

**AMBIENTE DE ENTRENAMIENTO VIRTUAL PARA PACIENTES CON
AMPUTACION DE MANO**



Trabajo de Grado

**Johny Daniel Rosero
Piedad Lucía Segura**

Director: Ing. Vladimir Trujillo

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Línea de I+D en Robótica y Control
Ingeniería en Automática Industrial
Popayán, Septiembre de 2013**

**AMBIENTE DE ENTRENAMIENTO VIRTUAL PARA PACIENTES CON
AMPUTACION DE MANO**

**Johny Daniel Rosero
Piedad Lucía Segura**

**Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de**

Ingeniero en Automática Industrial

**Director:
Director: Ing. Vladimir Trujillo**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Línea de I+D en Robótica y Control
Ingeniería en Automática Industrial
Popayán, Septiembre de 2013**

Hoja de Aprobación

Director _____
Ing. Vladimir Trujillo

Jurado _____

Jurado _____

Fecha de sustentación:

AGRADECIMIENTOS

Agradecemos a Dios por brindarnos la fuerza necesaria para llevar a buen término el presente trabajo de grado.

Agradecemos a nuestras familias por el apoyo durante estos años de formación profesional y personal.

Agradecemos también, al candidato a doctor, PhD(c) Ingeniero Cesar Quinayas por su guía y dedicación, vitales para el cumplimiento de los objetivos del presente proyectos.

También agradecemos encarecidamente a nuestro director de tesis, Ingeniero Vladimir Trujillo por su gestión y asesoría durante el transcurso del presente proyecto. Por último, agradecemos a la Facultad de Ingeniería Electrónica y Telecomunicaciones en representación de su personal por la excelente gestión prestada.

Agradezco a mis padres por darme la vida, el amor, la dedicación, la paciencia y el apoyo en todo momento

Agradezco a mi compañero de tesis Daniel, al Ingeniero Carlos Gaviria quien nos acompañó inicialmente en este proceso, al Ingeniero Vladimir Trujillo, Cesar Quinayas, Andrés Vivas quienes continuaron acompañándonos en este proyecto a todas las personas involucradas en mi formación personal e intelectual, a mis maestros, amigos y a la facultad de ingenierías por poner su servicio a nuestra disposición.

RESUMEN

Este trabajo de grado evidencia el proceso de desarrollo de un ambiente virtual de entrenamiento para amputados de mano, el cual fue creado con el fin de facilitar la adaptación de prótesis robóticas a estas personas. Este entrenador consiste de un ambiente virtual el cual le permite al paciente la visualización de los ejercicios sobre una prótesis virtual; y un componente hardware que permite la adquisición e identificación de señales bioeléctricas provenientes del paciente.

El software planteado se creó siguiendo las especificaciones de médicos Fisioterapeutas que trabajan en el proceso de rehabilitación de pacientes con amputación de miembro superior, para el desarrollo de la interfaz gráfica se tuvo en cuenta las observaciones de un diseñador gráfico buscando que las características del software nos permitieran desarrollar un programa de fácil aprendizaje y usabilidad para el usuario. Entre sus características adicionales consta de una base de datos, la cual permite guardar la información relevante de cada paciente y llevar un registro de los posibles candidatos a la adaptación de una prótesis que pueden ser analizados por el profesional de la salud; posteriormente se diseñó la navegabilidad de las ventanas, bajo las especificaciones de diseño del escenario, a partir de ello se crearon los objetos en 3D, para finalmente empezar a vincular las distintas herramientas software y darle forma al ambiente virtual de entrenamiento, acoplando los parámetros médicos de diseño y de funcionalidad requeridos.

Por otro lado se diseñó un sistema de adquisición de señales mioeléctricas consistente en una etapa de adquisición de la señal, una etapa de filtrado, una etapa de amplificación y una etapa de conversión analógica/digital el cual se integró con el trabajo [1] con el cual se obtienen las diferentes características para la identificación del movimiento de mano correspondiente, estas características se utilizaron para el entrenamiento de una red neuronal diseñada en el trabajo [2], obteniéndose así, un sistema de adquisición e identificación de señales EMG completo.

ABSTRACT

This paper evidence the development process of a virtual environment training for hand amputees, which was created in order to facilitate the adaptation of myoelectrical prostheses to these people. The trainer is a virtual environment which allows the visualization of exercises on a virtual interface, and a hardware component that enables identification and acquisition of bioelectric signals from the patient.

The proposed software is created according to the specifications of physicians Physiotherapists working in rehabilitation of patients with upper limb amputation. The GUI development took into account the comments of a graphic designer, looking forward the features software that allows developing an easy program for the user. The software consists of a database, which can store the relevant information of each patient and keep track of the patients, that information can be analysed by the health care professional; subsequently designed the navigability of the windows based on the design specifications, the 3D objects composing the scene were linking the various software tools and giving form the virtual environment training, engaging the medical parameters and design required functionality.

On the other hand designed an acquisition system consisting of a myoelectric signal acquisition phase signal a filtering stage, an amplification stage and a step of converting analogy / digital signal integrated with work [1] which is obtained different characteristics for the identification corresponding of different hand movement, these features are used to train a neural network designed in the work [2], thereby obtaining a system of identification and acquisition of EMG signals completely.

Contenido

1. INTRODUCCION.....	16
2. ESTUDIO DE LA SEÑALES MIOELECTRICAS	19
2.1 Señales mioeléctricas.....	19
2.2 Morfología subyacente en las señales electromiográficas	20
2.3 Características de la señal	21
2.4 Parámetros típicos de la señal	21
2.5 Adquisición de la señal.....	23
2.5.1 Electrodo de superficie de piel	23
2.5.2 Aspectos específicos de la aplicación de electrodo.....	24
2.6 Amplificación	24
2.7 Conversión Análoga/Digital.....	25
2.8 Fuentes de ruido e incertidumbre	25
2.9 Herramientas para entrenamiento de prótesis de miembro superior.....	25
3. DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE SEÑALES EMG	26
3.1 Acondicionamiento de las señales Electromiográficas.....	26
3.1.1 Obtención de la señal en miembro amputado.....	26
3.1.2 Pre amplificación	26
3.1.3 Filtrado	28
3.1.4 Amplificación	28
3.1.5 Amplificador Diferencial:.....	29
3.2 Digitalización de la señal electromiográfica	29
3.3 Psoc 5LP	30
3.3.1 Procedimiento de adquisición con PSOC 5LP	30
3.3.2 ADC DEL SIG:	31
3.3.3 UART.....	29
3.3.4 LCD.....	31
3.3.5 Adquisición de datos en Labview.....	32
3.3.6 Procesamiento de señales	34
3.4 FiO Family	35

3.4.1 FiOLite.....	35
3.4.2 Adquisición de señales EMG	36
3.4.3 Procesamiento de señales	37
3.4.4 Caracterización de patrones	37
3.4.5 configuración e implementación en tarjetas fio	38
3.4.6 Procedimiento para obtención de una base de datos de señales EMG	39
3.4.7 Red Neuronal	43
4.DISEÑO E IMPLEMENTACIÓN DE UN AMBIENTE VIRTUAL DE ENTRENAMIENTO	44
4.1 Factores de diseño	44
4.1.1 Usabilidad	44
4.1.2 Utilidad.....	44
4.1.3 Facilidad de uso.....	44
4.1.4 Facilidad de aprendizaje	44
4.2 Diseño de interfaz gráfica de usuario	44
4.3 Lista.....	43
4.4 Interfaz para entrenamiento	46
4.5 Estructura de Interfaz de usuario	46
4.5.1 Navegación entre ventanas	46
4.5.2 Ventana Selección de niveles	46
4.5.3 Niveles de entrenamiento:	47
4.5.3.1 Nivel 1.....	47
4.5.3.2 Nivel 2.....	47
4.5.3.3 Nivel 3.....	48
4.6 Introduccion a QT	48
4.6.1 Qt Designer.....	49
4.6.2 Qline edit.....	49
4.6.3 Lay out Horizontal y vertical	49
4.6.4 Labels	49
4.6.5 Group box.....	49
4.6.6 Spacers	49
4.7 Estructura de almacenamiento de datos del paciente	50
4.7.2 Clase Lista	51

4.8	Requerimientos para el diseño del entorno virtual.....	51
4.9	Modelado 3D de objetos	52
4.10	Diseño de mano virtual	53
4.11	Diseño de escenarios.....	56
4.11.1	Nivel 1 Y 2	56
4.11.2	Nivel 3.....	56
4.12	VTK.....	57
4.13	Diagrama UML.....	58
4.13.1	Interfaz de Usuario.....	60
4.13.2	Objeto Actor.....	60
4.13.3	Detección Colisiones.....	60
4.13.4	Cargar Mano.....	60
4.13.5	Cámara.....	60
4.13.6	Iluminación.....	60
4.13.7	Escenario.....	60
4.13.8	EscenarioN1N2.....	60
4.14	Clase Objeto Actor	61
4.15	Clase Cargamano.....	61
4.15.1	Funciones de la clase Cargamano	62
4.15.2	EnsambleMano	62
4.15.3	Función Origen	63
4.15.4	RotarArt	63
4.15.5	desMano	63
4.15.6	Giro.....	63
4.15.7	upDown.....	63
4.15.8	CargaMano	63
4.15.9	bposMano	63
4.15.10	orienMano	64
4.16	Clase Cámara	64
4.17	Clase Iluminación	65
4.18	Clase Detección de Colisiones	65
4.19	Clase InterfazUsuario	66

5.	PROTOCOLO DE ENTRENAMIENTO	68
5.1	Síndrome de dolor fantasma.....	68
5.2	Factores periféricos	68
5.3	Síndrome del miembro fantasma	68
5.4	Consideraciones a tener en cuenta	69
5.5	Procedimiento	70
5.5.1	Preparación de la piel	70
5.5.2	Colocación de electrodos.....	70
5.5.3	Procedimientos de verificación.....	71
6.	RESULTADOS	73
6.1	Electrodos.....	73
6.2	Pre Amplificación	73
6.3	Filtro pasa bandas	73
6.4	Amplificación.....	74
6.5	Base de datos de señales EMG obtenida.....	72
6.6	Resultados de la Red Neuronal	77
6.8	Resultados Software	78
6.10	Test de usabilidad.....	81
7.	CONCLUSIONES	843
8.	TRABAJO FUTURO	864

Lista de figuras

Figura 1 Señal electromiográfica [8].....	19
Figura 2 Potencial de unidad muscular [8]	19
Figura 3 Unidad motora [8]	20
Figura 4 Proceso de polarización/despolarización de la UM [8]	20
Figura 5 Interpolación de señales de UM activas [8]	21
Figura 6 Forma típica de una señal EMG [8].....	21
Figura 7 Parámetros de amplitud estándar [9]	22
Figura 8 Espectro de frecuencias de una señal EMG [8]	22
Figura 9 Diferencial de potencial entre electrodos [8].....	23
Figura 10 Diferentes tipos de electrodos [8].....	24
Figura 11 Colocación de Electrodo (Imagen propia).....	24
Figura 12 Circuito de un Monitor Medico de señales ECG [40]	27
Figura 13 Circuito Etapa de Pre Amplificación (Imagen propia)	27
Figura 14 Circuito etapa de Filtrado (Imagen Propia).....	28
Figura 15 Circuito Etapa de Amplificación (Imagen propia).....	29
Figura 16 Circuito Amplificador Diferencial (Imagen propia).....	29
Figura 17 Tarjetas Psoc 5, [20]	30
Figura 18 Modulo ADC de la Tarjeta PSoc (Imagen Propia)	31
Figura 19 Configuración del bloque UART (Imagen Propia).....	32
Figura 20 Programa implementado en la Tarjeta Psoc (Imagen Propia)	32
Figura 21 Interfaz Gráfica de LabView (Imagen propia)	33
Figura 22 Código desarrollado para Visualizar las señales en LabView (Imagen propia) .	34
Figura 23 Implementación Red Neuronal en la Tarjeta Psoc (Imagen Propia)	34
Figura 24 Tarjetas FIO, de izquierda a derecha (a), (b), (c) [21], [22].....	35
Figura 25 Librerías RapidSTM32.	36
Figura 26 Configuración bloques de adquisición.....	36
Figura 27 Configuración de los bloques	37
Figura 28 Red neuronal en Simulink	39
Figura 29 Verificación registro de la señal EMG (Imagen Propia)	40
Figura 30 Señales correspondientes a Movimientos de la mano.....	41
Figura 31 De izquierda a derecha Señal EMG1, EMG2 Agarre Grueso (Imagen Propia). 41	
Figura 32 De izquierda a derecha Señal EMG1, EMG2 Agarre Pinza (Imagen Propia).....	41
Figura 33 De izquierda a derecha señal EMG1, EMG2 Mano abierta (Imagen Propia)....	42
Figura 34 De izquierda a derecha señal EMG1, EMG2 Mano en reposo (Imagen Propia)42	
Figura 35 De izquierda a derecha Señal EMG1, EMG2 Todos los agarres (Imagen Propia)	43
Figura 36 Agarres de mano identificados en [1].....	43
Figura 37 Diseño ventana de inicio ambiente virtual (Imagen Propia)	45
Figura 38 Diseño ventana de niveles del entrenador virtual (Imagen Propia).....	47

Figura 39 Diseño ventana Nivel 1 entrenador virtual (Imagen Propia).....	47
Figura 40 Diseño ventana Nivel 2 entrenador virtual (Imagen Propia).....	48
Figura 41 Diseño ventana Nivel 3 entrenador virtual (Imagen Propia).....	48
Figura 42 Diseño Formulario de Datos del paciente (Imagen Propia)	50
Figura 43 Clase Datos persona (Imagen Propia)	51
Figura 44 Clase Lista	51
Figura 45 Capturas de imagen del software Blender. [33].	52
Figura 46 Imágenes muestras de las Manos (Imagen Propia)	53
Figura 47 Extrucción inicial de palma de la mano (Imagen Propia)	53
Figura 48 Detallado del dedo pulgar (Imagen Propia)	54
Figura 49 Extrucción inicial de dedos (Imagen Propia)	54
Figura 50 Detallado del modelado de dedos (Imagen propia)	54
Figura 51 Modelado de uñas y pliegues de flexión (Imagen propia).....	55
Figura 52 Modelado completo de dedos	55
Figura 53 Dimensionado del modelo de la mano (Imagen Propia)	55
Figura 54 Escenarios base de diseño [34],[35].	56
Figura 55 Plano del Nivel 3 [36].....	56
Figura 56 Visualización del escenario virtual (Imagen Propia)	57
Figura 57 Modelo virtual del escenario (Imagen Propia)	57
Figura 58 Diagrama UML Entrenador Virtual (Imagen propia)	59
Figura 59 Clase Objeto Actor.....	61
Figura 60 Clase Cargar Mano.....	62
Figura 61 Orientación de los dedos de la mano (Imagen propia)	64
Figura 62 Clase Cámara	64
Figura 63 Clase Iluminación.....	65
Figura 64 Configuración de la Iluminación (Imagen propia)	65
Figura 65 Clase Detección de Colisiones.....	66
Figura 66 Clase Interfaz de Usuario.....	67
Figura 67 Posiciones Anatómicas de colocación de electrodos superficiales, vista frontal y frotal y dorsal [8]	71
Figura 68 Objetos diseñados para el agarre grueso (Imagen Propia)	72
Figura 69 Etapas desarrolladas para la adquisición de señales EMG	73
Figura 70 Cable UTP, Cable especial para señales EMG	73
Figura 71 Obtención de Señal amplificada (Imagen propia).....	74
Figura 72 Señal completa EMG en Intervalos de diferentes movimientos (Imagen propia)	74
Figura 73 Circuito Adquisición de Señales EMG (Imagen propia)	75
Figura 74 Implementación circuito de adquisición de señales EMG (Imagen propia)	75
Figura 75 Base de datos de señales EMG obtenida (Imagen Propia)	76
Figura 76 Matriz de características (Imagen propia)	76
Figura 77 De izquierda a derecha Longitud de Onda, AVG (Imagen Propia)	76
Figura 78 De izquierda a derecha Energía, Valor Medio (Imagen Propia).....	77
Figura 79 Varianza (Imagen Propia)	77

Figura 80 Entrenamiento de la Red Neuronal (Imagen Propia)	77
Figura 81 Resultado del entrenamiento de la Red (Imagen Propia)	78
Figura 82 Formulario Datos del paciente (Imagen propia).....	78
Figura 83 Lista base de datos de los pacientes (Imagen propia).....	79
Figura 84 Interfaz inicial del ambiente virtual (Imagen Propia)	79
Figura 85 Interfaz anterior a los niveles de entrenamiento (Imagen Propia).....	79
Figura 86 Interfaz Primer Nivel del Entrenador virtual (Imagen Propia)	80
Figura 87 Movimientos correspondientes a Mano Abierta y Mano cerrada (Imagen propia)	80
Figura 88 Interfaz Segundo nivel de Entrenador virtual (Imagen propia).....	80
Figura 89 Agarre de diferentes objetos de la interfaz del segundo nivel (Imagen Propia)	81
Figura 90 Interfaz tercer nivel de Entrenador Virtual (Imagen Propia)	81

Lista de tablas

Tabla 1. Valores de 'BaseLine Offset' de los voluntarios.....	40
Tabla 2. Resultados test de usabilidad (Imagen propia).....	82

Lista de acrónimos

ADC: Conversión análoga digital

ANN: (*Artificial Neuronal Network*) Red neuronal artificial

ECG: Señales Electrocardiográficas

EMG: señales Electromiográficas

GUI: Interfaz gráfica de usuario

PA: Potencial de acción

PUM: Potencial de unidad muscular

UART: (*Universal Asynchronous Receiver-Transmitter*) Receptor transmisor asíncrono universal

UM: Unidad motora

UML: (*Unified Modeling Language*) Lenguaje de Modelado Unificado

1. INTRODUCCION

Una amputación se define como la ausencia total o parcial de un miembro o una extremidad; las amputaciones propiamente dichas, es decir que no son congénitas se denominan amputaciones adquiridas y consisten en la separación quirúrgica de una parte o de toda la extremidad, por una lesión traumática o por alguna enfermedad.

Las principales causas de estas amputaciones son el trauma, el cáncer, las infecciones y las enfermedades vasculares y neurológicas.

A nivel laboral las amputaciones traumáticas más frecuentes son las del miembro superior, en especial las de la mano.

En Colombia las amputaciones no solo se dan por enfermedades, sino por accidentes de toda índole, y por el conflicto armado; la Asociación Colombiana de Medicina Física y Rehabilitación estima que la incidencia de amputación en el país se da de 200 a 300 personas por cada 100000mil habitantes.

La incidencia de las amputaciones se ha elevado de forma constante debido a los mayores índices de accidentabilidad y a la prolongación media de vida que permite la supervivencia de personas con trastornos generales predisponentes a la amputación (como las isquemias periféricas). [3]

Las amputaciones en miembros superiores, en particular las producidas por debajo del codo, resultan en una importante pérdida de funcionalidad en las personas afectadas; la imposibilidad para restaurar o incorporar una extremidad afectada hasta un nivel compatible con la vida de los tejidos, constituye la razón fundamental de amputaciones y, consecuentemente, del surgimiento de instrumentos que reemplacen la función del miembro perdido, en torno a esto se han desarrollado prótesis que además de suministrar al paciente la posibilidad de realizar funciones que hacia anteriormente, son cada vez más semejantes a las extremidades humanas.

Para disminuir la sensación de rechazo inicial que supone la colocación de un elemento extraño (la prótesis) sobre el brazo es necesario realizar un entrenamiento, por esta razón se plantea el diseño de un entorno de aprendizaje virtual que ayude a mejorar la adaptación de la prótesis al paciente, un software basado en las observaciones medicas de un ortopedista de forma que la rehabilitación se acople a las condiciones iniciales del paciente y a mejorar progresivamente la destreza de los movimientos de su prótesis a través de la retroalimentación de estímulos visuales.

En la actualidad se han desarrollado diferentes sistemas encaminados a sustituir las prótesis reales en el proceso de aprendizaje y determinación de la adecuación del sujeto a los sistemas mioeléctricos, en estos casos la prótesis que el paciente acciona es una representación tridimensional de una mano en la pantalla del ordenador [4], aunque estos

han sido desarrollos significativos en el campo de las ayudas aumentativas a los discapacitados, han sido sistemas carentes de una interfaz amigable que permita capturar la atención del usuario, en consecuencia el diseño de este tipo de software es de difícil manejo y aprendizaje para el paciente; además este tipo de aplicaciones son de uso privativo y para uso exclusivo de la prótesis al cual está sujeto, razón por la cual se dificulta el acceso de este tipo de terapias a las personas con amputación de miembro superior, minimizando la posibilidad de hacerlas candidatas a una prótesis mioeléctrica.

Una prótesis mioeléctrica es un sistema accionado por servomotores controlado a partir de señales EMG recogidas en la parte sana de la extremidad del paciente para lo cual se precisa de un reconocimiento en tiempo real de las características de dichas señales [4]. Existe un porcentaje elevado entre los amputados de miembro superior que usan poco sus prótesis, esto se debe a la pérdida de información propioceptiva [5], la dificultad de aprender y a las dificultades de uso de la misma, por esta razón se ha trabajado en la fabricación y comercialización de prótesis cada vez más perfeccionadas donde además se hace hincapié en el proceso de adaptación de la prótesis al paciente.

En la actualidad se han desarrollado diferentes dispositivos para la adquisición de señales electromiográficas, comúnmente se hace uso de amplificadores operacionales, en diferentes configuraciones; implementando amplificadores de instrumentación y filtros para el manejo de las frecuencias donde se pueden captar dichas señales, además de utilizar conversores análogos digitales para obtener la representación digital de una señal analógica.

Existen dispositivos tales como Microcontroladores, DsPics, tarjetas programables, etc; que han sido usados en este tipo de sistemas los cuales son utilizados para la digitalización y procesamiento de la señal.

Al obtener la señal EMG se hace necesario identificar el movimiento al cual corresponde esta señal, para lograr este objetivo se han desarrollado algoritmos basados en sistemas difusos, redes neuronales, para lo cual se necesita la extracción de características de la señal y realizar su identificación.

Generalmente el Proceso de rehabilitación para pacientes de amputación consta de una etapa inicial de dolor, fortalecimiento y desensibilización del muñón, en la que se presenta el fenómeno conocido como síndrome de miembro fantasma [6] donde la persona aunque no posee más su extremidad, tiene la sensación de que esta aún está presente, posteriormente el paciente se somete a sesiones de fisioterapia que le ayudaran con la propiocepción, específicamente con el control central y el manejo de distancias, por último se trabaja en la funcionalidad de la prótesis, en el alcance y agarre de objetos haciendo uso de su prótesis.

Para el proceso de entrenamiento se propuso un entorno virtual de entrenamiento dirigido a personas con amputación de miembro superior, el cual supone asistir al proceso de adaptación y aceptación a una prótesis.

El sistema consiste en presentar al usuario un ambiente virtual basado en un entorno habitual, donde se presentan diferentes objetos, los cuales servirán para realizar diferentes tipos de agarres planteados en [1], que se podrán realizar con una mano virtual la cual generara algún movimiento a partir de señales mioelectricas, del mismo modo se implementó una base de datos que proporciona al profesional de la salud la alternativa de llevar un registro de los usuarios y guardar de manera organizada los datos relevantes de cada paciente; posteriormente se diseña el ambiente con el cual el usuario va a interactuar, vinculando las distintas herramientas software y hardware que agrupan los parámetros médicos, de diseño y de funcionalidad requeridos.

2. ESTUDIO DE LA SEÑALES MIOELECTRICAS

2.1 Señales mioeléctricas

Las señales mioeléctricas son aquellas que se producen al tensionar o distencionar un musculo, y resultan de la actividad química producida por la proteína fibrosa miosa [7], dichas señales son la suma de impulsos eléctricos generados por un grupo de células que conforman un musculo, como en la Figura 1.

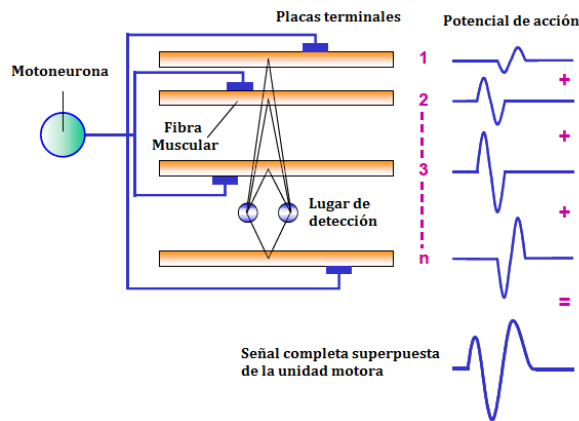


Figura 1 Señal electromiográfica [8]

Estas células producen un potencial que se conoce como potencial de unidad muscular (PUM) el cual se puede definir como un cambio en el voltaje de la membrana, que va desde la negatividad del estado de reposo hasta valores positivos durante un corto lapso como puede verse en la Figura 2.

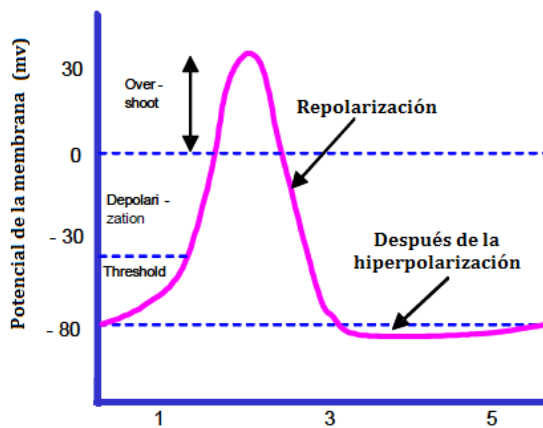


Figura 2 Potencial de unidad muscular [8]

2.2 Morfología subyacente en las señales electromiográficas

La unidad básica funcional que describe el control neuronal del proceso de contracción muscular se llama Unidad Motora, está definida como el cuerpo celular y las dendritas de una neurona motora, los múltiples brazos de su axón y las fibras musculares que intervienen, una UM puede observarse en la Figura 3.

