



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Diseño de un sistema simulador de las señales eléctricas del corazón y posibles aplicaciones

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO ELECTRÓNICO

PRESENTAN:

PADILLA MORALES EDGAR

VÁZQUEZ MARTÍNEZ LUIS ENRIQUE

INGENIERO MECÁNICO ELECTRICISTA

AREA: ELÉCTRICA ELECTRÓNICA

PRESENTAN

CORTÉS SOTO FRANCISCO OCTAVIO

GONZÁLEZ HUERTA NORBERTO RENÉ

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

NAVARRETE OROPEZA HÉCTOR

ASESOR DE TESIS:

M. I. JUAN FERNANDO SOLÓRZANO PALOMARES

MÉXICO, D.F., CIUDAD UNIVERSITARIA, SEPTIEMBRE 2012



AGRADECIMIENTOS

Al M. I. Juan Fernando Solórzano Palomares, por su confianza, su incondicional apoyo, paciencia, sus múltiples aportaciones, y por el tiempo dedicado a la asesoría de este proyecto.

A la Directora de Medicina del Deporte de Ciudad Universitaria, Dirección General de Actividades Deportivas y Recreativas, U.N.A.M. Dra. María Cristina Rodríguez Gutiérrez.

A la Lic. Mónica Juárez Hernández por las facilidades y orientaciones, prácticas y técnicas, brindadas para la realización de este trabajo.

A todos los profesores, dentro y fuera de la Universidad, que construyeron nuestra formación, fortaleciendo el criterio, y encaminando el futuro a la superación constante de todos nosotros.

A los compañeros y todas las personas que participaron directa o indirectamente, por estar presentes en el desarrollo de este proyecto.

Agradezco a mis dos abuelas Alejandra y Victoria por cuidarme donde quiera que estén, por enseñarme a luchar por lo que quiero. Quedo en deuda con mis abuelos, Aniceto y Luis, estoy seguro que en donde estén siguen divirtiéndose, muchas gracias por contagiar su alegría.

Gracias a Angelina y Luis, mis padres, por darme la vida, por consentirme, por darme educación, guiar y respetar mis decisiones, por ser mi ejemplo, por darme lo mejor de ustedes, y por ser felices tan solo porque yo lo soy.

Agradezco a mis hermanos Carlos y David, por saber que pase lo que pase siempre estaremos juntos en cualquier situación y somos uno para todos.

Gracias Lucía, por tener la fuerza para hacer realidad mis mejores sueños, hacerme tan feliz, por motivarme día a día, por oír y escucharme en cualquier momento, por siempre tener una solución, por esperarme, y por ayudarme a conseguir mi obra maestra. Gracias a Alvar por hacerme buscar mil y una formas para educar a un hijo.

Muchas gracias a la vida por Rodrigo, por su presencia, por darme sus sonrisas, por tener el cariño más sincero que un padre pueda pedir, por su motivación para seguir superándome, por todas sus ocurrencias, sus habilidades, su alegría, su energía, y darme la fuerza para intentar ser el mejor papá que pueda merecer.

Estoy muy agradecido con todos mis Maestros por darme una formación y educación, al Ing. Froilán Gutiérrez, M.I. Juan Manuel Gómez, Técnico Emmanuel Quintana, e Ing. Gonzalo García por su valiosa cooperación, a Tomás Tavera y Antonio Escutia que contribuyeron a valorar mi estado físico,.

Gracias a mi familia por su apoyo, mis tíos, primos, sobrinos, cuñadas, suegros, amigos, compañeros, alumnos, y a todos los que me faltó mencionar, a todos les estoy gradecido. Quiero que todos sepan que día a día intento ser una persona más valiosa para toda la gente cercana a mí. Muchas gracias por ser parte de mi mundo.

Edgar Padilla Morales

A mi bisabuela Aurora, mi abuela Esperanza, mi tía Margarita, quienes dieron todo por mi existir, a mi madre y padre por darme el hálito de vida, a mi hermano que sus "NO" me han impulsado al "SÍ" y que en momentos difíciles me ha apoyado, a mis maestros de las diversas disciplinas que me han formado y definen mi actual existencia, a mis amigos que en realidad son grandes hermanos. En especial al ser más hermoso que conozco y que admiro, a la MUJER, ser que en sus diversas manifestaciones en mi vida me ha acompañado y guiado mi camino y aún lo hace incondicionalmente, GRACIAS.

Francisco Octavio Cortés Soto

A Dios, quien es mi fortaleza en los momentos difíciles y me da la oportunidad de lograr este paso tan importante en mi vida personal y profesional.

A mis padres por su apoyo moral. Gracias por su paciencia.

Al Maestro Juan Gastaldi Pérez por haber aceptado con gusto ser director de esta tesis, gracias por su confianza, por su apoyo y tiempo.

Héctor Navarrete Oropeza

Ha sido un largo y difícil camino que tuve que recorrer y me siento feliz de haberlo terminado siempre con el apoyo de mi padre, quien me enseñó a seguir adelante y me dio las herramientas necesarias para no detenerme, de mi madre quien me apoyó día y noche y siempre me guió por el buen camino y por último mis hermanos que han compartido conmigo grandes momentos de alegría. Y ahora todo mi esfuerzo se los debo a ellos, mi única familia y apoyo que jamás olvidaré.

Luis Enrique Vázquez Martínez

A todos los ingenieros que me dieron clases que con sus consejos y enseñanzas lograron que yo amara a esta carrera, a los que participaron e hicieron posible que este proyecto llegara a su fin

A mi Mamá que nunca dejo que yo flaqueara o me rindiera, en ningún momento de mis estudios, a toda mi familia que siempre estuvo ahí para brindarme su apoyo incondicional

A mis amigos que siempre estuvieron en las buenas y en las malas, en los desvelos, entregas de proyectos, exámenes y en los múltiples triunfos que nos dio y ha dado esta facultad.

Finalmente a mi esposa e hijas que me sirvieron de gran de inspiración para cerrar este ciclo y el poder realizar este objetivo de la mejor manera, dándome fuerzas y entendimiento en los momentos que más lo necesite.

GRACIAS A TODOS

Norberto René González Huerta

ÍNDICE

CONTENIDO	Página
Capítulo 1. Introducción	10
Capítulo 2. Fisiología del corazón	17
2.1. Morfología	18
2.2. Latido del corazón	25
2.3. Actividad eléctrica del corazón	29
2.4. Electrocardiograma (ECG)	31
Capítulo 3. Bases electrónicas y principios de programación para PICs	54
3.1. Introducción	55
3.2. Lenguajes de programación	66
3.3. PIC16F887	69
3.4. Compilador Mikro C	78
3.5. Simulador PROTEUS para microcontroladores PIC	92
Capítulo 4. Sistema Simulador	97
4.1. Descripción general	98
4.2. Etapa inicial del Sistema	102
4.3. Programación del PIC16F887	106
4.4. Etapa de conversión de las señales digitales a analógicas	117
4.5. Etapa final, ECG patrón	126
Capítulo 5. Conclusiones: Resultados y Aplicaciones	136
5.1. Diferentes usos	137
5.2. Beneficios y conclusiones	140
5.3. Optimización del Sistema	142
Apéndices	144
Apéndice A Links y hojas de datos	144
Apéndice B Diagrama de flujo	146
Apéndice C Programa	147-149
Bibliografía	150
Índice de Figuras	
Figura 1.1. Primer electrocardiograma	11
Figura 1.2. Señal continua	14
Figura 1.3. Señal discreta	15
Figura 2.1.1. Cavidades y válvulas del corazón	18
Figura 2.1.2. Flujo sanguíneo del corazón durante el proceso de	

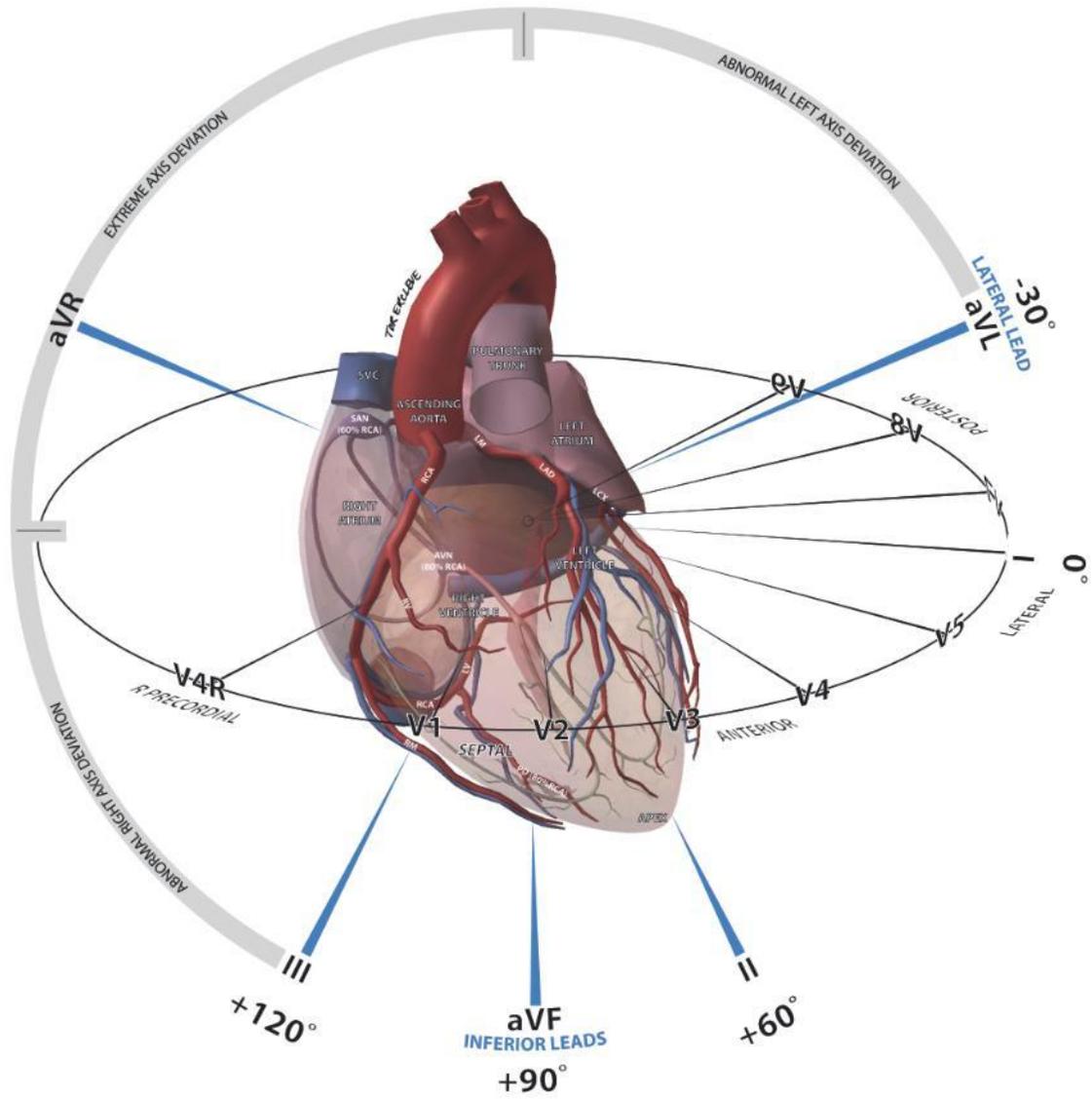
sístole y diástole	22
Figura 2.2.1. Tejido excitoconductor del corazón	27
Figura 2.3.1. Nodo SA y AV	30
Figura 2.4.1. Gráfica del potencial de acción (PA) de la célula contráctil miocárdica ventricular	32
Figura 2.4.2. Gráfica del potencial de acción (PA)	33
Figura 2.4.3. Actividad eléctrica del corazón en su recorrido vectorial, y sumatoria de estas señales	34
Figura 2.4.4. Sistema de conducción intrínseca del corazón (vectores de despolarización)	34
Figura 2.4.5. Generación del ECG	36
Figura 2.4.6. Componentes gráficos del ECG	39
Figura 2.4.7. Captura de un ECG	40
Figura 2.4.8. Deflexiones positiva y negativa en el ECG	41
Figura 2.4.9. Derivaciones Bipolares (Einthoven)	42
Figura 2.4.10. Derivaciones Unipolares (Goldberger)	42
Figura 2.4.11. Derivaciones Precordiales	43
Figura 2.4.12. Nomenclatura electrocardiográfica	44
Figura 2.4.13. Ejes y planos anatómicos	45
Figura 2.4.14. Triángulo de Einthoven	47
Figura 2.4.15. Ejes y planos geométricos del corazón en su posición anatómica	49
Figura 2.4.16. Posición anatómica del corazón	50
Figura 2.4.17. Vectores eléctricos del corazón	52
Figura 2.4.18. Vectores en los planos frontal y horizontal	53
Figura 3.1.1. Símbolo de la resistencia	56
Figura 3.1.2. Representación del capacitor	57
Figura 3.1.3. Representación del inductor	57
Figura 3.1.4. Diagrama, imagen real y circulación de corriente del diodo	58
Figura 3.1.5. Corte transversal simplificado de un transistor de unión bipolar NPN	59
Figura 3.1.6. Amplificador operacional común	60
Figura 3.1.7. Esquema de diseño de Un Microprocesador Tradicional.	62
Figura 3.3.1. Esquema físico del PIC 16F887	69
Figura 3.3.2. Representación interna del microcontrolador	71
Figura 3.3.3. Representación gráfica de bancos de memoria	76
Figura 3.5.1. Dispositivos disponibles en PROTEUS	94
Figura 3.5.2. Diseño de circuito electrónico	95
Figura 3.5.3. Edición de parámetros	96
Figura 3.5.4. Iconos de simulación	96
Figura 4.1.1. Diagrama a bloques del Sistema Simulador	98
Figura 4.2.1. Familia de PICs de Microchip	103
Figura 4.2.2. PIC16F887	103
Figura 4.2.3. Leds conectados al puerto D del PIC	104
Figura 4.2.4. ECG con 60 ppm	105
Figura 4.3.1. Ventana principal Mikro C	106

Figura 4.3.2. Pantalla de bienvenida: <u>N</u> ew Project	107
Figura 4.3.3. Paso 1	108
Figura 4.3.4. Paso 2	108
Figura 4.3.5. Paso 3	109
Figura 4.3.6. Paso 4	110
Figura 4.3.7. Paso 5	110
Figura 4.3.8. Paso 6	111
Figura 4.3.9. Nuevo programa Mikro C	112
Figura 4.3.10. Ventana principal ISIS de PROTEUS	115
Figura 4.3.11. Elementos del Sistema Simulador	116
Figura 4.4.1. Aplicación típica DAC0808	117
Figura 4.4.2. Circuito con el potenciómetro	119
Figura 4.4.3. Circuito del DAC	120
Figura 4.4.4. Circuito del filtro pasa bajos	121
Figura 4.4.5. Circuito del offset	122
Figura 4.4.6. Cálculo de R11	123
Figura 4.4.7. Bloque de atenuación	124
Figura 4.5.1. Aplicación típica regulador de voltaje	126
Figura 4.5.2. Aplicación típica del convertidor de voltaje	127
Figura 4.5.3. Fuente con batería de 9 [V]	127
Figura 4.5.4. Fuente rectificadora y fuente con batería de 9 [V]	128
Figura 4.5.5. Fuente rectificadora desconectada	129
Figura 4.5.6. Sistema Simulador completo	130
Figura 4.5.7. Derivación bipolar I	132
Figura 4.5.8. Derivación bipolar II	132
Figura 4.5.9. Derivación bipolar III	133
Figura 4.5.10. Derivación aumentada aVR	133
Figura 4.5.11. Derivación aumentada aVL	134
Figura 4.5.12. Derivación aumentada aVF	125
Figura 5.1.1. Quirófano	138
Figura 5.1.2. ECG patrón	139
Figura 5.2.1. ECG del Sistema Simulador	141
Figura 5.3.1. PICkit 2	142

Índice de Tablas

Tabla 2.4.1. Posición y polaridad de las derivaciones	51
Tabla 3.3.1. Composición general del PIC 16F887	69
Tabla 3.3.2. Descripción de pines PIC16F887	72-74
Tabla 3.4.1. Variables	80
Tabla 3.4.2. Operadores aritméticos	82
Tabla 3.4.3. Tabla de operadores	86
Tabla 4.1.1. Elementos con costos del Sistema Simulador	101
Tabla 4.5.1. Definición de derivaciones	131

CAPÍTULO 1



INTRODUCCIÓN

En el sector salud, tanto público como privado, los médicos y técnicos de la salud utilizan una gran variedad de aparatos electro-médicos, que les sirven para dar diferentes lecturas del comportamiento interno y externo del cuerpo humano, relacionados con la salud de los pacientes atendidos.

Uno de esos aparatos electro-médicos es el Electrocardiógrafo, que proporciona lecturas acerca de la actividad eléctrica del corazón de los pacientes. Dicha lectura es conocida como electrocardiograma (ECG o EKG), donde un ECG resulta fundamental para el diagnóstico del funcionamiento y posible anomalía cardíaca, por consiguiente alguna enfermedad relacionada con el corazón.

El electrocardiógrafo es un aparato, producto de una serie de avances tecnológicos, que permite manifestar de una manera gráfica la actividad eléctrica del corazón mediante electrodos conectados en determinados puntos sobre la superficie corporal, nombrados derivaciones. Este aparato está constituido por un galvanómetro que, como lo ideó Einthoven, está compuesto básicamente por una cuerda de platino, esta cuerda atraviesa un gran electroimán a través de su campo magnético, y es así como los pequeños cambios de potencial hacen que esta cuerda se desplace y grafique sobre un papel fotográfico. Uno de los primeros electrocardiógrafos se ilustra en la figura 1.1. [L06]



Figura 1.1. Primer electrocardiógrafo

Alrededor del año 1901, Willem Einthoven publicó su primer trabajo llamado "Un nouveau galvanometre", en el que habla de los resultados obtenidos con el galvanómetro de cuerda ideado por él y su equipo. Este electrocardiógrafo o galvanómetro de cuerda era enorme, pesaba más de 250 [kg], y necesitaba de 5 personas para su manejo. Einthoven sentó las bases de la electrocardiografía e introdujo la nomenclatura para las deflexiones registradas. Actualmente existen diseños electrónicos digitales con gran tecnología.

Ante el aumento en el mundo desarrollado de los padecimientos cardiacos, debidos al estrés de la vida moderna, sin duda alguna, hoy en día la cardiología se ha convertido quizás en la especialidad médica de mayor demanda, tanto en relación con las medidas de diagnóstico y terapéuticas como, principalmente, de prevención.

La utilidad clínica del ECG tiene varias ventajas, como diagnosticar enfermedades cardiovasculares, por mencionar algunos ejemplos: reconocimiento de arritmias, trastornos de conducción, isquemia miocárdica y trastornos metabólicos. Además de que el uso en los pacientes no representa riesgos, ya que se utiliza una técnica no invasiva, es decir, que no se introduce en el cuerpo. [L10]

Problemática detectada

Un inconveniente llega cuando la lectura del ECG no es correcta, debido al mal funcionamiento del electrocardiógrafo, ya sea por un mal o nulo mantenimiento. Una manera eficiente para prevenir lecturas erróneas sería mediante un mantenimiento preventivo, por medio de una calibración rutinaria.

En cualquier lugar donde se cuente con un aparato electro-médico se debe de partir de la premisa que el aparato se encuentre funcionando correctamente, ya que al valorar a los pacientes se debe tener plena confianza en contar con una lectura precisa para el diagnóstico cardiológico, y por consecuencia, de su estado de salud. Por tal motivo, es de suma importancia contar con equipos sin fallas y que estén debidamente calibrados.

Objetivo:

Diseñar un Sistema Simulador que proporcione un patrón de referencia para las señales eléctricas del corazón, y tenga una gran variedad de aplicaciones, de una forma sencilla, amigable y a bajo costo. Se obtendrá una señal idéntica al de un ECG, así como las principales derivaciones.

Método:

El diseño del Sistema Simulador se basa en los siguientes puntos:

- Obtener un ECG real, de una persona común, es decir, con una frecuencia cardíaca entre 60 y 90 pulsos por minuto.
- Ya con el ECG, realizar un muestreo con dos variables, una es el voltaje y la otra será el tiempo.
- Introducir la tabla obtenida por el muestreo a un microcontrolador PIC16F887, que a su vez proporcionará una señal digitalizada.
- Los datos digitales son convertidos a analógicos para que sean vistos en un osciloscopio para comprobarlo y ver una señal eléctrica, con la misma forma de onda de un ECG real, donde se podrá apreciar la variación de voltaje en función del tiempo. La conversión se lleva a cabo por medio de un convertidor digital – analógico, con el nombre comercial DAC0808.
- Se utilizará un proceso de filtrado para depurar la señal y obtener un ECG patrón para asegurar una copia fiel del ECG de una persona real.
- Ya con la señal depurada se adicionará un arreglo de resistencias para fungir como divisores de voltaje resistivos, para que tengan la función de simular cuatro partes del cuerpo, los dos brazos y las dos piernas, de esa forma cualquier electrocardiógrafo podrá conectarse al Sistema y obtener varias señales, que a su vez se ocuparán como las derivaciones, mismas que presenta el ECG de un electrocardiógrafo.
- El ECG patrón, que proporcionará el Sistema Simulador, podrá ser verificado por medio de una simulación, donde todos los elementos serán incluidos, y se podrá comprobar la copia exacta de un ECG real.

Resulta conveniente mencionar algunos conceptos básicos relacionados con el Sistema Simulador, tales como:

- Sistema: Es el conjunto de elementos que se organizan para realizar una función específica. Existen cinco tipos de sistemas: eléctricos, mecánicos (lineal y rotacional), hidráulicos, térmicos, e híbridos (combinación de los anteriores).
- Señal: Cuando un fenómeno físico tiene una variación con el tiempo y es observado o medido, es cuando se obtiene como resultado una señal. Todas las señales llevan información y de acuerdo a la aplicación serán analizadas.

Los conceptos de sistema y señal son muy amplios, ya que tienen varias subdivisiones. Cuando un sistema forma parte de otro sistema de mayor nivel, entonces se trata de un subsistema. Los subsistemas comparten información para cumplir con la finalidad u objetivo de todo el sistema. [L01]

Una señal puede depender de dos tipos de variables, con valores discretos y continuos. Para dejar claro los términos anteriores se puede mencionar: el voltaje de un fototransistor, en algún intervalo, puede tomar cualquier valor real, es decir, se trata de una señal continua; si se desea analizar el número de llamadas de algún teléfono, entonces sólo se tendrán valores enteros y por lo tanto es una señal discreta. A continuación se tiene la representación gráfica de una señal continua, la cual podría ser la forma de la señal de voltaje que llega a una casa, ver figura 1.2:

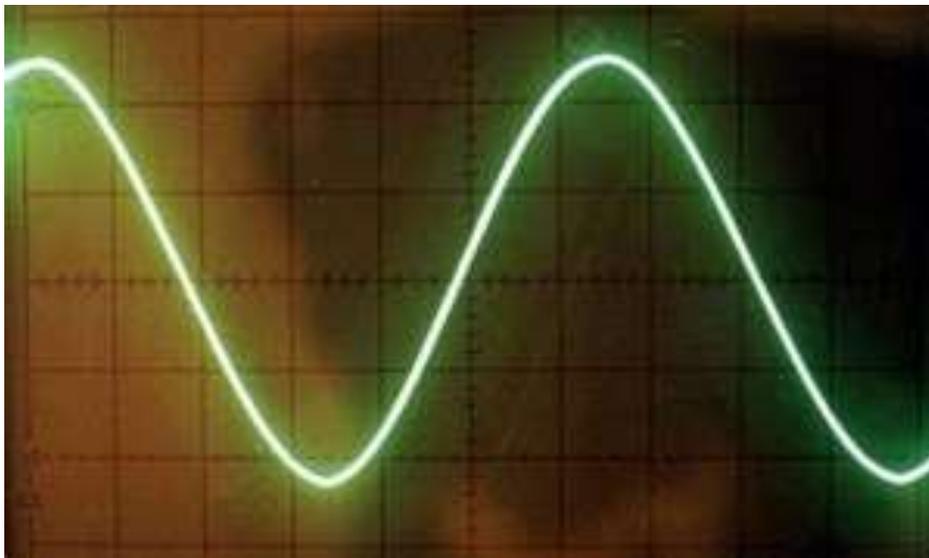


Figura 1.2. Señal continua

En la figura 1.3 se tiene una señal discreta a partir de tomarle muestras a una señal continua o análoga, como se aprecia, los valores del muestreo están bien definidos, es decir, tienen intervalos constantes, enseguida se presenta como se forma la señal discreta o también conocida como digital: [L16]

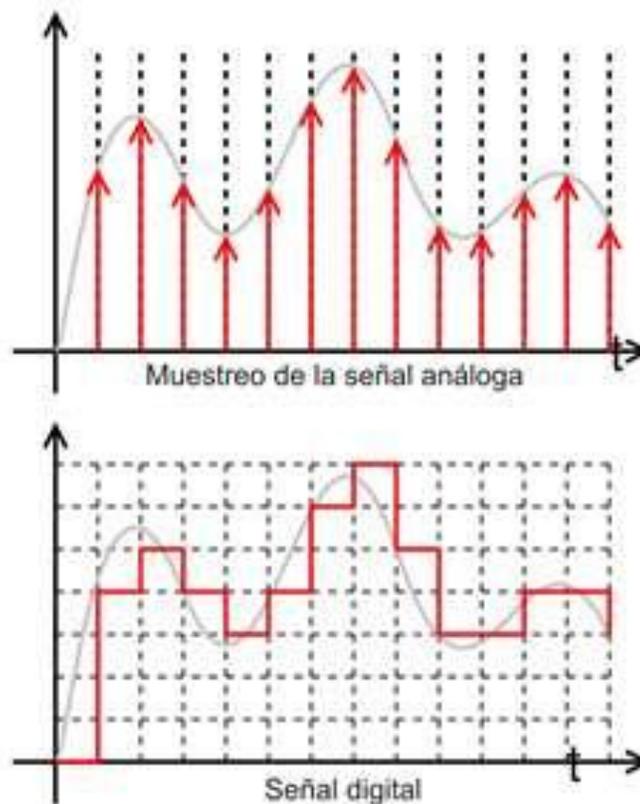


Figura 1.3. Señal discreta

En el capítulo 2 se dan los detalles con todo lo referente a la fisiología del corazón, incluyendo los aparatos electro-médicos, las diferentes enfermedades que se diagnostican con el ECG. Un breve panorama acerca de la morfología del corazón. Se explica también como el cuerpo humano genera señales eléctricas, que a su vez producen los latidos del corazón y finalmente se dan las bases para la generación de un ECG.

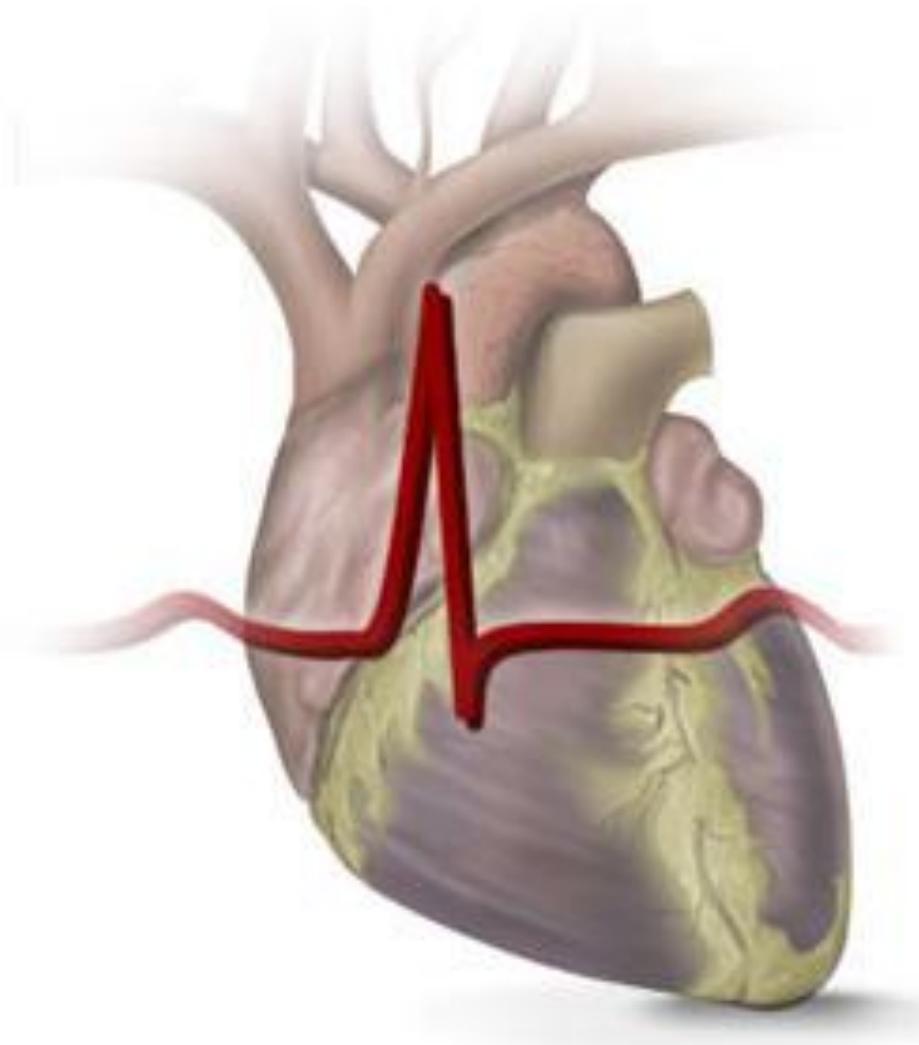
A través del capítulo 3 se dan las bases electrónicas, así como los principios de la programación para PICs; se incluye una breve reseña de la variedad de los lenguajes de programación que existen para utilizarlos en los microcontroladores. Se bosqueja el software Mikro C, el cual se usó para programar el PIC16F887, mostrando sus principales ventajas. Se muestran las herramientas básicas para el programa que se encargó de la simulación del Sistema Simulador, dicho programa se trata de ISIS de PROTEUS. Al final de este capítulo se proporcionan las

características del microcontrolador, que se encargó de digitalizar los valores que se consiguieron de un ECG real.

A lo largo del capítulo 4 se aborda la metodología que se siguió para diseñar el Sistema Simulador. Se tratan las diferentes etapas que integran al mismo, donde se explica a detalle la razón por la cual se eligieron cada uno de los elementos. También se muestra paso a paso la programación del microcontrolador, basándose en el software que se utilizó. En la etapa de conversión, debido a que el microcontrolador proporciona datos en forma digital, se considero conveniente transformarlos a datos analógicos para el despliegado de la señal eléctrica. Finalmente, se demuestra que al emplear el Sistema Simulador se obtiene un electrocardiograma con todas las características de uno real.

En el último capítulo se describen las diferentes aplicaciones que tiene el Sistema Simulador, tanto para el ámbito médico, educativo y para prevenir posibles fallas. Se comentan los beneficios con los que cuenta el Sistema, en funcionalidad y costo, principalmente. Se dan varios consejos para ir mejorando el funcionamiento, y una visión general para optimizar todo el Sistema Simulador. En este capítulo se exponen las conclusiones derivadas del uso del Sistema.

CAPÍTULO 2



FISIOLOGÍA DEL CORAZÓN

2.1. MORFOLOGÍA

El músculo cardíaco es del tamaño del puño de una persona, se divide en cuatro cavidades: dos superiores o atrios o aurículas y dos inferiores o ventrículos, según se muestra en la figura 2.1.1. Las aurículas reciben la sangre del sistema venoso, pasan a los ventrículos y desde ahí se envía a la circulación arterial.

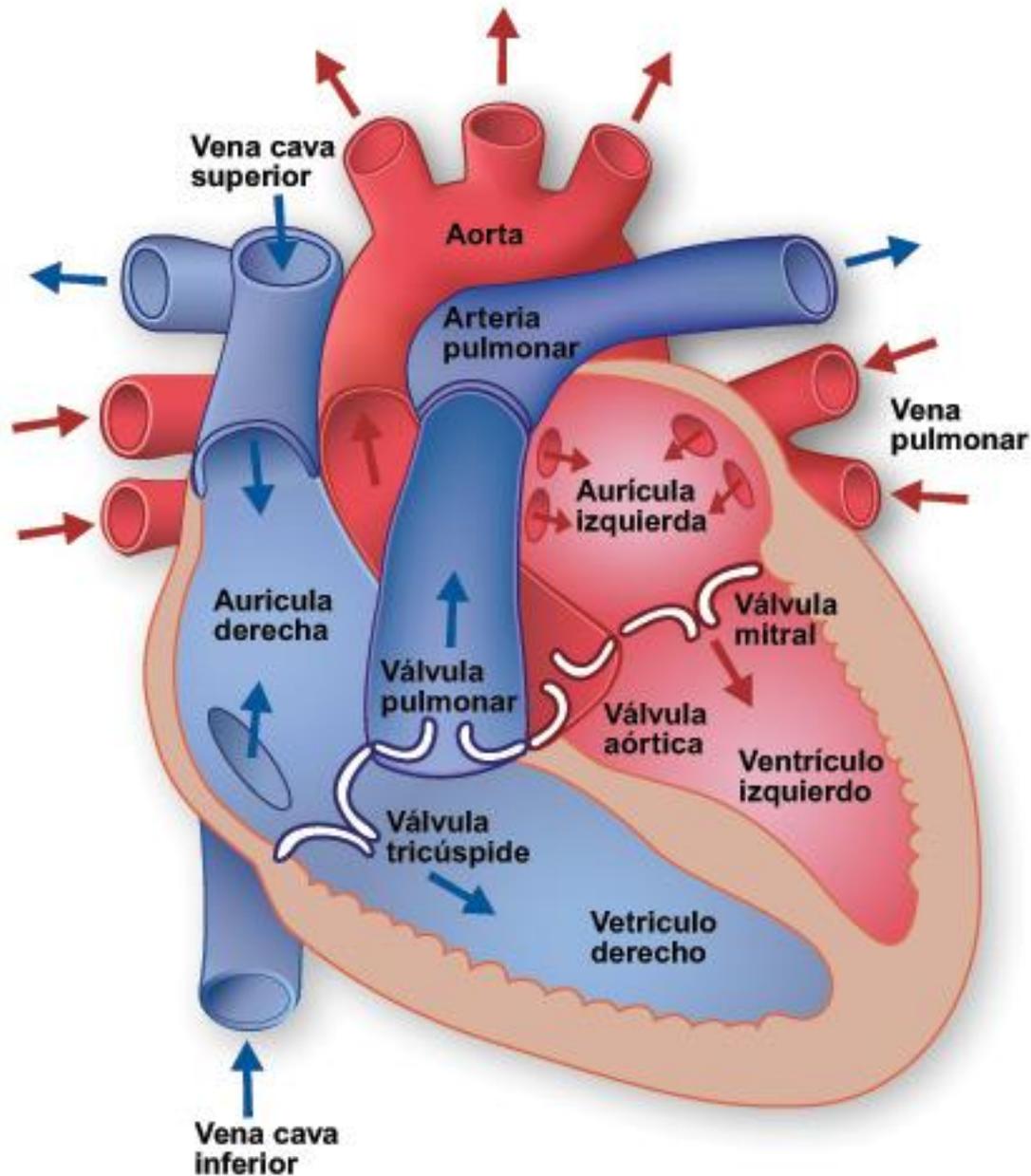


Figura 2.1.1. Cavidades y válvulas del corazón

La aurícula y el ventrículo derechos forman lo que comúnmente se denomina el corazón derecho, el cual recibe la sangre que proviene de todo el cuerpo que desemboca en la aurícula derecha a través de las venas cavas, superior e inferior; esta sangre (pobre en oxígeno) llega al ventrículo derecho, desde donde es enviada a la circulación pulmonar por la arteria pulmonar. Debido a que la resistencia de la circulación pulmonar es menor que la sistémica, la fuerza que el ventrículo debe realizar es menor, razón por la cual su tamaño muscular es considerablemente menor al del ventrículo izquierdo. Ver figura 2.1.1.

La aurícula izquierda y el ventrículo izquierdo forman el llamado corazón izquierdo. En esta parte del corazón se recibe la sangre de la circulación pulmonar que desemboca a través de las cuatro venas pulmonares en la porción superior de la aurícula izquierda. Esta sangre está oxigenada y proviene de los pulmones. El ventrículo izquierdo la envía por la arteria aorta para distribuirla por todo el organismo.

El tejido que separa el corazón derecho del izquierdo se denomina septo o tabique. Funcionalmente, se divide en dos partes no separadas: la superior o tabique interatrial, y la inferior o tabique interventricular. Éste último es especialmente importante, ya que por él discurre el haz de His, que permite llevar la pequeña corriente eléctrica hacia las partes más bajas del corazón. Más adelante se explica con más detalle la actividad eléctrica del corazón.

Las válvulas cardíacas son las estructuras que separan unas cavidades de otras, evitando que exista reflujo entre ellas, como se muestra en la figura 2.1.1. Están situadas en torno a los *orificios atrioventriculares* (o aurículo-ventriculares) y entre los ventrículos y las arterias de salida.

De la figura 2.1.1, las flechas indican el flujo normal de la sangre. El miocardio está formado por epicardio (capa exterior) y el endocardio (capa interior).

Dichas válvulas son:

- La válvula tricúspide, separa la aurícula derecha del ventrículo derecho.
- La válvula sigmoidea pulmonar, separa el ventrículo derecho de la arteria pulmonar.
- La válvula mitral o bicúspide, separa la aurícula izquierda del ventrículo izquierdo.
- La válvula sigmoidea aórtica, separa el ventrículo izquierdo de la arteria aorta.

El corazón presenta la siguiente estructura o capas, desde adentro hacia afuera:

- El endocardio, una membrana serosa de endotelio y tejido conectivo de revestimiento interno, con la cual entra en contacto la sangre. Incluye fibras elásticas y de colágeno, vasos sanguíneos y fibras musculares especializadas, las cuales se denominan Fibras de Purkinje. En su estructura encontramos las trabéculas carnosas, que dan resistencia para aumentar la contracción del corazón.
- El miocardio, una masa muscular contráctil, es el músculo cardíaco propiamente dicho; encargado de impulsar la sangre por el cuerpo mediante su contracción. Encontramos también en esta capa tejido conectivo, capilares sanguíneos, capilares linfáticos y fibras nerviosas. El miocardio está compuesto por células especializadas que cuentan con una capacidad que no tiene ningún otro tipo de tejido muscular del resto del cuerpo.
- El músculo cardíaco, como otros músculos, se puede contraer, pero también puede llevar un potencial de acción de conducción eléctrica, similar a las neuronas que constituyen los nervios, además, algunas de las células tienen la capacidad de generar un potencial de acción, conocido como automaticidad del músculo cardíaco.

- La irrigación sanguínea del miocardio es llevada a cabo por las arterias coronarias, ya que el corazón no puede nutrirse directamente de la sangre que circula a través de él a gran velocidad.
- La estimulación del corazón está coordinada por el sistema nervioso autónomo, tanto por parte del sistema nervioso simpático (aumentando el ritmo y fuerza de contracción) como por el sistema nervioso parasimpático (reduce el ritmo y fuerza cardíacos). Este sistema de conducción eléctrico explica la regularidad del ritmo cardiaco y asegura la coordinación de las contracciones auriculoventriculares.
- El epicardio, es una capa fina serosa mesotelial que envuelve al corazón llevando consigo capilares y fibras nerviosas. Esta capa se considera como parte del pericardio seroso.

La presión sanguínea constituye uno de los principales signos vitales y está determinada por la fuerza y el volumen de sangre bombeada, así como por el tamaño y la flexibilidad de las arterias, es decir, una medición de la fuerza que se aplica sobre las paredes de las arterias a medida que el corazón bombea sangre a través del cuerpo. El término presión sanguínea generalmente se refiere a la presión arterial, es decir, a la presión en las arterias más grandes, las arterias que forman los vasos sanguíneos que toman la sangre que sale desde el corazón. La presión de la sangre disminuye a medida que la sangre se mueve a través de arterias, arteriolas, vasos capilares y venas.

Conceptualmente, hay dos componentes o medidas de presión arterial que son:

- Presión arterial sistólica: corresponde al valor máximo de la tensión arterial en sístole (cuando el corazón se contrae). Se refiere al efecto de presión que ejerce la sangre eyectada del corazón sobre la pared de los vasos.
- Presión arterial diastólica: corresponde al valor mínimo de la tensión arterial cuando el corazón está en diástole o entre latidos cardíacos.

Depende fundamentalmente de la resistencia vascular periférica. Se refiere al efecto de distensibilidad de la pared de las arterias, es decir el efecto de presión que ejerce la sangre sobre la pared del vaso.

La presión arterial es medida por medio de un dispositivo que usa la altura de una columna de mercurio para reflejar la presión de circulación, por ello las lecturas de presión arterial o sanguínea se miden en milímetros de mercurio (mmHg) y usualmente se toma la lectura de dos números: por ejemplo, 120 sobre 75 (escrito como 120/75). La presión de pulso es la diferencia entre la presión sistólica y la diastólica.

El número superior es la lectura de la presión arterial sistólica y representa la presión máxima ejercida cuando el corazón se contrae. El número inferior es la lectura de la presión arterial diastólica y representa la presión en las arterias cuando el corazón se encuentra en reposo.

Estas medidas de presión sanguínea tienen grandes variaciones de un individuo a otro ya que no son estáticas, experimentan variaciones naturales entre un latido del corazón a otro y a través del día; también cambian en respuesta al estrés, por factores alimenticios, por medicamentos, o por enfermedades, y en general por el estilo de vida.

La hipertensión arterial es el aumento de la presión arterial, ya sea de la sistólica o de la diastólica. La hipertensión, junto con la hipercolesterolemia y el tabaquismo, es uno de los tres factores de riesgo cardiovascular más importante y modificable. Es una enfermedad silente (que no da síntomas, silenciosa), en sus primeros estados.

La hipotensión arterial es el descenso de la presión arterial por debajo de los límites promedio o considerados normales.

Junto con la temperatura del cuerpo, la presión sanguínea es el parámetro fisiológico más comúnmente medido.

La referencia a la presión sistólica y diastólica da pie para explicar la apertura y cierre de las válvulas del corazón, así como las entradas y salidas de sangre, como el resultado lógico de los cambios de presión que se producen en cada momento de la sístole y la diástole, ver figura 2.1.2.

