

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS



**ADQUISICIÓN DE LA SEÑAL PPG MEDIANTE UNA
PLATAFORMA ARDUINO**

PRESENTADO POR:

KENDAL ALFONSO SOSA MONTES

PARA OPTAR AL TITULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

CIUDAD UNIVERSITARIA, SEPTIEMBRE DE 2023

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

ING. FRANCISCO ALARCÓN SANDOVAL

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO:

PhD. EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS

DIRECTOR:

ING. RUDY WILFREDO CHICAS VILLEGAS

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS

Trabajo de Graduación previo a la opción al Grado de:
INGENIERO DE SISTEMAS INFORMATICOS

Título:

**ADQUISICIÓN DE LA SEÑAL PPG MEDIANTE UNA
PLATAFORMA ARDUINO**

Presentado por:

KENDAL ALFONSO SOSA MONTES

Trabajo de Graduación Aprobado por:

Docente Asesor:

ING. BLADIMIR DIAZ CAMPOS

SAN SALVADOR, SEPTIEMBRE DE 2023

Trabajo de Graduación Aprobado por:

Docente Asesor:

ING. BLADIMIR DIAZ CAMPOS



FACULTAD DE INGENIERÍA Y ARQUITECTURA
UNIVERSIDAD DE EL SALVADOR

TRABAJO DE GRADUACIÓN EN INGENIERÍA DE SISTEMAS INFORMÁTICOS

Adquisición de la señal PPG mediante una plataforma Arduino

AUTOR

Br. Kendal Alfonso Sosa Montes

Número de cédula SM13001

TUTOR

Dr. Javier de Pedro Carracedo

Universidad de Alcalá (UAH), España

COTUTOR

Ing. Bladimir Díaz Campos

Universidad de El Salvador (UES)

AÑO ACADÉMICO
2023

A los lectores empedernidos

Abstract

This document presents a way of how to create a device using Arduino and a heart rate sensor, which is capable of obtaining the PPG signal of a person using the photoplethysmography (PPG) method, a non-invasive technique that allows detecting variations in blood volume by means of a photosensor and a generally green light beam. This paper provides the necessary tools to build a photoplethysmograph using open source hardware, with which heartbeat measurements can be made in people and study the rhythm differences found in each patient to later search for possible symptoms of chronic stress. In addition, different source codes are proposed to obtain and process the PPG signal captured by Arduino.

Keywords: Photoplethysmography, Arduino, Heart rate sensor, PPG Signal, Chronic Stress detection.

Resumen

En el presente documento se presenta una forma de cómo crear un dispositivo utilizando Arduino y un sensor de pulso cardíaco, que sea capaz de obtener la señal PPG de una persona mediante el método de fotopleletismografía (PPG), una técnica no invasiva que permite detectar variaciones de volumen sanguíneo por medio de un fotosensor y un haz de luz generalmente verde. En este trabajo se brindan las herramientas necesarias para construir un fotopleletismógrafo utilizando hardware de código abierto, con el cual se pueden realizar mediciones del sistema cardiorrespiratorio en personas y estudiar las diferencias encontradas en cada paciente para posteriormente buscar posibles síntomas de estrés crónico. Además, se proponen diferentes códigos fuente para obtener y procesar la señal PPG capturada por Arduino.

Palabras clave: fotopleletismografía, Arduino, sensor de pulso cardíaco, señal PPG, detección de estrés crónico.

Contenido

Lista de figuras	XI
Lista de tablas	XIII
Lista de algoritmos	XVII
Lista de fragmentos de código	XVII
Lista de acrónimos	XIX
1 Introducción	1
1.1 Las señales biológicas	1
1.1.1 Necesidades de monitorización fisiológica de pacientes . . .	3
1.1.2 Adquisición de las señales biológicas	3
1.2 Clasificación de las señales biológicas	6
1.2.1 Tipo de variable fisiológica	6
1.2.2 Frecuencia de adquisición	7
1.3 Importancia de las señales biológicas	7
2 La señal PPG	9
2.1 Introducción	9
2.2 Origen físico	9
2.3 Origen fisiológico	11
2.3.1 Sistema de transporte de oxígeno	11
2.3.2 Hemoglobina y oxigenación	12
2.3.3 Propiedades ópticas de la Hemoglobina	14
2.3.4 Estrés	15

2.4	Fotopletismografía	16
2.5	Origen de la señal PPG	16
2.6	Representación de la señal PPG	17
2.7	Adquisición de la señal PPG	19
2.7.1	Configuraciones de sensores PPG	19
2.7.2	Adquisición simultánea de señales	21
2.8	Procesamiento de la señal PPG	22
2.9	Importancia de la señal PPG	24
3	Arduino	25
3.1	Historia de Arduino	25
3.1.1	Plataforma hardware Arduino	26
3.1.2	Plataforma software Arduino	28
3.2	Propósito	34
3.3	Materiales	35
3.3.1	Costo de materiales	37
3.3.2	Consideraciones	37
3.4	Construcción del fotopletismógrafo	39
3.4.1	Modelado en Protoboard	39
3.4.2	Modelado esquemático	41
3.4.3	Modelado PCB	41
3.4.4	Algoritmo sensor PPG	42
3.4.5	Librerías	43
3.4.6	Código fuente base	43
3.5	Verificación del funcionamiento del fotopletismógrafo	46
4	Software	47
4.1	Software	47
4.1.1	Obtención de lectura a través del monitor serial	49
4.1.2	Postprocesado de la señal PPG	54
4.1.3	Inversión de la señal PPG	59
4.1.4	Oportunidades de mejora	66
5	Costos y presupuesto	67

CONTENIDO

5.1	Metodología de desarrollo	67
5.1.1	Etapas	69
5.2	Estimación de costos	69
5.2.1	Materiales	69
5.2.2	Recursos	70
5.3	Planificación de actividades	70
5.3.1	Cronograma de actividades	70
5.3.2	Resumen de actividades	71
5.3.3	Diagrama de Gantt	72
6	Conclusiones y líneas futuras	73
	Referencias	75
	Agradecimientos	81

Lista de figuras

1.1	Algunas señales biológicas del corazón, tomadas de [4].	2
1.2	Esquema de un biosensor, tomado de [10].	4
1.3	Instrumentos médicos, tomados de [11].	5
1.4	Diagrama de bloques de un sistema de instrumentación biomédica, tomado de [12].	6
1.5	Electrocardiograma y su utilización en pacientes, tomado de [13]. .	7
2.1	Interpretación de la ley de Beer-Lambert, tomada de [20].	10
2.2	Proceso de transporte de oxígeno al organismo, tomado de [21]. . .	12
2.3	Curva de disociación de la oxihemoglobina, tomada de [22].	13
2.4	Curva de hemoglobina oxigenada (HbO ₂) y hemoglobina reducida (Hb), tomada de [23].	14
2.5	Representación típica de una señal PPG con ciclo anacrótico y catacrótico, tomada de [25].	18
2.6	Diferentes clases de señal PPG, tomadas de [27]. Las ondas de pulso pueden describir picos diastólicos en jóvenes, que se pueden difuminar a medida que aumenta la edad de la persona.	18
2.7	Sensor PPG en acción, tomado de [28].	19
2.8	Las dos formas de contacto en un fotopletismografo, tomado de [30]. A la izquierda en modo transmisión y a la derecha en modo reflexión.	20
2.9	Diagrama esquemático de un circuito PPG, tomado de [30].	21
3.1	Modelo de Arduino UNO R3 con sus pines, tomado de [33].	26
3.2	Algunos sensores para Arduino, tomados de [34].	27
3.3	Señal analógica y señal digital.	28
3.4	Arduino IDE con código de muestra. Elaboración propia.	29

LISTA DE FIGURAS

3.5	Circuito básico con Arduino y el Script Blink utilizando el pin 13 como salida digital. Elaboración propia.	31
3.6	Arduino UNO R3.	35
3.7	Sensor de Pulso KeyEstudio XD-58C.	36
3.8	Mini protoboard.	36
3.9	Display LCD 16x2 + Módulo I2C.	36
3.10	Cables Dupont macho-macho y macho-hembra.	37
3.11	Lectura PPG Limpia con sensor KeyEstudio XD-58C. Elaboración propia.	38
3.12	Lectura PPG ruidosa con Sensor IDUINO genérico. Elaboración propia.	38
3.13	Sensores PPG con señales obtenidas. Elaboración propia.	39
3.14	Arduino y sensor PPG. Elaboración propia.	40
3.15	Diagrama esquemático Arduino y sensor PPG. Elaboración propia.	41
3.16	PCB Arduino y sensor PPG. Elaboración propia.	41
3.17	Demostración del funcionamiento del sensor Arduino PPG. Elaboración propia.	46
4.1	Comunicación entre Arduino y una PC por monitor serial, tomada de [38].	48
4.2	Aplicación web con Arduino utilizando WebSockets.	53
4.3	Gráfica con datos obtenidos.	55
4.4	Gráfica con interpolación spline.	57
4.5	Gráfica con interpolación cúbica.	58
4.6	Representación de una señal PPG normal, tomada de [42].	59
4.7	Señal PPG invertida obtenida en el monitor serial con la función map.	62
4.8	Señal PPG invertida obtenida en el monitor serial por medio del diferencial.	64
4.9	Señal PPG invertida obtenida en el monitor serial por medio del complemento.	66
5.1	Diagrama de metodología de prototipo, tomado de [45].	68
5.2	Diagrama de Gantt para el desarrollo del fotopleletismógrafo.	72

Lista de tablas

3.1	Precio de componentes para sensor PPG.	37
3.2	Librerías para la elaboración del fotopleletismógrafo.	43
4.1	Librerías de lectura de puerto serial para algunos lenguajes de programación.	48
5.1	Precio de componentes, tomados de la tabla 3.1, para el sensor PPG durante el año 2022.	69
5.2	Recursos utilizados para la creación del proyecto.	70
5.3	Cronograma de actividades para el fotopleletismógrafo.	71
5.4	Resumen de actividades.	71

Lista de algoritmos

1	Blink	30
2	Liquid Crystal	33
3	Arduino PPG	42

Lista de fragmentos de código

3.1	Blink Sketch.	30
3.2	Ejemplo de declaración de librerías y constantes en Arduino.	32
3.3	Arduino PPG Sketch.	43
4.1	Contenido de <code>consumers.py</code>	50
4.2	Contenido de <code>routing.py</code>	51
4.3	<code>live graph script</code>	52
4.4	Gráfico de datos sin tratamiento.	54
4.5	Gráfico de datos con interpolación spline.	56
4.6	Gráfico de datos con interpolación cúbica.	57
4.7	Arduino map function.	60
4.8	Inversión de la señal PPG obtenida.	60
4.9	Inversión de la señal PPG por medio de la función map.	61
4.10	Inversión de la señal PPG obtenida por medio de valores máximos.	62
4.11	Inversión de la señal PPG obtenida por medio del complemento respecto del valor máximo.	64

Lista de acrónimos

QRS Complejo QRS

EKG Electrocardiograma

IDE Entorno de Software Integrado

PPG Fotopletismografía

RX Radiografía

GSR Respuesta Galvánica de la Piel

SNA Sistema Nervioso Autónomo

TAC Tomografía Axial Computarizada

UCC Unidad de Cuidados Críticos

UCI Unidad de Cuidados Intensivos

VFC Variabilidad de la Frecuencia Cardíaca

1

Introducción

En este capítulo se presenta una introducción a las señales biológicas presentes en el cuerpo humano, sus orígenes, las características que poseen y formas de adquisición de las señales biológicas.

1.1 LAS SEÑALES BIOLÓGICAS

Una señal biológica, o bioseñal, es cualquier señal en los seres vivos que se puede medir y monitorear continuamente. Usualmente se asocia el termino bioseñal para referirse a señales bioeléctricas; sin embargo, se puede referir tanto a señales eléctricas como no eléctricas [1]. Dichas señales se pueden representar como medidas escalares o en plano cartesiano. Por ejemplo un médico radiólogo busca anomalías en un paciente por medio de una radiografía de rayos X [2] y un cardiólogo utiliza un instrumento para obtener señales eléctricas del corazón de un paciente por medio de un electrocardiograma [3], o la temperatura de una persona, cuya medición brinda un dato tomando como referencia la nomenclatura de temperatura utilizada (en medicina se utilizan grados Celcius o Fahrenheit).

Las señales biológicas contienen información útil que permite describir características del cuerpo humano; además, pueden describir el funcionamiento de nuestros sistemas internos del cuerpo humano, esto con el fin de apoyar a los diagnósticos médicos o para estudios de nuevas enfermedades o anomalías en pacientes.

1.1. LAS SEÑALES BIOLÓGICAS

También permite detectar anomalías presentes en nuestro organismo y, por tanto, brindar un adecuado tratamiento para el mismo.

Estas señales pueden ser observadas por médicos y científicos mediante pruebas y análisis dando con ello diferentes interpretaciones sobre el funcionamiento del cuerpo humano. A su vez, algunas señales biológicas pueden ser presenciadas por nosotros mismos en el día a día, como es el caso de la temperatura y la presión arterial.

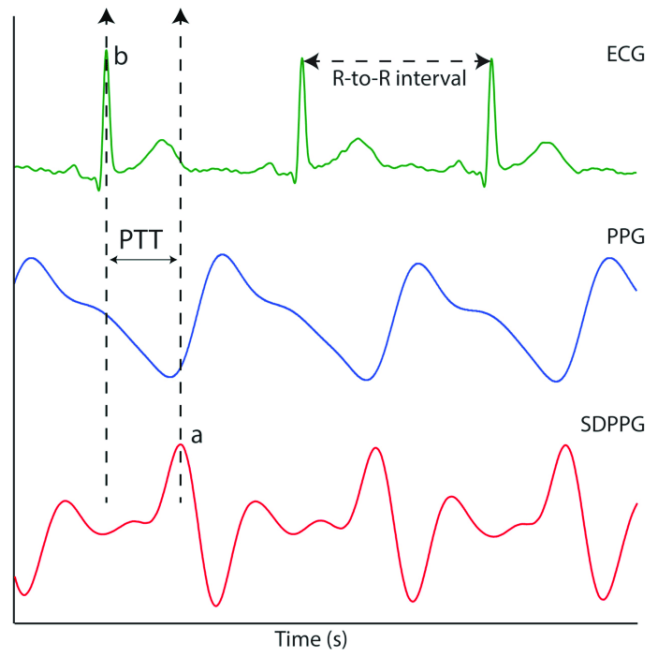


Figura 1.1: Algunas señales biológicas del corazón, tomadas de [4].

En nuestro cuerpo pueden llegar a apreciarse hasta siete diferentes tipos de señales biológicas [5], cada una caracterizada por como se transmiten en el cuerpo humano. Estas señales pueden ser capturadas y traducidas a señales electromagnéticas, ondas mecánicas, radiación ionizante o incluso impulsos eléctricos por medio de un dispositivo especializado para tal fin. Por ejemplo, podemos mencionar que el electrocardiograma es un dispositivo que estudia la actividad eléctrica del corazón detectando los impulsos eléctricos que generan los músculos cardíacos cuando este bombea sangre a través del sistema circulatorio, todo con el fin de obtener diversos indicadores sobre el funcionamiento del corazón de una persona [6].

1.1.1 NECESIDADES DE MONITORIZACIÓN FISIOLÓGICA DE PACIENTES

A mediados de la década de los años 50 en las unidades de atención a pacientes surge la necesidad de mantener una monitorización continua de los signos vitales de los pacientes gracias al surgimiento de la Unidad de Cuidados Intensivos (UCI) y la Unidad de Cuidados Críticos (UCC) y debido a la emergencia de atención de pacientes. Esto dio como resultado los primeros aparatos de monitoreo del paciente y su posterior evolución a aparatos más simplificados y precisos [7].

Inicialmente, en estas unidades se atendían pacientes con enfermedades que requerían intervención inmediata para las afecciones por las cuales se presentaron al hospital. Posteriormente, se perfeccionó el manejo de atención de los pacientes al punto que actualmente con pequeños dispositivos se pueden obtener signos vitales de un paciente [8].

Sumado a ello, en las décadas posteriores se crearon dispositivos capaces de detectar estas señales, inicialmente siendo estos instrumentos analógicos, y con el paso del tiempo se fueron implementando de manera digital gracias a la creación, acompañamiento y perfección de los circuitos integrados. A su vez dichos dispositivos permitían obtener señales fisiológicas de una manera cada vez mas práctica y oportuna, lo cual permitió mejorar los tiempos de atención de las Unidades de Emergencia de los Hospitales e incluso en centros de atención de salud.

La necesidad de monitorear constantemente los signos vitales de los pacientes es la que ha permitido que en el mercado se encuentren diversos dispositivos, tanto médicos como de uso personal, que permiten establecer y vigilar constantemente dichas señales biológicas.

1.1.2 ADQUISICIÓN DE LAS SEÑALES BIOLÓGICAS

La forma de adquisición de las señales biológicas se realiza a través de sensores receptores de la señal que deseamos obtener, y junto a este sensor se encuentra un transductor que se encarga de acondicionar adecuadamente la señal obtenida para convertirla a impulsos eléctricos o mecánicos para finalmente obtener una señal medible a partir de la traducción obtenida. En la figura 1.2 podemos ver un esquema de un biosensor con el que a partir de una muestra a analizar se tiene

1.1. LAS SEÑALES BIOLÓGICAS

información celular la cual se traduce a impulsos eléctricos [9].

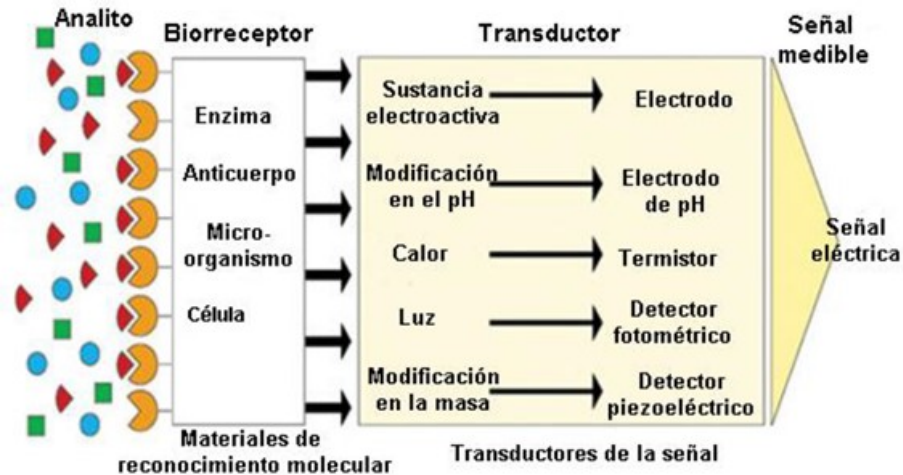


Figura 1.2: Esquema de un biosensor, tomado de [10].

El diseño de los diferentes sensores depende del fabricante y pueden tener diferentes propósitos de uso, como dispositivos de uso personal como Smart Watches o Smart Bands cuya finalidad es brindar información médica del usuario con fines meramente informativos, prescindiendo un poco de la precisión de la información, así como instrumentos orientados a medicina en donde se debe tener en cuenta una precisión exacta sobre las señales que estamos monitorizando, como es el caso de un electrocardiograma o un electroencefalograma.

Si bien estos dispositivos permiten obtener información importante sobre nuestro cuerpo humano y, además, brindan retroalimentación en cuanto al funcionamiento de nuestro organismo como tal, sucede que muchas veces estos dispositivos pueden ser de tipo invasivos en cuanto a la forma en cómo se obtienen los datos. Para nombrar algunos, un electrocardiograma requiere que se coloquen unos electrodos en diferentes áreas del tórax de la persona para poder obtener dicha información dependiendo del nivel de profundidad que se requiera para el estudio o toma de decisión en cuanto a la salud del paciente.

Es debido a la forma de obtención de las señales que se han diversificado las técnicas de medición de señales biológicas, utilizando principalmente variantes de tipo no invasivos. Unas técnicas basadas en ondas mecánicas como la ultrasonografía

o las ecografías, otras basadas en el reflejo de la luz como un oxímetro de pulso y otras fundamentadas en valores de referencia utilizando elementos de la naturaleza, como en el caso de un termómetro de mercurio.



Figura 1.3: Instrumentos médicos, tomados de [11].

