



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM ECATEPEC

“DISEÑO Y DESARROLLO DE UN SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES ELECTROCARDIOGRÁFICAS MONITORIALES DESDE UN SISTEMA WEB”

T E S I S

Que para obtener el título de
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

ING. MARTÍN URIEL RAMÍREZ DOMÍNGUEZ

Tutor:

DR. EN C. RODOLFO ZOLÁ GARCÍA LOZANO

Tutores adjuntos:

**M. EN I.S.C. CUAUHTÉMOC HIDALGO CORTEZ
DR. ISMAEL DÍAZ RANGEL**



Ecatepec de Morelos, Estado de México, Diciembre 2020



Índice

Capítulo 1. Introducción	10
1.1 Objetivo general	10
1.2 Objetivos particulares	10
1.3 Actividades	11
1.4 Justificación	12
1.5 Descripción del capitulado	13
Capítulo 2. Estado del arte y marco teórico	14
2.1 Estado del arte	14
2.2 Anatomía y fisiología del corazón	15
2.3 El electrocardiograma	17
2.4 Las enfermedades del corazón, sus estadísticas globales y de México	19
2.4.1 Estadísticas globales	19
2.4.2 Estadísticas del país	20
2.5 Dispositivos analógicos para adquirir la señal eléctrica del corazón	22
2.5.1 Electrodo	23
2.5.2 Amplificador de instrumentación y filtro pasa-altas	24
2.6 Digitalización de señales	25
2.6.1 Señales analógicas	25
2.6.2 Señales digitales	26
2.6.3 Conversión analógico/ digital	27
2.7 Microcontroladores	29
2.8 Microcomputadoras	30





2.8.1 Orange pi	31
2.8.2 PcDuino	32
2.8.3 Raspberry pi	33
2.9 Protocolos de transferencia de datos en internet	35
2.9.1 Protocolo de control de transmisión (TCP)	35
2.9.2 Protocolo de transferencia de archivos (FTP)	36
2.9.3 Protocolo de transferencia de hipertexto (HTTP)	36
2.9.4 Secure Shell (SSH)	37
2.10 Servidores	38
2.10.1 Servidor Privado Virtual (VPS)	38
2.10.2 Servidores Dedicados	38
2.10.3 Comparativa entre servidores dedicados y VPS	38
2.10.4 Amazon Web Services (AWS)	39
2.11 Bases de datos	40
2.11.1 Sistema gestor de base de datos (SGBD)	40
2.12 Aplicación web	41
2.13 Metodología de desarrollo	42
2.13.1 Modelos de proceso evolutivo	42
2.13.2 El modelo Prototipo	42
Capítulo 3. Desarrollo experimental	44
3.1 Adquisición de las señales electrocardiográficas	46
3.1.1 Electroodos	46
3.1.2 Acondicionamiento de la señal	46





3.1.3 Conversión analógico-digital y transmisión serial.	47
3.2 Captura, almacenamiento y envío de datos	50
3.3 Servidor remoto	52
3.3.1 Montaje del servidor	52
3.3.2 Gestión de archivos y visualización	59
Capítulo 4. Pruebas y análisis de resultados	65
4.1 Primera iteración	65
4.2 Segunda iteración	67
4.2.1 Primer experimento: Tiempo de conversión analógico digital (ADC)	69
4.2.2 Segundo experimento: Tiempo de transmisión serial (Tx)	70
4.2.3 Tercer experimento: Tiempo de conversión analógico digital y transmisión serial (ADC-Tx)	72
4.3 Tercera iteración	76
4.4 Cuarta iteración	81
4.5 Quinta iteración	83
4.5 Sexta iteración	87
4.5 Séptima iteración	89
Conclusiones y trabajos futuros	97
Referencias	99
Referencias de figuras	103
Anexos	104
A) Código de algoritmo en Arduino	104
B) Código de algoritmo en Raspberry	105





C) Códigos de algoritmos en el servidor	108
Index.php	108
Función leer.php	117
Función directorios.php	117
Hoja de estilos style.css	118
D) Plan de negocios	125
E) Constancias	131
F) Memoria del congreso	137



Capítulo 1. Introducción

A nivel mundial y nacional, las enfermedades cardiacas son la principal causa de muerte, están por encima del cáncer, enfermedades respiratorias y diabetes. Uno de los principales instrumentos utilizados para el estudio de las enfermedades del corazón es el electrocardiógrafo, y al estudio resultante se le denomina electrocardiograma; que consiste en una representación gráfica de la actividad eléctrica del corazón, generalmente se realiza en papel milimétrico y debe ser interpretada por un especialista. En el presente trabajo se muestra el desarrollo de un sistema de adquisición de señales electrocardiográficas, con una tasa de muestreo que cumple con el criterio de Nyquist; las mediciones se almacenan de manera local en una microcomputadora, con la que se asegura que no se pierdan los paquetes de datos adquiridos si no se cuenta con conexión a internet o interrupciones eléctricas, una vez que se restablece la conexión se envían a un servidor remoto; el electrocardiograma puede ser visualizado a través de cualquier dispositivo con navegador web y conexión a internet, permitiendo que un especialista pueda visualizar de manera remota el electrocardiograma del paciente, esto con la finalidad de que no tenga que trasladarse hacia un hospital o laboratorio, permitiendo que el paciente o la institución de salud donde se atiende, no gasten tiempo y dinero en estancias.

1.1 Objetivo general

Desarrollar un sistema para la adquisición y procesamiento de señales electrocardiográficas, que puedan ser visualizadas remotamente mediante un sistema Web.

1.2 Objetivos particulares

- Realizar una frecuencia de muestreo que cumpla con el criterio de Nyquist respecto al ancho de banda de las señales eléctricas del corazón.



- Almacenar las mediciones en una microcomputadora de manera local en paquetes que contengan datos equivalentes a 30 minutos de mediciones, y que cuando haya conectividad, sean enviados a un servidor remoto específico.
- Definir y configurar un servidor remoto que reciba los datos de la etapa de adquisición.
- Diseñar un método para la gestión de los paquetes de datos recibidos.
- Presentar el electrocardiograma en una aplicación web accesible desde cualquier navegador conectado a internet.
- Implementar la propuesta sobre un hardware de bajo costo, con prestaciones suficientes para el manejo confiable del sistema.
- Utilizar un sistema operativo y software de desarrollo de uso libre.

1.3 Actividades

- Examinar la anatomía y fisiología del corazón.
- Estudiar los electrocardiogramas.
- Conocer las enfermedades del corazón, sus estadísticas globales y de México.
- Investigar dispositivos analógicos para adquirir la señal eléctrica del corazón.
- Estudiar el criterio de Nyquist para la adquisición de señales analógicas y entender el proceso de digitalización de señales.
- Desarrollar algoritmos para su aplicación en un microcontrolador con el propósito de digitalizar y transmitir resultados cumpliendo con el criterio de Nyquist.
- Considerar microcomputadoras que satisfagan los requerimientos del proyecto.
- Crear un algoritmo que guarde los datos adquiridos para subirlos al servidor cuando se tenga conexión a internet.
- Investigar protocolos de transmisión de datos vía internet.
- Configurar un servidor remoto.
- Diseñar y crear una base de datos.



- Crear un algoritmo que permita guardar los datos recibidos en el servidor en la base de datos.
- Enlazar la base de datos con una aplicación web.
- Implementar la graficación en una aplicación web.
- Diseñar un circuito impreso para la etapa de adquisición.
- Crear un modelo en 3D de una carcasa protectora.

1.4 Justificación

A nivel mundial las enfermedades no transmisibles (ENT) son las responsables de la muerte de 41 millones de personas cada año. Esto equivale al 71% de las muertes que se producen en el mundo. Este tipo de enfermedades, también son conocidas como enfermedades crónicas y tienden a ser de larga duración. Regularmente son el resultado de la combinación de factores genéticos, fisiológicos, ambientales y conductuales [1].

Dentro de las ENT, las enfermedades cardiovasculares son la principal causa de muertes a nivel mundial (17.9 millones cada año), seguidas del cáncer (9 millones), las enfermedades respiratorias (3.9 millones) y la diabetes (1.6 millones) [1]. Estos cuatro grupos de enfermedades son responsables de más del 80% de todas las muertes prematuras por ENT.

Similar a la tendencia mundial, en México las enfermedades cardiovasculares se colocan como la primera causa de muerte, seguida de la diabetes mellitus y tumores malignos, según el Instituto Nacional de Estadística y Geografía [2]. El 70 por ciento de los ataques cardíacos ocurren en casa y en lugares públicos, y pocos reciben ayuda en ese momento [3].

Para poder monitorear a las personas que tienen estos padecimientos, es necesario, o bien, que acudan a un hospital constantemente, o hacerse estudios rutinarios en laboratorios. Un inconveniente que presentan los laboratorios es que tardan hasta días



en entregar los resultados, aparte de esto, un médico especializado tiene que interpretar estas gráficas; y el tiempo de traslado a un hospital puede ser costoso, tedioso y hasta grave para el paciente.

Este proyecto presenta un sistema de adquisición de señales electrocardiográficas, que cumple con una frecuencia de muestreo adecuada al ancho de banda de las señales electrocardiográficas, ya que cumple con el criterio de Nyquist. Los datos adquiridos pueden ser visualizados de manera remota desde cualquier dispositivo con navegador web y conexión a internet, por lo que un especialista puede monitorear de manera remota al paciente, sin la necesidad de que este tenga que salir. Con la situación que se vive en la actualidad a causa de la pandemia, los médicos podrían dar un diagnóstico y los pacientes no tendrían que salir de sus casas para poder realizarse el estudio.

1.5 Descripción del capitulado

Capítulo 2. Se describe el marco teórico de la propuesta, donde se presenta: el estado del arte, anatomía y fisiología del corazón, el electrocardiograma, digitalización de señales, dispositivos utilizados, servidores, protocolos de transferencia, bases de datos y los programas utilizados.

Capítulo 3. Se menciona el funcionamiento del sistema con diagramas y algoritmos, se menciona la construcción de la propuesta, configuraciones de las herramientas utilizadas y la construcción del prototipo de sistema de adquisición.

Capítulo 4. Se habla sobre las pruebas realizadas, los prototipos creados, adjuntando las retroalimentaciones hechas a cada prototipo y se describen los resultados obtenidos.

Para finalizar se mencionan las conclusiones del proyecto, y se mencionan mejoras futuras que se pudieran implementar.



Capítulo 2. Estado del arte y marco teórico

2.1 Estado del arte

Existen trabajos en los que se proponen soluciones para la adquisición de señales electrocardiográficas y puedan ser monitoriables a distancia. Como en [4] se presenta un sistema de adquisición de señales electrocardiográficas, utilizando electrodos de 3 derivaciones, para adquirir las señales eléctricas, después pasa por el filtrado y acondicionamiento de la señal a través de la tarjeta AD8232, luego se digitaliza con un convertidor analógico-digital para pasar a una Raspberry pi 3, la cual se ocupa para poder enviar los datos a un servidor por el protocolo MQTT; con esto se logra visualizar a través de una aplicación en el celular y también en la computadora. Sin embargo, no se asegura que no se pierdan los datos al momento de transferir a la nube de MQTT, y si llegara a cortar la conexión a internet o hacia el servidor, estos datos se pudieran perder. Otro inconveniente que se aprecia es que, no se especifica la frecuencia de muestreo para poder alcanzar el ancho de banda de las frecuencias cardíacas.

En otros trabajos para solucionar la forma de guardar los datos y no se lleguen a perder, proponen guardarlos de manera local en memorias externas, tal es el caso de [5], donde se presenta un sistema que captura las señales electrocardiográficas con electrodos de 3 derivaciones, la tarjeta AD8232, el convertidor analógico-digital de una tarjeta de desarrollo Arduino y guardando los datos en una memoria micro SD; después pasan la memoria micro SD a una computadora en donde se sube a un gestor de archivos en la nube, de donde se puede descargar para poder visualizarlo con ayuda de Matlab. Se puede encontrar en el documento que no realizan la frecuencia de muestreo adecuada, ya que menosprecian el tiempo que tarda en guardarse los datos en la memoria micro SD. Otro inconveniente pudiera darse a la hora de subir los archivos desde la computadora, pues se necesitaría de una persona que gestione esa parte no pudiendo ser tan sistemático. También, el proyecto costaría más por la ausencia de software libre, ya que se tendría que pagar un costo adicional por la licencia de Matlab.





En [6] se tiene un sistema de adquisición que contiene electrodos no invasivos y la tarjeta eHealth, esta tarjeta contiene varios módulos de adquisición de señales biomédicas, pero en ese trabajo solo se enfoca en el módulo de señales electrocardiográficas. Para hacer la digitalización utilizan un Arduino y cuenta con un módulo bluetooth que manda los datos hacia un celular y a su vez a una computadora en donde se visualiza el electrocardiograma. Se encontró que en su código de Arduino utilizan la función *delay* para producir un retraso de 4 milisegundos, y lograr una frecuencia de muestreo de 250 Hz; sin embargo, no consideran que la digitalización y transmisión serial de datos también requieren de tiempo de ejecución por parte del microcontrolador; por lo tanto, la frecuencia de muestreo no cumplirá con el requerimiento de los 250 Hz necesarios para adquirir todas las componentes de frecuencia de las señales eléctricas del corazón.

2.2 Anatomía y fisiología del corazón

El corazón es un órgano hueco muscular que ejerce la función de bombear continuamente sangre a todos los órganos del cuerpo. Tiene el tamaño de un puño y pesa entre 200 y 425 gramos, se encuentra situado en el lado izquierdo del pecho, entre los pulmones y protegido por las costillas [7].

El corazón tiene cuatro cavidades, a las superiores se les denomina aurículas, y los inferiores ventrículos. La sangre pasa a través de una válvula antes de salir de cada cavidad del corazón. Las válvulas evitan el retroceso del flujo de sangre [8].

Las cavidades aurículas se comunican con los ventrículos a través de las válvulas auriculoventriculares, tricúspide en lado derecho y mitral en lado izquierdo. El corazón está separado en dos mitades, izquierda y derecha, por una pared vertical muscular llamada tabique, interventricular arriba e interventricular abajo (ver fig. 2.1).

Del corazón salen las arterias del organismo y a el corazón llegan las siguientes venas: (ver figura 2.1) [7]:

- A la aurícula derecha llegan las venas cavas superior e inferior.



- Al lado aurícula izquierda llegan las cuatro venas pulmonares.
- Del ventrículo derecho sale la arteria pulmonar, se divide en dos ramas dirigiéndose a ambos pulmones.
- Del ventrículo izquierdo sale la arteria aorta que se divide en varias ramas arteriales, incluidas las arterias coronarias, derecha e izquierda.

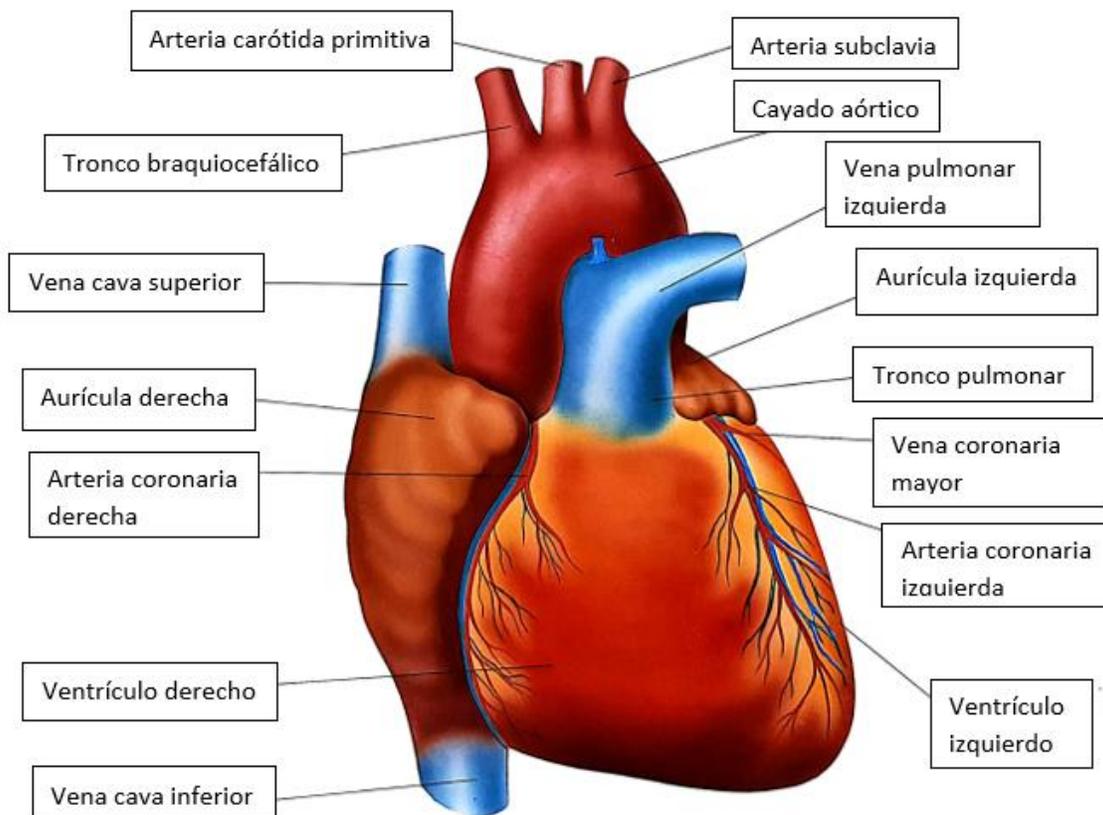


Figura 2.1 El corazón.

Las válvulas cardíacas que controlan el flujo de la sangre son cuatro:

- La válvula tricúspide es la que controla el flujo sanguíneo entre la aurícula derecha y el ventrículo derecho.



- La válvula pulmonar es la que controla el flujo sanguíneo del ventrículo derecho a las arterias pulmonares, que transportan la sangre a los pulmones para oxigenarla.
- La válvula mitral es la que permite que la sangre oxigenada proveniente de los pulmones pase de la aurícula izquierda al ventrículo izquierdo.
- La válvula aórtica es la que permite que la sangre oxigenada pase del ventrículo izquierdo a la aorta, que es la arteria más grande del cuerpo, y que transporta la sangre al resto del organismo.

2.3 El electrocardiograma

El electrocardiograma (ECG o EKG) es un registro gráfico que estudia las variaciones de voltaje con respecto al tiempo. Consiste en el registro de la actividad eléctrica que se desarrolla en el corazón en el transcurso del tiempo. Estos registros generalmente son en papel milimétrico; sin embargo, también se puede visualizar continuamente en un monitor que generalmente se encuentran en transporte sanitario medicalizado y en cuidados intensivos [9].

La señal de ECG tiene una amplitud de entre 0.5 y 5 mV, y un rango de frecuencias de utilidad diagnóstica de entre 0.05 y 125 Hz. Al igual que la mayoría de las señales biomédicas, la del ECG se caracteriza por ser no estacionaria, por la gran variabilidad que presenta entre distintos individuos y por su susceptibilidad al ruido. Las principales fuentes de ruido que afectan a la señal de ECG son [10].:

- La señal de electromiografía (EMG): que son las que se producen en los músculos del cuerpo, permiten su contracción.
- Desplazamientos de la línea de base: cuando el paciente llena sus pulmones al respirar, hace que cambie la posición de los electrodos con respecto al corazón y que se produzca un desplazamiento de la línea de base. El contacto variable entre los electrodos y la piel también puede provocar este tipo de ruido.



- Ruido de línea: los cables de los electrodos que registran el ECG y la línea eléctrica están acoplados por caminos capacitivos, lo cual hace que una corriente de 60 Hz (en México, 50 Hz en Europa) fluya por cada cable y a través del cuerpo hacia la tierra común.

El ECG recoge el trazado o el gráfico de un conjunto de ondas que Einthoven denominó P, Q, R, S y T de acuerdo con el orden en que aparecen en el tiempo. En la tabla 2.1, se describen los segmentos y ondas que conforman al electrocardiograma [11].

Tabla 2.1 Ondas y segmentos del ECG.

Onda P	Es la primera deflexión hacia arriba que aparece en el ECG. La cual describe la actividad auricular, que consiste en la contracción y envío de sangre hacia los ventrículos.
Segmento PR	Se encuentra entre la onda P y la Q. En este segmento, las aurículas terminan de vaciarse.
Complejo QSR	Representa el momento en que los ventrículos se contraen y expulsan la sangre contenida, consta de las ondas Q, R y S.
Onda Q	Es la deflexión negativa después del segmento PR y suelen ser de pequeño tamaño que casi no se note en ciertas derivaciones.
Onda R	Es la deflexión positiva más larga y es muy variable su altura dependiendo de la condición de las personas.
Onda S	Se observa enseguida del término de la onda R con deflexión negativa.
Segmento ST	Es la línea basal que se encuentra entre la onda S y el comienzo de la onda T y su valor de herramienta diagnóstica resulta insustituible.
Onda T	Es una deflexión generalmente positiva y representa el segmento donde el corazón está en relajación, después de expulsar la sangre de los ventrículos.

En la fig. 2.2 se observa la gráfica del electrocardiograma con cada segmento y onda antes mencionada [9].



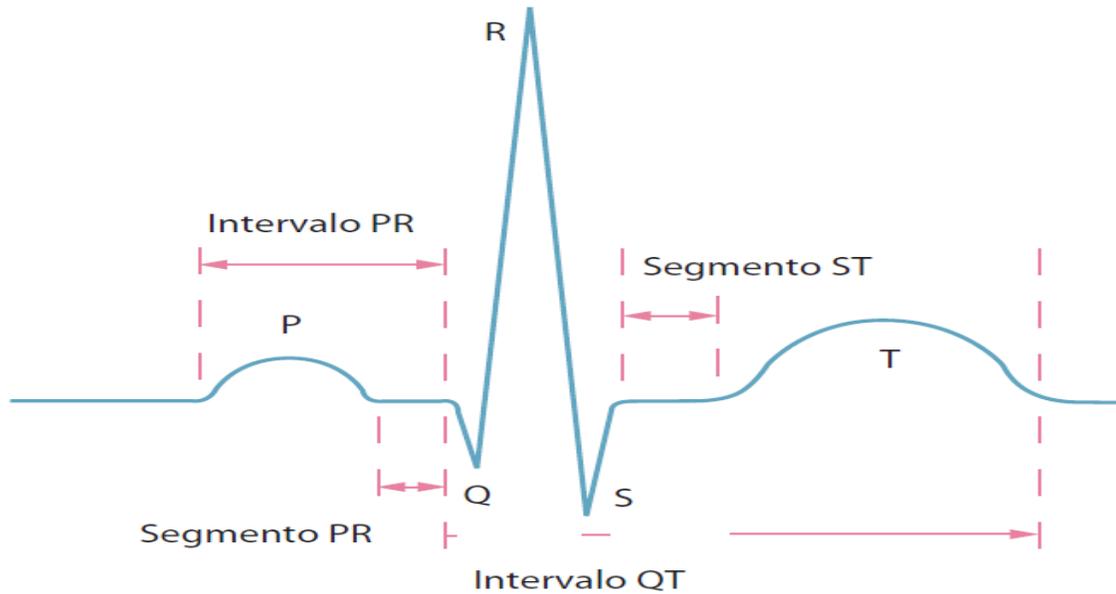


Figura 2. 2 Gráfica generada por electrocardiograma.

2.4 Las enfermedades del corazón, sus estadísticas globales y de México

2.4.1 Estadísticas globales

Las enfermedades no transmisibles (ENT), también conocidas como enfermedades crónicas, tienden a ser de larga duración y resultan de la combinación de factores genéticos, fisiológicos, ambientales y conductuales. Las principales causas de muerte por ENT son las enfermedades cardiovasculares (como los ataques cardíacos y los accidentes cerebrovasculares), el cáncer, las enfermedades respiratorias crónicas (como la enfermedad pulmonar obstructiva crónica y el asma) y la diabetes [1]. En la fig. 2.3 se muestra de forma gráfica las estadísticas globales de principales defunciones por ENT.

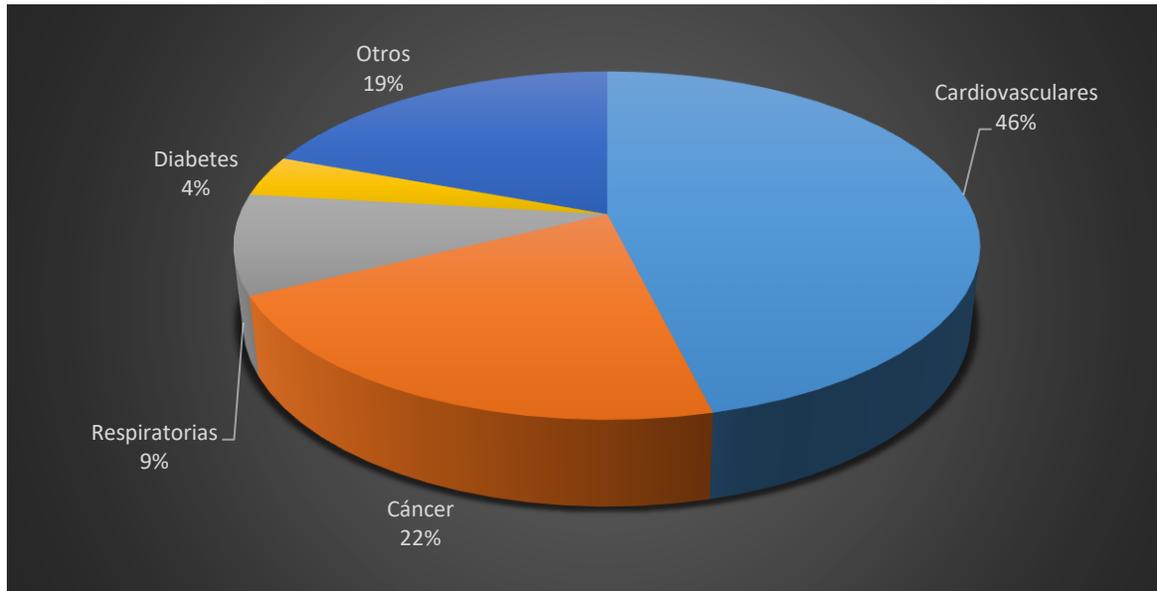


Figura 2. 3 Gráfica de principales causas de muerte a nivel mundial dentro de las ENT.