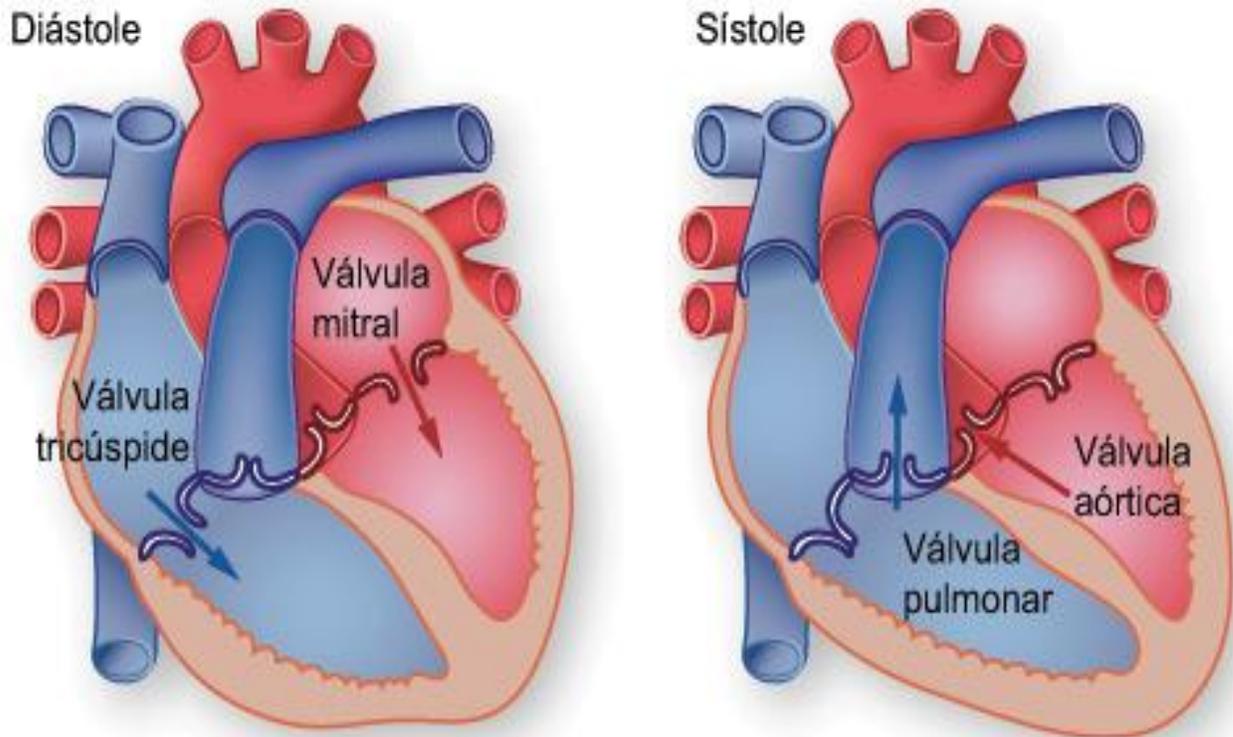


Figura 2.1.2. Flujo sanguíneo del corazón durante el proceso de sístole y diástole

. Las válvulas que controlan el flujo de la sangre por el corazón son cuatro:

- La válvula tricúspide controla el flujo sanguíneo entre la aurícula derecha y el ventrículo derecho.
- La válvula pulmonar controla el flujo sanguíneo desde el ventrículo derecho hacia las arterias pulmonares, las cuales transportan la sangre a los pulmones para oxigenarla.
- La válvula mitral o bicúspide permite que la sangre rica en oxígeno

(proveniente de los pulmones) pase de la aurícula izquierda al ventrículo izquierdo.

- La válvula aórtica permite que la sangre rica en oxígeno pase desde el ventrículo izquierdo hacia la aorta, la arteria más grande del cuerpo, la cual transporta la sangre al resto del organismo.

Cabe hacer notar que durante la expulsión de la sangre hacia las arterias, en la fase de sístole, las válvulas tricúspide y mitral o bicúspide deben permanecer cerradas para que la sangre no se devuelva al lecho venoso y para mantener la presión sanguínea hasta el máximo. Luego, en la fase de diástole, estas válvulas se abrirán para la entrada de un nuevo flujo de sangre, cerrándose las válvulas pulmonar y aórtica. [L11]

2.2. LATIDO DEL CORAZÓN

Cada latido del corazón desencadena una secuencia de eventos llamados ciclos cardiacos. Cada ciclo consiste principalmente en tres etapas: sístole auricular, sístole ventricular y diástole.

El ciclo cardíaco hace que el corazón alterne entre una contracción y una relajación aproximadamente 75 veces por minuto, es decir, el ciclo cardíaco dura 8 décimas de segundo en promedio. Para una persona común se considera un rango del ciclo cardíaco que va desde 60 hasta 100 pulsaciones por minuto.

Durante la sístole auricular, las aurículas se contraen y proyectan la sangre hacia los ventrículos, una vez que la sangre ha sido expulsada de las aurículas, las válvulas auriculoventriculares (ubicadas entre las aurículas y los ventrículos) se cierran. Esto evita el reflujo (retorno o devolución) de sangre hacia las aurículas. El cierre de estas válvulas produce el sonido familiar del latido del corazón. Esta etapa del proceso dura aproximadamente una décima de segundo.

La sístole ventricular implica la contracción de los ventrículos expulsando la sangre hacia el sistema circulatorio, una vez que la sangre es expulsada, las dos válvulas sigmoideas, la válvula pulmonar en la derecha y la válvula aórtica en la izquierda, se cierran. Esta etapa del proceso dura aproximadamente 3 décimas de segundo.

Por último, la diástole es la relajación de todas las partes del corazón que permite la llegada de nueva sangre. Esta etapa del proceso dura aproximadamente 4 décimas de segundo.

En el proceso completo se pueden escuchar dos golpecitos y como se mencionó anteriormente se lleva a cabo 75 veces por minuto en promedio: el de las válvulas al cerrarse (mitral y tricúspide) y la apertura de la válvula sigmoidea aórtica.

La expulsión rítmica de la sangre provoca el pulso que se puede palpar en las arterias: arteria radial, arteria carótida, arteria femoral, etcétera.

Si se observa el tiempo de contracción y de relajación se verá que las aurículas están en reposo aproximadamente 7 décimas de segundo y los ventrículos unas 5 décimas de segundo. Eso quiere decir que *el corazón pasa más tiempo en reposo que en trabajo*.

En cuanto a la excitación del corazón, el músculo cardíaco es biogénico, esto significa que se excita así mismo, a diferencia del músculo esquelético que necesita de un estímulo consciente o reflejo.

Las contracciones rítmicas del corazón se producen espontáneamente, pero su frecuencia puede ser afectada por las influencias nerviosas u hormonales, por el ejercicio físico o por la percepción de un peligro, etcétera.

El corazón tiene las siguientes características:

- Batmotropismo: que puede ser estimulado, manteniendo un umbral.
- Inotropismo: que se contrae bajo ciertos estímulos.
- Cronotropismo: que puede generar sus propios impulsos.
- Dromotropismo: es la conducción de los impulsos cardíacos mediante el sistema excitoconductor.
- Lusitropismo: es la relajación del corazón bajo ciertos estímulos.

Para entender cómo y por qué late el corazón se deben conocer las características básicas de la electrofisiología cardíaca, ver figura 2.2.1.

Físicamente, el corazón está constituido por dos tipos de tejidos: el tejido especializado excitoconductor y el miocardio contráctil.

El primero está representado por el nódulo sinusal o de Keith-Flack (también conocido como nódulo sinoauricular o marcapasos del corazón), el nódulo auriculoventricular o de Aschoff-Tawara, el haz de His, las ramas derecha e izquierda y la red de fibras de Purkinje.

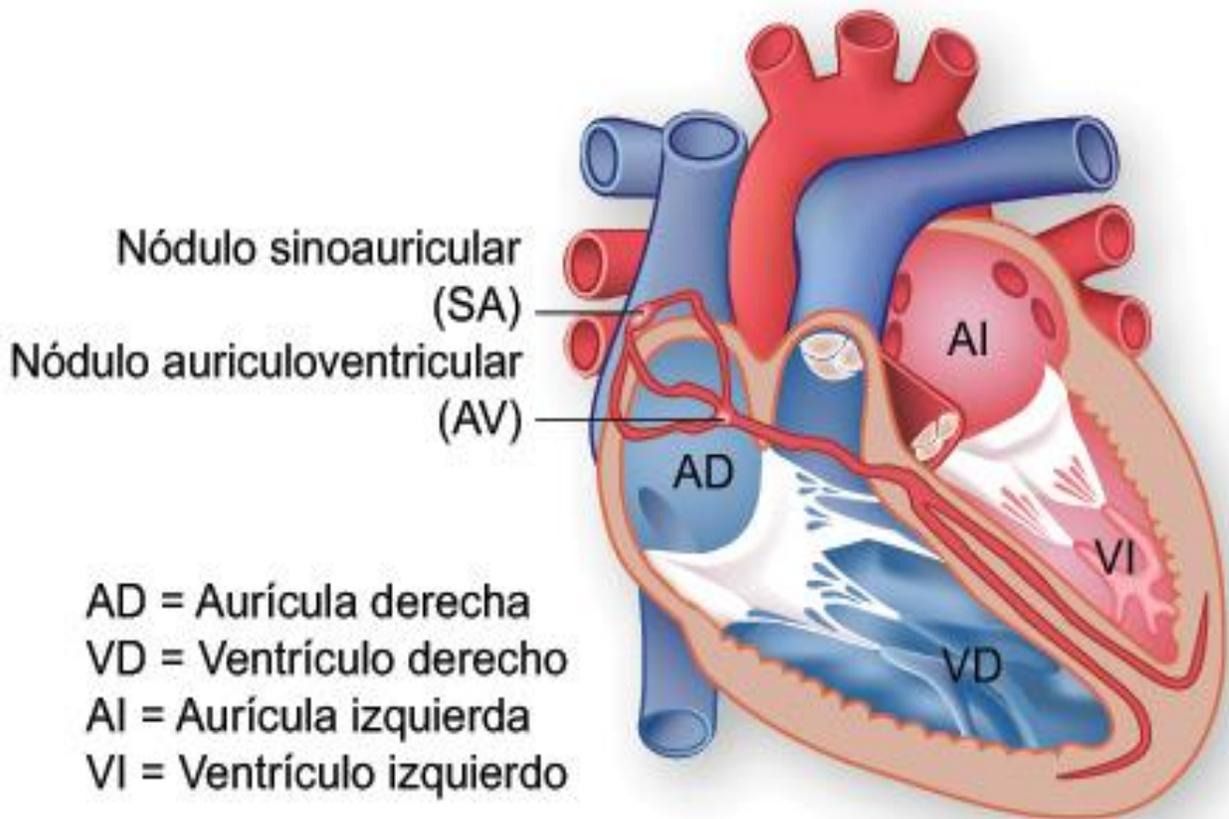


Figura 2.2.1. Tejido excitoconductor del corazón

Las fibras de Purkinje son fibras muy grandes y transmiten potenciales de acción a una velocidad seis veces mayor que la del músculo ventricular normal y 150 veces mayor que la de algunas fibras del nódulo auriculoventricular.

Estas corrientes iónicas producen un flujo continuo bidireccional a través de la membrana celular, generando potenciales eléctricos. Esta actividad eléctrica puede ser analizada con electrodos situados en la superficie de la piel, llamándose a esta prueba electrocardiograma (EKG o ECG).

Cuatro son las propiedades fundamentales del corazón:

- Automatismo, el corazón genera su propia actividad eléctrica, de acuerdo a lo anteriormente mencionado sobre las corrientes iónicas y los potenciales de acción. El ritmo cardíaco normal depende del automatismo del nódulo sinusal o sinoauricular (SA). La expresión que

se utiliza para denominar el ritmo cardíaco normal es ritmo sinusal. La frecuencia del automatismo sinusal oscila entre 60 y 100 ciclos por minuto. Se dice que hay bradicardia sinusal cuando hay una frecuencia menor a 60 latidos por minutos, y taquicardia sinusal cuando hay una frecuencia de más de 100 latidos por minuto. El automatismo intrínseco del nódulo auricoventricular (AV) oscila alrededor de 45 ciclos por minuto. El sistema His-Purkinje tiene una frecuencia aún más baja, alrededor de 30 por minuto. En condiciones de normalidad, el automatismo de estos focos no se hace evidente por la mayor frecuencia del nódulo sinusal.

- Conductibilidad, es la propiedad del tejido especializado de conducción y del miocardio contráctil que permite que un estímulo eléctrico originado en el nódulo sinusal o en cualquier otro sitio, difunda con rapidez al resto del corazón. La velocidad de conducción del estímulo varía en función del tejido considerado. Por ejemplo: el nódulo o nodo AV tiene una velocidad de conducción lenta, esta particularidad tiene su razón, sucede cuando se produce un retraso en la conducción del estímulo que permita la contracción de ambas aurículas en forma previa a la contracción ventricular.
- Excitabilidad, responde a un estímulo originando un potencial de acción propagado.
- Contractilidad es la capacidad intrínseca del músculo cardíaco de desarrollar fuerza y acortarse.

El automatismo es parte o consecuencia del tejido especializado excitoconductor, propiedad de la cual carece el miocardio contráctil. Sin embargo, ambos tipos de tejido tienen como característica común la propiedad de generar corrientes eléctricas de muy bajo voltaje como consecuencia de los desplazamientos iónicos debidos al Potasio (K+) y al Sodio (Na+), al Cloro (Cl-) y al Calcio (Ca++) fundamentalmente, que continuamente se están produciendo. [L10]

2.3. ACTIVIDAD ELÉCTRICA DEL CORAZÓN

El corazón funciona por impulsos eléctricos generados por algunas células del tejido muscular de dicho órgano. Son impulsos similares a los de otras células, como los generados por las neuronas y las células de los músculos esqueléticos.

Las células en cuestión son las del miocardio, tejido muscular del corazón responsable de bombear la sangre por el sistema circulatorio del cuerpo. El miocardio o músculo cardíaco, funciona involuntariamente ya que es auto-excitable. Posee y puede recibir inervación del sistema nervioso simpático, pero late sin el estímulo de éste.

El sistema nervioso simpático forma parte del sistema nervioso autónomo, también es conocido como sistema nervioso vegetativo, el cual transmite información desde las distintas vísceras del cuerpo a los músculos, las glándulas y los vasos sanguíneos, sin que seamos conscientes de ello. Así, el sistema nervioso simpático se diferencia del sistema nervioso somático que está formado por neuronas sensitivas y llevan información, de la que sí somos conscientes, a distintas partes del cuerpo.

En el miocardio hay diferentes tipos de células:

- Especializadas: son las que generan y conducen los impulsos eléctricos (o nerviosos)
- Y las contráctiles: son capaces de recibir dichos impulsos y contraer el músculo cardíaco.

El automatismo máximo radica en el nodo sinusal (SA), el cual constituye el marcapasos natural de este órgano, siendo el nodo auriculo ventricular (AV) el que toma el relevo si falla el sinusal. Ver figura 2.3.1.

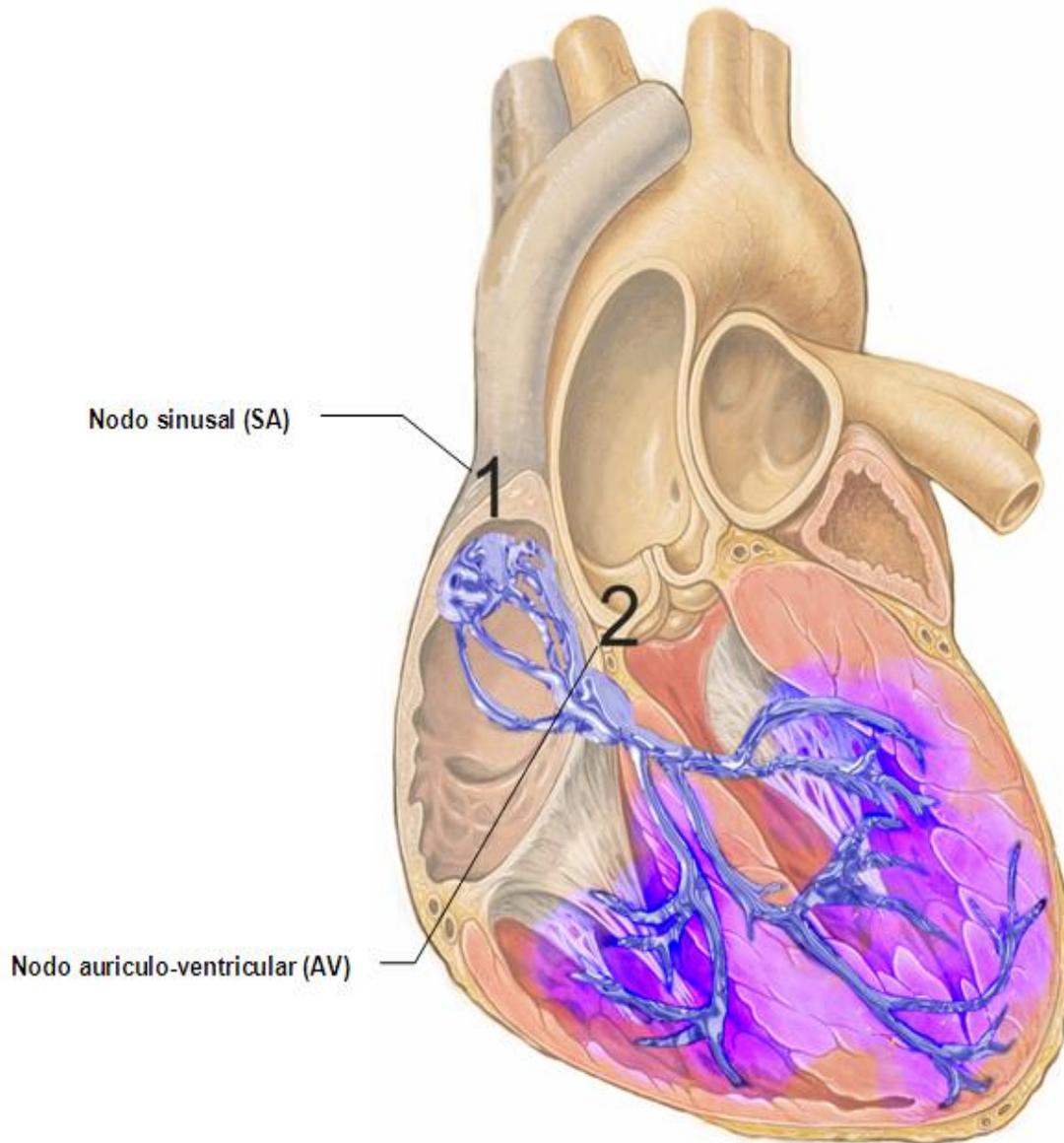


Figura 2.3.1. Nodo SA y AV

El potencial de acción (PA) o impulso eléctrico es una onda de descarga eléctrica que viaja a lo largo de la membrana celular. El PA se genera de un lado a otro de las membranas de las células del músculo del corazón por acumulación de iones positivos de sodio (Na^+) y de calcio (Ca^+) en el exterior de la célula, con una acumulación de iones de cloro en el mismo lado y por una acumulación de cargas negativas en el interior de la célula debida a ciertos aniones impermeables con una parte positiva proporcionada por el ión K^+ . En el siguiente tema se aborda con más detalle la parte eléctrica del corazón para conseguir un ECG.

2.4. ELECTROCARDIOGRAMA (ECG)

La electrocardiografía consiste en registrar gráficamente la actividad eléctrica que se genera en el corazón. Su análisis proporciona importante información que aporta datos sobre la función cardiaca, trastornos del ritmo y de la conducción, hipertrofia de cavidades y ayuda al diagnóstico de cardiopatías congénitas o adquiridas de tipo infeccioso, inflamatorio, metabólico o isquémico. Su normalidad no siempre descarta afectación cardiaca.

Un electrocardiograma (ECG) es un procedimiento de diagnóstico médico con el que se obtiene un registro gráfico de la actividad eléctrica del corazón en función del tiempo.

La actividad eléctrica son las variaciones de potencial eléctrico generadas por el conjunto de células cardiacas y que son recibidas en la superficie corporal. La formación del impulso cardiaco y su conducción generan corrientes eléctricas débiles que se diseminan por todo el cuerpo. Al colocar electrodos en diferentes sitios y conectarlos a un instrumento de registro como el electrocardiógrafo se obtiene el trazado característico de las ondas del corazón.

En condiciones de reposo una célula cardiaca tiene una carga negativa en su interior y positiva en el exterior, que se mantiene gracias a una permeabilidad selectiva de la membrana para los cationes extracelulares (Na y K) y a la actividad de la bomba de Na-K a nivel de los canales iónicos, esta diferencia de cargas a ambos lados de la membrana se denomina potencial de reposo.

Si el interior de la célula se hace menos negativo, llegando a un nivel crítico o umbral, aparece un cambio brusco en la permeabilidad, entrando masivamente cationes que invierten la carga a uno y otro lado de la membrana, generándose el llamado potencial de acción. Este cambio en la polaridad de la célula es lo que se conoce como despolarización. Este fenómeno activa a las células adyacentes, transmitiéndose por todo el corazón célula a célula y a través de sistemas especializados de conducción, ver figura 2.4.1.

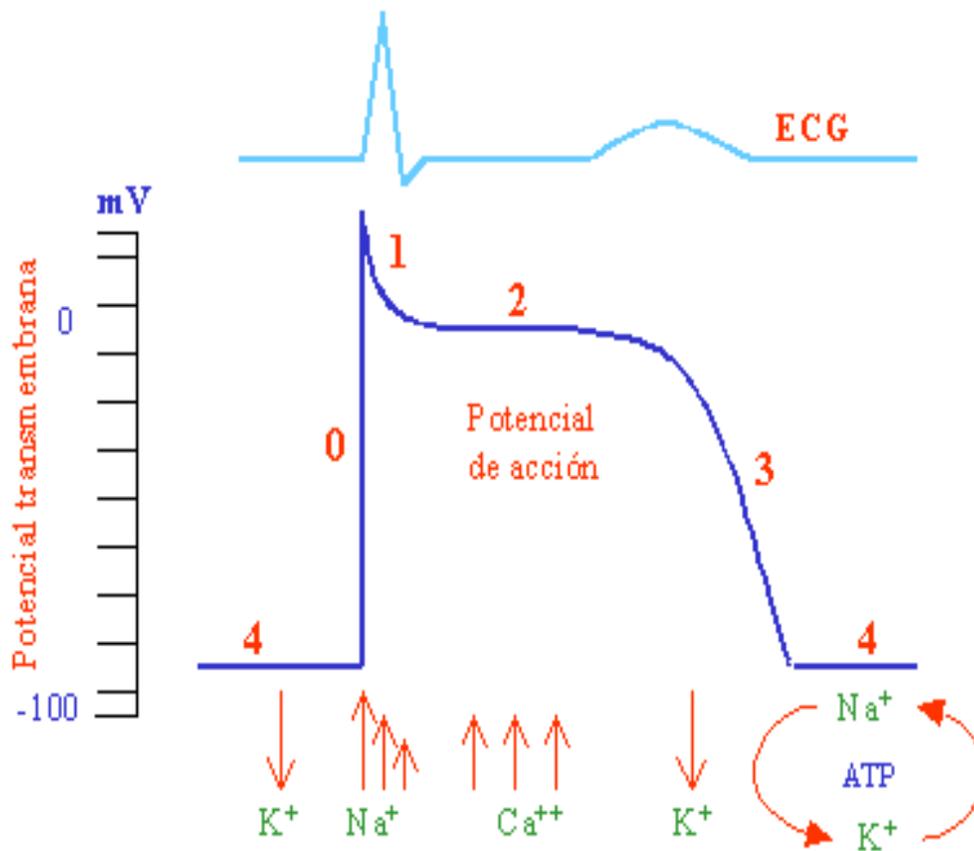


Figura 2.4.1. Gráfica del potencial de acción (PA) de la célula contráctil miocárdica ventricular

De la figura anterior, las flechas indican los tiempos de los principales movimientos de iones a través de la membrana celular.

La despolarización avanza por todo el corazón con un frente de onda positivo, mientras las células que quedan atrás se repolarizan, volviendo a su estado basal con carga negativa, como se ilustra en la gráfica del potencial de acción (PA) de la célula contráctil miocárdica ventricular y la sumatoria de todo el conjunto de células cardiacas que da el conocido trazo del ECG, ver figura 2.4.2.

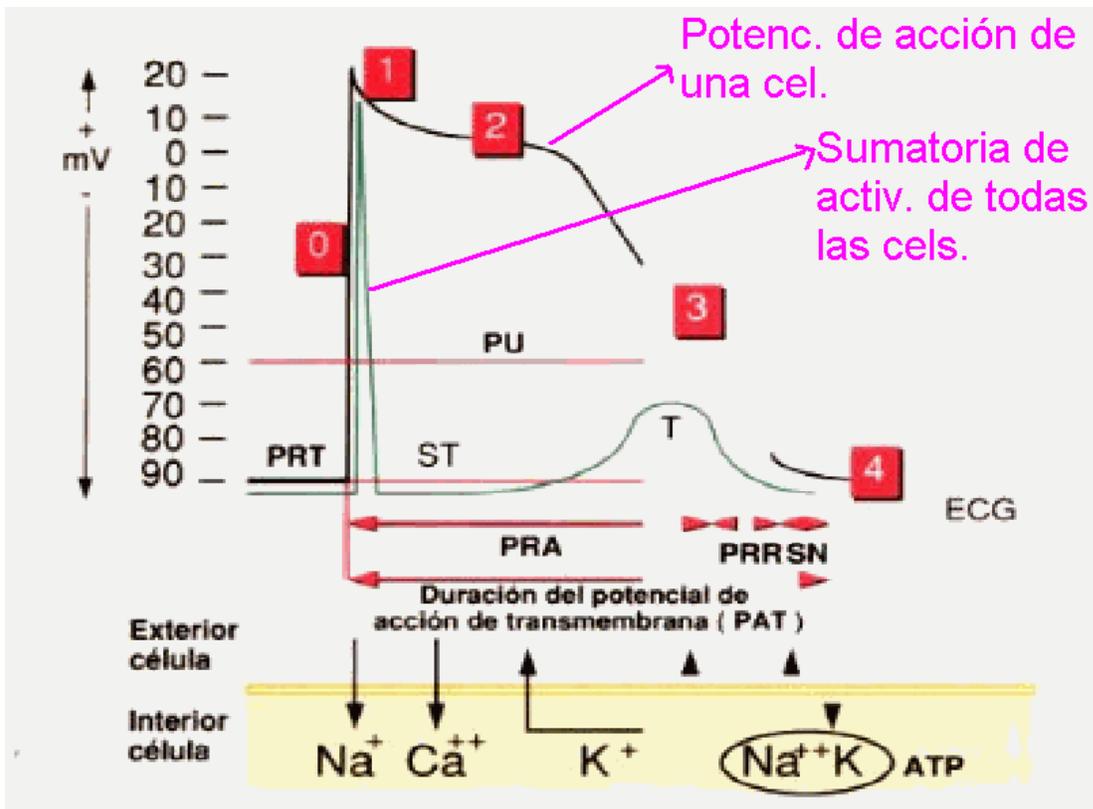


Figura 2.4.2. Gráfica del potencial de acción (PA)

La sumatoria de todas estas despolarizaciones sigue una dirección (vectores de despolarización), en función de las vías de conducción y la masa muscular del corazón, ver figuras 2.4.3 y 2.4.4.

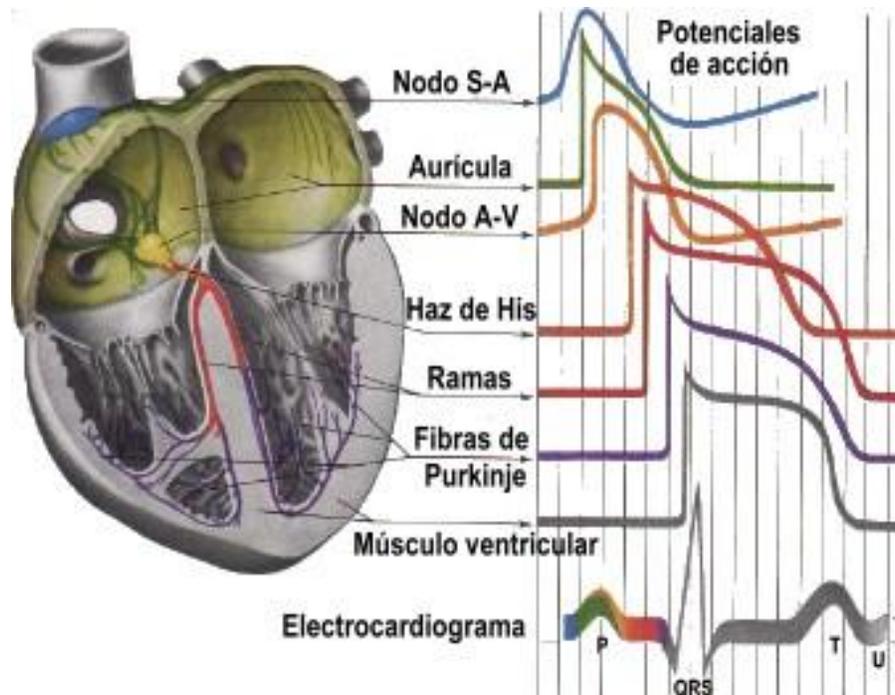


Figura 2.4.3. Actividad eléctrica del corazón en su recorrido vectorial, y sumatoria de estas señales

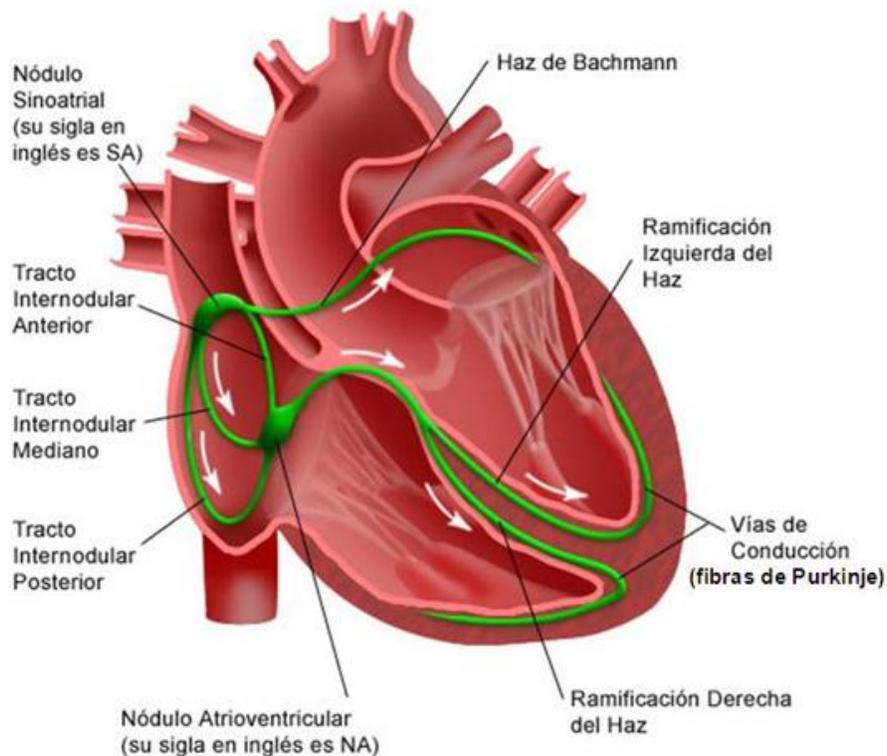


Figura 2.4.4. Sistema de conducción intrínseca del corazón (vectores de despolarización)

El registro de las señales cardiacas se fundamenta en la detección de la actividad eléctrica del corazón. Así pues, las contracciones rítmicas del corazón están controladas por una serie ordenada de descargas eléctricas, las cuales se explican mediante los siguientes principios electrofisiológicos

- Así pues, antes de cada contracción del músculo cardiaco se genera un impulso eléctrico por despolarización en el nodo o nódulo sinoauricular, este impulso eléctrico se propaga concéntricamente produciendo la onda P del electrocardiograma (ECG). Inicialmente se despolariza la aurícula derecha y posteriormente la aurícula izquierda. De este modo, en un ritmo normal, el nodo sinoauricular es el marcapasos cardiaco, es decir, es el lugar donde se origina el impulso cardiaco. El término despolarización, se refiere a la activación eléctrica del corazón que se produce por modificación de la polaridad de la membrana celular, al ingresar el sodio. La despolarización ventricular, por su parte, determina la onda QRS del ECG.
- La repolarización, es la recuperación de la polaridad. La repolarización auricular queda oculta o enmascarada en la curva QRS por lo que rara vez se observa una onda T auricular en el ECG. La repolarización ventricular se refleja en el segmento ST y en la onda T (una onda lenta).

Ambos fenómenos cardiacos, despolarización y repolarización, ocurren durante la sístole, es decir, durante la contracción del músculo cardiaco.

A continuación se presenta el trazo del ECG o EKG en la figura 2.4.5 y se explica cómo se generan las ondas:

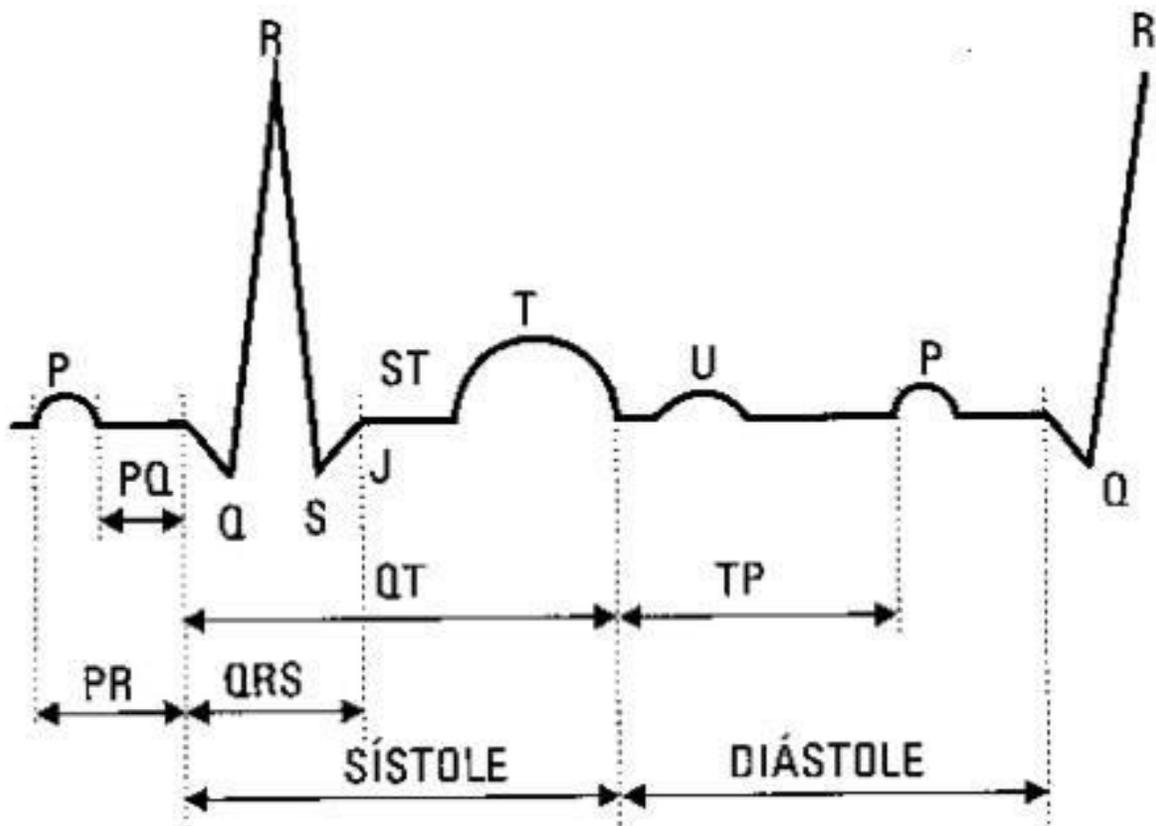


Figura 2.4.5. Generación del ECG

- La primera ondulación pequeña en la parte superior del trazado de un ECG se denomina "onda P", ésta indica que las aurículas (las dos cavidades superiores del corazón) son estimuladas en forma eléctrica (se despolarizan) para bombear la sangre hacia los ventrículos.
- En el trazado continua una sección corta descendente conectada con una sección alta ascendente. La misma se denomina "onda QRS o complejo QRS". Esta parte indica que los ventrículos (las dos cavidades inferiores del corazón) se están estimulando eléctricamente (despolarizando) para bombear la sangre hacia fuera.
- El segmento plano corto ascendente que sigue se llama "segmento ST", éste indica la cantidad de tiempo que transcurre desde el final de una contracción de los ventrículos hasta el comienzo del período de reposo (repolarización).

- La siguiente curva ascendente se denomina "onda T", ésta indica el período de recuperación o repolarización de los ventrículos.

Así pues, un **ciclo cardíaco** es representado por una sucesión de ondas en el trazado del ECG: la onda P, la onda o complejo QRS y la onda T. Estas ondas producen dos intervalos importantes: el intervalo PR y el intervalo QT, y dos segmentos: el segmento PQ y el segmento ST. Por ejemplo, durante el ejercicio físico el ciclo cardíaco se acelera y por tanto el intervalo P-T se acorta.

El registro del ECG se realiza en un papel milimetrado, con líneas más gruesas cada 5 milímetros. Cada milímetro de trazado horizontal es equivalente a 0.04 segundos del ciclo cardíaco, el espacio entre dos líneas gruesas representa 0.2 segundos, a la velocidad habitual del papel de 25 [mm/seg].

En sentido vertical, en el eje de las ordenadas, se representa el voltaje de las ondas electrocardiográficas, en [mV], las cuales pueden ser positivas o negativas según la línea isoeleétrica. El eje de las abscisas o sentido horizontal representa el tiempo.

Resulta interesante destacar que un trazado electrocardiográfico no es más que un gráfico de voltaje en función del tiempo. De esta manera, en él se reflejan ondas que tienen una determinada amplitud según la situación y una determinada duración. Entre las ondas muestra pausas que también tienen una determinada duración.

Es importante destacar las particularidades de los tejidos automáticos (como son los tejidos del músculo cardíaco), que son capaces de realizar despolarización espontánea, generando los estímulos eléctricos que derivan en la despolarización sincronizada del corazón.

Para que el corazón pueda realizar la contracción se necesita una estimulación y conducción eléctrica a través de fibras especiales para transmitir impulsos eléctricos. El latido del corazón es totalmente involuntario, por lo tanto el cuerpo no tiene control sobre sus latidos.

La actividad eléctrica del corazón se origina en el nodo o nódulo sinusal, de ahí pasa a las aurículas, luego llega al nódulo auriculoventricular, durante este proceso se disemina en un haz de fibras y acaba dividiéndose en cuatro ramas, como se observa en las figuras 2.4.3 y 2.4.4. Resumiendo las características de un ECG:

- El ECG es el registro de la actividad eléctrica del corazón.
- Se cuantifica la actividad eléctrica en un papel milimetrado.
- **Dicha cuantificación en la abscisa (X) es tiempo y en la ordenada (Y) es voltaje en mV.**
- La actividad eléctrica se representa morfológicamente como P, QRS, T.
- La despolarización y la repolarización son fenómenos sistólicos.
- **La onda P** coincide con la despolarización auricular.
- **La onda o complejo QRS** precede a la contracción ventricular.
- Durante el ejercicio el intervalo P-T se acorta.
- **La onda T** resulta por la repolarización ventricular.

Un electrocardiograma se obtiene con un electrocardiógrafo en forma de cinta continua colocando los pequeños electrodos de éste aparato en determinados puntos del cuerpo del paciente, a continuación se pone en funcionamiento y se realiza el registro en unos pocos minutos. La prueba es completamente indolora y no tiene ningún riesgo. En la figura 2.4.6 se aprecian los componentes gráficos de un ECG:

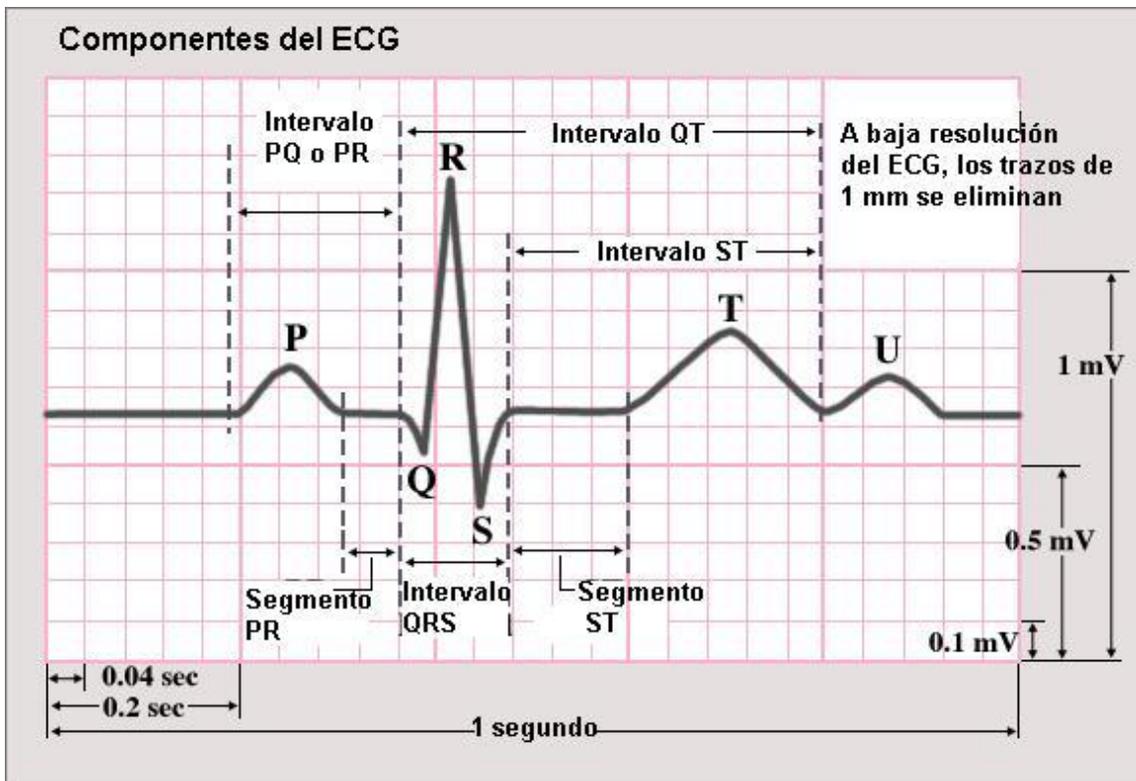


Figura 2.4.6. Componentes gráficos del ECG

El trazo de un ECG permite conocer las condiciones cardiovasculares del individuo. Así, para detectar cualquier irregularidad en la actividad eléctrica del corazón, un ECG es extremadamente útil para el diagnóstico y control de diversas afecciones, por ejemplo:

- arritmias,
- angina de pecho,
- infarto,
- alteraciones metabólicas,
- predisposición a una muerte súbita cardíaca,
- etcétera.