Es por ello que podemos mencionar que dentro de las formas de obtener señales biológicas se pueden clasificar en dos formas: a través de métodos invasivos o métodos no invasivos.

Para fines prácticos diremos que los métodos invasivos son aquellos que requieren una intervención en el cuerpo humano para poder llegar al área de estudio de interés. Por mencionar un método de tipo invasivo, el electrocardiograma requiere de conectar en muchos puntos de la caja torácica del paciente electrodos; los métodos no invasivos son aquellos que no requieren de mayor intervención dentro del cuerpo humano, como un ultrasonido o la fotopleletismografía [7].

1.2. CLASIFICACIÓN DE LAS SEÑALES BIOLÓGICAS

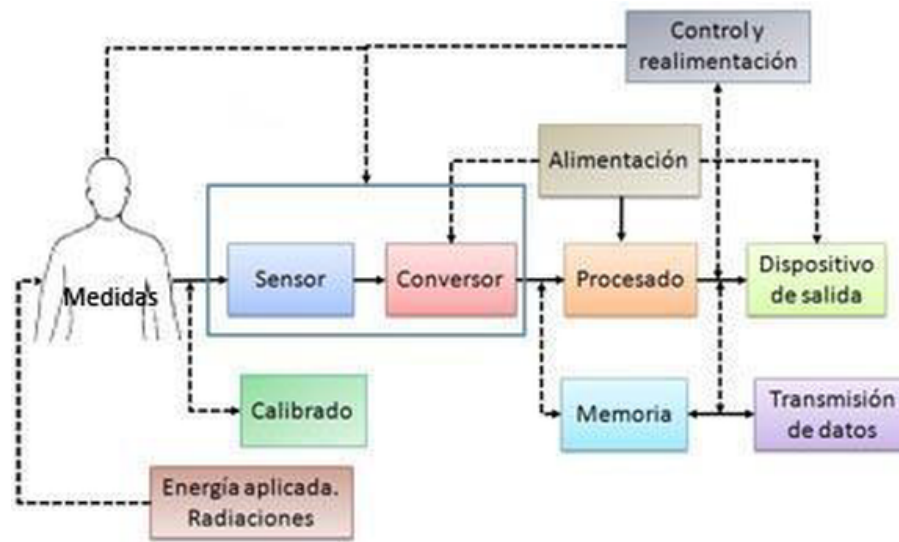


Figura 1.4: Diagrama de bloques de un sistema de instrumentación biomédica, tomado de [12].

1.2 CLASIFICACIÓN DE LAS SEÑALES BIOLÓGICAS

Según Presedo [7] podemos clasificar las señales biológicas de acuerdo con el tipo de variable fisiológica a estudiar o con su frecuencia de adquisición.

1.2.1 TIPO DE VARIABLE FISIOLÓGICA

La primera forma de clasificación sugiere que se maneje como referencia la dimensión de las variables. Podemos mencionar que hay señales de tipo unidimensional, como puede ser la temperatura, la presión arterial, la señal de un Electrocardiograma (EKG) o de un Fotopletismografía (PPG) debido a que interactúa únicamente una variable de medición de la señal. Luego se derivan las imágenes 2D y 3D. Estas ya son mecanismos más avanzados que pueden tener dos o más dimensiones de estudio, como, por ejemplo, una radiografía es una imagen 2D que tiene como referencia la vista del cuerpo humano a nivel de huesos y una débil masa muscular. Otro ejemplo puede ser una Tomografía Axial Computarizada (TAC), cuyo propósito es obtener imágenes de los órganos internos en dos o tres

dimensiones dependiendo del nivel de profundidad que se desee observar.

1.2.2 FRECUENCIA DE ADQUISICIÓN

La otra forma de clasificar las señales biológicas es por medio de la frecuencia de muestreo que esta posea, siendo así que podemos hablar de señales periódicas las cuales fundamentalmente son unidimensionales y con una elevada tasa de muestreo. Por ejemplo, la señal de un EKG. También hay de tipo aperiódica, las cuales pueden ser imágenes y otras señales que son adquiridas de manera manual, como, por ejemplo, una Radiografía (RX).

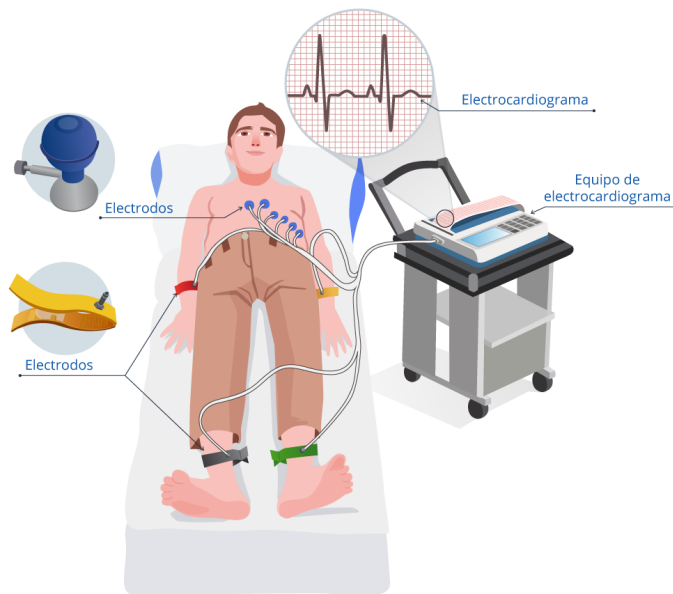


Figura 1.5: Electrocardiograma y su utilización en pacientes, tomado de [13].

1.3 IMPORTANCIA DE LAS SEÑALES BIOLÓGICAS

Como se ha mencionado anteriormente, muchas de las señales biológicas nos pueden brindar información relevante de un paciente, y, sobre todo, puede ayudar a predecir enfermedades patológicas o esporádicas por medio de la obtención de estas. Además, muchas de estas herramientas permiten brindar un diagnóstico más

1.3. IMPORTANCIA DE LAS SEÑALES BIOLÓGICAS

acertado en aquellos casos en los que no se puedan observar anomalías a simple vista.

Sin embargo, hay mecanismos que son invasivos y que requieren de una intervención del paciente para poder obtener dichas señales; es por ello que en este trabajo se brinda una alternativa para obtener mediciones que son igual de valiosas que las de un EKG pero que pueden ser capturadas por medio de mecanismos apenas invasivos, tal como sucede con la señal PPG, cuya caracterización se abordará en profundidad en el capítulo 2.

Si bien la señal de un electrocardiograma mide actividad eléctrica, en el caso de un fotopleletismógrafo se detecta la variación de volumen sanguíneo en un punto de vena o vaso capilar que también guarda relación con la saturación de oxígeno. Esta medición tiene una estrecha relación con otras señales biológicas, como lo es la del electrocardiograma debido a que en el proceso de transportación de oxígeno hacia nuestro sistema circulatorio intervienen tanto la actividad cardíaca como la respiratoria: la actividad cardíaca origina la formación del Complejo QRS (QRS) [14] del EKG que a su vez origina el bombeo de la presión arterial sistólica en la señal PPG ya que es justo el momento en el que se transportan las moléculas de oxígeno a través de nuestro organismo [15].

2

La señal PPG

En este capítulo se introduce la señal PPG o señal fotopletismográfica, sus orígenes fisiológicos, representación, formas de adquisición y procesamiento, y también la importancia que esta señal posee.

2.1 INTRODUCCIÓN

La señal PPG tiene su origen en la absorbancia del tejido. Esta absorbancia depende de la cantidad de tejido que tiene que atravesar un haz de luz. Si esta absorbancia cambia con respecto al tiempo, se puede obtener una señal para estudiar dichos cambios. En el caso de la PPG, la absorbancia cambia con respecto al ciclo cardíaco, lo que origina una variación pulsátil del volumen que se puede registrar. Esta señal puede utilizarse para obtener información sobre el sistema cardiorrespiratorio, como puede ser la frecuencia cardíaca o la saturación de oxígeno en sangre [16].

2.2 ORIGEN FÍSICO

La pletismografía es una técnica de medición cuyo principio se basa en los cambios de volumen y presión que puede sufrir un cuerpo en observación [17].

2.2. ORIGEN FÍSICO

Algunas de estas formas de pletismografía se valen de variaciones en el volumen o en la presión de los órganos del cuerpo humano para establecer diagnósticos médicos con arreglo a resultados de variaciones de volumen. Por ejemplo, una espirometría es capaz de detectar cambios del volumen pulmonar de un paciente simplemente soplando en un tubo todo el aire contenido en los pulmones y tomando valores de referencia de datos históricos de otros pacientes [18].

De acuerdo con Celi, Rocha y M. Yapur [19], podemos decir que «La finalidad de la pletismografía es registrar de una manera no invasiva las variaciones de volumen sanguíneo en las diferentes partes del cuerpo de una persona, especialmente sus extremidades».

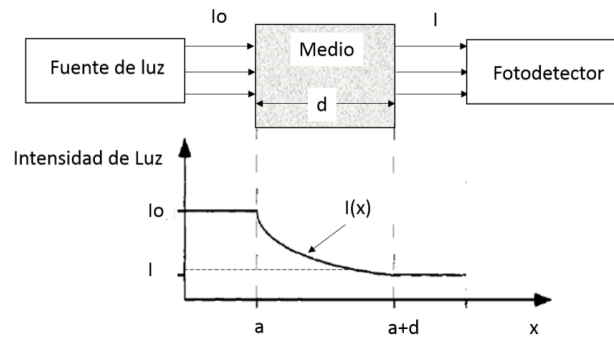


Figura 2.1: Interpretación de la ley de Beer-Lambert, tomada de [20].

Para interpretar la señal PPG que obtendremos nos basaremos en la ley de Beer-Lambert, la cual establece que «La absorbancia es directamente proporcional a la longitud \mathbf{b} de la trayectoria a través de la solución y a la concentración \mathbf{c} del analito o especie absorbente», como indica la ecuación (2.1):

$$A = \epsilon \mathbf{b} \mathbf{c}, \quad (2.1)$$

donde

A es la absorbancia total medida a la longitud de onda λ ,

ϵ es el *coeficiente de absorción* o la absorbancia molar de la sustancia,

\mathbf{b} es la distancia que la luz atraviesa por el cuerpo,

y c es la concentración de sustancia absorbente en el medio.

La ley de Beer-Lambert es una de las leyes de la absorción de la radiación [20], lo que explica que hay una relación exponencial entre la transmisión de luz a través de una sustancia y su concentración, así como también entre la transmisión y la longitud del cuerpo que atraviesa. En este caso podemos inferir que la sustancia que estudiaremos es la sangre y la longitud del medio puede ser los diferentes puntos en donde podemos obtener muestras del volumen sanguíneo.

2.3 ORIGEN FISIOLÓGICO

De acuerdo con la sección anterior, nuestro medio es el cuerpo humano. En este apartado se realizará una explicación más precisa sobre cómo nuestro cuerpo humano manifiesta la señal PPG, para ello se explicará cómo se realiza el transporte de oxígeno, el proceso de ventilación para luego profundizar con la sangre y la hemoglobina.

2.3.1 SISTEMA DE TRANSPORTE DE OXÍGENO

El paso fundamental para el transporte de oxígeno es la ventilación, la acción de mover el aire dentro y fuera de los pulmones. Esto es controlado por el cerebro a través de los mecanorreceptores y quimiorreceptores, que controlan todos los aspectos mecánicos y químicos involucrados en el proceso de respiración. Dichos aspectos pueden ser la medida de inflación de los pulmones, o la captura de oxígeno o dióxido de carbono (CO_2) en sangre, para adecuar la cantidad de oxígeno que ingresa al organismo [16].

La ventilación trabaja como un cambio de presión en la cavidad torácica debido a la contracción de los pulmones. Una presión menor que la presión atmosférica permite que el aire entre a los pulmones, el cual viaja a través de los pulmones encontrándose en un sistema de ramificaciones dividido en bronquios, bronquiolos, conductos alveolares y alvéolos rodeados por la pared capilar por donde el oxígeno se difunde en la sangre.

En los bronquios los glóbulos rojos transportan el oxígeno hacia las venas pulmonares, desde donde viaja al lado izquierdo del corazón, y desde ahí es

2.3. ORIGEN FISIOLÓGICO

bombada hacia el resto del organismo. La sangre con déficit de oxígeno regresa al lado derecho del corazón y es impulsada de nuevo a la arteria pulmonar hacia los pulmones, donde recoge el oxígeno y es liberado el dióxido de carbono [21].

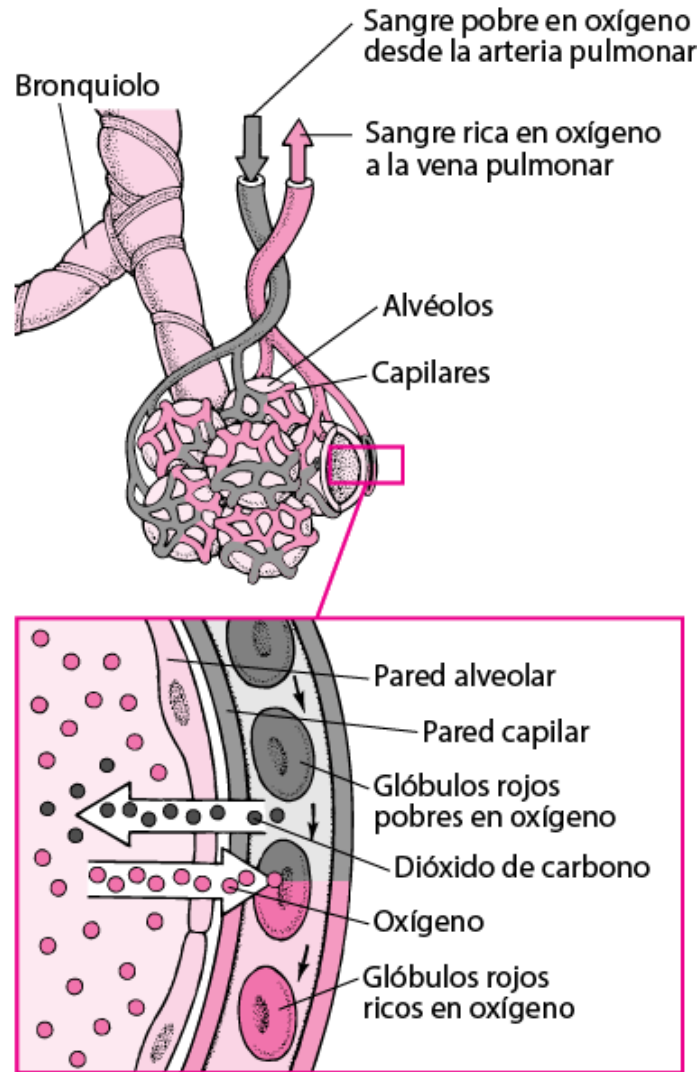


Figura 2.2: Proceso de transporte de oxígeno al organismo, tomado de [21].

2.3.2 HEMOGLOBINA Y OXIGENACIÓN

La sangre está compuesta mayormente por agua; por lo tanto, los gases necesitan una pequeña ayuda para poder viajar. Para ello existe la hemoglobina, una proteína

contenida en los glóbulos rojos que transporta el oxígeno desde los pulmones a los tejidos y órganos del cuerpo. También se encarga de enviar el dióxido de carbono de vuelta a los pulmones.

La hemoglobina puede estar unida o no al oxígeno, y no todas las moléculas de hemoglobina presentes en la sangre oxigenada están unidas a una molécula, esto se puede observar como un cambio de color, siendo de color rojo brillante cuando está unido al oxígeno y rojo oscuro cuando no lo está. Esta característica óptica es la base de la PPG y es fundamental en la oximetría de pulso. Una característica de esta molécula es que una vez que toma o libera una molécula de O_2 , es más fácil tomar o liberar otra.

Una curva de disociación de la hemoglobina es la representación entre la presión parcial de oxígeno en sangre y el porcentaje de oxígeno que se une a la hemoglobina, normalizado sobre el valor máximo, como ilustra la figura 2.3.

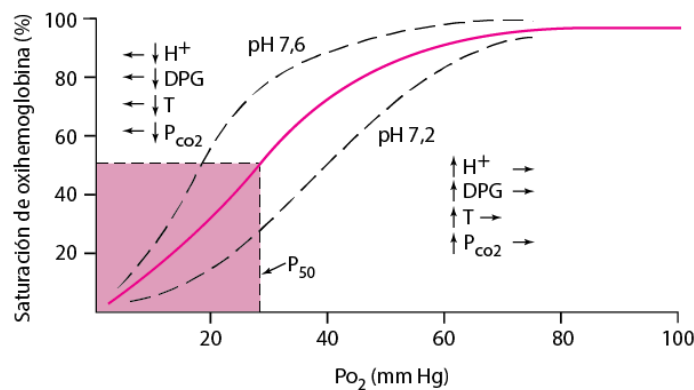


Figura 2.3: Curva de disociación de la oxihemoglobina, tomada de [22].

En esta curva hay algunos puntos que son de especial importancia: a 60 mmHg la saturación de O_2 es del 90 %, a 40 mmHg es alrededor del 75 % y a 27 mmHg es del 50 %. Esta curva puede estar desplazada a la derecha por algunos factores, de entre los cuales se encuentra la disminución del pH, el cual se encuentra relacionado con la disociación de CO_2 .

La cantidad de oxígeno que las moléculas de hemoglobina pueden capturar es lo que se conoce como saturación de oxígeno. Estos valores de saturación es dependiente de la presión parcial de oxígeno presente y del valor del pH presente

2.3. ORIGEN FISIOLÓGICO

en la sangre —se tiene de referencia valores de entre 7 y 8, como una solución neutra—; esto es conocido como efecto Bohr.

2.3.3 PROPIEDADES ÓPTICAS DE LA HEMOGLOBINA

Como se mencionó anteriormente, la hemoglobina muestra un color diferente dependiendo de si está unida al oxígeno o no. Aprovechando esta propiedad, es posible evaluar la saturación de oxígeno por medios ópticos. Además, con la ecuación de Beer-Lambert (véase la ecuación 2.1) podemos ver que los cambios en la intensidad de la luz dependen de la longitud de onda de luz emitida. A partir de esto se puede inferir que al utilizar diferentes longitudes de onda de luz, la absorción no será la misma, ni para la hemoglobina oxigenada (oxihemoglobina) ni para la hemoglobina libre (hemoglobina reducida). Por tanto, dado que tenemos dos cantidades desconocidas, podemos usar dos longitudes de onda diferentes para caracterizar estos valores.

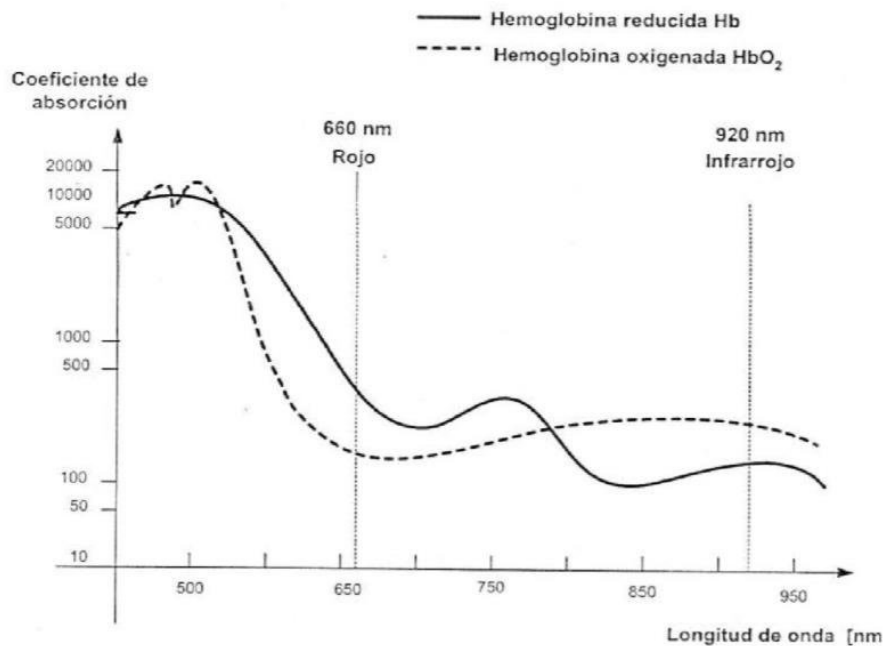


Figura 2.4: Curva de hemoglobina oxigenada (HbO₂) y hemoglobina reducida (Hb), tomada de [23].