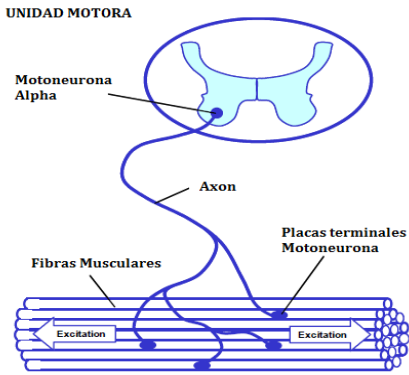


Figura 3 Unidad motora [8]

El proceso biológico que produce un potencial de acción está formado por dos etapas: despolarización y re polarización. La etapa de despolarización es producida por la estimulación del músculo. Cuando el músculo se excita por causa de un estímulo, la membrana axónica de las neuronas musculares abre sus canales de sodio y permiten el paso de los iones de sodio (Na^+) dentro del axón haciendo que el interior del axón se encuentre positivamente cargado. Este proceso provoca que un impulso eléctrico se disipe. Este impulso en estado de reposo se encuentra entre los -80 a -90 mV dependiendo de la fibra muscular, y al momento de la despolarización puede aumentar hasta $+35$ mV, este voltaje depende de la fibra muscular. El potencial de acción sólo dura un determinado periodo, este periodo puede variar según la magnitud del estímulo que produjo el PUM y del músculo. Al momento de la re polarización los canales de sodio se cierran y los canales de potasio se abren. Esto hace que los iones de sodio salgan del interior del axón haciendo que el mismo se polarice negativamente. En este momento el impulso eléctrico pasa de los $+mV$ a los $-mV$ nuevamente. Dicho proceso se puede observar en la Figura 4.

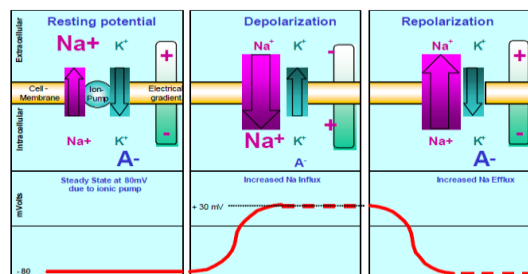


Figura 4 Proceso de polarización/despolarización de la UM [8]

2.3 Características de la señal

Una señal EMG es la suma de diferentes frecuencias fundamentales, depende de la cantidad de unidades motoras activas en el movimiento y sus respectivas frecuencias fundamentales o frecuencias de disparo, un ejemplo puede verse en la Figura 5.

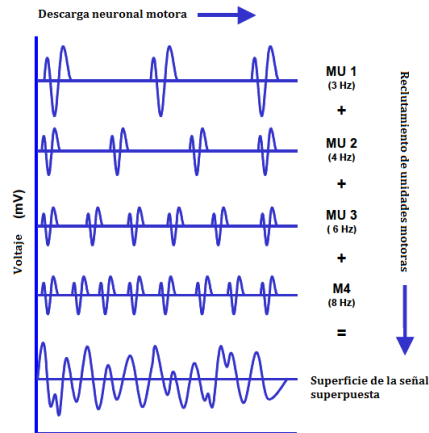


Figura 5 Interpolación de señales de UM activas [8]

Una señal EMG típica tiene una forma similar a la de la Figura 6 donde se observa picos de contracción activa y periodos de descanso. Dichas señales suelen contener picos de amplitud no representables que se pueden deber a ruidos producidos por diversas fuentes.

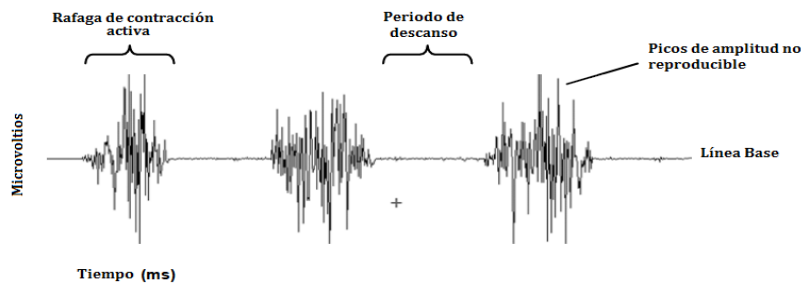


Figura 6 Forma típica de una señal EMG [8]

2.4 Parámetros típicos de la señal

Como cualquier otra curva, las señales EMG pueden ser calculadas con parámetros de amplitud estándar tales como el valor medio, valor mínimo, área e inclinación [8].

El valor pico de una señal EMG es el valor pico promedio de un intervalo de una cantidad de muestras de señales EMG, ya que los trazos que se obtienen son muy variados y dependen del proceso de rectificación de la señal [8].

El valor medio de la amplitud de un intervalo de análisis seleccionado es el cálculo más importante porque este es menos sensible a la diferencia de duración de los intervalos de análisis. El valor medio de una señal EMG es el que mejor describe el proceso de inervación del musculo, cuando este está realizando una tarea o trabajo [8].

El área es una integral verdadera bajo la amplitud de la señal EMG para un cierto periodo de análisis. Dependiendo del punto de vista esta puede ser una ventaja o desventaja ya que depende directamente del tiempo de duración [8].

En la Figura 7 se muestra en qué consisten estos parámetros.

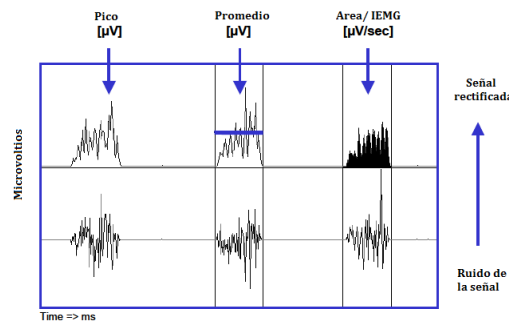


Figura 7 Parámetros de amplitud estándar [9]

Las señales mioeléctricas se encuentran en un rango de voltaje que va desde 1mV hasta 10mV, aunque este puede variar dependiendo del musculo que se esté midiendo y del tamaño del mismo, además se presentan diferencias de amplitud en el individuo logrando mayor amplitud en el hombre y menor en la mujer.

Por otra parte, el cálculo de contenido de frecuencia de una señal EMG puede considerarse como la sumatoria de señales senoidales de diferentes frecuencias donde se pueden utilizar varias técnicas matemáticas para la descomposición de la señal e identificar sus componentes en el rango de los Hertz para el análisis de su espectro total donde se puede identificar cual es la componente más dominante. Los rangos de frecuencia en los que se puede encontrar una señal EMG están dados en los valores de 0 Hz a 1 kHz pero la mayor amplitud de la señal se puede encontrar en los valores de frecuencia de 50Hz a 250Hz.

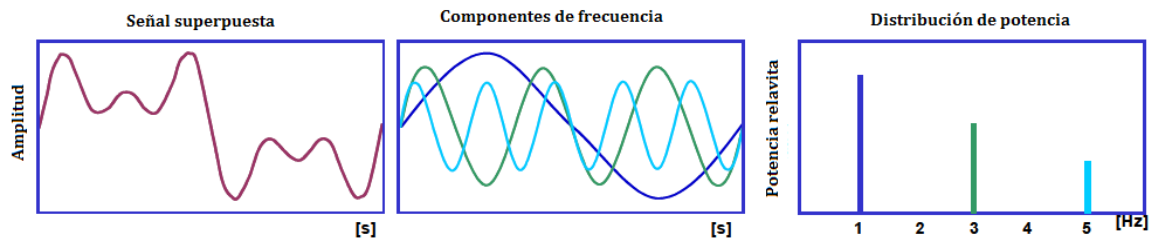


Figura 8 Espectro de frecuencias de una señal EMG [8]

2.5 Adquisición de la señal

El ciclo de polarización y re polarización de la señal EMG forma una onda o dipolo eléctrico el cual viaja a lo largo de la superficie de la fibra muscular. Típicamente la configuración de electrodos bipolares y amplificación diferencial es usada para la medición de señales EMG por simplicidad en la Figura 9 se muestra la detección y medición de una sola fibra muscular. Dependiendo de la distancia espacial entre los electrodos 1 y 2 se obtiene un diferencial de potencia entre los electrodos [8].

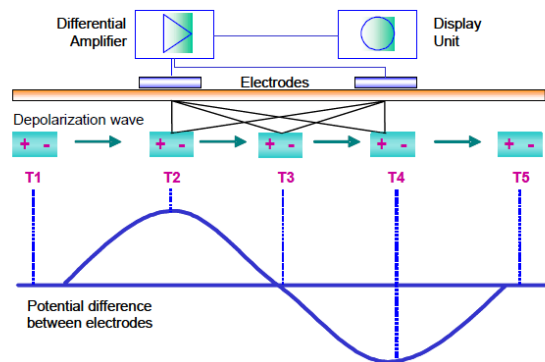


Figura 9 Diferencial de potencial entre electrodos [8]

En la figura 9 se puede apreciar como en T1 el potencial de acción es generado y como viaja a través del par de electrodos. Un incremento de diferencial de potencial es medido entre los electrodos, en la posición más alta de la parte positiva de la onda T2, si el dipolo alcanza una distancia igual entre los electrodos, la diferencia de potencial pasa por la línea cero y alcanza posición más baja de la parte negativa de la onda T4 lo cual significa la distancia más corta entre los electrodos [8].

2.5.1 Electrodo de superficie de piel: en muchos casos los electrodos de superficie de piel son más adecuados para el estudio del comportamiento promedio de la actividad eléctrica de un musculo o de un grupo de músculos [9], aunque esta técnica tiene muchos beneficios la principal limitación es que solo es detectable la superficie muscular. Por lo tanto la selección del tipo de electrodo depende en gran parte de la etapa de pre amplificación propia del electrodo y del material en el cual está construido ya que un tipo de electrodo no alcanza a cubrir todos los requerimientos posibles [8].

Para electrodos de superficie los más recomendados son los de combinación plata-cloruro de plata (Ag/AgCl); este tipo de electrodo es de fácil y cómodo manejo, además posee un aspecto higiénico lo cual evita problemas al momento de disponer de él. Las medidas preferidas para el diámetro del electrodo (área conductiva) son de 10 mm o menor [9], en la Figura 10 se muestran algunos modelos de electrodos superficiales.

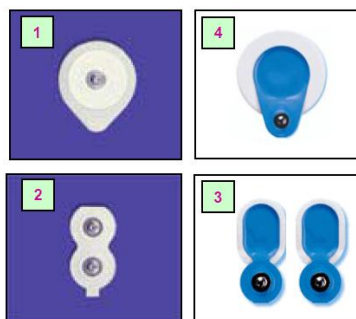


Figura 10 Diferentes tipos de electrodos [8]

2.5.2 Aspectos específicos de la aplicación de electrodo: debido al incremento en la inestabilidad de la señal, algunos investigadores recomiendan no ubicar los electrodos sobre la regiones del punto motor (área con alta densidad de placas terminales motoras) del musculo. Cuando un electrodo es ubicado cerca de una región de punto de motor se puede detectar frecuencias bajas estimuladas por la potencia generada por el movimiento del musculo [8].

Para el estudio de la señal EMG es muy importante colocar el electrodo en el centro del musculo que se va a medir. Otro aspecto es el encogimiento y estiramiento de la piel en sí misma, donde este problema se presenta en mayor intensidad en diferentes partes del cuerpo, por lo tanto la distancia recomendada para la colocación de los electrodos es típicamente entre 1 a 2 cm, un ejemplo puede apreciarse en la Figura 11. [8].



Figura 11 Colocación de Electrodos (Imagen propia)

2.6 Amplificación

Una señal EMG no amplificada se encuentra normalmente en el rango de 2 a 3 mili voltios cuando es leída desde la piel, la señal generalmente es amplificada por un factor de al menos 500, la entrada de impedancia del amplificador podría tener un valor al menos 10x la impedancia obtenida por el electrodo, se recomienda una impedancia de 1 a 10 MegaOmh. El rango de frecuencia de amplificación de una señal EMG en configuración pasa banda podría iniciar desde los 10 Hz con un pasa altas hasta los 500Hz con un pasa bajas, además se puede incluir un filtro Noch para eliminar la señal de las conexión eléctrica [8].

2.7 Conversión Análoga/Digital

Antes de que una señal pueda ser medida y mostrada en un computador, es necesario convertir el voltaje análogo a una señal digital (Conversión A/D).

2.8 Fuentes de ruido e incertidumbre

La señal EMG en el camino de las membranas musculares hasta la entrada de los electrodos se puede ver afectada por varios factores externos alterando la forma y características.

2.9 Herramientas para entrenamiento de prótesis de miembro superior

Un sistema de entrenamiento es vital en el proceso de aprendizaje para saber manejar la prótesis de forma eficiente y natural.

Para cumplir con dichas directrices, las herramientas usadas para entrenar al paciente en el uso de prótesis de miembro superior controladas con las señales mioeléctricas del muñón consisten en los siguientes tipos de dispositivos: despliegues de amplitud de señales mioeléctricas del muñón, videojuegos mioeléctricamente controlados, brazos robóticos, y simulaciones computarizadas de los mismos en el que el paciente controla los movimientos del robot a través de las señales mioeléctricas del muñón.

Algunas de las soluciones disponibles en este campo son los sistemas de despliegue de señales [10], [11], el sistema desarrollado por la universidad de Valladolid [4], los sistemas [12] y [13] desarrollados en la Universidad del Cauca y el Oto Bock's MyoBoy [14]. Este último incluye una función para despliegue de señales, un videojuego simple que simula a partir de gráficas en 2D un control proveniente de la información del paciente. Esta se determina de dos maneras: una virtual que se define desde la interfaz y una real experimentada en la mano robótica.

En "*The Development of a Myoelectric Training Tool for Above-Elbow Amputees*" [15] se indican algunos requerimientos claves y mejoras detectadas en los sistemas de entrenamiento para pacientes amputados. Los sistemas según lo explicado en "*Using virtual reality environment to facilitate training with advanced upper-limb prosthesis*" [16] deben ser, accesibles, portables y confiables, características que le permitan evaluar y grabar los datos de desempeño del paciente, esto con el fin de que pueda usarse en un centro de rehabilitación especializado o también garantizar un fácil acceso en el que el usuario sea capaz, dentro de sus posibilidades, de poder utilizarlo en casa. De ahí que, un sistema de entrenamiento eficiente debe ser diseñado bajo los parámetros modificables a las necesidades del paciente, a los esquemas de control convencional y a los nuevos aportes de punta. Los sistemas igualmente deben ser modulares, y ser capaces de adaptarse a la realidad del paciente brindando diferentes niveles de adaptabilidad según la gravedad de la amputación.

3. DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE SEÑALES EMG

Luego de haber consultado sobre las características y parámetros principales de las señales EMG, se puede definir una etapa de diseño Hardware, el cual permita unir un sistema de adquisición de señales con los algoritmos de caracterización e identificación de señales EMG [1]. Para este fin se implementan las siguientes etapas.

3.1 Acondicionamiento de las señales Electromiográficas

La etapa de acondicionamiento de la señal EMG está constituida por la obtención de esta señal a partir de electrodos superficiales, una etapa de pre amplificación, filtrado de la señal y una etapa final de amplificación.

3.1.1 Obtención de la señal en miembro amputado

Los electrodos utilizados para recoger la actividad eléctrica del musculo y capturar las señales EMG, son electrodos superficiales adhesivos para señales ECG, marca Meditrace®. La ubicación de los electrodos es muy importante en el proceso de captura de señales EMG y para esto se deben tener en cuenta que existe cierta dificultad al adquirir señales EMG en el antebrazo debido a la proximidad de unos músculos con otros, en donde pueden presentarse zonas de inervación [17].

Para el proceso de colocación de electrodos se estima una zona de potencial de acción en la cual se posicionaran los electrodos; se ubican de manera estratégica cinco electrodos, dos para cada señal, EMG1 y EMG2 en configuración diferencial y un electrodo de referencia, la distancia inter-electrodo recomendada es de 2cm [17]. De esta manera se podrá realizar la captura de las señales para ser caracterizadas e identificadas según la intención de movimiento definida en el trabajo de grado [2].

3.1.2 Pre amplificación

Como se mencionó anteriormente las señales obtenidas a partir de las conexiones con los electrodos superficiales son señales muy débiles y en consecuencia, se debe realizar una etapa de pre amplificación, con este fin, se empleó un amplificador de instrumentación AD620, el cual posee características tales como; un bajo nivel de ruido a corrientes de polarización, y un ancho de banda necesario para la captación de señales EMG, los cuales mejoran el rendimiento del dispositivo, A continuación en la Figura 12 se muestra la configuración del circuito de un monitor de señales ECG con un amplificador de instrumentación AD620, el cual es utilizado para capturar las señales EMG provenientes de los electrodos.

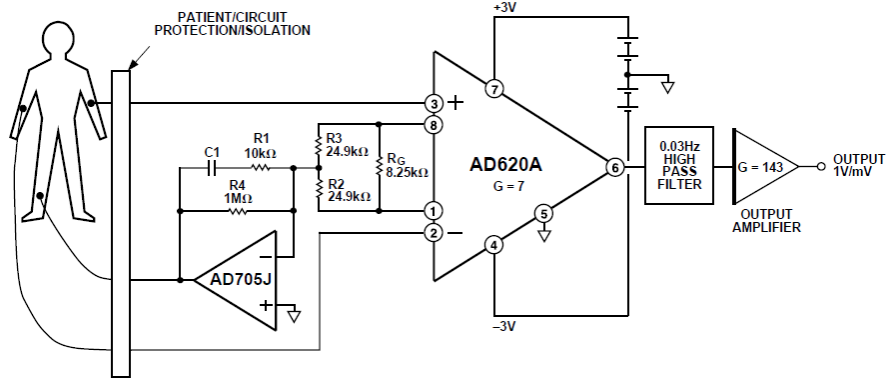


Figura 12 Circuito de un Monitor Medico de señales ECG [18]

La primera etapa del sistema de acondicionamiento de señales tal como lo recomienda el datasheet de este amplificador de instrumentación, se realizó como se muestra en la figura 13.

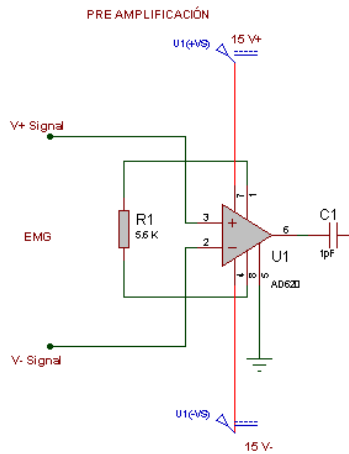


Figura 13 Circuito Etapa de Pre Amplificación (Imagen propia)

Se utilizaron dos amplificadores AD620 para capturar dos señales EMG provenientes de 5 electrodos, 4 para lograr capturar las dos señales EMG1 y EMG2 en configuración diferencial y un electrodo de referencia común a las dos señales. La ecuación (1) a continuación se empleó para calcular el valor de la resistencia de ganancia para una ganancia 10 veces mayor a la señal de entrada, por lo tanto el valor de la resistencia R_G se calculó en $5.49K\Omega$, se utilizó una resistencia R_G de $5.6 K\Omega$ debido a que es el valor más cercano conseguido en el mercado.

$$R_G = \frac{49.4K\Omega}{G-1} \quad (\text{Ecuación 1})$$

3.1.3 Filtrado

Luego de la etapa de pre amplificación de las señales EMG1 y EMG2, se pasó a implementar una etapa de filtrado cuya finalidad es limitar el rango de frecuencias en las cuales se encuentra la información necesaria para una apropiada caracterización de las señales electromiográficas, para esto se implementa un filtro pasa bandas.

El filtro implementado es un filtro Butterworth recomendado en [1], de primer orden, con frecuencias de corte establecidas entre 10Hz y 500Hz, el cual es el rango donde se obtiene la mayor cantidad de información de las señales electromiográficas.

Como era de esperarse la señal adquirida se vio afectada por una frecuencia de interferencia debida al ruido eléctrico proveniente de otros dispositivos y generado por la línea de potencia, por lo que surgió la necesidad de implementar un filtro notch, con una frecuencia de corte de 60 Hz para solucionar este problema. Este también es un filtro pasivo de primer orden, en la Figura 14 se puede apreciar el esquema de este filtro.

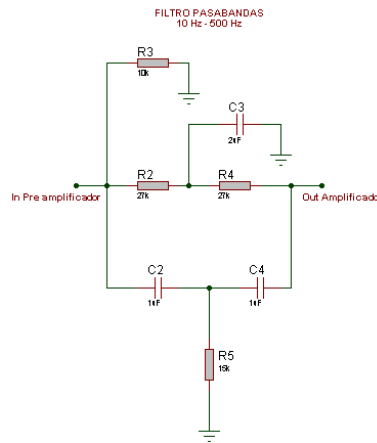


Figura 14 Circuito etapa de Filtrado (Imagen Propia)

3.1.4 Amplificación

Obteniendo la señal debidamente filtrada, se realizaron dos etapas de amplificación utilizando amplificadores operacionales TL074 [19], en configuración amplificador no inversor; la primera etapa de amplificación con una ganancia igual a 58, la salida de esta etapa corresponde a la entrada de la segunda etapa de amplificación con una ganancia 10 veces mayor a la señal de entrada, de manera que la señal que se obtiene para su correspondiente procesamiento no sea una señal muy débil, el diagrama correspondiente a la etapa de amplificación se puede observar en la figura 15.

A partir del acondicionamiento de la señal se procede a la adquisición de la señal para su posterior caracterización e identificación.

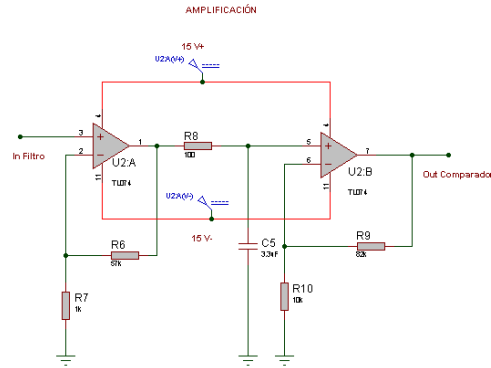


Figura 15 Circuito Etapa de Amplificación (Imagen propia)

3.1.5 Amplificador Diferencial:

Se implementó un circuito en configuración diferencial con una entrada de nivel DC constante a 0.4V, por medio de un divisor de tensión, al cual se le suma y resta el valor de la señal EMG, la ganancia de este amplificador es de uno, para evitar amplificar las señales de ruido, de esta forma se eleva el nivel DC de la señal EMG, en la figura 16 se presenta el diagrama de esta etapa.

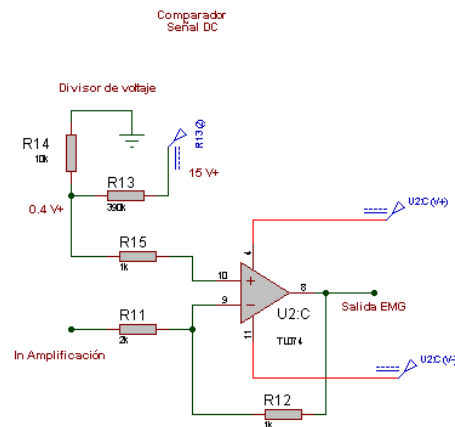


Figura 16 Circuito Amplificador Diferencial (Imagen propia)

3.2 Digitalización de la señal electromiográfica

Para la etapa de adquisición de la señal se pensó en opciones hardware que faciliten las operaciones de programación, minimicen el montaje y permitan embeber los algoritmos necesarios para el procesamiento de las señales.

De esta forma se realizó un análisis de las herramientas que más se aproximaban a las necesidades planteadas.

Existen actualmente dispositivos programables en los cuales se puede embeber algoritmos para el procesamiento de las señales, estos presentan una solución viable al sistema que se requiere implementar, a continuación se mencionaran los dispositivos que se emplearon para la adquisición, algunas características y el procedimiento que se realizó con cada uno de estos.

3.3 Psoc 5LP: Es un microcontrolador programable desarrollado por Cypress Semiconductor en 2002. Esta tecnología de microcontrolador incorpora todo un sistema configurable dentro de un único chip. Comprende una matriz configurable de funciones analógicas, solapada con otra de funciones digitales, proveyendo al sistema de la capacidad de asignar cualquier función a cualquier terminal del circuito integrado, lo que confiere una gran versatilidad al dispositivo [20].

Una de las ventajas de este dispositivo es su robustez puesto que implementa dispositivos analógicos y digitales dentro de un chip en consecuencia se reduce la sensibilidad al ruido, el hardware se reduce de manera considerable, además es compatible con programación en C y ensamblador, en la Figura 17 se puede observar esta tarjeta.



Figura 17 Tarjetas Psoc 5, [20]

Es relativamente fácil de programar debido a la combinación de una programación visual gráfica y scripts en C.

3.3.1 Procedimiento de adquisición con PSOC 5LP

Habiendo obtenido una señal EMG debidamente acondicionada se realiza la conversión análoga Digital. Para este fin se implementa un conversor ADC propio de la tarjeta Psoc 5LP; esta posee dos tipos de conversores ADC, ADC_DeISig y ADC_SAR, se escogió el tipo ADC_DeISig debido a que las frecuencias de muestreo son menores y permite una

configuración en modo diferencial, para la implementación de este conversor se realizan los siguientes pasos.

3.3.2 ADC DEL SIG: De manera general en este bloque se configura el modo de conversión, la resolución, frecuencia de muestreo y la referencia como se muestra en la Figura 18.

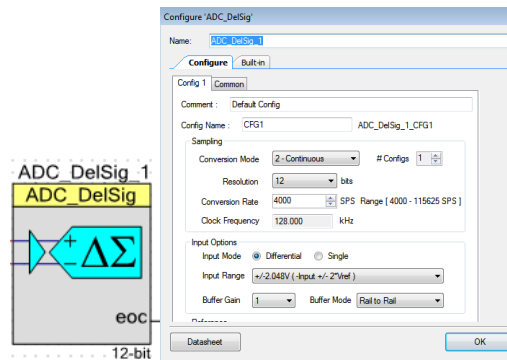


Figura 18 Modulo ADC de la Tarjeta PSoc (Imagen Propia)

Para cumplir con los requerimientos especificados en la tesis de maestría [1] el bloque de conversión ADC se configura con una resolución de 12 bits, un valor de referencia de 1.024v y el buffer de ganancia con un valor igual a uno en modo “level shift”. Para garantizar la obtención de los valores negativos de la señal se configura el rango de entrada con un valor de $(-input \pm 6 \cdot V_{ref})$ en el cual el rango de entrada es sumado y restado seis veces a 1.024v de manera que desde el punto inicial al punto final haya un voltaje pico a pico de 6.14v. La conversión AD se realiza con una frecuencia de muestreo de 8000 muestras por segundo, la cual es la mínima configurable en el bloque ADC_DelSig.