También es útil para determinar la duración del ciclo cardíaco y tiene la ventaja de ser un procedimiento médico no invasivo con resultados disponibles inmediatamente, además de ser un estudio económico. En la figura 2.4.7 se ejemplifica la captura de un ECG:

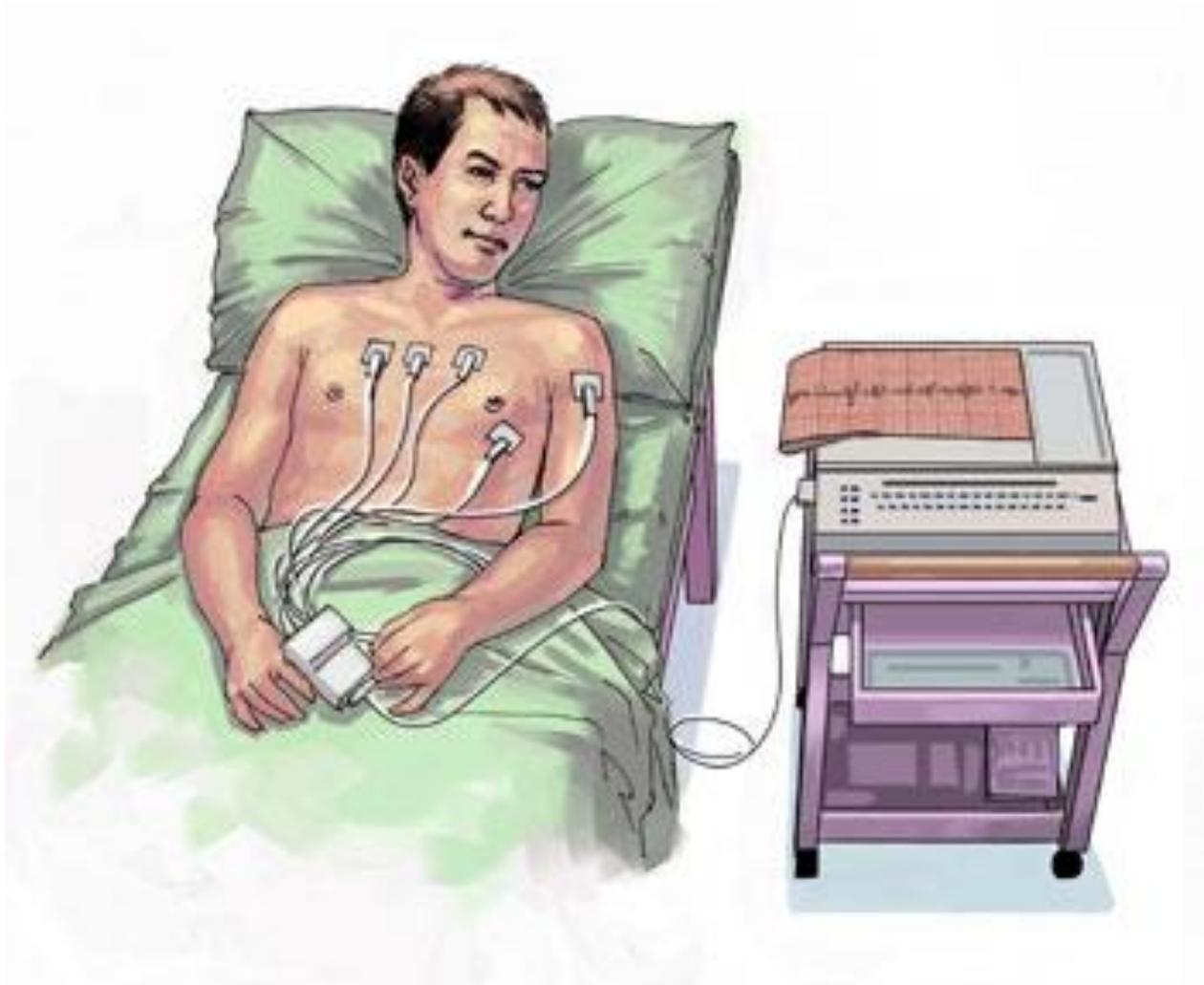


Figura 2.4.7. Captura de un ECG

El ECG debe ser siempre interpretado en conjunto con los hallazgos clínicos y observaciones de un médico. Es un examen que aislado no es diagnóstico de enfermedad cardíaca ni tampoco la puede excluir del todo. Las variaciones de potencial eléctrico durante el ciclo cardíaco producen las ondas características del ECG. Al colocar electrodos en diferentes sitios y conectarlos a un instrumento de registro como el electrocardiógrafo, se obtiene el trazado característico.

Las conexiones de entrada al aparato deben ser realizadas de tal forma que al obtener el gráfico, una deflexión hacia arriba indique un potencial positivo y una hacia abajo uno negativo, teniendo una línea base, ver figura 2.4.8.

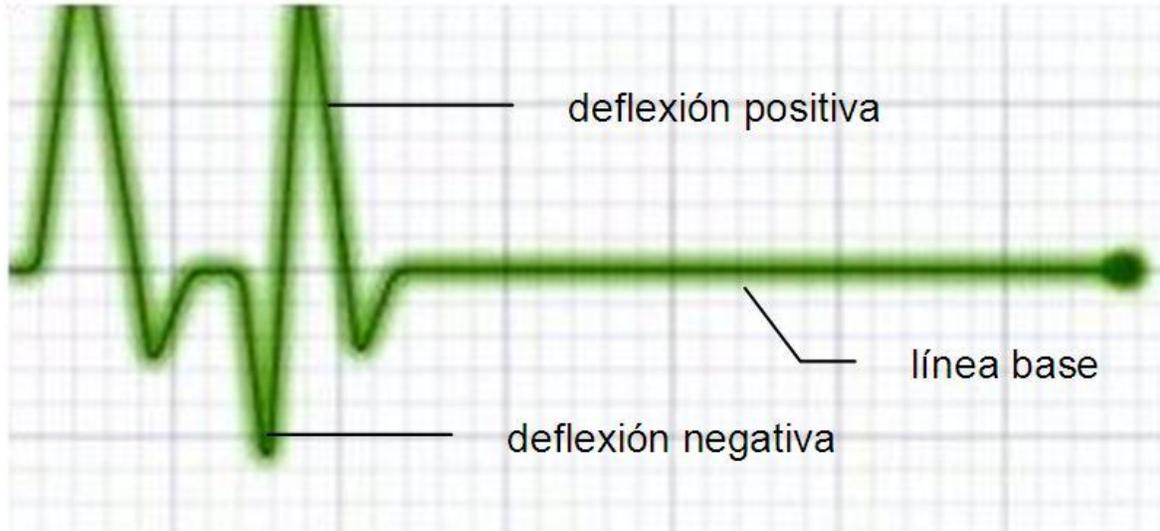


Figura 2.4.8. Deflexiones positiva y negativa en el ECG

Se han adoptado normas internacionales con respecto a la velocidad del papel, 25 [mm/seg], la amplitud de calibración $1 \text{ [mV]} = 1 \text{ [cm]}$, y los sitios de la colocación de los electrodos cutáneos. A las disposiciones específicas de los electrodos, se conocen como derivaciones y en la práctica clínica se utilizan un número estándar de doce, clasificadas en: bipolares, unipolares y precordiales. Las derivaciones bipolares o de Einthoven se observan en la figura 2.4.9; las derivaciones unipolares o de Goldberger aparecen en la figura 2.4.10; y las derivaciones precordiales se ilustran en la figura 2.4.11.

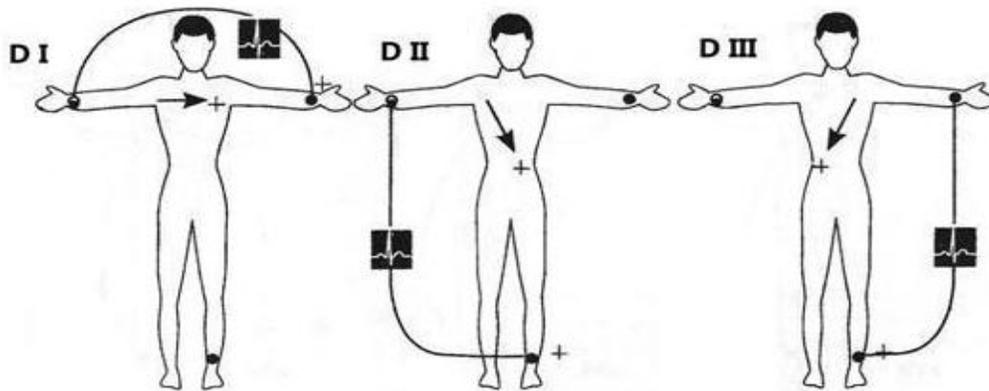


Figura 2.4.9. Derivaciones Bipolares (Einthoven)

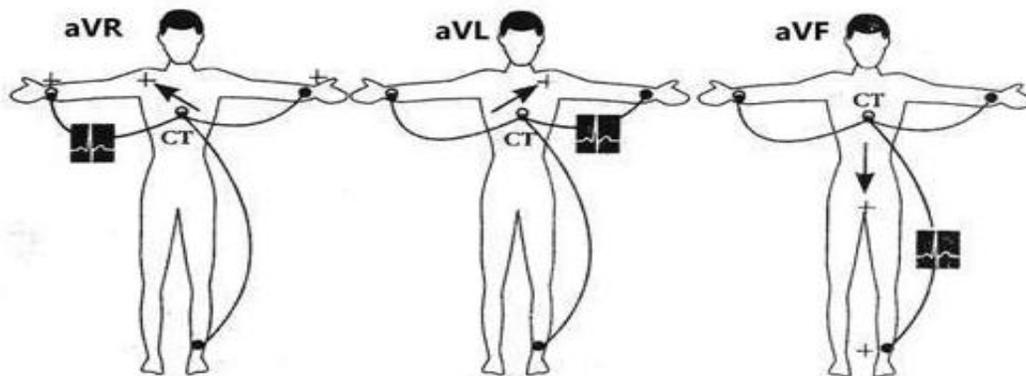


Figura 2.4.10. Derivaciones Unipolares (Goldberger)

En el caso de las derivaciones de las figuras 2.4.9 y 2.4.10 no se coloca el electrodo positivo en un miembro y el negativo en otro como en el caso anterior, sino que se coloca el electrodo positivo en uno de los miembros y se compara contra la sumatoria de los otros miembros conectados al polo negativo.

La ubicación en el cuerpo del paciente de las derivaciones precordiales es la siguiente:

- V1: 4º espacio intercostal con línea o borde paraesternal derecha.
- V2: 4º espacio intercostal con línea o borde paraesternal izquierda.
- V3: Punto equidistante entre V2 y V4.
- V4: 5º espacio intercostal izquierdo con línea medioclavicular izquierda.

- V5: 5º espacio intercostal con línea axilar anterior izquierda.
- V6: 5º espacio intercostal con línea axilar media izquierda.

Incluso en algunos casos, se llega hacer uso de una séptima colocación (V7). Ver figura 2.4.11.

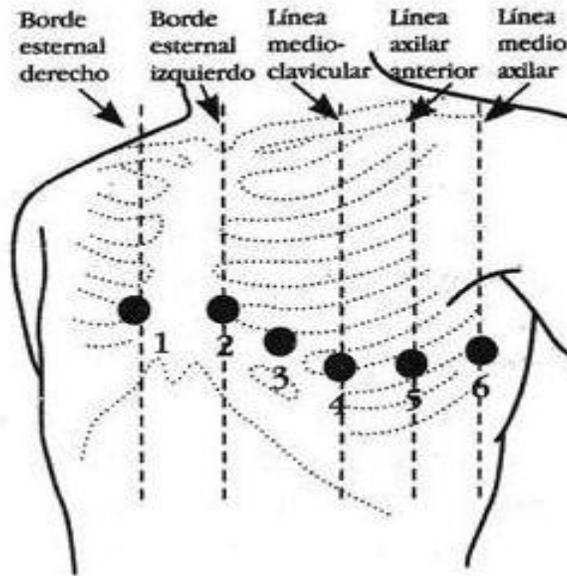


Figura 2.4.11. Derivaciones Precordiales

La nomenclatura empleada para la designación de las ondas electrocardiográficas, éstas se denominan en letras mayúsculas cuando tienen una amplitud mayor a 5 [mm], y en letras minúsculas cuando la amplitud es menor de 5 [mm], tal y como se observa en la figura 2.4.12.

Las posiciones de los electrodos en el cuerpo del paciente no registran solamente el potencial eléctrico de la pequeña área del miocardio subyacente, sino que registra los eventos eléctricos del ciclo cardiaco desde un sitio seleccionado.

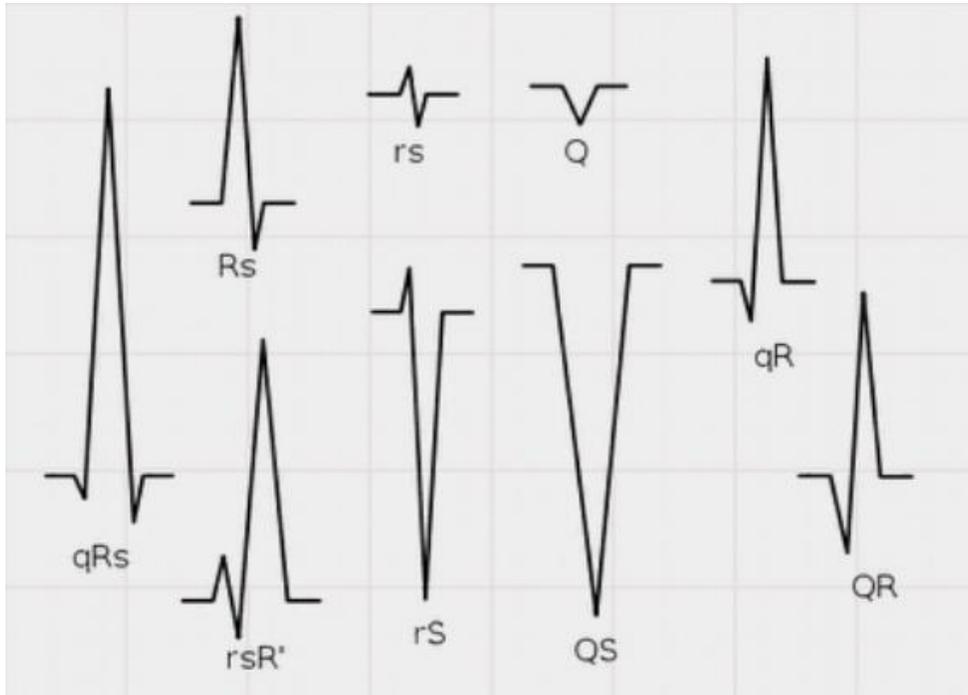


Figura 2.4.12. Nomenclatura electrocardiográfica

Para el estudio anatómico y biomecánico del cuerpo humano se hace uso de ejes y planos geométricos, ver figura 2.4.13.

A partir de estos ejes y planos anatómicos es que el holandés Willem Einthoven estableció las derivaciones, además de varias observaciones sobre el aspecto vectorial de la actividad eléctrica cardíaca.

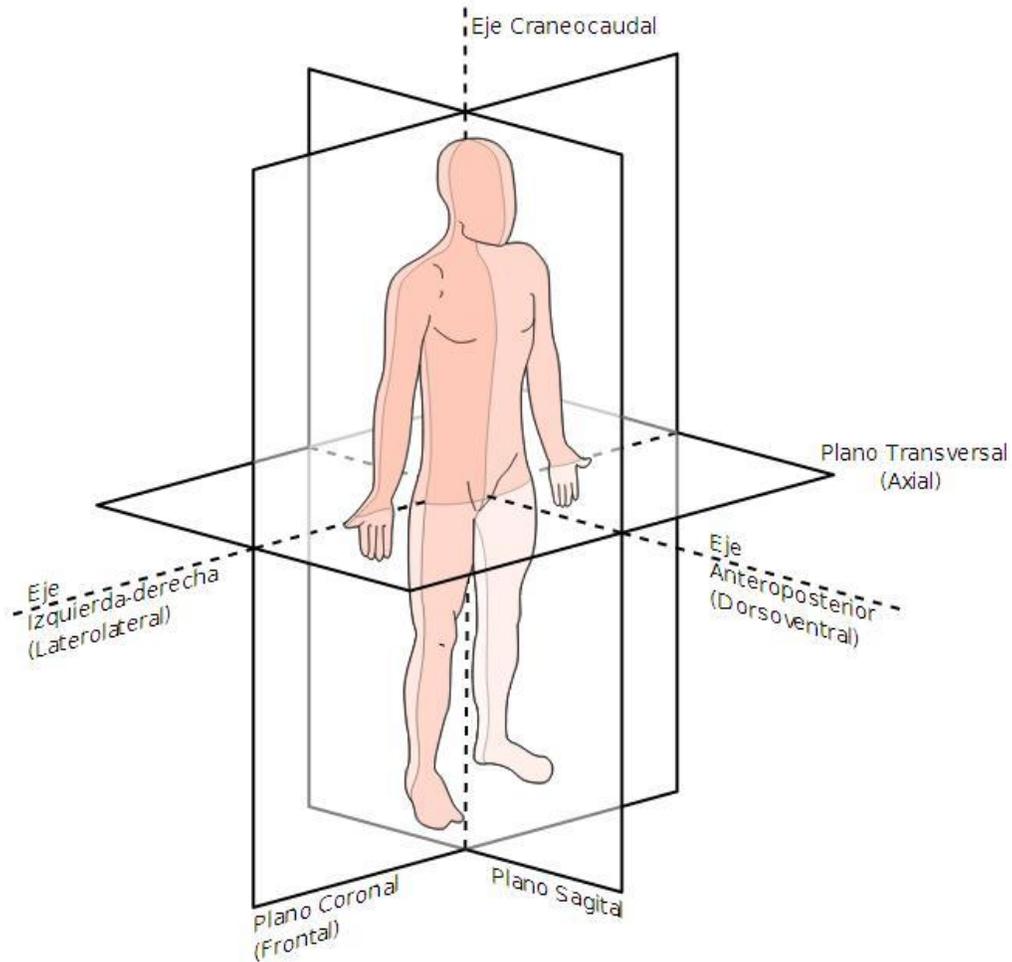


Figura 2.4.13. Ejes y planos anatómicos

Así, respecto al plano frontal estableció las derivaciones del plano frontal, derivaciones bipolares estándar, y las derivaciones amplificadas del plano frontal o monopolares.

Las derivaciones bipolares estándar (DI, DII, DIII) son las que originalmente eligió Einthoven para registrar los potenciales eléctricos en el plano frontal. Los electrodos son aplicados en los brazos derecho e izquierdo y en la pierna izquierda. Se coloca un electrodo en la pierna derecha que sirve como polo a tierra. Las derivaciones bipolares, registran las diferencias de potencial eléctrico entre los dos electrodos seleccionados.

La colocación para estas derivaciones es la siguiente:

- DI: Brazo izquierdo (+), Brazo derecho (-)
- DII: Pierna izquierda (+), Brazo derecho (-)
- DIII: Pierna izquierda (+), Brazo izquierdo (-)

El potencial eléctrico registrado en una extremidad, a más de doce centímetros del corazón, es el mismo sin importar el sitio en donde se coloque el electrodo sobre ella.

La ley del infinito eléctrico establece que a más de doce centímetros del corazón se puede obtener la misma medida del potencial ejercido por éste. Así, es lo mismo colocar los electrodos en el tronco del paciente o en sus extremidades (muñecas o tobillos), o por ejemplo si una extremidad ha sido amputada se puede colocar el electrodo en su porción más distal del corazón.

Las derivaciones unipolares fueron creadas por Frank Wilson en 1934 para medir la fuerza eléctrica absoluta de un electrodo positivo; la nomenclatura que se estableció para estas derivaciones unipolares fue con las letras V, R, L, y F (vector, right, left, foot), es decir, derivaciones VR, VL y VF (vector right, vector left, vector foot). Estas derivaciones fueron modificadas posteriormente por Goldberger (aVR, aVL y aVF) estableciendo así la suma algebraica de éstas igual a cero, cumpliendo así Leyes de Kirchhoff, como se aprecia en la figura 2.4.14. Al ser amplificadas dejan de ser unipolares quedando finalmente como bipolares.

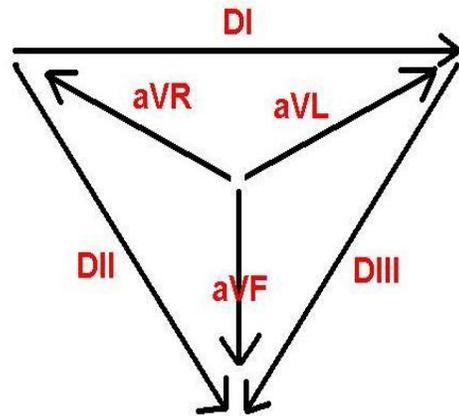
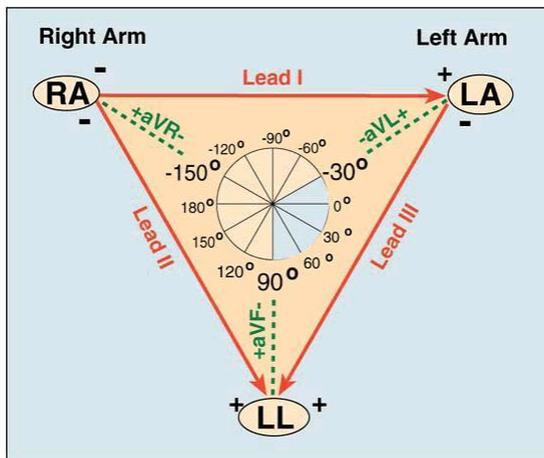


Figura 2.4.14. Triángulo de Einthoven

Finalmente, estas derivaciones amplificadas son las tres del plano frontal, que en los inicios de la electrografía eran monopares (VR, VL y VF), pero fueron modificadas para amplificarlas en el registro, convirtiéndose en bipolares amplificadas (**aVR**, **aVL** y **aVF**) la letra “a” del inglés augmented indica que la amplitud ha sido aumentada $\pm 50\%$ para facilitar su lectura (**augmented vector right** , **augmented vector left** , **augmented vector foot**).

En las derivaciones amplificadas del plano frontal, los electrodos se colocan de la siguiente forma:

- aVR: brazo derecho (+), brazo izquierdo (+), y pierna Izquierda (-)
- aVL: brazo izquierdo (+), brazo derecho (+), y pierna Izquierda (-)
- aVF: pierna izquierda (+), brazo derecho (+), y brazo izquierdo (-)

En la literatura sobre cardiología, esta clasificación puede prestarse a confusiones, pues la nomenclatura de las tres últimas derivaciones (aVR, aVL y aVF) se siguen denominando monopares de los miembros, para diferenciarlas de las bipolares estándar (I, II, III) siendo realmente bipolares.

Otro dato importante a mencionar es que a las derivaciones bipolares (DI, DII, DIII) y a las derivaciones “unipolares” (aVR, aVL y aVF), también se les denominan como derivaciones periféricas, o como derivaciones de extremidades o miembros.

Las derivaciones con respecto al plano anatómico horizontal (transversal o axial), son derivaciones verdaderamente monopolares o unipolares, pues comparan la actividad del punto en que se coloca el electrodo a nivel precordial o torácico (electrodo explorador) contra la suma de los tres miembros activos o Central Terminal (PI + BI + BD, que da como resultado 0). La ubicación precordial de los electrodos es la que da su nombre a este tipo de derivaciones.

Los electrodos se ubican en el cuerpo del paciente respecto a los ejes y planos anatómicos del cuerpo humano y en consecuencia de ahí el nombre de las derivaciones, pero falta mencionar los ejes y planos anatómicos del corazón y, su eje eléctrico que da la configuración eléctrica vectorial del triángulo de Einthoven.

La configuración anatómica del corazón y su nomenclatura, ver figura 2.4.15, es la siguiente:

- Plano A: plano del eje longitudinal o eje mayor del corazón.
- Plano B: plano coronal o plano de las 4 cámaras del corazón.
- Plano C: plano del eje menor del corazón.

Respecto a esta configuración, el eje mayor está orientado de arriba hacia abajo, de derecha a izquierda y de atrás hacia adelante y forma un ángulo de unos 40° con respecto al plano horizontal anatómico del cuerpo humano.

Habitualmente, el corazón se encuentra desplazado con respecto al plano sagital medio del cuerpo humano, de modo que dos terceras partes se encuentran a la izquierda y una tercera parte está a la derecha de este plano.

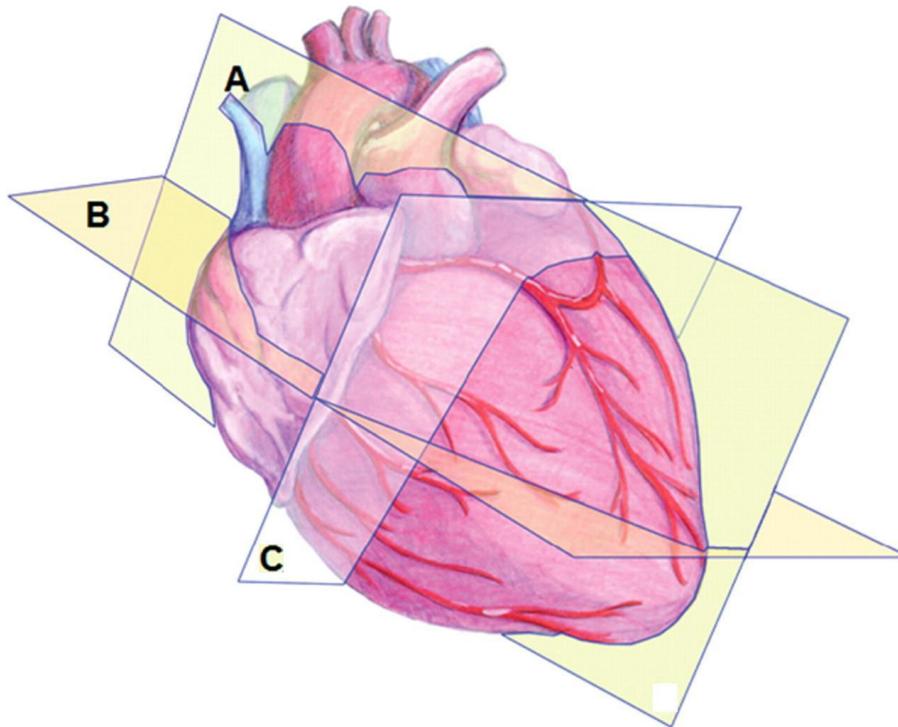


Figura 2.4.15. Ejes y planos geométricos del corazón en su posición anatómica

El corazón está en posición anatómica, es decir, su eje longitudinal no se dirige hacia abajo verticalmente sino de forma oblicua hacia abajo, a la izquierda y hacia adelante (ver figura 2.4.16).

Debido a la masa muscular de los ventrículos, los eventos eléctricos que originan durante su despolarización son de gran amplitud, por lo que prácticamente dominan sobre el resto. Por ello, si se representa el vector medio de todos los vectores generados en cada instante de despolarización ventricular, se obtiene el vector eléctrico medio o eje eléctrico medio cardiaco, que suele coincidir con el eje longitudinal anatómico del corazón, porque es el vector dominante dada la gran masa ventricular que se despolariza.

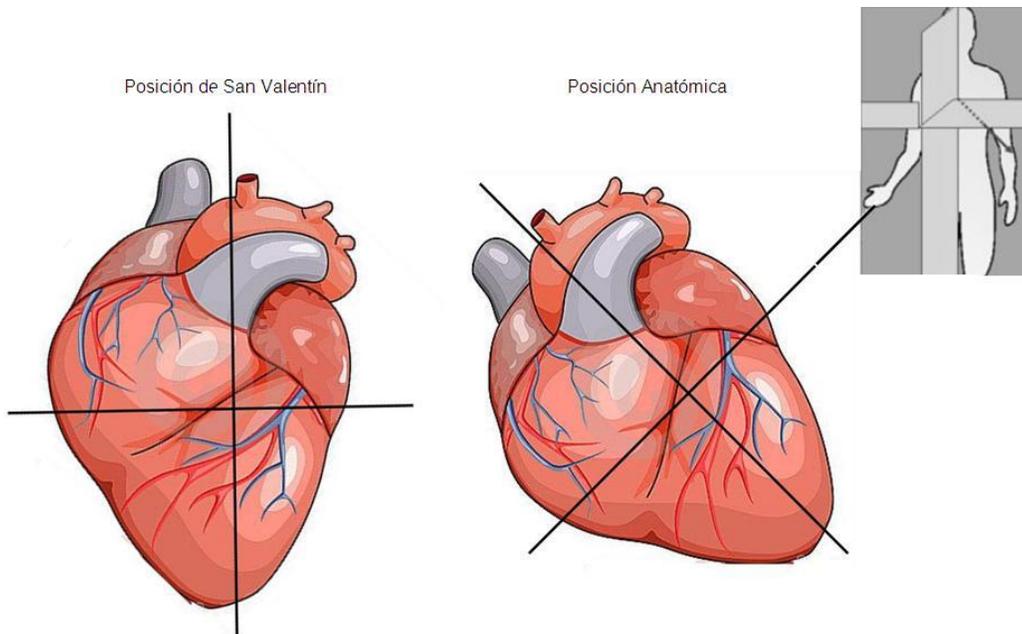


Figura 2.4.16. Posición anatómica del corazón

Este eje eléctrico medio es el resultante de la suma de tres vectores eléctricos medios que representan tres momentos importantes de la despolarización ventricular:

- El vector 1 que representa la despolarización de la masa septal del tabique interventricular y ventrículo derecho, de dirección hacia adelante y hacia la derecha y origen de la deflexión Q del ECG estándar.
- El segundo vector importante, el 2, se debe a la despolarización de la masa ventricular que origina la deflexión R.
- El vector 3 se debe a la activación de las porciones basales del corazón que se dirige hacia la derecha y hacia arriba y origina la deflexión S.

Considerando que todos ellos se originan en un punto común que sería el centro eléctrico del corazón, la suma de estos tres vectores medios dará el vector eléctrico medio de cada fase, cuya amplitud será igual a la amplitud de la onda correspondiente QRS en el ECG.

El eje QRS es el más importante y también el que se explora con mayor frecuencia en la práctica médica. Para calcular el eje de QRS debemos usar el sistema de referencia hexaxial (sistema de 6 ejes). Es el resultado de la combinación de los ejes (vectores) de las tres derivaciones clásicas de los miembros (DI, DII y DIII) y las 3 derivaciones aumentadas (aVR, aVL y aVF) del plano frontal. Sirve principalmente para la determinación de eje QRS en el plano frontal anatómico del cuerpo humano.

Las seis derivaciones se disponen como "radios de una rueda", y ésta queda dividida en 2 mitades (superior e inferior) por el eje (vector) de la derivación I. A los polos de cada derivación (una de las características de los vectores es la polaridad) en la mitad superior de la "rueda" se les asigna grados negativos (ejemplo: -60° , -120°) y los de la mitad inferior se les asigna grados positivos (ejemplo: $+30^\circ$, $+90^\circ$). No se debe confundir los grados positivos y negativos con los polos positivos y negativos de cada derivación (ver figuras 2.4.17. y 2.4.18).

La figura hexaxial se divide en 4 cuadrantes al cortar por la mitad los ejes de las derivaciones DI y aVF, las denominaciones de los cuadrantes son:

- I: de 0° a -90°
- II: de 0° a $+90^\circ$
- III: $+90^\circ \pm 180^\circ$
- IV: $-90^\circ \pm 180^\circ$

En la tabla 2.4.1 se resume la posición de las derivaciones, así como su polaridad:

Derivaciones estándar	Polo -	Polo +
DI	$\pm 180^\circ$	0°
DII	-120°	$+60^\circ$
DIII	-60°	$+120^\circ$
Derivaciones aumentadas	Polo -	Polo +
aVR	$+30^\circ$	-150°
aVF	$+150^\circ$	-30°
aVL	-90°	$+90^\circ$

Tabla 2.4.1. Posición y polaridad de las derivaciones

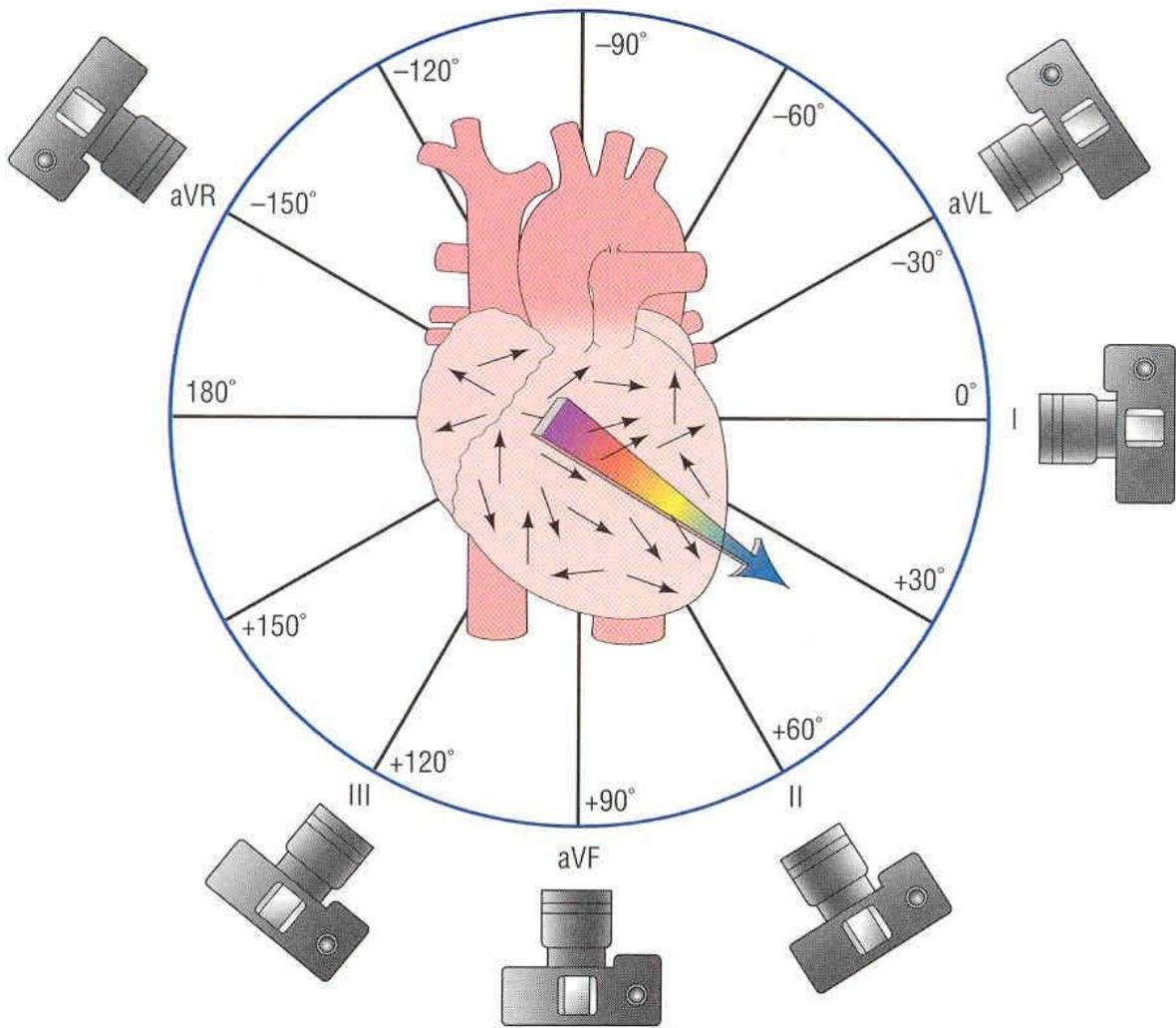


Figura 2.4.17. Vectores eléctricos del corazón

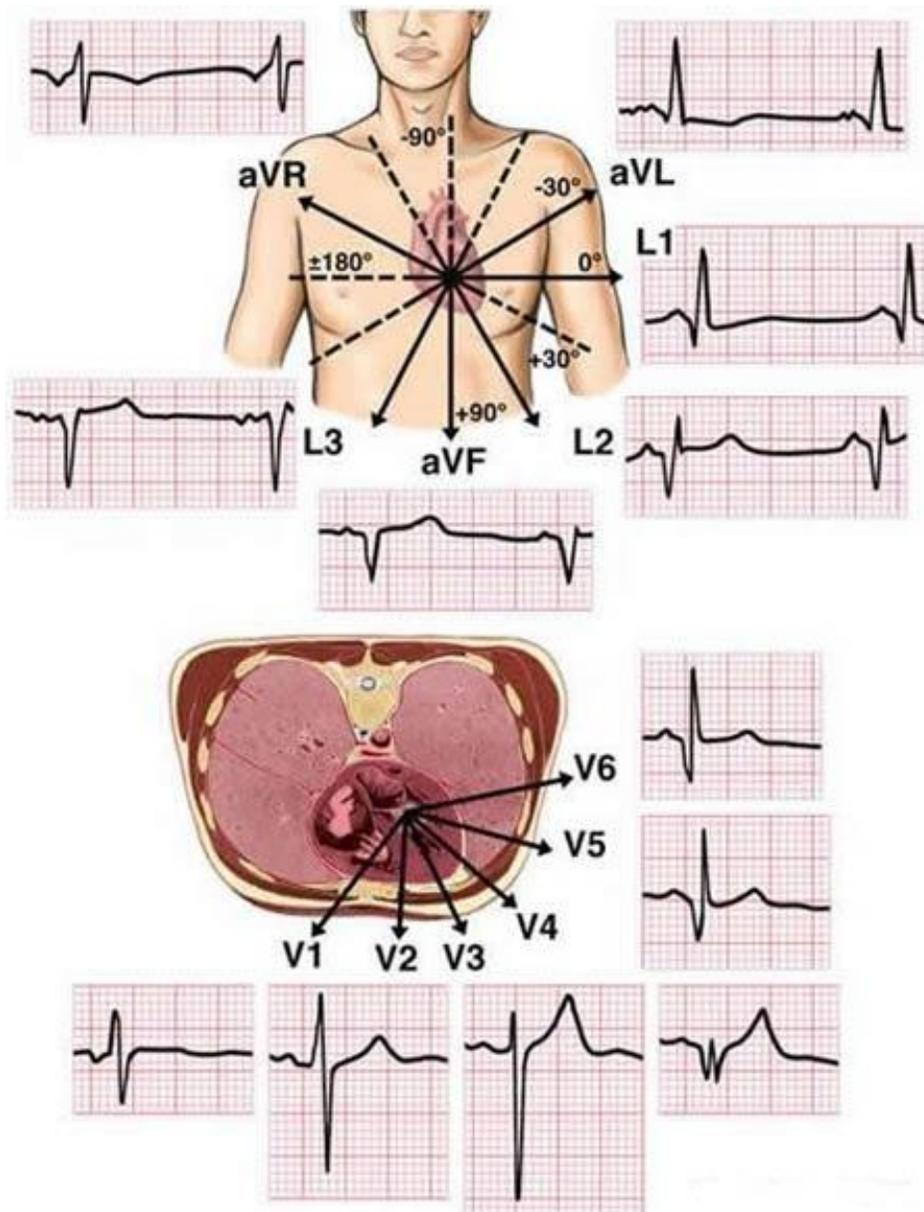


Figura 2.4.18. Vectores en los planos frontal y horizontal

Todo lo expuesto a lo largo de este capítulo sienta las bases para el propósito y desarrollo del presente trabajo, la propuesta de calibración de un electrocardiógrafo para tener la certeza de que el instrumento funciona correctamente antes de ser empleado y no considerar que su operación es adecuada con base en la experiencia observacional y estadística, es decir, a juicio y experiencia del operador del mismo. [L06]

3.1. INTRODUCCIÓN

El siguiente capítulo aborda los componentes que integran al Sistema Simulador, ejemplo de ello es la resistencia, el capacitor, el diodo, el transistor, entre otros.

Estos componentes al unirse con integrados más avanzados pueden ayudar a crear señales, impulsos o acciones, de múltiples formas para poder analizar o controlar sistemas completos y hacer tareas que en ocasiones ni nos imaginamos.

Al paso del tiempo, desde que el ser humano ha tratado de buscar que sus tareas cotidianas sean más sencillas, ha evolucionado e innovado, en gran medida, todos los componentes electrónicos. Resultado de ello son algunas tareas que se pueden hacer con dos o tres de estos componentes y que ocupan mucho menos espacio ya que su diseño se ha hecho más compacto, ejemplo de ello son los equipos celulares utilizados actualmente. [L07]

El inicio de estos cambios comienza con el descubrimiento del diodo en 1873 por Frederick Guthrie, considerado el componente básico para el desarrollo de circuitos más complejos, como lo son los transistores, los circuitos integrados, los microcontroladores y microprocesadores.

Resistencias

Se sabe que la resistencia es una particularidad de cualquier material conductor, que al hacerle pasar un flujo eléctrico y encuentra una fuerza opuesta, muy similar a la fricción mecánica. Por lo que es muy común encontrarla representada en circuitos como se muestra en la figura 3.1.1.

Se puede conocer la resistencia de cualquier material conductor, tomando en cuenta las siguientes características: tipo de Material, longitud, área transversal y temperatura. Lo que lleva a determinar que entre más longitud tenga el conductor mayor es la resistencia y entre mayor sea el área transversal menor será esta.

En las resistencias existe un tipo de ruido llamado ruido Johnson o ruido térmico ocasionado por la agitación térmica de los electrones y que en señales de muy pequeña amplitud llega a ocasionar una distorsión modificando los valores o comportamiento de la señal que se desea obtener.

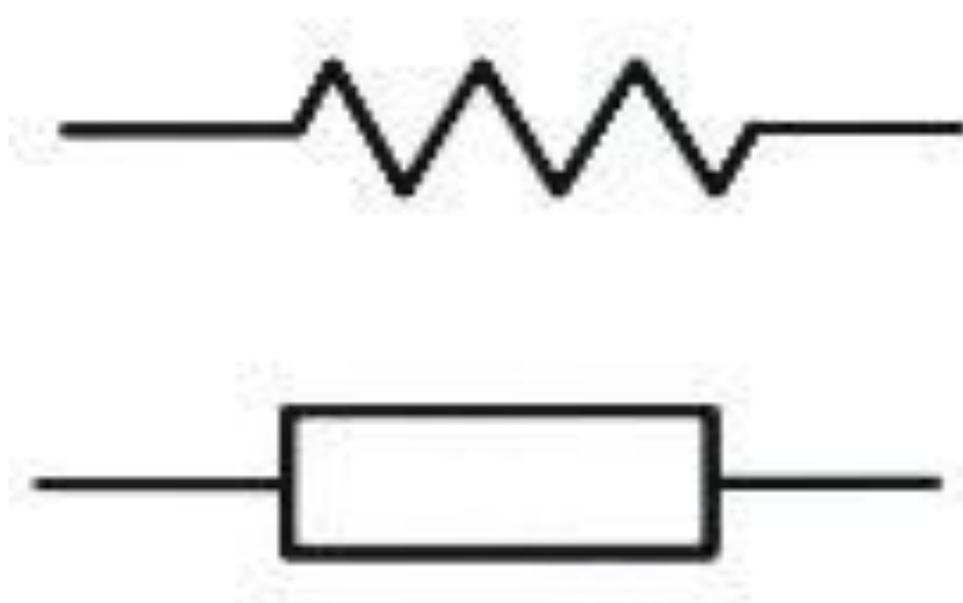


Figura 3.1.1. Símbolo de la resistencia

Capacitor

Se conoce que es un componente pasivo que almacena energía eléctrica, la cual puede ser usada cuando el circuito lo requiera, su forma más sencilla como se muestra en la figura 3.1.2 se conforma por dos placas metálicas paralelas separadas por un material dieléctrico.

En ocasiones se ocupa en los circuitos para que el capacitor sea el que de la tensión que necesita algún componente, como por ejemplo el flash de una cámara.

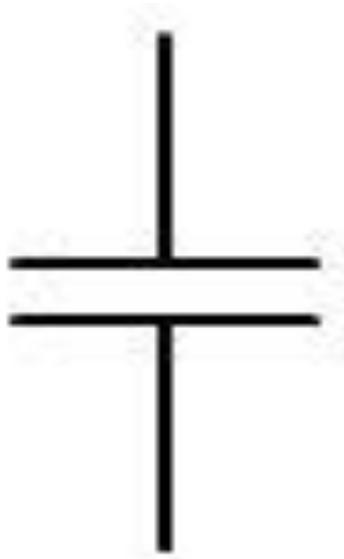


Figura 3.1.2. Representación del capacitor

Inductor

Es un alambre enrollado circularmente formando una bobina, la cual ocasiona que al pasar una corriente eléctrica forme un campo magnético y provoca un cambio en el flujo magnético hasta que la corriente se estabiliza. Los inductores son bobinas diseñadas para introducir inductancia en los circuitos la cual varía dependiendo de las propiedades magnéticas, por lo que se llegan a utilizar con núcleos ferromagnéticos para incrementar la inductancia se representa con una L y su representación grafica se muestra en la figura 3.1.3. [L15]



Figura 3.1.3. Representación del inductor

El diodo

Es un componente electrónico que cuenta con dos terminales, por las cuales circula una corriente eléctrica en un solo sentido como se observa en la figura 3.1.4. Los primeros diodos fueron unos tubos de vacío, que eran dos electrodos rodeados de vacío dentro de un tubo de cristal.