Para ello podemos elegir longitudes de onda que pertenecen a la luz roja y la luz infrarroja. Observando las curvas de extinción se puede observar que hay un punto en que la absorción de luz es exactamente igual para ambos tipos de hemoglobina, el cual es llamado punto isosbético. Este punto es algo que debemos evitar, debido a que no se obtendrá información sobre los estados de saturación con esa longitud de onda. Si se considera constante la cantidad total de hemoglobina, obtendríamos un valor constante. Los dos valores de longitud de onda que nos resultan de interés corresponden a:

$\lambda = 660 \text{ nm}$, con luz roja.

$\lambda = 940 \text{ nm}$, con luz infrarroja.

Como se puede observar en la figura 2.4, en 660 nm hay una gran diferencia de absorción entre la oxihemoglobina y la hemoglobina reducida, por lo que esta longitud de onda puede ser utilizada para estimar la presencia de hemoglobina unida al oxígeno. En 940 nm también hay una diferencia, pero en este caso la absorción de hemoglobina reducida es mayor que la de la oxihemoglobina. Estos dos puntos se pueden utilizar para realizar mediciones, siempre que el espectro de absorción sea continuo para ambas hemoglobinas y ambas absorban en cada longitud de onda, al menos un poco. No se pueden utilizar longitudes de onda más altas o bajas ya que esto podría implicar riesgos, tanto en lo que refiere a las mediciones como a la salud.

2.3.4 ESTRÉS

Una parte muy importante del sistema nervioso es el Sistema Nervioso Autónomo (SNA). Este a su vez se divide en sistema nervioso simpático, parasimpático y entérico. El SNA, entre otras cosas, controla diferentes órganos, como las glándulas sudoríparas, y en condiciones estables, los sistemas simpático y parasimpático se equilibran entre sí. En condiciones difíciles, el sistema simpático toma el control del cuerpo para aumentar la frecuencia cardíaca, la producción de las glándulas suprarrenales, o la frecuencia respiratoria, entre otras manifestaciones. La Variabilidad de la Frecuencia Cardíaca (VFC) es el intervalo entre dos picos de dos latidos cardíacos consecutivos; está conectada al sistema nervioso a través de células marcapasos

2.4. FOTOPLETISMOGRAFÍA

que la hacen posible. Este parámetro influye en la señal PPG ya que la señal es consecuencia de la actividad cardíaca. La VFC puede medirse utilizando la señal PPG, y, por tanto, también puede ser utilizada como un marcador de estrés.

2.4 FOTOPLETISMOGRAFÍA

La Fotopletismografía PPG es una técnica de pletismografía en la cual se detecta una variación de volumen utilizando un medio para alterar una propiedad del cuerpo en estudio (generalmente puede ser volumen) y un dispositivo capaz de determinar variaciones de volumen del mismo. Por lo general, un Fotopletismógrafo puede utilizar un haz de luz como medio para evaluar el volumen, y un diodo o fotosensor que es capaz de interpretar dichas variaciones de volumen.

2.5 ORIGEN DE LA SEÑAL PPG

El origen de la Fotopletismografía se remonta al año 1936, donde dos grupos de investigación (Molitor y Kniazuk del Instituto Merck de Investigación Terapéutica de Nueva Jersey, y Hanzlik *et al.* de la Escuela de Medicina de la Universidad de Stanford) describieron instrumentos similares para monitorear cambios en el volumen de la sangre en la oreja de un conejo después de la oclusión venosa. Al mismo tiempo Molitor y Kniazuk también describieron grabaciones realizadas con un sistema PPG en modo reflejo de dedos humanos. En 1937 un equipo de investigadores liderado por Alrick Hertzman del Departamento de Fisiología de la Universidad de St. Louis, presentaron un artículo sobre la PPG, describiendo el uso de un sistema en modo de reflexión para medir cambios en el volumen de sangre en los dedos inducidos por la maniobra de Valsalva, ejercicio y exposición al frío [24].

Posterior a ello se detectaron potenciales fuentes de error de medición, de entre las cuales Hertzman enfatizó que se necesita un buen contacto con la piel, pero sin una presión excesiva que pudiera resultar en un blanqueamiento de la señal. También aconsejó que se debe evitar el movimiento de la sonda de medición en la piel. Ello conllevó que posteriormente se desarrollaran dispositivos que mejoraran

el posicionamiento del mismo.

Hertzman también utilizó una bombilla de linterna alimentada por batería, que no era ideal debido a su espectro de iluminación relativamente amplio, particularmente el infrarrojo, el cual como se describió en la sección anterior, brinda medición de hemoglobina reducida en lugar de la oxihemoglobina, los errores debido a los efectos de la saturación de oxígeno en sangre, además del calentamiento local que ocasionaba dicha bombilla.

En décadas más recientes surgió la necesidad de emplear técnicas de evaluación no invasivas pequeñas, confiables, de bajo costo y de fácil aplicación, lo que permitió que se desarrollaran dispositivos cada vez mejores para utilizarse en la Fotopletismografía. Diferentes avances en optoelectrónica, e instrumentación clínica también contribuyeron significativamente a su avance. Las mejoras también en semiconductores, diodos emisores de luz, fotodiodos y fototransistores han permitido realizar mejoras cada vez más considerables en el tamaño, sensibilidad y confiabilidad en el diseño de sondas PPG [24].

2.6 REPRESENTACIÓN DE LA SEÑAL PPG

La señal PPG exhibe un patrón cuasiperiódico que consiste en una onda de pulso arterial para cada latido del corazón.

Esta onda de pulso se asemeja a una onda de pulso de presión sanguínea arterial, pero con diferencias importantes en cuanto al contorno de la onda. Cada onda de pulso PPG consta de dos fases distintas: la fase anacrótica y catacrótica, correspondientes a las ramas ascendente y descendente respectivamente [26].

La señal PPG consta de dos picos, el pico sistólico y el pico diastólico. Como en la electrocardiografía, la señal PPG muestra las manifestaciones diastólica/sistólica de nuestro corazón. A su vez muestra un intervalo temporal aparentemente periódica entre picos sistólicos, consecuencia de los latidos que nuestro corazón ejerce al momento de bombear sangre. Tanto los valores de los picos sistólicos y diastólicos como el intervalo entre picos pueden verse afectados por las siguientes condiciones:

- El corazón: este es el responsable de la eyección cardíaca, incluida la frecuencia cardíaca, el ritmo cardíaco y el volumen sistólico.

2.6. REPRESENTACIÓN DE LA SEÑAL PPG

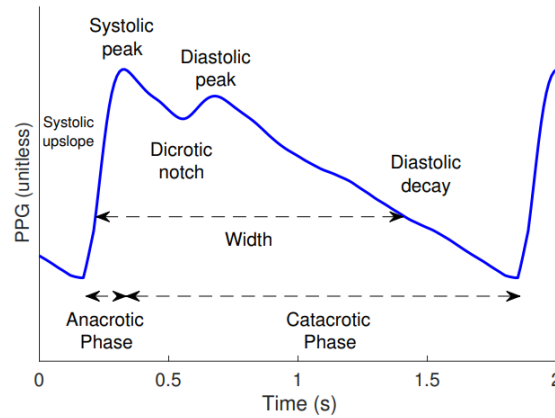


Figura 2.5: Representación típica de una señal PPG con ciclo anacrótico y catacrótico, tomada de [25].

- La circulación: Esta es responsable de las propiedades cardiovasculares, como la rigidez arterial y la presión arterial.
- Otros procesos fisiológicos: como la respiración y el Sistema Nervioso Autónomo SNA, que puede verse afectado por el estrés y otras enfermedades.

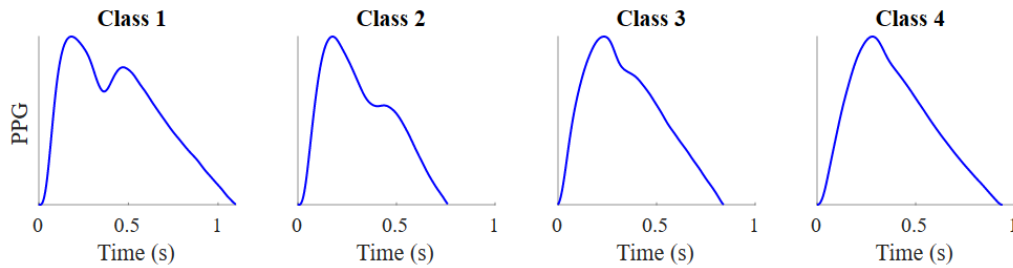


Figura 2.6: Diferentes clases de señal PPG, tomadas de [27]. Las ondas de pulso pueden describir picos diastólicos en jóvenes, que se pueden difuminar a medida que aumenta la edad de la persona.

Además de los factores previamente mencionados, la señal PPG puede presentarse de diferentes formas, en donde paulatinamente se difumina el pico sistólico y diastólico, esto se observa en cambios en el ciclo diacrótico [26]. En la figura 2.6 se puede observar.

2.7 ADQUISICIÓN DE LA SEÑAL PPG

La señal PPG es la medición que se obtiene utilizando la fotopleletismografía. En esencia se mide la variación volumétrica que experimenta el flujo sanguíneo en un punto considerado. En la figura 2.7 se puede observar cómo funciona un sensor PPG y a su vez se muestra la señal PPG capturada por la misma.

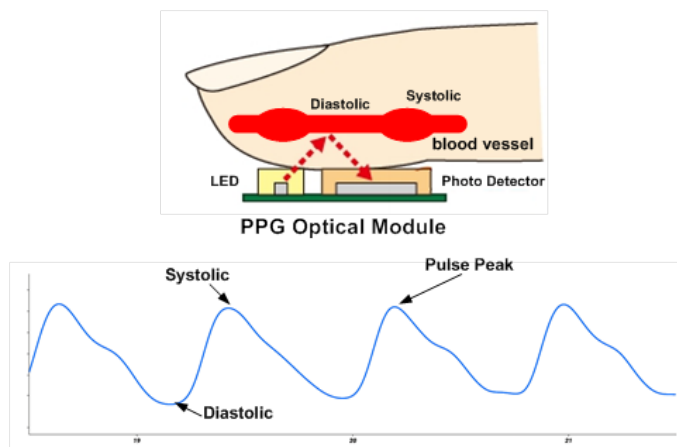


Figura 2.7: Sensor PPG en acción, tomado de [28].

Para realizar la captura de la señal PPG se utilizan diferentes sensores PPG, con dos tipos usuales de configuración, los cuales se tratarán en la subsección siguiente.

2.7.1 CONFIGURACIONES DE SENSORES PPG

La fotopleletismografía generalmente se realiza mediante la detección de contacto con el tejido, la configuración del contacto puede ser en modo de transmisión (transiluminación) o reflexión (adyacente). Generalmente se utiliza un fotodetector de alta sensibilidad y de un emisor de luz lo suficientemente fuerte para que el dispositivo pueda detectar dichas variaciones en el área a estudiar. Por lo general se utiliza luz verde para detectar dichos cambios aunque también hay algunos dispositivos PPG con luz ultravioleta. En la figura 2.9 se puede apreciar un circuito de un dispositivo PPG. El principio de la fotopleletismografía es detectar los cambios de pigmentación de la hemoglobina, en diferentes puntos de interés en la persona.

2.7. ADQUISICIÓN DE LA SEÑAL PPG

Esto se logra debido a que la hemoglobina presenta una absorción de la luz diferente al resto de los tejidos bajo ciertas longitudes de onda [29]. Las fluctuaciones ocurren por la distribución de proteínas y hemoglobina que viaja en la sangre debido a la oxigenación de la misma. Cuando se bombea sangre al cuerpo, el flujo que llega a los vasos capilares cercanos a la superficie de la piel aumenta, logrando así una mayor absorción de luz. En cambio, cuando la sangre regresa al corazón, el flujo disminuye, y asimismo disminuye la absorción de la luz. La finalidad de la PPG consiste en medir la variación volumétrica que experimenta el flujo sanguíneo y, no en pocas ocasiones, de la alteración del oxígeno en sangre que se envía a través de las venas.

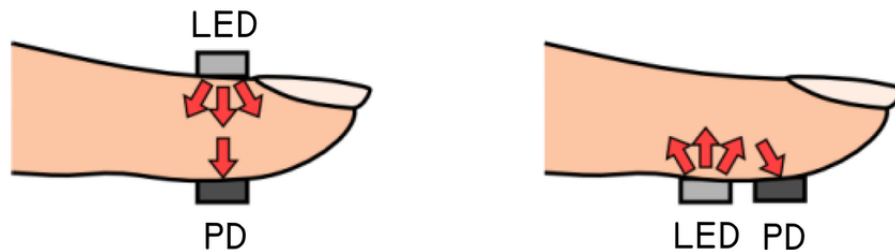


Figura 2.8: Las dos formas de contacto en un fotopletismógrafo, tomado de [30]. A la izquierda en modo transmisión y a la derecha en modo reflexión.

Actualmente existen a la venta diferentes tipos de sensores PPG con diferentes tipos de calibración y a su vez, con notables mejoras en el circuito integrado, tales como menor espacio entre diodo y luz, o la mayor precisión posible. Hay otros dispositivos que se pueden auxiliar con una pinza de sujeción para evitar movilidad en el punto de contacto, lo que a su vez permite también evitar el ruido u otras interferencias de la señal PPG.

Hay una serie de consideraciones que se deben tomar en cuenta al momento del diseño y elección del sensor PPG, entre los que se incluyen: si el modo reflexión o transmisión es el más adecuado para la aplicación que necesitemos, cómo evitar efectos de ruido o distorsión de la señal, la posición del dispositivo, la presión de la interfaz sonda-tejido debe ser lo suficiente para que el sensor se mantenga firme y en su lugar para minimizar movimientos, fuga de luz o permitir que otras fuentes de luz afecten a la lectura del sensor. También, que dicha presión a su vez no sea

tan alta como para distorsionar las características principales de la onda de pulso.

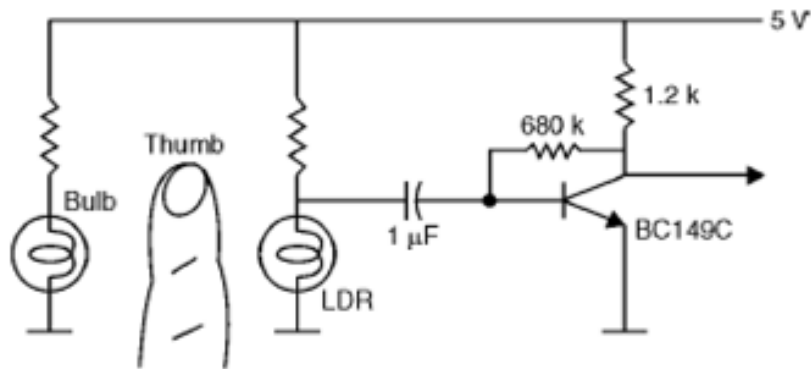


Figura 2.9: Diagrama esquemático de un circuito PPG, tomado de [30].

2.7.2 ADQUISICIÓN SIMULTÁNEA DE SEÑALES

La forma de onda de la señal PPG provee una gran cantidad de información sobre la hemodinámica cardiovascular y se puede utilizar para obtener varios parámetros vitales, como la frecuencia cardíaca, frecuencia respiratoria, saturación de oxígeno arterial y la presión arterial. Sin embargo, el verdadero potencial de la señal PPG queda demostrado cuando se puede combinar con otras señales biológicas como el electrocardiograma EKG, la acelerometría y la Respuesta Galvánica de la Piel (GSR) cuando se adquieren dichas señales simultáneamente [26].

La combinación de diferentes señales puede ofrecer la oportunidad de estudio de estrés mental, junto con la hemodinámica muscular, mientras que la señal de acelerometría se puede usar para eliminar la interferencia de movimiento de la señal PPG.

Entre los parámetros que se pueden obtener utilizando la PPG tenemos los siguientes:

- **Nivel de oxígeno en sangre:** este se obtiene con luz roja y cercana a la infrarroja debido a que esta luz es la más sensible en la piel y además es susceptible a cambios en el nivel de oxígeno en sangre.

2.8. PROCESAMIENTO DE LA SEÑAL PPG

- **Ritmo cardíaco:** este parámetro a monitorizar es fundamental en la medición de signos vitales de un paciente. Puede verse afectado por el movimiento o por una arritmia cardíaca.
- **Presión sanguínea:** este parámetro a monitorizar es fundamental de cara a enfermedades o problemas cardiovasculares.
- **Respiración:** este parámetro a monitorizar es esencial en el cuidado de recién nacidos o en estudios de sueño.

2.8 PROCESAMIENTO DE LA SEÑAL PPG

Una vez obtenida la señal PPG se pueden ejecutar diferentes algoritmos que nos permitan obtener información sobre la señal obtenida. Existen diferentes tipos de algoritmos que permiten acondicionar cualquier cantidad de valores obtenidos, como puede ser un algoritmo de regresión, y en el caso de la señal PPG no es la excepción. Uno de los algoritmos más utilizados es, por ejemplo, un filtro de reducción de ruido. Se pueden utilizar diferentes algoritmos que permitan obtener valores como valores máximos y mínimos de la curva descrita; se pueden obtener también de manera aproximada la función derivada de una serie de puntos obtenidos o una segunda derivada. Se pueden calcular además las diferencias de tiempo entre intervalos de latidos, para poder evaluar la actividad cardiorrespiratoria de la señal obtenida. Diferentes métodos matemáticos se pueden utilizar para estudiar nuestra señal PPG.

Dentro de todo el amplio espectro de procesamiento de la señal PPG tenemos las siguientes cualidades a obtener [24]:

1. Preprocesamiento:
 - Filtrado digital o reducción de ruido.
 - Eliminar artefactos de movimiento.
 - Calidad de la señal y su evolución.
 - Cálculo de derivadas.
 - Descomposición empírica modal.
2. Análisis en el dominio del tiempo:

- Identificación de ondas de pulso individuales.
- Identificación de tiempos fiduciaros.
- Cálculo de características de ondas de pulso.
- Obtención de medidas representativas a partir de registros.
- Identificación de ondas de pulso anormales.
- Análisis de señales simultáneas.

3. Análisis en el dominio de la frecuencia:

- Análisis espectral.
- Análisis tiempo-frecuencia.

4. Machine Learning:

- Regresión lineal.
- Algoritmos basados en árboles.
- Máquinas de vector soporte (SVM).
- Redes neuronales y deep Learning.

5. Análisis no lineal del espectro de fases.

6. Estimación de parámetros fisiológicos:

- Ritmo cardíaco.
- Intervalos entre latidos.
- Detección de fibrilación auricular.
- Variabilidad de frecuencia de pulso.
- Frecuencia respiratoria.
- Presión arterial.
- Rigidez arterial.
- Temporización de ondas de pulso.
- Amplitud de onda de pulso.
- Forma de onda de pulso.
- Combinación de características de onda de pulso.

7. Tiempo de eyección del ventrículo izquierdo.

2.9 IMPORTANCIA DE LA SEÑAL PPG

La señal PPG nos brinda una importante cantidad de parámetros en materia de salud, la cual nos permite conocer datos sobre la oxigenación en sangre de pacientes, también brinda información cardiorrespiratoria del paciente. En uso con más sensores puede brindar señales combinadas para estudiar ciertas condiciones como el estrés crónico u otros factores. Además, la señal PPG tiene la ventaja de que se puede obtener de manera relativamente sencilla con sensores no invasivos para el cuerpo humano por la cual es una de las técnicas de monitorización de signos vitales de pacientes en el área de salud con el oxímetro de pulso.



Arduino

Este capítulo describe la plataforma Arduino, una breve explicación de la misma y cómo se utilizará dicho hardware para elaborar nuestro fotopletismógrafo bajo la filosofía de hardware libre.

3.1 HISTORIA DE ARDUINO

Arduino es una plataforma de desarrollo de circuitos basado en una placa electrónica con un microcontrolador programable y una serie de pines hembra. Estos pines permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera sencilla [31].

La plataforma Arduino nació como una iniciativa para la elaboración de circuitos electrónicos de una manera rápida y bajo el principio de «soldar poco» (soldier less) orientada principalmente a entusiastas de la electrónica y a estudiantes de los campos de la electricidad. Para lograr este fin, la fundación Arduino provee todas las herramientas tanto de hardware como de software publicadas bajo una licencia basada en la General Public License (GPL), que consiente el uso, distribución, recreación y modificación de todos los componentes para uso personal y además se permite la distribución y modificación bajo ciertas condiciones estipuladas en su licencia [32].

