

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e Implementación de un *software* de video-monitoreo
en pacientes con epilepsia a través de electroencefalogramas
de alta densidad**

Trabajo de graduación presentado por Ramón Andrés Samayoa
Sagastume para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



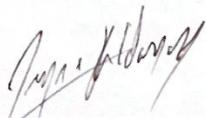
**Diseño e Implementación de un *software* de video-monitoreo
en pacientes con epilepsia a través de electroencefalogramas
de alta densidad**

Trabajo de graduación presentado por Ramón Andrés Samayoa
Sagastume para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

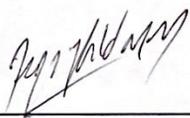
2019

Vo.Bo.:

(f) 

Ing. Dennis Moritz Aldana Moscoso

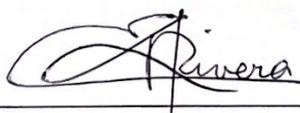
Tribunal Examinador:

(f) 

Ing. Dennis Moritz Aldana Moscoso

(f) 

Ing. Luis Pedro Montenegro Mejicanos

(f) 

Dr. Luis Alberto Rivera Estrada

Fecha de aprobación: Guatemala, 5 de diciembre de 2019.

Lista de figuras	VII
Lista de cuadros	IX
Resumen	XII
I. Introducción	1
II. Antecedentes	3
III. Justificación	5
IV. Objetivos	7
A. Objetivo general	7
B. Objetivos específicos	7
V. Alcance	9
VI. Marco teórico	11
A. Epilepsia	11
1. Definición	11
2. Signos y síntomas	11
B. Electroencefalografía	12
C. Oximetría de pulso	13
D. Frecuencia respiratoria	13
E. Python	14
1. PyQt	14
2. PyQtGraph	14
3. OpenCV	14
F. Paralelismo	15
1. Hilos	15
2. Procesos	15
3. Diferencias entre hilos y procesos	15

G. Filtros digitales	16
1. Tipos de filtros	16
H. <i>Open Source</i>	17
1. Tipos de licencias	17
VII. Selección de los sensores	19
A. Sensor de oximetría	20
B. Sensor de frecuencia respiratoria	21
VIII. Diseño de interfaz gráfica	23
A. Panel de configuración de señales	24
B. Panel de información del paciente	25
C. Panel de reproductor	25
D. Gráfica de señales vitales	26
E. Ventana de cámara	26
IX. Desarrollo del programa	27
A. Grabación de vídeo	28
B. Comunicación serial de sensores	28
C. Señales del electroencefalograma	29
1. Frecuencia de muestreo	29
2. Filtrado de las señales	29
D. Medio de transmisión de datos	30
E. Diagrama de flujo	31
X. Resultados	33
A. Señales de los sensores de signos vitales	34
B. Procesamiento y filtrado de electroencefalograma	35
C. Tiempo de ejecución	36
D. Velocidad de cámara	37
XI. Conclusiones	39
XII. Recomendaciones	41
XIII. Bibliografía	43
XIV. Anexos	45
A. Tablas con mediciones	46

1. Ejemplo de vídeo-EEG	12
2. Utilización de pulsioxímetro	13
3. Forma de los filtros ideales	16
4. Arduinos Nano Utilizados	19
5. Sensor CONTEC CMS50H	20
6. Clases de forma de onda según a la hendidura dícota	20
7. Función producida para simulación del sensor de oximetría	20
8. Sensor <i>Go Direct Respiration Belt</i> de Vernier	21
9. Señal de muestra del sensor de frecuencia respiratoria	21
10. Función producida para simulación del Sensor de Frecuencia Respiratoria	22
11. Diseño preliminar de la interfaz gráfica	23
12. Diseño final de la interfaz gráfica	24
13. Interfaz gráfica: Panel de configuración de señales	24
14. Interfaz gráfica: Panel de información del paciente	25
15. Interfaz gráfica: Panel de reproductor	25
16. Interfaz gráfica: Panel de gráficas de señales Vitales	26
17. Interfaz gráfica: Ventana con señal de vídeo de cámara	26
18. Primeras pruebas de software	27
19. Pruebas de señal de vídeo	28
20. Diagrama de flujo del funcionamiento	31
21. Última versión del <i>software</i>	33
22. Señal generada del sensor de oximetría	34
23. Señal generada del sensor de frecuencia respiratoria	34
24. Señal generada del sensor de oximetría	35
25. Señal generada del sensor de frecuencia respiratoria	35
26. Gráfica de cantidad de canales contra tiempo de promedio de operación	37
27. Comparación de velocidad entre implementaciones de cámara	37

Lista de cuadros

1. Diferencias entre hilos y procesos	15
2. Especificaciones de computadora utilizada	36
3. Medición de tiempos de operación	36
4. Primeras diez mediciones de tiempo con 8 canales	46
5. Primeras diez mediciones de tiempo con 16 canales	46
6. Primeras diez mediciones de tiempo con 32 canales	46
7. Primeras diez mediciones de tiempo con 64 canales	47
8. Primeras diez mediciones de tiempo con 128 canales	47
9. Primeras diez mediciones de tiempo con 256 canales	47

La electroencefalografía es un método para monitorear la actividad eléctrica de la corteza cerebral de manera no invasiva, y la electroencefalografía digital con el registro simultáneo de vídeo, ha dado paso al inestimable mejoramiento del diagnóstico de las crisis epilépticas, siendo según la literatura uno de los métodos más precisos para determinar la naturaleza epileptógena de la patología, y llevar a un manejo adecuado de la misma. A través del convenio del departamento de ingeniería Electrónica, Mecatrónica y Biomédica de la Universidad del Valle de Guatemala y HUMANA (Centro de Epilepsia y Neurocirugía Funcional) se planteó un proyecto con el objetivo de construir un EEG de alta densidad, en paralelo de un software para la lectura del mencionado.

El siguiente estudio se desarrolló a partir de la selección de sensores como herramientas para la medición de frecuencia respiratoria y oximetría, posteriormente se procedió al desarrollo del software a través de las librerías: PyQt, OpenCV y PyQtGraph y al diseño de la interfaz gráfica así como la implementación de *threads* para la mejora del rendimiento, y por ultimo la implementación de la cámara de vídeo, y graficación de las señales del electroencefalograma.

Obteniendo los siguientes resultados: las señales simuladas de los sensores de oximetría y frecuencia respiratoria, presentaron el comportamiento de las ondas se asemejan a las esperadas determinadas por las funciones matemáticas calculadas. En el *software*, las operaciones más demandantes de ejecutar fueron el procesamiento y el filtrado de las señales del electroencefalograma, absorbiendo la mayor cantidad del tiempo de ejecución, para que este segmento de instrucciones no impactaran de manera negativa en la fluidez del programa, se implementó un *thread* para realizar la tarea de manera concurrente. A medida que aumenta la cantidad de canales también lo hace el tiempo de operación obteniéndose una correlación lineal directamente proporcional entre las variables, cantidad de canales y el tiempo de operación promedio. Con la implementación de la cámara, se obtuvo una cantidad de 30 fotogramas por segundo al implementar *threads* se elimina el tiempo de espera en la obtención de un nuevo fotograma lográndose 519 fotogramas por segundo, obteniendo una mejora de 1730% en su rendimiento

Concluyendo lo siguiente: el medio de transmisión de datos establecido fue a través de USB, la herramienta necesaria para la medición de oximetría y frecuencia respiratoria es a

través de un sensor. Las librerías utilizadas fueron: PyQt, OpenCV y PyQtGraph, las cuales cumplen adecuadamente con las necesidades de velocidad y funcionamiento. Se evidencio que la utilización de *threads* en periféricos de entrada/salida como la cámara, incrementan el rendimiento de manera considerable.

La electroencefalografía es un método para monitorear la actividad eléctrica de la corteza cerebral de manera no invasiva, y la electroencefalografía digital con el registro simultáneo de vídeo, ha dado paso al inestimable mejoramiento del diagnóstico de las crisis epilépticas, siendo según la literatura uno de los métodos más precisos para determinar la naturaleza epileptógena de la patología, y consecuentemente llevar a un manejo adecuado de la misma.

En Guatemala actualmente existen alrededor de 325,000 habitantes con epilepsia, siendo una patología muy frecuente, que afecta a 2 de cada 100 habitantes, sin embargo los avances acerca del tratamiento y la mejora en la calidad de vida de las personas que lo padecen se encuentran actualmente limitados en nuestro país debido a que los recursos para su diagnóstico rara vez son accesibles para la población. El siguiente escrito pretende el desarrollo de las bases para la construcción de un software que permita el muestreo de las señales de un electroencefalograma de alta densidad incluyendo vídeo, asimetría y frecuencia respiratoria, y que este sea desarrollado a través de librerías totalmente gratuitas, con el propósito de brindar a la población guatemalteca a través de la asociación de la Universidad del Valle de Guatemala con HUMANA una atención de mayor calidad de forma mas accesible.