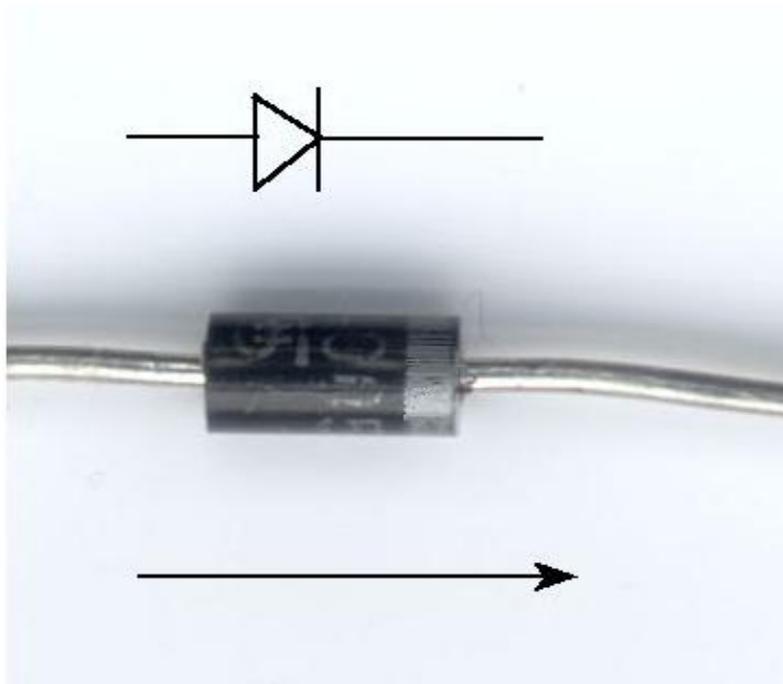


Figura 3.1.4. Diagrama, imagen real y circulación de corriente del diodo

El diodo está hecho de un material cuya conductividad es menor que la de un conductor y mayor que la de un aislante. La conductividad está muy relacionada con el número de electrones libres que este contenga. Gracias a esta característica el diodo tiene múltiples aplicaciones, por lo que se han creado diferentes tipos de diodos que funcionan de manera muy particular; dependiendo de su aspecto físico, impurezas, uso de electrodos y que cuentan con características eléctricas especiales. Algunos diodos liberan energía ya sea en forma de calor y en radiación ultravioleta o infrarroja, y como ejemplo se puede mencionar los diodos que se utilizan en los controles remotos. También existen diodos láser que se utilizan como fuentes de alimentación lumínicas para sistemas de comunicaciones, que a su vez se emplean en las fibras ópticas, otra aplicación son los diodos lectores de CD-ROM, estos diodos detectan luz. [L12]

Transistores

El transistor es un dispositivo electrónico semiconductor de tres terminales que puede ser utilizado para propósitos de amplificación y conmutación. Se le llama transistor **BJT (Bipolar Junction Transistor, transistor de unión bipolar)** por el hecho de que los huecos y los electrones participan en el proceso de inyección hacia el material polarizado de forma opuesta. Ver figura 3.1.5.

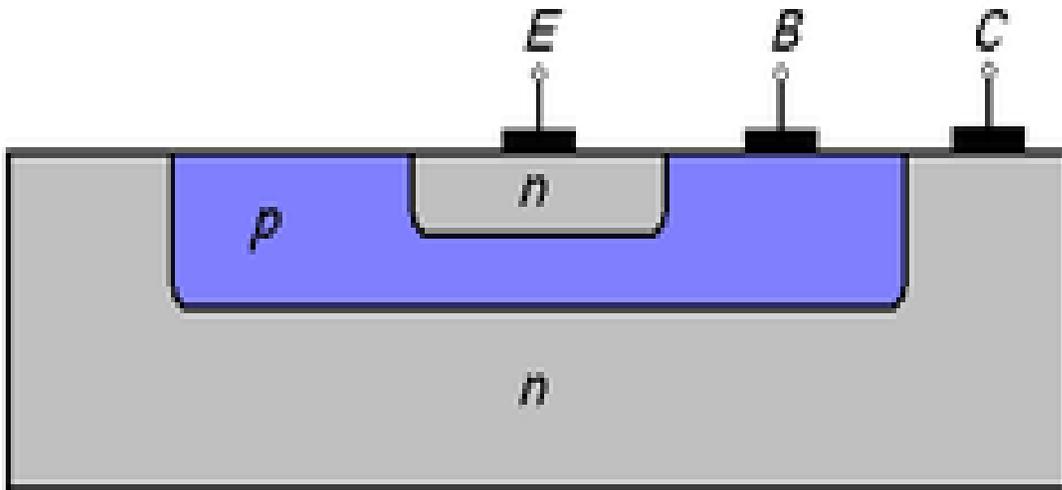


Figura 3.1.5. Corte transversal simplificado de un transistor de unión bipolar NPN

El transistor de la figura 3.1.5 es un dispositivo semiconductor de tres capas donde se puede apreciar como la unión base-colector es mucho más amplia que la base-emisor que consiste de dos capas de material tipo n y una capa tipo p; al que se le llama transistor npn, sus terminales se indican mediante las letras E para el emisor, C para el colector y B para la base; también existen de dos capas de material tipo p y una tipo n, se le llama transistor pnp.

Las principales características y ventajas son:

- El consumo de energía es bajo.
- Tamaño reducido.
- El peso es mínimo.
- Una vida larga útil (muchas horas de servicio).
- Puede permanecer mucho tiempo almacenado y no pierde propiedades.
- No necesita tiempo de calentamiento.

- Tiene resistencia mecánica elevada.
- Los transistores pueden reproducir el fenómeno de la fotosensibilidad (fenómenos sensibles a la luz)

Al aplicarle polarización al transistor se reduce la región de agotamiento por lo que el flujo mayoritario de portadores fluye desde el material tipo p al tipo n, y dependiendo de sus terminales la magnitud de la corriente de base casi siempre se encuentra en el orden de los micro amperes, comparando con los mili amperes para las corrientes del emisor y del colector, esto ocurre porque al ser polarizado fluyen los portadores mayoritarios de un material al otro. [L02]

Amplificadores Operacionales

El término de amplificador se dio en 1940 para designar una clase de amplificadores que permiten realizar una serie de operaciones tales como suma, resta, multiplicación, integración, derivación; aplicaciones muy importantes en esa época para la computación. Ver figura 3.1.6.

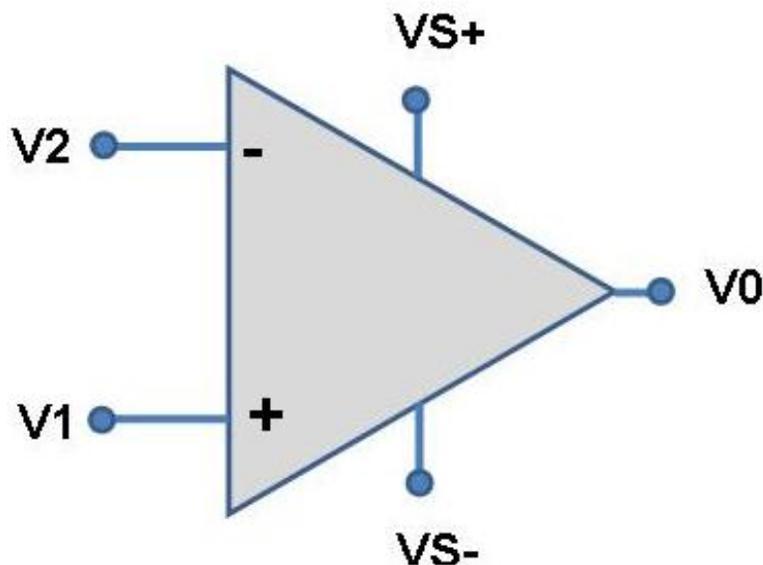


Figura 3.1.6. Amplificador operacional común

Los amplificadores para propósito general son sistemas de etapas múltiples. Una fuente bipolar de cd conectada en las terminales V1 y V2, y dependiendo de la aplicación, las señales de entrada VS+ y VS- pueden ser positivas, negativas o cero; el voltaje obtenido en la salida será según las señales de entrada y la característica del amplificador operacional.

El amplificador operacional por sus características nos puede ayudar a amplificar una señal que sea demasiado pequeña para poderla analizar con mejor detalle o que necesitemos darle una cierta amplitud para poderla integrar a otro sistema.

Los cambios que se le pueden hacer a una señal con los amplificadores operacionales pueden ser muy variadas por sus principales características y que al aplicarlas en una o varias señales se puede obtener el resultado deseado. [L04]

Chip

A finales de la década de 1960 se produjo una evolución tecnológica, gracias a la fabricación del chip, proporcionando grandes avances en ciertos campos, como la informática. Los circuitos integrados con unas delgadas obleas de silicio de tipo p, las cuales son obtenidas a partir de cristales de cuarzo cortados cilíndricamente. Los chips contiene diversos elementos como: transistores, resistencias, capacitores, diodos y conexiones. Los chips son fabricados mediante un proceso fotográfico que evita o permite la acción de ácidos corrosivos. [L13]

Microprocesador

Un Microprocesador, es un dispositivo integrado digital capaz de interpretar y ejecutar un conjunto secuencial de instrucciones, programa. Básicamente contiene circuitos electrónicos que realizan operaciones aritméticas, lógicas y de [control](#). Se trata de un sistema muy complejo, por eso debe ir integrado en chips.

Los microprocesadores no trabajan solos, sino que forman parte de un sistema mayor como se observa en la figura 3.1.7. El microprocesador es el [cerebro](#) o parte inteligente de este sistema, es un chip de actúa como unidad central de proceso del ordenador, incorpora un conjunto de circuitos y memoria. Su tamaño reducido facilita la fabricación de sistemas informáticos más pequeños, utilizados en computadoras, impresoras, coches, aviones, etc. [L08]

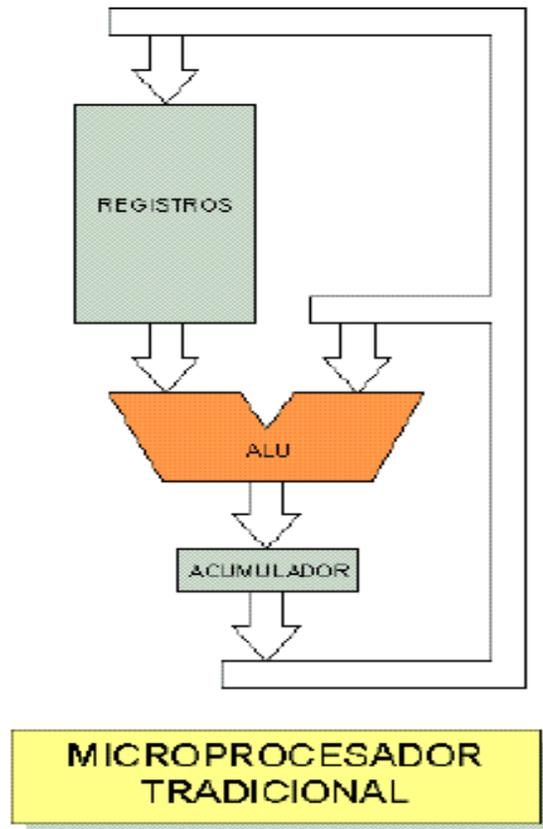


Figura 3.1.7. Esquema de diseño de Un Microprocesador Tradicional

Microcontroladores

Son computadores digitales, integrados en un chip que cuentan con un microprocesador o **CPU (Central Processing Unit, Unidad de Procesamiento Central)**, una memoria para almacenar el programa y otra para datos, puertos de entrada salida, un reloj interno, y uno o más componentes periféricos como son: un temporizador, convertidores analógicos a digital, comunicador serial y circuitos reguladores.

El funcionamiento de los microcontroladores está determinado por el programa que se almacena en su memoria. Este puede escribirse en distintos lenguajes de programación y puede hacerse repetidas veces.

Por las características antes mencionadas y su alta flexibilidad, son ampliamente utilizados como el cerebro de una gran variedad de sistemas que controlan máquinas, componentes de sistemas complejos, en aplicaciones industriales de

automatización, robótica, equipos médicos, sistemas, aeroespaciales, en sistemas especializados, automóviles, hornos de microondas, teléfonos y televisores; así como también están presentes en muchos controles monitoreando y procesando sistemas. [L17]

A continuación se describen los principales elementos que componen a los Microcontroladores:

- Unidad de Procesamiento Central (CPU Central): Típicamente de 8 bits, con arquitectura Harvard, con memoria/bus de datos separada de la memoria/bus de instrucciones de programa, o arquitectura de Von Neumann, con memoria/bus de datos y memoria/ bus de programa compartidas.
- Memoria de Programa: Es una memoria **ROM (Read-Only Memory, Memoria de solo Lectura)**, **EPROM (Electrically Programmable ROM, ROM Eléctricamente Programable)**, **EEPROM (Electrically Erasable, Programable ROM / ROM de programación y borrado eléctrico)** o memoria Flash que almacena el código del programa.
- Memoria de Datos: Es una memoria **RAM (Random Access Memory, Memoria de Acceso Aleatorio)** que puede ser de 1 hasta 32 kilobytes.
- Generador del Reloj: Por lo general es utilizado un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 40 [MHz].
- Interfaz de Entrada/Salida: Puertos paralelos, seriales
- Convertidor Analógico Digital: para convertir un nivel de voltaje en un cierto pin a un valor digital manipulable por el programa del microcontrolador.
- Moduladores PWM (**Pulse Width Modulation, Modulación Ancho de pulso**) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.

La alta integración de subsistemas que componen un microcontrolador reduce el número de chips, las pistas y espacio en un circuito impreso.

A través de los pines del chip asociados a las interfaces de entrada/salida se puede interactuar con otros circuitos externos enviándoles señales de comando o

recibiendo estímulos correspondientes a variables externas. Por lo general varios pines de datos funcionan como entradas o salidas, de acuerdo al programa. [L14]

Las diferencias entre los microprocesadores y microcontroladores son:

- La configuración básica de un Microprocesador está constituida por un Micro, Una memoria RAM, una memoria ROM y un decodificador.
- El Microcontrolador tiene incluido todos los elementos del Microprocesador en un solo Circuito Integrado por lo que simplifica los componentes.
- Los Microprocesadores se basan en la arquitectura de Von Newmann, lo cual indica que el procesador puede leer o escribir datos de instrucción pero no puede hacer los dos al mismo tiempo, mientras que los Microcontroladores trabajan con arquitectura de Harvard puede leer y escribir instrucciones y datos al mismo tiempo en la memoria.
- El tiempo de desarrollo de su proyecto electrónico es menor para los Microcontroladores.
- La principal diferencia entre ambos radica en la ubicación del registro de trabajo.
- En los microcontroladores tradicionales todas las operaciones (borrar, incrementar, decrementar, complementar) se realizan sobre el acumulador.
- En los microcontroladores PIC, la salida de la **ALU (Unidad Aritmética Lógica)** va al registro **W (Working Register / Registro de Trabajo)** y a la memoria de datos, por lo que puede guardarse en cualquiera de los dos destinos.

PIC

Los PIC (**Peripheral Interface Controller, Controlador de Interfaz Periférica**), es una familia de microcontroladores fabricado por Microchip Technology Inc , originalmente desarrollado por la división de microelectrónica de General Instrument, para ser usado en el procesador P1600 que tenía pocas innovaciones en los puertos de I/O (**Input/output, Entradas/Salidas**). En 1985 se le agregó una memoria EPROM creando un controlador programable.

El microcontrolador PIC es único en muchas maneras, este ocupa la arquitectura Harvard, la cual está referida al diseño de la computadora en la que datos e instrucciones usan diferentes caminos y espacios de memoria; se puede leer y escribir instrucciones y datos al mismo tiempo en la memoria. Esto resulta en una máquina rápida y compacta; a comparación del tradicional sistema de arquitectura Von Newman es que el procesador puede leer o escribir datos de instrucción pero no puede hacer los dos al mismo tiempo.

Estos procesadores tienen la memoria separada para el programa y datos con unas reducidas instrucciones. El uso de nombres en dispositivos por Microchip es usado en diferentes códigos que representan diferentes versiones de los componentes. Por ejemplo la primera letra después de la afiliación de la familia representa el tipo de memoria que contiene, a continuación se enlistan las nomenclaturas:

- La letra C en un PIC16CXXX hace referencia a que contiene una memoria EPROM
- La letra CR en un PIC16CRXXX hace referencia a que contiene una memoria ROM
- La letra F en un PIC16FXXX hace referencia a que contiene una memoria FLASH.
- La letra L después de la afiliación de la familia se refiere a un amplio rango de voltaje. [L18]

3.2. LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación son instrucciones que sirven para controlar una máquina, como un ordenador. Existen dos tipos de lenguajes de programación:

- Lenguaje de programación de alto nivel.
- Lenguaje de programación de bajo nivel: es un lenguaje dependiente de la máquina, es decir, el programa que se realiza en este tipo de lenguaje no se puede migrar a otras máquinas.

Cada tipo de CPU entiende su propio lenguaje de máquina. Las instrucciones en lenguaje de máquina son números almacenados como bytes en memoria. Cada instrucción tiene su propio y único código llamado código de operación. Las instrucciones del procesador 80X86 varían en tamaño. El opcode está siempre al inicio de la instrucción.

El lenguaje de máquina es muy difícil de programar directamente, descifrar el significado de las instrucciones codificadas numéricamente resulta tedioso para los programadores. Sin embargo, ellas pueden usar una interfaz de bajo nivel (a nivel del kernel).

Los programadores escriben instrucciones en diferentes lenguajes, algunos comprensibles en forma directa por la computadora, y otros que requieren pasos intermedios de traducción. Existen hoy en día cientos de lenguajes de computadora y los cuales se pueden agrupar en tres grandes tipos:

- Lenguajes de Máquina: Un programa o secuencia de instrucciones viene dado por una secuencia de códigos binarios y cualquier computadora sólo puede entender directamente su propio lenguaje de máquina, este lenguaje es el natural de una computadora particular. Está relacionado íntimamente con el diseño del hardware la computadora en cuestión. Estos lenguajes por lo general consisten en cadenas de números, unos y ceros, que instruyen a las computadoras para que ejecuten sus operaciones más elementales, una a la vez. Estos lenguajes son dependientes de los de máquina, sólo se puede usar en un lenguaje de máquina particular; en un

sólo tipo de computadora. Conforme las computadoras se hicieron más populares, el lenguaje máquina era demasiado lento y tedioso para la mayor parte de los programadores. En vez de utilizar las cadenas de números, que las computadoras pueden entender de forma directa, los programadores empezaron a usar abreviaturas similares en inglés para representar las operaciones elementales de la computadora; estas abreviaturas formaron la base de los lenguajes ensambladores.

- **Lenguajes ensamblador:** Se desarrollaron programas de traducción denominados ensambladores para convertir los programas en lenguaje ensamblador al lenguaje de máquina a la velocidad de las computadoras con mayor claridad. Un programa escrito en lenguaje ensamblador es almacenado como texto, tal como programas de alto nivel. Cada instrucción representa exactamente una instrucción de la máquina. El significado de la instrucción en este lenguaje es mucho más claro que el código de máquina. Por ejemplo la palabra `add` es el nemónico para la instrucción de suma. Un programa ensamblador es un programa que lee un archivo de texto con instrucciones de ensamblador y lo convierte en código de máquina. Los compiladores son programas que hacen conversiones similares para lenguajes de programación de alto nivel. Un ensamblador es mucho más simple que un compilador. Las instrucciones de un lenguaje de alto nivel son mucho más complejas y pueden requerir muchas instrucciones de máquina.
- **Lenguajes de Alto Nivel:** La utilización de las computadoras aumentó con rapidez con la llegada de los lenguajes ensambladores pero aun se necesitaban muchas instrucciones para llevar a cabo alguna tarea. Para acelerar el proceso de programación se desarrollaron los lenguajes de alto nivel, en los cuales se puede escribir un simple enunciado para poder llevar a cabo tareas sustanciales. Los programas de traducción que convierten los programas de alto nivel al lenguaje de la máquina se llaman Compiladores. Los lenguajes de alto nivel permiten a los programadores escribir instrucciones que están prácticamente en inglés y que contienen notaciones

matemáticas por lo que se ha hecho más deseable desde el punto de vista del programador. Cada tipo de CPU tiene su propio lenguaje de máquina, también tiene su propio lenguaje ensamblador. Trasladar programas entre arquitecturas de computador diferentes es mucho más difícil que en un lenguaje de alto nivel.

Se han desarrollado cientos de lenguajes de alto nivel, pero solo unos pocos han alcanzado una amplia aceptación como los que se mencionan a continuación:

- FORTRAN, desarrollado por IBM entre 1954 y 1957, para uso en aplicaciones científicas y de ingeniería.
- COBOL, fue desarrollado en 1959 por un grupo de fabricantes de computadoras, de usuarios industriales y de gobierno. COBOL se utiliza sobre todo en aplicaciones comerciales que requieren manipulación precisa y eficiente de grandes cantidades de datos:
- PASCAL se concibió para uso académico y se convirtió con rapidez en el lenguaje introductorio de programación de la mayor parte de las universidades. Por desgracia el programa carece de muchas características necesarias para hacerlo útil en aplicaciones comerciales, industriales y de Gobierno
- LENGUAJE C: fue derivado de lenguaje B por Dennis Ritchie, de los laboratorios Bell y al inicio se implantó en 1972 en una computadora PDP-11, al principio se hizo muy conocido como lenguaje de desarrollo del sistema operativo UNIX.

Conforme ha pasado el tiempo, C se ha hecho disponible para la mayoría de las computadoras, es independiente del hardware, cuenta con un diseño cuidadoso, es posible escribir programas en este lenguaje que sean portátiles hacia la mayor parte de las computadoras. También utiliza muchos de los conceptos importantes de BCPL y de B, además de añadir el tipo de datos y otras características poderosas.

Dado que C es un lenguaje independiente del hardware y ampliamente disponible, las aplicaciones que están escritas en este lenguaje pueden ejecutarse con poca o ninguna modificación en una amplia gama de sistemas distintos del cómputo. [L05]

3.3 MICROCONTROLADOR PIC 16F887

Los microcontroladores PIC tienen terminales de entrada y salida divididos en puertos que se encuentran numerados alfabéticamente A, B, C, D, etc. Cada puerto puede tener hasta 8 terminales que pueden comportarse como entrada o salida digital. Según las características del PIC cada puerto puede tener además asignado bloques funcionales A/D, I2C, etc. Esto es definido por el usuario en el programa que se esté desarrollando.

En general el PIC a utilizar contiene lo mostrado en la tabla 3.3.1:

Dispositivo	Memoria de Programa	Memoria de datos		I/O	10-bit A/D (ch)	ECCP/ CCP	comparadores	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F887	8192	368	256	35	14	1/1	2	2/1

Tabla 3.3.1 Composición general del PIC 16F887

Físicamente podemos encontrar las terminales distribuidas como lo muestra la figura 3.3.1.

40-pin PDIP

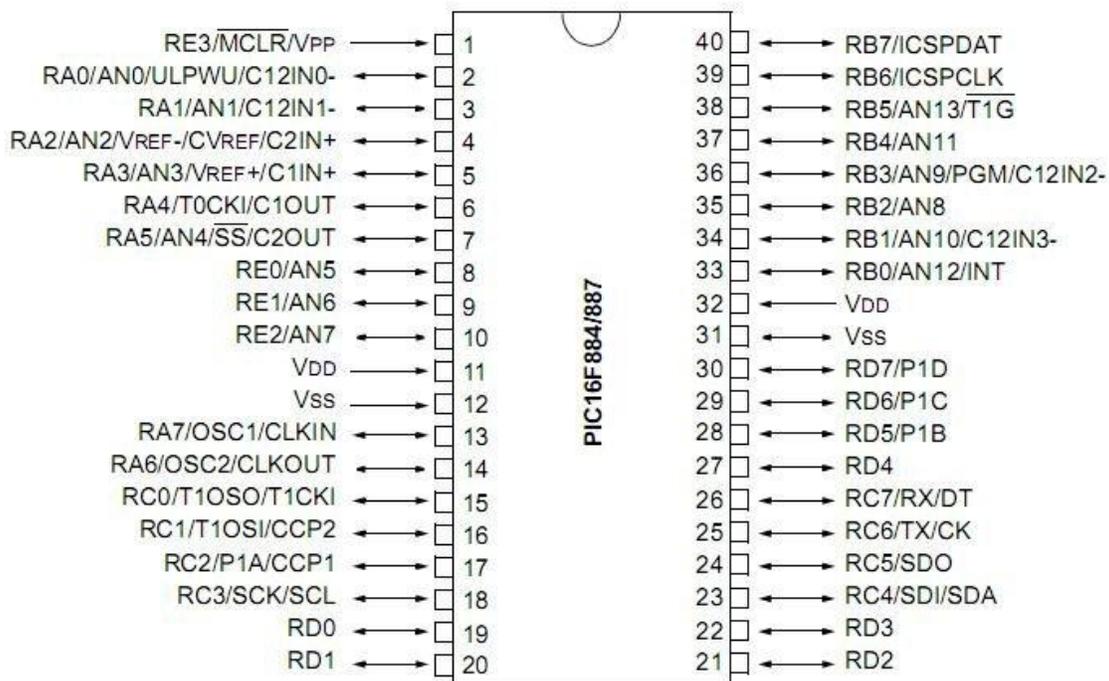


Figura 3.3.1 Esquema físico del PIC 16F887

A continuación se enlistan las características más importantes del PIC16F887:

- Arquitectura **RISC (Reduced Instruction Set Computer, Set Reducido de Instrucciones)**, el microcontrolador cuenta con solo 35 instrucciones diferentes.
- Frecuencia de operación 0-20 [MHz].
- Oscilador interno de alta precisión.
- Rango de frecuencia de 8 [MHz] a 31 [kHz] seleccionado por software.
- Voltaje de la fuente de alimentación de 2.0 [V] a 5.5 [V].
- Ahorro de energía en el Modo de suspensión.
- Brown-out Reset (BOR) con opción para controlar por software.
- 35 pines de entrada/salida.
- Memoria ROM de 8k con tecnología FLASH la cual puede ser re-programada hasta 100, 000 veces.
- 256 bytes de memoria EEPROM la cual puede ser grabada más de 1,000,000 veces.
- 368 bytes de memoria RAM.
- Convertidor A/D 14 canales con una resolución de 10 bits.
- 3 temporizadores/contadores independientes.
- Temporizador perro guardián (watchdog, WDT).
- Módulo comparador analógico.
- Módulo PWM incorporado.
- Módulo USART mejorado, soporta las comunicaciones seriales RS-485, RS-232 y LIN2.0.
- Puerto Serie Síncrono Maestro (MSSP), soporta los modos SPI e I2C.

Internamente se puede representar de acuerdo a la figura 3.3.2 la cual muestra de manera gráfica el contenido del PIC.

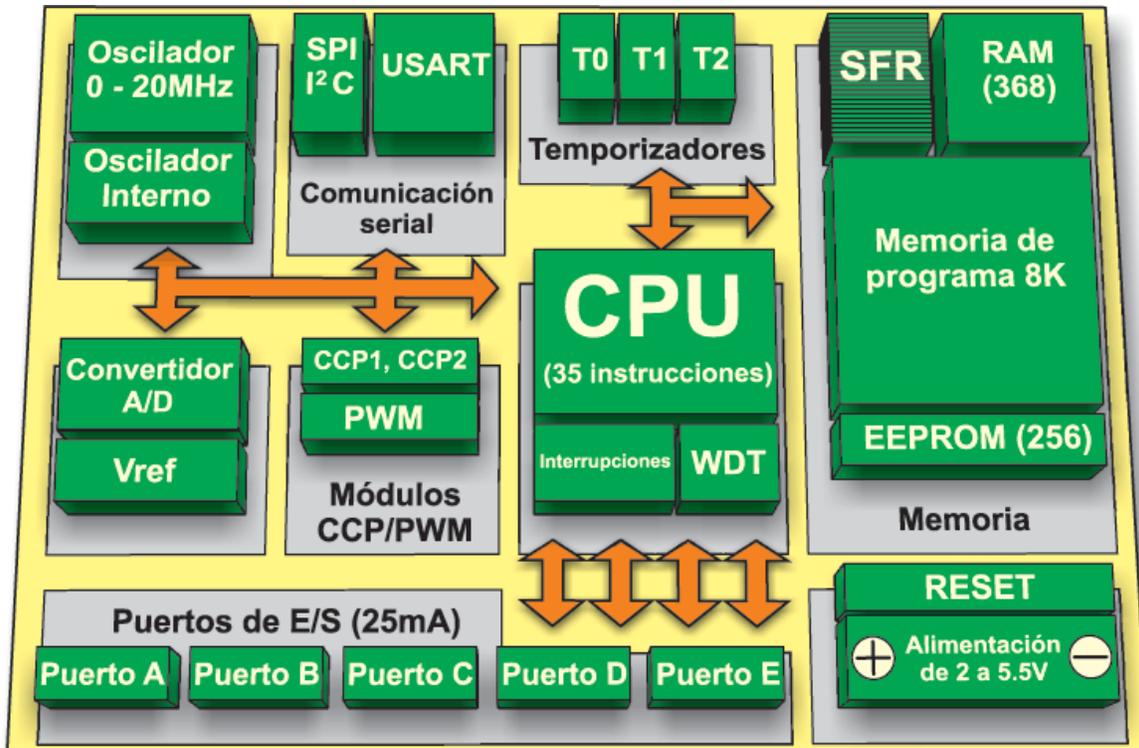


Figura 3.3.2. Representación interna del microcontrolador

Como se puede observar en la figura anterior se muestra la distribución física de los pines, descrito cada uno con una nomenclatura, ésta refiere a características propias del pin y que define el funcionamiento del mismo.

La mayoría de los pines del microcontrolador PIC16F887 son multipropósito como se muestra en la figura 3.3.1. Por ejemplo, la asignación RA3/AN3/Vref+/C1IN+ para el quinto pin del microcontrolador indica que éste dispone de las siguientes funciones:

- RA3 Tercera entrada/salida digital del puerto A.
- AN3 Tercera entrada analógica.
- Vref+ Referencia positiva de voltaje.
- C1IN+ Entrada positiva del comparador C1.

Esta asignación, realizada por el usuario o programador, es de gran utilidad puesto que permite un mejor aprovechamiento de los recursos del microcontrolador sin afectar su funcionamiento. La tabla 3.3.2 muestra las características de cada pin:

Nombre	Número (DIP 40)	Función	Descripción
RE3/MCLR/Vpp	1	RE3	Entrada de propósito general en el puerto PORTE
		MCLR	Pin de reinicio. El nivel lógico bajo en este pin reinicia al microcontrolador
		Vpp	Voltaje de programación
RA0/AN0/ULPWU/C12IN0-	2	RA0	E/S de propósito general en el puerto PORTA
		AN0	Entrada del canal 0 del convertidor A/D
		ULPWU	Entrada de desactivar el modo de espera
		C12IN0-	Entrada negativa del comparador C1 o C2
RA1/AN1/C12IN1-	3	RA1	E/S de propósito general en el puerto A
		AN1	Canal 1 del convertidor A/D
		C12IN1-	Entrada negativa del comparador C1 o C2
RA2/AN2/Vref-/CVref/C2IN+	4	RA2	E/S de propósito general en el puerto PORTA
		AN2	Canal 2 del convertidor A/D
		Vref-	Entrada de referencia negativa de voltaje del convertidor A/D
		CVref	Salida de referencia de voltaje del comparador
		C2IN+	Entrada positiva del comparador C2
RA3/AN3/Vref+/C1IN+	5	RA3	E/S de propósito general en el puerto PORTA
		AN3	Canal 3 del convertidor A/D
		Vref+	Entrada de referencia positiva de voltaje del convertidor A/D
		C1IN+	Entrada positiva del comparador C1
RA4/T0CKI/C1OUT	6	RA4	E/S de propósito general en el puerto PORTA
		T0CKI	Entrada de reloj del temporizador T0
		C1OUT	Salida del comparador C1
RA5/AN4/SS/C2OUT	7	RA5	E/S de propósito general en el puerto PORTA
		AN4	Canal 4 del convertidor A/D
		SS	Entrada del módulo SPI (<i>Selección del esclavo</i>)
		C2OUT	Salida del comparador C2
RE0/AN5	8	RE0	E/S de propósito general en el puerto PORTE
		AN5	Canal 5 del convertidor A/D
RE1/AN6	9	RE1	E/S de propósito general en el puerto PORTE
		AN6	Canal 6 del convertidor A/D
RE2/AN7	10	RE2	E/S de propósito general en el puerto PORTE
		AN7	Canal 7 del convertidor A/D
Vdd	11	+	Suministro de voltaje positivo
Vss	12	-	Tierra (ground - GND)

Nombre	Número (DIP 40)	Función	Descripción
RA7/OSC1/CLKIN	13	RA7	E/S de propósito general en el puerto PORTA
		OSC1	Entrada del oscilador de cristal
		CLKIN	Entrada del reloj externo
RA6/OSC2/CLKOUT	14	OSC2	Salida del oscilador del cristal
		CLKO	Salida en la que se presenta la señal $F_{osc}/4$
		RA6	E/S de propósito general en el puerto PORTA
RC0/T1OSO/T1CKI	15	RC0	E/S de propósito general en el puerto PORTC
		T1OSO	Salida del oscilador del temporizador 1
		T1CKI	Entrada de reloj del temporizador 1
RC1/T1OSO/T1CKI	16	RC1	E/S de propósito general en el puerto PORTC
		T1OSI	Entrada del oscilador del temporizador 1
		CCP2	E/S de los módulos CCP1 y PWM1
RC2/P1A/CCP1	17	RC2	E/S de propósito general en el puerto PORTC
		P1A	Salida del módulo PWM
		CCP1	E/S de los módulos CCP1 y PWM1
RC3/SCK/SCL	18	RC3	E/S de propósito general en el puerto PORTC
		SCK	E/S de reloj del módulo MSSP en el modo SPI
		SCL	E/S de reloj del módulo MSSP en el modo I ² C
RD0	19	RD0	E/S de propósito general en el puerto PORTD
RD1	20	RD1	E/S de propósito general en el puerto PORTD
RD2	21	RD2	E/S de propósito general en el puerto PORTD
RD3	22	RD3	E/S de propósito general en el puerto PORTD
RC4/SDI/SDA	23	RC4	E/S de propósito general en el puerto PORTC
		SDI	Entrada <i>Data</i> del módulo MSSP en el modo SPI
		SDA	E/S <i>Data</i> del módulo MSSP en el modo I ² C
RC5/SDO	24	RC5	E/S de propósito general en el puerto PORTC
		SDO	Salida <i>Data</i> del módulo MSSP en el modo SPI
RC6/TX/CK	25	RC6	E/S de propósito general en el puerto PORTC
		TX	Salida asíncrona del módulo USART
		CK	Reloj síncrono del módulo USART
RC7/RX/DT	26	RC7	E/S de propósito general en el puerto PORTC
		RX	Entrada asíncrona del módulo USART
		DT	Datos del módulo USART en modo síncrono

Nombre	Número (DIP 40)	Función	Descripción
RD4	27	RD4	E/S de propósito general en el puerto PORTD
RD5/P1B	28	RD5	E/S de propósito general en el puerto PORTD
		P1B	Salida del módulo PWM
RD6/P1C	29	RD6	E/S de propósito general en el puerto PORTD
		P1C	Salida del módulo PWM
RD7/P1D	30	RD7	E/S de propósito general en el puerto PORTD
		P1D	Salida del módulo PWM
Vss	31	-	Tierra (GND)
Vdd	32	+	Suministro de voltaje positivo
RB0/AN12/INT	33	RB0	E/S de propósito general en el puerto PORTB
		AN12	Canal 12 del convertidor A/D
		INT	Interrupción externa
RB1/AN10/C12INT3-	34	RB1	E/S de propósito general en el puerto PORTB
		AN10	Canal 10 del convertidor A/D
		C12INT3-	Entrada negativa de los comparadores C1 o C2
RB2/AN8	35	RB2	E/S de propósito general en el puerto PORTB
		AN8	Canal 8 del convertidor A/D
RB3/AN9/PGM/C12IN2-	36	RB3	E/S de propósito general en el puerto PORTB
		AN9	Canal 9 del convertidor A/D
		PGM	Habilita la programación del chip
		C12IN2-	Entrada negativa de los comparadores C1 o C2
RB4/AN11	37	RB4	E/S de propósito general en el puerto PORTB
		AN11	Canal 11 del convertidor A/D
RB5/AN13/T1G	38	RB5	E/S de propósito general en el puerto PORTB
		AN13	Canal 13 del convertidor A/D
		T1G	Entrada externa del temporizador 11
RB6/ICSPCLK	39	RB6	E/S de propósito general en el puerto PORTB
		ICSPCLK	Entrada de reloj de programación serial
RB7/ICSPDAT	40	RB7	E/S de propósito general en el puerto PORTB
		ICSPDAT	Pin de E/S para introducir los datos durante la programación ICSP™

Tabla 3.3.2. Descripción de pines PIC16F887

Ahora se describen las partes internas y la parte más importante es la Unidad Central de Procesos, funciona con tecnología RISC, que proporciona las siguientes ventajas:

- Cuenta con sólo 35 instrucciones simples lo que hace más rápido su proceso de ejecución del programa
- El tiempo de ejecución es igual para casi todas las instrucciones y tarda 4 ciclos de reloj. La frecuencia del oscilador se estabiliza por un cristal de cuarzo. Las

instrucciones de salto y de ramificación tardan ocho ciclos de reloj en ejecutarse. Esto significa que si la velocidad de operación del microcontrolador es 20 [MHz], el tiempo de ejecución de cada instrucción será 200 [nseg], es decir, el programa ejecutará 5 millones de instrucciones por segundo.

El PIC16F887 tiene tres tipos de memoria: ROM, RAM y EEPROM. La memoria ROM se utiliza para guardar permanente el programa que se está ejecutando. Es la razón por la que es frecuentemente llamada “memoria de programa” como ya se mencionó tiene 8 kb, en total 8192 localidades.

Similar a la memoria de programa, el contenido de memoria EEPROM está permanentemente guardado al apagar la fuente de alimentación. Sin embargo, a diferencia de la ROM (la cual puede ser cambiada al aplicarle voltaje de 13 [V] de un programador) el contenido de la EEPROM se puede cambiar durante el funcionamiento del microcontrolador. Es la razón por la que esta memoria (256 localidades) es perfecta para guardar algunos resultados creados y utilizados durante la ejecución del programa.

La tercera y última memoria es la RAM, es la más compleja debido a que se subdivide en dos partes: en registros de propósito general y en los registros de funciones especiales (SFR). Todos estos registros se dividen en cuatro bancos (ver figura 3.3.3).

Los registros de propósito general se utilizan para almacenar los datos temporales y los resultados creados durante el funcionamiento. Por ejemplo, si el programa realiza el conteo (de los productos en una cadena de montaje), es necesario tener un registro que representa una suma, y es necesario especificarle al microcontrolador la dirección de un registro de propósito general y asignarle esa función. Una vez creado el programa de suma el microcontrolador puede ejecutar el programa ya que sabe qué es y dónde está la suma que se va a incrementar. De manera similar, a cada variable de programa se le debe pre-asignar alguno de los registros de propósito general.

Los registros de funciones especiales a diferencia de los registros de propósito general, son predeterminados durante el proceso de fabricación y no se pueden cambiar. Como los bits están conectados a los circuitos particulares en el chip (convertidor A/D, módulo de comunicación serial, etc), cualquier cambio de su contenido afecta directamente al funcionamiento del microcontrolador o de alguno de los circuitos. Por ejemplo, el registro ADCON0 controla el funcionamiento del convertidor A/D. Al cambiar los bits se determina qué pin del puerto se configurará como la entrada del convertidor, el momento del inicio de la conversión así como la velocidad de la conversión. Otra característica de estas localidades de memoria

es que tienen nombres (tanto los registros como sus bits), lo que simplifica considerablemente el proceso de escribir un programa. Como el lenguaje de programación de alto nivel puede utilizar la lista de todos los registros con sus direcciones exactas, basta con especificar el nombre de registro para leer o cambiar su contenido mediante la asignación de valores en TRISC y PORTC.

En el caso de programar a TRISC = 0x00 (valor 0) todos los pines del puerto PORTC se configuran como salidas y si se coloca el valor lógico 1, quedan configuradas como entradas.

La figura 3.3.3 se ejemplifica la distribución de las memorias del PIC así como la subdivisión de la memoria RAM:

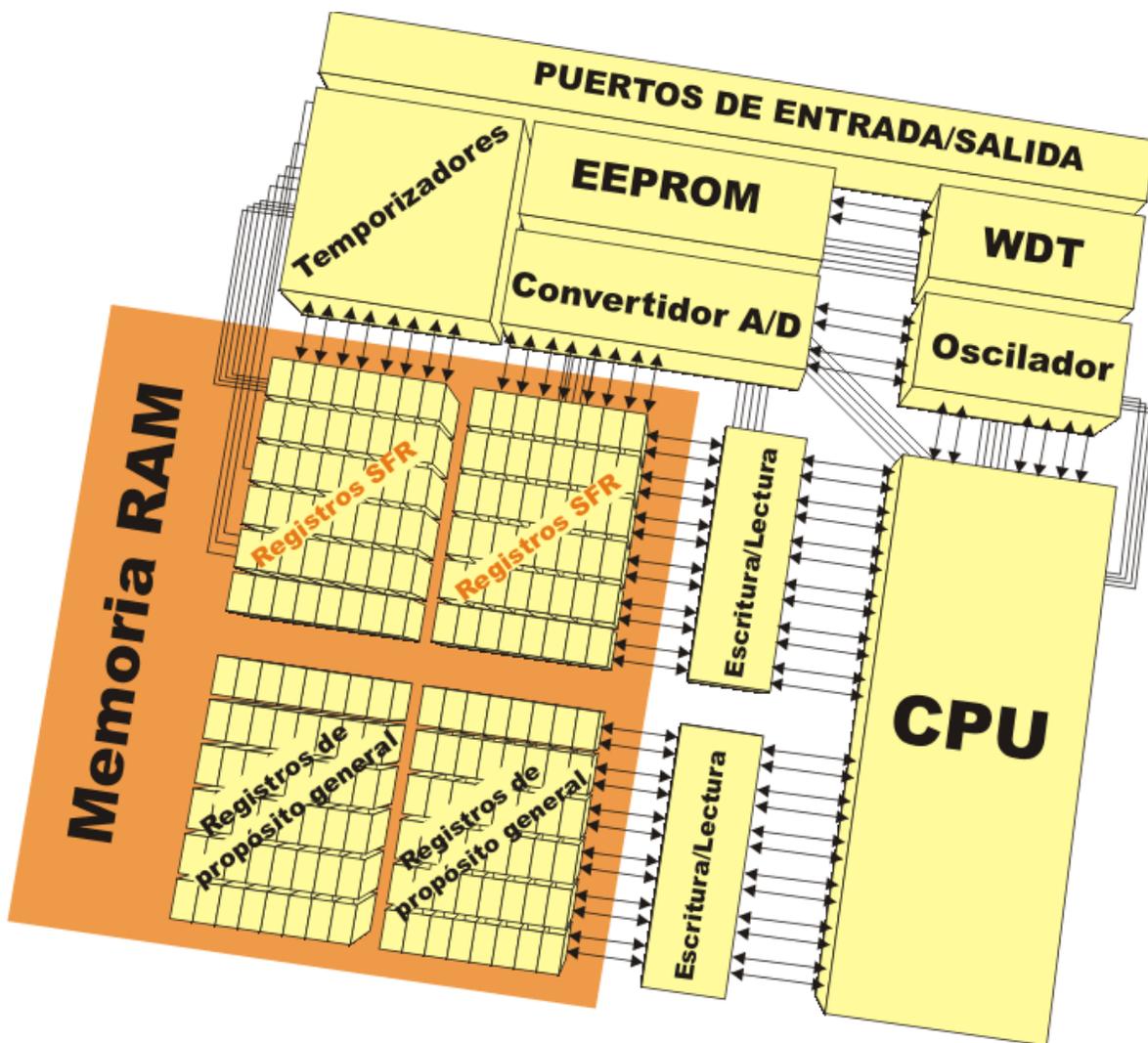


Figura 3.3.3. Representación gráfica de bancos de memoria

Otra de las funciones útiles del PIC son las interrupciones, estas son programadas para detectar algún estado no deseado del programa, la localidad en la que el programa se dirige al ser activada se le denomina vector de interrupción. En el caso de este microcontrolador está en la dirección es 0x0004h.