La fundación Arduino pone a nuestra disposición diferentes modelos de Arduino

3.1. HISTORIA DE ARDUINO

para diferentes propósitos.

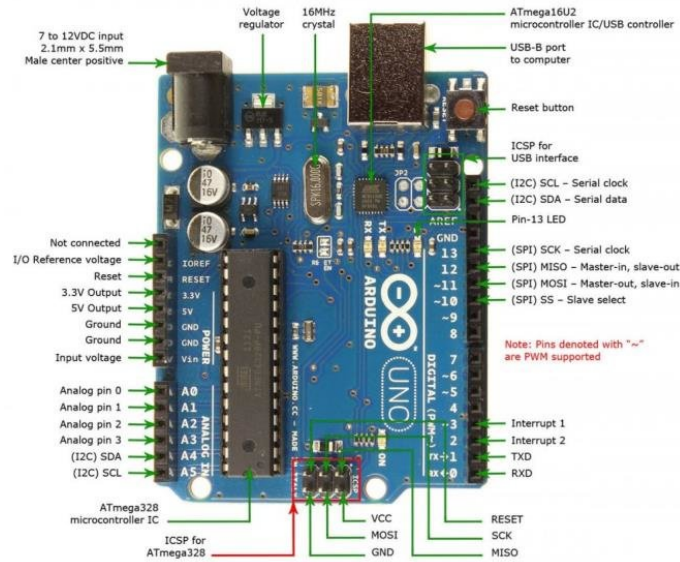


Figura 3.1: Modelo de Arduino UNO R3 con sus pines, tomado de [33].

El modelo más básico es el Arduino UNO R3 (véase la figura 3.1), el cual consta de 14 pines digitales, 6 pines analógicos y unos pines adicionales para maquetar nuestros circuitos, este modelo es perfecto para iniciarse en la plataforma y es el más adecuado para la mayoría de proyectos que se pueden elaborar con Arduino.

Para comprender el funcionamiento de la plataforma Arduino se hace la abstracción de sus dos elementos principales los cuales garantizan el funcionamiento de la misma: el hardware y el software.

3.1.1 PLATAFORMA HARDWARE ARDUINO

El hardware de Arduino consta de un circuito impreso en PCB (Printed Circuit Board), el cual consta de un microcontrolador programable por medio de un controlador USB el cual con el software de Arduino IDE se envía el código elaborado a la placa Arduino. Arduino también tiene diferentes pines hembra para diferentes usos, entre los cuales se tienen pines con lectura analógica, pines de salida digital, salida de corriente directa, conexión a tierra (o línea neutra) los

cuales por medio de estos podemos conectar diferentes sensores, luces, actuadores o cualquier dispositivo electrónico que nos permita prototipar nuestros circuitos.

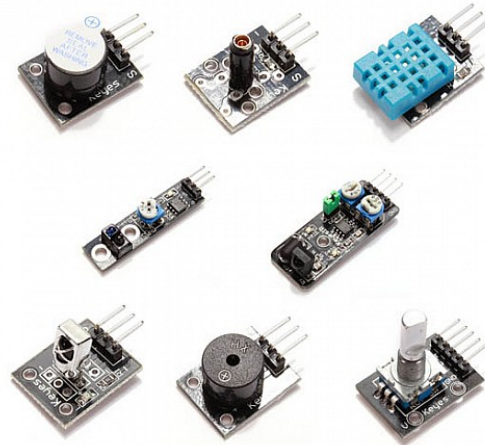


Figura 3.2: Algunos sensores para Arduino, tomados de [34].

A su vez, el hardware de Arduino es capaz de interactuar con diferentes circuitos electrónicos, como lo son sensores, pantallas LCD, motores, pulsadores y luces LED, los cuales permiten construir nuestro circuito auxiliándose de una placa de prototipado llamado protoboard. Puede haber diferentes sensores los cuales pueden interactuar con factores ambientales o con intervención humana (véase la figura 3.2), como por ejemplo sensores de temperatura, pantallas OLED o LCD, sensores ultrasónicos, sensores RFID, motores o servomotores, buzzers, diodos LED, pulsadores, etc. Dependiendo del propósito del circuito, estos nos brindarán información desde Arduino o retroalimentación que proviene de los sensores.

ENTRADAS Y SALIDAS ANALÓGICAS Y DIGITALES

La plataforma de Arduino es capaz de interactuar con entradas y salidas de tipo analógicas y digitales [35]. Una señal analógica permite obtener señales eléctricas de tipo variables en un rango de voltaje de entre 0 V y 5 V. Estas variaciones de voltaje se traducen en Arduino a un valor entre 0 y 1023. La lectura de este valor se

3.1. HISTORIA DE ARDUINO

hace por medio de software ejecutando una función que permite obtener el valor del sensor en tiempo de ejecución. Luego están las señales digitales las cuales están basadas en el sistema binario, cuya representación se da por medio de valores binarios oscilando entre cero y uno, siendo cero una ausencia de corriente y uno un flujo de corriente. En la práctica, en algunos circuitos puede tomarse de referencia un “cero” lógico un valor menor a 2.5 V o cuanto menos 0 V. Y un valor como “uno” lógico un valor mayor a 2.5 V hasta 5 V como máximo. La representación de dichas señales se puede ver en la figura 3.3.

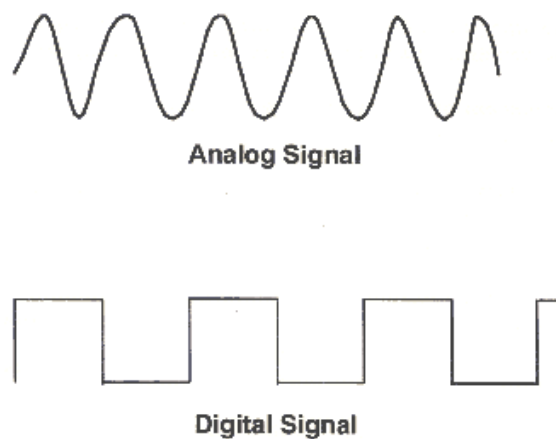


Figura 3.3: Señal analógica y señal digital.

Las lecturas de estos sensores puede obtenerse por medio de los pines que tiene incorporados la placa de Arduino (véase la figura 3.1). Dichos pines pueden configurarse programáticamente por medio de código en Arduino como entradas y salidas; en el caso de las entradas o salidas analógicas se recomienda la conexión a los pines analógicos, mientras que los pines que serán utilizados como entradas o salidas digitales se deben configurar en los pines digitales.

3.1.2 PLATAFORMA SOFTWARE ARDUINO

Una vez explicado el funcionamiento de la parte del hardware de Arduino, se dará una explicación a como interactúan los componentes de Arduino y la Placa PCB. Esto se realiza por medio del Software Arduino IDE, el cual es un Entorno

de Software Integrado (IDE) que permite que podamos brindarle instrucciones a nuestra placa Arduino por medio de un lenguaje de programación basado en el lenguaje C++.



```

36
37 void loop() {
38   ArduinoCloud.update();
39   if(moveServo){
40     loopServo();
41   }
42 }
43
44 void loopServo(){
45   unsigned long msNow = millis();
46   if(msNow - lastServoMove > SERVO_MOVE_INTERVAL){
47     int direction = garage ? 1 : -1;
48     currentAngle += direction * degreeSteps;
49     if(currentAngle > ANGLE_MAX || currentAngle < ANGLE_MIN){
50       moveServo = false;
51       currentAngle = (direction > 0) ? ANGLE_MAX : ANGLE_MIN;
52     }
53     Serial.println(currentAngle);
54     garageDoorServo.write(currentAngle);
55   }
56 }
57
58 void onGarageChange(){
59   Serial.print("Garage switch ");
60   Serial.println(garage ? "ON" : "OFF");
61   moveServo = true;

```

Figura 3.4: Arduino IDE con código de muestra. Elaboración propia.

Al ser un lenguaje de programación tenemos a disposición instrucciones de tipo secuenciales, selectivas y repetitivas. También tenemos a disposición diferentes librerías para poder interactuar de una manera rápida con todos los componentes de hardware que tengamos a disposición. El IDE cuenta con una herramienta que permite explorar las librerías que tenemos disponibles. Además, permite la instalación de librerías adicionales.

Las librerías de Arduino permiten utilizar funciones para facilitar la interacción de los sensores o demás componentes. Sin embargo, no siempre es necesario el uso de librerías; a veces con funciones e instrucciones propias de Arduino podemos realizar tareas básicas como la configuración de pines y salida de datos.

Arduino IDE provee además un monitor Serial, el cual es una herramienta

3.1. HISTORIA DE ARDUINO

que permite observar la información que es capturada por los sensores y cómo se presenta. El monitor se utiliza principalmente en salidas analógicas. La forma de visualización del monitor Serial puede ser a través de los datos leídos o por medio de un gráfico de las lecturas del sensor.

ESTRUCTURA DE UN CÓDIGO ESCRITO EN ARDUINO: BLINK SKETCH

En Arduino el código que escribimos se llama Sketch. Cada Sketch consiste en una declaración de librerías a utilizar en el mismo, y además tiene dos funciones que ejecuta la placa Arduino siempre que va a inicializarse. Estas funciones son `setup` y `loop`. Para explicar la estructura básica de un Sketch en Arduino usaremos de referencia uno de los Sketchs de ejemplo que provee el software Arduino IDE, el cual se llama «Blink», cuyo propósito es hacer que una luz led alterne entre encendido y apagado cada segundo. El algoritmo básico para el código del Sketch Blink está descrito en el Algoritmo 1.

Algoritmo 1 Blink

Input: none
Output: none
pin ← 13
set pin as OUTPUT
loop
 print Turn pin on
 Wait 1 second
 print Turn pin off
 Wait 1 second
end loop

El prototipo del circuito blink se puede ver en la figura 3.5. El código de ejemplo para que Arduino pueda realizar un parpadeo de una luz LED que tiene integrada en el circuito se puede ver en el Sketch 3.1.

```
1 // Language: Arduino (C++)
2 /*
3 Blink
4 Turns an LED on for one second, then off for one second, repeatedly.
5 This example code is in the public domain.
6 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
```

```

7  */
8
9  // the setup function runs once when you press reset or power the board
10 void setup() {
11     // initialize digital pin LED_BUILTIN as an output.
12     pinMode(LED_BUILTIN, OUTPUT);
13 }
14
15 // the loop function runs over and over again forever
16 void loop() {
17     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the
18     voltage level)
19     delay(1000); // wait for a second
20     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making
21     the voltage LOW
22     delay(1000); // wait for a second
23 }

```

Código 3.1: Blink Sketch.

El fragmento de código anterior permite que en el pin 13 (también utilizado como LED_BUILTIN) se emitirá una salida alta de energía al Pin 13 y al LED de Arduino durante 1 segundo y luego emitirá una salida baja de energía al Pin 13 y al LED durante 1 segundo.

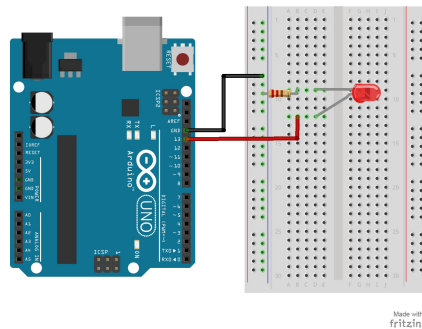


Figura 3.5: Circuito básico con Arduino y el Script Blink utilizando el pin 13 como salida digital. Elaboración propia.

En la figura 3.5 se puede ver una breadboard con una extensión del pin 13 conectado a un LED para simular el comportamiento del parpadeo a través del

3.1. HISTORIA DE ARDUINO

LED en el circuito diseñado. Se debe utilizar una resistencia de al menos 240Ω debido a que Arduino tiene salida de voltaje de 5 V, dado que al estar directamente conectado el LED con el Arduino, hará que el LED quede averiado en cuestión de segundos.

Fundamentalmente todos los Sketchs de Arduino se encuentran conformados por la siguiente secuencia de instrucciones:

- **Declaración de librerías.**
- **Función setup** (se ejecuta una sola vez en el ciclo de vida de Arduino).
- **Función loop** (se ejecuta repetidamente en el ciclo de vida de Arduino).

DECLARACIÓN DE LIBRERÍAS

Esta sección refiere usualmente a las primeras líneas de código del Sketch y sirve para invocar librerías que nosotros necesitaremos para nuestro proyecto.

También esta sección suele utilizarse para declarar variables que serán utilizadas como constantes dentro de nuestro proyecto, por ejemplo, pines que serán entradas o salidas para nuestro proyecto u otras variables que serán de proceso en nuestro Sketch.

```
1 // Language: Arduino (C++)
2 // include the library code:
3 #include <LiquidCrystal.h>
4
5 // initialize the library by associating any needed LCD interface pin
6 // with the arduino pin number it is connected to
7 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
8 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
9 // TODO Setup and Loop
```

Código 3.2: Ejemplo de declaración de librerías y constantes en Arduino.

En el código 3.2 tenemos la declaración de la librería `LiquidCrystal.h`, la cual permite que podamos utilizar Arduino con un Display LCD para salidas de datos por este medio. Además de ello tenemos la declaración de constantes que se utilizarán tanto para la instancia de un objeto `LiquidCrystal` que será el

que contendrá las funciones para utilizar con el Display LCD. En el caso de las variables se está diciendo que se utilizarán los pines 12, 11, 5, 4, 3 y 2 como pines reservados para nuestro Display LCD, Por lo tanto, en la parte del hardware tendremos que asegurarnos de conectar dichos pines a nuestro Display para que funcione correctamente. La secuencia de pasos para la declaración de la librería LiquidCrystal está en el Algoritmo 2.

Algoritmo 2 Liquid Crystal

```

rs ← 12
en ← 11
d4 ← 5
d5 ← 4
d6 ← 3
d7 ← 2
LiquidCrystal lcd(rs, en, d4, d5, d6, d7) {Creates a new LiquidCrystal instance}

```

FUNCIÓN SETUP

La función setup es llamada una sola vez en el Sketch cuando se inicia la ejecución de la aplicación [36]. Se utiliza para inicializar variables, configurar los modos de los pines, instanciar librerías, etc. Esta función solo se ejecuta una vez después de encender o reiniciar la placa Arduino.

En el Sketch 3.1 se puede observar que se está configurando el pin 13 como salida de Arduino; con la función pinMode podemos definir si el pin tendrá comportamiento de entrada o de salida, tanto en digital como en analógico. Las constantes para definir si será entrada o salida son: INPUT y OUTPUT.

La función setup instancia el uso de cada uno de los pines como entrada o salida respectivamente. También permite la configuración del monitor Serial el cual nos permitirá la salida de datos de las lecturas de los sensores analógicos a nuestro IDE Arduino, todo esto con fines de depuración o para validar la información obtenida por el sensor dentro de la PC.

3.2. PROPÓSITO

FUNCIÓN LOOP

La función loop se ejecuta después de realizar el llamado a la función setup. Esta función se ejecuta repetidamente hasta que apaguemos la placa Arduino o presionemos el botón de reiniciar [37].

En la función loop es donde nosotros escribiremos toda la lógica de nuestro hardware y en donde se realizarán todas las interacciones que tendrá nuestro prototipo. La idea de construir la lógica del circuito en la función loop es debido a que nuestro circuito realizará tareas repetidas en todo su tiempo de trabajo. Por tanto, se debe diseñar el prototipo pensando en las tareas que este realizará dentro de la función loop.

En la mayoría de los Sketchs se respeta la convención de programar las funciones setup y loop. Sin embargo, nosotros también podemos escribir funciones extras que nos permitan modularizar acciones dentro de nuestro Sketch.

En el caso del código 3.1, en la parte de la función loop es donde nosotros detallamos los pasos que hará la placa Arduino para que el LED pueda realizar el parpadeo respectivo. Este bucle se mantendrá en ejecución hasta que se reinicie la placa Arduino o se desconecte el circuito de la PC o de una fuente de energía.

3.2 PROPÓSITO

La finalidad de elegir la plataforma de Arduino para el desarrollo del fotopletismógrafo es principalmente por el coste de los materiales. Arduino es una plataforma libre en la cual todas las personas tienen acceso gratis a todas las herramientas y códigos fuente que se deseen utilizar. Asimismo, se tiene acceso a diferentes fabricantes de sensores y también a diferentes kits de Arduino con componentes básicos para poder iniciarse en el mundo de la electrónica.

El coste para los sensores y dispositivos compatibles con Arduino siguen la filosofía de Open Source Hardware, el cual dicta que todos los diseños y diagramas esquemáticos quedan públicos para que cualquiera pueda estudiarlos, modificarlos y distribuirlos. Además de poder producir y vender hardware basado en el diseño original, vale la pena comentar también que hay diferentes fabricantes de placas Arduino las cuales toman de base una placa Arduino original y añaden mejoras y/o

características adicionales sobre dichas placas, pudiéndose distribuirse y venderse como fabricantes de terceros, como en el caso del fabricante KeyEstudio, que además de brindar sensores, provee también placas Arduino con funcionalidades adicionales.

Los componentes a utilizar son de relativo acceso para todo el público y pueden ser adquiridos en cualquier tienda de componentes electrónicos local o en línea. Asimismo, hay diferentes fabricantes de componentes para Arduino; por tanto, se tiene variedad de componentes disponibles en el mercado.

También cabe destacar que los dispositivos como oxímetros de pulso o relojes inteligentes, los cuales integran sensores para fotopletismografía, por lo general, son circuitos cerrados con licencia propietaria, y no se puede conocer el funcionamiento interno de los mismos, así como tampoco se puede acceder a la información que ellos capturan. En el caso de los relojes inteligentes se tiene que utilizar una aplicación propietaria que procesa dichos datos y muestran la información que el fabricante considera oportuna. En el caso de Arduino no sucede esto, debido a que nosotros diseñamos el circuito y podemos extenderlo tanto como lo necesitemos o lo deseemos utilizar.

3.3 MATERIALES

Para la construcción del fotopletismógrafo básico se necesitan los siguientes componentes.

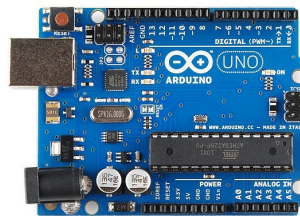


Figura 3.6: Arduino UNO R3.

3.3. MATERIALES



Figura 3.7: Sensor de Pulso KeyEstudio XD-58C.

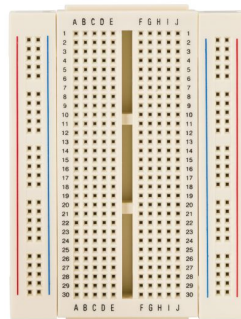


Figura 3.8: Mini protoboard.

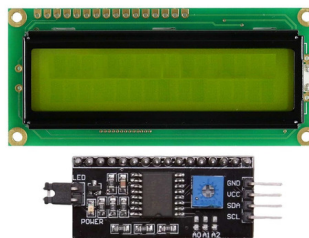


Figura 3.9: Display LCD 16x2 + Módulo I2C.

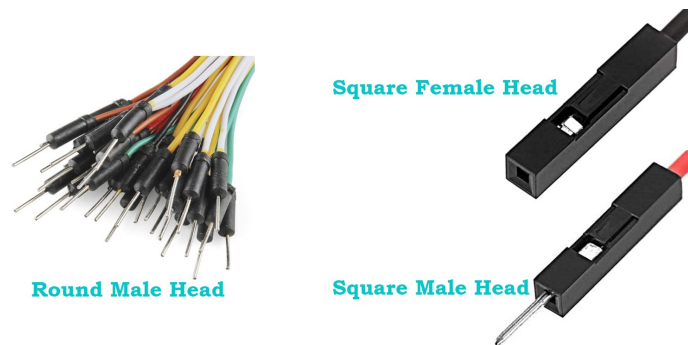


Figura 3.10: Cables Dupont macho-macho y macho-hembra.

3.3.1 COSTO DE MATERIALES

COMPONENTES	PRECIO (USD)
Arduino UNO R3	\$ 25.00
KeyEstudio XD-58C	\$ 14.00
Mini protoboard	\$ 5.00
Display LCD 16x2 + Módulo I2C	\$ 12.00
Cables Dupont Macho-Macho y Macho-Hembra (Surtido)	\$ 3.50
TOTAL	\$ 59.50

Tabla 3.1: Precio de componentes para sensor PPG.

