



UNIVERSIDAD DEL AZUAY

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

**Implementación de un sistema de control de actuadores para una
prótesis de antebrazo a partir de señales electromiográficas**

**Trabajo de Graduación previo a la obtención del título de
Ingeniero Electrónico**

Autor

Julio Edwin Álvarez Delgado

Director

Ing. Daniel Esteban Iturralde Piedra

Cuenca, Ecuador

2014

DEDICATORIA

Este trabajo de graduación va dedicado a mis padres, mi familia, ya que gracias a ellos desde temprana edad se despertó en mí el amor por la lectura y las ciencias, a Cecilia y Pamela que siempre estuvieron para brindarme una mano amiga cuando lo necesité y a todas las personas que con su aporte, consejos y apoyo permitieron la culminación de esta tesis.

Al gran arquitecto del universo, por poner a las personas adecuadas en el momento adecuado.

Julio Álvarez Delgado.

*Escritura N.
20-03-14*

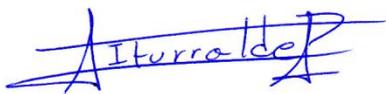
RESUMEN

IMPLEMENTACION DE UN SISTEMA DE CONTROL DE ACTUADORES PARA UNA PROTESIS DE ANTEBRAZO A PARTIR DE SEÑALES ELECTROMIOGRAFICAS

El presente trabajo de graduación busca desarrollar un sistema modular mediante el cual se pueda adquirir una señal EMG, cuantificarla, procesarla y producir un movimiento en actuadores eléctricos, además se propone la construcción de una mano robótica para las pruebas correspondientes del sistema y un software de control y pruebas de las placas PCB desarrolladas.

Como resultado se obtuvo un sistema modular escalable, que puede ser modificado según las necesidades que se presenten en desarrollos posteriores, además con la construcción de la mano robótica se tiene un primer acercamiento al diseño de una prótesis inteligente, y por último con el software de control también se pudo obtener un banco de datos con varias mediciones EMG las cuales pueden ser empleadas para el desarrollo de aplicaciones bioelectrónicas

Palabras clave: Bioelectrónica, EMG, prótesis, robótica, microcontroladores, Labview.



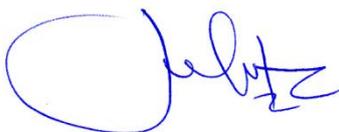
Ing. Daniel Esteban Iturralde Piedra

Director de Tesis



Ing. Francisco Eugenio Vásquez Calero

Director de Escuela



Julio Edwin Álvarez Delgado

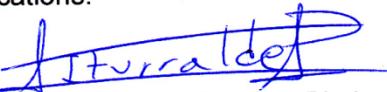
Autor

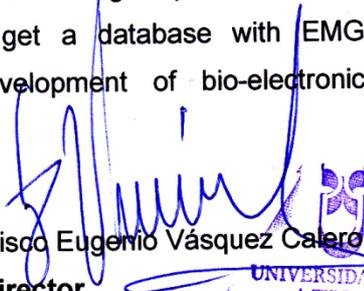
ABSTRACT

IMPLEMENTATION OF A CONTROL SYSTEM FOR FOREARM PROSTHESIS ACTUATORS FROM ELECTROMYOGRAPHIC SIGNALS

This graduation paper seeks to develop a modular system whereby an EMG signal can be acquired, quantified, and processed so as to produce movement in electric actuators. In addition, a construction of a robotic hand for testing the corresponding system and software for control and testing of the developed PCB plates is proposed.

The result is a scalable modular system that can be modified according to the needs that arise in later developments. Furthermore, with the construction of the robotic hand we can obtain a first approach to the design of an intelligent prosthesis; and finally, with the control software we could also get a database with EMG measurements which can be used for the development of bio-electronic applications.


Ing. Daniel Esteban Iturralde Piedra
Thesis Director


Ing. Francisco Eugenio Vásquez Calero
School Director


Julio Edwin Álvarez Delgado
Author


UNIVERSIDAD DEL
AZUAY
ESCUELA
Ingeniería Eléctrica


UNIVERSIDAD DEL
AZUAY
DPTO. IDIOMAS


Translated by,
Lic. Lourdes Crespo

ÍNDICE DE CONTENIDOS

| | |
|--|-----------|
| DEDICATORIA | ii |
| RESUMEN..... | iii |
| ABSTRACT..... | iv |
| ÍNDICE DE CONTENIDOS | v |
| INDICE DE FIGURAS..... | viii |
| INDICE DE TABLAS | xiii |
| INTRODUCCION | 1 |
| | |
| CAPITULO 1: REFERENTES TEORICOS | 3 |
| 1.1 Anatomía del antebrazo..... | 3 |
| 1.1.1 Tipos de movimiento | 3 |
| 1.1.2 Músculos y mecanismos de movimiento | 6 |
| 1.1.3 Musculatura y articulación de antebrazo y mano..... | 8 |
| 1.2 Señales electromiográficas | 9 |
| 1.2.1 Generación del potencial de acción..... | 10 |
| 1.2.2 Potencial de acción de la unidad motora..... | 11 |
| 1.2.3 Electrodo y derivaciones propuestas | 14 |
| 1.3 Métodos de adquisición de señales mioeléctricas | 16 |
| 1.3.1 Consideraciones físicas | 17 |
| 1.3.2 Consideraciones electrónicas | 18 |
| 1.4 Métodos de análisis de señales mioeléctricas..... | 20 |
| 1.4.1 Dominio del tiempo | 20 |
| 1.4.2 Dominio de la frecuencia | 21 |
| 1.4.3 Dominio tiempo-escala | 22 |
| 1.5 Clasificación de patrones de movimiento | 26 |
| | |
| CAPITULO 2 DISEÑO DE LA TARJETA DE ADQUISICIÓN DE SEÑALES MIOELECTRICAS | 28 |
| 2.1 Descripción de los bloques funcionales. | 28 |

| | | |
|-------|-------------------------|----|
| 2.1.1 | Pre amplificación | 29 |
| 2.1.2 | Filtrado | 33 |
| 2.1.3 | Amplificación..... | 35 |
| 2.2 | Producto Final | 37 |

CAPITULO 3: DISEÑO DE LA TARJETA DE CONTROL DE ACTUADORES.....40

| | | |
|-------|--|----|
| 3.1 | Descripción de los bloques funcionales | 40 |
| 3.2 | Módulo de control de motores MDrive | 41 |
| 3.3 | Módulo de sensores HSens..... | 43 |
| 3.4 | Módulo de procesamiento y control McuPro | 45 |
| 3.4.1 | Descripción del circuito | 46 |
| 3.4.2 | Firmware | 47 |

CAPITULO 4: DISEÑO DE MANO ROBOTICA.....53

| | | |
|-------|---------------------------------|----|
| 4.1 | Consideraciones de diseño | 53 |
| 4.2.2 | Sensores de presión..... | 60 |
| 4.2.3 | Acelerómetro | 62 |
| 4.3 | Diseño y construcción | 64 |
| 4.3.1 | Diseño de los dedos | 65 |
| 4.3.1 | Diseño de la palma..... | 67 |
| 4.4 | Montaje Final | 67 |

CAPITULO 5: SOFTWARE PARA EL CONTROL Y PRUEBA DE LA MANO ROBOTICA72

| | | |
|-------|--|----|
| 5.1 | Diagramas de flujo, bloques y descripción de funcionamiento..... | 72 |
| 5.1.1 | Pestaña EMG | 73 |
| 5.1.2 | Pestaña ACCELEROMETER..... | 75 |
| 5.1.3 | Pestaña SENS HUB | 76 |
| 5.1.4 | Pestaña MOTOR DRIVE..... | 76 |

| | |
|--|-----------|
| CONCLUSIONES..... | 78 |
| RECOMENDACIONES..... | 80 |
| BIBLIOGRAFIA..... | 82 |
| ANEXOS | 86 |
| ANEXO 1: ESQUEMAS Y DISEÑOS DE PCB..... | 86 |
| ANEXO 2: FIRMWARE | 99 |
| Anexo 2.1: Dextra PIC1 (Adquisición señal EMG, procesamiento y control de dedos pulgar e índice) Firmware | 99 |
| Anexo 2.2: Dextra PIC2 (control dedos medio y anular) Firmware..... | 111 |
| Anexo 2.3: Dextra PIC3 (control de muñeca) Firmware..... | 120 |
| ANEXO 3: PLANOS DE LA MANO ROBÓTICA | 127 |
| ANEXO 5: SOFTWARE DE MEDICIÓN | 145 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1. 1 Planos anatómicos del cuerpo humano..... | 4 |
| Figura 1. 2 Rangos de movimientos de muñeca, parte superior del brazo, hombros y codos. | 6 |
| Figura 1. 3 Conexión indirecta (a y b) y conexión directa (c) de músculos esqueléticos. | 7 |
| Figura 1. 4 Huesos antebrazo y mano. | 8 |
| Figura 1. 5 Músculos del antebrazo. | 9 |
| Figura 1. 6 Estructura de una neurona. | 10 |
| Figura 1. 7 Diferencia de potencial generada en el músculo por variación de los niveles de sodio y potasio en los extremos de la membrana celular..... | 11 |
| Figura 1. 8 Conexión existente entre una motoneurona y las fibras musculares. | 12 |
| Figura 1. 9 Potencial de acción del motor unitario MUAP y tren de potenciales de acción de motores unitarios MUAPT..... | 13 |
| Figura 1. 10 Sumatoria de MUAPTS para formar una señal EMG. | 13 |
| Figura 1. 11 Electrodo de inserción..... | 14 |
| Figura 1. 12 Tipos de electrodos superficiales, a) electrodos pasivos b) electrodos activos Delsys. | 15 |
| Figura 1. 13 Configuración bipolar..... | 16 |
| Figura 1. 14 Ubicación bipolar de electrodos, en el caso de los electrodos Delsys, estos contienen dos electrodos y la pre amplificación en su carcasa | 17 |
| Figura 1. 15 Ubicación de los electrodos en un músculo..... | 17 |
| Figura 1. 16 Configuración de amplificador diferencial donde m es la señal EMG y n es el ruido. | 19 |
| Figura 1. 17 Dominios para el análisis de señales. | 23 |
| Figura 1. 18 Relación escala-frecuencia en análisis de señales usando WT..... | 24 |
| Figura 1. 19 Bancos de Filtros, a) Descomposición DWT b) Descomposición WPT. | 26 |
| Figura 2. 1 Bloques funcionales del sistema de adquisición de señales sEMG..... | 29 |
| Figura 2. 2 Disposición de los electrodos superficiales y el electrodo de referencia. | 30 |

| | |
|--|----|
| Figura 2. 3 Circuito RLD propuesto en la hoja de datos del amplificador instrumental INA128..... | 31 |
| Figura 2. 4 Circuito de pre amplificación. | 32 |
| Figura 2. 5 Espectro de una señal EMG..... | 33 |
| Figura 2. 6 Topología filtro Sallen Key a.) pasa-bajo b.) pasa-alto..... | 34 |
| Figura 2. 7 Etapa de Filtrado del sistema de adquisición de señales EMG. | 35 |
| Figura 2. 8 Amplificador no inversor de ganancia unitaria. | 35 |
| Figura 2. 9 Amplificador diferencial con ganancia unitaria. | 36 |
| Figura 2. 10 Esquema de la etapa de amplificación..... | 37 |
| Figura 2. 11 Circuito PCB para la adquisición de señales electromiográficas. | 38 |
| Figura 3. 1 Diagrama de bloques de la interacción de los circuitos componentes del sistema..... | 41 |
| Figura 3. 2 Parte del esquema del módulo de control de motores Mdrive..... | 42 |
| Figura 3. 3 PCB del módulo de control de motores MDrive. | 43 |
| Figura 3. 4 Parte del esquema del módulo HSens..... | 44 |
| Figura 3. 5 PCB del módulo HSens. | 45 |
| Figura 3. 6 Parte del esquema del módulo de procesamiento y control McuPro..... | 46 |
| Figura 3. 7 PCB del módulo McuPro..... | 47 |
| Figura 3. 8 Diagrama de relación de procesos entre los Microcontroladores del módulo McuPro..... | 48 |
| Figura 3. 9 Algoritmo del MCU U25..... | 49 |
| Figura 3. 10 Algoritmo del MCU U27..... | 51 |
| Figura 3. 11 Algoritmo del microcontrolador U26. | 52 |
| Figura 4. 1 Huesos de la mano. | 54 |
| Figura 4. 2 Relaciones longitudinales entre los huesos de los dedos y ángulos de flexión. | 54 |
| Figura 4. 3 Esqueleto de una mano y modelo mecánico de la misma. | 55 |
| Figura 4. 4 Modelos prensiles humanos de Schlesinger, a) cilíndrica palmar, b) subtermino lateral, c) centralizado, d) tridigital, e) posición subterminal. | 56 |
| Figura 4. 5 Esquema interno típico de un motor de pasos a) secuencia para pasos de 90 grados b) pasos de 45 grados. | 58 |
| Figura 4. 6 Secuencia de giro de un motor de pasos..... | 58 |
| Figura 4. 7 Motor Faulhaber 1524E006S123..... | 59 |

| | |
|---|----|
| Figura 4. 8 Pulsos de salida del encoder del motor Faulhaber..... | 60 |
| Figura 4. 9 Composición de un sensor FSR..... | 61 |
| Figura 4. 10 Curva característica de un sensor FSR. | 62 |
| Figura 4. 11 Acelerómetro MMA7341L..... | 62 |
| Figura 4. 12 Salidas existentes en la tarjeta del acelerómetro MMA7341L..... | 63 |
| Figura 4. 13 Valores característicos de voltaje de MMA7341L..... | 64 |
| Figura 4. 14 Estructura base para la construcción de los dedos..... | 65 |
| Figura 4. 15 Puntas de los dedos propuestos. | 66 |
| Figura 4. 16 Soporte del dedo pulgar..... | 66 |
| Figura 4. 17 a) Diseño de la palma de la mano b) Mano con pulgar en posición de agarre cilíndrica palmar..... | 67 |
| Figura 4. 18 Ubicación de la circuitería en el dorso de la mano..... | 68 |
| Figura 4. 19 Disposición de los dedos en la mano robótica..... | 69 |
| Figura 4. 20 Disposición de la estructura del pulgar..... | 69 |
| Figura 4. 21 Ubicación de los pulsantes en el pulgar a) pulsante de posición final b) pulsantes de tipo de agarre..... | 70 |
| Figura 4. 22 Ubicación de los sensores de presión en a) Dedos índice, medio y anular b) Dedo pulgar y c) Palma de la mano..... | 70 |
| Figura 4. 23 Hub de montaje para el motor de pasos..... | 71 |
| Figura 4. 24 Montaje del Hub del motor de pasos en la mano robótica. | 71 |
| Figura 5. 1 Panel Frontal pestaña EMG..... | 73 |
| Figura 5. 2 Diagrama de flujo de la pestaña EMG..... | 74 |
| Figura 5. 3 Diagrama de bloques del panel ACCELEROMETER..... | 75 |
| Figura 5. 4 Diagrama de bloques del panel SENS HUB..... | 76 |
| Figura 5. 5 Diagrama de flujo del panel MOTOR DRIVE..... | 77 |
| Figura A1. 1 Cara superior del PCB Emg-Acq 4..... | 86 |
| Figura A1. 2 Cara inferior del PCB Emg-Acq 4..... | 87 |
| Figura A1. 3 Vista 3D cara superior EMG – Acq 4..... | 87 |
| Figura A1. 4 Vista 3D cara inferior EMG – Acq 4..... | 88 |
| Figura A1. 5 Esquemático del circuito EMG – Acq 4..... | 89 |
| Figura A1. 6 Cara superior del PCB H-Sens | 90 |
| Figura A1. 7 Cara inferior del PCB H-Sens..... | 90 |
| Figura A1. 8 Vista 3D cara superior H-Sens..... | 91 |

| | |
|--|-----|
| Figura A1. 9 Vista 3D cara inferior H-Sens..... | 91 |
| Figura A1. 10 Esquemático del circuito H-Sens..... | 92 |
| Figura A1. 11 Cara superior del PCB MDrive..... | 93 |
| Figura A1. 12 Cara inferior del PCB MDrive..... | 93 |
| Figura A1. 13 Vista 3D cara superior MDrive..... | 94 |
| Figura A1. 14 Vista 3D cara inferior MDrive..... | 94 |
| Figura A1. 15 Esquemático del circuito MDrive..... | 95 |
| Figura A1. 16 Cara superior del PCB MCUPro..... | 96 |
| Figura A1. 17 Cara inferior del PCB MCUPro..... | 96 |
| Figura A1. 18 Vista 3D cara superior MCUPro..... | 97 |
| Figura A1. 19 Vista 3D cara inferior MCUPro..... | 97 |
| Figura A1. 20 Figura A1.20 Esquemático del circuito MCUPro..... | 98 |
| Figura A3. 1 Plano pieza Bridge..... | 127 |
| Figura A3. 2 Plano pieza Hand A..... | 128 |
| Figura A3. 3 Plano pieza Hand B..... | 129 |
| Figura A3. 4 Plano pieza Hand W..... | 130 |
| Figura A3. 5 Plano pieza JointThumb..... | 131 |
| Figura A3. 6 Plano pieza JointSmall..... | 132 |
| Figura A3. 7 Figura A3. 7 Figura C1.7 Plano pieza MotorBracketH..... | 133 |
| Figura A3. 8 Plano pieza Motor Bracket Top Side..... | 134 |
| Figura A3. 9 Plano pieza Motor Bracket Top A..... | 135 |
| Figura A3. 10 Plano pieza Motor Bracket Top B..... | 136 |
| Figura A3. 11 Plano pieza Tips Side..... | 137 |
| Figura A3. 12 Plano pieza TipsTop A..... | 138 |
| Figura A3. 13 Plano pieza TipsTops B..... | 139 |
| Figura A3. 14 Plano pieza TipsTop C..... | 140 |
| Figura A3. 15 Plano pieza WDisc..... | 141 |
| Figura A4. 1 Ensamblaje final mano robótica..... | 142 |
| Figura A4. 2 Ensamblaje final mano robótica pulgar..... | 143 |
| Figura A4. 3 Ensamblaje final mano robótica dedos..... | 144 |
| Figura A5. 1 Panel Frontal pestaña EMG..... | 145 |
| Figura A5. 2 Diagrama de Bloques pestaña EMG (SELECT OPTION)..... | 145 |
| Figura A5. 3 Diagrama de Bloques pestaña EMG (WRITE FILE)..... | 146 |

| | |
|--|-----|
| Figura A5. 4 Figura E1.4 Diagrama de Bloques pestaña EMG (READ FILE) | 147 |
| Figura A5. 5 Panel Frontal pestaña ACCELEROMETER | 148 |
| Figura A5. 6 Diagrama de Bloques pestaña ACCELEROMETER | 148 |
| Figura A5. 7 Panel Frontal pestaña SENS HUB | 149 |
| Figura A5. 8 Diagrama de Bloques pestaña SENS HUB | 149 |
| Figura A5. 9 Panel Frontal pestaña MOTOR DRIVE | 150 |
| Figura A5. 10 Diagrama de Bloque pestaña MOTOR DRIVE (DC MOTOR OFF) | 150 |
| Figura A5. 11 Diagrama de Bloque pestaña MOTOR DRIVE (DC MOTOR ON AND LEFT) | 151 |
| Figura A5. 12 Diagrama de Bloque pestaña MOTOR DRIVE (STEPPER ON AND LEFT) | 152 |
| Figura A5. 13 Diagrama de Bloque pestaña MOTOR DRIVE (STEPPER ON AND RIGHT) | 153 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla 3. 1 Tabla de verdad del IC L293B..... | 42 |
|--|----|

Alvarez Delgado Julio Edwin
Trabajo de Graduación
Daniel Esteban Iturralde Piedra
Julio 2014

IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACTUADORES PARA UNA PRÓTESIS DE ANTEBRAZO A PARTIR DE SEÑALES ELECTROMIOGRÁFICAS

INTRODUCCION

Desde que el ser humano ha sufrido amputaciones por cualquier índole, se ha visto en la interminable búsqueda de un reemplazo para el miembro amputado, que permita recuperar, aunque sea en parte, algo de la funcionalidad del miembro perdido. Ejemplos de esta búsqueda se los puede encontrar en momias de hace 2000 años antes de cristo (revista la reunion, 2008), pasando por prótesis metálicas desarrolladas en los años 1500, hasta llegar al desarrollo de las prótesis articulares en 1818 por Peter Beil. Sin embargo, con la llegada de los microcontroladores y con el estudio de las señales electromiográficas, el desarrollo de prótesis inteligentes tomo un gran impulso hasta llegar a diseños como los propuestos por las compañías I-Limb o Be-bionic que ofertan dispositivos con prestaciones inimaginables hace una década atrás.