Una parte de programa que se ejecutará al hacer una petición de interrupción se le denomina rutina de interrupción. Su primera instrucción se encuentra en el vector de interrupción. Cuánto tiempo tardará en ejecutar esta subrutina y cómo será depende de la también del programador así como de la fuente de interrupción.

Se describieron las características que más nos interesan en el PIC debido a que son las que se manejan en el Sistema Simulador y se puede describir su uso de la siguiente manera:

Se utilizan las 8 salidas RC, que a su vez son enviadas a un convertidor analógico digital, el cual ayuda a obtener las señales analógicas requeridas. El otro puerto a utilizar es el puerto RE3, debido a una característica principal: posee una función secundaria llamada Master Clear. En el Sistema Simulador se usó esta función con el objetivo de tener una protección en el sistema, esta protección consta de un botón externo el cual, al ser activado, reseteará el sistema y así corregirá algún problema de código o algún bucle infinito al cual haya caído por error. Los circuitos electrónicos son muy susceptibles a errores y más si son manejados en condiciones no aptas para los mismos, es decir, uno de los casos más frecuentes es usarlos en lugares donde la temperatura es alta y/o en lugares donde el ruido eléctrico es muy alto.

3.4 COMPILADOR MIKRO C

El Sistema Simulador está basado en la programación de un PIC, el cual se programa en un lenguaje de bajo nivel “ensamblador”. Debido a los avances tecnológicos, tanto a nivel software como a nivel hardware, se han desarrollado nuevas herramientas que facilitan el manejo de la electrónica en general, en este caso se seleccionó el lenguaje C como compilador y manejador de los PICs como principal herramienta.

Un programa en sentido informático está constituido por un conjunto de instrucciones que se ejecutan ordinariamente de modo secuencial, es decir, cada una a continuación de la anterior. Con objeto de disminuir los tiempos de ejecución de programas críticos, en la actualidad se puede encontrar programas que funcionan de forma paralela, es decir, programas que se pueden ejecutar simultáneamente en varios procesadores. Este tipo de programación es mucho más complicada que la secuencial y no se hace referencia, pues está enfocada únicamente al modo secuencial.

Una de las mayores ventajas del compilador C es el uso de las funciones pues los programas que habitualmente se utilizan contienen decenas y aún cientos de miles de líneas de código fuente. A medida que los programas se van desarrollando y aumentan de tamaño, se convertirían rápidamente en sistemas poco manejables, para ello existe una herramienta llamada modularización, es el proceso que consiste en dividir un programa muy grande en una serie de módulos mucho más pequeños y manejables. A estos módulos se les ha denominado de distintas formas (subprogramas, subrutinas, procedimientos, funciones, etc.) según los distintos lenguajes. El lenguaje C hace uso del concepto de función. No importa la nomenclatura, el concepto es básicamente el mismo: dividir un programa grande en un conjunto de subprogramas o subprocesos más pequeñas que son llamados por el programa principal, éstas a su vez llaman a otras funciones más específicas, y así sucesivamente. La división de un programa en unidades más pequeñas o funciones presenta entre otras las ventajas siguientes:

- Cada función tiene una misión muy concreta, de modo que nunca tiene un número de líneas excesivo y siempre se mantiene dentro de un tamaño manejable. Además, una misma función puede ser llamada muchas veces en un mismo programa e incluso puede ser reutilizada por otros programas. Cada función puede ser desarrollada y comprobada por separado.
- En la medida en que una misma función es utilizada muchas veces, el número total de líneas de código del programa

disminuye y también lo hace la probabilidad de introducir errores en el programa lo que ayuda a ahorrar considerablemente el uso de memoria y ahorra el tiempo de desarrollo, cabe recordar que los PICs no contienen grandes cantidades de memoria.

- Es muy frecuente que al hacer una modificación para añadir una nueva funcionalidad o corregir un error, se introduzcan nuevos errores en partes del programa que antes funcionaban correctamente. Una función es capaz de mantener una gran independencia con el resto del programa, manteniendo sus propios datos y definiendo muy claramente la interfaz o comunicación con la función que la ha llamado y con las funciones a las que llama.

Existen varios compiladores para C, aparte del que se empleo para el Sistema Simulador, como CCS. MIKRO C presenta varias ventajas a comparación de CCS una de las más fundamentales es la cantidad de información disponible para su manejo, por ello de aquí en adelante el manejo de C está orientado a librerías, ambientes gráficos e instrucciones que son usados en MIKRO C, que haciendo un análisis más profundo no existe mucha diferencia entre ambos software de programación, la característica más notable y a su vez más útil, es el ambiente gráfico. Cabe mencionar que Mikro C es la versión estudiantil que el proveedor ofrece, restringiendo únicamente la dimensión del código el cual no debe exceder los 2 Kbytes.

El compilador de alto nivel C utiliza estructuras que facilitan la programación, optimiza las operaciones matemáticas y los procesos por medio de funciones y un conjunto de variables de tipo carácter, entero y de punto decimal. El compilador crea automáticamente el código ensamblador y el archivo *.hex, este archivo es el resultado del compilador dado que con este se programa eléctricamente el microcontrolador, en el caso del Sistema Simulador no será físicamente realizado. El compilador C soporta las siguientes variables básicas:

- bit: permite almacenar un valor lógico, es decir, verdadero o falso, 0 ó 1.
- char: se utiliza para almacenar caracteres codificados con el código ASCII, son útiles para guardar letras o textos.
- short: almacena un número entero de 8 bits corto puede valer de -127 a 127.
- int: guarda números enteros de 16 bits, está variable permite guardar números de -32767 a 32767.
- long: almacena números enteros largos de 32 bits, su rango puede ser de -2147483647 a 2147483647.

- float y double: permiten guardar números con punto decimal de $\pm 1.175 \times 10^{-38}$ a $\pm 3.402 \times 10^{+38}$.

La tabla 3.4.1 muestra las características de las variables:

Tipo	Tamaño	Rango	Descripción
Int1 Short	1 bit	0 a 1	Entero de 1 bit
Int Int8	8 bit	0 a 255	Entero
Int 16 Long	16 bit	0 a 65.535	Entero de 16 bit
Int32	32 bit	0 a 4294967295	Entero de 32 bit
Float	32 bit	$+1.175 \times 10^{-38}$ a $+3.402 \times 10^{+38}$	Coma Flotante
Char	8 bit	0 a 255	Carácter
Void	-	-	Sin valor
Signed Int8	8 bit	-128 a +127	Entero con signo
Signed Int16	16 bit	-32768 a +32767	Entero largo con signo
Signed Int32	32 bit	-2^{31} a $(2^{31}-1)$	Entero 32 bit con signo

Tabla 3.4.1. Variables

La declaración de las variables dentro del programa se realiza mencionando el tipo de variable seguido del nombre de la variable, con tantas declaraciones como variables se requieran, que cada programador lo asignará según sus necesidades, cabe mencionar que es posible inicializar las variables con un valor, sin embargo no es necesario, lo que sí se necesita es su declaración antes de su uso. Por último se coloca el punto y coma (;) que es el indicativo de término de asignación de valor a la variable o simplemente término de asignación del nombre.

A continuación se mostraran algunos ejemplos del uso de las variables según su tipo. (Nota: los caracteres "//texto" indican un comentario)

```

bit VARIABLE1_BIT; //Declaración de una variable tipo bit.
char CARÁCTER; //Declaración de una variable tipo char.
char CARACTER2='J'; //Declaración de una variable tipo char inicializada con
//el valor ASCII del carácter J.
int ENTERO=1234; //Declaración de una variable tipo entera inicializada con
//el valor 1234.
float DECIMAL=-12.45; //Declaración de una variable con punto decimal
//inicializada con el valor -12,45.
double DECIMAL2=56.68; //Declaración de una variable con punto decimal
//inicializada con el valor 56,68.

```

```
long ENTERO2=-954261;           //Demacración de una variable de tipo entero
                                //largo inicializada con el valor -954261.
```

Los siguientes ejemplos muestran como declarar variables sin signo:

```
unsigned char CHARACTER;       // Declaración de una variable tipo char sin signo
unsigned int ENTERO;           // Declaración de una variable tipo entera sin
                                //signo.
unsigned long ENTERO2;         // Declaración de una variable de tipo entero //largo
                                sin signo.
```

Variables con índice

Contienen valores diversos con un mismo nombre pero diferenciados mediante índices, como son los vectores y matrices (una y dos dimensiones).

El siguiente ejemplo muestra un vector de caracteres, o también conocido como una cadena de caracteres:

```
char Texto[20];                 //Cadena de caracteres con 20 posiciones de memoria.
```

De igual forma las cadenas de caracteres o de variables pueden ser declaradas con un valor inicial, este tipo de declaración se puede ver en el siguiente ejemplo:

```
char Texto[20] = "Nuevo Texto"; //Declaración de una cadena de caracteres
//inicializada con el texto: "Nuevo Texto".
int Enteros[5]={5,4,3,2,1};      //Declaración de una cadena de enteros con
//valores iniciales.
float Decimales[3]={0.8,1.5,5.8}; //Declaración de una cadena de números con
//punto decimal inicializadas.
```

Estructuras

Son variables que agrupan las características de un concepto mediante "miembros de la estructura", los cuales pueden asociarse a variables y arreglos, denominados: extensiones o apellidos de la estructura. Este tipo de estructuras se declarar, por medio de la directriz: *typedef struct*, y la forma de declarar una variable creada por el desarrollador es la siguiente:

```
typedef struct
{
char Nombre[10];
int Edad;
}Usuario;
```

La siguiente es la forma de usar una variable personalizada:

```
Usuario U;
U.Edad = 25;
```

```

U.Nombre[0]="L";
U.Nombre[1]="u";
U.Nombre[2]="i";
U.Nombre[3]="s";
U.Nombre[4]=0;

```

El compilador Mikro C maneja cantidades en sistema binario y sus representaciones octal, decimal y hexadecimal (bases 2, 8, 10, y 16). La escrita en decimal es la forma de mayor simplicidad ya que se escriben en lenguaje C de la misma manera convencional como que se aprendió en los primeros cursos escolares.

Los números binarios se escriben con el encabezado 0b seguidos del número en binario, un ejemplo de esta escritura es: 0b10100001 que es equivalente al número decimal 161.

Los números en base 16 o hexadecimales se denotan con el encabezado 0x precedidos de un número en hexadecimal, la expresión 0x2A, es un ejemplo de un número hexadecimal en lenguaje C, y equivale a 42 en decimal, pueden tener 16 dígitos que son; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F.

Operadores

Otra parte importante del lenguaje son las operaciones aritméticas o matemáticas básicas entre variables o constantes, los operadores aritméticos se usan para realizar operaciones matemáticas se listan en la tabla 3.4.2:

Carácter	Operador
+	Suma (enteros o reales) resul = var1 + var2
-	Resta (enteros o reales) resul = var1 - var2
*	Multiplicación (enteros o reales) resul = var1 * var2
/	División (enteros o reales) resul = var1 / var2
%	Módulo; resto de una división entera rango = n [A1]% 256
--	Decremento
++	Incremento
sizeof	Determina el tamaño, en bytes, de un operando

Tabla 3.4.2 Operadores aritméticos

La suma aritmética entre dos o más números, a continuación se presenta un ejemplo:

```

int A;
int B;
int C;
C = A+B; //Esta expresión guarda la suma de A y B en la variable C.
C = A+B+C; //Esta expresión guarda la suma de A, B y C en la variable C.

```

La resta aritmética entre dos o más números, se puede hacer como se ve en el siguiente ejemplo:

```

int A;
int B;
int C;
C = A-B; //Esta expresión guarda la diferencia entre A y B en la variable C.

```

`C = A-B-C;` //Está expresión guarda la diferencia entre A, B y C en la variable C.
La operación matemática de multiplicación se puede realizar entre dos o más números, la operación se relaciona con el carácter asterisco (*), esta expresión se puede apreciar con mayor claridad en el siguiente ejemplo:

```
int A;
int B;
int C;
C = A*B; //Esta expresión guarda la multiplicación entre A y B en la variable C.
C = A*B*C; //Esta expresión guarda la multiplicación entre A, B y C en la variable C.
```

La división aritmética en lenguaje C se especifica por medio de la barra inclinada (/), en el siguiente ejemplo se puede observar su implementación:

```
int A;
int B;
int C;
C = A/B; //Esta expresión guarda la división A entre B en la variable C.
```

La operación módulo calcula el módulo de una división aritmética es decir calcula el residuo de una división, el cálculo del módulo se denota con el carácter de porcentaje (%), la aplicación de esta operación se puede ver en el siguiente ejemplo:

```
int A;
int B;
int C;
C = A%B; //Esta expresión guarda el residuo de la división de A entre B en la //variable C.
```

Otros operadores matemáticos abreviados pueden ser utilizados cuando se requiere de conteos o cambios de una variable de forma constante, por ejemplo, es posible incrementar o decrementar una variable en términos de un número, de igual manera es posible hacer esta operación con la multiplicación y la división. Para entender de forma más clara este concepto se muestran los siguientes ejemplos:

```
int A=100;
int B=10;
A++; //Este operador incrementa en una unidad el valor de A.
A--; //Este operador decrementa en una unidad el valor de A.
A+=4; //Este operador incrementa en 4 el valor de A.
A-=5; //Este operador decrementa en 5 el valor de A.
A/=4; //Este operador divide el valor de A en 4.
A*=3; //Este operador multiplica el valor de A por 3. 25
A+=B; //Este operador incrementa el valor de A en el valor de B unidades.
A*=B; //Este operador multiplica el valor de A por B veces.
```

Es posible utilizar también operadores lógicos conocidas como operadores de Boole quien planteó las operaciones OR, AND, NOT, XOR, NOR, NAND, XNOR.

Las operaciones lógicas se realizan en lenguaje C entre dos variables o constantes, éstas se hacen bit a bit, del mismo peso en una variable o número.

Las operaciones lógicas en lenguaje C, se realizan con caracteres específicos, que denotan cada una de ellas, los siguientes ejemplos muestran las aplicaciones de los operadores lógicos.

Operación lógica NOT, negación, se denota con el carácter virgulilla (~).
unsigned short VALOR1=0b01010000; //Variable inicializada en binario con //el número 80.
unsigned short RESULTADO;
RESULTADO = ~VALOR1; //La variable RESULTADO guarda el //complemento de VALOR1.

Operación lógica OR, o inclusiva, está se denota con el carácter barra vertical (|).
unsigned short VALOR1=0b01010000; //Variable inicializada en binario //con el número 80.
unsigned short VALOR2=0b01011111; //Variable inicializada en binario con //el número 95.
unsigned short RESULTADO;
RESULTADO = VALOR1|VALOR2; //El valor de la operación lógica or, es //guardado en RESULTADO.

Operación lógica XOR, o exclusiva, está se denota con el carácter acento circunflejo (^).
unsigned short VALOR1=0b01010000; //Variable inicializada en binario con //el número 80.
unsigned short VALOR2=0b01011111; //Variable inicializada en binario //con el número 95.
unsigned short RESULTADO;
RESULTADO = VALOR1^VALOR2; //El valor de la operación lógica //AND es guardado en RESULTADO, 15.

Operación lógica AND, y, está se denota con el carácter ampersand (&).
unsigned short VALOR1=0b01010000; //Variable inicializada en binario con //el número 80.
unsigned short VALOR2=0b01011111; //Variable inicializada en binario con //el número 95.
unsigned short RESULTADO;
RESULTADO = VALOR1&VALOR2; //El valor de la operación lógica o //exclusiva, es guardado en //RESULTADO, 80.

La implementación de las operaciones lógicas NAND, NOR, XNOR, son similares a AND, OR, y XOR, agregando el carácter de negación virgulilla.

```
unsigned short VALOR1=0b01010000; //Variable inicializada en binario con //el número 80.
```

```
unsigned short VALOR2=0b01011111; //Variable inicializada en binario con //el número 95.
```

```
unsigned short RESULTADO;
```

```
RESULTADO = ~(VALOR1|VALOR2); //El valor de la operación lógica o negada // exclusiva, es guardado en RESULTADO.
```

```
RESULTADO = ~(VALOR1&VALOR2); //El valor de la operación lógica y negada, // es guardado en RESULTADO.
```

```
RESULTADO = ~(VALOR1^VALOR2); //El valor de la operación lógica o negada // exclusiva negada, es guardado en //RESULTADO.
```

Otra de las herramientas básicas en mikroC es el desplazamiento de bits dentro de una variable, es útil para realizar procesos y tareas que impliquen la manipulación de datos a nivel de los bits. El corrimiento a la derecha en una variable o valor constante se realiza por medio del doble carácter mayor que (>>), de la misma manera el corrimiento a la izquierda se ejecuta con el doble carácter menor que (<<). La operación de corrimiento hace perder los valores de los bits que salen, e ingresa ceros en los nuevos bits. En los siguientes ejemplos se muestra su implementación:

```
short Dato=0xFF; //Declaración de variables.
```

```
short Resultado;
```

```
Resultado = Dato>>4; //Está operación guarda en la variable Resultado //el corrimiento de 4bits a la derecha de la //variable Dato, valor final de Resultado es 0x0F.
```

```
Resultado = Dato<<4; //Está operación guarda en la variable Resultado //el corrimiento de 4 bits a la izquierda de la //variable Dato, valor final de Resultado es 0xF0.
```

La tabla 3.4.3 muestra los operadores vistos previamente:

Descripción de la operación	Operador
Suma	+
Resta	-
División	/
Módulo o residuo de la división entera	%
Incremento	++
Decremento	--
Operación OR	
Operación AND	&
Operación XOR	^
Complemento a 1	~
Corrimiento a la derecha	>>
Corrimiento a la izquierda	<<

Tabla 3.4.3. Tabla de operadores

Condiciones e iteraciones en Lenguaje C: Es fácil realizar un programa, pero dependiendo de la complejidad del mismo es necesario hacer uso de otras herramientas como son los condicionales, permiten realizar ciclos desde n hasta infinito o bien toman decisiones según sea el caso. En los siguientes ejemplos se muestra su uso:

```
short A; 31
```

```
short B;
```

```
(A==10)&&(B==20) //La condición es verdadera si A vale 10 y B vale 20.
```

```
(A!=5)&&(B>2) //La condición es verdadera si A es diferente de 5 y B es //mayor que 2.
```

```
(A<=4)&&(B>=6) //La condición es verdadera si A es menor o igual que 4 //o si B es mayor o igual que 6.
```

```
!(A>3)&&(B!=4) //La condición es verdadera si A no es mayor que 3 y si //B es diferente de 4.
```

Sentencia condicional if e if else

La estructura de la sentencia if evalúa una condición lógica y ejecuta una porción de código si y solo si el resultado de la evaluación es verdadera. La estructura se puede ver en el siguiente ejemplo:

```
short Valor;
```

```
if( Valor>100 ) //Este if evalúa si la variable Valor es mayor que 100.
```

```
{
```

```
// Porción de código que se ejecuta si el if es verdadero.
```

```
}
```

La estructura if else es igual que la sentencia if, su única diferencia es que en el dado caso de que la evaluación de la condición sea falsa se ejecuta la porción de código asociada al else. Para entender esta situación se ejemplifica de la siguiente forma:

```
short Valor;

if( Valor>100 ) //Este if evalúa si la variable Valor es mayor que 100.
{
// Porción de código que se ejecuta si la condición del if es verdadero.
}

else
{
// Porción de código que se ejecuta si la condición del if es falsa.
}
```

Las sentencias if e if else en conjunto se pueden anidar para crear estructuras más completas y complejas, haciendo que su uso sea más útil.

Sentencia switch case

La sentencia switch case, funciona de manera similar a la sentencia if, difiere en que no se evalúa una condición si no el valor de una variable y ejecuta una porción de código para cada valor posible de la variable. Los valores contemplados en los casos o case posibles de la variable los escoge el desarrollador arbitrariamente en función de su criterio y necesidad. Los casos o valores que no son de interés para el programador se ejecutan en un fragmento de código por defecto, por medio de la directriz default. Cada uno de los fragmentos de código editados deben terminar con la directriz break para romper el flujo de la estructura switch case, esta acción es necesaria dado que si dos casos están seguidos los fragmentos de código seguidos se ejecutarán hasta encontrar la directriz break o hasta finalizar la estructura switch case. Para comprender de forma clara el funcionamiento de la sentencia switch case se propone el siguiente ejemplo:

```
short Valor;

switch( Valor ) //Evaluación de la variable Valor.
{
case 0:        //Fragmento de código correspondiente al valor 0 de la variable Valor.
break;        //Ruptura del caso 0.
```

```

case 1:      //Fragmento de código correspondiente al valor 1 de la variable Valor.
break;      //Ruptura del caso 1.

case 10:    //Fragmento de código correspondiente al valor 10 de la variable
//Valor.
break;      //Ruptura del caso 10.

case 20:    //Fragmento de código correspondiente al valor 20 de la variable
//Valor.
break;      //Ruptura del caso 20.

case 50:    //Fragmento de código correspondiente al valor 50 de la variable
//Valor.
break;      //Ruptura del caso 50.

default:    //Código correspondiente para cualquier otro valor por defecto.
break;      //Ruptura del default.

}

```

Ciclo iterativo while y do while

El ciclo iterativo while repite o ejecuta un fragmento de programa siempre y cuando la condición contenida en el while sea verdadera, esta sentencia es útil cuando se requiere generar bucles o ciclos que serán rotos hasta que se cumpla el valor condicionante, por ejemplo:

```

short CONT=0;    //Declaración de variable entera que cuenta hasta 100.
while( CONT<100 ) //Declaración del ciclo while.
{
CONT++;          //Incremento de la variable CONT.
}

```

La implementación de un ciclo do while, es similar al ciclo while, con la diferencia puntual de que en este ciclo el fragmento de código se ejecuta y después evalúa la condición del while. En conclusión el código se ejecuta por lo menos una vez antes de evaluar la condición. En el siguiente ejemplo se puede ver la forma de usar este tipo de ciclo:

```

short CONT=0;    //Declaración de variable entera que cuenta hasta 100.
do               //Declaración del ciclo do while.
{ 33
CONT++;          //Incremento de la variable CONT.
} while( CONT<100 );

```

El ciclo iterativo for

Es una estructura parecida al ciclo while, la diferencia es una estructura más compleja en los paréntesis de la condición. La condición cuenta con tres

parámetros, el primero es un espacio de código para dar valores iniciales a una o más variables, el segundo campo permite crear una condición lógica que hace repetir el fragmento de código del for cuando es verdadera, el último espacio realiza cambio sobre una o más variables, tal como: incrementos, decrementos, etc. Las variables que se cambian en el último campo son generalmente las mismas que se inician en el primer campo, aunque no necesariamente es una regla. Su estructura básica es la siguiente:

```
for( _INICIO_ ; _CONDICION_ ; _CAMBIO_ )  
{  
}
```

Para conocer el funcionamiento del ciclo for, se presenta el siguiente ejemplo:

```
short A;           //Declaración de variables.  
short B;  
for( A=0, B=100; (A<100)|| (B>20); A++, B-- ) //Estructura del ciclo for.  
{  
//Fragmento de código que se ejecuta cuando la condición del for es verdadera.  
}
```

Uso anidado de ciclos iterativos

Muchas de las aplicaciones incrementan su funcionalidad y optimizan el tamaño del código de máquina final al usar los ciclos iterativos de forma anidada. La anidación de ciclos consiste en ejecutar un ciclo dentro de otro. Un ciclo puede contener uno o más ciclos anidados, para comprender este concepto de mejor manera se ejemplifica de la siguiente forma.

```
short COL;         //Declaración de variables.  
short FIL;  
short MATRIZ[10][10]; //Declaración de un vector bidimensional.  
for( COL=0; COL<10; COL++ ) //Declaración de ciclo for.  
{  
//Fragmento del primer ciclo.  
for( FIL=0; FIL<10; FIL++ ) //Declaración de ciclo for anidado.  
{  
//Fragmento del ciclo anidado.  
MATRIZ[COL][FIL] = COL * FIL;  
}  
}
```

El ejemplo anterior, guarda valores en una matriz de 10 por 10, recorriendo una a una las posiciones de la misma.

Con todo lo visto previamente es posible desarrollar un programa básico en Lenguaje C, por ello se mostrará la estructura de un programa.

Primero es necesario declarar variables globales que el programador tenga consideradas para el desarrollo del programa, estas variables declaradas inicialmente serán reconocidas en todo el programa.

El siguiente paso es declarar la función main(), declarando las variables que se requieran en la misma. Posteriormente se deben poner el código que deben inicializar puertos y módulos del microcontrolador que le permitan identificar entradas y salidas.

En el siguiente programa es un ejemplo básico.

```
int Variable A;           //Declaración de variables
char VariableB;
float VariableC;
//Declaración de funciones propias del desarrollador.
float Valor_PI ( void )
{
float PI=3.1416;
return PI;
}
float Calculo_Area_Circulo( float Radio )
{
float Area;
Area = Valor_PI()*Radio*Radio; //Cálculo del área.
return Area;
}
void main( void )        //Declaración de la función main o principal
{
float Valor_Radio;      //Declaración de variables de la función main.
float Valor_Area;
TRISB=0;                //Configuración de puertos y módulos.
TRISC=0;
PORTB=0;
PORTC=0;
while( 1 )              //Código concreto del aplicativo.
```

```
{  
Valor_Area=Calculo_Area_Circulo( Valor_Radio );  
}  
}
```

La información previamente vista es la base para cualquier programa, es posible desarrollar un sistema tan complejo como se desee, el objetivo primordial es dar a conocer las principales herramientas que componen a un programa en lenguaje C que permitan entender todo aquello que se muestra en este proyecto.

3.5 SIMULADOR PROTEUS PARA MICROCONTROLADORES PIC

ISIS DE PROTEUS VSM

El entorno del diseño electrónico PROTEUS brinda la posibilidad de simular circuitos electrónicos, esto permite el diseño tanto a nivel hardware como software. Para ello se utilizan tres subentornos como son **ISIS** para el diseño gráfico, **VSM (Virtual System Modeling, Modelado de Sistema Virtual)** para la simulación y **ARES** para el diseño de placas. Las herramientas a utilizar serán únicamente ISIS y VSM dejando ARES para la realización de futuras aplicaciones del proyecto.

ISIS es un programa especializado en el diseño electrónico que realiza esquemas para ser simulados en el entorno VSM o pasados en un circuito impreso en el entorno ARES, posee un entorno de trabajo formado por varias librerías de modelos tanto para dibujar y/o simular el circuito generando su respectiva placa. Además permite la creación de nuevos componentes, su modelización para su simulación e incluso, la posibilidad de solicitar un nuevo modelo.

En el caso de los microcontroladores existe una valiosa forma de simularlos, cargando en el microcontrolador el archivo de programa .HEX generado en la compilación en el programa MIKRO C. También es posible modificar parámetros del circuito como es la frecuencia de reloj (lo que nos evita el uso de cristales externos en la simulación), memorias de programa, la memoria de datos RAM o la EEPROM, los registros especiales (FSR) y la pila (STACK).

Una de las principales características de PROTEUS VSM es la capacidad de depurar programas fuente de distintos lenguajes de programación. Además de aceptar el archivo Intel Hex. También admite ficheros IAR UBROF (D39), Byte-Craf COD (COD), Microchip Compatible (COF) y Crownhill Proton Plus (BAS), que pueden abrirse mediante la fuente llamada SOURCE CODE, mediante el cual se puede seguir el programa fuente línea a línea de código, es decir, en caso de ser necesaria una modificación del programa fuente (en nuestro caso *.HEX), no será necesario hacerlo en MIKRO C.

Una vez cargado el microcontrolador con el programa fuente se puede proceder a la simulación teniendo las opciones MARCHA, PASO A PASO, PAUSA Y PARADA.

MARCHA: La simulación se inicia y funciona en modo continuo no siendo en tiempo real pues depende de la carga de trabajo de la PC, debido a esto, PROTEUS tiene una opción muy útil que nos indica la carga de la CPU del PC (a mayor carga menos real será la simulación) y el tiempo de ejecución; este tiempo es el que tardaría, en la realidad, el circuito en realizar un proceso (por ejemplo

esto implica que, dependiendo de la carga de la CPU, un tiempo de un segundo en el circuito, puede significar varios minutos en la simulación)

STOP: La opción STOP para totalmente la simulación, mientras que PAUSE la detiene momentáneamente, hace uso de las herramientas de depuración.

La opción PASO a PASO permite trabajar en tramos de tiempo predefinidos y, además permite utilizar las herramientas de depuración. Esta opción está ligada a la configuración de la animación.

Al producirse una pausa el menú DEBUG se modifica, mostrando las correspondientes herramientas de depuración.

Herramientas de depuración:

- SIMULATION LOG: Mensajes resultantes de la simulación.
- WATCH WINDOWS: Ventana de visualización de posiciones de memoria. Permite añadir lo que se desee ver.
- PIC CPU REGISTERS: Muestra los registros FSR del PIC.
- PIC CPU DATA MEMORY: Muestra la memoria de datos (RAM)
- PIC CPU EPROM MEMORY: Muestra la memoria de datos EPROM
- PIC CPU PROGRAM MEMORY: Muestra la memoria del programa.
- PIC CPU STACK: Muestra la pila.

La ventana WATCH windows es la más versátil puesto que pueden añadir variables y modificar su visualización. Estas herramientas serán las bases para poder visualizar, modificar y simular el circuito deseado.

Características básicas para la simulación:

El entorno de trabajo del simulador contiene diversas herramientas en sus menús y sub menús, tales como barras de archivo, vista, edición y diseño. En la parte izquierda se encuentra la paleta de dispositivos, una vez seleccionado despliega una gran cantidad de dispositivos electrónicos (ver figura 3.5.1), esta gran variedad hace que PROTEUS sea una poderosa herramienta para la simulación de circuitos y es la razón por la que fue elegida para el desarrollo del Sistema.

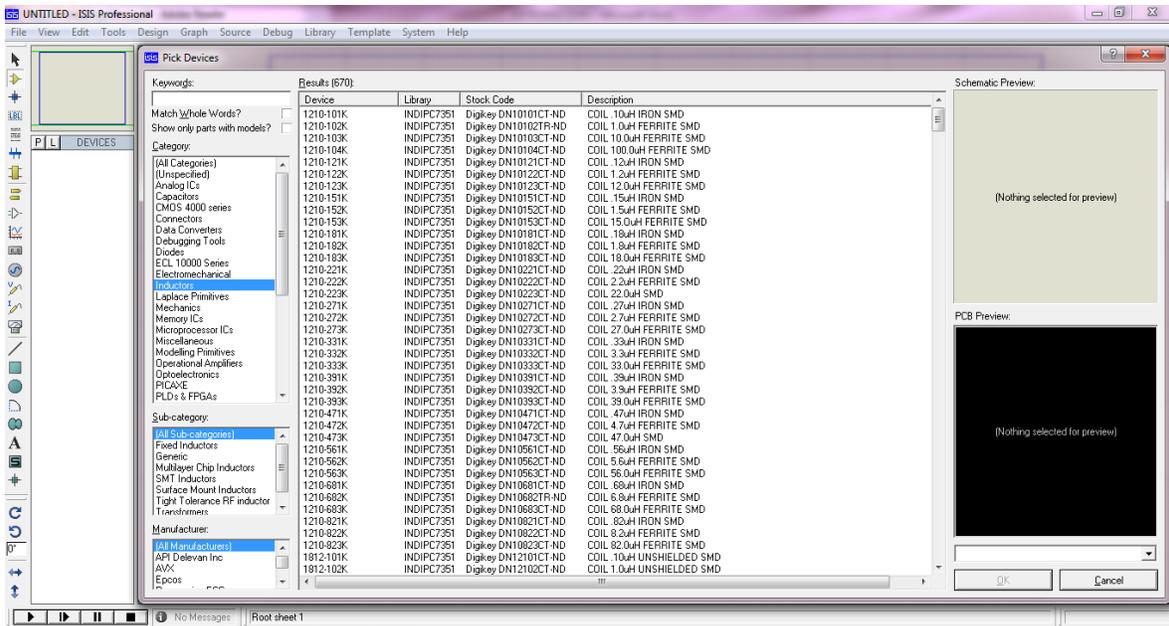


Figura 3.5.1. Dispositivos disponibles en PROTEUS

El siguiente paso para la simulación, es la elección de dispositivos adecuados que permitan plasmar el circuito electrónico deseado, ello implica las fuentes, tierras y aparatos de medición, el circuito simulado debe contener los mismos elementos que un circuito real. En el siguiente ejemplo se muestra un circuito que emplea un microcontrolador, figura 3.5.2:

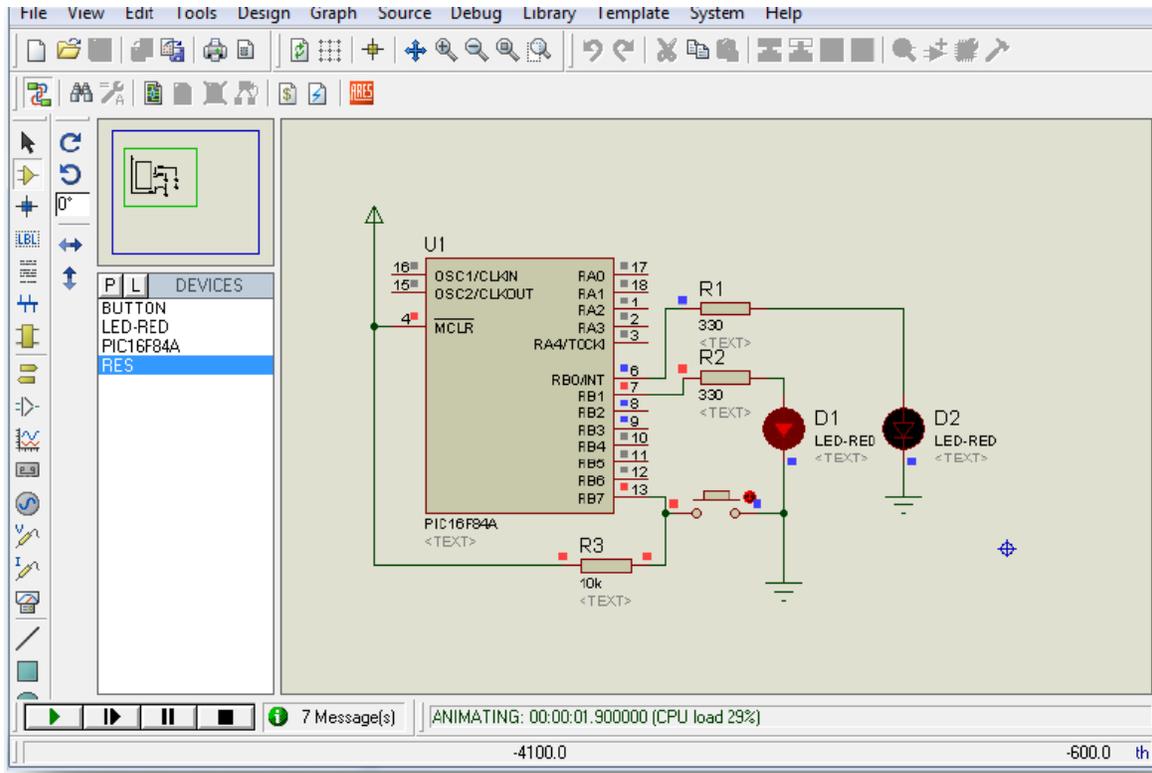


Figura 3.5.2 Diseño de circuito electrónico

Antes de comenzar la simulación el PIC debe tener el archivo .hex que es el programa desarrollado por el programador, para simular todo el sistema. Haciendo click con el botón derecho del mouse sobre el PIC y seleccionando la opción “edición del componente” es posible ingresar el archivo para su correcta simulación. Ver figura 3.5.3.

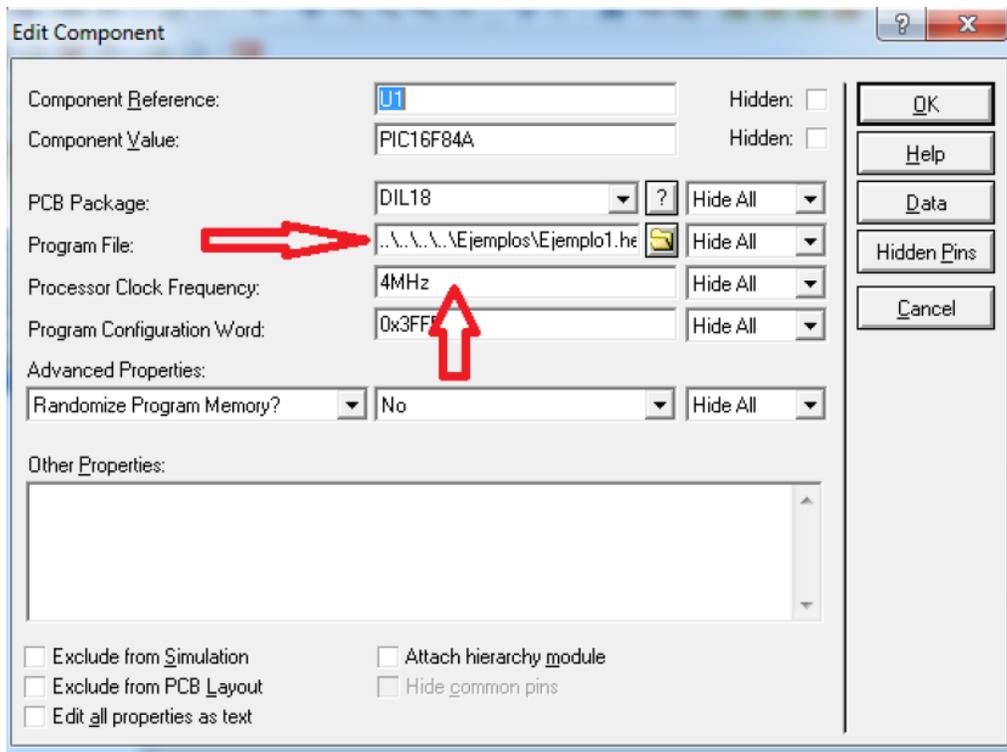


Figura 3.5.3. Edición de parámetros

Una vez cargado el archivo, se puede proceder a la simulación y para poder lograrlo se busca la barra mostrada en la figura 3.5.4.

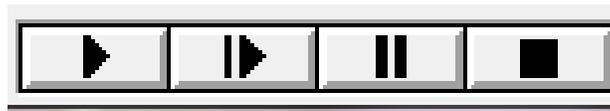


Figura 3.5.4 Iconos de simulación

El icono ► (play) se utiliza para ejecutar la simulación inmediatamente, por ende los demás detienen o pausan según sea el caso. Una de las cosas más importantes en la simulación es que para obtener resultados, lo más cercanos posible a la realidad, es necesario tener la computadora en el estado de mayor reposo, es decir, no hacer uso del procesador pues entre mayor sea el consumo de los recursos de la computadora en el momento de la simulación, mayor será la variación de valores simulados. [L09]

CAPÍTULO 4



SISTEMA SIMULADOR

4.1. DESCRIPCIÓN GENERAL.

El siguiente capítulo se encarga de detallar todos los elementos que conforman las etapas correspondientes al Sistema Simulador, tal y como se muestra en el diagrama a bloques de la figura 4.1.1, para poder obtener un electrocardiograma (ECG) patrón que tenga todas las características de uno real:

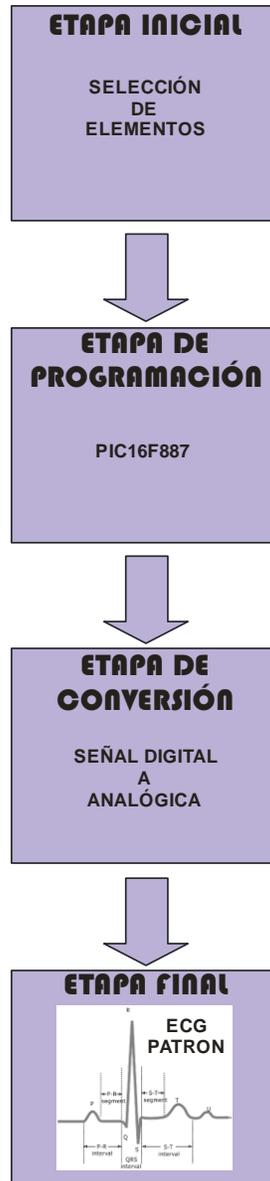


Figura 4.1.1. Diagrama a bloques del Sistema Simulador

La forma en que el Sistema Simulador se llevó a cabo se describe en los siguientes puntos:

- Se obtuvo un ECG real de una persona común, es decir, con una frecuencia cardíaca de 60 pulsos por minuto.
- Se digitalizó el ECG real por medio de un muestreo considerando dos variables involucradas en cualquier ECG, el voltaje y el tiempo.
- Introducir los datos obtenidos por el muestreo al microcontrolador, que a su vez digitalizó esos datos.
- Los datos digitales son convertidos a analógicos para que sean vistos en un osciloscopio para comprobarlo y ver una señal eléctrica similar a un ECG real, donde se podrá apreciar, en esa señal principal, la variación de voltaje en función del tiempo.
- Utilizar un arreglo de resistencias para la señal principal, donde el ECG patrón se pueda descomponer en las señales que proporciona un electrocardiógrafo, dichas señales son conocidas como derivaciones, ya descritas en el capítulo 2.
- El ECG patrón, que proporciona el Sistema Simulador, puede ser verificado por medio de una simulación, donde todos los elementos están incluidos.

Cabe señalar que una vez que se seleccionó un elemento, se puso mucho cuidado en considerar el aspecto económico, además de colocarlo y configurarlo de acuerdo a todas las recomendaciones señaladas en su hoja de datos. Para proporcionar una mejor referencia y justificación, se colocará en el Apéndice A las hojas de datos utilizadas en el Sistema Simulador.