3.3.2 CONSIDERACIONES

Existen algunos sensores de pulso cardíaco los cuales son de un costo inferior o de calidad de componentes genéricos, los cuales pueden ser tentadores de comprar. Sin embargo, la calidad de captura de la señal puede llegar a ser inferior o con muchísima interferencia, Un ejemplo de una señal limpia y una señal con ruido se muestra en las figuras 3.11 y 3.12.

El origen de la señal con ruido proviene de los circuitos que se utilizaron en el ensamblaje de los componentes. La calidad de los componentes es suficiente para su venta comercial, al punto de sacrificar hasta cierto punto la calidad de las lecturas que puede ofrecer el sensor. Es por ello que es crucial tener presente la calidad de

3.3. MATERIALES

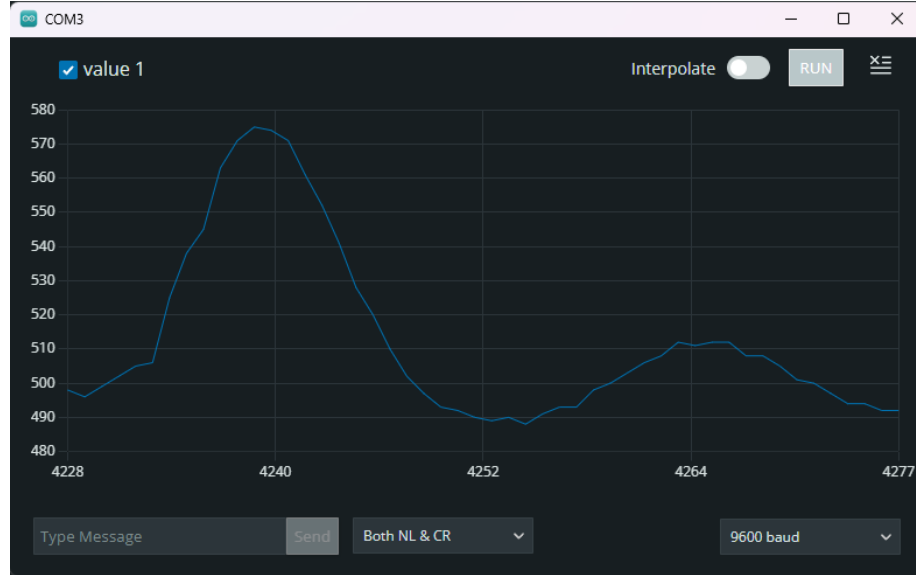


Figura 3.11: Lectura PPG Limpia con sensor KeyEstudio XD-58C. Elaboración propia.



Figura 3.12: Lectura PPG ruidosa con Sensor IDUINO genérico. Elaboración propia.

los mismos a la hora de comprar dichos componentes. Los sensores que originaron dichas lecturas están en la figura 3.13: a la izquierda el sensor KeyEstudio XD-58C y a la derecha un sensor IDUINO.



Figura 3.13: Sensores PPG con señales obtenidas. Elaboración propia.

3.4 CONSTRUCCIÓN DEL FOTOPLETISMÓGRAFO

Para la construcción del fotopletismógrafo con Arduino nos auxiliaremos de una mini protoboard para poder realizar las conexiones pertinentes.

En la protoboard los pines están etiquetados por números y letras del alfabeto y es similar a una notación de tipo tablero de ajedrez. A modo de ejemplo, a su vez, tiene unos pines que están etiquetados como '+' y '-' o con unas franjas rojas y azules en sus orillas. La transmisión de energía en la protoboard es de la siguiente manera: se transmite en las líneas Vivo y Tierra a lo largo de las orillas; dicho de otra manera, siguen la dirección de las letras. En los pines numerados se transmite de manera paralela a las líneas de energía y neutro; o sea, viaja a través de los números.

3.4.1 MODELADO EN PROTOBOARD

Para la conexión de los componentes utilizaremos los pines del Arduino y los del sensor realizando los siguientes pasos:

- Conectar a Arduino en los pines GND y 5 V 2 cables dupont macho-macho a la protoboard, estos se conectarán en los pines señalados como '+' y '-' en la protoboard.

3.4. CONSTRUCCIÓN DEL FOTOPLETISMÓGRAFO

- Conectar 4 cables dupont macho-hembra en la pantalla LCD 16x2 I2C del lado de la interfaz I2C.
- Conectar 3 cables dupont macho-macho en Arduino en los siguientes pines: (SCL, SDA y A0) y se conectarán a 3 líneas de la protoboard en los pines etiquetados.
- En el Display LCD I2C conectaremos los pines SDA, SCL, VCC y GND a 4 pines de la protoboard, a su vez, conectaremos los respectivos pines de Arduino a sus correspondientes en la Pantalla LCD I2C teniendo el cuidado de verificar las conexiones.
- Para el sensor XD-58C conectaremos los respectivos pines GND, VCC y Signal a nuestra protoboard, utilizando 3 pines extras. Utilizaremos el pin A0 de Arduino como Signal para el Sensor PPG. A su vez conectaremos el sensor PPG a 5 V y GND de la protoboard revisando correctamente las conexiones realizadas.
- Una vez está todo conectado, procedemos a conectar Arduino a la PC y procedemos a transferir el código fuente del proyecto a nuestra placa Arduino. Luego de eso abriremos el puerto Serial para verificar el correcto funcionamiento de este.
- Opcional: se puede conectar el pin 13 de Arduino a la protoboard, a fin de poder obtener la retroalimentación del sensor PPG a través de un LED conectado a la protoboard en lugar de usar el pin 13 de Arduino. Al conectar un LED es necesario colocar una resistencia de al menos 240 Ω para no dañar el LED.

La señalización del sensor está en la parte trasera del mismo. El prototipo debería quedar como en la figura 3.14.

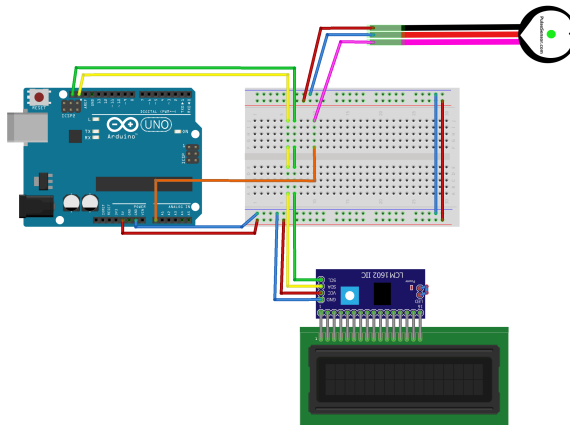


Figura 3.14: Arduino y sensor PPG. Elaboración propia.

3.4.2 MODELADO ESQUEMÁTICO

El modelo esquemático es una representación simplificada de los diferentes componentes que existen entre los circuitos electrónicos. Este diagrama es bastante útil para poder brindar las ideas base de los diferentes proyectos que se desarrollan tanto en la electrónica en general como en Arduino específicamente. El diagrama esquemático para el fotopletismógrafo se muestra en la figura 3.15.

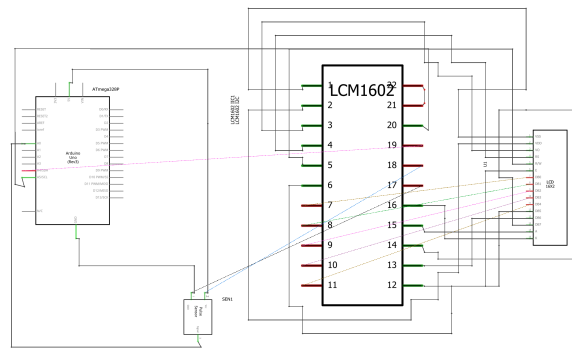


Figura 3.15: Diagrama esquemático Arduino y sensor PPG. Elaboración propia.

3.4.3 MODELADO PCB

Un diagrama PCB es un plano de un circuito electrónico que permite fijar todos nuestros componentes dentro de un circuito impreso, el cual permite a su vez la correcta conexión de todos nuestros componentes. El modelo PCB propuesto para el fotopletismógrafo se muestra en la figura 3.16.

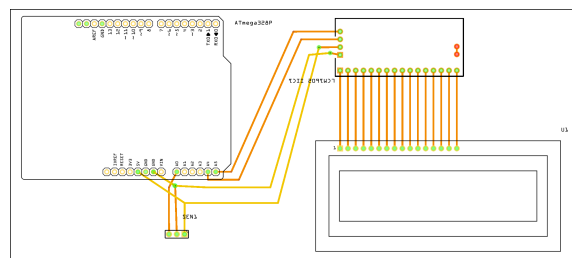


Figura 3.16: PCB Arduino y sensor PPG. Elaboración propia.

3.4. CONSTRUCCIÓN DEL FOTOPLETISMÓGRAFO

3.4.4 ALGORITMO SENSOR PPG

El algoritmo para el sensor PPG que construiremos contendrá el siguiente conjunto de instrucciones mostrados en el algoritmo 3.

Algoritmo 3 Arduino PPG

Input: Arduino Hardware connections

Output: PPG Read in loop

PulseWire \leftarrow 0

LED13 \leftarrow 13

Threshold \leftarrow 550

Signal \leftarrow 0

Creates pulseSensor instance

Initialize LCD Screen

Initialize Serial \leftarrow 9600 baud

set pulseSensor input \leftarrow PulseWire

set pulseSensor blink \leftarrow LED13

set pulseSensor threshold \leftarrow Threshold

Print on LCD "Heart Rate Monit"{End setup}

loop

myBPM \leftarrow pulseSensor.getBeatsPerMinute

if pulseSensor.sawStartOfBeat **then**

Print BPM Value on LCD Screen

end if

Signal \leftarrow read analog value from PulseWire pin

print Signal {Send the Signal value to Serial Plotter }

sleep for 20 milliseconds {delay until the next PPG read}

end loop{End loop}

Se define la variable Threshold, la cual es un valor de sensibilidad que nos permite conocer cuál es el valor hipotético de llenado capilar que se encuentra en nuestros vasos sanguíneos. Por defecto se deja el valor de 550 como un valor de referencia. Sin embargo, puede ser ajustado para que sea más sensible al cambio, en ese caso a no menos de 470, o reducir su sensibilidad (implica aumentar el valor de

Threshold a un máximo de 630). Cabe recordar que las lecturas que toma nuestro fotopleletismógrafo van en un rango de 0 a 1023 según la interfaz del pin analógico, pero en el caso del sensor PPG este estará oscilando entre 350 y 650.

3.4.5 LIBRERÍAS

Previo a la compilación de nuestro código en Arduino será necesaria la instalación de las siguientes librerías.

Librería	Versión
LiquidCrystal	>= 1.0.7
LiquidCrystal I2C	>= 1.1.2
PulseSensor Playground	>= 1.7.0

Tabla 3.2: Librerías para la elaboración del fotopleletismógrafo.

3.4.6 CÓDIGO FUENTE BASE

Al verificar las conexiones con la protoboard y los componentes, procedemos a conectar nuestro circuito a la PC por medio de un cable USB tipo B, abrimos Arduino IDE y copiamos y pegamos el siguiente código.

```

1 // Language: Arduino (C++)
2 #define USE_ARDUINO_INTERRUPTS true // Set-up low-level interrupts for
   most accurate BPM math.
3 #include <PulseSensorPlayground.h> // Includes the
   PulseSensorPlayground Library.
4
5 #include <Wire.h>
6 #include <LiquidCrystal_I2C.h>
7
8 #if defined(ARDUINO) && ARDUINO >= 100
9 #define printByte(args) write(args);
10 #else
11 #define printByte(args) print(args, BYTE);
12 #endif
13
14 // Heart character

```

3.4. CONSTRUCCIÓN DEL FOTOPLETISMÓGRAFO

```
15 uint8_t heart[8] = {0x0,0xa,0x1f,0x1f,0xe,0x4,0x0};
16
17 // Initialize a LiquidCrystal_I2C object:
18 LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
    16 chars and 2 line display
19
20 const int PulseWire = 0; // PulseSensor PURPLE WIRE connected to ANALOG
    PIN 0
21 const int LED13 = LED_BUILTIN; // The on-board Arduino LED, close to
    PIN 13.
22 int Threshold = 550; // Determine which Signal to "count as a beat" and
    which to ignore.
23 // Use the "Gettting Started Project" to fine-tune Threshold Value
    beyond default setting.
24 // Otherwise leave the default "550" value.
25
26 int Signal; // The signal will be sent into serial
27
28 PulseSensorPlayground pulseSensor; // Creates an instance of the
    PulseSensorPlayground object called "pulseSensor"
29
30 void setup() {
31     // put your setup code here, to run once:
32     lcd.init(); // initialize the lcd
33     lcd.backlight();
34     lcd.createChar(1, heart);
35     lcd.home();
36
37     // Initialize a serial monitor at 9600 baud
38     Serial.begin(9600);
39
40     pulseSensor.analogInput(PulseWire);
41     pulseSensor.blinkOnPulse(LED13);
42     pulseSensor.setThreshold(Threshold);
43
44     // Double-check the "pulseSensor" object was created and "began"
    seeing a signal.
45     if (pulseSensor.begin()) {
46         lcd.setCursor(0,0);
47         lcd.print("Heart Rate Monit");
```

```

48     }
49 }
50
51 void loop() {
52     // put your main code here, to run repeatedly:
53     int myBPM = pulseSensor.getBeatsPerMinute(); // Calls function on
54     our pulseSensor object that returns BPM as an "int".
55     // "myBPM" hold this BPM value now.
56     if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if "a
57     beat happened"
58     lcd.setCursor(0,1);
59     clearLCDLine(1);
60     lcd.setCursor(3,1);
61     lcd.printByte(1);
62     lcd.print(" BPM: "); // Print phrase "BPM: "
63     lcd.print(myBPM);
64     }
65
66     Signal = analogRead(PulseWire); // Read the PulseSensor's value.
67     // Assign this value to the "Signal
68     " variable.
69     Serial.println(Signal);
70
71     delay(20); // considered best practice in a simple sketch.
72 }
73
74 void clearLCDLine(int line)
75 {
76     lcd.setCursor(0,line);
77     for(int n = 0; n < 16; n++) // 20 indicates symbols in line. For 2
78     x16 LCD write - 16
79     {
80         lcd.print(" ");
81     }
82 }

```

Código 3.3: Arduino PPG Sketch.

3.5 VERIFICACIÓN DEL FUNCIONAMIENTO DEL FOTOPLETISMÓGRAFO

Una vez subido el código fuente a la placa Arduino, podremos ver que en la pantalla LCD saldrán las palabras «Heart Rate Monit» y cuando se coloque el sensor de pulso PPG al dedo anular veremos los BPM (beats per minute) tal como se muestra en la figura 3.17.

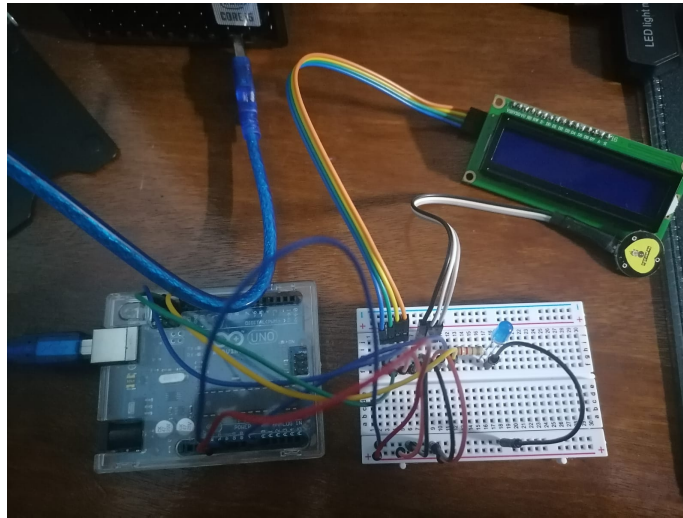


Figura 3.17: Demostración del funcionamiento del sensor Arduino PPG. Elaboración propia.

Al abrir el serial plotter en Arduino IDE → Tools → Serial Plotter veremos las lecturas que tomará el sensor PPG tal como se refleja en la figura 3.11.

Ahora que ya tenemos elaborado nuestro lector PPG con Arduino y al verificar que efectivamente se realizan las lecturas correctamente podremos realizar tomas de datos y con la consiguiente recopilación de señales podremos realizar aplicaciones de software que se explicarán en el siguiente capítulo.

4

Software

En este capítulo se describirán algunas aplicaciones de software que se pueden realizar con nuestro lector PPG. También se brindarán algunas sugerencias para poder extender más su potencial en el contexto de Arduino.

4.1 SOFTWARE

Las aplicaciones de software son programas de computador que tienen un propósito específico, como la consecución de tareas puntuales, ya sean procesadores de texto u hojas de calculo, productos de software o paquetes software *ad hoc*. Por ejemplo, Microsoft Word y Excel son clasificados como productos de software, mientras que un sistema informático específico para las actividades operativas de una empresa es clasificado como un software *ad hoc*.

En el caso de nuestro fotopletismógrafo podemos tomar las lecturas realizadas en crudo que son brindadas de salida en el monitor serial y proceder con su procesamiento posterior con cualquier lenguaje de programación que soporte comunicación por puerto serial. En la figura 4.1 se provee un diagrama de interacción entre Arduino y una aplicación que toma las lecturas del monitor serial del mismo en un ordenador.

4.1. SOFTWARE

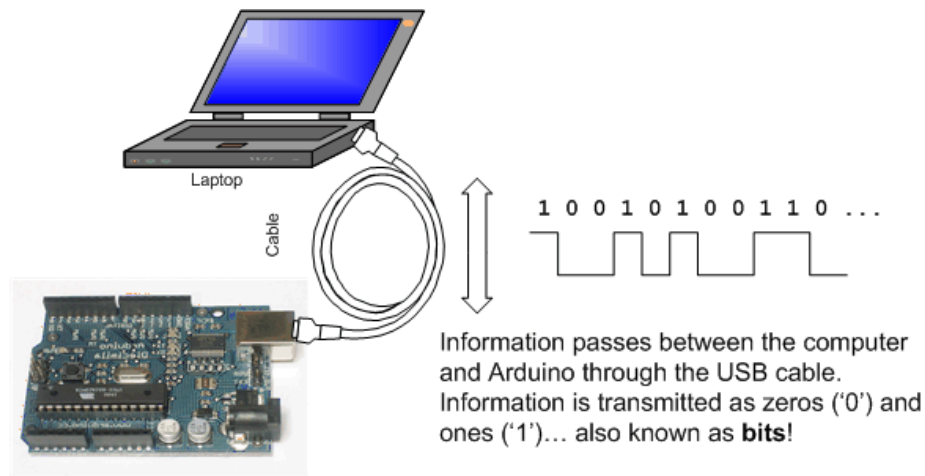


Figura 4.1: Comunicación entre Arduino y una PC por monitor serial, tomada de [38].

Cabe recordar que en el capítulo 3 se introdujo el funcionamiento del monitor serial y se comentaba que todos los valores que pueden ser leídos o procesados en hardware pueden ser enviados a través del monitor serial. Aprovecharemos esta característica de Arduino y lo que podremos realizar con nuestras aplicaciones de software es abrir el puerto serial con el que se instancia Arduino con objeto de adquirir la información o datos que este captura. En la tabla 4.1 se presentan las librerías que se encuentran disponibles en los diferentes lenguajes de programación para poder leer desde nuestro monitor serial.

Lenguaje de programación	Librerías serial
PHP	PHP Serial
Python	Pyserial
C#	Integrado en Librería System
Java	jSerialComm
Node.js	SerialPort

Tabla 4.1: Librerías de lectura de puerto serial para algunos lenguajes de programación.

En todos los lenguajes de programación tendremos que implementar un mecanismo para poder abrir el puerto serial disponible en Arduino, y, posteriormente,

realizar diferentes actividades que se necesiten, como un postprocesado de la señal o tomar el valor del monitor serial y mostrar los datos capturados al usuario.