2.4.2 Estadísticas del país

En la población general, la principal causa de muerte desde que se tiene información (1998) han sido las enfermedades del corazón. En los adultos, las cinco primeras causas de muerte en hombres del 2017 fueron las enfermedades del corazón (20.1%), diabetes (14.1%), tumores malignos (10.8%), enfermedades del hígado (7.6%) y homicidios (7.3%). Por otro lado, la muerte en mujeres se debió principalmente a enfermedades del corazón (22.7%), diabetes (18.6%), tumores malignos (14.5%), enfermedades cerebrovasculares (6.1%) y enfermedades pulmonares obstructivas crónicas (3.8%) [12].

En la tabla 2.2 se muestra las principales causas de muerte más actuales que se encuentran en la página del sistema de información de la Secretaría de Salud (2018), donde las tasas se refieren al número de defunciones por cada 100 mil habitantes.





Tabla 2.2 Ondas y segmentos del ECG.

Principales Causas	Total(personas)	Tasas¹
Enfermedades del corazón	146,923	117.2
Diabetes mellitus	100,350	80.1
Tumores malignos	85,011	67.8
Enfermedades del hígado	38,563	30.8
Enfermedades cerebrovasculares	34,931	27.9
Accidentes	32,075	25.6
Agresiones (homicidios)	31,656	25.3
Neumonía e influenza	27,766	22.2
Enfermedades pulmonares obstructivas crónicas	23,219	18.5
Insuficiencia renal	13,710	10.9
Ciertas afecciones originadas en el período perinatal	11,943	9.5
Malformaciones congénitas, deformidades y anomalías cromosómicas	8,929	7.1
Desnutrición y otras deficiencias nutricionales	6,931	5.5
Lesiones autoinfligidas intencionalmente (suicidios)	6,470	5.2
Enfermedad por virus de la inmunodeficiencia humana	4,975	4.0
Sepsis	4,826	3.9
Bronquitis crónica, enfisema y asma	4,512	3.6
Enfermedades infecciosas intestinales	3,435	2.7
Anemias	3,297	2.6
Úlceras gástrica y duodenal	2,640	2.1
Mal definidas	8,789	7.0
Las demás causas	103,852	82.9

Como se puede observar, las muertes por enfermedades cardiacas superan por gran número a las muertes por las demás enfermedades y según la cruz roja, el 70 de los ataques al corazón ocurren en casas y en espacios públicos, de los que solo el 5 por ciento recibe atención inmediata [3].

¹ Número de defunciones por cada 100 mil habitantes.





2.5 Dispositivos analógicos para adquirir la señal eléctrica del corazón

Para adquirir la señal eléctrica del corazón es necesario acondicionarla para que pueda ser leída o interpretada por otros dispositivos, por ejemplo, un osciloscopio o un digitalizador. Para lograr esto es necesario:

- Adquirir la señal con electrodos, de tipo no invasivos
- Amplificador de instrumentación, que permite amplificar la señal proveniente de los electrodos
- Filtros pasa-bajas y pasa-altas, los cuales quitan los ruidos de la señal ya amplificada
- Amplificador operacional para amplificar aún más la señal y pueda ser leída por otros dispositivos

La tarjeta de adquisición está desarrollada con el chip ad8232 contiene cada uno de estos procesos. Como se observa en la fig. 2.4, la tarjeta cuenta con un conector de 3.5 mm para el cable de donde provienen los electrodos. Requiere una tensión de alimentación de entre 2 y 3.5 V (la tensión recomendada es de 3.3 V) cuenta con un LED indicador de la frecuencia cardíaca y pines: dos para la alimentación, uno para la salida analógica, dos salidas indicadoras de desconexión de los electrodos y uno para colocar el chip en modo *stand-by*.

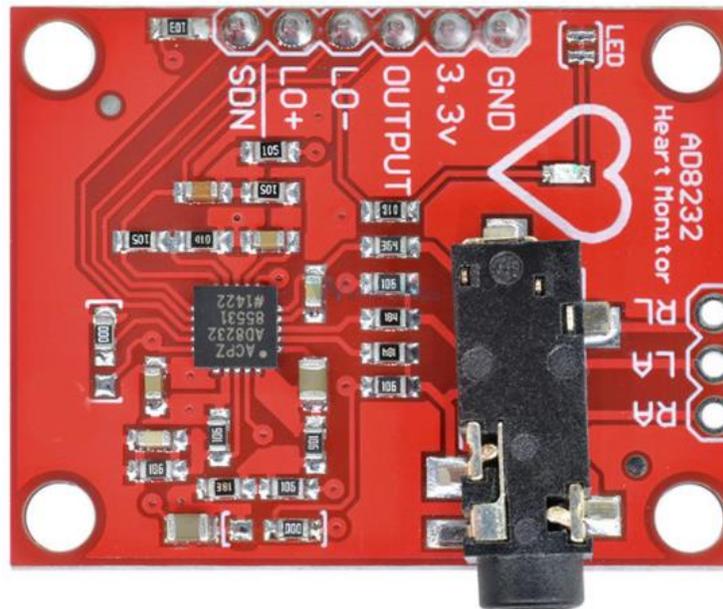


Figura 2. 4 Tarjeta de adquisición con chip AD8232.

2.5.1 Electrodo

Las corrientes en el cuerpo humano son conducidas por iones, mientras que, en los cables y la instrumentación electrónica, son conducidas por electrones. Es por esto que, para poder registrar o capturar señales bioeléctricas es necesario un elemento que sea capaz de actuar como transductor entre ambos tipos de corriente, este elemento es el electrodo [13].

Para que sea eficaz el contacto entre los electrodos y la piel, es recomendable limpiar la piel con alcohol antes de su colocación para eliminar grasas y remover el vello en áreas en las que se ubicarán los electrodos. Se utilizan generalmente electrodos de tipo no invasivos, es decir, de parches externos, colocados en forma de triángulo (derivaciones) y conectado por cables hacia la etapa de acondicionamiento de la señal electrocardiográfica.



Las derivaciones son la diferencia de potencial entre dos electrodos. Dependiendo del tipo de derivación, es como deben de colocarse los electrodos en el paciente. Los tipos de derivaciones más comunes son las bipolares y las unipolares:

Las bipolares, están compuestas por dos electros, uno positivo y uno negativo, también se integra un tercer electrodo que se conecta al común o tierra del sistema. En las **unipolares**, cada derivación es la diferencia de potencial entre cada electrodo con el común o tierra. En la figura 2.5 se observa la forma de colocar los electrodos, en este caso, el de tres derivaciones. Cada círculo de color es un electrodo.

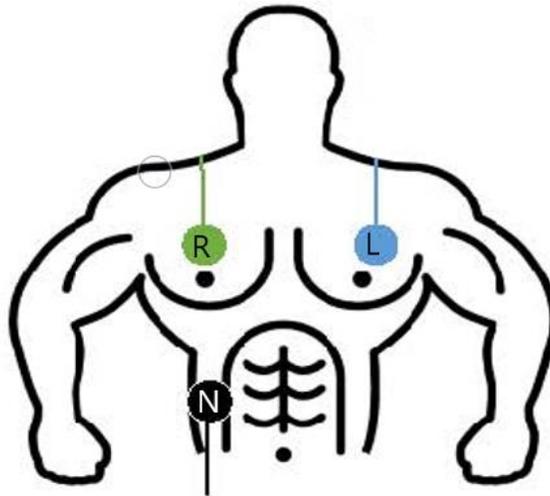


Figura 2. 5 Colocación de los electrodos, ECG tres derivaciones.

A continuación, se menciona más a fondo el funcionamiento de la etapa de acondicionamiento de la señal, incorporada en el chip AD8232.

2.5.2 Amplificador de instrumentación y filtro pasa-altas

La etapa fundamental del chip AD8232 es el amplificador de instrumentación. Es un circuito que amplifica el voltaje de entrada entre los electrodos y al mismo tiempo rechaza la señal de ruido e interferencia, estos dispositivos se encuentran internamente en el dispositivo, por lo que permite que las especificaciones estén garantizadas y el usuario



no tenga que hacer un diseño propio. Las características fundamentales de un amplificador de instrumentación son [13]:

- Coeficiente de Rechazo al Modo Común alto (CMRR) e impedancia de entrada alta; el cual se mide en decibeles y ohm, respectivamente. En el chip AD8232 es de 80 dB para un rango de frecuencias de entre 0 y 60 Hz y una impedancia de entrada de 10Gohms.
- Baja corriente de polarización de entrada; que es la corriente que pasa por las entradas del amplificador. La diferencia entre ambas corrientes de polarización se denomina corriente de *offset* de entrada. En el chip AD8232, las corrientes de polarización típicas son de 50 pA, y las corrientes de *offset* de entrada es de 25 pA.

Dentro de esta etapa se incluye un amplificador que permite hacer un filtrado pasa-altas de 2 polos y eliminar la corriente continua. Esto permite eliminar los *offsets* de los electrodos de hasta 300 milivolts y también señales de alta frecuencia como la desviación de la línea de base.

2.6 Digitalización de señales

2.6.1 Señales analógicas

Una señal analógica es un tipo de señal continua generada por algún fenómeno físico, representando una función matemática en la que es variable su amplitud y su periodo.

Estas señales son predominantes en nuestro entorno de vida, ya que se pueden encontrar en las variaciones de temperatura, presión, velocidad, sonido, etc. Y para poder tratarlas electrónicamente es necesario transformarlas en señales eléctricas, mediante el transductor que se adecue a estas [14].

Generalmente se pueden representar en un diagrama de amplitud contra tiempo o amplitud contra frecuencia, como se muestra en la figura 2.6.

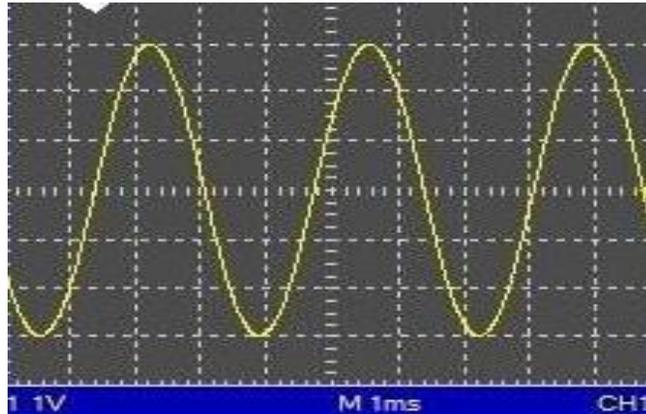


Figura 2. 6 Señal analógica.

2.6.2 Señales digitales

Una señal digital es un tipo de señal discontinua con el tiempo y solo toma ciertos valores discretos, estos valores son bien definidos que alternan en el tiempo transcurrido que generalmente se les asocian a los símbolos 0 o 1, llamados también pulsos o señal cuadrada. Estas señales no se producen en el mundo físico como tal, a diferencia de las señales analógicas estas señales están creadas por el hombre y estas se tratan con una técnica particular desde dispositivos digitales.

Resumiendo, las señales digitales solo adquieren un numero finito de estados diferentes, se clasifican según su número de estados (binarias, ternarias, etc.) y según su naturaleza eléctrica (unipolares y bipolares) [14]. En la fig. 2.7 se encuentra un diagrama de la representación de este tipo de señales.



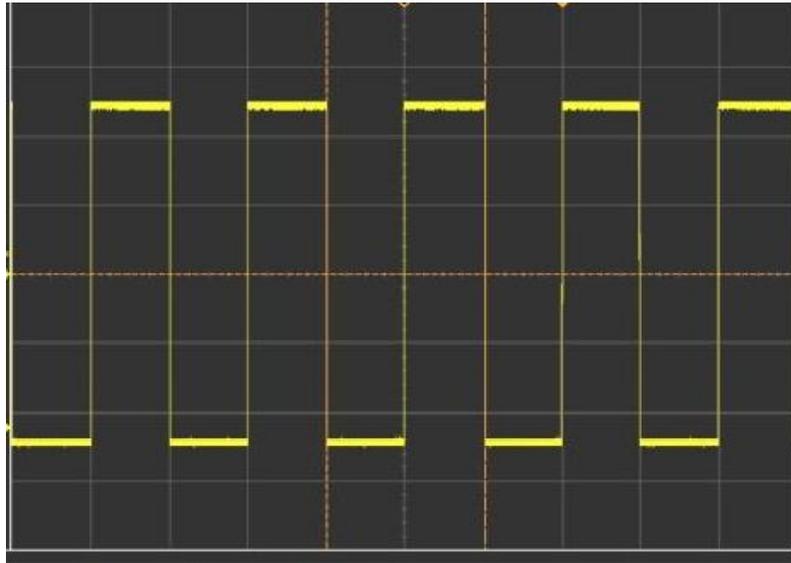


Figura 2. 7 Señal Digital.

2.6.3 Conversión analógico/ digital

En un sistema de adquisición de datos, la parte más importante del desarrollo es el convertidor analógico /digital, ya que esta etapa determina la precisión que pueda tener el sistema.

El convertidor analógico digital tiene como objetivo obtener una representación digital, es decir, obtener un conjunto de unos y ceros provenientes de una magnitud analógica. En otras palabras, los convertidores analógicos/ digital (ADC, por sus siglas en inglés) realizan la conversión o traducción de una señal continua en el tiempo (analógica) en valores o conjunto de valores discretos (digital). Para que la digitalización sea adecuada, es necesario tomar muy en cuenta la resolución y el muestreo a la hora de digitalizar. A continuación, se describen estos procesos [14].

Resolución: es la cantidad de bits entregados a la salida luego de la conversión. Puede expresarse también como el porcentaje o partes por millón (ppm) que representa cada bit menos significativo (LBS) o ancho de código, en el rango total de entrada. En otras palabras, el voltaje que ingresa de la señal analógica es dividida en porcentajes



dependiendo de la resolución del ADC (8 bits, 10 bits, 12 bits, etc.). Por ejemplo, un ADC de 10 bit con una entrada analógica de 0 a 5 volts, divide ese rango de 0 a 5 volts en un rango de 1024 porciones. En la fig. 2.8 se muestra que entre más resolución tenga el ADC, mayor será la cantidad de proporciones a tomar.

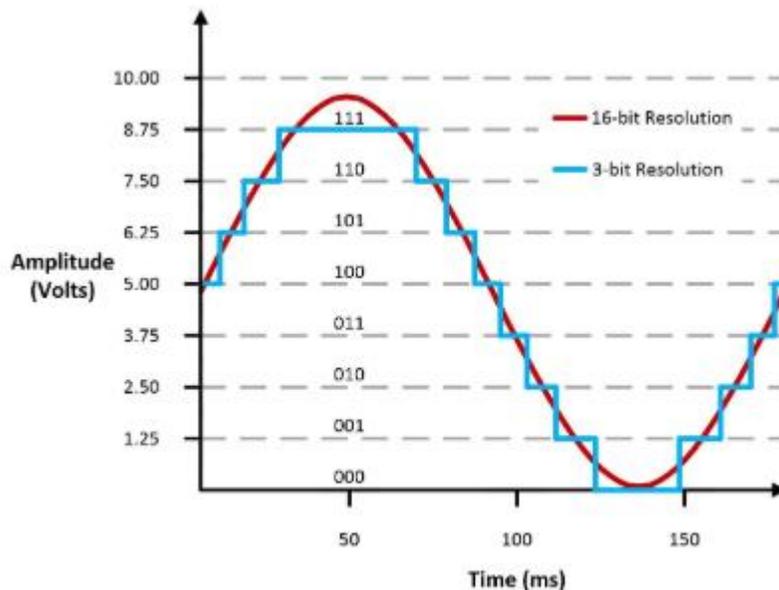


Figura 2. 8 Resolución de la señal.

Muestreo: En la mayoría de los casos es necesario muestrear y convertir señales que varían en el tiempo en forma arbitraria, con la sola restricción de algún límite en el ancho de banda requerido. El teorema de Shannon/Nyquist establece que “una señal debe ser muestreada con una frecuencia por lo menos del doble del componente de máxima frecuencia en la señal”. Si la frecuencia de muestreo es menor a la mínima establecida por el teorema de Shannon/Nyquist, la resultante de la gráfica podría ser distorsionada y pudiera perderse algún contenido [15]. En la fig. 2.9 se muestra un ejemplo de cómo es una onda analógica y como sería el muestreo de ésta.

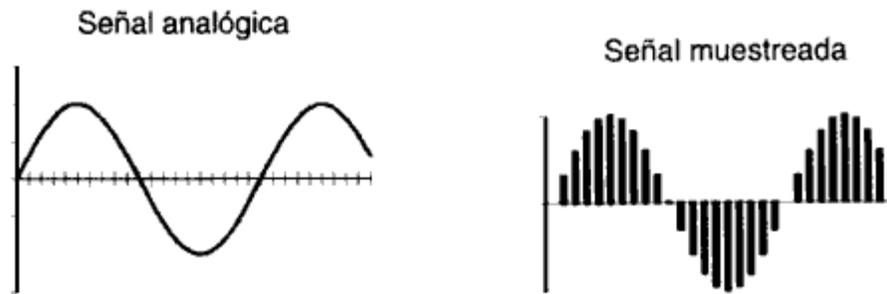


Figura 2. 9 Señal analógica muestreada.

2.7 Microcontroladores

En electrónica, para toda serie de procesos se ocupa una unidad de procesamiento (CPU) concentrado en un circuito integrado. Esta unidad de procesamiento requiere por lo menos puertos de entrada y salida, y memoria para almacenar datos o para almacenar la programación del microprocesador. Al conjunto de dispositivos adicionales al microprocesador se le llaman periféricos, y a la unión del microprocesador con los periféricos en un circuito integrado se le denomina microcontrolador [16].

En la fig. 2.10 se muestra el diagrama a bloques general de un microcontrolador.

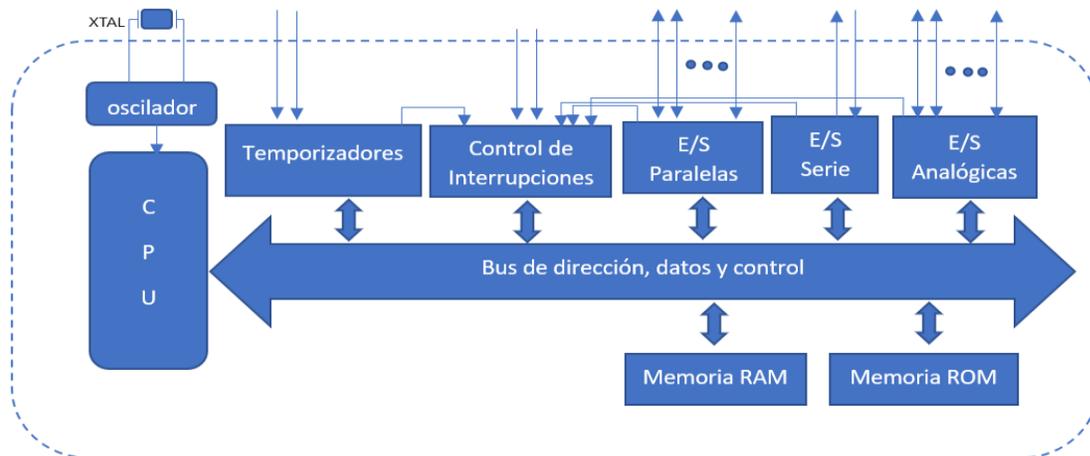


Figura 2. 10 Diagrama a bloques general de un microcontrolador.



Los microcontroladores cuentan con un oscilador que le permite la sincronización de todas las operaciones internas, generalmente son de cuarzo (XTAL) por su gran estabilidad de frecuencia.

Muchos de los microcontroladores pueden incluir puertos de comunicación serial (UART, USART, I²C, etc.), convertidores analógico-digital (ADC) y digital-analógico (DAC), memoria volátil que pierde la información almacenada cuando falte energía (RAM), memoria no volátil (ROM, EPROM, OTP, FLASH, etc.), que es donde se almacena permanentemente el programa que debe ejecutarse, y en peculiar importancia, las entradas y salidas (E/S) para poder comunicarse con el exterior [17].

Las entradas y salidas pueden ser analógicas, seriales y digitales, éstas van conectadas físicamente al microcontrolador para comunicarse con infinidad de dispositivos de diferentes características para cada tipo de puerto.

Para poder controlar el tiempo de ejecución de las instrucciones se ocupan los temporizadores, son módulos de hardware incluido en el microcontrolador que está diseñado para incrementar el valor de un registro. Los temporizadores reciben pulsos o señales de reloj de un origen externo o interno [16].

Entre la variedad de microcontroladores, se procura que el diseñador tenga un microcontrolador hecho a la medida de sus necesidades y de su aplicación. Es por esto que los fabricantes ofrecen diferentes tipos de familias de microcontroladores, donde cada familia ejecuta el mismo repositorio de instrucciones, pero con diferentes componentes físicos (capacidad de memoria, variedad de dispositivos de entrada y salida, etc.), teniendo en cuenta esto, los diseñadores pueden elegir el más idóneo para su aplicación.

2.8 Microcomputadoras

Las microcomputadoras o computadoras personales (Pc's), son computadoras para uso personal que tuvieron su origen junto con la creación de los microprocesadores. Son de





bajo precio y actualmente se encuentran en oficinas, escuelas, trabajos y casi en cualquier hogar [18].

El término PC surge en el año de 1981, cuando IBM sacó a la venta su modelo “IBM PPC”, la cual se convirtió en un tipo de computadora ideal para el uso personal, después este término se estandarizó y los computadores sacados por otras empresas fueron llamados “PC y compatibles”, los cuales usaban el mismo procesador que las de IBM, con un costo menor y ejecutando los mismos programas [18].

Entonces, se puede entender como microcomputadora a las computadoras personales de torre y monitores o a las laptops que dependiendo de las características de hardware que posean es su costo y generalmente son accesibles para el público común o general.

También, actualmente existen otras alternativas que a pesar de considerarse como “placa computadora” por ser, en esencia, placas con capacidad computacional, sin nada más, no deja de ser una microcomputadora, ya que incluye un microprocesador. Son microcomputadoras de bajo costo por ser de software libre, con características similares a una convencional y que generalmente se son usadas por desarrolladores de tecnologías. Entre estas microcomputadoras se encuentra:

- Orange pi
- PCDuino
- Raspberry pi

2.8.1 Orange pi

Es una computadora de una sola placa de código abierto fabricada por Shenzhen Xunlong Software CO., Limited. Se puede ejecutar Android, Ubuntu, Debian, así como las imágenes de Raspberry Pi y Banana Pi. Utilizan los microprocesadores AllWinner H2, H3 y H5 SoC, A64 Quad-core Cortex-A53 64bit o ARM Cortex-A5 32bit, entre otros; tienen desde 256MB las placas más básicas y hasta 2 GB DDR3 SDRAM de RAM





pudiendo tener según el modelo Ethernet, Bluetooth, WiFi por medio de un módulo, e incluso 2G (ver fig. 2.11) [19].



Figura 2. 11 Orange pi.

2.8.2 PcDuino

PcDuino es una computadora de bajo costo, múltiples funciones y de alto rendimiento. Ejecuta sistemas operativos como Ubuntu Linux y Android. pcDuino tiene una interfaz HDMI para poder visualizarla en su monitor. Podría admitir decodificador de video multiformato de 1080p a 60 fps y codificador de video 1080p a 30 fps H.264 y MPEG4 con su procesador de video incorporado. Se dirige especialmente a la comunidad de código abierto. PcDuino proporciona múltiples herramientas fáciles de usar y es compatible con el popular ecosistema Arduino, como Arduino Shields (Ver figura 2.12) [20]. La placa cuenta con las siguientes características:

- CPU: 1GHz ARM Cortex A8.
- GPU: OpenGL ES2.0, OpenVG 1.1 Mali 400 core.
- DRAM: 1GB.
- Almacenamiento: 2GB Flash y slot de memoria SD expandible hasta 32GB.
- Salida de video: HDMI.



- Sistemas operativos compatibles: Linux3.0 + Ubuntu 12.04, Android ICS 4.0.
- Requisitos de alimentación: 5VDC, 2A.
- 2 puertos USB.
- 1 puerto Ethernet 10/100Mbps.