Antecedentes

En 2012 en el Hospital Pediátrico Docente "Juan Manuel Márquez", La Habana, Cuba, se realizó un estudio sobre la experiencia en el uso del video-electroencefalograma en el servicio de neurofisiología clínica, para la evaluación de los eventos paroxísticos neurológicos de naturaleza no precisada y/o en la epilepsia de difícil control en pacientes pediátricos, se estudiaron 121 niños divididos en 4 grupos: I. Determinar la naturaleza (epiléptica o no) del evento paroxístico; II. Confirmar el síndrome epiléptico; III. Cuantificar las crisis epilépticas y IV. Estudiar candidatos para cirugía de la epilepsia. La eficacia diagnóstica fue de un 86.77 %, y provocó modificaciones en el diagnóstico y/o el tratamiento inicial en un 32.38 %. Las modificaciones en el diagnóstico en el grupo I fueron en 16/43 pacientes. En el grupo II se clasificaron 47 síndromes epilépticos. Para el grupo III el número de crisis registradas fue $7,31 \pm 4,21$, y el tiempo de registro del video-electroencefalograma fue de $3,86 \pm 1,10$ horas. En el grupo IV se estudiaron 2 pacientes, y se confirmaron como candidatos para hemisferectomía funcional. Se concluyó que el monitoreo por video-electroencefalograma ratificó ser una herramienta de gran valor y utilidad en el diagnóstico de los eventos paroxísticos neurológicos en niños, lo que incide en establecer una adecuada conducta terapéutica y contribuye a disminuir ciertas limitaciones en la aplicación de los criterios de la Liga Internacional contra la Epilepsia [1].

En 2006 se realizó un estudio prospectivo, transversal, que incluye a 22 pacientes adultos con diagnóstico en el ambulatorio de Epilepsia de Lóbulo Temporal (ELT) de difícil control, sometidos a monitorización por video-EEG, donde los objetivos fueron correlacionar datos clínicos y electroencefalográficos obtenidos por la monitorización prolongada por video-EEG e inferir los grados de precisión del diagnóstico por la aplicación del modelo COPPE/Cosenza de jerarquía confusa; en dicho estudio se demostró que la monitorización prolongada por video-EEG estableció el diagnóstico clínico, sindrómico y topográfico en esos pacientes; la lógica fuzzy mostró definir mejor un valor matemático de la correspondencia entre el diagnóstico en el ambulatorio de ELT y la monitorización prolongada por video-EEG ya que el diagnóstico clínico se modificó en dos pacientes (9 %), el sindrómico en seis (27.2 %), y el topográfico en 16 (72.7 %). Los indicadores fuzzy de precisión del diagnóstico fueron: 0.91 para diagnóstico clínico, 0.74 para diagnóstico sindrómico y 0.36 para diagnós-

tico topográfico; se considera 1 el valor de máxima precisión de diagnóstico atribuyéndose a los resultados de la monitorización por vídeo-EEG [2].

En la actualidad, la epilepsia aun siendo una enfermedad poco entendida, constituye un problema a escala mundial, siendo un trastorno que afecta a individuos de todas las edades. Se estima que a nivel mundial alrededor de 50 millones de personas padecen epilepsia, y el 80 % de quienes lo padecen viven en países subdesarrollados, si bien estas personas viven en países de ingresos socioeconómicos bajos, no son diagnosticados adecuadamente, lo cual representa un problema crucial en el manejo y nivel de vida de estos individuos, pues el 70 % de las personas con epilepsia podrían vivir sin convulsiones si se diagnosticaran y trataran adecuadamente, causando un impacto transcendental en la vida de quien padece dicho trastorno³.

En Guatemala se encuentran aproximadamente 325,000 habitantes con epilepsia, afectando a 2 de cada 100 personas (HUMANA, 2019). Teniendo en cuenta la alta prevalencia de la epilepsia y que esta representa uno de los motivos de consultas más frecuentes, y siendo el nuestro un país con una carencia evidente en servicios de salud, es relevante el uso de métodos de diagnóstico factibles como el electroencefalograma de alta densidad (128 o 256 canales), en conjunto con un vídeo del paciente en un área controlada, lo cual generará un diagnóstico altamente preciso ya que correlaciona el estado clínico de la convulsión con las ondas en el electroencefalograma y a su vez con los signos vitales, siendo la construcción de un software en paralelo el primer paso para el inicio de una revolución en el diagnóstico y seguimiento de la epilepsia en Guatemala.

Se plantea el desarrollo de un software capaz de poder transmitir la información de un electroencefalograma de alta densidad, para que pueda ser utilizado como una herramienta de diagnóstico accesible en tiempo real y que cuente además con la emisión de signos vitales para monitoreo del paciente, y de esta manera poder esclarecer el futuro del diagnóstico en epilepsia en Guatemala y ofrecer a los pacientes una atención integral.

A. Objetivo general

Implementación de una aplicación de *software* de computadora para el muestreo de las señales de un electroencefalograma de alta densidad, incluyendo vídeo, oximetría y frecuencia respiratoria del paciente.

B. Objetivos específicos

- Establecer un medio de transmisión de datos para la transferencia de la información necesaria de un electroencefalograma de alta densidad.
- Determinar las herramientas adecuadas para la medición de oximetría y la transmisión de esta, como parte del monitoreo del paciente.
- Determinar las herramientas adecuadas para la medición de la frecuencia respiratoria y la transmisión de esta, como parte del monitoreo del paciente.
- Desarrollar el programa utilizando librerías totalmente gratis y *open source*, capaces de cumplir con las necesidades de velocidad y funcionamiento.
- Asegurar un rendimiento de tiempo real de los datos del electroencefalograma, así como los sensores de oximetría y frecuencia respiratoria, en conjunto con la señal de vídeo.

Este trabajo de investigación sentará las bases para el desarrollo de un software para el muestreo de señales de un electroencefalograma de alta densidad incluyendo señales de vídeo, oximetría y frecuencia respiratoria, un procedimiento esencial para pacientes farmacoresistentes en quienes se considera un abordaje quirúrgico como tratamiento.

Realizado en la Universidad del Valle de Guatemala en conjunto con HUMANA, definiendo un software inicial, a través de un medio de transmisión de datos, basado en fuentes de código abierto, que permitan la descripción visual de las señales de electroencefalografía, frecuencia respiratoria y oximetría. Asegurando una base para el funcionamiento y desarrollo de las herramientas de software integrado.

Esto sentará las bases para la innovación del diagnóstico de epilepsia en Guatemala dando paso a un diagnóstico más factible y más preciso de forma más accesible para la población

A. Epilepsia

1. Definición

Se ha debatido si la epilepsia debería considerarse como un trastorno o una enfermedad, pero la Liga Internacional contra la Epilepsia "ILAE" por sus siglas en inglés, y la Oficina Internacional para la Epilepsia por sus siglas en inglés "IBE" acordaron que es preferible tratarla como una enfermedad^[4]. La Organización Mundial de la Salud (OMS) define la epilepsia como una enfermedad crónica por diversa etiología y se caracteriza por la recurrencia de crisis neurológicas debidas a descargas eléctricas en neuronas cerebrales, también llamadas convulsiones^[3].

2. Signos y síntomas

Se considera que un paciente sufre de epilepsia cuando este presenta un mínimo de dos crisis no provocadas con más de 24 horas de separación, entendiéndose por crisis epiléptica a un episodio autolimitado, en el que se altera la función cerebral, causado por una actividad anormal y excesiva de un grupo de neuronas corticales, produciendo descargas eléctricas sincrónicas, teniendo como resultado diferentes tipos de espasmos corporales según el área cerebral que se vea afectada. Las crisis que se pueden presentar podrían afectar a solo una parte del cuerpo (parciales) o su totalidad (generales), además varían en su tiempo, tanto en la duración de la crisis como en la presentación entre ellas. Dependiendo del área del cerebro donde se inicia y la forma en que se propaga puede llegar a crear síntomas temporales, como pueden ser: pérdida del conocimiento o la conciencia, pérdida de los sentidos, cambio en el estado de ánimo u otras funciones cognitivas, o combinación entre ellas^[3].

B. Electroencefalografía

La electroencefalografía o EEG es un método para monitorear la actividad eléctrica de la corteza cerebral de manera no invasiva, el cual consiste en el fijamiento de electrodos sobre el cuero cabelludo y para detectan los impulsos eléctricos de las diferentes áreas del cerebro^[5].

El electroencefalograma digital con el registro simultáneo de vídeo, también llamado vídeo-EEG, ha dado paso al inestimable mejoramiento del diagnóstico de las crisis epilépticas^[6]. Porque nos proporciona mayor información del paciente al mostrar lo que le ocurre, siendo estas sesiones de duraciones que comprenden desde una alta cantidad de horas hasta incluso días. La alta densidad nos proporciona una mejora indiscutible en la precisión para la localización de la actividad eléctrica del cerebro, indicando la ubicación del foco epileptógeno. Este es un procedimiento esencial para los paciente que son farmacoresistentes que consideran la cirugía como un posible tratamiento^[7].

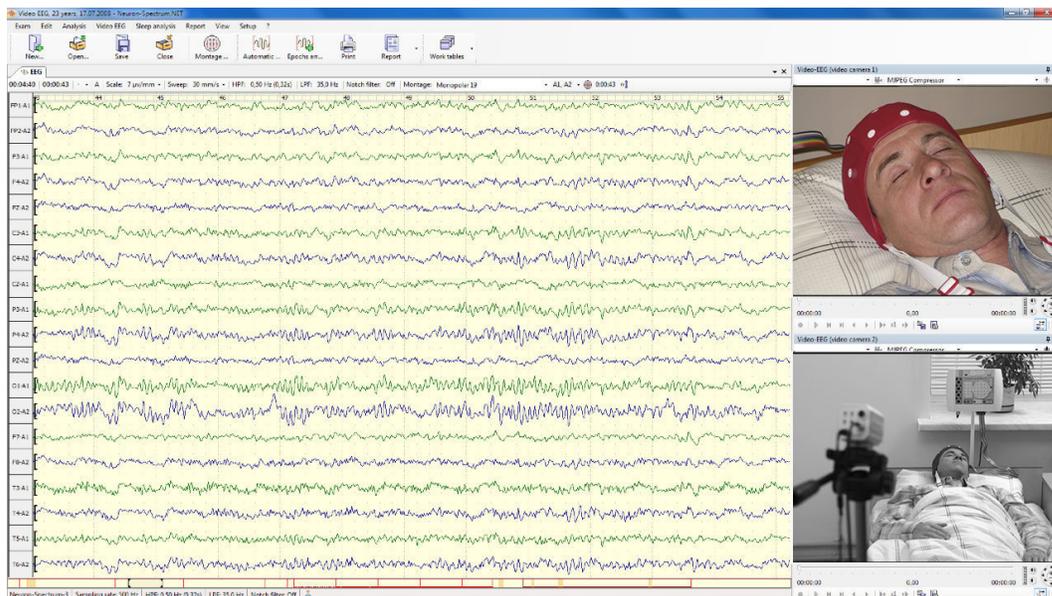


Figura 1: Ejemplo de vídeo-EEG

C. Oximetría de pulso

La oximetría de pulso, también llamada pulsioxímetro, es un método no invasivo que consiste en la medición de la saturación de oxígeno en la hemoglobina. Este instrumento se ha vuelto indispensable para la atención de pacientes.