A pesar de los avances en prestaciones y funcionalidad que tienen en la actualidad las prótesis inteligentes existentes en el mercado, existe un gran problema con ellas, y es precisamente el hecho de que su principal objetivo, que es el ayudar a la gente a recuperar su movilidad, se ve reducido en gran parte por los costos que éstas prótesis poseen (pueden llevar a valer hasta 30000 dólares), y es que, hay que tomar en cuenta que por lo general una persona que sufre una amputación además de los problemas físicos y psicológicos que conlleva ésta pérdida, también sufren serios

problemas económicos pues al no poder emplear sus manos, su capacidad de trabajar y por ende su capacidad de ganar dinero también se ve reducida.

Con lo expuesto anteriormente, se propone el diseño de un sistema modular que permita adquirir señales electromiográficas y procesarlas para controlar actuadores eléctricos, de esta forma los costos de desarrollo y su posterior implementación se reducen, pues al ser modular permite escalar las funcionalidades iniciales sin tener que reemplazar todo el sistema. Además para el punto de vista del usuario final, si se desea hacer una mejora al sistema simplemente tendría que adquirir el modulo correspondiente y no una prótesis nueva.

También es necesario (sin perder de vista el enfoque del presente trabajo de investigación) de un sistema que permita hacer pruebas con los prototipos desarrollados, es por tal razón, que también se cuenta con un diseño bastante básico de una mano robótica con el fin de emular las prestaciones de prótesis inteligentes existentes en el mercado. Además como es necesario calibrar el sistema con los valores únicos que genera cada individuo, se cuenta con un software de control y pruebas que permite tener un enfoque adicional del comportamiento del sistema implementado.

El diseño y construcción de un prótesis inteligente constituye una empresa multidisciplinaria, que abarca varias áreas del conocimiento que se extienden mucho más allá del tema investigativo propuesto, sin embargo, el presente trabajo puede ser empleado como una guía inicial para cualquier emprendimiento futuro de este tipo.

CAPITULO 1:

REFERENTES TEORICOS

Para el desarrollo de cualquier proyecto bioelectrónico, es necesario conocer los aspectos básicos fisiológicos y funcionales del cuerpo humano que se desea estudiar o replicar, puesto que, con este conocimiento se puede determinar la mejor forma de aplicar una interface entre el cuerpo y la electrónica. Este capítulo pretende dar una breve recorrido por los aspectos anatómicos, electro génesis de las señales electromiográficas, métodos utilizados para la adquisición, análisis y clasificación de estas señales, además de una breve introducción al reconocimiento de patrones de movimiento utilizados en proyectos similares en la actualidad; todo esto sin perder el enfoque y la profundidad requerida para el tema de tesis propuesto.

1.1 Anatomía del antebrazo

En el desarrollo de un sistema de adquisición de señales mioeléctricas, es necesario familiarizarse con la terminología empleada en la medicina, ya sea para, la descripción de movimientos, así como con los nombres de los músculos involucrados en los mismos. De esta forma, se puede tener un panorama de las “mejores zonas” donde colocar los electrodos para adquirir una señal electromiográfica en determinada contracción muscular.

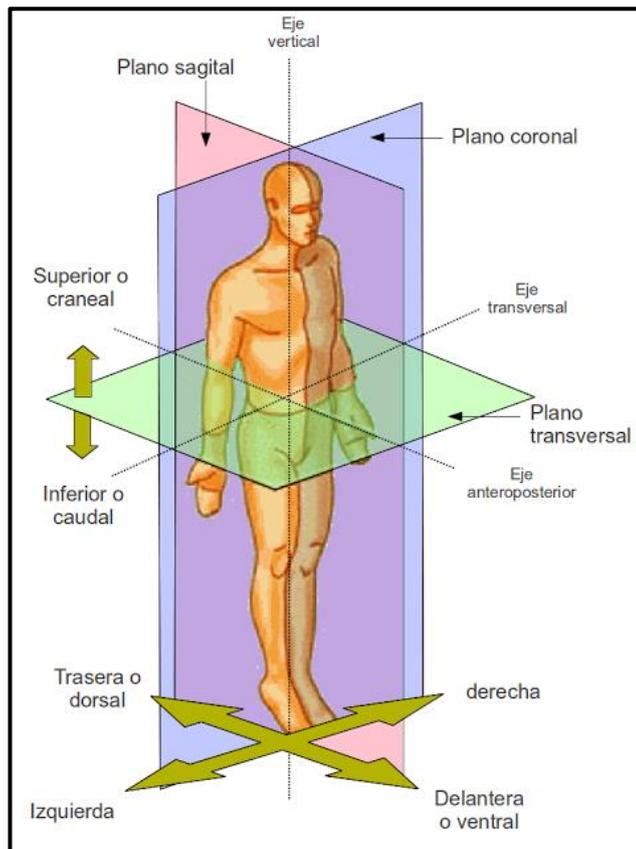
1.1.1 Tipos de movimiento

Para el estudio del movimiento de las extremidades, es muy común el uso de la terminología anatómica de localización para describir la posición relativa y movimiento de un órgano en el espacio. En la figura 1.1 se puede apreciar los

denominados “planos anatómicos del cuerpo humano” que, partiendo de la posición anatómica estándar, a manera de ejes cartesianos, dividen al ser humano en tres ejes:

- Eje vertical denominado “cráneo-caudal” que va desde la cabeza a los pies
- Eje transversal denominado “latero-lateral” que va de lado a lado
- Eje anteroposterior denominado “ventro-dorsal” que va de adelante hacia atrás.

Figura 1. 1 Planos anatómicos del cuerpo humano.



Fuente: <http://anatolandia.blogspot.com>

De igual manera que los ejes, también existen tres planos imaginarios que atraviesan el cuerpo humano:

- El plano sagital medio, es un plano vertical que atraviesa longitudinalmente el cuerpo y lo divide en dos mitades, derecha e izquierda.
- El plano frontal o coronal, es cualquier plano vertical perpendicular al plano sagital medio y permite dividir al cuerpo en dos mitades, anterior y posterior.

- El plano transversal, horizontal o axial, es perpendicular al eje longitudinal, sirve para dividir al cuerpo en dos zonas, superior e inferior.

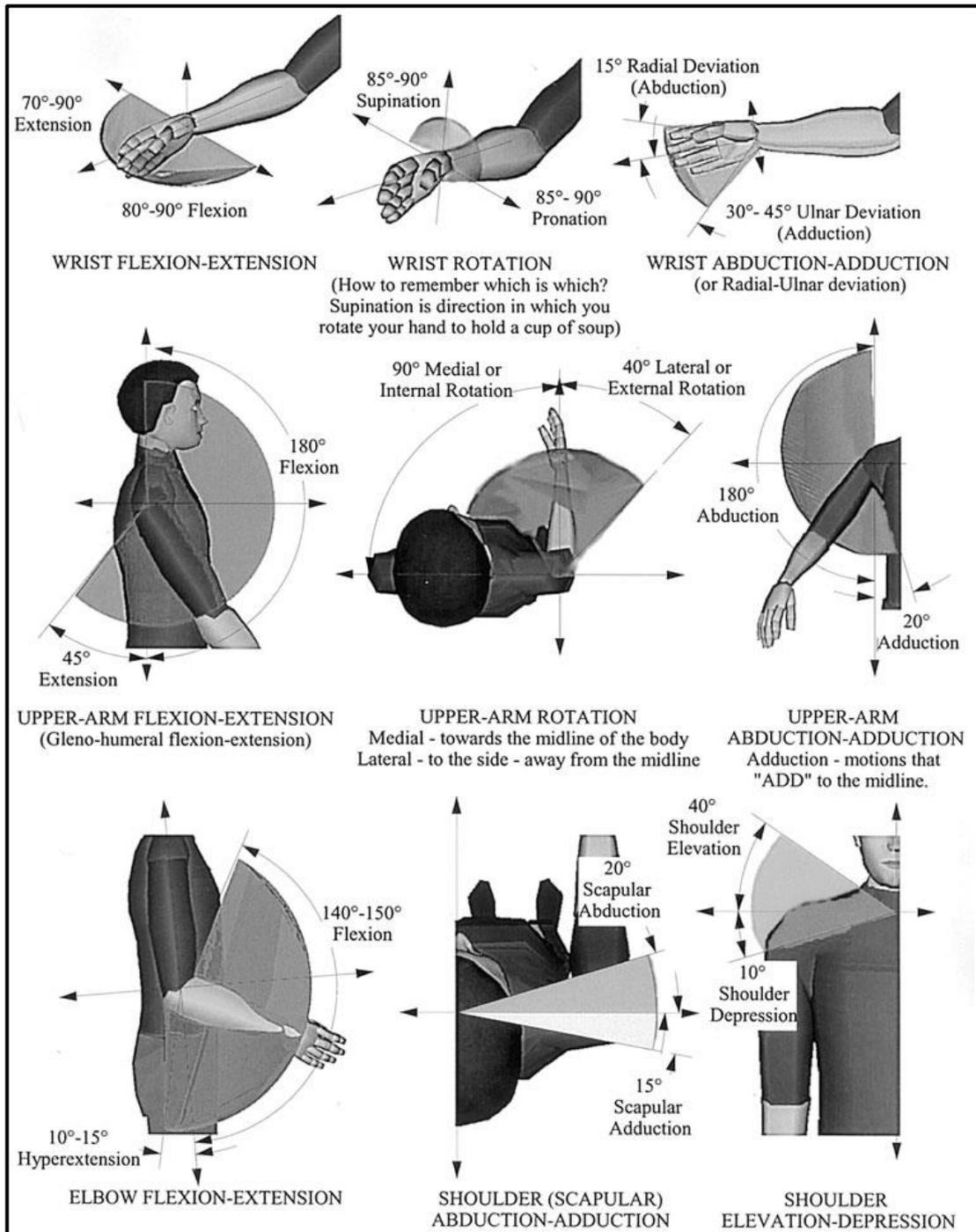
-

Como se mencionó con anterioridad, estos planos además de dividir las partes del cuerpo humano en posiciones relativas, también son empleados para determinar qué tipo de movimiento realiza una parte del cuerpo en el espacio (Weir, et al., 2009) (figura 1.2). Estos a su vez se dividen en tres clases:

- Deslizamiento, es cuando dos superficies articulares pasan una sobre otra sin girar ni describir ningún ángulo.
- Movimientos angulares, a su vez se dividen en abducción, aducción, extensión y flexión. Abducción, es cuando se aleja un segmento del eje medio o plano sagital medio, por ejemplo, separar la pierna derecha hacia ese mismo lado. Aducción, es cuando se acerca un miembro al plano sagital medio, por ejemplo, juntar el brazo hacia el tronco.
- Flexión, es aproximar un segmento hacia otro disminuyendo el ángulo formado entre ellos, por ejemplo, aproximar el antebrazo hacia el hombro por medio de los bíceps. Extensión, es lo opuesto a la flexión, es decir, aumentar el ángulo formado por dos segmentos sin sobrepasar los 180°, por ejemplo, alinear el antebrazo con respecto al brazo por medio de los tríceps.
- Movimiento de rotación, es cuando un miembro rota sobre su eje, por ejemplo, la rotación del cubito y el radio permiten ubicar a la palma de la mano hacia arriba (supinación) o hacia abajo (pronación).

Existe otro grupo de movimientos denominados “movimientos especiales” pero su contenido va más allá del objetivo de este trabajo.

Figura 1. 2 Rangos de movimientos de muñeca, parte superior del brazo, hombros y codos.



Fuente: Biomedical Engineering and Design Handbook

1.1.2 Músculos y mecanismos de movimiento

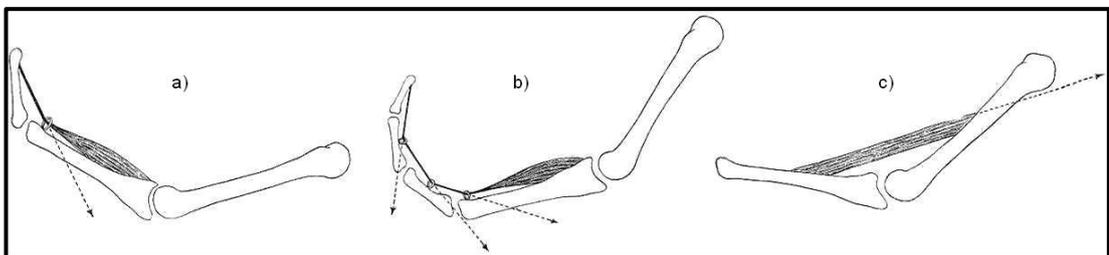
El cuerpo humano tiene varios tipos de músculos que realizan diversas funciones específicas, sin embargo, para no perder el enfoque del tema propuesto, se pone

énfasis en aquellos músculos que producen el movimiento de las extremidades superiores y en particular del antebrazo.

Los músculos esqueléticos, como su nombre lo indica, son órganos contráctiles conectados de forma directa o indirecta a los huesos; su principal función es la de determinar los movimientos de diversas partes del esqueleto, además de mantener la posición y postura del cuerpo. Se los suele definir como voluntarios, puesto que su acción puede ser controlada directamente por el cerebro mediante el sistema nervioso central (en el apartado 1.2 Señales electromiográficas, se abarcará con más detalle el proceso que involucra el movimiento de un músculo).

Como ya se acoto en párrafos anteriores, este tipo de músculos se conectan a los huesos, cartílagos, ligamentos e inclusive la piel, tanto de forma directa o mediante tendones (figura 1.3). Dependiendo del tipo de conexión el movimiento se clasifica en directo e indirecto, siendo este último, bastante similar a un sistema mecánico conformado por poleas y resortes.

Figura 1. 3 Conexión indirecta (a y b) y conexión directa (c) de músculos esqueléticos.



Fuente: <http://anatolandia.blogspot.com>

Es importante destacar que, no necesariamente la interacción de los miembros con sus músculos circundantes es un factor determinante para analizar los movimientos generados en un miembro. Aunque a primera vista el movimiento de un músculo puede ser muy notorio, no se puede afirmar que éste sea el responsable principal de una acción motora; esto se debe a que, a pesar de que un movimiento puede ser muy definido, son varios los músculos que intervienen para realizar una acción motora.

En el sistema nervioso central, no existen ordenes específicas para el movimiento de un músculo en particular, sino de acciones motoras a desarrollar, es decir, para

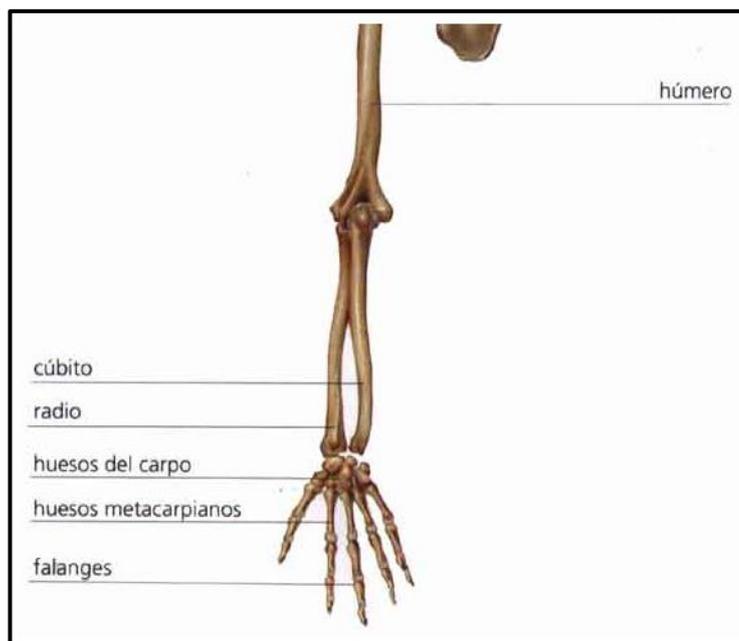
realizar determinada acción (rotar la muñeca por ejemplo), se ejecuta una combinación de acciones de varios músculos involucrados en este movimiento, la persona no tiene la capacidad de agregar o disminuir la cantidad de músculos involucrados en determinada acción motora. Esta característica se debe tener en cuenta para determinar derivaciones apropiadas para obtener aquellas señales electromiográficas relevantes a la acción motora que se desee adquirir.

1.1.3 Musculatura y articulación de antebrazo y mano

Los movimientos desarrollados por las extremidades son, en esencia, la combinación de la acción de varios músculos a la vez, teniendo esto en cuenta, y considerando el objetivo del tema planteado, es necesario conocer la anatomía del antebrazo y mano para tener una noción básica de cómo interactúan estos músculos entre sí.

El esqueleto del antebrazo (figura 1.4) está constituido por dos huesos longitudinales relativamente paralelos denominados Radio y Cúbito, mientras que la mano está conformada por 27 huesos agrupados en tres áreas (carpo, metacarpo y falanges). Las uniones e interacciones de estos huesos mediante las articulaciones permiten realizar la mayoría de los movimientos

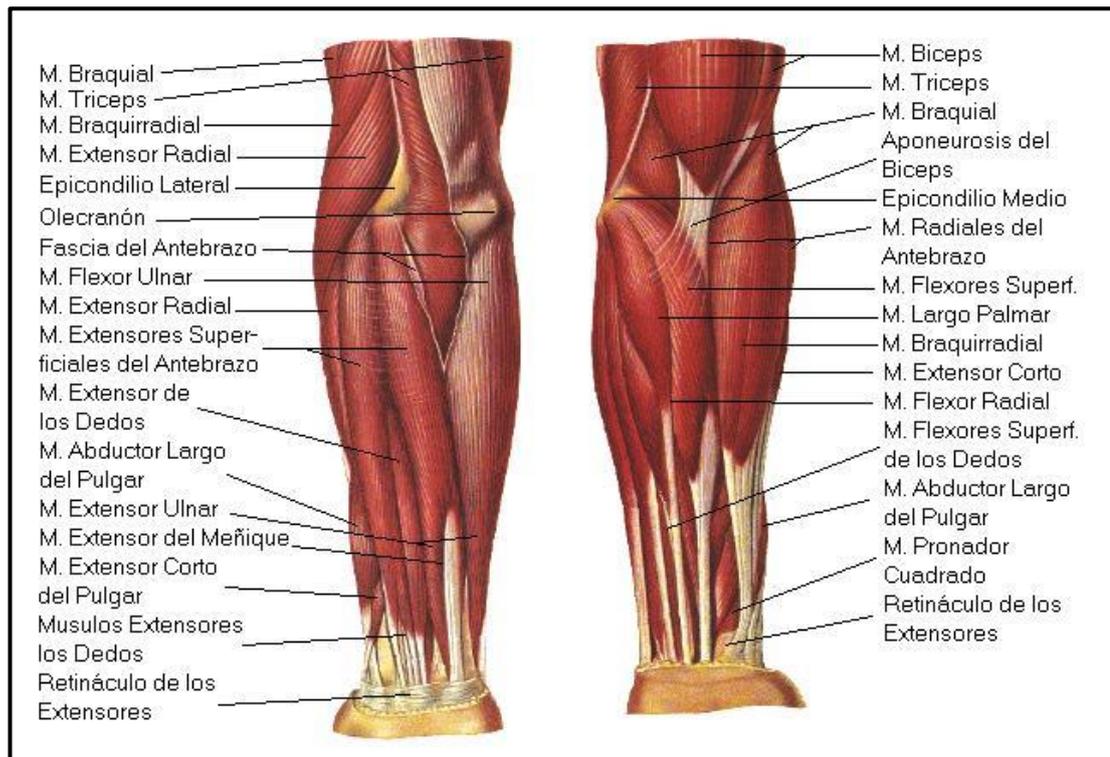
Figura 1. 4 Huesos antebrazo y mano.



Fuente: <http://educacion.uncomo.com>

Los músculos del antebrazo son los encargados de accionar estas articulaciones de forma directa como indirecta. La disposición de estos músculos es sumamente compleja y entrelazada (figura 1.5) y precisamente la mayoría de los músculos relevantes para el desarrollo de una prótesis de mano, se encuentran en la capa muscular interna, por lo que resulta bastante difícil identificar cuál de ellos está actuando en determinado movimiento en particular.

Figura 1. 5 Músculos del antebrazo.