4.1.1 OBTENCIÓN DE LECTURA A TRAVÉS DEL MONITOR SERIAL

Para poder capturar la señal PPG y mostrarla en aquellas aplicaciones de software que necesitemos, se debe utilizar un lenguaje para la adquisición de datos y comunicar la placa Arduino con la aplicación que deseamos conectar. Para ello debemos conocer el puerto habilitado en Arduino y posterior a ello se procede a tomar las lecturas obtenidas del sensor y estas serán mostradas por medio de nuestra aplicación.

Un ejemplo puede ser una aplicación web con conexión a Arduino. En este tipo de aplicaciones ha de tenerse presente que se debe tener conectado nuestro Arduino a la PC y este a su vez deberá ejecutar en paralelo la aplicación en Arduino y la aplicación web. Teniendo en cuenta ello, podemos crear una aplicación web que permita una conexión serial con Arduino para así obtener las lecturas del mismo por medio de WebSockets.

Los WebSockets son una tecnología avanzada que permite establecer una sesión de comunicación interactiva entre un cliente y un servidor. Por medio de este es posible el envío y recepción de mensajes controlados por eventos sin la necesidad de consultar a un servidor por la respuesta [39].

Dicho de otra manera, los WebSockets permiten crear aplicaciones interactivas que simulan un comportamiento en tiempo real por medio de la comunicación de eventos. Para esta aplicación se utilizará la ventaja del tiempo real para poder emitir en cada lectura de nuestro fotopletismógrafo un envío de datos a nuestra aplicación. En otras palabras, se puede utilizar WebSockets para enviar la información obtenida desde nuestro sensor PPG.

TOMA DE DATOS EN APLICACIÓN WEB POR MEDIO DE WEBSOCKETS

Antes de brindar la explicación de cómo tomar lecturas por medio de Arduino hacia nuestra aplicación web se trabajará bajo el supuesto de que se tiene una aplicación web de tipo MVC u orientada a microservicios y que existe una URL en donde se puede establecer una conexión hacia un WebSocket. En el ejemplo que

4.1. SOFTWARE

se mostrará en esta sección se utilizará un framework de Python llamado Django [40], utilizando una aplicación MVC. Además, en nuestra aplicación se instalará un paquete llamado Django Channels [41], que permite a nuestra aplicación utilizar WebSockets, o protocolos de chat, entre otros.

No se profundizará mucho en la creación del proyecto debido a que la finalidad de este documento es la construcción del fotopletismógrafo. Sin embargo, se brindarán los pasos relevantes para la comunicación entre Arduino y la aplicación web, en caso que se desee construir una aplicación y tomar lecturas de Arduino.

Lo primero que se realizará será configurar un proyecto en Django para que permita conexión a través de WebSockets. Para ello, en la documentación de Django Channels realizamos los pasos necesarios para crear una Aplicación WSGI. Después de ello, queda a discreción crear una aplicación en donde se trabajará específicamente la interfaz. Dentro de ella se crearán dos archivos nuevos: `consumers.py` y `routing.py`. En el archivo `consumers.py` colocaremos toda nuestra lógica del WebSocket que trabajaremos; el archivo `routing.py` recoge nuestro manejador de URL para cada función de socket que crearemos en el archivo `consumers.py`.

En el archivo `consumers.py` colocaremos lo reflejado más abajo. Nótese que nuestro WebSocket deriva de una clase llamada `AsyncWebsocketConsumer`; esto significa que nuestro WebSocket se comunicará de manera asíncrona, esto debido a que nuestro cliente espera una comunicación asíncrona entre el WebSocket y las lecturas.

```
1 # Language: Python (Django)
2 import json
3 from asyncio import sleep
4 import serial
5
6 from channels.generic.websocket import AsyncWebsocketConsumer
7
8
9 class GraphConsumer(AsyncWebsocketConsumer):
10     async def connect(self):
11         await self.accept()
12         # Recibimos por URL el puerto COM al cual consultaremos.
13         com = self.scope["url_route"]["kwargs"]["comport"]
14
```

```

15     # Establecemos una nueva comunicacion serial
16     ser = serial.Serial()
17     ser.baudrate = 9600
18     ser.port = com
19     ser.open()
20
21     for i in range(100):
22         # Lectura de datos desde el puerto serial
23         line = ser.readline()
24         string = line.decode()
25         stripped_string = string.strip()
26         if (stripped_string == ''):
27             stripped_string = 0
28
29         num_int = int(stripped_string)
30
31         # Enviamos nuestro valor a traves de un objeto JSON
32         await self.send(json.dumps({'value': num_int}))
33         # Tiempo de demora entre lecturas, deberia ser similar al
tiempo que le toma a Arduino una lectura de datos. 50/1000 da un
periodo de tiempo de demora.
34         await sleep(50/1000)
35
36     # Cerramos la comunicaci\on serial y la conexion con el
WebSocket
37     ser.close()
38
39     await self.close()

```

Código 4.1: Contenido de consumers.py.

En el archivo routing.py crearemos nuestra URL para poder enlazar el WebSocket por medio de una URL con la que nuestro cliente se conectará.

```

1 # Language: Python (Django)
2 from django.urls import re_path
3
4 from .consumers import GraphConsumer
5
6
7

```

4.1. SOFTWARE

```
8 ws_urlpatterns = [  
9     re_path(r"^ws/graph/(?P<comport>\w+)$", GraphConsumer.as_asgi())  
10 ]
```

Código 4.2: Contenido de routing.py.

En nuestra URL hemos definido una especie de parámetro llamado «comport» el cual es el puerto que se envía al WebSocket para poder abrir la conexión.

Una vez finalizada nuestra configuración del WebSocket, en nuestra página cliente bastará realizar una conexión con la URL creada en nuestro routing.py. La lógica que se realizó en la pantalla de ejemplo es la de mostrar por medio de un gráfico las lecturas del fotopletismógrafo; a su vez se guardan las muestras adquiridas y opcionalmente se copian las muestras obtenidas en un cuadro de texto en la interfaz.

La porción de código relevante para establecer la comunicación entre Arduino y la aplicación web se muestra en el código 4.3.

```
1 // Language: JavaScript  
2 let btn_show = document.getElementById("btn_show");  
3 btn_show.addEventListener('click', function (evt) {  
4     toastr.info('Start connection');  
5     listenSocket();  
6 });  
7  
8 // TODO Code  
9  
10 async function listenSocket(socket) {  
11  
12     var com = document.getElementById('comport').value;  
13     window.socket = new WebSocket('ws://localhost:8000/ws/graph/' + com  
14 );  
15  
16     window.socket.onmessage = function (e) {  
17         var djangoData = JSON.parse(e.data);  
18  
19         var newGraphData = graphData.datasets[0].data;  
20         //newGraphData.shift();  
21         newGraphData.push({ x: null, y: djangoData.value });  
22         graphData.datasets[0].data = newGraphData;
```

```

23     var newLabelData = graphData.labels;
24     //newLabelData.shift();
25     newLabelData.push('');
26     graphData.labels = newLabelData;
27
28     myChart.update();
29 }
30 }

```

Código 4.3: live graph script.

El resultado se puede ver en la figura 4.2.

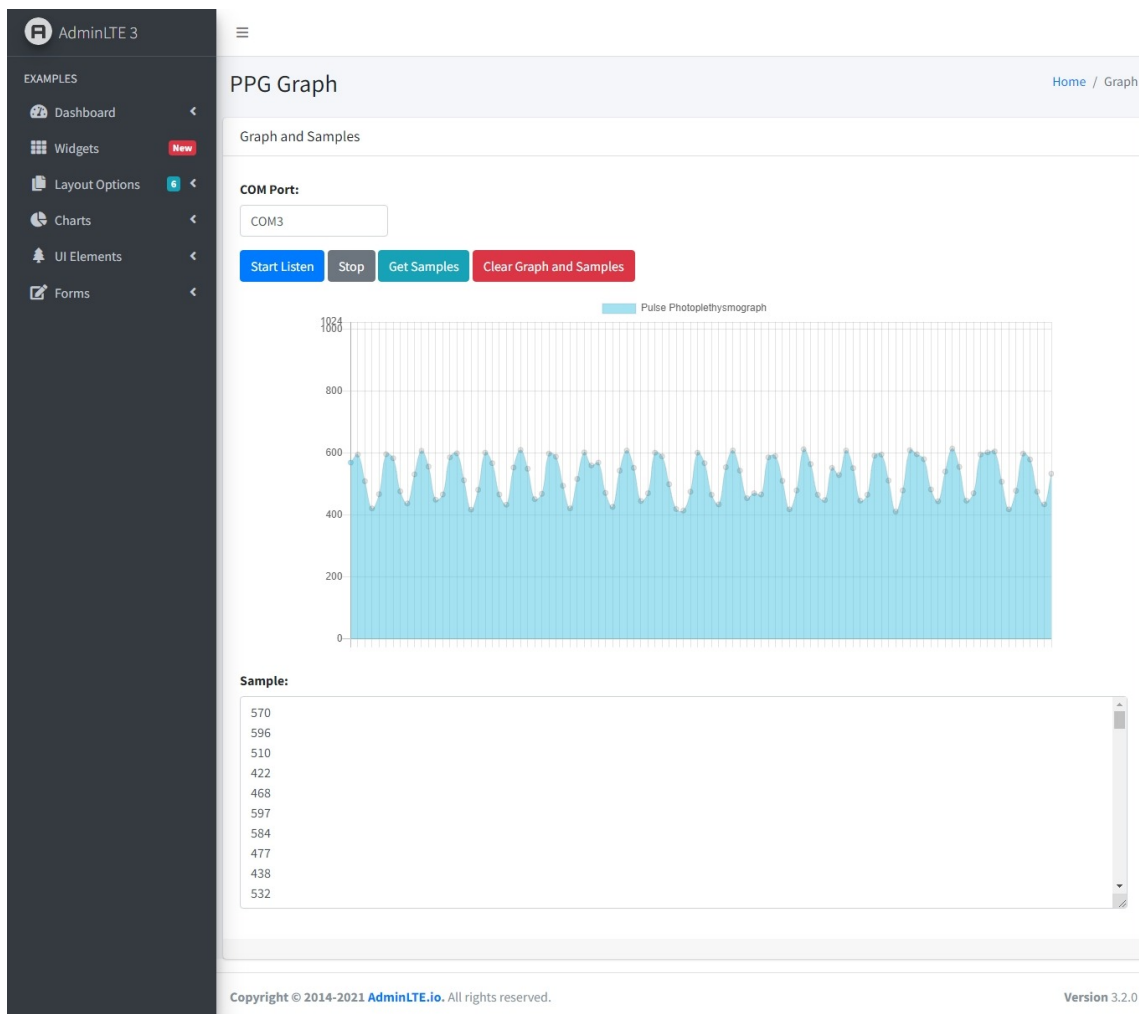


Figura 4.2: Aplicación web con Arduino utilizando WebSockets.

4.1. SOFTWARE

Si se desea conocer más en detalle el funcionamiento del proyecto, se deja un acceso al código fuente en la dirección siguiente: <https://gitlab.com/f4790/django-ppg>.

4.1.2 POSTPROCESADO DE LA SEÑAL PPG

El postprocesado de la señal es un tratamiento que se realiza sobre los datos, una vez han sido adquiridos. Puede haber diferentes mecanismos de postprocesado de datos, uno puede ser el «smoothing» o suavizado, un tratamiento de datos cuyo propósito es realizar un suavizado de distancias entre toma de datos, la cual permite realizar una especie de ajuste de tipo regresión, cuya finalidad es la de prevenir puntos de referencia bruscos mediante un ajuste lineal de valores. Un smoothing simple se puede realizar por medio de librerías de gráficos o con lenguajes de programación que soporten dicho procesamiento.

Los lenguajes que soportan de manera nativa un suavizado de curvas son los lenguajes de programación orientados a matemáticas y análisis numérico o estadístico. También se pueden utilizar lenguajes de programación de propósito general como Python utilizando librerías como matplotlib o numpy.

En el código 4.4 podemos observar un ejemplo de gráfico de datos obtenido por nuestra aplicación. En este ejemplo únicamente estamos mostrando los datos obtenidos en crudo.

```
1 # Language: Python
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Dataset
6 x = np.array(list(range(0, 100)))
7 y = np.array([
8     533, 525, 516, 509, 506, 507, 508, 509, 508, 507,
9     509, 509, 509, 509, 508, 509, 507, 505, 504, 504,
10    503, 503, 505, 506, 507, 508, 507, 507, 506, 507,
11    505, 505, 506, 508, 506, 504, 504, 506, 506, 505,
12    506, 505, 505, 506, 509, 517, 527, 534, 534, 527,
13    520, 513, 508, 506, 503, 510, 509, 506, 506, 506,
14    508, 509, 508, 507, 506, 506, 505, 504, 503, 504,
15    503, 504, 503, 504, 504, 505, 505, 505, 507, 507,
```

```

16     507, 506, 506, 506, 506, 505, 503, 505, 506, 506,
17     506, 507, 511, 521, 530, 534, 532, 524, 516, 509
18 ])
19
20 # Ploteando el grafico
21 plt.plot(x, y)
22 plt.title("Curve plotted using the given points")
23 plt.xlabel("X: Time")
24 plt.ylabel("Y: Signal")
25 plt.show()

```

Código 4.4: Gráfico de datos sin tratamiento.

El código anterior muestra el gráfico en la figura 4.3.

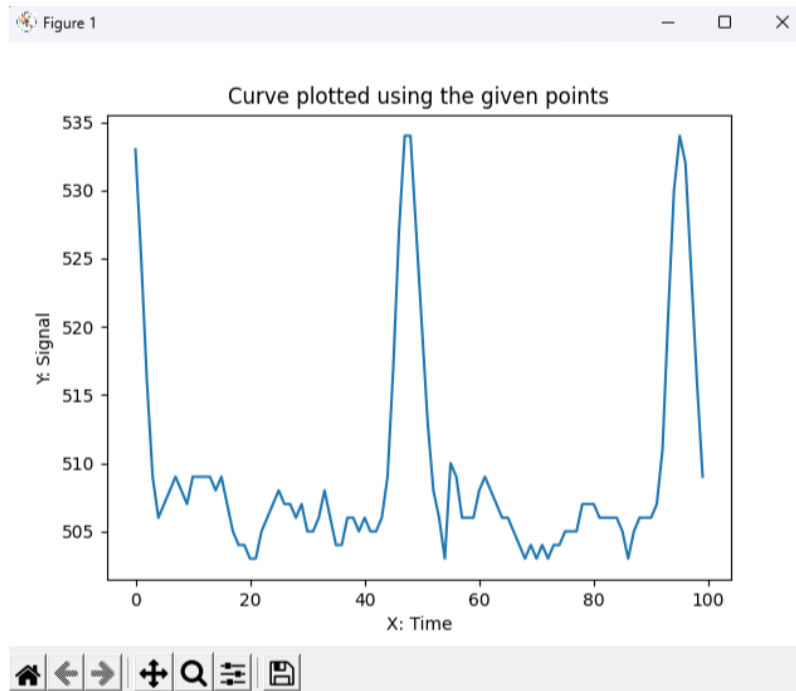


Figura 4.3: Gráfica con datos obtenidos.

Para el tratamiento de los datos podemos utilizar dos algoritmos simples que provee la librería *scipy*, los cuales son «Interpolación spline» e «Interpolación cúbica». Ambos métodos brindarán resultados deseables para el suavizado de nuestra muestra de datos.

4.1. SOFTWARE

El código 4.5 muestra el tratamiento por interpolación spline y su respectivo gráfico generado aparece en la figura 4.4.

```
1 # Language: Python
2 import numpy as np
3 import numpy as np
4 from scipy.interpolate import make_interp_spline
5 import matplotlib.pyplot as plt
6
7 # Dataset
8 x = np.array(list(range(0, 100)))
9 y = np.array([
10     533, 525, 516, 509, 506, 507, 508, 509, 508, 507,
11     509, 509, 509, 509, 508, 509, 507, 505, 504, 504,
12     503, 503, 505, 506, 507, 508, 507, 507, 506, 507,
13     505, 505, 506, 508, 506, 504, 504, 506, 506, 505,
14     506, 505, 505, 506, 509, 517, 527, 534, 534, 527,
15     520, 513, 508, 506, 503, 510, 509, 506, 506, 506,
16     508, 509, 508, 507, 506, 506, 505, 504, 503, 504,
17     503, 504, 503, 504, 504, 505, 505, 505, 507, 507,
18     507, 506, 506, 506, 506, 505, 503, 505, 506, 506,
19     506, 507, 511, 521, 530, 534, 532, 524, 516, 509
20 ])
21
22 X_Y_Spline = make_interp_spline(x, y)
23
24 # Retornar cada uno de los numeros espaciados
25 # sobre un intervalo especificado
26 X_ = np.linspace(x.min(), x.max(), 500)
27 Y_ = X_Y_Spline(X_)
28
29 # Ploteando el grafico
30 plt.plot(X_, Y_)
31 plt.title("Plot Smooth Curve Using the scipy.interpolate.
32           make_interp_spline() Class")
33 plt.xlabel("X: Time")
34 plt.ylabel("Y: Signal")
35 plt.show()
```

Código 4.5: Gráfico de datos con interpolación spline.

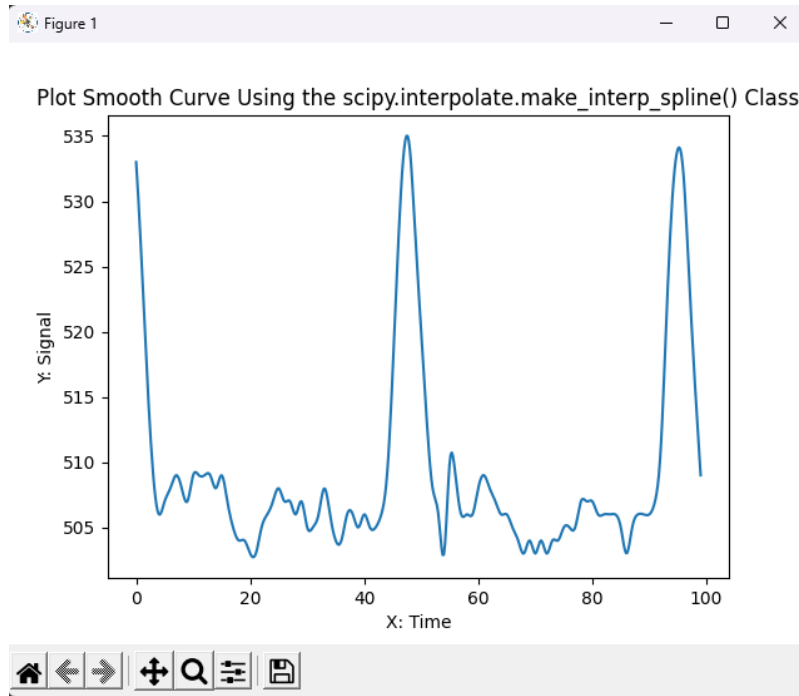


Figura 4.4: Gráfica con interpolación spline.

El código 4.6 muestra el tratamiento por Interpolación cúbica y su respectivo gráfico generado aparece en la figura 4.5.

```

1 # Language: Python
2 import numpy as np
3 from scipy.interpolate import interp1d
4 import matplotlib.pyplot as plt
5
6 # Dataset
7 x = np.array(list(range(0, 100)))
8 y = np.array([
9     533, 525, 516, 509, 506, 507, 508, 509, 508, 507,
10    509, 509, 509, 509, 508, 509, 507, 505, 504, 504,
11    503, 503, 505, 506, 507, 508, 507, 507, 506, 507,
12    505, 505, 506, 508, 506, 504, 504, 506, 506, 505,
13    506, 505, 505, 506, 509, 517, 527, 534, 534, 527,
14    520, 513, 508, 506, 503, 510, 509, 506, 506, 506,
15    508, 509, 508, 507, 506, 506, 505, 504, 503, 504,
16    503, 504, 503, 504, 504, 505, 505, 505, 507, 507,

```

4.1. SOFTWARE

```
17     507, 506, 506, 506, 506, 505, 503, 505, 506, 506,  
18     506, 507, 511, 521, 530, 534, 532, 524, 516, 509  
19 ]) )  
20 cubic_interpolation_model = interp1d(x, y, kind="cubic")  
21  
22 # Ploteando el grafico  
23 X_ = np.linspace(x.min(), x.max(), 500)  
24 Y_ = cubic_interpolation_model(X_)  
25  
26 plt.plot(X_, Y_)  
27 plt.title("Plot Smooth Curve Using the scipy.interpolate.interp1d Class  
28 ")  
29 plt.xlabel("X: Time")  
30 plt.ylabel("Y: Signal")  
31 plt.show()
```

Código 4.6: Gráfico de datos con interpolación cúbica.