El dispositivo utiliza la espectrofotometría, que es la utilización de dos diodos emisores de luz con diferentes longitudes de onda, normalmente uno de luz roja (640-660nm) y otro con luz infrarroja (910-940nm), atravesando el pulpejo del dedo hacia un fotodiodo detector.



Figura 2: Utilización de pulsioxímetro

El receptor mide la cantidad de luz roja e infrarroja que no fueron absorbidas por la desoxihemoglobina (hemoglobina reducida) y la oxihemoglobina (hemoglobina oxigenada), respectivamente, y el radio de absorción entre cada una de estas resulta en el grado de oxigenación de la hemoglobina^[8].

D. Frecuencia respiratoria

La frecuencia respiratoria es la cantidad de respiraciones que realiza una persona durante un minuto. Se puede medir de varias formas, una es visual observando las veces que el pecho se eleva, y otra, de manera palpada con la mano en el tórax y contabilizando la cantidad de respiraciones^[9].

Un método más vanguardista es la utilización de un sensor de fuerza en una correa ajustable que se coloca alrededor del pecho y con que mide la frecuencia respiratoria^[10].

E. Python

Python es un lenguaje de programación multiparadigma por su variada cantidad de usos, su sintaxis sencilla lo vuelve fácil de escribir como de leer [11]. Este lenguaje también es multiplataforma y se puede utilizar tanto en Windows, Linux, MacOS y otros sistemas basados en UNIX. Actualmente, después de 28 años, se encuentra en su tercera versión (3.7).

1. PyQt

PyQt es la adaptación del framework Qt, que permite el desarrollo de interfaces gráficas de usuario, en las versiones dos y tres del lenguaje de programación Python. Esta librería es soportada por todos los sistemas operativos que permiten el framework Qt, entre estos se encuentra Windows, OS X, Linux, iOS y Android [12].

2. PyQtGraph

Es una librería exclusiva de Python desarrollada para la creación de gráficas e interfaces de usuario. Dicha librería fue desarrollada en PyQt. Tiene implementaciones en el área de matemáticas, ciencias e ingeniería. PyQtGraph tiene como principal objetivo desarrollar gráficas de respuesta rápida y proveer las herramientas necesarias para desarrollo de aplicaciones gráficas [13].

3. OpenCV

OpenCV (Open Source Computer Vision Library) es una librería abierta creada en C++ especializada en visión computacional y aprendizaje de máquina. Esta librería fue construida en su mayoría para la fácil implementación de aplicaciones de visión artificial o el uso de productos comerciales, tales como cámaras. Las licencias con las que cuenta OpenCV permite a las empresas una fácil modificación del código para la implementación de sus productos. Con más de 2500 algoritmos optimizados, OpenCV permite clasificar acciones humanas en vídeo, rastrear movimientos, etc. OpenCV se utiliza en el mercado como una herramienta para el manejo de sistemas de cámaras, identificación de objetos, y distintos tipos de procesos de visión [14].

F. Paralelismo

El paralelismo a nivel de computadora se define como la forma en que se puede subdividir una tarea. Cada proceso es dividido en instrucciones secuenciales mediante hilos y subprocesos con el fin de mejorar la velocidad de respuesta^[15].

1. Hilos

Los hilos, o *threads* en inglés, son la unidad más básica de ejecución, que consta en una agrupación de tareas a realizar en un tiempo estimado. Una librería para Python se encuentra incluida en la Python *Standard Library* llamada *threading*^[16]. En Python, el interprete contiene una limitante llamada *Global Interpreter Lock*, el cual es un mecanismo que no permite la ejecución del código en múltiples núcleos de un procesador, por lo que se ejecuta en solo uno de ellos. Estos son recomendados para pequeños pedazos de código que esperan por una entrada, como lo son las operaciones de periféricos de entrada/salida^[17].

2. Procesos

Los procesos, o *Processes* en inglés, es una tarea que se le asigna a uno o múltiples procesadores y puede funcionar como un programa totalmente nuevo. Su utilización es recomendada para tareas intensas del procesador donde se realiza una gran cantidad de operaciones^[17]. Una librería para Python se encuentra incluida en la Python *Standard Library* llamada *multiprocessing*^[18].

3. Diferencias entre hilos y procesos

La principal diferencia entre ambos métodos es concurrencia contra paralelismo. la primera diferencia es en el caso de los threads, donde ocurre solo una instancia de código a la vez y varían entre ellos para trabajar de manera compartida. Y en la segunda, los procesos sí crean varias instancias de código las que se ejecutan de manera simultánea^[15].

Procesos	Hilos
No se comparte la memoria del programa	Se utiliza la memoria del programa principal
Requieren de más recursos del sistema	"Procesos ligeros", requieren menos recursos
Varios procesadores a la vez	Un procesador a la vez
Puede recaer en un Dead Lock	Es menos propenso a un Dead Lock

Cuadro 1: Diferencias entre hilos y procesos

G. Filtros digitales

Los filtros digitales son algoritmos que se implementan a través de hardware o software que se operan sobre una señal digital, discreta en el tiempo y cuantificada en amplitud, generando una nueva señal de salida ya filtrada. Todos estos filtros funcionan a través de valores numéricos de las muestras realizadas de las señales analógicas obtenidas por un conversor análogo-digital, o simplemente en valores ya almacenados en una computadora o microprocesador [19].

1. Tipos de filtros

Los tipos más comunes de filtros de frecuencias se encuentran en la siguiente lista, y sus formas ideales se observan en la figura 3 [20].

- **Paso bajo:** permite el paso de las frecuencias bajas y atenúa las altas.
- **Paso alto:** permite el paso de las frecuencias altas y atenúa las bajas.
- **Paso banda:** también llamado filtro pasabanda, se basa en la combinación entre un filtro paso bajo y paso alto, permitiendo el paso de un rango de frecuencias y atenuando lo que se encuentra fuera de este.
- **Banda eliminada:** también llamado en inglés filtro *notch*, elimina un rango de frecuencias o una en específica.

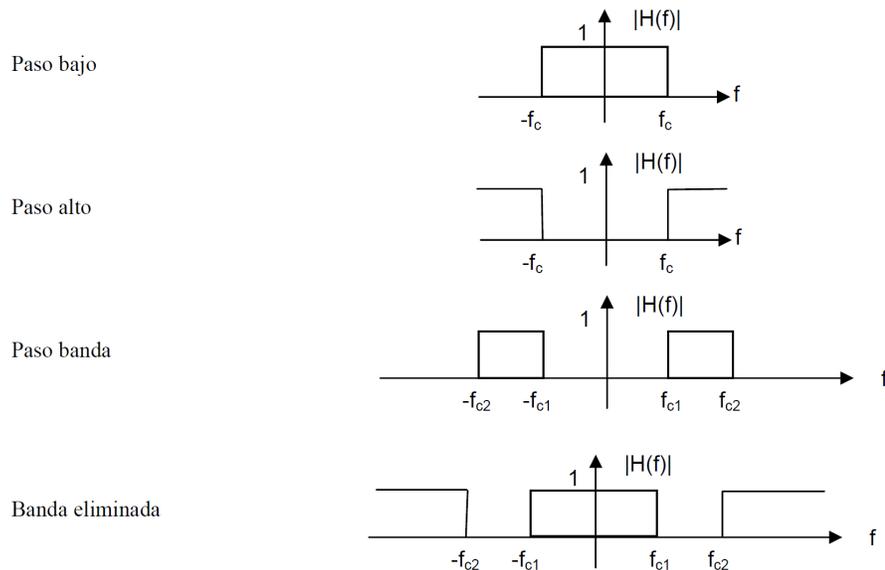


Figura 3: Forma de los filtros ideales

H. *Open Source*

El código abierto (*Open Source*) es el término que fue acuñado para hacer referencia a software que tiene licencia o términos de uso con el fin de tener acceso al código de programación para su utilización sin caer en un problema legal.

Dicha licencia, se centra en fines éticos y en beneficios de practicidad que se pueden obtener como individuo o en conjunto

El movimiento *Open Source* se enfoca más en el trabajo colaborativo con fin de llegar más lejos en una investigación. Esto quiere decir que, si se quiere realizar una investigación puede empezarse desde este punto de partida y obtener nuevos resultados y conocimientos. Su analogía más directa sería el de una patente libre^[21].

1. Tipos de licencias

- **Académica:** Muchos derechos de uso, pocas obligaciones de retribución.
- **CopyLeft:** Muchos derechos de uso, obligaciones derivadas de los descubrimientos obtenidos en el trabajo realizado.
- **Strong Copyleft:** Muchos derechos de uso, obligaciones con el primer autor con todas las atribuciones que se tengan con el trabajo obtenido^[21].