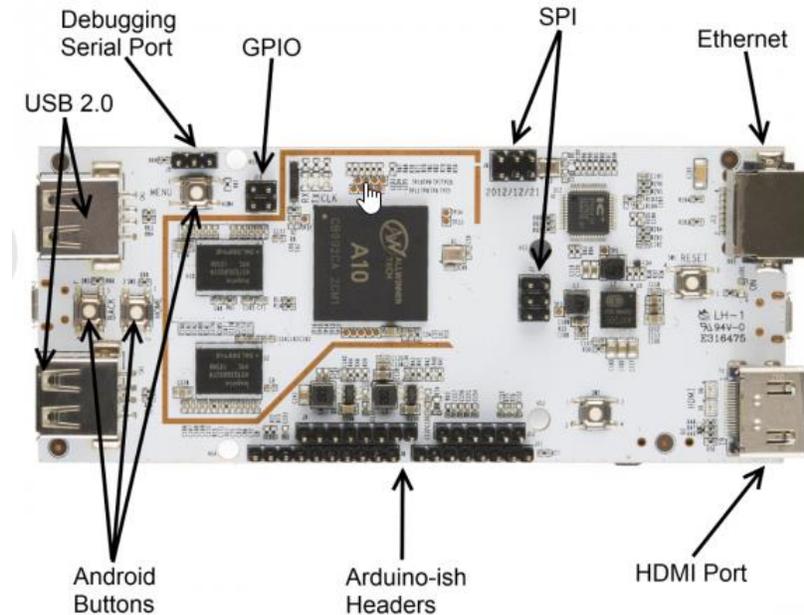


Figura 2. 12 pcDuino.

2.8.3 Raspberry pi

Raspberry Pi es una placa o microcomputadora de bajo costo, de tamaño reducido similar al de una tarjeta de crédito, desarrollado en Reino Unido por la fundación Raspberry por la Universidad de Cambridge en el año 2006, teniendo de objetivo incentivar el estudio de la informática en las escuelas, luego se hizo comercial en el año 2012.

La Raspberry Pi 3 Modelo B + es la revisión final en la gama Raspberry Pi 3, la cual cuenta con las siguientes especificaciones [21]:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) SoC de 64 bits a 1.4GHz.



- SDRAM LPDDR2 de 1 GB.
- LAN inalámbrica IEEE 802.11.b / g / n / ac de 2.4GHz y 5GHz, Bluetooth 4.2, BLE.
- Gigabit Ethernet sobre USB 2.0 (rendimiento máximo 300 Mbps).
- Cabecera GPIO extendida de 40 pines.
- HDMI de tamaño completo.
- 4 puertos USB 2.0.
- Puerto de cámara CSI para conectar una cámara Raspberry Pi.
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi.
- Salida estéreo de 4 polos y puerto de video compuesto.
- Puerto Micro SD para cargar su sistema operativo y almacenar datos.
- Entrada de alimentación de 5V / 2.5A DC.
- Compatibilidad con alimentación por Ethernet (PoE) (requiere un PoE HAT separado).

Como se puede observar, la Raspberry pi 3 b+, a diferencia de las anteriores microcomputadoras cuenta con wifi incorporado en la tarjeta, que es de mucha utilidad para proyectos de innovación; muy bajo costo, pero, sobre todo, muy fácil y común de conseguir. Sus comunidades son inmensas, por lo que se puede encontrar mucho soporte para los proyectos e innovaciones que se requieran. En la figura 2.13 se visualiza la Raspberry pi.



Figura 2. 13 Raspberry pi 3b+.

2.9 Protocolos de transferencia de datos en internet

A continuación, se enlistan algunos de los protocolos más utilizados en internet y se mencionan sus principales características:

- Protocolo de control de transmisión (TCP).
- Protocolo de transferencia de archivos (FTP).
- Protocolo de transferencia de hipertexto (HTTP).
- Secure Shell (SSH).

2.9.1 Protocolo de control de transmisión (TCP)

TCP es el protocolo de transporte más utilizado por las aplicaciones. Es un protocolo orientado a la conexión, la cual se lleva a cabo en tres fases: *establecimiento de conexión*, *transmisión de datos* y *finalización de la conexión* [22]. Con este protocolo se proporciona una comunicación fiable y libre de errores.

Su unidad de transmisión es la de paquete, entonces cuando se quiere enviar información a un destino, se divide en paquetes de un tamaño fijo. Se marca cada paquete con un número de secuencia y lo envía después de haber establecido una conexión con el destino [22].



2.9.2 Protocolo de transferencia de archivos (FTP)

El protocolo FTP es, que como su nombre lo dice, un protocolo de transferencia de archivos.

Es un protocolo de aplicación TCP/IP, que implementa la función de transferencia de archivos, tiene una especie protocolos múltiples de conexión, las conexiones FTP se dividen en conexiones de control y conexión de datos. Las conexiones de control usan o derivan del protocolo telnet para transmitir los comandos y las respuestas adoptan el protocolo TCP para cargar o descargar los archivos [23]. El objetivo del FTP es:

- Permitir que equipos remotos puedan compartir archivos.
- Permitir la independencia entre los sistemas del equipo cliente y el equipo servidor.
- Permitir una transferencia de datos eficaz.

2.9.3 Protocolo de transferencia de hipertexto (HTTP)

El HTTP es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/1 está recogida en el RFC 2616.

HTTP está basado en operaciones de solicitudes y respuestas. Un cliente establece una conexión con un servidor mediante la solicitud de una página web y envía un mensaje con los datos a solicitar. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web es conocido por su URL (Uniform Resource Locator por sus siglas en inglés) [24].

Es decir, cuando un usuario ingresa a una página web, crea una solicitud hacia el servidor donde se aloja esta página y el servidor da como respuesta la página, por lo regular creada a partir de HTML (Lenguaje de marcado de hipertextos), sin la necesidad de realizar ninguna otra acción. Cabe mencionar que las páginas que cuentan con cifrado





SSL (Capa de Sockets Seguros) aparecen como HTTPS (protocolo seguro de transferencia de hipertexto), esto hace que sea seguro tanto para el solicitante o cliente, como para el servidor.

2.9.4 Secure Shell (SSH)

Cuando se contrata un servidor dedicado o un VPS con una empresa de alojamiento web, debe utilizar algún tipo de mecanismo para conectarse con él remotamente y poder interactuar con todas sus funciones. Debido a la importancia de los datos que se suelen mover en este tipo de máquinas, es fundamental contar con algún protocolo de conexión que nos garantice total seguridad a la hora de comunicarnos con el servidor. Esta seguridad nos la proporciona el protocolo SSH (Secure Shell), probablemente el sistema más utilizado actualmente para comunicarse entre equipos con sistema operativo Linux [25].

Hasta la aparición de este protocolo, la forma que utilizaban los usuarios para conectarse con los servidores era por medio del uso de otros protocolos como Telnet o FTP, protocolos que tienen una gran desventaja respecto al SSH, ya que en este último encripta la sesión de conexión, impidiendo que alguien pueda capturar las credenciales de autenticación [25]. Algunas características más relevantes son:

- Las claves de encriptación utilizadas sólo son conocidas por quien emite la información y por quien la recibe.
- Cuando un usuario se autentica, entre él y el servidor se crea un canal seguro cifrado por donde intercambiar la información con total garantía.
- Los datos enviados y recibidos mediante el uso de SSH se realiza por medio de algoritmos de encriptación de 128 bits, lo que hace que sea muy complicado de descifrar y de leer.
- Una sola alteración de la clave modifica el mensaje original, lo que permite que si por alguna razón un tercero descubre la clave no acceda al mensaje completo, entre otros.





2.10 Servidores

2.10.1 Servidor Privado Virtual (VPS)

El servidor privado virtual (VPS, por sus siglas en inglés) es una partición virtual situado en un servidor físico dedicado con recursos exclusivos para cada partición, dependiendo de las necesidades que se requieran; en otras palabras, consiste en particionar un servidor físico en varios servidores virtuales [26].

2.10.2 Servidores Dedicados

El servidor dedicado es una maquina física completamente independiente, es decir, la maquina es completamente para el cliente y solo el cliente tiene acceso a este. El cliente desde un principio elige las características técnicas que requiera.

2.10.3 Comparativa entre servidores dedicados y VPS

En la tabla 2.3 se muestran las diferencias entre los servidores tradicionales y los VPS, con la finalidad de poder entender mejor sus diferencias, ventajas y desventajas.

Tabla 2.3 Ondas y segmentos del ECG.

Servidores Privados Virtuales	Servidores Dedicados
La máquina física funciona como un ordenador de hogar, todos pueden tener acceso a ella con su respectivo segmento.	La máquina es exclusivamente del cliente y tendrá derecho a todos los recursos de toda la máquina.
Pensado para sitios o proyectos que empiezan y no tienen mucho tráfico, ya que tiene opciones de gama alta y baja.	Pensado para sitios o proyectos que tienen más de 500,000 al mes.
No requiere de un técnico para poder aumentas sus capacidades físicas (RAM, ROM, etc.), ya que, al ser solo una partición virtual puedes aumentar su capacidad de recursos sin perder la conexión.	Se debe contar con el personal técnico necesario para poder dar solución en caso que se requiera una expansión de recursos o para solucionar un problema, perdiendo por un tiempo la conexión del servidor.
Existen plataformas donde puede incluir los servicios que mejor se adapten a tu proyecto	Al contratar un servidor dedicado, se contratan los servicios y aplicaciones que se requieran desde un





con solo solicitarlo, en cuestión de segundos se tendría el servicio deseado.	inicio. Esto es ideal para proyectos sólidos y que se sabe cuáles aplicaciones o servicios se usaran.
Al comenzar con recursos básicos y al ser compartidos el recurso físico, son más baratos.	El servidor dedicado es más caro por las características que se mencionan.

Teniendo en cuenta las características anteriores, se puede concluir que es conveniente la opción del servidor privado virtual, ya que es de bajo costo, se tiene control de administrador como servidor dedicado; en caso de requerir más recursos, se puede ampliar sin complicaciones de manera sencilla y si el proyecto creciera, se podría migrar a un servidor dedicado.

2.10.4 Amazon Web Services (AWS)

AWS es una plataforma en la nube que ofrece servicios a nivel global. Está pensada para innovar de manera rápida, ya que cuenta con muchas aplicaciones a un bajo costo. Cuenta con muchas aplicaciones como máquinas virtuales, bases de datos e incluso inteligencia artificial [27].

Amazon EC2 (Amazon Elastic Compute Cloud) es una función o aplicación de AWS, es un entorno virtual que permite tener máquina virtual con diferentes sistemas operativos, pudiendo cargar el entorno de aplicaciones deseadas y administrando permisos de red.

Además, cuenta con las siguientes características:

- **Instancias dedicadas:** permiten que las aplicaciones tengan acceso directo al procesador y memoria del servidor.
- **Instancias de informática con GPU (Unidad de Procesamiento Gráfico):** para clientes que necesiten de mayor capacidad de procesamiento gráfico.
- **Instancias de almacenamiento HDD:** para los clientes que necesiten mayor espacio en el servidor, ya sea para base de datos o servicio web.
- **Auto scaling:** EC2 permite ajustar la escala de la capacidad automáticamente de acuerdo a las condiciones que defina, entre muchos otros.





Algo que resalta de AWS y EC2, es que te dan 1 año gratis de servicio con respectivas restricciones, pero que son adecuadas para probar el servicio y probar con proyectos que se vayan creando. Después de este periodo el cobro es conforme se use este servidor y las aplicaciones que se le vayan incorporando.

2.11 Bases de datos

Una base de datos o DB (por sus siglas en inglés) es un conjunto de datos del mismo contexto y almacenados sistemáticamente para su posterior uso. Entonces bien, si una biblioteca almacena libros y textos para consultarlos, podría considerarse como una base de datos. Sin embargo, debido al crecimiento electrónico e informático, la mayoría de bases de datos se encuentran en formato digital, ofreciendo una gran variedad de soluciones al problema de almacenar datos [28].

Para poder gestionar toda la información que se tiene en las bases de datos, es necesario contar con un manejador o gestor que nos permita facilitar las consultas a dichas bases. A continuación, se menciona y describe el sistema gestor de base de datos, que lo conforma y que lenguaje utiliza.

2.11.1 Sistema gestor de base de datos (SGBD)

Un sistema gestor de base de datos permite almacenar y acceder a los datos de forma rápida y estructurada, utilizando una interfaz de acceso. Cumplen con tareas fundamentales para resguardar, dar mantenimiento y actuar ante un fallo físico [28].

Se compone principalmente del hardware donde se aloja físicamente la base de datos, el sistema manejador de base de datos (por ejemplo, MySQL) y los datos almacenados.

El modelo relacional se emplea en gran variedad de sistema de bases de datos como Oracle, MySQL, Microsoft SQL Server, etc. Ocupa algebra relacional con el fin de poder hacer operaciones con los datos entre las tablas, es decir, relacionar las tablas con sus datos.



SQL (Lenguaje de Consulta Estructurado) es el lenguaje que más se utiliza para poder definir y construir bases de datos, además para hacer consultas o búsquedas de los datos almacenados. Se caracteriza por manejar el álgebra y el cálculo relacional, permitiendo hacer búsquedas de manera más rápida y sencilla la información que se desea [28].

2.12 Aplicación web

Una aplicación web es una herramienta informática accesible desde cualquier navegador, ya sea a través de internet (lo más común) o bien a través de una red local. A través del navegador se puede acceder a toda la funcionalidad y herramientas proporcionadas y adaptadas para cada aplicación. Estas aplicaciones cuentan con 3 niveles [29] (ver figura 2.14):



Figura 2.14 Diagrama aplicación web.

Usuarios. – Este nivel es cuando el usuario accede desde el navegador web hacia la aplicación, se puede incluir el administrador o solo el usuario final.



La aplicación. – En este nivel está el código y la gestión de la aplicación, es decir, lo que hará nuestra aplicación.

Base de datos. – Este nivel se almacenarán lo datos de la aplicación, como son: usuarios, datos de los usuarios, contraseñas, etc.

2.13 Metodología de desarrollo

2.13.1 Modelos de proceso evolutivo

Con frecuencia los requerimientos y productos del proyecto cambian conforme avanza el desarrollo, haciendo que no se avance de forma lineal hacia el producto final. Los modelos evolutivos son iterativos. Es decir, permiten desarrollar versiones cada vez más completas o mejorar detalles detectados.

2.13.2 El modelo Prototipo

Es un tipo de modelo de proceso evolutivo que sirve principalmente, y como su nombre lo dice, para hacer prototipos del proyecto a desarrollar dando una vista preliminar al usuario [30].

Esta metodología se utiliza cuando los objetivos o las características finales de los proyectos a desarrollar no son muy claras o no están completamente definidas desde el inicio. Se basa en la creación de prototipos rápidos que permiten, conforme se va desarrollando, visualizar ciertos aspectos de interés del proyecto. Al término de cada prototipo es fundamental la comunicación entre el desarrollador y cliente con el fin de recibir una retroalimentación que sea la base del trabajo futuro.

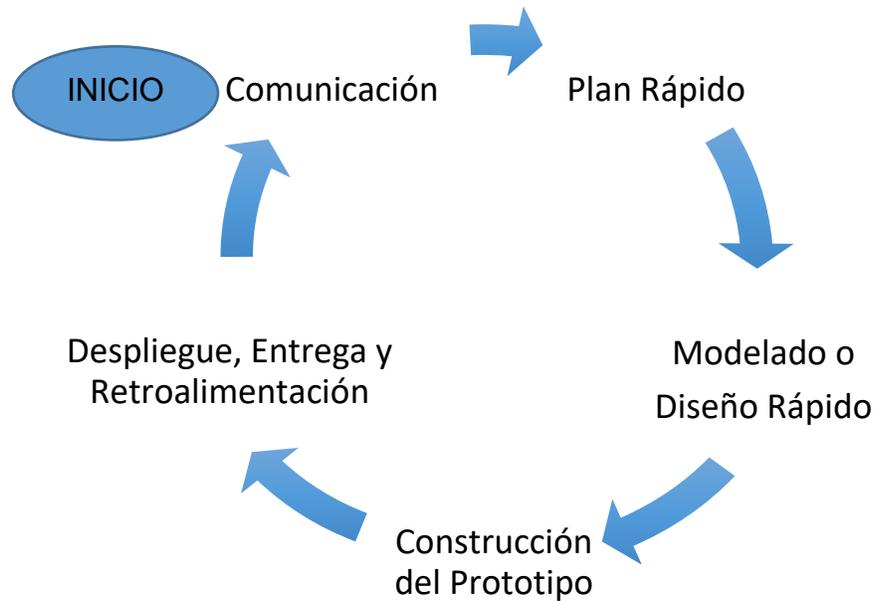


Figura 2.15 Modelo de prototipos.

En la figura. 2.15 se muestra el proceso del modelo de prototipos, el cual comienza con la comunicación. Consiste en reunirse con otros participantes para plantear y definir los objetivos generales. Se planea una iteración rápida para hacer el prototipo y se lleva a cabo un “diseño rápido”. El diseño se centra en la representación de los aspectos que interactúen más con el usuario. Del diseño se pasa a la construcción del prototipo. El prototipo se entrega y se evalúa por los que participen en el proyecto, para que den retroalimentación y así mejorar los parámetros que se requieren [30].





Capítulo 3. Desarrollo experimental

El diagrama que representa al diseño del sistema propuesto se muestra de manera general en la fig. 3.1.

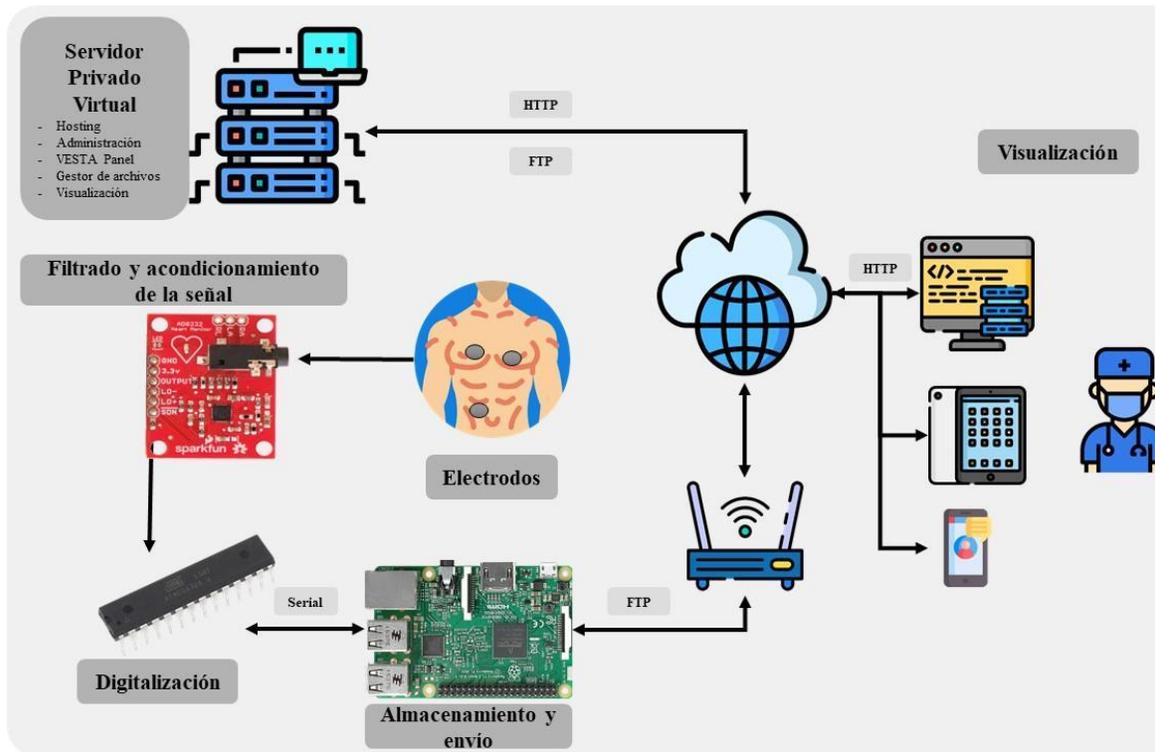


Figura 3.1 Diagrama del sistema general.

Mediante los electrodos se adquieren las señales electrocardiográficas. El acondicionamiento de la señal se realiza con la tarjeta AD8232, donde la señal proveniente de los electrodos es filtrada y amplificada. La señal biomédica es digitalizada utilizando el convertidor analógico-digital integrado en el microcontrolador Atmega328; posteriormente, estos datos digitalizados son transmitidos de manera serial a la micro computadora Raspberry pi.



La Raspberry pi, almacena los datos que le llegan del puerto serial en un documento de texto plano y cuando tiene conexión a internet, sube esos archivos al servidor, de no tener conexión, los sigue almacenando hasta que tenga la oportunidad de subirlos.

En este servidor se guardan los archivos de texto plano y mediante una interfaz, se pregunta al usuario que medición quiere visualizar con su respectiva fecha y hora. Entonces, el usuario puede acceder a la página para visualizar y recuperar los datos, esto a través de cualquier dispositivo que cuente con un navegador web y salida a internet.

Como se puede observar y resumiendo, el sistema se compone de 3 etapas principales:

- Adquisición de las señales electrocardiográficas: captura las señales eléctricas que produce el corazón por medio de electrodos; acondiciona la señal; realiza la conversión analógico-digital y transmite vía serial los datos a la siguiente etapa.
- Captura, almacenamiento y envío de datos: recibe la información por el puerto serial; almacenas lecturas, y cuando tiene la conectividad a la red, hace el envío al servidor.
- Servidor remoto: que consiste en montar un servidor; gestión de los archivos para que el usuario seleccione fechas y horas de las mediciones; presentación de manera gráfica en una aplicación web.

En el siguiente apartado se describe de manera más detallada cada una de estas etapas del sistema.



3.1 Adquisición de las señales electrocardiográficas

Se empieza con la etapa de adquisición de las señales electrocardiográficas, la cual incluye (con respecto al diagrama general, figura 3.1): los electodos, filtrado, acondicionamiento de la señal, digitalización y transmisión serial.

3.1.1 Electroodos

Para adquirir la señal se necesitan electodos como los que se muestran en la figura 3.2, se usaron electodos del tipo no invasivo; esto es, que solo se colocan como parches formado un triángulo (3 derivaciones) sin tener que perforar la piel del paciente, la colocación se observa en la figura. 2.5. Inicialmente se le agrega gel conductivo para adquirir mejor la señal eléctrica del corazón, luego se conectan por medio de cables a la siguiente etapa.



Figura 3.2 Electroodos no invasivos.

3.1.2 Acondicionamiento de la señal

En la fig.3.3 se muestra el diagrama a bloques del proceso que realiza la tarjeta AD8232 para poder filtrar y acondicionar la señal proveniente de los electodos y la pueda interpretar el convertidor analógico-digital en la etapa posterior.



Figura 3.3 Diagrama a bloques de Acondicionamiento de la señal.



Los cables que provienen de los electrodos se conectan a la tarjeta AD8232 por medio de un plug de 3.5 milímetros, la tarjeta amplifica la señal proveniente de los electrodos (a este amplificador se le denomina amplificador de instrumentación). Al amplificar la señal también se amplifican los ruidos externos al sistema como: corrientes de línea, ruidos o señales emitidas por los músculos, etc.; es por esto que se tiene que pasar por otra etapa de filtrado, con un filtro pasa-bajas y un pasa-altas (incluidas en la tarjeta AD8232) para limpiar la señal; una vez filtrada la señal se vuelve a amplificar o acondicionar para que lo pueda leer o interpretar el convertidor analógico digital (ADC) en la siguiente etapa.

Cabe mencionar que la tarjeta AD8232 tiene pines de polarización y también cuenta con un pin de salida donde entrega la señal. En la fig. 3.4 se muestra como entrega la señal de forma analógica.

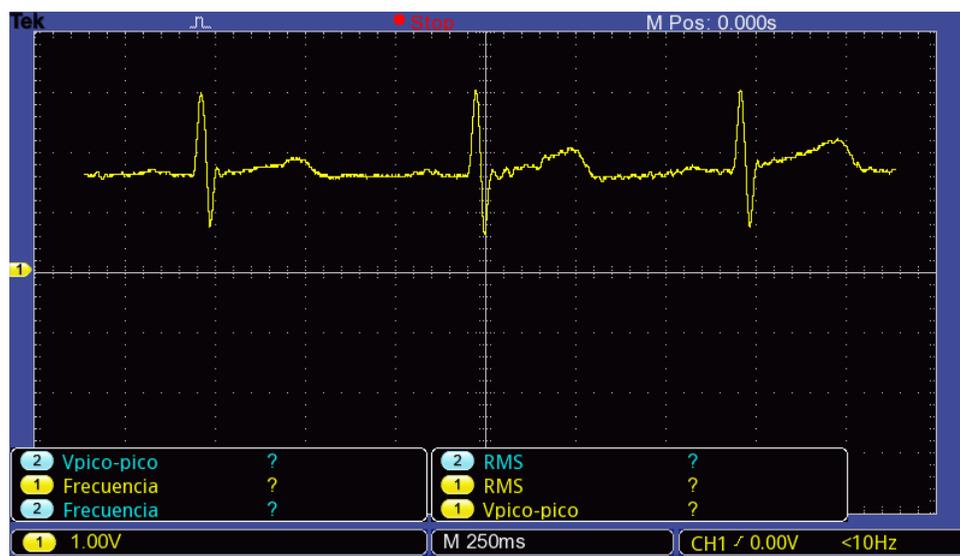


Figura 3.4 forma de la salida de señal del AD8232.