Fuente: tesisdeinvestigadores.blogspot.com

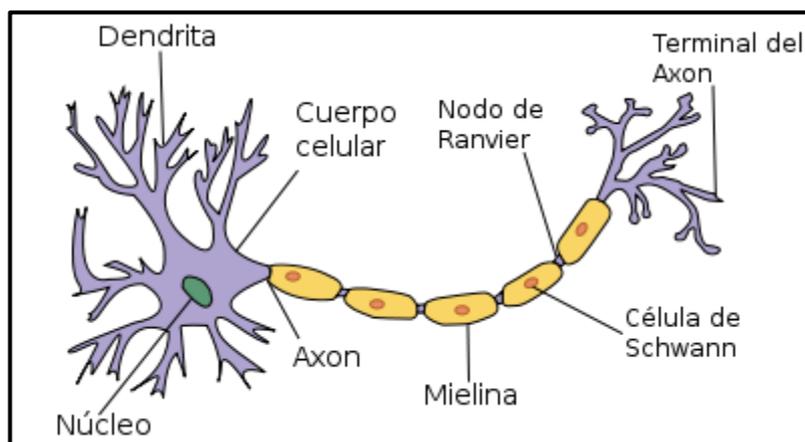
1.2 Señales electromiográficas

La comprensión de la relación existente entre los impulsos nerviosos y la contracción de los músculos, es la base fundamental, para el desarrollo de cualquier dispositivo que utilice señales electromiográficas, de hecho, la propagación de estos impulsos a los tejidos circundantes cuando existe una contracción muscular, hace posible su registro mediante el uso de electrodos diseñados específicamente para tal fin. En el presente apartado se explicará con mayor profundidad como se generan estas señales en el cuerpo humano.

1.2.1 Generación del potencial de acción

Las neuronas son células especializadas en la transmisión y conducción de impulsos, constituyen la base del sistema nervioso. En la figura 1.6 se puede apreciar la estructura de una neurona, en donde el axón es la vía por donde transitan los impulsos nerviosos o potenciales de acción hacia la siguiente célula. Los axones pueden agruparse formando fibras nerviosas.

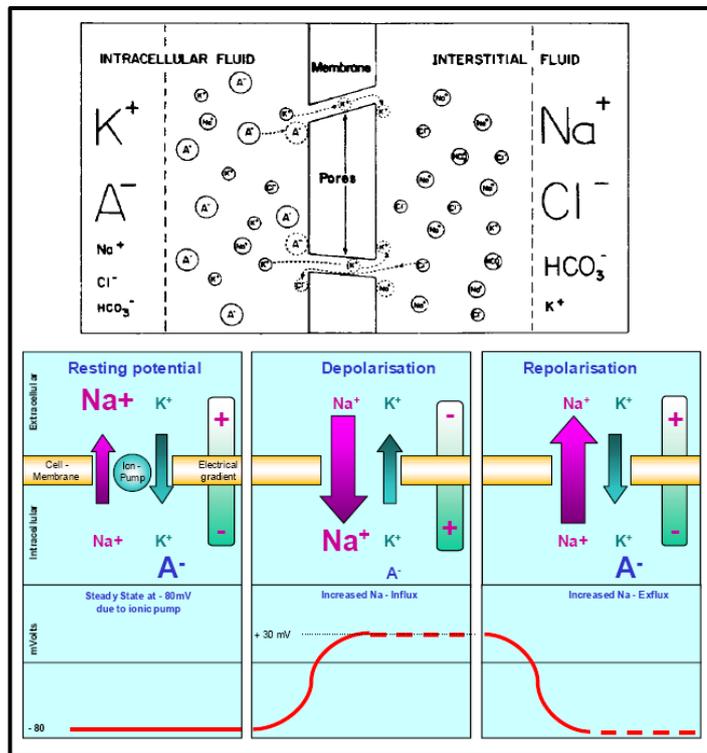
Figura 1. 6 Estructura de una neurona.



Fuente: www.innovacd.eu

Cuando se produce un estímulo, ya sea interno o externo, en una célula sensorial o en las neuronas, se genera una onda eléctrica denominada impulso nervioso. En el caso específico de las neuronas, este estímulo se genera por una descompensación de los iones existentes en ambos lados de la membrana celular. La descompensación se produce por el incremento o disminución de los niveles de Sodio (Na^+) y de Potasio (K^+) en ambos extremos de la membrana celular (figura 1.7) generando así una diferencia de potencial.

Figura 1. 7 Diferencia de potencial generada en el músculo por variación de los niveles de sodio y potasio en los extremos de la membrana celular.



Fuente: scienceblogs.com

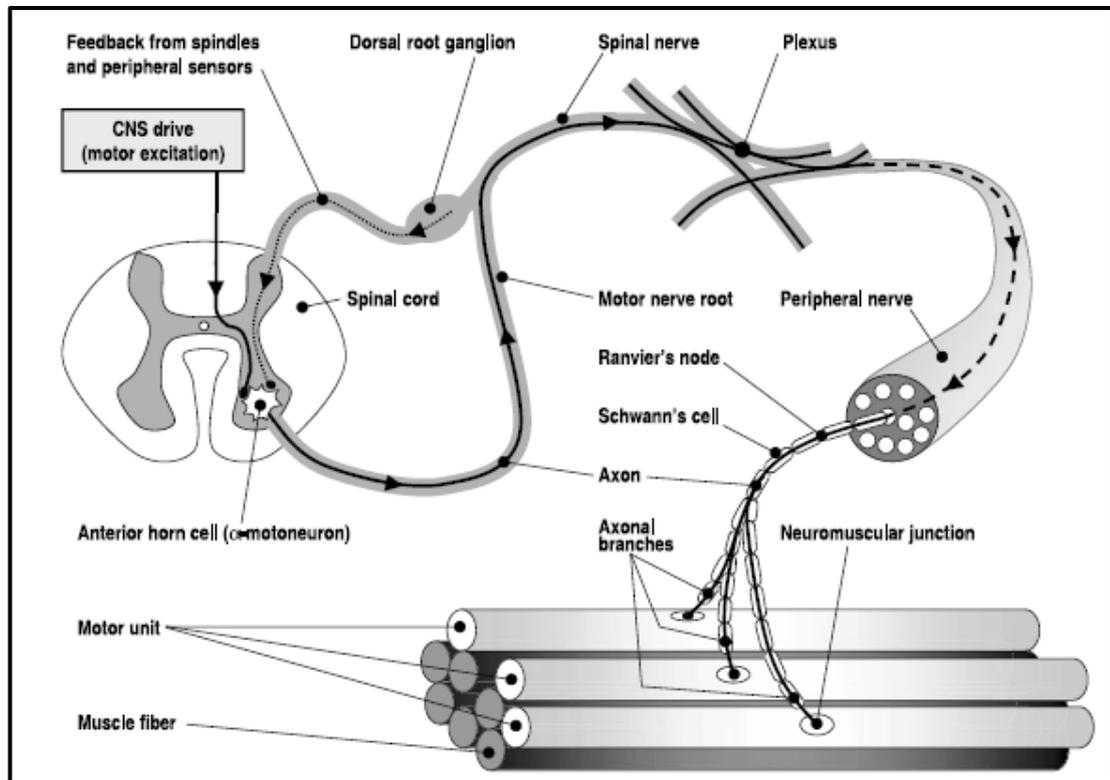
Como se puede ver en la figura anterior, existen tres estados musculares dependiendo de los cambios químicos existentes en los extremos de la membrana celular:

- Reposo, cuando los niveles de Na^+ y K^+ están en balance.
- Depolarización, se produce un incremento en el nivel de Na^+ provocando la contracción muscular.
- Repolarización, se produce cuando un musculo cesa de realizar una acción y retorna a su posición original

1.2.2 Potencial de acción de la unidad motora

Los músculos esqueléticos, están compuestos de unidades funcionales llamadas motores unitarios (MU por sus siglas en ingles "Motor Units"). Un MU está conformado por una motoneurona cuyo axón está conectado a las fibras musculares (figura 1.8), las cuales se contraen cuando su correspondiente MU es activada.

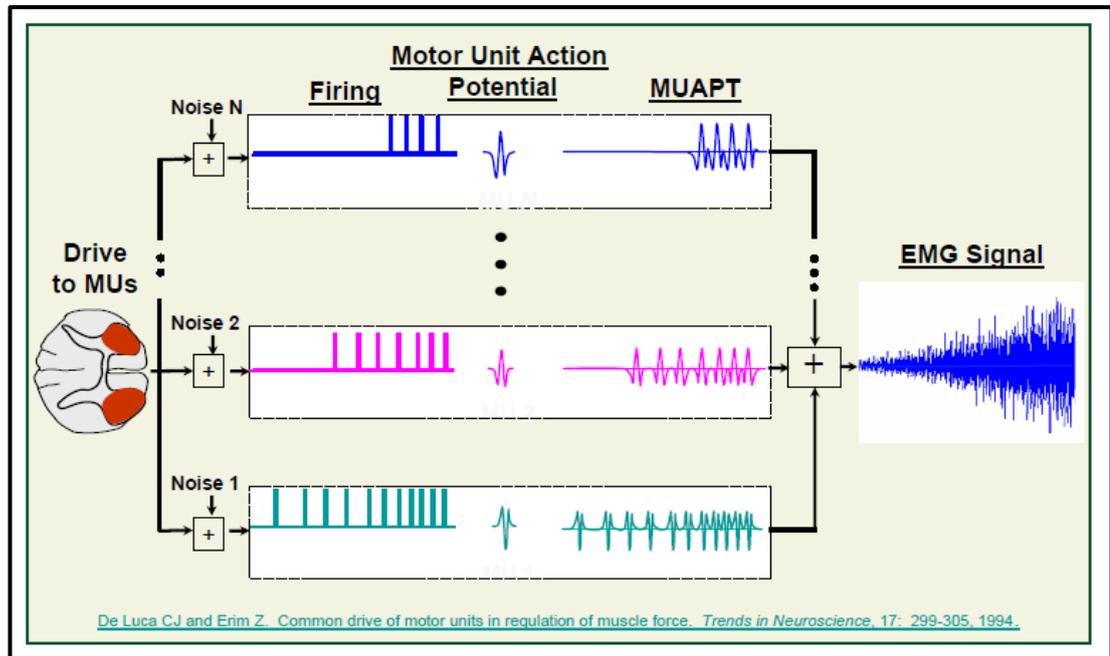
Figura 1. 8 Conexión existente entre una motoneurona y las fibras musculares.



Fuente: en.wikibooks.org

El potencial de acción motora (MUAP “Motor Unit Action Potential”), se define como la señal eléctrica que emite la activación de las fibras musculares de una MU. La presencia de un MUAP produce una contracción de las fibras musculares afectadas, sin embargo, para que se mantenga esta contracción en el tiempo, es necesario que las unidades motoras sean accionadas repetidamente. Este accionamiento repetitivo, hace que se cree un “tren de potenciales de acción de la unidad motora” (MUAPT por sus siglas en inglés), el cual se mantendrá mientras el músculo requiera generar fuerza (figura 1.9).

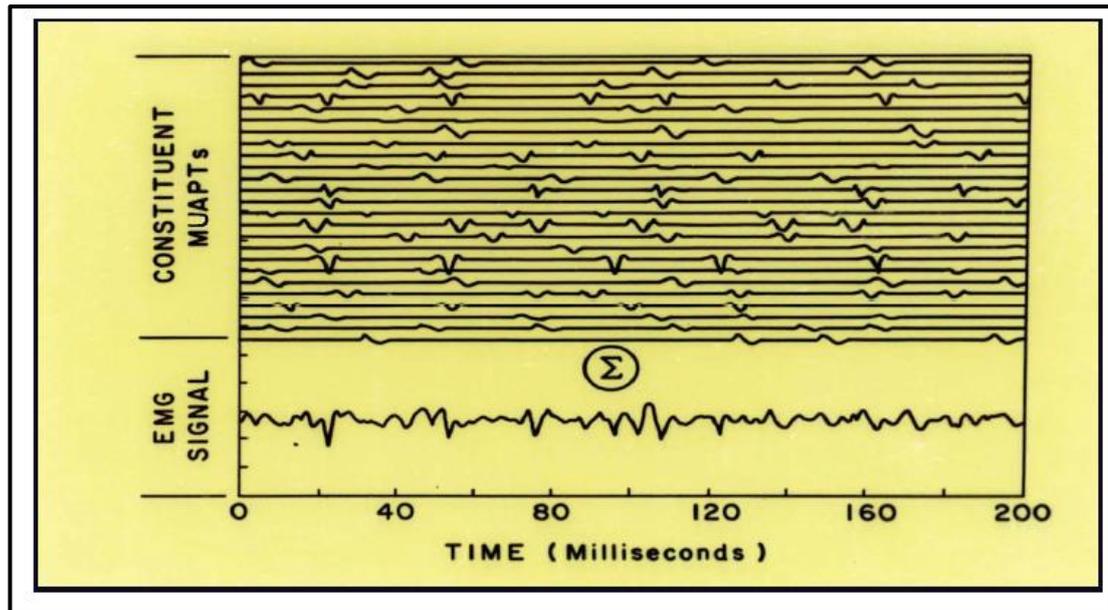
Figura 1. 9 Potencial de acción del motor unitario MUAP y tren de potenciales de acción de motores unitarios MUAPT.



Fuente: delsys.com

La sumatoria de los MUAPT activos en determinado espacio de tiempo generan una señal electromiográfica (EMG) (figura 1.10). Cuando se requiere mayor cantidad de fuerza en el músculo, mayor número de unidades motoras son activadas y por ende se incrementa la tasa de activación de fibras musculares (en inglés, "firing rate").

Figura 1. 10 Sumatoria de MUAPTS para formar una señal EMG.



Fuente: delsys.com

1.2.3 Electrodo y derivaciones propuestas

En tema anterior, se explicó la génesis de las señales EMG, sin embargo, para poder registrar estas señales es necesario un elemento que funcione como interface entre el cuerpo y el dispositivo de adquisición; precisamente los electrodos cumplen con tal función.

Un electrodo recoge las señales eléctricas existentes en el musculo durante una contracción y las envía al instrumento de adquisición. Según el método de registro, se los puede dividir en electrodos de inserción y electrodos superficiales, los primeros suelen ser agujas o finos filamentos que son insertados dentro de los músculos, que si bien es cierto, adquieren señales muy precisas, su uso e implementación requiere de procedimientos quirúrgicos (figura 1.11), mientras que los electrodos superficiales simplemente se los coloca sobre la piel.

Figura 1. 11 Electrodo de inserción.



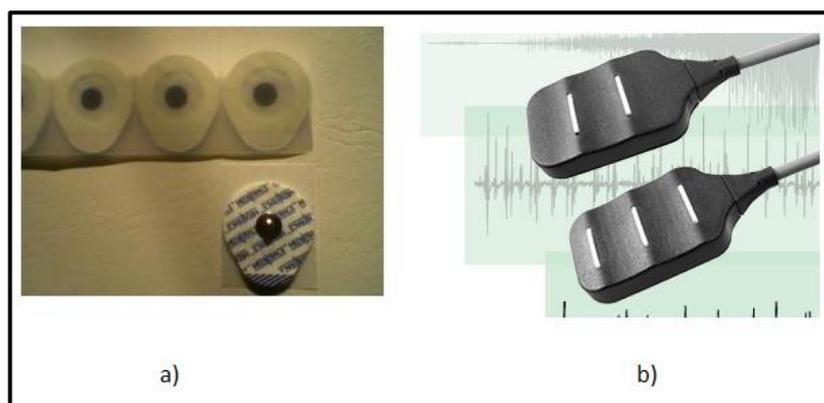
Fuente: www.girishsurgical.com

Por las razones expuestas en el párrafo anterior, para la adquisición de las señales EMG se optará por el uso de electrodos superficiales, ya que únicamente se requiere colocarlos sobre la piel y no es necesario intervenciones quirúrgicas en absoluto.

En el mercado, existen básicamente dos tipos de electrodos superficiales (figura 1.12):

- Electrodos pasivos, que simplemente son discos de plata adheridos a la piel.
- Electrodos activos, incluyen en el electrodo un amplificador de alta impedancia para obtener mejores mediciones al reducir la impedancia de la interface piel-electrodo.

Figura 1. 12 Tipos electrodos superficiales, a) electrodos pasivos b) electrodos activos Delsys.



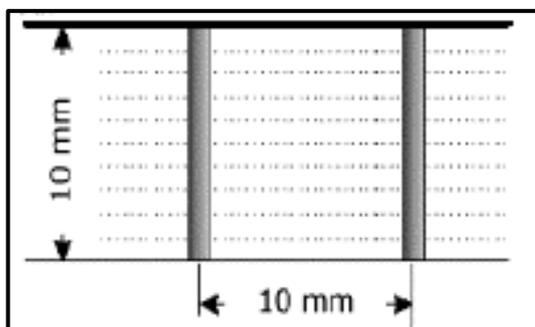
Fuente: delsys.com

Según el tipo de geometría y su cantidad, a los electrodos superficiales para la medición de señales EMG se los puede agrupar en electrodos monopolares,

bipolares, tripolares y multipolares. La clasificación a utilizar durante el desarrollo de este proyecto de tesis, será la de electrodos bipolares.

Para implementar la modalidad bipolar, se colocan dos electrodos a las entradas específicas del amplificador diferencial (figura 1.13), además de un tercer punto de conexión de referencia conocido como RLD (“Right Leg Drive”). Como se explicó anteriormente, no es posible registrar únicamente la señal de un musculo en particular, de hecho, lo que se obtiene es la superposición de varias señales, generadas en músculos adyacentes al punto de interés. Al utilizar una configuración bipolar, estas señales generadas por los músculos adyacentes son eliminadas en gran parte por el amplificador diferencial (De Luca, 2002).

Figura 1. 13 Configuración bipolar.



Fuente: delsys.com

1.3 Métodos de adquisición de señales mioeléctricas

Para la correcta adquisición de señales EMG, se debe encontrar un balance entre los métodos físicos (ubicación de electrodos por ejemplo) y los métodos electrónicos (filtrado, amplificación, etc.) para obtener una señal de calidad. En el presente apartado se explica los requerimientos necesarios para obtener señales EMG con la menor distorsión posible.

1.3.1 Consideraciones físicas

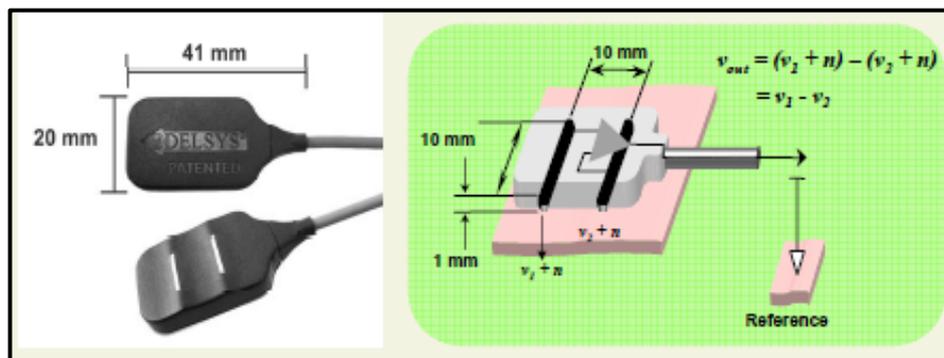
Básicamente todas las consideraciones físicas están encaminadas en la prevención de la inducción del ruido y “Cross-Talk”, al momento de adquirir señales EMG. A continuación se explican varias recomendaciones a tener en cuenta al momento de adquirir señales EMG.

La inducción de ruido del medio ambiente (radiofrecuencias, líneas de poder, etc.) se reduce en gran medida con el uso de cables apantallados, cuya longitud, en el caso específico de este proyecto, no debe ser mayor a los 30 cm.

Es muy recomendable ubicar la etapa de pre amplificación, si es posible, en la carcasa de los electrodos.

Cuando se utiliza una configuración bipolar de electrodos, la distancia entre estos debe ser aproximadamente de 1 cm (figura 1.14).

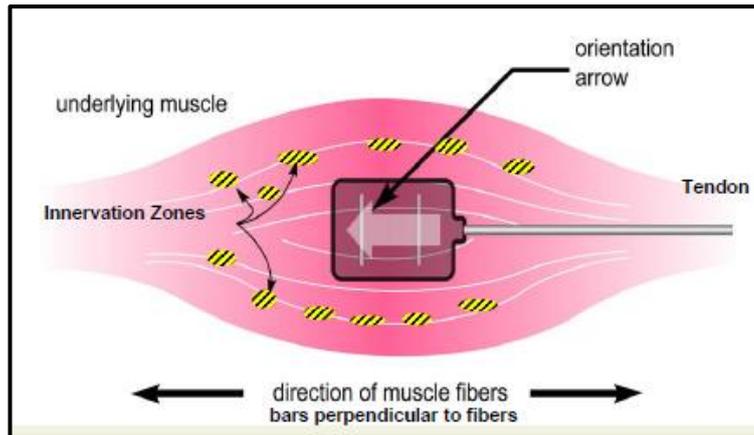
Figura 1. 14 Ubicación bipolar de electrodos, en el caso de los electrodos Delsys, estos contienen dos electrodos y la pre amplificación en su carcasa



Fuente: delsys.com

Los electrodos se deben ubicar en el medio del músculo, perpendicularmente a las fibras musculares y lo más alejados de las zonas de inervación y de los orígenes de los tendones (De Luca, 2008), de esta forma se garantiza obtener las mayores amplitudes de señal cuando éste sea contraído (figura 1.15).

