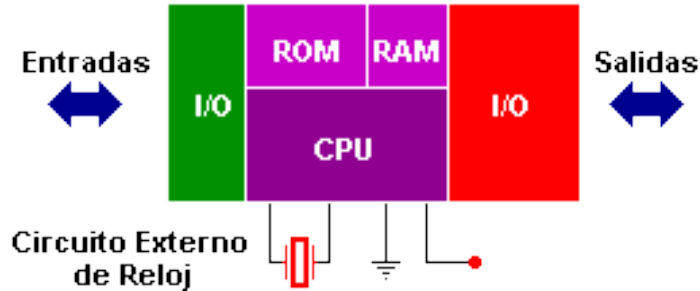


Figura 2.1.4.1 Diagrama general interno de un microcontrolador

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM/FLASH.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, puertos serie y paralelo, CAD: Convertidores Analógico/Digital, CDA: Convertidores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.

- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

El Programa se encargara de verificar si se reciben datos por medio del puerto serial, de ser así el programa continua. Primero lee los datos recibidos en los puertos analogicos del microcontrolador. Luego se guarda en un arreglo para que cuando se llene, se envíe al puerto serial y se repite el proceso. En la figura 2.1.4.2 se mostrara el diagrama de flujo que se usara para programar el PIC.

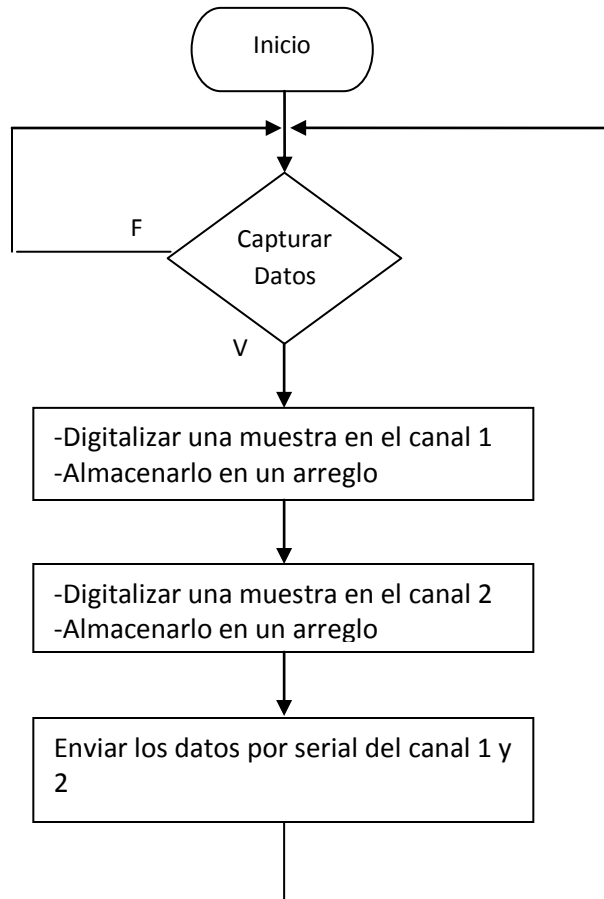


Figura 2.1.4.2 Diagrama de flujo para el control del PIC

La entrada ADC será utilizada para convertir la señal analógica resultante de nuestra etapa de acondicionamiento en una señal digital que pueda ser leída por nuestro modulo Bluetooth.

El programa del PIC se hará en un lenguaje de alto nivel como C. Hay varios compiladores tales como el C18 de Microchip, CCS de CCS Inc., MikroC, MikroC PRO de Mikroelektronika. Se uso en este caso el MikroC PRO versión

4.60, ya que este compilador tiene incluida una documentación bastante completa, la cual es de ayuda al momento de poder escribir el código.

El microcontrolador que se utilizó fue el PIC18F4550, ya que tiene un módulo de comunicación serial, que es compatible con el módulo bluetooth.

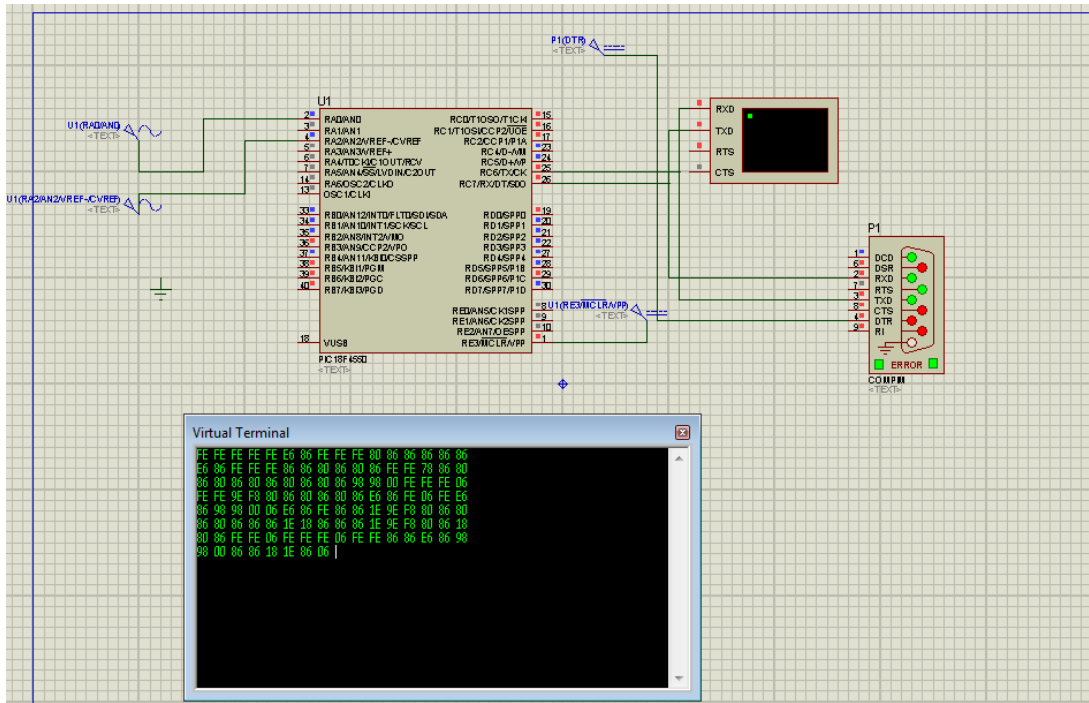


Figura 2.1.4.3 Simulación en Proteus

En la figura 2.1.4.3 se muestra el sistema con el microcontrolador usando el programa ISIS de Proteus. Como se podrá observar, se usarán como entradas cualquiera de los pines analógicos 2 y 4. Se utilizan los pines 25 y 26, ya que es en estos pines donde se ubica el puerto serial que se conecta de forma directa al módulo bluetooth. Para hacer la simulación, se usó el puerto serial dado en Proteus, ya que el módulo Bluetooth usa el mismo protocolo de comunicación que el puerto serial. Usando la terminal y probando con señales sinusoidales se puede notar la transmisión de datos en formato serial.

### 2.1.5 TRANSMISIÓN DE DATOS

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja, y basados en transceptores de bajo costo.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una clase con los de las otras. En la tabla 2.1.5.1 se muestran las diferentes clases con su respectivas potencias y distancias.

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

Tabla 2.1.5.1 Clases de Bluetooth

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal

llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor sensibilidad del dispositivo de clase 1 permite recibir la señal del otro pese a ser más débil [6].

Los dispositivos con Bluetooth también pueden clasificarse según su ancho de banda tal como se muestra en la Tabla 2.1.5.2

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s

Tabla 2.1.5.2 Ancho de Banda de Bluetooth

La especificación de Bluetooth define un canal de comunicación de máximo 720 kb/s (1 Mbps de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores) [27].

La frecuencia de radio con la que trabaja está en el rango de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos/s. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez [27].

La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0 dBm (1 mW), mientras que la versión de largo alcance transmite entre 20 y 30 dBm (entre 100 mW y 1 W) [27].

Para lograr alcanzar el objetivo de bajo consumo y bajo costo, se ideó una solución que se puede implementar en un solo chip utilizando circuitos CMOS. De esta manera, se logró crear una solución de 9x9 mm y que consume aproximadamente 97% menos energía que un teléfono celular común [28].

El protocolo de banda base (canales simples por línea) combina conmutación de circuitos y paquetes. Para asegurar que los paquetes no lleguen fuera de orden, los slots pueden ser reservados por paquetes síncronos, un salto diferente de señal es usado para cada paquete. Por otro lado, la conmutación de circuitos puede ser asíncrona o síncrona. Tres canales de datos síncronos (voz), o un canal de datos síncrono y uno asíncrono, pueden ser soportados en un solo canal. Cada canal de voz puede soportar una tasa de transferencia de 64 kb/s en cada sentido, la cual es suficientemente adecuada para la transmisión de voz. Un canal asíncrono puede transmitir como mucho 721 kb/s en una dirección y 56 kb/s en la dirección opuesta, sin embargo, para una conexión síncrona es posible soportar 432,6 kb/s en ambas direcciones si el enlace es simétrico [28].

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes:

- **un dispositivo de radio**, encargado de modular y transmitir la señal
- **un controlador digital**, compuesto por una CPU, por un procesador de señales digitales (DSP - Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de las interfaces con el dispositivo anfitrión.

El LC o Link Controller está encargado de hacer el procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de capa física. Además, se encarga de las funciones de transferencia (tanto asíncrona como síncrona), codificación de Audio y cifrado de datos.

El CPU del dispositivo se encarga de atender las instrucciones relacionadas con Bluetooth del dispositivo anfitrión, para así simplificar su operación. Para ello, sobre el CPU corre un software denominado Link Manager que tiene la función de comunicarse con otros dispositivos por medio del protocolo LMP.

Entre las tareas realizadas por el LC y el Link Manager, destacan las siguientes:

- Envío y Recepción de Datos.
- Empaginamiento y Peticiones.
- Determinación de Conexiones.
- Autenticación.
- Negociación y determinación de tipos de enlace.
- Determinación del tipo de cuerpo de cada paquete.
- Ubicación del dispositivo en modo sniff o hold.

#### 2.1.5.1 Enlace físico.

En la especificación Bluetooth se definen dos tipos de enlace:

- Enlace de sincronización de conexión orientada (SCO)
- Conexión simétrica punto a punto entre maestro y esclavo.
- El maestro utiliza rebanadas de tiempo reservadas a intervalos regulares.

- El maestro puede soportar más de 3 enlaces simultáneos mientras que los esclavos 3 como máximo.
- Los paquetes de configuración nunca son retransmitidos.
- Principalmente este tipo de enlace se utiliza para transmitir información de voz con un ratio de transmisión de 64kB/s.
- Enlace asíncrono de baja conexión (ACL)
- Conexiones simétricas o asimétricas punto-multipunto entre maestro y esclavo.
- Conexión utilizada para la transmisión de datos.
- Se aplica retransmisión de paquetes.
- Se definen para este tipo de conexión los slots 1,3 y 5.
- La máxima velocidad de envío es de 721 kbit/s en una dirección y 57.6 kbit/s en la otra.

#### 2.1.5.2 Arquitectura de protocolo.

La especificación Bluetooth define una arquitectura de protocolos semejante a la torre OSI. Estos protocolos son utilizados dependiendo el tipo de sistema que se quiere implementar. Cada protocolo lo podemos tratar como una capa independiente con sus correspondientes interfaces para que el sistema sea homogéneo.

En la figura 1.3.2.3.3.1 se puede ver una imagen donde se muestra la pila de protocolos [29]. Como se podrá observar, la pila completa se compone tanto de protocolos específicos de Bluetooth (LM (Link Manager) y L2CAP (Logical Link Control Adaption Protocol), por ejemplo, como de protocolos no específicos de Bluetooth como son OBEX (Objects Exchange Protocol), UDP (User Datagram Protocol), TCP, IP, etc., eso con el propósito de que la hora de diseñar la torre de protocolos, el objetivo principal ha sido maximizar el número de protocolos existentes que se puedan reutilizar en las capas más altas para diferentes propósitos.

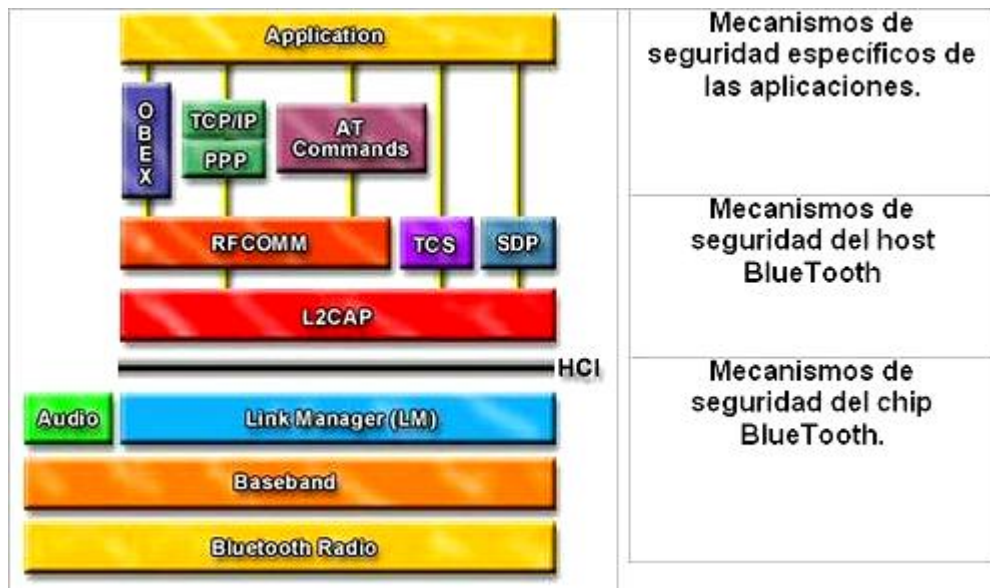


Figura 1.3.2.3.3.1 Pila de protocolos del Bluetooth

En este apartado no se revisara la pila de protocolos ya que solo se utilizara el protocolo RFCOMM para comunicarse con el puerto serial del microcontrolador, pero es necesario nombrarlo para estar al tanto que hay más usos que se le pueden dar a este tipo de comunicación y saber que su uso no está limitado solamente a ello.

### 2.1.5.3 Modulo Bluetooth

El modulo Bluetooth utilizado en este proyecto usa chipsets externos BlueCore4. Esto incrusta 8 Mbit de flash para almacenamiento de software, y suporta 3.3 V de voltaje. BC04 es un módulo de multifunción. Puede ser usado en diferentes productos de acuerdo a las características del firmware. También está especializado para la transferencia de datos.

La segunda generación de modulo Bluetooth UART tiene 2 modos de trabajo: modo de comando AT, y modo automático de transferencia de datos de forma transparente. En modo transparente de datos, puede ser configurado a maestro, esclavo o bucle, y conectarse o ser conectado por otros dispositivos que soporten protocolo SPP para configuración. En modo comandos AT, el usuario puede configurar el modulo y enviar comandos de control.

Sus componentes principales son:

- Chipset CSR BC417143 (BlueCore4External)
- Bluetooth version V2.0+EDR
- Output power Class II
- Flash 8 Mbit
- Power Supply 3.3 V
- Interface I2C UART PCM USB1.2

- Size 26.9 mm\*13 mm\*2.2 mm
- ROHS: Yes

Para procesar los datos y visualizarlos en la PC, se envían los datos de forma inalámbrica utilizando el protocolo RFCOMM por medio del modulo Bluetooth, ya que nos permite movilidad, aparte de su bajo costo y estandarización.

En este caso se usara un modulo Bluetooth HC-05 diseñado para una conexión transparente [30].

Sus características vendrían siendo:

- Sensibilidad típica de -80 dB
- Poder de transmisión de radiofrecuencia arriba de +4 dBm
- Interfaz UART con velocidad de transmisión programable desde 4800 hasta 1382400 baudios, entre otras características.

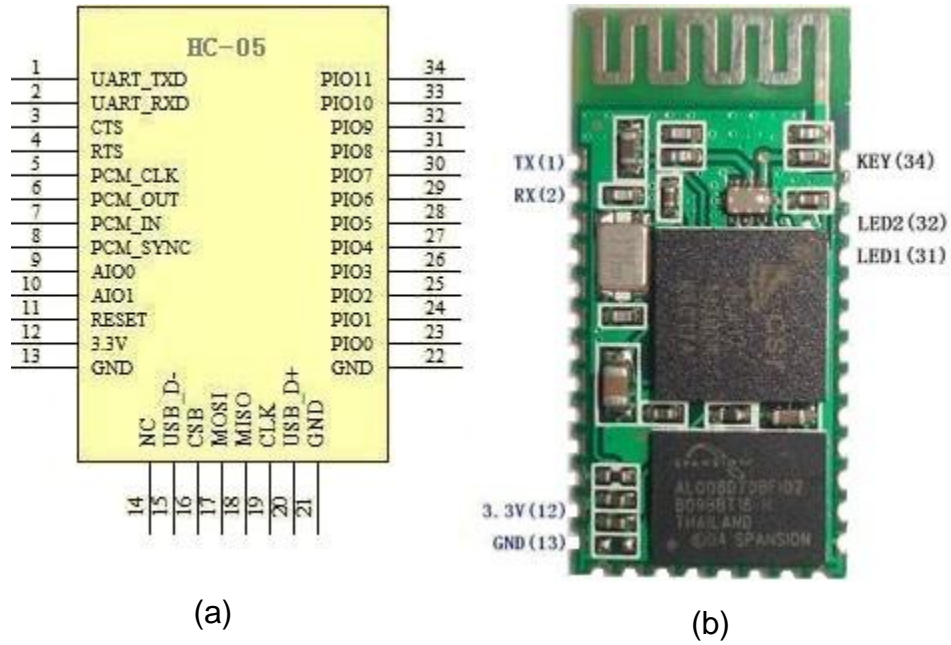


Figura 2.1.5.1 Diagrama de pines (a) y módulo Bluetooth HC-05 (b)

El diagrama del módulo en general se presenta en la figura 2.1.5.1, ya en la tabla 2.1.5.1 se define la función de cada pin [31]. En esta tabla se pueden observar las diferentes funciones de cada pin siendo la mayor parte pines programables, que se pueden usar por medio de comandos AT. Pero los que vamos a utilizar serán pin 1 y 2 para envío/recepción de datos y los pines que corresponden tanto a fuente de alimentación como a la tierra (GND).