3.1.3 Conversión analógico-digital y transmisión serial.

Para hacer la conversión analógico digital, se usó el microcontrolador Atmega328. El cual cuenta con un convertidor analógico-digital (ADC) con una resolución de 10 bits, por lo





que la digitalización será de 0 a 1023 bits, también incorpora un módulo de transmisión serial (USART) que se ha configurado con una velocidad de transferencia de 115200 baudios.

De manera general el proceso de digitalización se lleva a cabo de la siguiente manera:

1. Lectura y digitalización de la señal analógica.
2. Transmisión de la señal digital por el puerto serial.
3. Tiempo de espera para realizar la siguiente lectura.

Para este trabajo, la tasa de muestreo debe ser de 250 Hz, o lo que es lo mismo, cada 4 milisegundos, cumpliendo con el criterio de Nyquist para lograr obtener componentes de frecuencias de hasta 125Hz. Para ello se diseñó un método que toma en cuenta el tiempo de digitalización, se suma el tiempo de transmisión serial y se espera lo que resta para completar los milisegundos establecidos. En la figura 3.5 se muestra el algoritmo utilizado para satisfacer la frecuencia de muestreo a 250Hz, es decir, para hacer el proceso cada 4 milisegundos incluyendo todas las acciones del microcontrolador.

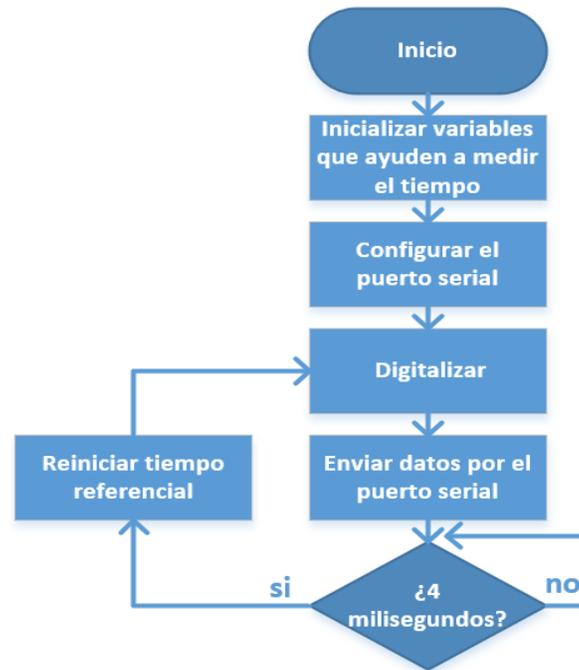


Figura 3.5 Diagrama de flujo del algoritmo en el microcontrolador.

El proceso descrito en el algoritmo es el siguiente: se inicializa las variables que se usaran para medir el tiempo, luego se configuran los pines y puertos a usar. Posteriormente se digitalizan los valores que llegan a través del pin del ADC, para después enviarlo por el puerto serial hacia la siguiente etapa; si la variable inicializada para medir el tiempo, alcanzó los 4 milisegundos, después de haber hecho la digitalización y transmisión serial, vuelve al proceso de digitalizar y transmitir por el puerto serial (aunque esto no sucede según los pruebas experimentales que se presentarán en el siguiente capítulo, ya que se comprobó que la digitalización y transmisión serial consumen tiempo) ; en caso de no haber llegado a los 4 milisegundos, espera hasta que esto ocurra.

Entonces, al microcontrolador se le conecta la tarjeta AD8232 en el pin del ADC, digitaliza y manda los datos por medio de los pines seriales a la microcomputadora Raspberry pi que realiza el proceso en la siguiente etapa.





3.2 Captura, almacenamiento y envío de datos

Para esta etapa se utilizó una Raspberry pi 3 B+ en la cual se instaló el sistema operativo Raspbian, el cual se consiguió entrando a la página oficial de Raspberry pi en la sección de descargas². Se guardaron los archivos descargados en una memoria microSD que se insertó a la Raspberry y se procedió a la instalación.

Se definió esta microcomputadora con la finalidad de poder almacenar y transmitir al servidor la información obtenida de la etapa de adquisición. También se definió guardar los datos en archivos de texto plano que por nombre llevan la fecha y hora en la que se crearon, y cuyo contenido corresponde a cada una de las lecturas provenientes de la etapa de adquisición de las señales, las cuales son recibidas por el puerto serial de la Raspberry.

En la figura 3.6 se muestra el diagrama de flujo del algoritmo propuesto y utilizado para poder almacenar los datos y el envío de los paquetes (archivos de texto plano) al servidor. Esta etapa fue programada en lenguaje Python.

² <https://www.raspberrypi.org/downloads/raspberry-pi-os/>



y hora en el momento de la medición en forma de cadena de caracteres. Después, se va escribiendo en el archivo cada dato que va llegando del microcontrolador, con un salto de línea entre cada uno, hasta que se llega a los 5 minutos (750,000 datos) o el tiempo que se quiera establecer; dentro de este proceso, si no llegará un dato por cualquier razón, se cierra el documento como si hubiese llegado al tiempo establecido, con la intención de evitar que se creen errores en los archivos, es decir que dejen líneas vacías, y también para evitar que se pierda la información ya recolectada.

Una vez que se termina de crear el archivo, se hace la comprobación si existe conexión a internet, en caso de no tener conexión, el archivo se guarda en una carpeta específica y se vuelve al paso anterior creando un nuevo archivo, este proceso es repetitivo hasta que se encuentra conexión a internet; cuando se tiene la conexión, se verifica cuántos archivos existen guardados y se introducen en un arreglo para después subir los datos de estos archivos, uno por uno, al servidor mediante FTP. Cada vez que se va enviando un archivo, se elimina para no tener problemas de saturación de memoria en un futuro.

3.3 Servidor remoto

3.3.1 Montaje del servidor

Para montar el servidor se hizo uso de la plataforma Amazon Web Service (AWS), el proceso que se llevó a cabo para crear el Servidor Privado Virtual (VPS) es el siguiente:

- Creación de cuenta en Amazon AWS
- Creación de una máquina virtual con EC2
- Configuración de la máquina virtual (VPS)
 - Configuración de SSH y acceso a VPS
 - Configuración de IP estática
 - Configuración de VESTA para el web hosting
 - Creación de Dominio

Se creó una cuenta en Amazon AWS, Amazon ofrece gratis un año de servicio de prueba con recursos determinados por la empresa, la cuenta se crea con un correo electrónico y una contraseña. Adicionalmente, Amazon pidió una tarjeta de crédito, de la cual no



llegó ningún cobro, se utiliza para que terminando el año de prueba se empiece a cobrar, pero con la opción de recordar antes de hacer el cobro o deshabilitar este cargo.

Después de que se creó la cuenta, se accedió al panel de administración. En el panel de administración se encuentra Elastic Compute Cloud (EC2), que es el recurso para poder crear máquinas virtuales; es decir, con este recurso se creó el VPS. Cuando se accedió a la herramienta EC2, manda a una serie de pasos para crear el servidor, en el paso uno se muestran los diferentes tipos de sistemas operativos que se le pueden montar al servidor (ver fig. 3.7).

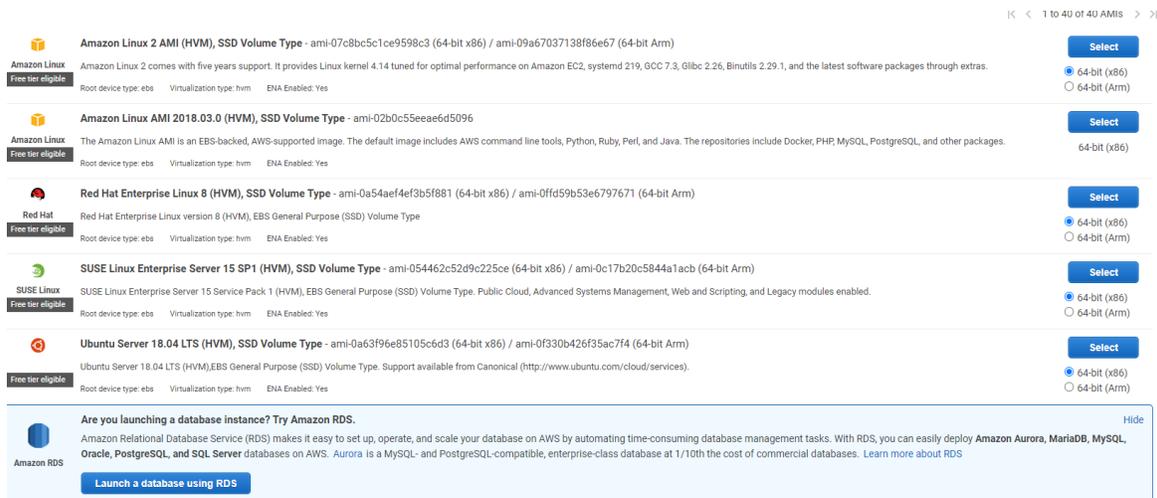


Figura 3.7 Lista de sistemas operativos EC2 de AWS.

Se eligió Ubuntu server, que es de las opciones de prueba gratis que maneja Amazon. En el paso 2 se despliega una lista de tipos de interfaces, se eligió la que viene por defecto por ser la versión gratis que da AWS. En el paso 3 se dejó tal cual están las configuraciones de instancia. En el paso 4 se elige el que viene por defecto (Propósito General) y en la capacidad se configuró a 15 Gb. En el paso 5 se dejaron por defecto los tags. En el paso 6 se configuró un grupo de seguridad, este grupo sirve para poder abrir los puertos por donde se comunica el servidor, se eligió la opción de crear un nuevo





grupo, luego se le asignó un nombre y una pequeña descripción, posteriormente se configuraron los puertos de la siguiente manera:

- En type, se seleccionó para que tipo de puerto se utilizará.
- En protocol, que protocolo utilizará.
- En port Range, se colocó el puerto que se utilizará.
- En source, se puso que direcciones IP tienen permitido conectarse.
- Y en Description, se puso una pequeña descripción.

Para cada uno de los puertos (22 para SSH, 8083 para Panel de control Vesta, 80 para HTTP, 443 para HTTPS y 21 para FTP) se le configuraron los parámetros anteriores, como se muestra en la figura 3.8.

Step 6: Configure Security Group

Assign a security group: Create a new security group
 Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP 190.195.30.145/32	SSH
Custom TCP	TCP	8083	Custom 0.0.0.0/0, ::/0	VESTA
Custom TCP	TCP	80	Custom 0.0.0.0/0, ::/0	HTTP
Custom TCP	TCP	443	Anywhere 0.0.0.0/0, ::/0	HTTPS
Custom TCP	TCP	21	Anywhere 0.0.0.0/0, ::/0	FTP

Add Rule

Cancel Previous Review and Launch

Figura 3.8 Puerto configurados en el Grupo de Seguridad AWS.

Para el último paso, solo mostró un resumen de toda la configuración, para terminar, se dio clic en launch y se abrió una nueva ventana, la cual es para crear una llave pública y privada para poder ingresar remotamente. Entonces, se seleccionó “Create a new key pair” y se le ingresó un nombre para poder reconocerla, se dio clic en “Download Key Pair” y enseguida descargó un archivo con extensión .pem. Se





concluyó dando “Launch Instances” y comenzó a cargar y poner en conexión el VPS. Cabe destacar que se tuvo especial atención en el archivo descargado (con extensión .pem), ya que se usó más adelante para conectarse de manera remota por SSH. Además, este archivo no se puede perder o compartir por seguridad del servidor. En la figura 3.9 se muestra la instancia o VPS funcionando y en línea.

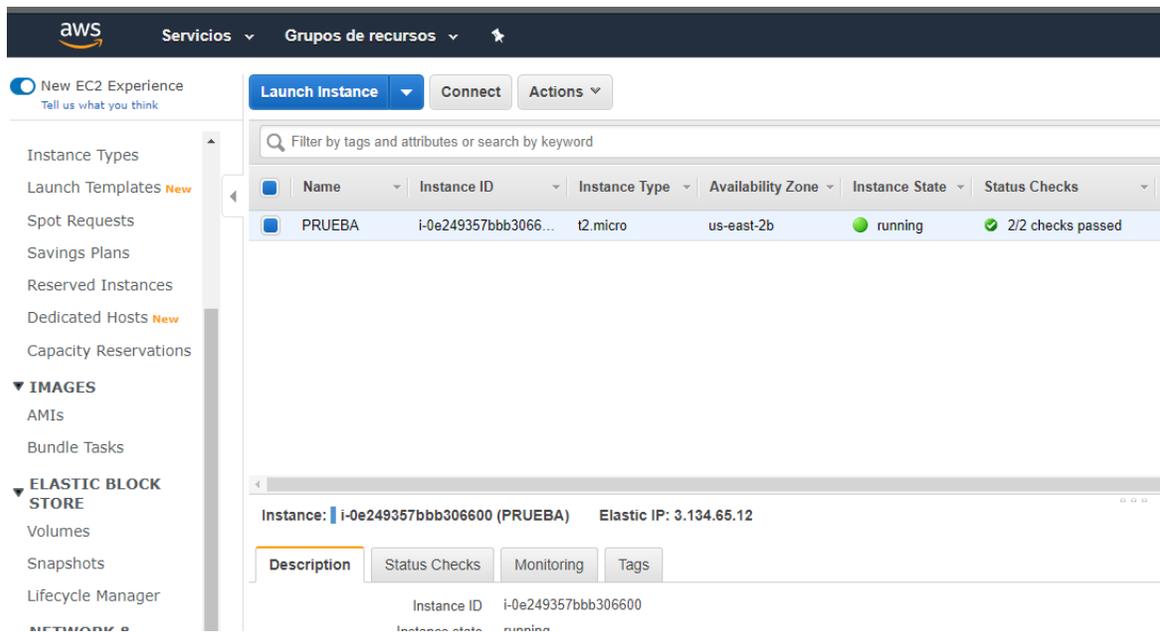


Figura 3.9 Panel de servicios y configuraciones de EC2.

Se configuró una IP estática del servidor, con el objetivo de que no cambie continuamente. Para poder dejar una IP estática en el servidor se procedió de la siguiente manera: en columna izquierda se muestra un menú en forma de lista donde se eligió “Elastic IPs”, luego se dio clic en “Allocate new Address”, luego se dio clic en “Allocate” y se cerró esa ventana. En la ventana de “Elastic IPs” se muestra ya la IP estática, se puede configurar para poder ponerle un nombre para no confundirse por si se llega a usar otra más adelante. En la parte superior se dio clic en el menú desplegable “Actions” y se seleccionó “Associate address”. Se desplegó una nueva ventana donde se seleccionó “Instance” y en “Instance” se eligió el nombre de la instancia creada





anteriormente, en “Private IP” se seleccionó la IP privada que antes se guardó y luego se dio clic en “Associate” para culminar. Entonces el Servidor ya tiene asociada la IP estática creada anteriormente.

Para poder acceder al VPS de forma remota es necesario conectarnos a través de SSH, esto se logra en Windows mediante el uso de Putty. Es un software que permite la comunicación con el servidor mediante una consola de comandos, donde se conecta por el puerto 22 del VPS. En la página oficial³ se puede descargar de manera gratuita Putty. Cuando se instaló Putty se crearon dos programas: uno es la terminal de comandos y el otro es Putty Key Generator. Primero se abrió Putty Key Generator, en la barra de herramientas se dio clic en “Conversions”, luego en “Import key” y en la ventana nueva, se seleccionó el archivo .pem previamente descargado, para finalizar en la ventana emergente se dio clic en “Save private key”. Cabe mencionar que estas acciones fueron realizadas para que putty pueda interpretar el archivo .pem; es decir, se convirtió a un archivo legible para Putty.

Se abrió Putty para configurar la conexión al servidor, en el campo de Hostname o IP se escribió la IP fija antes configurada del VPS y en el campo de puerto se puso el 22. En el panel izquierdo se dio clic en “SSH”, luego “Auth” y se seleccionó la llave convertida en el paso anterior. Se volvió a login y para no tener que repetir estos pasos en subsecuentes conexiones, en “Save Sessions” se colocó un nombre y se dio clic en “Save”. Para poder inicial la conexión se dio clic en Open y abre la terminal, empezó con un “login as:” que se refiere al nombre del usuario que se conectará. Se puso Ubuntu para referirse que entraba con ese usuario y la conexión fue exitosa. En la figura 3.10 se muestra la conexión al VPS desde Putty mediante líneas de comandos.

³ www.putty.org



```
root@rdecg: /
https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

67 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Fri May 22 04:11:29 2020 from 187.189.146.252
ubuntu@rdecg:~$ sudo su
root@rdecg:/home/ubuntu# cd..
cd.: command not found
root@rdecg:/home/ubuntu# cd ..
root@rdecg:/home# cd ..
root@rdecg:/# ls
backup      etc          lib64       proc        srv          vmlinuz
bin         home        lost+found  root        sys          vmlinuz.old
boot       initrd.img  media       run         tmp          vst-install-ubuntu.sh
deb_signing.key  initrd.img.old  mnt        sbin       usr          vst-install.sh
dev        lib         opt         snap       var
```

Figura 3.10 Acceso mediante SSH al VPS.

Con el propósito de facilitar las configuraciones y accesos al VPS y teniendo en cuenta que se usa como Hosting web, se decidió usar VESTA Panel. Esta interfaz permite configurar de manera gráfica servicios de Firewall, DNS, Mail y base de datos, entre otros; es decir, con esta herramienta se obtuvo un web hosting con todas las interfaces que incluye. Se puede ingresar la interfaz desde un navegador web para hacer pruebas como si estuviese de manera local, además de ser más intuitiva y cómoda que si se estuviera trabajando por línea de comandos desde Putty mediante SSH.

Para instalar VESTA Panel fue necesario ingresar desde su página oficial⁴. En esta página se pueden configurar las opciones de instalación, donde se configuró como se muestra la fig. 3.11, para después copiar la línea de comandos generada y ponerla desde Putty en el servidor, estas líneas de comandos son las que están situadas en el recuadro verde de la parte inferior de la figura. Cabe mencionar que para poner el “Hostname” fue

⁴ <https://vestacp.com>





necesario crear un dominio, en este caso con Hostinger por su bajo precio, ya que más fácil recordar un nombre que una dirección IP.

ADVANCED INSTALL SETTINGS

WEB nginx + apache	FTP vsftpd	MAIL no
DNS named	Firewall iptables + fail2ban	SOFTACULOUS yes
Additional Repository remi	File System Quota no	DB <input checked="" type="checkbox"/> MySQL <input type="checkbox"/> PostgreSQL
Hostname rdecg.tech	Email	Password

```
1 # Connect to your server as root via SSH
ssh root@your.server

2 # Download installation script
curl -O http://vestacp.com/pub/vst-install.sh

3 # Run it
bash vst-install.sh --nginx yes --apache yes --phpfpm no --named yes --remi yes --vsftpd yes --proftpd no --iptables
yes --fail2ban yes --quota no --exim no --dovecot no --spamassassin no --clamav no --softaculous yes --mysql yes --
postgresql no --hostname rdecg.tech
```

Generate Install Command

Figura 3.11 Comandos para instalar y configurar VESTA Panel.

En la figura 3.12 se muestra el panel de control Vesta con el que podemos acceder desde el navegador a nuestro servidor de manera gráfica con el usuario (admin) y la contraseña antes creada, esto por el puerto 8083 que se configura al crear el VPS.





The screenshot shows the Vesta control panel interface. At the top, there is a navigation bar with tabs for 'Planes', 'IP', 'Gráficos', 'Estadísticas', 'Registros', 'Actualizaciones', 'Cortafuegos', and 'Servidor'. Below this, there are several summary cards for different services: USUARIO, WEB, DNS, CORREO, BD, TAREAS, and RESPALDOS. Each card displays counts for active and suspended items. For example, the 'USUARIO' card shows 1 active user and 0 suspended. Below the summary cards, there is a search bar and a list of users. The 'admin' user is selected, and their details are shown, including their role as 'System Administrator', bandwidth usage (38 mb), disk usage (7 mb), and various system settings like email, SSH access, and IP addresses.

Figura 3.12 Panel de control Vesta desde navegador web.

Como ya se mencionaba, con la finalidad de poder ingresar al servidor, ya sea para administrar/configurar o para que el usuario pueda ingresar y visualizar la aplicación web, se decidió usar un dominio. Existen diferentes páginas, por ejemplo: Hostinger, GoDaddy, Hosttinet, por mencionar algunas, que ofrecen los servicios de Domain Name System (DNS), que simplificando, este sistema es capaz de convertir o asociar las IP (por ejemplo: 85.11.29.231) a nombres que podemos recordar.

Se decidió contratar por un año el Hostinger, por el bajo precio y el soporte que brindan. Una vez asociando el dominio a la IP del VPS, se puede ingresar con el nombre elegido y configurado desde un navegador web.

3.3.2 Gestión de archivos y visualización

Para poder seleccionar y visualizar los datos de los archivos que llegaron desde la Raspberry pi, se dividió en 3 funciones o acciones:

1. Mostrar fechas existentes de archivos.





2. Mostrar horas de la fecha seleccionada correspondiente al archivo.
3. Mostrar Gráfica del archivo seleccionado.

El algoritmo de la fig. 3.13 se utilizó para realizar la acción correspondiente número 1.

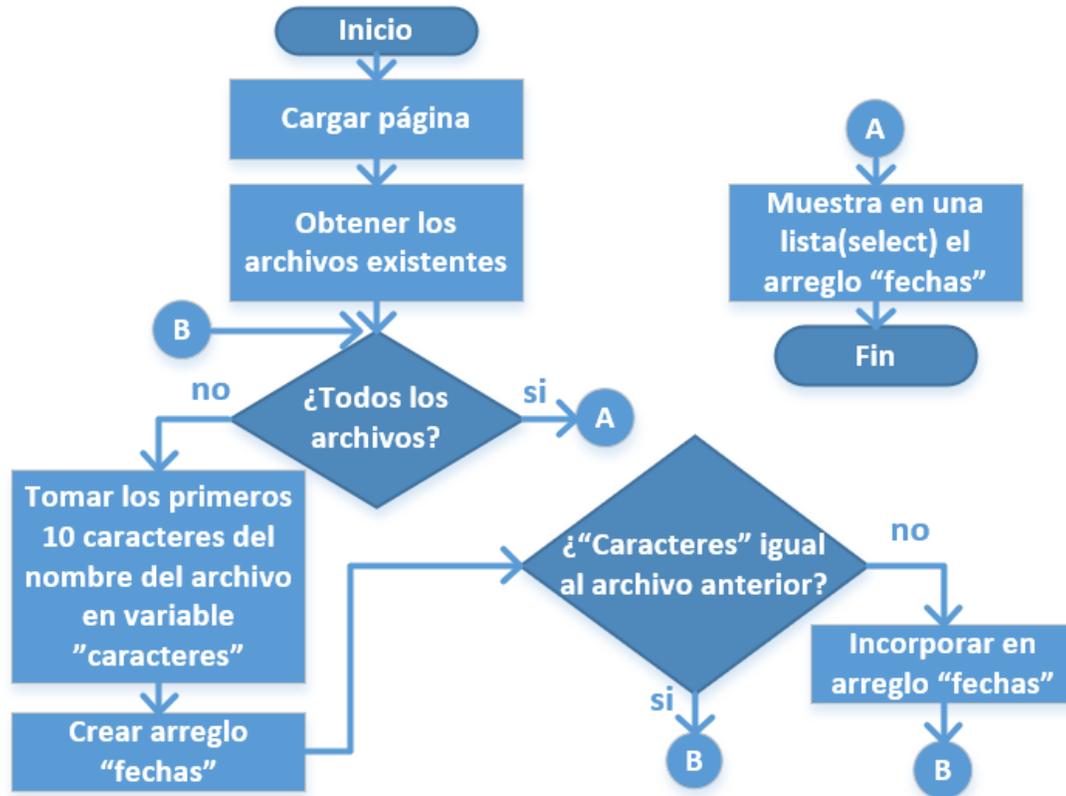


Figura 3.13 Diagrama de flujo del algoritmo Mostrar fechas existentes de archivos.

A continuación, se explica de forma más detallada el funcionamiento del algoritmo de la figura 3.13. Al entrar a la página web se verifica en la carpeta específica (de donde llegan los archivos provenientes de la Raspberry) cuantos archivos existen y se guardan en una variable de tipo arreglo; mediante un ciclo, los archivos son obtenidos de uno en uno tomando los 10 primeros caracteres del nombre del archivo seleccionado (estos caracteres son referentes a la fecha en que fueron creados); se crea una variable del tipo arreglo, en el que se guardan estos caracteres, y en dado caso de que se llegaran a repetir estos caracteres con los del archivo anterior, no se guarda y regresa al ciclo





mencionado. Cuando se termina el ciclo o bucle, el arreglo es presentado en forma de lista por HTML hacia el usuario, la cual contiene las fechas de los archivos existentes en el servidor.

El algoritmo de la fig. 3.14 se utilizó para realizar la acción correspondiente número 2 (Mostrar horas).