Figura 1. 15 Ubicación de los electrodos en un músculo



Fuente: delsys.com

Se recomienda limpiar la superficie que se desea registrar con alcohol, además, en lo posible se debe eliminar los vellos existentes para evitar distorsiones en la señal EMG

1.3.2 Consideraciones electrónicas

Al adquirir señales EMG, lo que se desea es obtener la máxima cantidad de información relevante de la señal EMG y a su vez, la mínima cantidad de ruido eléctrico presente en ella. Sin embargo, la maximización de la tasa de señal a ruido debe ser hecha afectando en lo mínimo a la señal EMG (De Luca, 2002). A continuación se explican los aspectos electrónicos básicos a tener en cuenta para el desarrollo de un sistema de adquisición de señales EMG (De Luca, 2002).

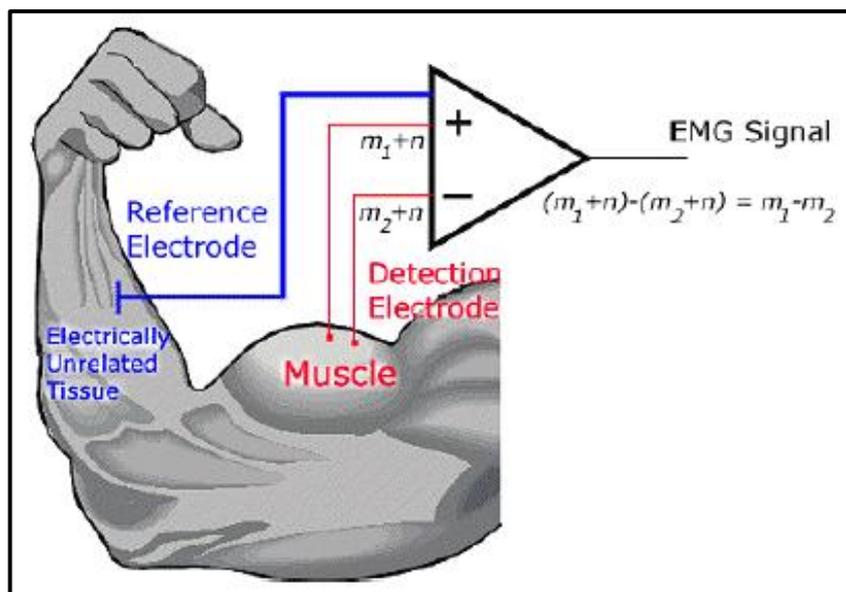
1.3.2.1 Amplificación diferencial

Para eliminar la mayoría del ruido inducido en la señal, se recurre al uso de técnicas de amplificación diferencial (figura 1.16). Con esta técnica, se obtienen dos señales detectadas en dos sitios, se sustrae las señales obtenidas y la diferencia es amplificada, por lo tanto, si se tiene una señal que se origina lejos del punto de

detección, aparecerá como una señal común para los dos puntos (ruido) y será suprimida, amplificando únicamente la señal EMG.

Sin embargo, no cualquier amplificador diferencial substrahe las señales de forma perfecta. Para tal fin se recurre al uso del CMRR (en inglés, Common Mode Rejection Ratio) como parámetro para determinar la eficiencia de un amplificador para restar el ruido. Un valor CMRR de 90dB es suficiente para eliminar el ruido eléctrico externo.

Figura 1. 16 Configuración de amplificador diferencial donde m es la señal EMG y n es el ruido.



Fuente: delsys.com

1.3.2.2 Impedancia de entrada

La impedancia existente entre la unión de la piel seca y la superficie de detección del electrodo, se encuentra en un rango entre los miles de ohmios hasta varios mega ohmios, esto hace que se produzcan distorsiones o atenuaciones en la señal que se desea adquirir. Para prevenir esto, la impedancia de entrada de los amplificadores diferenciales debe ser lo más grande posible (en la actualidad, existen dispositivos electrónicos que presentan una impedancia de 10^{12} ohms en paralelo a un condensador de 5pF).

1.3.2.3 Filtrado

Una etapa de filtrado es recomendable para eliminar el ruido que pudiera estar presente (a pesar de las consideraciones anteriores) en la señal adquirida. Ya que las señales EMG se encuentran entre los 20 a los 500 Hz se recomienda un filtrado entre esas frecuencias con un roll-off de 12 dB/oct. Adicionalmente, se debe ubicar una etapa de amplificación adicional inmediatamente después de la etapa de filtrado.

1.4 Métodos de análisis de señales mioeléctricas

Dependiendo del enfoque y la capacidad de procesamiento que se disponga, se pueden emplear diversos métodos para el análisis de señales EMG, sin embargo, a pesar del amplio espectro de formas de abarcar el análisis de señales EMG, se los puede agrupar en métodos de dominio del tiempo, dominio de la frecuencia y tiempo-escala (Beng Guey, 2009).

1.4.1 Dominio del tiempo

El análisis de la señal EMG en el dominio del tiempo es la técnica más utilizada cuando se trata de detectar la activación de un músculo. Existen varios parámetros que pueden ser extraídos de la señal mediante las siguientes técnicas:

Rectificación de onda completa, una de las operaciones básicas cuando se trata de señales EMG superficiales (sEMG), es la rectificación de onda completa que puede ser realizada digitalmente empleando:

$$sEMG_{rec}(i) = |sEMG(i)| \quad (\text{Ecuación 1.1})$$

Valor eficaz o RMS (del inglés, Root Mean Square), se lo emplea para determinar la amplitud y sirve en algunos casos como indicador de fuerza de la señal sEMG.

$$RMS = \sqrt{\frac{\sum_{i=0}^n A_i^2}{N}} \quad (\text{Ecuación 1.2})$$

Donde A_i es la amplitud de la i muestra y N corresponde al número total de muestras. El valor medio absoluto MAV (en inglés, Mean Absolute Value), sirve para estimar el valor MAV de la señal x_i en el segmento i que contiene N muestras de largo.

$$\bar{X}_i = \frac{1}{N} \sum_{i=1}^N |X_i| \quad (\text{Ecuación 1.3})$$

Amplitud Wilson, es básicamente un conteo por cada cambio realizado en la amplitud de la señal sEMG cuyo valor exceda a un valor umbral predeterminado, esta función es muy importante ya que se la puede utilizar como un indicador de disparo del MUAP, por ende, es un indicador del nivel de contracción muscular.

$$WAMP = \sum_{i=1}^N f(|x_i - x_{i-1}|) \quad (\text{Ecuación 1.4})$$

Donde $f(x) = 1$ si $x > \text{umbral}$, caso contrario su valor es de 0.

Varianza, es una medición del poder de la señal y es calculada por:

$$VAR = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (\text{Ecuación 1.5})$$

Longitud de onda, es determinada por:

$$l_0 = \sum_{i=1}^N |\Delta x_i| \quad (\text{Ecuación 1.6})$$

Donde $\Delta x_i = x_i - x_{i-1}$. Este parámetro otorga una medición de la amplitud, frecuencia y duración de la forma de onda en una sola ecuación.

1.4.2 Dominio de la frecuencia

El análisis de señales sEMG en el dominio de la frecuencia (también llamado espectral), es ampliamente empleado para el estudio de la fatiga muscular. Por lo tanto, el uso de este tipo de análisis se limita a escenarios cuando oscilaciones o patrones repetitivos de movimiento están presentes, por ejemplo en el caso de temblores musculares. Por tal motivo, y tomando en cuenta la meta del presente

trabajo de investigación, este apartado únicamente se limitara a introducir, a manera informativa, algunos aspectos básicos utilizados en este tipo de análisis.

El análisis de Fourier permite transformar una señal desde el dominio del tiempo al dominio de la frecuencia, esto se logra descomponiendo a la señal en sinusoides consecutivas a diferentes frecuencias. La transformada de Fourier se expresa como:

$$F(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} \quad (\text{Ecuación 1.7})$$

Donde ω es la frecuencia angular ($\omega = 2\pi f$), f es la frecuencia de entrada, $x(t)$ es la señal en el dominio del tiempo y $F(\omega)$ corresponde a la transformada de Fourier en el dominio de la frecuencia.

Además de la transformada de Fourier, existen otros parámetros a considerar en el análisis de una señal sEMG en el dominio de la frecuencia:

- Frecuencia media, es utilizado como un indicador de la fatiga muscular, se expresa como:

$$\bar{F} = \frac{\sum_{i=0}^n f_i A_i^2}{\sum_{i=0}^n A_i^2} \quad (\text{Ecuación 1.8})$$

Donde \bar{F} es la frecuencia media, f_i y A_i corresponden a la frecuencia y amplitud en la muestra i , por último, n representa el número total de muestras.

- Frecuencia mediana, es otro parámetro que se puede usar para estudiar la fatiga muscular, es la frecuencia a la que se divide el espectro de potencia en dos regiones que poseen el mismo nivel de potencia.

1.4.3 Dominio tiempo-escala

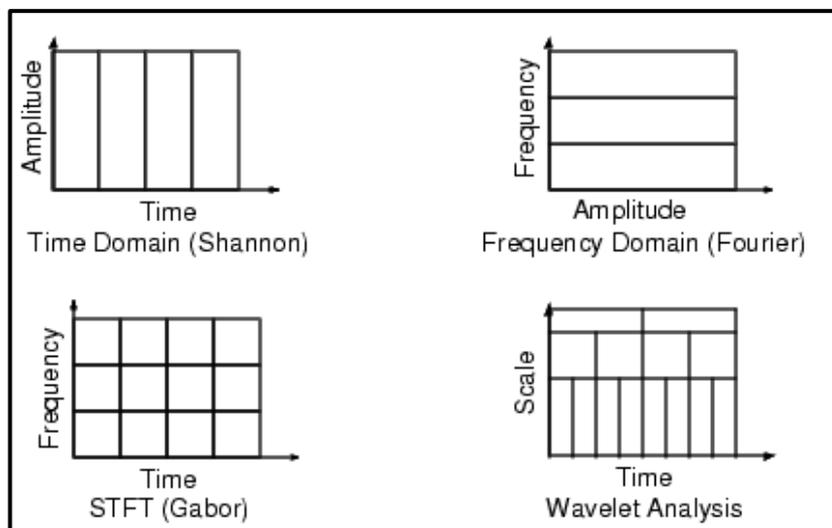
El principal problema de los dos tipos de análisis anteriormente expuestos, es que son excluyentes entres sí, es decir, cuando se analiza una señal en el dominio del tiempo es imposible determinar sus características de frecuencia y cuando se realiza una transformada de Fourier (a excepción de la transformada de tiempo corto de

Fourier) para pasar del dominio del tiempo al dominio de la frecuencia a la señal de interés, se pierde toda la información del dominio del tiempo que estaba contenida en ella.

Al usar una transformada de tiempo corto de Fourier STFT (en inglés, Short Time Fourier Transform) se puede determinar dónde y a qué frecuencia ocurre un evento en la señal estudiada, no obstante, al emplear una técnica de “ventana” para realizar dicho análisis, su resolución está limitada por el tamaño de la ventana y por las características estacionaria o no estacionaria de la señal (por ejemplo, las señales sEMG no son estacionarias). El uso de las transformadas de Wavelet (WT por sus siglas en inglés) permite solucionar este problema ofreciendo una resolución flexible de tiempo-frecuencia usando un análisis de tiempo-escala.

La WT es una herramienta matemática que permite al análisis focalizado o local de una señal no estacionaria y de rápida transición, empleando regiones de tiempo-escala en lugar de regiones de tiempo-frecuencia para el análisis de señales, lo que proporciona una resolución flexible de tiempo-frecuencia (figura 1.17).

Figura 1. 17 Dominios para el análisis de señales.

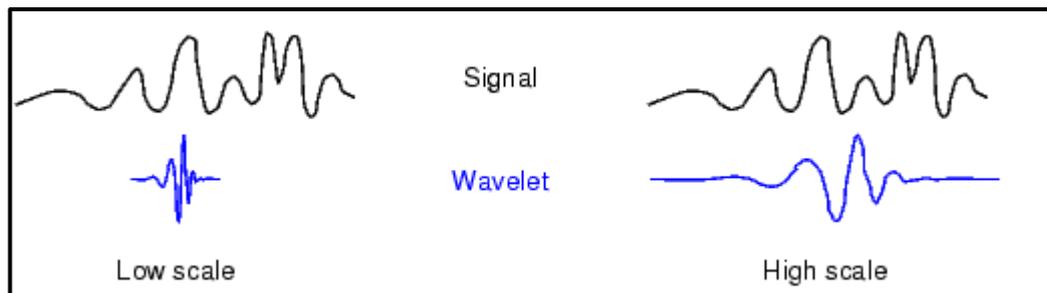


Fuente: Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano.

Las transformadas de Wavelet convierte a una señal analizada en una representación de tiempo-escala, esto permite, presentar la información contenida de la señal en una forma más útil.

En el análisis de una señal mediante WT, se puede apreciar una relación entre escala y frecuencia; a baja frecuencia de la señal, se utiliza altas escalas obteniendo así una buena resolución en frecuencia, mientras que, a altas frecuencias, se utilizan bajas escalas obteniendo así una buena resolución en el tiempo (Romo, et al., 2007) (figura 1.18).

Figura 1. 18 Relación escala-frecuencia en análisis de señales usando WT.



Fuente: Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano

Dependiendo del enfoque con el que es analizada la señal, se puede dividir a las transformadas de Wavelet en continuas, discretas y de paquetes (Romo, et al., 2007), a continuación se describen sus características básicas.

1.4.3.1 Transformada Wavelet continua CWT

La transformada Wavelet continua, expresa una señal $x(t)$ (continua en el tiempo y de energía finita) mediante una expansión de términos con coeficientes proporcionales al producto interno entre la señal $x(t)$ y versiones desplazadas y escaladas de una función prototipo $\Psi(t)$ (de energía finita) denominada Wavelet madre (Romo, et al., 2007), obteniendo:

$$CWT(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \Psi\left(\frac{t-b}{a}\right) dt \quad (\text{Ecuación 1.9})$$

Donde a y b son reales, $a \neq 0$ y “ a ” corresponde a una variable de escala que permite comprimir o dilatar la función $\Psi(t)$, por lo tanto establece la resolución a la cual es analizada la señal $x(t)$, “ b ” es una variable de desplazamiento que indica la ubicación de la función $\Psi(t)$ sobre la señal analizada.

1.4.3.2 Transformada Wavelet discreta DWT

El uso de transformadas Wavelet discretas se da más por razones de índole computacional y de practicidad, básicamente esta transformada es una discretización de una CWT, mediante una escala diádica en donde $a = 2^{-j}$ y $b = k2^{-j}$ de tal manera que la ecuación 1.9 se convierta en:

$$\Psi_{j,k}(t) = 2^{\frac{j}{2}}(2^j t - k) \quad (\text{Ecuación 1.10})$$

En donde $2^{\frac{j}{2}}$ es la constante de normalización y garantiza la condición de ortonormalidad (Romo, et al., 2007). De igual forma, la función $\Psi(t)$ lleva asociada consigo una función escala $\phi(t)$, de esta forma es posible aproximar cualquier función $x(t) \in L^2(\mathfrak{R})$ mediante la expresión:

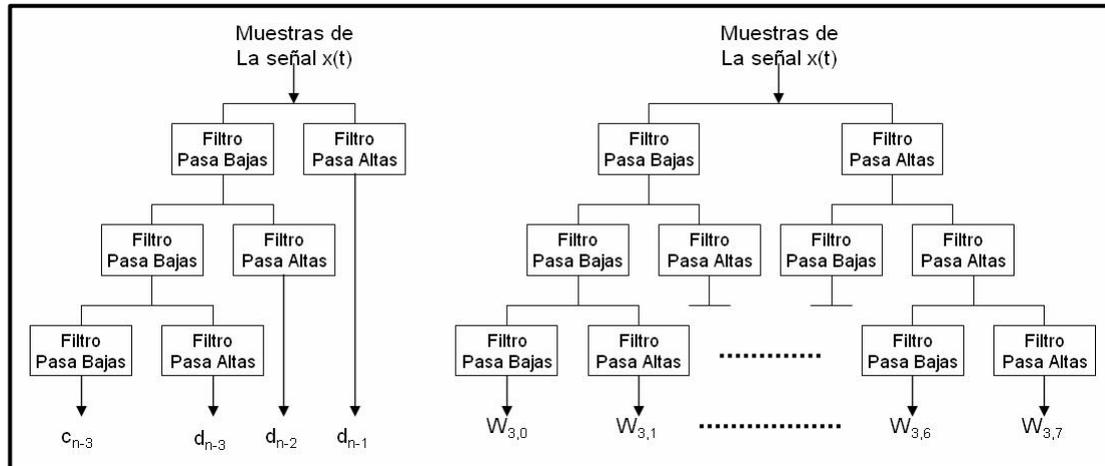
$$x(t) = \sum_k \sum_j c_{j,k} \phi(t) + \sum_k \sum_j d_{j,k} \Psi(t) \quad (\text{Ecuación 1.11})$$

Donde $c_{j,k}$ son los coeficientes escala o aproximación y $d_{j,k}$ son los de Wavelet o detalle de la señal original $x(t)$ con respecto a $\phi(t)$ y $\Psi(t)$.

1.4.3.3 Transformada de paquetes de Wavelet WPT

Al emplear algoritmos de “bancos de filtros” (Romo, et al., 2007) en DWT (figura 1.19 a), se reitera el proceso de filtraje únicamente en los coeficientes de escala $c_{j,k}$ dejando intactos los coeficientes de detalle $d_{j,k}$ en cada nivel de descomposición. En el caso de la transformada de paquetes de Wavelet (figura 1.19 b) el proceso de filtraje incluye también a los coeficientes $d_{j,k}$ de detalle, esto permite un análisis multi resolución en los dominios tiempo frecuencia.

Figura 1. 19 Bancos de Filtros, a) Descomposición DWT b) Descomposición WPT.



Fuente: Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano

La principal desventaja de este tipo de análisis, es su alto costo computacional y mayor dimensionalidad del espacio transformado.

1.5 Clasificación de patrones de movimiento

La base para el desarrollo de un sistema de control basado en señales sEMG, es la identificación de qué tipo de movimiento el usuario está tratando de realizar en base a la señal adquirida. Estas técnicas de discriminación y de toma de decisiones reciben el termino de clasificación de patrones de movimiento, a continuación se describirá brevemente, como referencia, algunas de las técnicas más utilizadas (Romo, et al., 2007), sin embargo esto no implica que sean las únicas existentes o que necesariamente se tenga que recurrir a ellas.

Clasificador Bayesiano, básicamente es un método estadístico que estima la probabilidad “a posteriori” de la pertenencia de una muestra de prueba a una de clases dadas. Esta probabilidad es evaluada en función de las probabilidades “a priori” de cada clase y la probabilidad condicional, resultante de la distribución de las muestras de entrenamiento en cada clase y evaluadas según el teorema de Bayes (Romo, et al., 2007).

Redes neuronales artificiales (ANN, Artificial Neural Network), basado en la estructura de una red neuronal biológica, una ANN es un método computacional que emula las características de aprendizaje y de interconexión de un ser vivo, de esta forma logra “aprender” de experiencias anteriores, se adapta a cambios ambientales e incluso puede procesar información degradada o inclusive incompleta. Sin embargo, la gran limitación de una ANN, radica en que, requiere una gran cantidad de muestras para su procesamiento y los tiempos de entrenamiento para el “aprendizaje” adecuado son bastante largos.

Lógica difusa (Fuzzy Logic), este método, toma dos valores aleatorios pero contextualizados y referidos entre sí (Wikipedia, [s.a]) (por ejemplo, determinar que hace calor porque la temperatura es de 30°C, basado en que a 10°C es considerado frío). De esta forma logra simular la toma de decisiones de un ser humano, basado en estructuras de tipo “IF-THEN”.