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación

PIN Name	PIN #	Pad type	Description
GND	13	VSS	Ground Pot
	21		
	22		
3.3 Vcc	12	3.3 V	Integrated 3.3 V (+) supply with On-Chip linear regulator Output within 3.15-3.3 V
AIO0	9	Bi-Directional	Programmable input/output line
AIO1	10	Bi-Directional	Programmable input/output line
PIO0	23	Bi-Directional RX EN	Programmable input/output line control output for LNA (if fitted)
PIO1	24	Bi-Directional TX EN	Programmable input/output line control output for PA (if fitted)
PIO2	25	Bi-Directional	Programmable input/output line
PIO3	26	Bi-Directional	Programmable input/output line
PIO4	27	Bi-Directional	Programmable input/output line
PIO5	28	Bi-Directional	Programmable input/output line
PIO6	29	Bi-Directional	Programmable input/output line
PIO7	30	Bi-Directional	Programmable input/output line
PIO8	31	Bi-Directional	Programmable input/output line
PIO9	32	Bi-Directional	Programmable input/output line
PIO10	33	Bi-Directional	Programmable input/output line
PIO11	34	Bi-Directional	Programmable input/output line
RESETB	11	CMOS input with weak internal pull-up	Reset if low, input debouncde so must be low for >5MS to cause a reset
UART_RTS	4	CMOS output tri-stable with weak internal pull-up	UART request to send, active low
UART_CTS	3	CMOS input with weak internal pull-up	UART clear to send, active low
UART_RX	2	CMOS input with weak internal pull-up	UART Data input
UART_TX	1	CMOS output tri-stable with weak internal pull-up	UART Data output
SPI_MOSI	17	CMOS input with weak internal pull-up	Serial Peripheral interface data input
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface data input
SPI_CLK	19	CMOS input with weak internal pull-up	Serial Peripheral interface clock
SPI_MISO	18	CMOS input with weak internal pull-up	Serial Peripheral interface data output
USB_-	15	Bi-Directional	
USB_+	20	Bi-Directional	
NC	14		
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock
PCM_OUT	6	CMOS output	Synchronous PCM data output
PCM_IN	7	CMOS input	Synchronous PCM data input
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe

Tabla 2.1.5.1 Pines del Módulo HC-05

En la figura 2.1.5.2 se podrá ver la configuración básica del módulo Bluetooth.

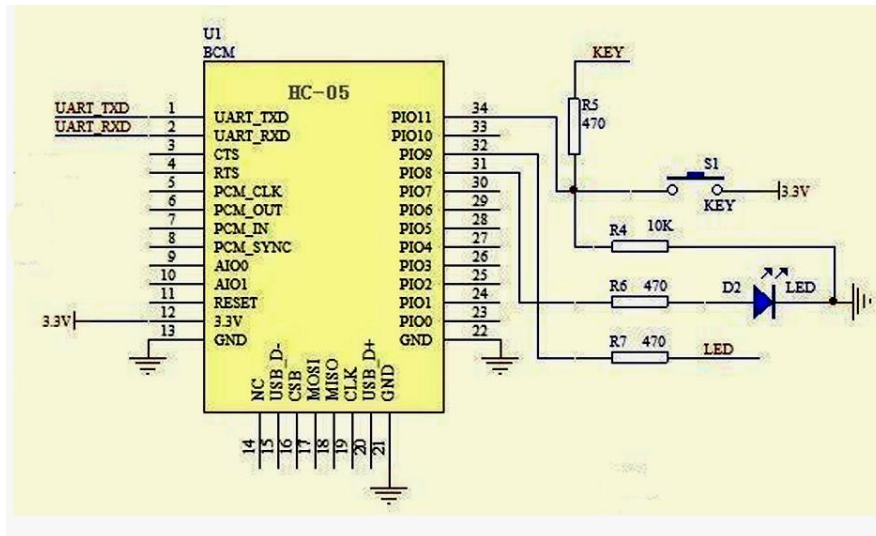


Figura 2.1.5.2 Configuración básica para el módulo Bluetooth

El diseño final del circuito para la obtención y envío-recepción de los datos quedaría como se muestra en la figura 2.1.5.3. Solo cabe aclarar que la salida mostrada en el diseño va conectada directamente a lo que sería el módulo Bluetooth, que no aparece ya que no hay un componente que lo represente en el Proteus. En la figura 2.1.5.3 se muestran el diseño de todo el sistema requerido para la adquisición de los datos.

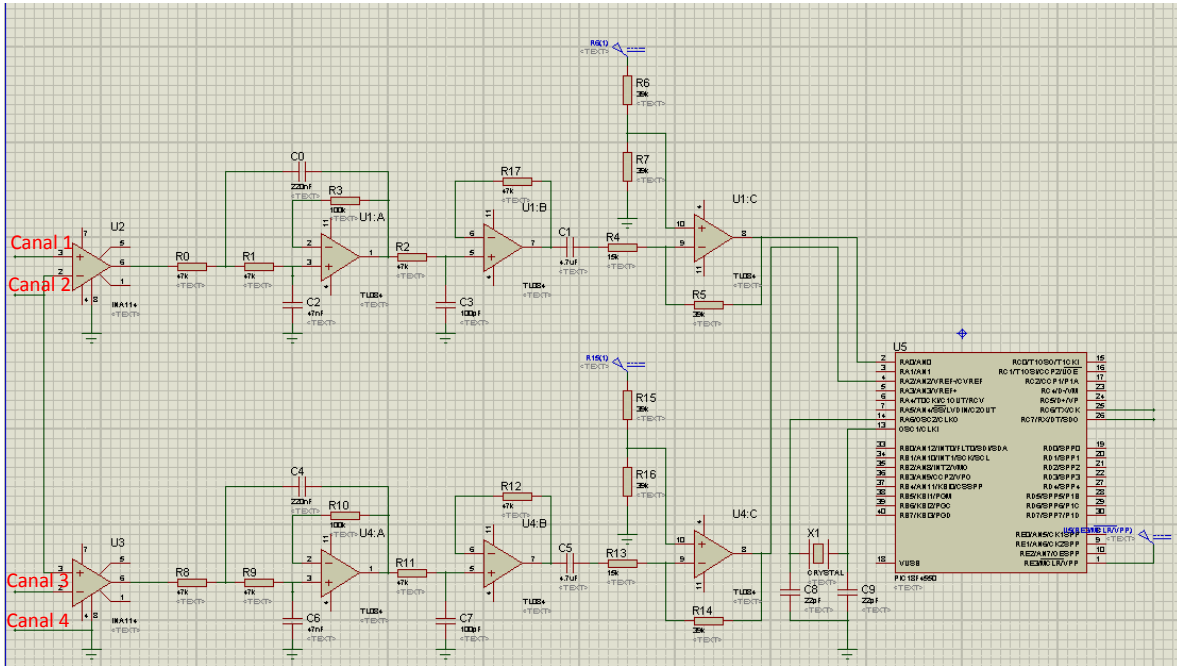


Figura 2.1.5.3 Diseño Final del sistema

Se puede observar en la figura 2.1.5.3 las diferentes etapas del diseño, empezando desde el amplificador de instrumentación (INA114), para enviar la información por el filtro Butterworth, luego por el filtro pasabajas para después enviarlo al microcontrolador. Se usó el amplificador INA114 y uno de cada uno de los filtros por canal, también se puede observar cómo se conectan las entradas que irán hacia los INA114. Estas entradas es donde irán los electrodos que van hacia el cerebro, en este caso 4 canales. En cuanto a la salida del microcontrolador esta irá hacia el módulo Bluetooth.

# Capítulo 3. Desarrollo de Software

## Introducción

En este capítulo se tratará el tema del Software. Este por el momento se encargará de poder recibir la señal enviada por el módulo descrito en la sección de hardware para luego poder visualizarla por medio de la computadora. Se explicará a detalle cómo se hace la recepción de datos y su posterior visualización, además de permitir un análisis en frecuencia, para mostrar un análisis más preciso.

Uno de los lenguajes más utilizados y difundidos en estos últimos años ha sido Java, por ello el sistema se hizo utilizando una paquetería que permite compilar el lenguaje, en este caso el JCreator.

Una de las razones por la que se seleccionó este lenguaje es debido a su gran portabilidad, ya que puede ser utilizado en cualquier plataforma, es decir se puede hacer el código en Windows y compilarlo en Linux sin necesidad de cambiar ninguna parte de código o ser ejecutado forzosamente en alguna plataforma. Esta portabilidad también se ha extendido hacia los dispositivos móviles, lo cual puede permitir al sistema poder ser utilizado en uno de ellos.

Otra razón por la que se escogió este sistema fue por la gran variedad de APIS que contiene, la cual permite desarrollar distintas aplicaciones ya sean sockets, archivos, crear ventanas, etc. En este caso se utilizara el Bluecove (descargable en su página oficial), el cual permitirá a la computadora acceder al módulo Bluetooth de la computadora y poder leer los datos enviados por el microcontrolador.

### 3.1 REQUERIMIENTOS DEL SOFTWARE

Los requerimientos básicos para el funcionamiento del sistema es el siguiente:

- Sistema operativo WindowsXP/Vista/7, Linux, o cualquier sistema que tenga la máquina virtual java.
- JCreator para poder compilar el código fuente.
- Librería Bluecove, el cual permite utilizar el API JSR-82 para el manejo del módulo Bluetooth en la computadora personal.
- Modulo Bluetooth (interno o externo)

### 3.2 MÓDULO BLUETOOTH

El Bluetooth es una tecnología de comunicación inalámbrica omnidireccional. Se ideó pensando en dispositivos de bajo consumo y comunicaciones a corta distancia (10 metros).

Se trata de una tecnología barata con un ancho de banda reducido. Es ideal para periféricos de computadora (ratón, teclado, manos libres,...) y dispositivos móviles (teléfonos móviles, PDAs, Pocket PCs, etc.).

Mediante Bluetooth es posible formar pequeñas redes de dispositivos conectados denominadas piconets. Se pueden conectar varias piconets formando lo que se denomina una scatternet.

Las principales aplicaciones de Bluetooth son: transferencia de archivos, sincronización de dispositivos y conectividad de periféricos.

El lenguaje de programación Java otorga una API llamada JSR-82, la cual permite la comunicación con el módulo Bluetooth.

Este API está dividida en dos partes: el paquete `javax.bluetooth` y el paquete `javax.obex`.

Los dos paquetes son totalmente independientes. El primero de ellos define clases e interfaces básicas para el descubrimiento de dispositivos, descubrimiento de servicios, conexión y comunicación. La comunicación a través de `javax.bluetooth` es a bajo nivel: mediante flujos de datos o mediante la transmisión de arrays de bytes.

Por el contrario el paquete `javax.obex` permite manejar el protocolo de alto nivel OBEX (OBject EXchange). Este protocolo es muy similar a HTTP y es utilizado sobre todo para el intercambio de archivos. El protocolo OBEX es un estándar desarrollado por IrDA y es utilizado también sobre otras tecnologías inalámbricas distintas de Bluetooth [32].

La librería Bluecove usa la API JSR-82, para utilizarla en una computadora personal y también en dispositivos móviles (incluido en el J2ME) con esto se tiene acceso al módulo Bluetooth, ya sea interno o externo.

Bluecove puede ser descargada de su página oficial <http://bluecove.org>, donde la cual permite obtener un archivo `.jar` el cual va incluido en el CLASSPATH de java, permitiendo así poder empezar a trabajar con la comunicación. En la figura 3.2.1 se muestra el diagrama de flujo general del programa, donde el programa se encargara de localizar los dispositivos y sus servicios, para luego poder conectarse y empezar la comunicación. Después se empiezan a leer los

datos, para esto primero se crean 2 paneles, que cada uno será un canal para la representación de datos, luego se guardan en un arreglo para evitar pérdida de datos y luego se graficarán. Por último se guardan los datos en 2 archivos, cada uno siendo un canal.

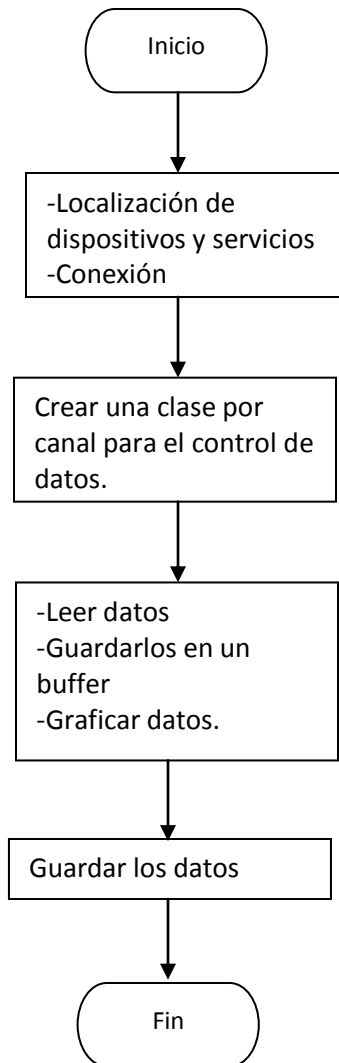


Figura 3.2 Diagrama de flujo general para el sistema completo.

### 3.2.1 CLASE PARA CONTROL DEL MÓDULO BLUETOOTH.

Aquí se explica la parte que se encargara de la búsqueda de dispositivos, localización de servicios y lectura de los datos enviados por medio del módulo Bluetooth. En la figura 3.2.1.1 se muestra cómo funcionaría de forma básica este apartado que localizara los dispositivos que estén cercanos a la computadora, luego se descubrirán los servicios que tiene cada dispositivo para luego poder conectarse a alguno de ellos, en este caso el electroencefalógrafo. Por último se empieza la lectura de datos.

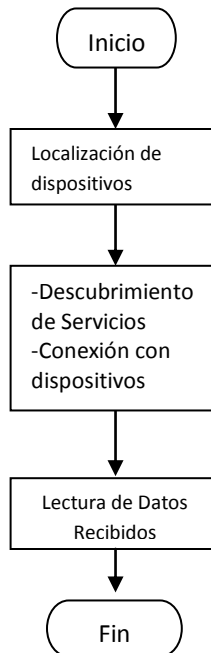


Figura 3.2.1.1 Diagrama de flujo general para el funcionamiento de la recepción y captura de datos.

#### 3.2.1.1 Clase encargada para la localización de dispositivos.

El proceso de detección de dispositivos Bluetooth forma parte de las funciones de la capa HCI.

- Inicialmente, el dispositivo origen envía paquetes inquiry y se mantiene en espera de recibir respuestas de otros dispositivos presentes en su zona de cobertura.
- Si los dispositivos destino están configurados en modo visible (discoverable) se encontrarán en estado inquiry\_scan y en predisposición de atender estos paquetes. En este caso, al recibir un paquete inquiry cambiarán a estado inquiry\_response y enviarán una respuesta al dispositivo origen con sus direcciones MAC y otros parámetros.
- Los dispositivos que estén configurados en modo no visible (no discoverable) nunca entrarán en modo inquiry\_response y, por tanto, no responderán al dispositivo origen y permanecerán ocultos. [33]

Esta clase que viene en los ejemplos dados en la página de Bluecove nos permitirá poder descubrir los dispositivos que estén a nuestro alrededor. La clase usa los métodos de la clase LocalDevice. La clase LocalDevice provee el método “getDiscoveryAgent” que retorna una instancia DiscoveryAgent. Este DiscoveryAgent puede ser usado para descubrir los dispositivos Bluetooth remotos (usando búsqueda HCI)

Como se podrá ver a continuación en la figura 3.2.1.1.1, la clase RemoteDevice representa un dispositivo Bluetooth remoto. Por lo tanto provee información acerca de un dispositivo remoto como la dirección Bluetooth del dispositivo y su alias (friendly name) [34].

```
DiscoveryListener listener = new DiscoveryListener() {
    public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
        System.out.println("Dispositivo " + btDevice.getBluetoothAddress() + " encontrado");
        devicesDiscovered.addElement(btDevice);
        try {
            System.out.println(" Nombre " + btDevice.getFriendlyName(false));
        } catch (IOException cantGetDeviceName) {
        }
    }

    public void inquiryCompleted(int discType) {
        System.out.println("Búsqueda de dispositivo completa");
        synchronized(inquiryCompletedEvent){
            inquiryCompletedEvent.notifyAll();
        }
    }
    .
    .
    .
};

synchronized(inquiryCompletedEvent) {
    boolean started =
LocalDevice.getLocalDevice().getDiscoveryAgent().startInquiry(DiscoveryAgent.GIAC,
listener);
    if (started) {
        System.out.println("Espere hasta que la búsqueda se complete...");
        inquiryCompletedEvent.wait();
        System.out.println(devicesDiscovered.size() + " Dispositivo(s) encontrado");
    }
}
```

Figura 3.2.1.1.1 Código para la localización de dispositivos.

Como se puede notar en la figura 3.2.1.1.1, este código es el `DiscoveryListener`, la cual es una interfaz que permite a una aplicación recibir los eventos de dispositivos descubiertos y descubrimiento de servicios. De los cuatro métodos que proporciona esta interfaz (`deviceDiscovered`, `inquiryCompleted`, `servicesDiscovered`, `serviceSearchCompleted`), se usan los métodos `deviceDiscovered` e `inquiryCompleted`. El primero es llamado cuando un dispositivo es encontrado durante una búsqueda. Una búsqueda para servicios que son descubribles, mientras que el segundo método que es llamado cuando la búsqueda esta completa.

Entre los métodos incluidos en la clase `RemoteDevice`, cabe destacar dos los cuales son:

`getBluetoothAddress`: Este método recupera la función del o los dispositivos Bluetooth. La dirección consta de 12 caracteres que van del 0 al 9 y de A a F (sistema hexadecimal). Esta función no retorna nunca null (valor que indica ausencia de datos) [34].

`getFriendlyName` : Devuelve el nombre del dispositivo. La especificación del Bluetooth llama a esto nombre del dispositivo Bluetooth o “user-friendly name”. Este método solo se comunicara con el dispositivo remoto si el nombre no es conocido, o si su parámetro de entrada es verdadero. El parámetro de entrada es un booleano, el cual si es verdadero retornara el nombre aun sin haber hecho contacto con los dispositivos [34].

En el método de sincronización, se usa la clase `LocalDevice`, la cual define las funciones básicas del *Bluetooth Manager*. El *Bluetooth Manager* provee el más bajo nivel de interface con la pila de Bluetooth. Esta clase provee el acceso y control del dispositivo local de Bluetooth [35].