Las licencias más reconocidas y utilizadas en la comunidad de programación son las siguientes:

- GPL, o *Global Public Licenses*
- BSD, o *Berkeley Software Distribution*
- Licencia MIT

Selección de los sensores

Para la medición de la oximetría y la frecuencia cardiaca fue necesario el uso de un sensor, sin embargo no fue posible la adquisición del mismo por lo que se optó por simularlo a través de un microcontrolador. Se utilizó una Arduino Nano para la creación de las señales similares a las que pueden proveer este, y por medio de comunicación serial a través de USB se realizó la transmisión de las señales hacia la computadora.

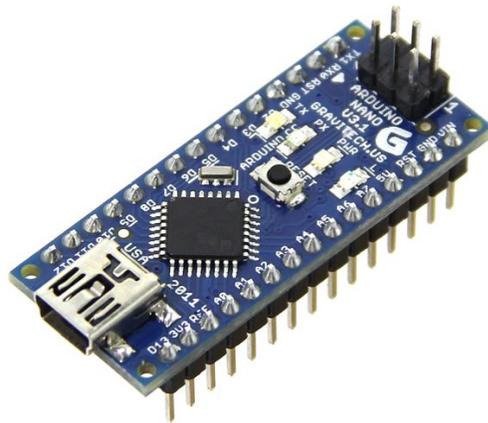


Figura 4: Arduinos Nano Utilizados

Para la simulación de los sensores se decidió realizar la frecuencia de muestreo de 250 Hz y una velocidad de 115,200 baudios/s, porque esta es la que utiliza la placa del programa *openBCI* para sus datos.

A. Sensor de oximetría

Se seleccionó el CONTEC CMS50H, por las siguientes funcionalidades: medición de la oxigenación de la sangre y el pulso cardiaco, conexión por USB, transmisión de datos en tiempo real y grabación de datos hasta por 24 horas continuas. Características son necesarias para el cumplimiento de las necesidades planteadas.



Figura 5: Sensor CONTEC CMS50H

Se considera normal un valor mayor al 95 % de oxígeno en la sangre [22], se seleccionó un valor de 98 % para su simulación.

Para la forma de onda que provee el sensor respecto al pulso, se utilizó aquella que se acerca mas a la normalidad, la cual corresponde a la clase I.

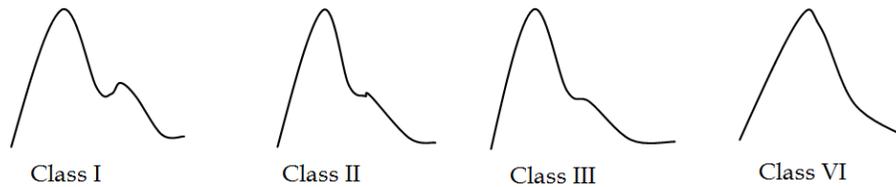


Figura 6: Clases de forma de onda según a la hendidura dicrota

al simular esta señal, se buscó una función matemática con una forma de onda similar a la seleccionada. Para visualizar se utilizó Desmos, la cual es una calculadora gráfica que se encuentra como aplicación web para la graficación de ecuaciones matemáticas. Determinando su período para calcular su frecuencia la cual resultaría siendo la frecuencia cardiaca.

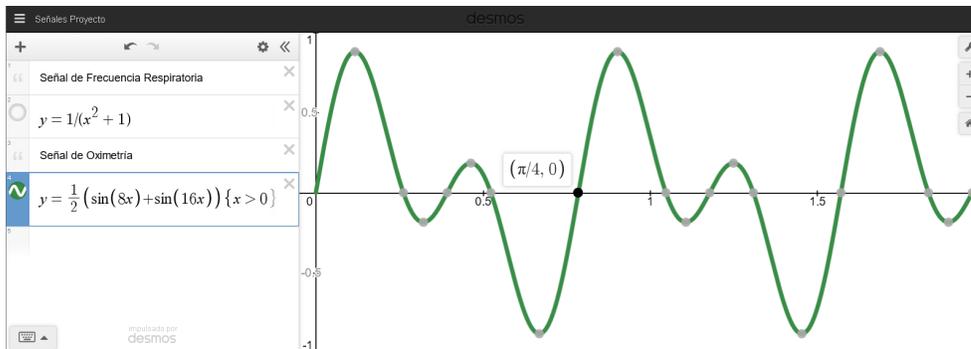


Figura 7: Función producida para simulación del sensor de oximetría

B. Sensor de frecuencia respiratoria

Se seleccionó el sensor *Go Direct Respiration Belt* para la medición de la frecuencia respiratoria, el cual mide los cambios de fuerza realizada por el abdomen el cual también puede utilizarse por USB o Bluetooth para la transmisión de datos con el computador.



Figura 8: Sensor *Go Direct Respiration Belt* de Vernier

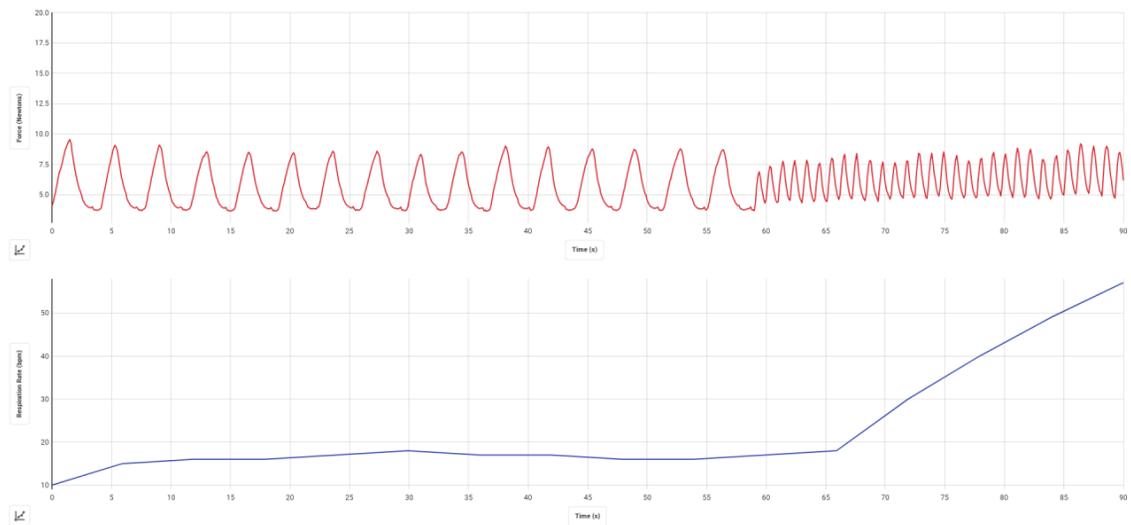


Figura 9: Señal de muestra del sensor de frecuencia respiratoria

Al igual que con el sensor anterior, se determinó una función matemática con una forma similar a la señal real del sensor. La frecuencia respiratoria tiene un valor normal entre 14 a 18 respiraciones por minuto^[23], se seleccionó la media igual a 15 respiraciones, resultando en un período de 4 segundos por cada respiración. El resultado final se graficó en Desmos y puede observarse en la Figura ^[10].

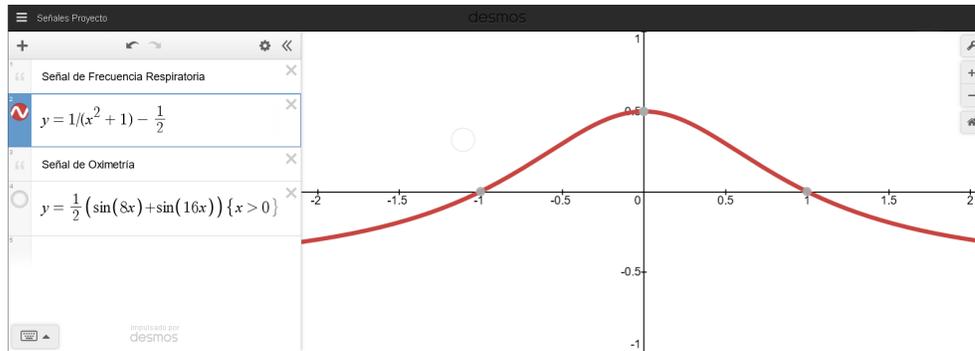


Figura 10: Función producida para simulación del Sensor de Frecuencia Respiratoria

Diseño de interfaz gráfica

Se utilizó la librería PyQt, esta proporciona una herramienta llamada QtDesigner que facilita la creación de la interfaz gráfica de usuario (GUI, siglas en inglés). El diseño preliminar se puede observar en la Figura 11.