3.3.3 UART: Para verificar el funcionamiento correcto de conversión se implementó un programa en Labview con una etapa de comunicación serial a un computador, el bloque que se utilizó para lograr la comunicación entre la tarjeta Psoc y el computador es el bloque UART el cual permite la transmisión de datos al computador; en este bloque se configuran los pines de recepción y de transmisión.

Las configuraciones que se realizan en el bloque UART de comunicación de la tarjeta Psoc 5LP son; una rata de transmisión 115200 baudios, con una resolución de 8 bits y un bit de parada, el bloque es configurado solamente para la transmisión de datos; la ventaja de este bloque es que permite la transmisión de hasta 4 bytes en el buffer, lo cual permite el envío de datos de las dos señales a la vez, esta configuración se puede observar en la Figura 19.

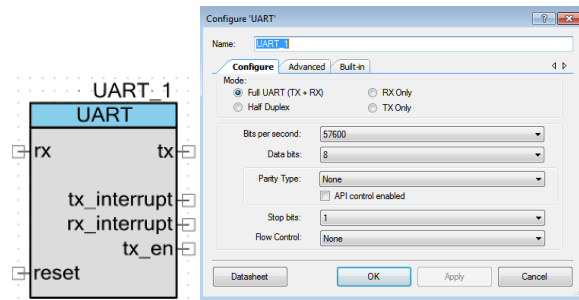


Figura 19 Configuración del bloque UART (Imagen Propia)

3.3.4 LCD: Se utilizó un bloque correspondiente a la configuración de la tarjeta Psoc 5LP, para visualizar el valor de los canales CH1 y CH2

Finalmente el programa implementado en la interfaz de programación grafica de Psoc 5LP, se puede observar en la figura 20

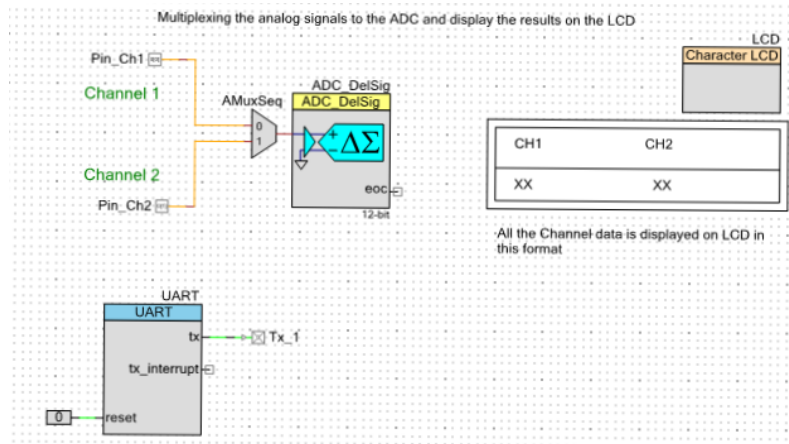


Figura 20 Programa implementado en la Tarjeta Psoc (Imagen Propia)

3.3.5 Adquisición de datos en Labview

En Labview se implementó un programa que permite la recepción, almacenamiento y visualización de los datos digitalizados desde la tarjeta Psoc 5LP correspondientes a las señales EMG. El medio de comunicación entre la tarjeta y el computador es un convertor de datos externo Serial-USB.

A continuación, se realiza una explicación del programa desarrollado en Labview. En la ventana de interfaz se colocaron dos osciloscopios para la representación visual de las señales, uno correspondiente a la señal digital y otra a la señal analógica, también se colocaron visores de cadenas de datos para comprobar que el valor enviado por la tarjeta fuera el mismo que se estuviera leyendo en el programa de Labview, y un cuadro de

dialogo el cual permite ubicar la dirección del archivo TXT en el cual se van a guardar los datos de las señales.

En la Figura 21 se muestra la ventana correspondiente a la interfaz de visualización.

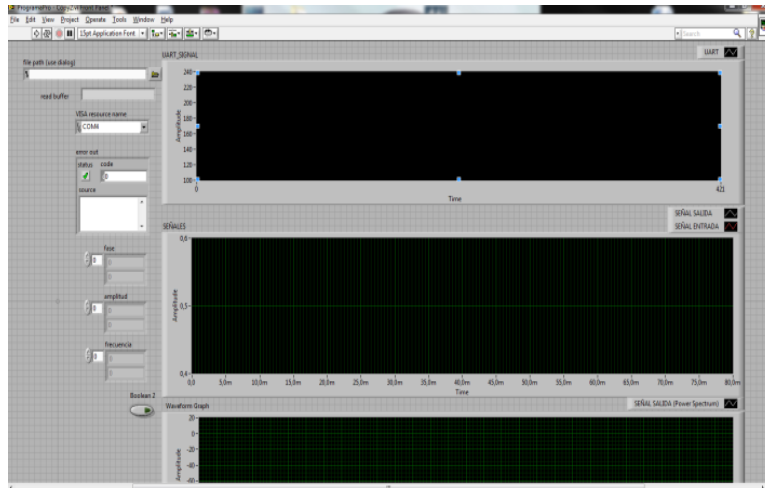


Figura 21 Interfaz Gráfica de LabView (Imagen propia)

En la ventana de programación visual de Labview se implementaron los siguientes bloques y funciones para la lectura de los datos desde la tarjeta PSoc.

Dentro de una estructura de secuencias se implementa un bloque VISA para comunicación de puerto serial, el cual se configura con los mismos valores de comunicación serial de la tarjeta PSoc. También se agrega un ciclo “for” en el cual se repite de forma consecutiva la lectura de los datos por medio de un bloque de lectura en el cual se especifican el puerto de comunicación y los bytes que se van a leer.

Ya que la transmisión serial permite un envío de 8 bits a la vez (un byte) y el dato proveniente desde la tarjeta es de diez y seis bits, (dos bytes), hubo la necesidad de partir el dato proveniente de la tarjeta para transmitir un byte a la vez. En el programa de Labview se implementa una función de concatenación del dato, en el cual primero se recibe un byte correspondiente a la parte baja de dato y luego la parte alta, para luego proceder a la reconstrucción del dato en dos bytes, obteniendo la información completa correspondiente a la señal EMG. Para luego poder guardar el dato en un archivo .txt por medio de la función ‘write to text file’ en la cual se especifica la ruta del archivo y los datos que se van a guardar, el código realizado para la adquisición en LabView se puede observar en la Figura 22.

3.4 FiO Family: Las tarjetas de la familia FiO, son tarjetas basadas en STMicroelectronics STM32, estas han sido diseñadas inicialmente como una solución asequible para sistemas embebidos [21], [22]. Estas tarjetas son:

- FiOStd (a)
- FiOLite (b)
- STM32VLDISCOVERY (c)



Figura 24 Tarjetas FiO, de izquierda a derecha (a), (b), (c) [21], [22]

La principal ventaja de estas tarjetas, mostradas en la figura 24, es la programación a partir los bloques de Simulink de Matlab, esto facilita la sistematización de los procedimientos que se van a llevar a cabo para la identificación y caracterización de las señales EMG, que se realizaron en el trabajo [1] desarrollado en la herramienta software Matlab.

Las tarjetas anteriormente nombradas, poseen diferentes características tales como, capacidad de memoria, velocidad de procesamiento, cantidad de bloques internos, y comunicación con el computador; por lo tanto se realizaron pruebas con cada uno de los dispositivos. A continuación se explican los pasos que se desarrollaron con cada una de estas tarjetas.

A continuación se explica el procedimiento que se realizó utilizando la tarjeta FiOLite de las tarjetas de programación FIO Family.

3.4.1 FiOLite

Anteriormente se mencionaron algunas de las características de la tarjeta FiOLite, la mayor ventaja que presentan estos dispositivos es la programación desde la interfaz Simulink de Matlab. De la misma manera como se hizo la implementación en la tarjeta Psoc, se parte de una señal acondicionada mediante Hardware, para proceder a la adquisición y posterior caracterización e identificación de las señales EMG.

Para poder hacer uso de las tarjetas de la familia FiO, surgió la necesidad de implementar las librerías de RapidSTM32 en Matlab Simulink, los bloques que componen esta librería se pueden observar en la Figura 25.

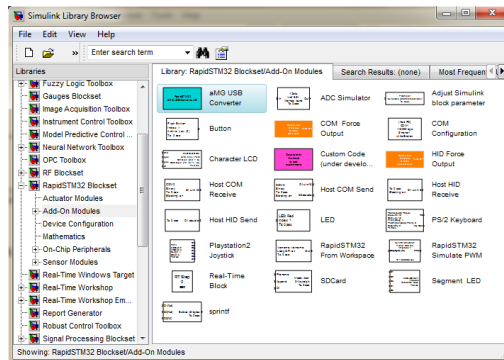


Figura 25 Librerías RapidSTM32.

3.4.2 Adquisición de señales EMG

Para este proceso de adquisición de señales se realizan dos programas diferentes en Simulink, los cuales corresponden, el primero a la conversión de la señal y la transmisión de los datos digitalizados y el segundo a la lectura y procesamiento de estos.

En el primer segmento se implementan los bloques correspondientes para la adquisición de señales provenientes del circuito de acondicionamiento, se realizan las configuraciones necesarias para la digitalización de las señales, el envío de datos desde la tarjeta FiOLite al computador, un bloque correspondiente a la compilación del programa que se va a embeber en la tarjeta y por último la configuración del reloj del sistema.

En la Figura 26 se puede observar el diagrama general para el envío de datos. Este programa es compilado y embebido en la tarjeta FiOLite.

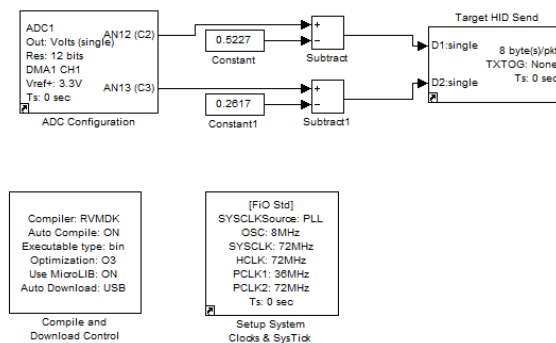


Figura 26 Configuración bloques de adquisición

En la Figura 27 se muestra la configuración de cada bloque.

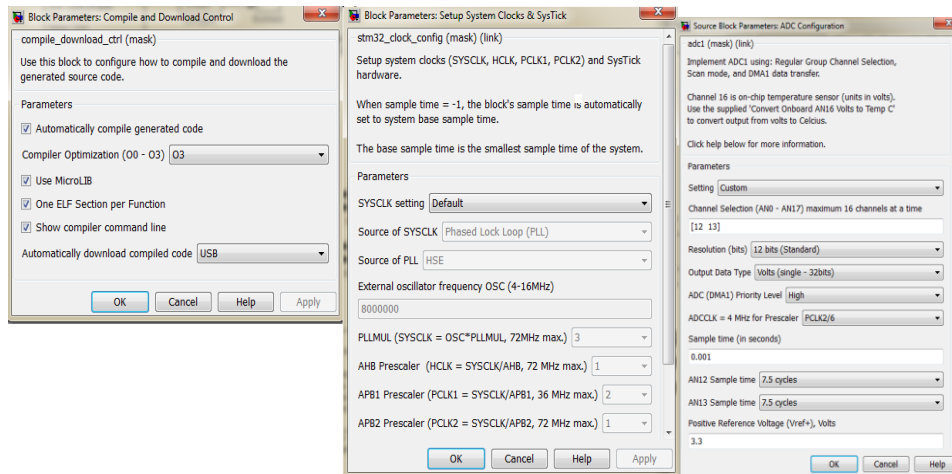


Figura 27 Configuración de los bloques

3.4.3 Procesamiento de señales

Para la posterior lectura y procesamiento de los datos, se realiza la implementación de los bloques necesarios en el programa que va a ser ejecutado en el computador.

Previo a la configuración de los bloques para la lectura y procesamiento de los datos, se realizó un estudio de los algoritmos para la caracterización e identificación de movimiento, el cual se muestra a continuación.

3.4.4 Caracterización de patrones

Para el procesamiento y caracterización de las señales EMG se realizó la implementación de un algoritmo basado en el trabajo de maestría '*Aplicación de Wavelets en el análisis y la extracción de características para la clasificación de señales EMGS*', y posteriormente se realizó la implementación de una red neuronal diseñada en el trabajo de grado '*Sistema Detector de intención de movimiento a partir de señales electromiográficas*, la cual entrega la señal clasificada según la intención de movimiento ejecutado.

A continuación se definen los pasos efectuados en el desarrollo del sistema:

- Inicialmente se revisó y analizó el programa de clasificación de las señales electromiográficas [1], identificando las variables, funciones y la estructura de este algoritmo.
- El siguiente paso fue identificar los bloques de Simulink que permitan la implementación de este algoritmo.
- Por último, transformar el código en lenguaje de Matlab a código C, necesario para la programación de la tarjeta.

3.4.5 Configuración e implementación en tarjetas fio

Para el análisis del algoritmo de clasificación de señales electromiográficas [1] se realizó un desglose de las funciones que componen este programa, lo cual permitió conocer cuáles son las acciones que se ejecutan en cada función.

Por ejemplo, la función nPacked permite dividir la muestra en los paquetes necesarios para el procesamiento de la señal, estos paquetes corresponden a 256, 128, 64 y 32 datos.

Para la implementación de este algoritmo en Simulink se utilizaron bloques 'embedded Matlab Function', este tipo de bloque permite la edición de código, como primera prueba se transcribió el código para la clasificación de señales [1] y se embebió en la tarjeta, este código, en el momento de ejecutarlo se presentaron algunos problemas, esto debido a que el compilador que utiliza la tarjeta FIO Lite no logra una traducción completa del código implementado en Matlab, por ejemplo, la declaración de un ciclo en la siguiente forma: 'for(i=1:n)' en Matlab no es reconocido por el compilador de la tarjeta.

Por lo tanto se realizó una implementación de este código de tal forma que el compilador de la tarjeta pudiera traducirlo por ejemplo la declaración de un ciclo 'for' en Matlab de la siguiente forma 'for(i=1:1:n)' es traducido por el compilador como 'for(int i=1;i<n+1;i++)'. Logrando una correspondencia a un código en lenguaje C.

De esta forma se completa el bloque correspondiente a la parte de caracterización e identificación de señales EMG.

Cabe aclarar que aunque las pruebas del procesamiento de señales se realizaron en un computador, el código está diseñado para poder ser embebido en la tarjeta.

Al realizar la pruebas con el código completo embebido en la tarjeta FIO Lite se presentaron algunos problemas, tales como el tamaño del programa diseñado excedía la capacidad de memoria de la tarjeta, y la cantidad de datos enviados por la tarjeta hacia el computador superaba el permitido por la tarjeta.

En la Figura 28 se muestra el diagrama de bloques que componen esta parte del sistema.

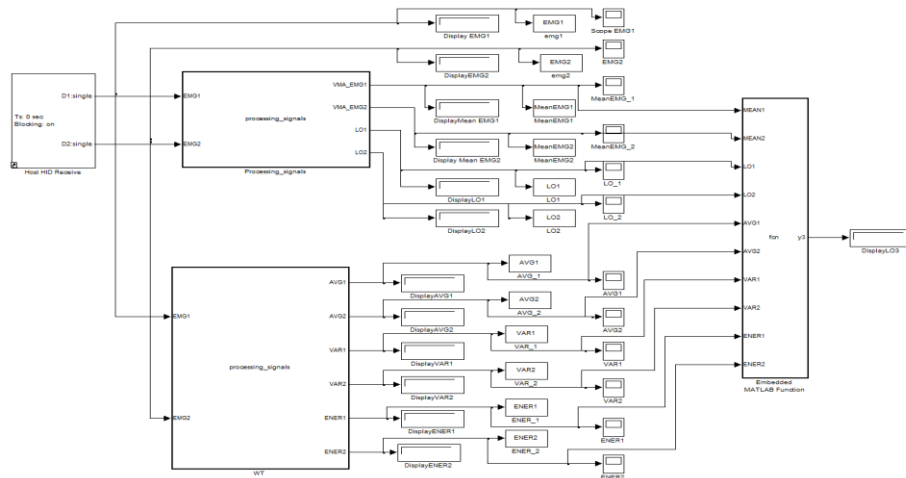


Figura 28 Red neuronal en Simulink

Se logró la adquisición de señales EMG con los dispositivos utilizados, fue necesario realizar la elaboración de una nueva base de datos, ya que se desconocen los parámetros con los que se obtuvo la base de datos que se utilizó en las tesis anteriores [1] y [2].

3.4.6 Procedimiento para obtención de una base de datos de señales EMG

Para la obtención de la base de datos se tomaron muestras de cuatro voluntarios, los cuales ejecutaran los cuatro movimientos identificados, estos datos se guardaran de forma matricial en un arreglo de 256x160, con este arreglo se procede a calcular nuevos parámetros para el re entrenamiento de la red neuronal.

Se definió un tiempo para la toma de señales de 102 segundos lo que equivale a 1024 muestras, que corresponden a un movimiento. Este paso se realiza para cada movimiento.

1. Como primer paso se hace una verificación visual y táctil de la zona del musculo que se desea medir, para evitar ubicar los electrodos en dos músculos diferentes, ya que se pueden ocasionar fenómenos como el “cross-talk” y la colocación de electrodos sobre zonas de inervación, después de haber seleccionado el musculo se realiza una limpieza de la zona seleccionada con algodón empapado de alcohol para asegurar un buen contacto de la piel con los electrodos
2. Se colocan los electrodos sobre el musculo seleccionado con una distancia de 1cm inter electrodo y se conectan los cables especiales para las señales EMG. Se debe tener especial cuidado en la colocación de los electrodos ya que se pueden obtener malos registros de las señales. En la Figura 29 se puede ver un mal registro de la señal EMG

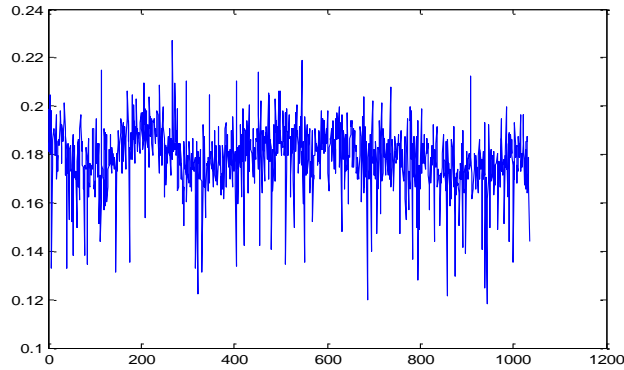


Figura 29 Verificación registro de la señal EMG (Imagen Propia)

3. Se ubica al paciente en una posición cómoda, se le explica el procedimiento, el tiempo y los movimientos que se van a realizar
4. En estado de mano relajada se toman muestras con un intervalo de tiempo de 30 segundos, para encontrar el valor medio de la señal el cual corresponde al desplazamiento de la señal (BaseLine Offset), este valor se suma al nivel DC constante y es restado a la señal. Para los cuatro voluntarios se realizó este proceso obteniendo los resultados observados en la tabla 1, de 'Baseline Offset'

Tabla 1. Valores de 'BaseLine Offset' de los voluntarios

Voluntarios	Baseline Offset	
	EMG1	EMG2
Voluntario 1: Hombre	0.5541	0.5674
Voluntario 2: Mujer	0.5563	0.6547
Voluntario 3: Hombre	0.552	0.61
Voluntario 4: Mujer	0.561	0.571

5. Se solicita al paciente realizar movimientos a diferentes intervalos de tiempo, para verificar que las señales EMG se están obteniendo correctamente, como se muestra en la Figura 30, se pidió al voluntario ejecutar movimientos de mano relajada, agarre grueso, mano relajada, agarre pinza, mano relajada, mano abierta y por ultimo mano relajada, en ese orden.

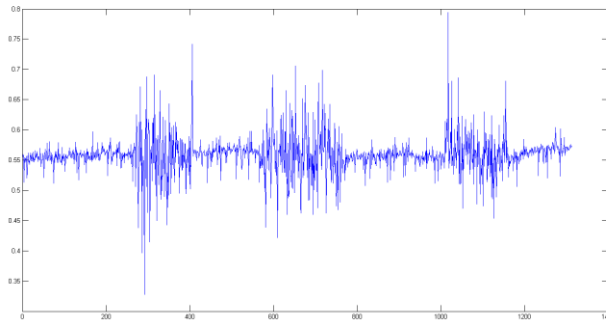


Figura 30 Señales correspondientes a Movimientos de la mano

6. Se procede a tomar las señales correspondientes a cada movimiento en el tiempo especificado y se guardan los datos de las señales EMG correspondientes a cada canal y las características de cada una, este proceso se realiza con cada movimiento que el voluntario ejecuta.

En las Figuras 31, 32, 33, 34 se mostraran los resultados de las diferentes señales que representan los movimientos de interés obtenidos con un voluntario.

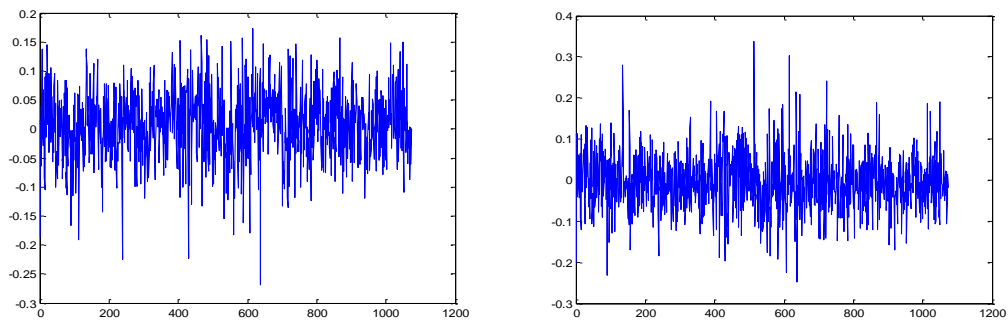


Figura 31 De izquierda a derecha Señal EMG1, EMG2 Agarre Grueso (Imagen Propia)

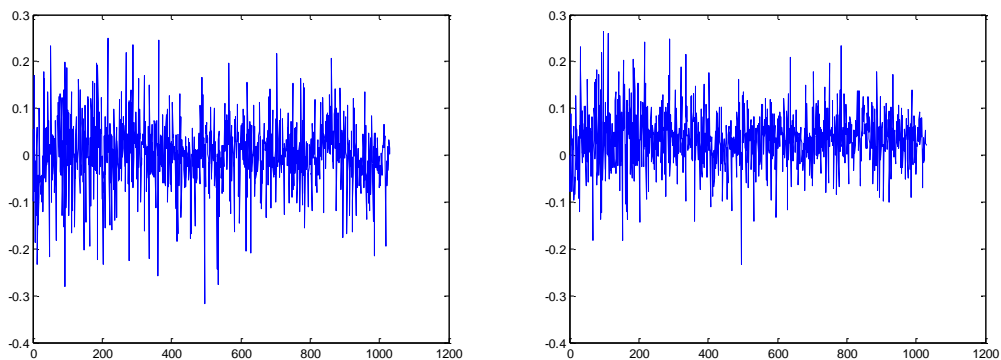


Figura 32 De izquierda a derecha Señal EMG1, EMG2 Agarre Pinza (Imagen Propia)

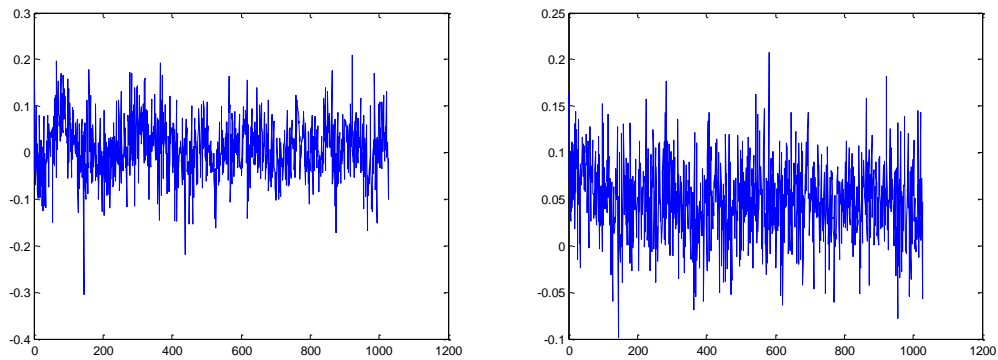


Figura 33 De izquierda a derecha señal EMG1, EMG2 Mano abierta (Imagen Propia)

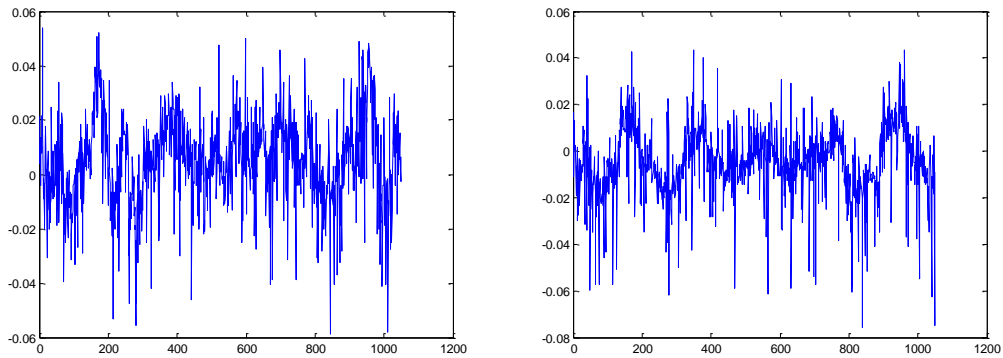


Figura 34 De izquierda a derecha señal EMG1, EMG2 Mano en reposo (Imagen Propia)

Finalmente se realizó una toma de los diferentes movimientos a intervalos de tiempo como se puede observar en la Figura 35; un tiempo inicial de Mano en reposo de 10 segundos, 20 segundos movimiento Agarre grueso, seguidos de 10 segundos de mano en reposo, 20 segundos de Agarre pinza seguidos de 10 segundos de mano en reposo, 20 segundos de mano abierta y finalmente 10 segundos de mano en reposo nuevamente, para obtener 120 segundos de muestreo.