### 3.2.1.2 Clase encargada del descubrimiento de servicios.

Esta clase nos permitirá encontrar los servicios que ofrece el o los dispositivos localizados previamente para luego poder utilizarlos mediante el `DiscoveryAgent`.

La clase `DiscoveryAgent` para configurar los dispositivos y realizar la búsqueda de dispositivos. Un dispositivo local debe tener solo un objeto `DiscoveryAgent`.

Descubrimiento de dispositivos: Hay dos maneras de descubrir dispositivos. La primera, una aplicación debe usar `startInquiry()` para empezar una búsqueda para encontrar dispositivos próximos al dispositivo local. Los dispositivos descubiertos son retornados por el método `deviceDiscovered()` de la interfaz `DiscoveryListener`. La segunda manera para descubrir dispositivos es por el método `retrieveDevices()`. Este método retornara dispositivos que han sido descubiertos por medio de una búsqueda previa o dispositivos que son clasificados como pre-conocidos (los dispositivos pre-conocidos son aquellos que son definidos en el Centro del Control de Bluetooth como los dispositivos que se contactan frecuentemente). El método `retrieveDevices()` no realiza una búsqueda, pero provee una forma para obtener dispositivos que deberían estar en el área.

Descubrimiento de servicios: La clase `DiscoveryAgent` también encapsula la funcionalidad provista por el perfil de aplicación del servicio de descubrimiento. La clase provee una interface para una aplicación para buscar y recibir atributos para un servicio particular. Hay 2 formas para buscar por servicios. Para buscar por un servicio en un simple dispositivo, el método `searchServices()` debe ser utilizado. En otra manera, si a ti no te importa cual servicio está activo en un servicio, el método `selectService()` hace un servicio de búsqueda en un grupo de dispositivos remotos[36].

```
DiscoveryListener listener = new DiscoveryListener() {

    public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
        for (int i = 0; i < servRecord.length; i++) {
            String url =
servRecord[i].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT, true);
            if (url == null) {
                continue;
            }
            serviceFound.add(url);
            DataElement serviceName = servRecord[i].getAttributeValue(0x0100);
            if (serviceName != null) {
                System.out.println("service " + serviceName.getValue() + " found " + url);
            } else {
                System.out.println("service found " + url);
            }
        }
    }

    public void serviceSearchCompleted(int transID, int respCode) {
        System.out.println("service search completed!");
        synchronized(serviceSearchCompletedEvent){
            serviceSearchCompletedEvent.notifyAll();
        }
    }

};
```

Figura 3.2.1.2.1 Código para la localización de los servicios que ofrece cada dispositivo.

Este método usa el mismo tipo de Listener usado en la clase encargada en la localización de dispositivos, solo que ahora se usan los métodos restantes los cuales son: `servicesDiscovered` y `serviceSearchCompleted` como se puede observar en la figura 3.2.1.2.1. A continuación se describe brevemente que hace cada método.

`serviceDiscovered` es el método que es llamado cuando los servicios son encontrados durante una búsqueda de servicios, mientras que `serviceSearchCompleted` es llamado cuando la búsqueda se ha terminado o hay algún error.

```
synchronized(serviceSearchCompletedEvent) {
    System.out.println("search services on " + btDevice.getBluetoothAddress() + " " +
btDevice.getFriendlyName(false));
    LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs,
searchUuidSet, btDevice, listener);
    serviceSearchCompletedEvent.wait();
}
```

Figura 3.2.1.2.2 Código llamado cuando se completa la búsqueda de dispositivos

La figura 3.2.1.2.2 es la función llamada por el evento de cuando `serviceSearchCompleted` es activado. Como se podrá ver el `DiscoveryAgent` se obtiene de `LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs, searchUuidSet, btDevice, listener)`, y se usa para obtener los servicios del dispositivo al que se quiera acceder. Sus parámetros de entrada se describen a continuación:

`attrIDs`: indica los atributos cuyos valores se recuperarán en los servicios que tienen los UUID especificados en `uuidSet`

`searchUuidSet`: el conjunto de UUID que están por los que se buscan; Los servicios encontrado retornaran de vuelta todos los UUID especificado aquí. La clase `UUID` define identificadores únicos universales. Estos enteros sin signo de

128-bits se garantiza que sean únicos en todo tiempo y espacio. En consecuencia, una instancia de esta clase es inmutable. La especificación Bluetooth proporciona un algoritmo que describe cómo un UUID de 16-bit o 32-bit podría ser promovido a un UUID de 128 bits. En consecuencia, esta clase proporciona una interfaz que ayuda en la creación de aplicaciones de 16-bits, 32-bits y 128-bits UUID largos. Los métodos admitidos por esta clase de pruebas permiten la igualdad de dos objetos UUID.

El *Bluetooth Assigned Numbers* [37] define un largo número de UUIDs para protocolos y clases de servicios. La tabla 3.2.1.2.1 provee una corta lista de los más comunes UUIDs definidos en el Bluetooth Assigned Numbers.

Name	Value	Size
Base UUID Value (Used in promoting 16-bit and 32-bit UUIDs to 128-bit UUIDs)	0x0000000000001000800000805F9B34FB	128-bit
SDP	0x0001	16-bit
RFCOMM	0x0003	16-bit
OBEX	0x0008	16-bit
HTTP	0x000C	16-bit
L2CAP	0x0100	16-bit
BNEP	0x000F	16-bit
Serial Port	0x1101	16-bit
ServiceDiscoveryServerServiceClassID	0x1000	16-bit
BrowseGroupDescriptorServiceClassID	0x1001	16-bit
PublicBrowseGroup	0x1002	16-bit
OBEX Object Push Profile	0x1105	16-bit
OBEX File Transfer Profile	0x1106	16-bit
Personal Area Networking User	0x1115	16-bit
Network Access Point	0x1116	16-bit
Group Network	0x1117	16-bit

Tabla 3.2.1.2.1. UUIDs comúnmente utilizadas

btDevice: Activa la búsqueda para servicios remotos de Bluetooth.

Listener: El objeto que recibirá los eventos cuando los servicios sean encontrados.

### 3.2.1.3 Lectura de Datos.

Aquí se mostrara la parte que es encargada de poder leer los datos para así luego poder graficarlos. Primero se localizara el dispositivo al que se quiere acceder (en este caso el dispositivo Bluetooth que está conectado al microcontrolador), luego de localizar el dispositivo, se buscarán los servicios que otorga y conectarse al servicio que permite la comunicación adecuada (En este caso los relacionados con RFCOMM). Luego de eso se abrirá un flujo, el cual permitirá entablar comunicación entre el microcontrolador y la computadora por medio de sus respectivos dispositivos Bluetooth. En la figura 3.2.1.3.1 se puede observar el funcionamiento de la adquisición de los datos por medio del módulo Bluetooth.

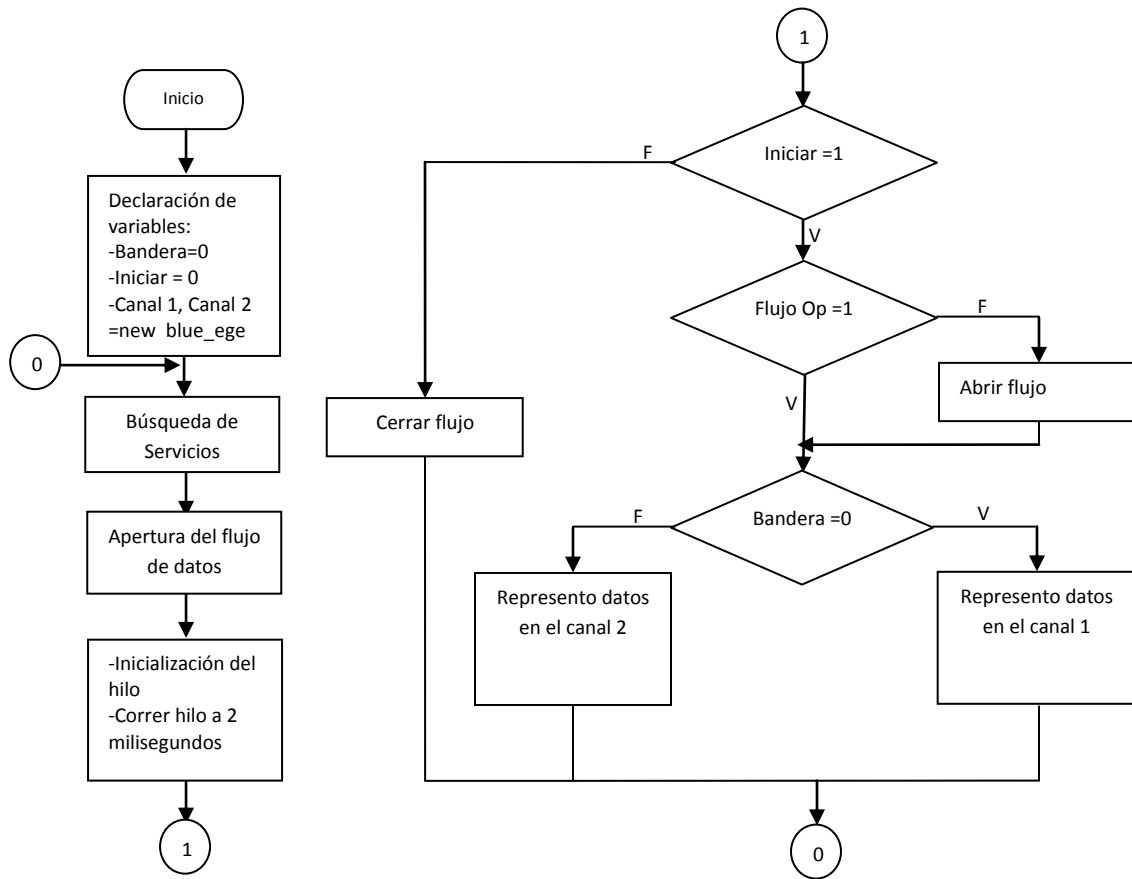


Figura 3.2.1.3.1. Diagrama de flujo que indica cómo se obtienen los datos.

Primero se llamara a la clase ServicesSearch, la cual previamente hicimos y se encarga de localizar los servicios para así tener una lista tanto de los dispositivos que se localizaron como sus respectivos servicios. En la Figura 3.2.1.3.1 se puede observar el código que se encarga de esta labor.

```
ServicesSearch.main(null);
    if (ServicesSearch.serviceFound.size() == 0) {
        System.out.println("Servicio no Encontrado");
        return;
    }
}
```

Figura 3.2.1.3.2. Código que se encarga de llamar el método que se encarga de buscar los servicios.

Aquí se manda a llamar la lista del primer servicio que se encontró y al cual se conectara, en este caso el modulo del microcontrolador. En la figura 3.2.1.3.2 se muestra el código que se encarga de crear un flujo usando la interfaz `StreamConnection` en el método `Connector.open`. Este último permite poder acceder al servicio previamente encontrado y escogido, mientras que la interfaz `StreamConnection` definirá las capacidades que una conexión de flujo debe tener. En una implementación típica de esta interface, todos los `StreamConeccion` tienen un `InputStream` y un `OutputStream`. Un `DataInputStream` equivale a un `InputStream` y un `DataOutputStream` cuenta como un `OutputStream`. Luego de abrir los flujos de entrada y salida, se envía un 1 para decirle al microcontrolador que envíe la información.

```
cadenaConexion=(String)ServicesSearch.serviceFound.elementAt(0);
conn=(StreamConnection)Connector.open(cadenaConexion,Connector.READ_WRITE,true);
System.out.println("Encefalografo conectado...");
din = new DataInputStream(conn.openInputStream());
dout= new DataOutputStream(conn.openOutputStream());
dout.write(1);
```

Figura 3.2.1.3.3. Inicialización de los flujos de datos para la comunicación Bluetooth.

Ahora se declara y ejecuta un hilo (figura 3.2.1.3.3).

```
tLectura = new Thread( this );  
tLectura.start();
```

Figura 3.2.1.3.4. Inicialización del hilo.

Este hilo manda a llamar la función leer\_dato y luego dormirá el hilo 2 milisegundos como se muestra en la figura 3.2.1.3.4. Este tiempo se da para que se sincronice con el microcontrolador, además de que como los datos son almacenados en un arreglo de 256, este tiempo permite que el almacenamiento sea lo suficientemente rápido como para cambiar entre muestreo y muestreo cada segundo.

```
public void run()  
{  
    try  
    {  
        while(true){  
            this.leer_dato();  
            Thread.sleep( 2 );  
        }  
    }  
    catch( InterruptedException e ) {System.out.println("Exception Occured: " +  
e.toString());}  
}
```

Figura 3.2.1.3.5. Método que se manda a llamar con el hilo.

La función leer dato, que se mencionó previamente y se muestra en la figura 3.2.1.3.5, se encarga de procesar los datos enviados por el Bluetooth para luego poder graficarlos. También permite activar las banderas encargadas para poder aplicar la transformada de Fourier sobre la gráfica. En cuanto a la visualización de los datos, estos se hacen por medio del método serial de la clase lector, la cual se controla por medio de banderas, ya que el electroencefalógrafo es de 2 canales y por lo tanto se tiene que intercalar los datos de canal a canal. Esto se hace así, ya que el envío de datos por medio del microcontrolador lo hace

cada 2 milisegundos, enviando primero el dato que recibe del primer canal analógico y luego el segundo para empezar de nuevo.

```

public void leer_datos(){
    try{
        lector1.ActFourier(fourier);
        lector2.ActFourier(fourier);
        if(iniciar){
            init=true;
            if (close_stream==true){
                conn=(StreamConnection)Connector.open(cadenaConexion,Connector.REA
D_WRITE,true);
                din = new DataInputStream(conn.openInputStream());
                dout= new DataOutputStream(conn.openOutputStream());
                dout.write(1);
                System.out.println("Stream Abierto");
                close_stream=false;
            }
            if(tambuff1<256||tambuff2<256){
                if (bandera==0){
                    buffercillo1[tambuff1]=din.read();
                    driver.Llenar(buffercillo1[tambuff1]);
                    tambuff1++;
                    bandera=1;
                }
                if(bandera==1){
                    buffercillo2[tambuff2]=din.read();
                    driver2.Llenar(buffercillo2[tambuff2]);
                    tambuff2++;
                    bandera=0;
                }
            }
            else{
                dout.write(0);
                for(int i=0;i<256;i++){
                    lector1.serial(buffercillo1[i]);
                    lector2.serial(buffercillo2[i]);
                }
                tambuff1=0;
                tambuff2=0;
                dout.write(1);
            }
        }
        else{
            if (init==true){
                dout.write(0);
                tambuff1=0;
                tambuff2=0;
                din.close();
                dout.close();
                conn.close();
                System.out.println("Stream Cerrado");
                close_stream=true;
                init=false;
            }
        }
    }
    catch (Exception e) {System.out.println("Exception Occured: " + e.toString());
    }
}
    
```

Figura 3.2.1.3.6. Método que se encarga de la adquisición de datos.

### 3.3 VISUALIZACIÓN DE DATOS ENVIADOS POR LA PLACA DE DESARROLLO.

En esta sección se revisa lo que serían las clases encargadas para la visualización de los datos obtenidos por medio de la comunicación entre la computadora y la placa de diseño. Con esto se hablara básicamente de dos clases: Una que se encargara de recoger la información y la otra que se encargara de dibujar las líneas. Se empezara primero por esta última ya que es la base de poder hacer esta tarea. En la figura 3.3.1 se puede ver el funcionamiento general de este método, con el cual se podrá representar los datos obtenidos por medio del Bluetooth.

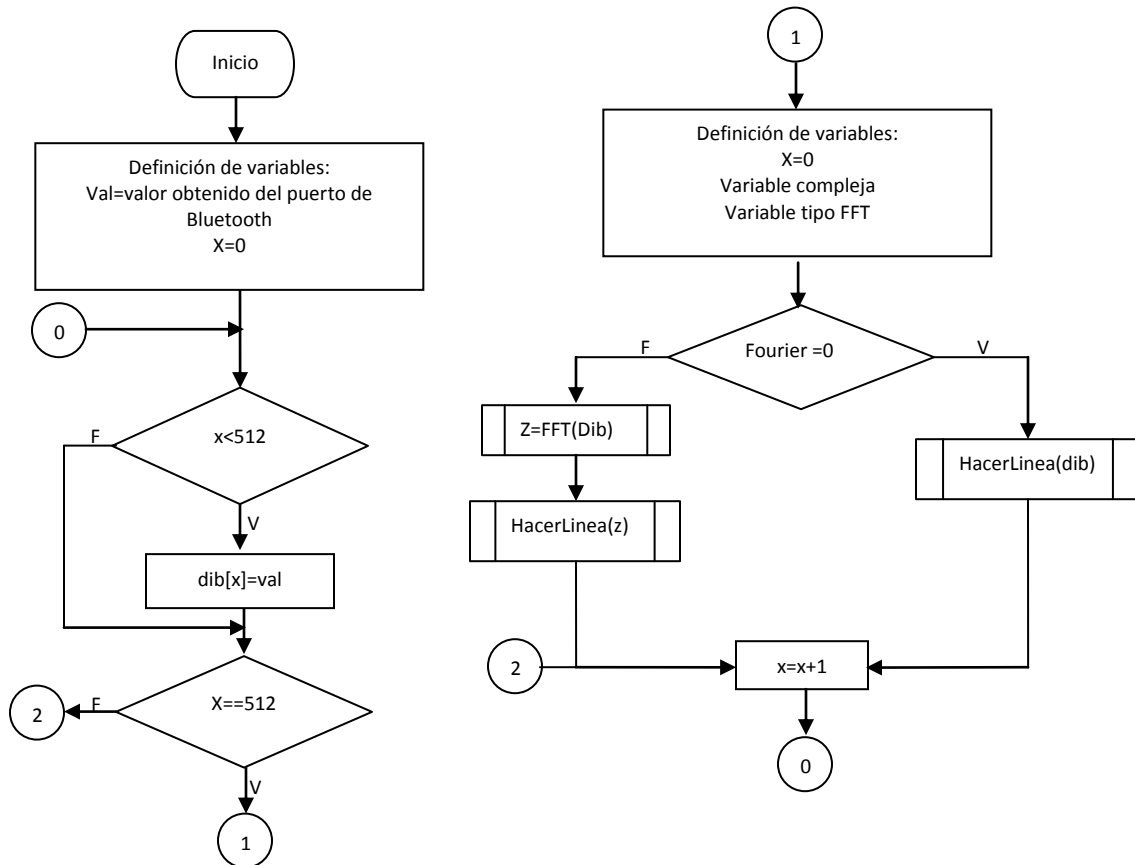


Figura 3.3.1. Diagrama de flujo del método que se encarga de representación de datos

### 3.3.1 CLASE QUE GRAFICA EL ARREGLO DE DATOS

La figura 3.3.1.1 es un método que se encarga de dibujar en el lienzo los datos obtenidos por medio del módulo Bluetooth. Simplemente de un array de valores, obtenido de la clase `blue_ege`, va uniendo los puntos para luego plasmarlos en el lienzo. Usa la API `awt`, la cual contiene todas las clases y métodos necesarios para la creación de interfaces graficas [38], aunque en este caso se usara para poder dibujar en un lienzo. La clase heredara de la clase `Panel`, la cual es la clase contenedora más simple y se encargara de poder proveer un espacio donde una aplicación pueda agregar cualquier otro componente [39]. De esta clase se usara el método `paint`, que pinta el contenedor [40] y recibe como parámetro un objeto de tipo `Graphics`, una clase abstracta base para todo el contexto gráfico que permite a alguna aplicación dibujar adentro de los componentes que son realizados en varios dispositivos [41].

```
public void paint(Graphics w){
    int y2=0; int y1=0; int x2=0; int x1=0; int c=0; int cont=0;
    y1=0; x1=0; x2=1;
    w.setColor(Color.white);
    w.drawString ("Número de pulsos", 90,20);
    while(c<512) {
        y2=vectorp[c]*3;
        w.setColor(Color.white);
        w.drawLine(0,150,512,150);
        w.drawLine(0,152,512,152);
        w.drawLine(0,0,0,300);
        w.drawLine(2,0,2,300);
        w.setColor(Color.red);
        w.drawLine(x1,150-y1,x2,150-y2);
        w.setColor(Color.red);
        y1=y2;
        x1=x2;
        x2=x2+1;
        c++;
        //aca se cuentan los pulsos
        if (y2>20) //&& (y2<60))
        {
            w.setColor(Color.black);
            w.fillRect(200,10,50,12);
            w.setColor(Color.blue);
            contador++;
            w.drawString(String.valueOf(contador), 200, 20);
        }
    }
}
```

Figura 3.3.1.1. Código que dibuja los arreglos de datos en el lienzo.

