



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN ELECTRÓNICA E
INSTRUMENTACIÓN

TEMA: DISEÑO Y CONSTRUCCIÓN DE UN
ELECTROESTIMULADOR Y ELECTROMIOGRAMA CON
COMUNICACIÓN INALÁMBRICA PARA LA DETECCIÓN Y
TRATAMIENTO DE ENFERMEDADES NEUROMUSCULARES

AUTORES: KARINA ETELVINA TUTILLO TAIPE

NELSON ROLANDO SINCHIGUANO CHILIQUE

DIRECTOR: ING. JOSÉ GIOBERTY BUCHELI ANDRADE

CODIRECTOR: ING. EDDIE EGBERTO GALARZA
ZAMBRANO

LATACUNGA

2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICADO

Ing. José Bucheli (DIRECTOR)
Ing. Eddie Galarza (CODIRECTOR)

CERTIFICAN

Que el trabajo titulado “DISEÑO Y CONSTRUCCIÓN DE UN ELECTROESTIMULADOR Y ELECTROMIOGRAMA CON COMUNICACIÓN INALÁMBRICA PARA LA DETECCIÓN Y TRATAMIENTO DE ENFERMEDADES NEUROMUSCULARES”, realizado por la Señorita Karina Etelvina Tutillo Taipe y el Señor Nelson Rolando Sinchiguano Chiliquinga, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas ESPE.

Debido a que constituye un trabajo de excelente contenido científico y aplicable para el desarrollo profesional, SI recomiendan su aplicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan al Señor Nelson Rolando Sinchiguano Chiliquinga y a la Señorita Karina Etelvina Tutillo Taipe, que lo entregue al Ing. Franklin Silva, en su calidad de Director de Carrera.

Latacunga, Agosto del 2015

Ing. José Bucheli
DIRECTOR

Ing. Eddie Galarza
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Nosotros:

KARINA ETELVINA TUTILLO TAIPE
NELSON ROLANDO SINCHIGUANO CHILIQUE

DECLARAMOS QUE

El proyecto de grado denominado “DISEÑO Y CONSTRUCCIÓN DE UN ELECTROESTIMULADOR Y ELECTROMIOGRAMA CON COMUNICACIÓN INALÁMBRICA PARA LA DETECCIÓN Y TRATAMIENTO DE ENFERMEDADES NEUROMUSCULARES”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Agosto del 2015

Karina Etelvina
Tutillo Taipe
C.C: 0503321358

Nelson Rolando
Sinchiguano Chilique
C.C: 0502874902

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros:

KARINA ETELVINA TUTILLO TAIPE
NELSON ROLANDO SINCHIGUANO CHILIQUE

Autorizamos a la UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE, la publicación, en la biblioteca virtual de la Institución del Trabajo “DISEÑO Y CONSTRUCCIÓN DE UN ELECTROESTIMULADOR Y ELECTROMIOGRAMA CON COMUNICACIÓN INALÁMBRICA PARA LA DETECCIÓN Y TRATAMIENTO DE ENFERMEDADES NEUROMUSCULARES”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, Agosto del 2015

Karina Etelvina
Tutillo Taipe
C.C: 0503321358

Nelson Rolando
Sinchiguano Chilique
C.C: 0502874902

DEDICATORIA

Dedico este proyecto de tesis principalmente a Dios y a la Santísima Virgen del Quinche, por darme la vida y sus bendiciones para permitirme concluir un paso más en mi vida profesional.

A a mis Padres, Oswaldo y Etelvina, que con su esfuerzo, perseverancia, amor y apoyo incondicional me han ayudado a cumplir con este sueño.

A mis hermanos, quienes de una forma u otra han abierto paso para que el recorrido hacia el éxito sea fructífero, con su ejemplo, amistad, amor y consejos

Karina Etelvina

El presente trabajo va dedicado a la Stma. Virgen del Quinche, por regalarme la vida y darme la sabiduría suficiente para culminar con éxito cada objetivo que me propongo.

A mis padres quienes han sido mis consejeros y con su ejemplo me han inculcado valores éticos y morales que me han permitido llegar a este momento tan especial en mi vida.

A mi hermana, amigos y familiares que de una u otra manera supieron brindarme su apoyo desinteresadamente para seguir adelante y culminar mi carrera con éxito.

Nelson Rolando

AGRADECIMIENTO

A Dios, por brindarnos sabiduría y fortaleza durante nuestra carrera universitaria, a la Stma. Virgen del Quinche, por ser la luz que guía nuestro camino y por las bendiciones recibidas en el transcurso de nuestra vida.

A nuestras familias, nuestro eterno agradecimiento, por su incondicional apoyo, sin ellos este logro no hubiera sido posible.

Al Ing. José Bucheli Director e Ing. Eddie Galarza Codirector, por su colaboración, tiempo y acertada guía durante el desarrollo del proyecto de fin de carrera.

De manera especial al PhD. Luis Enrique Sánchez mentalizador de este sueño, que ha sido el pilar fundamental en la realización del proyecto, quien aportó con sus conocimientos y experiencia para llevar a feliz término este proyecto incentivado por él.

Y a todas las personas que de una u otra forma aportaron para que el presente proyecto haya llegado a su feliz término.

Karina y Nelson

ÍNDICE DE CONTENIDO

CARÁTULA	i
CERTIFICADO	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xiii
RESUMEN	xiv
ABSTRACT	xv
INTRODUCCIÓN	xvi
CAPÍTULO I.....	1
1. FUNDAMENTACIÓN TEÓRICA	1
1.1. Antecedentes	1
1.1.1. Históricos	1
1.1.2. La actividad eléctrica del músculo	1
1.2. Técnica de Electromiografía.....	1
1.2.1. La Electromiografía (EMG)	1
1.2.2. Principio Fisiológico del Electromiograma	2
1.2.3. El Electromiógrafo	2
1.3. Procedimiento del Electromiograma.....	3
1.3.1. Contracciones Musculares	3
1.3.2. Electromiograma Normal	5
1.3.3. Electromiograma patológico	10
1.4. Electrodo.	12
1.4.1. Electrodo de captación de aguja.....	13
1.4.2. Electrodo de captación de superficie	13
1.5. Electroterapia y electroestimulación.....	14
1.5.1. Corrientes más utilizadas en electroestimulación	15

1.5.2.	Dosis adecuada para electroterapia	20
1.6.	Microcontroladores.....	21
1.6.1.	Arduino uno	22
1.7.	Plataforma de desarrollo Java.....	24
1.8.	Sistema Android.....	25
1.9.	Android Studio.....	26
1.10.	Comunicación inalámbrica.....	26
1.10.1.	Estándar Bluetooth (802.15.1).....	27
CAPÍTULO II.....		32
2. DISEÑO E IMPLEMENTACIÓN		32
2.1.	Implementación del EMG	32
2.1.1.	Electrodos.....	33
2.1.2.	Tarjeta de adquisición de la señal bioeléctrica	33
2.2.	Implementación del electroestimulador	37
2.2.1.	Diseño del módulo de alto voltaje	38
2.2.2.	Diseño del módulo de control	43
2.2.3.	Configuración de las funciones del electroestimulador	44
2.3.	Módulo Bluetooth HC-05 para la Comunicación Serial	46
2.4.	Diseño del Software	49
2.4.1.	Entorno de desarrollo para Arduino	49
2.4.2.	Desarrollo de la aplicación en Android	51
CAPÍTULO III.....		60
3. PRUEBAS Y RESULTADOS OBTENIDOS		60
3.1.	Pruebas del EMG	60
3.1.1.	Adquisición de señales de EMG	61
3.1.2.	Prueba de contracción isométrica.....	61
3.1.3.	Prueba de contracción isotónica.....	62
3.1.4.	Prueba de músculo en fatiga	62
3.2.	Análisis de resultados de las pruebas del EMG	62
3.2.1.	Resultados de Adquisición de señales de EMG	62

3.2.2.	Resultados de prueba contracción isométrica	66
3.2.3.	Resultados de pruebas de contracción isotónica.....	67
3.2.4.	Resultados de pruebas de músculo en fatiga	69
3.3.	Comparación de señales de EMG con el Módulo KL-730002..	70
3.3.1.	Experimento de contracción isométrica	70
3.3.2.	Experimento de contracción isotónica	71
3.4.	Pruebas del Electroestimulador	72
3.4.1.	Función Bíceps	72
3.4.2.	Función Flexores	73
3.4.3.	Función Muslos.....	75
3.4.4.	Función Peroné	76
3.4.5.	Pruebas de intensidades del ectroestimulador	78
CAPÍTULO IV	81
4. ANÁLISIS ECONÓMICO	81
4.1.	Alcances y limitaciones	84
4.1.1.	Alcances	84
4.1.2.	Limitaciones.....	84
4.2.	Proyectos futuros	85
CAPÍTULO V	86
5. CONCLUSIONES Y RECOMENDACIONES	86
5.1.	Conclusiones.....	86
5.2.	Recomendaciones.....	88
REFERENCIAS BIBLIOGRÁFICAS	89
ANEXOS	91

ÍNDICE DE FIGURAS

Figura 1.1 Contracciones Musculares.....	3
Figura 1.2 Contracción concéntrica	4
Figura 1.3 Contracción excéntrica	4
Figura 1.4 Contracción isométrica	5
Figura 1.5 Registro del músculo en reposo.....	6
Figura 1.6 Interferencias en registro de músculo en reposo	6
Figura 1.7 Registro de contracción muscular leve	7
Figura 1.8 Potenciales de Unidad Motora Polifásica.....	8
Figura 1.9 Formas de Registro Electromiográfico.....	9
Figura 1.10 Potenciales de Fasciculación.....	10
Figura 1.11 Potenciales de Fibrilación.....	11
Figura 1.12 Ondas positivas patológicas	12
Figura 1.13 Potenciales polifásicos gigantes	12
Figura 1.14 Electrodo de captación de aguja.....	13
Figura 1.15 Corriente galvanica	16
Figura 1.16 Corriente galvánica interrumpida	17
Figura 1.17 Corriente alterna	17
Figura 1.18 Comportamiento de electrones en corriente alterna	18
Figura 1.19 Corriente interrumpida alterna	18
Figura 1.20 Corriente modulada en amplitud.....	19
Figura 1.21 Corriente Kotz	20
Figura 1.22 Arduino uno.....	22
Figura 1.23 Shield ECG / EMG	24
Figura 1.24 Clasificación de las comunicaciones inalámbricas.....	27
Figura 1.25 Estándares de comunicación inalámbrica.....	27
Figura 1.26 Aplicaciones Bluetooth.....	28
Figura 1.27 Estructura piconet.....	30
Figura 1.28 Estructura de scatternet	30
Figura 2.1 Diagrama de bloques de electromiógrafo inalámbrico	32
Figura 2.2 Conversion VCCD a VCCA.....	34
Figura 2.3 Etapa de recepción de señal en Shield EKG/EMG	34
Figura 2.4 Amplificador de instrumentación de Shield EKG/EMG	35

Figura 2.5 Filtros analógicos de la Shield EKG/EMG	37
Figura 2.6 Diagrama de bloques de electroestimulador inalámbrico	38
Figura 2.7 Elevador de voltaje	38
Figura 2.8 Variación de la tensión de salida según el ciclo de trabajo ...	39
Figura 2.9 Etapa de ganancia	41
Figura 2.10 Modulador de amplitud	42
Figura 2.11 Generador de pulsos diferenciales	43
Figura 2.12 Conexión ATMEGA328P	43
Figura 2.13 Conexión Bluetooth con Arduino.....	47
Figura 2.14 Direcciones MAC de módulos Bluetooth.....	48
Figura 2.15 Entorno de programación arduino.....	50
Figura 2.16 Ventana de bienvenida de Android Studio.....	52
Figura 2.17 Pantalla de configuración de un nuevo proyecto	52
Figura 2.18 Configuración de plataformas de Android Studio.....	53
Figura 2.19 Creación de nueva actividad proyecto Android.....	53
Figura 2.20 Configuración de actividad en blanco proyecto Android	54
Figura 2.21 Pantalla Principal de Android Studio.....	54
Figura 2.22 Diagrama de flujo de Actividad Principal.....	55
Figura 2.23 Layout de Actividad Principal	56
Figura 2.24 Diagrama de flujo de Actividad Electromiógrafo	56
Figura 2.25 Layout de Actividad Electromiógrafo.....	57
Figura 2.26 Layout de Actividad Electroestimulador	57
Figura 2.27 Diagrama de flujo de Actividad Electroestimulador	59
Figura 3.1 Posición de electrodos	61
Figura 3.2 Preparación del paciente	63
Figura 3.3 Ubicación de los electrodos en el paciente.....	63
Figura 3.4 Señal con ruido	64
Figura 3.5 Paciente EMG con ruido	64
Figura 3.6 EMG sin peso	65
Figura 3.7 Paciente EMG sin Peso	65
Figura 3.8 EMG con peso de 1Kg.....	66
Figura 3.9 Contracción isométrica	67
Figura 3.10 Contracción isotónica concéntrica	68
Figura 3.11 Contracción isotónica excéntrica	69

Figura 3.12 Contracción isotónica concéntrica y excéntrica	69
Figura 3.13 Músculo en fatiga.....	70
Figura 3.14 Contracción isométrica con Módulo KL-73002.....	70
Figura 3.15 Contracción isométrica dispositivo EMG.....	71
Figura 3.16 Contracción isotónica con Módulo KL-73002.....	71
Figura 3.17 Contracción isotónica dispositivo EMG	71
Figura 3.18 Ubicacion de electrodos función Bíceps	72
Figura 3.19 Onda Alfa función Bíceps.....	72
Figura 3.20 Onda alfa y beta de funcion bíceps.....	73
Figura 3.21 Ubicacion de electrodos función flexores.....	73
Figura 3.22 Onda Alfa función Flexores.....	74
Figura 3.23 Onda alfa y beta de funcion flexores.....	74
Figura 3.24 Ubicacion de electrodos función muslos	75
Figura 3.25 Onda Alfa función muslos	75
Figura 3.26 Onda alfa y beta de funcion flexores.....	76
Figura 3.27 Ubicacion de electrodos función peroné	76
Figura 3.28 Onda Alfa función peroné	77
Figura 3.29 Onda alfa y beta de funcion peroné	77
Figura 3.30 Paciente de electroestimulacion	78
Figura 3.31 Nivel de intensidad 1 para electroestimulador	79
Figura 3.32 Nivel de intensidad 2 para electroestimulador	79
Figura 3.33 Nivel de intensidad 3 para electroestimulador	80
Figura 4.1 Compex My Sport 500	81

ÍNDICE DE TABLAS

Tabla 1.1 Características del potencial de unidad motora simple	7
Tabla 1.2 Características del Potencial de Unidad Motora Polifásica	9
Tabla 1.3 Clasificación de corrientes	15
Tabla 1.4 Características de la tarjeta Arduino uno	23
Tabla 1.5 Características Shield EKG-EMG.....	24
Tabla 2.1 Voltaje y resistencia según el nivel de intensidad	41
Tabla 2.2 Ajustes Función Bíceps.....	45
Tabla 2.3 Función Flexores.....	45
Tabla 2.4 Ajustes Función Muslos	46
Tabla 2.5 Ajustes Función Peroné	46
Tabla 2.6 Especificaciones técnicas del Módulo Bluetooth HC-05	48
Tabla 2.7 Botones de acceso rápido de Arduino IDE	50
Tabla 4.1 Comparación Compex My Sport 500 vs. Prototipo	82
Tabla 4.2 Costo total Electromiograma y Electroestimulador.....	82

RESUMEN

El presente proyecto consiste en el diseño y construcción de un electroestimulador y electromiograma con comunicación inalámbrica para la detección y tratamiento de enfermedades neuromusculares, este equipo consta de sensores que permiten adquirir señales bioeléctricas del cuerpo humano generadas por los músculos esqueléticos con la ayuda de electrodos no invasivos permitiendo la detección de las enfermedades neuromusculares. Además para lograr la rehabilitación del grupo muscular afectado permite enviar impulsos eléctricos al nervio motor para desencadenar la excitación de músculo. Las señales son transmitidas inalámbricamente vía Bluetooth para la visualización respectiva de los datos en un dispositivo móvil que contenga la plataforma Android. Para la adquisición de la señal electromiográfica se utiliza el módulo shield EKG/EMG que permite obtener la señal bioeléctrica minimizando el ruido. En el desarrollo del electroestimulador se diseñó una etapa de alto voltaje para conseguir los diferentes tipos de corrientes de estimulación eléctrica. Este dispositivo cumple a cabalidad con los requerimientos médicos, sin peligro para su utilización en humanos tanto en el diagnóstico de enfermedades neuromusculares como para el tratamiento en rehabilitación y fisioterapia ayudando a los centros de Fisioterapia y Rehabilitación, siendo más accesible para los pacientes.

PALABRAS CLAVE:

- ❖ **ELECTROMIOGRAFÍAS**
- ❖ **ELECTROESTIMULADORES**
- ❖ **DISPOSITIVOS MÓVILES**
- ❖ **CONEXIONES INALÁMBRICAS**

ABSTRACT

This project involves the design and construction of a stimulator and EMG with wireless communication for the detection and treatment of neuromuscular diseases, this equipment consists of sensors to acquire bioelectric signals of the human body generated by skeletal muscles with the help of electrodes not invasive allowing detection of neuromuscular diseases. In addition to the rehabilitation of the affected muscle group to send electrical pulses to the motor nerve to muscle trigger arousal. The signals are transmitted wirelessly via Bluetooth to the respective data display on a mobile device containing the Android platform. To acquire the electromyographic signal the shield EKG / EMG module which allows for bioelectrical minimizing noise signal is used. In the development stage electrostimulator high voltage design for different types of electrical stimulation current. This device complies fully with medical requirements, safe for use in humans both in the diagnosis of neuromuscular diseases and treatment in rehabilitation and physical medicine helping Physiotherapy and Rehabilitation centers, being more accessible to patients.

KEYWORDS:

- ❖ **ELECTROMYOGRAPHY**
- ❖ **ELECTRO STIMULATOR**
- ❖ **MOBILE DEVICES**
- ❖ **WIRELESS CONNECTIONS**

INTRODUCCIÓN

En la sociedad actual la tecnología evoluciona a un ritmo acelerado en todos los campos, incluido el campo de la medicina, en donde la electrónica ha tomado un papel fundamental, presentando avances de gran nivel logrando una mejora en lo referente al tratamiento y diagnóstico de diversas enfermedades.

Existe mucho interés de la ciencia en buscar herramientas para el tratamiento de diversas enfermedades neuromusculares, una de estas herramientas es la electroestimulación, la cual a lo largo de los años ha evolucionado, no solamente en el ámbito de la medicina sino también en la ingeniería, mediante la ayuda de equipos médicos se puede tratar de manera eficaz diversas enfermedades.

En Ecuador más de 112.072 [1] personas se ven afectadas por alguna forma de enfermedad neuromuscular, sin embargo, se tienen un índice mínimo en lo referente al desarrollo y fabricación de instrumentos médicos, por lo tanto hospitales y centros médicos los adquieren en el mercado internacional a precios altos que no brindan las suficientes garantías ni funcionalidades de usos.

El presente proyecto plantea la construcción de un equipo biomédico de electroestimulación, el cual consiste en adquirir señales Bioeléctricas del cuerpo Humano mediante el EMG con la ayuda de electrodos no invasivos y enviar impulsos eléctricos al nervio motor para desencadenar la excitación de los músculos.

Con este proyecto se pretende ayudar a centros de Fisioterapia y Rehabilitación, para aprovechar al 100% los equipos con un menor costo de adquisición, conllevando un costo más accesible para los pacientes acorde a la funcionalidad del equipo.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Antecedentes

1.1.1. Históricos

Dentro de los estudios acerca de la electroestimulación y electromiografía se puede rescatar los trabajos realizados por el médico y científico italiano Luigi Galvani, quien en 1780, descubrió que la aplicación de la corriente eléctrica en una preparación neuromuscular de las patas de una rana, producía la contracción muscular, llegando a la conclusión de la existencia de electricidad en la musculatura de los organismos vivos. En el siglo XIX los trabajos de Galvani dieron la pauta para Duchenne de Boulogne dedicado al estudio de la estimulación eléctrica, descubre que era posible estimular los músculos por encima de la piel y que la aplicación sistemática permitía estudiar la dinámica del músculo estriado. Duchenne construyó un equipo de estimulación neuromuscular, inicialmente con fines terapéuticos, después investigativos y diagnósticos.

1.1.2. La actividad eléctrica del músculo

Uno de los exámenes comúnmente realizados que brinda información importante acerca del estado de los músculos, es la Electromiografía (EMG), que hace referencia a un examen que evalúa y proporciona información acerca de la actividad eléctrica de los músculos esqueléticos, comprendiendo su estado fisiológico.

1.2. Técnica de Electromiografía

1.2.1. La Electromiografía (EMG)

La Electromiografía (EMG), es el registro del microfenómeno cinético de la contracción muscular, por lo tanto, es el registro gráfico de la despolarización y repolarización de la membrana muscular, como expresión del proceso electroiónico de la unidad motora. [2]

La electromiografía es una de las principales técnicas para adquirir información del cuerpo humano, en la cual se realiza el estudio electrofisiológico del sistema neuromuscular, es decir el estudio de los potenciales eléctricos generados por los músculos durante el movimiento.

El EMG tiene por objeto el estudio de la actividad muscular a través de la señal eléctrica producida por los mismos. Con esta prueba se puede identificar si las alteraciones neuromusculares se deben al músculo o a las fibras nerviosas. Para ello se analiza básicamente:

- ❖ La amplitud de las corrientes eléctricas.
- ❖ El número de fibras musculares que se contraen.
- ❖ El tiempo que tardan en contraerse.
- ❖ El tiempo que se mantienen contraídas.

1.2.2. Principio Fisiológico del Electromiograma

Los fenómenos registrados en el estudio del electromiograma reciben el nombre de potenciales de unidad motora o potenciales de acción muscular. Varían en número, forma, amplitud y duración en relación con el estado funcional normal o patológico del músculo. [2]

La unidad anatómica es la llamada fibra muscular y la unidad funcional es la unidad motora. Una unidad motora es un grupo de fibras musculares inervado por una única motoneurona de la médula espinal o de un núcleo motor del tallo cerebral. Cuando una unidad motora se activa por simulación, se genera una forma de onda de potencial con una amplitud de 20-2000 μV , una frecuencia de descarga de 6-30 Hz, y un intervalo de 3 - 10. Por lo tanto, la contracción de las fibras musculares conduce a una señal de potencial con mayor amplitud y una frecuencia más alta que se llama la electromiografía (EMG).

1.2.3. El Electromiógrafo

Es un equipo utilizado para el estudio del EMG, está diseñado para la captación y el registro de los potenciales electroiónicos producidos en la

membrana de la fibra muscular, y de la respuesta muscular a nivel de un área de exploración.

1.3. Procedimiento del Electromiograma

El estudio EMG, tiene como propósito investigar la degeneración del tejido nervioso y los procesos patológicos de la fibra muscular mediante el registro de sus manifestaciones a nivel del músculo.

1.3.1. Contracciones Musculares

La contracción muscular es un proceso fisiológico, que por lo general es dirigida y controlada por el cerebro en la cual los músculos según la tensión, se estiran o se acortan permitiendo producir fuerza motora.

Las contracciones musculares surgen como respuesta a un estímulo, la Figura 1.1 expone los diversos tipos de contracciones musculares que existen.

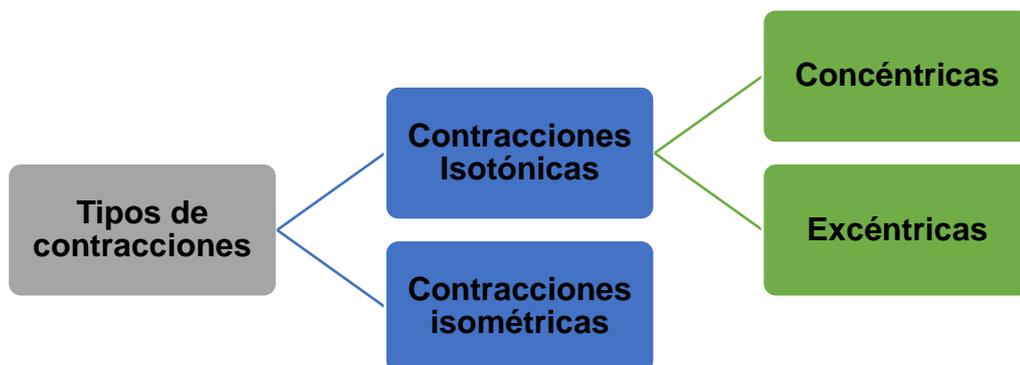


Figura 1.1 Contracciones Musculares

a. Contracciones Isotónicas

Desde el punto de vista fisiológico, las contracciones isotónicas son aquellas contracciones en la que las fibras musculares además de contraerse, modifican su longitud.

- ❖ **Contracciones Concéntricas:** Tiene lugar cuando un músculo desarrolla una tensión suficiente para superar una resistencia, de tal forma que éste se acorta y moviliza una parte del cuerpo venciendo dicha resistencia. La Figura 1.2 expone un ejemplo, al acercar una mancuerna al hombro, se produce un acortamiento muscular concéntrico en el brazo.



Figura 1.2 Contracción concéntrica

Fuente: [3]

- ❖ **Contracciones Excéntricas:** Se producen cuando una resistencia dada es mayor que la tensión ejercida por un músculo determinado, de forma que éste se alarga. En la Figura 1.3 se continúa con el ejemplo anterior, la contracción muscular excéntrica se produce al llevar la mancuerna desde el hombro hacia adelante.

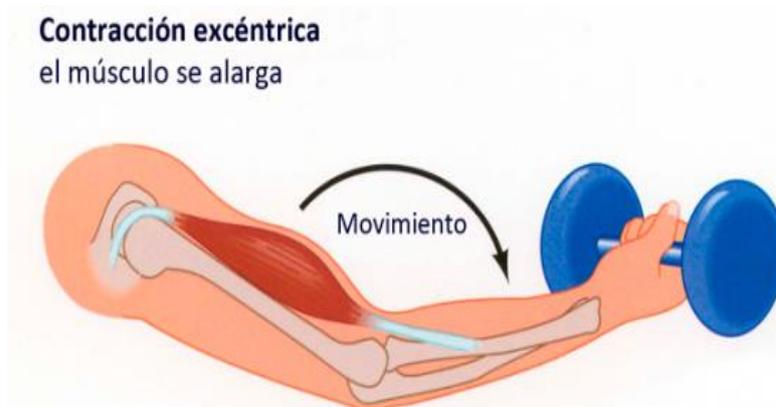


Figura 1.3 Contracción excéntrica

Fuente: [3]

b. Contracciones Isométricas

La contracción muscular isométrica se basa en la estabilidad del músculo, es decir, al contrario que ocurre en la contracción muscular isotónica, el músculo permanece estático, sin acortarse ni alargarse, pero aunque permanece estático, sigue generando tensión. La Figura 1.4 muestra un ejemplo en el cual se sostiene la mancuerna de manera constante, pese a que el brazo permanece estático, se genera tensión para evitar que la mancuerna caiga al suelo.

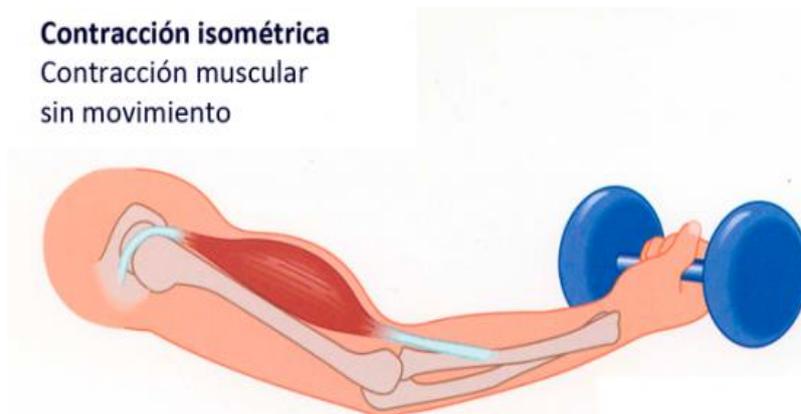


Figura 1.4 Contracción isométrica

Fuente: [3]

1.3.2. Electromiograma Normal

a. Músculo en reposo

Al colocar los electrodos, normalmente aparece en la pantalla una sola línea que corresponde a una línea isoelectrica que representa el estado de equilibrio electroiónico del músculo y de su membrana. La línea isoelectrica se acompaña de un silencio eléctrico por la ausencia de sonidos.

La Figura 1.5 muestra el registro de un músculo en reposo, en la cual existe una señal continua llamada línea isoelectrica.

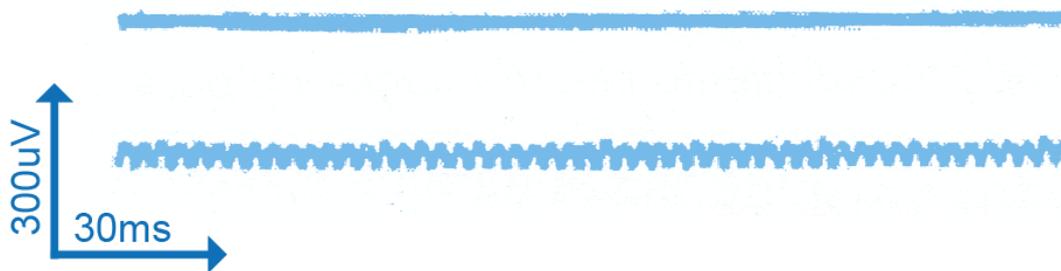


Figura 1.5 Registro del músculo en reposo

Se caracteriza por la presencia de una que otra onda positiva monofásica y a veces difásica, de una duración muy corta, menos de 100 ms. Son de voltaje muy variable, su forma en ocasiones puede ser bifásica, se presenta con una frecuencia de 2 a 5 por segundo.

Este tipo de potenciales también son conocidos como ondas agudas u ondas en "V". Aparecen en el reposo, en la relajación y a veces en la contracción ligera. Rara vez se observan descargas de potenciales en serie o trenes con frecuencias superiores a los 100 por segundo. Sin embargo, la onda en "V", puede presentar el único hallazgo de actividad del músculo, por lo cual es necesario interpretarlo correctamente.

En este estado aparecen también los trazados de interferencias relacionados con los aparatos de electroterapia u otros equipos electrónicos ubicados cerca del área de exploración, las señales más usuales de este tipo de interferencia se muestran en la Figura 1.6.

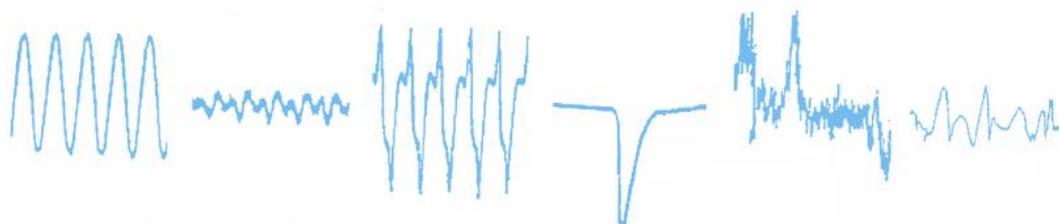


Figura 1.6 Interferencias en registro de músculo en reposo

b. Contracción muscular ligera

El paciente realiza una contracción isométrica leve, entonces la línea isoelectrica se borra y aparece un escaso número de Potenciales de Acción

Motora provocando el fenómeno conocido como Interferencia Parcial o Patrón de Ondas Parciales.

Este fenómeno permite analizar la amplitud, la forma y la duración de los potenciales de acción motora.

Por lo general el fenómeno o fase de interferencia parcial está constituido por un gran porcentaje de unidades trifásicas acompañadas de unidades bifásicas y escasas unidades polifásicas (menos las 10% de las polifásicas).

La Figura 1.7 muestra el registro de una contracción muscular leve, en donde: (a) representa a un registro electromiográfico normal ya que existe un potencial de acción al momento de realizar la contracción, en (b) se muestra un potencial de acción trifásico doble y finalmente en (c) se muestra un potencial de acción simple trifásico.

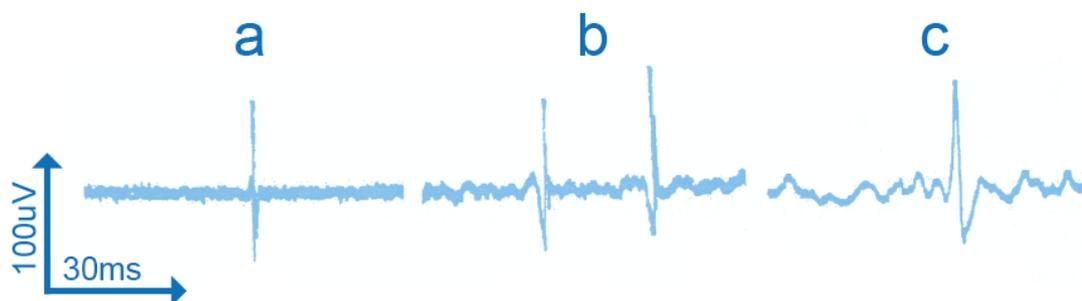


Figura 1.7 Registro de contracción muscular leve

La Tabla 1.1 muestra las características del potencial de unidad motora simple trifásica.

Tabla 1.1

Características del potencial de unidad motora simple.

Forma de la onda	2 a 4 fases, generalmente trifásica
Voltaje	100 a 5000 V.
Duración	2 a 10ms
Frecuencia	1 a 60 x segundo

c. Contracción Muscular Máxima

El paciente realiza una contracción muscular voluntaria lo más energética posible.

En ocasiones es necesario que utilice todos los grupos musculares agonistas del segmento explorado.

La línea isoelectrica del osciloscopio se borra totalmente apareciendo una gran cantidad de potenciales de unidad motora, producto del fenómeno de sumación de las fibras musculares despolarizadas y como consecuencia de la intervención de un gran número de unidades motoras que constituye ese grupo.

El fenómeno EMG que se reconoce como patrón de interferencia total y tiene que ver con la interferencia de la línea isoelectrica se muestra en la Figura 1.8, es posible analizar el tamaño, la forma predominante de los potenciales de acción motora, y la cantidad de los potenciales que forman el conjunto del mencionado fenómeno de interferencia total.

Normalmente la interferencia de la línea isoelectrica se realiza uniformemente.



Figura 1.8 Potenciales de Unidad Motora Polifásica

La Tabla 1.2 expone las características del Potencial de Unidad Motora Polifásica.