Sistemas Neuro-Fuzzy, es la unión de sistemas ANN y de lógica difusa (Fuzzy Logic), esto permite tomar ventaja de la capacidad de aprendizaje de las redes neuronales y la eficiencia en la toma de decisiones de un sistema Fuzzy Logic.

CAPITULO 2:

DISEÑO DE LA TARJETA DE ADQUISICIÓN DE SEÑALES MIOELECTRICAS

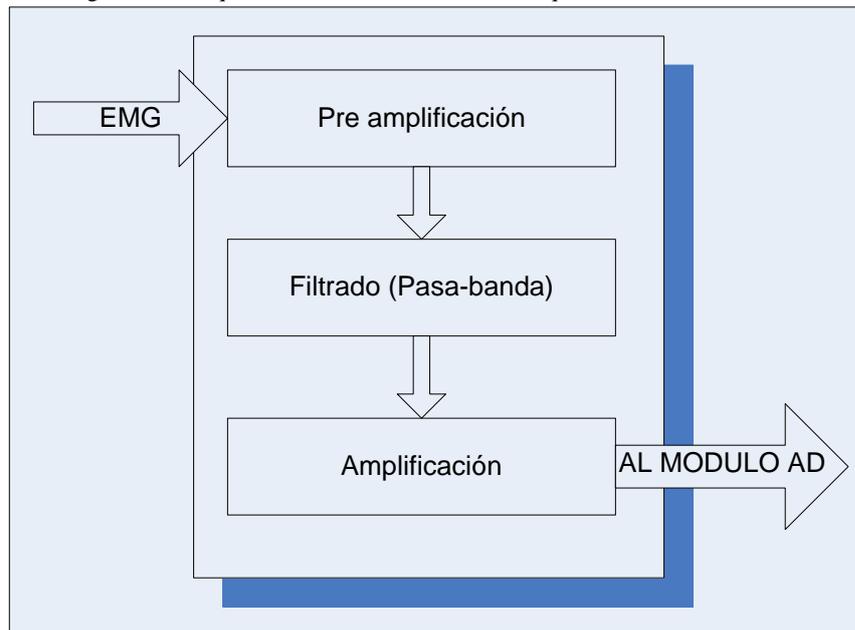
Como se pudo apreciar en el capítulo 1, las señales electromiográficas poseen niveles de tensión demasiado pequeños como para ser empleados directamente en cualquier aplicación electrónica, por tal motivo, es indispensable transformar (sin perder información) estos niveles de tensión, de tal manera que puedan ser utilizados para su adquisición y procesamiento.

En el presente capítulo, se estudiará las técnicas empleadas para el diseño de una tarjeta de adquisición de sEMG, sus bloques constitutivos y sus principales componentes.

2.1 Descripción de los bloques funcionales.

Básicamente todo sistema de adquisición de señales sEMG, posee los bloques funcionales de la figura inferior. Al ingresar una señal EMG al sistema de adquisición, pasa por el bloque de pre amplificación para que de esta forma la señal alcance niveles de voltaje adecuados para los siguientes procesos de adecuación de la señal, luego pasa por un filtro pasa banda, esto permite tener una señal acotada en los niveles de frecuencia deseados (ver capítulo 1) con el fin de eliminar parte del ruido de la señal, y como última parte se tiene un amplificador, el cual permite transformar la señal a niveles de voltaje trabajables (entre 0 y 5V) para su uso en sistemas de conversión Analógico - Digitales para su posterior procesamiento.

Figura 2. 1 Bloques funcionales del sistema de adquisición de señales sEMG.



2.1.1 Pre amplificación

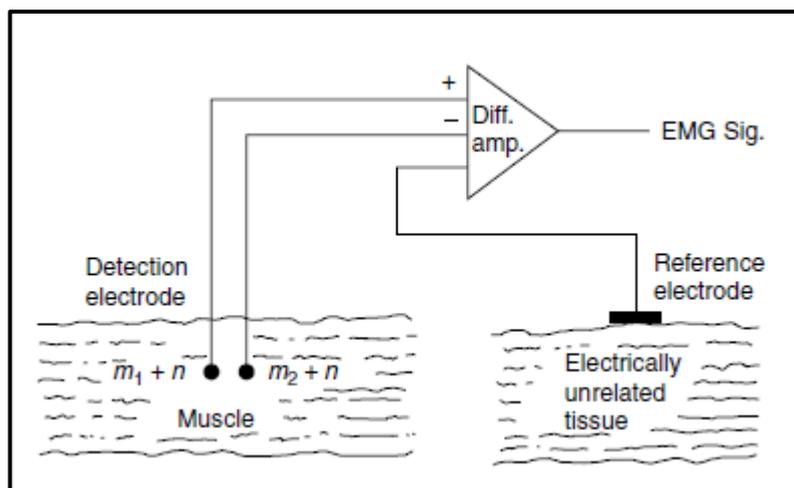
Las señales electromiográficas poseen niveles de tensión en el orden de los micro voltios hasta 1.5 voltios (De Luca, 2002) lo que impide que puedan ser empleadas directamente en cualquier sistema electrónico para su procesamiento, por tal razón, es indispensable contar con una etapa de pre amplificación que transforme estos voltajes sumamente bajos, en niveles que eviten cualquier interferencia electrónica, y a su vez, puedan ser empleados por cualquier sistema electrónico.

El bloque de pre amplificación está conformado, básicamente, en un amplificador diferencial que posee las siguientes características (De Luca, 2006) :

- Impedancia de modo común mayor que $10^{15} \Omega$ y en paralelo con menos de 7 pF.
- CMRR > 85 dB
- Corriente de polarización de entrada menor que 5 fA
- Ruido menor que 1.5 μ V (con entradas cortocircuitadas).
- Ancho de banda 15-500 Hz.

Como se explicó en los párrafos anteriores, la etapa de pre amplificación, básicamente es un amplificador diferencial, por tal motivo, y tomando en cuenta la configuración de electrodos a emplear en el desarrollo de este proyecto (configuración bipolar), es necesario implementar un circuito que permita obtener un voltaje de referencia con respecto al cuerpo, para lograr este requerimiento, se recurre al uso del denominado circuito de pierna derecha o RLD (por sus siglas en inglés Right Leg Drive), este circuito permite que, en el caso de existir cambios en la temperatura del cuerpo o se presenten ruidos inducidos por el mismo, se mantengan los niveles de voltaje correctos. Para que esto se cumpla, es necesario que la ubicación del electrodo de referencia sea en un tejido aislado del músculo a estudiar.

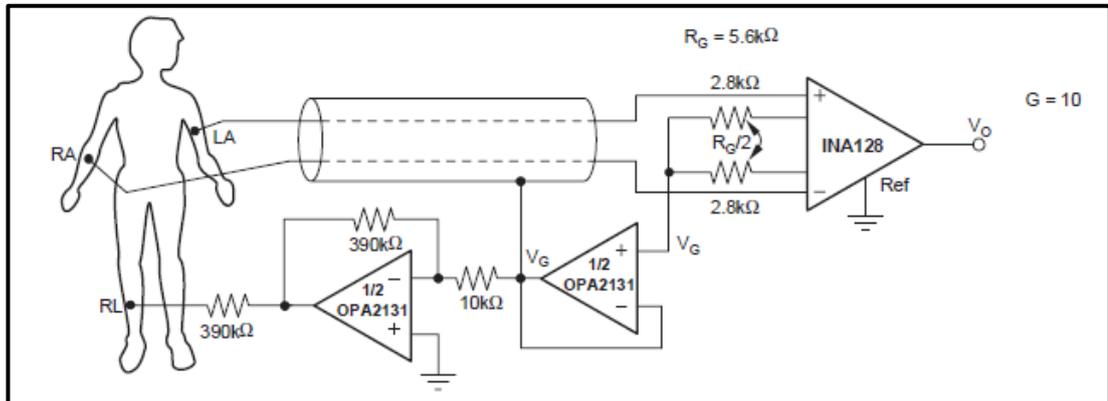
Figura 2. 2 Disposición de los electrodos superficiales y el electrodo de referencia.



Fuente: Encyclopedia of Medical Devices and Instrumentation

Como consideración adicional, en la hoja de datos del amplificador operacional INA128 (que es el que cumple con las características que necesita el pre amplificador), se puede apreciar el circuito RLD optimizado para su uso en adquisición de señales ECG.

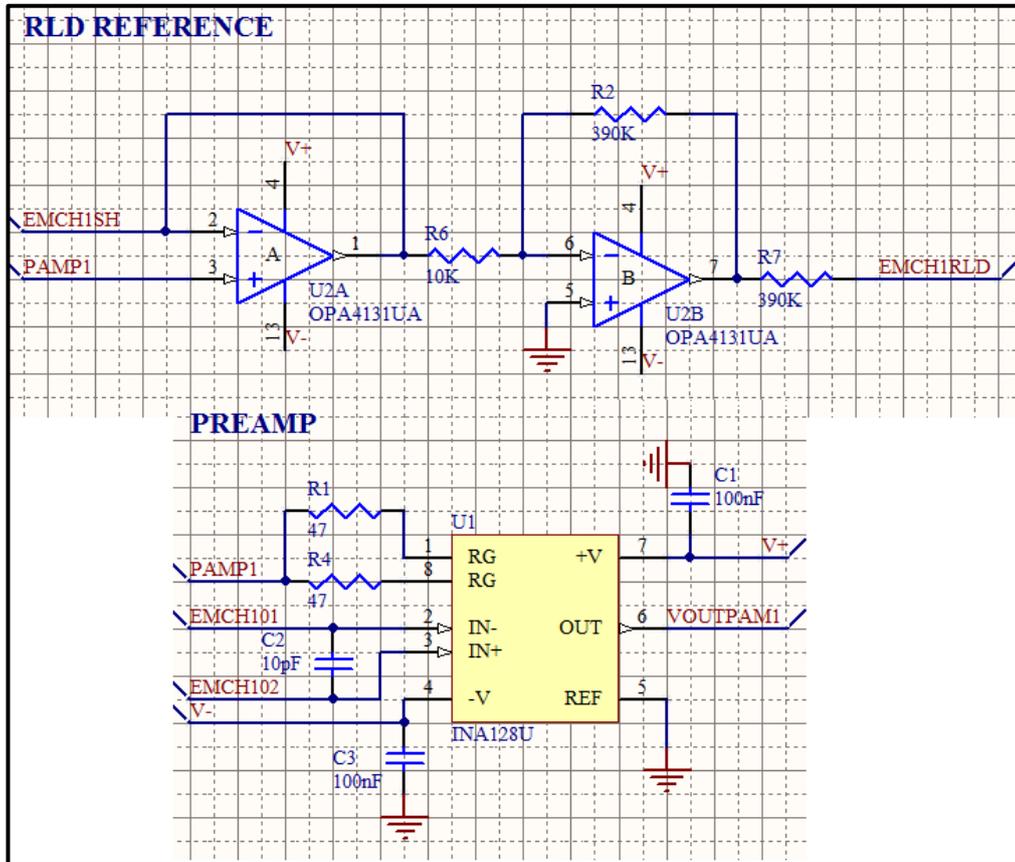
Figura 2. 3 Circuito RLD propuesto en la hoja de datos del amplificador instrumental INA128.



Fuente: Burr Brown, Precision, Low Power INA 128-129 Datasheet

En la figura superior, se puede apreciar que además del circuito RLD y de las conexiones de los electrodos en cada brazo, también existe una conexión con la malla que blindo a los electrodos, esto se realiza para eliminar el ruido que se pudiera inducir en el cable. De todo lo expuesto en este apartado, se obtiene el siguiente circuito de la figura 2.4.

Figura 2. 4 Circuito de pre amplificación.



En la figura 2.4, se puede observar que las resistencias R4 y R1 son las encargadas de regular la ganancia obtenida a la salida del preamplificador (Burr Brown, 2005), por lo tanto para obtener una ganancia de 500 se tiene que:

$$G = 1 + \frac{50k}{R_G} \tag{Ecuación 2.1}$$

Despejando

$$R_G = \frac{50k}{499} \tag{Ecuación 2.2}$$

Donde

$$R_G = 100.2 \Omega \tag{Ecuación 2.3}$$

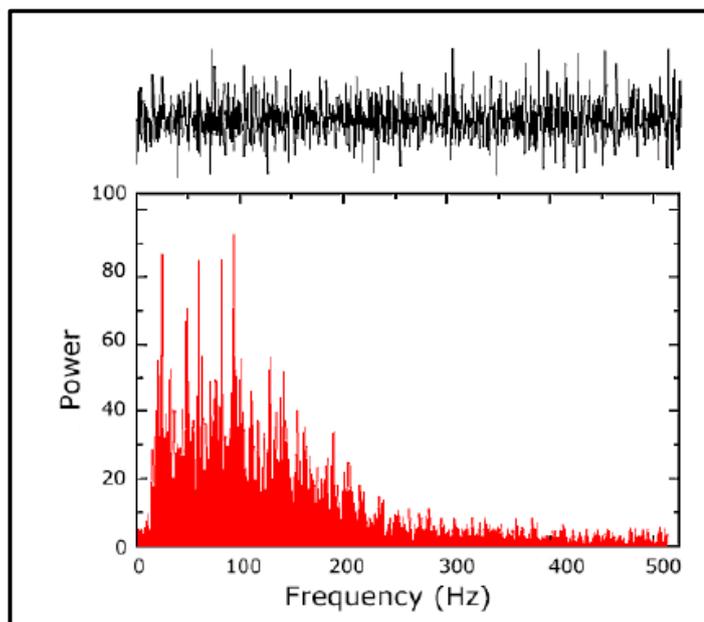
Sin embargo, la hoja de datos del INA128 (Burr Brown, 2005) especifica que, las resistencias que se deben colocar en esos pines deben tener un valor igual a la mitad

de R_G , es decir un valor de 50.1Ω , lamentablemente en el mercado no existe una resistencia de ese valor, por lo tanto, se recurre al uso de resistencias de 47Ω . De esta forma, empleando la ecuación 2.1, la ganancia real que se tendrá en el circuito será de 532.91Ω .

2.1.2 Filtrado

Luego de amplificar las señales a niveles de tensión “trabajables”, es necesario filtrar estas señales con el fin de eliminar toda la información que no sea relevante para el proceso de adquisición. Las señales EMG, se encuentran en el rango de los 15 a los 500 Hz, sin embargo, en el rango de los 15 a los 150 Hz, es donde, está concentrada la mayor cantidad de información (figura 2.5) (De Luca, 2006) (De Luca, 2002).

Figura 2. 5 Espectro de una señal EMG.

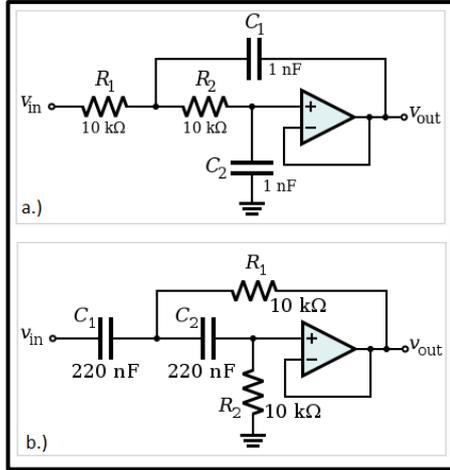


Fuente: delsys.com

Al observar la figura superior, la mayor concentración de energía (información) se encuentra contenida en un rango bien definido de frecuencias (entre los 15 a los 150Hz), por esta razón lo más lógico resulta en la implementación de un filtro pasa banda, lo cual se logra conectando en cascada un filtro pasa bajos con un filtro pasa altos.

Un filtro pasa banda fácil de implementar y de excelentes prestaciones es el filtro activo Sallen Key (Wikipedia, [s.a]) cuya topología se indica en la figura 2.6.

Figura 2. 6 Topología filtro Sallen Key a.) pasa-bajo b.) pasa-alto.



Fuente: en.wikipedia.org

En la figura 2.6 se aprecia que, para cambiar de un filtro pasa bajo a uno pasa alto, simplemente se cambian las posiciones de las resistencias por la de los condensadores. La frecuencia de corte en este tipo de filtro viene dada por:

$$F_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (\text{Ecuación 2.4})$$

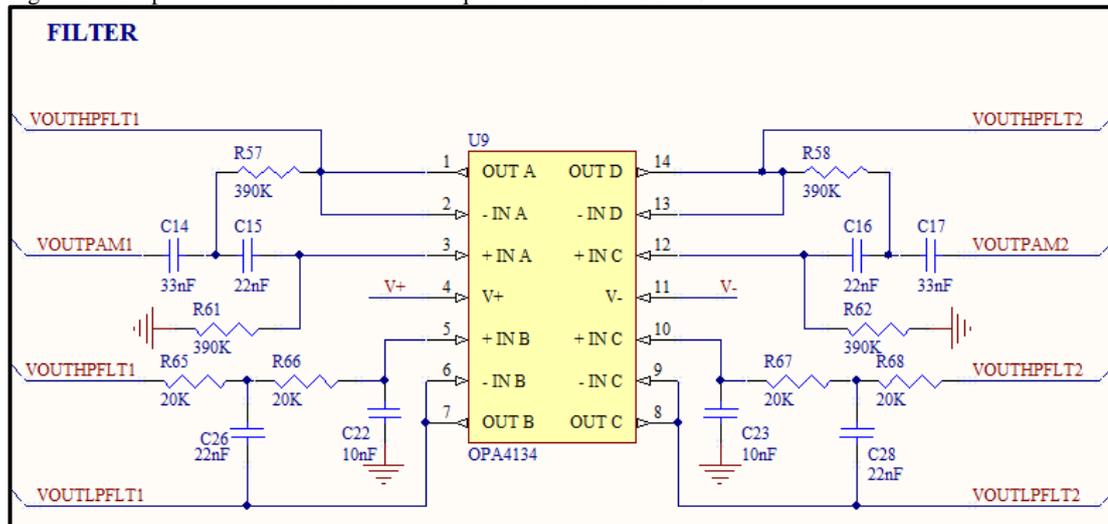
Por lo tanto para obtener un filtro pasa banda con un rango de 15 a 500 Hz se tiene:

$$F_c (\text{Pasa altos}) = \frac{1}{2\pi\sqrt{390K*390K*22nF*33nF}} = 15.14Hz \quad (\text{Ecuación 2.5})$$

$$F_c (\text{Pasa bajo}) = \frac{1}{2\pi\sqrt{20K*20K*22nF*10nF}} = 536.51Hz \quad (\text{Ecuación 2.6})$$

Por consiguiente, tomando las consideraciones de topología y de los cálculos realizados, se obtiene el circuito de la figura 2.7.

Figura 2. 7 Etapa de Filtrado del sistema de adquisición de señales EMG.



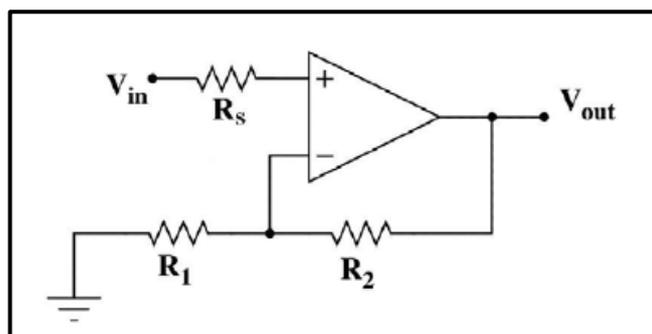
2.1.3 Amplificación

Luego de la etapa de filtrado, es necesario amplificar estas señales de tal forma que puedan ser utilizados por un microcontrolador u otro elemento electrónico para su procesamiento, para esto es indispensable transformar estas señales a niveles TTL para que puedan ser utilizados en la conversión analógico digital.

El bloque funcional de amplificación consiste en un amplificador no inversor (Wikipedia, [s.a]), y de un amplificador diferencial de ganancia unitaria (Wikipedia).

La primera parte de este bloque, como ya se indicó, consta de un amplificador no inversor (figura 2.8), esto básicamente se realiza para calibrar la señal que ingresa al módulo de conversión analógico digital.

Figura 2. 8 Amplificador no inversor de ganancia unitaria.