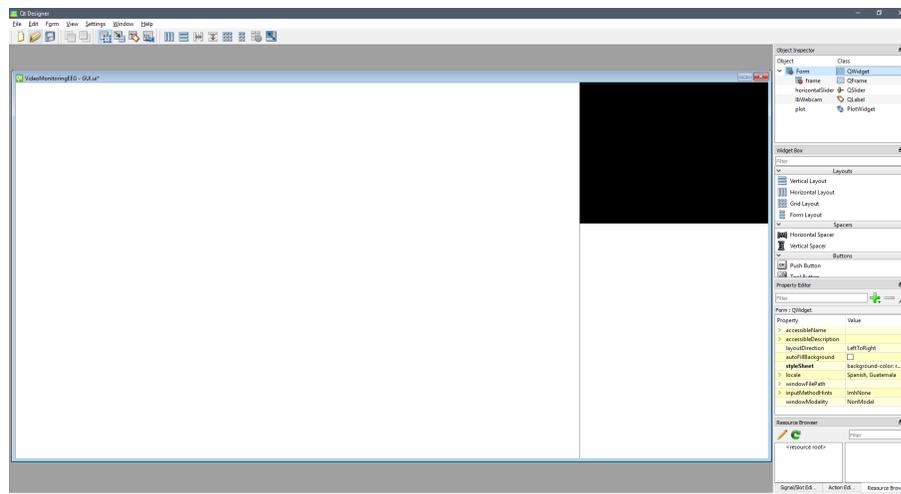


Figura 11: Diseño preliminar de la interfaz gráfica

Tomando en consideración la posibilidad de que médicos y otros usuarios hagan uso del software, se intentó generar la interfaz de la manera más clara posible y fácil de utilizar, incluyendo las opciones esperadas en un software de este tipo.

El resultado final de la interfaz gráfica realizada con la herramienta se puede observar en la Figura 12, en ella se ve su modo inicial donde el programa no se ha inicializado.

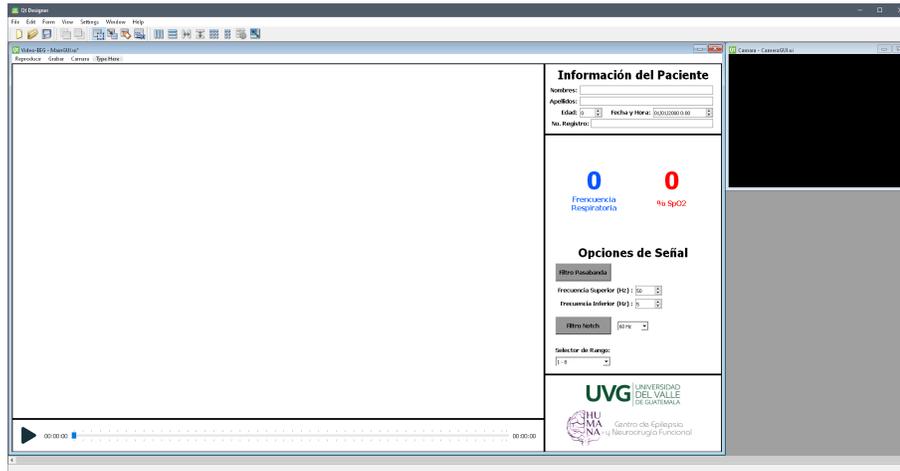


Figura 12: Diseño final de la interfaz gráfica

A. Panel de configuración de señales

En este panel se encuentran las opciones para la filtración y el manejo de las señales, además de las mediciones realizadas de la frecuencia respiratoria y el porcentaje de oxígeno en la sangre en una manera resaltada para facilitar el monitoreo. en la parte derecha de la pantalla se encuentran las opciones de filtrado, con el botón se puede activar o desactivar cada uno de ellos, con el pasabanda ambas frecuencias pueden ser modificadas de manera independiente y ser totalmente variables. Con el filtro *notch* solo se puede seleccionar entre las dos frecuencias más comunes de la red eléctrica, 50 o 60 Hz. Al final se encuentra un selector de rango en el cual hay 32 opciones, separando las 256 señales en segmentos de 8 cada una y así facilitar la navegación.



Figura 13: Interfaz gráfica: Panel de configuración de señales

B. Panel de información del paciente

En esta sección se incluye la información más inmediata del paciente, como lo son sus nombres, apellidos y su edad, el numero de registro que el centro de atención haya asignado al paciente. Con el objetivo de facilitar el control de sesiones del paciente, se agregó un campo para editar la fecha y hora.



Figura 14: Interfaz gráfica: Panel de información del paciente

C. Panel de reproductor

En este se encuentra un reproductor, en el cual se tiene la línea de tiempo con los valores del tiempo actual y el tiempo total, además de un botón tanto para la reproducción y pausa.



Figura 15: Interfaz gráfica: Panel de reproductor

D. Gráfica de señales vitales

Este es el panel de mayor tamaño, ya que es el más relevante de todos. En este se incluyen las 256 señales del electroencefalograma y las dos provenientes de los sensores de oximetría y frecuencia respiratoria.

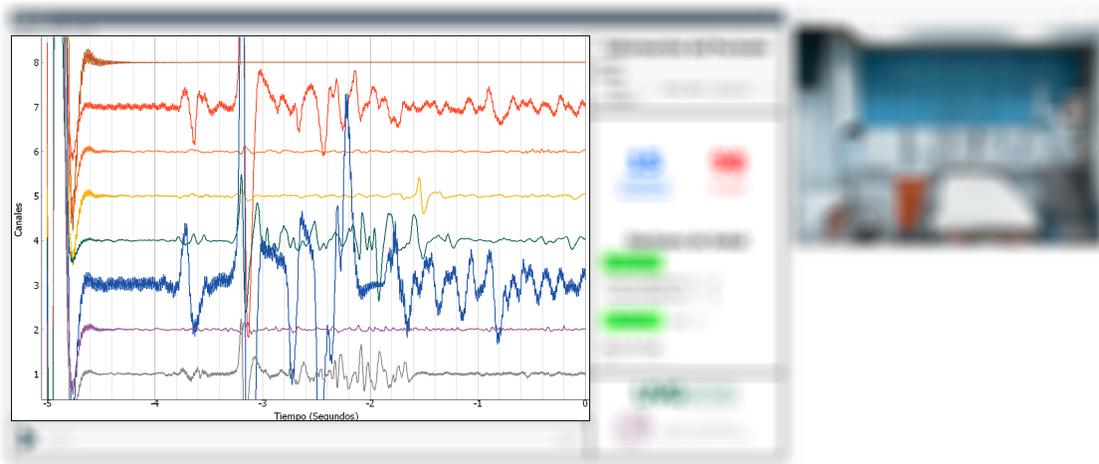


Figura 16: Interfaz gráfica: Panel de gráficas de señales Vitales

E. Ventana de cámara

La señal de vídeo se desplegó en una ventana por separado para permitir la manipulación del tamaño y su posición a conveniencia del usuario.



Figura 17: Interfaz gráfica: Ventana con señal de vídeo de cámara

Desarrollo del programa

El programa se inició con el objetivo de lograr un rendimiento cercano al tiempo real, para poder visualizar todas las señales fisiológicas de la mejor manera posible. El primer paso fue instalar las librerías que no incluye Python y que son de importancia para nuestro programa y así comenzar con la familiarización e implementación de estas.

- PyQt5, con licencia *General Public License*.
- PyQtGraph, con licencia MIT.
- OpenCV, con licencia *Berkeley Software Distribution*.

Las primeras pruebas se realizaron con una interfaz con ambas señales simuladas aunadas a la señal de vídeo, se evidencio que pese a ser tres señales únicamente, se presentaron problemas de rendimiento que repercutían en todo el programa.

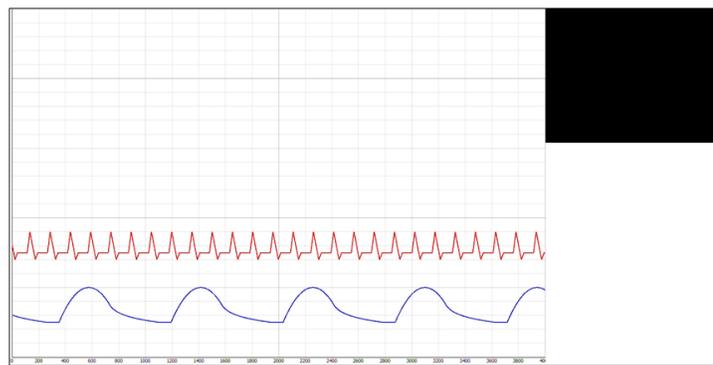


Figura 18: Primeras pruebas de software

A. Grabación de vídeo

La librería de OpenCV ofrecía mucha documentación, por lo que la implementación de esta no fue complicada, desde las primeras pruebas se pudo mostrar la señal de vídeo de la cámara. sin embargo al integrarlo al software se descubrió que la cámara tenía sus limitantes de velocidad al tener una tasa de fotogramas de 30 fotogramas por segundo, esto retenía todo el programa en la espera que la cámara tuviera un nuevo fotograma para mostrar.

Enseguida se pensó en dos posibles soluciones, la primera consistía en que al conocer la velocidad de la cámara solo permitir su actualización cada debido tiempo y la segunda, la implementación de hilos para la obtención de fotogramas de la cámara de manera independiente del principal. Ambas opciones fueron utilizadas, pero se optó por la segunda opción debido a que resolvía la problemática y también aportaba una verdadera mejora en el rendimiento de la cámara de 300 %.