Tabla 1.2

Características del Potencial de Unidad Motora Polifásica

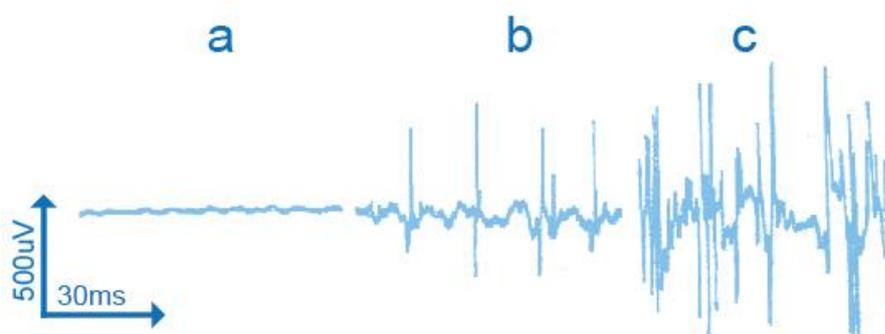
Forma de onda	5 a 25 fases.
Voltaje	20 a 5000 V.
Duración	2 a 25 ms
Frecuencia	20 a 30 x segundo.

d. Músculo en Relajación

El paciente realiza el proceso de relajación de su músculo y lo mantiene en reposo hasta que el médico termine la exploración. La exploración del periodo de relajación muscular debe realizarse después de la contracción mínima y de la contracción máxima.

Es la etapa más importante del estudio EMG. Normalmente debe reaparecer en la pantalla osciloscópica la línea isoelectrica acompañada del mencionado silencio eléctrico. La presencia de cualquier otro fenómeno en este periodo debe ser considerada como patología.

Las formas de Registro Electromiográfico obtenidas en las diferentes etapas del examen se muestran en la Figura 1.9, en donde (a) es el silencio eléctrico del músculo en reposo, (b) es el registro de la contracción muscular leve con un patrón e interferencia racial, (C) Registro de contracción muscular máxima con un patrón de interferencia total con algunos potenciales de tipo polifásico.

**Figura 1.9 Formas de Registro Electromiográfico**

1.3.3. Electromiograma patológico

a. Potenciales de fasciculación.

Se los conocen también como mioquimias. Se presentan durante el reposo y la relajación. Se producen espontáneamente, es decir, no están controlados por voluntad.

Los potenciales de fasciculación son visibles en el sujeto y se manifiesta por una sensación subjetiva y objetiva de reptación por debajo de la piel. El paciente y el examinador aprecian un movimiento involuntario similar al "tic" nervioso de los párpados.

La forma del potencial de fasciculación se muestra en la Figura 1.10, el cual registrado en la pantalla puede ser trifásico o polifásico, su número, amplitud, duración, ritmo, son también variables.

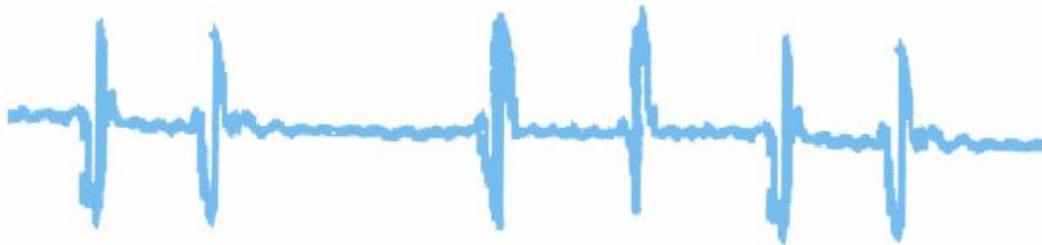


Figura 1.10 Potenciales de Fasciculación

b. Potenciales de fibrilación.

Los potenciales de fibrilación tienen una duración muy corta, 1 ms o menos y son bifásicos.

La Figura 1.11 muestra los potenciales de fibrilación, en donde la primera fase es siempre positiva y se grafica por encima de la línea isoelectrica, y la segunda es negativa y se grafica por debajo de la línea isoelectrica.

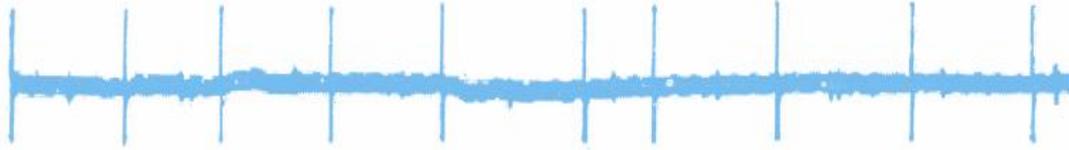


Figura 1.11 Potenciales de Fibrilación

Una observación detenida del fenómeno puede demostrar la presencia de una tercera fase, casi imperceptible. La amplitud de la fibrilación es de 10 a 600V, generalmente por debajo de 100V, y se repite con una frecuencia regularmente de 2 a 30 por segundo.

c. Potenciales de ondas positivas

Son potenciales espontáneos que aparecen irregularmente pero mantienen su forma constante. Tiene una duración de hasta 100 ms, se produce con una frecuencia de 2 a 100 por segundo.

Se producen por la presencia de fibras nerviosas dañadas, como se muestra en la Figura 1.12. La onda positiva conocida como onda aguda u onda (V invertida), por lo general es monofásica y se registra por encima de la línea isoelectrica.

La onda aguda aparece en reposo, la relajación e inclusive en la contracción muscular ligera, en los procesos que cursan con denervación muscular. Puede acompañarse de una pequeña deflexión negativa a veces imperceptible. Comienza con un ascenso brusco seguido de un descenso lento hasta un poco por debajo de la línea isoelectrica.



Figura 1.12 Ondas positivas patológicas

d. Potenciales gigantes

Si la unidad motora tiene una amplitud mayor de 5 mv y hasta 10 mv, se trata de una unidad motora gigante o potencial gigante, tal como se muestra en la Figura 1.13.

Corresponde a un proceso de hipertrofia de la unidad motora que se ve abocada a suplir la función de sectores de su propio territorio que han dejado de funcionar.



Figura 1.13 Potenciales polifásicos gigantes

1.4. Electrodo.

Es un dispositivo de conducción para registrar un potencial eléctrico (electrodo de registro) o para aplicar una corriente eléctrica (electrodo estimulador). [4]

El electrodo recoge la actividad eléctrica del músculo, generalmente son desechables y de cloruro de plata-plata, se utilizan debido a sus propiedades

de estabilidad y reducción del ruido. Se conoce varios tipos de electrodos de captación, clasificándose en electrodos de aguja y los de superficie.

1.4.1. Electrodo de captación de aguja

Permiten registrar selectivamente la actividad de unas pocas unidades motoras o incluso de una sola de ellas, pudiendo realizarse un estudio preciso y exhaustivo del estatus neurológico.

La Figura 1.14 muestra los electrodos de captación de aguja, que son utilizados en un método invasivo que representa dolor para el paciente y requiere de alta experiencia técnica.



Figura 1.14 Electrodo de captación de aguja

Fuente: [4]

1.4.2. Electrodos de captación de superficie

Dispositivo conductor para la estimulación o registro eléctrico del músculo, colocado sobre una superficie de la piel.

Existen una relación indirecta entre el tamaño del electrodo y la intensidad de la corriente, a mayor tamaño del electrodo la intensidad de la corriente disminuye, a menor tamaño la intensidad de la corriente aumenta porque la cantidad de electrones se concentra más en una superficie menor.

Los electrodos cutáneos son los más utilizados ya que corresponden a un método no invasivo, que consiste en dispositivos autoadhesivos que recogen la actividad de la musculatura perineal.

Dependiendo del tamaño y la condición de la estimulación los electrodos se clasifican en electrodos monopolares y bipolares.

a. Electrodos monopolar

De aproximadamente 0.4 mm en su parte más gruesa y 18 mm cuadrados en la punta. Para el estudio es necesario utilizar adicionalmente un electrodo dispersivo que actúa como un verdadero electrodo de tierra.

b. Electrodos bipolares.

Puede ser coaxial simple o coaxial doble de un diámetro similar a una aguja hipodérmica número 26 o 23 respectivamente. En el coaxial simple los dos alambres conductores se disminuyen en tal forma que uno de ellos termina en la punta.

1.5. Electroterapia y electroestimulación

La electroterapia se basa en la aplicación científica de los efectos de la corriente eléctrica de baja, mediana y alta frecuencia para el tratamiento de las afecciones y dolor que se producen en el sistema muscular.

Al aplicar corriente, se sustituye los estímulos fisiológicos naturales del organismo con estímulos eléctricos provocados desde el exterior con un adecuado potencial y forma de corriente.

Las formas de aplicación electroterapéutica se derivan ó parten de los dos tipos fundamentales, la corriente directa y la corriente alterna. La transformación de ellas en otros tipos de mayor complejidad, mediante sistemas rectificadores de corriente, ha hecho posible la transformación en una amplia gama de posibilidades electroterapéuticas. [2]

Para que esto suceda, se habrán provocado variaciones físicas de los trenes de impulsos, en frecuencia, duración de la onda, intensidad, forma y

longitud, parámetros que caracterizan individualmente a las diversas formas de estímulos. Le adjudican un efecto y determinan su indicación, contraindicaciones o su ventaja para el tratamiento de determinado trastorno.

1.5.1. Corrientes más utilizadas en electroestimulación

La Tabla 1.3 muestra la clasificación generalizada de las corrientes que componen toda la gama de electroterapia.

Tabla 1.3
Clasificación de corrientes

Clasificación	Sub clasificación
Efectos en el organismo	Efectos electroquímicos Efectos motores sobre nervio y músculo Efectos sensitivos sobre nervio sensitivo Efectos por aporte energético mejora metabolismo
Modo de aplicación	Puntos aislados Trenes o ráfagas Frecuencia fija
Frecuencia	Corrientes con modulación Baja Frecuencia de 0 a 1.000 Hz Media Frecuencia de 1.000 a 500.000 Hz Alta Frecuencia de 500.000 Hz hasta el límite de las ultravioletas.
Forma de onda	De flujo constante y mantenida polaridad De flujo interrumpido y mantenida polaridad De flujo constante e invertida polaridad De flujo interrumpido e invirtiendo la polaridad Modulando la amplitud Modulando la frecuencia Aplicación simultanea de dos o más corrientes

A continuación, se va a describir la clasificación de las corrientes según su forma de onda.

a. Flujo constante y mantenida polaridad: Galvánica o corriente continua

Es una corriente de valor constante, que consiste en aplicar corriente continua al organismo, subiendo lentamente la intensidad y manteniéndola

sin alteración, al mismo tiempo que no varía la polaridad durante toda la sesión.

La Figura 1.15 muestra la corriente galvánica, la cual no posee frecuencia, ya que se mantiene constante en el tiempo, sin embargo se la considera con una frecuencia infinita.

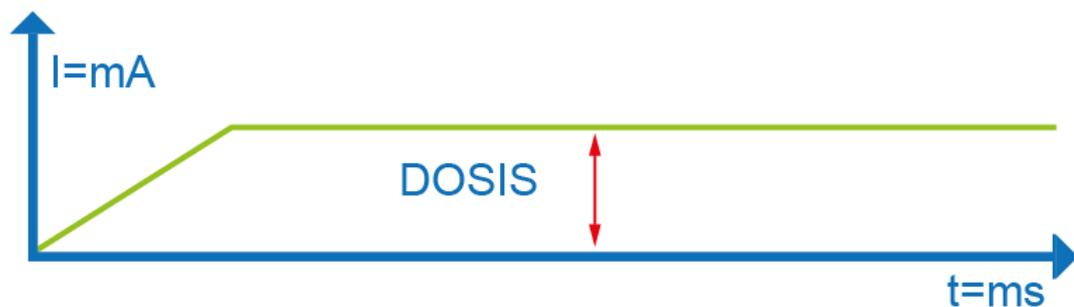


Figura 1.15 Corriente galvanica

Los electrones van a entrar en la materia viva por el electrodo negativo o cátodo y salen de ella por el polo positivo o ánodo, provocando efectos electrolíticos y electroforéticos sobre el organismo.

Los generadores capaces de producir una corriente galvánica son: las pilas eléctricas y baterías. En forma industrial se obtiene mediante circuitos electrónicos por medio de rectificadores o mediante la producción directa por medio de dinamos o magnetos.

b. Flujo interrumpido y polaridad mantenida: Galvánicas interrumpidas

Consiste en un corriente galvánica con polaridad mantenida, pero con interrupciones en su intensidad durante un lapso de tiempo denominado pausa, y devolver después esta corriente a su forma habitual. Los pulsos pueden ser de diferentes formas y frecuencias, así como agrupados en trenes, impulsos aislados, modulados o frecuencia fija. Son las más características de la señales de baja frecuencia.

La Figura 1.16 muestra una corriente galvánica interrumpida, la cual se obtiene al interrumpir una corriente galvánica.

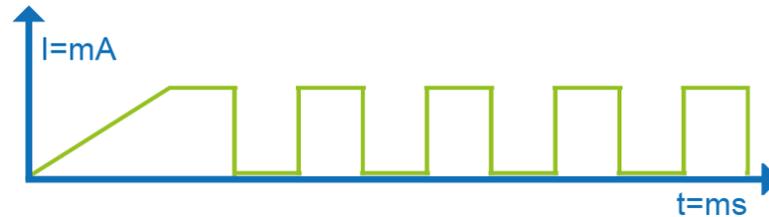


Figura 1.16 Corriente galvánica interrumpida

c. Flujo constante e invertida polaridad: Alterna

Es una corriente eléctrica sin interrupciones con alternancia rítmica en su polaridad, la forma más característica es la sinusoidal perfecta de mayor o menor frecuencia, empleada en media y alta frecuencia.

En la Figura 1.17 se muestran señales alternas de diferentes frecuencias, con lo cual se demuestra que una señal alterna puede variar en frecuencia manteniendo su flujo constante e invirtiendo su polaridad, de allí surgen las llamadas corrientes bifásicas.

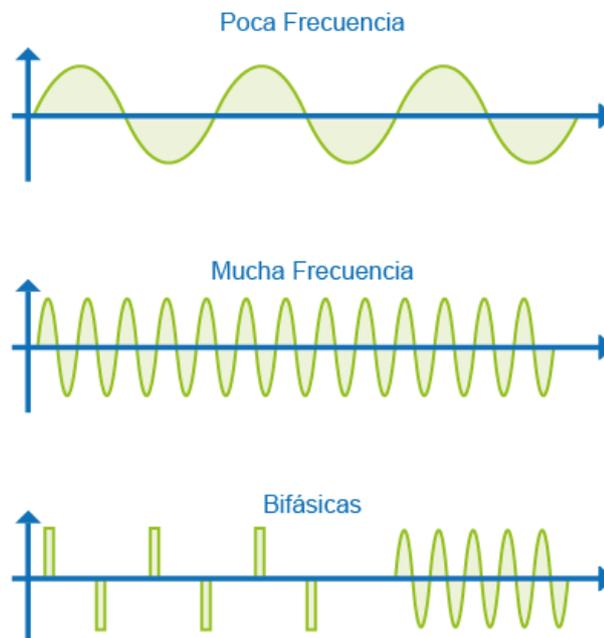


Figura 1.17 Corriente alterna

La Figura 1.18 muestra que debido a los cambios de polaridad, los electrones no se desplazan en un único sentido, sino que durante la onda

positiva lo harán en un sentido y durante el tiempo que dure la negativa lo harán en el sentido contrario. [5]

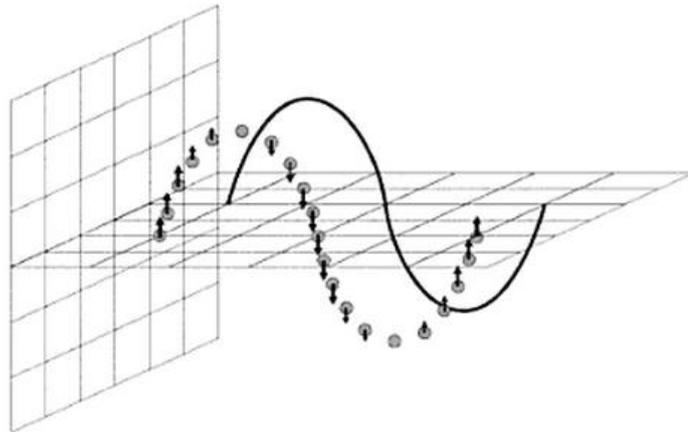


Figura 1.18 Comportamiento de electrones en corriente alterna
Fuente: [5]

d. Flujo interrumpido e invirtiendo la polaridad: Interrumpidas alternas

Consisten en aplicar interrupciones en una corriente alterna para formar pequeñas ráfagas o paquetes denominados pulsos o trenes de ondas. Una característica de este tipo de onda es la posibilidad de regular el efecto térmico manteniendo siempre la misma amplitud o potencia de las ondas, además de establecer tiempos de duración distintos para la misma frecuencia.

Los trenes de ondas al interrumpir una corriente alterna se muestran en la Figura 1.19.

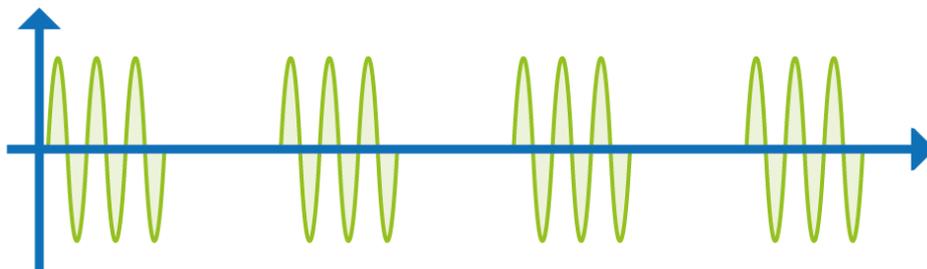


Figura 1.19 Corriente interrumpida alterna

e. Moduladas en amplitud: Interferenciales – TNS

Son corrientes normalmente de media frecuencia que se caracterizan por sufrir cambios constantes durante toda la sesión.

El fin terapéutico de este tipo de corriente es realzar los efectos de las formas simples, dotándolos de una modulación rítmica, para que el efecto estimulante de la forma base original sea modificado según la forma del ritmo o tren.

La Figura 1.20 muestra que la amplitud de la corriente modulada aumenta y disminuye al tiempo que simultáneamente oscila.

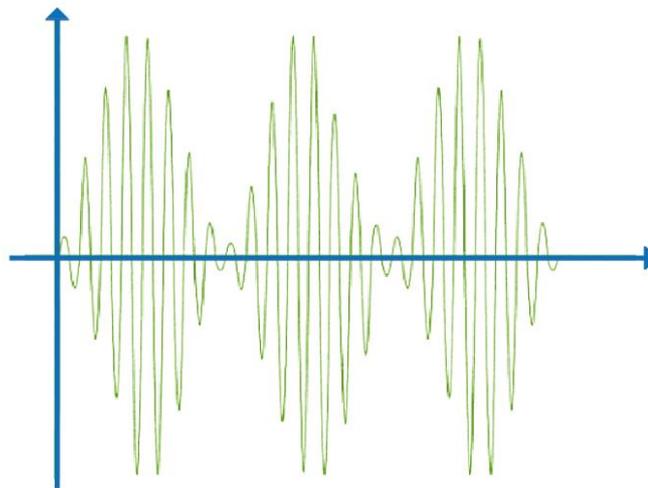


Figura 1.20 Corriente modulada en amplitud

Estas formas moduladas de corrientes tienen mucha importancia en las aplicaciones fisioterapéuticas, ya que las estimulaciones logradas por este doble sistema es más vigorosa al dar tiempo a la musculatura y a su inervador a que se recupere de la excitación recibida por la forma básica o fundamental entre cada pausa o módulo rítmico. [6]

Las corrientes moduladas en amplitud presentan algunas ventajas terapéuticas:

- ❖ La sensación de corriente es menos molesta.
- ❖ Se puede dosificar con mayor intensidad.
- ❖ Efecto estimulante mayor

f. Modulada en frecuencia: Corriente Kotz

Son corrientes que generan impulsos a una frecuencia variable entre dos límites, comienza con la frecuencia baja y se hace un barrido durante unos segundos por todas las frecuencias hasta llegar a la frecuencia alta y se repite el proceso.

La Figura 1.21 muestra que la señal empieza con la frecuencia mínima y a medida que transcurre el tiempo la frecuencia aumenta hasta llegar a la frecuencia máxima, luego de lo cual vuelve a disminuir a la frecuencia mínima.

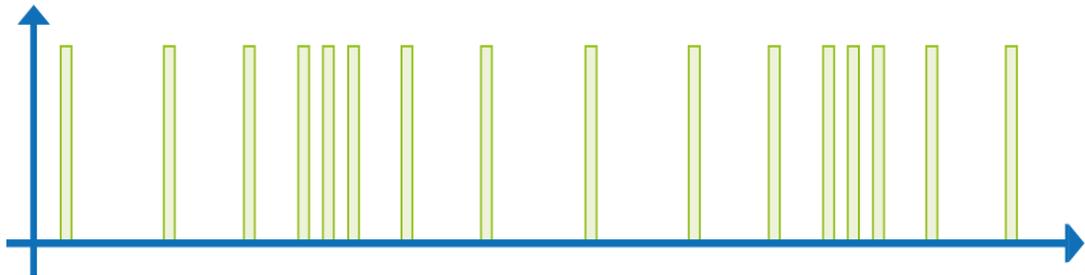


Figura 1.21 Corriente Kotz

Este método de tratamiento es seguro y no tiene efectos indeseables, alcanza profundidades significativas, permite el uso de grandes electrodos e invadir amplias masas musculares.

1.5.2. Dosis adecuada para electroterapia

Aún no se puede establecer de manera concreta el nivel de energía que necesitan determinados tejidos para que sean estimulados con una respuesta terapéutica, por lo tanto el valor de la dosis depende de los siguientes factores:

- ❖ Tamaño de electrodos
- ❖ Distancia entre los electrodos y tejidos
- ❖ Resistencia de la piel
- ❖ Rendimiento del electrodo
- ❖ Apreciación personal del paciente

El parámetro más importante es la subjetividad del paciente, sin embargo es posible determinar una intensidad media por unidad de superficie basada en cada centímetro cuadrado de superficie corporal que recibe la corriente, soporte siempre la misma sin que influyan otros parámetros. [5]

Según [5] con respecto a la dosis de corriente “La mayoría de autores establecen entre un mínimo de $0,05\text{mA/cm}^2$ y un máximo de 1 mA/cm^2 . Pero, nuestra práctica habitual, basada en la experiencia repetida, nos lleva a elegir la media de: $0,15\text{ mA/cm}^2$ ”.

1.6. Microcontroladores

Un microcontrolador es un circuito integrado, que incorpora en su interior los bloques básicos para formar un sistema embebido o una PC en menor escala, es decir el microcontrolador es un chip en cuyo interior encontramos una CPU, Memoria, Reloj, Puertos de Comunicación y Módulos Periféricos de E/S. [7]. Cada bloque interno del microcontrolador es encargado de cumplir con una función específica:

- ❖ **Procesador Central:** La CPU es la Unidad Central de Procesos encargada de codificar y ejecutar las instrucciones almacenadas en la memoria.
- ❖ **Memoria:** Almacena el programa destinado a gobernar una aplicación, realizando operaciones con las variables de programa definidas.
- ❖ **Temporizador:** Conocido también como reloj, provee una señal de sincronización para todas las tareas del sistema.
- ❖ **Puertos de comunicación:** Permiten tener comunicación con otros microcontroladores.
- ❖ **Módulos de E/S:** Son la conexión con el mundo exterior, soportan la conexión de sensores y actuadores del dispositivo a controlar.

Un microcontrolador es un computador complejo, contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea.

Existen diversos fabricantes de microcontroladores, los AVR son una familia de microcontroladores fabricada por la compañía noruega ATMEL.

1.6.1. Arduino uno

Arduino es una plataforma electrónica de diseño abierto, básicamente es un chip de la familia AVR de marca ATMEL, que adicionalmente posee toda la circuitería de soporte como se observa en la Figura 1.22.



Figura 1.22 Arduino uno

Fuente: [8]

Arduino Uno es una placa electrónica basada en el ATmega328 que cuenta con:

- ❖ 14 pines digitales de entrada / salida, de los cuales 6 se pueden utilizar como salidas PWM.
- ❖ 6 entradas analógicas.
- ❖ Un reloj cerámico 16 MHz
- ❖ Puerto de conexión USB.
- ❖ Conector de alimentación.
- ❖ Header ICSP.
- ❖ Botón de reinicio.

El Arduino Uno puede ser alimentado a través del propio cable USB o con un suministro de energía externo, proveniente de un pequeño transformador o una batería de 9v.

Las características técnicas de ARDUINO UNO se exponen de mejor manera en la Tabla 1.4.

Tabla 1.4
Características de la tarjeta Arduino uno

Microcontrolador	ATmega328
Voltaje de operación	5V
Voltaje de entrada	7-12V
Voltaje de salida	6-20V
Pines de entrada/salida digital	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica	6
Corriente continua pin IO	40 mA
Corriente continua pin 3.3V	50 mA
Memoria Flash	32 KB (0.5 KB ocupados por el bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Frecuencia de reloj	16 MHz

Fuente: [8]

La placa puede operar con un suministro de energía recomendada entre los 6 y 12 voltios. Si es suministrada con menos de 7V la salida del regulador de tensión a 5V puede dar menos que este voltaje, y si se sobrepasa los 12V probablemente la placa se dañe.

g. Shield EKG-EMG

Las Shields son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Shield ECG / EMG permite a Arduino capturar las señales de electromiografía, mediante el uso de Shield EKG EMG se abren nuevas posibilidades de experimentar con retroalimentación biológica. [9]

La Figura 1.23 muestra la vista frontal y posterior de una placa Shield EKG / EMG.

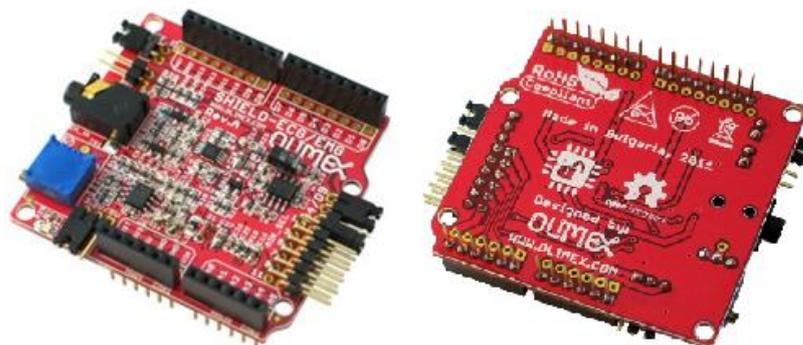


Figura 1.23 Shield ECG / EMG

Fuente: [9]

Las características principales de la tarjeta Shield ECG/EMG se exponen en la Tabla 1.5.

Tabla 1.5
Características Shield EKG-EMG

Fabricante	Olimex Ltd.
Licencia de Hardware	Creative Commons Atribución (BY)
Licencia de Software	Licencia GPL
Producto	Development Boards
Tipo	Bio-Feedback Shields
Herramienta de Evaluación	Olimex's ARDUINO Boards
Voltaje de alimentación operativo	3.3 V - 5 V
Marca	Olimex Ltd.
Descripción/Función	Módulo de extensión de arduino
Dimensiones	2 in x 2 in
Para utilizar con	Olimex's ARDUINO Boards

1.7. Plataforma de desarrollo Java

La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo.

En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de

las aplicaciones, y un conjunto de bibliotecas estándar que ofrecen una funcionalidad común. [10]

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil, por ello es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa.

Como lenguaje de programación Java permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios ya que posee diversas características:

- ❖ Lenguaje simple
- ❖ Orientado a objetos
- ❖ Distribuido
- ❖ Interpretado
- ❖ Sólido
- ❖ Seguro
- ❖ Arquitectura neutral
- ❖ Portable
- ❖ Alto desempeño
- ❖ Dinámico

1.8. Sistema Android

ANDROID es un sistema operativo para teléfonos móviles que permite crear aplicaciones de todo tipo, fue diseñado para teléfonos móviles, es similar a los sistemas operativos iOS (Apple), Symbian (Nokia) y Blackberry OS, sin embargo actualmente se instala no sólo en móviles, sino también en múltiples dispositivos, como tabletas. GPS, televisores, discos duros multimedia, mini ordenadores, etc. [11]

Android está basado en Linux, por lo tanto es libre, gratuito y multiplataforma. Android está estructurado de aplicaciones que se ejecutan

en un framework Java gracias a la máquina virtual Dalvik. Esto hace que desarrollar aplicaciones para Android sea relativamente sencillo en un lenguaje tan extendido como Java, ya que el sistema proporciona todas las interfaces necesarias para que las aplicaciones desarrolladas accedan a las funciones del teléfono.

1.9. Android Studio

Android Studio es un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones.

Existen diversas ventajas de Android Studio frente a entornos de iguales características como Eclipse.

- ❖ Un entorno de desarrollo claro y robusto.
- ❖ Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- ❖ Asistentes y plantillas para los elementos comunes de programación en Android.
- ❖ Un completo editor con muchas herramientas extra para agilizar el desarrollo de nuestras aplicaciones.

1.10. Comunicación inalámbrica

Son aquellas comunicaciones entre dispositivos móviles o no, que intercambian información utilizando el espectro electromagnético, engloba desde una comunicación Bluetooth entre un teléfono móvil y un ordenador portátil hasta una comunicación de dos terminales de telefonía móvil GSM que utilizan el espectro electromagnético como un canal para el intercambio de información. [12]

La Figura 1.24 muestra la clasificación de las comunicaciones inalámbricas según su alcance y la manera de controlar el acceso a la red.

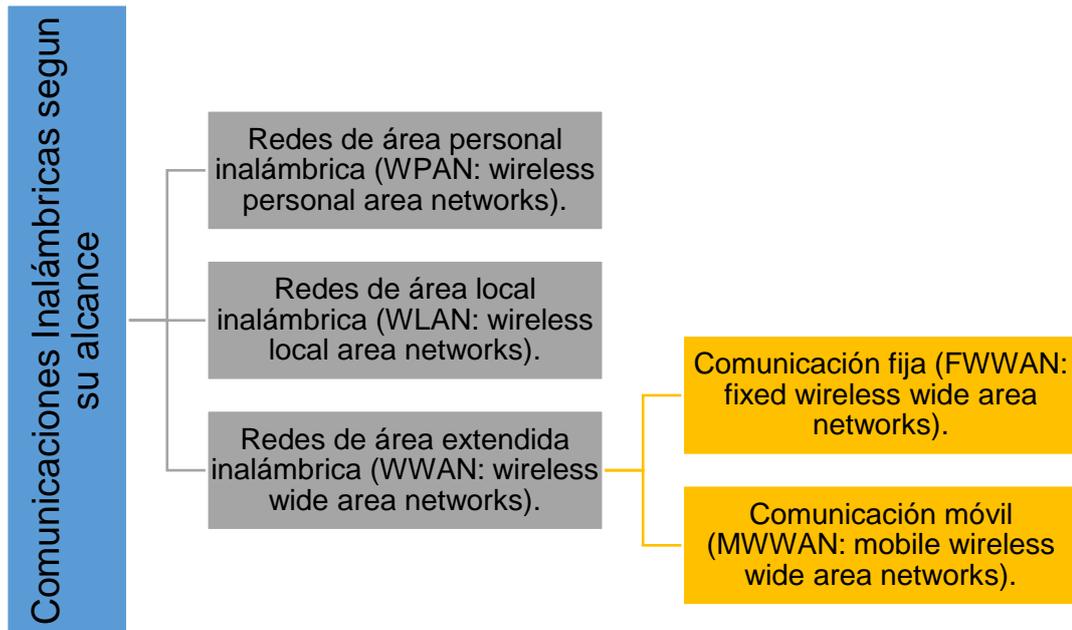


Figura 1.24 Clasificación de las comunicaciones inalámbricas

En las comunicaciones inalámbricas existen estándares y protocolos, los cuales se muestran en la Figura 1.25.

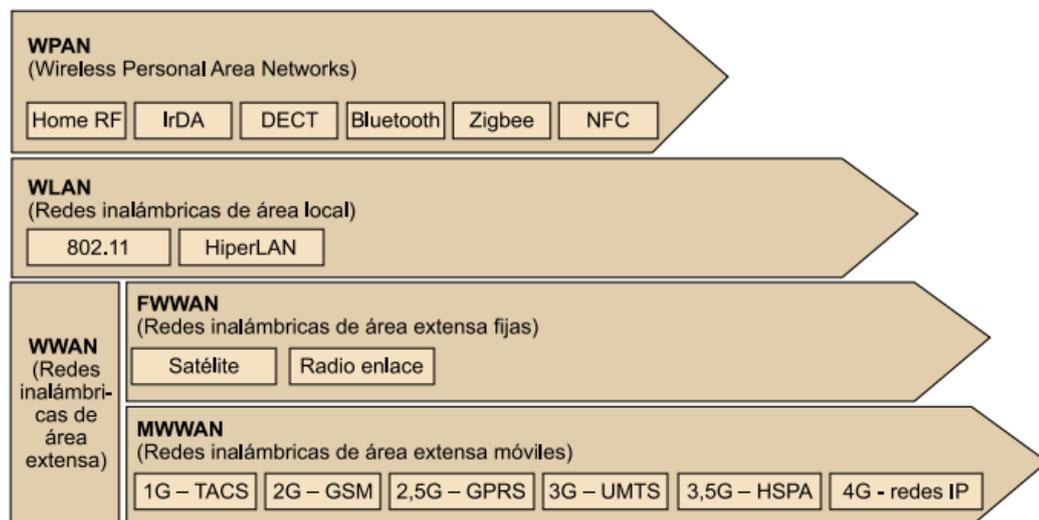


Figura 1.25 Estándares de comunicación inalámbrica

Fuente: [12]

1.10.1. Estándar Bluetooth (802.15.1)

Bluetooth es una especificación regulada por el grupo de trabajo IEEE 802.15.1, que permite la transmisión de voz y datos entre diferentes

dispositivos mediante un enlace de radiofrecuencia en la banda ISM de 2,4 GHz.

Bluetooth pertenece a las Redes personales inalámbricas (WPAN), define un estándar de comunicaciones inalámbricas de corto alcance mediante señales de radiofrecuencia que permite la transmisión de datos y voz. Bluetooth define un alcance corto (alrededor de 10 m) y, opcionalmente, un alcance medio (alrededor de 100 m).