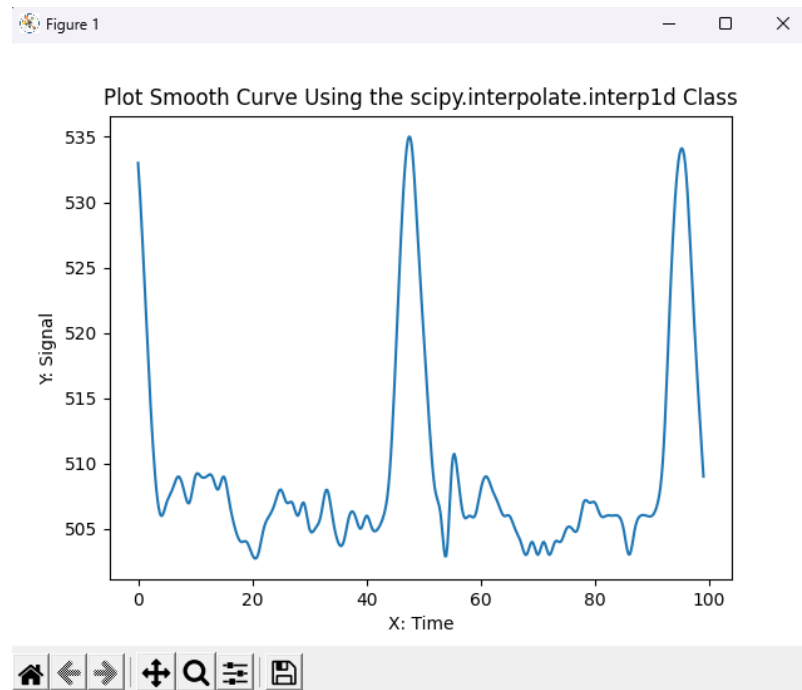


Figura 4.5: Gráfica con interpolación cúbica.

4.1.3 INVERSIÓN DE LA SEÑAL PPG

Algunos fabricantes pueden brindar sensores PPG que miden la difusión de la luz en un punto de medición arterial, lo cual es una manera de obtener una señal cardíaca del cuerpo humano. Sin embargo, la medición se basa en la difusión de luz en lugar de la absorción de luz. Bajo este tipo de casos decimos que podemos encontrarnos con una señal PPG normal. No obstante, para otros estudios puede resultar de utilidad estudiar la señal PPG invertida.

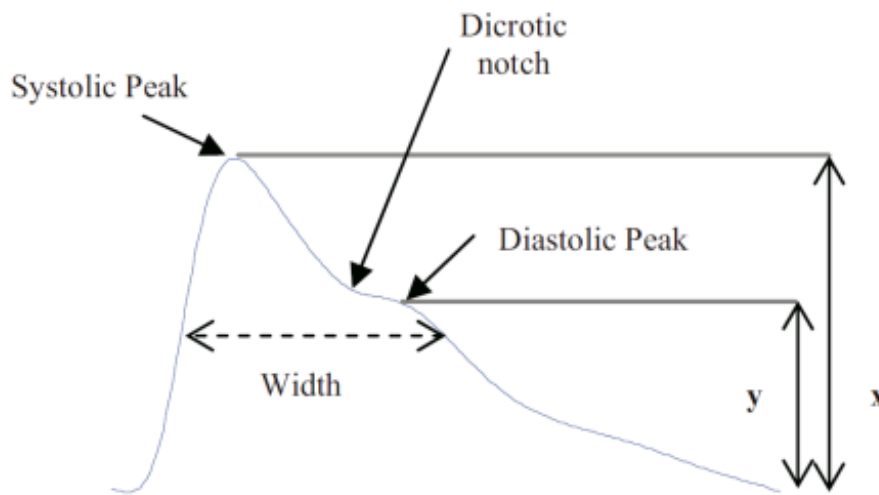


Figura 4.6: Representación de una señal PPG normal, tomada de [42].

La señal PPG invertida es una variación de la señal PPG obtenida, en la que en lugar de medir la difusión de la luz generada por el haz de luz se lee la tasa de absorción de luz que detecta el sensor. En un sentido práctico podemos decir que es una «inversión» de la señal PPG normal.

Para obtener la señal PPG invertida podemos realizarlo a través de hardware directamente con Arduino; en este documento se presentan tres maneras de cómo invertir dicha señal obtenida.

INVERTIR SEÑAL PPG POR MEDIO DE MAP

En Arduino existe una función llamada `Map()` [43] cuya sintaxis es la siguiente.

4.1. SOFTWARE

```
1 // Language: Arduino (C++)
2 // map function
3 value = map(value, fromLow, fromHigh, toLow, toHigh);
```

Código 4.7: Arduino map function.

Parámetros:

- Value: el número a mapear.
- fromLow: el límite inferior del rango actual del valor.
- fromHigh: el límite superior del rango actual del valor.
- toLow: el límite inferior del rango objetivo del valor.
- toHigh: el límite superior del rango objetivo del valor.

Definición:

La función map vuelve a asignar un número de un rango a otro, es decir, un valor de fromLow se asignaría a un toLow, un valor de fromHigh a un toHigh, y los valores intermedios a valores intermedios. No restringe los valores dentro del rango.

La función map utiliza matemáticas enteras, por lo que no generará fracciones, cuando las matemáticas podrían indicar que debería hacerlo. Los restos fraccionarios se truncan y no se redondean ni promedian.

Básicamente, la función map nos permite redefinir el valor que obtenemos para que este pueda ser configurado según nuestras necesidades. En nuestro caso, en el que queremos invertir la señal PPG nos bastaría con poner al revés los valores que deseamos obtener, recordemos que en Arduino las lecturas analógicas van de 0 a 1023; por tanto, bastará con reescribir la función map de la siguiente manera para que podamos voltear la señal recibida.

```
1 // Language: Arduino (C++)
2 // signal inverted
3 signal = map(signal, 0, 1023, 1023, 0);
```

Código 4.8: Inversión de la señal PPG obtenida.

En el siguiente Sketch podemos ver en acción la función map para poder invertir dicha señal.

```

1 // Language: Arduino (C++)
2 // Variables
3 int PulseSensorPurplePin = 0;
4 int LED = LED_BUILTIN; // The on-board Arduino LED
5
6 int Signal;
7 int Threshold = 550;
8
9
10 // The SetUp Function:
11 void setup() {
12     pinMode(LED,OUTPUT);
13     Serial.begin(9600);
14 }
15
16 // The Main Loop Function
17 void loop() {
18     Signal = analogRead(PulseSensorPurplePin);
19
20     // map value to invert signal.
21     Signal = map(Signal, 0, 1023, 1023, 0);
22
23     Serial.println(Signal);
24
25     // The Threshold will be affected due to the map values.
26     if(Signal > Threshold){
27         digitalWrite(LED,HIGH);
28     } else {
29         digitalWrite(LED,LOW);
30     }
31     delay(10);
32 }

```

Código 4.9: Inversión de la señal PPG por medio de la función map.

4.1. SOFTWARE

El resultado en el Monitor serial es el siguiente.

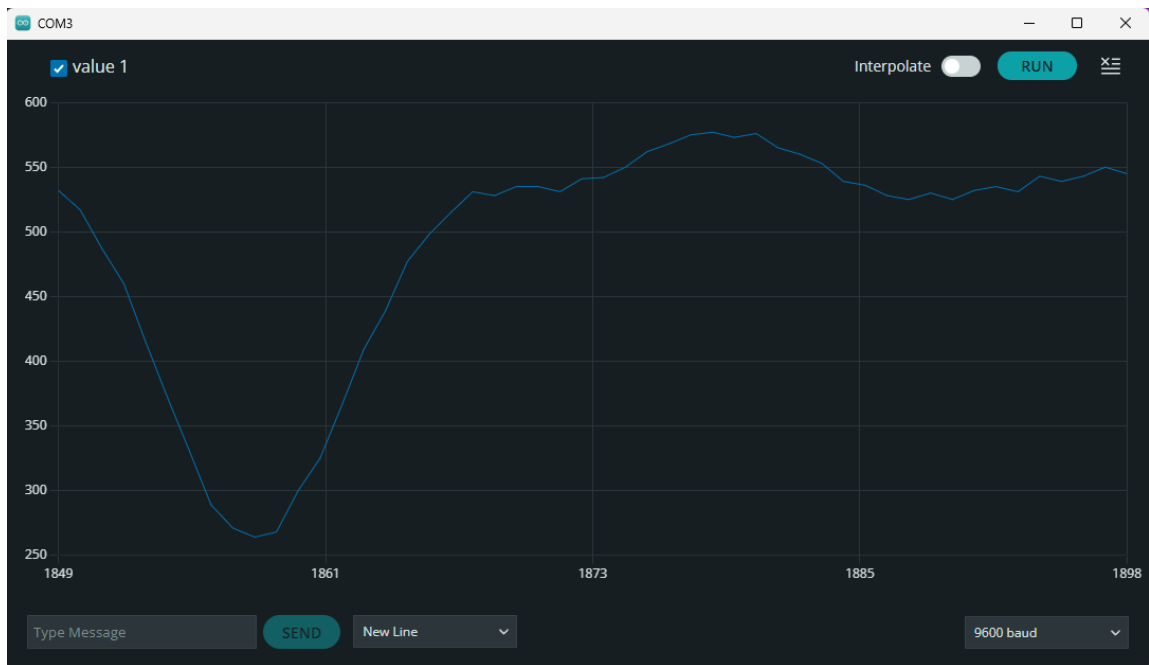


Figura 4.7: Señal PPG invertida obtenida en el monitor serial con la función map.

INVERTIR LA SEÑAL PPG POR MEDIO DEL DIFERENCIAL CON VALOR MÁXIMO OBTENIDO.

La siguiente forma de obtener la señal PPG invertida es por medio de una diferencia de la lectura obtenida con el valor máximo medido con Arduino. Esta forma es un poco más burda en cuanto al procedimiento. Sin embargo, para fines demostrativos es práctica de utilizar.

Lo que se realizará en el Sketch es inicializar una nueva variable llamada «max» e igualarla a 0. Posteriormente, se hace una verificación del valor máximo obtenido, para poder reemplazar el valor máximo en tiempo de ejecución.

El Sketch sugerido quedaría de la siguiente manera.

```
1 // Language: Arduino (C++)
2 // Variables
3 int PulseSensorPurplePin = 0;
4 int LED = LED_BUILTIN; // The on-board Arduino LED
5
6 int Signal;
```

```

7 int Threshold = 550;
8 int max = 0;
9
10
11 // The SetUp Function:
12 void setup() {
13     pinMode(LED,OUTPUT);
14     Serial.begin(9600);
15 }
16
17 // The Main Loop Function
18 void loop() {
19     Signal = analogRead(PulseSensorPurplePin);
20
21     // verify is signal is greather than max, then substitute max value
22     .
23     if(Signal > max){
24         max = Signal;
25     }
26
27     Signal = max - Signal;
28
29     Serial.println(Signal);
30
31     // The Threshold will be affected due to the signal changes.
32     if(Signal > Threshold){
33         digitalWrite(LED,HIGH);
34     } else {
35         digitalWrite(LED,LOW);
36     }
37     delay(10);
38 }

```

Código 4.10: Inversión de la señal PPG obtenida por medio de valores máximos.

4.1. SOFTWARE

Las muestras obtenidas por medio del monitor serial son las siguientes.

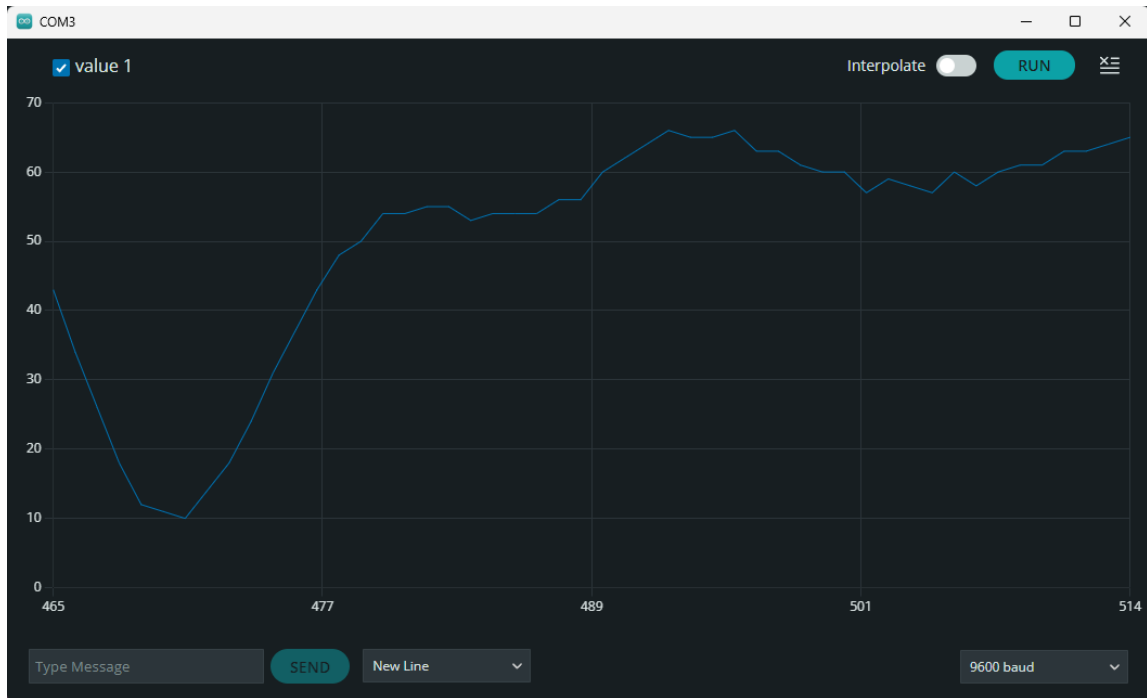


Figura 4.8: Señal PPG invertida obtenida en el monitor serial por medio del diferencial.

INVERTIR LA SEÑAL PPG POR MEDIO DEL COMPLEMENTO DE LA LECTURA OBTENIDA.

En este método se hace una simplificación del valor máximo obtenido; en lugar de ir utilizando un valor máximo e ir actualizando dicho valor en tiempo de ejecución de Arduino, podemos asumir que el valor máximo obtenido es de 1024 y a este valor restar el valor de Signal obtenido, ya que es conocido que el valor máximo que se puede obtener analógicamente en Arduino es 1024.

El Sketch quedaría de la siguiente manera.

```
1 // Language: Arduino (C++)
2 // Variables
3 int PulseSensorPurplePin = 0;
4 int LED = LED_BUILTIN; // The on-board Arduino LED
5
6 int Signal;
```



```
7 int Threshold = 550;
8 int max = 1024;
9
10
11 // The SetUp Function:
12 void setup() {
13     pinMode(LED,OUTPUT);
14     Serial.begin(9600);
15 }
16
17 // The Main Loop Function
18 void loop() {
19     Signal = analogRead(PulseSensorPurplePin);
20
21     Signal = max - Signal;
22
23     Serial.println(Signal);
24
25     // The Threshold will be affected due to the signal changes.
26     if(Signal > Threshold){
27         digitalWrite(LED,HIGH);
28     } else {
29         digitalWrite(LED,LOW);
30     }
31     delay(10);
32 }
```

Código 4.11: Inversión de la señal PPG obtenida por medio del complemento respecto del valor máximo.

4.1. SOFTWARE

El resultado en Monitor Serial sería el siguiente.



Figura 4.9: Señal PPG invertida obtenida en el monitor serial por medio del complemento.

4.1.4 OPORTUNIDADES DE MEJORA

En cuanto a las oportunidades de mejora se puede utilizar un módulo de Wi-Fi compatible con Arduino, con la finalidad de prescindir por completo de una conexión por cable entre nuestras aplicaciones y Arduino, Un modelo de ejemplo puede ser el siguiente: Placa de desarrollo de Internet en ESP8266 ESP-12F para Arduino. Cabe mencionar que utilizando comunicación por Wi-Fi implicaría realizar modificaciones a nuestro proyecto para que en lugar de mandar por WebSockets los valores del puerto Serial, se pueda enviar directamente el valor capturado en Arduino hacia una dirección URL dentro de la red local en la que ambos dispositivos estén conectados.

5

Costos y presupuesto

En este capítulo se brinda una estimación del costo de elaboración del proyecto, además de las actividades involucradas en la construcción del mismo.

5.1 METODOLOGÍA DE DESARROLLO

El desarrollo del proyecto se elaboró bajo la metodología de desarrollo denominada Prototyping o Prototipado [44].

Un prototipo es una versión preliminar de un sistema informático con fines de demostración o evaluación.

El prototipo genera una implementación parcial de un sistema, con fines específicos de aprender sobre los requisitos de un sistema. Este es construido de una manera tan rápida como sea posible.

Esto es debido a que los usuarios, clientes o stakeholders están interesados en realizar un experimento y una manera de brindar ese primer acercamiento es con un prototipo. Luego estos usuarios pueden brindar retroalimentación sobre lo que a ellos les gustó y no les gustó acerca del prototipo.

Un prototipo puede ser utilizado como parte de la fase de determinación de requisitos o justo antes de la parte de desarrollo. En otros casos un prototipo puede servir antes de alguno o todo el proceso en el desarrollo incremental o evolutivo.

5.1. METODOLOGÍA DE DESARROLLO

Ventajas

- Flexibilidad de desarrollo.
- Permite experimentar con una arquitectura o desarrollo de sistema.
- Rapidez de desarrollo.
- Se puede utilizar el prototipo como producto final.

Desventajas

- Puede dar una falsa ilusión al usuario sobre la velocidad de desarrollo.
- No se conoce cuándo se tendrá un producto aceptable.
- No se conoce cuántas iteraciones serán necesarias.
- Se puede volver producto aun cuando no cumpla con los estándares de desarrollo.

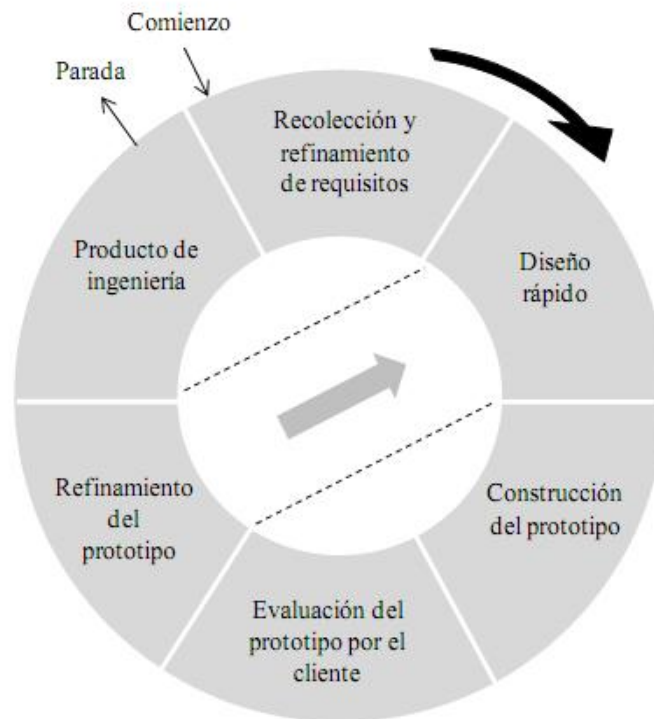


Figura 5.1: Diagrama de metodología de prototipo, tomado de [45].

5.1.1 ETAPAS

- Recolección y refinamiento de requisitos.
- Modelado, diseño rápido.
- Construcción del prototipo.
- Desarrollo, evaluación del prototipo por el cliente.
- Refinamiento del prototipo.
- Producto de ingeniería.

5.2 ESTIMACIÓN DE COSTOS

Para el costo de elaboración del fotopleletismógrafo se incurrió en los siguientes costos.

5.2.1 MATERIALES

El costo de materiales es el siguiente.

COMPONENTES	PRECIO (USD)
Arduino UNO R3	\$ 25.00
KeyEstudio XD-58C	\$ 14.00
Mini protoboard	\$ 5.00
Display LCD 16x2 + Módulo I2C	\$ 12.00
Cables Dupont Macho-Macho y Macho-Hembra (Surtido)	\$ 3.50
TOTAL	\$ 59.50

Tabla 5.1: Precio de componentes, tomados de la tabla 3.1, para el sensor PPG durante el año 2022.