### 3.3.2 CLASE QUE SE ENCARGA DEL CONTROL DE DATOS.

Esta es la clase (blue\_ege) que se encarga de controlar los datos enviados por el módulo antes de su graficación. En esta clase también se decide si aplicar la transformada de Fourier o no. Como se podrá ver en el código siguiente, se usa el método paint, pero esta vez para mandar a llamar un objeto de tipo graficador (declarado como graf) el cual se usa para mandar a llamar los métodos declarados en esa clase. Con ello se crea un arreglo de 512 llamado dib y se pasa directamente a la función hace línea. Adentro de esta clase se declaran otros arreglos, pero estos servirán para aplicar la transformada de Fourier.

```
public void paint(Graphics g)
    {
        Complex[] x = new Complex[512];
        int z[]=new int[512];
        FFT frecuencia;
        frecuencia=new FFT();

        if(!banderafourier){
            graf.HaceLinea(dib);
            graf.repaint(5);
        }
    }
```

Figura 3.3.2.1. Código que se encarga de dibujar línea e inicializar las variable necesarias para la FFT

Como se ve en la figura 3.3.2.1, se checa una bandera denominada banderafourier. Si esta activa, se empieza aplicar la FFT para un análisis en base a frecuencia [11]. Las se describe en las últimas secciones de este capítulo. Como se podrá analizar, primero se convierte el arreglo con números enteros a complejos, luego se pasa ese arreglo como parámetro en el método fft de la clase FFT. Ya después para poder graficar los datos se elevan al cuadrado ambas partes de un número Complejo (parte real, parte imaginaria). Estas 2 partes se pueden obtener por medio de los métodos re e im de la clase Complex.

```
else{  
  
    for (int i = 0; i < 512; i++) {  
        x[i] = new Complex(dib[i], 0);  
    }  
    Complex[] y = frecuencia.fft(x);  
  
    for (int i = 0; i < 512; i++) {  
        z[i] = (int)(((y[i].re()*y[i].re())+(-1*y[i].im()*y[i].im()))/2000);  
    }  
    graf.HaceLinea(z);  
    graf.repaint(5);  
    }  
}
```

Figura 3.3.2.2. Continuación de 3.3.2.1 que se encarga ahora de ajustar la FFT para su representación

El método mostrado en la figura 3.3.2.2 recibe como parámetro el valor obtenido por medio del dispositivo Bluetooth, para luego introducirlo en el arreglo que será utilizado en el método anterior. Se podrá notar que este método se encarga de escoger el tamaño del área que se utilizará para la visualización. En este caso el 512 será el tamaño tanto del arreglo dib como del panel que se utiliza.

```
public void serial( int val)
{
    // Aca son lineas de control para que el puerto
    // sea vigilado
    if(x<512)
    {
        dib[x] = val;
        System.out.print(dib[x]);
        System.out.print(" ");
    }
    if(x==512)
    {
        x=0;
        grafica.resize(512,300);
        grafica.setResizable(false);
        grafica.show();
    }
    x++;
}
```

Figura 3.3.2.2. Manejo de los valores para poder llenar el arreglo, el cual será representado en el lienzo

### 3.4 INTERFAZ DE USUARIO.

En este apartado se mostrara la forma final que tendrá el programa. Para ser de fácil comprensión solo se explicaran a detalle las funciones que influyen directamente con el procesamiento de la imagen. En la imagen 3.4.1 se muestra la apariencia que tendría el electroencefalógrafo (sin haber empezado la comunicación). Como se podrá ver se le inserto una interfaz especial para poder almacenar los datos del paciente (Nombre, Apellido, DNI, Edad) además de tener el radiobutton de nombre Fourier, el cual se encarga de aplicar o no la FFT y un botón Iniciar/Parar que permite controlar la comunicación con el microcontrolador.

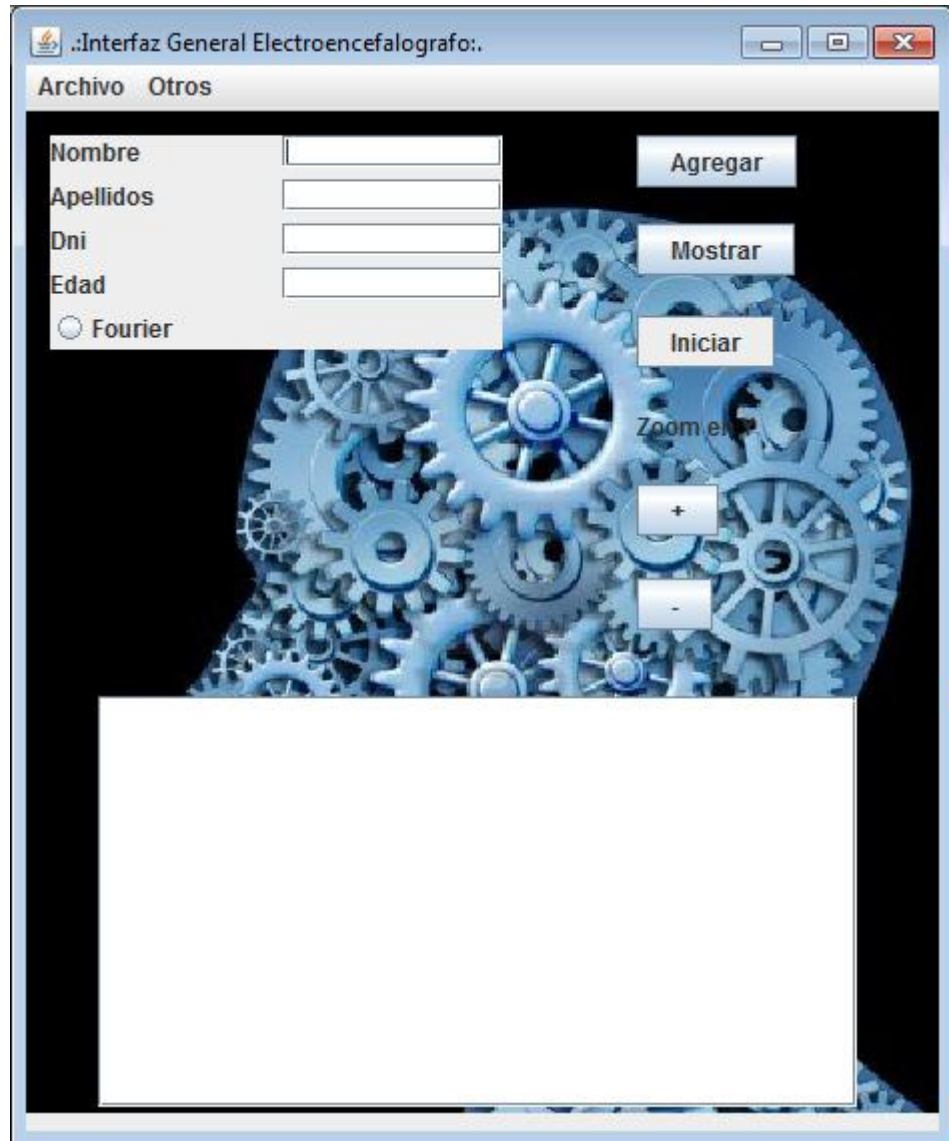


Figura 3.4.1 GUI de la aplicación

El código de la figura 3.4.2 se encarga de que cada vez que se dé un clic encima del radiobutton Fourier (que se aprecia en la figura 3.4.1) permite aplicar la transformada de Fourier (método mostrado anteriormente en el apartado 3.3.2.2), lo único que haces es solo cambiar una bandera que es pasada como parámetro al método `ActFourier(boolean b)` del objeto de clase `blue_ege`.

```
private void jbr1ActionPerformed(java.awt.event.ActionEvent evt){  
    if(jrb1.isSelected()== true ){  
        fourier=true;  
    }  
    else{  
        fourier=false;  
    }  
}
```

Figura 3.4.2. Acción realizada por el radiobutton Fourier

En la figura 3.4.3 el código mostrado se encarga de simplemente de poder parar la lectura de electroencefalógrafo, dando luz verde al método leer\_datos() para abrir o cerrar, dependiendo del caso, los streams de datos. Esto se hace cambiando la bandera iniciar, la cual es declarada de forma global y es utilizada en la clase que se encarga de controlar los datos.

```
private void jButtonIniciarActionPerformed(java.awt.event.ActionEvent evt) {  
    if(!iniciar){  
        iniciar=true;  
        jButtonIniciar.setText("Parar");  
    }  
    else{  
        iniciar=false;  
        jButtonIniciar.setText("Iniciar");  
    }  
}
```

Figura 3.4.3. Control de datos

## 3.5 CLASES COMPLEX Y FFT (TRANSFORMADA RÁPIDA DE FOURIER)

FFT es la abreviatura usual (del inglés Fast Fourier Transform) de un eficiente algoritmo que permite calcular la transformada de Fourier discreta (DFT)

y su inversa. La FFT es de gran importancia en una amplia variedad de aplicaciones, desde el tratamiento digital de señales y filtrado digital en general a la resolución de ecuaciones en derivadas parciales o los algoritmos de multiplicación rápida de grandes enteros. El algoritmo pone algunas limitaciones en la señal y en el espectro resultante. Por ejemplo: la señal de la que se tomaron muestras y que se va a transformar debe consistir de un número de muestras igual a una potencia de dos. La mayoría de los analizadores FFT permiten la transformación de 512, 1024, 2048 o 4096 muestras. El rango de frecuencias cubierto por el análisis TRF depende de la cantidad de muestras recogidas y de la proporción de muestreo [42].

En este caso la FFT nos permitirá hacer un análisis en frecuencia a las ondas cerebrales para su posterior análisis por los expertos en el área. En las secciones siguientes se explicara tanto la FFT, que es la que nos permitirá hacer la transformada de Fourier, así como la clase Complex, que nos permitirá poder utilizar la transformada de Fourier, ya que este método trabaja en base de números complejos y además devuelve un valor complejo.

### 3.5.1 CLASE COMPLEX

De los constructores ubicados en la clase complex, el que se usara será el que se muestra en la figura 3.5.1.1, ya que nos permite convertir de entero (o número real) a número complejo. Esto se logra pasando como parámetro cada uno de los números del arreglo que contiene los datos en la parte real y pasar un 0 en la parte imaginaria, para así poder tener una representación adecuada para aplicar la *transformada rápida de Fourier* (FFT por sus siglas en ingles de Fast Fourier Transform)

```
public Complex(double real, double imag) {  
    re = real;  
    im = imag;  
}
```

Figura 3.5.1.1. Constructor que vuelve de real a complejo

Ya una vez aplicada la transformada rápida de Fourier, la clase retornara un número complejo, compuesto de parte real e imaginaria. Para poderlo representar en números enteros para poderlo graficar, es necesario obtener la magnitud y fase del complejo y para eso se necesita obtener ambas partes a partir de la real y la compleja. Para obtener estos valores se usaron los métodos que se muestran en la figura 3.5.1.2.

```
public double re() { return re; }  
public double im() { return im; }
```

Figura 3.5.1.2. Funciones que regresan tanto parte real como imaginaria.

### 3.5.2 CLASE FFT (TRANSFORMADA RÁPIDA DE FOURIER)

Esta clase se encarga de calcular la FFT y la inversa de la FFT de un tamaño N de una secuencia de complejos. Como limitaciones de este código se encuentra:

- Asume que N es una potencia de 2
- No es el algoritmo de memoria más eficiente (porque usa un object type para representar números complejos y porque re-direcciona memoria para el subarreglo, luego de hacerlo en su lugar o usando un arreglo temporal)

En este caso el método que se usa fue el que se encarga de sacar la transformada rápida de Fourier. Para calcular la FFT se utiliza la ecuación (1) [43].

$$y_p = \sum_{k=0}^{n-1} x_k \left( \cos\left(2\pi \frac{kp}{n}\right) + i \sin\left(2\pi \frac{kp}{n}\right) \right) \quad (1)$$

El código mostrado en la figura 3.5.2.1 es el método encargado de aplicar la FFT. Se inicializa N con el tamaño del arreglo que es pasado como parámetro. Luego se verifica si es potencia de 2, de ser así se empieza primero sacando los valores correspondientes al índice k y k+1, para ordenarlos en pares. Ya después por medio de recursividad, se va aplicando las operaciones correspondientes y así obtener la salida de los datos ya transformados.

```

public static Complex[] fft(Complex[] x) {
    int N = x.length;

    /* caso base*/
    if (N == 1) return new Complex[] { x[0] };

    /*Comprueba si el tamaño de x[] es potencia de 2.*/
    if (N % 2 != 0) { throw new RuntimeException("N is not a power of 2"); }

    // fft of even terms
    Complex[] even = new Complex[N/2];
    for (int k = 0; k < N/2; k++) {
        even[k] = x[2*k];
    }
    Complex[] q = fft(even);

    // fft of odd terms
    Complex[] odd = even; // reusa el arreglo
    for (int k = 0; k < N/2; k++) {
        odd[k] = x[2*k + 1];
    }
    Complex[] r = fft(odd);

    /*Combina*/
    Complex[] y = new Complex[N];
    for (int k = 0; k < N/2; k++) {
        double kth = -2 * k * Math.PI / N;
        Complex wk = new Complex(Math.cos(kth), Math.sin(kth));
        y[k] = q[k].plus(wk.times(r[k]));
        y[k + N/2] = q[k].minus(wk.times(r[k]));
    }
    return y; }

```

Figura 3.5.2.1. Función de la transformada rápida de Fourier.

### 3.6 MANEJO DE DATOS.

En este capítulo, se hablará de la parte que se encarga de poder guardar los datos producidos por el electroencefalógrafo, así como poder abrirlos en el programa, para poder analizar nuevamente los resultados en caso que sea necesario para el especialista. La clase que se encarga de esto maneja los métodos que tanto se encarga de almacenar los datos como de poder leerlos, para luego poder abrirlos. En las siguientes secciones se hablaran de estos métodos.

#### 3.6.1 MÉTODO QUE SE ENCARGA DE GUARDAR LOS DATOS EN UN ARCHIVO.

Para poder hacer un análisis posterior de la prueba hecha al paciente, es necesario guardas los datos obtenidos del módulo Bluetooth. En la figura 3.6.1.1 e muestra el diagrama de flujo de este método.

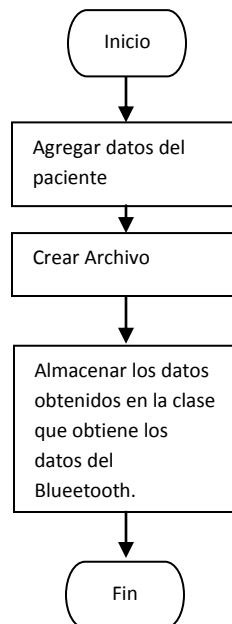


Figura 3.6.1.1 Método que se encarga de almacenamiento de datos

Básicamente el archivo se crea al momento de pulsar el botón agregar que está en la interfaz. Primero se introducen los datos en los campos que aparecen en la pantalla inicial. En la figura 3.6.1.2 se puede observar el código que es ejecutado al pulsar este botón creando un archivo por canal del electroencefalógrafo. Estos archivos que se generan se guardan utilizando la información obtenida del campo DNI.

```
private void jButtonAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    boolean f=false;
    if(this.verString(this.datos().getDni())){
        f=personas.adicionarPersona(this.datos());
        if(f){
            driver.Guardar(this.datos().getDni()+"_Canal1.txt");
            driver2.Guardar(this.datos().getDni()+"_Canal2.txt");
            Nombre1=this.datos().getDni()+" "+this.datos().getNombre()+"
"+this.datos().getApellidos()+" "+this.datos().getEdad()+"_Canal1";
            Nombre2=this.datos().getDni()+" "+this.datos().getNombre()+"
"+this.datos().getApellidos()+" "+this.datos().getEdad()+"_Canal2";
        }
    }
    this.reset();
    if(!f){
        javax.swing.JOptionPane.showMessageDialog(this,"ya se encuentra registrado");
    }
}
```

Figura 3.6.1.2 Método que se encarga agregar los datos del paciente y generar los respectivos archivos de cada canal.