El Sistema Simulador se conforma de los siguientes elementos:

- Microcontrolador PIC16F887
- Convertidor digital – analógico DAC0800L
- Amplificadores operacionales
- Resistencias (o resistores)
- Capacitores
- Otros
- Hardware, programador PICKit 2
- Software, compilador Mikro C
- Software, simulador ISIS de PROTEUS

Existen varios tipos de elementos plasmados en la lista anterior, los electrónicos, los de software, el hardware y otros que fueron empleados para la programación del microcontrolador, el diseño y la simulación de todo el Sistema. En la tabla 4.1.1 se colocaron los precios de una tienda de electrónica, se consideró el costo más accesible [1]:

¹ Precios consultados el cuarto mes del año en curso, y no incluyen IVA

Elemento	Valor	Cantidad	Precio [\$]
Micro controlador de 8 BIT	PIC16F887	1	43.103
Convertidor digital/analógico 8 BIT	DAC0808	1	23.31
Amplificador Operacional	LF351	1	7.559
Amplificador Operacional	TL081	1	6.897
Regulador de voltaje 5V	7805	1	0.455
Convertidor de voltaje	ICL7660	1	22.414
Resistencia de carbón, 1/4 watt, 5%	10[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	4.7[KΩ]	3	0.862
Resistencia de carbón, 1/4 watt, 5%	470[Ω]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	100[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	47[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	27[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	22[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	2.2[KΩ]	2	0.862
Resistencia de carbón, 1/4 watt, 5%	3.9[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	1[KΩ]	2	0.862
Resistencia de carbón, 1/4 watt, 5%	10[Ω]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	8.2[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	1.8[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	56[KΩ]	1	0.862
Resistencia de carbón, 1/4 watt, 5%	33[KΩ]	1	0.862
Capacitor electrolítico	0.33[μF]	1	1.724
Capacitor electrolítico	0.1[μF]	2	1.724
Capacitor electrolítico	10[μF]	2	1.724
Capacitor cerámico	0.1[μF]	1	0.862
Capacitor cerámico	0.001[μF]	1	1.724
Capacitor de tantalio	100[nF]	2	9
Capacitor electrolítico	2200[μF]	1	16
Led	Rojo	1	1.724
Diodo rectificador	1N4002	2	2
Diodo rectificador	1N4148	3	2
Diodo Zener de 12 [V]	BZX55C	1	5
Transistor	2N3055	1	44
Preset Horizontal	1[KΩ]	2	4.31
Preset Horizontal	5[KΩ]	1	4,31
Preset Horizontal	250[KΩ]	1	4,31
Batería	9[V]	1	26
Micro switch, de push	2 terminales NA	1	4.31
Transformador de 12 [V]	Transformador	1	89

Tabla 4.1.1. Elementos con costos del Sistema Simulador

4.2. ETAPA INICIAL DEL SISTEMA

Ya con todos los elementos seleccionados anteriormente, resulta primordial conocer las bases para el manejo de los programas que se utilizaron. Una característica que comparten dichos programas es que pertenecen a una versión estudiantil y para los fines del Sistema Simulador resultan suficientes.

A continuación se describe la primera parte de la figura 4.1.1, que corresponde a la Etapa inicial, mostrando los argumentos para la selección de los elementos que conforman al Sistema Simulador. Todos los elementos electrónicos (resistencias, capacitores y amplificadores operacionales), que se utilizaron en el diseño, fueron seleccionados en base a la Hoja de datos del PIC16F887, el DAC0800, tal y como se hace referencia más adelante en su hoja de datos correspondiente, donde se indican **Typical applications (aplicaciones típicas)** que recomiendan los fabricantes.

Un elemento principal del Sistema Simulador es el PIC16F887; se trata de un microcontrolador, es decir, cuenta con un CPU, memoria RAM y ROM y puertos que pueden ser empleados como entrada o salida. El PIC16F887 lo fabrica la empresa Microchip desde el 2007. En el Apéndice A se encuentra el link de la hoja de datos correspondiente al PIC16F887. Las principales características del microcontrolador son:

- La empresa cuenta con tres familias de microcontroladores, de acuerdo a sus características y funciones, que se dividen en PICs de 8, 16, y 32 bits. El PIC16F887 se encuentra en la primera parte, pero es de los que cuenta con más herramientas. En la figura 4.2.1 ^[2] se observan las familias de los PICs.

² La figura 4.2.1 está basada en la proporcionada por Microchip.

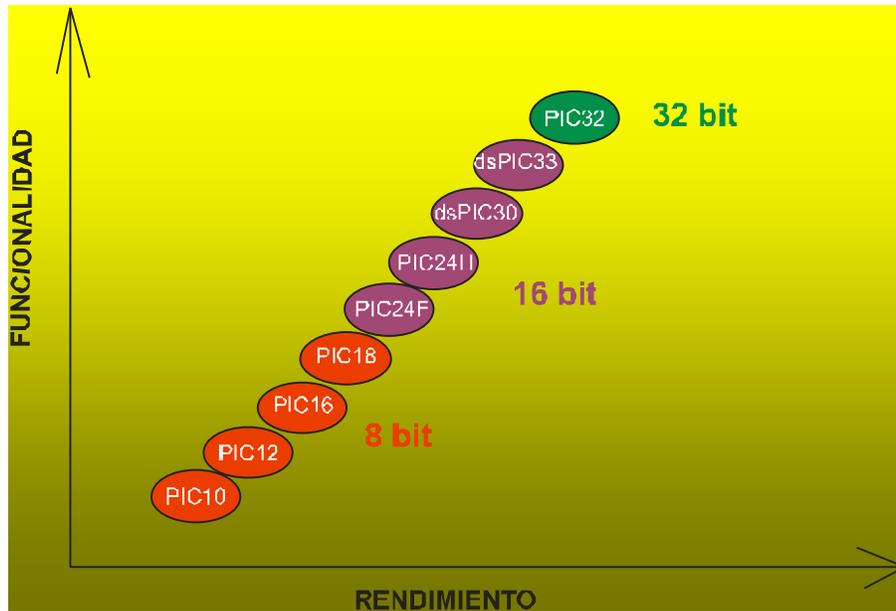


Figura 4.2.1. Familia de PICs de Microchip

- El microcontrolador viene en un encapsulado PDIP de 40 pines, tal y como se aprecia en la figura 4.2.2:



Figura 4.2.2. PIC16F887

- El PIC puede excitar directamente leds, debido a que cuenta con una corriente elevada, de tal forma que no se necesite de algún otro elemento electrónico para que se conecten terminales del led a las terminales correspondientes del microcontrolador, tal y como se muestra en el ejemplo de la figura 4.2.3, donde algunos leds están conectados al puerto D del microcontrolador:

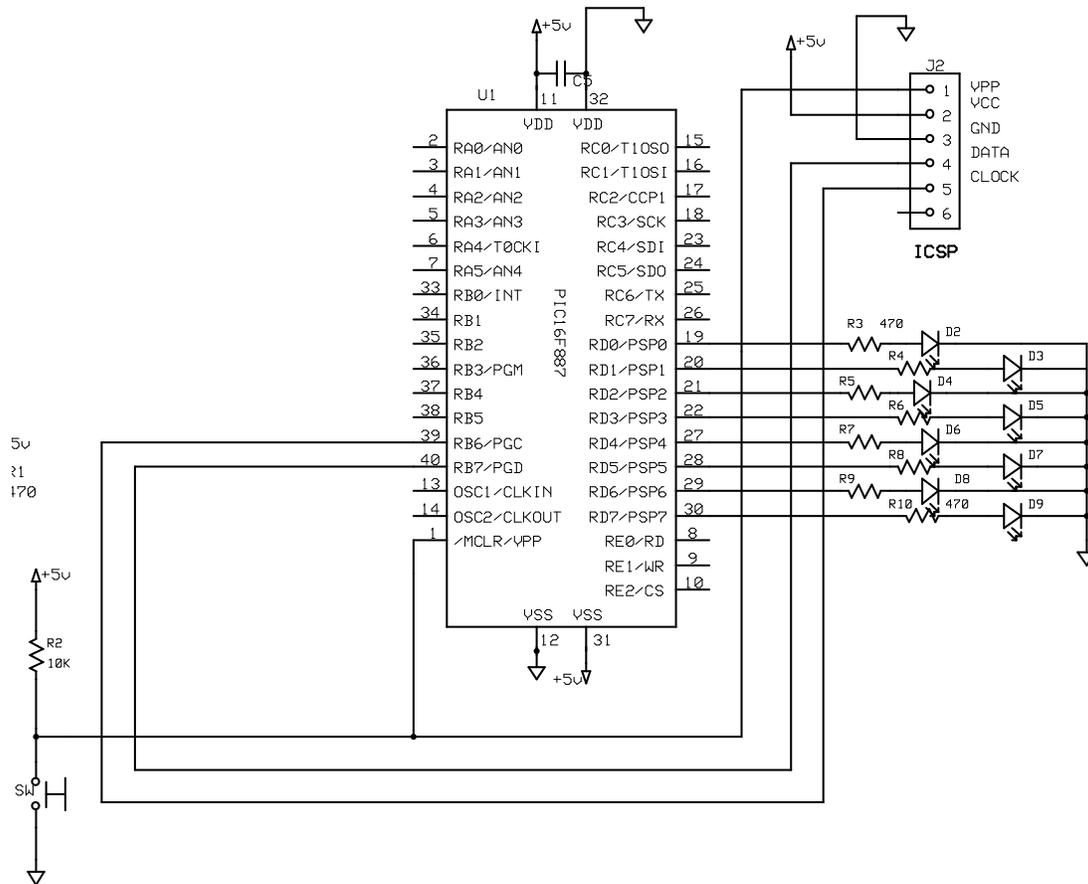


Figura 4.2.3. Leds conectados al puerto D del PIC

- Cuenta con un convertidor analógico – digital de 10 bits de resolución, lo cual indica que se pueden utilizar hasta 2^{10} , es decir, 1024 valores. Como por ejemplo si se requiere digitalizar alguna variable física, como algún voltaje y existe un rango de 0 a 5 [V], los 10 bits de resolución indican que se pueden realizar muestreos desde 0 hasta 5 [V], cada $5/1024$ [V].

- Maneja una arquitectura Harvard, lo que indica que tanto la memoria de programa y la de datos se encuentran en buses diferentes, esto implica una mayor velocidad de acceso, tanto para la escritura como para la lectura.
- Trabaja con instrucciones RISC, se requieren pocas instrucciones para programar un PIC. El PIC puede ser reprogramado hasta 100, 000 veces.
- El PIC16F887 puede adquirirse en muchos lugares del país, además de contar con un bajo costo, y poder conseguir elementos para su programación y captura del programa para el PIC, como: ejemplos, aplicaciones, etc. [L09]

La señal que se consideró como referencia de un ECG real o principal se muestra en la figura 4.2.4:

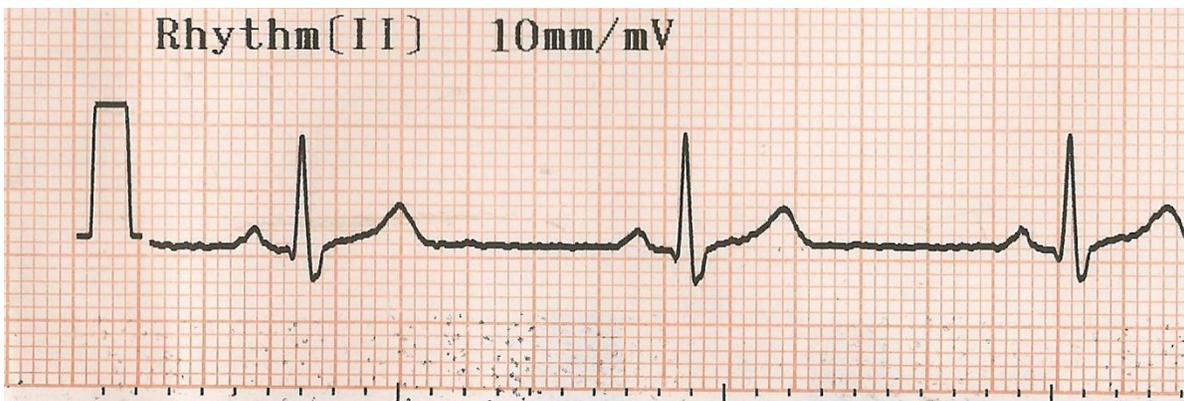


Figura 4.2.4. ECG con 60 ppm

En la figura anterior se observa el ECG de una persona con 60 latidos por minuto, en papel milimétrico, donde: un milímetro en la parte horizontal corresponde a 0.04 segundos y cada milímetro vertical corresponde a 0.1 milivolts. Los Médicos pueden distinguir el dato de los pulsos por minuto de dos formas, la primera opción es revisar el dato que arroja el ECG; otra forma es hacer una división de 300 entre los cuadros grandes de 5 milímetros que haya de separación entre dos ondas R consecutivas, las ondas R son la que tienen la cresta más grande, aproximadamente existen 5 cuadros de separación entre ellas, por lo tanto se trata de un paciente con 60 pulsos por minuto.

4.3. PROGRAMACIÓN DEL PIC16F887

Para la programación del PIC se utilizó el compilador Mikro C. Es un programa sencillo de manejar, el cual proporciona un ambiente para editar programas en C, y compilarlos. Existen muchas herramientas que facilitan y apoyan al usuario al escribir un programa en C, en las siguientes líneas se irán describiendo las herramientas mencionadas. En la figura 4.3.1 aparece la ventana principal del compilador:

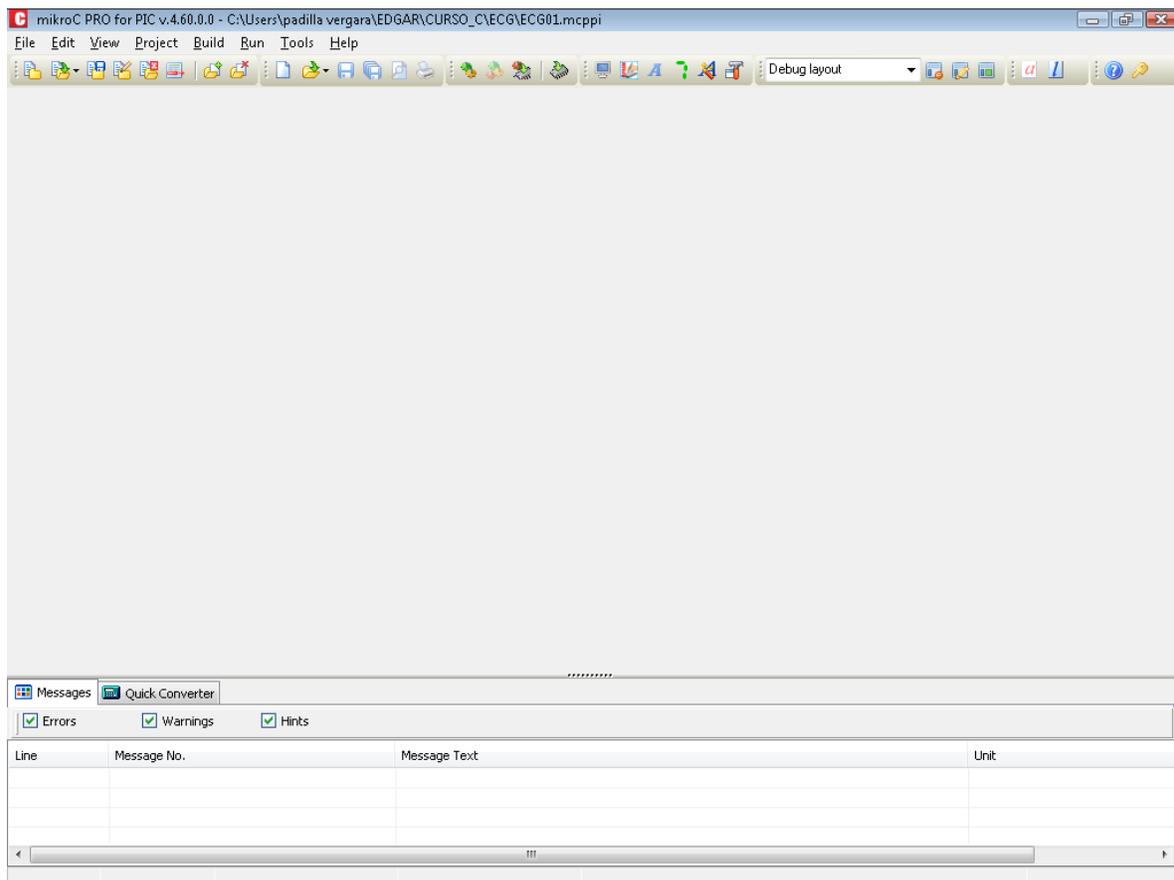


Figura 4.3.1 Ventana principal Mikro C

Cuando se abre el programa se selecciona File, luego New y enseguida New Project, aparecerá la siguiente pantalla, tal y como se muestra en la figura 4.3.2:



Figura 4.3.2 Pantalla de bienvenida: New project

Ahora se selecciona Next, y se desplegará la siguiente pantalla con 6 pasos a seguir con las indicaciones para comenzar con un nuevo programa:

1. La figura 4.3.3 es el paso 1, donde se selecciona el microcontrolador en el que se va a programar, en este caso el PIC16F887, en la pantalla no aparece completo, solo aparece P16F887, y se selecciona Next:
2. En la figura 4.3.4 aparece una pantalla con el paso 2, en ella se selecciona el valor para el reloj interno del PIC, 8 mega hertz y se selecciona Next:
3. Como se observa en la figura 4.3.5 se elige la dirección del archivo donde se guardará el programa y se selecciona Next:

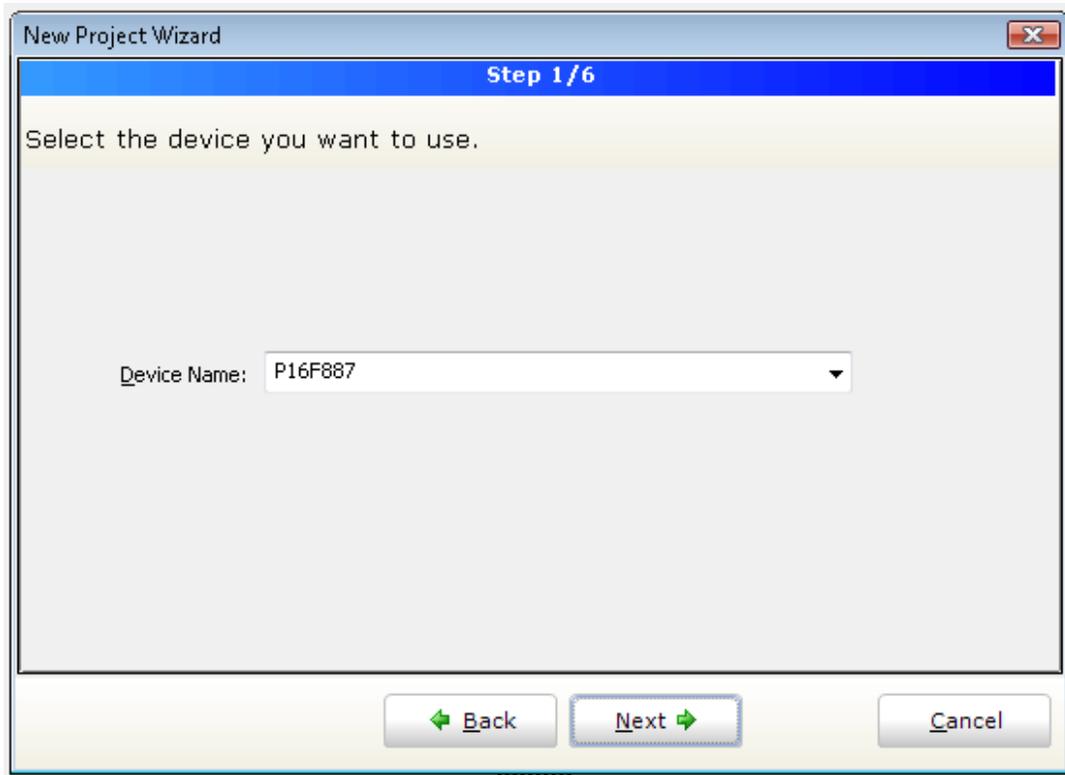


Figura 4.3.3. Paso 1

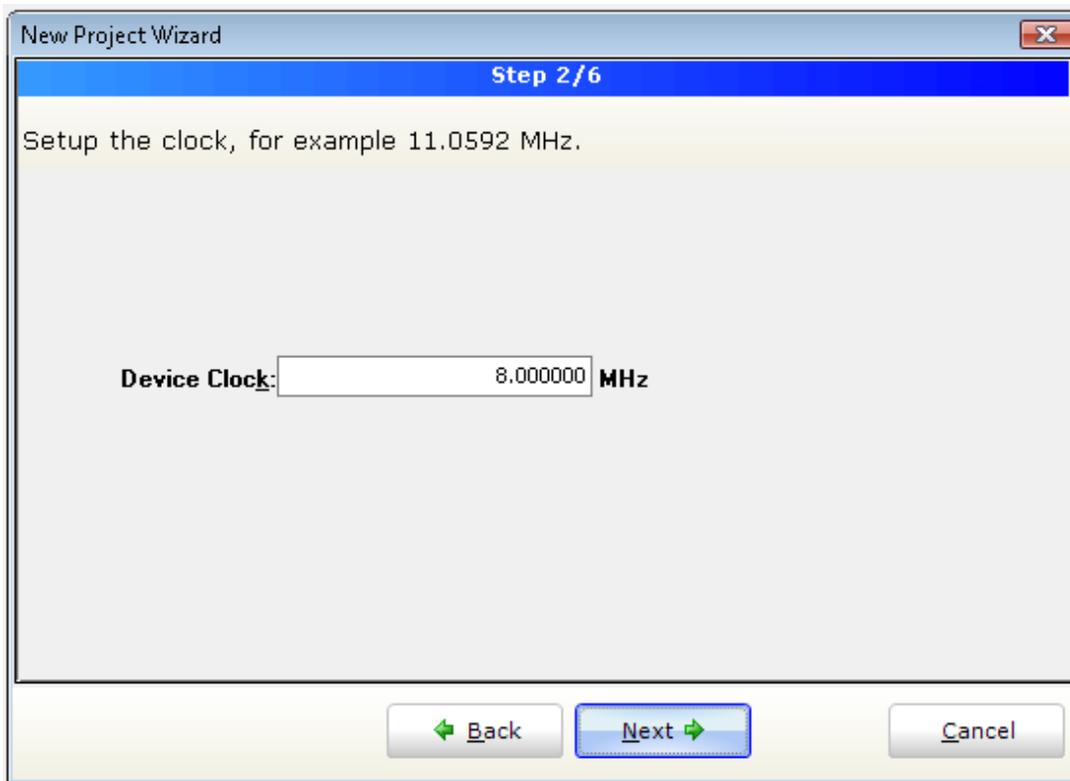


Figura 4.3.4. Paso 2

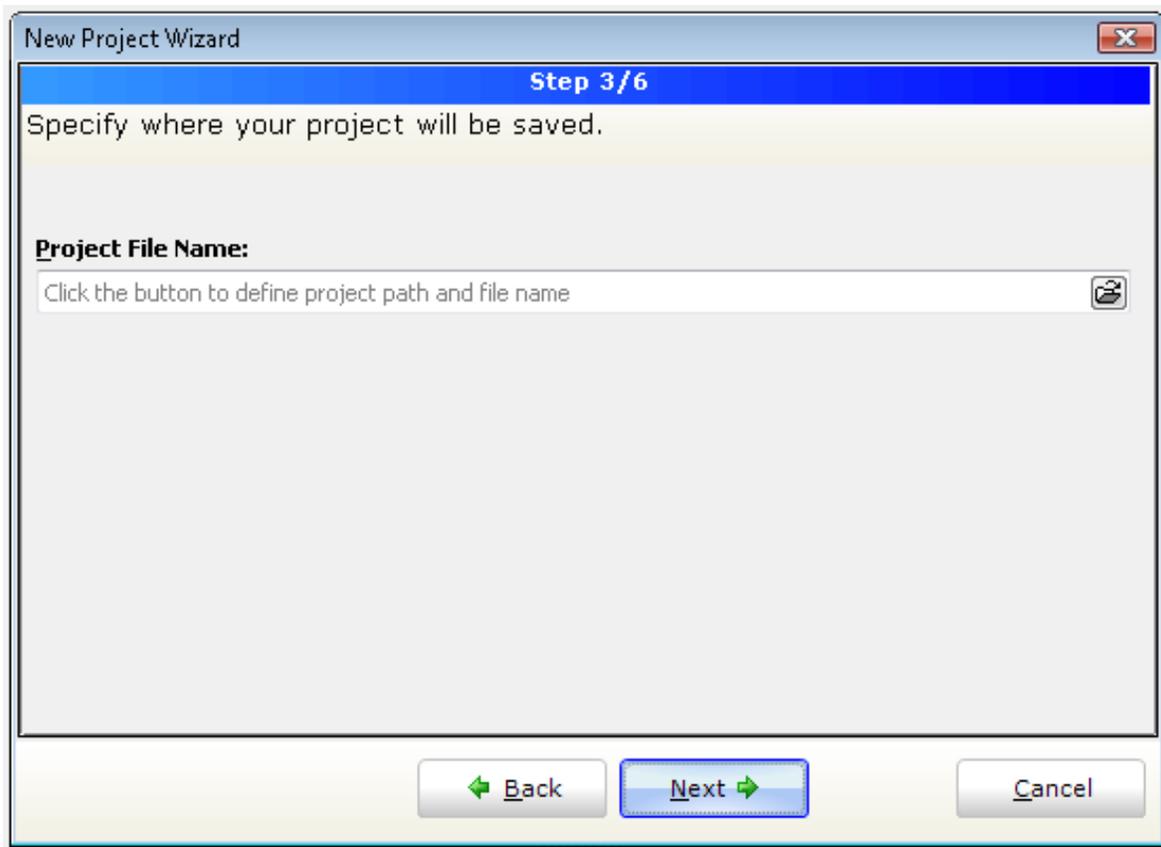


Figura 4.3.5. Paso 3

4. El cuarto paso se utiliza sólo para el caso donde existan más proyectos que se requieren para el programa o que interactúen, en el caso del presente programa no se necesita de otro proyecto, por lo tanto no se adiciona nada y se selecciona Next, como aparece en la figura 4.3.6:
5. En el quinto paso es recomendable elegir el uso de todas las librerías, tal y como se muestra en la figura 4.3.7:
6. El último paso solo es para dar inicio al nuevo proyecto, es decir, comenzar con el programa, a menos que se haya cometido un error en los pasos anteriores, se elige Finish, figura 4.3.8:

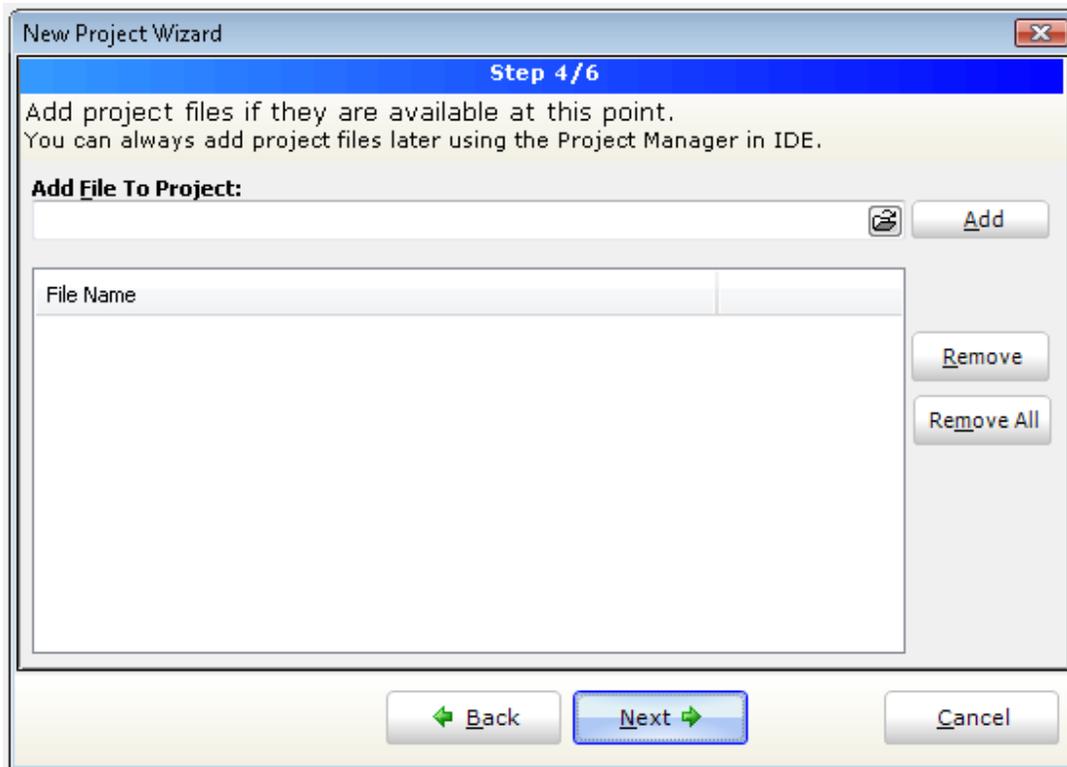


Figura 4.3.6. Paso 4

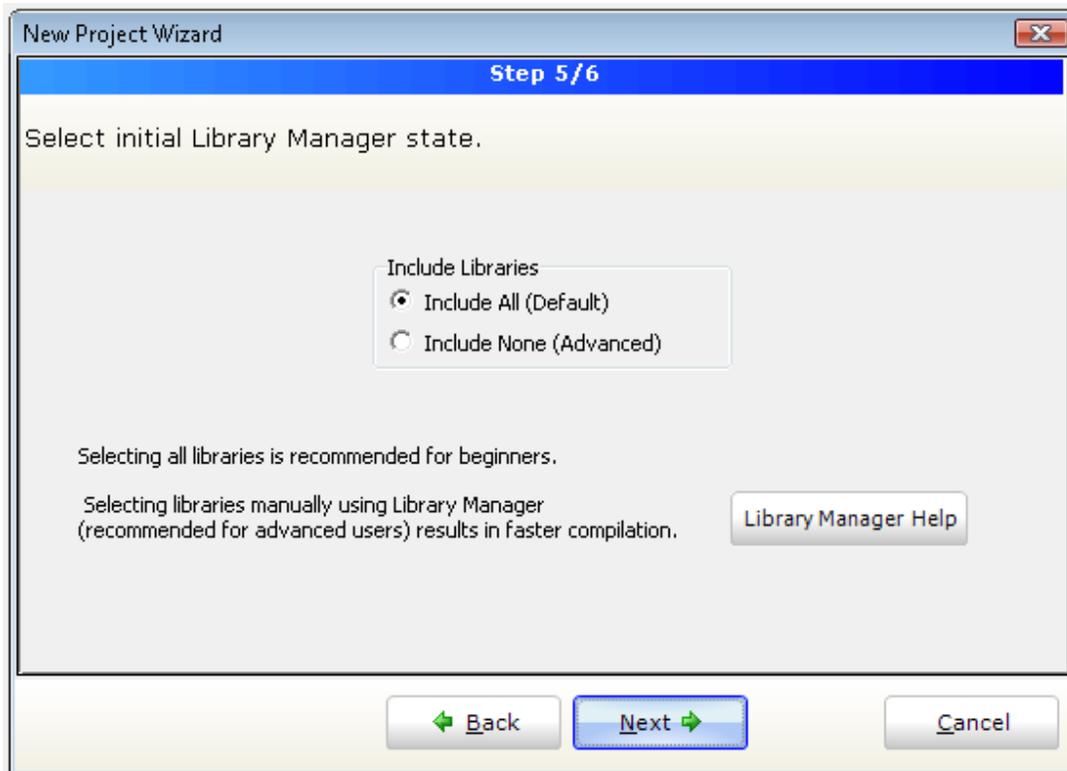


Figura 4.3.7. Paso 5

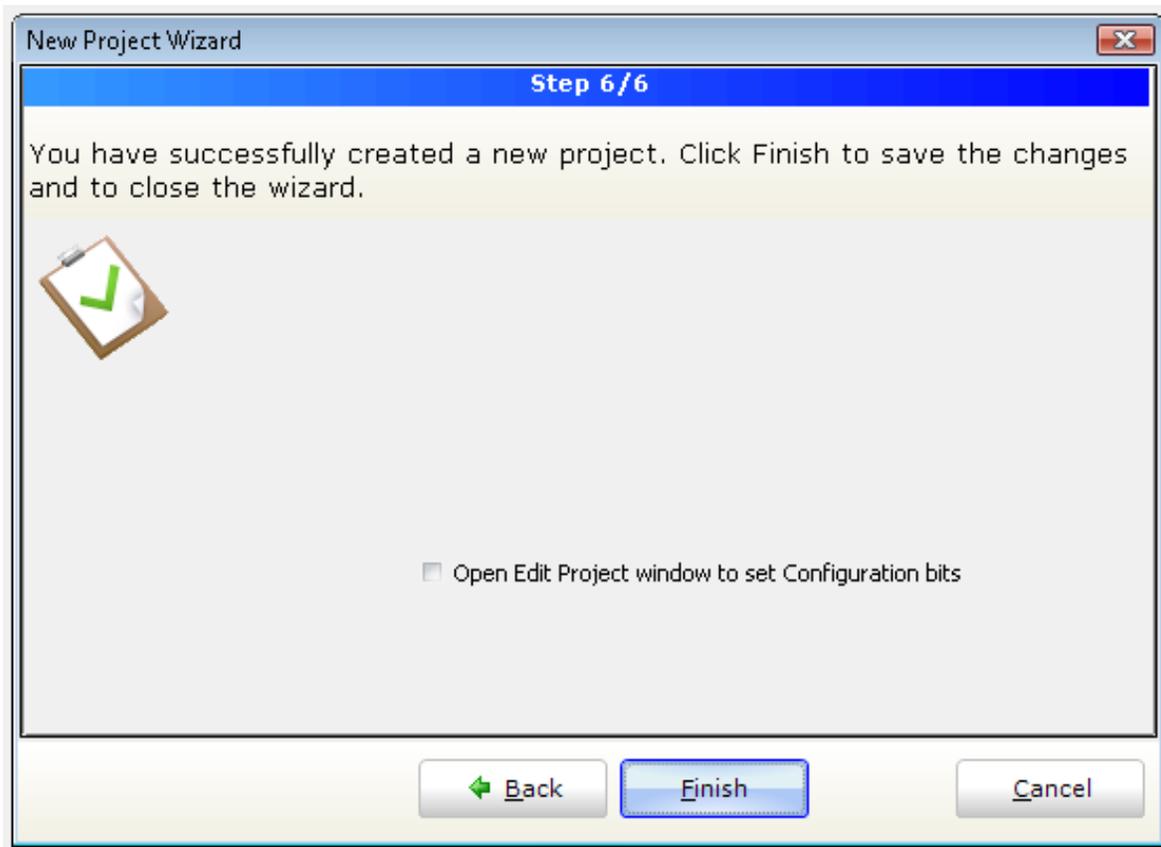


Figura 4.3.8. Paso 6

El nuevo proyecto está listo para utilizarse, tal y como se muestra en la figura 4.3.9, donde se aprecia la pantalla para teclear las instrucciones:

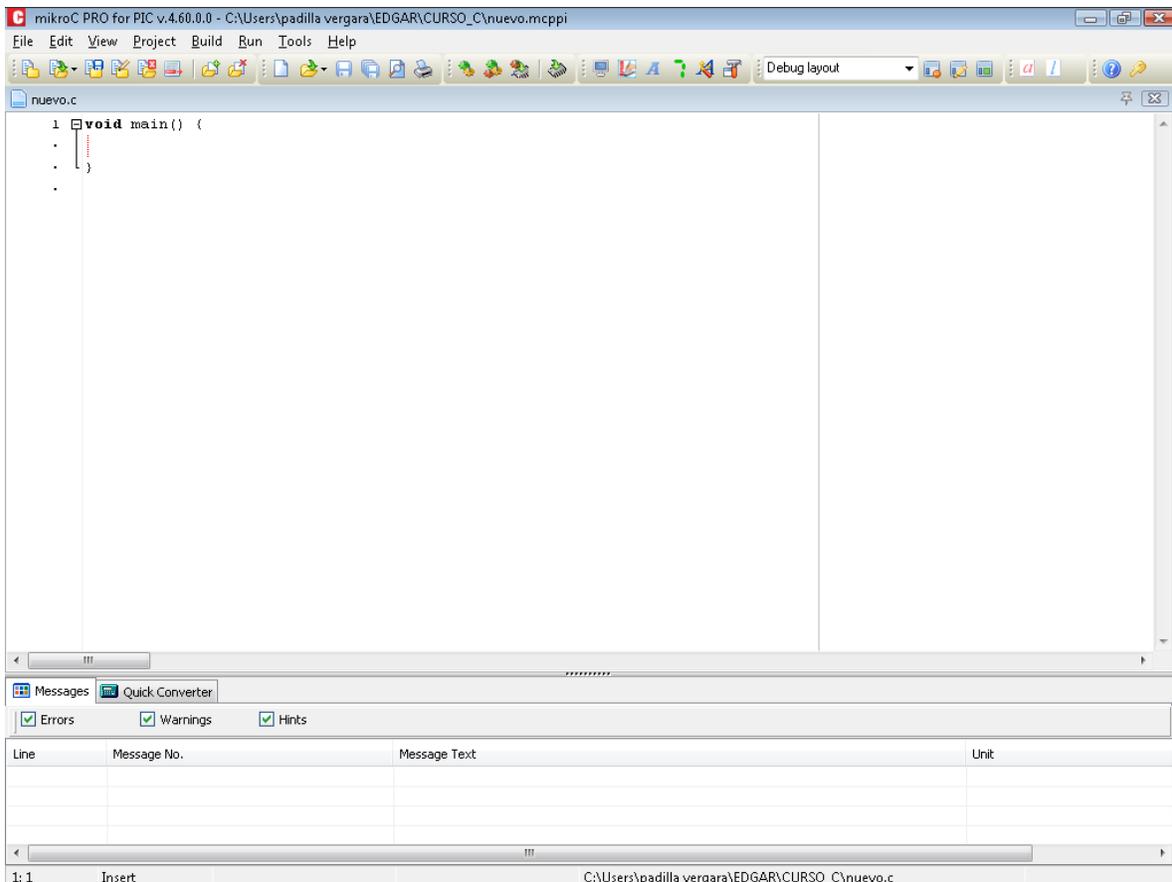


Figura 4.3.9. Nuevo programa Mikro C

El programa para generar la señal ECG patrón tiene la siguiente estructura:

```

// Definición de variables globales

// Definición de funciones

void main(void)

{

    // Definición de variables locales

    // Configuración de registros (recursos y puertos)

    // ciclo infinito

while ( 1 )

```

```

        {
            // Programa de usuario
        }
    }

```

El diagrama de flujo del programa en Mikro C se encuentra en el Apéndice B.

A continuación se mencionan algunas instrucciones y detalles que se emplearon para el programa.

- unsigned int: define variables locales enteras sin signo o positiva y puede tener valores de 0 a 65535, y con un tamaño de 2 bytes.
- unsigned char: define variable local de 0 a 255 y tiene un tamaño de un byte.
- const unsigned char: define un arreglo de caracteres constantes sin signo.
- ANSEL: define cual de las primeras ocho entradas se configuran como entradas analógicas, se inicializa con 0 si no se utilizan y 1 cuando sí son entradas.
- ANSELH: define cual de las últimas entradas se configuran como entradas analógicas. Esta instrucción, junto con la anterior son indispensables para configurar el Puerto A y E como entrada o salida de datos digitales.
- C1ON_bit: deshabilita comparador 1.
- C2ON_bit: deshabilita comparador 2.
- TRISE: registro tri-estado para configurar el Puerto E, entrada/salida.
- PORTC: configuración del Puerto C
- PORTD: configuración del Puerto D

- PORTE: configuración del Puerto E
- PORTD.F1: para tener acceso individual para las variables de 8 bits, utilizando .F0, .F1,...,F7, siendo F0 el bit menos significativo
- TRISD: registro tri-estado para configurar el Puerto D, entrada/salida.
- OSCCON: es un registro que sirve para controlar el oscilador, ya sea interno o con un cristal externo. Tiene 8 bits que se configuran como sigue en forma binaria: el bit 7 se pone en 0 porque no tiene implementación; bit 6, 5 y 4 con 1 porque se trabaja con la frecuencia de 8 [MHz]; el bit 3 con 1 ya que se utilizaron los tres bits anteriores; bit 2 con 1 porque se trabaja con una frecuencia que oscila en el rango de 8 [MHz] a 125 [kHz]; el bit 1 con 1, para que las bajas frecuencias sean estables; y el bit 0 con 1 porque el oscilador interno se utiliza para el reloj del microcontrolador. De tal forma que en número binario se tiene 01111111, lo que equivale a 7F en hexadecimal.
- Delay_ms: se utiliza para poner un retardo en milisegundos.
- PORTD.F1^1: cambia el estado anterior del bit 1 del Puerto D, si estaba prendido lo apaga y viceversa. [L03]

Una vez ya listo el programa en C y compilado satisfactoriamente, lo siguiente es pasarlo al microcontrolador. En el Apéndice C aparece el programa completo. Mikro C, después de compilar el programa, guarda varias extensiones del programa, la c, cfg, cp, dbg, dct, dlt, y hex, esta última será la que se coloca en el software de simulación ISIS de PROTEUS, ver figura 4.3.10:

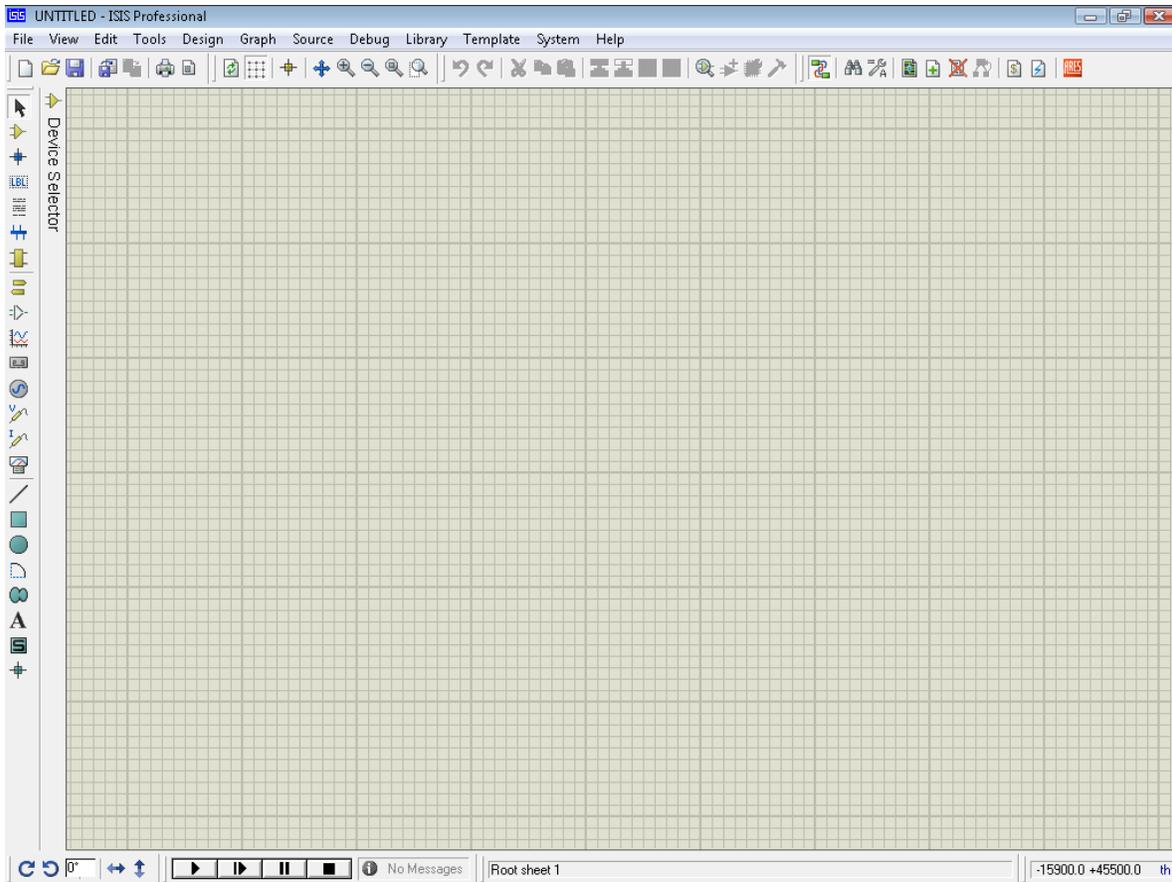


Figura 4.3.10. Ventana principal ISIS de PROTEUS

Cuando se abre la ventana principal se comienzan a colocar todos los elementos que se van a emplear para el Sistema, tal y como se muestra en la figura 4.3.11.