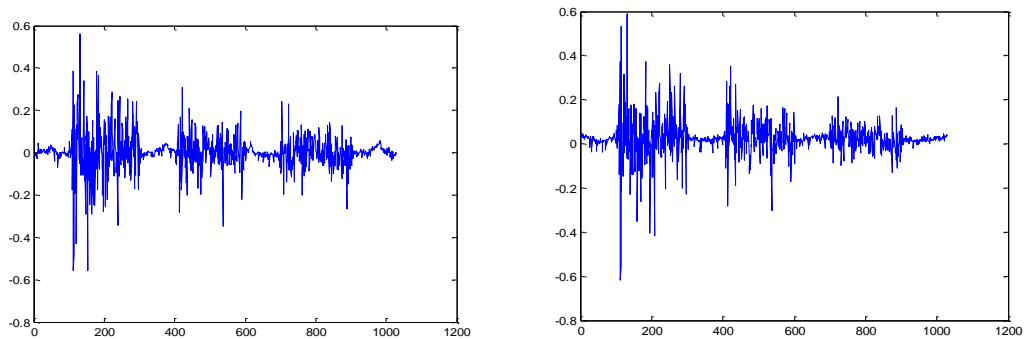


Figura 35 De izquierda a derecha Señal EMG1, EMG2 Todos los agarres (Imagen Propia)

3.4.7 Red Neuronal

A partir de los patrones que se lograron extraer, se implementó en ambos dispositivos hardware la red neuronal diseñada en el trabajo de grado [2], con la cual se obtiene la identificación de los 4 movimientos que se observan en la Figura 36.



Figura 36 Agarres de mano identificados en [1]

4. DISEÑO E IMPLEMENTACIÓN DE UN AMBIENTE VIRTUAL DE ENTRENAMIENTO

En este capítulo se presenta el proceso de diseño e implementación de una herramienta software, en donde se definen los objetos, eventos y acciones que permiten al usuario interactuar con la interfaz del ambiente virtual de entrenamiento, para llevar a cabo esta etapa de desarrollo es necesario tener en cuenta conceptos como interfaz gráfica de usuario, factores de diseño, usabilidad, utilidad, facilidad de uso, facilidad de aprendizaje.

Interfaz gráfica de usuario: Son los elementos gráficos que ayudan a una persona a comunicarse con un sistema o estructura a través de la pantalla de un ordenador [23]. Está compuesta por ventanas, contenedores, menús, barras, botones, campos de texto, etc. Además hace uso de posicionadores (Layout) que permiten gestionar la organización de los componentes gráficos que componen la interfaz y hace uso de eventos para generar la interactividad y respuesta a la entrada del usuario, estos eventos pueden ser desplazamiento del ratón, selección en un menú, botón pulsado, etc. [23].

4.1 Factores de diseño: para el desarrollo de la interfaz gráfica se tuvieron en cuenta factores de diseño como:

- **4.1.1 Usabilidad:** La usabilidad de un sistema o herramienta se define como la utilidad, facilidad de uso, facilidad de aprendizaje y apreciación para una tarea, un usuario y un contexto dado [23].
- **4.1.2 Utilidad:** La utilidad es la capacidad que tiene una herramienta para ayudar a cumplir tareas específicas [23].
- **4.1.3 Facilidad de uso:** Está relacionada directamente con la eficiencia efectividad, medida como velocidad o cantidad de posibles errores [23].
- **4.1.4 Facilidad de aprendizaje:** es la medida del tiempo requerido para trabajar con un cierto grado de eficiencia en el uso de la herramienta, y alcanzar un cierto grado de retención de estos conocimientos luego de cierto tiempo de no usar la herramienta o sistema [23].

4.2 Diseño de interfaz gráfica de usuario

Teniendo en cuenta las definiciones anteriores en relación con los factores de diseño, y la necesidad de los médicos fisioterapeutas, se bosquejaron las ventanas que compondrán la interfaz de usuario a diseñar; esta interfaz estará compuesta por una ventana donde se presentara un formulario para la consignación de datos del paciente, una ventana de inicio al ambiente de entrenamiento y tres ventanas, donde cada una corresponderá a un nivel de dificultad en el entrenamiento.

Habiendo aclarado los elementos tales como botones y espaciadores que se van a colocar en cada ventana, se realizó un modelo de interfaz a lápiz, en este se ubicaron los

componentes en diferentes posiciones para obtener la mejor distribución de los elementos dentro de la ventana, para que el usuario pueda ubicar los diferentes elementos de una forma sencilla. Además se definieron los eventos que se ejecutarán en la interfaz por ejemplo, para la ventana de inicio la cual se muestra en la Figura 37, se coloca un botón de inicio; este botón se ubica en el centro de la ventana y se le da un tamaño relativamente grande con una geometría circular, esto con el fin de enviar un mensaje claro y centrar la atención del usuario.

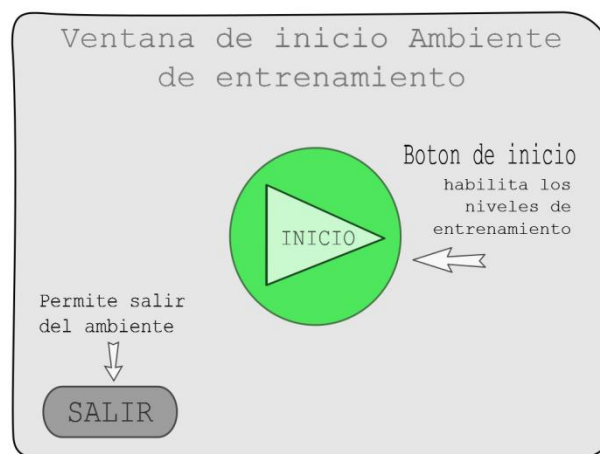


Figura 37 Diseño ventana de inicio ambiente virtual (Imagen Propia)

También se coloca un botón para finalizar la ejecución del programa, este en cambio es más pequeño. De una forma rectangular y está ubicado en una esquina de la ventana, ya que se espera que este botón sea usado solo cuando el usuario realmente desee salir de la aplicación.

4.2.1 Interfaces para el ambiente virtual: se planteó un diseño de interfaz el cual consiste en tres ventanas; la primera ventana que aparecerá será la ventana de formulario de datos de los pacientes, la cual estará diseñada según un formulario de fuerzas utilizado en el departamento de fisioterapia [24], [25], [26], la segunda ventana permite el despliegue de la información obtenida con el formulario, en una tabla tipo excel; la tercera ventana permite el inicio del ambiente virtual de entrenamiento, la cuarta ventana habilita los botones de acceso a cualquiera de los tres niveles del ambiente virtual, finalmente las últimas tres ventanas corresponden a los niveles 1, 2 y 3 de entrenamiento progresivo.

En cada ventana se implementaron los botones necesarios para lograr la interacción del usuario con la interfaz gráfica, a continuación se definen las funciones de cada uno de los botones implementados en el ambiente virtual de entrenamiento.

4.2.2 Botones: Se emplearán para ejecutar diferentes funciones según el evento que le sea asignado. Por ejemplo en la ventana de formulario se encuentran los botones Guardar, Salir y Lista; cada uno de ellos con una función diferente asignada.

Botón Guardar: Se encarga de enviar los datos del paciente a una base de datos.

Botón Salir: Permite cerrar la ventana del formulario de datos del paciente y abrir la ventana de inicio del entrenador virtual.

Botón Lista: Este botón despliega una ventana que contiene una tabla de datos en los cuales están consignados los datos del paciente provenientes de la base de datos.

4.3 Lista: Esta ventana está compuesta de una tabla, un botón para borrar un registro y un botón de salir el cual permite cerrar la ventana actual y volver al formulario, lo importante de esta ventana es que permite la visualización de los datos del paciente de una manera estructurada [27].

4.4 Interfaz para entrenamiento: Esta interfaz es una ventana compuesta por cinco marcos internos correspondientes a las etapas del entrenador virtual, desde el inicio hasta los niveles de entrenamiento propuestos [28].

4.5 Estructura de Interfaz de usuario

A continuación se explica la estructura planteada.

4.5.1 Navegación entre ventanas: A partir de la observación de interfaces de diferentes juegos de computador existentes se planteó la forma de navegación para el ambiente virtual de entrenamiento, evitando el proceso de cierre y apertura de ventanas para acceder a los diferentes niveles de entrenamiento y disminuyendo la dificultad de uso por parte del usuario.

La ventana de Inicio está compuesta por dos botones, uno de inicio y otro de salir. El Botón 'inicio' permite el desplazamiento a una segunda ventana, el botón 'salir' finaliza la ejecución del software [27], [28], el diseño de esta interfaz se puede observar en la Figura 37.

4.5.2 Ventana Selección de niveles: esta ventana está compuesta por cuatro botones, tres de ellos correspondientes a los niveles de entrenamiento propuestos y un botón 'Anterior' que permite regresar a la ventana de inicio [27], [28], Figura 38.

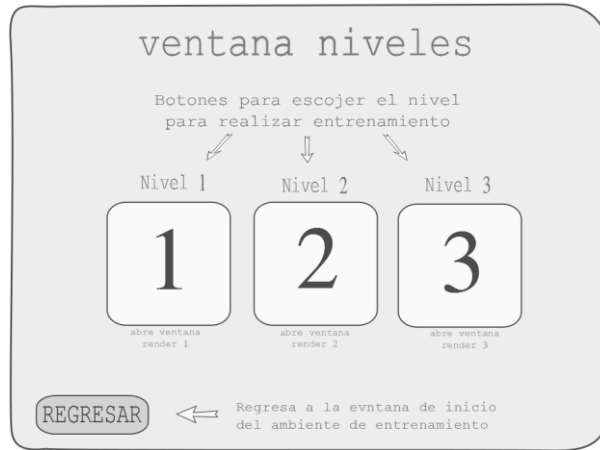


Figura 38 Diseño ventana de niveles del entrenador virtual (Imagen Propia)

4.5.3 Niveles de entrenamiento:

4.5.3.1 Nivel 1: Esta ventana está compuesta por un marco de renderizado y un botón que permite regresar a la ventana de selección de niveles. En el marco de render se podrá hacer la visualización de los ejercicios propuestos para el primer nivel [27], [28], Figura 39.

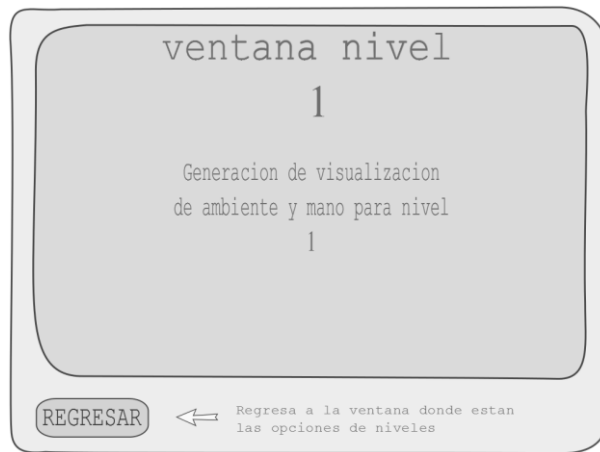


Figura 39 Diseño ventana Nivel 1 entrenador virtual (Imagen Propia)

4.5.3.2 Nivel 2: Esta ventana está compuesta por un botón el cual permite retornar a la ventana de selección de niveles, dos botones que permiten ejecutar el giro de la mano sobre el eje longitudinal haciendo uso de eventos presionar y liberar (pressed, release), los cuales consisten en ejecutar la función asociada mientras se tiene presionado el botón y terminarla cuando este es liberado [27], [28], Figura 40.



Figura 40 Diseño ventana Nivel 2 entrenador virtual (Imagen Propia)

4.5.3.3 Nivel 3: Esta ventana está compuesta por los mismos elementos del nivel 2, además de cinco botones correspondientes a la navegación, tres de ellos permiten el desplazamiento horizontal y el giro transversal de la mano y los dos restantes permiten el desplazamiento vertical. Estos botones también son manejados por eventos de presionar y liberar [27], [28], Figura 41.

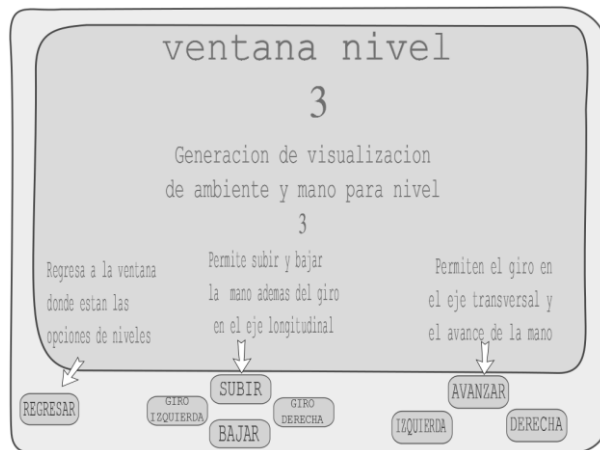


Figura 41 Diseño ventana Nivel 3 entrenador virtual (Imagen Propia)

De acuerdo con el modelo de interfaz planteado, se procedió a la implementación de este modelo en una herramienta software de desarrollo de interfaz, para este caso se utilizó Qt Y Qt Designer.

4.6 Introduccion a QT

QT es un entorno de trabajo basado en C++ para el desarrollo de software en plataforma cruzada. Junto a una amplia librería en C++, QT incluye las herramientas necesarias para escribir las aplicaciones de una forma muy rápida y sencilla. Estas librerías incluyen un conjunto de widgets (“controles” en la terminología de Windows) encargados de

proporcionar las funciones estándar de una interfaz o GUI. QT presenta una alternativa para la comunicación inter-objeto denominada “signals y slots” que reemplaza al antiguo sistema de callbacks utilizado por otros sistemas además de proporcionar un modelo de eventos convencional para la captura de las pulsaciones del ratón, teclas u otras entradas del usuario haciendo posible la creación de aplicaciones multiplataforma mediante funciones estándar, incluyendo drivers para Linux, Windows, e incluso drivers nativos para bases de datos como Oracle®, Microsoft® SQL Server, IBM DB2®, Borland [29].

4.6.1 Qt Designer: es un diseñador de interfaces gráficas para aplicaciones basados en las bibliotecas QT. Es posible escribir aplicaciones completamente en código fuente, o utilizando QT designer para un desarrollo más rápido.

Diseñar un archivo con Qt Designer es un proceso simple, consiste en arrastran los ítems desde un menú a un formulario, utilizando herramientas de edición para seleccionar, cortar, pegar y cambiar el tamaño de los diferentes componentes.

El desarrollo de la interfaz gráfica de usuario del ambiente virtual de entrenamiento se llevó a cabo en la plataforma de Qt como se mencionó anteriormente. A continuación se describe paso a paso las acciones que se llevaron a cabo en Qt para el desarrollo del entorno grafico de usuario GUI [29].

A partir del diseño se definieron dos ventanas principales, una ventana correspondiente a los datos del paciente y la otra al ambiente de entrenamiento.

En base al diagrama de fuerzas proporcionado por la unidad de fisioterapia de la Universidad del Cauca y el diseño del formulario se escogen los componentes de Qt que cumplen con las funciones propuestas anteriormente. Estos widgets son: botones, labels, Qline edit, radio buttons y combo box.

4.6.2 Qline edit: permite ingresar los datos de los pacientes por medio de eventos de teclado, se utilizaron 13 editores que permiten el ingreso de caracteres alfanuméricos por eventos de teclado [27].

4.6.3 Lay out Horizontal y vertical: Permiten alinear los diferentes elementos de la ventana [30].

4.6.4 Labels: permiten colocar nombres específicos a cada texto, para indicar los campos por ejemplo, el label ‘Nombre’ indica que el dato que se pide es el nombre [31].

4.6.5 Group box: Permite dividir la ventana en grupos de elementos para una mejor organización de esta y una redimensión proporcional de la ventana al cambiar de tamaño, se utilizaron dos de estos para diferenciar los datos personales y los datos estadísticos del paciente [31].

4.6.6 Spacers: permiten dar espacio y posicionar los elementos de la ventana, para una mejor disposición de los elementos en la ventana, ofreciendo una mejor visualización [31].

Finalmente para ajustar al tamaño de la ventana se utiliza la propiedad de Qt Lay out In a grid.

Se procedió a dar propiedades a los botones como salir, cancelar, guardar, mostrar lista para que se ejecutaran dichas acciones al presionar estos botones, en la Figura 42 se puede observar un ejemplo de la implementación de los elementos anteriormente nombrados en el FrameWork de Qt Designer.

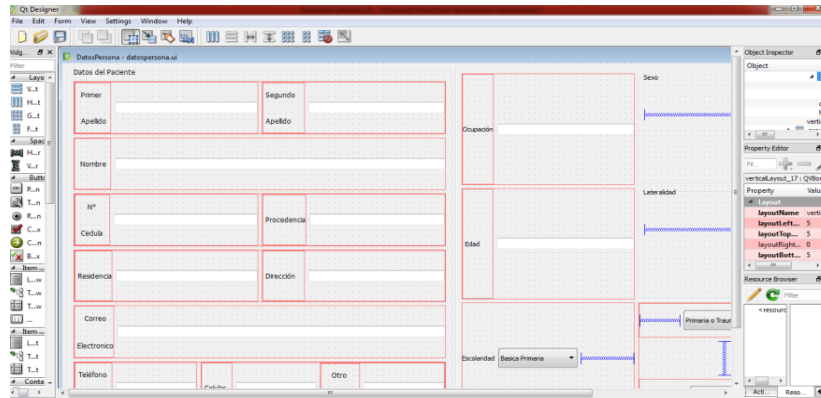


Figura 42 Diseño Formulario de Datos del paciente (Imagen Propia)

Después de realizar la interfaz en Qt designer se definieron los eventos asociados a los widgets mencionados anteriormente.

4.7 Estructura de almacenamiento de datos del paciente

Después de haber definido los datos relevantes del usuario, es importante almacenar estos datos y la mejor forma de hacerlo es implementando una base de datos, Qt posee una librería de base de datos SQLite, la cual permite la interacción con tablas exclusivas de Qt y al poder realizar una programación orientada a objetos se hizo la implementación de clases para la creación y manejo de la base de datos, a continuación se explica las clases que componen el módulo de almacenamiento de datos.

4.7.1 Clase Datos persona: Esta clase está compuesta por cuatro parámetros, cuatro eventos y un atributo, en ella se guardan los datos provenientes del formulario en un vector y se realiza la validación de los campos numéricos definiendo un límite de caracteres según la información requerida, por ejemplo para la edad se estableció un rango entre 1 y 150 años, además permite diferenciar entre un valor numérico o carácter; el cual permitirá ingresar los valores numéricos en los campos correspondientes a estos, el modelo de esta clase se puede observar en la Figura 43.

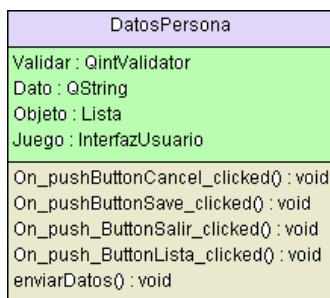


Figura 43 Clase Datos persona (Imagen Propia)

4.7.2 Clase Lista: Esta clase es la encargada del manejo de la base de datos que administrara los datos del paciente, Figura 44.

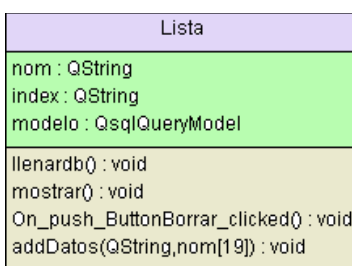


Figura 44 Clase Lista

Para el desarrollo de esta clase se implementaron funciones propias de SQL, por ejemplo, la función encargada de inicializar la base de datos ‘prepare’. Esta función carga la base de datos si está aún no ha sido asignada.

El funcionamiento de cada uno de los atributos se comprobó utilizando advertencias desplegadas en el terminal de Qt. Por ejemplo en la creación de la base de datos se programaron mensajes de error tales como ‘Fallo en conexión’, ‘Fallo en adición de datos a la tabla’, entre otros; esto facilito encontrar y corregir errores de programación en la clase Lista.

4.8 Requerimientos para el diseño del entorno virtual

Posteriormente se inició el desarrollo del entorno virtual de entrenamiento; como primer paso se realizó una asesoría con médicos Fisioterapeutas de la Universidad del Cauca, en donde se definió la estructura del ambiente, se especificó la cantidad de niveles y su respectiva dificultad, el tipo de objetos que deben aparecer en la escena y el acondicionamiento del ambiente.

Las recomendaciones de los médicos Fisioterapeutas se mencionan a continuación:

- La posición de la mano debe ser frontal a la perspectiva visual de la persona, para simular el realismo, ya que esta, es la tendencia natural de las personas al hacer agarre de objetos.

- Los niveles de entrenamiento deben permitir al paciente una mejoría de manera progresiva, con esto se plantearon tres niveles con distintos tipos de dificultad, contando además con el concepto del PhD(c). Ingeniero Cesar Quinayas de la Universidad del Cauca, autoridad en dicho campo.

Los niveles propuestos para la herramienta virtual de entrenamiento son:

Nivel 1: Este nivel corresponde a la apertura y cierre de la mano.

Nivel 2: En este nivel se ejercitan los movimientos identificados en el trabajo de grado [2] con diferentes objetos.

Nivel 3: En este nivel también se ejercitan los movimientos identificados con la posibilidad de navegar sobre un ambiente virtual, el cual permite la interacción con diferentes objetos.

Después de conocer los objetos con los cuales el usuario va a interactuar en la escena y el aspecto visual que se desea lograr se procedió a diseñar estos objetos en una herramienta de modelado 3D.

4.9 Modelado 3D de objetos

Blender es un software de modelado y animación 3D multiplataforma (Compatible con Mac Os x, Windows, Linux, Solaris, IRIX y FreeBSD). Con él se pueden crear escenas y vídeos generados por computadora, con muchas prestaciones y facilidades. En la parte izquierda de la Figura 45 se puede observar la interfaz de usuario de Blender en la cual se crea el modelo 3D y en la parte derecha de dicha figura se muestra un ejemplo gráfico creado con Blender [33].

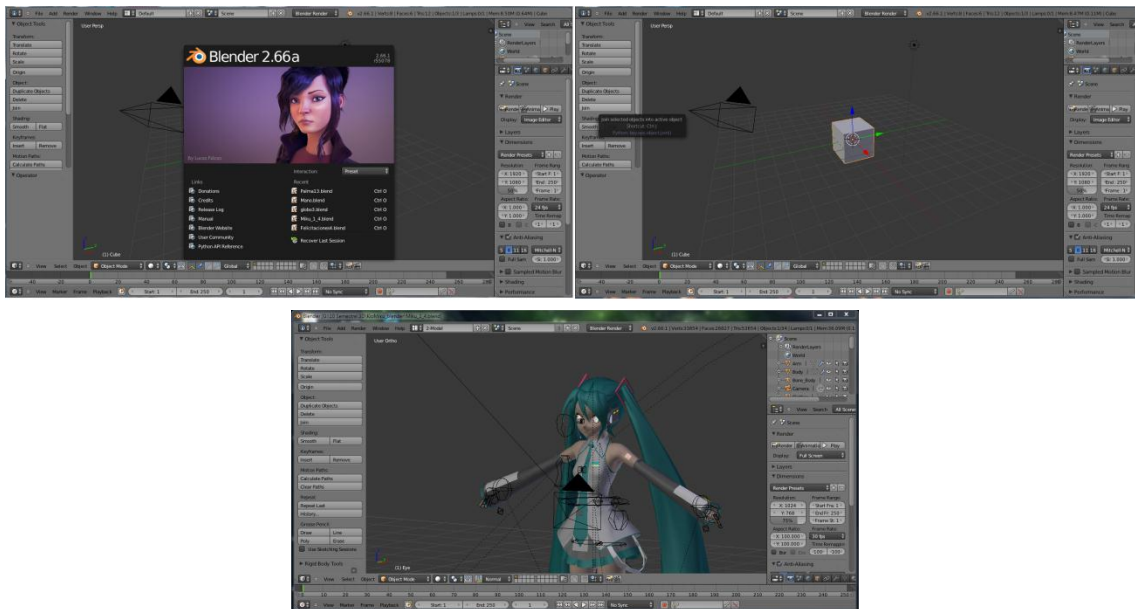


Figura 45 Capturas de imagen del software Blender. [33].

Después de haber hecho un proceso de familiarización con el software de modelado y haber identificado las funciones necesarias para el desarrollo, se inició con la fase de modelado de la mano el cual es el principal objeto de la escena y posteriormente se modelaron los demás objetos de la escena, el proceso llevado a cabo se especificara en los numerales 4.10 y 4.11.

4.10 Diseño de mano virtual

Se tomaron imágenes de diferentes manos ejemplificadas en la Figura 46, se hizo un proceso de selección y se escogió una imagen nítida que pudiera servir de muestra para la elaboración del modelo 3D.



Figura 46 Imágenes muestras de las Manos (Imagen Propia)

Después de seleccionar la imagen se inició con el modelamiento de la mano en la interfaz de Blender. Se cargó un cubo el cual se ubicó en la coordenadas (0,0,0) de Blender y se cargó la imagen de la mano seleccionada. Para poder hacer que la caja tome la forma de la mano se procedió a modificarla con las herramientas extruir, dividir caras, cortar caras y soldar vértices; también se hizo uso de la manipulación de las posiciones de vértices para modificar la apariencia de la maya.

A continuación se explica este proceso:

1. Se extruye el cubo en doce partes; en una matriz 3x4 donde los cuatro cubos consecutivos corresponden a los dedos de la mano y se realiza un extrucción más en la primera parte, la cual corresponde al dedo pulgar, esta matriz representa la palma de la mano, imágenes de dicho proceso pueden verse en la Figura 47.

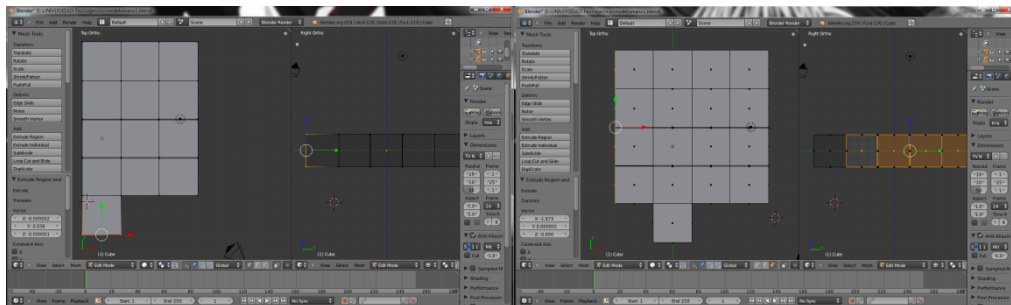


Figura 47 Extrucción inicial de palma de la mano (Imagen Propia)

2. Se procede a dividir la parte del dedo pulgar, en partes más pequeñas con el fin de lograr una edición de la mano mucho más fácil como se muestra en la Figura 48.