Bluetooth permite conectar inalámbricamente diferentes dispositivos electrónicos, como asistentes digitales personales (PDA), teléfonos móviles, ordenadores portátiles, etc., lo que facilita, abarata y garantiza la interoperabilidad entre dispositivos de diferentes fabricantes a una velocidad aceptable.

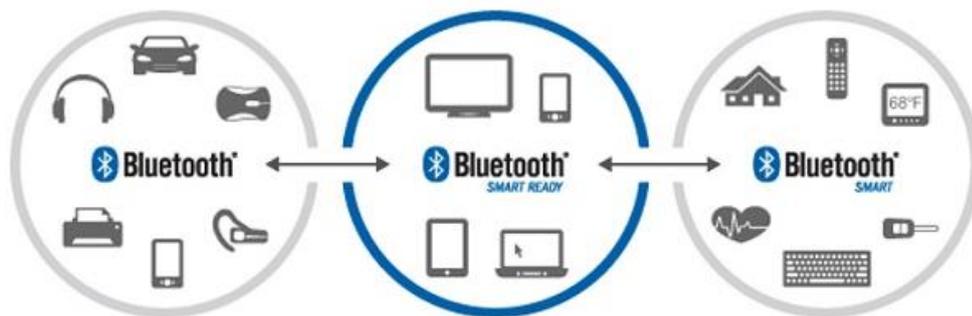


Figura 1.26 Aplicaciones Bluetooth

Fuente: [13]

La Figura 1.26 muestra que, mediante esta tecnología se puede acceder desde un ordenador, una cámara fotográfica digital o cualquier otro dispositivo electrónico a otro dispositivo Bluetooth situado en un teléfono móvil como punto de acceso a la red GSM/GPRS o UMTS. También permite la interconexión de ordenadores creando redes adhoc. Otra de las aplicaciones es la sustitución de los cables, RS-232, audio, etc., que conectan distintos dispositivos electrónicos entre sí. [13]

Debido a que la banda ISM es de libre uso, cumpliendo ciertas restricciones, la señal de radiofrecuencia del sistema Bluetooth deberá estar preparada para que las múltiples interferencias que se pudieran producir no

mermen su capacidad. Para ello Bluetooth utiliza el método de salto de frecuencia debido a que esta tecnología puede ser integrada en equipos de baja potencia y bajo costo. Este sistema divide la banda de frecuencia en varios canales de salto: los transceptores, durante la conexión, van cambiando de uno a otro canal de salto de manera pseudoaleatoria. La potencia de transmisión se especifica según tres tipos de clases de dispositivos: 1 mw para alcances inferiores a 5 metros, 2.5 mw para alcances de hasta 30 metros, y hasta 100 mw para cobertura de hasta 300 metros. [13]

La topología de las redes Bluetooth puede ser punto a punto o punto a multipunto, con todos los dispositivos iguales. Si un equipo se encuentra dentro del radio de cobertura de otro, éste puede establecer conexión con cualquiera de ellos. El control del enlace lo asume la unidad que ha iniciado la conexión según un protocolo de maestro-esclavo. De todas formas, los dispositivos pueden intercambiar el control de la conexión pasando el dispositivo que actúa como maestro a esclavo y viceversa. Un dispositivo que actúa como maestro puede estar conectado de forma simultánea hasta con siete dispositivos esclavos Bluetooth. Cuando dos o más dispositivos Bluetooth establecen una conexión a través de un único dispositivo que actúa como maestro forman una piconet. Cada piconet establece una secuencia de salto de frecuencia que depende de la dirección y de un reloj interno del dispositivo maestro. También existen diversos estados de bajo consumo en los que los dispositivos esclavos son aparcados, aunque se mantienen sincronizados, a la espera de la reiniciación del enlace tal y como se muestra en la Figura 1.27. Los dispositivos que no establecen ningún tipo de conexión están en un estado de espera.

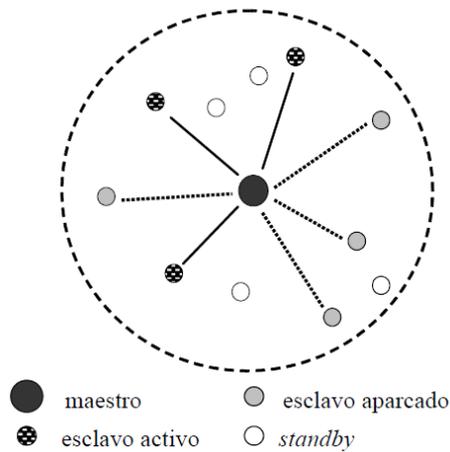


Figura 1.27 Estructura piconet

Fuente: [13]

Las unidades Bluetooth pueden establecer potencialmente comunicaciones entre ellas cuando las aplicaciones que los gestionan lo requieran, de modo que en una misma zona pueden existir distintos dispositivos que establezcan comunicaciones simultáneas entre sí. Esto provocará que se creen varias piconets, cada una con una secuencia de salto de frecuencia distinto en áreas de cobertura superpuestas, tal y como se muestra en la Figura 1.28.

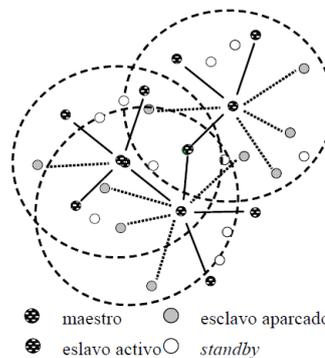


Figura 1.28 Estructura de scatternet

Fuente: [13]

A un grupo de piconets se le denomina scatternet. Debido a que individualmente cada piconet tiene un salto de frecuencia diferente, diferentes piconets pueden coincidir sin interferirse. De todas formas, si en un área reducida se establecen muchas piconets, más de 20 de forma simultánea a máxima velocidad, el rendimiento de cada una de ellas

disminuirá. Las distintas piconets pueden intercomunicarse puesto que un dispositivo puede ser maestro de una piconet y esclavo de otra, o esclavo de dos piconets distintas.

La seguridad de las transmisiones y los enlaces entre dispositivos se obtiene mediante:

- ❖ Saltos de frecuencia pseudoaleatorios, que dificultan que dispositivos ajenos a la red puedan interceptar o ver el tráfico de información.
- ❖ Autenticación, que permite a un usuario controlar la conectividad sólo para dispositivos especificados.
- ❖ Encriptación, mediante uso de claves secretas.

CAPÍTULO II

2. DISEÑO E IMPLEMENTACIÓN

En este capítulo se detalla, el proceso para la fabricación de cada una de las etapas que constituyen el Electromiograma y Electroestimulador, así como la correcta utilización de los sensores para captar las señales bioeléctricas del cuerpo humano.

2.1. Implementación del EMG

En el desarrollo del EMG se realizaron varias etapas funcionales que se muestran en el diagrama de bloques de la Figura 2.1.

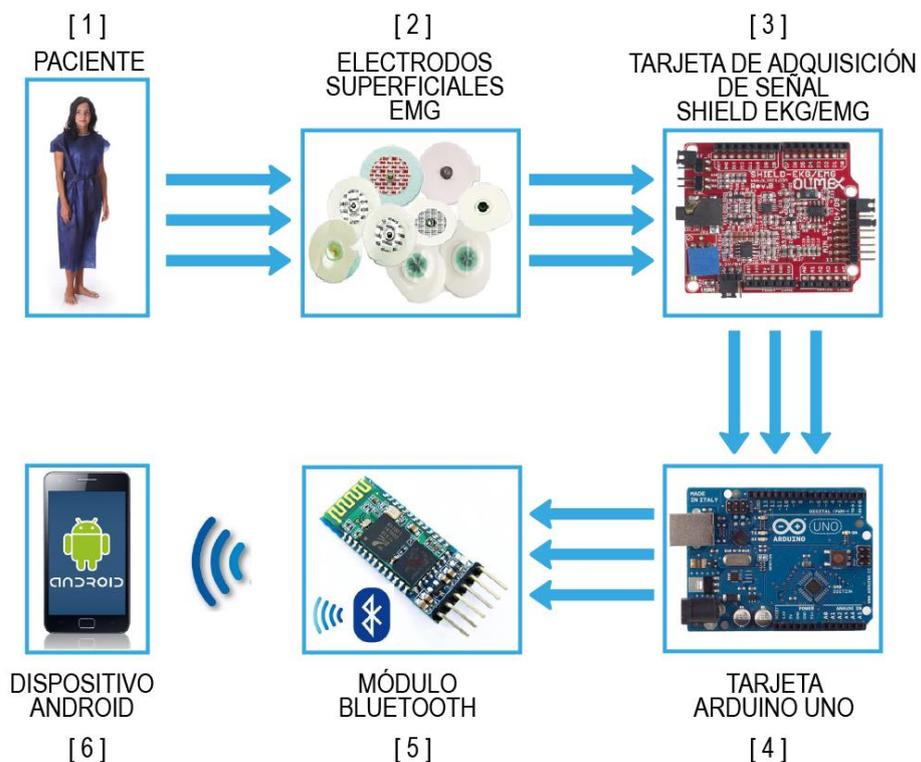


Figura 2.1 Diagrama de bloques de electromiógrafo inalámbrico

En el primer y segundo bloque se encuentra el paciente a quien se le colocan los sensores bioeléctricos en posiciones determinadas del cuerpo para registrar las señales bioeléctricas producto de la actividad de los músculos esqueléticos. En el tercer bloque está la tarjeta de adquisición de señales en donde se realiza la etapa de amplificación y prefiltrado de las señales electromiográficas y la eliminación de la interferencia inherente a la

medición de potenciales tan minúsculos. Después con la ayuda de la tarjeta Arduino uno se digitaliza estas señales para poder transmitir las inalámbricamente mediante el módulo Bluetooth a un dispositivo Android en donde se visualizará las señales correspondientes a los músculos.

2.1.1. Electrodo

Los electrodos, son los elementos primarios de un EMG que convierten las corrientes iónicas de los músculos del cuerpo humano en corrientes eléctricas, se utilizan electrodos de superficie, que están en contacto con la piel del paciente, por lo cual se debe considerar los siguientes aspectos:

- ❖ La impedancia normal de la piel varía desde 0.5 k Ω para piel sudorosa hasta 20 k Ω para piel seca.
- ❖ Problemas de la piel, especialmente resequeza, piel escamosa, o enfermedades en la piel producen un incremento en la impedancia en el rango de 500 k Ω .
- ❖ Los electrodos de superficie son considerados como una fuente de voltaje con muy alta impedancia.
- ❖ El adhesivo no permanece mucho tiempo, produciendo el movimiento o desprendimiento del electrodo.
- ❖ Se deberá evitar la colocación del electrodo en protuberancias óseas.
- ❖ El deslizamiento del electrodo causa un cambio abrupto del grosor de pasta o gel, lo que genera la producción de artefactos en la señal.

Para obtener una señal pura es recomendable rasurar al paciente en las áreas donde se colocarán los electrodos, después limpiar la superficie de la piel con alcohol y al final colocar el gel conductor este último ayuda a disminuir la impedancia producida por la piel.

2.1.2. Tarjeta de adquisición de la señal bioeléctrica

La mayor parte de la adquisición y medida de las señales biofísicas y bioeléctricas del cuerpo humano se realiza a través de circuitos electrónicos, de esta manera la adquisición de la señal del EMG se la realiza mediante la

shield EKG/EMG, dentro de la cual se encuentran amplificadores de instrumentación, filtros analógicos y un amplificador de ganancia.

La shield es alimentada directamente desde la tarjeta electrónica Arduino, por lo cual se realiza una conversión de voltaje digital VCCD a voltaje analógico VCCA para energizar todos los elementos, el cual se muestra en la Figura 2.2.

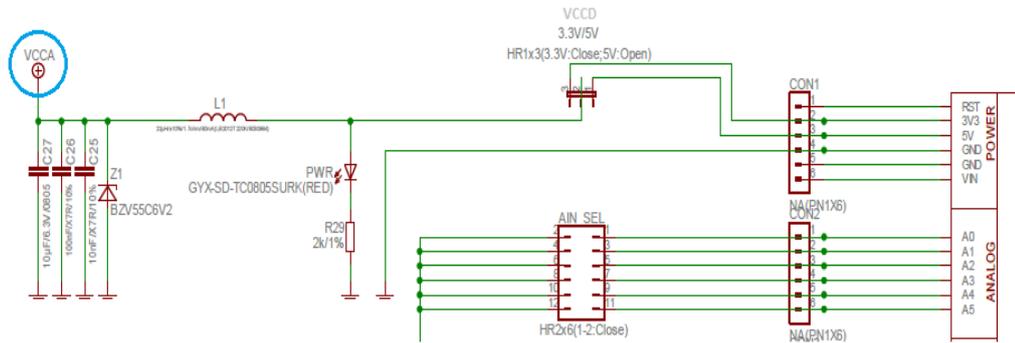


Figura 2.2 Conversion VCCD a VCCA

Fuente: [9]

La Figura 2.3 muestra que la entrada de la señal proveniente de los electrodos superficiales hacia la shield EKG/EMG, se dispone de tres sensores, por lo cual la entrada de voltaje es en modo diferencial entre dos sensores, siendo la referencia el tercer sensor.

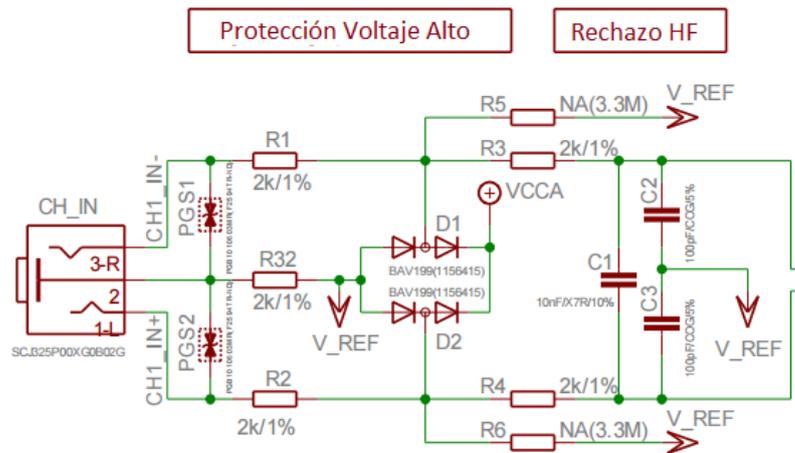


Figura 2.3 Etapa de recepción de señal en Shield EKG/EMG

Fuente: [9]

A pesar que la amplitud generada en el estudio de la electromiografía va de 0.1 a 5mV, en la etapa inicial de la Shield se coloca una protección de alto voltaje que impide que ingrese un voltaje mayor a 5V que pueda perjudicar a la tarjeta electrónica, lo cual puede ocurrir al generarse picos de alto voltaje por la estática proveniente del cuerpo humano.

A continuación de la protección de voltaje la tarjeta shield EKG/EMG cuenta con una etapa de rechazo de frecuencia para prevenir que la interferencia electromagnética EMI afecten en la adquisición de la señal.

La tarjeta de adquisición de datos cuenta con un amplificador de instrumentación Figura 2.4, cuya ganancia puede establecerse de forma muy precisa y ha sido optimizado para que opere de acuerdo a sus propias especificaciones aún en un entorno hostil. Este circuito amplifica la diferencia entre las dos señales de entrada y rechazan cualquier señal que sea común a ambas señales. Se utilizan principalmente para amplificar señales diferenciales muy pequeñas en aplicaciones médicas con una ganancia establecida.

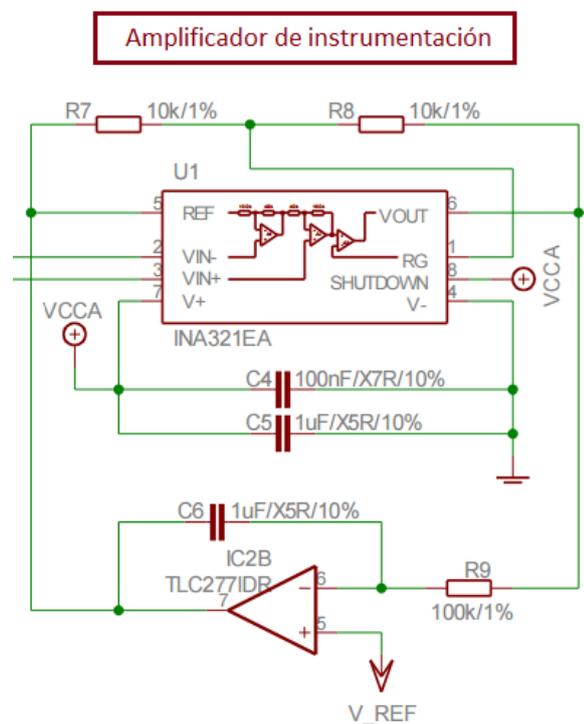


Figura 2.4 Amplificador de instrumentación de Shield EKG/EMG

Fuente: [9]

Para trabajar con la tarjeta Arduino la magnitud de la señal bioeléctrica debe ser necesariamente amplificada porque la señal adquirida de un paciente es demasiado pequeña, varía entre los 0.1mV y 5mV en función del tiempo-amplitud, a esta señal se le amplifica inicialmente dándole una ganancia de 10.

La ganancia del amplificador de instrumentación de la Shield EKG/EMG está dada por la siguiente ecuación:

$$G = 5 * \left(1 + \frac{R_8}{R_7}\right)$$

Para una ganancia

$$G = 10$$

Se asume

$$R_8 = 10 \text{ K}\Omega. \text{ y } R_7 = 10 \text{ K}\Omega$$

$$G = 5 * \left(1 + \frac{10 \text{ K}\Omega}{10 \text{ K}\Omega}\right)$$

$$G = 5 * (1 + 1)$$

$$G = 10$$

El ruido del ambiente puede interferir en la señal del EMG, y para conseguir una señal con el mínimo ruido se ha desarrollado dentro de la shield EKG/EMG un conjunto de filtros mostrados en la

Figura 2.5 que permiten obtener una señal electromiográfica aceptable.

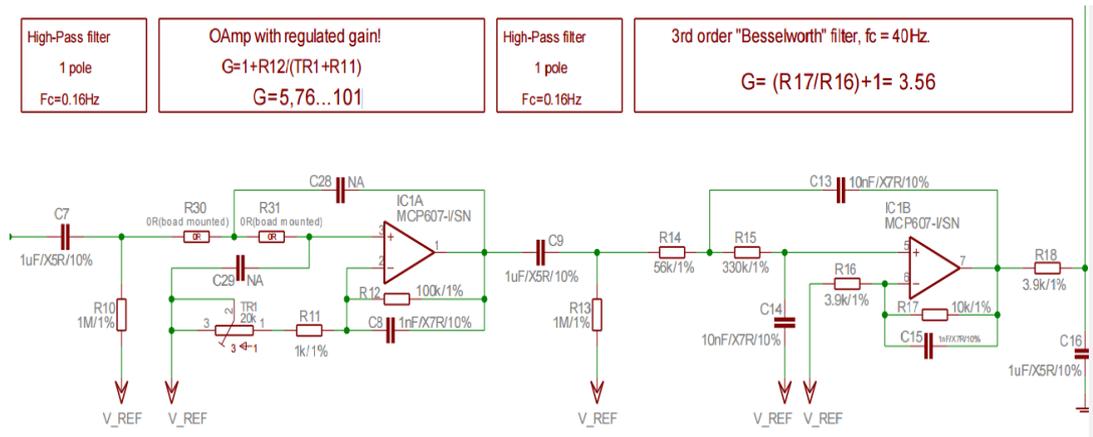


Figura 2.5 Filtros analógicos de la Shield EKG/EMG

Fuente: [9]

Para reducir las frecuencias indeseables o ruido generado por el ambiente que se introduce en la señal EMG, se emplea un filtro paso alto de 0.16Hz, luego del cual va un amplificador de ganancia variable, a continuación otro filtro de tercer orden Besselworth de 60 Hz con el que se logra reducir el ruido generado por la red eléctrica.

2.2. Implementación del electroestimulador

De manera similar al desarrollo del EMG en el electroestimulador se realizaron varias etapas funcionales que se muestran en el diagrama de bloques de la Figura 2.6.

En el primer bloque se encuentra el dispositivo Android, en el cual mediante la aplicación realizada se configura los parámetros del electroestimulador como: grupo muscular, tiempo de sesión, intensidad. Los datos son enviados a través del módulo bluetooth hacia la etapa de control en el tercer bloque, a continuación en la etapa de alto voltaje acorde a los datos ingresados en la aplicación se generan las corrientes que serán suministradas en el grupo muscular seleccionado del paciente a través de electrodos superficiales que están en contacto con la piel.

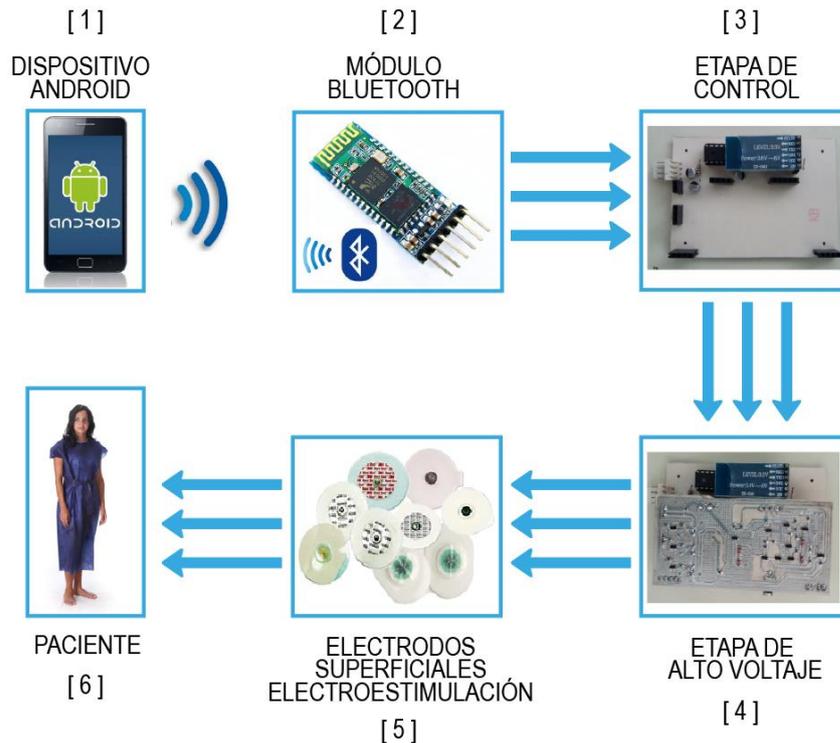


Figura 2.6 Diagrama de bloques de electroestimulador inalámbrico

2.2.1. Diseño del módulo de alto voltaje

a. Diseño del elevador de alto voltaje

Los electroestimuladores tienen como característica fundamental utilizar alto voltaje, por lo cual se ha diseñado un elevador de voltaje que se lo realiza mediante el convertidor dc-dc que se muestra en la Figura 2.7 para obtener a la salida una tensión continua mayor a la tensión de entrada.

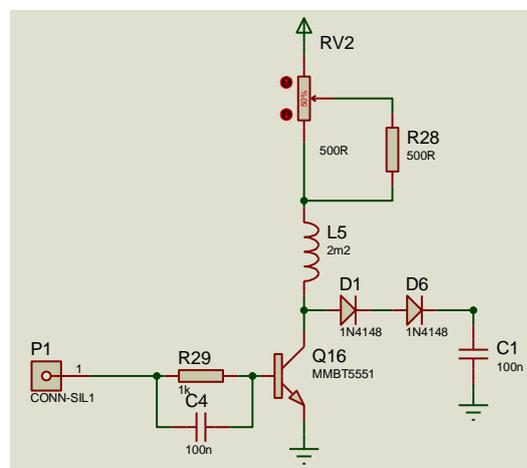


Figura 2.7 Elevador de voltaje

El elevador de voltaje cuenta con una etapa de oscilación, esta tiene la función de generar los pulsos necesarios para excitar al resto del conjunto. Las características de la señal de pulsos se obtienen de la Figura 2.8 considerando una tensión de salida de 70V, la señal PWM se genera mediante el microcontrolador ATMEGA328P que cuenta con timer internos que facilitan la ejecución en tiempo real, sin interferir con el desarrollo del resto de código.



Figura 2.8 Variación de la tensión de salida según el ciclo de trabajo
Fuente: [14]

$$frecuencia = 7.5 \text{ KHz}$$

$$\delta = \text{ciclo de trabajo oscilador}$$

$$\delta = 0,64\%$$

Para lograr la elevación de voltaje se utiliza una bobina, elemento que se carga de corriente almacenando energía, el valor de la bobina se calcula considerando la siguiente fórmula de diseño del conversor elevador Boost.

$$\delta = 0,64$$

$$R = \text{Resistencia de carga} = 220\Omega$$

$$f = \text{frecuencia oscilador}$$

$$f = 7.5 \text{ KHz}$$

$L_{min} = \text{Inductancia mínima bobina}$

$$L_{min} = \frac{\delta \cdot (1 - \delta)^2 \cdot R}{2 \cdot f}$$

$$L_{min} = \frac{0,64 \cdot (1 - 0,64)^2 \cdot 220}{2 \cdot 7500} = 1,21mH$$

Con el fin de tener un margen para asegurar corriente permanente se define:

$$L = 2.2mH$$

Mediante un interruptor electrónico Q16, se cortocircuita a tierra la bobina L5, que se cargará de corriente, almacenando energía. Mientras, C1 es el encargado de alimentar la carga que esté conectada a la salida.

Cuando el interruptor se vuelve a abrir, la corriente que ha almacenado L5 llega, a través del diodo D1 y D6, hasta C1, cargándolo con una tensión mayor a la entrada. El interruptor es un transistor de alta frecuencia que hará la función de abrir y cerrar miles de veces por segundo la conexión entre la bobina L1 y el terminal negativo del circuito. El diodo D1 garantiza que la corriente circule en un solo sentido, impidiendo que se cortocircuite a tierra C1 mientras el interruptor está cerrado.

La bobina es atravesada por una corriente que magnetiza progresivamente su núcleo hasta saturarlo. Cuando esto sucede, cesa interiormente el flujo de corriente y el campo magnético que rodea la bobina colapsa, generándose una fuerza contraelectromotriz (fem), en este momento se produce la transferencia de energía a través del diodo desde el inductor al capacitor de salida y también a la carga.

En serie con la bobina se coloca una resistencia variable R28 mediante la cual se puede generar diversos valores de voltaje, generando un voltaje máximo de:

$$V = 68V$$

b. Diseño de la etapa de ganancias

Para el diseño del electroestimulador se ha establecido tres niveles de intensidad, para lo cual se ha realizado una etapa de ganancias Figura 2.9, en donde varía la resistencia que se encuentra en serie con la bobina del elevador de tensión, ya que a mayor resistencia menor voltaje de salida.

Para la etapa de ganancia se utilizan transistores de alta frecuencia que funcionan en las zonas de corte y saturación, las señales de activación de los transistores según el nivel de voltaje que se requiera viene dada por el microcontrolador, el cual recibe los datos mediante el módulo bluetooth de la aplicación Android en donde se elige la configuración de los parámetros de nivel de intensidad.

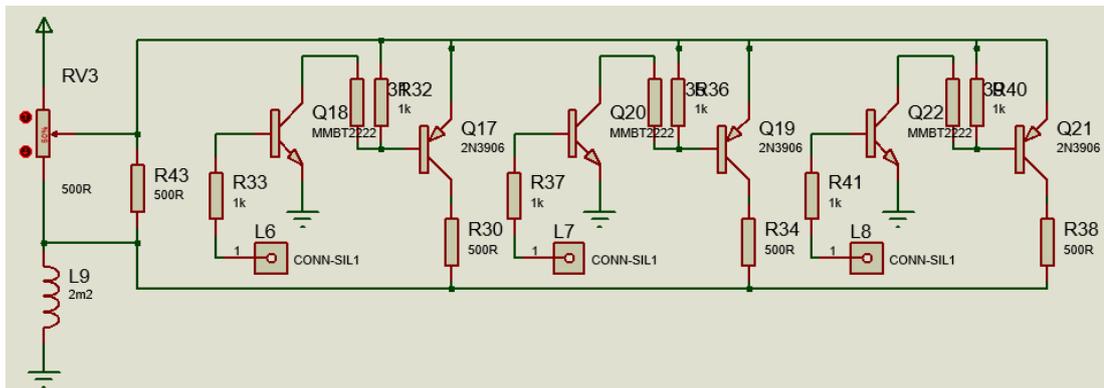


Figura 2.9 Etapa de ganancia

La Tabla 2.1 muestra el voltaje y la resistencia correspondiente según el nivel de intensidad:

Tabla 2.1

Voltaje y resistencia según el nivel de intensidad

Nivel	Voltaje	Resistencia en serie
Nivel 0	5 V	500 Ω
Nivel 1	14 V	470 Ω
Nivel 2	24 V	330 Ω
Nivel 3	30 V	250 Ω
Máximo	68 V	220 Ω

c. Diseño del modulador AM

Para lograr el efecto terapéutico en los electroestimuladores es necesario trabajar con frecuencias moduladas, la modulación de amplitud es un tipo de modulación no lineal que consiste en modificar la amplitud de una señal de alta frecuencia, denominada portadora, en función de una señal de baja frecuencia, denominada moduladora. La Figura 2.10 muestra el esquema del modulador en amplitud.

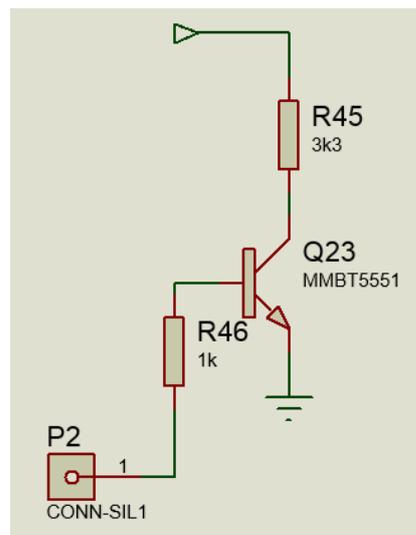


Figura 2.10 Modulador de amplitud

La onda moduladora será generada desde el microcontrolador según el grupo muscular al que se va a aplicar la electroestimulación, mientras que la onda portadora es resultado de la etapa del generador de pulsos diferenciales.

d. Diseño del generador de pulsos diferenciales

Para realizar el electroestimulador se utilizan dos electrodos biomédicos, los cuales producen una diferencia de voltaje entre ellos, ya que no cuentan con un tercer electrodo de referencia.

Para realizar el generador de pulsos diferenciales mostrado en la Figura 2.11. se utiliza dos señales PWM generadas desde el microcontrolador, en donde una es complemento de la otra, es decir mientras la señal PWMA se encuentra en estado alto, la señal PWM \bar{A} está en estado bajo y viceversa.

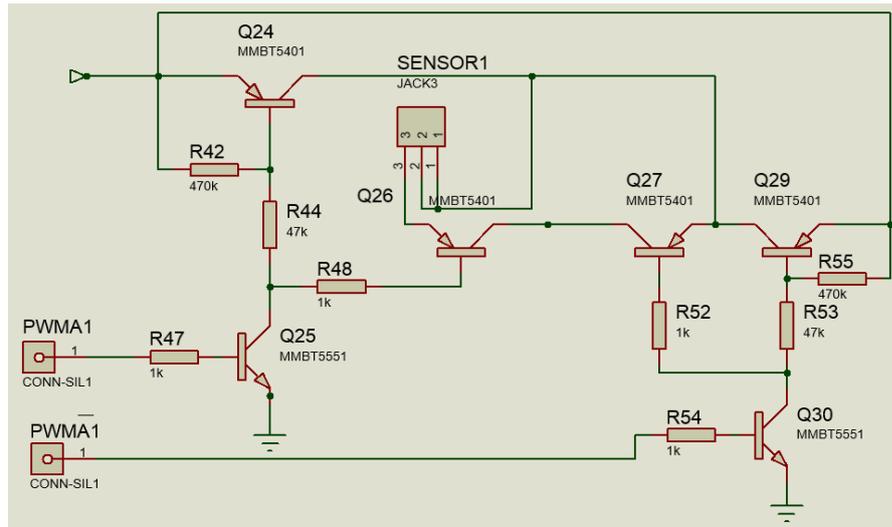


Figura 2.11 Generador de pulsos diferenciales

2.2.2. Diseño del módulo de control

Para el desarrollo del electroestimulador se ha realizado un módulo de control en donde se va a ubicar el microcontrolador ATMEGA328P, el cual es encargado de generar las señales PWM, dar las señales de control para la etapa de ganancia y verificar la conexión del sensor.

A continuación en la Figura 2.12 se observa la conexión del microcontrolador con el módulo de alto voltaje.

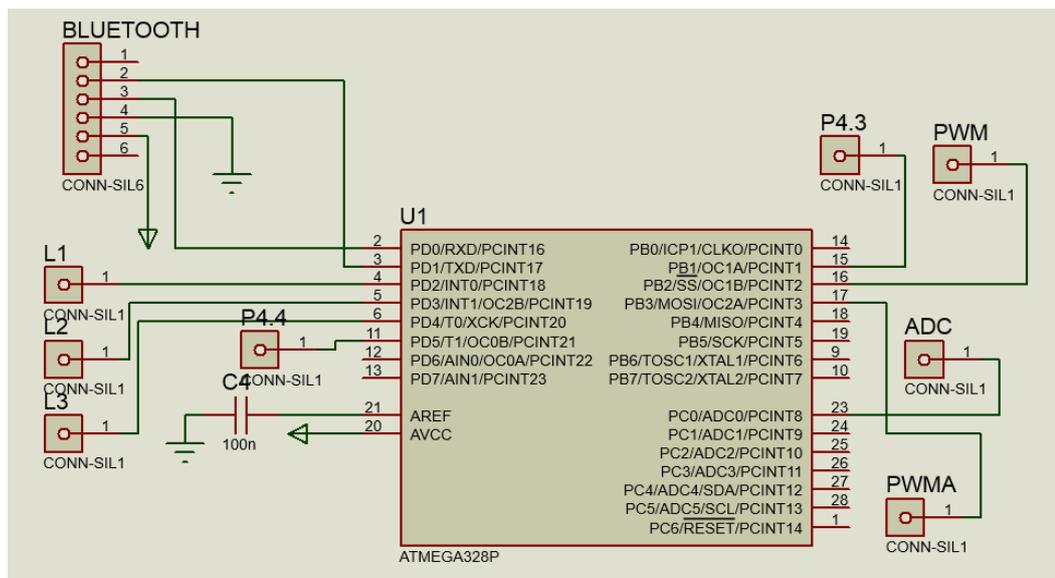


Figura 2.12 Conexion ATMEGA328P

En el contexto del desarrollo de instrumentos médicos es importante el factor seguridad, atendiendo principalmente a prevenir la posibilidad de shock eléctrico que se podría darse al conectar el equipo a la red eléctrica, lo cual podría ocurrir al momento de cargar las baterías, por ello nunca se debe utilizar el equipo si las baterías se encuentran en proceso de carga conectadas a la red eléctrica.