Fuente: es.wikipedia.org

En donde su ganancia viene expresada por (Wikipedia, [s.a]):

$$V_{OUT} = V_{IN} \left(1 + \frac{R_2}{R_1} \right) \quad (\text{Ecuación 2.7})$$

Dado que lo que se desea es poder regular la ganancia, se reemplaza R2 con un potenciómetro, obteniendo de esta manera los siguientes resultados:

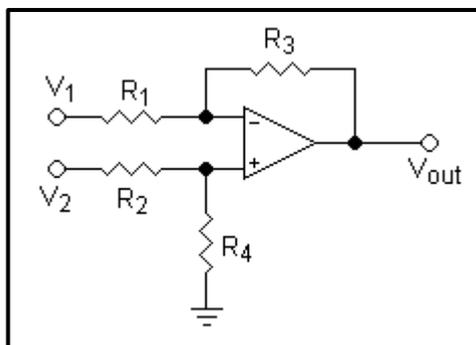
$$R_2 = 0 \rightarrow V_{OUT} = V_{IN} \left(1 + \frac{0}{10K} \right) = V_{IN} \quad (\text{Ecuación 2.8})$$

$$R_2 = 200K \rightarrow V_{OUT} = V_{IN} \left(1 + \frac{200K}{10K} \right) = 21V_{IN} \quad (\text{Ecuación 2.9})$$

Mientras que el valor de Rs es igual al paralelo de R2 y R1, al ser R2 mucho mayor que R1, Rs tiende a obtener un valor cercano a R2. De esta forma se puede regular que la señal de salida del amplificador no sobrepase los valores máximos de -2.5v y 2.5v pico.

Sin embargo, es necesario sumar una componente de 2.5v con el fin de que el rango de salida del amplificador coincida con el rango de conversión del módulo AD, es decir entre 0 y 5v, para ello se recurre al uso de un circuito amplificador diferencial con ganancia unitaria (figura 2.10) y al circuito integrado TPS71525 el cual genera un voltaje de 2.5V constantes.

Figura 2. 9 Amplificador diferencial con ganancia unitaria.



Fuente: es.wikipedia.org

En donde su voltaje de salida es:

$$V_{OUT} = V_2 \left(\frac{(R_3+R_1)R_4}{(R_4+R_2)R_1} \right) - V_1 \left(\frac{R_3}{R_1} \right) \quad \text{(Ecuación 2.10)}$$

Por lo tanto, si $R_1 = R_2 = R_3 = R_4$ se tiene:

$$V_{OUT} = V_2 - V_1 \quad \text{(Ecuación 2.11)}$$

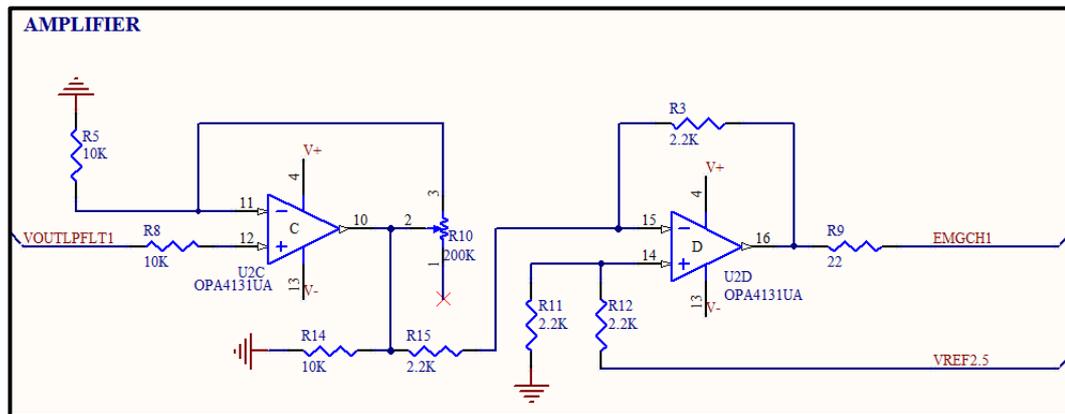
Al tener $V_1 = EMG$ y V_2 igual a un nivel de referencia de 2.5v se obtiene:

$$V_1 = -2.5V \rightarrow V_{OUT} = 2.5V - (-2.5V) = 5V \quad \text{(Ecuación 2.12)}$$

$$V_1 = 2.5V \rightarrow V_{OUT} = 2.5V - (2.5V) = 0V \quad \text{(Ecuación 2.13)}$$

De esta forma, se logra una señal centrada en 2.5v e invertida a la salida del amplificador, esto no distorsiona la información contenida en la señal EMG, pues simplemente se puede mediante software cambiar la polaridad de la señal obtenida en el conversor AD.

Figura 2. 10 Esquema de la etapa de amplificación.

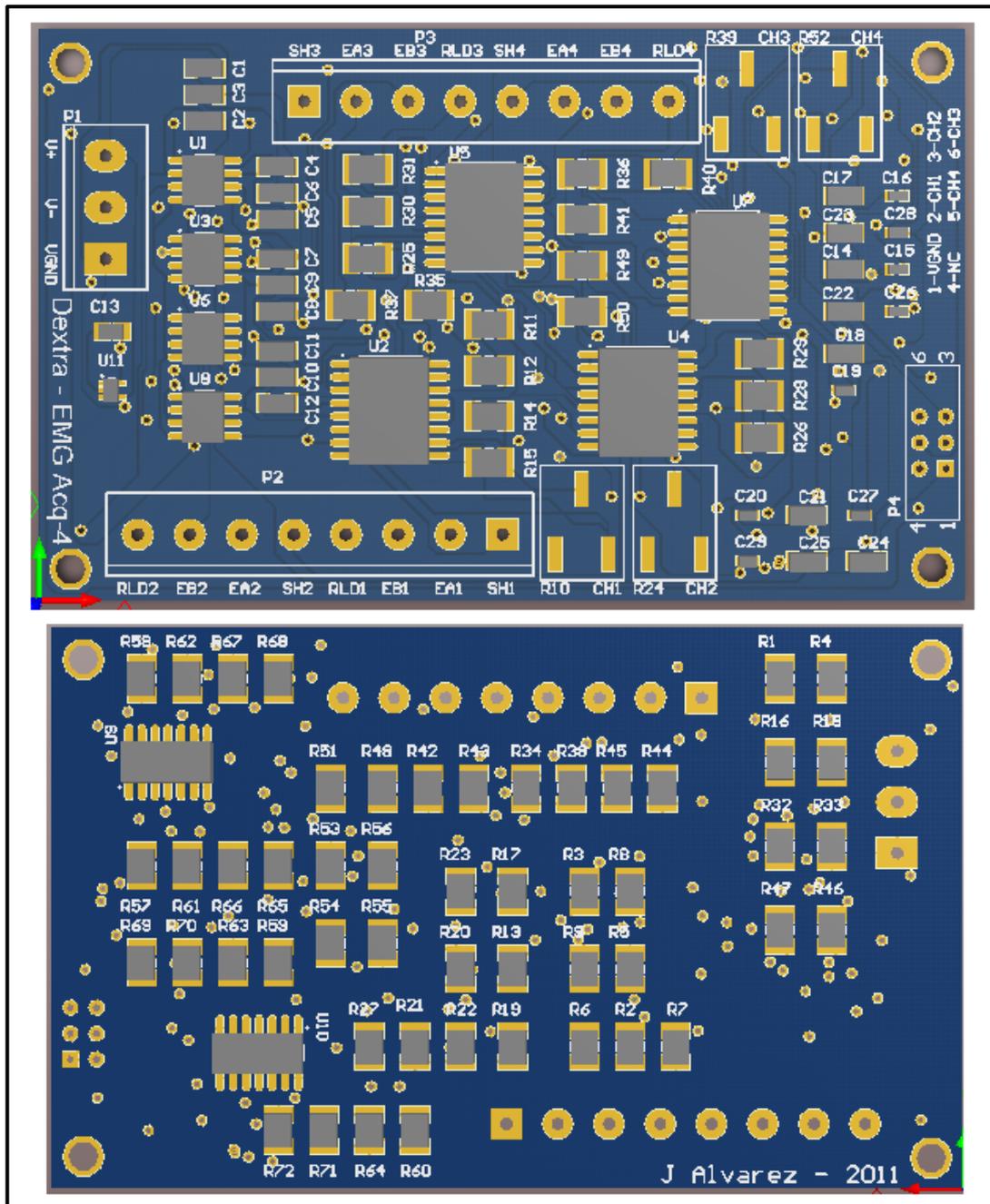


2.2 Producto Final

Para la elaboración del PCB final del módulo de adquisición, se utilizó el software de diseño electrónico Altium Designer 10, el cual permite obtener placas PCB a partir de esquemas de circuitos electrónicos. Para el posicionamiento, ruteo, elaboración de pistas y pads y demás procesos que implican la elaboración de un

PCB, se tomó en cuenta las recomendaciones de diseño (ancho de pista, distancia entre pistas, planos de tierra entre otras) para la adquisición de este tipo de señales, sin embargo, el presente apartado no explica las consideraciones de diseño del PCB, pues se considera que el estudio del diseño de circuitos impresos es un tema que abarca campos que van más allá del enfoque del presente proyecto (Jones, 2004) (Khandpur, 2006) (Coombs, 2001) (Wilson, 2012).

Figura 2. 11 Circuito PCB para la adquisición de señales electromiográficas.



En la figura 2.11, se puede apreciar la vista frontal y posterior del PCB para la adquisición de señales EMG, las características que posee el circuito son las siguientes:

- 4 canales independientes para la adquisición de señales EMG, (RLDx para el circuito de pierna derecha, SHx para la conexión de la malla de blindaje y Exy para la conexión de los cables que adquieren las señales EMG).
- Filtro pasa banda de 15 a 500 Hz.
- 4 potenciómetros para la calibración de las señales EMG adquiridas.
- 1 salida P4 con conector tipo shrouded box. En donde el pin 1 es la salida VGND, y los pines 2 al 5 corresponden a las salidas en niveles TTL de las señales EMG adquiridas.

Para mayores detalles, consulte el anexo 1 en donde constan todos los esquemas de los circuitos implementados y sus diseños en PCB respectivamente.

CAPITULO 3:

DISEÑO DE LA TARJETA DE CONTROL DE ACTUADORES

En el presente capítulo, se abordará el diseño de los diferentes módulos de circuitos que permiten el control de los actuadores mediante el uso de señales sEMG.

Si bien es cierto que, la mayoría de los componentes y circuitos empleados para el control de actuadores y de interface de sensores son ampliamente conocidos, es necesario profundizar, en el caso de ser necesario, en su interacción con la concepción global del sistema de control.

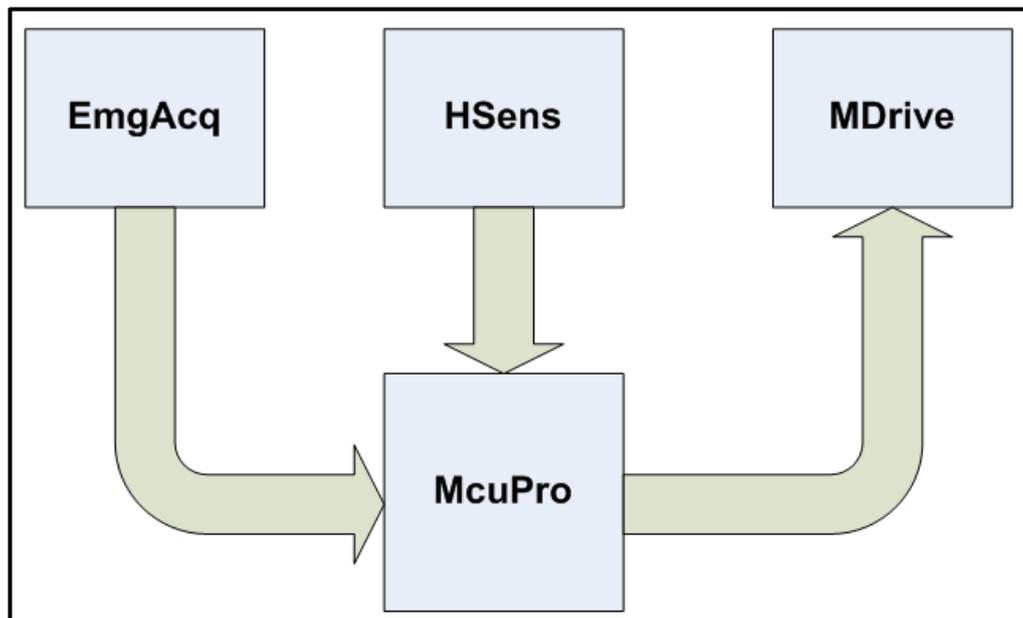
3.1 Descripción de los bloques funcionales

Una de las principales consideraciones al momento de realizar el presente proyecto, fue brindar cierta flexibilidad para futuros desarrollos, por tal motivo, se decidió dividir en varios circuitos o módulos a todo el sistema de control y de adquisición, de esta forma, si se desea, se puede emplear únicamente los módulos que se crean necesarios. Ya sea que se desee solamente adquirir señales EMG, controlar motores, utilizar otro tipo de sensores o simplemente que el proceso de análisis se realice de otra forma o con otros componentes, el sistema está constituido por los siguientes módulos:

- Una tarjeta para el control de motores de DC, denominada MDrive.
- Un hub de sensores de la mano robótica, denominada HSens
- Una tarjeta de procesamiento denominada McuPro
- Una tarjeta de adquisición de señales EMG denominada EmgAcq 4 (véase el capítulo 2).

Esta flexibilidad, permite que los tiempos de desarrollo de futuras aplicaciones, se reduzcan, pues no es necesario volver a desarrollar todo el sistema de adquisición y control, sino simplemente se reemplaza el modulo que no cumple con las especificaciones de determinado nuevo proyecto.

Figura 3. 1 Diagrama de bloques de la interacción de los circuitos componentes del sistema.



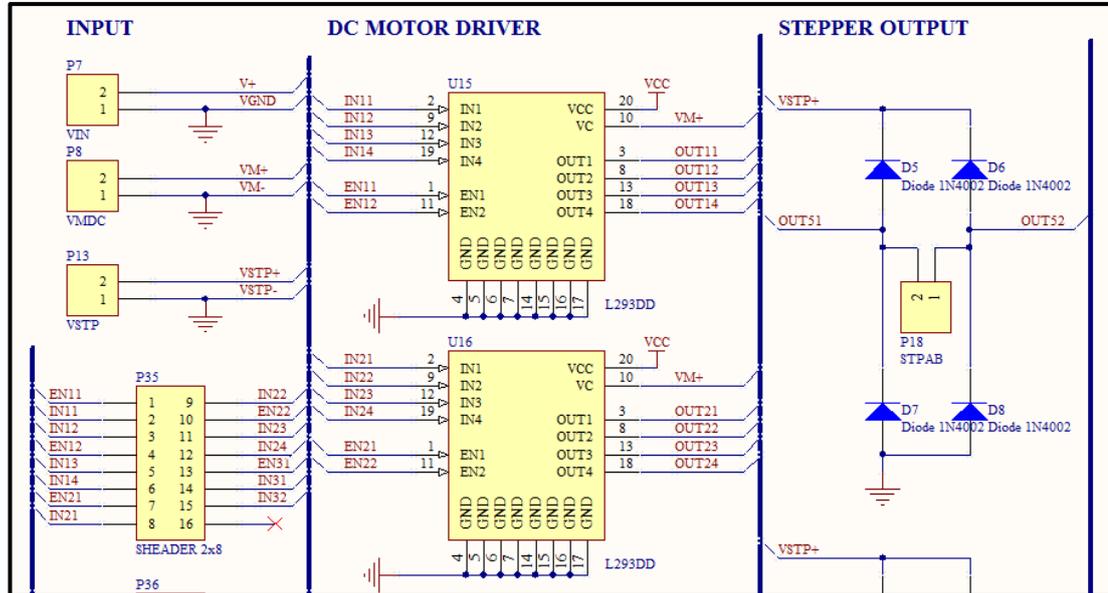
Como se puede apreciar en la figura 3.1, luego de ser adquiridas las señales sEMG en el módulo EmgAcq, pasan al módulo de procesamiento McuPro, éste es el encargado de tomar las decisiones en base a las señales que recibe. En el caso de que las señales recibidas correspondan con cierta actividad muscular, McuPro enviará una señal a MDrive, de tal forma que, activará los motores hasta que cumplan determinado movimiento por completo, o hasta que, alguno de los sensores ubicados en el HSens envíe una señal a McuPro que indique que debe detener a los actuadores. En las siguientes páginas se explica con mayor detalle el funcionamiento de cada uno de estos módulos.

3.2 Módulo de control de motores MDrive

El módulo de control de motores, permite manejar un motor de pasos (para la rotación de la muñeca) y hasta 8 motores de DC. Tanto la alimentación del circuito,

como de los motores DC y el de pasos es independiente, sin embargo, la limitación de corriente que deben manejar estos motores es de 600 mA como máximo.

Figura 3. 2 Parte del esquema del módulo de control de motores Mdrive.



El módulo está conformado, básicamente, por 4 circuitos integrados L293DD (STMicroelectronics, 2003) y un L293B (STMicroelectronics, [s.a]), estos circuitos integrados al recibir un determinado pulso en sus entradas, activan o desactivan a uno o varios motores dependiendo de la conexión externa que sea dispuesta. La diferencia entre estos dos integrados radica en que, en el primer integrado los diodos de protección se encuentran incorporados mientras que en el L293B éstos son externos.

Tabla 3. 1 Tabla de verdad del IC L293B.

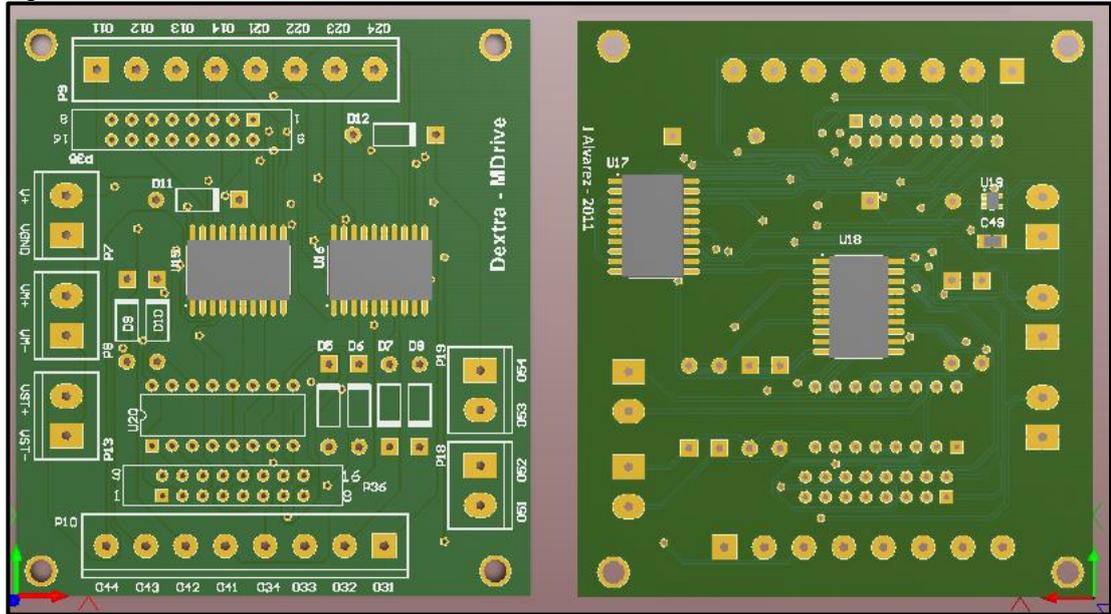
| TRUTH TABLE | | |
|----------------------|-------|------------------|
| V_i (each channel) | V_o | $V_{inh}^{(**)}$ |
| H | H | H |
| L | L | H |
| H | X (*) | L |
| L | X (*) | L |

(*) High output impedance
 (**) Relative to the considerate channel

Fuente: Push-pull four channel driver L293B-L293E Datasheet.

Las señales de control provenientes del McuPro ingresan a través de los conectores P35 y P36. En el anexo 1 se puede observar el esquema completo del circuito de control de motores y su correspondiente diseño de PCB.