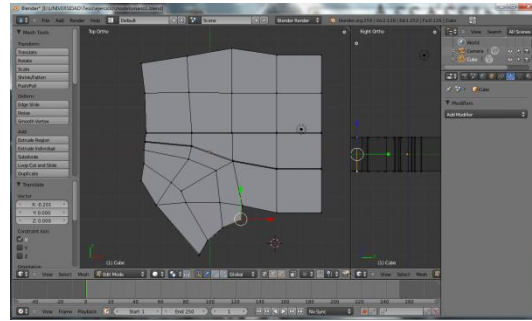


Figura 48 Detallado del dedo pulgar (Imagen Propia)

3. Del mismo cubo se extruye un cara correspondiente a un dedo, esto se repite varias veces con el fin de lograr las divisiones de las falanges del dedo, Figura 49.

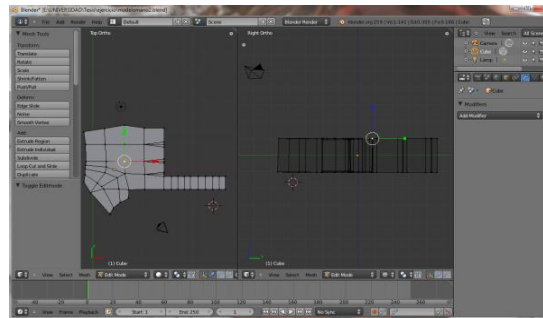


Figura 49 Extrucción inicial de dedos (Imagen Propia)

4. Teniendo la parte de la palma y el dedo se procede a dar pequeños cambios de posición en los vértices con el objetivo de dar una forma más natural a una mano real logrando un nivel de detalle como el de la Figura 50.

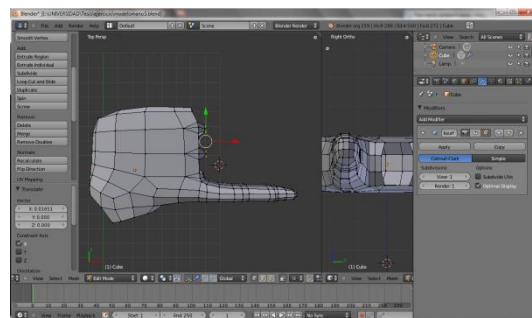


Figura 50 Detallado del modelado de dedos (Imagen propia)

- Después de tener un modelo general ajustado a la mano real se hacen divisiones en la malla para definir los detalles como son las uñas y los pliegues de flexión de la mano de la Figura 51.

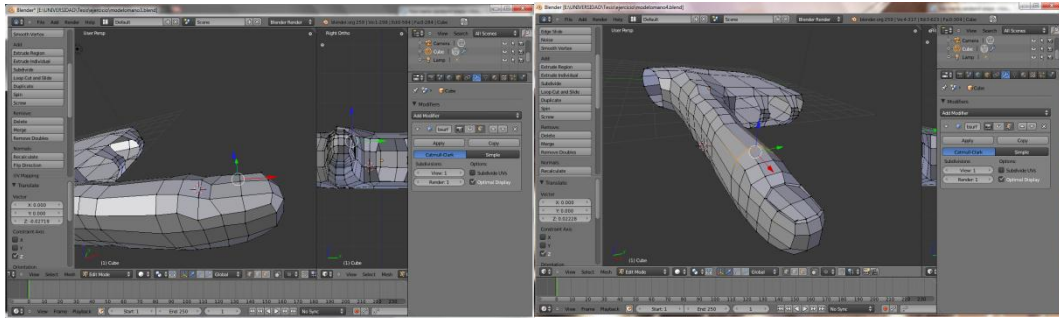


Figura 51 Modelado de uñas y pliegues de flexión (Imagen propia)

- Al obtener una forma definida del dedo se hace una copia de este para generar los dedos restantes que completan el modelo 3D para obtener un resultado como el de la Figura 52.

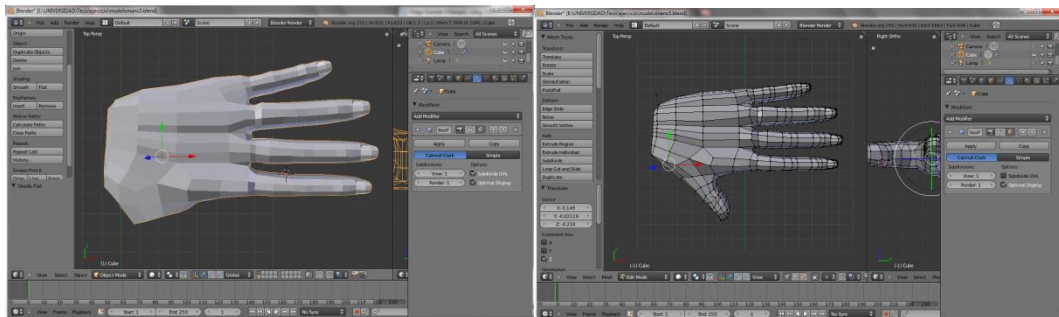


Figura 52 Modelado completo de dedos

Estando el modelo 3D completo, se dimensiona a la imagen muestra de la mano, mejorando la forma y la posición de la palma y los dedos, obteniendo así, el modelo final de la mano, mostrado en la Figura 53.



Figura 53 Dimensionado del modelo de la mano (Imagen Propia)

Cabe aclarar que en el modelamiento de la mano se evitó la generación de celdas triangulares ya que tienden a deformarse incorrectamente, por esta misma razón se prescindió del uso de cilindros en el modelado de los dedos ya que los puntos del cilindro

convergen al centro dando paso a la formación de triángulos, y dificultando la operación de suavizado de la maya, si en el modelo aparecieran caras triangulizadas al momento de exportar de Blender a archivos .OBJ, algunos vértices de los objetos quedarían desligados.

4.11 Diseño de escenarios

4.11.1 Nivel 1 Y 2: Ya que se deseaba un ambiente más cotidiano se pensó en un escenario basado en una sala de estudio en la cual se encuentran diferentes objetos tales como libros escritorios, etc. Se tomaron como referencias los siguientes escenarios contenidos en la Figura 54.



Figura 54 Escenarios base de diseño [34],[35].

A partir de estos escenarios se planteó un diseño para obtener un modelo 3D atractivo del ambiente y se definió la mejor distribución de los objetos, perspectiva general, colores e iluminación del ambiente.

4.11.2 Nivel 3: se definió un escenario basado en el plano de una casa ya que se dispone del espacio necesario para lograr la navegación y se puedan integrar distintos objetos con los que se puede interactuar, el plano que se escogió está en la Figura 55.

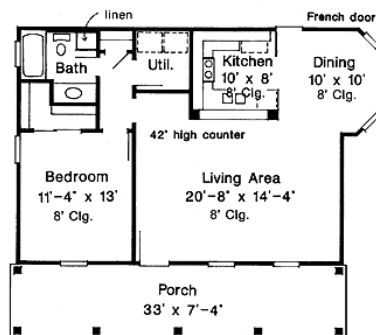


Figura 55 Plano del Nivel 3 [36]

Después de haber definido los ambientes para cada nivel se inició con el proceso de modelado de cada uno de estos en Blender; Haciendo uso de las herramientas y el proceso descrito en el modelamiento de la mano, para los niveles 1, 2 y 3 se

implementaron objetos geométricos tales como esferas, cubos, cilindros, etc; manipulándolos de forma tal que se asemejen a tazas, vasos, balones, juguetes, etc.

Estos modelos están hechos manteniendo la escala 1:1 con el fin de conservar la correspondencia del plano; para comprobar las longitudes del modelo se eligió la opción en Blender que permite la visualización de las caras que componen al objeto, a continuación se muestra una gráfica del modelo en la Figura 56.



Figura 56 Visualización del escenario virtual (Imagen Propia)

Luego de haber modelado el escenario, el siguiente paso es diseñar y ubicar todos los objetos que componen el ambiente, en este punto se diseñaron sillas, mesas, juguetes, tazas, vasos, etc. que simularan un entorno como el que se planteó inicialmente.

A continuación se muestra el primer prototipo de la escena en la Figura 57.

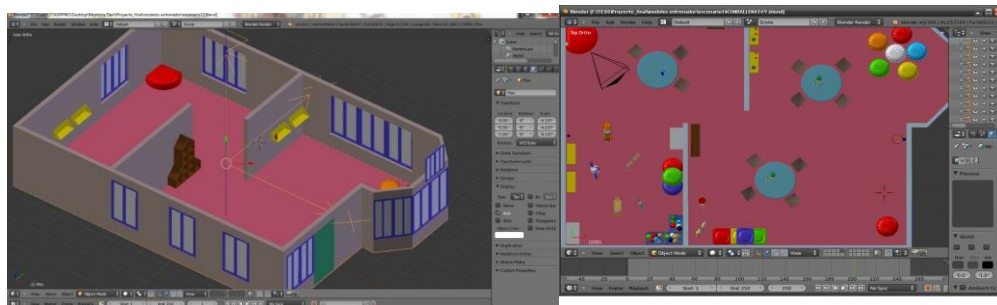


Figura 57 Modelo virtual del escenario (Imagen Propia)

Finalmente se definió el escenario, las acciones que se quieren desarrollar, las herramientas software que se van a utilizar (Qt para desarrollo de interfaces y VTK para la visualización gráfica 3D) y el paradigma de programación (Programación orientada a objetos). A continuación se hace una breve definición de las librerías de VTK utilizadas en el proceso de visualización del ambiente.

4.12 VTK

VTK (Visualization Toolkit) es un conjunto de bibliotecas desarrolladas por Kitware, contienen un sistema que permite la visualización de geometrías 3D, soportando una amplia variedad de algoritmos de visualización y modelado. Es una herramienta open

source que impulsa una gran difusión de la misma, extendiendo su aplicación a prácticamente todos los campos en los que se emplean objetos 3D entre los que se destacan la medicina y las herramientas industriales [37].

Las funciones que integran VTK se basan en los principios de orientación a objetos conformando un completo sistema jerárquico de filtros y fuentes que dotan de gran flexibilidad a todo el sistema [37]. VTK no consiste únicamente en un sistema de visualización sino que aporta una amplia gama de algoritmos para tratar las imágenes y objetos creados, permitiendo entre otras cosas el añadir texturas, aplicar tensores, reducir el número de polígonos, además de proporcionar una gran variedad de clases de datos para trabajar datos poligonales, imágenes, volúmenes y mallas [37].

Esta librería se utilizó básicamente para cargar y visualizar los objetos modelados y poder realizar una interacción con ellos, logrando una realimentación visual del ambiente de entrenamiento virtual.

4.13 Diagrama UML

Para proseguir con el desarrollo de la herramienta de entrenamiento, se realiza un diagrama UML, en el cual se segmenta todo el desarrollo de la interfaz en clases [38], este modelo junto con la definición de las diferentes clases se puede apreciar en la Figura 58.

Las clases que se definen son:

- **4.13.1 Interfaz de Usuario:** Esta es la clase principal la cual se encarga de manejar todos los eventos de teclado y ratón, además de la navegación entre ventanas, eventos de tiempo y procesos de renderización de VTK.
- **4.13.2 Objeto Actor:** permite cargar los archivos .OBJ (objetos que fueron modelados en Blender), generar el mapa de vértices y obtener datos de los polígonos de los objetos, además permite dar las propiedades de color, posición, desplazamiento y giro.
- **4.13.3 Detección Colisiones:** Esta clase es la encargada de realizar un proceso de detección de triángulos entre dos objetos, lo cual permite identificar los puntos en los cuales los objetos hacen contacto.
- **4.13.4 Cargar Mano:** En esta clase se llama al objeto contenido en una dirección de memoria, además realiza los diferentes ensambles de los objetos, ubica las posiciones iniciales, permite la implementación de funciones de desplazamiento y giro, y coloca los valores correspondientes al color asignado a cada objeto.
- **4.13.5 Cámara:** En clase cámara se configuraron las opciones de posición, punto focal, ángulo de elevación, el giro horizontal y vertical. Para lograr un ángulo de vista superior se configuro la función 'SetViewAngle' con valor en el eje y; la función Dolly se utiliza para establecer el ángulo de perspectiva.
- **4.13.6 Iluminación:** La clase iluminación se utilizó para darle las propiedades de iluminación de luz que se colocara en los diferentes escenarios, las propiedades que se definen en esta clase son: el color, la posición, el punto focal, la intensidad y el ángulo de cono.
- **4.13.7 Escenario:** En esta clase se hace el llamado de los objetos que componen el escenario principal en el nivel 3.
- **4.13.8 EscenarioN1N2:** En esta clase se hace el llamado de los objetos que componen el escenario principal en el nivel 1 y 2.

Las clases a continuación realizan un proceso similar de llamado a los objetos respectivos a los nombres de cada clase, los cuales componen el escenario en los niveles 1,2 y 3.

- Juguetes
- Mesa
- Dados
- Balón
- Cojín
- Monitor
- Mueble
- Objetos varios

A continuación se exponen algunas clases en las cuales se detalla cuáles son los objetos y funciones más utilizadas en el proceso de desarrollo del ambiente virtual.

4.14 Clase Objeto Actor

Esta clase está compuesta por cinco objetos y tres funciones exclusivos de las librerías de VTK, mostrados a continuación, Figura 59.

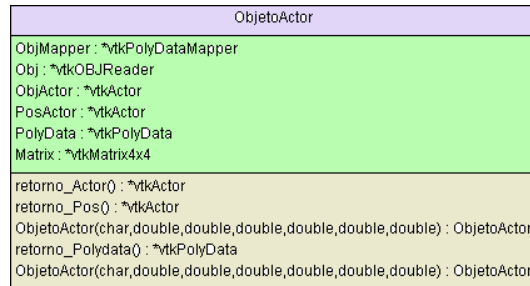


Figura 59 Clase Objeto Actor

4.15 Clase Cargamano

En esta clase se crean tres objetos privados, dos son agregados desde la clase objetoActor, con los atributos mencionados anteriormente y el objEnsamble, cinco objetos públicos, también se definen diez y siete funciones públicas y una privada el modelo de esta clase se puede observar en la Figura 60.

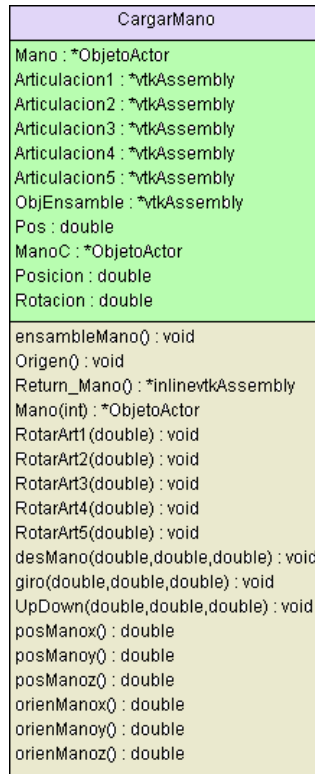


Figura 60 Clase Cargar Mano

4.15.1 Funciones de la clase Cargamano

4.15.2 EnsambleMano: Esta función permite el llamado del objeto tipo 'objActor' que fue definido como un vector de 15 posiciones, correspondientes a las 15 partes que componen el modelo de la mano, al cual se le transfiere la posición de memoria donde está el archivo.OBJ, y atributos como posición inicial y color [41].

En esta función se implementa el proceso de ensamble de las partes que componen el modelo de la mano, se adicionan los objetos tipo actor según el orden en que se quiere ensamblar el modelo completo, el primer objeto asignado al ensamble será el primer hijo, como ejemplo se explica el proceso de ensamble de un dedo.

1. El objeto el cual será el primer hijo es la parte de la falange que contiene la parte de la uña y la yema del dedo, este objeto es el primero que se asigna a la primera posición del vector ensamble.
2. En la segunda posición del vector ensamble se coloca la segunda falange, la parte media del dedo esta parte se convierte en el padre del objeto agregado al vector ensamble en la primera posición.

3. Por ultimo en la tercera posición del vector ensamble se adiciona la falange que va unida a la palma, está a su vez se convierte en el padre del objeto que está en la segunda posición del vector ensamble.

De esta forma se logra obtener el modelo ensamblado de los demás dedos y la palma de la mano. Dentro de esta función se hace un llamado a la función origen, explicada a continuación:

4.15.3 Función Origen: Esta función permite calcular los límites máximos y mínimos de cada objeto y definir estos valores como el punto de origen de rotación del objeto, para este fin se utiliza `retorno_Pos` que es una función de la clase `ObjetoActor` la cual permite encontrar los valores de posición y rotación del objeto, estos valores se guardan en el vector tipo `'double'` de seis posiciones correspondientes a los valores máximos y mínimos del objeto, obtenidos con la función `'GetBounds'` implementada en la librería `vtkActor` [39].

Se obtienen los límites máximos y mínimos del objeto actor y se guardan en un vector de seis posiciones utilizando la función `'GetBounds'`, estos valores son ubicados como punto de origen del ensamble correspondiente al Actor al cual se le encontraron los límites, para esto se utilizó la función `'SetOrigin'` de VTK. De la misma manera se realiza el ensamble de los demás objetos del modelo de la mano [39].

4.15.4 RotarArt: Se definieron cinco funciones de este tipo, las cuales reciben como parámetro un valor `'double'` correspondiente a un ángulo de giro, el cual define el valor de rotación de cada articulación; todas las articulaciones de cada dedo son programadas para rotar la misma cantidad de grados en el mismo instante, así se obtiene un movimiento independiente de rotación por cada dedo [41].

4.15.5 desMano: esta función recibe tres valores tipo `'double'` que corresponden a los valores de los ejes `x,y,z`; los cuales determinan la posición del objeto, esto se logra con la función `'SetPosition'` de VTK, esta función es aplicada al objeto ensamblado [42].

4.15.6 Giro: Esta función recibe tres valores que corresponden al ángulo de giro en los ejes `x,y,z` del objeto ensamblado, este cambio de rotación se realiza implementando la función `'SetOrientation'` de VTK [42].

4.15.7 upDown: Esta función es la que se encarga de manejar el ascenso y descenso en el eje transversal del objeto ensamblado, Se realiza implementando la función `'SetPosition'` de VTK [42].

4.15.8 CargaMano: Esta función es de tipo `'ObjetoActor'` y permite el retorno de las propiedades de un objeto desde la clase `Actor`, recibe como parámetro un valor entero correspondiente al actor con el cual se quiere operar [39].

4.15.9 posMano: Se definen tres funciones correspondientes a los ejes `(x,y,z)`; las cuales permiten obtener los valores de posición actuales del objeto ensamblado [42].

4.15.10 orienMano: También se definen tres funciones correspondientes a los ejes (x,y,z), los cuales permiten obtener los valores de rotación actuales del objeto ensamblado [42].

Se implementó un programa que permite la visualización de un objeto, en este, se carga cada una de las partes del ensamble de la mano para observar que el modelo de la mano coincidiera con la imagen de la mano de muestra.

Además para comprobar el correcto funcionamiento de esta clase se implementó un programa compuesto de una ventana de renderizado y 'slides' los cuales permiten cambiar los valores de posición y giro de la mano, para poder observar el comportamiento de la posición del modelo de la mano se adiciono un modelo 3D estático, que sirvió como punto de referencia para poder verificar el desplazamiento en los diferentes ejes, como se puede observar en la Figura 61.

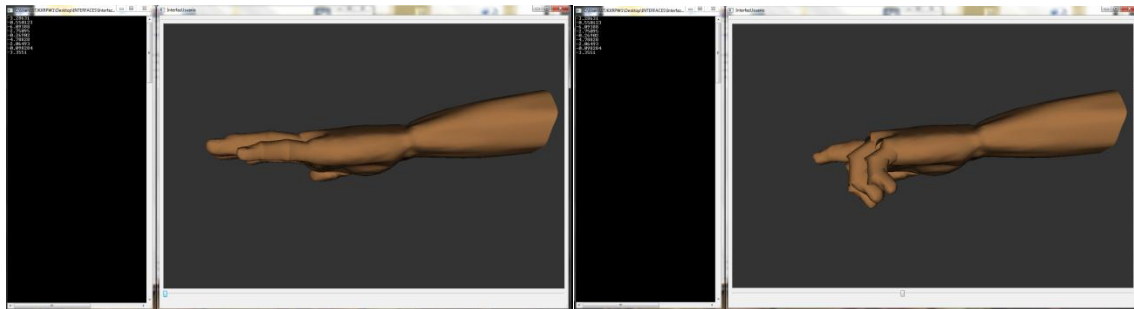


Figura 61 Orientación de los dedos de la mano (Imagen propia)

4.16 Clase Cámara

En esta clase se operan los ángulos de giro horizontal y vertical, la elevación, el punto focal, el desplazamiento y la perspectiva de la cámara [43] que se va a implementar en el ambiente virtual de entrenamiento, el modelo de esta clase se puede observar en la Figura 62.

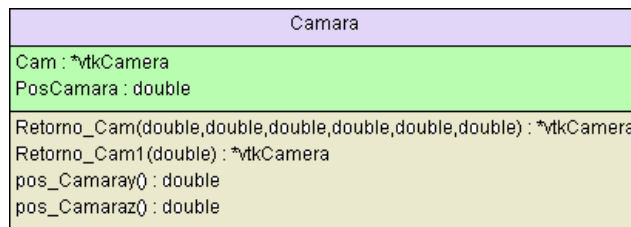


Figura 62 Clase Cámara

4.17 Clase Iluminación

Esta clase está compuesta por dos objetos y dos funciones de tipo vtkLight, en el constructor de esta se definen las propiedades que tendrá el objeto vtkLight por defecto, como el color, posición, punto focal e intensidad de la iluminación, se crean dos objetos ya que el ambiente diseñado para los niveles 1 y 2 es diferente al nivel 3, por lo tanto los valores de iluminación deben variar conforme al ambiente [44] Figura 63.



Figura 63 Clase Iluminación

Se realizó un programa para comprobar los valores más convenientes de intensidad, foco, posición y ángulo de área de iluminación para mejorar el aspecto visual del ambiente; además se hicieron pruebas en el color de la iluminación, las cuales no están implementadas en el este programa. En la Figura 64 se pueden observar diferentes configuraciones en la iluminación.

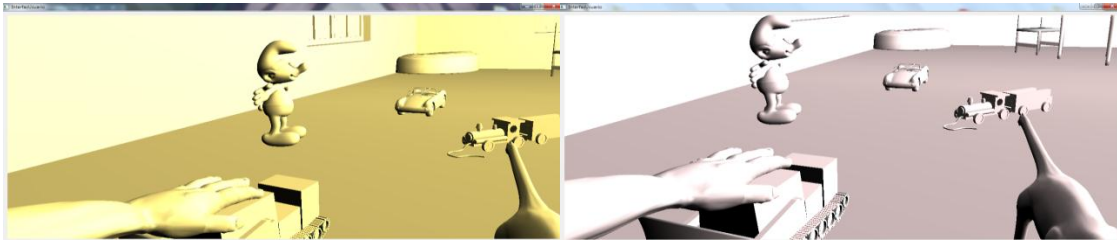


Figura 64 Configuración de la Iluminación (Imagen propia)

4.18 Clase Detección de Colisiones

Esta clase permite identificar las celdas que componen la maya de los diferentes objetos los cuales en algún proceso de desplazamiento hacen contacto, para poder implementar correctamente esta clase, las mallas de los objetos que se cargan deben estar triangulizadas, para cumplir con este requerimiento en el software de modelado Blender se habilitan las siguientes opciones:

- Triangle face
- writeNubs
- Include Normals

Con esto se logra triangulizar las caras del objeto además de exportarlo suavizado y en alta definición.

Esta clase contiene un objeto tipo VCollide, un objeto VCRReport y dos variables tipo entero y cinco funciones, el modelo de esta clase se puede observar en la Figura 65.

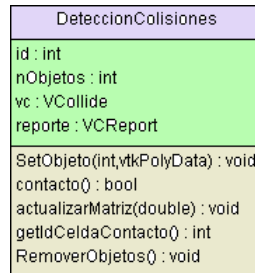


Figura 65 Clase Detección de Colisiones

4.19 Clase InterfazUsuario

Esta es la clase más importante , ya que es la encargada de manejar todos los eventos de teclado, ratón y de tiempo requeridos para el control de la interfaz del ambiente virtual; también se implementan las funciones necesarias para el proceso de visualización y renderizado en VTK, a esta clase se agregan todos los objetos de las diferentes clases expuestas anteriormente, los cuales componen el ambiente virtual, igualmente se realiza la implementación y llamado de las funciones que permiten la generación de movimiento de las diferentes partes que componen el modelo de la mano virtual, el modelo de esta clase se puede observar en la Figura 66.

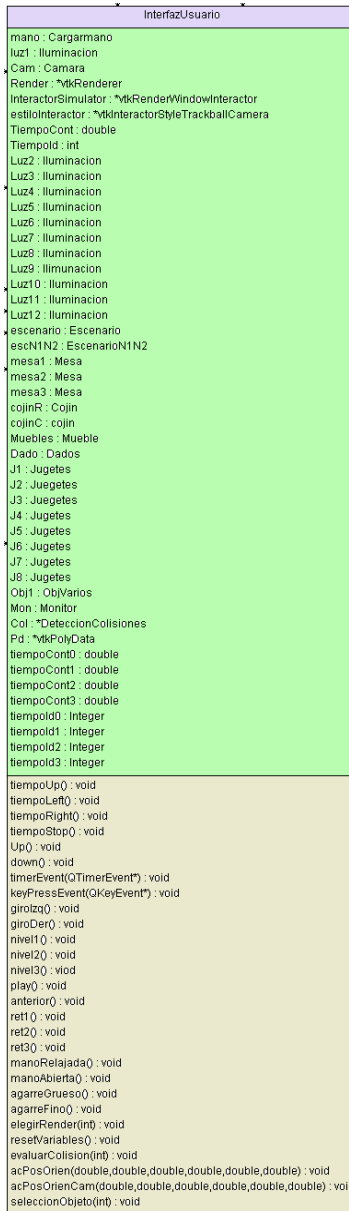


Figura 66 Clase Interfaz de Usuario

5. PROTOCOLO DE ENTRENAMIENTO

Este protocolo se establece con el propósito de determinar una guía para el manejo del entrenador virtual por pacientes con amputación de mano; se fundamenta en las descripciones médicas suministradas por coordinadores en el área de fisioterapia de la Universidad del Cauca.

Cuando sucede una amputación y se pierde una parte del cuerpo en este caso, la mano; inmediatamente después de la amputación se inicia una etapa de tratamiento.

El tratamiento de un paciente con amputación requiere de un equipo multidisciplinario involucrado en el proceso de rehabilitación, desde el médico quien realiza la valoración respectiva del paciente, siguiendo por el fisioterapeuta, hasta el técnico ortopedista.