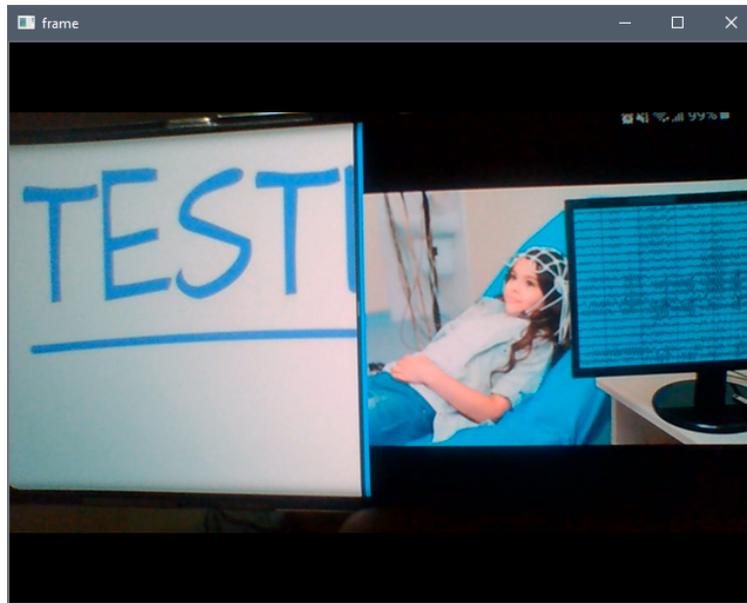


Figura 19: Pruebas de señal de vídeo

B. Comunicación serial de sensores

Las señales de oximetría y las de frecuencia respiratoria son enviadas al sistema a través de comunicación serial por USB desde ambos Arduinos al computador. Para asegurar un mejor rendimiento y calidad en la transferencia de datos se realizó la misma implementación de hilos.

C. Señales del electroencefalograma

Ya que el casco de alta densidad también se encuentra en proceso de investigación y desarrollo, se utilizó uno con el que ya contaba el Departamento de Electrónica, Mecatrónica y Biomédica de la universidad. La desventaja de este casco es que solo cuenta con ocho canales, por lo que se realizó una captura con el software de openBCI. Luego de obtener el archivo grabado se extrajeron los datos de los ocho canales y el registro de tiempo hacia un archivo CSV (Comma Separated Values), generando los demás canales a partir de una copia de los originales.

El resultado fue un archivo con 21,270 filas y 256 columnas, obteniendo así un total de 5,445,120 para un tiempo total de 84.51 segundos de grabación. Determinando que son 64,400 puntos por segundo los que deben ser correctamente procesados y filtrados para su graficación, cuando originalmente eran 2,015 puntos.

Esta cantidad de datos ralentizó el programa, dejando de ser funcional, por lo que se buscó la manera de mejorar el rendimiento. Se pensó utilizar paralelismo para mejorar el rendimiento y por lo tanto se implementó un hilo el cual solamente procesara los datos del electroencefalograma. La actualización de la gráfica se realizó en una función.

1. Frecuencia de muestreo

La frecuencia de muestreo de los datos tomados es de 250 Hz, siendo un dato relevante ya que es en base a esta frecuencia que se crean los filtros para las señales, y se determinan las velocidades de las gráficas en software.

2. Filtrado de las señales

En electroencefalografía normalmente se utilizan dos tipos de filtros, *notch* y pasabanda. El primero se utiliza para eliminar el posible ruido generado por la frecuencia de la red eléctrica, en Guatemala esta frecuencia es de 60 Hz por lo que esta se encuentra por defecto. El segundo se utiliza más para el análisis de las señales, este puede ser más variable, pudiendo asignar tanto la frecuencia inferior (por defecto 5 Hz) y la frecuencia superior (por defecto 50 Hz) para su filtrado.

D. Medio de transmisión de datos

Para el medio de transmisión de datos entre el electroencefalograma y la computadora, se seleccionó USB como la interfaz, debido a que en HUMANA los datos viajan de una habitación a otra y esto hace que la transmisión inalámbrica no sea posible. En la versión 2.0 de USB la velocidad de transmisión es de 480 Mbps, mientras que a través de Ethernet la velocidad es de 100 Mbps [24].

Para el protocolo de comunicación se propone un formato de datos que contenga un byte de inicio. Seguido de este estarían los datos de los canales del electroencefalograma, en tipo de dato entero de 32 bits, siendo un total de cuatro bytes por canal, resultando en un total de 1024 bytes. Terminando con un byte de fin para finalizar la comunicación.

La cantidad total sería igual a 1026 bytes, u 8208 bits, y con una frecuencia de muestreo de 250 Hz resultaría en una tasa de transmisión de datos de 2,052,000 bits por segundo o 2.052 Mbps.

E. Diagrama de flujo

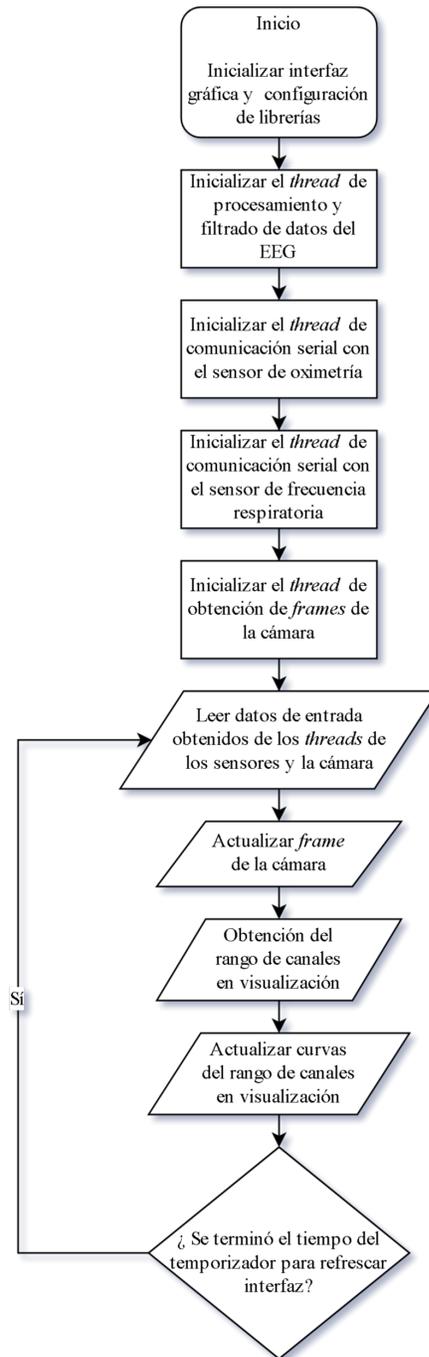


Figura 20: Diagrama de flujo del funcionamiento

Resultados

En la Figura 21 se puede observar la última versión del *software* desarrollado.

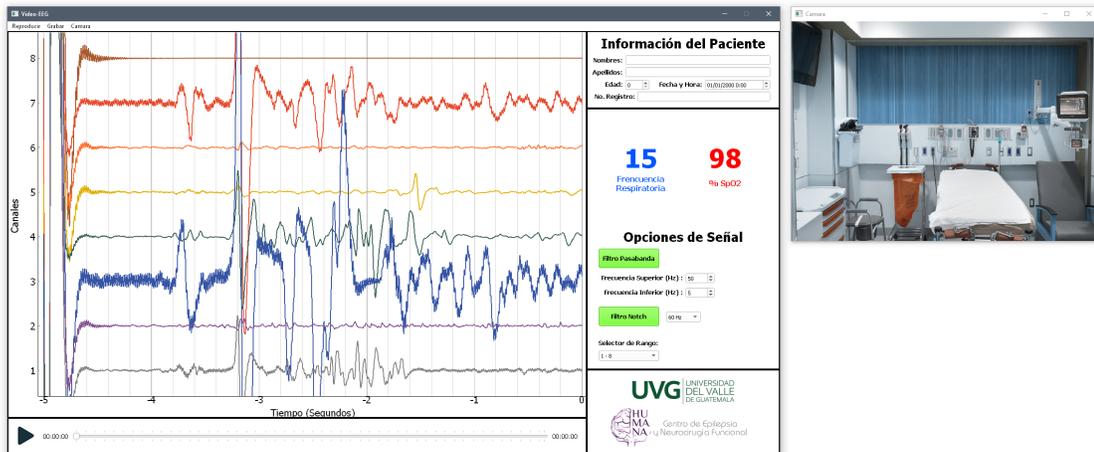


Figura 21: Última versión del *software*

A. Señales de los sensores de signos vitales

Las señales simuladas de los sensores de oximetría y frecuencia respiratoria se pueden observar en la Figura 22 y la Figura 23. Estas capturas fueron obtenidas en la última versión del *software*. En ambos resultados, el comportamiento de las ondas se asemejan a las esperadas determinadas por las funciones matemáticas previamente calculadas.

Gracias a la implementación de *threads* en la comunicación serial de los sensores, se evitó el tiempo de espera en la lectura del puerto en el proceso principal del programa, obteniendo así, una fluidez en la obtención de los datos y el programa en general.