2.2.3. Configuración de las funciones del electroestimulador

Se han seleccionado los grupos musculares que más frecuentemente presentan complicaciones. Para realizar la electroestimulación se diseñan dos ondas con diferente ancho de pulso y frecuencia, a las cuales se han denominado onda alfa y onda beta.

Para cada sesión se ha diseñado una distribución de tiempo de la siguiente manera:

- ❖ La sesión inicia con 2 minutos de relajación con la onda alfa
- ❖ A partir de los primeros 2 minutos se aplica por 5 segundos la onda beta y por 10 segundos la onda alfa.
- ❖ Se aplica una combinación de la onda alfa y onda beta hasta concluir el tiempo de la sesión.

La combinación de la onda alfa por 10 segundos y onda beta por 5 segundos genera el efecto analgésico que se requiere para realizar la estimulación eléctrica.

Tanto la onda alfa como la onda beta generan un pulso que tiene un tiempo de estímulo y un tiempo de pausa acorde con la función seleccionada.

a. Función “Bíceps”

En esta función se realiza la electroestimulación del músculo bíceps, para lo cual se ha determinado la configuración mostrada en la Tabla 2.2.

Tabla 2.2
Ajustes Función Bíceps

Ajuste	Función Bíceps	
Modo	Onda modulada	
	Onda alfa	Onda beta
Ancho de pulso	1 ms	500 us
Frecuencia de pulso	6,25 Hz	70 Hz
Nivel de salida	Adaptable al nivel de intensidad más cómodo	
Tiempo de tratamiento	Configurable por experto	
Distribución de tiempo	120 segundos: Onda alfa 5 segundos: Onda beta 10 segundos: Onda alfa	

b. Función “Flexores”

En esta función se realiza la electroestimulación de los músculos flexores, para lo cual se ha determinado la configuración mostrada en la Tabla 2.3.

Tabla 2.3
Ajustes Función Flexores

Ajuste	Función Flexores	
Modo	Onda modulada	
	Onda alfa	Onda beta
Ancho de pulso	400 ms	500 us
Frecuencia de pulso	6 Hz	65 Hz
Nivel de salida	Adaptable al nivel de intensidad más cómodo	
Tiempo de tratamiento	Configurable por experto	
Distribución de tiempo	120 segundos: Onda alfa 5 segundos: Onda beta 10 segundos: Onda alfa	

c. Función “Muslos”

En esta función se realiza la electroestimulación del músculo muslos, para lo cual se ha determinado la configuración mostrada en la Tabla 2.4.

Tabla 2.4
Ajustes Función Muslos

Ajuste	Función Muslos	
Modo	Onda modulada	
	Onda alfa	Onda beta
Ancho de pulso	5 us	1 ms
Frecuencia de pulso	4 Hz	75 Hz
Nivel de salida	Adaptable al nivel de intensidad más cómodo	
Tiempo de tratamiento	Configurable por experto	
Distribución de tiempo	120 segundos: Onda alfa 5 segundos: Onda beta 10 segundos: Onda alfa	

d. Función “Peroné”

En esta función se realiza la electroestimulación de los músculos peroné, para lo cual se ha determinado la configuración mostrada en la Tabla 2.5.

Tabla 2.5
Ajustes Función Peroné

Ajuste	Función Peroné	
Modo	Onda modulada	
	Onda alfa	Onda beta
Ancho de pulso	2 ms	160 us
Frecuencia de pulso	6 Hz	75 Hz
Nivel de salida	Adaptable al nivel de intensidad más cómodo	
Tiempo de tratamiento	Configurable por experto	
Distribución de tiempo	120 segundos: Onda alfa 5 segundos: Onda beta 10 segundos: Onda alfa	

2.3. Módulo Bluetooth HC-05 para la Comunicación Serial

La transmisión de señales EMG hacia el dispositivo Android se realiza mediante el protocolo de comunicación inalámbrica Bluetooth, para ello se

selecciona el módulo HC-05 que utiliza el protocolo UART RS 232 serial, el cual es ideal para la implementación con Arduino.

El módulo HC-05 viene configurado de fábrica como esclavo, sin embargo puede funcionar en el modo maestro para conectarse con otros módulos bluetooth realizándose necesariamente la configuración a través de comandos AT, mientras que en el modo esclavo queda a la escucha de peticiones de conexión.

Agregando este módulo se puede realizar comunicaciones inalámbricamente, gracias a sus puertos TxD y RxD que nos permite comunicarnos a una distancia máxima de 10 mts desde un celular o una tablet, aprovechando todas las funcionalidades deseadas.

La conexión entre el módulo bluetooth y la tarjeta Arduino se realiza como se muestra en la Figura 2.13.

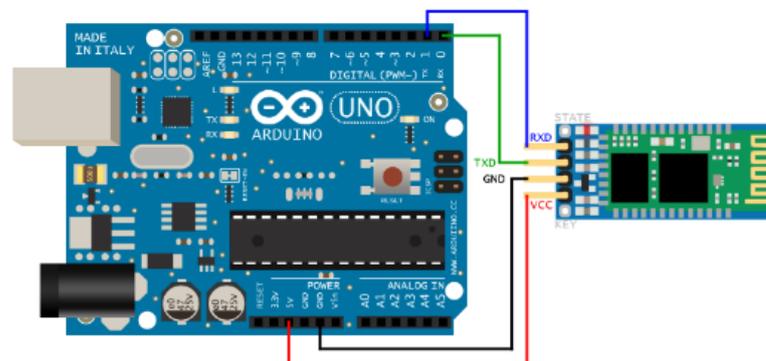


Figura 2.13 Conexión Bluetooth con Arduino

Fuente: [15]

El módulo HC 05 tiene un reducido tamaño lo que facilita la implementación del proyecto, además cuenta con las especificaciones técnicas que se muestra en la Tabla 2.6.

Tabla 2.6

Especificaciones técnicas del Módulo Bluetooth HC-05

Parámetro	Módulo Bluetooth HC-05
Protocolo Bluetooth	Bluetooth v2.0+EDR
Frecuencia	banda 2.4 GHz
Modulación	GFSK
Producto	Development Boards
Potencia de Emisión	4dBm, Class 2
Sensibilidad	-84dBm
Velocidad Asíncrona	2.1Mbps(Max) / 160 kbps
Velocidad Síncrona	1Mbps/1Mbps
Alimentación:	+3.3VDC 50mA
Velocidad de transmisión ajustable	200, 2400, 4800,9600, 19200, 38400, 57600, 115200 Baud Rate
Corriente de operación	<40mA
Corriente modo suspendido	<1mA
Temperatura de operación	- 40°C~86°C
Rango de alcance	10 metros

Para la transferencia de datos se ha utilizado dos módulos bluetooth configurados como esclavos, los cuales se conectan directamente con la aplicación mediante su dirección MAC, esto con el fin de facilitar la transmisión de datos evitando interferencia entre ellos. A continuación en la Figura 2.14 se muestran las direcciones MAC de los módulos bluetooth.

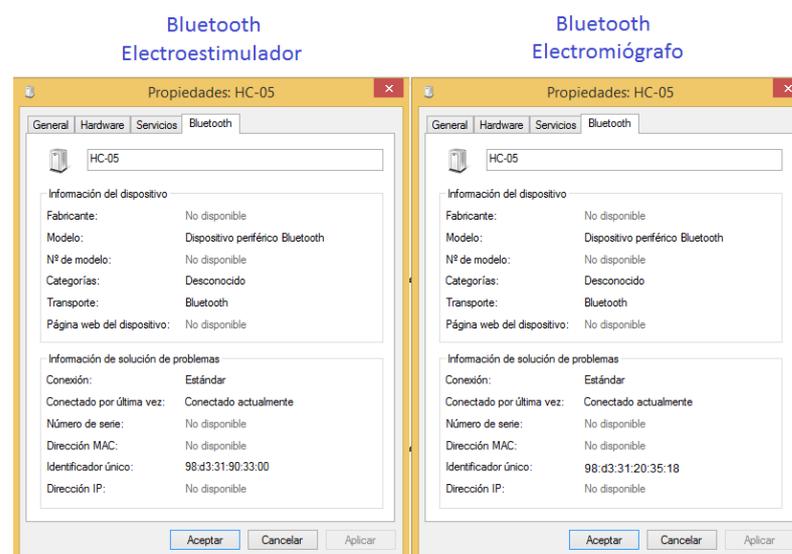


Figura 2.14 Direcciones MAC de módulos Bluetooth

2.4. Diseño del Software

Una vez finalizado la implementación del hardware, se procede a explicar el diseño del software tanto en Arduino, Android y Bascom.

Receptar la señal del EMG, así como la configuración de los parámetros del electroestimulador deben ser realizados en tiempo real, en un dispositivo con las características necesarias, para dicho propósito se utilizará una Tablet, que transmitirá la información proveniente de un dispositivo bluetooth, la información será procesada por la Tablet mediante una aplicación Android que permite visualizar la señal EMG en la pantalla y enviar los parámetros del electroestimulador.

Una vez finalizada las placas del proyecto, se procede a explicar el diseño del software tanto en Arduino, Android y Bascom.

2.4.1. Entorno de desarrollo para Arduino

Arduino cuenta con su propio entorno de desarrollo (IDE), el cual al ser un proyecto Open-source puede ser descargado de la página web de Arduino, en donde se dispone de versiones para Windows y para MAC.

El entorno de programación cuenta con un ambiente gráfico para fines didácticos que está compuesto por editor de texto, consola de texto, área de mensajes, barra de herramientas con botones de acceso rápido, y menús que se muestran en la Figura 2.15.

El IDE Arduino ofrece múltiples beneficios que facilita la programación como:

- ❖ Lenguaje simple, basado en C/C++.
- ❖ Programación directamente sobre el hardware.
- ❖ Proyecto open-source de libre distribución y uso.
- ❖ Comunidad de desarrollo a nivel mundial.

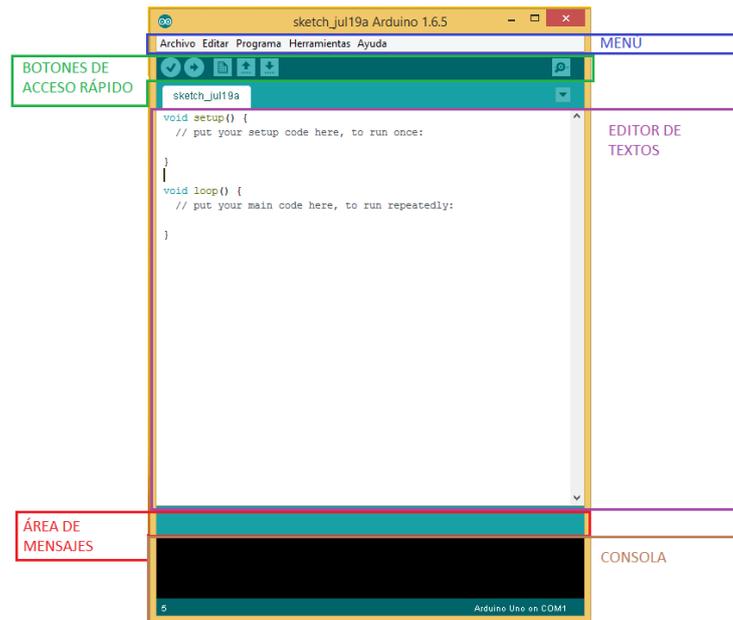


Figura 2.15 Entorno de programación arduino

La interfaz de Arduino además de ser un entorno de desarrollo gráfico posee en la barra de herramientas iconos con funciones específicas que facilitan la programación, a continuación en la Tabla 2.7 se muestran las funciones específicas de los botones de acceso rápido.

Tabla 2.7

Botones de acceso rápido de Arduino IDE

Icono	Función
	Verificar Compilar y Chequea el código identificando los errores
	Subir Cargar el programa compilado desde el PC hasta la tarjeta arduino.
	Nuevo Crear un nuevo sketch
	Abrir Presenta un menú para abrir los programas sketch previamente guardados y también muestra los sketch de ejemplos.
	Salvar Guarda el sketch actual
	Monitor Serie Realiza el monitoreo del puerto serial, visualiza el dato enviado desde la tarjeta arduino.

2.4.2. Desarrollo de la aplicación en Android

Para el desarrollo de la aplicación se ha seleccionado la plataforma Android Studio, que es el nuevo IDE (Integrated Development Environment - Entorno de Desarrollo Integrado) que Google ha puesto a disposición de los desarrolladores de Android de forma gratuita. Android Studio se basa en IntelliJ IDEA, un IDE que ofrece un buen entorno de desarrollo Android.

Al igual que en Arduino IDE la descarga de Android Studio se hace en la web directamente desde el sitio oficial de desarrolladores de Google, este programa proporciona las bibliotecas API y todas las herramientas de desarrollo para crear, probar y depurar las aplicaciones en Android, además se debe tener la actualización del Java Development Kit o (JDK), es un software que proporciona herramientas de desarrollo para la creación de programas en Java que son indispensables para ejecutar el emulador de Android y algunas herramientas de depuración.

Una vez que Android Studio este instalado correctamente se puede comenzar a desarrollar el proyecto.

a. Creación de un nuevo proyecto

Nuestro proyecto Android contiene todos los archivos que componen el código fuente de la aplicación Android. Las herramientas del SDK de Android facilitan el inicio de un nuevo proyecto proporcionando un conjunto de directorios y archivos de proyecto por defecto.

Para comenzar un nuevo proyecto de Android se sigue los siguientes pasos:

1. Abrir el programa Android Studio y en la ventana de Bienvenida dar click en Start a new Android Studio Project como se observa en la Figura 2.16, de esta manera abriremos el asistente que nos guiará por las distintas opciones de creación y configuración de un nuevo proyecto Android.

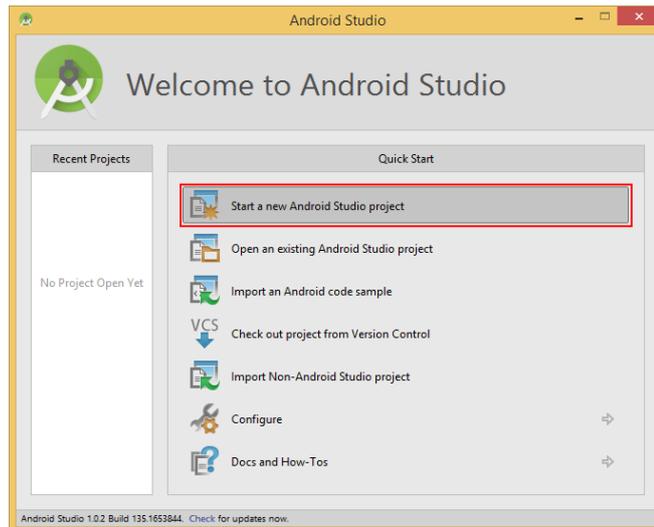


Figura 2.16 Ventana de bienvenida de Android Studio

2. En la pantalla mostrada en la Figura 2.17 se completan los campos de configuración del proyecto los cuales son:
 - ❖ Application Name: Nombre de nuestra aplicación.
 - ❖ Company Domain: Nombre de espacio de nuestra aplicación, se utiliza un nombre de dominio web.
 - ❖ Package name: Este campo se llena solo, es una combinación del nombre de la aplicación y el dominio anterior.
 - ❖ Project Location: Este es el directorio por defecto donde se creará la aplicación.

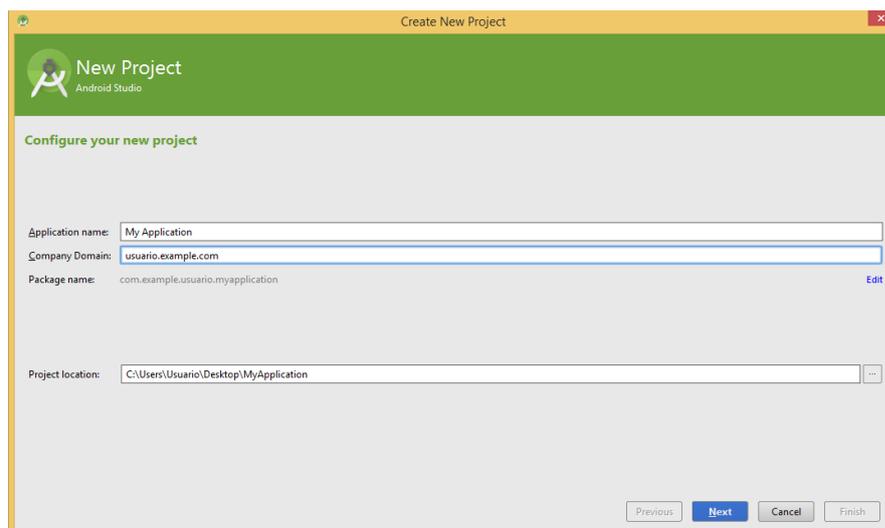


Figura 2.17 Pantalla de configuración de un nuevo proyecto

3. En la siguiente pantalla del asistente mostrada en la Figura 2.18 configuraremos las plataformas y APIs que va a utilizar nuestra aplicación. Nos centraremos en aplicaciones para teléfonos y tablets, en cuyo caso tan sólo tendremos que seleccionar la API mínima que soportará la aplicación, para este proyecto Android 4.0 como versión mínima (API 14).

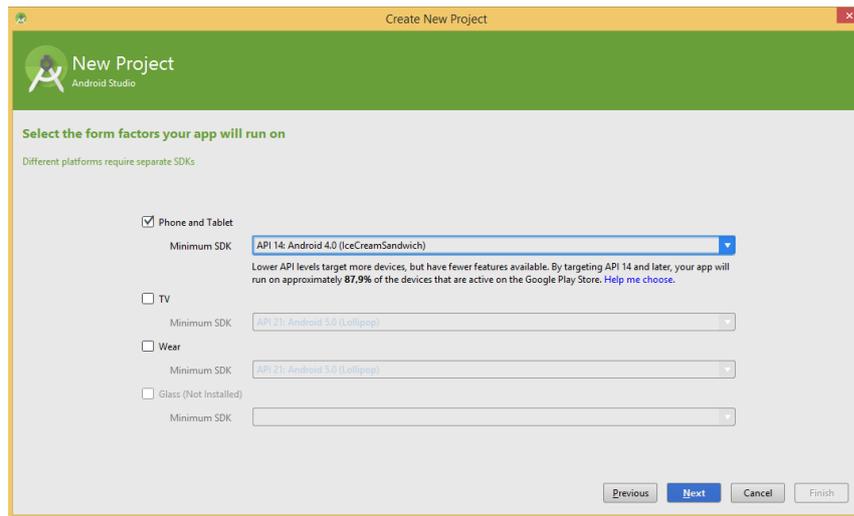


Figura 2.18 Configuración de plataformas de Android Studio

4. En la siguiente pantalla se muestra la plantilla en la que se va a trabajar. Por defecto aparecen varias opciones como se ilustra en la Figura 2.19, para nuestro proyecto se seleccionó BlankActivity, que es el tipo más sencillo.

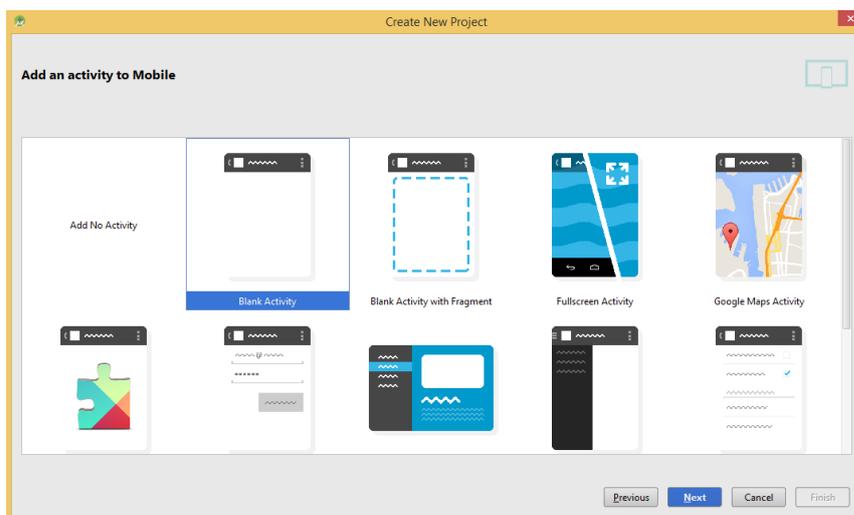


Figura 2.19 Creación de nueva actividad proyecto Android

- En la siguiente pantalla del asistente se configura los datos asociados a la actividad principal, indicando el nombre de su clase java asociada (Activity Name) y el nombre de su layout xml, su título, y el nombre del recurso XML correspondiente a su menú principal.

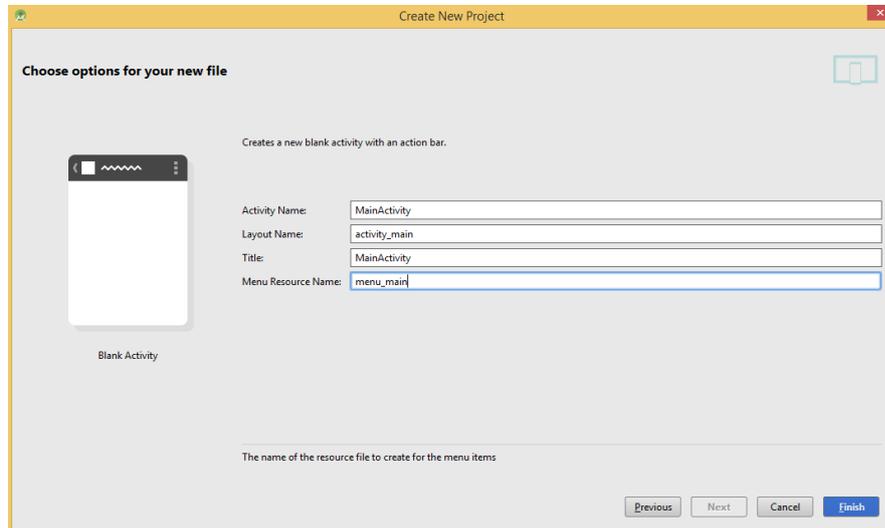


Figura 2.20 Configuración de actividad en blanco proyecto Android

- Una vez configurados todos los parámetros se pulsa el botón Finish y Android Studio creará toda la estructura del proyecto y los elementos indispensables que debe contener. Al finalizar aparecerá la pantalla principal de Android Studio con el nuevo proyecto creado como se muestra en la Figura 2.21.

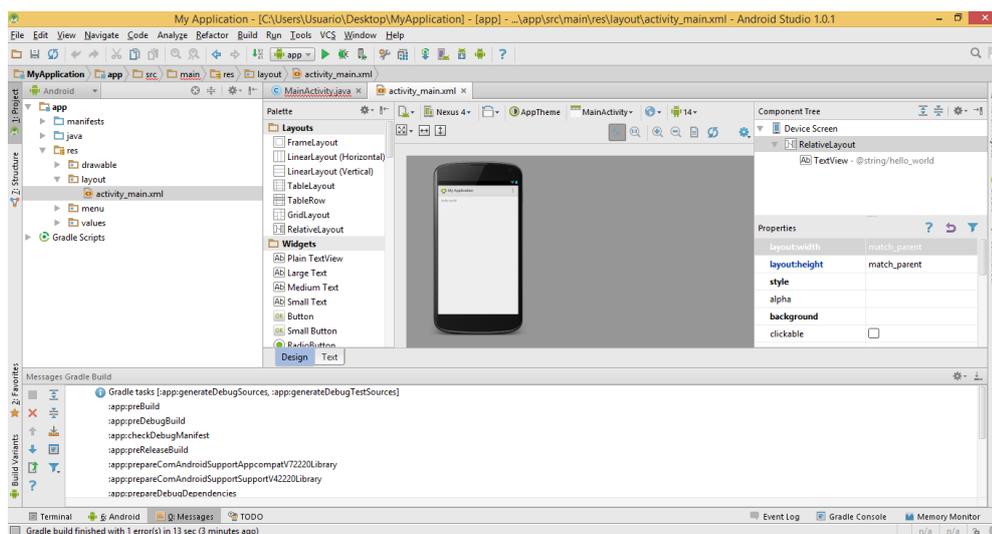


Figura 2.21 Pantalla Principal de Android Studio

b. Diagrama de flujo de la aplicación en Android

A continuación se describen los diagramas de flujo de la programación del dispositivo móvil bajo la plataforma Android, tanto de la actividad principal como del resto de actividades.

La Figura 2.22 muestra el diagrama de flujo general de la aplicación en el dispositivo móvil, en la pantalla principal se despliega un menú en donde se encuentra dos botones llamados “Electromiógrafo” y “Electroestimulador”, al pulsar el botón Electromiógrafo se despliega una ventana en donde se visualiza las señales del electromiograma, al pulsar el botón Electroestimulador se despliega una ventana en donde se configura los parámetros de la electroestimulación.

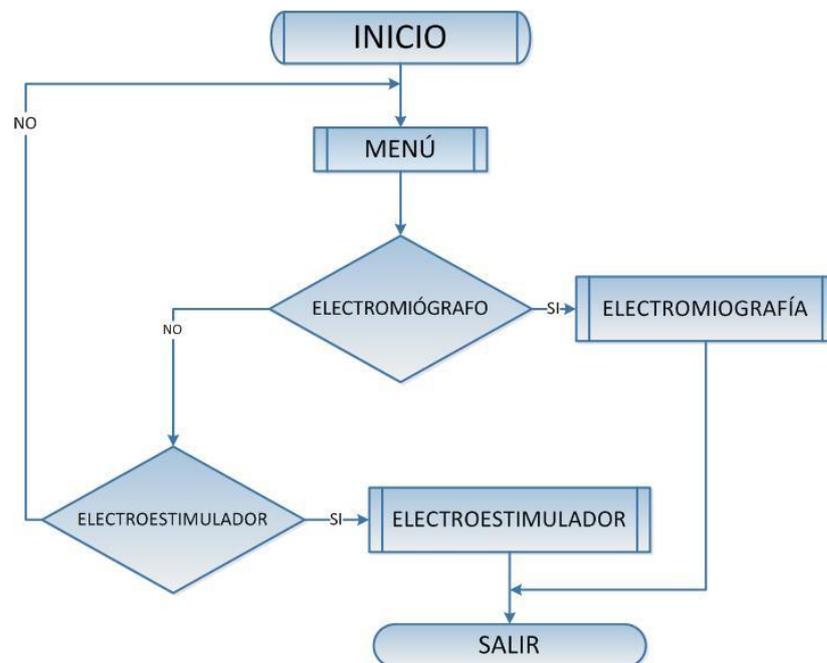


Figura 2.22 Diagrama de flujo de Actividad Principal

Cada actividad en el software Android Studio permite diseñar un Layout, que es una distribución de los elementos dentro de un diseño, la Figura 2.23. muestra el Layout de la Actividad Principal, en la cual se distribuyen los botones del menú explicados previamente.



Figura 2.23 Layout de Actividad Principal

En la Figura 2.24, se muestra el diagrama de flujo de la actividad Electromiógrafo para la obtención y visualización de los datos, en primer lugar se configura la comunicación Bluetooth, para leer los datos de la variable de la señal del EMG ya asignada en la tarjeta Arduino con su valor correspondiente, para luego ser visualizada en el dispositivo móvil.

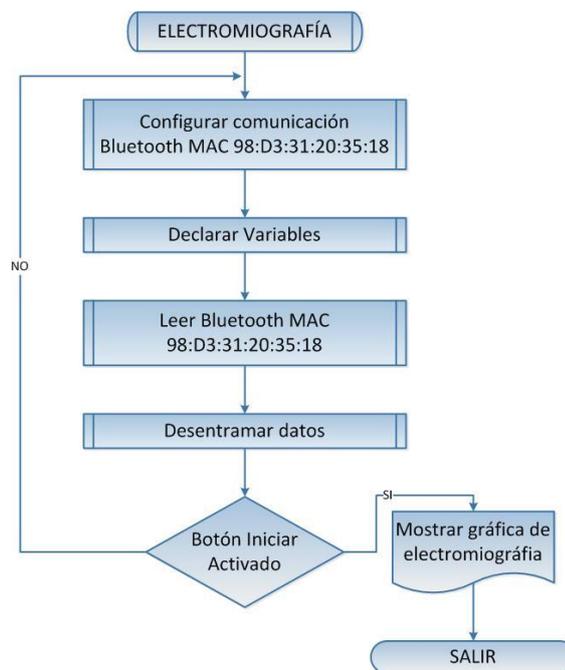


Figura 2.24 Diagrama de flujo de Actividad Electromiógrafo

La Figura 2.25 muestra el layout de la actividad Electromiógrafo, en la cual se observa el espacio adecuado para la gráfica del EMG y adicionalmente existe un botón que brinda la posibilidad de detener y reanudar la gráfica del EMG.



Figura 2.25 Layout de Actividad Electromiógrafo

La Figura 2.26 muestra el layout de la actividad electroestimulación.



Figura 2.26 Layout de Actividad Electroestimulador

En el layout de la actividad electroestimulador se configura los siguientes parámetros:

- ❖ Grupo Muscular: Corresponde a la sección muscular que va a recibir el tratamiento.
- ❖ Dosis: Se refiere al nivel de intensidad de corriente que va a recibir el paciente.
- ❖ Sesión: Configura el tiempo en minutos que va a tardar la sesión de electroestimulación.

La Figura 2.27 muestra el diagrama de flujo de la actividad Electroestimulador para la configuración de los parámetros de electroestimulación, en primer lugar se configura la comunicación Bluetooth, para enviar los datos de los parámetros hacia el módulo de control, para luego en el módulo de alto voltaje generar las corrientes de electroestimulación.

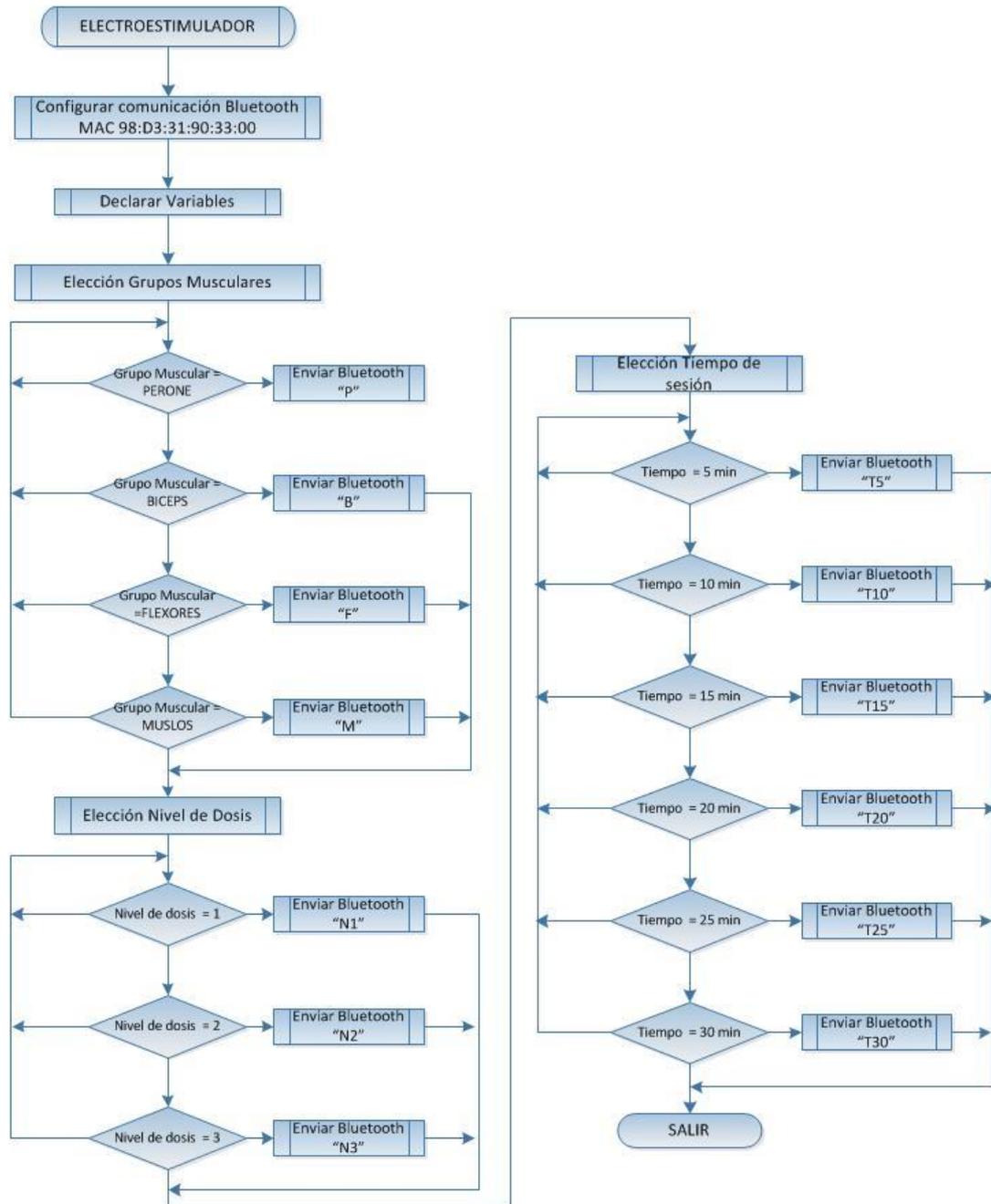


Figura 2.27 Diagrama de flujo de Actividad Electroestimulador

CAPÍTULO III

3. PRUEBAS Y RESULTADOS OBTENIDOS

En este capítulo se describe el análisis del funcionamiento tanto del dispositivo electromiógrafo como del electroestimulador, aplicando pruebas a diferentes voluntarios. En tanto que los resultados de las pruebas ayudaron a la determinación de importantes conclusiones.