La función del fisioterapeuta en este proceso de rehabilitación es ejecutar la terapia prescrita, valorar los progresos del paciente, instruir sobre las técnicas de rehabilitación y colocación de la prótesis; así mismo el técnico ortopedista tiene la función de construir la prótesis de acuerdo a las características prescritas, realizar los ajustes necesarios en la prótesis y reunirse con el equipo médico para la valorar conjuntamente el proceso de rehabilitación.

En las visitas realizadas al personal de fisioterapia de la Universidad del Cauca, se definieron algunas particularidades que se manifiestan posterior a la amputación de miembro superior como son: Síndrome de dolor y miembro fantasma, Manejo de control central y propiocepción [5].

5.1 Síndrome de dolor fantasma: El dolor del miembro fantasma podría estar relacionado con una memoria de dolor somatosensorial [6]. Es posible que los factores periféricos sean responsables de esta memoria.

5.2 Factores periféricos: Hace referencia a las sensaciones dolorosas de espasmos y compresión en un miembro fantasma reflejadas en la tensión muscular del miembro residual, estos cambios en la tensión muscular se hacen evidentes en una electromiografía [6].

5.3 Síndrome del miembro fantasma: Este fenómeno ocurre debido a que el cerebro retiene imágenes sobre el miembro superior antes de ser removido, dejando presente la percepción sensorial después de la amputación. El síndrome de miembro fantasma se puede presentar en los pacientes inmediatamente después de la amputación, ya sea accidental o programada, posterior a la amputación o inclusive su aparición puede tardar muchos años [6].

La sensación del miembro fantasma es una sensación no dolorosa de la continuidad del miembro amputado, lo que genera en el paciente gran incomodidad, fastidio y malestar psicológico; además este fenómeno afecta el control central y el manejo de distancia.

En la asesoría que se realizó con el personal del departamento de Fisioterapia se hicieron aclaraciones importantes para tener en cuenta como preámbulo para el desarrollo de un protocolo de entrenamiento, en estas consultas se nombraron los fenómenos ya señalados, uno muy importante es el manejo de control central, el cual hace mención a la discapacidad del paciente de ubicarse en el espacio debido a la sensación presente de un miembro ahora inexistente en él; esto conlleva a la frustración de la persona al realizar movimientos inexactos, en consecuencia la rehabilitación del paciente incluye además de la etapa inicial de control del dolor, manejo de control central y distancia y algunos ejercicios de propiocepción [5].

Después del tratamiento médico para el dolor del paciente y mientras el segmento amputado que tiende a perder tamaño se moldea, la persona afectada debe someterse a por lo menos 15 sesiones de fisioterapia para trabajar en la etapa inicial de rehabilitación, trabajando en el control central, el cual presenta deterioro debido al síndrome del miembro fantasma manifestado después de la pérdida de la mano.

Estas sesiones serán realizadas por el equipo médico, quienes además capacitaran al paciente para el desarrollo de los ejercicios en el entrenador virtual propuesto en este trabajo de grado.

La primera etapa de estas sesiones consta de la implementación de sesiones de sensibilización y fortalecimiento del muñón. Posteriormente a una amputación, la parte donde sucedió la amputación llamada muñón va cambiando de volumen hasta tomar una forma final, en este proceso al paciente no se le puede implantar una prótesis aún, así que se estima un tiempo de por lo menos un año para la adaptación de la prótesis.

Después de que el paciente realice en conjunto con el personal médico las actividades mencionadas anteriormente, se procede a la ejecución de un protocolo para el manejo de un entrenador virtual diseñado para aumentar el potencial bioeléctrico en pacientes con amputación de mano, el cual se basa en las pautas de los médicos fisioterapeutas de la Universidad del Cauca.

5.4 Consideraciones a tener en cuenta

- El diseño del ambiente virtual se planteó para personas con amputación por debajo del primer tercio del codo.
- El paciente a tratar debe tener no más de 2 años de haber sufrido la amputación de la mano.
- El paciente con amputación debe haberse sometido previamente a los tratamientos de rehabilitación de muñón, manejo de control central y propiocepción.

5.5 Procedimiento

5.5.1 Preparación de la piel: la calidad de la medición de una señal EMG depende en gran parte de una preparación apropiada de la piel y posición de los electrodos, la estrategia principal de la preparación de la piel es establecer un contacto de baja impedancia con los electrodos. No existen reglas generales para este procedimiento, hay muchas posibilidades de buscar una buena condición de la piel para la medición de las señales EMG. Una consideración importante es probar la condición de la parte donde se van a tomar las señales, mediante pruebas en las que el paciente realice diferentes movimientos y el medico fisioterapeuta detecte por observación o palpación las zonas de mayor potencial. Los pasos que se deben desarrollar son:

- ✓ Remover el cabello: Es necesario para mejorar la adhesión de los electrodos, especialmente bajo condiciones húmedas, tipos de piel sudorosa o condiciones de mucho movimiento.
- ✓ Limpieza de la piel: El uso de esponjas para baño permite remover las células muertas de la piel, quienes producen alta impedancia también se utiliza alcohol puro frotado sobre la superficie donde se van a colocar los electrodos.
- ✓ Comprobación de las condiciones: Una forma de comprobar la adecuada condición de impedancia de la zona en la que se va a poner el electrodo es observar la piel con un leve tono rojo lo cual indica la baja impedancia de la zona seleccionada de la piel [8].

5.5.2 Colocación de electrodos

- ✓ Los electrodos de gel tienen mejores valores de impedancia con la piel.
- ✓ El uso de electrodos pequeños incrementa la selectividad de las mediciones, evitando el fenómeno de 'cross-talk' [8].
- ✓ Seleccionar la distancia inter electrodo lo más cerca posible para incrementar la selectividad.
- ✓ Es importante localizar el par de electrodos en la posición central del musculo en el cual se van a medir las señales EMG (evitar colocar los electrodos sobre diferentes músculos) debido a que por el encogimiento y alargamiento de la piel, cambia la posición del electrodo.
- ✓ Usar un sistema de mapeo del sitio de la colocación de los electrodos en lugares específicos de la anatomía de la persona, como se puede apreciar en la figura 67.

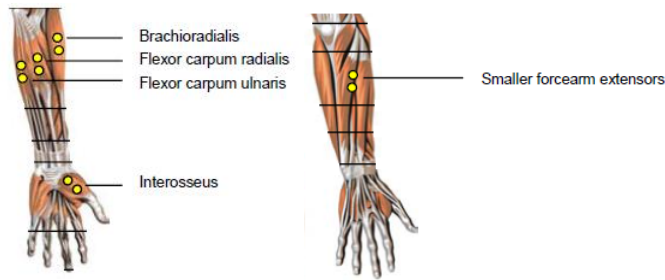


Figura 67 Posiciones Anatómicas de colocación de electrodos superficiales, vista frontal y frotal y dorsal [8]

- ✓ Electrodo de Referencia: Se necesita colocar al menos un electrodo neutro de referencia, normalmente este electrodo no es afectado eléctricamente pero se coloca cerca del área seleccionada en áreas tales como articulaciones y zonas óseas.

5.5.3 Procedimientos de verificación: A partir de la correcta colocación de los electrodos, se realizan pruebas para validar la señal EMG, en cada canal, para ello se hace una inspección visual de las señales en una interfaz computacional, se sugiere que el paciente relaje sus músculos para inspeccionar el desplazamiento de la referencia 'Baseline Offset' debido a que dependiendo del paciente el valor medio de la señal se desplaza y se requiere poner este valor en cero para adquirir valores correctos de amplitud de las señales EMG.

Después de que el paciente haya realizado las actividades en conjunto con los fisioterapeutas que lo ayuden con el manejo de distancia que ahora es diferente debido a la ausencia de su mano, procedemos a definir la parte del protocolo para el diseño del entrenador virtual de manera que este sirva para mejorar la adaptación del paciente a su prótesis.

Se determinaron en ese orden tres niveles a manera de un juego:

Primer nivel del entrenador virtual: El usuario intentara ejecutar movimientos de abrir y cerrar la mano, a través de un proceso de realimentación visual desplegado en la primera ventana del ambiente virtual, estos ejercicios se deberán ejecutar cierto número de veces a intervalos de tiempo definido, este proceso deberá realizarse de forma incremental hasta que el paciente logre un control de los movimientos propuestos.

Segundo Nivel del entrenador virtual: En este nivel el paciente visualizara la interfaz de un espacio cotidiano como un estudio donde el usuario se encontrara con distintas clase de objetos para los cuales deberá intentar generar algún tipo de agarre de manera que la mano virtual realice los agarres pertinentes para una correcta visualización en el movimiento que se realiza virtualmente.

En este nivel se describen cuatro agarres, los cuales fueron definidos en [1] como son mano relajada, agarre grueso, agarre pinza y mano abierta.

Para poder simular estos movimientos se diseñaron objetos con los que regularmente se utilizan este tipo de agarres, como se mencionara a continuación.

En la figura 68 se muestran algunos objetos para ejercitar el agarre grueso.



Figura 68 Objetos diseñados para el agarre grueso (Imagen Propia)

De la misma manera se diseñaron objetos para los demás tipos de agarres mencionados.

De esta manera se espera que el paciente ejercite los movimientos definidos para captar las señales electromiográficas y lograr que estas se intensifiquen para la posterior adaptación a una prótesis robótica accionada por medio de las señales EMG.

Después de que el paciente ha experimentado con el entrenamiento básico se plantea un último nivel.

Tercer Nivel del entrenador virtual: Este nivel se diseñó como un ambiente tridimensional en el que el paciente se va a poder desplazar y va a poder interactuar con distintos objetos que se encuentran distribuidos en el ambiente, además que el paciente podrá ejercitar los movimientos que se definieron en el nivel anterior.

Este nivel se planteó como una alternativa para que el paciente tenga acceso a una navegación a través de un ambiente virtual que estimula al paciente visualmente para realizar los ejercicios.

6. RESULTADOS

Después de haber realizado las pruebas correspondientes a la obtención, clasificación e identificación de las señales EMG, diseño del ambiente virtual y desarrollo del software se obtuvieron los resultados correspondientes a estos, los cuales se exponen a continuación en el orden anteriormente señalado.

Como se mencionó en el capítulo II 'Desarrollo de Hardware', se realizaron distintas etapas para la obtención, clasificación e identificación de señales EMG, como se ve en la Figura 69.

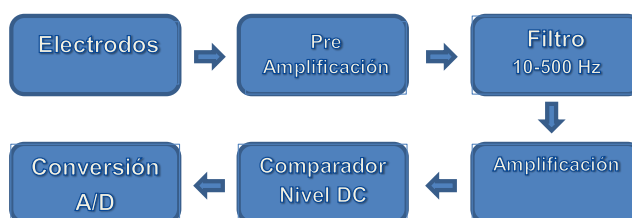


Figura 69 Etapas desarrolladas para la adquisición de señales EMG

6.1 Electrodo: Se realizaron pruebas conectando los electrodos a cables blindados UTP, con los cuales se captaron las señales EMG poco legibles ya que se veían afectadas por el ruido externo; posteriormente se utilizaron cables especiales para capturar las señales electromiográficas, con los cuales se logró registrar una señal más limpia, en la Figura 70 se observan los dos tipos de cables usados.



Figura 70 Cable UTP, Cable especial para señales EMG

6.2 Pre Amplificación: En esta etapa se realizaron pruebas con los amplificadores de instrumentación INA128p y AD620, obteniendo resultados similares debido a sus características, se decidió utilizar AD620 ya que fue de más fácil adquisición

6.3 Filtro pasa bandas: Se realizaron pruebas con un filtro activo y un filtro pasivo y se compararon las señales obtenidas con cada uno; se observó que con el filtro activo la señal EMG es muy afectada por el ruido externo y muy sensible a los cambios de movimiento de la mano lo cual ocasionaba una saturación en la señal de salida, obteniendo valores de los milivoltios a voltios, lo cual causaría un daño en la tarjeta en la

cual se realiza la conversión y el procesamiento de la señal, por lo tanto se escogió el filtro pasivo, ya que con este se obtenían mejores resultados. Se obtuvieron señales legibles en el espectro de rangos de frecuencia de interés (10-500 Hz), atenuando las señales fuera de este rango con un rango de voltaje estable sin importar la fuerza de ejecución de cada movimiento.

6.4 Amplificación: En esta etapa se probaron diferentes ganancias para las dos etapas de amplificación que se diseñaron, obteniendo como las ganancias más óptimas 58 para la amplificación de la primera etapa y de 10 para la segunda. Con estas ganancias y en esta configuración se logró disminuir el ruido de la señal y se obtenía una señal más legible, como se ve en la Figura 71.

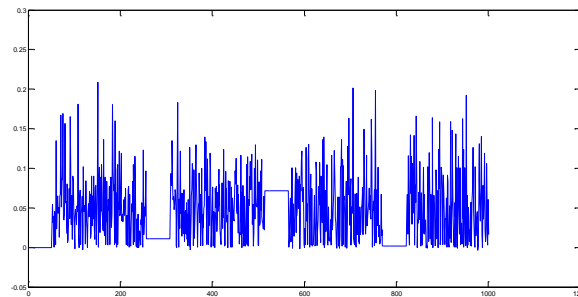


Figura 71 Obtención de Señal amplificada (Imagen propia)

Como se puede observar se pierde información en la parte correspondiente a los valores negativos de la señal y la señal EMG no alcanza a ser representada en su totalidad, esto se debe a la conversión analógica digital de la tarjeta ya que esta no está diseñada para convertir valores negativos. Para resolver este inconveniente se implementó un amplificador operacional en configuración Comparador- Amplificador.

Comparador: Esta etapa consta de un amplificador operacional en configuración comparador, en la entrada positiva del amplificador se introduce una señal DC constante de 0.4V, el cual es usado para subir el nivel DC de la señal EMG para que la conversión ADC de esta permita el registro de la señal EMG completa, observar Figura 72.

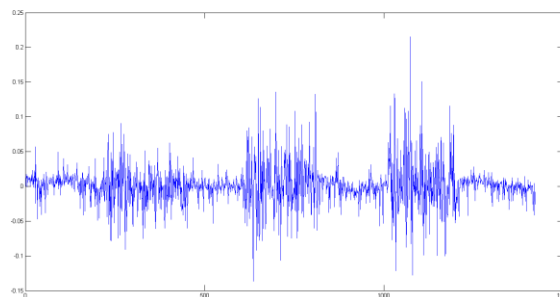


Figura 72 Señal completa EMG en Intervalos de diferentes movimientos (Imagen propia)

Con esto se obtuvo el diseño y el montaje completo del circuito, como se ve en la Figura 73 y 74 respectivamente.

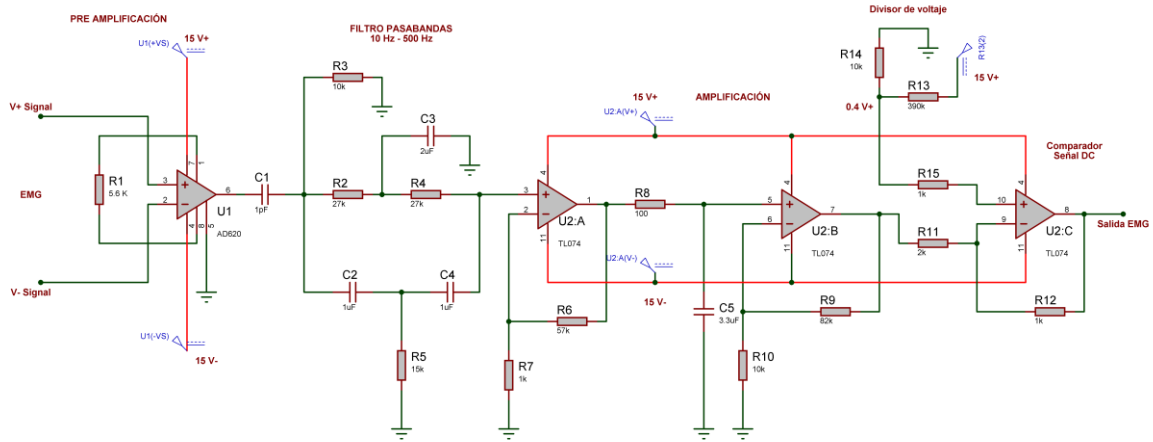


Figura 73 Circuito Adquisición de Señales EMG (Imagen propia)

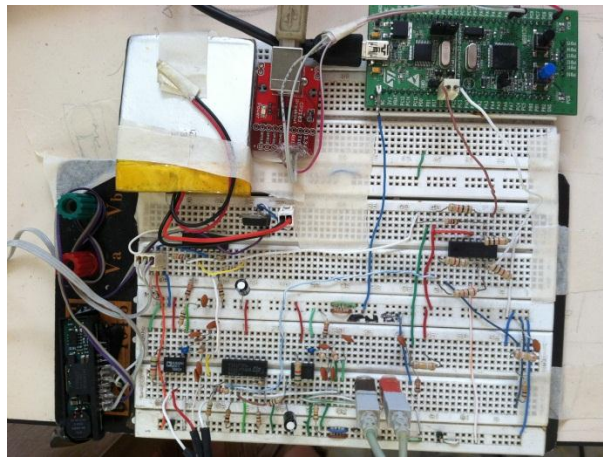


Figura 74 Implementación circuito de adquisición de señales EMG (Imagen propia)

Base de datos de señales EMG obtenida: Se realizó la elaboración de una nueva base de datos de señales EMG, utilizando el circuito de adquisición de señales implementado en este trabajo de grado, tomando 5000 muestras por sesión para cada canal para entreno y prueba de la red neuronal, en este orden, en la Figura 75 se puede observar la imagen de un segmento de la base de datos obtenida.

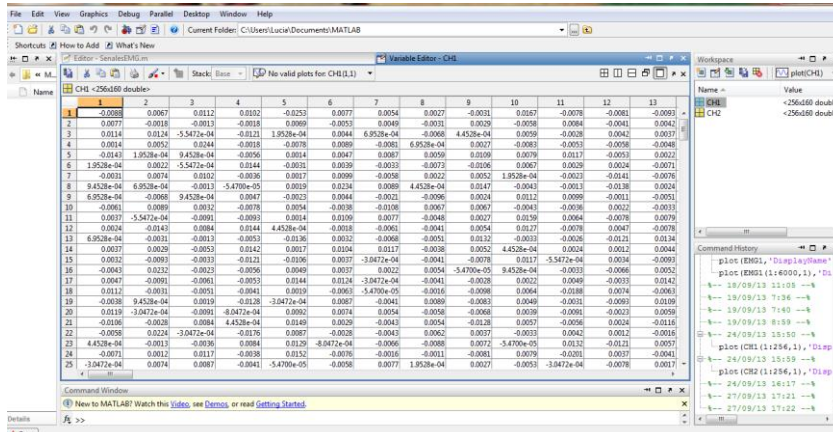


Figura 75 Base de datos de señales EMG obtenida (Imagen Propia)

Con la base de datos obtenida se realiza la extracción de características de dichas señales y se obtiene una matriz de características como se muestra en la Figura 76.

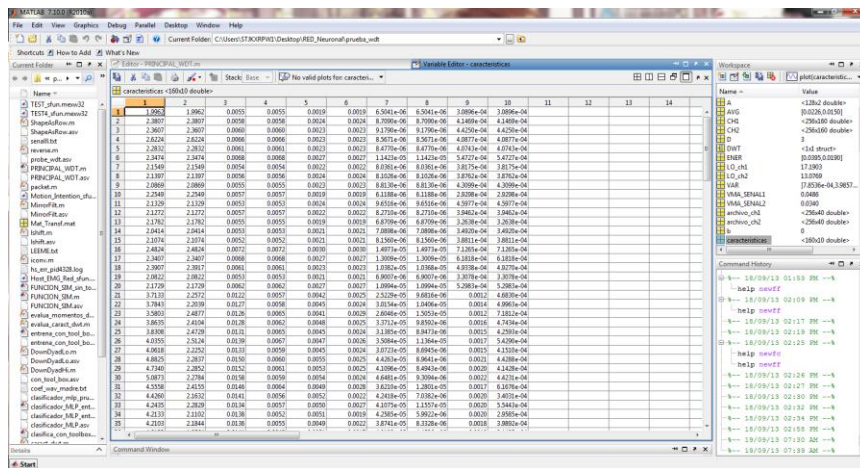


Figura 76 Matriz de características (Imagen propia)

Finalmente se realiza la extracción de las características correspondientes a longitud de onda, AVG, valor medio, energía y varianza de la señal EMG para los canales CH1 y CH2; utilizando la base de datos creada en este trabajo de grado, como se observa en las Figuras 77, 78, 79; en azul se muestran las características de la señal del Canal CH1 y en verde se muestran las correspondientes al canal CH2.

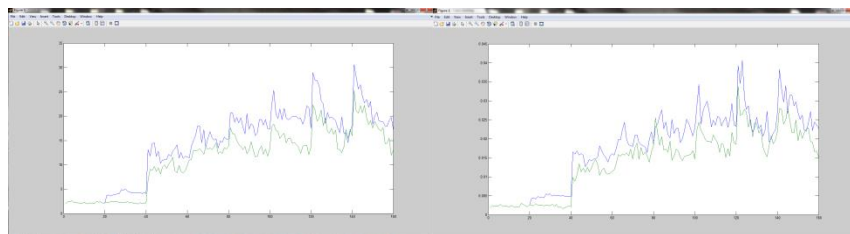


Figura 77 De izquierda a derecha Longitud de Onda, AVG (Imagen Propia)

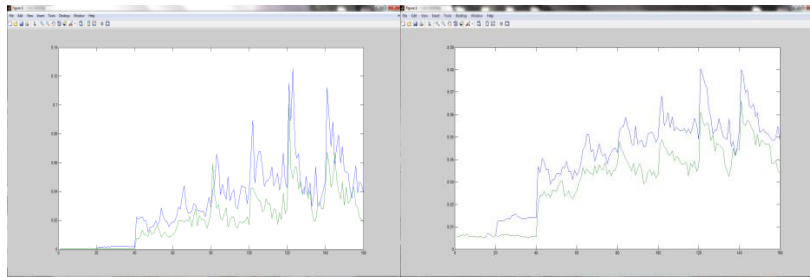


Figura 78 De izquierda a derecha Energía, Valor Medio (Imagen Propia)

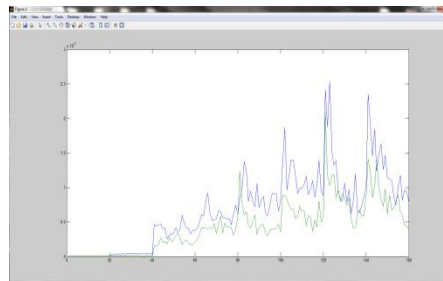


Figura 79 Varianza (Imagen Propia)

6.6 Resultados de la Red Neuronal: Después de obtener la matriz con las características de las señales EMG tomadas de la base de datos nueva, se procede a realizar el entrenamiento de la red Neuronal, realizada en [2], en la figura 80 se puede observar el proceso de entrenamiento de la red

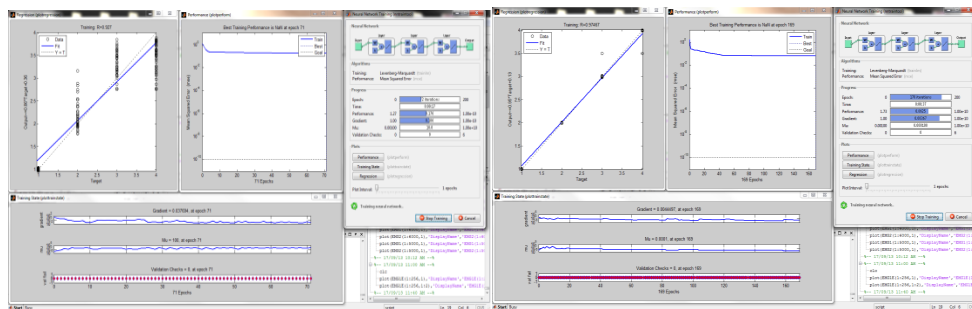


Figura 80 Entrenamiento de la Red Neuronal (Imagen Propia)

En la Figura 81 se observa el resultado de entrenamiento de la red con un error de 5.189%, logrando el alcance de la meta (goal) de la red, lo que significa que el proceso de entrenamiento fue satisfactorio.

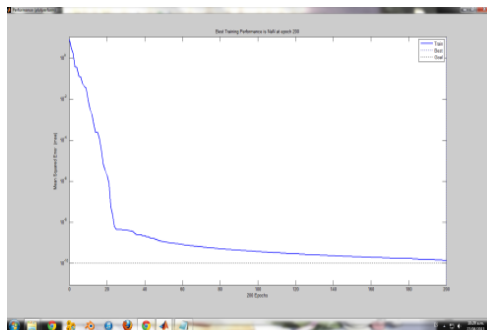


Figura 81 Resultado del entrenamiento de la Red (Imagen Propia)

Se obtuvieron las matrices de peso correspondientes a la red entrenada y se embebieron en las tarjetas utilizadas, aunque el proceso de entrenamiento fue satisfactorio no se logró obtener un resultado sobre la identificación de la intención de movimiento a partir de señales EMG tomadas del antebrazo de la persona ya que los datos que se obtuvieron cambian muy rápido y no se mantiene una respuesta constante mientras se está ejecutando el movimiento.

6.8 Resultados Software

De la misma forma en el desarrollo del software se obtuvieron diferentes resultados, con respecto a la base de datos del usuario; se logró un formulario el cual permite la obtención de los datos relevantes del paciente en los campos correspondientes diferenciando entre datos alfanuméricos y datos numéricos, además permite escoger opciones en barras desplegadas y finalmente guardar la información de cada paciente en una tabla de datos. En la Figura 82 y 83 se muestran el formulario y la tabla que se plantearon respectivamente.

Datos del Paciente

Primer Apellido Segundo Apellido

Nombre

Nº Cedula Procedencia

Residencia Dirección

Correo Electronico

Teléfono Casa Celular Otro (Indiqué)

Ocupación

Sexo Femenino Masculino

Edad

Lateralidad Izquierda Derecha

Causa de Amputación

Escolaridad

Tipo de Amputación

Lista Salir Cancelar Guardar

Figura 82 Formulario Datos del paciente (Imagen propia)

Id	Cedula	PrimerApellido	SegundoApellido	Nombre	Procedencia	Celular	Correo	Dir
1	1061710741	Segura	Cometa	Lucia	Valle	3013076422	lucia.sgr@gmail...	Calle 4A...