Como se puede observaren la figura anterior, se manda a llamar la función guardar. En la figura 3.6.1.3 se ve el código de esta función, que recibe como parámetro un string que será el nombre del archivo, en este caso el DNI del paciente más el número de canal. Ya después se crea el archivo.

```
public boolean Guardar(String Name){  
    try{  
        Archivo=new PrintWriter(new FileWriter(Name));  
        return true;  
    }  
    catch(IOException ex){  
        System.out.print(ex);  
        return false;  
    }  
}
```

Figura 3.6.1.3 Método que se encarga de crear el archivo.

Luego para almacenar los datos se utilizar una función la cual recuperara el dato obtenido para luego guardarlo en el archivo. En la figura 3.6.1.4 se puede observar esta función.

```
public void Llenar(int A)throws IOException,FileNotFoundException{  
    Archivo.printf("%d\n",A);  
}
```

Figura 3.6.1.4 Método que se encarga de guardar datos en el archivo.

### 3.6.2 LECTURA DE DATOS.

Los métodos que se estudian a continuación son los que se encargan de leer los datos generados por el método anterior y poder a la vez abrirlos para su representación gráfica. En la figura 3.6.2.1 se puede observar el diagrama de flujo de este método.

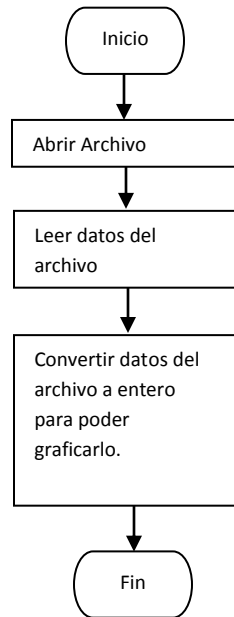


Figura 3.6.2.1 Método que se encarga de la lectura de datos

En la figura 3.6.2.2 se puede observar el método que lee los datos del archivo. Se pasa por parámetro el archivo que se desea leer y luego, hasta el final del archivo, empieza a graficar usando la función serial de un objeto de la clase blue\_ege.

```

public void Leer(String archi,String title){
    try{
        Arc = new File (archi);
        Archer = new FileReader (Arc);
        br = new BufferedReader (Archer);
        blue_ege lector1 = new blue_ege(title);
        String linea;
        while((linea=br.readLine())!=null)
            lector1.serial(Integer.parseInt(linea));
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
    
```

Figura 3.6.2.2 Método que se encarga de leer datos del archivo.

Para poder abrir los archivos por medio del programa, se hizo en la clase principal un evento de ratón ligado al textarea, el cual al pulsar el botón del ratón sobre un renglón, permite abrir el archivo relacionado con el paciente. Después de esto, se parte el texto entero en renglones. Luego se mide el tamaño de cada renglón y se compara con el valor de carreo que se retorna al hacer clic. Esto se hace en un ciclo en el cual comparan que el tamaño del renglón sea mayor que el carreo. Cuando no se cumple esta función, se retorna el valor que sería el renglón, el cual se obtiene de incrementar una variable por cada ciclo. Luego se accede al número de renglón en el arreglo de renglones, para luego fragmentar el renglón y obtener así el DNI, el cual se pasara como parámetro a la función que se encarga de leer los datos. En la figura 3.6.2.3 se muestra el código utilizado para lograr hacer esta tarea.

```

private class ManejadorClicsRaton extends MouseAdapter {
    public void mouseClicked(MouseEvent evento)
    {
        int tam=jTextAreaMostrar.getLineCount();
        String renglones [] =new String [tam];
        String texto= new String(jTextAreaMostrar.getText());
        StringTokenizer partir=new StringTokenizer(texto,"\\n");
        int Nelementos=0;
        while(partir.hasMoreTokens()){
            renglones[Nelementos]=partir.nextToken();
            Nelementos++;
        }
        int carreo=jTextAreaMostrar.getCaretPosition();
        int renglon=0;
        int sumarreglos=0;
        do{
            sumarreglos=sumarreglos+renglones[renglon].length();
            renglon++;
        }while(carreo>=sumarreglos&&renglon<Nelementos);
        int val=renglon-1;
        String frase=new String(renglones[val]);
        StringTokenizer partir2=new StringTokenizer(frase,"\\t");
        String palabra [] =new String [20];
        int Npal=0;
        while(partir2.hasMoreTokens()){
            palabra[Npal]=partir2.nextToken();
            Npal++;
        }
        String NombreArchivo1= palabra[0]+"_Canal1.txt";
        String NombreArchivo2= palabra[0]+"_Canal2.txt";
        String Nombre1=frase+"_Canal1";
        String Nombre2=frase+"_Canal2";
        driver.Leer(NombreArchivo1,Nombre1);
        driver2.Leer(NombreArchivo2,Nombre2);

    }
}

```

Figura 3.6.2.3 Evento del Ratón

# Capítulo 4. Pruebas y Resultados

## Introducción.

Se realizaron pruebas en varios sujetos de prueba para mostrar el funcionamiento tanto en la parte de software como en la parte de hardware.

Las pruebas se hicieron sobre 2 etapas y un proceso:

La etapa de acondicionamiento que recibe la señales del cerebro, el proceso de transmisión, el cual se encarga de enviar la información recibida por la fase de acondicionamiento a la computadora por medio del microcontrolador y el módulo Bluetooth. En este punto el microcontrolador recibe la información y la codifica vía serial para enviarla por medio del módulo de comunicación y la computadora se encargara de rastrear el módulo de comunicación.

Y por último la etapa de visualización, la cual se encargará de graficar los datos en dos paneles diferentes tanto el lóbulo derecho como el izquierdo.

## 4.1 APLICACIÓN

La actividad bioeléctrica cerebral puede captarse por diversos procedimientos:

- Sobre el cuero cabelludo.
- En la base del cráneo.
- En cerebro expuesto.
- En localizaciones cerebrales profundas.

Para captar la señal se utilizan diferentes tipos de electrodos:

- Electrodo superficial: Se aplican sobre el cuero cabelludo.
- Electrodo basal: Se aplican en la base del cráneo sin necesidad de procedimiento quirúrgico.
- Electrodo quirúrgico: para su aplicación es precisa la cirugía y pueden ser corticales o intracerebrales.

El registro de la actividad bioeléctrica cerebral recibe distintos nombres según la forma de captación:

- Electroencefalograma (EEG): cuando se utilizan electrodos de superficie o basales.
- Electroencefalograma (ECoG): si se utilizan electrodos quirúrgicos en la superficie de la corteza.
- Estéreo Electroencefalograma (E-EEG): cuando se utilizan electrodos quirúrgicos de aplicación profunda.

Existen diferentes sistemas para obtener las señales del cerebro (Illinois, Montreal, Aird, Cohn, Lennox, Merlis, Oastaut, Schwab, Marshall, etc.), el sistema internacional «Diez-Veinte» es el más utilizado actualmente [44].

En la figura 4.1.1 [45] se muestra el diagrama que se sigue para cumplir con este sistema. Por ahora se usarán los electrodos que van conectados a los nodos F7, Fp1, Fp2, F8. Se escogió esta área ya que el electroencefalógrafo diseñado está compuesto por 4 electrodos de entrada, además de que esta área, conformada por la sien, es la que presenta menos impedimentos a la hora de tomar las señales por cuestión de que en esa zona no hay cabello que llegue a interferir con la señal.

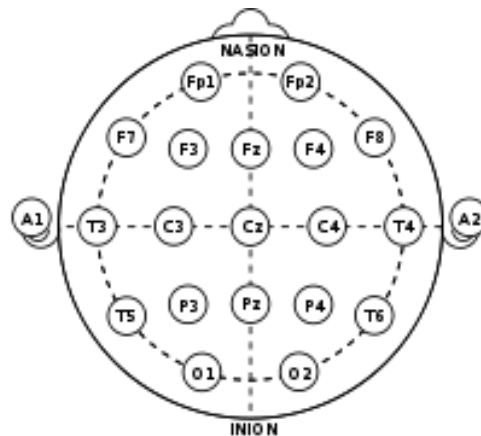


Figura 4.1.1 Diagrama que muestra la forma de conectar los electrodos en base al sistema internacional 10-20

Para poder mejorar la conducción de la señal y así poder ayudar a tomar una muestra más fiable, fue necesario usar un elemento denominado electrogel, además fue necesario que el paciente se lavara la zona de la cabeza usando jabón natural y no usar compuesto alguno (fijador, spray, etc.) para no interferir con la señal. En la Figura 4.1.2 se puede observar como quedarían puestos los electrodos en el paciente.



Figura 4.1.2 Sujeto de pruebas con electrodos ya implantados.

En la tabla 4.1.1 se pueden observar los diferentes pacientes de pruebas, los cuales están entre los 20 y 30 años de edad.

	Edad	Sexo	Clima	Observaciones
Paciente 1	20	F	Nublado	Estrés agudo, fue estimulada con un beso. Se notaron cambios en la gráfica
Paciente 2	32	M	Nublado con posibilidad de lluvia	Estrés agudo episódico. Gráfica bastante alterada.
Paciente 3	24	M	Cálido	Sin problema aparente cuando se hicieron las pruebas
Paciente 4	22	M	Templado	Se nota ligeramente alterado lobulo izquierdo del paciente.
Paciente 5	22	M	Templado	Sin problema aparente cuando se hicieron las pruebas
Paciente 6	19	M	Cálido	Ondas demasiado estimuladas.
Paciente 7	21	M	Nublado	Sin problema aparente cuando se hicieron las pruebas. Se dio a consumir un refresco, en donde se observaron que la gráfica se excitaba un poco más
Paciente 8	28	M	Templado	Sin problema aparente cuando se hicieron las pruebas.
Paciente 9	20	F	Cálido	Resfriado común
Paciente 10	21	F	Cálido	Migraña ligera en el momento del análisis
Paciente 11	24	M	Templado	Sin problemas aparente cuando se hicieron las pruebas.

Tabla 4.1.1 Sujetos de pruebas que se usaron para las pruebas.

En la figura 4.1.3 se puede observar ejemplos de estas muestras.

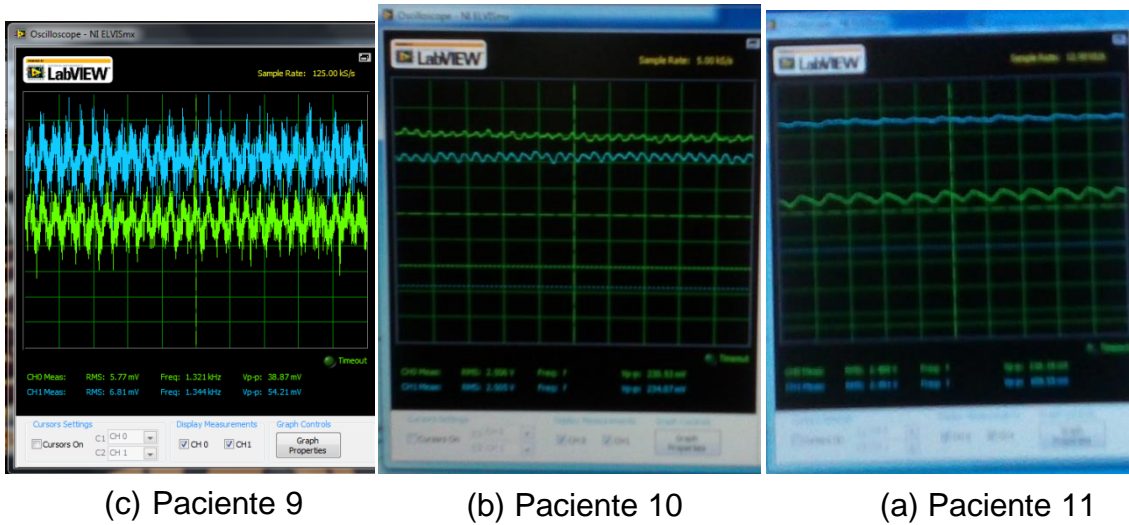


Figura 4.1.3 Algunos sujetos que se usaron para las pruebas.  
Abajo se indica el número que corresponde al sujeto de pruebas

## 4.2 HARDWARE.

Se realizaron pruebas sobre la parte de acondicionamiento por medio de la ELVIS II y un osciloscopio con transformada incluida. Las primeras pruebas se muestran a continuación en la figura 4.2.1, donde se observan los resultados realizados con la primera placa, donde se muestra que recibe la señal cerebral. Como se puede observar, las señales recibidas de ambos canales (lóbulo izquierdo y lóbulo derecho), mostradas en una escala de 10 mV con una escala de tiempo de 10 ms, presenta una forma demasiado irregular, dado que es mayor el ruido entrante que la señal de interés

. Esto se debe a que se probó la placa con ganancia unitaria, para así ver la señal resultante sin amplificación de la señal.

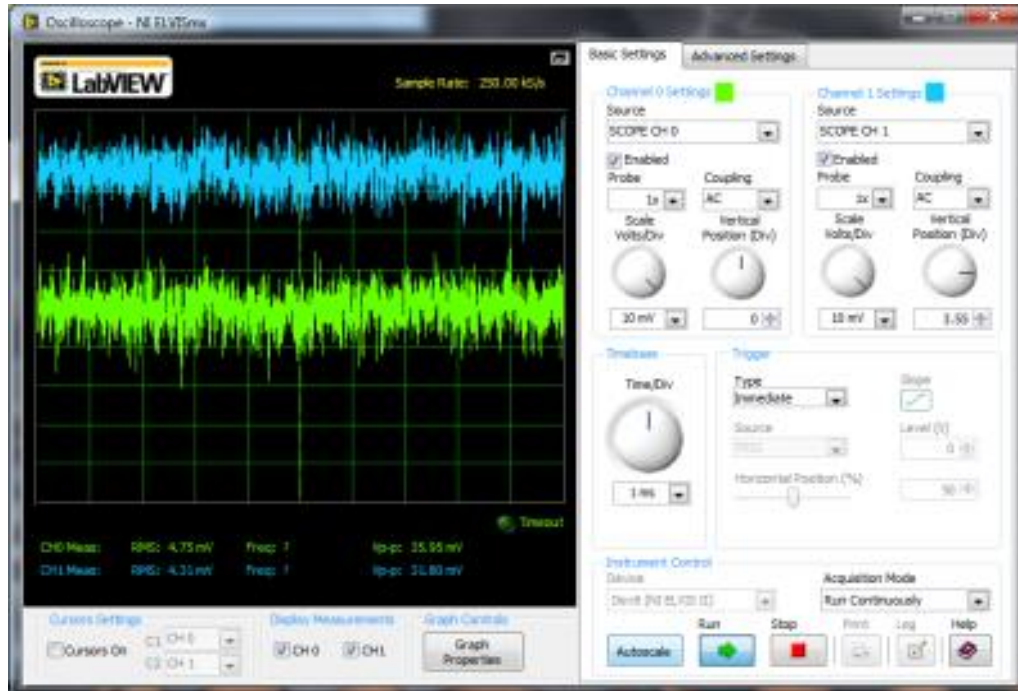


Figura 4.2.1 Primeras pruebas del electroencefalógrafo. La gráfica verde indica el lóbulo derecho y la azul el lóbulo izquierdo

Después de esto, se usó una resistencia de  $47 \Omega$  para así obtener una ganancia de 1000 en lugar de la ganancia unitaria que se estaba utilizando. En la figura 4.2.3 se muestra el resultado final de esto, haciendo que se eliminara el ruido y así mostrar una señal más nítida y congruente.

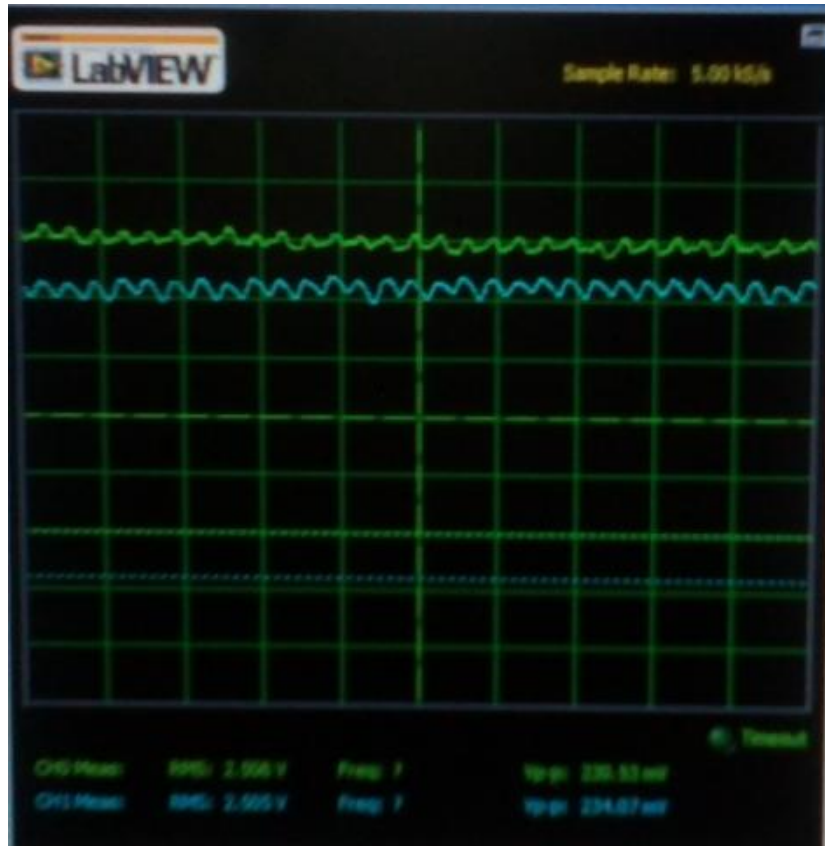


Figura 4.2.3. Encefalograma de sujeto femenino de 21 años, estudiante con migraña ligera.

En la figura 4.2.4 se muestra el diseño correspondiente a la parte de acondicionamiento. Donde se observa la conexión de los electrodos a la placa de acondicionamiento.