3.1. Pruebas del EMG

Una unidad motora al activarse envía un impulso o potencial de acción que se desplaza de la motoneurona al músculo. Cuando el potencial de acción alcanza el electrodo negativo, se produce una diferencia de potencial que se manifiesta en el registro como una onda positiva. Cuando el impulso llega al electrodo positivo, la diferencia de potencial se registra como una onda negativa. Un músculo se contrae mediante la activación simultánea de varias unidades motoras, por lo que en el registro eléctrico se mezclan las ondas producidas por cada unidad motora.

Durante la realización del electromiograma los diferentes electrodos captan las señales eléctricas, que se almacenan en una memoria. Al mismo tiempo que se realiza la prueba se pueden observar estas mediciones en la pantalla del dispositivo android.

De una electromiografía se puede analizar la amplitud, que se usa como una medida del estímulo neural del músculo, donde la amplitud de pico a pico de una onda es proporcional al número de unidades motoras activadas por la estimulación eléctrica del nervio periférico. La frecuencia proporciona información fisiológica sobre la velocidad de conducción de las fibras musculares y la tasa de disparo de las unidades motoras, e información no fisiológica sobre ciertos tipos de contaminación de ruido de las interferencias eléctricas dentro de la señal electromiográfica.

Para la realización de las pruebas de EMG se ha seleccionado el grupo muscular correspondiente al músculo bíceps. A continuación se mostrarán algunas gráficas de las diferentes pruebas realizadas en el sistema del EMG.

3.1.1. Adquisición de señales de EMG

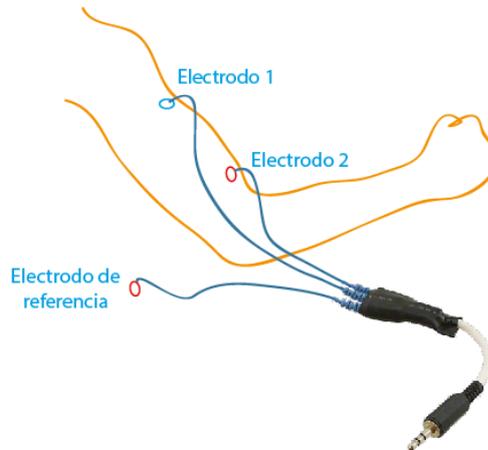


Figura 3.1 Posición de electrodos

Para adquirir las señales del EMG se debe considerar los siguientes pasos para la preparación del paciente:

- ❖ Quitar reloj y decoración de las manos del paciente.
- ❖ Limpiar el área de la piel a ser examinada con alcohol antiséptico para la colocación de los Electrodos y colocar gel conductor para reducir la resistencia de la piel.
- ❖ Dos electrodos se colocan en la parte superior del brazo derecho como se observa en la Figura 3.1. cada uno aparte de 6-8 cm. La línea que conecta los dos electrodos debe estar en paralelo con el eje del brazo.
- ❖ Colocar el electrodo de referencia en el brazo izquierdo.

3.1.2. Prueba de contracción isométrica

- ❖ El paciente debe estar en una posición que se sienta cómodo, para relajarse.
- ❖ Observar si la señal es estable, si existe un ruido excesivo, significa que existe una alta impedancia entre el electrodo y la piel. Entonces, es necesario reubicar el electrodo de referencia.
- ❖ Ejercer una fuerza consciente en el músculo y observar si la amplitud de la forma de onda aumenta.

- ❖ Luego, el paciente debe relajar el músculo.
- ❖ Sujete la pesa de 1 kg fuertemente.
- ❖ Después de 2 segundos de grabación, levante la mancuerna en una condición de 2 segundos en el que el codo debe mantenerse a 90 grados y la parte superior del brazo en 45 grados.

3.1.3. Prueba de contracción isotónica

- ❖ El paciente debe estar en una posición que se sienta cómodo, para relajarse.
- ❖ Sujete la pesa de 1 kg fuertemente.
- ❖ Después de 2 segundos de grabación, levante la mancuerna en un segundo. Repita la acción tres veces antes de tomar un descanso de 2 segundos.

3.1.4. Prueba de músculo en fatiga

- ❖ Al paciente se le pide estar en una posición de pie, para relajar naturalmente su mano derecha (los dedos apuntan a la tierra), y de putt que la palma de mirar hacia adelante.
- ❖ Sujete la pesa de 1 kg.
- ❖ Después de 2 segundos de grabación, levante la mancuerna y no cambie esta postura hasta que haya una alteración evidente en las señales de EMG.

3.2. Análisis de resultados de las pruebas del EMG

3.2.1. Resultados de Adquisición de señales de EMG

Para adquirir las señales eléctricas del músculo, es importante preparar adecuadamente al paciente para que se disminuya la resistencia de su piel y los electrodos puedan obtener una señal sin ruido.

De la correcta ubicación de los electrodos depende el éxito del experimento.



Figura 3.2 Preparación del paciente



Figura 3.3 Ubicación de los electrodos en el paciente

A continuación se presentan las gráficas obtenidas de la señales del EMG.

a. EMG con Ruido

La Figura 3.4 corresponde a un EMG con ruido, ya que los electrodos todavía no son colocados al paciente como se observa en la Figura 3.5.



Figura 3.4 Señal con ruido



Figura 3.5 Paciente EMG con ruido

Como se observa en la gráfica no existe una línea base definida, y se presentan interferencias que no representan ningún impulso motor. Si se obtiene esta señal a pesar de haber conectado los electrodos al paciente, se debe reubicar el electrodo de referencia.

b. EMG sin peso

La Figura 3.6 presenta la señal del paciente sin sostener ningún peso, como se observa la señal es estable y de baja amplitud.



Figura 3.6 EMG sin peso



Figura 3.7 Paciente EMG sin Peso

Mediante el electromiograma se puede confirmar o descartar la existencia de enfermedades musculares o neurológicas de carácter degenerativo, como distrofias, esclerosis y otras.

En un electromiograma normal, cuando el músculo está en reposo se visualiza solamente la línea base. Es decir, cuando el músculo se encuentra en reposo no se registra ninguna actividad en el EMG.

Si en estas condiciones cuando no hay que observar registro de ninguna actividad, se lleva a cabo una contracción débil por parte del músculo, se visualizará un reducido número de potenciales de unidad motora, siendo cada una de ellas representaciones de un electromiograma patológico.

c. EMG con peso de 1Kg.

En la Figura 3.8 se presenta la señal del paciente sosteniendo una mancuerna de 1Kg, como se observa la señal es estable y con una amplitud superior a la señal sin peso.



Figura 3.8 EMG con peso de 1Kg.

3.2.2. Resultados de prueba contracción isométrica

Para este experimento se le pide al paciente que sostenga la mancuerna de 1kg para que el músculo desarrolle tensión pero no cambia su longitud externa (constante) y después vuelva al estado de relajación.

El resultado de este experimento se observa en la Figura 3.9, en la cual existe una variación en la señal debido a la tensión desarrollada por el músculo al sostener la mancuerna por un tiempo prolongado, ya que si

aumentamos la fuerza de contracción se incorporan nuevos potenciales que densifican el trazado, siendo ya escasos los trechos netos con línea de base. Si la contracción llega a ser máxima desaparecerá por completo la línea de base y se habrá conseguido un patrón de interferencia.



Figura 3.9 Contracción isométrica

En la contracción isométrica se modifica la tensión del músculo, manteniendo constante la longitud, en el electromiograma de una contracción isométrica, según la gradación de la fuerza muscular, se observa que:

- ❖ En las contracciones débiles las unidades motoras tienen un ritmo de actividad (frecuencia de disparo) bajo.
- ❖ En las contracciones medianas las unidades motoras aumentan el ritmo de actividad y aparecen nuevos potenciales, resultantes del reclutamiento de nuevas unidades motoras.
- ❖ En las contracciones máximas las unidades motoras tienen un ritmo muy alto.

3.2.3. Resultados de pruebas de contracción isotónica

Las contracciones isotónicas son las más comunes en la mayoría de los deportes, actividades físicas y actividades correspondientes a la vida diaria, ya que en la mayoría de las tensiones musculares ejercidas suelen ir

acompañadas por acortamiento y alargamiento de las fibras musculares de un músculo determinado. Para la prueba se realiza un ejercicio, en donde el paciente debe sujetar la pesa de 1 kg fuertemente; subir la mancuerna a la parte superior del brazo y la bajar a 45 grados con intervalos de tiempo constantes. En la contracción isotónica se va analizar dos tipos de movimientos.

La Figura 3.10 muestra una contracción isotónica concéntrica, en la cual la amplitud de la señal aumenta por la tensión que se ejerce en la contracción del músculo.



Figura 3.10 Contracción isotónica concéntrica

La Figura 3.11 muestra una contracción isotónica excéntrica, en la cual la amplitud de la señal disminuye porque la tensión del músculo también disminuye al alargarse.



Figura 3.11 Contracción isotónica excéntrica

Finalmente en la Figura 3.12 se puede observar la señal de la contracción isotónica general, de la cual se puede concluir cada que se genera un alargamiento y/o contracción del músculo la señal se modifica acorde con la tensión que desarrolla el músculo.



Figura 3.12 Contracción isotónica concéntrica y excéntrica

3.2.4. Resultados de pruebas de músculo en fatiga

Se realiza un ejercicio en el cual el paciente sujeta la pesa de 2 kg fuertemente por un tiempo prolongado, los resultados de este experimento se observa en la Figura 3.13, en la cual la amplitud de la señal es alta ya que el músculo ha sido sometido a una tensión por un tiempo prolongado.



Figura 3.13 Músculo en fatiga

3.3. Comparación de señales de EMG con el Módulo KL-730002

Se ha utilizado el Módulo KL-730002 el cual posee la función de EMG y está disponible en el laboratorio de instrumentación biomédica, con el fin de realizar una comparación de esas señales con las señales registradas como resultados de las pruebas con el EMG.

3.3.1. Experimento de contracción isométrica

En la Figura 3.14 se muestra la gráfica de la contracción isométrica tomada del módulo KL-730002 y en la Figura 3.15 se muestra la señal obtenida en el dispositivo Android, se observa que las señales no difieren, ya que en ambas la amplitud aumenta conforme aumentan el esfuerzo muscular.

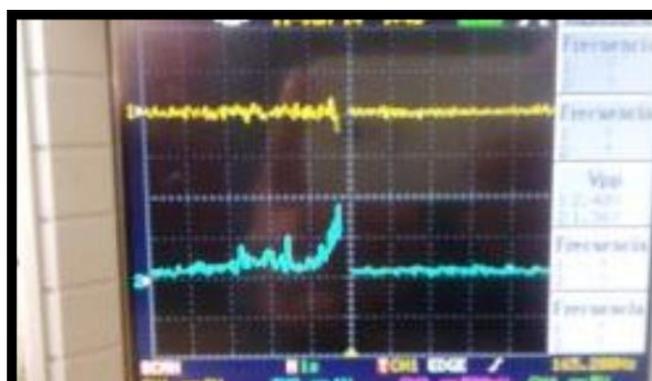


Figura 3.14 Contracción isométrica con Módulo KL-730002

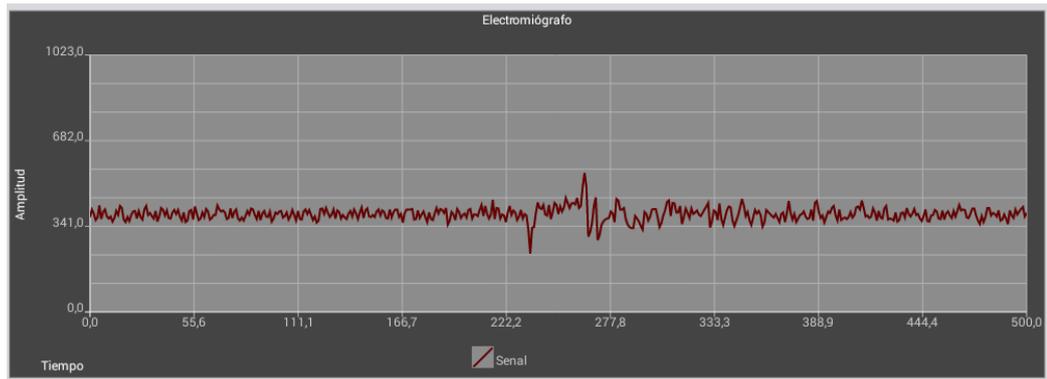


Figura 3.15 Contracción isométrica dispositivo EMG

3.3.2. Experimento de contracción isotónica

En la Figura 3.16 se muestra la gráfica de la contracción isotónica tomada del módulo KL-73002 y en la Figura 3.17 se muestra la señal obtenida en el dispositivo Android, las señales se comportan de manera similar, ya que en ambas la amplitud disminuye cuando el músculo entra en un estado de relajación.

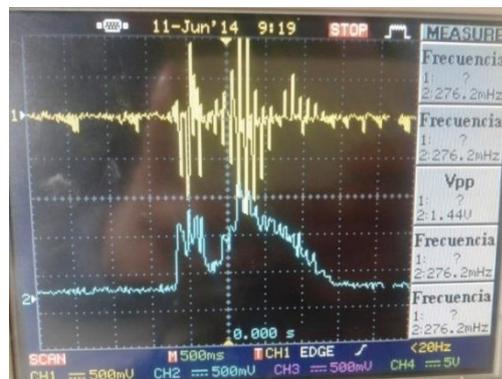


Figura 3.16 Contracción isotónica con Módulo KL-73002

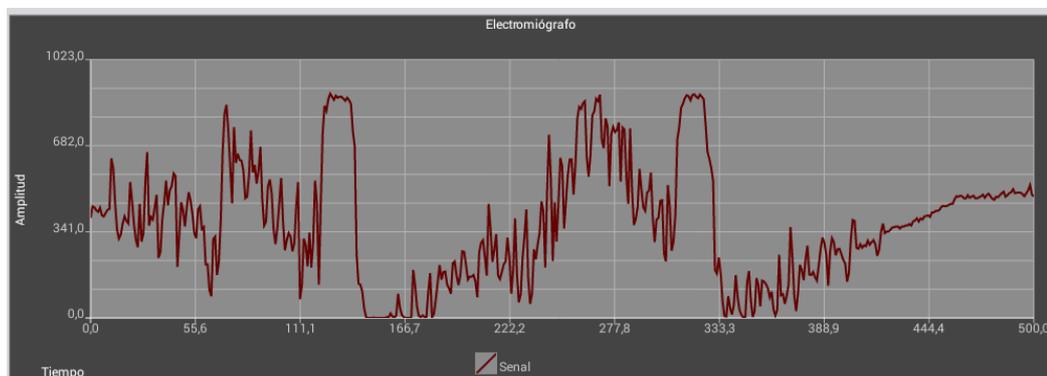


Figura 3.17 Contracción isotónica dispositivo EMG

3.4. Pruebas del Electroestimulador

Se han programado cuatro funciones para el electroestimulador de acuerdo con el grupo muscular que va a ser estimulado.

- ❖ Función Bíceps
- ❖ Función Flexores
- ❖ Función Peroné
- ❖ Función Muslos

3.4.1. Función Bíceps

Para la función bíceps se deben colocar los electrodos como se indica en la Figura 3.18.

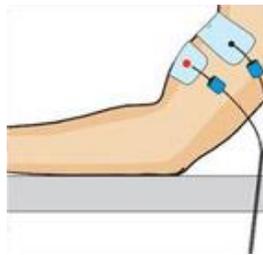


Figura 3.18 Ubicacion de electrodos función Bíceps

La Figura 3.19 muestra la onda alfa de la función bíceps, la cual cumple con la frecuencia y ancho de pulso programado.

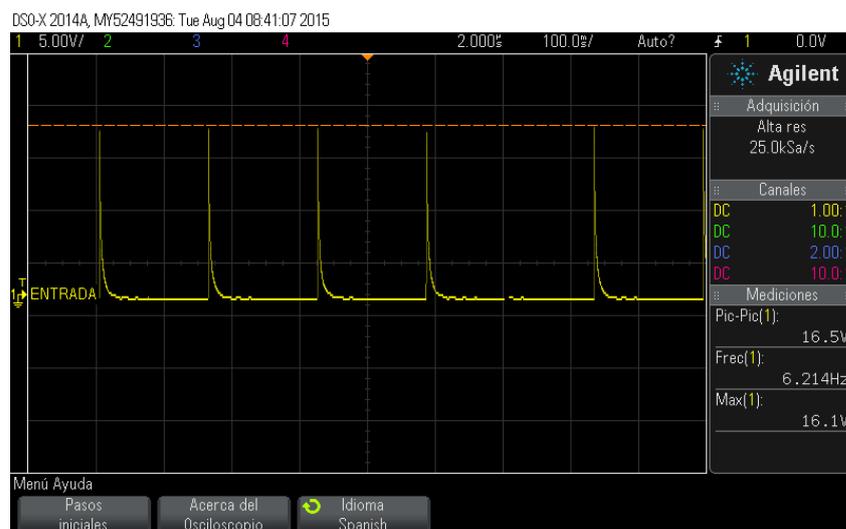


Figura 3.19 Onda Alfa función Bíceps

La Figura 3.20 presenta la combinación de la onda alfa y beta, se observa que la onda alfa dura 10 segundos mientras que la onda beta dura 5 segundos.

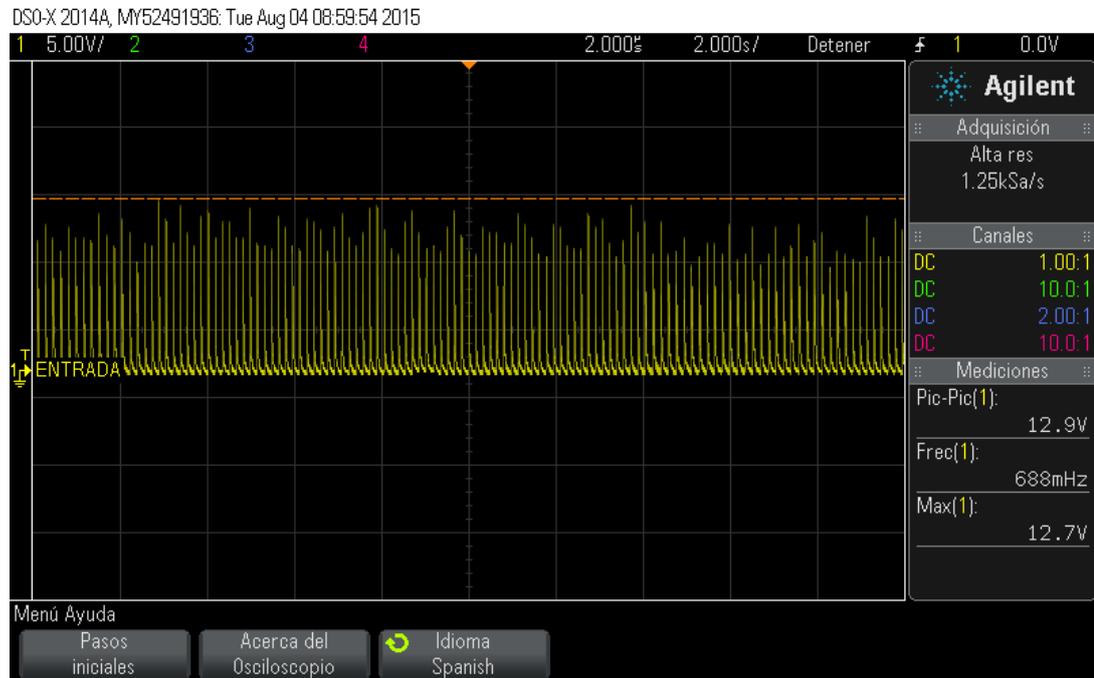


Figura 3.20 Onda alfa y beta de función bíceps

3.4.2. Función Flexores

Para la función flexores se deben colocar los electrodos como se indica en la Figura 3.21.

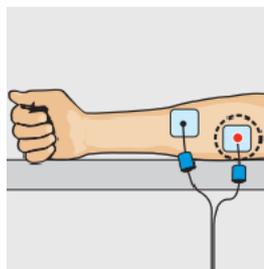


Figura 3.21 Ubicacion de electrodos función flexores

La Figura 3.22 muestra la onda alfa de la función flexores, la cual cumple con la frecuencia y ancho de pulso programado.

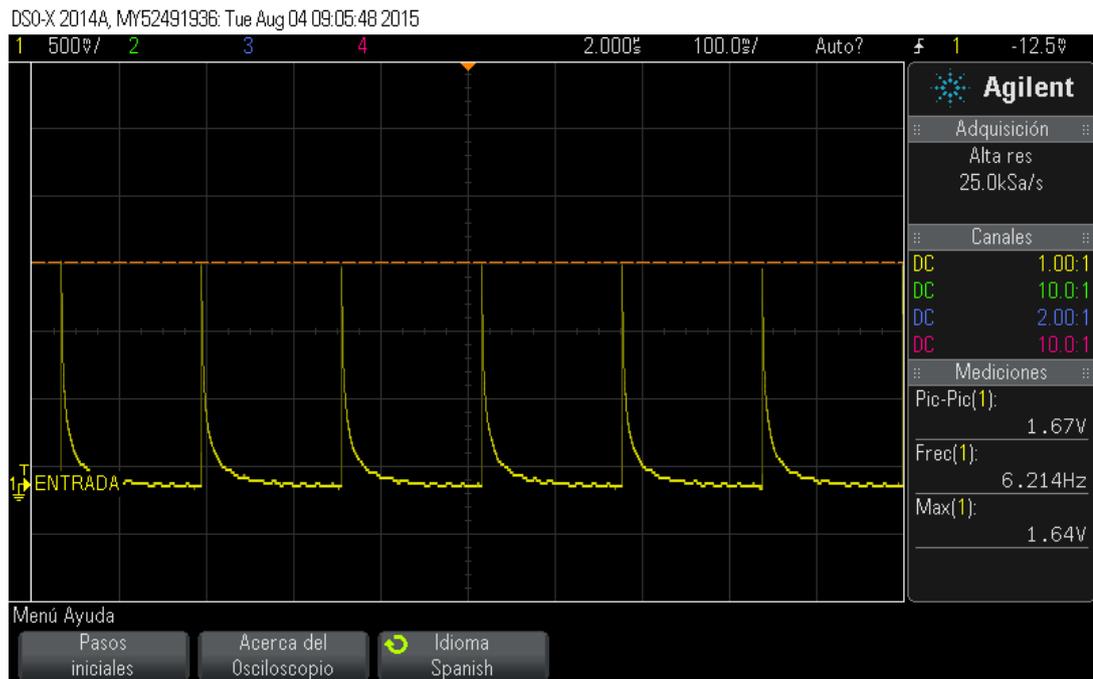


Figura 3.22 Onda Alfa función Flexores

En la Figura 3.23 se presenta la combinación de la onda alfa y beta, se observa que la onda alfa dura 10 segundos mientras que la onda beta dura 5 segundos.

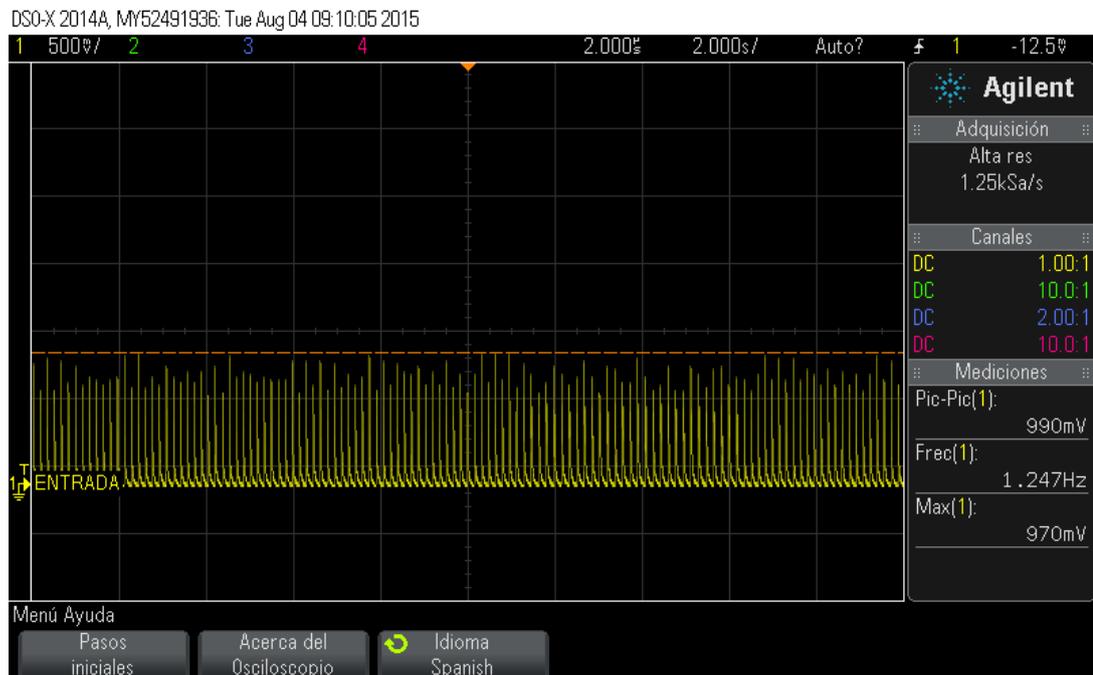


Figura 3.23 Onda alfa y beta de función flexores

3.4.3. Función Muslos

Para la función muslos se deben colocar los electrodos como se indica en la Figura 3.24.

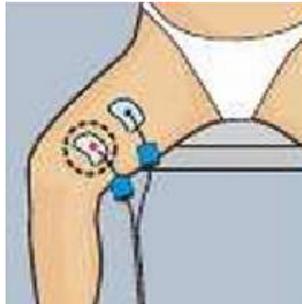


Figura 3.24 Ubicación de electrodos función muslos

La Figura 3.25 muestra la onda alfa de la función muslos, la cual cumple con la frecuencia y ancho de pulso programado.

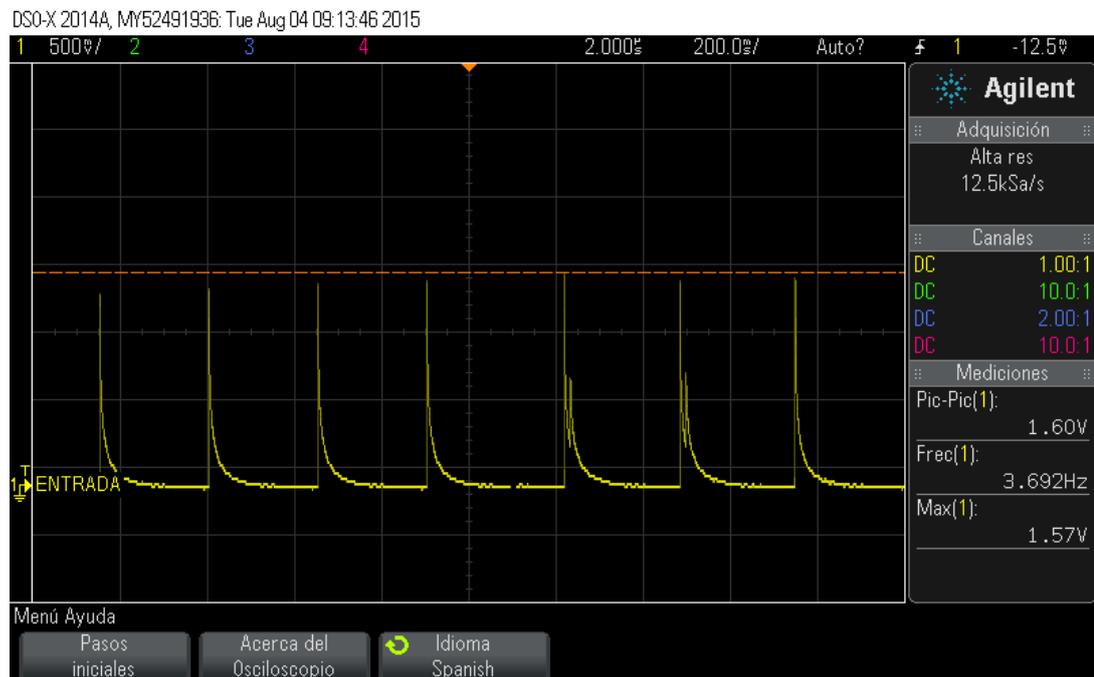


Figura 3.25 Onda Alfa función muslos

La Figura 3.26 muestra la combinación de la onda alfa y beta, se observa que la onda alfa dura 10 segundos mientras que la onda beta dura 5 segundos.

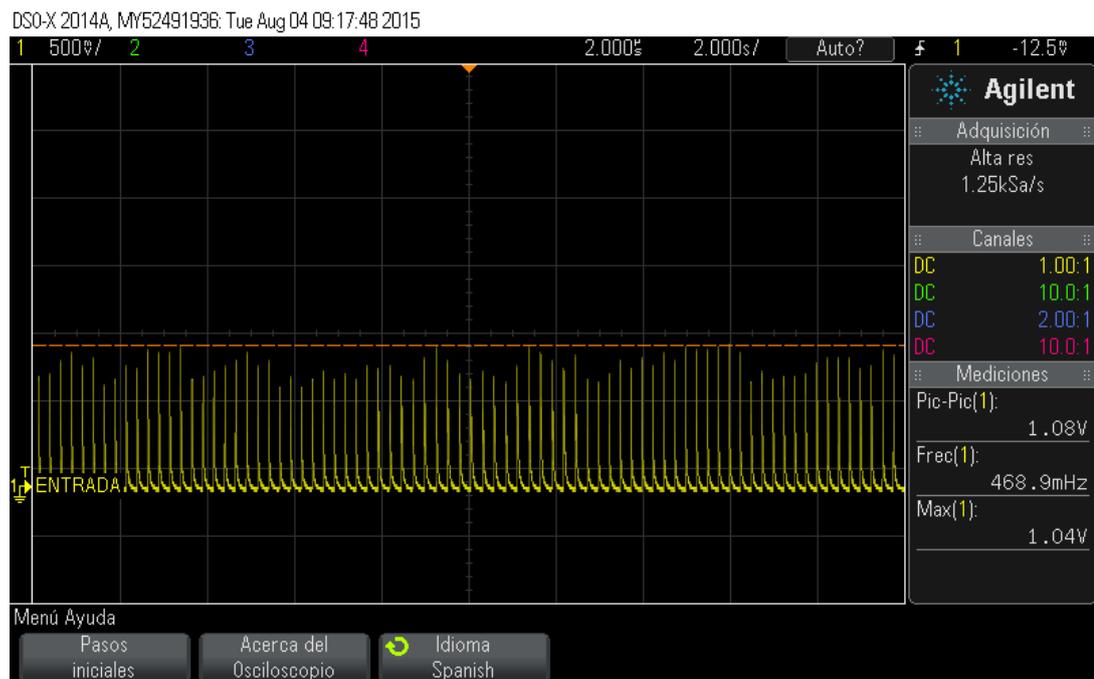


Figura 3.26 Onda alfa y beta de función flexores

3.4.4. Función Peroné

Para la función muslos se deben colocar los electrodos como se indica en la Figura 3.27.

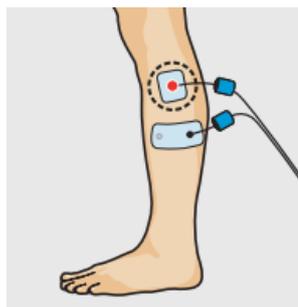


Figura 3.27 Ubicacion de electrodos función peroné

La Figura 3.28 muestra la onda alfa de la función peroné, la cual cumple con la frecuencia y ancho de pulso programado.

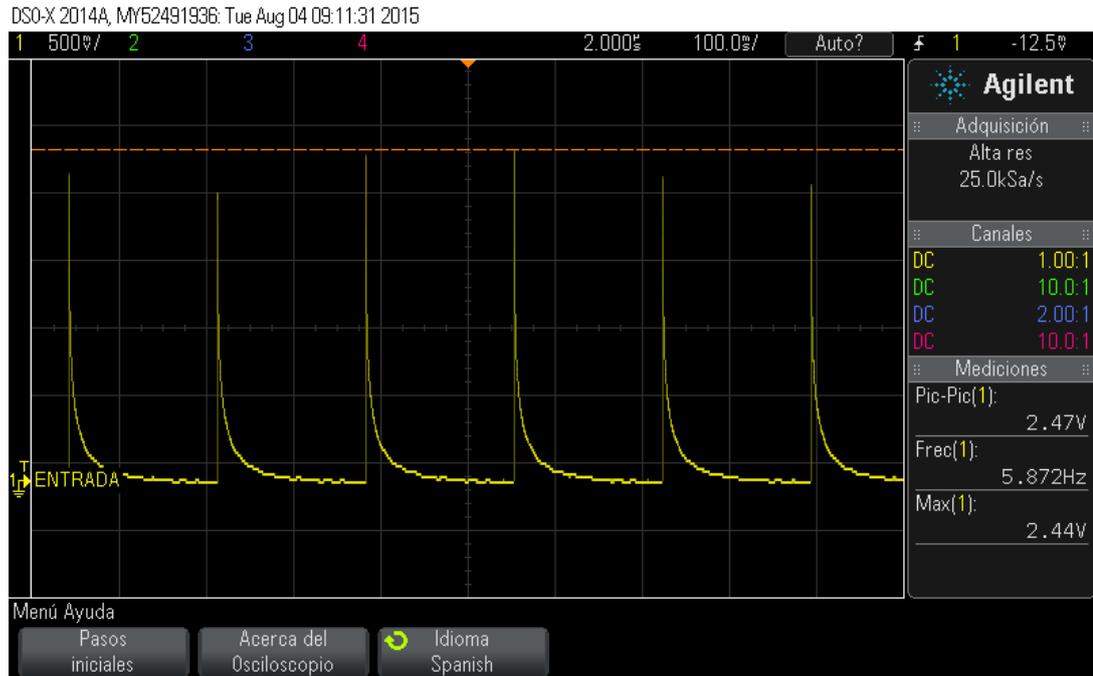


Figura 3.28 Onda Alfa función peroné

La Figura 3.29 muestra la combinación de la onda alfa y beta, se observa que la onda alfa dura 10 segundos mientras que la onda beta dura 5 segundos.