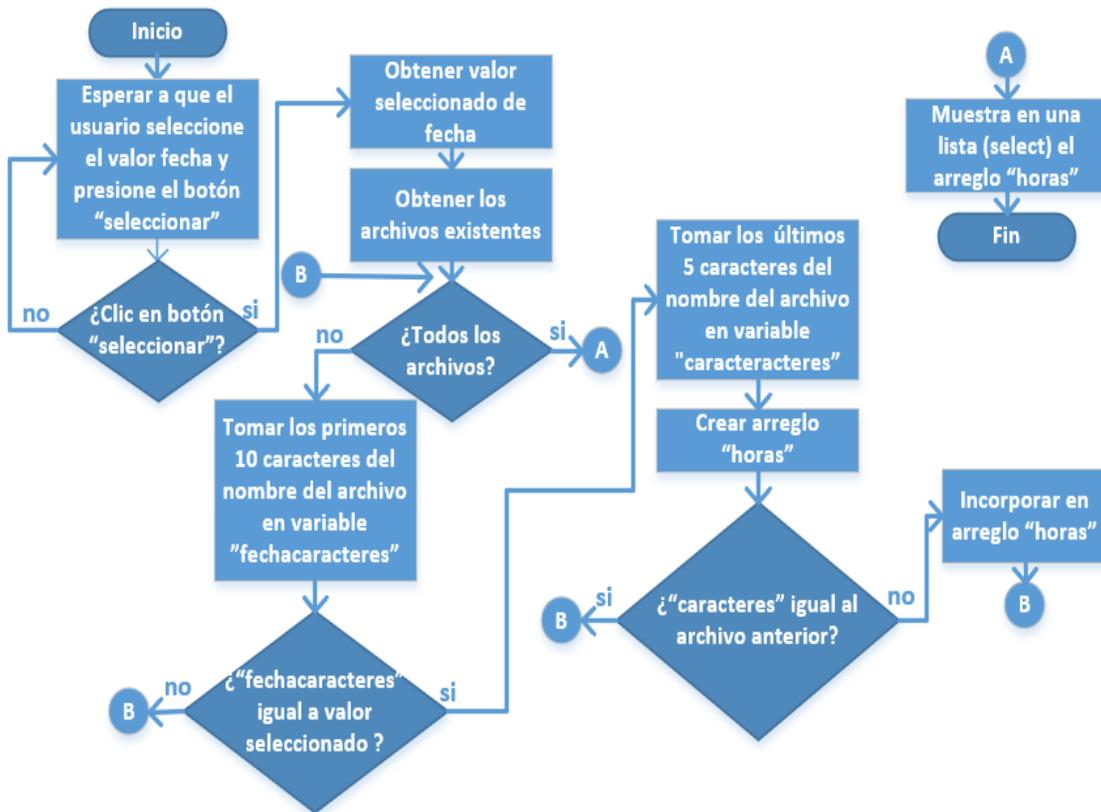


Figura 3.14 Diagrama de flujo del algoritmo que muestra las horas.

Comienza con la espera de la selección del valor fecha y del clic en el botón seleccionar; una vez se haya presionado el botón seleccionar, se guarda en una variable los caracteres de la fecha seleccionada; se obtienen los archivos que existen en la carpeta específica y se guardan en un arreglo, se verifica archivo por archivo que coincida con la hora seleccionada con los 10 primeros caracteres del archivo, si no coincide pasa al





siguiente archivo, pero si coincide, obtiene los 10 últimos caracteres del archivo (que representan la hora) y se van guardando en un arreglo. Cuando se terminan de verificar todos los archivos y guardar las horas de las fechas que sean iguales, son mostrados en forma de lista al usuario.

El algoritmo de la figura 3.15 se utilizó para realizar las funciones o acciones correspondientes del paso numero 3 (Mostrar la gráfica del archivo seleccionado).

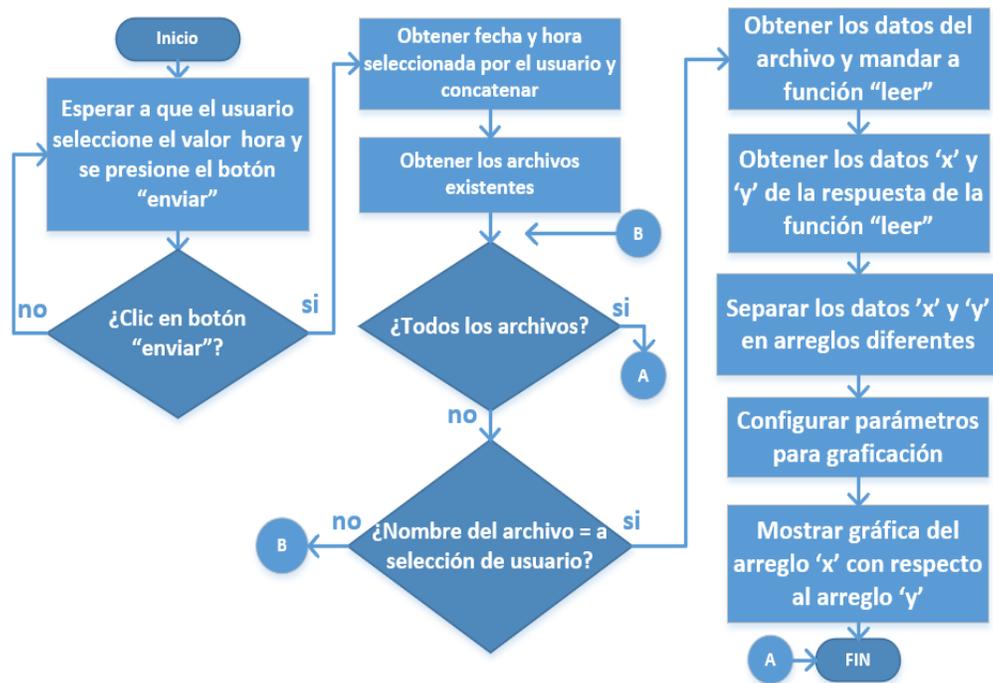


Figura 3.15 Diagrama de flujo del algoritmo que muestra la gráfica.

A continuación, se describe el algoritmo de la fig. 3.15. Se empieza esperando a que el usuario seleccione el valor de la hora y presione el botón enviar; cuando se da clic en el botón enviar, se obtiene la fecha y la hora seleccionada por el usuario y se concatena con la cadena ".txt" en una variable; se obtienen los archivos provenientes de la Raspberry que existen en la carpeta del servidor y se van tomando para verificar si el nombre del archivo es igual a la concatenación seleccionada por el usuario; si son iguales, se obtiene o selecciona ese archivo y se envía por método POST hacia una





función llamada “leer” (en la figura 3.16 se muestra la función “leer”). La función “leer” retorna como respuesta una agrupación de dos arreglos donde se obtienen los datos a graficar ‘x’ e ‘y’; luego son enviados a una función que separa estos arreglos quedando de forma independiente; se configuran los parámetros para la gráfica (tipo de gráfica, tipo de línea, color de línea, etc.); por último, se muestran los valores del arreglo ‘x’ que son graficados con respecto del arreglo ‘y’ en una división de la página, hay que destacar que se debe esperar un aproximado de 10 segundos para poder visualizar, esto debido a la gran cantidad de datos que se maneja y capacidad del servidor.

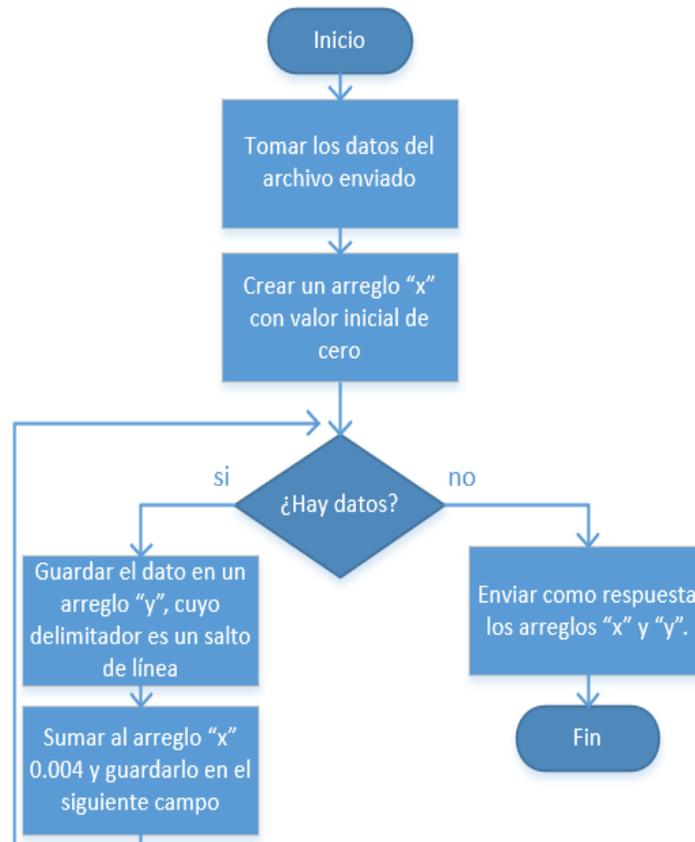


Figura 3.16 Diagrama de flujo del algoritmo de la función “leer”.



A continuación, se describe el proceso que realiza la función “leer” de la fig. 3.16. Este proceso empieza tomando el archivo que se envía por método POST desde el programa principal; se toman los datos de ese archivo y hasta que se tomen todos los datos, se van guardando en un arreglo ‘y’ (los datos) y para saber dónde acaba cada valor, se propuso que por delimitante de cada dato se agregará un salto de línea; luego se hace una suma de 0.004 en 0.004 y cada valor se guarda en un arreglo ‘x’ (el tiempo). Cuando se hayan tomado todos los datos, esta función envía como respuesta el grupo donde contiene los arreglos ‘x’ y ‘y’. Estos arreglos son enviados y recibidos por medio de “JSON”, esta herramienta permite poder enviar y recibir arreglos.

Cabe mencionar que en dado caso que no se llene un campo de selección del formulario y se presione alguno de los dos botones (seleccionar o enviar), se muestra un mensaje emergente de alerta al usuario y le dice que es necesario llenar los campos.

Como aparecen todos los datos de 30 minutos, de primera impresión pareciera que se ven puras líneas verticales, es por esto que se cuenta con el panel de herramientas (ver figura 3.17) que se describe a continuación.



Figura 3.17 Panel de Herramientas.

Esta barra de herramientas cuenta con captura en imagen en formato .PNG (), alejar y acercar la imagen ( ), auto escala (), regresar al punto original (), mover la gráfica de izquierda a derecha () y seleccionar solo la parte que se desee visualizar (), esta última es la opción por default al cargar la gráfica. Para trabajar con la gráfica, se da clic y sin soltar se selecciona la parte a revisar (esta se va marcando de un color más oscuro) y cuando se suelta el botón del ratón, automáticamente se ve de forma ampliada lo que se seleccionó. Además, al pararse sobre un punto sobre la gráfica se muestra el valor de dicho punto que puede servir para posterior análisis.





Capítulo 4. Pruebas y análisis de resultados

Para realizar las pruebas y teniendo en cuenta que se ocupó un modelo evolutivo de prototipado, esta sección se divide conforme a las pruebas y resultados obtenidos en cada prototipo propuesto.

4.1 Primera iteración

Objetivo.

El primer prototipo tenía como objetivo la implementación de la medición y visualización básica de las señales electrocardiográficas. Mediante el desarrollo del mismo se iniciaría con el aprendizaje de los conceptos relacionados con la adquisición, procesamiento y visualización de las señales biomédicas.

Requerimientos.

Los requerimientos para el primer prototipo fueron:

- Que el prototipo capte las señales eléctricas del corazón a través de los electrodos.
- Se realice el acondicionamiento de la señal mediante el filtrado y amplificación analógica de la misma.
- Se realice la conversión de la señal analógica a una señal digital y su transmisión serial.
- Visualizar la señal en la pantalla de una computadora

Pruebas y resultados.

Se hicieron pruebas usando electrodos de tipo no invasivos, con gel conductor y para el filtrado y acondicionamiento de la señal, se usó la tarjeta AD8232; para hacer la conversión analógico-digital se usó la tarjeta de desarrollo Arduino, que mediante transmisión serial, transmitió las mediciones a una computadora para poder visualizar los datos mediante una gráfica con la ayuda de Matlab.

Como resultado se obtuvieron las primeras mediciones de las señales biomédicas que se muestran en la fig. 4.1. En la cual, se observa una gráfica del electrocardiograma,





visualizada y procesada con Matlab, es decir, que se aplicó la Transformada Rápida de Fourier para ver las componentes de frecuencia, además del oscilograma (que en este caso corresponde al electrocardiograma).

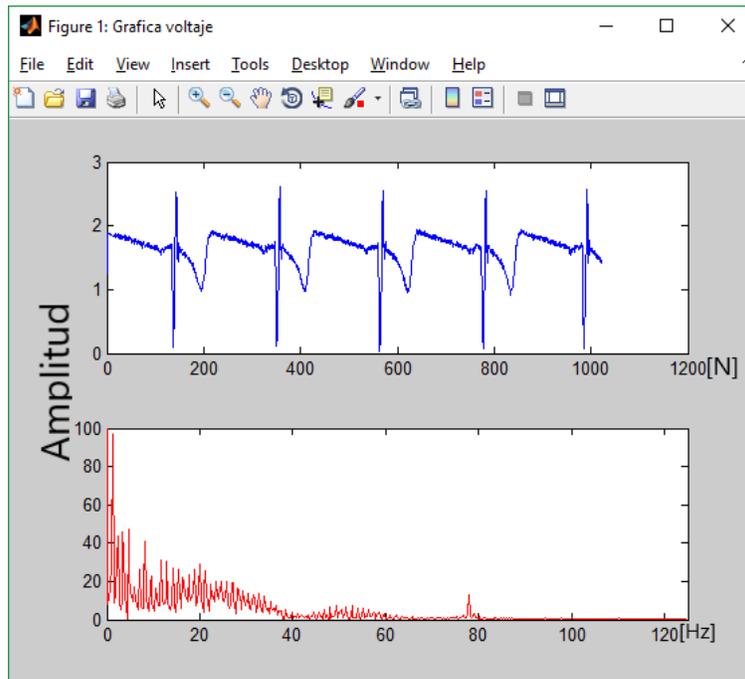


Figura 4.1 Oscilograma y espectro de frecuencias.

Retroalimentación y trabajo a futuro.

Para este primer prototipo se pudo estudiar parte de la teoría sobre la señal eléctrica del corazón, el electrocardiograma; la tarjeta AD8232 y el funcionamiento de la tarjeta de desarrollo Arduino.

Se lograron los requerimientos previos, haciendo el oscilograma y el espectrograma de las señales adquiridas de una persona.

A pesar de los resultados obtenidos, era necesario ajustar algunas cuestiones encontradas.

- Probar con una alternativa de software libre.
- Estudiar e implementar frecuencias de muestreo.





- Guardar los datos obtenidos de la etapa de adquisición.

4.2 Segunda iteración

Objetivo.

Para este segundo prototipo se propuso estudiar las frecuencias de muestreo y lo relacionado en la conversión analógico digital, crear un algoritmo para implementarlo al microcontrolador, revisar los tiempos que le toma al microcontrolador realizar cada una de las acciones para tener una frecuencia de muestreo, y cumplir con el criterio de Nyquist.

Requerimientos.

Los requerimientos del segundo prototipo fueron:

- Medir y analizar el tiempo que tarda el microcontrolador en realizar una digitalización, la transmisión serial y ambas acciones
- Realizar un algoritmo para tener una frecuencia de muestreo adecuada para obtener componentes de frecuencia de hasta 125 Hz; esto es, una frecuencia de muestreo de 250 Hz (cada 4 milisegundos).

Pruebas y resultados.

De manera general el proceso adquisición de señales se llevó a cabo en las siguientes etapas:

1. Lectura y digitalización de la señal analógica
2. Transmisión de la señal digital por el puerto serial
3. Tiempo de espera para realizar la siguiente medición.

En este apartado se analizaron los tiempos necesarios para realizar el proceso de adquisición de señales y se relacionaron con las frecuencias máximas de muestreo para comprobar si son congruentes con las frecuencias requeridas para la reconstrucción de la señal a partir de las muestras.



Para la parte de transmisión se usaron 8 bits a 115200 baudios, esto se configuró en el PC para que no existieran cuellos de botella en la transmisión entre el Arduino y la computadora. En la fig. 4.2 se observa la configuración de las propiedades del puerto serial (COM) en Windows, esto se logró entrando en administración de equipo, administración de dispositivos y en puertos COM seleccionando el puerto en el que el Arduino se encontraba conectado.

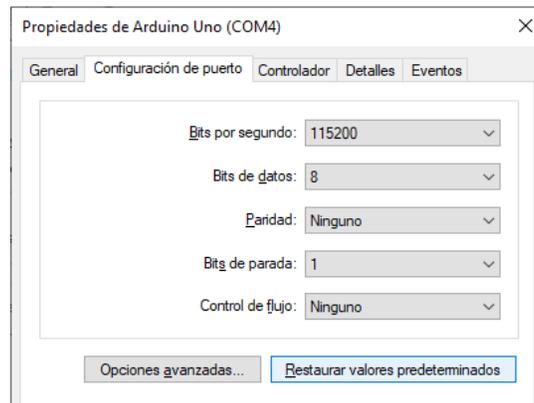


Figura 4.2 Configuración de transferencia serial en puerto de la computadora.

Para realizar los experimentos se siguió el algoritmo de la fig. 4.3.

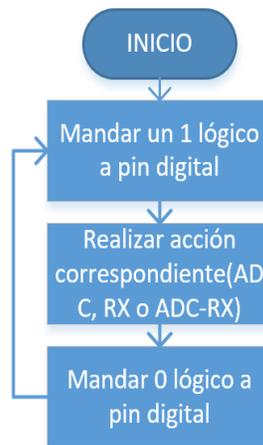


Figura 4.3 Algoritmo para realizar los experimentos de medición de tiempos.





4.2.1 Primer experimento: Tiempo de conversión analógico digital (ADC)

Con base al diagrama mostrado en la fig. 4.3 y con el objetivo de medir el tiempo de conversión del ADC del Arduino se realizó la siguiente secuencia:

1. Se manda un 1 lógico (estado alto) por el pin digital del Arduino. La transición a un voltaje alto de esta terminal permitirá identificar el inicio del proceso.
2. Se recibe una señal por el pin analógico la cual será digitalizada. Durante el tiempo que tarde este proceso la señal digital del paso anterior se mantiene en alto.
3. Se manda un 0 lógico (estado bajo) por el pin digital del Arduino. La transición a nivel bajo de este pin indica el final del proceso.
4. Se repiten los pasos anteriores.

En la fig. 4.4a) se observa el circuito electrónico. El pin A0 se conectó a un potenciómetro con el que se varía la señal analógica. A la salida del pin 8 se conectó el osciloscopio donde se visualizaron los cambios en la señal digital producidos por los pasos 1 y 3.

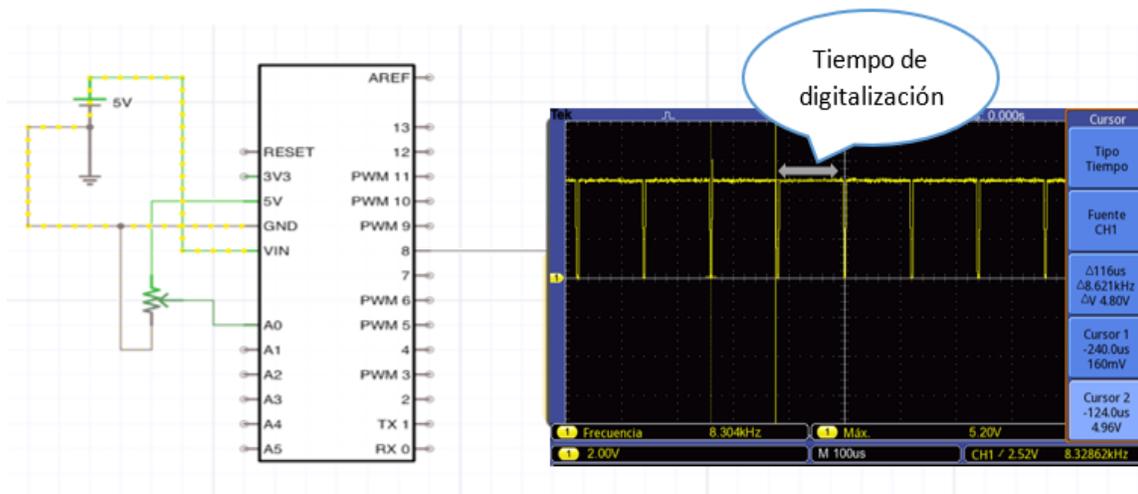


Figura 4.4 a) Circuito electrónico.

Figura 4.4 b) Osciloscopio prueba ADC.

En la fig. 4.4b) se observa que el proceso de conversión analógico a digital requiere de un tiempo de 116 microsegundos. Esto significa que la frecuencia máxima con la que se podrían realizar conversiones de analógico a digital sería de 8.6 KHz. Es importante mencionar que el tiempo de conversión es independiente del valor analógico a la entrada y no varía el tiempo o frecuencia.





En la fig. 4.4b) se observa que la transición del valor bajo al alto es muy corta, esto es debido a que el tiempo requerido para reiniciar el ciclo de medición es muy reducido.

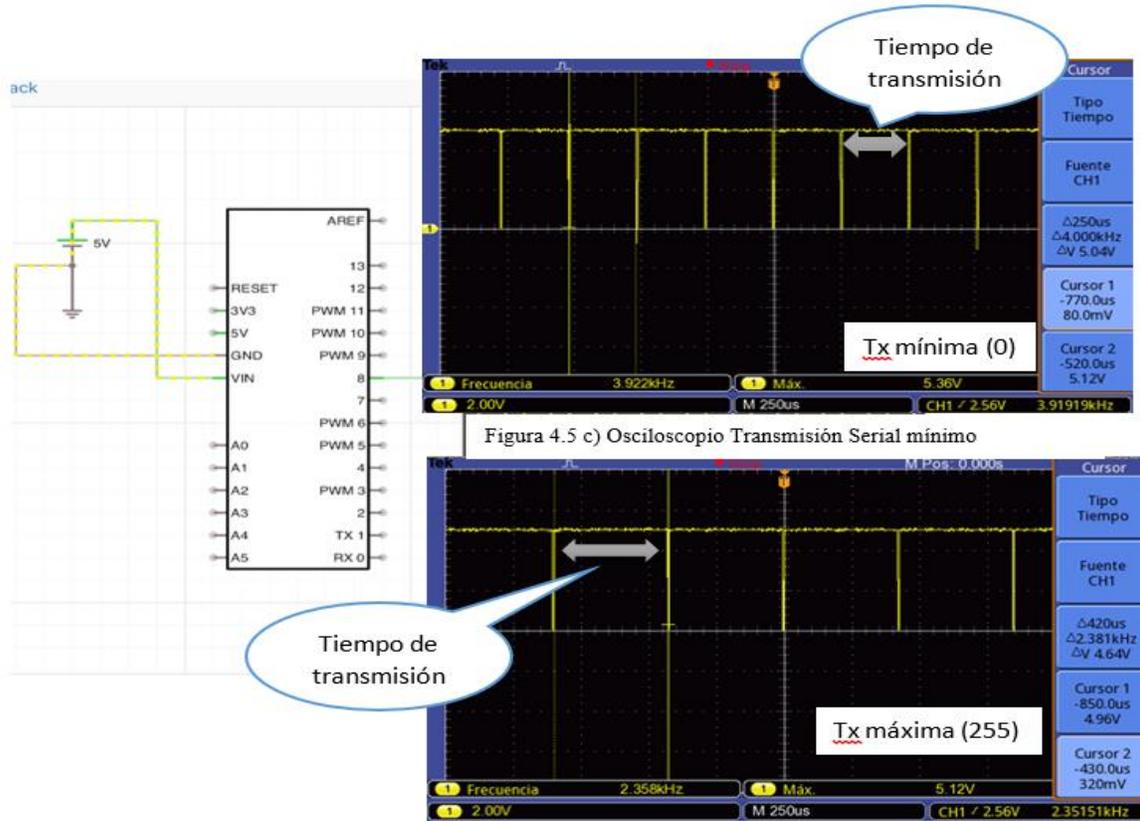
4.2.2 Segundo experimento: Tiempo de transmisión serial (Tx)

Con el objetivo de medir el tiempo de transmisión serial, se usa el mismo algoritmo (fig. 4.3) en donde se midió el tiempo de transmisión serial de un byte. A continuación, se describen los pasos:

1. Se manda un 1 lógico (estado alto) por el pin digital del Arduino. La transición a un voltaje alto de esta terminal permitirá identificar el inicio del proceso.
2. Se manda por el puerto serial un byte con el equivalente a $0_{10}=00000000_2$ y luego el valor $255_{10}=11111111_2$, que serían los posibles valores máximos y mínimos. Durante el tiempo que tarde este proceso la señal digital del paso anterior se mantiene en alto.
3. Se manda un 0 lógico (estado bajo) por el pin digital del Arduino. La transición a nivel bajo de este pin indica el final del proceso.
4. Se repiten los pasos anteriores.