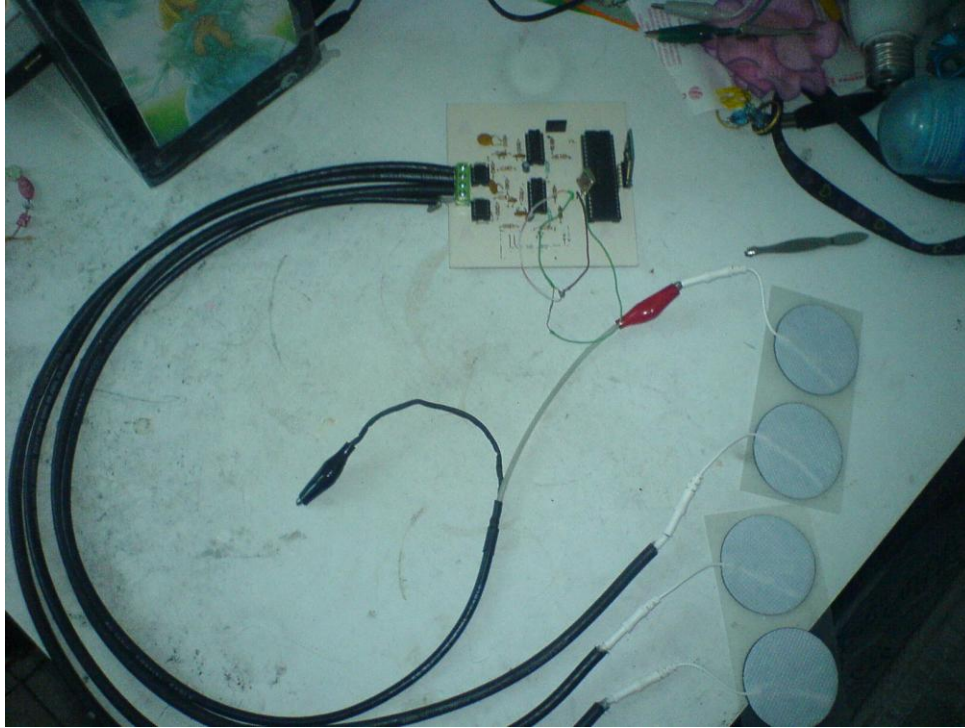


Figura 4.2.4. Diseño de placa donde se localizan acondicionamiento y digitalización.

Se utilizó cable coaxial para blindar la conexión a la placa.

### 4.3 SOFTWARE

La aplicación se encarga de recibir los datos que son enviados por el microcontrolador por medio del módulo Bluetooth y graficarlos para poder ser mostrados al analista. Para probar esta parte, se hizo una prueba en la cual se validaba señales específicas, esto se hacía pasando una señal con el voltaje más bajo posible del generador de funciones (usando un rango de mili voltios) para enviar una señal conocida al microcontrolador, y luego ser enviadas por el modulo Bluetooth perteneciente a la parte de digitalización al módulo Bluetooth de la computadora, para luego ser leído por el programa.

Para ver que la frecuencia es correcta, se cuantifico la gráfica por medio de marcas en el software, las cuales valían cada una 10 ms. Se dividió uno entre el valor de la frecuencia para el periodo de oscilación. En este caso, en una frecuencia de 30 Hz y usando

$$F = \frac{1}{T} \quad (1)$$

Despejando

$$T = \frac{1}{F} \quad (2)$$

Igualando F con 30 en (2)

$$T = \frac{1}{30} = 0.033 \text{ s} \quad (3)$$

con lo que implica que la onda debe volver a empezar cada 33 ms. Esto se probó y como se muestra en la figura 4.3.1, se puede observar que se cumple con esto, probándola en una señal de 30 Hz.

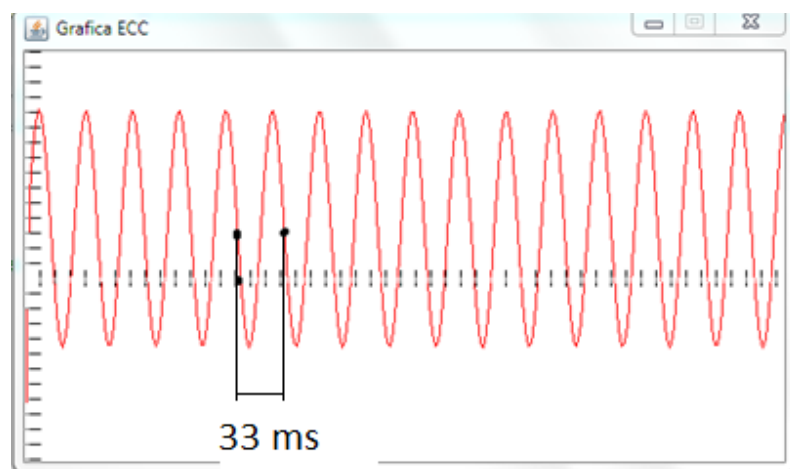
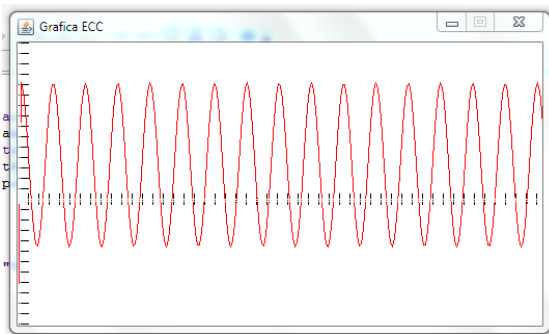
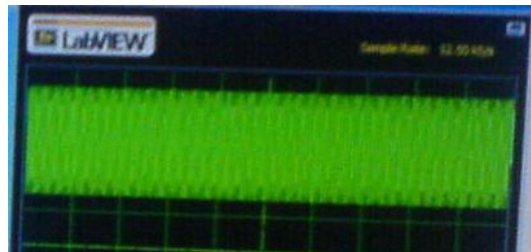
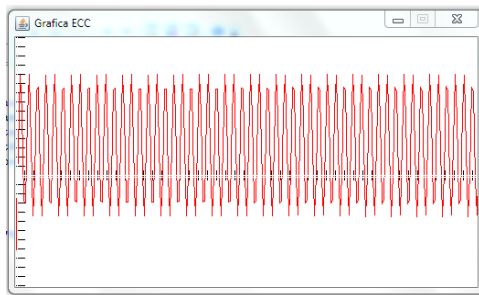


Figura 4.3.1 Señal donde se demuestra que la frecuencia corresponde a la producida por el generador de funciones

En la figura 4.3.2 se puede observar la comparación entre la señal de 30 Hz (a) y 100 Hz (b) del software y la del osciloscopio.



a) Señal de 30 Hz.



b) Señal de 100 Hz.

Figura 4.3.2. Comparación entre la señal mostrada en el osciloscopio y la señal mostrada en el software de las frecuencias sinusoidales de 30 y 100 Hz.

En la tabla 4.3.1 se muestra a continuación las diferentes señales que se utilizaron, y donde se muestra cuáles fueron mostradas correctamente.

Frecuencia (Hz)	Tipo de Señal	Tipo de Pruebas	Resultados
1	Sinusoidal	Cualitativa	Se mostró correctamente
	Cuadrada	Cualitativa	Se mostró correctamente
	Triangular	Cualitativa	Se mostró correctamente
10	Sinusoidal	Cualitativa/Cuantitativa	Mostro problemas de ajuste al principio, pero se corrigió el problema
	Cuadrada	Cualitativa	Se mostró correctamente
	Triangular	Cualitativa	Se mostró correctamente
30	Sinusoidal	Cualitativa/Cuantitativa	Se mostró correctamente
	Cuadrada	Cualitativa	Se mostró correctamente
	Triangular	Cualitativa	Se mostró correctamente
100	Sinusoidal	Cualitativa	Se mostró correctamente
	Cuadrada	Cualitativa	Se mostró correctamente
	Triangular	Cualitativa	Se mostró correctamente

Tabla 4.3.1 Pruebas que se hicieron con el programa para obtener datos del generador de funciones

Como se puede observar en la tabla anterior, todas las señales fueron mostradas correctamente por el software, para lo cual luego de esto, se procedió a probar si la transformada de Fourier era correcta. Para esto cada marca del eje X es igual a 10 Hz cada una (figura 4.3.3). Primero se probó de forma cuantitativa en la señal de 30 Hz sinusoidal, donde se puede apreciar en la figura 4.3.3 que la frecuencia es correcta, ya que la línea cruza por medio de la tercera marca.

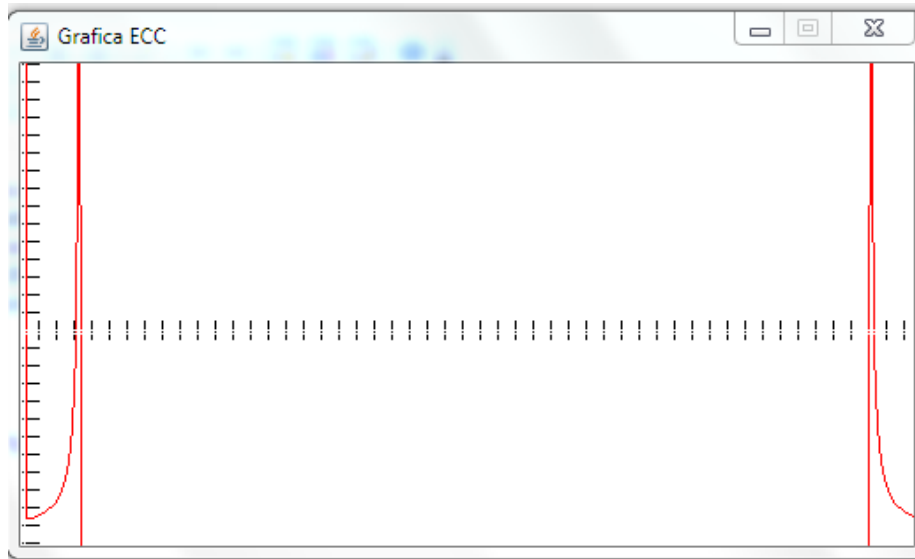


Figura 4.3.3 FFT de una señal sinusoidal de 30 Hz

En la tabla 4.3.2 se muestra las frecuencias en las que se aplicó la Transformada Rápida de Fourier, donde se puede observar que fue correcto en todos los casos.

Frecuencia (Hz)	Resultados Fourier
1	Se mostró correctamente
10	Se mostró correctamente
30	Se mostró correctamente
100	Se mostró correctamente

Tabla 4.3.2 Pruebas de la transformada de Fourier.

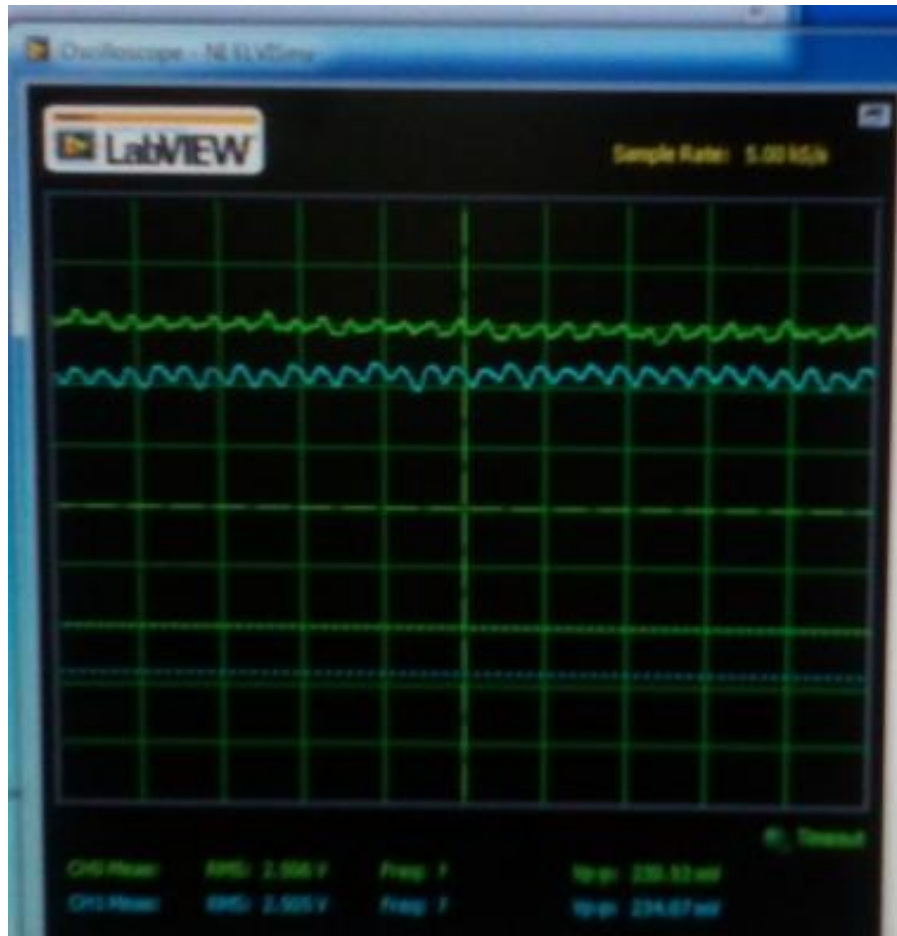
Por lo tanto la representación de la FFT es correcta.

#### 4.4 FUNCIONAMIENTO COMPLETO DEL SISTEMA.

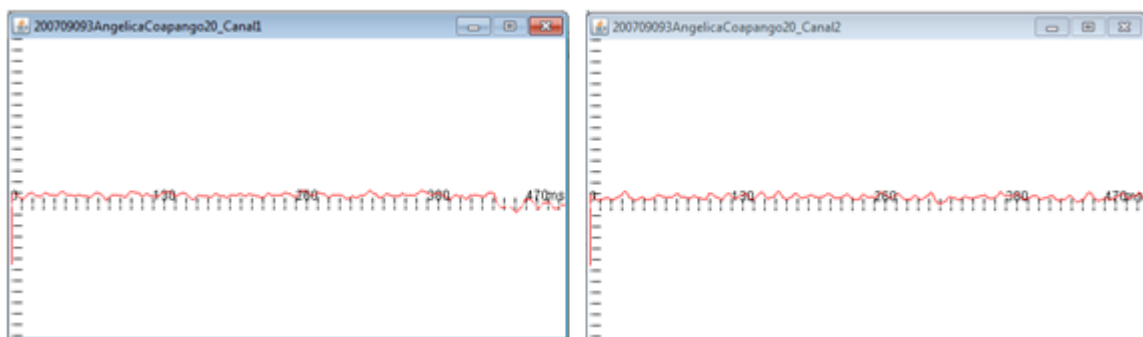
Después de validar las señales eléctricas con el programa, se prosigue a probarlo junto con la etapa de acoplamiento y registrar los datos en la computadora y permitir aplicar FFT (Transformada Rápida de Fourier) en el electroencefalograma. En la figura 4.2.3.1 se puede observar el funcionamiento

del sistema completo, mientras se compara tanto los resultados mostrados con el osciloscopio de la ELVIS II y el software hecho en Java, la señal obtenida con el osciloscopio (figura 4.2.3.1.a) a un muestreo de 10 ms y la figura 4.2.3.1.b muestra la señal obtenida con el sistema realizado en Java a 10 ms.

Como se puede observar ambas señales son iguales al mismo tiempo de muestreo, por lo tanto se puede deducir que la señal entrante es mostrada correctamente por el sistema en Java.



(a) Señal usando la ELVIS II



(b) Señal usando software en Java

Figura 4.2.3.1 Señal mostrada tanto en ELVISII como en software de Java.

En la figura 4.2.3.2, se muestra la comparación entre los datos tomados por medio del software y un electroencefalograma obtenido de una revista de neuro-psiquiatría [46], en el cual se debe prestar atención en este último en las ondas correspondientes a Fp1-F7 y Fp2-F8, ya que estas ondas son las correspondientes al lóbulo frontal izquierdo y derecho de acuerdo al sistema internacional 10-20. Se puede observar de forma cualitativa que las ondas son muy similares. En este ejemplo se quitó los ejes para poder mostrar mejor esta comparación. La amplitud en ambos casos va de 20 a 100 mV y los tiempos son de 10 ms.

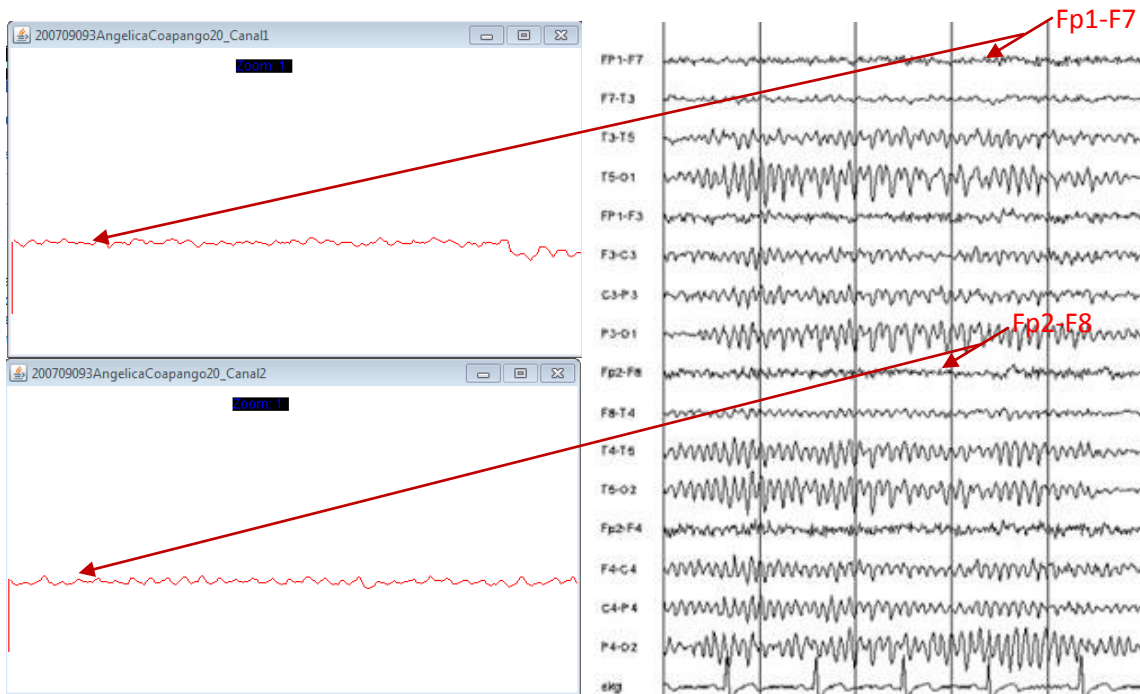


Figura 4.2.3.2 Comparación entre las señales obtenidas por el software (izquierda) y un electroencefalograma obtenido de una revista de neuro-psiquiatría (derecha).

En la figura 4.2.3.3 se puede observar la imagen del sistema en Java mientras se le aplica un zoom de 3 respecto a amplitud a la figura 4.2.3.2, donde se puede observar que no hay distorsión en la gráfica.