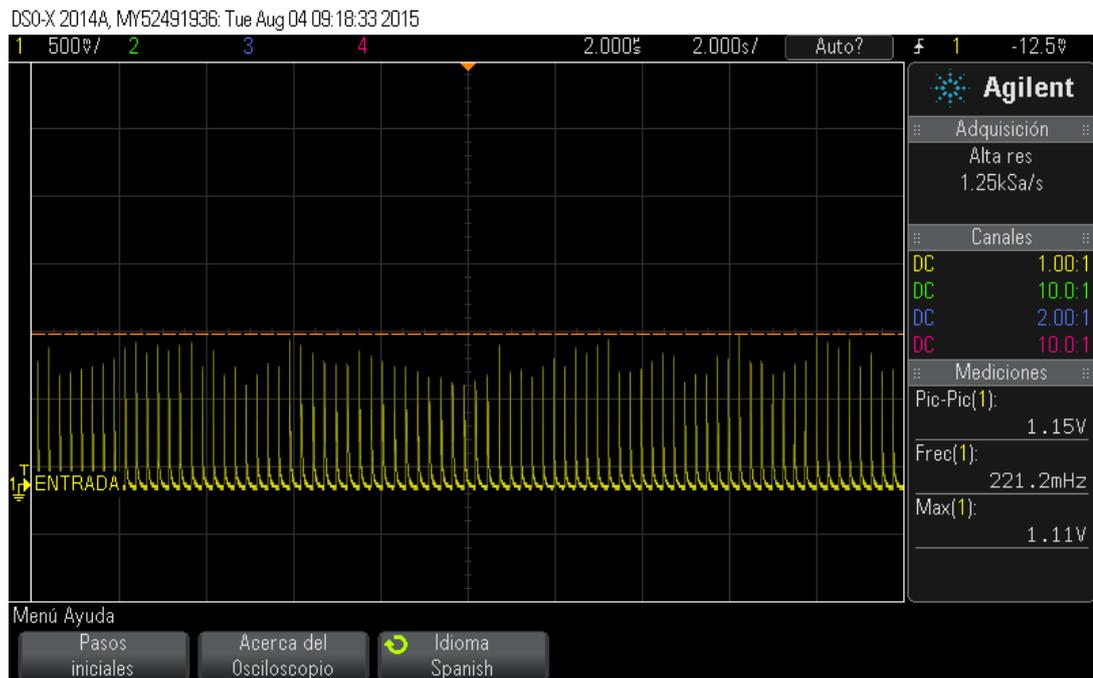


Figura 3.29 Onda alfa y beta de función peroné

3.4.5. Pruebas de intensidades del electroestimulador

Cuando estimulamos un músculo, el número de fibras que trabajan depende de la energía aplicada a la estimulación, a mayor intensidad aplicada se llega más profundamente al músculo.

Siempre se debe trabajar al límite de lo que el paciente pueda soportar, avanzando progresivamente de intensidad de sesión en sesión.

La Figura 3.30 muestra al paciente de electroestimulación, el cual debe estar ubicado en una posición cómoda y segura para evitar que las extremidades presenten movimientos peligrosos.



Figura 3.30 Paciente de electroestimulación

Se ha realizado pruebas de intensidades de electroestimulación, la Figura 3.31 muestra la onda alfa para un nivel de intensidad 1, en donde se observa que el voltaje es:

$$V = 14,4V$$

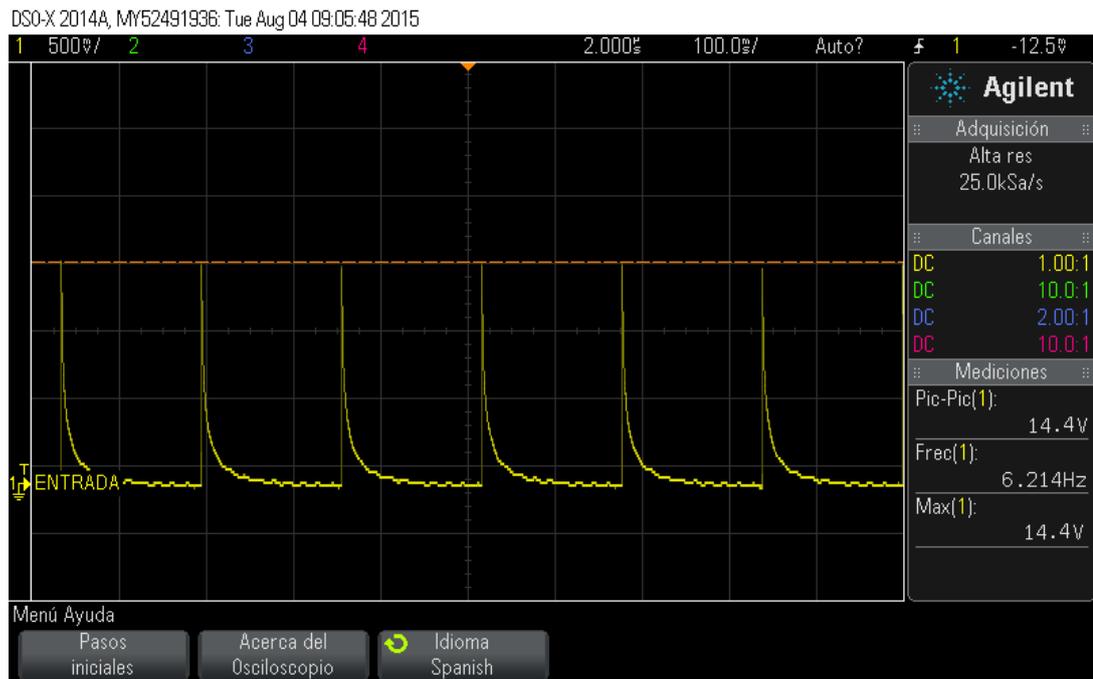


Figura 3.31 Nivel de intensidad 1 para electroestimulador

La Figura 3.32 muestra la onda alfa para un nivel de intensidad 2, en donde se observa que el voltaje es:

$$V = 23,1V$$

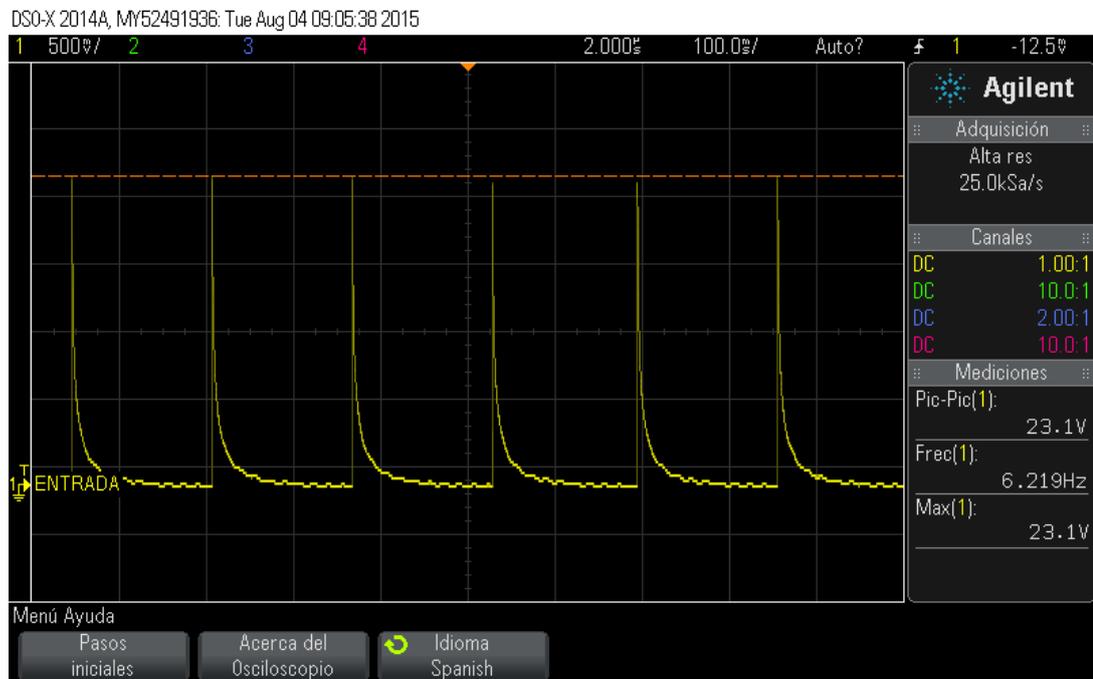


Figura 3.32 Nivel de intensidad 2 para electroestimulador

La Figura 3.33 muestra la onda alfa para un nivel de intensidad 3, en donde se observa que el voltaje es:

$$V = 27,5V$$

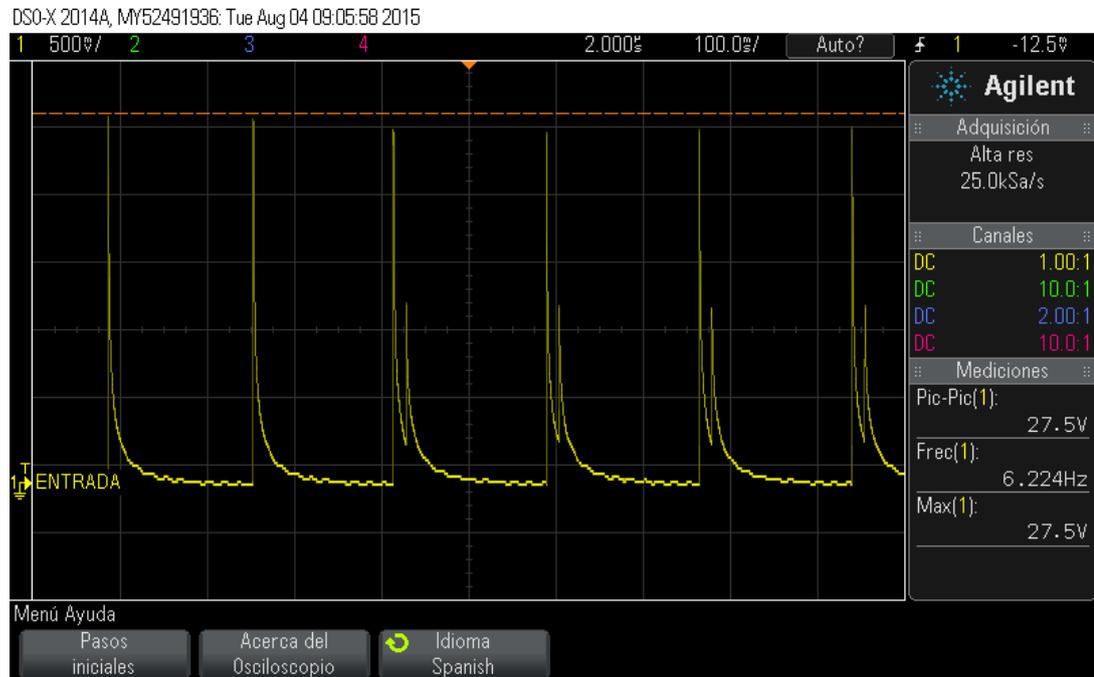


Figura 3.33 Nivel de intensidad 3 para electroestimulador

CAPÍTULO IV

4. ANÁLISIS ECONÓMICO

Uno de los motivos que impulsó al desarrollo de este equipo fue el de construir dispositivo económico que pueda satisfacer las necesidades de las personas con síntomas de enfermedades neuromusculares, para que puedan realizar el tratamiento mediante electroestimulación facilitando su recuperación en menor tiempo.

Compex es una marca del Grupo DJO, proveedores de aparatos ortopédicos de calidad que cuenta con una amplia gama de productos para la rehabilitación, el tratamiento del dolor y fisioterapia. La empresa crea, fabrica y distribuye equipos electroestimuladores, entre la cartera de sus productos se encuentra el producto My sport 500



Figura 4.1 Compex My Sport 500

Fuente: [16]

El precio de este equipo en el Ecuador es de \$2295.00 USD [16]. A continuación se presenta en la Tabla 4.1 una comparación entre las características del Compex My Sport 500 y nuestro prototipo

Tabla 4.1
Comparación Compex My Sport 500 vs. Prototipo

Características	My Sport 500	Prototipo diseñado
Tecnología	Con cables	Con cables
Conexión a internet	No	No
Conexión bluetooth	No	Si
Carga de objetivos	No	No
Histórico de utilización	No	No
Número de canales	4	1
Número de programas	10	4
Ondas moduladas	Si	Si
Pantalla	Monocromática	Color
Compatible Android	No	Si
Suministro de energía	Batería recargable	Batería recargable
Potencia	1 mA 150 Hz	1 mA 150 Hz
Niveles de intensidad	5	3
Electromiografía	No	Si
Cable EMG	No	Si
Cable	Si	Si
Electroestimulador		
Accesorios	Bolsa de transporte Cargador de batería Electrodos	Bolsa de transportes Cargador de batería Electrodos

Por otro lado se coloca el detalle económico del equipo de electromiograma y electroestimulador inalámbrico en la Tabla 4.2.

Tabla 4.2
Costo total Electromiograma y Electroestimulador

Cantidad	Referencia	Valor unitario	Valor total
1	Potenciómetros precisión	0,65	0,65
2	Capacitor 100nF	0,25	0,50
1	Capacitor 47 μ F	0,14	0,14
5	Transistores SMD MMBT5551	0,15	0,75
4	Transistores SMD MMBT5401	0,15	0,60
3	Transistores SMD MMBT2222	0,16	0,48
3	Transistores 2N3906	0,12	0,36
4	Diodos rectificadores 1N4148	0,15	0,60
1	Bobina 2.2mH	0,55	0,55
27	Resistencias	0,02	0,54
1	Jack USB	0,60	0,60
1	Placa PCB	5,00	5,00
20	Cables de Protoboard	0,15	3,00
5	Borneras de 3 Pines	0,35	1,75

CONTINÚA



5	Borneras 2 Pines	0,28	1,40
3	Zócalos de 8 Pines	0,12	0,36
4	Zócalos de 14 Pines	0,16	0,64
1	Zócalos de 24 Pines	0,20	0,20
1	Shield EKG/EMG	45,00	45,00
1	Arduino UNO	35,00	35,00
1	Microcontrolador AVR ATMEGA328P	6,00	6,00
2	Módulos Bluetooth HC-05	27,50	55,00
1	Batería	20,00	20,00
1	Caja de Acrílico	30,00	30,00
6	Postes	0,40	2,40
1	Cable apantallado para EMG	50,00	50,00
1	Cable USB Macho / Macho	3,00	3,00
1	Maleta y Bordado diseñado	15,00	15,00
1	Galón Gel para Ultra Sonido	10,00	10,00
20	Electrodos de prueba	1,25	25,00
1	Tablet Samsung Tab4	400,00	400,00
1	Aplicación Android	100,00	100,00
50	Hora de Ingeniería	5,00	250,00
TOTAL \$			1064,52

Para la implementación del prototipo se empleó la cantidad de mil sesenta y cuatro dólares, en donde los costos más elevados corresponden a la Tablet, y a los módulos necesarios para el funcionamiento del equipo de Electromiograma y Electroestimulador.

Realizando una comparación de precios entre el Compex y nuestro prototipo se puede observar que es más económico y versátil, esto se debe a que nuestro prototipo cuenta con una aplicación para el sistema operativo Android, dotándole a la vez de una pantalla a color que resulta más atractiva para el usuario.

Existe un ahorro de \$1230.48 dólares, en el primer prototipo, y tomando en cuenta que la producción en masa reduciría el costo del equipo en un 40% del precio normal; dando una reducción en el precio de \$1869.20, resultando cada equipo con un precio comercial para el mercado nacional de \$425.80.

4.1. Alcances y limitaciones

4.1.1. Alcances

En el transcurso del desarrollo del prototipo se pudo notar los siguientes alcances, los mismos que se pone a consideración.

- ❖ Posibilita la visualización de la señal del electromiograma en tiempo real, registrando la actividad eléctrica de los músculos en diversas pruebas.
- ❖ Es posible detener la señal del electromiograma y guardarla en el dispositivo android para que pueda ser analizada por un experto.
- ❖ Permite realizar la electroestimulación de cuatro grupos musculares seleccionándolos desde la aplicación Android.
- ❖ El electroestimulador dispone de tres niveles de intensidad que pueden ser configurados en tiempo real.
- ❖ El tiempo de la sesión de electroestimulación puede ser configurado como máximo de 30 min. dando la posibilidad de establecer intervalos de 5 min.
- ❖ Concentra la información necesaria (configuración y visualización) en una sola interfaz, reduciendo así la complejidad de manejo del equipo.
- ❖ El dispositivo es de fácil transportación y puede ser llevado a diferentes lugares, ya que cuenta con un sistema de batería recargable integrado.

4.1.2. Limitaciones

Al finalizar el proyecto se pudo confirmar ciertas limitaciones que presenta el sistema y son expuestas a continuación.

- ❖ La aplicación de la Tablet no puede ser instalada en dispositivos móviles con una versión inferior de Android 4.0.3, esto se debe a que no cumple con las características necesarias de compatibilidad.
- ❖ La comunicación bluetooth tiene un alcance aproximadamente de 10 m, al estar fuera de este rango la transferencia de los datos están propensos a interrupciones y pérdidas de los mismos.

- ❖ El sistema de baterías recargables permite cierto tiempo de autonomía, es importante que no se trabaje con un nivel bajo de carga.
- ❖ El tamaño del cable del EMG es corto por lo cual el paciente no se puede mover cómodamente.

4.2. Proyectos futuros

A un futuro se puede realizar mejoras en este equipo, como la utilización de nuevas tecnologías como son los electrodos inalámbricos para la implementación del EMG, que ayudaría a la reducción de cables dando una mejor comodidad al paciente.

Se puede aumentar el número de funciones en el electroestimulador para tener un equipo más completo y de mayor eficiencia.

Al tener la aplicación en un dispositivo Android con conectividad a Internet se lo puede introducir al mundo de la Telemedicina apoyando al paciente desde la comodidad de su hogar por un experto en cualquier parte del mundo.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- ❖ Se cumplió con el objetivo principal del proyecto, el cual consiste en el diseño y construcción de un electroestimulador y electromiograma inalámbrico para la detección y tratamiento de enfermedades neuromusculares.
- ❖ Se estudió e identificó claramente, cada uno de los componentes y elementos que intervienen en el diseño y construcción de un electroestimulador y electromiograma inalámbrico.
- ❖ Se desarrolló un dispositivo compacto que integra las funciones de un electromiograma y electroestimulador en una sola aplicación para dispositivos móviles bajo la plataforma Android.
- ❖ Se implementó una aplicación móvil sencilla, amigable con el usuario que satisface las necesidades tanto de la adquisición de la señal del EMG como de la configuración de los parámetros del electroestimulador, además permite una fácil conexión entre el dispositivo móvil con el sistema Android, a través del protocolo de comunicación inalámbrica Bluetooth.
- ❖ El dispositivo desarrollado cumplió con las condiciones y requerimientos para dispositivos médicos donde se utilizó una velocidad de 9600 que garantizó que los datos transmitidos cumplan con las especificaciones de monitoreo en equipos médicos.
- ❖ El electromiograma implementado permite la medición de la señal eléctrica de los músculos, con una alta velocidad de actualización de los datos.
- ❖ Los impulsos eléctricos generados por el electroestimulador permiten desencadenar potenciales de acción en distintos tipos de fibras nerviosas, imponiendo una respuesta mecánica muscular dependiente de los parámetros de estimulación.
- ❖ Las gráficas obtenidas difieren en un pequeño porcentaje de los valores medidos con las obtenidas con equipos patrones de medicina,

debido principalmente a las tolerancias de los elementos pasivos utilizados.

- ❖ La visualización de la señal electromiográfica (EMG) se lo hizo con la ayuda de la librería Androidplot que permite graficar un vector numérico de n posiciones, de una forma sencilla.
- ❖ La comunicación entre el electromiograma y electroestimulador con el dispositivo móvil con sistema Android fue efectiva dentro de un rango de 10 metros con línea de vista, luego del cual la comunicación se ve interrumpida.
- ❖ Se notó que la forma de la señal del EMG tomada en diferentes pacientes difiere debido a las características de la piel que presentan diversas resistencias.
- ❖ El aumento en el orden del filtro Butterworth tipo Notch implementado disminuye notablemente el ruido en las señales del EMG.
- ❖ Con la electroestimulación, la excitación se produce directamente sobre el nervio motor gracias a los impulsos eléctricos perfectamente controlados que garantizan la eficacia, la seguridad y el confort en el uso. De este modo, el músculo es incapaz de notar la diferencia entre una contracción voluntaria ordenada por el cerebro y una contracción inducida eléctricamente.

5.2. Recomendaciones

- ❖ Para una buena recepción de la señal electromiográfica se recomienda una buena limpieza con alcohol y si es necesario rasurarse la zona donde van colocados los electrodos con la finalidad de reducir al máximo la resistencia eléctrica causada por los restos de grasa y existencia de vellos.
- ❖ No recargar nunca el aparato cuando los cables estén conectados al electroestimulador y los electrodos en contacto con el usuario, ya que existe el riesgo de shock eléctrico.
- ❖ No desconectar los cables del electroestimulador durante una sesión mientras el aparato siga conectado, previamente debe ser parado.
- ❖ No utilizar un mismo juego de electrodos más de quince sesiones, ya que la calidad del contacto entre el electrodo y la piel se deteriora de forma progresiva.
- ❖ El EMG es muy sensible al ruido por ello se debe tomar en cuenta una correcta aplicación y sujeción de los electrodos en el paciente para evitar cualquier ruido externo.
- ❖ El dispositivo móvil debe contener la versión Android 4.0.3 o superior.
- ❖ Tomar la precaución de retirar del paciente todo tipo de objetos que puedan alterar las mediciones como relojes, decoraciones, etc.

REFERENCIAS BIBLIOGRÁFICAS

- [1] INEC Instituto Nacional de Estadística y Censos, *Estadísticas de Camas y Egresos Hospitalarios*, Quito, 2014.
- [2] L. Cifuentes, *Electroterapia, Electrodiagnóstico, Electromiografía*, Quito: PH Ediciones, 2006.
- [3] D. Montoya, «Educación Física,» [En línea]. Available: <http://edfis13.blogspot.com/2015/03/contraccion-muscular-una-muscular-por.html>. [Último acceso: 26 Marzo 2015].
- [4] R. Jules, R. Serge y W. Steven, *MANUAL DEL ESPECIALISTA EN REHABILITACIÓN (Cartoné y bicolor)*, Barcelona: Paidotribo, 2005.
- [5] J. M. Rodriguez Martin, *Electroterapia en fisioterapia*, vol. II, España: Editoria Médica Panamericana S.A., 2004.
- [6] W. Bracero Tobar, *Electroestimulador muscular*, Quito, 2003.
- [7] O. Flores, «BATALLA DE MICROCONTROLADORES ¿AVR o PIC?,» 1 SEPTIEMBRE 2009. [En línea]. Available: https://microcontroladores2utec.files.wordpress.com/2009/11/180909_articulo_colaboracion_boletin_fica_omar_otoniel_flores.pdf. [Último acceso: 13 Abril 2015].
- [8] Arduino, «Arduino Uno,» 1 Enero 2015. [En línea]. Available: <http://www.arduino.cc/en/Main/ArduinoBoardUno>. [Último acceso: 20 Abril 2015].

- [9] OLIMEX, «SHIELD-EKG-EMG bio-feedback shield USER MANUAL,» 12 Junio 2014. [En línea]. Available: <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/resources/SHIELD-EKG-EMG.pdf>. [Último acceso: 5 Marzo 2015].
- [10] JAVA, «JAVA,» 12 Enero 2015. [En línea]. Available: www.java.com. [Último acceso: 9 Abril 2015].
- [11] GOOGLE, «ANDROID,» Digital Globe, 23 Octubre 2014. [En línea]. Available: www.android.com. [Último acceso: 26 Abril 2015].
- [12] J. Prieto, Introducción a los Sistemas de Comunicación Inalambrica, Catalunya: Universidad Oberta de Catalunya, 2012.
- [13] R. Sallent, J. Valenzuela y R. Agustí, Principios de comunicaciones móviles, Barcelona: Ediciones UPC Universidad Politécnica de Catalunya, 2003.
- [14] D. W. Hart, Electrónica de Potencia, Madrid: Prentice Hall, 2001.
- [15] PROMETEC, «EL MÓDULO BLUETOOTH HC-05,» [En línea]. Available: <http://www.prometec.net/bt-hc05/>. [Último acceso: 15 JUNIO 2015].
- [16] COMPEX, «Compex Your intelligent training parther,» 1 Enero 2015. [En línea]. Available: http://www.compex.info/es_ES/index.html. [Último acceso: 28 Junio 2015].

ANEXOS

ÍNDICE DE ANEXOS

ANEXO A: GLOSARIO DE TÉRMINOS

ANEXO B: CIRCUITOS ESQUEMATICOS

ANEXO C: DIAGRAMAS IMPRESOS

ANEXO D: PLACAS TERMINADAS

ANEXO E: ARDUINO UNO

ANEXO F: SHIELD ECK / EMG

ANEXO G: MICROCONTROLADOR AVR 328P

ANEXO H: PROGRAMACIÓN EN ANDROID STUDIO

ANEXO I: PROGRAMACIÓN EN ARDUINO UNO

ANEXO J: PROGRAMACION EN BASCOM - AVR

ANEXO K: CERTIFICACIÓN MÉDICA

ANEXO L: MANUAL DE USUARIO

ANEXO A: GLOSARIO DE TÉRMINOS

Arduino.- Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo.

AVR.- Son una familia de microcontroladores RISC del fabricante estadounidense Atmel.

Bloqueo anodal.- bloqueo local de la conducción nerviosa causado por hiperpolarización de la membrana de la célula nerviosa mediante un estímulo eléctrico.

Electrodo activo.- sinónimo de electrodo explorador. También se refiere a un electrodo EMG de superficie con una impedancia de entrada alta.

Electrodo de registro.- dispositivo que se utiliza para registrar una diferencia de potencial eléctrico.

Electrodo de tierra.- un electrodo conectado al paciente y a un cuerpo conductor grande (como el suelo) que se utiliza como una guía de regreso común para un circuito eléctrico y como un punto de referencia potencial cero arbitrario.

Electrodo estimulador.- Dispositivo que se utiliza para aplicar corriente. Toda estimulación eléctrica necesita de dos electrodos: la terminal negativa se denomina cátodo y la terminal positiva ánodo.

Electrodos estimuladores bipolares.- se denominan bipolares si están encerrados o unidos juntos.

Electrodos estimuladores monopolares.- se denominan monopolares si no están unido o encerrados juntos. Generalmente la estimulación eléctrica para los estudios de conducción nerviosa requiere la aplicación del cátodo para producir la despolarización de las fibras del tronco del nervio. Si el ánodo se coloca inadvertidamente sobre el cátodo y los electrodo de

registro, puede ocurrir un bloqueo focal de la conducción nerviosa (bloqueo anodal) y dar lugar a un estudio técnicamente insatisfactorio.

Electromiografía (EMG).- definida de forma exacta, registro y estudio de la actividad eléctrica de inserción, espontánea y voluntaria del músculo.

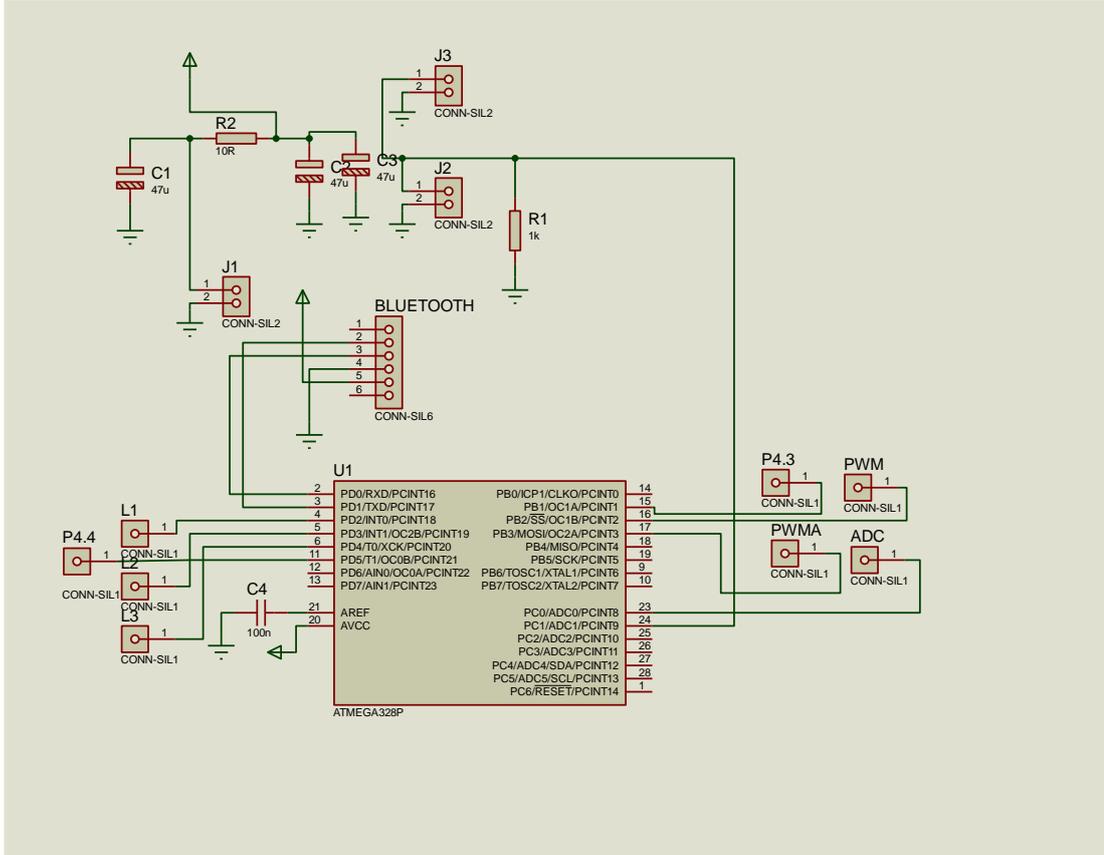
Electromiógrafo.- material que se usa para activar, registrar, llevar a cabo y exponer potenciales de acción nerviosos y musculares con el propósito de evaluar la función nerviosa y muscular.

Electromiograma.- es el registro obtenido por una electromiografía

Estimulo.- cualquier agente externo, estado o cambio que puede influir en la actividad de una célula, tejido u organismo

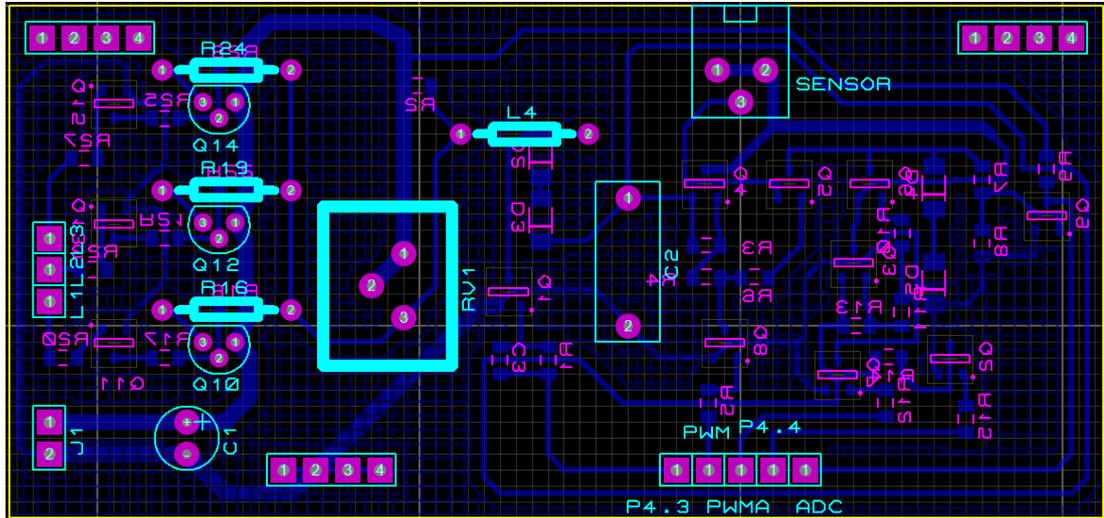
Ruido.- definido exactamente, potenciales producidos por electrodos, cables, amplificador o medios de almacenaje y sin relación con potenciales de origen biológico.