Entonces, de manera similar al experimento anterior, el tiempo en hacer una transmisión serial se pudo medir a través de tiempo en alto de la señal digital que se monitoreó con un osciloscopio.

En al fig. 4.5a) se observa el diagrama electrónico del experimento. Inicialmente, por programación, se ingresaron los valores a transmitir. A la salida del pin 8 se conectó el osciloscopio donde se visualizaron los tiempos entre las líneas verticales.



Como se puede observar en la fig. 4.5 b), el tiempo de transmisión del 0_{10} es de 250 microsegundos. Con el objetivo de verificar si el tiempo de transmisión depende del dato a transmitir se repitió el experimento transmitiendo el valor $255_{10}=11111111_2$. Para este experimento se obtuvo un tiempo de transmisión serial de 420 microsegundos (ver fig. 4.5 c).





4.2.3 Tercer experimento: Tiempo de conversión analógico digital y transmisión serial (ADC-Tx)

Con el objetivo de medir el tiempo requerido para realizar la conversión analógico-digital de una señal y transmitirla a través del puerto serial se realizó el siguiente procedimiento (mismo algoritmo de la fig. 4.3):

1. Se manda un 1 lógico (estado alto) por el pin digital del Arduino. La transición a un voltaje alto de esta terminal permitirá identificar el inicio del proceso.
2. Se recibe una señal por el pin analógico la cual será digitalizada. Durante el tiempo que tarde este proceso la señal digital del paso 1 se mantiene en alto.
3. Se transmite por el puerto serial la señal digitalizada. Durante el tiempo que tarde este proceso la señal digital del paso 1 sigue en alto.
4. Se manda un 0 lógico (estado bajo) por el pin digital del Arduino. La transición a nivel bajo de este pin indica el final del proceso.
5. Se repiten los pasos anteriores.

En la fig. 4.6a) se observa el circuito electrónico. El pin A0 se conectó a un potenciómetro con el que se varia la se señal analógica y se mandó por el puerto serial. A la salida del pin 8 se conectó el osciloscopio donde se visualizaron los cambios en la señal digital producidos por los pasos 1 y 4.

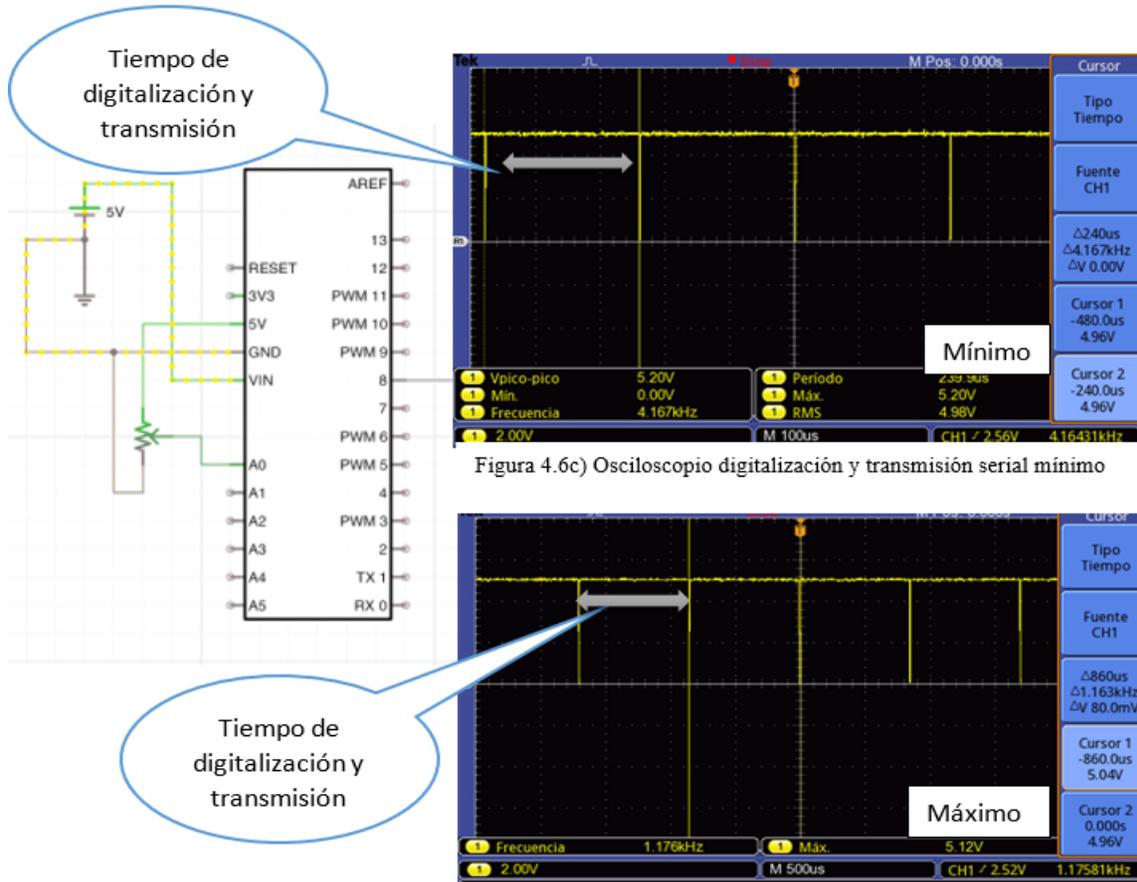


Figura 4.6c) Osciloscopio digitalización y transmisión serial mínimo

Figura 4.6 a) Circuito electrónico.

Figura 4.6 b) Osciloscopio digitalización y transmisión serial máximo.

Para este experimento se obtuvieron los tiempos de 240 microsegundos para un valor máximo de digitalización junto con la transmisión serial, y 860 microsegundos para un valor mínimo de digitalización y transmisión serial.

Se muestra en la Tabla 4.1 los tiempos obtenidos para cada experimento con su frecuencia, con la finalidad de poder visualizar mejor estos datos obtenidos.





Tabla 4.1 Tiempos de experimentos.

Arduino		
	Tiempo	Frecuencia
ADC min	116 us.	8.621 KHz
ADC max	116 us.	8.621 KHz
Tx 0	250 us.	4.000 KHz
TX 255	420 us.	2.381 KHz
ADC + TX min	240 us.	4.167 KHz
ADC + TX max	860 us.	1.163 KHz

Se pudo observar en los experimentos realizados, que en el caso de tener un valor máximo de digitalización y transmisión serial, podría llegar a tardar hasta casi otro milisegundo ocasionando alteraciones en la tasa de muestreo deseada.

Es por esto que, para adecuar el tiempo a cada 4 milisegundos se usó la función millis de Arduino. Esta función permite obtener el tiempo en milisegundos desde que la placa Arduino es energizada.

Con la finalidad de conseguir un muestreo cada 4 milisegundos (250 Hz), sin que afecte el tiempo de la digitalización y transmisión serial, se realizó el algoritmo de la fig. 3.5.

En la fig. 4.7 se presenta una barra para comprender mejor como se engloba el tiempo con el algoritmo de la fig. 3.5.



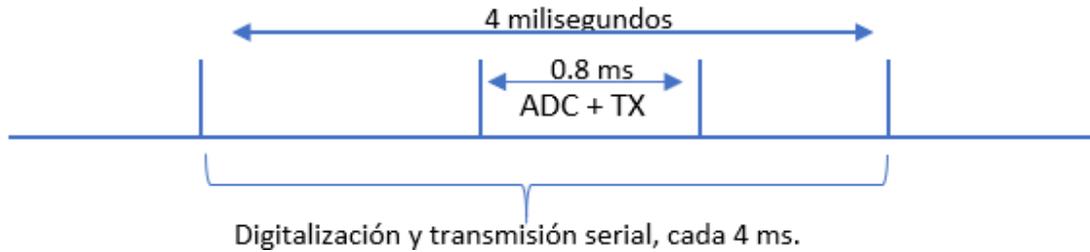


Figura 4.7 Globalización de tiempos con algoritmo propuesto.

En donde, todo el proceso, que implica la digitalización, transmisión de datos y espera transcurre cada 4 milisegundos, teniendo la certeza que la tasa de muestreo será de 250 Hz estables.

Los resultados obtenidos al enviar datos con este algoritmo se visualizaron en la fig. 4.8. Se observó que efectivamente cada 4 milisegundos se realiza la digitalización y transmisión, la cual al igual que en las pruebas anteriores se mide con el nivel alto y bajo que sale del pin 8 del Arduino hacia un osciloscopio.

Cabe destacar que, aunque sea un valor máximo o un mínimo no presentó cambio alguno en la frecuencia de muestreo, esto gracias al algoritmo propuesto y que, sin importar el valor de la digitalización y transmisión, siempre lo hizo cada 4 milisegundos.

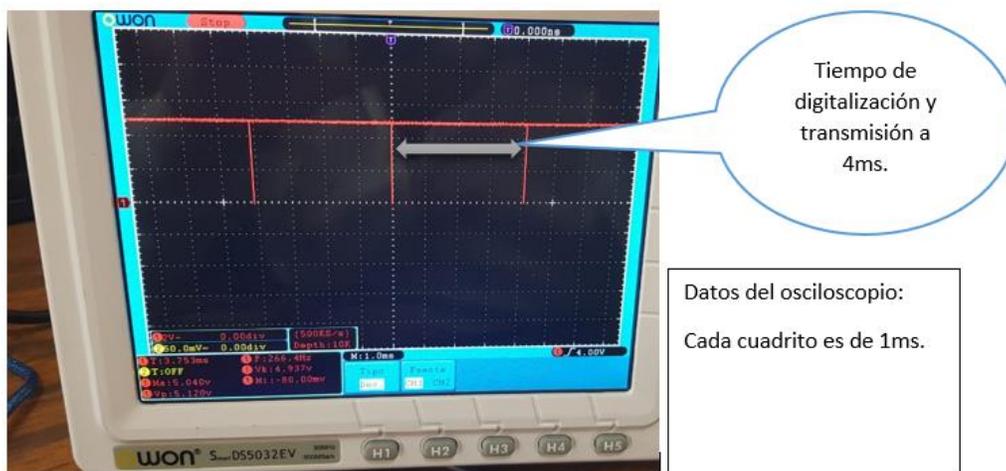


Figura 4.8 Tiempo de algoritmo propuesto a 4 milisegundos.



Retroalimentación y trabajo a futuro.

En estas pruebas se pudo observar que es importante el tiempo que se tarda en adquirir la señal, esto con la finalidad de no hacer sobre muestreo o submuestreo de la señal, lo cual afecta la frecuencia de muestreo que se desea (250 Hz), pudiendo afectar la reconstrucción de la onda que se digitaliza.

A pesar de la gran importancia y solución del problema de muestreo es necesario:

- Buscar un método para que las mediciones se guarden.
- Conseguir un método de transmisión hacia un visualizador en un navegador web.
- Que el sistema sea echo en software de libre acceso.

4.3 Tercera iteración

Objetivo.

Para esta iteración se necesitaba un sistema de hardware y software necesario para poder guardar los datos adquiridos y estudiar algún método para poder mandarlo hacia un servidor, Así mismo ver e investigar los tipos de servidores que se podrían usar.

Requerimientos.

- Tener un método para poder guardar los datos adquiridos
- Que este hardware pueda enviar los datos a través de internet
- Buscar opciones de servidores y elegir el más conveniente para el sistema

Pruebas y resultados.

Como el objetivo de esta iteración era buscar un hardware y/o software adecuado para poder guardar los datos y trasmitirlos por internet, de primer momento se pensó en la forma de transmitir dato por dato a través de internet; se pensó en usar una tarjeta de desarrollo Wemos, la cual cuenta con el microcontrolador ESP8266. La idea era poder tomar los datos adquiridos y mandarlos mediante del protocolo MQTT (Message Queing Telemetry Transport, por sus siglas en ingles), muy utilizado para el internet de las cosas.



De primer momento las pruebas de tiempo para la digitalización arrojaban resultados igual que las de la tarjeta Arduino, por lo que no afectaría la frecuencia de muestreo. Por otro lado, se analizó el protocolo MQTT y no satisfacía las necesidades para poder mandar los datos por ese protocolo, ya que en dado caso de que se llegara a quedar sin internet en la localidad donde se estuviera haciendo las mediciones o el servidor estuviera fuera de línea, los datos adquiridos datos podrían perderse.

También surgieron las incógnitas del que pasaría si se llegara a quedar sin suministro eléctrico, se podría perder la información. Por esto se hicieron pruebas con la tarjeta WeMos y un lector de memorias microSD, pero se tuvo problemas con las frecuencias de muestreo, ya que excedía el tiempo al guardar los datos en la memoria microSD, y no satisfizo el criterio de Nyquist.

Se llegó a la conclusión de usar una microcomputadora, eligiendo la Raspberry pi 3B+ por su accesibilidad en la compra y por su inmensa comunidad, que, en caso de tener alguna problemática, existe mucho material de apoyo para solucionarlo y una comunidad extensa que pude servir para investigar soluciones y apoyos.

Se hicieron pruebas desconectado el Arduino de la Raspberry, logrando que no realizaría ninguna acción, solo se manda un mensaje que dice que no existe un dispositivo conectado (ver fig. 4.9). Cuando se conecta el Arduino, se comienza con la creación del documento y a guardar los datos provenientes del puerto serial.



```
File Edit Shell Debug Options Window Hel
No hay dispositivo conectado
```

Figura 4.9 No existe dispositivo conectado por puerto serial.

Otra prueba fue dejar sin conexión a internet a la Raspberry, y como se puede observar en la fig. 4.10 se van guardando los archivos, se verificó que no se perdieran y que se fueran guardando en la carpeta que se especificó; lo mismo ocurre si el servidor se encuentra inactivo o fuera de línea; es decir, que mientras no exista una conexión entre la Raspberry y el servidor, se guardarán los archivos hasta que se puedan conectar.

Se realizó otra prueba donde se deja sin suministro eléctrico a la microcomputadora, en este caso se guarda el documento hasta la última la medición que le llegó por el puerto serial. En otras palabras, el archivo se cierra con los datos guardados hasta la falta de suministro de energía, cuando se reanuda la energía y se vuelve a usar el programa, se crea otro archivo nuevo, pero manteniendo el que se quedó antes de la falta de suministro eléctrico y cuando se termina de introducir todos los datos en este archivo nuevo es enviado el actual y el que se cerró por falta de energía, Aunque el tamaño es inferior al tiempo establecido para los paquetes, no es un problema para la trasmisión y posterior visualización (ver fig. 4.10).



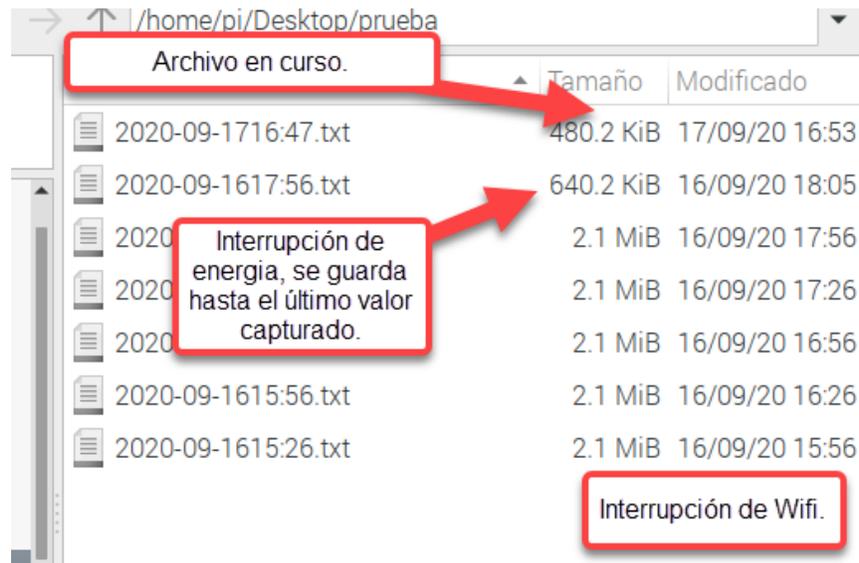


Figura 4.10 Prueba de archivos guardados.

A la par que se creaba el algoritmo que gestionaría el almacenamiento y envío de los documentos, se fue investigado la mejor opción para el servidor, este tenía que ser conveniente para la recepción de archivos provenientes de la Raspberry y para graficarlos. Se tomó en cuenta que se necesitaba un servidor dedicado para poder administrar correctamente al sistema. Para poder montarlo se necesitaba, en primer lugar, un proveedor de servicio de internet que brinde una Dirección IP fija o bien contratar un servicio que resuelva este primer problema. Además, se tendría que comprar o adquirir el equipo de cómputo o servidor, con las características necesarias para alojar el sistema web. Esto hacía que se necesitasen más recursos y aumentararan los costos para la realización del proyecto.

Se estudiaron y verificaron las posibles alternativas que se pudieran utilizar, eligiendo la plataforma Amazon Web Services; es gratis por un año y cuenta con las herramientas necesarias e ideales para acceso, administración y visualización que requería el sistema web. Además, se verificó que tuviera un amplio soporte por si se llegaran a tener dificultades a la hora de montar el servidor y el sistema web.



Teniendo ya montado el Servidor Privado Virtual sobre la plataforma, se realizaron pruebas para corroborar que los archivos enviados desde la microcomputadora llegaran al servidor y manteniendo todos los datos; es decir, que estuvieran guardados en el servidor tal cual se mandaron. En la fig. 4.11 se muestra la conexión al servidor mediante protocolo SSL con Putty, donde se ingresó a la carpeta donde se guardaron los archivos recibidos, y a su vez, se verificó que estuvieran todos los datos de los archivos enviados.

```
root@rdecg: /home/admin/web/rdecg.tech/public_html/archivos
root@rdecg:/home/admin/web/rdecg.tech# ls
cgi-bin document_errors logs private public_html public_shtml stats
root@rdecg:/home/admin/web/rdecg.tech# cd public_html/
root@rdecg:/home/admin/web/rdecg.tech/public_html# ls
DS_Store archivos css directorios.php img index.php leer.php librerias
root@rdecg:/home/admin/web/rdecg.tech/public_html# cd archivos/
root@rdecg:/home/admin/web/rdecg.tech/public_html/archivos# ls
2020-07-1009:30.txt 2020-08-1323:20.txt 2020-08-2119:13.txt
2020-07-1016:15.txt 2020-08-1506:33.txt 2020-08-2119:43.txt
2020-07-2216:39.txt 2020-08-1509:30.txt 2020-08-3112:20.txt
2020-07-2221:05.txt 2020-08-1516:15.txt 2020-08-3112:53.txt
2020-07-2323:20.txt 2020-08-1612:20.txt 2020-09-0120:19.txt
2020-07-2821:05.txt 2020-08-1616:34.txt 2020-09-0120:49.txt
2020-07-2916:39.txt 2020-08-1617:30.txt 2020-09-1613:56.txt
2020-07-2923:20.txt 2020-08-1710:30.txt 2020-09-1614:26.txt
2020-08-0812:20.txt 2020-08-1711:15.txt 2020-09-1614:56.txt
```

Figura 4.11 Prueba de archivos guardados en el servidor.

Retroalimentación y trabajo a futuro.

En la realización de pruebas se tuvieron problemáticas con el nombre del archivo, ya que existen caracteres especiales que afectaban para guardar y ver el contenido desde el servidor. Es por esto que se decidió poner guiones medios (-) en lugar de barra (/) para separar el año, mes y día, quitando el espacio entre fecha y la hora, y separando con dos puntos (:) los minutos. Por ejemplo, el texto asignado al nombre del archivo quedó de la siguiente manera: 2020-07-1009:30.txt.

Con esto ya se pudo verificar que efectivamente se guarden los archivos y se consiguió verificar que los datos contenidos fueras los que se enviaron desde la microcomputadora.

Ahora era necesario en la siguiente iteración:





- Crear un algoritmo para gestionar y guardar los archivos en una base de datos
- Crear e implementar una interfaz para poder visualizar los datos de manera grafica

4.4 Cuarta iteración

Objetivo.

Ya teniendo el servidor privado virtual y comprobando que le llegaran los archivos con la información completa, era necesario instalar una base de datos que guardara los datos provenientes de los archivos, también era necesario que se pudieran visualizar esos archivos en una aplicación web.

Requerimientos.

- Crear una base de datos para almacén de los archivos
- Crear una ampliación web donde se visualicen los datos

Pruebas y resultados.

Se usó lenguaje MySQL para gestionar la base de datos usando el programa Heidi SQL que fue más fácil de entender por su interfaz intuitiva. En la fig. 4.12 se muestra la página principal, al cargar esta página el algoritmo verificaba si había archivos existentes y las guardaba en la base de datos, cuando se terminaba de pasar todos los datos del archivo, era eliminado para no causar conflicto de datos repetidos cuando se volviera a cargar la página.

Se tenía que hacer clic en la parte azul (“aquí”) para que se dirigiera a la siguiente página donde se visualizaba la gráfica.



Se cargaron los archivos existentes a la base de datos.

Haga clic [aquí](#) para ver las gráficas.

Figura 4.12 Página principal que carga los archivos a la base de datos.

En la página donde se podía visualizar el electrocardiograma, cargaba todos los datos contenidos en la base y los graficaba con respecto a un campo auto incrementable que hacía la función de la intersección con el eje X. En la fig. 4.13 se muestra como quedaba la gráfica, cabe mencionar que la herramienta comentada en el capítulo anterior para graficación se implementó en esta iteración, por lo que las herramientas para poder enfocar la gráfica fueron las mismas.

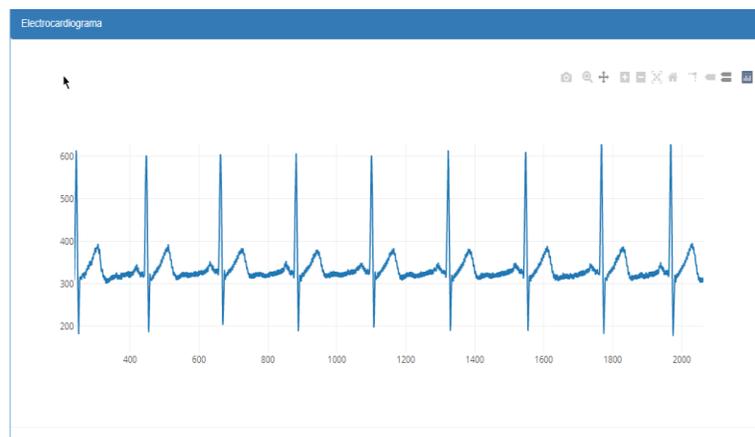


Figura 4.13 Gráfica del electrocardiograma en la página web.



Retroalimentación y trabajo a futuro.

A pesar de haber conseguido guardar los datos y visualizarlos, se tuvieron varios inconvenientes, como en las pruebas se usaron paquetes de datos de 2 minutos equivalentes a 30,000 datos, no se notaron problemas, pero a la hora de realizar las pruebas con archivos de 5 minutos en adelante era muy complicado e incómodo poder visualizar la zona deseada.

Es por esto que se decidió que el sistema web, en la siguiente iteración, tuviera las opciones de elegir la fecha y hora que se desee visualizar, para que así el usuario o especialista le sea más fácil buscar el periodo de mediciones que desee visualizar.

4.5 Quinta iteración

Objetivo.

Esta iteración tenía como objetivo resolver el problema de la iteración anterior. En otras palabras, se tenía que incluir en el sistema web una lista de opciones con las fechas y horas de las mediciones existentes, para que el usuario seleccionara la que quería ver.

Requerimientos.

- Que se puedan elegir la fecha de las mediciones existentes
- Que se seleccione el horario que se desee ver y se grafique solo ese horario requerido
- Conservar la aplicación y herramientas de graficación de la iteración anterior

Pruebas y resultados.

Para poder mandar los campos que usaríamos para saber la fecha y hora, era necesario que en los primeros campos del archivo creado por la Raspberry contuviera estos parámetros de fecha y hora. Siguiendo el principio de la iteración anterior, la página principal queda casi igual, al terminar de cargarla sube los datos de los archivos existentes a la base de datos para después eliminarlos, pero ahora guardando en



campos de fecha y hora en la base de datos lo que se le agregó desde la Raspberry. En la fig. 4.14 se muestra el archivo generado por la Raspberry y los campos que se guardarían en la base de datos en los campos de fecha y hora.

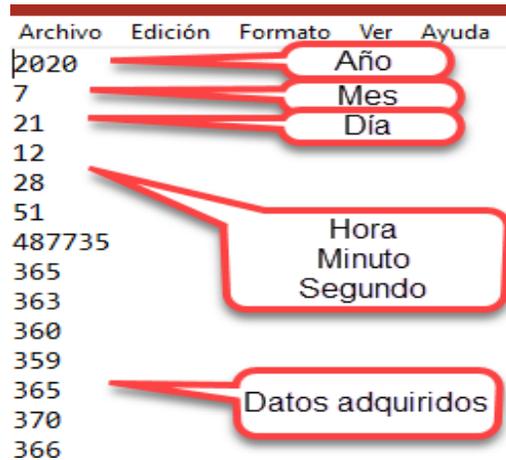


Figura 4.14 Archivo generado por la Raspberry con los campos para asignar en la base de datos.