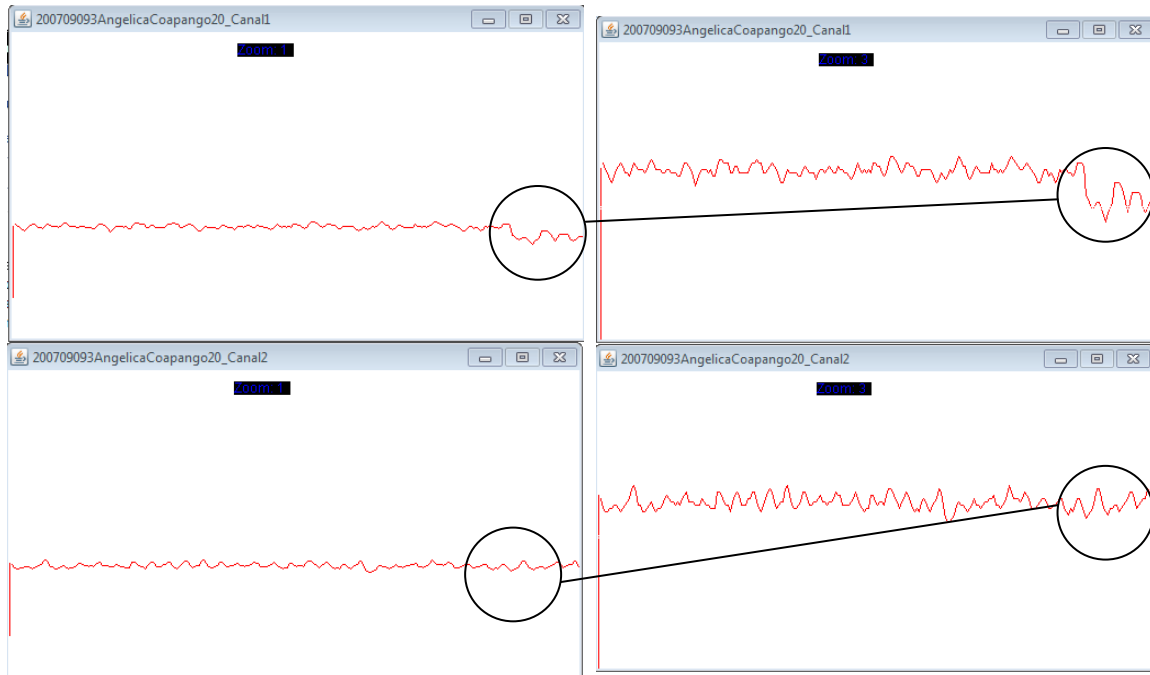


Figura 4.2.3.3. Imagen normal y aplicando zoom x3 sobre el eje y, donde con los círculos se muestra los puntos donde más claramente se muestra esto

En cuanto a las pruebas que se realizaron para guardar archivos, primero se llenan los campos de la interfaz y luego se oprime el botón agregar. Estos campos indicaran lo que sería el nombre y apellidos del paciente, el campo DNI con el cual identificaremos al paciente con un número y la edad. También se observa la aplicación de Fourier usando un radiobutton, los botones agregar para poner más pacientes, mostrar para ver los pacientes guardados e iniciar para empezar a funcionar el sistema. También se incluye 2 botones los cuales se encargan de aumentar y reducir el zoom. En la figura 4.2.3.4 se muestra esta interfaz.

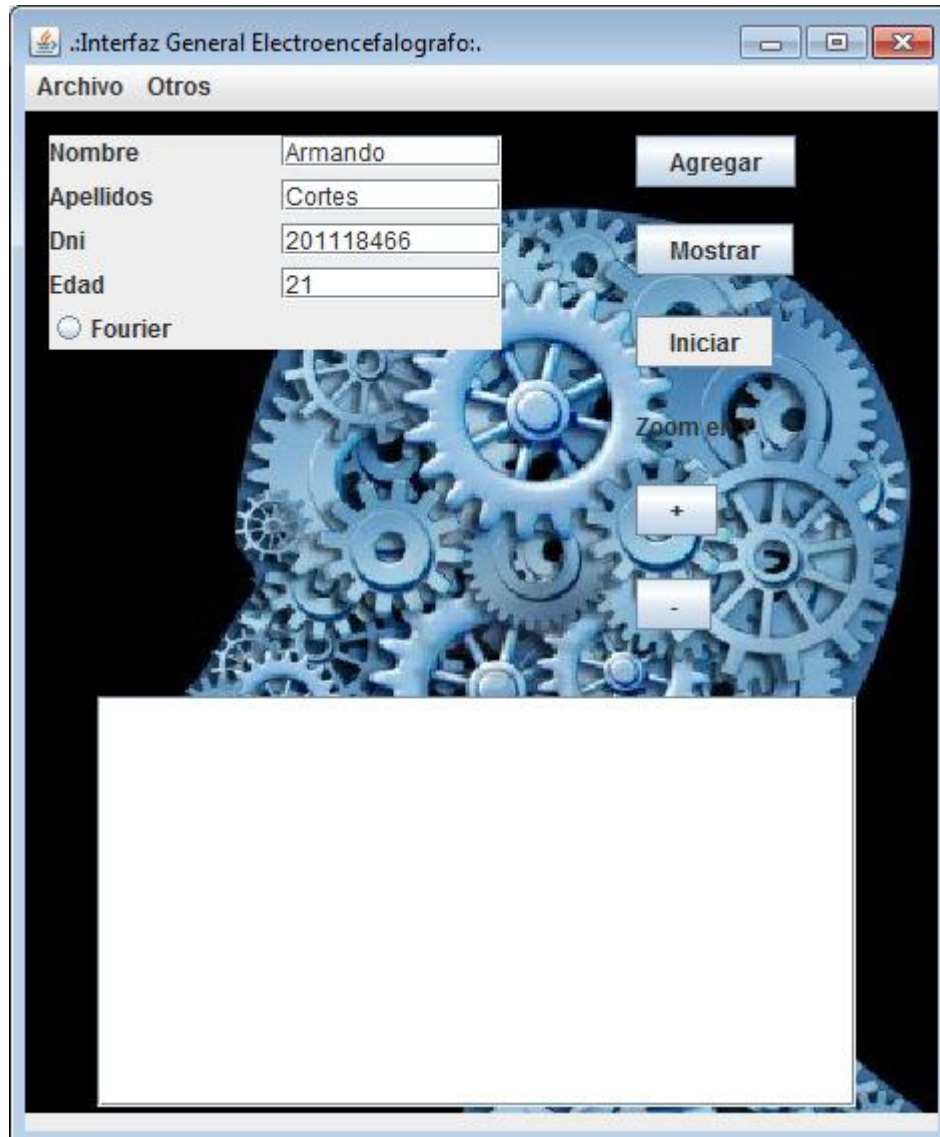


Figura 4.2.3.4. Imagen donde se muestra que campos llenar, para luego introducir el botón agregar.

Luego de hacer esto, se generarán dos archivos, los cuales se encargarán de guardar los datos de su respectivo canal del electroencefalógrafo. En la figura 4.2.3.5 se muestra estos archivos.

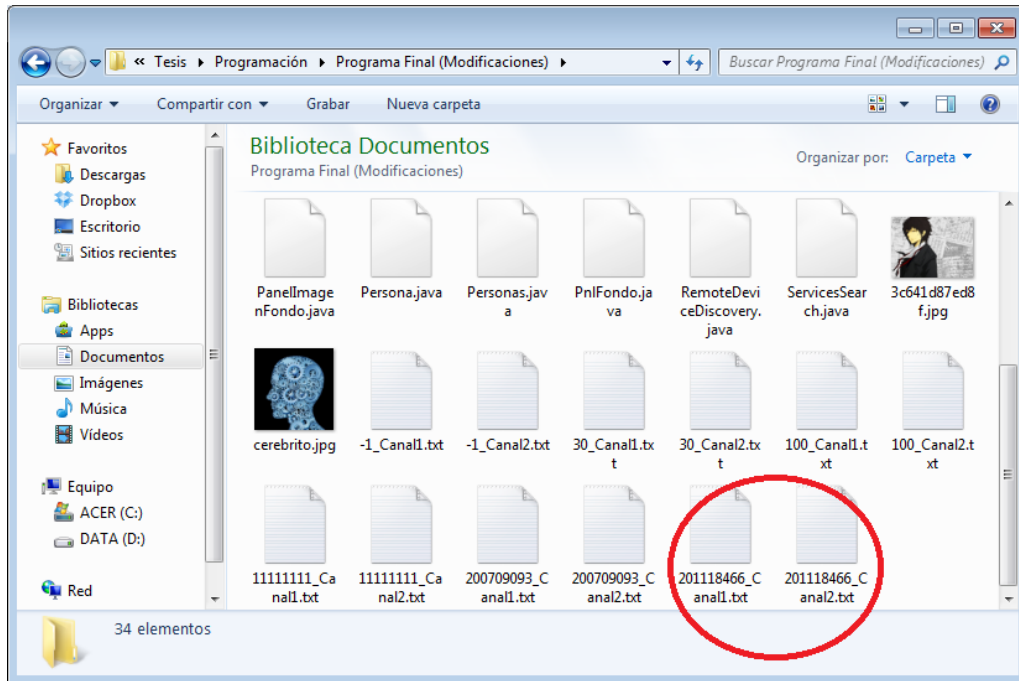


Figura 4.2.3.5. Imagen donde se muestra los archivos generados después de agregar los campos. Como se puede ver, su nombre es el campo DNI más el número de canal, en este caso es el DNI usado en la figura 4.2.3.3

Ahora después de hacer esto, para evitar que estos archivos se pierdan, es necesario hacer el siguiente paso el cual consiste en guardar el campo agregado a un archivo que tendrá el listado de pacientes. Se puede usar ya sea “guardar como” en caso de que no exista dicho archivo, o “guardar” en caso contrario. En la figura 4.2.3.6 se puede observar la localización de esos botones en el menú archivo.

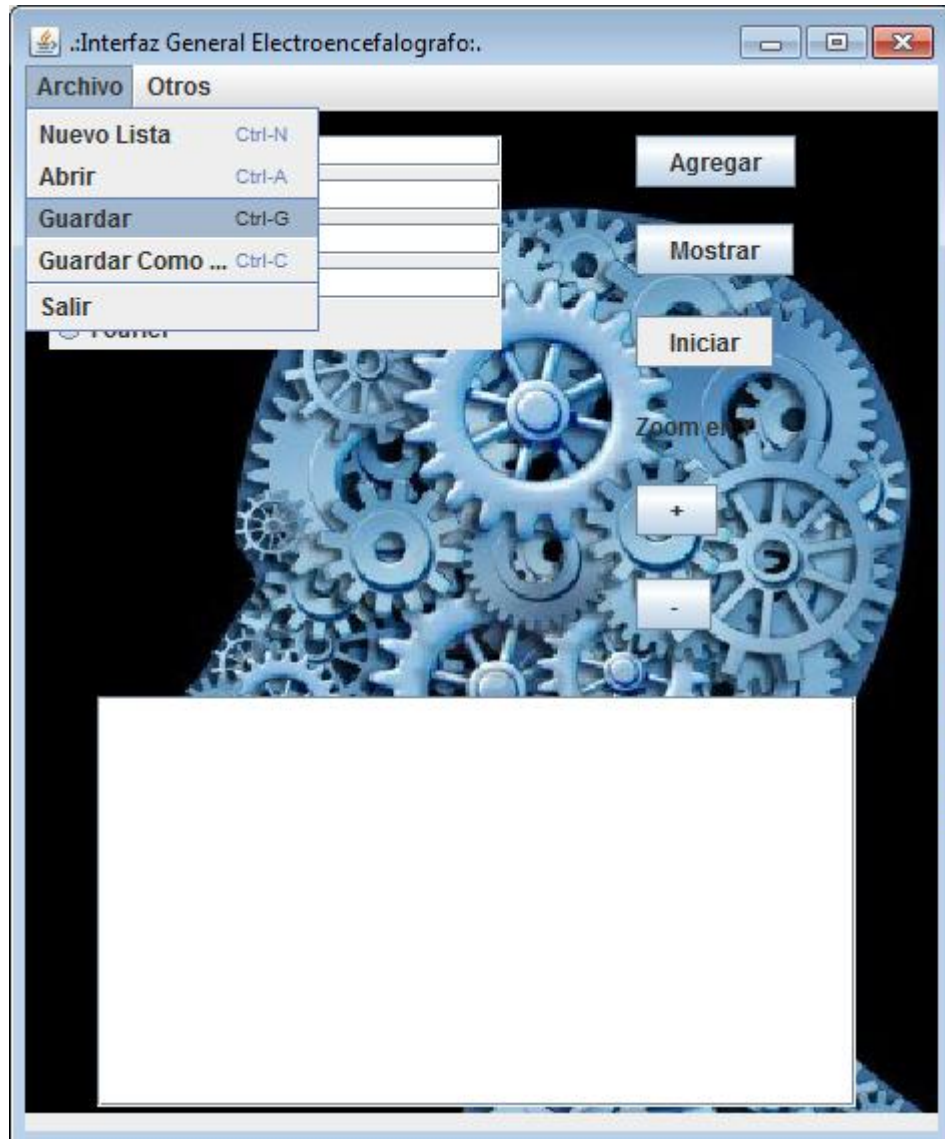


Figura 4.2.3.6. Localización de los botones de guardar y guardar como.

En cuanto a la parte de abrir el archivo, para luego mostrar sus graficas en pantalla, primero se abre la lista con los nombres de los pacientes que se hayan guardado previamente, con lo cual se accede al menú archivo y luego en la opción abrir (Figura 4.2.3.6). Para poder acceder al archivo, luego se presiona el botón mostrar, lo cual, en el área de texto, mostrara los datos de los pacientes. En la figura 4.2.3.7 se observa cómo se muestran los datos.

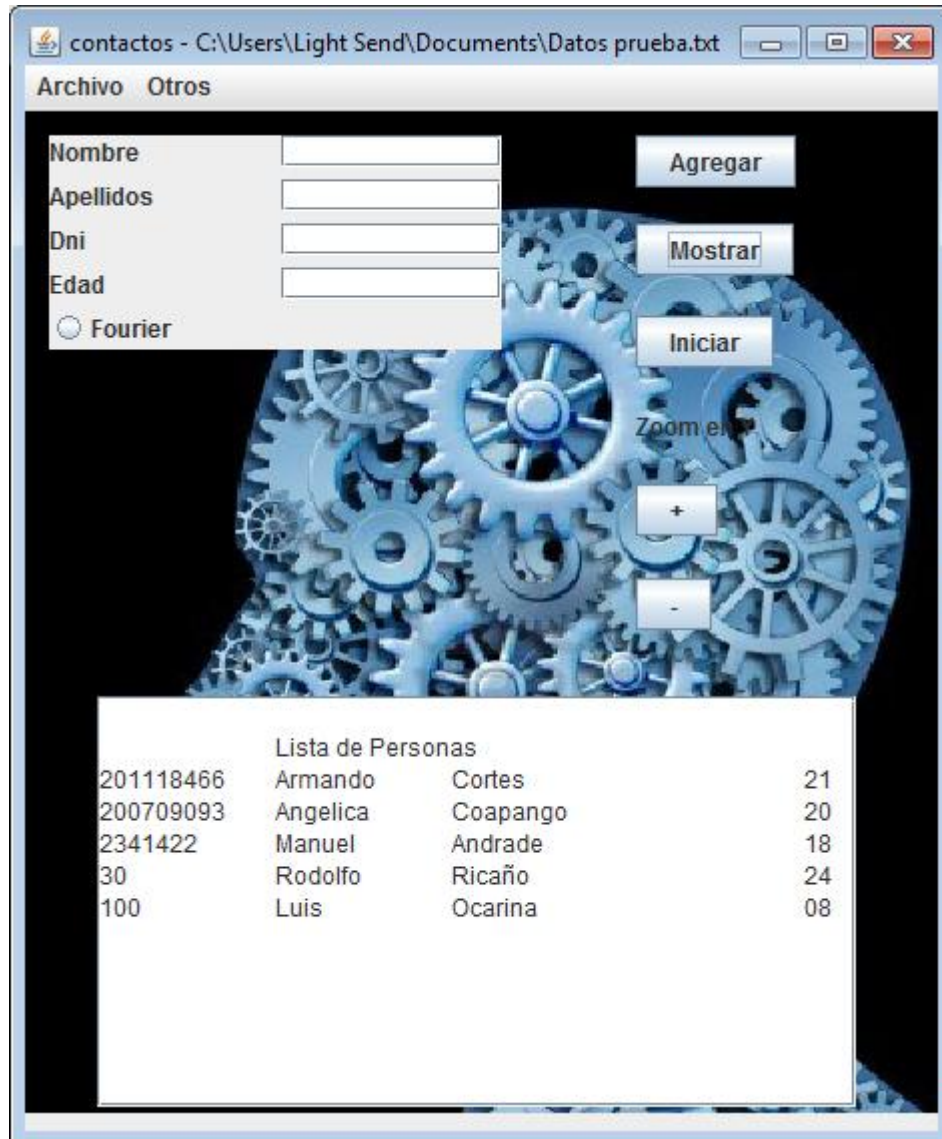


Figura 4.2.3.7. Lista datos del paciente.

Ya que se muestra la lista, se procede a hacer clic sobre uno de los pacientes, para que así, por medio de su número de DNI, se busquen los archivos que le corresponden al paciente (en este caso el que tiene DNI 200709093) y luego los grafica como se muestra en la figura 4.2.3.8.