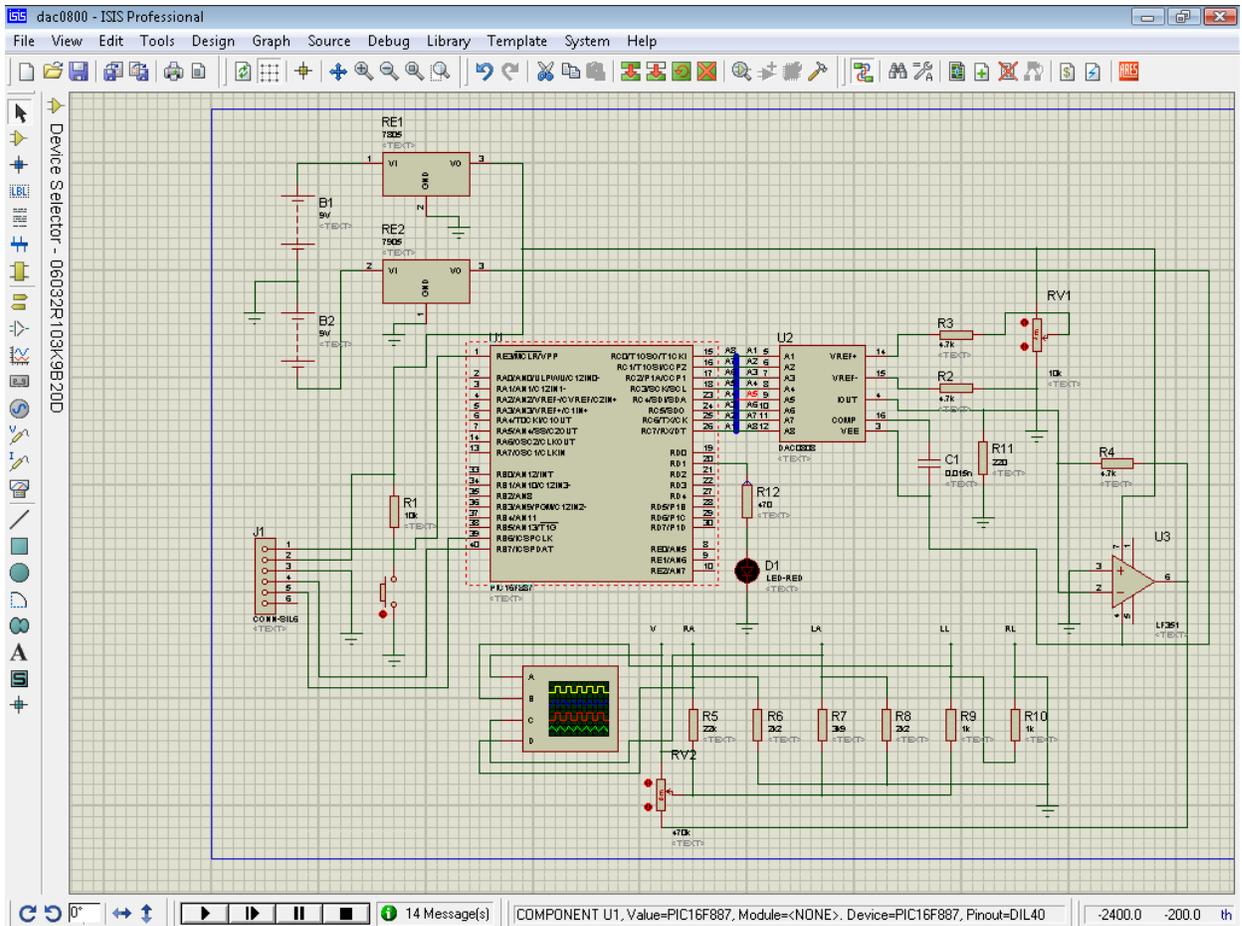


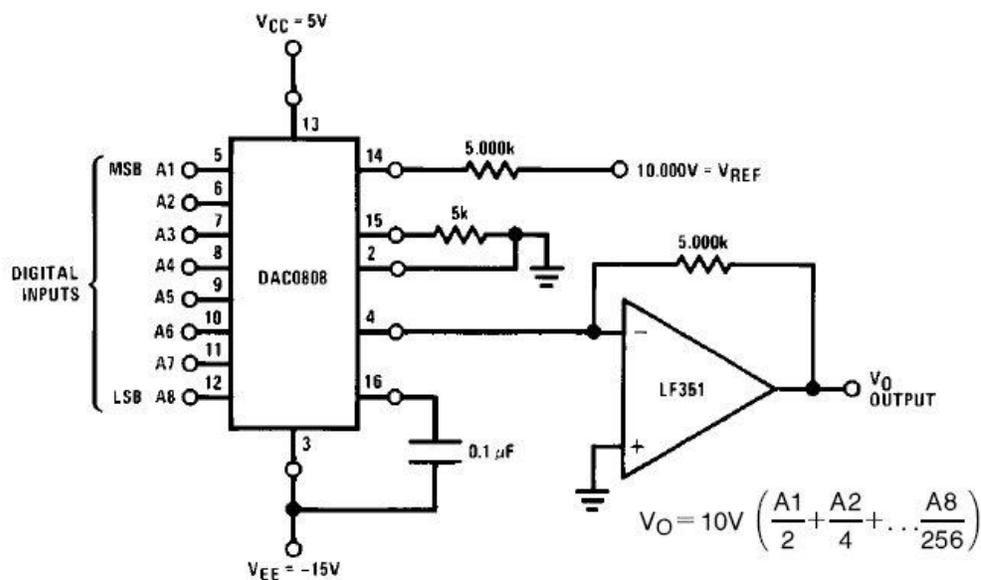
Figura 4.3.11. Elementos del Sistema Simulador

4.4. ETAPA DE CONVERSIÓN DE LAS SEÑALES DIGITALES A ANALÓGICAS.

Para la Etapa de conversión, se implementó un pequeño circuito conversor digital - análogo, basado en el uso del convertidor DAC0808, el cual posee 8 entradas con lo que se obtiene 256 (2^8) diferentes niveles de corriente. Las entradas digitales del convertidor reciben la información por parte del microcontrolador que varían entre +5 y 0 volts. Sus características principales son:

- Exactitud relativa: $\pm 0.19\%$ error máximo
- Rápido tiempo de estabilización: 150 nanosegundos típico
- Entradas digitales no inversoras TTL y CMOS compatible
- Alta velocidad de multiplicación: 8 mA / μ S
- Rango de voltaje: ± 4.5 [V] a ± 18 [V]
- Bajo consumo: 33 [mW] para ± 5 [V]
- Convertidor paralelo, los datos binarios que se van a convertir llegan en paralelo a la entrada.

La aplicación típica del DAC se muestra en la figura 4.4.1:



TL/H/5687-3

Figura 4.4.1. Aplicación típica DAC0808

Existen varias aplicaciones para el uso de un convertidor digital analógico, como:

- Control
- Pruebas automáticas
- Reconstrucción de señal

Para el caso del Sistema Simulador, el DAC fue utilizado para reconstruir la señal del ECG. La señal de salida del DAC0808 depende directamente del voltaje de referencia (V_{ref}), por lo cual se colocó un potenciómetro para obtener un voltaje de 1 V a la salida, con el fin de establecer la amplitud deseada en la Etapa final.

Los cálculos para saber el valor del potenciómetro del voltaje de referencia son:

- Con la fórmula proporcionada por el fabricante del convertidor, se calcula V_{ref} , considerando el valor de $V_o = 1$ [V].

$$V_o = V_{ref} (A_n / 2^n), \text{ se despeja } V_{ref},$$

$$V_{ref} = V_o / (A_n / 2^n)$$

$$V_{ref} = 1 (1/2 + 1/4 + 1/8 + 1/16 + 1/32 + 1/64 + 1/128 + 1/256)$$

$$V_{ref} = 0.996 \text{ [V]}$$

- El valor obtenido es aproximadamente 1[V], teniendo este valor como referencia se calcula el valor del potenciómetro, el circuito propuesto para proporcionar V_{ref} del DAC se muestra en la figura 4.4.2.

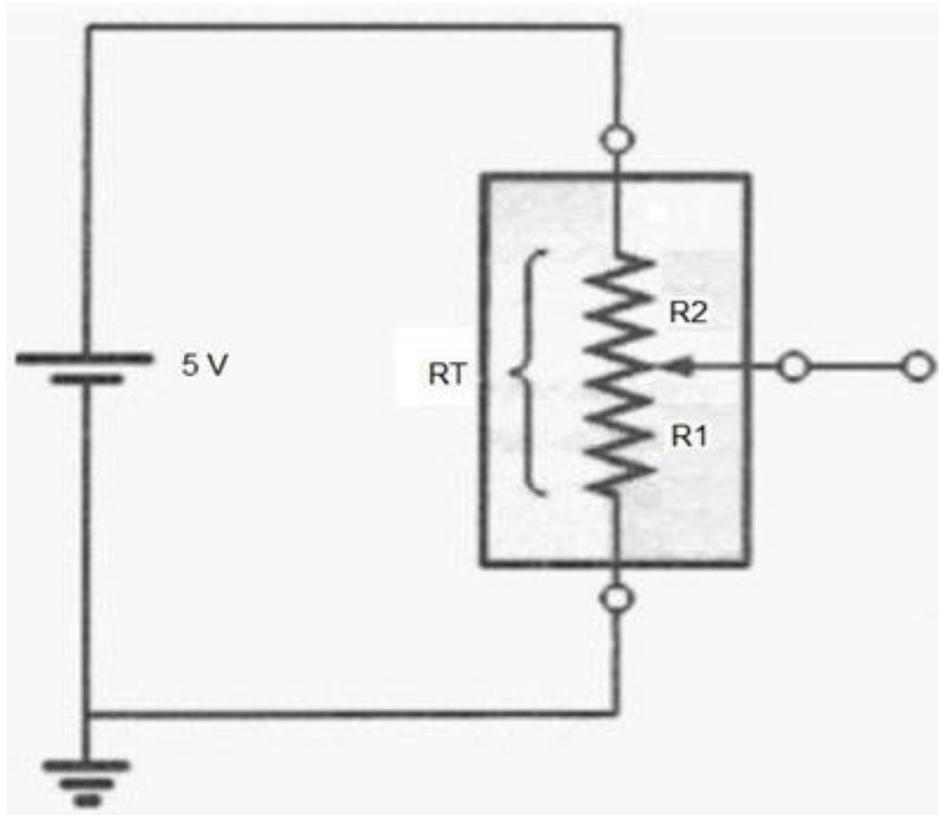


Figura 4.4.2. Circuito con el potenciómetro

- Se consideran los siguientes datos para continuar con los cálculos:

$$V_T = 5 \text{ [V]}$$

$$V_{R1} = V_{\text{ref}} = 1 \text{ [V]}$$

$$V_{R2} = 4 \text{ [V]}$$

donde:

$$R1 = 1 \text{ [k}\Omega\text{]} \text{ y } R2 = ?$$

aplicando la regla del divisor de voltaje.

$$V_{R1} = \frac{R_1 \times V_T}{R_1 + R_2}$$

despejando R_2 para obtener el valor del potenciómetro R_T

$$R_2 = \frac{R_1 \times V_T}{V_{R1}} - R_1 = \frac{1 \times 10^3 \times 5}{1} - 1 \times 10^3 = 4 \times 10^3 \text{ [}\Omega\text{]}$$

$$R_T = R_1 + R_2 = 1 \times 10^3 + 4 \times 10^3 = 5 \times 10^3 \text{ [}\Omega\text{]}$$

Por lo tanto se utiliza un potenciómetro de 5 [kΩ], que es un valor comercial. En la figura 4.4.3 se muestran el potenciómetro calculado y la aplicación típica del DAC colocado en el Sistema.

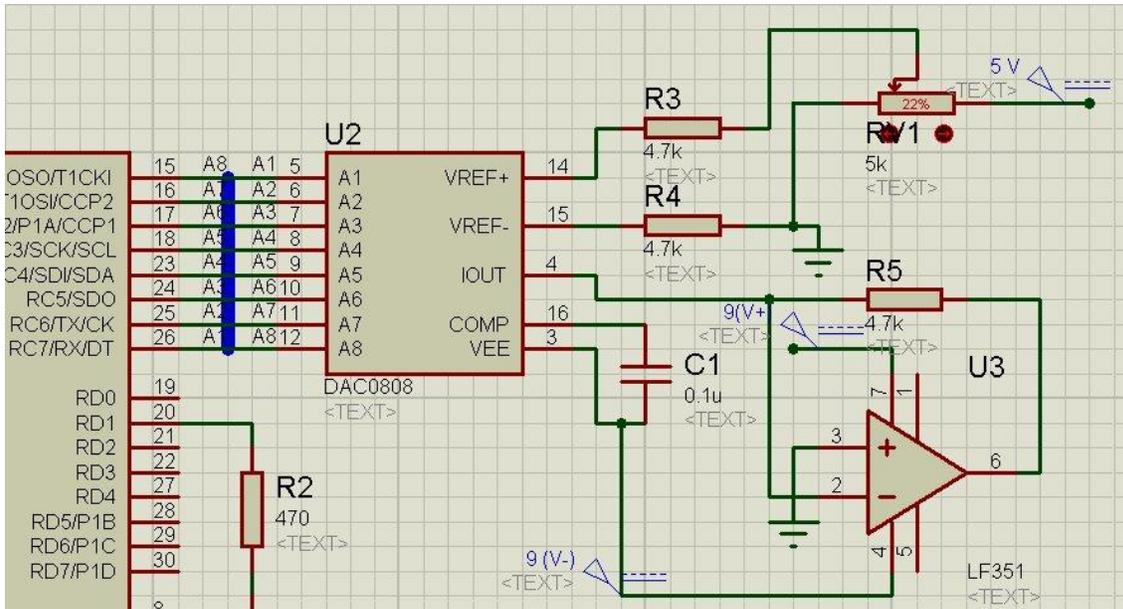


Figura 4.4.3. Circuito del DAC

Otro circuito para el Sistema Simulador se trata de un filtro pasa bajos con una frecuencia de corte de 1500 [Hz] y una ganancia aproximadamente de 1.5, se encarga de filtrar la señal y reducir las posibles interferencias ya que todo el Sistema ofrece a la salida voltajes muy pequeños, por lo que es propenso al ruido, electrónicamente hablando.

Considerando la fórmula:

$$F_{corte} = \frac{1}{2\pi R_6 C_2}$$

despejando R_6 :

$$R_6 = \frac{1}{2\pi C_2 F_{corte}}$$

se considera el valor del capacitor, $C_2 = 0.001 [\mu F]$.

$$R_6 = \frac{1}{(2\pi) (0.001 \times 10^{-6} [F]) (1.5 \times 10^3 [Hz])} = 106.103 \times 10^3 [\Omega]$$

por lo tanto se coloca $R_6 = 100 [k\Omega]$, por ser el valor comercial más cercano.

Se procede a calcular la ganancia de voltaje:

$$A_v = 1 + (R_8 / R_7)$$

donde:

$$R_7 = 47 \text{ [k}\Omega\text{]} \text{ y } R_8 = 27 \text{ [k}\Omega\text{]}$$

$$A_v = 1 + (27 \times 10^3 / 47 \times 10^3)$$

$$A_v = 1.57$$

En la figura 4.4.4 se muestra el circuito del filtro diseñado:

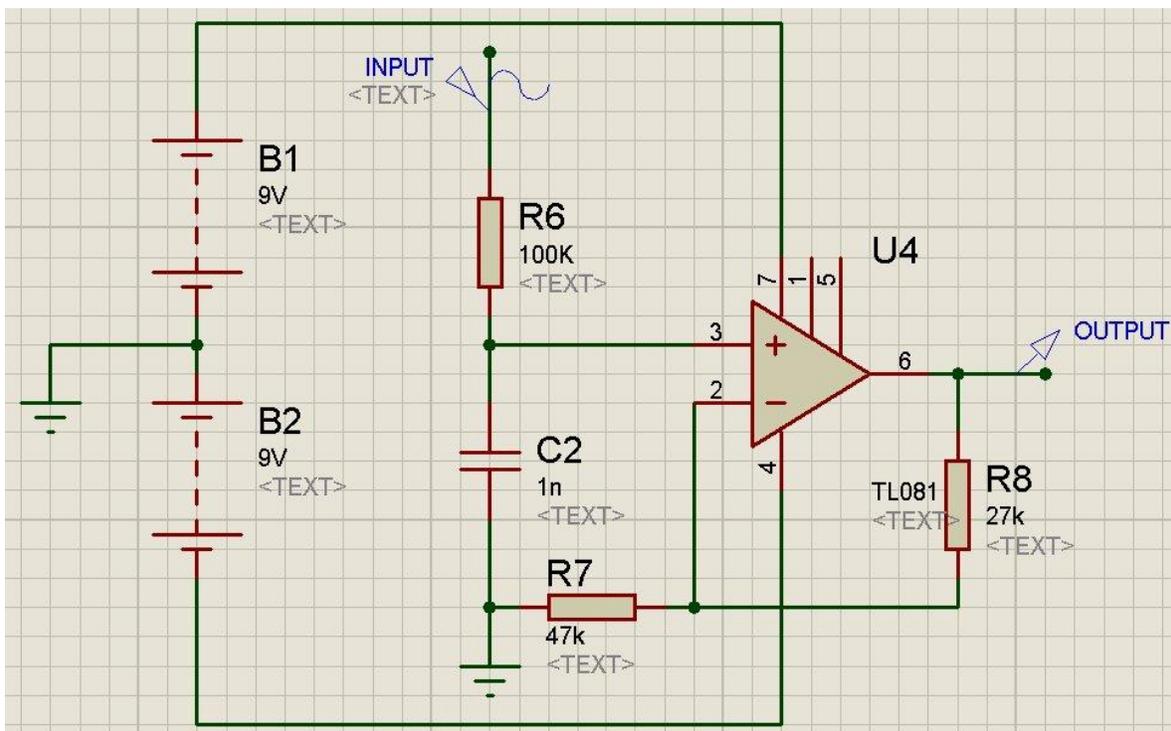


Figura 4.4.4. Circuito del filtro pasa bajos

Los datos que se introdujeron al PIC para generar el ECG patrón no tienen la parte negativa en el eje de las ordenadas, para ajustar dichos valores y obtener el ECG patrón idéntico al real se adicionó un offset después de la señal que proporciona el circuito paso bajas. El offset consiste en añadir un voltaje de -1 volt de corriente directa a la señal que se obtiene después del filtrado, y así desplazarlo hacia la parte negativa utilizando un potenciómetro (RV2). En la Figura 4.4.5 se muestra el circuito propuesto para el offset:

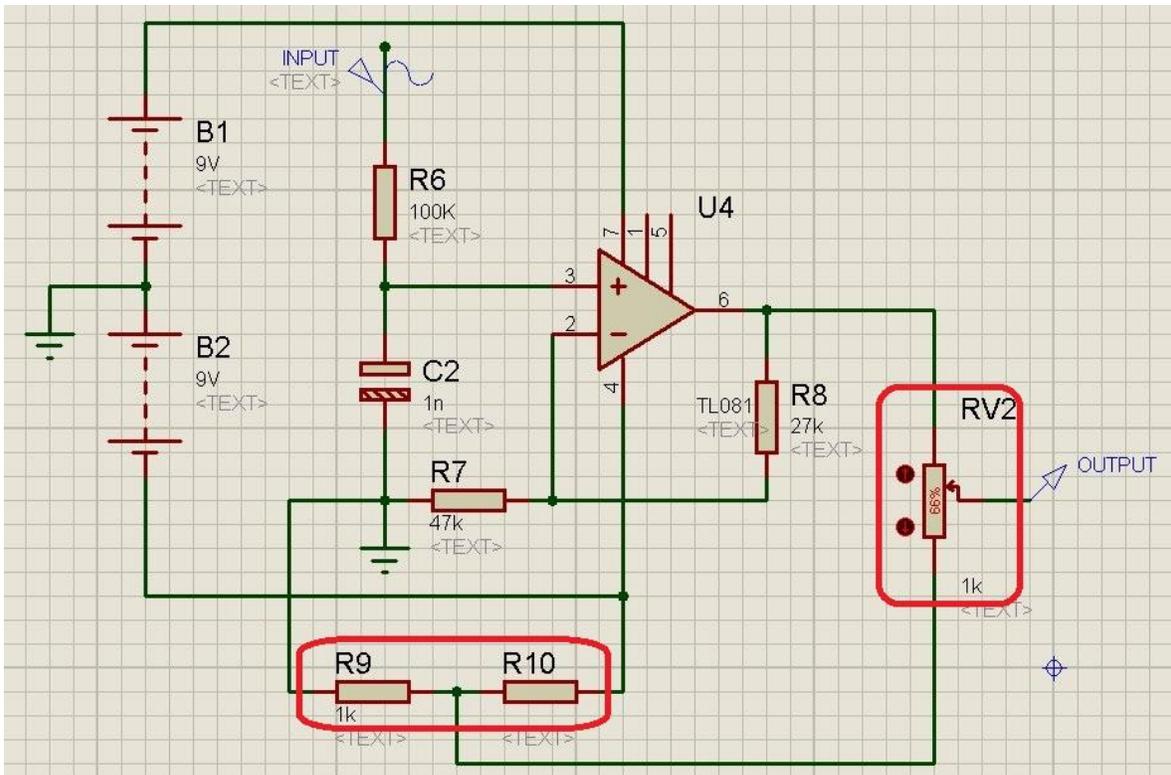


Figura 4.4.5. Circuito del offset

La figura anterior, consta de un divisor resistivo (R9 y R10) para aumentar de -9 [V] a -1 [V] y después conectarlo a un potenciómetro (RV2) con un valor de 1 [kΩ] como se muestra en la figura 4.4.5. [L02]

Para el cálculo del divisor resistivo se consideró $R_9 = 1 \text{ [k}\Omega\text:]$:

- donde:

$$V = -9 \text{ [V]}$$

$$V_{R_9} = -1 \text{ [V]}$$

$$R_9 = 1 \text{ [k}\Omega\text]}$$

$$R_{10} = ?$$

- aplicando la regla del divisor de voltaje para encontrar R_{10} .

$$V_{R_9} = \frac{R_9 \times V}{R_9 + R_{10}}$$

$$R_{10} = \frac{R_9 \times V}{V_{R_9}} - R_9 = \frac{(1 \times 10^3)(-9)}{-1} - 1 \times 10^3 = 8 \times 10^3 \text{ [}\Omega\text]}$$

Valor comercial de $R_{10} = 8.2 \text{ [k}\Omega\text]}$

Los cálculos para el circuito de atenuación son para obtener las derivaciones correspondientes, a continuación se presenta el desarrollo:

- Se tiene un voltaje de entrada de 685 [mV], que se obtuvo después del ajuste del offset, por medio de un divisor resistivo que proporcione un voltaje de 5 [mV]. En la figura 4.4.6 se presenta el circuito propuesto:

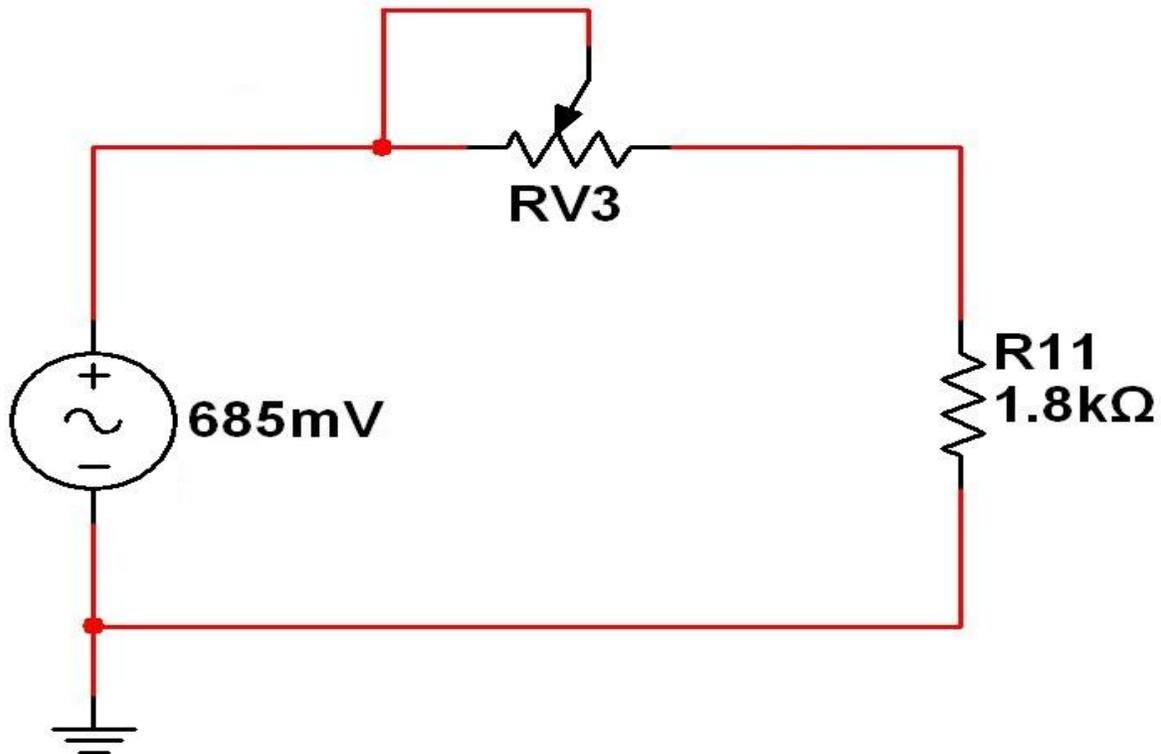


Figura 4.4.6. Cálculo de R11

- del circuito anterior se coloca una resistencia variable para asegurar el valor deseado de voltaje:

$$V = 685 \times 10^{-3} \text{ [V]}$$

$$V_{R11} = 5 \times 10^{-3} \text{ [V]}$$

$$R_{V3} = ?$$

$$R_{11} = 1.8 \text{ [k}\Omega\text{]}$$

aplicando la regla del divisor de voltaje,

$$V_{R11} = \frac{R_{11} \times V}{R_{V3} + R_{11}}$$

despejando R_{V3}

$$R_{V3} = \frac{R_{11} \times V}{V_{R11}} - R_{11} = \frac{1.8 \times 10^3 \times 685 \times 10^{-3}}{5 \times 10^{-3}} - 1.8 \times 10^3 = 244.8 \times 10^3 [\Omega]$$

por lo tanto se utiliza un potenciómetro de 250 [k Ω] por ser un valor comercial. Al tener una alta impedancia el voltaje puede variar, de tal forma que para asegurar el valor deseado de voltaje es preciso colocar un seguidor de voltaje utilizando un amplificador operacional LM741, tal y como se presenta más adelante en el Sistema Simulador completo.

Para obtener las derivaciones que se utilizan en un electrocardiógrafo se implementó el circuito de la figura 4.4.7, donde las señales tengan una magnitud del orden de mili volts.

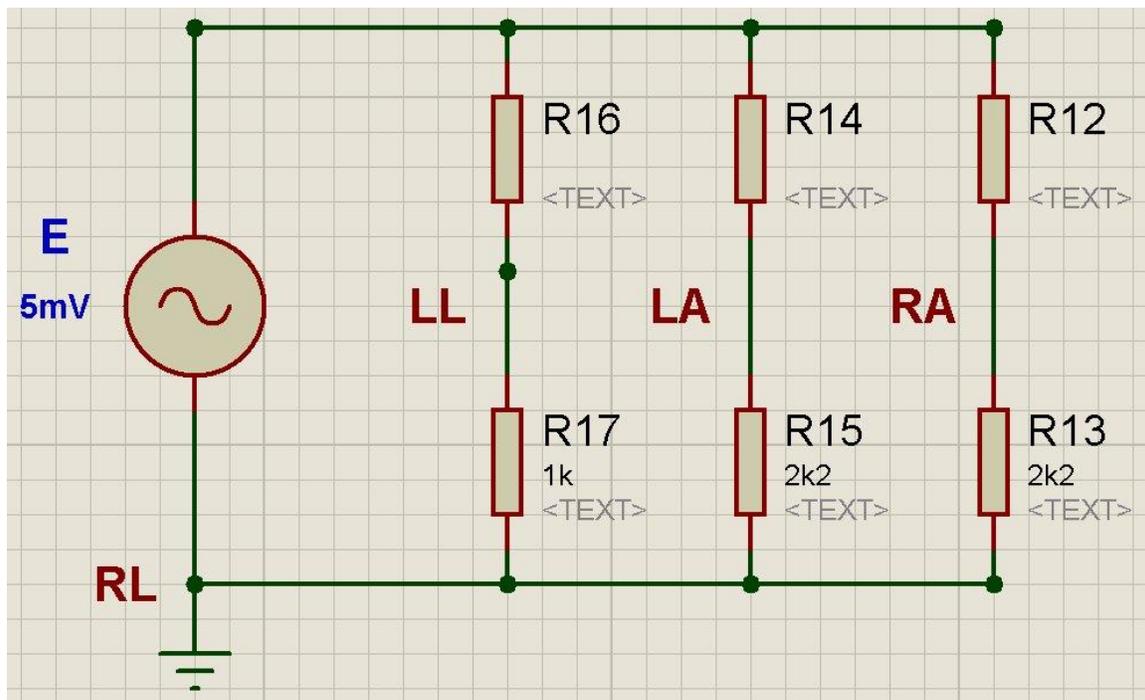


Figura 4.4.7. Bloque de atenuación

La señal recibida por el divisor de voltaje calculado anteriormente, tiene un valor pico de 5 [mV]; por esta razón se emplea un arreglo de resistores para atenuar la señal.

Cada divisor de voltaje representa una parte del cuerpo, en donde se debe colocar el electrodo:

- RA: Simula al electrodo que va en el brazo derecho
- LA: Simula al electrodo que va en el brazo izquierdo

- LL: Simula al electrodo que va en el pie izquierdo
- RL: Simula al electrodo que va en el pie derecho

Datos:

$$E = 5 \times 10^{-3} \text{ [V]}$$

$$V_{13} = 0.2 \times 10^{-3} \text{ [V]}$$

$$V_{15} = 0.3 \times 10^{-3} \text{ [V]}$$

$$V_{17} = 1 \times 10^{-3} \text{ [V]}$$

$$R_{12} = ?$$

$$R_{13} = 2.2 \text{ [k}\Omega\text{]}$$

$$R_{14} = ?$$

$$R_{15} = 2.2 \text{ [k}\Omega\text{]}$$

$$R_{16} = ?$$

$$R_{17} = 1 \text{ [k}\Omega\text{]}$$

aplicando la regla del divisor de voltaje para encontrar R_{12} .

$$V_{13} = \frac{R_{13} \times E}{R_{12} + R_{13}}$$

$$R_{12} = \frac{R_{13} \times E}{V_{13}} - R_{13} = \frac{2.2 \times 10^3 \times 5 \times 10^{-3}}{0.2 \times 10^{-3}} - 2.2 \times 10^3 = 52.8 \times 10^3 \text{ [\Omega]}$$

Valor comercial de $R_{12} = 56 \text{ [k}\Omega\text{]}$

Cálculo de R_{14}

$$V_{15} = \frac{R_{15} \times E}{R_{14} + R_{15}}$$

$$R_{14} = \frac{R_{15} \times E}{V_{15}} - R_{15} = \frac{2.2 \times 10^3 \times 5 \times 10^{-3}}{0.3 \times 10^{-3}} - 2.2 \times 10^3 = 34.467 \times 10^3 \text{ [\Omega]}$$

Valor comercial de $R_{14} = 33 \text{ [k}\Omega\text{]}$

Cálculo de R_{16}

$$V_{17} = \frac{R_{17} \times E}{R_{16} + R_{17}}$$

$$R_{16} = \frac{R_{17} \times E}{V_{17}} - R_{17} = \frac{1 \times 10^3 \times 5 \times 10^{-3}}{1 \times 10^{-3}} - 1 \times 10^3 = 4 \times 10^3 \text{ [\Omega]}$$

Valor comercial de $R_{16} = 3.9 \text{ [k}\Omega\text{]}$

4.5. ETAPA FINAL, ECG PATRÓN.

La parte de alimentación del Sistema consta de dos fuentes de voltaje: una hecha a base de baterías y la segunda es una fuente rectificadora. A continuación se explica el desarrollo de cada una de las fuentes:

La fuente con baterías requiere de:

- Una fuente simétrica para poder alimentar los amplificadores operacionales, para lo cual se utilizará una batería de 9 Volts para darle portabilidad al Sistema Simulador. También se requiere una fuente de 5 [V] para poder alimentar al microcontrolador y al convertidor análogo digital.
- Para cumplir con los 5 [V] se utilizó un regulador de voltaje KA7805 de tres terminales para generar un voltaje fijo de +5 Volts, utilizando el circuito recomendado por el fabricante en la aplicación típica que se muestra en la figura 4.5.1.

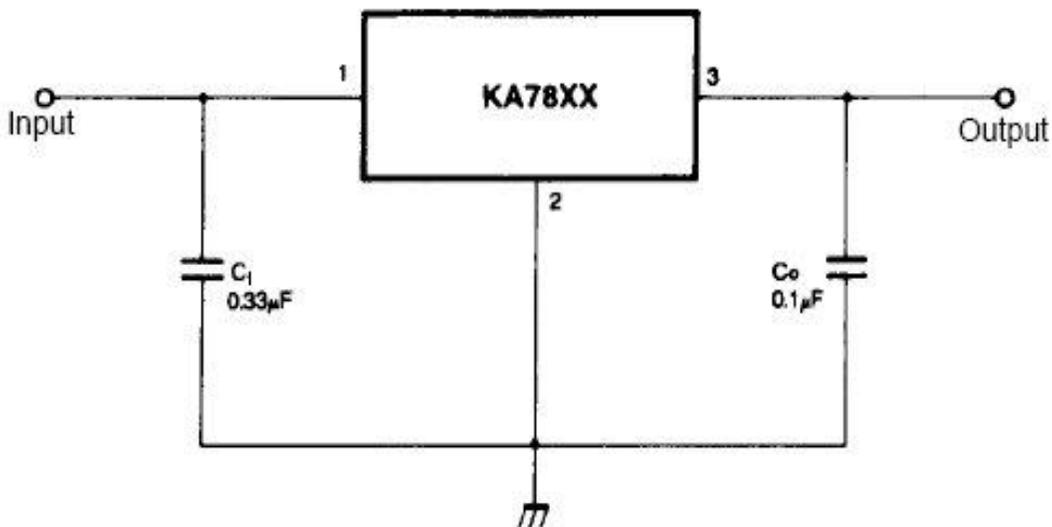


Figura 4.5.1. Aplicación típica regulador de voltaje

El regulador de voltaje KA7805 tiene las siguientes características:

- Rango de alimentación de 8 a 20 [V]
- Voltaje de salida típico: 5 [V]
- Corriente máxima de salida: 1 Amper

Para generar - 9 volts se utilizó un circuito convertidor de voltaje negativo ICL7660, en la figura 4.5.2 se presenta la aplicación típica recomendada por el fabricante:

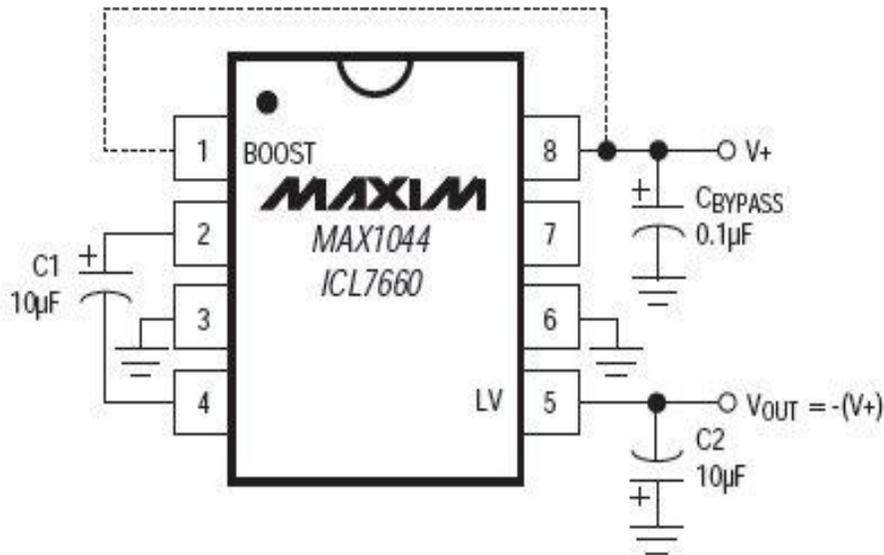


Figura 4.5.2. Aplicación típica del convertidor de voltaje

Las características principales del convertidor de voltaje son:

- Rango de alimentación: de 1.5 a 10 volts
- Eficiencia de transformación: 98%
- Corriente máxima de salida: 10 [mA]

El circuito de alimentación diseñado se muestra en la figura 4.5.3.

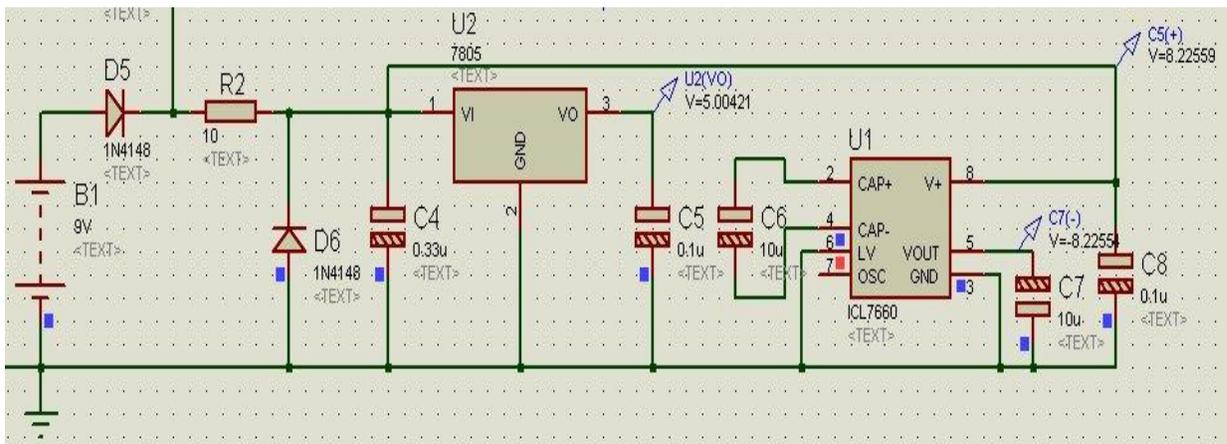


Figura 4.5.3. Fuente con batería de 9 [V]

Para la fuente rectificadora se consideró lo siguiente:

- A partir de la alimentación de corriente alterna como la que existe en la mayoría de los hogares, se obtiene un voltaje de corriente directa mediante la rectificación y filtrado de la señal de voltaje en corriente alterna, y por último regularlo para obtener el voltaje fijo de 10 [V] en corriente directa. En la figura 4.5.4 se presenta el circuito:

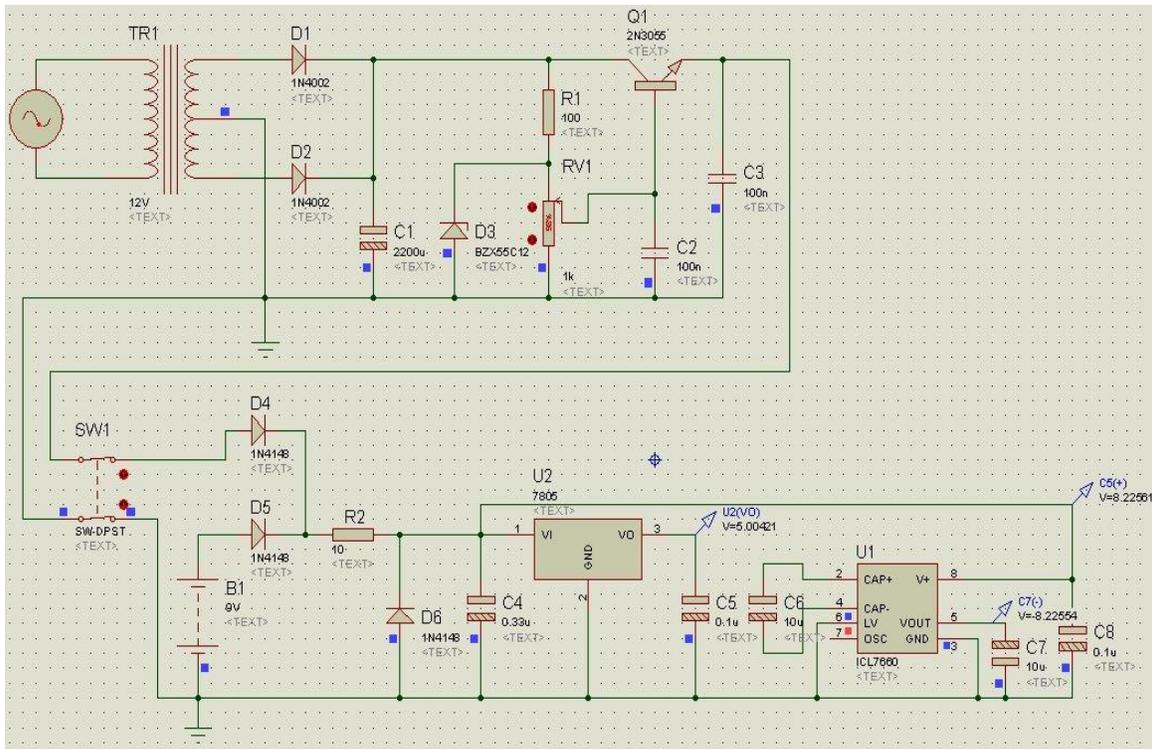


Figura 4.5.4. Fuente rectificadora y fuente con batería de 9 [V]

En la figura anterior se observan las dos fuentes conectadas por medio de un interruptor SW1, que representa la conexión y desconexión de la fuente rectificadora, las fuentes están aisladas, una de otra, por medio de los diodos D4 y D5. A continuación se explica el funcionamiento de los diodos:

- Cuando la fuente rectificadora entrega 10 [V], D4 se encuentra conduciendo. D5 se encuentra en polaridad inversa, es decir, circuito abierto, lo que ocasiona que la batería de respaldo 9 [V] este desconectada.

- Cuando no se ocupa la fuente rectificada, se queda abierto SW1, como se muestra en la Figura 4.5.5, D4 ya no conduce, D5 está polarizado directamente por la batería de 9 [V].