Borrar

Figura 83 Lista base de datos de los pacientes (Imagen propia)

Habiendo definido las interfaces que se van a implementar en el entrenador virtual, se realizó la implementación de estas interfaces desde el diseño obtenido a lápiz de cada interfaz, junto con los ambientes propuestos para cada una de estas, en la Figura 84, 85 y 86 se puede apreciar las ventanas correspondientes a inicio, escoger nivel de entrenamiento y ambiente de entrenamiento del primer nivel respectivamente.

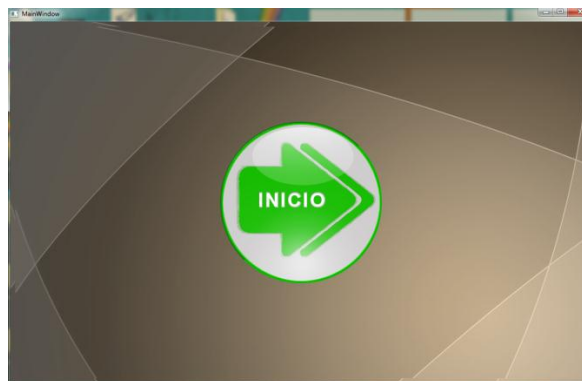


Figura 84 Interfaz inicial del ambiente virtual (Imagen Propia)

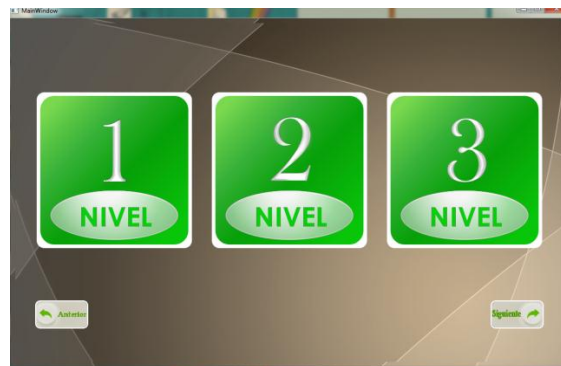


Figura 85 Interfaz anterior a los niveles de entrenamiento (Imagen Propia)

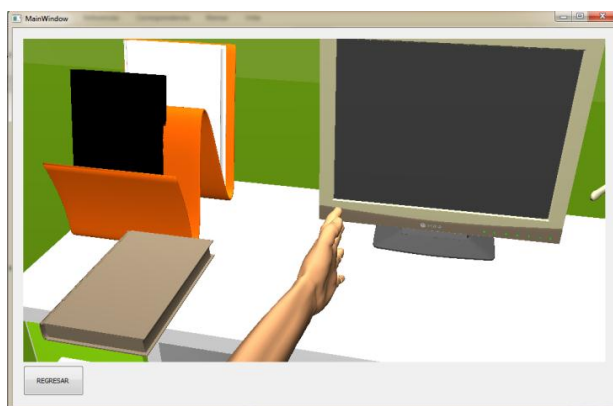


Figura 86 Interfaz Primer Nivel del Entrenador virtual (Imagen Propia)

En el primer nivel se realizan los movimientos correspondientes a la apertura y cierre de la mano, mostrados en la Figura 87, los cuales corresponden al inicio de entrenamiento que se recomienda en el protocolo después de las sesiones de fisioterapia recomendadas.

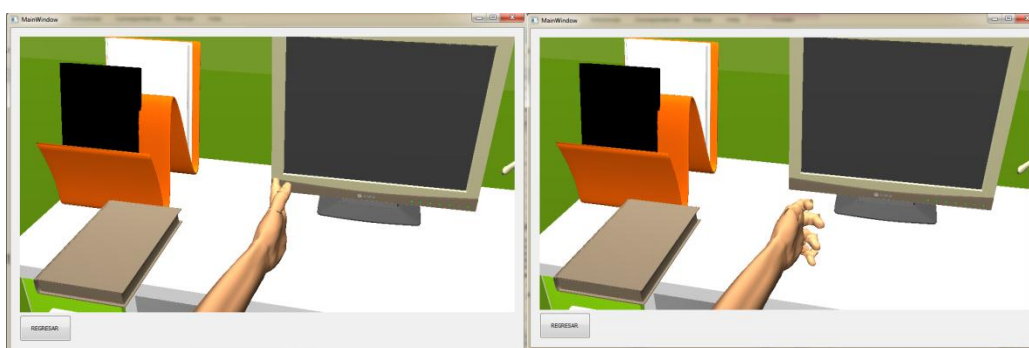


Figura 87 Movimientos correspondientes a Mano Abierta y Mano cerrada (Imagen propia)

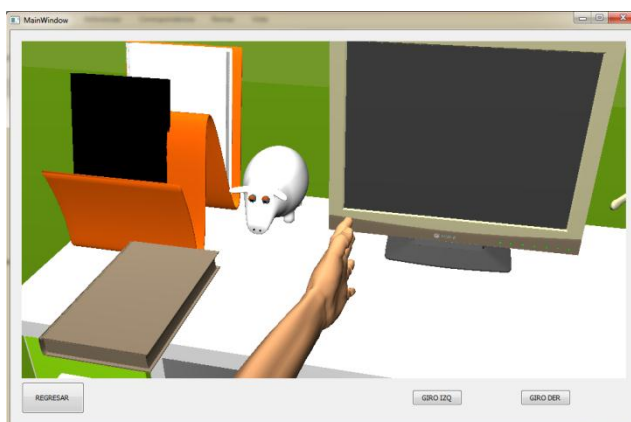


Figura 88 Interfaz Segundo nivel de Entrenador virtual (Imagen propia)

En el segundo nivel de entrenamiento (Figura 88) se plantea una serie de movimientos correspondientes a los cuatro agarres y diferentes objetos con los cuales se pueden ejecutar estos agarres, en la Figura 89 se puede apreciar un ejemplo.

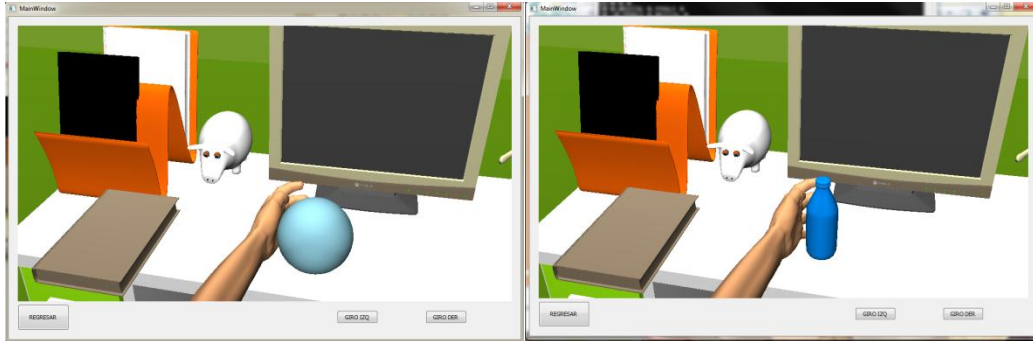


Figura 89 Agarre de diferentes objetos de la interfaz del segundo nivel (Imagen Propia)

Para el nivel tres se plantea un ambiente para la navegación virtual en el cual el usuario puede realizar los distintos movimientos y desplazarse por medio de botones sobre un ambiente virtual que está representado en un contexto cotidiano, como se puede observar en la Figura 90.



Figura 90 Interfaz tercer nivel de Entrenador Virtual (Imagen Propia)

6.9 PROTOCOLO DE ENTRENAMIENTO

A partir de la obtención de las señales, el diseño del ambiente virtual y teniendo en cuenta las consideraciones para la elaboración de un protocolo de entrenamiento, se obtuvo el siguiente protocolo de entrenamiento para personas con amputación de mano:

Este protocolo se basa en las recomendaciones del personal médico de la Universidad del Cauca involucrado en el área de rehabilitación y fisioterapia, se divide en cuatro fases de entrenamiento.

6.9.1 PRIMERA FASE (15 sesiones, tiempo estimado de mínimo seis meses)

Se planteó una fase inicial al proceso de rehabilitación de pacientes que han sufrido amputación de mano, donde se realizaran al menos quince sesiones de fisioterapia, con el fin de realizar ejercicios guiados por la persona encargada de la rehabilitación, acerca de la propiocepción, control central y manejo de distancia del amputado.

Esta fase será llevada a cabo por el personal médico asignado para la rehabilitación del paciente

6.9.2 SEGUNDA FASE (tiempo estimado, una semana)

En esta fase se van a identificar los músculos más adecuados para poder realizar una adquisición de señales electromiográficas legible, se conectan al paciente los electrodos en diferentes puntos, se le pide al sujeto intentar realizar movimientos y se registran las señales obtenidas, corroborando los puntos del paciente en los que se puede obtener mayor potencial eléctrico.

6.9.3 TERCERA FASE (Tiempo estimado, según la evolución el paciente)

En esta fase ya se han encontrado los puntos del paciente donde se conectarán los electrodos superficiales, se inicia la ejecución del ambiente virtual de entrenamiento y se le pide al usuario que intente realizar el movimiento de abrir la mano representada en el ambiente virtual y retornarla a un estado de reposo; generando un estímulo de realimentación visual por medio de la interfaz del nivel 1 del ambiente virtual de entrenamiento, ante este estímulo se debe producir una señal mioeléctrica y se recomienda, realizar este ejercicio del primer nivel las veces que sean necesarias para lograr una amplitud de señal suficiente y realizar los siguientes niveles de entrenamiento con menor dificultad.

6.9.4 CUARTA FASE (Tiempo estimado, según la evolución del paciente)

Esta última fase se plantea para desarrollarse en los niveles dos y tres del ambiente virtual de entrenamiento, para este nivel se supone que el paciente ha adquirido destreza en la realización del movimiento abrir mano y en el retorno a un estado de reposo, y ya puede intentar realizar otro tipo de movimientos como son un agarre grueso y un agarre pinza bajo el estímulo visual que genera el ambiente de entrenamiento virtual del segundo nivel en el cual aparecerán distintos objetos, los cuales el paciente debe sujetar y en el tercer nivel el paciente podrá navegar por la interfaz haciendo uso del ratón, podrá interactuar con objetos presentes en este escenario e intentar realizar los cuatro tipos de movimientos (Mano relajada, mano abierta, agarre pinza, agarre grueso).

6.10 Test de Usabilidad

Se realizó un test (Ver Anexo E) en el cual se califican criterios como la facilidad de uso, facilidad de aprendizaje, apreciación y utilidad, con el fin de obtener una evaluación del software diseñado que permitan determinar la viabilidad del software sobre los anteriores ítems, los encargados de realizar dicha evaluación fueron los fisioterapeutas involucrados en el proceso de desarrollo del macro proyecto "Prótesis de mano Robótica", En la tabla número dos se pueden observar los resultados de calificación en ítems numéricos de 1 hasta 10, siendo 10 la calificación más alta de la variable

Tabla 2. Resultados test de usabilidad (Imagen propia)

Evaluadores Fisioterapia	Facilidad de Uso	Facilidad de aprendizaje	Apreciación	Utilidad
1 Evaluador	10	9	10	9
2 Evaluador	10	8	9	9
Porcentaje	100	85	95	90

Se obtuvieron porcentajes altos en los diferentes criterios, sin embargo también se realizaron recomendaciones acerca de la interfaz gráfica, como la implementación de estímulos visuales para los usuarios.

7. CONCLUSIONES

El propósito de este trabajo de grado fue desarrollar un ambiente virtual de entrenamiento para amputados de mano con el fin de contribuir en el proceso de adaptación a una prótesis mioeléctrica. Para dar cumplimiento a este objetivo se especificaron diferentes etapas, expuestas en el siguiente orden; adquisición, procesamiento e identificación de las señales EMG, diseño de un entorno virtual con herramientas de software libre, determinación de un protocolo de entrenamiento basado en observaciones médicas y una posterior valoración del sistema propuesto. En este orden de ideas inicialmente se realizó un estudio sobre el comportamiento de las señales EMG, lo cual permitió definir las zonas más propicias del brazo para la obtención de las señales y como mejor opción el uso de electrodos de gel húmedo y cables especiales para la señales EMG con los cuales se logra una considerable disminución del ruido externo. Este tipo de señales son de difícil adquisición debido a sus bajos valores de amplitud y a la sensibilidad sobre el ruido; por esta razón se diseñó un sistema de adquisición por etapas en las cuales se realizaron pruebas de funcionamiento, obteniendo un sistema integrado con pocos componentes el cual garantiza la obtención de las señales en los rangos de frecuencia de interés. La etapa de filtraje presento algunas dificultades por lo cual se probaron diferentes configuraciones, obteniendo mejores resultados con un filtro pasivo de primer orden.

Se utilizaron distintos dispositivos para el proceso de conversión Análoga digital de la señal obtenida, en los cuales se obtuvieron resultados de conversión similares, en estos dispositivos el conversor análogo digital está diseñado para realizar la conversión de valores positivos, con lo cual se pierde la información correspondiente a la parte negativa de la señal EMG, para obtener una señal EMG completa fue necesario implementar un sistema diseñado con amplificadores operaciones en configuración sumador y comparador, siendo el comparador la mejor opción para subir el nivel DC de la señal.

Se ha definido una única zona de ubicación de electrodos para la adquisición de señales EMG, con la cual se realizaron todas las pruebas necesarias, además se obtuvo la base de datos de estas señales para cada voluntario.

Aunque los electrodos se colocan en la misma zona que está definida, cada persona posee una línea de base 'Baseline offset' diferente, por lo cual es necesario tomar algunas muestras cada vez que se va a realizar un proceso de captura de señales, este valor de offset se ve afectado por el estado de animo de la persona y otros factores que aunque son valores muy pequeños es importante tenerlos cuenta ya que esto podría dificultar el proceso de caracterización de la señal, ya que esta no estaría referenciada a un nivel de valor cero, necesario para la correcta clasificación de esta.

Los pasos implementados en el proceso de diseño de las diferentes interfaces que componen el ambiente virtual evito que se realizaran correcciones sobre la programación del mismo, ya que se podían observar errores en el modelamiento de las interfaces,

cuando estas se estaban definiendo en bosquejos a lápiz. Y así comprobar la ubicación, y forma de los diferentes componentes que las componen.

La retroalimentación por parte del personal de fisioterapia fue vital para el desarrollo de este sistema, ya que las recomendaciones que se realizaron permitieron definir un diseño de los diferentes modelos 3d implementados acorde con los ejercicios que se desean entrenar en este ambiente, además del diseño de un formulario y una base de datos con los datos más relevantes requeridos por los fisioterapeutas para el control de los pacientes que hagan uso de este sistema.

El manejo de colores e iluminación en el ambiente es importante ya que permite dar mayor sensación de realismo, además mejora la percepción de los objetos en cuanto a profundidad y ubicación de los mismos en el ambiente diseñado.

El manejo de una herramienta de edición 3d es importante, ya que aunque en la WEB se encuentren objetos ya modelados, algunos de estos no cumplen con los requerimientos del software de visualización o simplemente son objetos con muchos polígonos lo cual ocasiona una mayor carga computacional y en consecuencia una lenta ejecución del programa.

Aunque el ambiente virtual planteado está diseñado con diferentes niveles de dificultad progresiva, se vio la necesidad de dar al usuario la posibilidad de escoger el nivel que desea entrenar, consiguiendo repetir un nivel las veces que sean necesarias.

Para la verificación de los patrones obtenidos por medio del sistema desarrollado, se realizó una comparación ejecutando los algoritmos del código realizado en [Referencia] obteniendo valores y graficas similares que permitieron corroborar el funcionamiento correcto del algoritmo implementado en este trabajo.

Para el proceso de animación fue importante el manejo de eventos de tiempo, ya que permitieron el control de las diferentes posiciones, orientaciones y velocidad con que se ejecutan los movimientos de los diferentes objetos en el ambiente.

Se realizaron pruebas para verificar el correcto funcionamiento en el proceso de animación correspondiente al traslado y orientación del modelo de la mano virtual, comprobando que cada una de sus partes continúen ensambladas en el proceso de animación, ya que se presentaron muchas dificultades tales como separación e inmovilización de las partes que componen este modelo, obteniendo como resultado un proceso de animación coherente con el movimiento de una mano real.

Al realizar el proceso de animación correspondiente a la navegación del nivel tres, se presenta un problema con respecto a la cámara, esto se debe a que esta no es capaz de seguir las mismas trayectorias que están definidas para el modelo de la mano, lo cual se verifica en saltos que esta presenta en algunos ángulos de giro, este problema aún no se ha podido solucionar del todo.

8. TRABAJO FUTURO

El actual desarrollo permitió observar posibles trabajos futuros en relación al proceso de adquisición de señales electromiográficas y al entrenamiento virtual como una herramienta útil para personas con amputación de mano, a continuación se plantean las siguientes propuestas:

1. Realizar la validación de la adquisición de señales EMG y las características con pacientes amputados.
2. Obtener nuevas bases de datos con diferentes ubicaciones de los electrodos a las propuestas en este trabajo de grado.
3. Disminuir la sensibilidad al ruido del sistema de adquisición de señales EMG implementado en este trabajo de grado, utilizando dispositivos más precisos.
4. Implementar un sistema de retroalimentación sensitiva, haciendo uso de los algoritmos de detección de colisiones implementado en este trabajo de grado, para que además de la retroalimentación visual el usuario pueda sentir ya sea por medios vibratorios u otros cuando colisiona un objeto en el ambiente virtual.
5. Identificar movimientos de muñeca implementados en el entrenador virtual como son el giro y el desplazamiento longitudinal, los cuales pueden ser entrenados en los niveles dos y tres del ambiente virtual de entrenamiento planteado en este trabajo de grado.

Bibliografía

- [1] H. Romo. "Aplicación de wavelets en el análisis y la extracción de características para la clasificación de señales EMG". Trabajo de grado de maestría en ingeniería, Universidad del Cauca
- [2] M. Guerrero, J Cortes. "Sistema detector de intención de movimiento a partir de señales electromiográficas superficiales" Trabajo de grado de Ingeniería en Automática Industrial, Universidad del Cauca
- [3] G. España, "Enfermedad vascular periférica: Isquemia de miembros inferiores". Sección de cirugía vascular periférica. Hospital gral. U. Gregorio Marañón Madrid pp 161
- [4] A. Alonso, R. Sanchez, P. Hurtado, R. Steinz, L. Liptak. "Entrenador Mioelectrico de prótesis para amputados de brazo y mano". MAPFRE MEDICINA. Vol 3, No 1, pp 11-22. 2001
- [5] D.N.Lee, E. Aronson. "Visual proprioceptive control of standing in human infants". Perception & Psychophysics. Vol 15, No 3, pp 529-532. 1974
- [6] D. Scheinberg."Síndrome del Miembro fantasma. Beth Israel Deconess medical center [en línea] Disponible en <http://www.bidmc.org/YourHealth/ConditionsAZ.aspx?ChunkID=127556>>"]
- [7] RAMOS Mario, VERGARA Ángel, VÁZQUEZ Gustavo, HERNÁNDEZ Edgar, JUÁREZ Ramiro Luis. Detección y Acondicionamiento de Señales Mieléctricas. [en línea]. <disponible en><http://www.mecamex.net/anterior/cong10/trabajos/art36.pdf> [citado 3/07/2013]
- [8] KONRAD Peter. THE ABC OF EMG. A Practical Introduction to Kinestiological Electromyography. Noraxon U.S.A. Inc. Scottsdale, Arizona. 2006. Pag 7-41 [en línea]. Disponible en <<http://www.noraxon.com/docs/education/abc-of-emg.pdf?sfvrsn=0>> ISBN 0-9771622-1-4
- [9] Cifuentes Irving DISEÑO Y CONSTRUCCION DE UN SISTEMA PARA LA DETECCION DE SEÑALES ELECTROMIOGRAFICAS Universidad Autónoma de Yucatán Facultad de matemáticas 2010 Trabajo de grado para optar al título de Licenciatura en Ingeniería en Computación. Pag 9-13
- [10] "The iLimb Hand - Touch Bionics - BioSim". Noviembre 2010. Disponible: <http://www.touchbionics.com/downloads/document-library/>. Consultado: [Febrero de 2012]
- [11] "MyoLab II EMG Tester and Trainer Motion Control / Utah Arm", Noviembre 2010. Disponible: <http://www.utaharm.com/files/>. Consultado: [Febrero de 2012].

- [12] Vallejos D, García J. “Entorno Gráfico de un Entrenador Virtual de Prótesis de Mano” Trabajo de grado Ingeniería Automática Industrial. Universidad del Cauca. Colombia 2009.
- [13] Cabrera J. “Conjunto normalizado de interfaces para los componentes de un simulador virtual de prótesis de mano robótica” Trabajo de grado de maestría en ingeniería, Universidad del Cauca.
- [14] “Otto Bock – Myoboy”, Noviembre 2010. Disponible: <http://www.ottobock.com/cps/rde/xchg/ob.com/en/hs.xsl/3795.html?id=teaser2#teaser2>. Consultado: [Febrero de 2012].
- [15] Dawson M, Fahimi F, Carey J. “The Development of a Myoelectric Training Tool for Above-Elbow Amputees” The Open Biomedical Engineering Journal Vol. 6, pp 5-15, Enero 2011. Consultado: [Marzo de 2012].
- [16] Resnik L, Etter K, Klinger S L, Kambe Charles. “Using virtual reality environment to facilitate training with advanced upper-limb prosthesis” Journal of Rehabilitation Research & Development. Vol. 48, pp 707-718. Noviembre
- [17] M. Garcia, M. Rojas. “Valoración de la electromiografía de superficie multicanal para el análisis de la mialgia de esfuerzo en músculos del antebrazo”. Universidad politécnica de Catalunya. Barcelona. 2010
- [18] Analog devices AD620. www.analog.com [En línea] Disponible en < http://www.stanford.edu/class/ee122/Parts_Info/datasheets/ad620.pdf >
- [19] TL074. www.datasheetcatalog.com [En línea] Disponible en < http://www.datasheetcatalog.com/datasheets_pdf/T/L/0/7/TL074.shtml>
- [20] <http://www.cypress.com/>
- [21] FiOLite Datasheet. www.aimagin.com
- [22] http://www.aimagin.com/learn/index.php/Getting_Ready#Set_FiO_Std_in_USB_IAP_Mode
- [23] E. Mecovich. “Ponencia sobre diseño de Interfaces y Usabilidad, cómo hacer productos más útiles, eficientes y seductores”. [En línea] Disponible en <<http://planeta.gaiasur.com.ar/infoteca/disenio-de-interfaces-y-usabilidad.html>>
- [24] J. Thelin. “Rapid application development using Qt”. Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 33-54
- [25] J. Thelin. “Widgets and layouts”. Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 57-58, 62-68, 75-85, 87-89
- [26] J. Thelin. “The main window”. Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 95-120

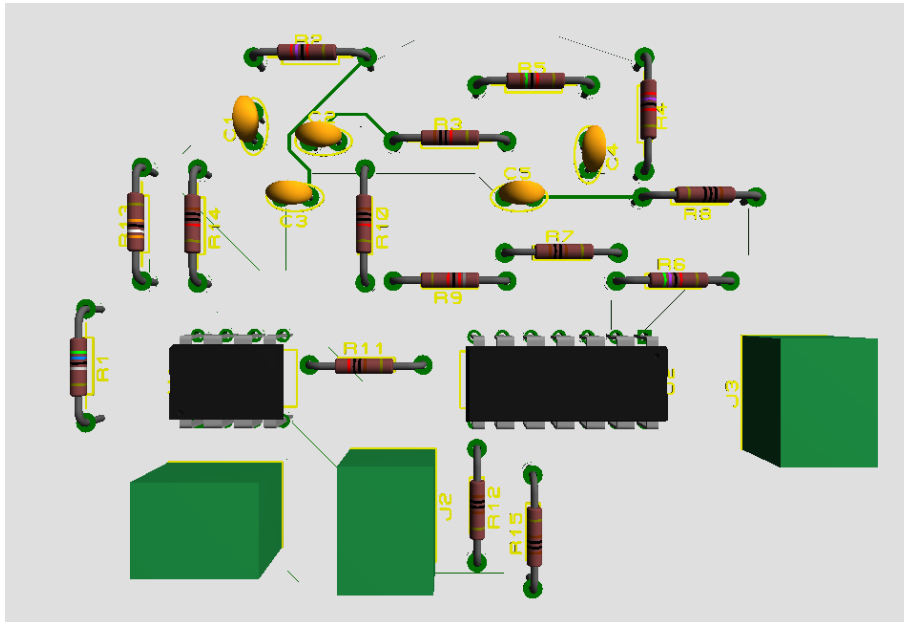
- [27] J. Thelin. "The model view framework". Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 123-142
- [28] J. Thelin. "Creating widgets". Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 157-163
- [29] <http://qt.digia.com/>
- [30] J. Blanchette, M. Summerfield ."Layout management" C++ GUI programming with Qt, 2da edición, westford, Massachusetts,prentice hall, 2008, pp 150-154
- [31] J. Blanchette, M. Summerfield ."Creating main windows" C++ GUI programming with Qt, 2da edición, westford, Massachusetts,prentice hall, 2008, pp 63-72
- [32] J. Thelin. "Data Bases". Foundation of Qt development. Vol 1, Jason Gilmore, Ed New york, après, 2007, pp 371-402
- [33] <http://www.blender.org/download>
- [34] https://picasaweb.google.com/lh/photo/xKp_CZE-unJfmL_0RIKRmQ
- [35] <http://www.estilodemujer.com/2011/07/19/muebles-basicos-para-la-zona-de-estudio-o-trabajo/>
- [36] <http://planosdecasasgratis.blogspot.com/2010/10/casa-de-campo.html>
- [37] <http://www.vtk.org/>
- [38] <http://staff.ustc.edu.cn/~zhuang/cpp/tools/ArgoUML%20Tutorial.pdf>
- [39] L. Avila, S Barre "Models". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 52
- [40] Avila, S Barre "Working with Data Attributes". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 83-86, 105-106
- [41] Avila, S Barre "Assembles". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 62,63
- [42] Avila, S Barre "Controlling 3D props". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 56,61
- [43] Avila, S Barre "Controlling the Camara". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 52
- [44] Avila, S Barre "Controlling lights". The VTK user's guide update for versión 4.0. vol 1, William J. Schrueder, 2001, pp 55,56

[45] Avila, S Barre "Volume Rendering". The VTK user's guide update for versión 4.0. vol 1, William J. Schroeder, 2001, pp 136,138