Hay que tener en cuenta que es necesario buscar una tienda en donde vendan productos específicos para Arduino y robótica orientada a estudiantes y aficionados de la electrónica. Hay diferentes tiendas en el área metropolitana de San Salvador,

5.3. PLANIFICACIÓN DE ACTIVIDADES

como son Steren, Casa Rivas y Electrónica 2001. Para este proyecto los materiales fueron adquiridos en Electrónica 2001.

5.2.2 RECURSOS

Los recursos son insumos que se han utilizado para la creación del proyecto, no forman parte directa de los materiales, sino que forman parte del proceso para obtener como resultado el dispositivo a construir:

Recurso	Descripción
Cautín	Utilizado para soldar módulo I2C del display LCD
Estaño	
Pasta de soldar	

Tabla 5.2: Recursos utilizados para la creación del proyecto.

5.3 PLANIFICACIÓN DE ACTIVIDADES

En esta sección, se detalla la planificación de actividades para llevar a cabo el trabajo de graduación. Para una organización efectiva, se emplean tres elementos clave: el cronograma, el resumen de actividades y el diagrama de Gantt.

La combinación de estas tres herramientas de planificación proporciona un enfoque estructurado y eficiente para el desarrollo del trabajo de graduación. A través del cronograma, el resumen de actividades y el diagrama de Gantt, se demuestra una organización efectiva y un plan de acción sólido.

5.3.1 CRONOGRAMA DE ACTIVIDADES

El cronograma es una herramienta esencial para establecer fechas límite y asignar tiempos estimados a cada actividad. A través de él, se visualiza el orden y la duración de las tareas del proyecto, permitiendo un seguimiento y control adecuados del avance.

Nombre	Fecha inicio	Fecha fin	Duración (días)
Proyecto Memoria PPG	20/02/2023	21/04/2023	45
Obtención de materiales	20/02/2023	24/02/2023	5
Análisis de requisitos	27/02/2023	03/03/2023	5
Construcción	06/03/2023	24/03/2023	15
Prototipo de software	27/03/2023	14/04/2023	15
Prototipo de scripts de procesamiento	17/04/2023	21/04/2023	5
Pruebas	24/04/2023	12/05/2023	15
Pruebas de lectura	24/04/2023	28/04/2023	5
Detección de ruido	01/05/2023	05/05/2023	5
Ajuste de señal	08/05/2023	12/05/2023	5
Redacción de Memoria	15/05/2023	01/09/2023	80
Capítulo 1	15/05/2023	09/06/2023	20
Capítulo 2	12/06/2023	14/07/2023	25
Capítulo 3	17/07/2023	04/08/2023	15
Capítulo 4	07/08/2023	18/08/2023	10
Capítulo 5	21/08/2023	01/09/2023	10

Tabla 5.3: Cronograma de actividades para el fotopletismógrafo.

5.3.2 RESUMEN DE ACTIVIDADES

El resumen de actividades proporciona una visión general de las principales tareas que se llevarán a cabo en el trabajo de graduación. Enumera de manera precisa y concisa las actividades clave necesarias para alcanzar los objetivos propuestos. Esto ayuda a tener una comprensión clara de los pasos a seguir en el proyecto.

Nombre	Fecha inicio	Fecha fin	Duración (días)
Construcción del fotopletismógrafo	20/02/2023	21/04/2023	45
Presentación Defensa 1	17/04/2023	21/04/2023	5
Documento de Investigación	15/05/2023	01/09/2023	80
Presentación Defensa 2	12/06/2023	16/06/2023	5
Presentación Defensa Final	28/08/2023	01/09/2023	5
TOTAL	20/02/2023	01/09/2023	140

Tabla 5.4: Resumen de actividades.

5.3. PLANIFICACIÓN DE ACTIVIDADES

5.3.3 DIAGRAMA DE GANTT

El diagrama de Gantt es una herramienta visual que muestra las actividades del proyecto en un calendario. A través de barras, se representan las tareas y su duración estimada. Este diagrama permite identificar las interdependencias entre las actividades, los solapamientos y las fechas críticas. Así se optimiza la gestión del tiempo y los recursos, evitando retrasos y facilitando la planificación.

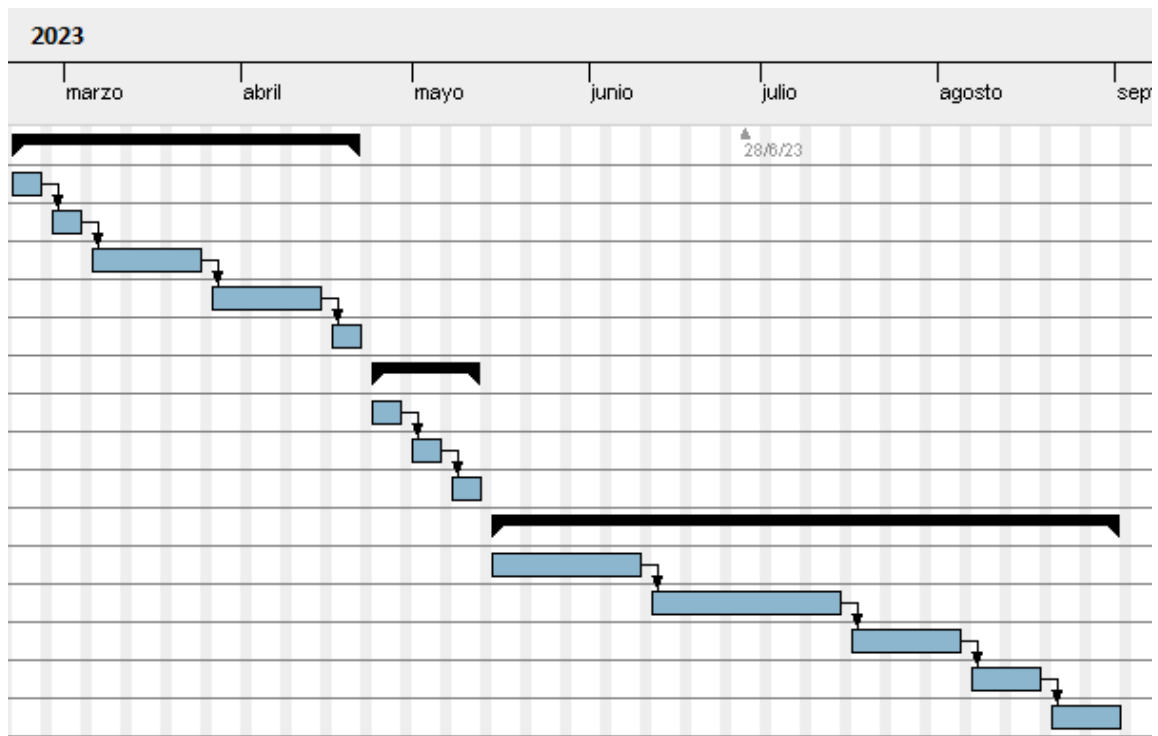


Figura 5.2: Diagrama de Gantt para el desarrollo del fotopletismógrafo.



Conclusiones y líneas futuras

El diseño y desarrollo del sensor PPG por medio de la plataforma Arduino, mediante la metodología de prototipo, nos brinda los siguientes resultados:

- La implementación del dispositivo PPG ha resultado exitosa en cuanto a la obtención de valores de referencia para la señal PPG a obtener.
- La construcción del sensor PPG es viable por medio de la plataforma Arduino, siempre y cuando los sensores PPG sean adecuados y de buena calidad para obtener una señal PPG limpia.
- Se brinda toda la documentación técnica en materia de construcción, diagramas esquemáticos, y propuestas de software para poder utilizar el dispositivo PPG.
- Se presentan propuestas de software para poder estudiar la señal obtenida y obtener tanto su representación como los valores de referencia de la misma.
- Se brindan sugerencias y oportunidades de mejora en cuanto a las maneras de obtener la señal PPG.
- La metodología de prototipo nos permite diseñar software o hardware piloto que puede ser posteriormente implementado a gran escala para todos los usuarios e interesados en el producto.
- La señal PPG muestra patrones que pueden ser de interés para diagnósticos clínicos, fisiológicos e incluso psicológicos. Tan pronto se estudien los diferentes patrones de comportamiento asociados con enfermedades crónicas es factible atender una nueva manera de detectar anomalías fisiológicas.

Referencias

- [1] PhD Monty Escabí. «Introduction to Biomedical Engineering (Third Edition)». En: *Introduction to Biomedical Engineering*. Ed. por Elsevier Inc. Elsevier, 1 de ene. de 2012, págs. 667-746. DOI: <https://doi.org/10.1016/B978-0-12-374979-6.00011-3>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123749796000113>.
- [2] MedLinePlus. *Rayos X*. 2021. URL: <https://medlineplus.gov/spanish/xrays.html>.
- [3] MedLinePlus. *Electrocardiograma*. 2020. URL: <https://medlineplus.gov/spanish/pruebas-de-laboratorio/electrocardiograma/>.
- [4] ResearchGate, ed. *The ECG signal, the PPG signal and the second derivative photoplethysmographic (SDPPG)*. 1 de ene. de 2019. URL: https://www.researchgate.net/figure/The-ECG-signal-the-PPG-signal-and-the-second-derivative-photoplethysmographic-SDPPG_fig1_330258585.
- [5] Dr Sonzini Astudillo. *Tipos de señales biológicas*. 2012. URL: <https://guiasdeneuro.com/tipos-de-senales-biologicas/>.
- [6] MD Larissa Hirsch. *El corazón y el sistema circulatorio*. Ed. por Connecticut Children's. 3 de sep. de 2018. URL: <https://www.connecticutchildrens.org/health-library/es/parents/heart-esp/>.
- [7] Jesús María Rodríguez Presedo. *Adquisición de señales biológicas*. Ed. por Universidad de Santiago. URL: <https://www.usc.gal/catedras/telemedicina/2005/materialAsignatura/AdquisicionSenalesBiologicas.pdf>.

REFERENCIAS

- [8] Pedro Luis Véliz Martínez y Ana Rosa Jorna Calixto. «Evolución histórica y perspectivas futuras de la Medicina Intensiva y Emergencia como especialidad en Cuba». es. En: *Educación Médica Superior* 28 (sep. de 2014), págs. 592-602. ISSN: 0864-2141. URL: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412014000300018&nrm=iso.
- [9] Karthik Budidha y Panayiotis A. Kyriacou. «Photoplethysmography for Quantitative Assessment of Sympathetic Nerve Activity (SNA) During Cold Stress». En: *Frontiers in Physiology* 9 (ene. de 2019). DOI: 10.3389/fphys.2018.01863.
- [10] Albert Saavedra. «Diseño y desarrollo de sensores y biosensores para monitoreo de procesos biohidrometalúrgicos». es. En: (2018). DOI: 10.13140/RG.2.2.12917.63205.
- [11] *Diferentes instrumentos médicos*. 1 de mar. de 2018. URL: <https://es.vecteezy.com/arte-vectorial/372743-diferentes-instrumentos-medicos>.
- [12] Cristina Pérez Barreiro. *Tecnología electrónica en ingeniería biomédica: sensores*. Ed. por Universidad de Valladolid. 1 de sep. de 2020. URL: https://www.researchgate.net/figure/Figura-19-Eschema-de-un-biosensor-Figura-reproducida-de-Vargas-Bernal-et-al-2012_fig6_329983275.
- [13] Joel Gutierrez. *Electrocardiograma: ¿Qué es un electrocardiograma?* Ed. por Tu Significado. URL: <https://tusignificado.com/salud/electrocardiograma>.
- [14] Ecocardio, ed. *Complejo QRS*. 1 de ene. de 2021. URL: <https://ecocardio.com/documentos/biblioteca-preguntas-basicas/preguntas-al-cardiologo/1046-complejo-qrs.html>.
- [15] Stephen Evanczuk. *Utilice un módulo de biosensor portátil combo PPG/ECG para mediciones cardiovasculares certificables por la FDA*. Ed. por Colaboración de Editores de Digi-Key de América del Norte. 19 de mar. de 2019. URL: <https://www.digikey.com/es/articles/use-a-ppg-ecg-combo-wearable-biosensor-module>.
- [16] Gabriel Cano García. *Diseño de un sistema óptico de detección de la tensión emocional*. 31 de dic. de 2018.

- [17] MedLinePlus. *Pletismografía*. 7 de mar. de 2020. URL: <https://medlineplus.gov/spanish/ency/article/003771.htm>.
- [18] Mayo Clinic. *Espirometría*. 24 de feb. de 2018. URL: <https://www.mayoclinic.org/es-es/tests-procedures/spirometry/about/pac-20385201>.
- [19] M. Yapur G. Celi M. Rocha. *Mediciones Fotopletismográficas*. Ed. por Research Gate. 1 de sep. de 2011. URL: https://www.researchgate.net/publication/277801575_Mediciones_fotopletismograficas.
- [20] quimica.es, ed. *Ley de Beer-Lambert*. URL: https://www.quimica.es/enciclopedia/Ley_de_Beer-Lambert.html.
- [21] Rebecca Dezube. *Intercambio de oxígeno y dióxido de carbono*. Ed. por Manual MSD. URL: [https://www.msmanuals.com/es/hogar/trastornos-del-pulm%C3%B3n-y-las-v%C3%ADas-respiratorias/biolog%C3%ADa-de-los-pulmones-y-de-las-v%C3%ADas-respiratorias/intercambio-de-ox%C3%ADgeno-y-di%C3%B3xido-de-carbono#:~:text=La%20sangre%20oxigenada%20circula%20desde,obtenga%20m%C3%A1s%20informaci%C3%B3n%20\)..](https://www.msmanuals.com/es/hogar/trastornos-del-pulm%C3%B3n-y-las-v%C3%ADas-respiratorias/biolog%C3%ADa-de-los-pulmones-y-de-las-v%C3%ADas-respiratorias/intercambio-de-ox%C3%ADgeno-y-di%C3%B3xido-de-carbono#:~:text=La%20sangre%20oxigenada%20circula%20desde,obtenga%20m%C3%A1s%20informaci%C3%B3n%20)..)
- [22] Manual MSD, ed. *Curva de disociación de la oxihemoglobina*. URL: <https://www.msmanuals.com/es/professional/multimedia/figure/curva-de-disociaci%C3%B3n-de-la-oxihemoglobina>.
- [23] Alejandro Calderon Gamio. *Curva de absorción de luz para hemoglobina oxigenada y desoxigenada*. Ed. por ResearchGate. URL: https://www.researchgate.net/figure/Ondas-de-senal-que-muestran-la-absorcion-de-luz-para-la-hemoglobina-oxigenada-y_fig4_331742636.
- [24] John Allen. «Photoplethysmography and its application in clinical physiological measurement». En: *Physiological Measurement* 28.3 (feb. de 2007), R1-R39. DOI: 10.1088/0967-3334/28/3/r01.
- [25] *Photoplethysmogram (PPG) pulse wave*. 27 de abr. de 2021. URL: [https://commons.wikimedia.org/wiki/File:Photoplethysmogram_\(PPG\)_pulse_wave.svg](https://commons.wikimedia.org/wiki/File:Photoplethysmogram_(PPG)_pulse_wave.svg).

REFERENCIAS

- [26] Elisa Mejía-Mejía et al. «Photoplethysmography signal processing and synthesis». En: *Photoplethysmography*. Elsevier, 2022, págs. 69-146. DOI: 10.1016/b978-0-12-823374-0.00015-3.
- [27] *Classes of photoplethysmogram (PPG) pulse wave shape.svg*. 4 de mayo de 2021. URL: [https://commons.wikimedia.org/wiki/File:Classes_of_photoplethysmogram_\(PPG\)_pulse_wave_shape.svg](https://commons.wikimedia.org/wiki/File:Classes_of_photoplethysmogram_(PPG)_pulse_wave_shape.svg).
- [28] Richtek, ed. *ECG/PPG Measurement Solution*. URL: <https://www.richtek.com/Design%20Support/Technical%20Document/AN057>.
- [29] Noé San Fabián. *Índice de calidad y evaluación de un sistema PPG*. Ed. por Universidad Pública de Navarra. 18 de sep. de 2018. URL: https://academica.e.unavarra.es/xmlui/bitstream/handle/2454/31636/TFM_NoSanFabian.pdf;jsessionid=1FC55CD2484AA5CBF0F1561CD4D18818?sequence=1.
- [30] Nicolas de Pinho Pereira. *Transmission and reflection PPG working principle*. Ed. por ResearchGate. URL: https://www.researchgate.net/figure/Transmission-and-reflection-PPG-working-principle-13_fig4_340789526.
- [31] Arduino. *What is Arduino?* 5 de feb. de 2018. URL: <https://www.arduino.cc/en/Guide/Introduction>.
- [32] Arduino, ed. *Arduino License*. URL: <https://support.arduino.cc/hc/en-us/articles/4415094490770-Licensing-for-products-based-on-Arduino>.
- [33] Arthur Daniel Limantara. *Arduino Uno ATmega328 Microcontroller*. Ed. por ResearchGate. URL: https://www.researchgate.net/figure/Arduino-Uno-ATmega328-Microcontroller_fig1_328376805.
- [34] OCompra, ed. *Kit Sensores Arduino*. URL: <https://www.ocompra.com/argentina/item/kit-sensores-arduino-38-color-ir-humedad-temperatura-y-mas-681071164/>.
- [35] Tim Youngblood. *How To Use Arduino's Analog and Digital Input/Output (I/O)*. Ed. por All About Circuits. 10 de jun. de 2015. URL: <https://www.allaboutcircuits.com/projects/using-the-arduinios-analog-io/>.
- [36] Arduino. *setup()*. 18 de feb. de 2019. URL: <https://www.arduino.cc/reference/en/language/structure/sketch/setup/>.

- [37] Arduino. *loop()*. 21 de jul. de 2019. URL: <https://www.arduino.cc/reference/en/language/structure/sketch/loop/>.
- [38] Ladyada. *Arduino Tutorial*. 27 de abr. de 2012. URL: <https://www.ladyada.net/learn/arduino/lesson4.html>.
- [39] Mozilla Developer Network. *WebSockets*. Ed. por Mozilla Developer Network. 4 de ago. de 2022. URL: https://developer.mozilla.org/es/docs/Web/API/WebSockets_API.
- [40] Django. 25 de oct. de 2022. URL: <https://www.djangoproject.com/start/>.
- [41] Django Channels. 25 de oct. de 2022. URL: <https://channels.readthedocs.io/en/stable/>.
- [42] Greeshma Joseph et al. «Photoplethysmogram (PPG) signal analysis and wavelet de-noising». En: *2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD) (2014)*, págs. 1-5.
- [43] Arduino. *map()*. 6 de sep. de 2022. URL: <https://reference.arduino.cc/reference/en/language/functions/math/map/>.
- [44] Julián Maida Esteban Gabriel; Pacienza. *Metodologías de desarrollo de software*. Ed. por Pontificia Universidad Católica Argentina. 1 de dic. de 2015. URL: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>.
- [45] Woodyweb, ed. *Metodologías de desarrollo*. 24 de ago. de 2015. URL: <https://woodyweb.wordpress.com/2015/08/24/metodologias-de-desarrollo/>.

Agradecimientos

Quiero expresar mi profundo agradecimiento al Dr. Javier de Pedro Carracedo por su valiosa guía y asesoramiento a lo largo de este trabajo de graduación, todo su conocimiento en cuanto a señales biológicas ha sido abrumador a la vez que apasionante de estudiar, y, sobre todo, agradecerle por recordarnos que los jóvenes podemos aspirar a muchísimo más y que tenemos el deber de hacer el cambio para bien en nuestra sociedad.

Deseo agradecer también a mis compañeros de la carrera de Ingeniería de Sistemas Informáticos los cuales compartimos la pasión de la informática y su vertiginosa evolución continua. Estoy agradecido por todo el aprendizaje que compartimos mutuamente como compañeros de estudios y estoy seguro que todos seremos profesionales de cambio para nuestro país.

Quiero expresar mi más sincero agradecimiento a mi familia y amigos por su incondicional apoyo y aliento a lo largo de mi carrera y trabajo de grado. Su amor, motivación y comprensión han sido motor para mi éxito académico incluso en los tiempos difíciles.