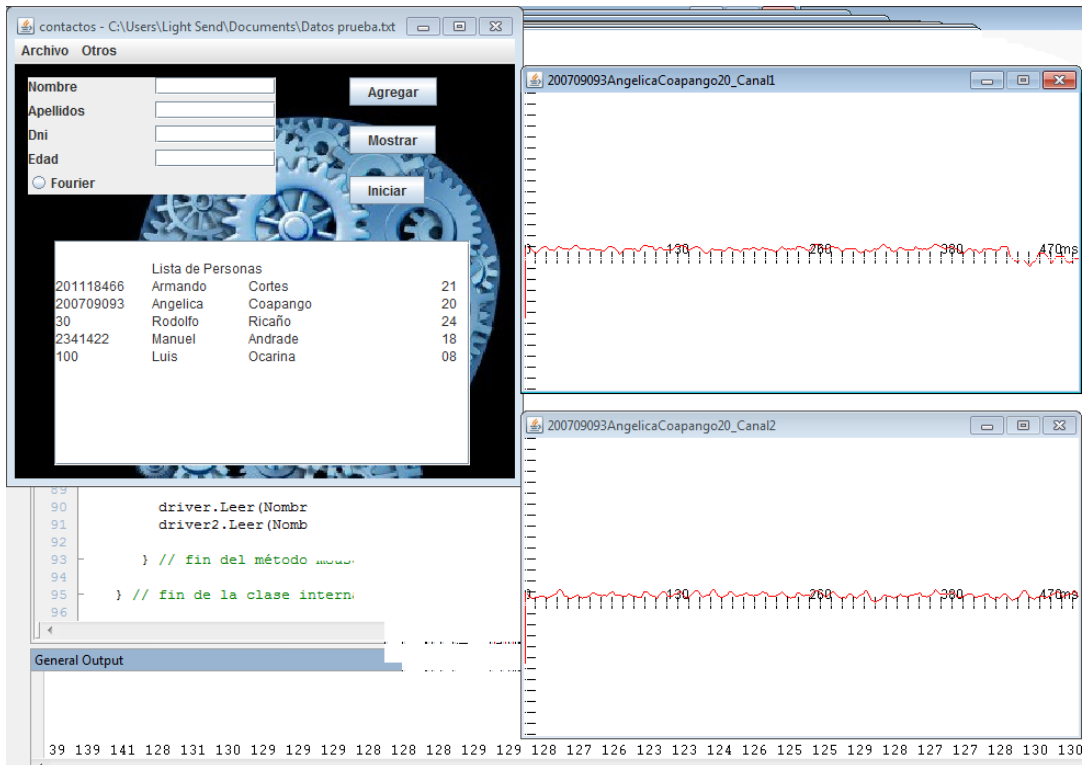


Figura 4.2.3.8. Electroencefalogramas obtenidos de archivos previamente guardados. En la derecha se muestra los nombres de los pacientes, mientras a la izquierda las gráficas utilizadas.

Con todo lo anterior se puede concluir que el sistema funciona correctamente, ya que las señales recuperadas por el sistema, además de ser leídas correctamente, son congruentes a los electroencefalogramas investigados. Dicho sistema podrá ser utilizado por expertos en el área para ser interpretados. Y contribuir a detección de trastornos cerebrales e investigación.

# Capítulo 5. Conclusiones y Trabajo a Futuro.

Con este trabajo se obtuvo un sistema capaz de poder representar las señales bioeléctricas del cerebro, el cual permitirá facilitar el diagnóstico para enfermedades de tipo neurológico, además de poder ser utilizado para pruebas como el FeedBack o análisis donde se quiera ver su comportamiento al consumir diferentes sustancias psicoactivas.

También se permite el uso de archivos, donde se guardan los datos de los sujetos de prueba. Con esto se puede hacer un análisis posterior ya sea por ausencia del paciente o para comparar tanto viejos como nuevos resultados.

Se tuvo que profundizar tanto en nociones de medicina, así como la amplificación de las señales cerebrales y su representación usando una comunicación inalámbrica, en este caso Bluetooth. Al hacer esto se tuvo que estudiar tanto el funcionamiento de los amplificadores de instrumentación, así como filtros que ayudarán a eliminar el ruido de la señal. Se profundizó también en la digitalización de las señales por medio del microcontrolador y su correcto envío a la computadora, con lo que se tuvo que aprender a usar la API Bluecove en Java para poder comunicarse. Ya al final se representaron los datos, graficándolo en 2 paneles por separado.

El sistema realizado es funcional solo faltaría ser evaluado por expertos en el área médica y comparar los resultados con uno comercial para validar los obtenidos.

Se plantea a futuro poder hacerlo más cómodo, haciéndolo ya sea más pequeño y con una fuente de alimentación interna (baterías) para tener un sistema completamente móvil. También se puede buscar otros tipos de comunicación inalámbrica como el Wifi, el cual es mejor para el trabajo de oficina y tiene una mayor velocidad de transmisión. En cuanto al software se podría modificarlo de tal forma que se pueda implementar en un dispositivo móvil la visualización de los electroencefalogramas, dando una mayor portabilidad y flexibilidad. También se buscará poderlo hacer con el microcontrolador de 32 bits como se tenía planeado previamente.

## Capítulo 6. Bibliografía

- [1] Definición de Electroencefalograma (EEG). *Enciclopedia de la salud*. Recuperado el día 27 de septiembre del 2012, de <http://www.enciclopediasalud.com/definiciones/electroencefalograma/>
- [2] André Zúquete, Bruno Quintela, João Paulo Silva Cunha (2010). *Biometric authentication using brain responses to visual stimuli*. Recuperado el día 20 de diciembre del 2012, de [http://www.researchgate.net/publication/221334402\\_Biometric\\_Authentication\\_using\\_Brain\\_Responses\\_to\\_Visual\\_Stimuli](http://www.researchgate.net/publication/221334402_Biometric_Authentication_using_Brain_Responses_to_Visual_Stimuli).
- [3] Claudio Lavin, René San Martín, Diego Bravo, Luis Contreras, Pablo Isla (2011). *Potenciales Cerebrales relacionados a FeedBack en el estudio del aprendizaje y la toma de decisiones económicas*. Recuperado el día 20 de diciembre del 2012, de <http://www.scielo.org.co/pdf/rlps/v43n3/v43n3a06.pdf>.
- [4] Wu Di, Chen Zihua, Feng Ruifang, Li Guangyu, Luan Tian (2010). *Study on Human Brain After Consuming Alcohol Based on EEG signal*. *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Volume:5)*. Recuperado el día 2 de enero del 2013, de [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5564084&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D5564084](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5564084&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5564084)
- [5] Pengjing, Yangbo, Zhaoxu, Liyonghong (2012). *Correlation Properties Applied Detrended Fluctuation Analysis Method for Cue-induced EEG in Drug Dependence*. *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on (Volume:1)*. Recuperado el día 2 de enero del 2013, de <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6187882&url=http%3A%2F>

%2Fieeexplore.ieee.org%2Fiel5%2F6187453%2F6187799%2F06187882.pdf%3Farnumber%3D6187882

[6] Joan F. Alonso, Miquel A. Mañanas, Sergio Romero, Jordi Riba, Manel J. Barbanoj, and Dirk Hoyer (2007). *Connectivity analysis of EEG under drug therapy. Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. Recuperado el día 2 de enero del 2013, de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4353768>*

[7] Naveen Verma, Ali Shoep, Jose Bohorquez, Joel Dawson, John Gutttag, Anantha P. Chandrakazan (2010). *A Micro-Power EEG Acquisition SoC With Integrated Feature Extraction Processor for a Chronic Seizure Detection System. Solid-State Circuits, IEEE Journal of (Volume:45 , Issue: 4 ). Recuperado el día 3 de enero del 2013, de <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5437484&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4%2F5437471%2F05437484.pdf%3Farnumber%3D5437484>*

[8] J. G. Webster (1998). *Medical Instrumentation: Application and Design, 3rd ed.* New York: Addison-Wesley.

[9] David Lewis (1988). *Poder Mental: El Plan Alfa.* Barcelona: Martinez Roca.

[10] Tipos de ondas cerebrales (s.f.). En *Megabrain*. Recuperado el día 27 de septiembre del 2012, de [http://www.megabrain.net/novedades/megabrain/ondas\\_cereb.htm](http://www.megabrain.net/novedades/megabrain/ondas_cereb.htm)

[11] Electroencefalógrafo (s.f.). En *Mimi*. Recuperado el día 1 de enero del 2013, de <http://es.mimi.hu/medicina/electroencefalografo.html>

[12] Compra. Especialidad EEG clínico (s.f.). En *Neuronic*. Recuperado el día 2 de febrero del 2013, de <http://www.neuronica.com/modulos/compra/eeg%20clinico.htm>

- [13] Electroencefalógrafo - Siemens - Mingograf EEG 10 (s.f.). *MedWoW. Plataforma de equipamientos médicos*. Recuperado el día 2 de febrero del 2013, de <http://es.medwow.com/used-eeg-unit/siemens/mingograf-eeg-10/661035305.item>
- [14] Electroencefalógrafo - Nihon Kohden - EEG-9200 (s.f.). *MedWoW. Plataforma de equipamientos médicos*. Recuperado el día 2 de febrero del 2013, de <http://es.medwow.com/used-eeg-unit/nihon-kohden/eeg-9200/928395352.item>
- [15] Electroencefalógrafo - Nihon Kohden - Neuropack MEB-9200 (s.f.). *MedWoW. Plataforma de equipamientos médicos*. Recuperado el día 2 de febrero del 2013, de <http://es.medwow.com/used-eeg-unit/nihon-kohden/neuropack-meb-9200/993862368.item>
- [16] Moisés Adrian Chávez Méndez, Luis Fernando Corona Trevizo, Juan Pablo de León García, Rogelio González Chávez, Julio Félix Hernández Ochoa y Edgar Zendejas Hernández (2004). *Proyecto EEG. Instituto Tecnológico de Chihuahua*. Recuperado el día 4 de enero del 2013, de [www.elet.itchihuahua.edu.mx/academia/jnevarez/Mecatronica II/Electroencefalografo EEG Reporte de proyecto ago dic 09.htm](http://www.elet.itchihuahua.edu.mx/academia/jnevarez/Mecatronica II/Electroencefalografo EEG Reporte de proyecto ago dic 09.htm)
- [17] Thomas J. Sullivan, Stephen R. Deiss, and Gert Cauwenberghs (2010). *A low-Noise, non-Contact EEG/ECG sensor. Biomedical Circuits and Systems Conference, 2007. BIOCAS 2007. IEEE*. Recuperado el día 4 de enero del 2013, de [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4463332&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D4463332](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4463332&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4463332)
- [18] Vincent Lee, Jennifer Monski, Winthrop Williams, Bharathwaj Muthuswamy, Tom Swiontek, Michel Maharbiz, Vivek Subramanian and Ferenc Kovac (2012). *A mixed-signal EEG interface circuit for use in first year electronics courses. Circuits*

*and Systems (ISCAS), 2012 IEEE International Symposium. Recuperado el día 4 de enero del 2013, de*

<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6271861&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6257548%2F6270389%2F06271861.pdf%3Farnumber%3D6271861>

[19] Molina Culqui Néstor Marcelo y Álvarez Rueda Robin (2011). *Sistema de monitoreo de las ondas cerebrales (electroencefalograma o EEG) presentes en el sueño: análisis de frecuencia y coherencia del EEG en ambos hemisferios. XXII Jornadas en Ingeniería Eléctrica y Electrónica. Recuperado el día 6 de octubre del 2012, de*

[http://bibdigital.epn.edu.ec/bitstream/15000/9938/1/SISTEMA%20DE%20MONITOREO%20DE%20LAS%20ONDAS%20CEREBRALES%20\(ELECTROENCEFALOGRAMA\).pdf](http://bibdigital.epn.edu.ec/bitstream/15000/9938/1/SISTEMA%20DE%20MONITOREO%20DE%20LAS%20ONDAS%20CEREBRALES%20(ELECTROENCEFALOGRAMA).pdf)

[20] Maksim Jenihhin, Maksim Gorev, Vadim Pesonen, Hiie Hinrikus, Maie Bachmann, Jaanus Lass (2011). *EEG Analyzer Prototype Based on FPGA. Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium. Recuperado el día 4 de enero del 2013, de*

[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6046588&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D6046588](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6046588&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6046588)

[21] Sistemas microcontrolados (s.f.). *Tutorial Básico. Recuperado el día 6 de enero, de* <http://r-luis.xbot.es/pic1/pic01.html>.

[22] José Mopín Poblet (1988). *Introducción a la Bioingeniería*. España: Boixareu Editores.

[23] Molina Culqui Néstor Marcelo y Álvarez Rueda Robin (2011). *Sistema de monitoreo de las ondas cerebrales (electroencefalograma o EEG) presentes en el sueño: análisis de frecuencia y coherencia del EEG en ambos hemisferios. XXII*

*Jornadas en Ingeniería Eléctrica y Electrónica. Recuperado el día 6 de octubre del 2012, de*

[http://bibdigital.epn.edu.ec/bitstream/15000/9938/1/SISTEMA%20DE%20MONITOREO%20DE%20LAS%20ONDAS%20CEREBRALES%20\(ELECTROENCEFALOGRAMA\).pdf](http://bibdigital.epn.edu.ec/bitstream/15000/9938/1/SISTEMA%20DE%20MONITOREO%20DE%20LAS%20ONDAS%20CEREBRALES%20(ELECTROENCEFALOGRAMA).pdf)

[24] Jose María Drake Moyano (2005). *Instrumentación electrónica de comunicaciones. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Recuperado el día 5 de octubre del 2012, de*  
[http://www.ctr.unican.es/asignaturas/instrumentacion\\_5\\_IT/IEC\\_1.pdf](http://www.ctr.unican.es/asignaturas/instrumentacion_5_IT/IEC_1.pdf).

[25] *Hoja de Especificaciones (Datasheet) del INA114. Recuperado el día 5 de octubre del 2012, de*  
<http://www.datasheetcatalog.org/datasheet/BurrBrown/mXsqwzu.pdf>

[26] Revisión de los tipos de filtros analógicos más comunes (2009). *Escola Tècnica Superior d' Enginyeria. Universitat de València. Recuperado el día 4 de enero del 2013 de* [http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema\\_2.\\_revisio\\_n\\_de\\_los\\_tipos\\_de\\_filtros\\_analogicos\\_mas\\_comunes.pdf](http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_2._revisio_n_de_los_tipos_de_filtros_analogicos_mas_comunes.pdf)

[27] Acerca de la Tecnología Bluetooth (s.f.). *Dissmo. Recuperado el día 9 de noviembre del 2012, de* <http://dissmo.globalbit.biz/acerca-de-la-tecnologia-bluetooth/>

[28] Como elegir un Bluetooth (s.f.). *Guías MercadoLibre. Recuperado el día 22 de octubre del 2012, de* <http://guia.mercadolibre.com.ar/elegir-un-bluetooth-6925-VGP>

[29] Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) (2005). *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific*

*Requirements*. Recuperado el día 13 de octubre del 2012, de <http://standards.ieee.org/about/get/802/802.15.html>

[30] Seguridad en Bluetooth (s.f.). *Bluehack*. Recuperado el 4 de enero del 2013, de <http://bluehack.elhacker.net/proyectos/bluesec/bluesec.html>

[31] HC Serial Bluetooth Products (s.f.). *User Instructional Manual*. Recuperado el 5 de noviembre del 2012, de <http://cxem.net/arduino/download/HC%20Serial%20Bluetooth%20Products%20201104.pdf>

[32] Alberto Gimeno Brieba (2004). *JSR-82: Bluetooth desde Java*. Recuperado el día 15 de noviembre del 2012 de <http://mami.uclm.es/j2me/J2ME/java-bluetooth.pdf>

[33] Detección de dispositivos móviles (s.f.). *Seguridad mobile*. Recuperado el día 12 de noviembre del 2012, de <http://www.seguridadmobile.com/bluetooth/especificacion-bluetooth/identificacion-bluetooth/deteccion-de-dispositivos.html>

[34] Class RemoteDevice (s.f.). *Bluecove documentation*. Recuperado el día 20 de noviembre del 2012, de <http://bluecove.org/bluecove/apidocs/javax/bluetooth/RemoteDevice.html>

[35] Class LocalDevice (s.f.). *Bluecove documentation*. Recuperado el día 20 de noviembre del 2012, de <http://bluecove.org/bluecove/apidocs/javax/bluetooth/LocalDevice.html>

[36] Class Discovery Agent (s.f.). *Bluecove documentation*. Recuperado el día 21 de noviembre del 2012, de <http://bluecove.org/bluecove/apidocs/javax/bluetooth/DiscoveryAgent.html>

[37] [http://www.bluetooth.org/foundry/assignnumb/document/service\\_discovery](http://www.bluetooth.org/foundry/assignnumb/document/service_discovery)

[38] Package java.awt. Summary (s.f.). *Java™ 2 SDK, Standard Edition Documentation*. Recuperado el día 21 de noviembre del 2012, de <http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/package-summary.html>

[39] Package java.awt. Class Panel (s.f.). *Java™ 2 SDK, Standard Edition Documentation*. Recuperado el día 21 de noviembre del 2012, de <http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/Panel.html>

[40] Package java.awt. Class Container (s.f.). *Java™ 2 SDK, Standard Edition Documentation*. Recuperado el día 22 de noviembre del 2012, de [http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/Container.html#paint\(java.awt.Graphics\)](http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/Container.html#paint(java.awt.Graphics))

[41] Package java.awt. Class Graphics (s.f.). *Java™ 2 SDK, Standard Edition Documentation*. Recuperado el día 22 de noviembre del 2012, de <http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/Graphics.html>

[42] Fast Fourier Transform (s.f.). *Wolfram MathWorld*. Recuperado el día 7 de febrero del 2013, de <http://mathworld.wolfram.com/FastFourierTransform.html>

[43] An application of Discrete Fast Fourier Transform algorithm (s.f.). *School of Engineering and Desing, Bridgeport University*. Recuperado el día 8 de febrero del 2013, de [http://www1bpt.bridgeport.edu/sed/projects/cs597/Summer\\_2002/kunhlee/index.html](http://www1bpt.bridgeport.edu/sed/projects/cs597/Summer_2002/kunhlee/index.html)

[44] Instrumentación Biomédica. Tema 5: Electroencefalografía (s.f.). *Universidad de Alcalá. Departamento de electrónica*. Recuperado el día 20 de marzo del 2013,

de

<http://www.bioingenieria.edu.ar/academica/catedras/bioingenieria2/archivos/apuntes/tema%205%20-%20electroencefalografia.pdf>

[45] Ernst Niedermeyer, Fernando Lopes da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Field*. Referencia recuperada de [http://en.wikipedia.org/wiki/10-20\\_system\\_\(EEG\)](http://en.wikipedia.org/wiki/10-20_system_(EEG))

[46] Julia Santin, Jaime Godoy, Loreto Ríos, Tomás Mesa, Luis Aranda (2003). *Incidencia de variantes normales de aspecto epileptiforme en electroencefalogramas normales. Revista Chilena de neuro-psiquiatría. Estudio prospectivo y revisión de la literatura. Recuperado el día 20 de marzo del 2013, de [http://www.scielo.cl/scielo.php?pid=S0717-92272003000400004&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0717-92272003000400004&script=sci_arttext)*