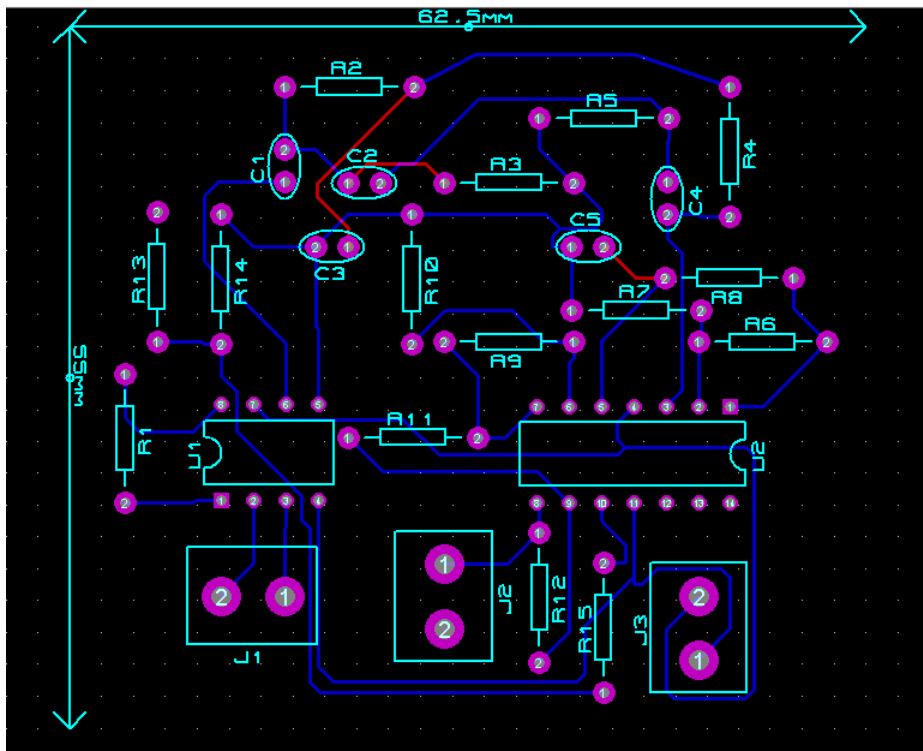
[46] J. Blanchette, M Sumerfield. "Event Processing ". C++ GUI programming with Qt, 2da edición, Westford, Massachussetts, prentinces hall, 2008, pp 167,174

ANEXOS A

Placa Circuito adquisición de señales EMG diseñado

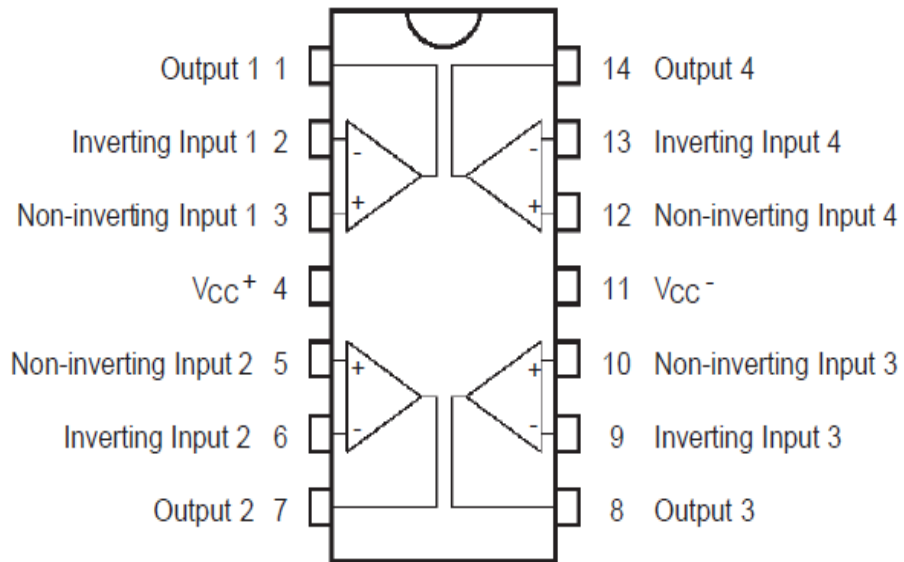


Planos circuito adquisición de señales EMG diseñado



ANEXO B

Conexión pines TL074



Conexión de pines AD620

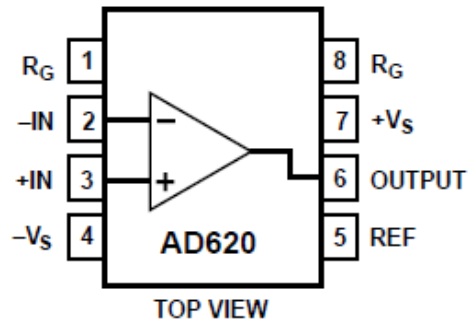
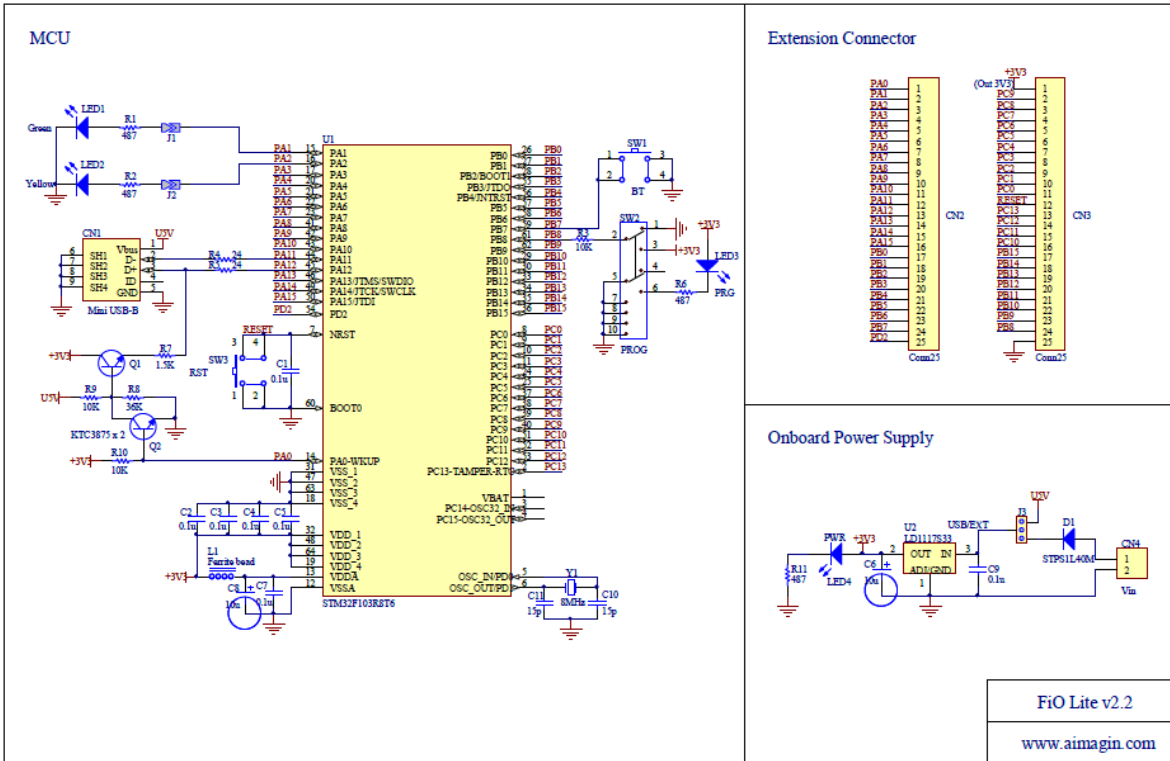


Diagrama esquemático Tarjeta FiOLite



ANEXO C

Código desarrollado en el primer bloque de Simulink Matlab 'Processing signals' para la caracterización de las señales EMG1 y EMG2

```
function [VMA_EMG1,VMA_EMG2,LO1,LO2] = processing_signals(EMG1,EMG2)
%#codegen
sample_EMG1 = single(zeros(256,1));
sample_EMG2 = single(zeros(256,1));
%medial = single(0);
%media2 = single(0);

for I=1:256
    sample_EMG1(I) = EMG1;
    sample_EMG2(I) = EMG2;
end

%% 1. Valor Medio Absoluto VMA
medial = mean(abs(sample_EMG1));
media2 = mean(abs(sample_EMG2));
VMA_EMG1 = medial;
VMA_EMG2 = media2;

%% 2. Longitud de Onda: mide complejidad de la onda LO.
LO_ch1 = single(sample_EMG1(1));
LO_ch2 = single(sample_EMG2(1));
for i = 2:256,
    delta_xi_ch1 = sample_EMG1(i) - sample_EMG1(i-1);
    LO_ch1 = LO_ch1 + abs(delta_xi_ch1);
    delta_xi_ch2 = sample_EMG2(i) - sample_EMG2(i-1);
    LO_ch2 = LO_ch2 + abs(delta_xi_ch2);
end
LO1 = LO_ch1;
LO2 = LO_ch2;
```

Código desarrollado en el segundo bloque de simulink 'processing signals' para las caracterización de las señales EMG1 y EMG2

```
function[AVG1, AVG2, VAR1, VAR2, ENER1, ENER2]=
processing_signals(EMG1,EMG2)
%#eml
sample_EMG1 = single(zeros(256,1));
sample_EMG2 = single(zeros(256,1));
senal= single(zeros(256,1));
senal2= single(zeros(256,1));
s = single(zeros(1,256));
sum=single(0);
sum1=single(0);
sum2=single(0);
d=0;
matrix =single(zeros(4,128));
for I=1:1:256
    sample_EMG1(I) = EMG1;
    sample_EMG2(I) = EMG2;
end
% senall=[sample_EMG1 sample_EMG2];
AVG = single(zeros(2,1));
VAR = single(zeros(2,1));
ENER = single(zeros(2,1));

for ind=1:2,
    for i=1:1:256,
        senal(i,1)=sample_EMG1(i);
        senal2(i,1)=sample_EMG1(i);
    end
    %%% declaracion de QMF
    qmf = [0.2352 0.5706 0.3252 -0.0955 -0.0604 0.0249];
    D=3;
    n = length(senal); %se calcula longitud de la señal.
    wt = single(zeros(n,D+1));
for id=1:1:3,
    if id==1
        n=256;
    else if id==2
        n=128;
    else if id==3
```

ANEXO D

Código implementado para traducir en un bloque de Simulink de Matlab la Red Neuronal

```
function y3 = fcn(MEAN1, MEAN2, LO1, LO2, AVG1,AVG2,VAR1, VAR2, ENER1,
ENER2)
%#eml
% w1=single(zeros(10,10));
% w2=single(zeros(5,10));
% w3=single(zeros(5));
bias1=[1.8182 -13.3056 0.5422 -4.3184 3.2676 5.2546 -4.4344 -0.7881
1.0900 3.2248];
bias2=[2.0561 0.1606 0.4278 0.7825 -1.6481];
bias3=0.7514;
i1=single(0);
i2=single(0);
i3=single(0);
y1=single(zeros(10));
y2=single(zeros(5));
% y3=single(0);
patron = single(zeros(10));
patron(1)=LO1;
patron(2)=LO2;
patron(3)=MEAN1;
patron(4)=MEAN2;
patron(5)=AVG1;
patron(6)=AVG2;
patron(7)=VAR1;
patron(8)=VAR2;
patron(9)=ENER1;
patron(10)=ENER2;

w1=[-2.3786 3.3123 3.4364 -13.4183 -5.1783 28.6256 -24.6496 23.4213
1.7795 2.4282;
0.5450 0.2130 -0.4890 -0.2006 -9.7648 19.0423 72.6313 227.1784
1.1162 3.0428;
-0.0267 0.8754 3.2177 -14.0038 0.4298 -1.1302 -80.1166 29.9725 1.3095
0.9813;
-0.1909 1.1399 -6.0655 15.0630 14.8903 25.7686 41.2168 -86.3204
0.7453 1.5653;
0.2925 0.6500 -5.9656 -12.7103 3.9292 -19.6626 21.7534 50.5248
1.4181 -3.8091;
-1.1812 1.2781 6.4858 -15.3351 -12.9639 -6.3783 52.6269 -62.6946 -
2.0702 -1.2851;
1.4313 1.6044 7.8701 2.8745 -5.3693 37.1601 -24.2288 173.9898 -
0.5906 -0.9485;
-0.7277 -0.3996 9.2188 9.8204 -15.3209 25.5621 41.8422 -41.0584
0.5026 -2.4233;
0.0335 0.2562 -9.9086 -8.2554 0.0939 16.1136 69.0381 -4.4853 -0.9983
2.8283;
-0.0036 0.5005 -8.3079 -14.2951 -2.9363 9.0210 -37.0267 115.6257 -
1.5340 4.5217];
```



```

w2= [-1.4411 -2.2813 0.8115 0.1875 1.7279 1.6050 0.0056 -1.7123 -0.5120
0.9957;
    1.7553 0.8729 0.5151 -3.1853 0.2321 0.6489 0.2612 0.5730 0.2776
2.2702;
    -1.7513 -0.2513 0.5647 0.8236 1.5786 -0.4322 -0.4217 -1.8606 0.8105
-0.5077;
    1.0777 2.3178 0.2668 1.0937 -0.7029 1.0129 0.1994 0.8076 -0.3083 -
0.3428;
    -1.2784 1.9790 -0.7843 0.0128 -0.7578 -0.4585 -0.2899 -0.1194 0.0912
-0.7706];

w3=[-0.0826 -0.5009 0.8404 1.5015 -1.4927];

%%primer for de la capa oculta

for j=1:1:10,
    for j1=1:1:10,
        i1=(i1+ patron(j1)*w1(j,j1));
    end
    i1=i1+bias1(j);
    y1(j)=2/(1+exp(-2*i1))-1;
end
%% segunda capa de la red

for j2=1:1:5,
    for j3=1:1:10,
        i2=i2+(y1(j3)*(w2(j2,j3)));
    end
    i2=i2+bias2(j2);
    y2(j2)=2/(1+exp(-2+i2))-1;
end
%%salida de la red
for j4=1:1:5,
    i3=i3+y2(j4)*w3(j4);
end
i3=i3+bias3;
y3=floor(i3+0.5);

```

ANEXO E

TEST DE USABILIDAD ENTRENADOR VIRTUAL PARA PACIENTES CON AMPUTACIÓN DE MANO

Nombre del encuestador: Lucía Segura

Nombre del encuestador: Daniel Rosero

Hora comienzo:

Fecha:

Perfil del encuestado:

Nombre:

Profesión: Fisioterapeuta

Vinculado al proyecto prótesis de mano de la Universidad del Cauca: Si No

Nota: Las personas vinculadas al proyecto ‘prótesis de mano’ de la universidad del Cauca, delegaron a dos personas para realizar la encuesta, debido a que ellos se encontraban fuera de la ciudad y no podía realizar la evaluación.

El entrenador virtual para pacientes con amputación de mano se diseñó bajo especificaciones médicas para contribuir a los pacientes con amputación de mano en la adaptación a una prótesis mioeléctrica, de igual forma para asistir al personal de fisioterapia encargado en el proceso de rehabilitación que se realiza a continuación de una amputación de mano.

1. En una escala de 1 a 10 donde 10 es “muy interesante” y 1 es “Nada interesante”
¿Cómo de interesante cree que es el ambiente virtual de entrenamiento para pacientes con amputación de mano?
2. En una escala de 1 a 10 donde 10 es “muy interesante” y 1 es “Nada interesante”
¿Cómo de interesante cree que es el ambiente virtual de entrenamiento para los profesionales en el área de fisioterapia?
3. Cuáles de las siguientes características le atraen del producto

Utilidad Facilidad de uso Apreciación Facilidad de aprendizaje

Otras variables (Especifique) _____

Debilidades del producto

4. Califique las siguientes características de 1 a 10, siendo 10 el puntaje más alto.

Utilidad	Facilidad de Uso	Apreciación	Facilidad de aprendizaje

5. ¿Cuáles de las siguientes características no le atraen del producto?

No lo necesito Es difícil de usar Otras variables Estoy interesado en el producto.

Otra (Especifique) _____

6. Tiene alguna sugerencia o comentario para realizar sobre este ambiente virtual de entrenamiento para pacientes con amputación de mano?

ANEXO F

MANUAL DE USUARIO AMBIENTE VIRTUAL DE ENTRENAMIENTO PARA PACIENTES CON AMPUTACIÓN DE MANO



INDICE

CONTENIDO

Introducción al software

1. Características del software
2. Instalación
3. Ayuda
4. Manual del Usuario

Introducción al Software

- ✓ El siguiente software está pensando para acelerar el proceso de asimilación y adaptación de una persona con amputación de mano a una prótesis mioeléctrica por medio de retroalimentación visual
- ✓ Se podrán almacenar los datos de los usuarios y la información relevante de estos en una base de datos
- ✓ El entrenador virtual para amputados de mano es un software escalable el cual permite extender sus funciones a partir de programación en C++.

Características

Este software le permitirá al usuario:

- ✓ Acceder a una interfaz amigable que le permita al paciente con amputación de

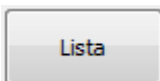
- ✓ Mejorar progresivamente la contracción de los músculos involucrados en la adquisición de las señales electromiográficas correspondientes
- ✓ Llevar datos estadísticos acerca de los pacientes que acceden a este tipo de tratamientos
- ✓ mano generar movimientos en una interfaz a través de estímulos visuales
- ✓ Se puede utilizar en las versiones GNU/Linux™, Microsoft Windows™
- ✓ Los requisitos mínimos de hardware serían los siguientes:
 - Procesador, al menos, Pentium M
 - Al menos 1GB de memoria RAM
 - Tarjeta gráfica incorporada

Instalación del programa

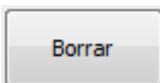
La instalación del programa se realiza al ejecutar la aplicación, de esta manera se cargan los elementos de la interfaz y se accede a esta.

Ayuda: Permite consultar las funciones del software

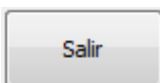
Los botones con los que el usuario podrá acceder a las opciones del ambiente virtual de entrenamiento se pueden ver a continuación con su respectiva función



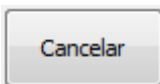
El botón lista despliega la lista de los usuarios registrados junto con su información relevante.



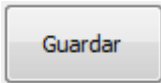
El botón borrar tiene la función de eliminar usuarios seleccionados de la lista.



El botón salir en la ventana de la base de datos posee la función de cerrar la ventana actual y ejecutar el inicio del ambiente virtual de entrenamiento



El botón cancelar finaliza la ejecución del programa



El botón guardar, archiva los datos ingresados en lista correspondiente a la base de datos



El botón Inicio del ambiente virtual, permite al usuario ingresar a otra ventana donde se encuentran los niveles 1,2 y 3, para que este pueda tener acceso a cada uno de ellos, según lo desee.



El botón del nivel 1 permite al usuario acceder al primer nivel de entrenamiento



El botón del nivel 2 permite al usuario acceder al segundo nivel de entrenamiento



El botón del nivel 3 permite al usuario acceder al tercer nivel de entrenamiento



El botón anterior ubicado en la ventana de los 3 niveles permite regresar a la ventana de inicio



El botón siguiente permite al usuario continuar con el siguiente nivel de entrenamiento



El botón girar 'izquierda' permite al usuario visualizar el giro de la mano hacia la izquierda o movimiento de pronación, este botón se encuentra en los niveles 2 y 3 del ambiente virtual de entrenamiento



El botón girar 'derecha' permite al usuario visualizar el giro de la mano hacia la izquierda o movimiento de supinación, este botón se encuentra en los niveles 2 y 3 del ambiente virtual de entrenamiento



El botón subir permite al usuario desplazar la mano verticalmente hacia arriba por medio de acciones de teclado, este botón se encuentra en el nivel del ambiente virtual de entrenamiento



El botón bajar permite al usuario desplazar la mano verticalmente hacia abajo por medio de acciones de teclado, este botón se encuentra en el nivel 3 del ambiente virtual de entrenamiento



El botón avanzar permite al usuario navegar por el ambiente virtual diseñado en el nivel 3, funciona por medio de acciones de teclado



El botón izquierda permite al usuario girar hacia la izquierda por medio de acciones de teclado, este botón se encuentra en el nivel 3 del ambiente virtual de entrenamiento



El botón derecha permite al usuario girar hacia la derecha, este botón se encuentra en el nivel 3 del ambiente virtual de entrenamiento

Manual de Usuario Permite acceder al manual del Usuario

Al ejecutar el programa, el usuario tendrá acceso a una ventana inicial que pide ingresar la información relevante del paciente y así poder obtener un listado de cada uno de ellos, posibilitando realizar estadísticas poblacionales

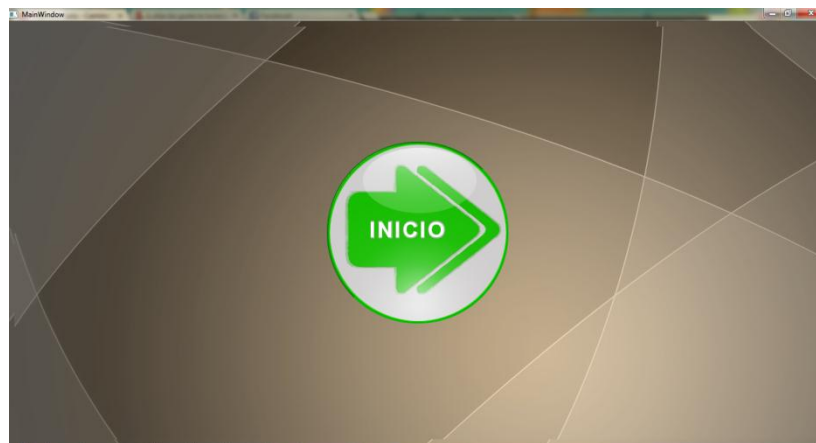
The screenshot shows a software window titled "DatosPersona" with a form for entering patient information. The form is divided into two main columns. The left column contains fields for "Primer Apellido", "Segundo Apellido", "Nombre", "N° Cedula", "Procedencia", "Residencia", "Dirección", "Correo Electronico", and "Teléfono" (with sub-fields for "Casa", "Celular", and "Otro (Indiqué)"). The right column contains fields for "Ocupación", "Sexo" (with radio buttons for "Femenino" and "Masculino"), "Edad", "Lateralidad" (with radio buttons for "Izquierda" and "Derecha"), "Causa de Amputación" (with a dropdown menu showing "Primaria o Traumatica"), "Escolaridad" (with a dropdown menu showing "Basica Primaria"), and "Tipo de Amputación" (with a dropdown menu showing "Muñeca"). At the bottom of the window, there are four buttons: "Lista", "Salir", "Cancelar", and "Guardar".

The screenshot shows a window titled 'Form' containing a table with the following data:

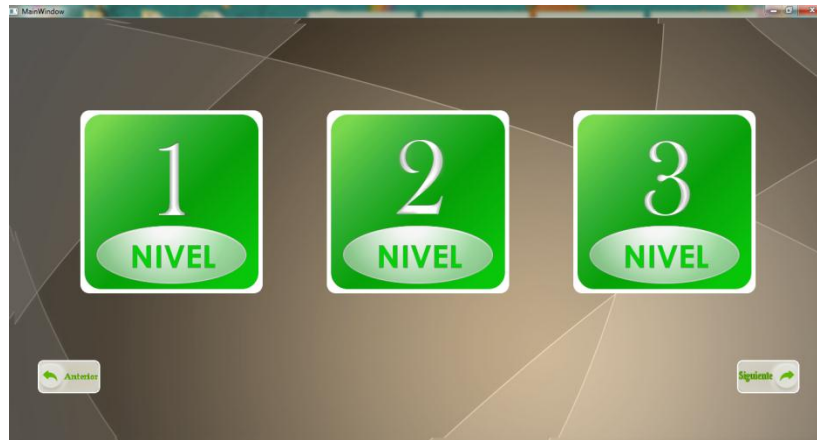
	Id	Cedula	PrimerApellido	SegundoApellido	Nombre	Procedencia	Celular	Correo	Direc
1	1		Segura	Cometa	Piedad				

Below the table is a 'Borrar' button.

Si el usuario guarda los datos y presiona el botón 'salir' podrá acceder a la ventana inicial del ambiente virtual de entrenamiento que se ve a continuación

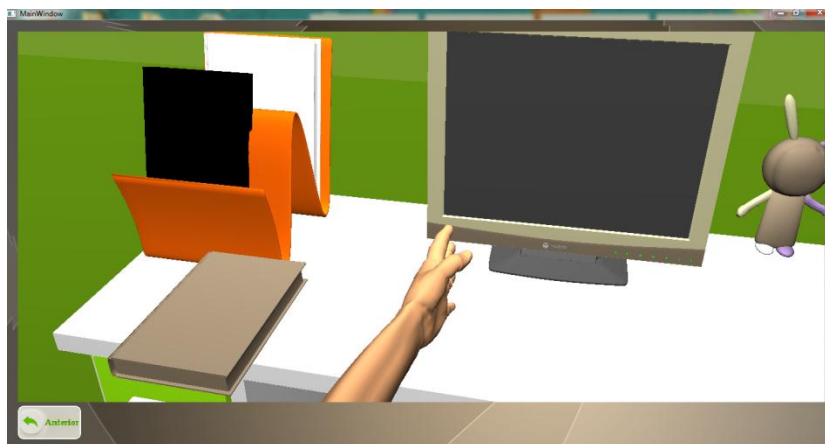


Al pulsar la ventana inicio el usuario ingresara a una interfaz que desplegara las opciones de los tres niveles a los cuales puede acceder siendo estos nivel 1, nivel 2, nivel 3 como se ve en la figura



El usuario podrá pulsar los botones correspondientes a los niveles 1,2 y 3 en el orden que este lo requiera y podrá volver a este ventana pulsando el botón 'anterior'.

Al pulsar los botones de los niveles respectivos el usuario, este se encontrara con distintas interfaces, en el primer y segundo nivel la ventana que se despliega es como se ve a continuación.



En esta el usuario podrá realizar algunos ejercicios principales como son abrir y cerrar la mano en el primer nivel, el segundo nivel aunque se desarrolla bajo la misma interfaz gráfica permitirá al usuario la interacción con distintos objetos y posibilitara la ejecución de otros movimientos como son 'agarre pinza' y ' agarre grueso' además de los ya mencionados anteriormente.

El nivel 3 permitirá al usuario entrar en un ambiente más complejo en el que podrá navegar e interactuar con distintos objetos, por medio del teclado del computador para avanzar, subir, bajar, izquierda, derecha y girar, de la misma manera que en el nivel 2 el usuario podrá realizar los movimientos correspondientes a 'mano abierta', 'mano relajada', 'agarre pinza' y 'agarre grueso' regresar a la ventana inicial pulsando el botón 'anterior'.