Módulo de control

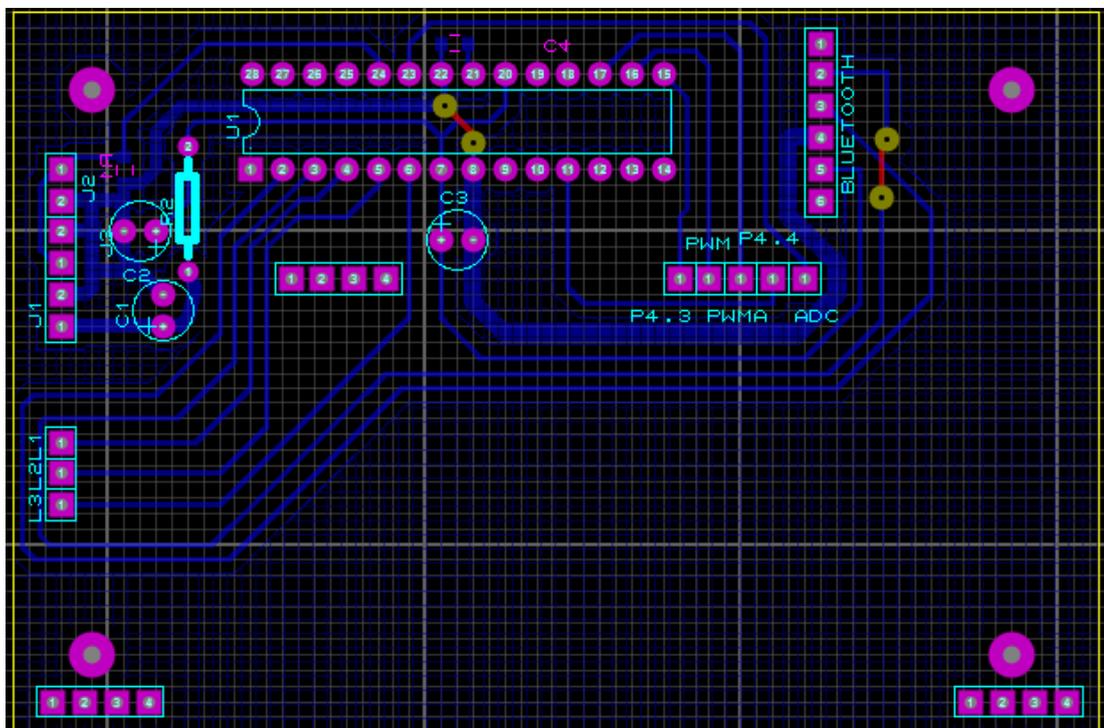


ANEXO C: DIAGRAMAS IMPRESOS

PCB Módulo de alto voltaje



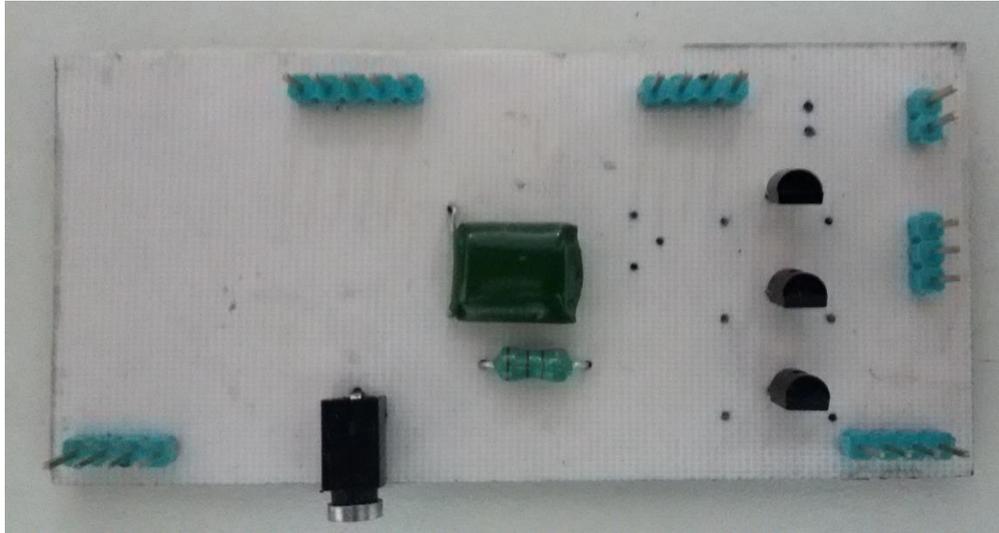
PCB Módulo de control



ANEXO D: PLACAS TERMINADAS

Módulo de alto voltaje

Vista superior

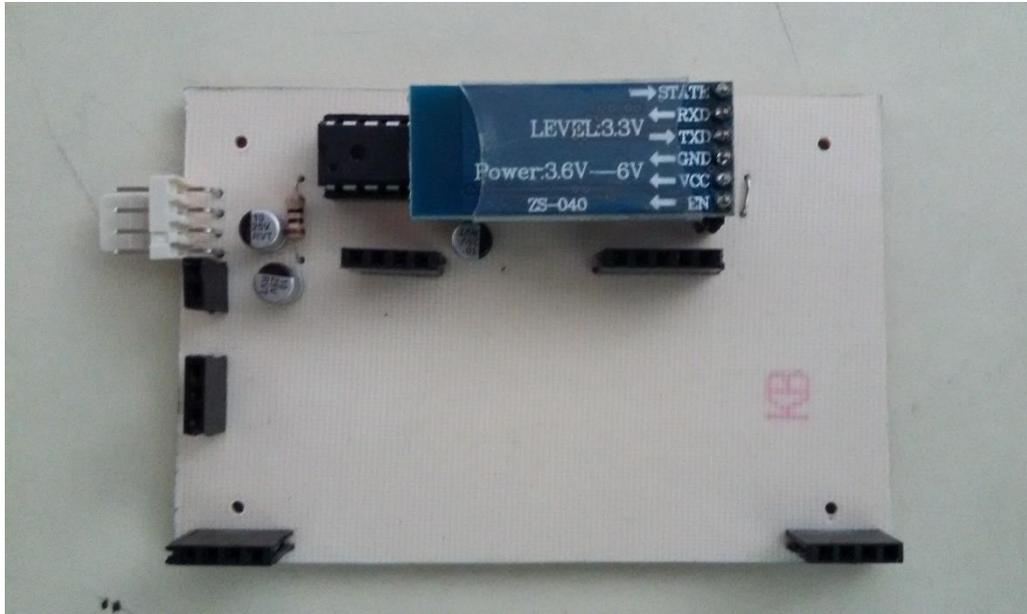


Vista inferior

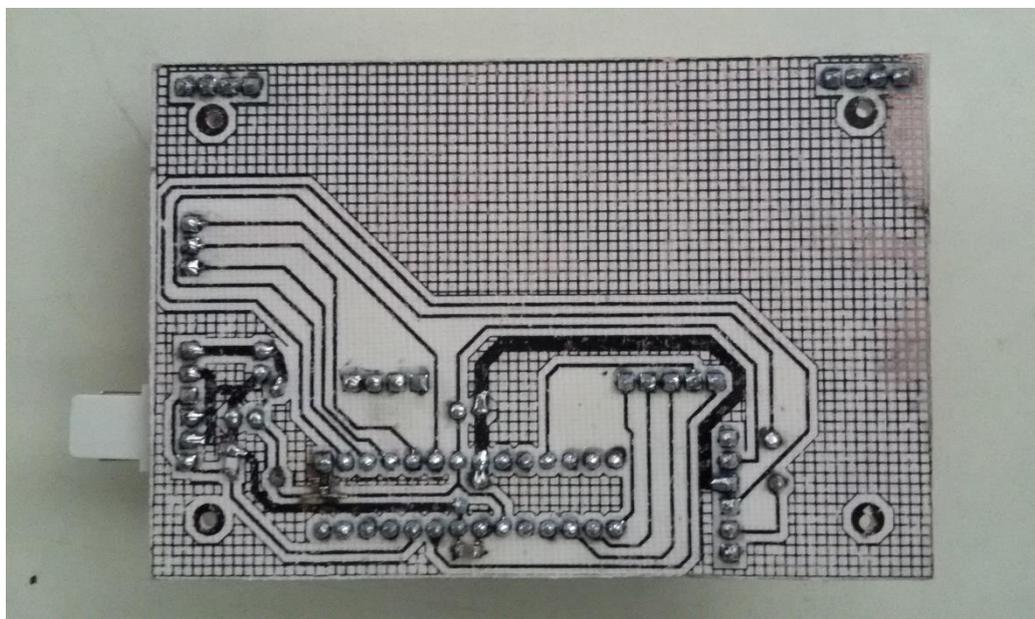


Módulo de control

Vista superior



Vista inferior



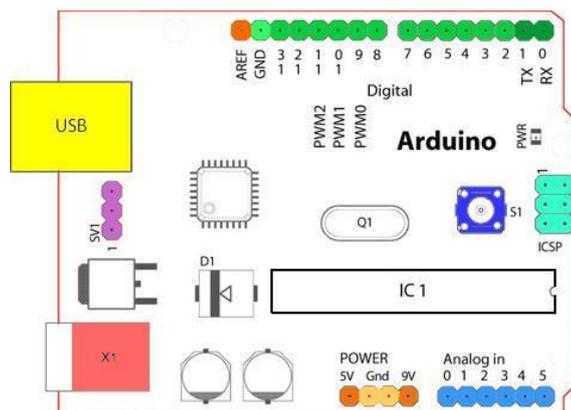
ANEXO E: ARDUINO UNO

El Uno Arduino es una placa electrónica basada en el ATmega328 (ficha técnica). Tiene 14 entradas / salidas digitales pines (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reset. Contiene todo lo necesario para apoyar el microcontrolador, basta con conectarlo a un ordenador con un cable USB o alimentarlo con un adaptador AC-DC o batería para empezar.

“Uno” significa uno en italiano y se le llamo así por el lanzamiento de Arduino 1.0. El Uno y la versión 1.0 serán las versiones de referencia para los siguientes Arduinos. El Uno es el último de una serie de placas Arduino USB, y el modelo de referencia para la plataforma Arduino.

Componentes de la Placa Arduino UNO

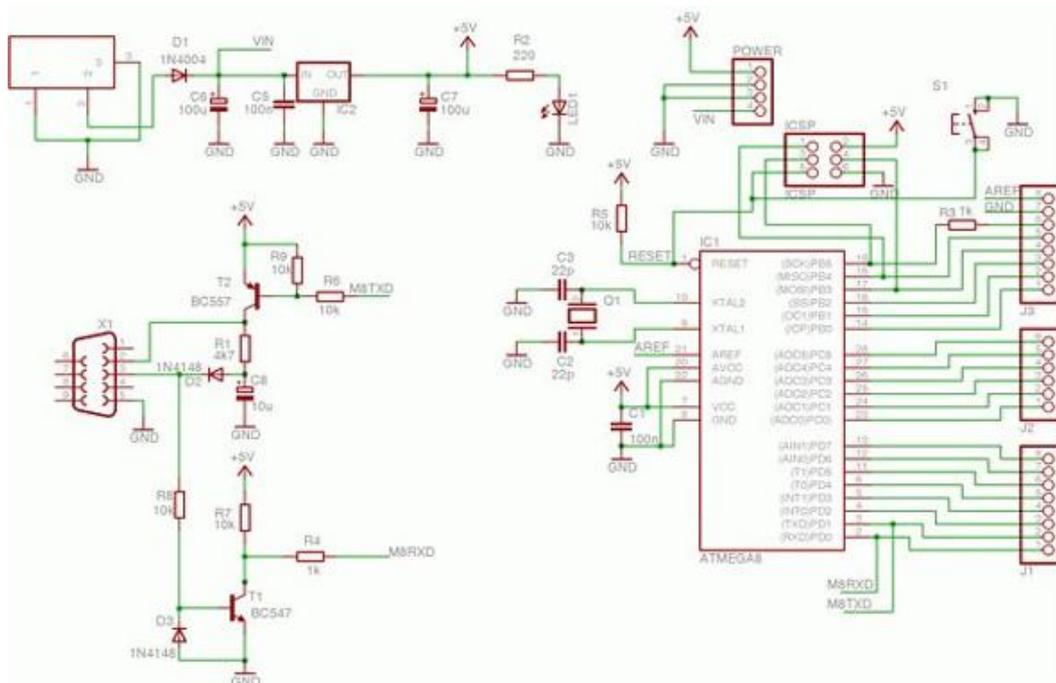
Mirando a la placa desde la parte de arriba, este es el esquema de lo que puedes ver (los componentes de la placa con los que puedes interactuar en su uso normal están resaltados tal como se muestra en la siguiente figura).



Esquema eléctrico del Arduino UNO

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así

pues eres libre de adaptarlos a tus necesidades. En la siguiente figura se muestra el esquema eléctrico del Arduino uno.



Arduino recibió una Mención Honorífica en la sección *Digital Communities* de la edición del 2006 del *Ars Electronica Prix*. El equipo Arduino (Arduino team) es: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis.

Esta placa es sencilla pero ofrece muchas prestaciones y por poco dinero se pueden hacer muchos montajes y disfrutar de la electrónica y los microcontroladores, hacer robots y jugar con ellos por poco dinero, en el mercado hay cientos de componentes preparados para esta placa sin necesidad de fabricarlos.

ANEXO F: SHIELD ECK / EMG

Se trata de un escudo ECG / EMG que permite Arduino como tablas para capturar señales Electrocardiografía electromiografía. El escudo abre nuevas posibilidades para experimentar con retroalimentación biológica. Usted puede controlar su ritmo cardíaco y registrar el pulso, reconocer gestos por el seguimiento y analizar la actividad muscula como se hace en este proyecto

CARACTERÍSTICAS

- ❖ Cabeceras apilables hasta 6 canales pueden ser apilados y conectados a entradas analógicas A0-A6
- ❖ Generación de la señal de calibración de salida digital D4 / D9
- ❖ Potenciómetro Trimmer preciso para la calibración (todas las placas se envían completamente montado, probado y calibrado de modo que usted no tiene que hacer esto a menos que usted quiere ver cómo funciona cosas)
- ❖ Conector de entrada para electrodos normales o activos
- ❖ Funciona tanto con 3.3V y 5V Arduino tableros.

Conexión de la Shield con Arduino

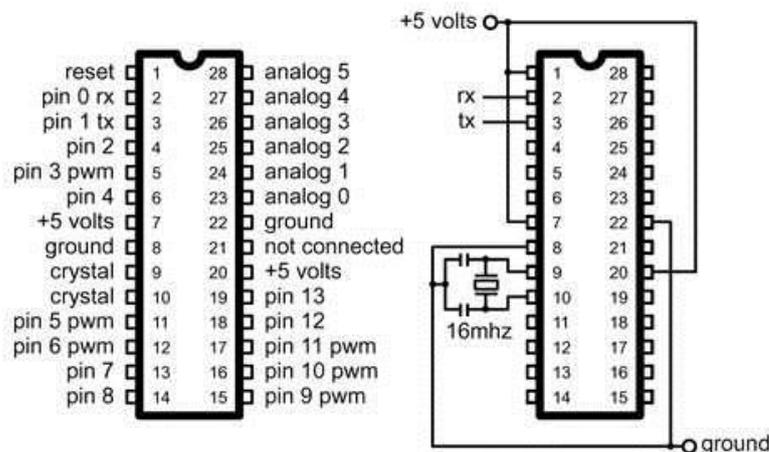
Los conectores siguen la especificación ARDUINO para conexión de la shield. El escudo viene con conectores soldadas dejándolo listo para el montaje a bordo compatible con la posibilidad de tener otra shield montado en él.

En la figura siguiente figura se muestra el diagrama esquemático de la shield EKG/EMG

ANEXO G: MICROCONTROLADOR AVR 328P

Es un circuito integrado de alto rendimiento Atmel picoPower 8 bits microcontrolador RISC AVR combina memoria flash de 32 KB ISP con capacidades de lectura y escritura, mientras 1024B EEPROM, SRAM 2 KB, 23 líneas de propósito general de E / S, 32 registros de propósito generales de trabajo, tres temporizadores flexibles / contadores con comparan los modos, las interrupciones internas y externas, USART programable de serie, una interfaz en serie de 2 hilos byte-orientada, puerto serial SPI, un 6 canales de convertidor A / D de 10 bits (8-canales en TQFP y QFN / FML paquetes), temporizador de vigilancia programable con oscilador interno, y cinco de software modos de ahorro de energía seleccionable. El dispositivo funciona entre 1.8 hasta 5 voltios.

En la siguiente figura se observa la distribución de pines del microcontrolador utilizado para la etapa de control del electroestimulador.



Este chip trabaja en modo IDE, necesitará un cristal de 16MHz externo o resonador, una alimentación de 5V, y una conexión en serie.

El Microcontrolador de 8 bits ATmega328 de Atmel viene en un encapsulado DIP de 28 pines. Es como el ATmega168, con el doble de espacio flash. 32K de espacio de programa. 23 I / O líneas, 6 de las cuales

son los canales para el ADC de 10 bits. Funciona hasta 20MHz con cristal externo.

En la siguiente tabla se muestra los parámetros clave para atmega328P.

PARAMETRO	VALOR
Flash (Kbytes)	32 Kbytes
Número de pines	32
Max. Frecuencia de operación. (MHz)	20 MHz
CPU	8-bit AVR
Número de Canales Touch	16
Hardware QTouch adquisición	No
Max I / O Pins	23
Ext Interrupciones	24
UART	1
LCD gráfico	No
Decodificador del vídeo	No
Interfaz de la cámara	No
Canales de ADC	8
Resolución ADC (bits)	10
Velocidad ADC (ksps)	15
Comparadores analógicos	1
Resistiva Pantalla táctil	No
Resolución DAC (bits)	0
Temperatura Sensor	Si
Crypto Motor	No
SRAM (Kbytes)	2
EEPROM (Bytes)	1024
Auto de memoria de programa	Si
Interfaz de bus externo	0
DRAM Memoria	No
NAND Interfaz	No
picoPower	Si
Temperatura Range (grados C)	-40 A 85
E / S de alimentación Clase	1.8 a 5.5
Tensión de funcionamiento (Vcc)	1.8 a 5.5
Timers	3
Comparar canales de salida	6
Canales de captura de entrada	1
Canales PWM	6
32kHz RTC	Si
Calibrado oscilador RC	Si
Watchdog	Si

ANEXO H: PROGRAMACIÓN EN ANDROID STUDIO

Clase EMG

```
package ru.sash0k.bluetooth_terminal;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    Button bElectromiografo, bElectroestimulador;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bElectromiografo = (Button) findViewById(R.id.EMIO);
        bElectroestimulador = (Button) findViewById(R.id.ESTI);

        bElectromiografo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showConsoleActivity(1);
            }
        });

        bElectroestimulador.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showConsoleActivity(2);
            }
        });
    }

    private void showConsoleActivity(int pag) {
        switch (pag){
            case 1:
                Electromiografo.show(this);
                break;
            case 2:
                Electroestimulador.show(this);
                break;
            default:
                MainActivity.show(this);
        }
    }
}
```

```

        break;
    }
}

static void show(Context context) {
    final Intent intent = new Intent(context, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP
Intent.FLAG_ACTIVITY_NO_HISTORY);
    context.startActivity(intent);
}
}
}

```

CLASE ELECTROMIÓGRAFO

```

package ru.sash0k.bluetooth_terminal;
import android.app.Activity;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.preference.PreferenceManager;
import android.text.InputFilter;
import android.text.InputType;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.TextView;

import com.actionbarsherlock.app.ActionBar;
import com.androidplot.util.PlotStatistics;
import com.androidplot.xy.BoundaryMode;
import com.androidplot.xy.LineAndPointFormatter;
import com.androidplot.xy.SimpleXYSeries;
import com.androidplot.xy.XYPlot;

import java.io.UnsupportedEncodingException;
import java.lang.ref.WeakReference;
import java.util.Timer;
import java.util.TimerTask;

```

```

import ru.sash0k.bluetooth_terminal.bluetooth.DeviceConnector;
import ru.sash0k.bluetooth_terminal.bluetooth.DeviceListActivity;

public final class Electromiografo extends BaseActivity {
    private static final String DEVICE_NAME = "DEVICE_NAME";
    private static final String LOG = "LOG";

    private static String MSG_NOT_CONNECTED;
    private static String MSG_CONNECTING;
    private static String MSG_CONNECTED;

    private static DeviceConnector connector;
    private static BluetoothResponseHandler mHandler;

    private boolean hexMode, needClean;
    private String command_ending;
    private String deviceName;

    public double amp=0;
    public int ss=0;
    private static final int HISTORY_SIZE = 500;           // number of points to
plot in history
    private XYPlot aprHistoryPlot = null;
    private CheckBox hwAcceleratedCb;
    private SimpleXYSeries azimuthHistorySeries = null;
    public Button bPause;
    int pause=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        PreferenceManager.setDefaultValues(this, R.xml.settings_activity,
false);

        if (mHandler == null) mHandler = new BluetoothResponseHandler(this);
        else mHandler.setTarget(this);

        MSG_NOT_CONNECTED = getString(R.string.msg_not_connected);
        MSG_CONNECTING = getString(R.string.msg_connecting);
        MSG_CONNECTED = getString(R.string.msg_connected);

        setContentView(R.layout.activity_electromiografo);
        bPause = (Button) findViewById(R.id.EMIP);

        bPause.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View view, MotionEvent event) {
                if(event.getAction() == MotionEvent.ACTION_DOWN) {

```

```

        pause=1;
    } else if (event.getAction() == MotionEvent.ACTION_UP) {
        pause=0;
    }
    return false;
}
});

// setup the APR History plot:;
aprHistoryPlot = (XYPlot) findViewById(R.id.Gráfica);
azimuthHistorySeries = new SimpleXYSeries("Senal");
azimuthHistorySeries.useImplicitXVals();
aprHistoryPlot.setRangeBoundaries(0, 1023, BoundaryMode.FIXED);
aprHistoryPlot.setDomainBoundaries(0, HISTORY_SIZE,
BoundaryMode.FIXED);
aprHistoryPlot.addSeries(azimuthHistorySeries, new
LineAndPointFormatter(Color.rgb(100, 0, 0),0x000000,0x000000,null));
aprHistoryPlot.setDomainStepValue(10);
aprHistoryPlot.setTicksPerRangeLabel(3);
aprHistoryPlot.setDomainLabel("Tiempo");
aprHistoryPlot.getDomainLabelWidget().pack();
aprHistoryPlot.setRangeLabel("Amplitud");
aprHistoryPlot.getRangeLabelWidget().pack();

// setup checkboxes:
hwAcceleratedCb = (CheckBox) findViewById(R.id.hwAccelerationCb);
final PlotStatistics levelStats = new PlotStatistics(1000, false);
final PlotStatistics histStats = new PlotStatistics(1000, false);

aprHistoryPlot.addListener(histStats);
hwAcceleratedCb.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton,
boolean b) {
        if (b) {
            //aprHistoryPlot.setLayerType(View.LAYER_TYPE_NONE, null);
        } else {

//aprHistoryPlot.setLayerType(View.LAYER_TYPE_SOFTWARE, null);
        }
    }
});
}
}
}

```

```

//
=====
===

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putString(DEVICE_NAME, deviceName);

}
//
=====
===

private boolean isConnected() {
    return (connector != null) && (connector.getState() ==
DeviceConnector.STATE_CONNECTED);
}
//
=====
===

private void stopConnection() {
    if (connector != null) {
        connector.stop();
        connector = null;
        deviceName = null;
    }
}
//
=====
===

@Override
public void onStart() {
    super.onStart();

    // hex mode
    final String mode = Utils.getPreference(this,
getString(R.string.pref_commands_mode));
    this.hexMode = mode.equals("HEX");
    if (hexMode) {
    } else {
    }
}

```

```

        this.command_ending = getCommandEnding();

        this.needClean = Utils.getBooleanPreference(this,
getString(R.string.pref_need_clean));

        BluetoothDevice device =
btAdapter.getRemoteDevice("98:D3:31:20:35:19");
        if (super.isAdapterReady() && (connector == null))
setupConnector(device);

    }
    //
=====
===

    @Override
    public void onDestroy(){
        super.onDestroy();
        stopConnection();
    }

    private String getCommandEnding() {
        String result = Utils.getPreference(this,
getString(R.string.pref_commands_ending));
        if (result.equals("\r\n")) result = "\r\n";
        else if (result.equals("\n")) result = "\n";
        else if (result.equals("\r")) result = "\r";
        else result = "";
        return result;
    }
    //
=====
===

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {
            case REQUEST_CONNECT_DEVICE:
                // When DeviceListActivity returns with a device to connect
                if (resultCode == Activity.RESULT_OK) {
                    String address =
data.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
                    BluetoothDevice device = btAdapter.getRemoteDevice(address);

```



```

        azimuthHistorySeries.addLast(null, amp);
    }
    aprHistoryPlot.redraw();
}
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
}
}
}
//

```

```

=====
===

```

```

void setDeviceName(String deviceName) {
    this.deviceName = deviceName;
    getSupportActionBar().setSubtitle(deviceName);
}
//

```

```

=====
===

```

```

private static class BluetoothResponseHandler extends Handler {
    private WeakReference<Electromiografo> mActivity;

    public BluetoothResponseHandler(Electromiografo activity) {
        mActivity = new WeakReference<Electromiografo>(activity);
    }

    public void setTarget(Electromiografo target) {
        mActivity.clear();
        mActivity = new WeakReference<Electromiografo>(target);
    }
    @Override
    public void handleMessage(Message msg) {
        Electromiografo activity = mActivity.get();
        if (activity != null) {
            switch (msg.what) {
                case MESSAGE_STATE_CHANGE:

                    Utils.log("MESSAGE_STATE_CHANGE: " + msg.arg1);

                    final ActionBar bar = activity.getSupportActionBar();
                    switch (msg.arg1) {

```

```

        case DeviceConnector.STATE_CONNECTED:
            bar.setSubtitle(MSG_CONNECTED);
            break;
        case DeviceConnector.STATE_CONNECTING:
            bar.setSubtitle(MSG_CONNECTING);
            break;
        case DeviceConnector.STATE_NONE:
            bar.setSubtitle(MSG_NOT_CONNECTED);
            break;
    }
    break;

    case MESSAGE_READ:
        final String readMessage = (String) msg.obj;
        if (readMessage != null) {
            activity.appendLog(readMessage);
        }
        break;
    case MESSAGE_DEVICE_NAME:
        activity.setDeviceName((String) msg.obj);
        break;

    case MESSAGE_WRITE:
        // stub
        break;

    case MESSAGE_TOAST:
        // stub
        break;
    }
}

static void show(Context context) {
    final Intent intent = new Intent(context, Electromiografo.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP
Intent.FLAG_ACTIVITY_NO_HISTORY);
    context.startActivity(intent);
}

=====
===
}

```

CLASE ELECTROESTIMULADOR

```
package ru.sash0k.bluetooth_terminal;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.preference.PreferenceManager;
import android.text.InputFilter;
import android.text.InputType;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.actionbarsherlock.app.ActionBar;

import java.lang.ref.WeakReference;

import ru.sash0k.bluetooth_terminal.bluetooth.DeviceConnector;
import ru.sash0k.bluetooth_terminal.bluetooth.DeviceListActivity;

public final class Electroestimulador extends BaseActivity {

    private static final String DEVICE_NAME = "DEVICE_NAME";
    private static final String LOG = "LOG";

    private static String MSG_NOT_CONNECTED;
    private static String MSG_CONNECTING;
    private static String MSG_CONNECTED;

    private static DeviceConnector connector;
    private static BluetoothResponseHandler mHandler;

    Button bBiceps, bMuslos, bFlexores, bPerone, bSubirN, bBajarN, bSubirT,
    bBajarT, bDetenerN, bDetenerT;
    private boolean hexMode;
    private String command_ending;
    private String deviceName;
    TextView tNivel, tTiempo;
```

```

int nivel=2;
int tiempo=15;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    PreferenceManager.setDefaultValues(this, R.xml.settings_activity,
false);

    if (mHandler == null) mHandler = new BluetoothResponseHandler(this);
    else mHandler.setTarget2(this);

    MSG_NOT_CONNECTED = getString(R.string.msg_not_connected);
    MSG_CONNECTING = getString(R.string.msg_connecting);
    MSG_CONNECTED = getString(R.string.msg_connected);

    setContentView(R.layout.activity_electroestimulador);
    if (isConnected() && (savedInstanceState != null)) {
        setDeviceName(savedInstanceState.getString(DEVICE_NAME));
    } else getSupportActionBar().setSubtitle(MSG_NOT_CONNECTED);

    bPerone = (Button) findViewById(R.id.BPERONE);
    bFlexores = (Button) findViewById(R.id.BFLEXORES);
    bBiceps = (Button) findViewById(R.id.BBICEPS);
    bMuslos = (Button) findViewById(R.id.BMUSLOS);
    bSubirN = (Button) findViewById(R.id.BSUBIRN);
    bSubirT = (Button) findViewById(R.id.BSUBIRT);
    bBajarN = (Button) findViewById(R.id.BBAJARN);
    bBajarT = (Button) findViewById(R.id.BBAJART);
    bDetenerN = (Button) findViewById(R.id.BDETN);
    bDetenerT = (Button) findViewById(R.id.BDETT);
    tTiempo = (TextView) findViewById(R.id.TTIEMPO);
    tNivel = (TextView) findViewById(R.id.TNIVEL);

    bPerone.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            sendString("P");
        }
    });
    bFlexores.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            sendString("F");
        }
    });
    bBiceps.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            sendString("B");
        }
    });

```

```

    }
});
bMuslos.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendString("M");
    }
});

bSubirN.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        nivel = nivel+1;
        if(nivel>3)
            nivel=3;
        sendString("N"+Integer.toString(nivel));
        tNivel.setText("Nivel: " + Integer.toString(nivel));
    }
});

bSubirT.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        tiempo = tiempo+5;
        if(tiempo>30)
            tiempo=30;
        sendString("T"+Integer.toString(tiempo));
        tTiempo.setText("Tiempo: " + Integer.toString(tiempo));
    }
});

bBajarN.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        nivel = nivel-1;
        if(nivel<0)
            nivel=0;
        sendString("N"+Integer.toString(nivel));
        tNivel.setText("Nivel: " + Integer.toString(nivel));
    }
});

bBajarT.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        tiempo = tiempo-5;
        if(tiempo<0)
            tiempo=0;
        sendString("T"+Integer.toString(tiempo));
        tTiempo.setText("Tiempo: " + Integer.toString(tiempo));
    }
});
bDetenerN.setOnClickListener(new View.OnClickListener() {

```

```

        public void onClick(View v) {
            nivel = 0;
            sendString("N"+Integer.toString(nivel));
            tNivel.setText("Nivel: " + Integer.toString(nivel));
        }
    });
    bDetenerT.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            tiempo = 0;
            sendString("T"+Integer.toString(tiempo));
            tTiempo.setText("Tiempo: " + Integer.toString(tiempo));
        }
    });
}
//
=====
=====

```

```

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putString(DEVICE_NAME, deviceName);
}
//
=====
=====

```

```

private boolean isConnected() {
    return (connector != null) && (connector.getState() ==
DeviceConnector.STATE_CONNECTED);
}
//
=====
=====

```

```

private void stopConnection() {
    if (connector != null) {
        connector.stop();
        connector = null;
        deviceName = null;
    }
}
}

```

```

//
=====
=====

@Override
public void onStart() {
    super.onStart();

    // hex mode
    final String mode = Utils.getPrefence(this,
getString(R.string.pref_commands_mode));
    this.hexMode = mode.equals("HEX");
    if (hexMode) {
    } else {
    }

    this.command_ending = getCommandEnding();

    BluetoothDevice device =
btAdapter.getRemoteDevice("98:D3:31:20:35:18");
    if (super.isAdapterReady() && (connector == null))
setupConnector(device);

}
//
=====
=====

@Override
public void onDestroy(){
    super.onDestroy();
    stopConnection();
}

private String getCommandEnding() {
    String result = Utils.getPrefence(this,
getString(R.string.pref_commands_ending));
    if (result.equals("\r\n")) result = "\r\n";
    else if (result.equals("\n")) result = "\n";
    else if (result.equals("\r")) result = "\r";
    else result = "";
    return result;
}

```

```

//
=====
=====

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case REQUEST_CONNECT_DEVICE:
            // When DeviceListActivity returns with a device to connect
            if (resultCode == Activity.RESULT_OK) {
                String address =
data.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
                BluetoothDevice device = btAdapter.getRemoteDevice(address);
                if (super.isAdapterReady() && (connector == null))
setupConnector(device);
            }
            break;
        case REQUEST_ENABLE_BT:
            // When the request to enable Bluetooth returns
            super.pendingRequestEnableBt = false;
            if (resultCode != Activity.RESULT_OK) {
                Utils.log("BT not enabled");
            }
            break;
    }
}
//
=====
=====

private void setupConnector(BluetoothDevice connectedDevice) {
    stopConnection();
    try {
        String emptyName = getString(R.string.empty_device_name);
        DeviceData data = new DeviceData(connectedDevice, emptyName);
        connector = new DeviceConnector(data, mHandler);
        connector.connect();
    } catch (IllegalArgumentException e) {
        Utils.log("setupConnector failed: " + e.getMessage());
    }
}
//
=====
=====

```

```

public void sendString(String mm) {
    if (mm != null) {
        String commandString = mm;
        if (commandString.isEmpty()) return;

        if (hexMode && (commandString.length() % 2 == 1)) {
            commandString = "0" + commandString;
        }
        byte[] command = (hexMode ? Utils.toHex(commandString) :
commandString.getBytes());
        if (command_ending != null) command = Utils.concat(command,
command_ending.getBytes());
        if (isConnected()) {
            connector.write(command);
            //appendLog(commandString);
        }
    }
}
//

```

```

=====
=====

```

```

void appendLog(String message) {

    //textlN.setText(message);

}
//

```

```

=====
=====

```

```

void setDeviceName(String deviceName) {
    this.deviceName = deviceName;
    getSupportActionBar().setSubtitle(deviceName);
}
//

```

```

=====
=====

```

```

/**
 * ?????????? ?????? ?????? ?? bluetooth-??????
 */

```

```

private static class BluetoothResponseHandler extends Handler {
    private WeakReference<Electroestimulador> mActivity;

    public BluetoothResponseHandler(Electroestimulador activity) {
        mActivity = new WeakReference<Electroestimulador>(activity);
    }
}

```

```

}

public void setTarget2(Electroestimulador target2) {
    mActivity.clear();
    mActivity = new WeakReference<Electroestimulador>(target2);
}

```

@Override

```

public void handleMessage(Message msg) {
    Electroestimulador activity = mActivity.get();
    if (activity != null) {
        switch (msg.what) {
            case MESSAGE_STATE_CHANGE:

                Utils.log("MESSAGE_STATE_CHANGE: " + msg.arg1);
                final ActionBar bar = activity.getSupportActionBar();
                switch (msg.arg1) {
                    case DeviceConnector.STATE_CONNECTED:
                        bar.setSubtitle(MSG_CONNECTED);
                        break;
                    case DeviceConnector.STATE_CONNECTING:
                        bar.setSubtitle(MSG_CONNECTING);
                        break;
                    case DeviceConnector.STATE_NONE:
                        bar.setSubtitle(MSG_NOT_CONNECTED);
                        break;
                }
                break;

            case MESSAGE_READ:
                final String readMessage = (String) msg.obj;
                if (readMessage != null) {
                    activity.appendLog(readMessage);
                }
                break;

            case MESSAGE_DEVICE_NAME:
                activity.setDeviceName((String) msg.obj);
                break;

            case MESSAGE_WRITE:
                // stub
                break;

            case MESSAGE_TOAST:
                // stub
                break;
        }
    }
}

```

```

    }
    }
}

static void show(Context context) {
    final Intent intent = new Intent(context, Electroestimulador.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP
Intent.FLAG_ACTIVITY_NO_HISTORY);
    context.startActivity(intent);
}
//
=====
===
}

```

PROGRAMACIÓN LAYOUT'S

Activity_main.xml

```

RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffd9d8dc"
    tools:context="ru.sash0k.bluetooth_terminal.MainActivity">

```

```

<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true">

```

```

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="960dp"
    android:layout_height="100dp"
    android:layout_column="0">

```

```

<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

```

```

<TableRow
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <LinearLayout
    android:orientation="horizontal"
    android:layout_width="200dp"
    android:layout_height="100dp"
    android:layout_column="0">

    <FrameLayout
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">

      <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/imageView"
        android:src="@drawable/espe"
        android:layout_gravity="center" />
      </FrameLayout>
    </LinearLayout>

    <LinearLayout
      android:orientation="horizontal"
      android:layout_width="760dp"
      android:layout_height="100dp"
      android:layout_column="1">

      <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

        <TableRow
          android:layout_width="fill_parent"
          android:layout_height="fill_parent">

          <LinearLayout
            android:orientation="horizontal"
            android:layout_width="760dp"
            android:layout_height="50dp"
            android:layout_column="0">

            <TextView
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:textAppearance="?android:attr/textAppearanceLarge"