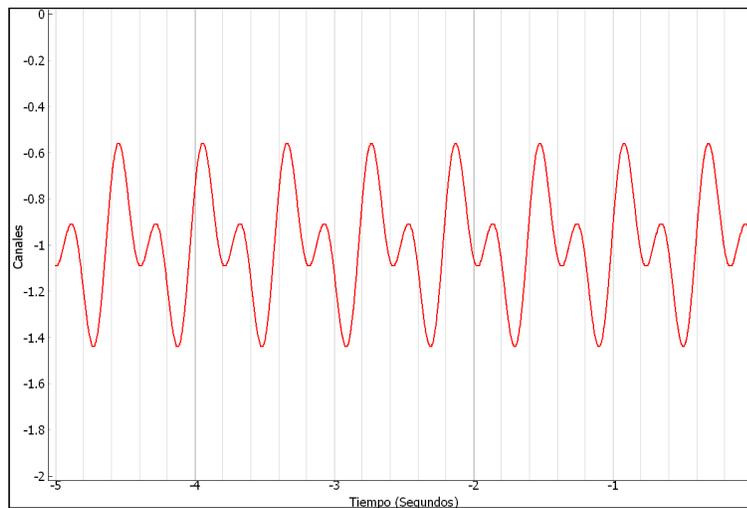


Figura 22: Señal generada del sensor de oximetría

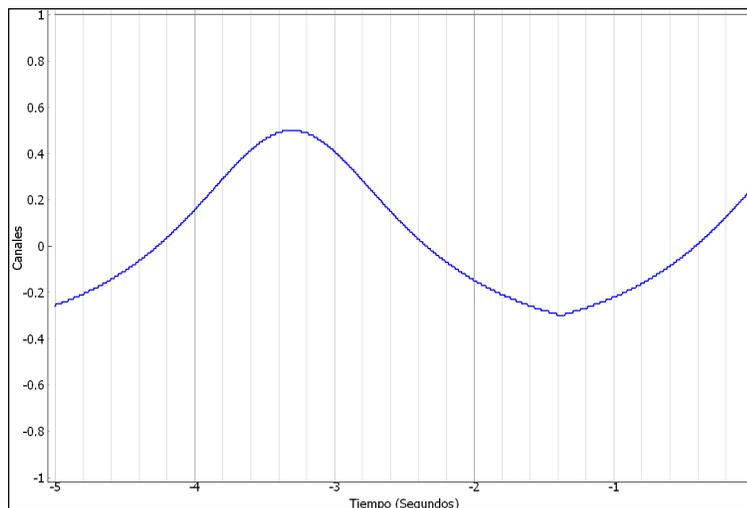


Figura 23: Señal generada del sensor de frecuencia respiratoria

B. Procesamiento y filtrado de electroencefalograma

Se aplicaron los filtros *notch* y pasabanda por defecto del programa, esperando la obtención de las formas de las señales similares a las de la Figura 25, siendo esta nuestra referencia. Las señales obtenidas por el *software* desarrollado se observan en la Figura 24. Comparando ambas figuras, se pueden apreciar los mismos comportamientos en las señales, al igual que amplitudes similares entre sí.

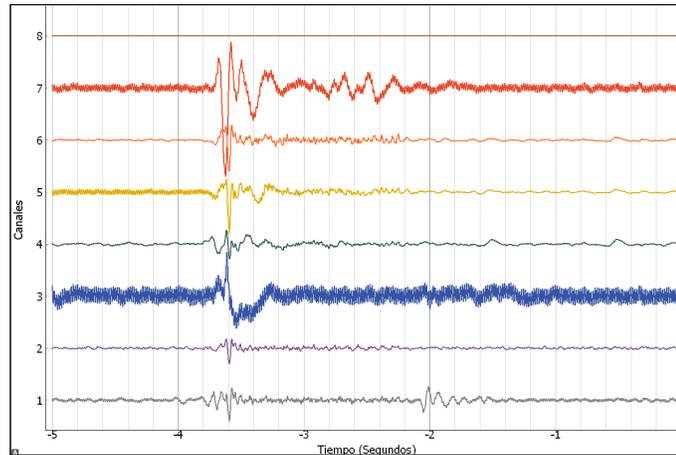


Figura 24: Señal generada del sensor de oximetría

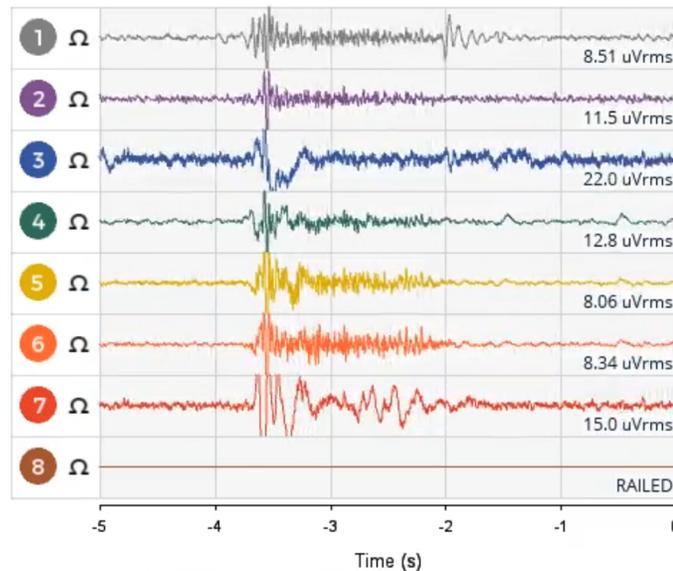


Figura 25: Señal generada del sensor de frecuencia respiratoria

Es importante mencionar que el orden en el que se aplican estos filtros es importante, las primeras pruebas se realizaron únicamente con el filtro pasabanda. Al evidenciar que los resultados no eran similares a los previamente grabados, se aplicó el filtro *notch* pero luego del filtro pasabanda. Obteniendo de nuevo resultados diferentes a los esperados, al realizarse el cambio en el orden de filtrado, es decir aplicando primero el filtro *notch* y posteriormente el filtro pasabanda, finalmente se obtuvieron los resultados deseados.

C. Tiempo de ejecución

Todos los resultados obtenidos fueron a través de una computadora portátil con las especificaciones que se observan en el Cuadro 2.

Modelo CPU	Núcleos CPU	Frecuencia Base	Frecuencia Máxima	Memoria RAM
Intel i7-7700HQ	4	2.80 GHz	3.8 GHz	16.0 GB

Cuadro 2: Especificaciones de computadora utilizada

En el *software*, las operaciones más demandantes de ejecutar fueron el procesamiento y el filtrado de las señales del electroencefalograma, absorbiendo la mayor cantidad del tiempo de ejecución. Para que este segmento de instrucciones no impactaran de manera negativa en la fluidez del programa, se implementó un *thread* para realizar la tarea de manera concurrente.

Como se puede evidenciar en el Cuadro 3, a medida que aumenta la cantidad de canales también lo hace el tiempo de operación (procesado y filtrado de todos los canales seleccionados).

La obtención de los valores promedio evidenciados en el Cuadro 3, fueron calculados a través de una muestra de 1,250, correspondiendo a un valor por encima del 5% de un total de 21,270 puntos.

Cantidad de Canales	Tiempo promedio de operación (s)	Uso promedio de CPU (%)
8	0.0191	15.50
16	0.0380	15.26
32	0.0757	15.05
64	0.1611	15.23
128	0.3096	15.57
256	0.5966	15.27

Cuadro 3: Medición de tiempos de operación

Se evidenció, que el uso del procesador se mantuvo constante, demostrando así las limitantes de los *threads* en Python, ya que aún siendo una tarea tan pesada para el procesador, no fue posible aprovechar todo el potencial de la computadora.

La relación entre las variables, cantidad de canales y el tiempo de operación promedio, puede apreciarse en la Figura 26. Observándose una correlación lineal directamente proporcional.

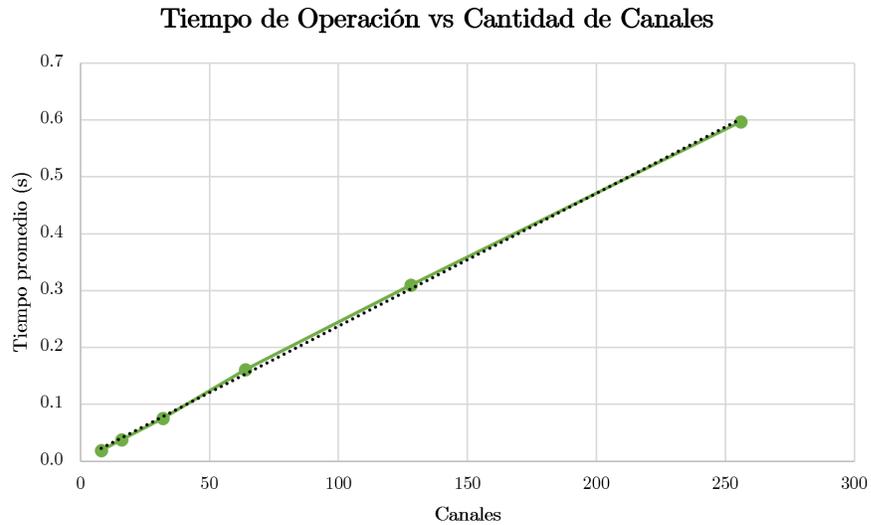
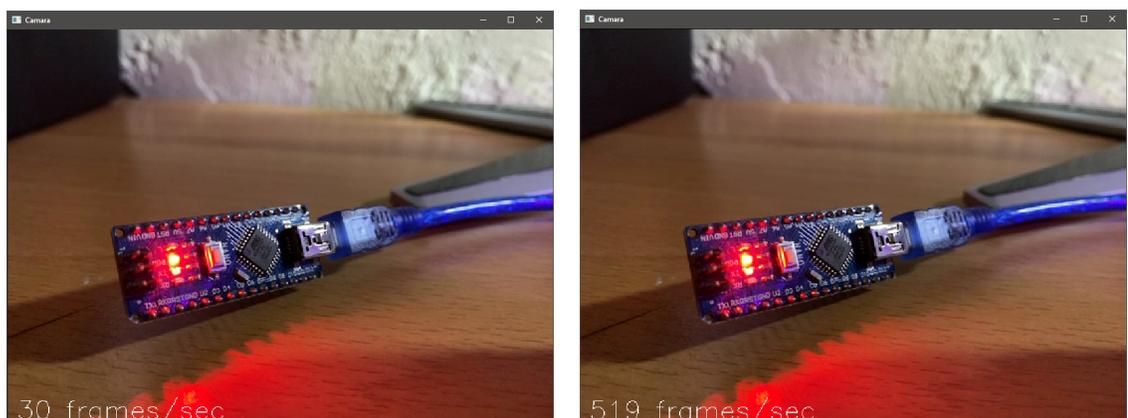


Figura 26: Gráfica de cantidad de canales contra tiempo de promedio de operación

D. Velocidad de cámara

Con la implementación de la cámara, se obtiene una cantidad de 30 fotogramas por segundo. Este resultado puede observar en la Figura 27a.