La página de graficación cambió, ya que cuando se cargaba la página se verificaban los campos de la fecha en la base de datos con la de las mediciones existentes y se presentaban en forma de lista. En la fig. 4.15 se muestra la página con las opciones, en ellas también se agregaron 2 campos para poder elegir desde que hora hasta que hora se quiere graficar, se eligen y se da clic en el botón enviar para que aparezca la gráfica debajo de las opciones. En caso de que no se llegará a elegir un campo del horario, el sistema manda una alerta de que necesitaban ser llenados los campos.





Seleccione Alguna Fecha Existente De Medición

2020-07-22

Hora Inicial

01:00 a.m.

Hora Final

11:59 p.m.

enviar

Figura 4.15 Opciones para graficar en la página.

Al graficar se tomaban los datos seleccionados por el usuario y se buscaba en la base de datos las fecha y hora que correspondieran a las seleccionadas para poder graficar, en dado caso que el usuario haya elegido horarios donde no hubiera mediciones, saltaba una alerta que manifestaba que no había mediciones en el horario seleccionado.

En la fig. 4.16 se muestra la gráfica, que similar a la anterior, entregaba resultados apropiados, pero que en caso de que los archivos fueran demasiados grandes se tenían problemas, estos se explican en la parte de la retroalimentación.

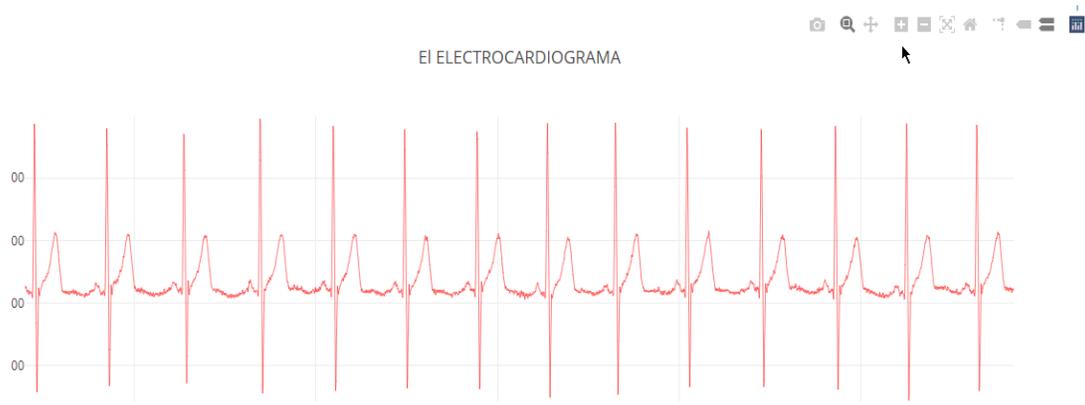


Figura 4.16 Grafica del electrocardiograma quinta iteración.

Retroalimentación y trabajo a futuro.

A pesar que se logró dar la opción de elegir el día de mediciones y la hora que se quisiera seleccionar, se tuvieron las siguientes consideraciones o problemáticas:

- Se propuso que el médico o usuario pudiera elegir un horario, pidiendo poner rangos desde las 0 horas y hasta las 24 horas, pero si no se llegara a saber en qué horario hay mediciones existentes, podría ser molesto que siempre le estuviera saliendo el mensaje de que no existen mediciones en el rango de horario seleccionado.
- Al cargar los documentos a la base de datos en la página principal tardaba un tiempo un poco considerable cuando los archivos eran de media hora ya que eran muchas mediciones; entonces si se llegará a dar clic en la siguiente página para graficar inmediatamente que cargo la página principal, no se cargarían todos los archivos a la base de datos.
- Cuando se deseaba graficar un rango largo de mediciones, tardaba demasiado y el navegador mostraba un error de que se esperó mucho tiempo a que cargara la página y no graficaba nada.



Por estas razones se comenzó a estudiar la posible solución, tratando de solucionar la gestión de la base de datos y del sistema de graficación, se observó, que, quizá podría usarse el mismo documento de texto plano para poder hacer la gestión de la selección de fechas y horas y también la graficación. Entonces quedaba pendiente para la siguiente iteración, hacer pruebas para manejar la selección de fechas y horas desde los archivos de texto plano, así como las gráficas.

4.5 Sexta iteración

Objetivo.

En esta iteración se tenía como objetivo poder gestionar las fechas y horas desde los documentos para poder mostrarlos y darle la oportunidad de que el usuario pudiera elegir la fecha y hora, ahora con las mediciones existentes para que no se le complique a la hora de querer visualizar dicho archivo de datos. También poder graficar desde el archivo de texto plano elegido por el especialista o usuario.

Requerimientos.

- Implementar un sistema que sea capaz de dar las fechas de los archivos existentes.
- Que el sistema pueda dar opción de los horarios solo del archivo elegido.
- Graficar los datos desde el archivo de texto plano que se eligió.

Pruebas y resultados.

Se empezó modificando de nuevo los datos del archivo desde la Raspberry quitando los valores de fecha y hora que se le agregaron en la iteración anterior, dejando solo los datos adquiridos, ya que como en el nombre del archivo contiene la fecha y hora en que se creó, nos ayudaría para hacer la gestión.



Se decidió quitar la página principal de la que se hablaba en las iteraciones anteriores, ya que los archivos ya no serían guardados en una base de datos, si no que ahora se conservarían los archivos de texto plano en una carpeta específica. Por esta razón también se decidió usar mediciones de 30 minutos para que en dado caso de que se tuviera una medición continua, solo se tuvieran 48 opciones de selección de hora, y no fuera complicado al momento de que el usuario o especialista eligiera.

En la fig. 4.17 se muestra la página, la cual al entrar se cargaron las fechas de los archivos existentes, al seleccionar alguna de esas fechas se visualizaron los horarios de mediciones relacionados a la fecha elegida, al graficar se muestra adecuadamente el archivo seleccionado.



Figura 4.17 Pagina de selección y graficación de la sexta iteración.

Cabe mencionar que, aunque hay que esperar alrededor de 10 segundos para poder ver la gráfica, no es tan tardado como pasaba en la iteración pasada, que se quedaba





cargado y no graficaba. Entonces se solucionó por completo con la gestión y graficación desde los archivos de texto plano.

Retroalimentación y trabajo a futuro.

El sistema trabaja de manera adecuada y se solucionó el problema con el empleo y gestión desde los archivos de texto plano, cabe mencionar que en vez de graficar contra un id que incrementaba en la base de datos, se hizo con tiempo. En el algoritmo se creó un arreglo donde va sumando de 0.004 en 0.004 para poder graficar las mediciones contra tiempo.

Ahora quedaba en la siguiente y última iteración, hacer una prueba de los pasos a seguir para que el usuario o especialista interactúe con el sistema y la aplicación web.

4.5 Séptima iteración

Objetivo.

El propósito de esta iteración era dar los pasos necesarios para poder usar el sistema. Es decir, la guía para poder utilizar el sistema de adquisición y el sistema web.

Requerimientos.

- Realizar guía de como colocar los electrodos
- Efectuar los pasos para poder usar el sistema de la Raspberry
- Hacer una guía para visualizar los datos en el sistema web

Pruebas y resultados

Para empezar a usar el sistema, primero se deben colocar los electrodos de forma triangular como se muestra en la fig. 2.5. Cada conector del cable con el electrodo trae una la letra, donde R se refiere al del lado derecho, L al del lado izquierdo y COM al que va en la parte inferior del dorso (ver fig. 4.18).



Figura 4.18 Referencia en electrodos para colocación.

Antes de colocar los electrodos al cuerpo, se agrega a cada uno gel conductor para hacer una mayor adherencia y se capte mejor la señal, el gel debe ir bien extendido en todo el electrodo.

Posterior a esto, se coloca el plug del cable de los electrodos a la caja donde contiene el AD8232 y el Arduino. Se conecta esta caja vía USB a la microcomputadora Raspberry. La Raspberry debe estar conectada a un monitor vía HDMI y después conectarla a la fuente de corriente de 5 volts a 2 amperes, esto puede ser con un adaptador de corriente o una power bank si se desea tener movilidad. Se espera a que encienda y se da doble clic en el archivo del escritorio llamado electrocardiograma (ver fig. 4.19). Al abrir el archivo se da en la tecla F5 para que empiece a ejecutar el algoritmo.

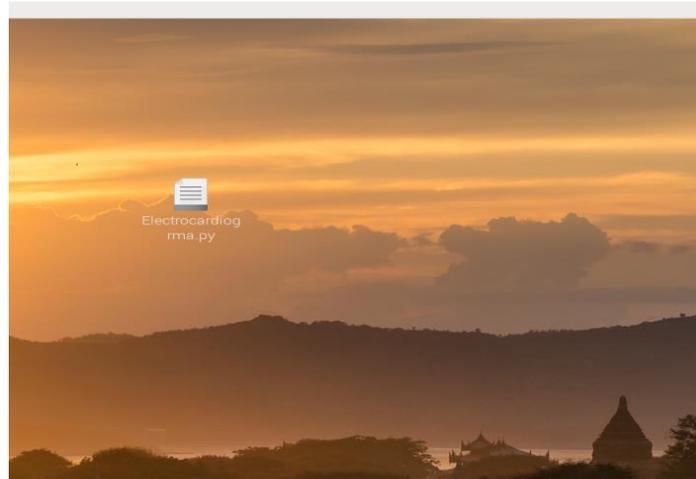


Figura 4.19 Programa electrocardiograma en Raspberry pi.

En la parte del sistema web es necesario entrar a al dominio de la página alojada, por ejemplo: rdecg.tech, desde cualquier navegador con conexión a internet. En este caso se hace desde una computadora y con Google Chrome.

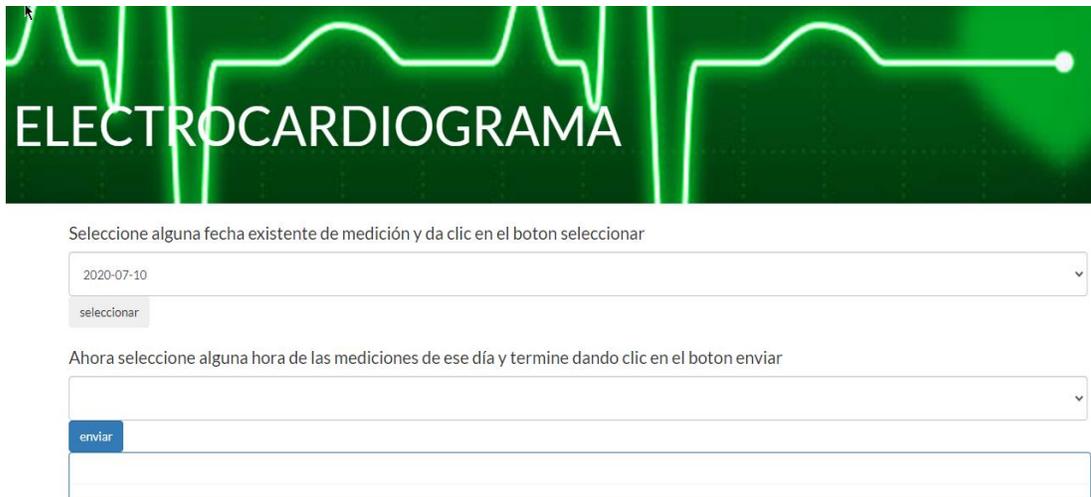


Figura 4.20 Pagina web para la visualización del electrocardiograma.

En la fig.4.20, como ya se comentaba, al ingresar a la página se ven dos listas de opciones, en la parte superior, por defecto al cargar la página se muestran todas las



fechas de las mediciones existentes. Entonces, para elegir una, se da clic en la lista y se selecciona la fecha de la medición que se desea obtener (ver fig. 4.21).

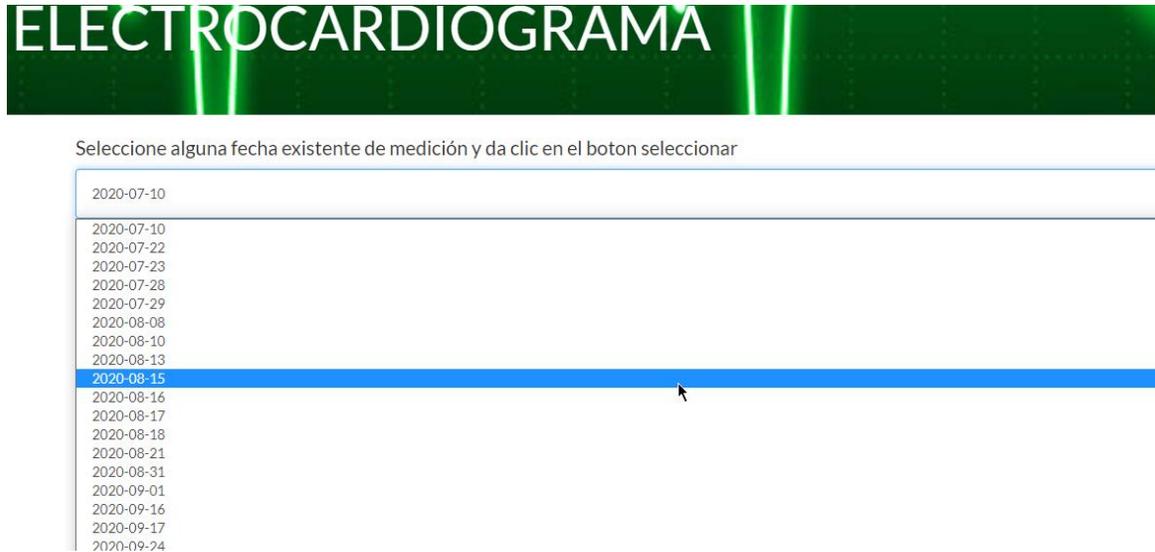


Figura 4.21 elección de fecha de mediciones existentes.

Una vez que se eligió, debe darse clic en el botón seleccionar, esto hará que en la siguiente lista aparezcan las horas de las mediciones de esa fecha seleccionada. Al igual que al seleccionar la fecha, ahora ocurre lo mismo para la hora, se da clic en la lista y se selecciona la hora deseada (ver fig. 4.22).





Seleccione alguna fecha existente de medición y da clic en el boton seleccionar

2020-08-21

seleccionar

Ahora seleccione alguna hora de las mediciones de ese día y termine dando clic en el boton enviar

11:05

11:05

11:35

12:05

12:35

13:05

19:13

19:43

Figura 4.22 Selección de hora de medición de la fecha seleccionada.

Una vez que se ha elegido la fecha y hora de las mediciones que se desean visualizar, se da clic en el botón enviar y se espera aproximadamente 10 segundos a que se cargue la gráfica. De primera impresión parecen que son puras líneas sin sentido (mostrado en la fig. 4.23), ya que por la gran cantidad de datos que se grafican es imposible visualizar de principio, pero gracias a las herramientas que se incluyen en la gráfica, se pueden ver con mayor calidad los datos graficados. A continuación, se explican las herramientas para poder interactuar con la gráfica.



Seleccione alguna fecha existente de medición y da clic en el boton seleccionar

2020-08-21
seleccionar

Ahora seleccione alguna hora de las mediciones de ese día y termine dando clic en el boton enviar

12:35
enviar

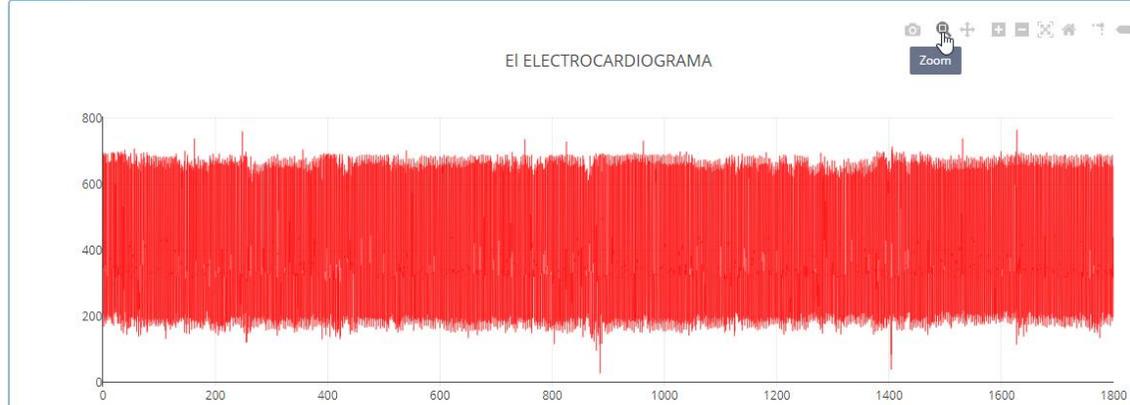


Figura 4.23 Grafica de electrocardiograma sin acercamiento.

Para poder visualizar la gráfica se tiene una barra de herramientas presentada en la fig. 3.17. Por defecto se tiene la tecla de zoom (🔍), con esta herramienta se puede seleccionar la parte de la gráfica que se deseada analizar; para esto se da clic, y sin soltar, se selecciona la parte a realizar el acercamiento, en la fig. 4.24 se muestra cómo se va tomando la parte que se desea hacer un acercamiento.

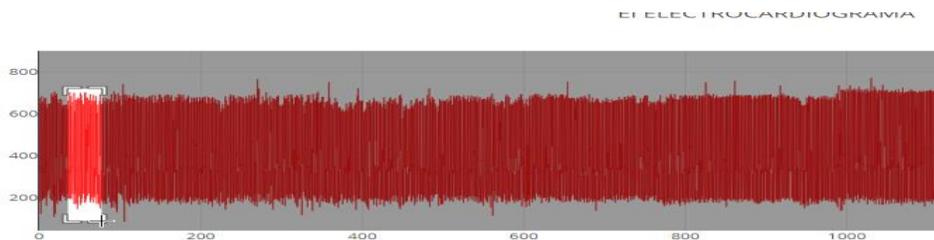


Figura 4.24 Parte seleccionada para acercar la gráfica.

Una vez que ya se tiene el acercamiento deseado, con la herramienta Pan (➕) se puede mover la gráfica para ir avanzando e ir visualizar más datos de izquierda a derecha. Si





se desea capturar o guardar lo que se está visualizando en ese momento, con la herramienta Download () se puede guardar una imagen png, solo se presiona el botón y se selecciona en el explorador de archivos la carpeta y nombre que se guardará la imagen. En la fig. 4.25 se observa una imagen capturada con esta herramienta.

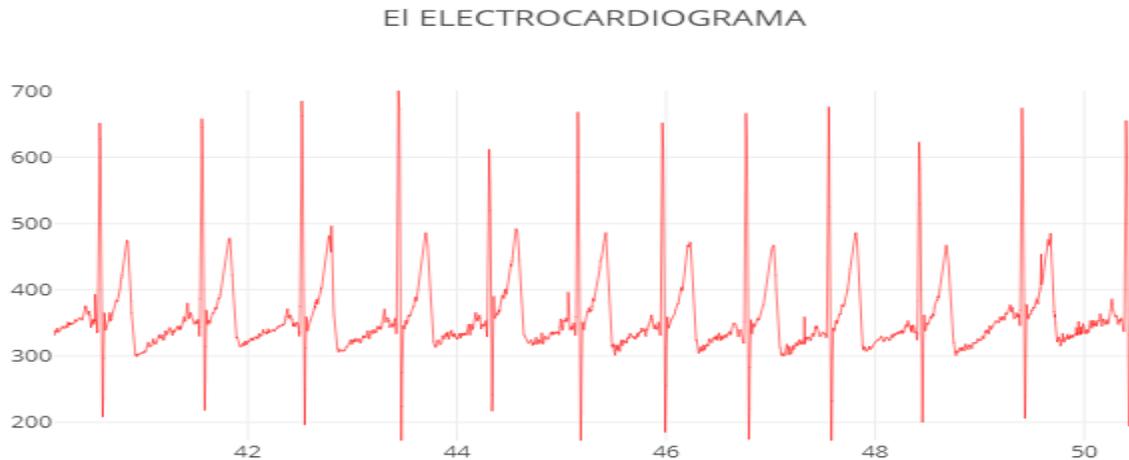


Figura 4.25 Imagen capturada.

Se usa la herramienta de reset () si se desea regresar al origen de la gráfica; es decir, a la parte inicial que se obtuvo después de presionar el botón enviar, esto ayuda si se desea elegir o seleccionar de nuevo otra parte del periodo de la medición, repitiendo el paso de la fig. 4.24.

También se puede notar en la gráfica que, en la parte inferior se muestran los segundos, entonces para poder tener una idea de la hora en que fueron capturadas esas mediciones, se toma la hora que se seleccionó y se le suman los segundos que se encuentran debajo de la gráfica para tener el horario de medición exacto.

Retroalimentación.

Como ya se comentaba, el sistema y en especial la microcomputadora Raspberry puede ir conectada a una fuente de alimentación con un eliminador o con un banco de baterías





portátil o power bank. En caso de que se requiera algo de movilidad es recomendable usar el banco de baterías, pero haciendo pruebas se pudo apreciar que, en caso de realizar actividades o movimientos bruscos, produce mediciones alteradas como se muestra en la fig. 4.26. En este caso se hicieron actividades hogareñas durante el periodo de medición, viéndose afectada la adquisición por las variaciones de contacto con los electrodos al cuerpo en esos movimientos bruscos.

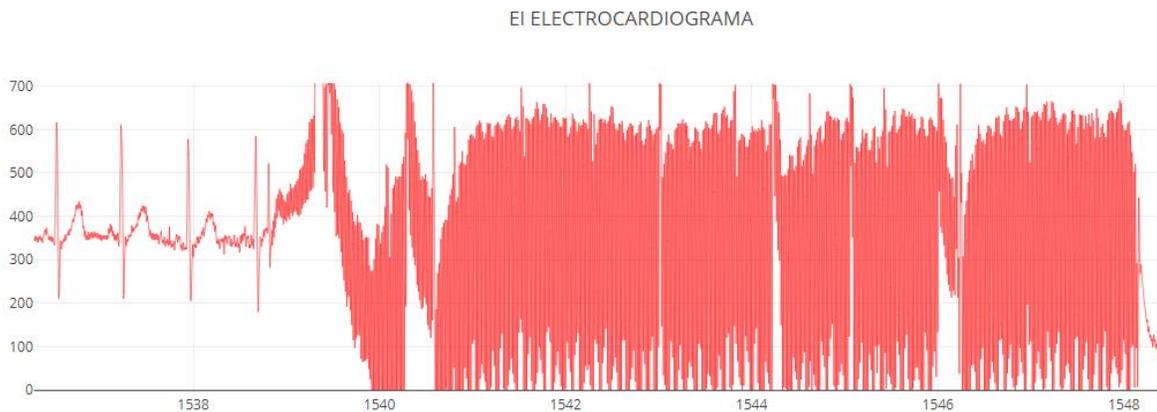


Figura 4.26 Afectación de la gráfica por el no buen contacto de los electrodos.

Cabe mencionar que los electrodos influyen mucho en las mediciones, y de no estar bien adheridos a la piel, como se puede observar, afecta en las mediciones. Otro factor importante es el buen estado de los electrodos, ya que si se encuentran desgastados, también pueden verse afectadas las mediciones, teniendo datos o gráficas similares a la de la fig. 4.26.





Conclusiones y trabajos futuros

Se pudo verificar que, gracias al algoritmo propuesto, el sistema puede realizar frecuencias de muestreo de 4 ms equivalentes a 250Hz, alcanzando componentes de frecuencia de 125Hz satisfaciendo el criterio de Nyquist.

El sistema captura los datos y los va guardando de manera local en la microcomputadora en archivos de texto plano; en dado caso que no exista conexión a internet, se quedan guardados hasta que tenga la posibilidad de conectarse al servidor y poder enviarlos. De manera similar pasa si se llega a quedar sin suministro de corriente eléctrica, se queda guardado el archivo con la última medición obtenida y se cierra, quedando en espera para ser enviado al servidor.

En la parte del servidor, recibe y guarda los archivos provenientes de la microcomputadora. Asimismo, el algoritmo propuesto muestra las fechas y horas de los archivos que están actualmente almacenados, para después poderlos graficar, simplificando que medición se desea visualizar, con herramientas para poder apreciar mejor los resultados de forma fácil de usar.

Los componentes de hardware son los más adecuados para que pueda funcionar el sistema, ya que como se verificó si se usaban un lector de tarjetas micro sd para guardar los archivos, se alteraba la frecuencia de muestreo. También si se usaba la tarjeta WeMos, en caso de que no hubiera conexión hacia el servidor, los datos se perderían, de ahí la propuesta del uso de la microcomputadora Raspberry, con la cual el sistema funciona confiablemente y con la ventaja de ser de bajo costo.

Cabe mencionar que el sistema operativo y herramientas de desarrollo utilizadas en la Raspberry y el servidor son de uso libre, esto hace que este sistema tenga un costo reducido.