```

```

ARMADAS
        android:text="UNIVESIDAD DE LAS FUERZAS
        "
        android:id="@+id/textView"
        android:textColor="#ff000000"
        android:gravity="center|bottom" />
</LinearLayout>
</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:layout_column="0"
        android:gravity="center">

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"

            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="ESPE - EXTENSIÓN LATACUNGA
            "

            android:id="@+id/textView2"
            android:textColor="#ff000000"
            android:gravity="center|top" />
        </LinearLayout>
    </TableRow>
</TableLayout>
</LinearLayout>
</TableRow>
</TableLayout>
</LinearLayout>
</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="430dp"
        android:layout_column="0">

```

```

<TableLayout
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
      android:orientation="horizontal"
      android:layout_width="960dp"
      android:layout_height="50dp"
      android:layout_column="0">

      <TextView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="SISTEMA"
        android:id="@+id/textView3"
        android:textColor="#ff000000"
        android:gravity="center|bottom" />
      </LinearLayout>
    </TableRow>

    <TableRow
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">

      <LinearLayout
        android:orientation="horizontal"
        android:layout_width="960dp"
        android:layout_height="50dp"
        android:layout_column="0">

        <TextView
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"

          android:textAppearance="?android:attr/textAppearanceLarge"
          android:text="MÉDICO"
          android:id="@+id/textView4"
          android:textColor="#ff000000"
          android:gravity="center|top" />
        </LinearLayout>
      </TableRow>

```

```

<TableRow
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="330dp"
    android:layout_column="0">

    <FrameLayout
      android:layout_width="700dp"
      android:layout_height="100dp"
      android:layout_gravity="center">

      <Button
        android:layout_width="200dp"
        android:layout_height="80dp"
        android:text="ELECTROMIÓGRAFO"
        android:id="@+id/EMIO"
        android:layout_gravity="left|center_vertical" />

      <Button
        android:layout_width="250dp"
        android:layout_height="80dp"
        android:text="ELECTROESTIMULADOR"
        android:id="@+id/ESTI"
        android:layout_gravity="right|center_vertical" />
    </FrameLayout>
  </FrameLayout>
</TableRow>
</TableLayout>
</LinearLayout>
</TableRow>
</TableLayout>

</RelativeLayout>

```

Activity_electromiografo.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:background="#ffd9d8dc">

  <LinearLayout

```

```
android:orientation="horizontal"
android:layout_width="960dp"
android:layout_height="100dp"
android:id="@+id/ad">
```

```
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="200dp"
    android:layout_height="100dp"
    android:layout_column="0">
```

```
<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<ImageView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/imageView"
    android:src="@drawable/espe"
    android:layout_gravity="center" />
</FrameLayout>
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="760dp"
    android:layout_height="100dp"
    android:layout_column="1">
```

```
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<LinearLayout
```

```

        android:orientation="horizontal"
        android:layout_width="760dp"
        android:layout_height="50dp"
        android:layout_column="0">

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"

android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="UNIVESIDAD DE LAS FUERZAS
ARMADAS
            "
            android:id="@+id/textView"
            android:textColor="#ff000000"
            android:gravity="center|bottom" />
        </LinearLayout>
    </TableRow>

    <TableRow
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="50dp"
            android:layout_column="0"
            android:gravity="center">

            <TextView
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"

android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="ESPE - EXTENSIÓN LATACUNGA
                "
                android:id="@+id/textView2"
                android:textColor="#ff000000"
                android:gravity="center|top" />
            </LinearLayout>
        </TableRow>
    </TableLayout>
</LinearLayout>

```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="960dp"
  android:layout_height="20dp"></LinearLayout>
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="fill_parent"
  android:layout_height="270dp"
  android:layout_alignParentTop="true"
  android:layout_alignParentLeft="true"
  android:id="@+id/L3">
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="980dp"
  android:layout_height="fill_parent"
  android:id="@+id/L5">
```

```
<TableLayout
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
```

```
<TableRow
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="750dp"
  android:layout_height="wrap_content"
  android:layout_column="0"
  android:id="@+id/l">
```

```
<com.androidplot.xy.XYPlot
  android:id="@+id/Gráfica"
  android:layout_width="fill_parent"
  android:layout_height="270dp"
  androidPlot.title="Electromiógrafo"
  androidPlot.domainLabel="Tiempo"
  androidPlot.rangeLabel="Amplitud"
  androidPlot.graphWidget.marginTop="20dp"
  androidPlot.graphWidget.marginLeft="15dp"
  androidPlot.graphWidget.marginBottom="25dp"
  androidPlot.graphWidget.marginRight="10dp"
```

```
androidPlot.legendWidget.iconSizeMetrics.heightMetric.value="15dp"
```

```
androidPlot.legendWidget.iconSizeMetrics.widthMetric.value="15dp"  
    androidPlot.legendWidget.heightMetric.value="25dp"/>
```

```
<LinearLayout android:orientation="horizontal"  
    android:gravity="center"  
    android:layout_width="50dp"  
    android:layout_height="50dp">  
    <CheckBox android:id="@+id/hwAccelerationCb"  
        android:visibility="gone"  
        android:text="HW Acceleration"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<TableLayout  
    android:layout_width="200dp"  
    android:layout_height="fill_parent"  
    android:layout_column="1">
```

```
<TableRow  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">
```

```
</TableRow>
```

```
<TableRow  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">
```

```
<LinearLayout  
    android:orientation="horizontal"  
    android:layout_width="fill_parent"  
    android:layout_height="50dp">
```

```
</LinearLayout>
```

```
</TableRow>
```

```
<TableRow  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">
```

```
</TableRow>
```

```

        <TableRow
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">

            </TableRow>
        </TableLayout>
    </TableRow>
</TableLayout>
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="75dp">

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="30dp"
                android:layout_height="75dp"
                android:layout_column="0"></LinearLayout>

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="200dp"
                android:layout_height="75dp"
                android:layout_column="2" >

                <FrameLayout
                    android:layout_width="fill_parent"
                    android:layout_height="fill_parent">

                    <Button
                        android:layout_width="120dp"
                        android:layout_height="50dp"
                        android:text="DETENER"
                        android:id="@+id/EMIP"
                        android:layout_gravity="center" />
                </FrameLayout>
            </LinearLayout>

```

```

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="10dp"
        android:layout_height="75dp"
        android:layout_column="3" >

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="75dp"
    android:layout_height="75dp"
    android:layout_column="4" >

    <FrameLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

        </FrameLayout>
    </LinearLayout>

</TableRow>
</TableLayout>
</LinearLayout>
</LinearLayout>

```

Activity_electroestimulador.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffd9d8dc"
    tools:context="ru.sash0k.bluetooth_terminal.Electroestimulador">

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="960dp"
    android:layout_height="100dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:id="@+id/linearLayout2">

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

```

```
<TableRow
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <LinearLayout
    android:orientation="horizontal"
    android:layout_width="200dp"
    android:layout_height="100dp"
    android:layout_column="0">

    <FrameLayout
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">

      <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/imageView"
        android:src="@drawable/espe"
        android:layout_gravity="center" />
    </FrameLayout>
  </LinearLayout>

  <LinearLayout
    android:orientation="horizontal"
    android:layout_width="760dp"
    android:layout_height="100dp"
    android:layout_column="1">

    <TableLayout
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">

      <TableRow
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

        <LinearLayout
          android:orientation="horizontal"
          android:layout_width="760dp"
          android:layout_height="50dp"
          android:layout_column="0">

          <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
```

```

android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="UNIVESIDAD DE LAS FUERZAS
ARMADAS
        "
        android:id="@+id/textView"
        android:textColor="#ff000000"
        android:gravity="center|bottom" />
</LinearLayout>
</TableRow>

```

```

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

```

```

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:layout_column="0"
        android:gravity="center">

```

```

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"

```

```

android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="ESPE - EXTENSIÓN LATACUNGA
"
        android:id="@+id/textView2"
        android:textColor="#ff000000"
        android:gravity="center|top" />
</LinearLayout>
</TableRow>
</TableLayout>
</LinearLayout>
</TableRow>
</TableLayout>

```

```

</LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="960dp"
    android:layout_height="20dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/linearLayout2"

```

```
android:id="@+id/linearLayout3"></LinearLayout>
```

```
<LinearLayout  
  android:orientation="horizontal"  
  android:layout_width="960dp"  
  android:layout_height="500dp"  
  android:layout_below="@+id/linearLayout3">
```

```
<TableLayout  
  android:layout_width="match_parent"  
  android:layout_height="match_parent">
```

```
<TableRow  
  android:layout_width="match_parent"  
  android:layout_height="match_parent">
```

```
<LinearLayout  
  android:orientation="horizontal"  
  android:layout_width="500dp"  
  android:layout_height="50dp"  
  android:layout_column="0">
```

```
<TextView  
  android:layout_width="fill_parent"  
  android:layout_height="fill_parent"  
  android:textAppearance="?android:attr/textAppearanceLarge"  
  android:text="GRUPOS MUSCULARES"  
  android:id="@+id/textView5"  
  android:textColor="#ff000000"  
  android:gravity="center" />
```

```
</LinearLayout>
```

```
<LinearLayout  
  android:orientation="horizontal"  
  android:layout_width="30dp"  
  android:layout_height="fill_parent"  
  android:layout_column="1"></LinearLayout>
```

```
<TextView  
  android:layout_width="100dp"  
  android:layout_height="fill_parent"  
  android:textAppearance="?android:attr/textAppearanceLarge"  
  android:text="DOSIS"  
  android:id="@+id/textView6"  
  android:textColor="#ff000000"  
  android:gravity="center"  
  android:layout_column="2" />
```

```
<LinearLayout
  android:orientation="horizontal"
  android:layout_width="30dp"
  android:layout_height="fill_parent"
  android:layout_column="3" />
```

```
<TextView
  android:layout_width="100dp"
  android:layout_height="fill_parent"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:text="SESIÓN"
  android:id="@+id/textView8"
  android:textColor="#ff000000"
  android:gravity="center"
  android:layout_column="4" />
```

```
</TableRow>
```

```
<TableRow
  android:layout_width="match_parent"
  android:layout_height="match_parent">
```

```
<FrameLayout
  android:layout_width="fill_parent"
  android:layout_height="120dp"
  android:layout_column="0">
```

```
<FrameLayout
  android:layout_width="230dp"
  android:layout_height="120dp"
  android:layout_gravity="left|center_vertical">
```

```
<ImageView
  android:layout_width="100dp"
  android:layout_height="wrap_content"
  android:id="@+id/imageView2"
  android:src="@drawable/zona1"
  android:layout_gravity="left|center_vertical"
  android:layout_column="1" />
```

```
<Button
  android:layout_width="120dp"
  android:layout_height="50dp"
  android:text="PERONE"
  android:id="@+id/BPERONE"
  android:layout_below="@+id/BMUSLOS"
  android:layout_alignParentLeft="true"
  android:layout_column="2"
```

```

        android:layout_gravity="right|center_vertical" />
</FrameLayout>

<FrameLayout
    android:layout_width="230dp"
    android:layout_height="120dp"
    android:layout_gravity="right|center_vertical" >

    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/imageView3"
        android:src="@drawable/zona3"
        android:layout_gravity="left|center_vertical" />

    <Button
        android:layout_width="120dp"
        android:layout_height="50dp"
        android:text="BICEPS"
        android:id="@+id/BBICEPS"
        android:layout_alignParentLeft="true"
        android:layout_column="1"
        android:layout_gravity="right|center_vertical" />
</FrameLayout>
</FrameLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="1"></LinearLayout>

<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="2">

    <TextView
        android:layout_width="120dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nivel: 2"
        android:id="@+id/TNIVEL"
        android:textColor="#ff000000"
        android:gravity="center"
        android:layout_column="2"
        android:layout_gravity="left|center_vertical" />
</FrameLayout>

```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="3" />
```

```
<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="4" >
```

```
<TextView
    android:layout_width="120dp"
    android:layout_height="50dp"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Tiempo: 15"
    android:id="@+id/TTIEMPO"
    android:textColor="#ff000000"
    android:gravity="center"
    android:layout_column="2"
    android:layout_gravity="left|center_vertical" />
</FrameLayout>
```

```
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="120dp"
    android:layout_column="0">
```

```
<FrameLayout
    android:layout_width="230dp"
    android:layout_height="120dp"
    android:layout_gravity="left|center_vertical" >
```

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/imageView6"
    android:src="@drawable/zona4"
    android:layout_gravity="left|center_vertical"
    android:layout_column="1" />
```

```

<Button
    android:layout_width="120dp"
    android:layout_height="50dp"
    android:text="FLEXORES"
    android:id="@+id/BFLEXORES"
    android:layout_alignParentLeft="true"
    android:layout_column="1"
    android:layout_gravity="right|center_vertical" />

</FrameLayout>

<FrameLayout
    android:layout_width="230dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right|center_vertical" >

    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/imageView7"
        android:src="@drawable/zona5"
        android:layout_gravity="left|center_vertical"
        android:layout_column="1" />

    <Button
        android:layout_width="120dp"
        android:layout_height="50dp"
        android:text="MUSLOS"
        android:id="@+id/BMUSLOS"
        android:layout_below="@+id/BMUSLOS"
        android:layout_alignParentLeft="true"
        android:layout_column="2"
        android:layout_gravity="right|center_vertical" />
</FrameLayout>
</FrameLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="1"></LinearLayout>

<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="2" >

    <Button

```

```
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:id="@+id/BBAJARN"
    android:layout_gravity="center_horizontal|bottom"
    android:background="@drawable/abajo" />
```

```
<Button
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:id="@+id/BSUBIRN"
    android:layout_gravity="center_horizontal|top"
    android:background="@drawable/arriba" />
```

```
<Button
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:id="@+id/BDETN"
    android:layout_gravity="right|center_vertical"
    android:background="#fff0010" />
```

```
</FrameLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="3" />
```

```
<FrameLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_column="4" >
```

```
<Button
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:id="@+id/BBAJART"
    android:layout_gravity="center_horizontal|bottom"
    android:background="@drawable/abajo" />
```

```
<Button
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:id="@+id/BSUBIRT"
    android:layout_gravity="center_horizontal|top"
    android:background="@drawable/arriba" />
```

```
<Button
    android:layout_width="30dp"
```

```
        android:layout_height="30dp"  
        android:id="@+id/BDETT"  
        android:layout_gravity="right|center_vertical"  
        android:background="#ffff0500" />  
    </FrameLayout>
```

```
</TableRow>
```

```
</TableLayout>  
</LinearLayout>
```

```
</RelativeLayout>
```

ANEXO I: PROGRAMACIÓN EN ARDUINO UNO

```
#include <compat/deprecated.h>
#include <FlexiTimer2.h>
//http://www.arduino.cc/playground/Main/FlexiTimer2

// All definitions
#define NUMCHANNELS 6
#define HEADERLEN 4
#define PACKETLEN (NUMCHANNELS * 2 + HEADERLEN + 1)
#define SAMPFREQ 256 // ADC sampling rate 256
#define TIMER2VAL (1024/(SAMPFREQ)) // Set 256Hz sampling
frequency
#define LED1 13
#define CAL_SIG 9

// Global constants and variables
volatile unsigned char TXBuf[PACKETLEN]; //The transmission packet
volatile unsigned char TXBufEMG[25]; //The transmission packet
volatile unsigned char TXIndex; //Next byte to write in the transmission
packet.
volatile unsigned char TXIndexEMG; //Next byte to write in the
transmission packet.
volatile unsigned char CurrentCh; //Current channel being sampled.
volatile unsigned char counter = 0; //Additional divider used to generate
CAL_SIG
volatile unsigned int ADC_Value = 0; //ADC current value

//~~~~~
// Functions
//~~~~~

/*****/
/* Function name: Toggle_LED1 */
/* Parameters */
/* Input : No */
/* Output : No */
/* Action: Switches-over LED1. */
/*****/
void Toggle_LED1(void){

if((digitalRead(LED1))==HIGH){ digitalWrite(LED1,LOW); }
else{ digitalWrite(LED1,HIGH); }

}

/*****/
```

```

/* Function name: toggle_GAL_SIG          */
/* Parameters                               */
/* Input  : No                             */
/* Output : No                             */
/* Action: Switches-over GAL_SIG.         */
/*****/
void toggle_GAL_SIG(void){

if(digitalRead(CAL_SIG) == HIGH){ digitalWrite(CAL_SIG, LOW); }
else{ digitalWrite(CAL_SIG, HIGH); }

}

/*****/
/* Function name: setup                    */
/* Parameters                               */
/* Input  : No                             */
/* Output : No                             */
/* Action: Initializes all peripherals     */
/*****/
void setup() {

noInterrupts(); // Disable all interrupts before initialization

// LED1
pinMode(LED1, OUTPUT); //Setup LED1 direction
digitalWrite(LED1,LOW); //Setup LED1 state
pinMode(CAL_SIG, OUTPUT);

//Write packet header and footer
TXBuf[0] = 0xa5; //Sync 0
TXBuf[1] = 0x5a; //Sync 1
TXBuf[2] = 2; //Protocol version
TXBuf[3] = 0; //Packet counter
TXBuf[4] = 0x02; //CH1 High Byte
TXBuf[5] = 0x00; //CH1 Low Byte
TXBuf[6] = 0x02; //CH2 High Byte
TXBuf[7] = 0x00; //CH2 Low Byte
TXBuf[8] = 0x02; //CH3 High Byte
TXBuf[9] = 0x00; //CH3 Low Byte
TXBuf[10] = 0x02; //CH4 High Byte
TXBuf[11] = 0x00; //CH4 Low Byte
TXBuf[12] = 0x02; //CH5 High Byte
TXBuf[13] = 0x00; //CH5 Low Byte
TXBuf[14] = 0x02; //CH6 High Byte
TXBuf[15] = 0x00; //CH6 Low Byte
TXBuf[2 * NUMCHANNELS + HEADERLEN] = 0x01; // Switches state

```

```

TXIndexEMG=1;
TXBufEMG[0] =20;
TXBufEMG[23] =13;
TXBufEMG[24] =10;
// Timer2
// Timer2 is used to setup the analog channels sampling frequency and
packet update.
// Whenever interrupt occurs, the current read packet is sent to the PC
// In addition the CAL_SIG is generated as well, so Timer1 is not required in
this case!
FlexiTimer2::set(TIMER2VAL, Timer2_Overflow_ISR);
FlexiTimer2::start();

// Serial Port
Serial.begin(57600);
//Set speed to 57600 bps

// MCU sleep mode = idle.
//outb(MCUCR,(inp(MCUCR) | (1<<SE)) & ~(1<<SM0) | ~(1<<SM1) |
~(1<<SM2));

interrupts(); // Enable all interrupts after initialization has been completed
}

/*****/
/* Function name: Timer2_Overflow_ISR          */
/* Parameters                                */
/* Input   : No                               */
/* Output  : No                               */
/* Action: Determines ADC sampling frequency. */
/*****/
void Timer2_Overflow_ISR()
{
  // Toggle LED1 with ADC sampling frequency /2
  Toggle_LED1();

  //Read the 6 ADC inputs and store current values in Packet

  for(CurrentCh=0;CurrentCh<6;CurrentCh++){
    ADC_Value = analogRead(CurrentCh);
    TXBuf[((2*CurrentCh) + HEADERLEN)] = ((unsigned char)((ADC_Value &
0xFF00) >> 8)); // Write High Byte
    TXBuf[((2*CurrentCh) + HEADERLEN + 1)] = ((unsigned
char)(ADC_Value & 0x00FF)); // Write Low Byte
  }

  if(TXIndexEMG<22){

```

```

TXBufEMG[TXIndexEMG] =TXBuf[5]/2;
TXBufEMG[TXIndexEMG+1] =TXBuf[4];
if(TXIndexEMG==21){
for(TXIndexEMG=0;TXIndexEMG<25;TXIndexEMG++){
  Serial.write(TXBufEMG[TXIndexEMG]);
}
TXIndexEMG=1;
}else{
  TXIndexEMG=TXIndexEMG+2;
}
}

// Increment the packet counter
TXBuf[3]++;

// Generate the CAL_SIGnal
counter++; // increment the divider counter
if(counter == 12){ // 250/12/2 = 10.4Hz ->Toggle frequency
  counter = 0;
  toggle_GAL_SIG(); // Generate CAL signal with frequ ~10Hz
}
}

/*****/
/* Function name: loop */
/* Parameters */
/* Input : No */
/* Output : No */
/* Action: Puts MCU into sleep mode. */
/*****/
void loop() {

  __asm__ __volatile__ ("sleep");

}

```

ANEXO J: PROGRAMACION EN BASCOM - AVR

```
$regfile = "m328pdef.dat"  
$crystal = 8000000  
$hwstack = 40  
$swstack = 16  
$framesize = 32  
$baud = 9600
```

```
'niveles  
'N0 abierto  
'N1 470r  
'N2 330r  
'N3
```

```
'funciones  
'P perone  
'F flexores  
'B bicep  
'M muslos
```

```
'tiempo  
'T0  
'T5  
'T10  
'T15  
'T20  
'T25  
'T30  
Dim T1 As Byte  
Dim T2 As Byte
```

```
Dim Tiempo As Byte  
Dim Us As Byte  
Dim Ds As Byte  
Dim Um As Byte  
Dim Dm As Byte
```

```
Dim Nivel As Byte  
Dim Funcion As Byte
```

```
Dim Inicio As Byte
```

```
Dim S As Byte  
Dim A As Word  
Dim Segundo As Byte  
Dim Segundos As Byte  
Dim Dato As String * 10
```

Config Serialin = Buffered , Size = 20

'Config Adc = Single , Prescaler = Auto , Reference = Avcc
'Start Adc

Config Timer0 = Timer , Prescale = 256
On Timer0 Titilar
Enable Timer0

Config Timer2 = Timer , Prescale = 8 , Compare A = Toggle , Clear Timer = 1
Compare2a = 66

'Ddr IN OUT 1 OUT 0 ENTRADA
'Port Out
'Pin IN

Ddrb.0 = 1
Led Alias Portb.0
Dim B As Byte

Ddrd.2 = 1
Ddrd.3 = 1
Ddrd.4 = 1

G1 Alias Portd.2
G2 Alias Portd.3
G3 Alias Portd.4

Enable Interrupts

Ddrb.1 = 1
Ddrb.2 = 1
Reset Portb.1
Reset Portb.2

Do

```
If Segundos > 125 Then
    Print Dm ; Um ; ":" ; Ds ; Us           ; " T1= " ; T1 ; " T2=" ; T2
    Segundos = 0
End If
S = Inkey()
If S = "N" Then
    Input Dato Noecho
    Nivel = Val(dato)
    Print Nivel
End If
```

```
If S = "T" Then
Input Dato Noeche
If Dato = "0" Then
  Us = 0
  Ds = 0
  Um = 0
  Dm = 0
End If
```

```
If Dato = "5" Then
  Us = 0
  Ds = 0
  Um = 5
  Dm = 0
End If
```

```
If Dato = "10" Then
  Us = 0
  Ds = 0
  Um = 0
  Dm = 1
End If
```

```
If Dato = "15" Then
  Us = 0
  Ds = 0
  Um = 5
  Dm = 1
End If
```

```
If Dato = "20" Then
  Us = 0
  Ds = 0
  Um = 0
  Dm = 2
End If
```

```
If Dato = "25" Then
  Us = 0
  Ds = 0
  Um = 5
  Dm = 2
End If
```

```
If Dato = "30" Then
  Us = 0
  Ds = 0
  Um = 0
```

```
    Dm = 2
End If
Tiempo = Val(dato)
Print Tiempo
End If
```

```
If S = "F" Then
    Funcion = 1
    Inicio = 1
    T1 = 0
End If
```

```
If S = "M" Then
    Funcion = 2
    Inicio = 1
    T1 = 0
End If
```

```
If S = "P" Then
    Funcion = 3
    Inicio = 1
    T1 = 0
End If
```

```
If S = "B" Then
    Funcion = 4
    Inicio = 1
    T1 = 0
End If
```

```
If Nivel = 0 Or Tiempo = 0 Or Funcion = 0 Then
    Reset G1
    Reset G2
    Reset G3
    Reset Portb.1
    Reset Portb.2
    Inicio = 0
    Funcion = 0
```

```
Else
    If Funcion = 1 Then
        If T1 < 120 Then
            Set Portb.1
            Waitus 400
            Set Portb.2
            Waitus 50
            Reset Portb.1
            Reset Portb.2
            Waitms 170
        End If
        If T2 < 5 Then
            Set Portb.1
```

```
    Waitus 500
    Set Portb.2
    Waitus 100
    Reset Portb.1
    Reset Portb.2
    Waitms 14
Else
    Set Portb.1
    Waitus 400
    Set Portb.2
    Waitus 50
    Reset Portb.1
    Reset Portb.2
    Waitms 170
End If
End If
```

```
If Funcion = 2 Then
    If T1 < 120 Then
        Set Portb.1
        Waitus 500
        Set Portb.2
        Waitus 50
        Reset Portb.1
        Reset Portb.2
        Waitms 250
    End If
    If T2 < 5 Then
        Set Portb.1
        Waitus 1000
        Set Portb.2
        Waitus 100
        Reset Portb.1
        Reset Portb.2
        Waitms 14
    Else
        Set Portb.1
        Waitus 500
        Set Portb.2
        Waitus 50
        Reset Portb.1
        Reset Portb.2
        Waitms 250
    End If
End If
```

```
If Funcion = 3 Then
    If T1 < 120 Then
```

```
Set Portb.1
Waitus 252
Set Portb.2
Waitus 50
Reset Portb.1
Reset Portb.2
Waitms 160
End If
If T2 < 5 Then
Set Portb.1
Waitus 600
Set Portb.2
Waitus 100
Reset Portb.1
Reset Portb.2
Waitms 13
Else
Set Portb.1
Waitus 252
Set Portb.2
Waitus 50
Reset Portb.1
Reset Portb.2
Waitms 160
End If
End If
```

```
If Funcion = 4 Then
If T1 < 120 Then
Set Portb.1
Set Portb.2
Waitus 1000
'Set Portb.2
'Waitus 50
Reset Portb.1
Reset Portb.2
Waitms 160
End If
If T2 < 5 Then
Set Portb.1
Set Portb.2
Waitus 500
'Set Portb.2
'Waitus 50
Reset Portb.1
Reset Portb.2
Waitms 14
Else
```

```
Set Portb.1
Set Portb.2
Waitus 1000
'Set Portb.2
'Waitus 50
Reset Portb.1
Reset Portb.2
Waitms 160
End If
End If
End If
```

```
If Nivel = 3 Then
Set G1
Reset G2
Reset G3
End If
```

```
If Nivel = 2 Then
Set G2
Reset G1
Reset G3
End If
```

```
If Nivel = 1 Then
Set G3
Reset G1
Reset G2
End If
Loop
```

End

Titular:

```
Incr Segundo
Incr Segundos
If Segundo = 125 Then          '125
Segundo = 0
Toggle Led
.....
```

```
If Inicio = 1 Then
If T1 < 10 Then Incr T1
If T1 = 10 Then Incr T2
If T2 > 15 Then T2 = 0
Decr Us
If Us = 0 And Ds = 0 Then
If Um = 0 And Dm = 0 Then
Inicio = 0
```

```
        Funcion = 0
        Tiempo = 0
    End If
End If
If Us = 255 Then
    Us = 9
    Decr Ds
    If Ds = 255 Then
        Ds = 5
        Decr Um
        If Um = 255 Then
            Um = 9
            Decr Dm
        End If
    End If
End If
End If
End If
.....
End If
Return
```

ANEXO K: CERTIFICACIÓN MÉDICA



HOSPITAL "INGLES M & C"

EXCELENCIA EN MEDICINA ESPECIALIZADA

Quito, 25 de julio de 2015

CERTIFICADO MEDICO

Mediante la presente certifico haber utilizado y probado, un ELECTRO - ESTIMULADOR y ELECTRO - MIOGRAMA CON COMUNICACIÓN INALÁMBRICA, diseñado y desarrollado por Nelson Sinchiguano y Karina Tutillo, en mi consulta el día de hoy.

Los parámetros establecidos y normados por la OMS se cumplen a cabalidad, sin peligro para su utilización en humanos tanto en el Diagnóstico de enfermedades musculares como para el tratamiento en Rehabilitación y Fisiatría.

Adicionalmente, puedo comunicarle que se ha probado en mi consulta que ha acudido el día de hoy un total de 8 pacientes.

Atentamente,

Dr. Gino Manciatì J.
Médico - C.C.M.
L:20 F:88 N° 263

Dr. Gino Manciatì J.

Médico Tratante - CCM

ANEXO L: MANUAL DE USUARIO



CONTENIDO

Introducción	2
Que es un Electromiografía (EMG)?	2
Que es la Electroestimulación?	2
Posición de los electrodos	3
Cuando es hora de reemplazar los electrodos?	3
Medidas de seguridad	4
Características técnicas del instrumento	5
Preparando el instrumento	6
Electromiografía	7
Electroestimulador	8
Preguntas frecuentes y posibles fallos	11

Introducción

Qué es una Electromiografía (EMG)?

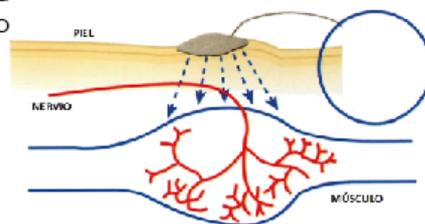
Es una técnica para la evaluación y registro de la actividad eléctrica producida por los músculos esqueléticos.

2

La EMG se desarrolla utilizando un instrumento médico llamado electromiógrafo, para producir un registro llamado electromiograma.

Qué es la electroestimulación?

La electroestimulación muscular produce una contracción visible del músculo mediante un estímulo eléctrico que se genera directamente sobre el músculo mediante un aparato adecuado y unos electrodos.



Posición de los electrodos

La posición de los electrodos depende del grupo muscular donde se requiera realizar la electromiografía o la electroestimulación.

Es importante que los electrodos estén bien fijados a los conectores de los cables; para ello presione con fuerza el conector sobre el electrodo hasta que se oiga un "click"

Reemplazo de los electrodos

Se recomienda que los electrodos deben ser reemplazados cuando las almohadillas autoadhesivas ya no se pegan bien o si usted comienza a sentir una sensación de ardor en su piel.

No se debe utilizar en más de 15 sesiones el mismo juego de electrodos.

3



Medidas de seguridad

No utilizar el instrumento dentro del agua o en un lugar húmedo (sauna, hidroterapia, etc.).

No desconectar los cables del electroestimulador durante la sesión mientras el aparato siga conectado.

4

Nunca conecte los cables del electroestimulador a una fuente eléctrica externa, porque existe el riesgo de tener un shock eléctrico.

No cargue el instrumento cuando los cables y electrodos estén conectados a alguien.



Características técnicas del instrumento

Alimentación: Baterías de níquel recargable 5 VDC \approx 2000 mA

Carcasa: Acrílico

Dimensiones: Longitud: 15 cm; Anchura: 10 cm ; Altura: 10 cm

Peso: 350 g

Entradas de carga de baterías

Entrada cable EMG

Entrada cable Electroestimulador

Activación: Botón de Reset



5

Preparando el instrumento

- A. Accionar la batería.
- B. Encender el instrumento
- C. Conecte los cables a los electrodos autoadhesivos.
- D. Limpie la piel.
- E. Retire la cubierta de plástico de los electrodos antes de conectar a la piel. La parte adhesiva de los electrodos no debe ser tocada.
- F. Ingrese a la aplicación desde un dispositivo móvil con sistema Operativo Android y seleccione el programa a realizar.

6



Electromiografía

- A. Conecte el cable EMG en la entrada respectiva
- B. Ejecutar la aplicación en el dispositivo Android, seleccionar la opción electromiografo.



7

Electroestimulador

- A. Conecte el cable electroestimulación en la entrada respectiva
- B. Ejecutar la aplicación en el dispositivo Android, seleccionar la opción electroestimulador.
- C. Acorde con la función a realizar colocar lo electrodos como se muestra en la aplicación.
- D. Configurar parámetros de tiempo e intensidad.

8



9



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
EXTENSIÓN LATAQUINGA
INNOVACIÓN PARA LA EXCELENCIA

ACCIONAMIENTO
BATERIA

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

TEMA:
DISEÑO Y CONSTRUCCIÓN DE UN ELECTROESTIMULADOR Y
ELECTROMIOGRAMA CON COMUNICACIÓN INALÁMBRICA PARA
LA DETECCIÓN Y TRATAMIENTO DE ENFERMEDADES
NEUROMUSCULARES

AUTORES:
SINCHIGUANO CHILIQUINGA NELSON ROLANDO
TUTILLO TAIPE KARINA ETELVINA
2015

Cara Frontal del Instrumento

Lateral Izquierdo
del Instrumento

CABLE
ELECTROMIOGRAMA

10

ELECTROESTIMULADOR Y
ELECTROMIOGRAMA

CABLE
ELECTROESTIMULADOR

CARGA DE BATERIA

ACCIONAMIENTO
INSTRUMENTO



Cara Posterior del Instrumento

Lateral Derecho
del Instrumento

INDICADOR
NIVEL DE BATERIA



PREGUNTAS FRECUENTES Y POSIBLES FALLOS

1. La aplicación en el dispositivo android no se inicia.

Sol. Revise que el instrumento este encendido y correctamente conectado los sensores.

2. El Instrumento no se enciende.

Sol. Revise que las baterías estén cargadas y el fusible en buen estado, caso contrario retirar la tapa del módulo, con ello poder extraer la placa o las baterías.

3. La señal mostrada en el Electromiograma no corresponde a la de una persona normal.

Sol. Revise la conexión de los electrodos, que estén en contacto con la piel correctamente y de ser necesario reubique el electrodo de referencia.

11



UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE

ING. ELECTRÓNICA

Nelson Sinchiguano
Karina Tutillo
2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por la Srta. Karina Etelvina Tutillo Taipe y el Sr. Nelson Rolando Sinchiguano Chiliquinga, bajo nuestra supervisión.

ING. JOSÉ BUCHELI

DIRECTOR DEL PROYECTO

ING. EDDIE GALARZA

CODIRECTOR DEL PROYECTO

ING. FRANKLIN SILVA

DIRECTOR DE LA CARRERA DE
INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN

DR. RODRIGO VACA

SECRETARIO ACADÉMICO