Al implementar *threads* se elimina el tiempo de espera en la obtención de un nuevo fotograma, lo cual comprendía una limitante en la ejecución del *software*. Implementándola sin ninguna otra instrucción en ejecución, se logró un resultado de 519 fotogramas por segundo, obteniendo una mejora de 1730 % en su rendimiento. Este resultado puede observar en la Figura 27b.



(a) Velocidad normal de cámara

(b) Velocidad de cámara con thread

Figura 27: Comparación de velocidad entre implementaciones de cámara

- El mejor medio de transmisión de datos se estableció a través de USB, por ser 480 % más rápida.
- Un microcontrolador es capaz de simular un sensor ideal de oximetría.
- Un microcontrolador es capaz de simular un sensor ideal de frecuencia respiratoria.
- Las librerías utilizadas PyQt, PyQtGraph y OpenCV cumplen con las licencias de *open source* GPL, MIT y BSD.
- La utilización de *threads* en periféricos de entrada/salida como la cámara, incrementando el rendimiento en un 1730 %.

- Se recomienda la futura implementación de la librería *multiprocessing* de Python para mejorar el rendimiento, así evitando las limitantes de *Global Interpreter Lock* y alcanzar velocidades mucho mayores en el programa.
- Investigar las diferencias en el manejo de recursos entre los múltiples sistemas operativos, como podrían ser: dificultades, beneficios, compatibilidad y limitantes.
- Procesamiento y filtrado de las señales del electroencefalograma en base de hardware a través del microcontrolador para la comunicación entre el casco con electrodos y el computador.
- Se recomienda la adquisición de sensores para la medición de oximetría y frecuencia respiratoria.

-
- [1] L. Aguilar Fabr , R. F. Rodr guez Vald s, R. M. Morgade Fonte, I. Valdivia  lvarez, R. Garc a Garc a y E. Bonet Quesada, “Experiencia en el uso del video-electroencefalograma en el servicio de neurofisiolog a cl nica”, es, *Revista Cubana de Pediatr a*, vol. 84, p gs. 234-243, sep. de 2012, ISSN: 0034-7531. direcci n: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0034-75312012000300004&nrm=iso.
- [2] M. E. Cosenza-Andraus, C. Nunes-Cosenza, R. Gomes-Nunes, R. Fantezia-Andraus y S. Alves-Leon, “Monitorizaci n prolongada por videoelectroencefalograf a de pacientes con diagn stico ambulatorio de epilepsia del l bulo temporal de dif cil control: aplicaci n del modelo de l gica fuzzy”, *Revista de neurolog a*, vol. 43, n.  1, p gs. 7-14, 2006.
- [3] *Epilepsia*, [en l nea; accedido 29. May 2019]. direcci n: <https://www.who.int/es/news-room/fact-sheets/detail/epilepsy>.
- [4] R. S. Fisher, C. Acevedo, A. Arzimanoglou, A. Bogacz, J. Helen Cross, C. E. Elger y S. Wiebe, “Definici n cl nica pr ctica de la epilepsia”, *Epilepsia*, vol. 55, n.  4, p gs. 475-482, 2014.
- [5] *Electroencefalograf a (EEG) - Mayo Clinic*, [en l nea; accedido 29. May 2019]. direcci n: <https://www.mayoclinic.org/es-es/tests-procedures/eeg/about/pac-20393875>.
- [6] F. Villanueva-G mez, “Descripci n v deo-EEG de las crisis epil pticas”, *RevNeurol*, vol. 30, n.  S1, p g. 0004,
- [7] A. Russi. (31 de oct. de 2017). V deo electroencefalograma (VEEG) de alta densidad, Viviendo con epilepsia, direcci n: <https://epilepsiarussi.com/blog/video-electroencefalograma-veeg-de-alta-densidad/> (visitado 30-09-2019).
- [8] G. P. L pez-Herranz, “Oximetr a de pulso: A la vanguardia en la monitorizaci n no invasiva de la oxigenaci n”, *Revista M dica del Hospital General de M xico*, vol. 66, n.  3, p gs. 160-169, 2003.

- [9] *Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure) - Health Encyclopedia - University of Rochester Medical Center*, [en línea; accedido 29. May 2019]. dirección: <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963>.
- [10] *Go Direct Respiration Belt | Vernier*, [en línea; accedido 29. May 2019]. dirección: <https://www.vernier.com/products/sensors/respiration-monitors/gdx-rb/#section4>.
- [11] M. Summerfield, *Programming in Python 3: A Complete Introduction to the Python Language*. Pearson Education, 2008, ISBN: 9780321606594.
- [12] *Riverbank | Software | PyQt | What is PyQt?*, [en línea; accedido 29. May 2019]. dirección: <https://riverbankcomputing.com/software/pyqt/intro>.
- [13] *Introduction — pyqtgraph 0.10.0 documentation*, [en línea; accedido 29. May 2019]. dirección: <http://www.pyqtgraph.org/documentation/introduction.html>.
- [14] *About*, [en línea; accedido 29. May 2019]. dirección: <https://opencv.org/about>.
- [15] J. Anderson. (14 de ene. de 2019). Speed up your python program with concurrency – real python, dirección: <https://realpython.com/python-concurrency/> (visitado 17-09-2019).
- [16] P. S. Foundation. (16 de sep. de 2019). threading — Thread-based parallelism — Python 3.7.4 documentation, dirección: <https://docs.python.org/3/library/threading.html> (visitado 17-09-2019).
- [17] G.-B. I. Ivanov. (20 de abr. de 2017). Introduction to Parallel and Concurrent Programming in Python, dirección: <https://code.tutsplus.com/articles/introduction-to-parallel-and-concurrent-programming-in-python--cms-28612> (visitado 17-09-2019).
- [18] P. S. Foundation. (16 de sep. de 2019). multiprocessing — Process-based parallelism — Python 3.7.4 documentation, dirección: <https://docs.python.org/3/library/multiprocessing.html> (visitado 17-09-2019).
- [19] S. W. Smith. (2011). Filter Basics, Chapter 14: Introduction to Digital Filters, dirección: <https://www.dspguide.com/ch14/1.htm> (visitado 30-09-2019).
- [20] A. Moreno, *Filtros*, jun. de 2009. dirección: https://ocw.upc.edu/sites/all/modules/ocw/estadistiques/download.php?file=11480/2011/1/52847/tema3._filtros_v2.3-2742.pdf (visitado 30-09-2019).
- [21] T. A. Alspaugh, H. U. Asuncion y W. Scacchi, “Software licenses, open source components, and open architectures”, pág. 23, 2012.
- [22] Organización Mundial de la Salud, *Manual de Oximetría de Pulso Global*, 2010. dirección: <http://www.lifebox.org/wp-content/uploads/WHO-Pulse-Oximetry-Training-Manual-Final-Spanish.pdf> (visitado 30-09-2019).
- [23] A. Dumas, *Lecciones de Semiología*, 7ma. Guatemala, 2012.
- [24] J. Claerr, “What Is the Difference Between USB & Ethernet?”, *Techwalla*, jul. de 2017. dirección: <https://www.techwalla.com/articles/what-is-the-difference-between-usb-ethernet>.

CAPÍTULO XIV

Anexos

A. Tablas con mediciones

8 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.019994259	14
1	0.020000219	10
2	0.020001173	26.7
3	0.019999981	12.5
4	0.020004034	11.1
5	0.023995876	12.5
6	0.023999453	12.5
7	0.0280056	31.2
8	0.020000696	12.5
9	0.022420168	25

Cuadro 4: Primeras diez mediciones de tiempo con 8 canales

16 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.037867546	13.4
1	0.046907187	20
2	0.050862789	15.6
3	0.045871019	15.8
4	0.047903776	13.8
5	0.046861887	16.7
6	0.048869133	20.8
7	0.037925482	28.6
8	0.036875248	11.1
9	0.037925243	11.1

Cuadro 5: Primeras diez mediciones de tiempo con 16 canales

32 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.081781387	13.4
1	0.084774017	14.3
2	0.092751741	12.8
3	0.097739697	14.3
4	0.089760303	17.4
5	0.09175396	16.7
6	0.082779884	21.4
7	0.080783606	31.8
8	0.083775759	11.4
9	0.079799175	15

Cuadro 6: Primeras diez mediciones de tiempo con 32 canales

64 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.171068668	14.1
1	0.174556494	14.4
2	0.176551819	19.3
3	0.145643711	15.1
4	0.148600101	14.6
5	0.149597168	12.5
6	0.171545506	16.9
7	0.21242547	16.4
8	0.227391958	19.3
9	0.21642065	12

Cuadro 7: Primeras diez mediciones de tiempo con 64 canales

128 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.344112158	14.1
1	0.323128462	14
2	0.31017518	15.4
3	0.307172537	12.7
4	0.308183908	16.2
5	0.329087257	18.9
6	0.325161457	25.6
7	0.306175232	12.8
8	0.308180332	13.5
9	0.302191734	13.8

Cuadro 8: Primeras diez mediciones de tiempo con 128 canales

256 Canales		
Muestra	Tiempo (s)	Uso de CPU (%)
0	0.646273136	13.5
1	0.58543396	14.5
2	0.590420723	16.4
3	0.581444502	16
4	0.589422464	15.3
5	0.57945013	14.5
6	0.601391554	13.5
7	0.583439112	14.4
8	0.586431026	16.8
9	0.593412638	16.9

Cuadro 9: Primeras diez mediciones de tiempo con 256 canales