Por lo anterior dicho, y de acuerdo a las pruebas presentadas, se puede concluir que sistema cumple con los objetivos planteados. Adicionalmente se tiene una oportunidad de mejora para realizarse como trabajo futuro, de lo que se puede citar lo siguiente:

- Implementar una forma de que el sistema este toda integrado en una caja o algo similar donde solo salgan los cables para los electrodos. Es decir, que contenga una interfaz gráfica de usuario en una pantalla touch pequeña donde se puedan iniciar las funciones del algoritmo de la microcomputadora, pero manteniendo un costo reducido.
- Que el sistema web tenga una mejor interfaz, para que el usuario o paciente se pueda dar de alta y hacer una cuenta, asociándola con la microcomputadora. Esto con la finalidad de que el sistema sea más global y cada que un usuario quiera ingresar para ver sus mediciones ingrese con su usuario y contraseña.
- Con el propósito de poder revisar más a fondo el estudio, es necesario agregarle una función que realice la transformada rápida de Fourier y se pueda obtener el espectrograma, pudiendo servir para revisiones más detalladas de los datos.
- También se propone que mediante un tipo de inteligencia artificial el sistema pueda predecir o advertir si el paciente tiene alguna afectación, mandando de inmediato un mensaje para que el especialista tenga especial atención a esos datos del ese paciente.



Referencias

- [1] OMS, «Organización Mundial de la Salud,» 1 Junio 2018. [En línea]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/noncommunicable-diseases>. [Último acceso: 20 Septiembre 2019].
- [2] «INEGI,» 2016. [En línea]. Available: <http://www.inegi.org.mx/est/contenidos/proyectos/registros/vitales/mortalidad/tabulados/PC.asp?t=14&c=11817>.
- [3] Cruz Roja, «El universal,» 27 06 2019. [En línea]. Available: <https://www.eluniversal.com.mx/articulo/nacion/sociedad/2017/04/21/ataques-cardiacos-primera-causa-de-muerte-en-mexico>.
- [4] M. Ayaskanta, K. Akanksha, S. Pooja y P. Pranjal , «researchgate.net,» 04 Abril 2018. [En línea]. Available: https://www.researchgate.net/profile/Ayaskanta_Mishra/publication/324686561_Remote_web_based_ECG_Monitoring_using_MQTT_Protocol_for_IoT_in_Healthcare/links/5adcadfb0f7e9b285940b8c0/Remote-web-based-ECG-Monitoring-using-MQTT-Protocol-for-IoT-in-Healthcare.. [Último acceso: 18 abril 2020].
- [5] P. Guaman y P. Cesar, «Universidad Católica de Santiago de Guayaquil,» 22 Junio 2018. [En línea]. [Último acceso: 20 Abril 2020].
- [6] C. Palao Cruz, Desarrollo de un sistema IoT integrado con dispositivos eHealth para la detección automática de la variabilidad cardíaca, Valencia, 2017.
- [7] Ó. García, de *Atlas del cuerpo humano*, Barcelona, Thema, 2007, pp. 36-39.





- [8] I. J. Sastre Pérez y R. Pérez Galán, «faeditorial.es,» [En línea]. Available: <https://www.faeditorial.es/capitulos/infarto-miocardio.pdf>. [Último acceso: 11 Julio 2019].
- [9] L. Azcona, «BBVA,» [En línea]. Available: https://www.fbbva.es/microsites/salud_cardio/mult/fbbva_libroCorazon_cap4.pdf. [Último acceso: 15 08 2019].
- [10] M. Gonzales Villada y X. Martinez Osorio, ANÁLISIS DE SEÑALES ELECTROCARDIOGRÁFICAS (ECG) CON ISQUEMIA CARDIACA USANDO TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES, Manizales, 2009.
- [11] C. Lydia, «Edic College,» 2015. [En línea]. Available: <http://ediccollege.edu/wp-content/uploads/2016/01/Lectura-Comprension-EKG.pdf>. [Último acceso: 15 08 2019].
- [12] DGIS, «Sistema de informacion de la Secretaria de Salud,» [En línea]. Available: <http://sinaiscap.salud.gob.mx:8080/DGIS/>. [Último acceso: 16 Junio 2020].
- [13] A. Sgarlatta, «Sensor inalámbrico de ECG conectado vía Bluetooth,» Universidad Nacional de Cordoba, Cordoba, 2016.
- [14] F. Miyara, «Conversores D/A y A/D,» Universidad Nacional de Rosario, Rosario, 2004.
- [15] R. Martorell, «el Hospital,» el Hospital, Abril 2016. [En línea]. Available: <http://www.elhospital.com/temas/Analisis-de-la-frecuencia-de-muestreo-en-electrocardiografia+111828?tema=10000003&pagina=1>. [Último acceso: 12 Junio 2020].
- [16] F. E. Valdés Pérez y R. Pallás Areny, Microcontroladores: Fundamentos y aplicaciones con PIC, Cuba: Marcombo, 2007.



- [17] M. A. Pérez Rodas, DISEÑO E IMPLEMENTACIÓN DE UN ELECTROCARDIOGRAFO PORTÁTIL Y DEL SISTEMA DE PROCESAMIENTO DIGITAL DE SEÑALES ELÉCTRICAS DEL CORAZÓN, PARA MONITOREO Y ANÁLISIS MÉDICO, Guatemala, 2011.
- [18] M. A. Quintero M., «Introduccion a la Computacion,» Universidad de Los Andes, Mérida, 2002.
- [19] «Orangepi en español,» Orange pi, [En línea]. Available: <https://orangepiweb.es/index.php#gsc.tab=0>. [Último acceso: 16 mayo 2020].
- [20] «Link Sprite,» [En línea]. Available: <https://www.linksprite.com/linksprite-pcduino3/>. [Último acceso: 2020 Mayo 17].
- [21] «Raspberry pi,» [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. [Último acceso: 2020 Mayo 20].
- [22] J. L. Villada Romero, «UF1217: Instalación y configuración del software de un servidor Web,» IC, Málaga, 2014.
- [23] X.-g. Zou, Q. Li, S.-H. Sun y X. Niu, «SpringerLink,» 2005. [En línea]. Available: https://link.springer.com/chapter/10.1007/11553939_151. [Último acceso: 2020 Junio 20].
- [24] M. w. docs, «Generalidades del protocolo HTTP,» 7 Agosto 2020. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>. [Último acceso: 2020 Junio 21].
- [25] C. Gardoki, «Hostalia».





- [26] F. Segarra, «Hostinger,» [En línea]. Available: <https://www.hostinger.mx/tutoriales/que-es-un-vps#Que-es-un-VPS>. [Último acceso: 19 02 2020].
- [27] Amazon, «AWS,» [En línea]. Available: <https://aws.amazon.com/es/what-is-aws/>. [Último acceso: 19 02 2020].
- [28] M. E. Millán, «Fundamentos de bases de datos,» Universidad del Valle, Santiago de Cali, 2012.
- [29] S. Lujan Mora, «Programación de aplicaciones web: historia, principios básicos y clientes web,» Club Universitario, Alicante, 2002.
- [30] P. Roger S., Ingeniería del software Un enfoque practico, Connecticut: McGrawHill, 2010.
- [31] A. Prieto, «CENETEC,» 15 03 2019. [En línea]. Available: http://www.cenetec.gob.mx/cd_inter/guiapaciente/electrocardiograma.pdf. [Último acceso: 15 03 2019].





Referencias de figuras

Figura 2.1. El corazón. -recuperado de [7].

Figura 2.2. Gráfica generada por electrocardiograma. – recuperado de [9].

Figura 2.3. Gráfica de principales causas de muerte a nivel mundial dentro de las ENT. – datos tomados de [1].

Figura 2.4. Tarjeta de adquisición con chip AD8232. – recuperado el 16 de junio de 2019 en

https://www.vlnssport.com/index.php?main_page=product_info&products_id=63017

Figura 2.6. Señal analógica. – recuperado el 5 de septiembre de 2019 en

<https://marinaebg.wordpress.com/2014/10/21/practica-v-manejo-del-osciloscopio/>

Figura 2.7. Señal Digital. – recuperado el 5 de septiembre de 2019 en

http://black-darkjona.blogspot.com/2012/08/medicion-de-frecuencia-osciloscopio_26.html

Figura 2.8. Resolución de la señal. – recuperado el 8 de septiembre de 2019 en

https://www.reddit.com/r/askscience/comments/4cjgl9/if_i_had_a_very_fast_and_accurate_camera_could_i/

Figura 2.9. Señal analógica muestreada. – recuperado el 8 de septiembre de 2019 en

http://fcaenlinea1.unam.mx/anexos/1467/1467_u1_act3.pdf

Figura 2.11. Orange pi. – recuperado de [19].

Figura 2.12. PcDuino. – recuperado de [20].

Figura 2.13. Raspberry pi 3b+. – recuperado de [21].

Las figuras no referenciadas son de autoría propia.





Anexos

A) Código de algoritmo en Arduino

```
#include <Arduino.h>

unsigned long time;

unsigned long t=0;

int Dt = 3;

void setup() {
  Serial.begin(115200);
}

void loop()
{
  time= millis();
  if (time-t> Dt){
    t = time;
    int x = analogRead(A0); //
    Serial.println(x);
  }
}
```





B) Código de algoritmo en Raspberry

```
#!/usr/bin/python

import serial

from ftplib import FTP, error_perm

import socket

import time

import datetime

from os import listdir, remove

while True:

    try:

        s= serial.Serial('/dev/ttyACM0', 115200)

        while True:

            fecha=str(datetime.date.today())

            hora = time.strftime("%H:%M")

            t=1

            f = open ('/home/pi/Desktop/prueba/'+fecha+hora+'.txt', 'a')

            while t <=450000:

                v=s.readline().decode("utf-8","ignore")

                if not v:

                    print("No llego dato")

                else:
```





```
f.write(v)

t += 1

f.close()

print("Termino de crearse un archivo. Cotinua...")

con = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

try:

    con.connect(('rdecg.tech', 80))

    print ("online")

    try:

        ftp=FTP('rdecg.tech')

        ftp.login('admin_rdecg','3104')

        ftp.cwd('archivos')

        fielist=[]

        fielist=listdir("/home/pi/Desktop/prueba")

        for i in fielist:

            print(f"ahora vale esto: {i}")

            ftp.storbinary('STOR %s' %i, open('%s'

%/home/pi/Desktop/prueba/'+i,'rb'))

            ftp.dir()

            print("Se subio archivo")

            remove ('/home/pi/Desktop/prueba/'+i)
```





```
print(f"se borro archivo: {i}")
```

```
except error_perm as e:
```

```
    print (e)
```

```
ftp.quit()
```

```
except:
```

```
    print ("offline")
```

```
con.close()
```

```
except serial.SerialException:
```

```
    print("No hay dispositivo conectado")
```



C) Códigos de algoritmos en el servidor

Index.php

```
<!DOCTYPE html>

<html>

<head>

<title>Graficos title</title>

<link rel="stylesheet" type="text/css" href="librerias/bootstrap/css/bootstrap.css">

<script src="librerias/jquery-3.5.1.min.js"></script>

<script src="librerias/plotly-latest.min.js"></script>

<link href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900&display=swap"
rel="stylesheet">

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<header class="site-header inicio">

<div class="contenedor contenido-header">

<div class="barra">

<a href="/">



</a>
```





```
</div>

<h1>ELECTROCARDIOGRAMA</h1>

</div> <!-- contenedor-->

</header>

<body>

<div class="container">

<div class="col-4">

<h3>Seleccione alguna fecha existente de medición y da clic en el boton seleccionar</h3>

<select class="col-12 form-control" name="select_fecha" id="select_fecha">

</select>

</div>

<div class="col-2">

<button type="button" name="bt_select" class="btn" id="bt_select">seleccionar</button>

</div>

<div class="col-4">

<h3 for="">Ahora seleccione alguna hora de las mediciones de ese día y termine dando clic
en el boton enviar</h3>

<!--input type="time" name="hora_inicial" id="hora_inicial" class="form-control"-->

<select class="col-12 form-control" name="select_hora" id="select_hora">

</select>
```





```
</div>

<div class="col-4 d-flex justify-content-center" >

  <!--input type="time" name="hora_final" id="hora_final" class="form-control"-->

</div>

<div class="row">

  <div class="col-12" id="mensaje_error">

    </div>

  </div>

  <div class="col-2">

    <button      type="button"      name="bt_enviar"      class="btn      btn-primary"
id="bt_enviar">enviar</button>

    </div>

  </div>

  <div class="container">

    <div class="row">

      <div class="col-sm-12">

        <div class="panel panel-primary">

          <!--div class="panel panel-heading">Electrocardiograma

          </div-->

          <div class="panel panel-body">

            <div class="row">
```





```
var fecha = nom.substr(0,10);

if (fechas.includes(fecha)) {

}else{

    fechas.push(fecha);

}

}

var texto = "";

for (var i = 0; i < fechas.length; i++) {

    texto += `

        <option value="${fechas[i]}">${fechas[i]}</option>

    `;

}

$("#select_fecha").append(texto);

});

//*****boton select*****////////////////////////////////////

$("#bt_select").click(function(){

    $("#select_hora").empty();

    var date = $("#select_fecha").val();

    $.get('directorios.php',function(archivos){

        archivos=JSON.parse(archivos);

        var horas=[];
```





```
for (var i = 0; i < archivos.length; i++) {  
  
    var nom = archivos[i]  
  
    var fecha = nom.substr(0,10);  
  
    if (fecha == date) {  
  
        var hora = nom.substr(-9,5);  
  
        if (horas.includes(hora)) {  
  
            }else{  
  
                horas.push(hora);  
  
            }  
  
        }else{  
  
            }  
  
        }  
  
    var texto ="";  
  
    for (var i = 0; i < horas.length; i++) {  
  
        texto += `  
  
            <option value="${horas[i]}">${horas[i]}</option>  
  
        `;  
  
    }  
  
    $("#select_hora").append(texto);  
  
});  
  
});
```





```
//*****boton enviar*****

$("#bt_enviar").click(function(){

    $("#mensaje_error").empty();

    $("#mensaje_error").removeClass("alert alert-danger");

    $.get('directorios.php',function(archivos){

        archivos=JSON.parse(archivos);

        var fecha = $("#select_fecha").val();

        var hora = $("#select_hora").val();

        var nomC = fecha.concat(hora);

        nomC += ".txt";

        for (var i = 0; i < archivos.length; i++) {

            if (archivos[i] == nomC) {

                var nombre = "archivos/"+archivos[i];

                $.get(nombre, function(lecturas){

                    respuesta = {

                        'lecturas' : lecturas

                    };

                });

                $.post('leer.php', respuesta, function(respuesta){

                    respuesta = JSON.parse(respuesta);

                    console.log(respuesta);

                    datosX=crearCadenaLineal(respuesta['x']);

                });

            }

        }

    });

});
```





```
console.log(datosX);

datosY=crearCadenaLineal(respuesta['y']);

console.log(datosY);

var trace1 = {

  x: datosX,

  y: datosY,

  type: 'scatter',

  line: {

    color: 'red',

    width: 0.5

  }

}

var data = [trace1];

var layout = {

  title: 'El ELECTROCARDIOGRAMA',

  xaxis: {

    title: ""

  },

  yaxis: {

    title: ""

  }

}
```



```
};  
  
var lineDiv = document.getElementById('cargaLineal');  
  
Plotly.newPlot( lineDiv, data, layout);  
  
});  
  
});  
  
}else {  
  
}  
  
}  
  
});  
  
});  
  
});  
  
function crearCadenaLineal(json){  
  
var parsed =json;  
  
var arr = [];  
  
for (var x in parsed) {  
    arr.push(parsed[x]);  
  
}  
  
console.log(arr);  
  
return arr;  
  
}  
  
</script>
```



Función leer.php

```
<?php
    $lectura = $_POST['lecturas'];
    $delimitador = "\n";
    $datos_separados = explode($delimitador, $lectura);
    $datos=[];
    $tiempo=[];
    $final = sizeof($datos_separados);
    for($i = 0; $i<$final; $i++){
        $datos[] = $datos_separados[$i];
        $tiempo[] = $i*0.004;
    }
    $respuesta = [
        'x' => $tiempo,
        'y' => $datos
    ];
    echo json_encode($respuesta);
?>
```

Función directorios.php

```
<?php
$files = array_slice(scandir('/home/admin/web/rdecg.tech/public_html/archivos'), 2);
```



```
echo json_encode($files);
```

```
?>
```

Hoja de estilos style.css

```
html {  
    font-size: 62.5%; /** reset para REMS = 62.5% = 10 px de 16 px **/  
}  
*, *:before, *:after {  
    box-sizing: inherit;  
}  
body{  
    font-family: 'Lato', sans-serif;  
    font-size: 1.6rem;  
}  
/** Globales **/  
.contenedor {  
    width: 98%;  
    max-width: 130rem; /** = 1200px; **/  
    margin: 0 auto;  
}  
h1{  
    font-size: 6rem;
```





```
}  
h2{  
    font-size: 3rem;  
}  
h3{  
    font-size: 2rem;  
}  
/*utilidades*/  
.seccion{  
    margin-top: 2rem;  
    margin-bottom: 2rem;  
}  
.fw-300 {  
    font-weight: 300;  
}  
.centrar-texto {  
    text-align: center;  
}  
.contenido-centrado {  
    max-width: 900px;  
}  
/** Botones **/
```



```
.boton {  
    color: #ffffff;  
    font-weight: 700;  
    text-decoration: none;  
    font-size: 1.8rem;  
    padding: 1rem 3rem;  
    margin-top: 3rem;  
    display: inline-block;  
    text-align: center;  
    border: none;  
    display: block;  
    flex: 0 0 100%;  
}  
  
@media (min-width: 768px) {  
    .boton {  
        display: inline-block;  
        flex: 0 0 auto;  
    }  
}  
  
.boton:hover {  
    cursor: pointer;  
}
```





```
.boton-verde {  
    background-color: rgb(67, 105, 0);  
}  
  
/*header */  
  
.site-header {  
    background-color: #333333;  
    padding: 0;  
    min-height: 15rem;  
}  
  
.logo{  
    width: 20rem;  
    height: 15rem;  
}  
  
.site-header.inicio {  
    background-image: url(../img/logo.jpg);  
    background-position: center center;  
    background-size: cover;  
    height: 40vh;  
    min-height: 40rem;  
}  
  
.contenido-header {
```



```
height: 100%;  
  
display: flex;  
  
flex-direction: column;  
  
justify-content: space-between;  
  
}  
  
.contenido-header h1{  
  
color: #ffffff;  
  
padding-bottom: 3rem;  
  
max-width: 50rem;  
  
line-height: 0.5;  
  
}  
  
.barra {  
  
display: flex;  
  
justify-content: space-between;  
  
align-items: center;  
  
padding-top: 2rem;  
  
}  
  
/*Navegacion */  
  
.navegacion a {  
  
color: #ffffff;  
  
text-decoration: none;
```



```
font-size: 1.8rem; /*= a 18px*/  
  
margin-right: 2rem;  
  
}  
  
.navegacion a:hover {  
  
    color: #71B100;  
  
}  
  
.navegacion a:last-of-type{  
  
    margin: 0;  
  
}  
  
/* Formulario*/  
  
.formulario {  
  
    font-size: 2rem;  
  
    color: #4f4f4f;  
  
    margin: 2rem 0 0 0;  
  
}  
  
legend {  
  
    font-size: 2rem;  
  
    color: #4f4f4f;  
  
}  
  
label {  
  
    font-weight: 700;  
  
    text-transform: capitalize;
```



```
display: block;
}
input:not([type="submit"]) {
padding: 1rem;
display: block;
width: 98%;
background-color: #e1e1e1;
margin-bottom: 3rem;
border: none;
border-radius: 2rem;
}
input[type="radio"]{
width: auto;
margin: 0;
}
```



D) Plan de negocios

Nombre

RD Systems mx;



Figura D.1 Logo.

Eslogan

Monitoreamos tu corazón, cuidando tu bolsillo.

Propuesta de valor

Sistema de adquisición y procesamiento de señales electrocardiográficas visualizables desde un sistema web.

El cual posee las siguientes características:

- Dispositivo (hardware) cómodo para el usuario
- Interfaz gráfica amigable con el cliente
- Visualización desde cualquier dispositivo con un navegador web y conexión a internet



Segmento de mercado

Hospitales o clínicas.

Pueden ser para especialidad en problemas cardiacos o simplemente en clínicas generales, ya que en ocasiones no se cuenta con un instrumento para estudios de este tipo.

Público en general.

Cualquier familia D+, C, C+, etc., que tenga un familiar que necesite monitoreo constante de su sistema cardiaco, y para que algún medico pueda verlo desde donde se encuentre y valorar su situación médica. Por lo general son a personas mayores, pero no se restringe solo ellos, también al que lo requiera o tenga problemas del corazón y necesite estarse monitoreando constantemente.

Canales de distribución

Se realizará una comunicación directa con las personas interesadas (hospitales, clínicas, familias o personas interesadas, tiendas especializadas, etc.) y paralelamente en un sitio web, donde se venderá y dará a conocer el sistema, también en las redes sociales se dará a conocer y venderá el sistema.

Este sitio web también sirve para el monitoreo, consulta de datos, y a su vez servicios post venta (mantenimiento, soporte y asesoría).

Relación con los clientes

Como ya se mencionaba, podrían realizarse las compras en línea con la posibilidad de diferentes formas de pago (tarjetas, pago contra entrega, mercado pago, etc), con la posibilidad de recibirlo en la comodidad de su hogar o de la dirección proporcionada por el comprador.





En el caso de la página comercial viene por predeterminado la valoración del artículo, pero en las descripciones se dará el link de la página oficial para que pueda contestar un pequeño cuestionario de satisfacción y control de calidad.

Así mismo, dentro del producto se le proporcionara al consumidor:

Instructivo de uso, con pasos muy claros y precisos de cómo funciona.

El sitio oficial (ya sea el link, pagina o código QR) donde se le invitara a contestar el cuestionario de calidad y satisfacción del producto servicio.

En esta misma página se atenderá de manera personalizada las instrucciones de uso o alguna cuestión que tenga el usuario.

Recursos clave

Físicos

Instalaciones de diseño y desarrollo:

- Espacio rentado o adquirido

Mobiliario:

- Escritorio amplio
- Silla semi ejecutiva
- Mesas de trabajo
- No break con supresor de picos

Equipo

Computo:

- PC con Intel Core I7, 16 Gb de RAM, con teclado, mouse y monitor.

Electrónica y componentes electrónicos:

- Arduino
- Raspberry Pi
- Memorias microSD
- Carrete de estaño



- Pasta para soldar

Herramientas:

- Impresora 3d
- Estación de soldado
- Multímetro

Comunicación:

- Kit cable, teléfono e internet de al menos 30 Mbps de descarga
- Smartphone con servicios de renta en compañía telefónica
- Dominio y hosting de pagina

Intelectuales

- Curso aplicación y desarrollo web
- Cursos de Python
- Cursos Marketing

Humano

- Ingeniero en desarrollo web full stack.
- Administrador de la empresa que se encargue de ventas, almacén, bitácoras, etc.
- Ingeniero electrónico encargado del sistema físico(hardware).
- Asistente que sepa todo sobre el producto para ventas y que se pueda transportar con los usuarios.

Actividades clave

Producción de calidad tanto físico como software.

Manual de usuario físico y digital, explicado muy específicamente.

Soporte técnico 24/7 con la ayuda de un chat Bot (respuestas automáticas), asimismo correo, página y número telefónico para asistencia personalizada.

Aliados clave

Hospitales, clínicas y público en general: será la principal fuente de ingresos y de donde se retroalimentará el proyecto para posibles mejoras.



Proveedor de componentes electrónicos: facilite los materiales, de garantía y sobre todo, que de un excelente precio.

Estructura de costos

Tabla D.1 Presupuesto de inversión.

Activo fijo			
Concepto	Cantidad	Precio unitario	Total
Computadoras completas	3	\$12,500	\$37,500
Sillas semi ejecutivas	5	\$950	\$4,750
Mesas de trabajo	2	\$2,800	\$5,600
Escritorio principal	1	\$4,500	\$4,500
Laptop	1	\$13,000	\$13,000
Impresora	1	\$4,000	\$4,000
Microondas	1	\$2,500	\$2,500
Dispensador de agua	1	\$3,500	\$3,500
Lápices	20	\$3	\$60
Bolígrafos	20	\$3	\$60
Hojas	5	\$89	\$445
Acondicionamiento y remodelación de local	1	\$15,000	\$15,000
Herramientas requeridas	1	\$10,285	\$10,285
Subtotal			\$101,200

Capital de trabajo			
Producto	Cantidad	Precio unitario	Total
Mano de obra Jornal	320	\$112	\$35,840
Insumos	1	\$1,958	\$1,958
Administración	1	\$12,000	\$12,000
Agua	1	\$1,085	\$1,085
Luz	1	\$1,850	\$1,850
Servicio Teléfono internet	1	\$1,000	\$1,000
Renta de servicios de hosting	1	\$750	\$750





Subtotal			\$54,483
TOTAL			\$155,683

Fuente de ingresos

La principal fuente de ingreso es la venta del sistema.

Actualización del sistema, si se llega a tener mejoras o prototipos posteriores se dará mejor precio si adquirió la versión anterior.

Venta de consumibles(electrodos).

Tabla D.2 Proyección de ingresos.

Concepto	Ventas al mes	Precio	6 meses	1 año	2 años	3 años
Paquete completo	10	\$ 5,111	\$306,660	\$613,320	\$ 1,226,640	\$1,839,960
Electrodos	15	\$30	\$2,700	\$5,400	\$ 10,800	\$16,200
		TOTAL	\$309,360	\$618,720	\$1,237,440	\$1,856,160