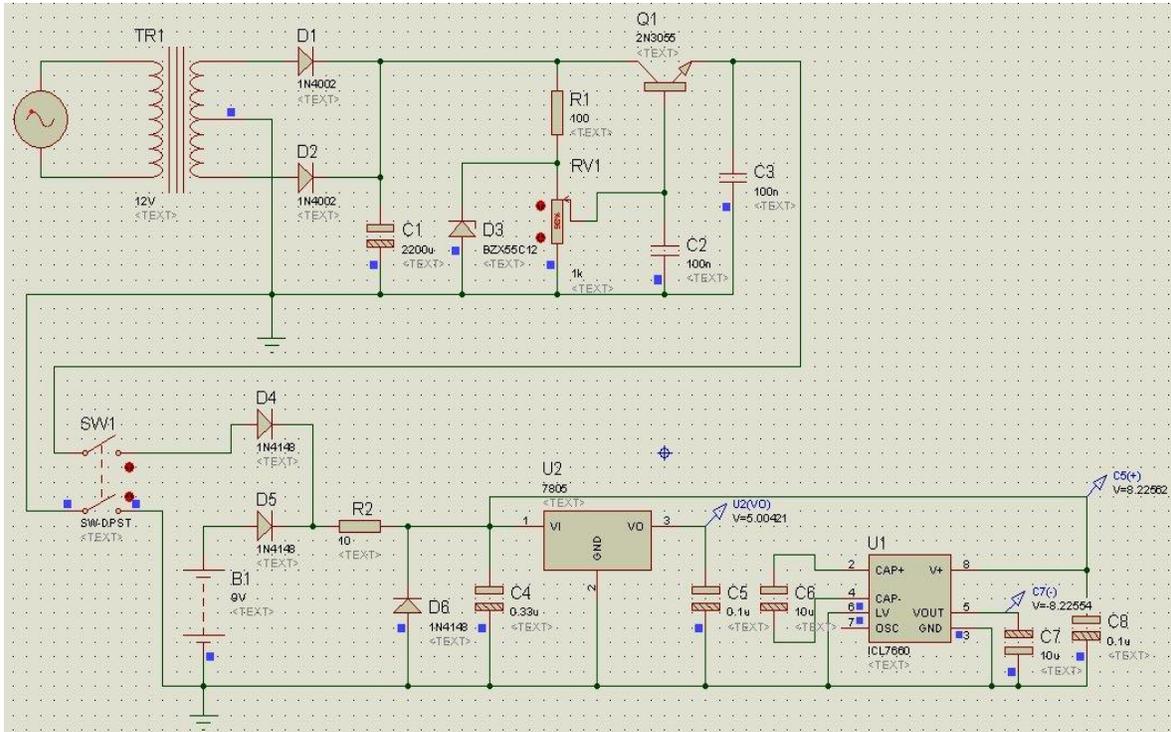


Figura 4.5.5. Fuente rectificada desconectada

La Etapa final corresponde a la obtención del ECG patrón, como se muestra en la Figura 4.5.6, aparecen todos los elementos que integran al Sistema. Se debe hacer mención de un led conectado al bit 1 del Puerto D del PIC16F887; la función del led es tener la certeza de que el Sistema está funcionando antes de conectarlo a un electrocardiógrafo, ya que al activar el Sistema Simulador el led se prenderá y apagará cada medio segundo por instrucciones del programa.

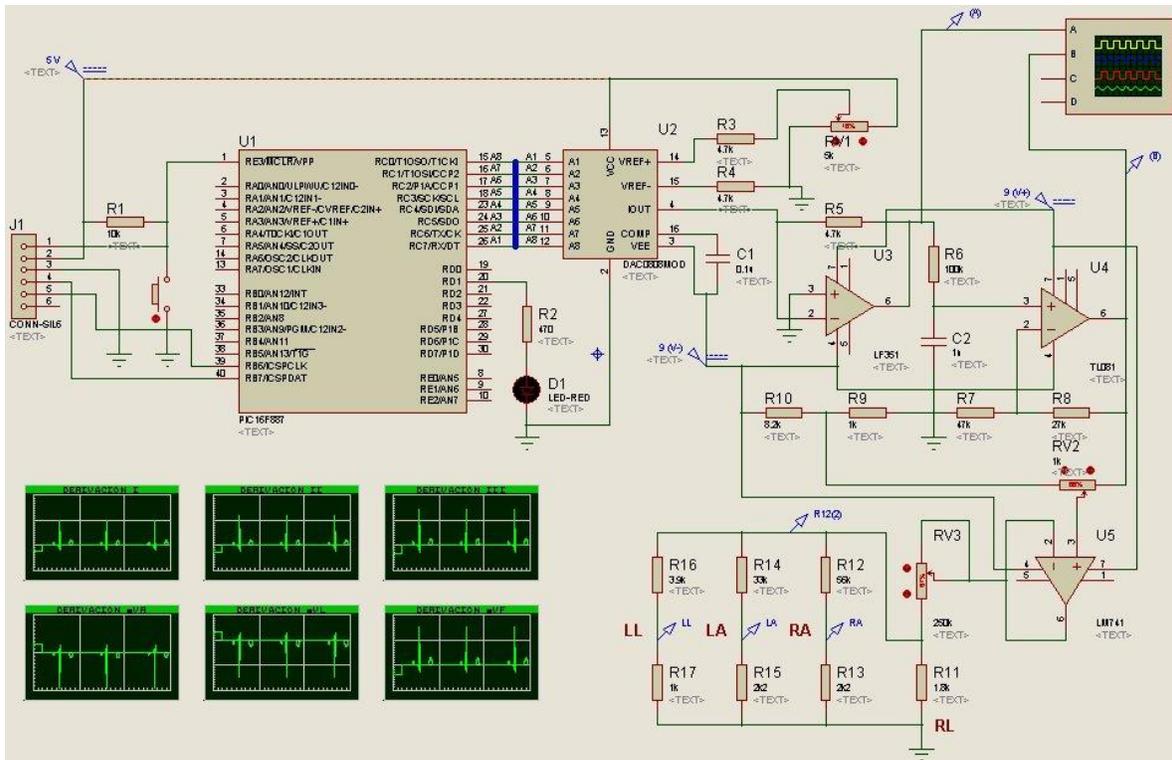


Figura 4.5.6. Sistema Simulador completo

En la figura anterior se puede apreciar en la parte baja las resistencias que sirven como divisores de voltaje para obtener las derivaciones. Como se mencionó anteriormente, a la salida del potenciómetro RV2 fue necesario conectar un seguidor de voltaje (U5), es un circuito que consta de un amplificador operacional LM741, cuya función es la de producir una tensión de salida que es exactamente igual a la tensión de entrada (o suficientemente parecida como para satisfacer casi cualquier aplicación). La ventaja de esta conexión es que la carga conectada a la salida no altera el voltaje y la corriente proporcionada, es decir, se trata de un seguidor de voltaje para proporcionar la señal de salida.

También en la figura 4.5.6 se muestran unas sondas de voltaje que son LL, LA y RA, para visualizar las derivaciones por medio de las gráficas que proporciona ISIS de PROTEUS. En este caso cada derivación se obtiene usando las configuraciones de acuerdo a la tabla 4.5.1:

Derivación Nomenclatura	Definición	Nombre de la derivación
I	$I = LA + RA$	Derivación bipolar de miembros (Einthoven)
II	$II = LL + RA$	
III	$III = LL - LA$	
aVR	$aVR = RA - (LA + LL)/2$	Derivaciones Aumentadas (Goldberger)
aVL	$aVL = LA - (LL + RA)/2$	
aVF	$aVF = LL - (LA + RA)/2$	

Tabla 4.5.1. Definición de derivaciones

Los resultados de las señales obtenidas, conforme al orden de la tabla 4.5.1 son:

- La derivación bipolar I se muestra en la figura 4.5.7, donde se aprecian las ondas R, las que tienen la cresta más grande, y entre dos ondas consecutivas R existe de separación un segundo. Lo cual indica que en el ECG patrón hay 60 pulsos por minuto.
- La derivación bipolar II se aprecia en la figura 4.5.8.
- En la figura 4.5.9 aparece la derivación bipolar III.
- En la figura 4.5.10 se muestra la derivación aumentada aVR.
- En la figura 4.5.11 se observa la derivación aVL.
- Por último se presenta la derivación aVF en la figura 4.5.12.

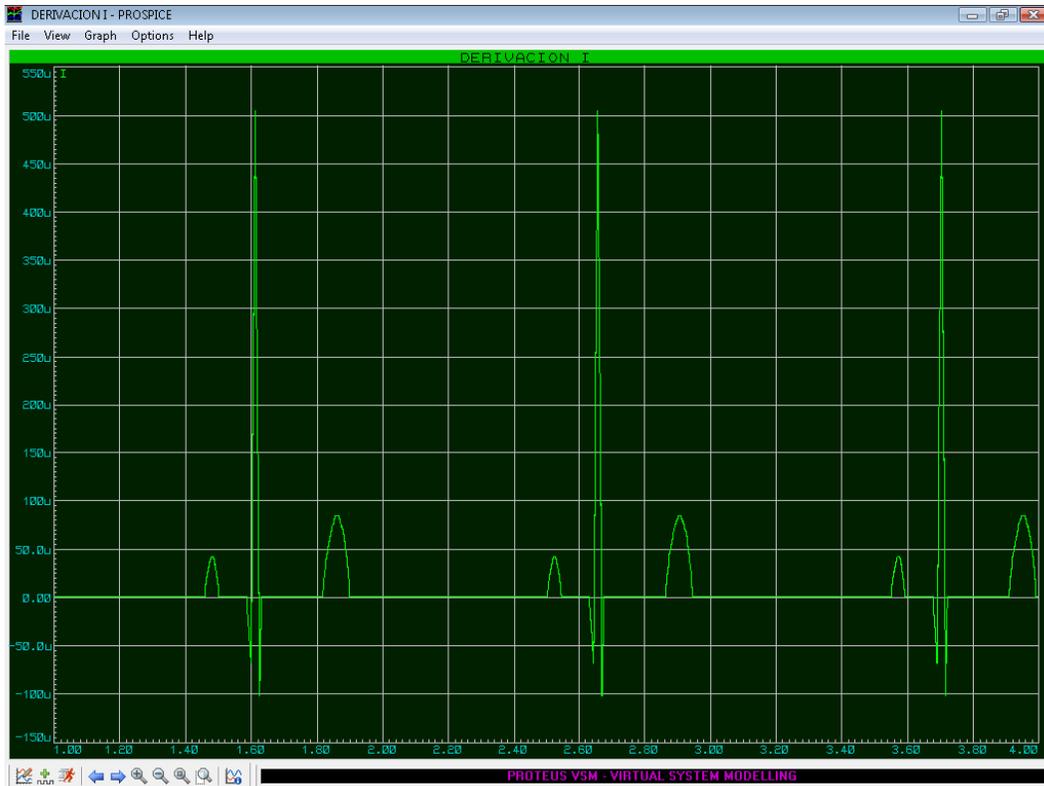


Figura 4.5.7. Derivación bipolar I

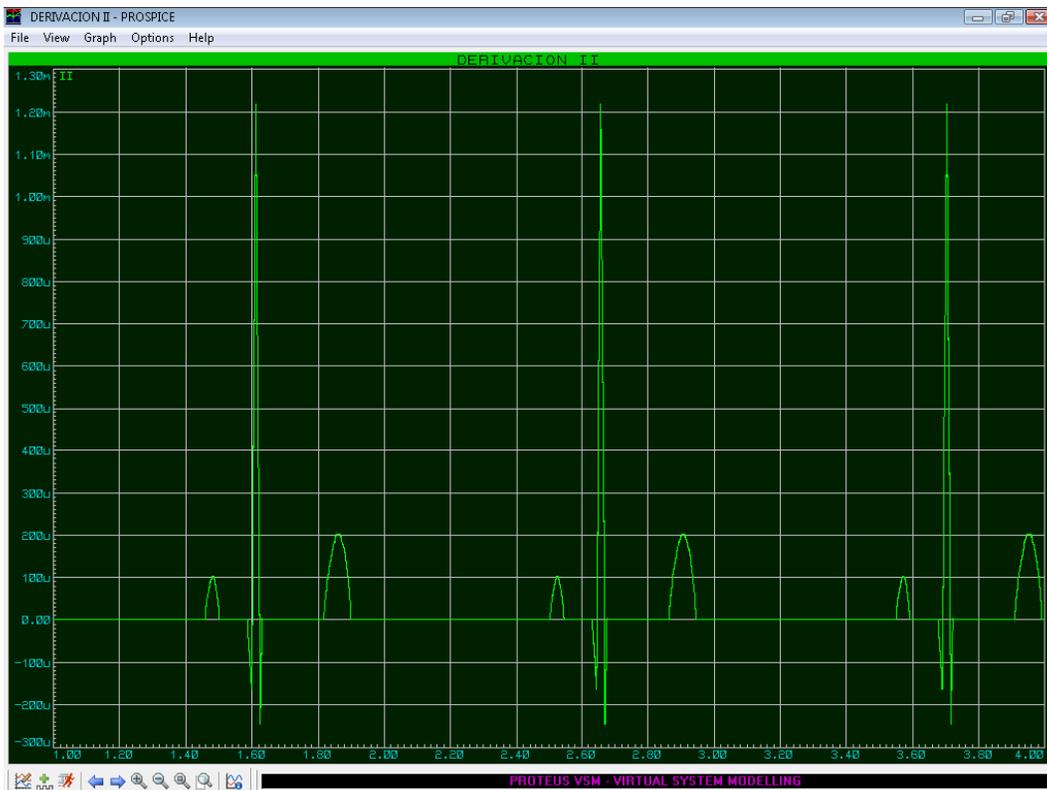


Figura 4.5.8. Derivación bipolar II

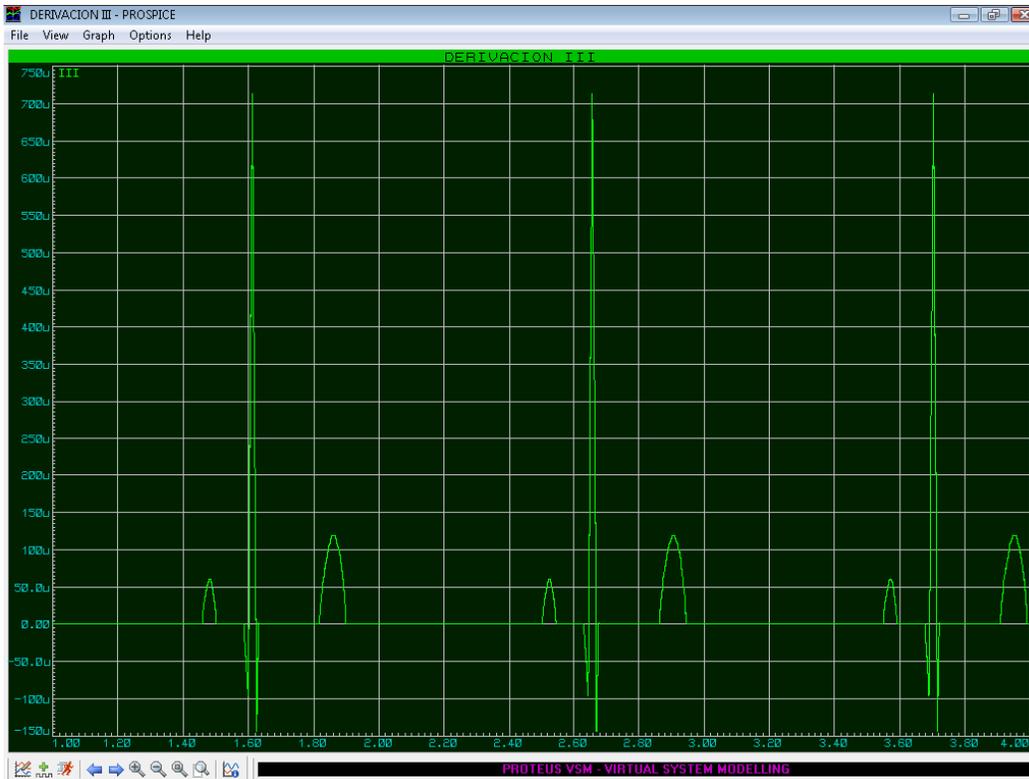


Figura 4.5.9. Derivación bipolar III

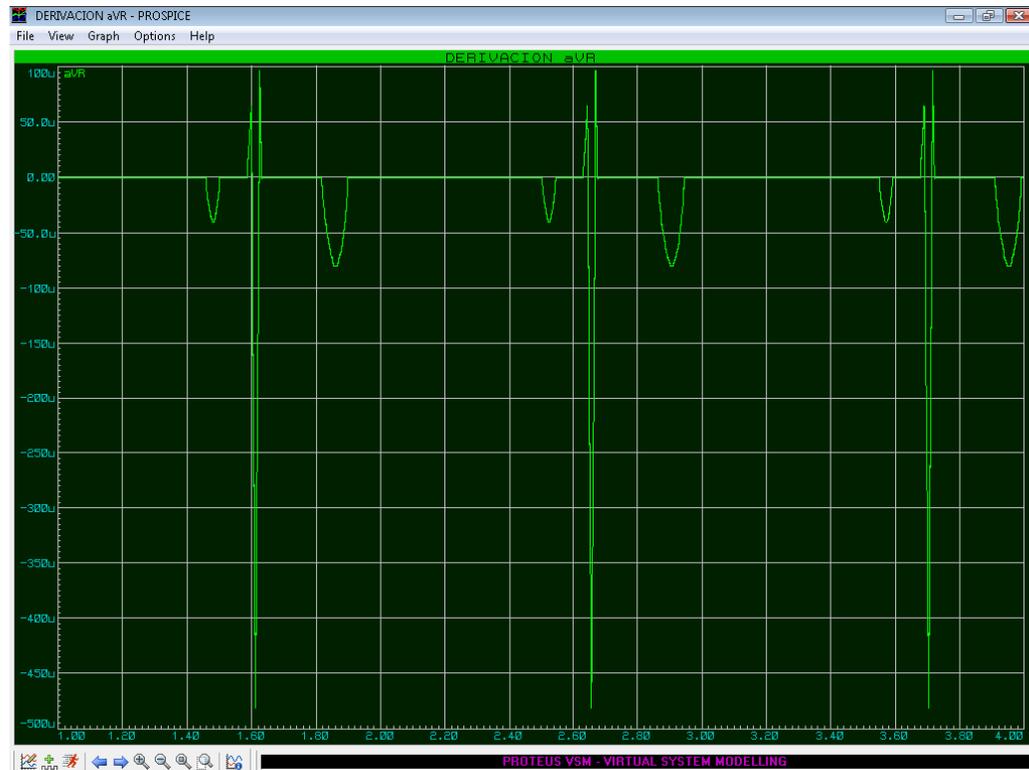


Figura 4.5.10. Derivación aumentada aVR

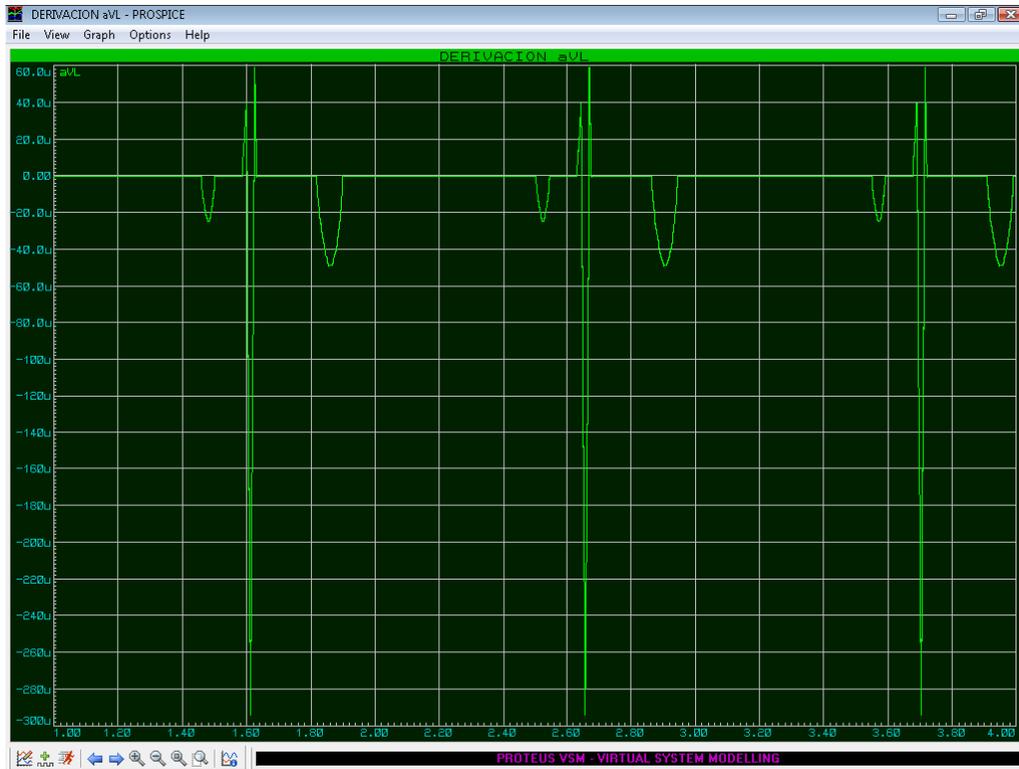


Figura 4.5.11. Derivación aumentada aVL

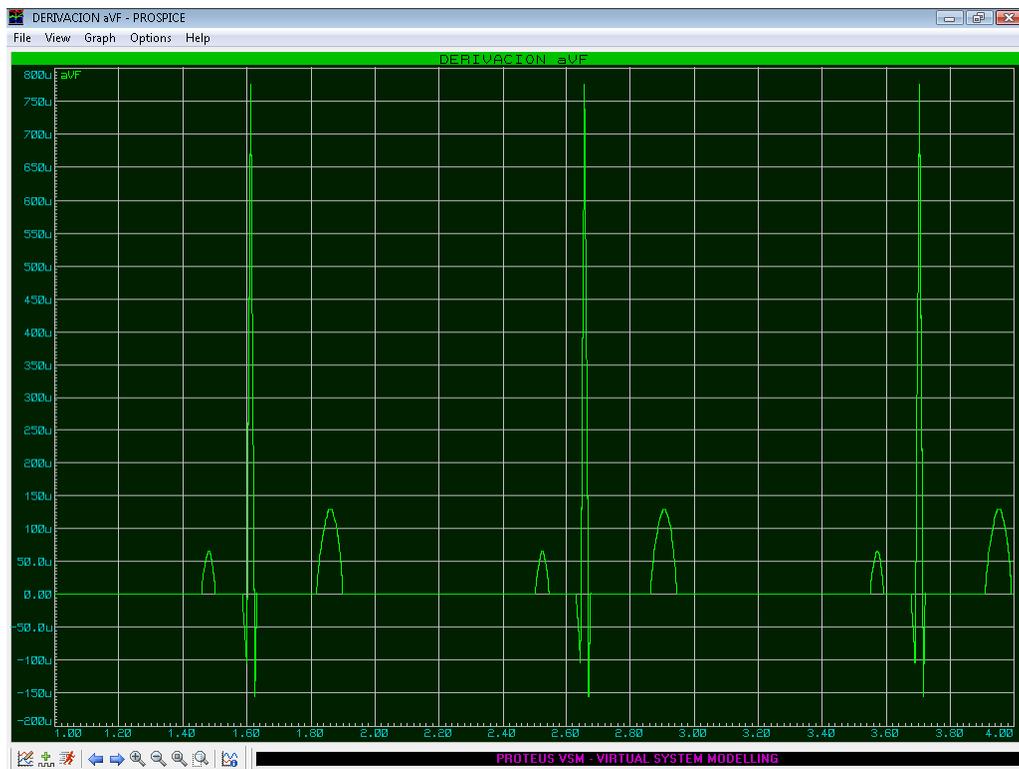


Figura 4.5.12. Derivación aumentada aVF

Las graficas obtenidas anteriormente representan al ECG patrón que proporciona el Sistema Simulador, con una frecuencia cardíaca de 60 pulsos por minuto, de tal forma que se puede comprobar la copia de un ECG real, considerando la figura 4.2.4 y la 4.5.7.

CAPÍTULO 5



CONCLUSIONES: RESULTADOS Y APLICACIONES

5.1. DIFERENTES USOS

Ya integrados todos los elementos que conforman al Sistema simulador, ahora se mencionarán sus diferentes usos. Existen varias formas en las que se podría aprovechar el Sistema simulador, como:

- En varias cirugías médicas los especialistas conectan al paciente un electrocardiógrafo, el cual les da la lectura del comportamiento del corazón, para saber cómo está respondiendo a la cirugía. Es en esta parte donde tiene gran funcionalidad el Sistema Simulador, ya que antes de intervenir al paciente, se le puede hacer al electrocardiógrafo pruebas de desempeño para asegurar el funcionamiento correcto durante la cirugía. En la figura 5.1.1 se observa el área de un quirófano donde también se muestran los instrumentos con los que cuentan regularmente. En dicha figura se puede apreciar del lado izquierdo, el instrumento que se encuentra más arriba se trata de un electrocardiógrafo, el cual será conectado al paciente para monitorear el comportamiento de su corazón y así asegurar que no presente problemas cardíacos durante la cirugía.
- En diferentes instituciones médicas se hacen estudios relacionados con deportistas de alto rendimiento, con diferentes propósitos, desde estadísticos hasta estudios amplios relacionados con el rendimiento físico de los pacientes, y en este caso no serán pacientes enfermos, sino personas que serán monitoreadas para revisar como aumenta o se mantiene su capacidad cardiovascular. Existen muchos factores que se deben tomar en cuenta, como el tiempo de entrenamiento, la alimentación, tiempo de descanso, estados del ánimo, de tal forma que existen muchos aspectos que se consideran variables, pero al emplear el Sistema simulador se podrá tener como un factor fijo: el estado del electrocardiógrafo, el cual se planea utilizar para ese tipo de estudio, ya que con cierta frecuencia se estará calibrando.



Figura 5.1.1. Quirófano

- En la actualidad existen varias instituciones educativas de nivel superior donde se han realizados trabajos para el diseño y construcción de electrocardiógrafos, de tal forma que resultaría muy benéfico contar con una señal que permitiera probar dichos aparatos.
- En el estado actual del Sistema Simulador, tiene una gran utilidad didáctica, para los estudiantes del área médica, o cualquier persona interesada en el tema, la forma de onda que presenta el ECG patrón, donde se podrán observar y medir todas las características que tiene un ECG real, tal y como se observa en la figura 5.1.2:

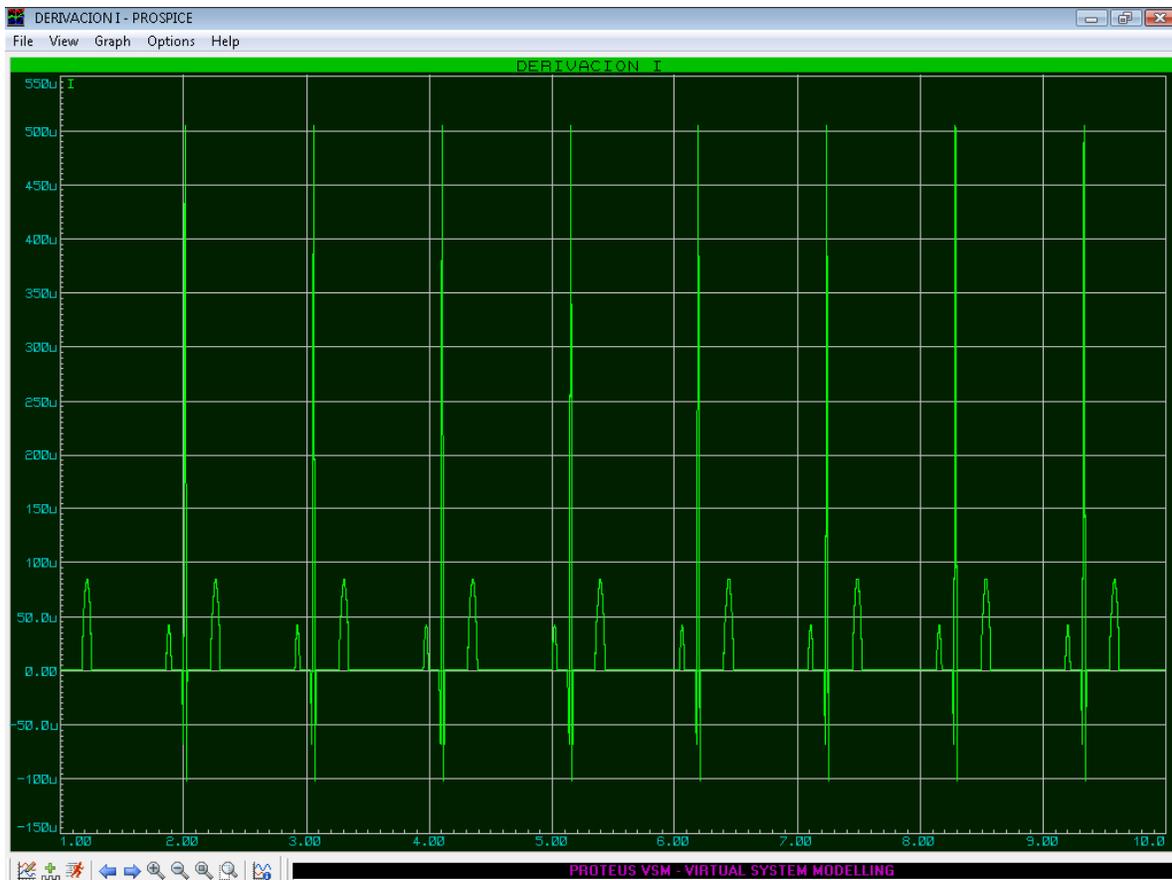


Figura 5.1.2. ECG patrón

- En resumen, con el Sistema simulador se podrán calibrar todos los electrocardiógrafos y de esta forma asegurar el óptimo desempeño para tener una confianza plena en los resultados que proporcionen.

5.2. BENEFICIOS Y CONCLUSIONES

Entre los beneficios que presenta el Sistema Simulador, a continuación se mencionan los principales:

- Se puede calibrar cualquier electrocardiógrafo en el momento que se requiera, de tal forma que si antes se necesitaba de un paciente real para saber el buen funcionamiento de los electrocardiógrafos, ahora con el Sistema Simulador se podrán verificar a cualquier hora. Simplemente se conecta el Sistema Simulador y se obtiene un ECG, y si el electrocardiógrafo no presenta ningún problema, entonces deberá desplegar la señal del ECG patrón, es decir, una señal con 60 pulsos por minuto.
- Si un paciente real, al cual se le toman dos ECG, podrían presentar diferencias, ya que si el paciente habla, está estresado o no guarda una respiración constante, todos estos factores afectan en la captura del ECG, pero con el Sistema Simulador se tendrá la misma señal, sin importar la circunstancia.
- Otro aspecto importante es el económico, como se mostró en el capítulo anterior, en la tabla 4.1.1 aparecen los costos de todos los componentes que se incluyen en el Sistema Simulador. Si el Sistema se llega a construir, sus componentes tienen un costo bajo, además de contar con una gran vida útil, y si alguno de los elementos llegara a fallar, se podría reemplazar fácilmente.
- Como se mencionó anteriormente, dentro de varias Universidades y escuelas se realizan trabajos relacionadas con el diseño y construcción de electrocardiógrafos, por ejemplo: en la Universidad Nacional Autónoma de México se cuentan con por lo menos doce tesis, de manera que el Sistema Simulador sería una herramienta fundamental para determinar el buen funcionamiento de los electrocardiógrafos diseñados y construidos.

- Cabe recordar y aclarar que en el diseño del Sistema Simulador se tuvieron que realizar varias aproximaciones en la elección de los componentes electrónicos, debido a que los valores comerciales no son iguales a los calculados, lo cual ocasionó que al obtener el ECG patrón no se tuviera una separación de, exactamente, un segundo entre dos ondas consecutivas (P, Q, R, S, o T), el tiempo de separación que tienen dos ondas consecutivas es de 1.05 [seg], como se muestra en la figura 5.2.1 el ECG patrón del Sistema Simulador:

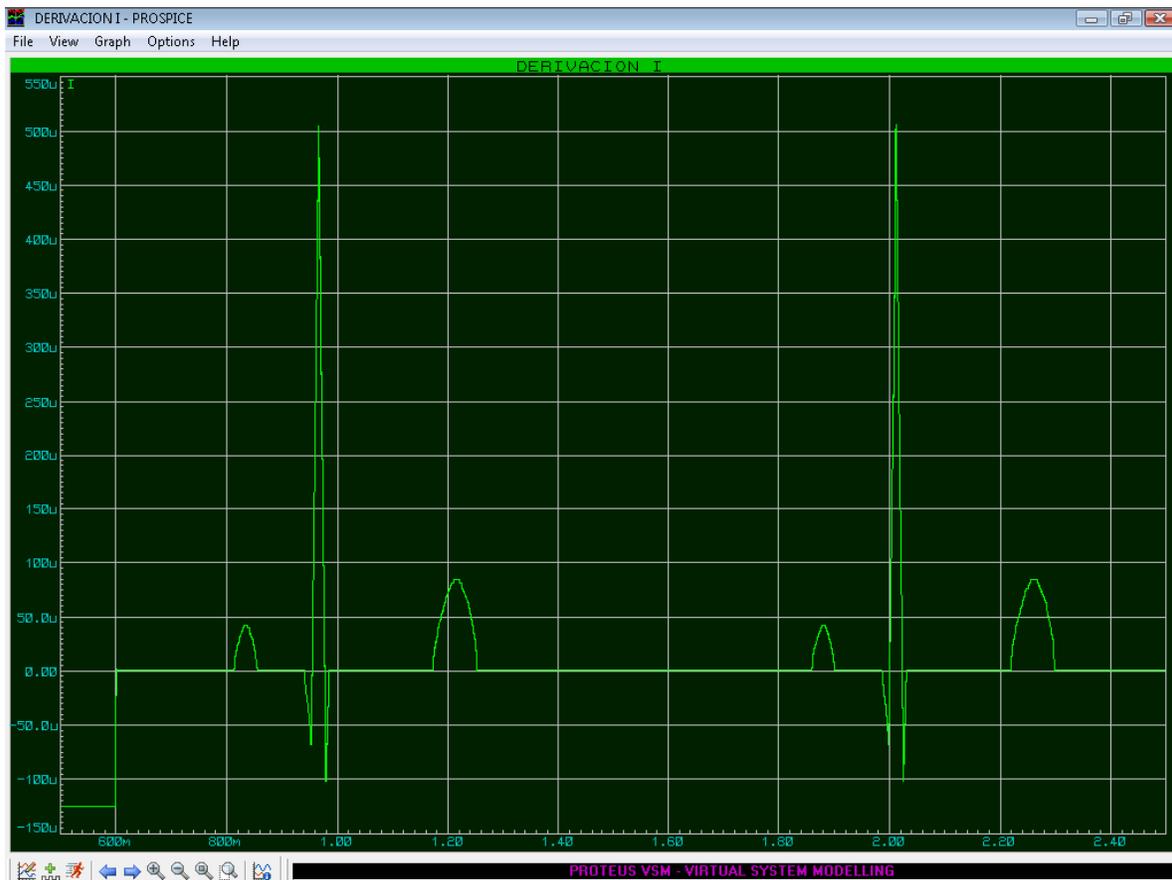


Figura 5.2.1. ECG del Sistema Simulador

- Con el uso del Sistema Simulador se podrá reducir el tiempo de diagnosticar a un paciente, debido a que el electrocardiógrafo permanecerá en óptimas condiciones por contar con una calibración periódica.

5.3. OPTIMIZACIÓN DEL SISTEMA

Como se explicó en el capítulo 3 y 4, el PIC16F887 tiene muchas herramientas que se pueden usar para ampliar las capacidades del Sistema Simulador, de hecho, el Sistema tiene un conector extra de 6 pines que se emplea para colocar el programador directamente de la computadora al PIC16F887, el programador lo fabrica Microchip y se llama PICkit 2, ver figura 5.3.1, es un pequeño dispositivo que se conecta por medio de un conector USB – miniUSB al puerto USB.



Figura 5.3.1. PICkit 2

Para optimizar el Sistema se recomienda:

- Construir físicamente el Sistema Simulador, para demostrar los alcances que tendrá.
- Como se había mencionado en el capítulo anterior, el PIC tiene varias herramientas extras, una de ellas es la facilidad de cambiar el programa que, inicialmente, sirvió solo para digitalizar la señal muestreada y obtener el ECG. Considerando el programador de la figura 5.3.1, se podrán añadir varios tipos de señales digitalizadas de electrocardiogramas, con bradicardia, con taquicardia, etc.
- El Sistema Simulador cuenta con sólo una señal para verificar el funcionamiento de un electrocardiógrafo, se sugiere meterle muchas más señales al PIC16F887 para poder verificar en un intervalo amplio las señales del ECG, como por ejemplo desde 30 hasta 300 ppm.
- Para una mayor precisión se requiere componentes con “calidad militar”.
- Adicionar un teclado matricial para programar el tipo de ECG que se requiera, además modificar el ancho de banda y la amplitud de la señal, y obtener varios tipos de ECG.

APÉNDICES

Apéndice A

Links y hojas de datos

- Fisiología:

www.mailxmail.com

www.medynet.com/usuarios/jraguilar/Manual%20de%20urgencias%20y%20Emergencias/ecg.pdf

www.urgenciauc.com/profesion/pdf/ecg_tutorial.pdf

www.spapex.es/pdf/taller_ekg.pdf

www.cymasa.com.ar/pdf/urgencias_emergentologia/ECG%20%20NORMAL.pdf

www.uhu.es/francisco.cordoba/asignaturas/CUERPOHUMANO/TEMAS%20PDF/T6-Circulacion.pdf

www.nasajpg.com/e-books-digitales-free/

www.ecocardio.com/rincon_alumno/pdf/Capitulo_01.pdf

www.dalcame.com/ecg.html

www.eccpn.aibarra.org/temario/seccion4/capitulo56/capitulo56.htm

www.portalbiomedico.com/equipamiento-biomedico/ecg-/-ekg/ecg-/-ekg-derivaciones-bipolares-aumentadas-y-precordiales-conceptos-basicos.html

www.slideshare.net/venecia90210/interpretacion-electrocardiograma-ekg

- Electrónica

www.dte.us.es/tec_inf/itis/estr_comp/docweb/tema4/tema4ec-3.pdf

www.dte.us.es/tec_inf/itis/estr_comp/docweb/tema4/tema4ec-3.pdf

www.mikroe.com.

- Hoja de datos

PIC16F887

ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf

DAC0808

www.national.com/ds/DA/DAC0808.pdf

LF351

www.ece.ucsb.edu/Faculty/rodwell/Classes/ECE137B/LF351.pdf

TL081

www.ti.com/lit/ds/snosbw6a/snosbw6a.pdf

7805

www.datasheetcatalog.org/datasheets/228/390068_DS.pdf

ICL7660

datasheets.maxim-ic.com/en/ds/ICL7660-MAX1044.pdf

1N4002

www.datasheetcatalog.org/datasheet/motorola/1N4002.pdf

1N4148

www.nxp.com/documents/data_sheet/1N4148_1N4448.pdf

BZX55C

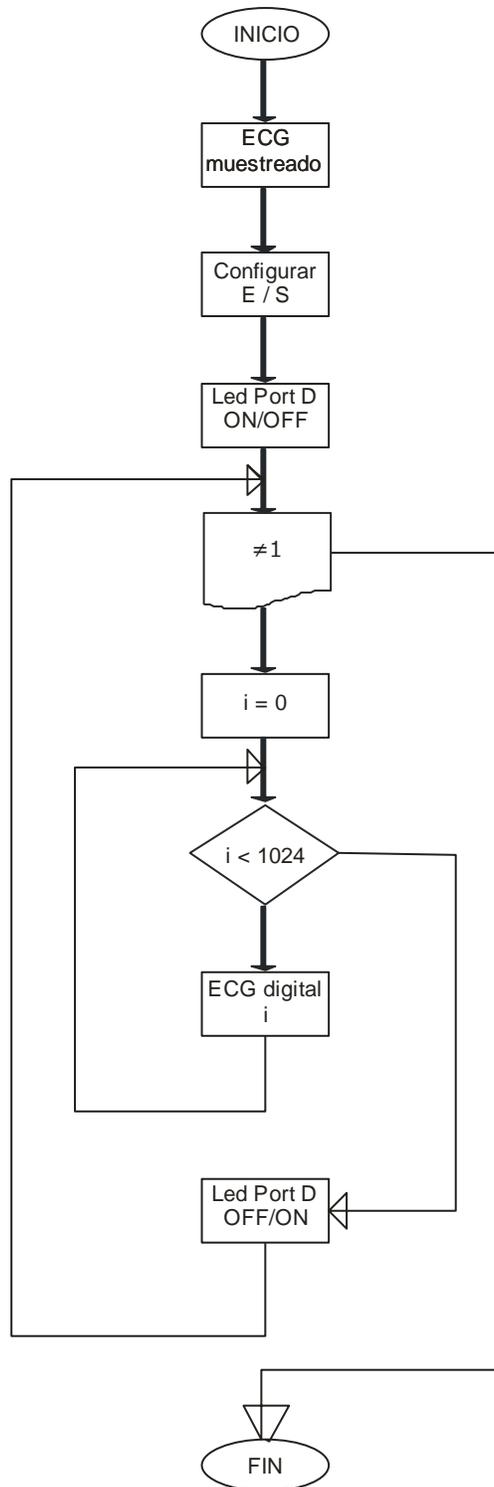
www.datasheetcatalog.org/datasheets/37/113628_DS.pdf

2N3055

www.datasheetcatalog.org/datasheet/motorola/2N3055.pdf

Apéndice B

Diagrama de flujo




```
while (1){          //Ciclo infinito

    for(i=0; i<1024; i++) {
        var = ecg[i];
        PORTC=var;
        Delay_ms(1);

    }
    PORTD.F1 = PORTD.F1 ^1;
}
}
```

BIBLIOGRAFÍA



- [L01] Alvarado Moya, Pablo, *Señales y sistemas, fundamentos matemáticos*. Centro de desarrollo de material bibliográfico, Costa Rica, 2005. 337 p.p.
- [L02] Boylestad, Robert L, *Introducción al análisis de circuitos*. Pearson Prentice Hall, E. U. A., 2004. 1228 p.p.
- [L03] Clavijo Mendoza, J. R. *Diseño y simulación de sistemas microcontrolados en lenguaje C*. Mikro C, Colombia, 2011.
- [L04] Coughlin, Robert F., *Amplificadores operacionales y circuitos integrados lineales*. Prentice Hall Hispanoamericana, México, 1999.
- [L05] Deitel, Harvey M., *Cómo programar en C/C++*. Prentice Hall Hispanoamericana, México, 1995.
- [L06] Dubin, D., *Electrocardiografía práctica*. McGraw-Hill, México, 1974.
- [L07] Edminister, Joseph, *Circuitos Eléctricos*. McGraw-Hill Interamericana, México, 1997. 575 p.p.
- [L08] Fletcher, William, *An engineering aproach to digital desing*. Ed Prentice Hall, E. U. A., 1997.
- [L09] García Breijo, Eduardo, *Compilador C CCS y simulador PROTEUS para microcontroladores PIC*. Alfaomega, México, 2011. 265 p.p.
- [L10] García Fernández, M. A., *Ecocardiografía básica*. ENE ediciones, España, 2000.
- [L11] Guyton, Arthur C., *Tratado de fisiología médica*. Elsevier Saunders, España, 2003.
- [L12] Hayt, William Hart, *Análisis de circuitos en Ingeniería*. McGraw-Hill, México, 2007.
- [L13] Malvino, Albert Paul, *Principios de electrónica*. McGraw-Hill Interamericana, México, 2000.
- [L14] Mano, M. Morris, *Fundamentos de diseño lógico y computadoras*. Prentice Hall Hispanoamericana, México, 1998.
- [L15] Pozo, Julio, y Chorbadian, Rosa, *Tópicos de electricidad y magnetismo*. Universidad Diego Portales, Chile, 2006. 256 p.p.
- [L16] Savant, C. J., *Diseño electrónico: circuitos y sistemas*. Pearson Educacion, México, 2000.

- [L17] Schilling, Donald L., *Circuitos electrónicos discretos e integrados*. Alfaomega Marcombo, México, 1991.
- [L18] Taub, Herbert, *Digital integrated electronics*. McGraw-Hill, México, 1977.