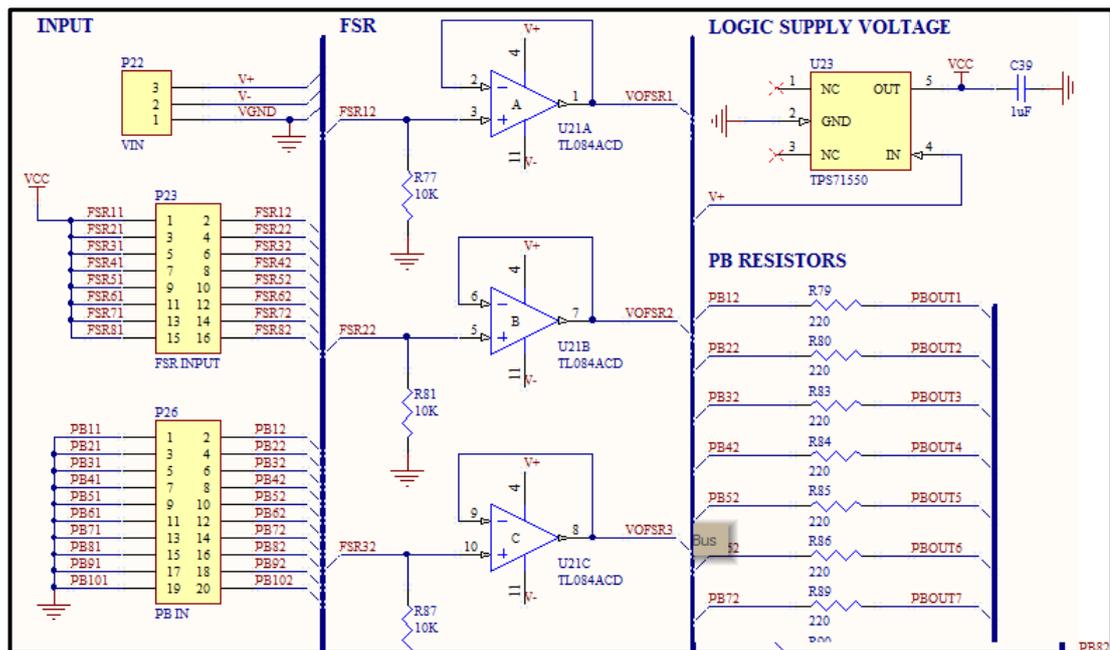
Figura 3. 3 PCB del módulo de control de motores MDrive.



3.3 Módulo de sensores HSens

El módulo HSens, es simplemente un hub de sensores, donde convergen todos los implementos electrónicos que permiten tener una retroalimentación de variables como la fuerza y la posición inicial de la prótesis robótica. En este módulo van conectados 10 pulsantes, los cuales, permiten conocer si la prótesis se encuentra en su posición inicial; además también están conectados sensores de presión denominados FSR, los cuales permiten mediante software calibrar la fuerza del agarre de la mano.

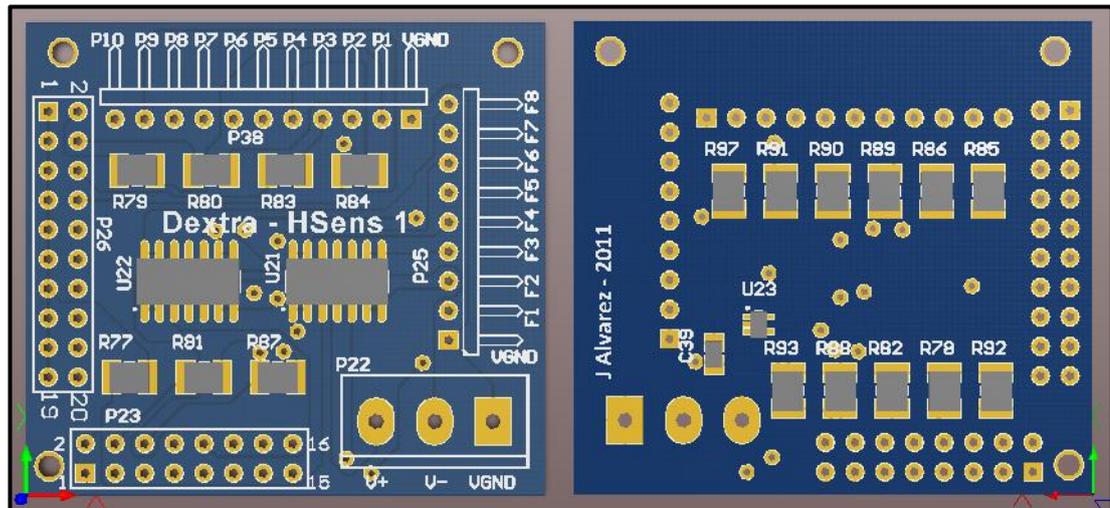
Figura 3. 4 Parte del esquema del módulo HSens.



Una consideración importante a tener en cuenta, es que, la conexión para los pulsantes, está configurada de tal forma, que puedan ser conectados a través de las resistencias de pull-ups internas del microcontrolador (Microchip Technology Incorporated, 2006).

Por otra parte, la configuración del amplificador TL084 es la recomendada por el fabricante para obtener una salida de hasta 3.5v aplicando una fuerza de 1Kg (Electronics, Interlink, [s.a]). En el capítulo 4, se profundizará con mayor detalle sobre este tipo de sensores.

Figura 3. 5 PCB del módulo HSens.

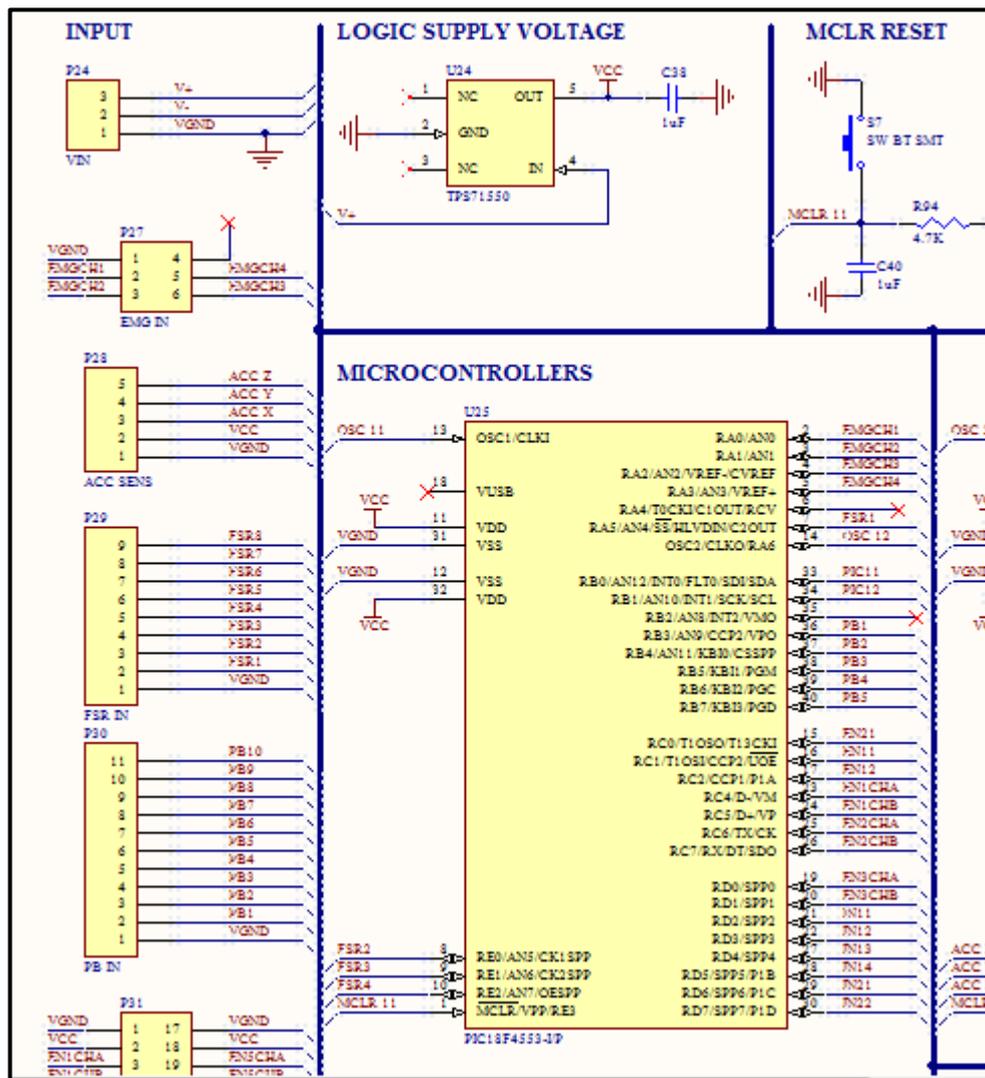


3.4 Módulo de procesamiento y control McuPro

El módulo McuPro, es donde convergen todas las señales necesarias para la toma de decisiones del sistema, es decir, es el encargado de realizar los diferentes tipos de movimientos, dependiendo de los valores de EMG y de los valores provenientes de los diferentes sensores ubicados en la mano robótica, que recibe.

Está constituido por tres micro controladores PIC 18F4553, los cuales poseen un módulo de conversión analógico digital de 12 bits (Microchip Technology Incorporated, 2007), lo cual permite tener una mejor resolución, esto es muy importante cuando se trata de analizar señales electromiográficas, debido a la pequeña diferencia entre el valor de voltaje existente entre el músculo en reposo y el músculo en activación.

Figura 3. 6 Parte del esquema del módulo de procesamiento y control McuPro.



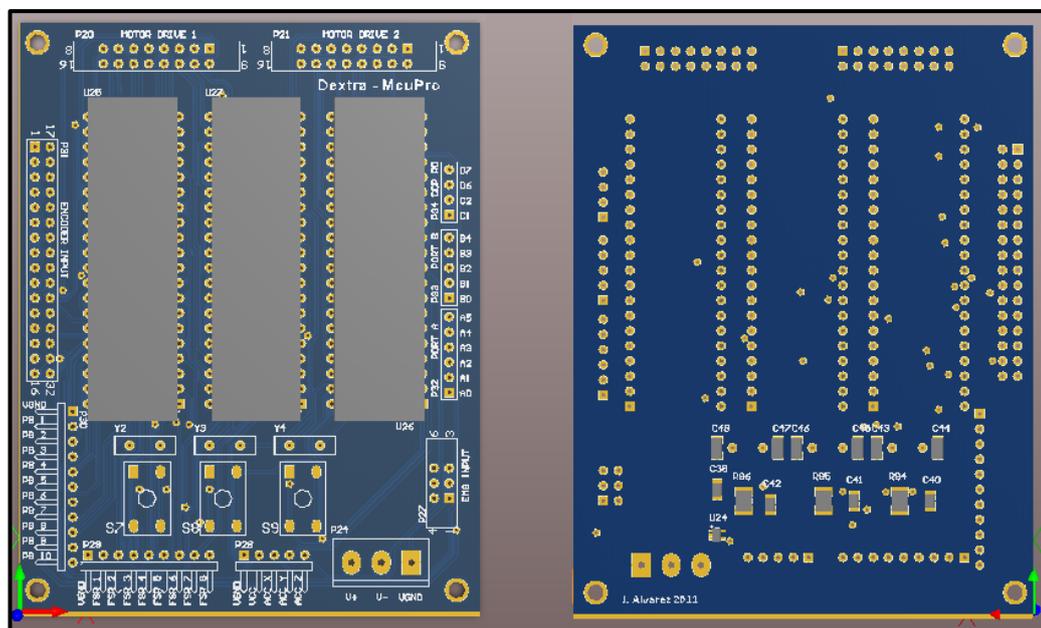
3.4.1 Descripción del circuito

El módulo McuPro, posee las siguientes características (para mayor información sobre el esquema completo consultar el anexo 1):

- Tres micro controladores para el procesamiento de la señal, con sus respectivos botones de reset
- Un entrada de alimentación.
- Un regulador de tensión TPS1550
- Una entrada tipo shrouded box para las señales EMG
- Una entrada para las señales de los ejes x, y, z del acelerómetro

- Una entrada para las señales de hasta 8 FSR.
- Una entrada para los pulsos de hasta 10 pulsantes (conectados con los pull ups internos del microcontrolador)
- Dos salidas tipo shrouded box para los driver de los motores
- Una entrada-salida para los pulsos generados por los encoders incorporados en los motores y para su respectiva alimentación
- Salidas directas de los pines RA0-RA5, RB0-RB4, RC1, RC2, RD6 y RD7 del microcontrolador U26 para su uso en cualquier aplicación adicional.

Figura 3. 7 PCB del módulo McuPro.



3.4.2 Firmware

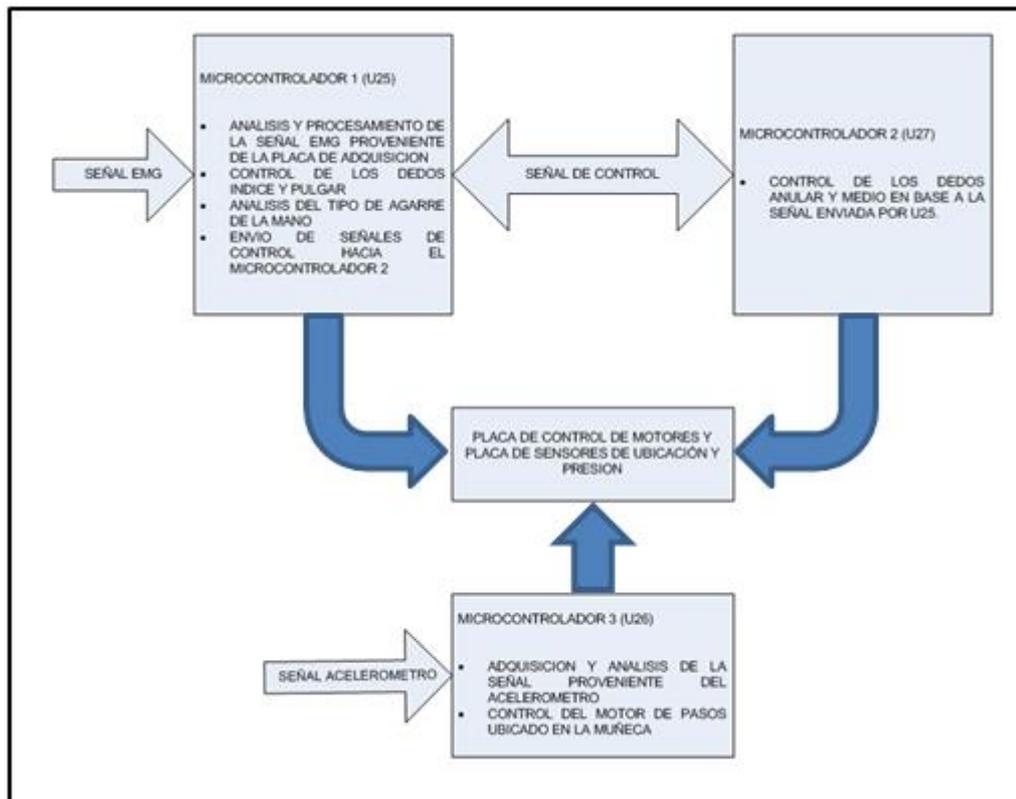
Antes de explicar el diagrama de flujo del firmware de control y procesamiento, es necesario explicar las funciones que desempeñan cada uno de los microcontroladores (también denominados PIC o MCU) empleados en el módulo McuPro y los sensores implicados en cada movimiento.

Como se puede observar en la figura 3.7 (diagrama completo en el anexo 1), existen tres microcontroladores encargados de los procesos de control, el primer PIC (en el esquema denominado como U25) se encarga principalmente de tomar las señales EMG y analizarlas para generar el movimiento de los diferentes motores, además

también controla el movimiento de los dedos pulgar e índice de la mano y envía señales hacia U27 para la activación de los dedos faltantes.

El segundo microcontrolador U27, al recibir la señal enviada por U25, genera el movimiento de los dedos medio y anular. Todo esto tomando en cuenta las diferentes señales que adquiere de los diferentes sensores asociados a cada dedo. Por último el microcontrolador U26 se encarga básicamente del movimiento de la muñeca del prototipo, esto lo realiza analizando la señal que proviene del acelerómetro (se abordara con mayor detalle su funcionamiento en el capítulo 4). Todo lo anteriormente explicado se resume en el diagrama de la figura 3.8.

Figura 3. 8 Diagrama de relación de procesos entre los Microcontroladores del módulo McuPro.

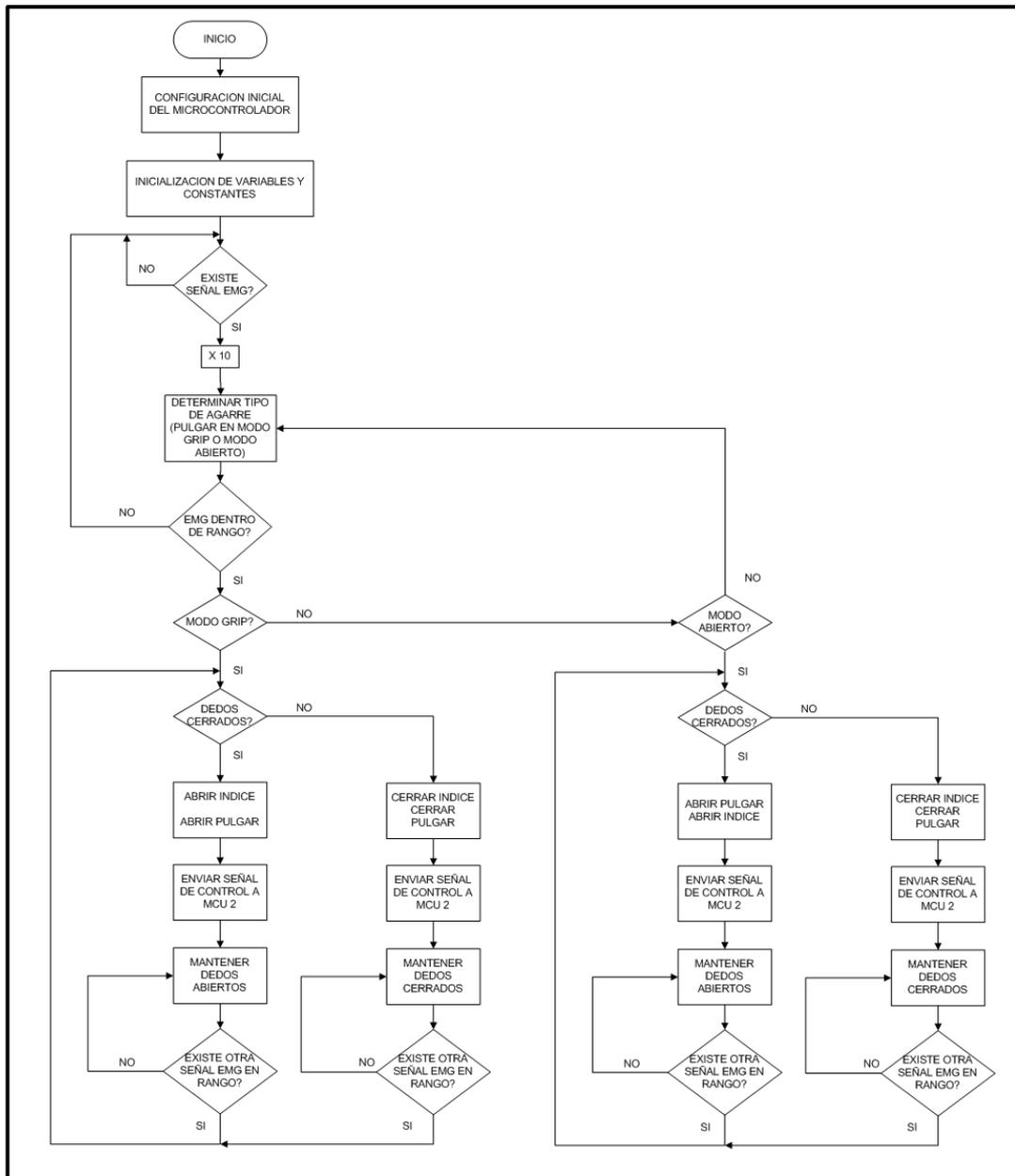


El código completo de cada uno de los microcontroladores se los puede consultar en el anexo 2.

3.4.2.1 Algoritmo del microcontrolador U25

Este MCU es el corazón de todo el proceso de control, pues es el encargado de analizar la señal EMG proveniente del módulo EmgAcq 4 y en base a estos valores de tensión generar las señales necesarias para la activación de los motores asociados a determinado movimiento.

Figura 3. 9 Algoritmo del MCU U25.



En la figura anterior, se puede apreciar que el algoritmo de control es bastante simple, básicamente luego de configurar los valores constantes en el microcontrolador (los valores de voltaje de la señal emg deseada, los valores de tensión registrados por los sensores de presión, valores de conteo de los encoders, etc.) lo único que realiza el código es adquirir la señal emg que proviene de la placa EmgAcq 4 y multiplicarla por 10 para así obtener al menos un decimal más de resolución para nuestras comparaciones, luego, se determina si éste valor se encuentra dentro del rango deseado (para esto es necesario conocer de antemano las características de la señal de la persona que operará la prótesis, véase el capítulo 5) y mediante comparaciones, determinar si la mano se encuentra ya sea en modo grip o en modo de mano abierta. Dependiendo del tipo de modo de agarre, los dedos se abrirán o cerraran en diferente orden, esto permite un sinnúmero de opciones de movimiento que pueden ser configuradas mediante software.

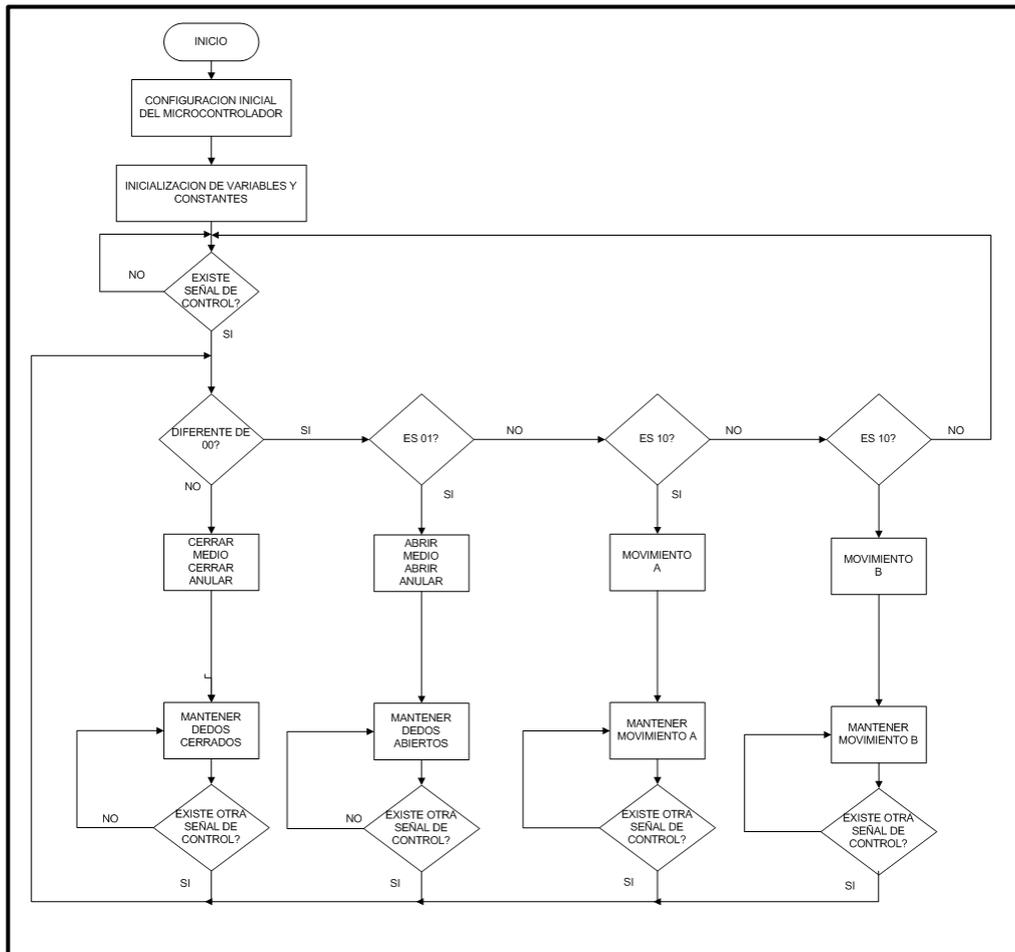
3.4.2.2 Algoritmo del microcontrolador U27

El microcontrolador U27 funciona de modo similar a una configuración maestro-esclavo, en donde el maestro es el PIC U25, esto implica que si U25 no envía la señal de control hacia U27, éste no realiza ninguna acción sobre los dedos medio y anular de la prótesis robótica.

La señal de control es básicamente un código binario de 2 bits, esto implica que podría tener hasta 4 diferentes órdenes para el control de los dedos, esta particularidad permite que el tipo de agarre sea plenamente personalizable por el usuario.

En el diagrama de flujo de la figura 3.10, se observa los movimientos posibles de acuerdo a la señal de ingreso que puede ejecutar en el microcontrolador U27, sin embargo, por cuestiones de simplicidad, exclusivamente se definen los movimientos básicos de abrir y cerrar de cada dedo, por tal motivo para los dos últimos movimientos (que como ya se indicó anteriormente, son completamente configurables) únicamente existe una notación, pues el tipo de agarre quedaría a la discreción y requerimiento de la persona que utilice la prótesis robótica.

Figura 3. 10 Algoritmo del MCU U27.



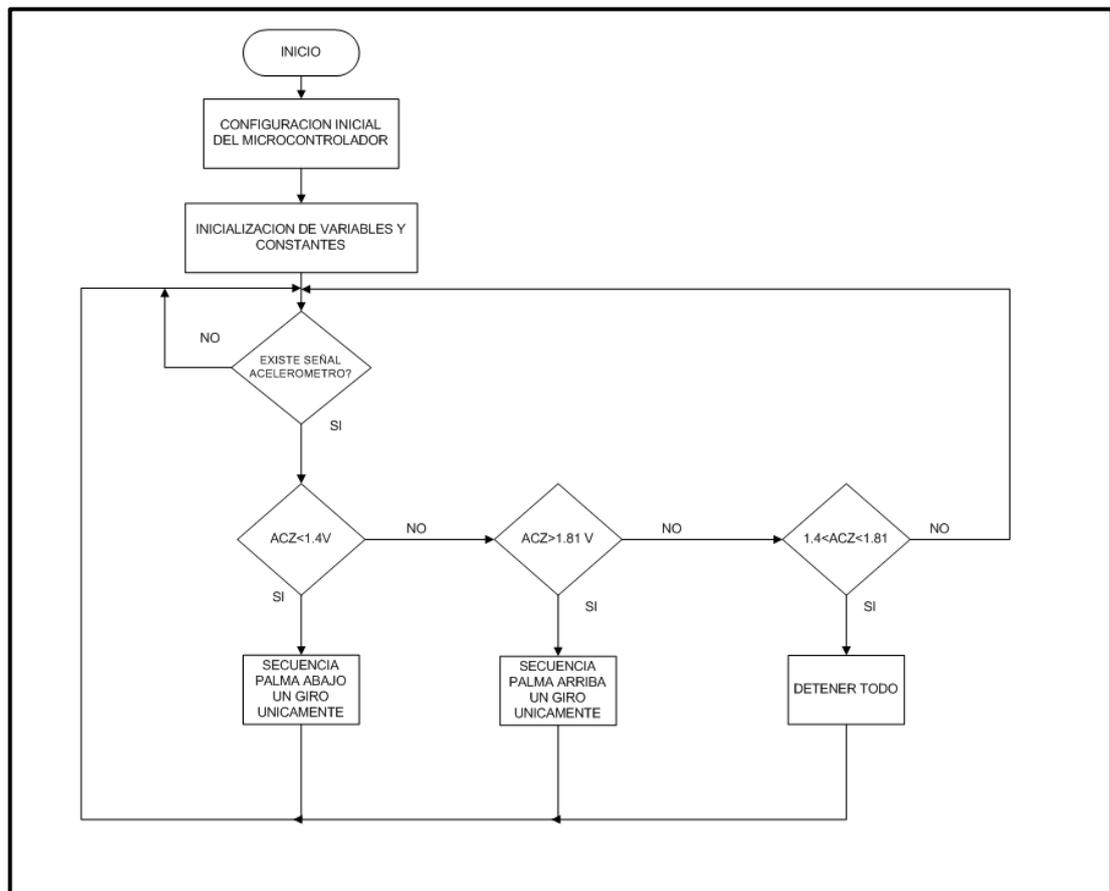
3.4.2.3 Algoritmo del microcontrolador U26

El MCU U26, este se encarga exclusivamente del control de giro de la muñeca, para esto adquiere la señal que genera el acelerómetro y en base al nivel de voltaje que obtiene en sus terminales, puede determinar si la intención del usuario es rotar la muñeca o no.

El diagrama de flujo de este microcontrolador es bastante básico; una vez que se adquiere la señal, se comprueba si ésta se encuentra en rango (ya sea para ubicar la palma hacia arriba, hacia abajo o simplemente detener el giro) y el microcontrolador envía la señal correspondiente para que el motor de pasos encargado de la muñeca describa una única rotación en el sentido deseado, una vez concluida la rotación, se vuelve a consultar el estatus de la señal que proviene del acelerómetro para repetir nuevamente el ciclo.

Los valores utilizados para para determinar estos rangos son proporcionados por el fabricante del acelerómetro (Freescale Semiconductor, 2012), y mediante mediciones realizadas con el software de calibración (capítulo 5).

Figura 3. 11 Algoritmo del microcontrolador U26.



CAPITULO 4:

DISEÑO DE MANO ROBOTICA

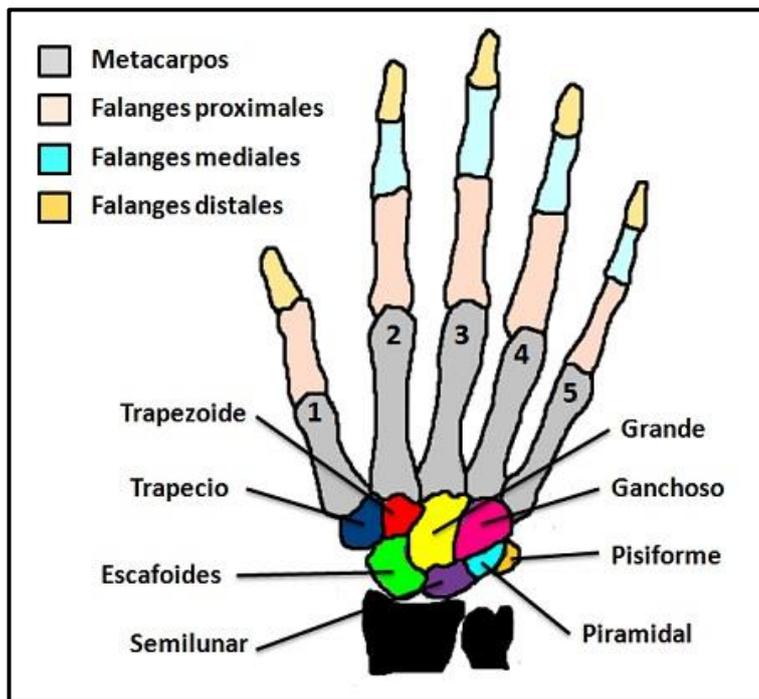
A lo largo del presente capitulo, se explicarán los diferentes parámetros tomados en cuenta para el diseño de la mano robótica empleada para comprobar el funcionamiento de los circuitos electrónicos implementados para el control de actuadores mediante señales EMG.

Como consideración adicional, hay que tener en cuenta que el estudio para el diseño de una mano robótica abarca varias áreas del conocimiento las cuales se extienden mucho más allá del tema de investigación planteado en el presente texto, es por tal razón que en este capítulo se busca explicar, a breves rasgos, los conceptos más relevantes inherentes a la mano empleada, tales como las consideraciones antropomórficas de diseño, descripción de componentes esenciales y los actuadores a ser empleados.

4.1 Consideraciones de diseño

Teniendo en cuenta que lo que se busca es tratar de replicar el comportamiento de una mano humana, es necesario, partir precisamente, de las características morfológicas presentes en el elemento a replicar y tratar de emular su comportamiento con los componentes electrónicos disponibles y tomando siempre en cuenta las limitaciones tanto de tamaño como de grados de libertad a replicar.

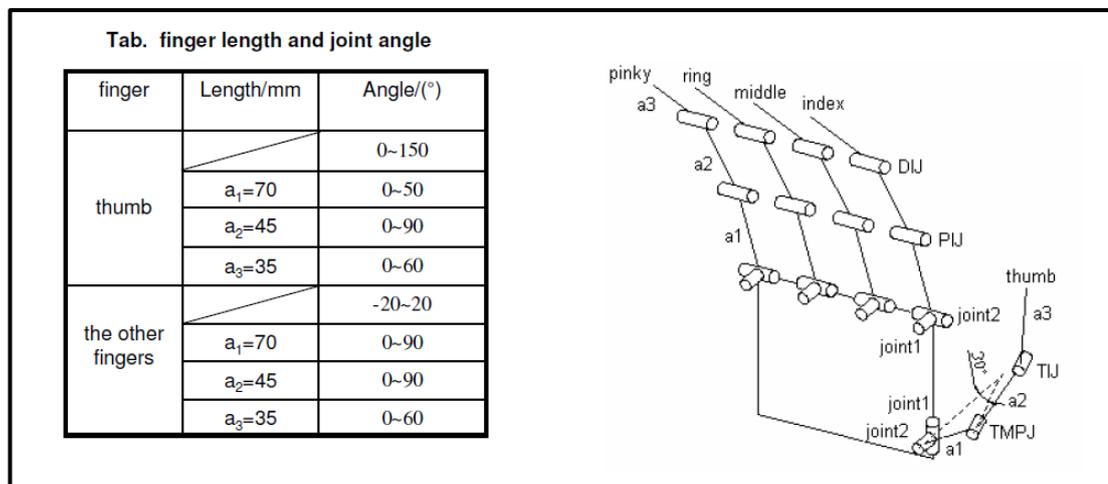
Figura 4. 1 Huesos de la mano.



Fuente: <http://www.saberespractico.com>

Un primer acercamiento para reproducir el comportamiento de una mano es conocer como está configurada su estructura interior, en este caso, que relaciones tienen los huesos de los dedos entre sí. Como se puede apreciar en la figura anterior, existen numerosos huesos que constituyen una mano, sin embargo, únicamente se tomarán en cuenta los huesos que constituyen los diferentes dedos, pues los que se encuentran presentes en la palma no son relevantes para el diseño de una prótesis robótica.

Figura 4. 2 Relaciones longitudinales entre los huesos de los dedos y ángulos de flexión.

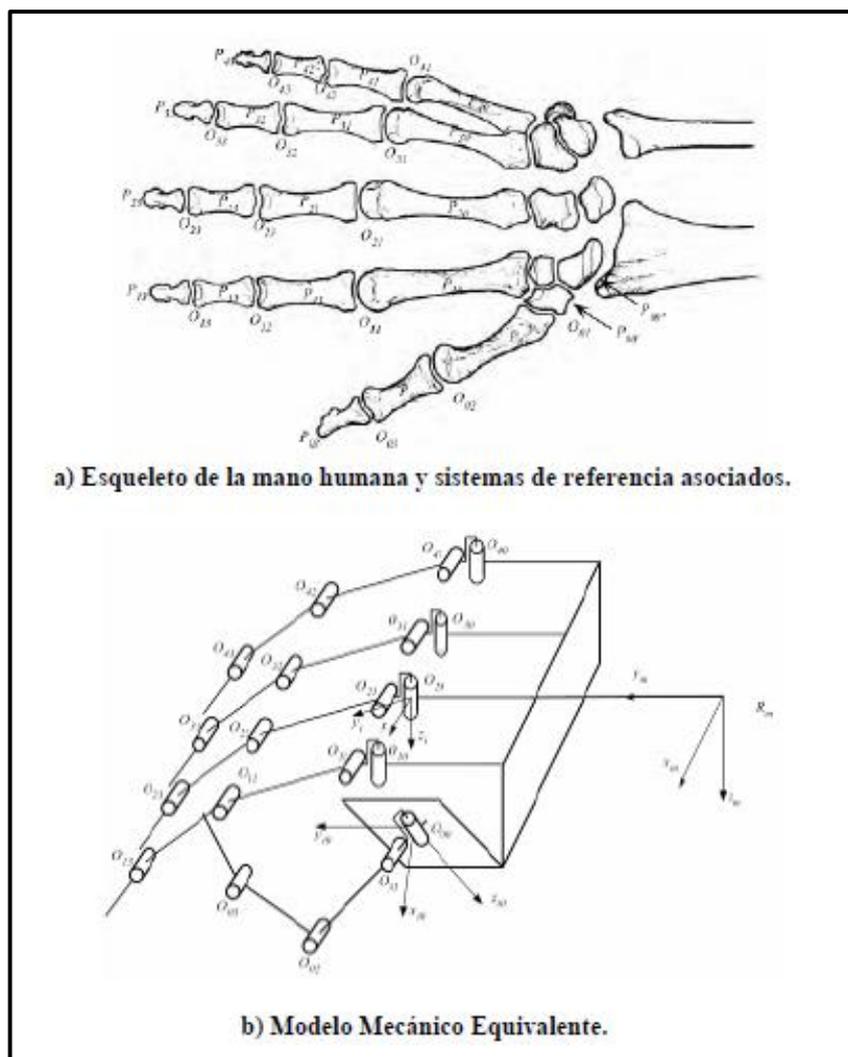


Fuente: Beng Guey, An Intelligent Prosthetic Hand using Hybrid.

En la figura 4.2 se puede apreciar las relaciones longitudinales y los ángulos de flexión característicos existentes entre los huesos de los dedos, esto permite determinar las proporciones anatómicas que deben tener los elementos que conformaran los dedos de la mano robótica para tratar de replicar tanto en apariencia como en funcionalidad a su homóloga humana.

Hasta el momento, en los párrafos anteriores, solo se ha tomado en cuenta las características “físicas” de la mano (dimensiones y proporciones), sin embargo, para replicar efectivamente a este miembro, es necesario tomar en cuenta sus características “activas” o los grados de libertad DOF (por sus siglas en inglés) y por ende los movimientos básicos que puede realizar la misma.

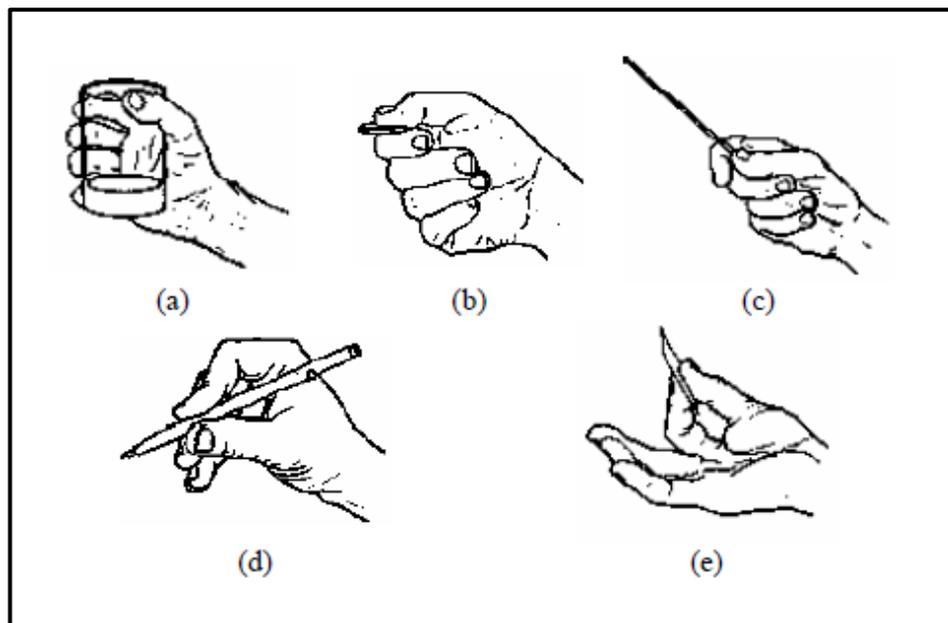
Figura 4. 3 Esqueleto de una mano y modelo mecánico de la misma.



Fuente: Oscar Avilés, Diseño y construcción de un dedo para grippers robóticos

La gran de músculos y juntas presentes en la mano permiten hasta 20 DOF, esto posibilita muchas variaciones de agarre, la cuales en muchos casos varían de persona en persona. Sin embargo, con el fin de unificar estos diversos tipos de agarres, Schlesinger desarrolló una clasificación de la taxonomía para el estudio de la destreza de las manos humanas (Oscar F. Aviles, [s.a]). Este autor agrupó en seis categorías las estrategias de agarre de la mano humana.

Figura 4. 4 Modelos prensiles humanos de Schlesinger, a) cilíndrica palmar, b) subterminal lateral, c) centralizado, d) tridigital, e) posición subterminal.



Fuente: Oscar Avilés, Diseño y construcción de un dedo para grippers robóticos

Considerando el gráfico anterior, para lograr realizar la mayoría de los agarres allí mostrados, es necesario contar con al menos cuatro dedos que se muevan de manera independiente (pulgar, índice, medio y anular), además hay que tomar en cuenta el papel que desempeña el pulgar de acuerdo a su ubicación con respecto a la palma, pues dependiendo de su orientación, los tipos de agarres que se pueden realizar cambian de manera substancial.

4.2 Descripción de los componentes

En el tema anterior se puede tener un bosquejo de lo que implica el diseño de una mano robótica, sin embargo, es necesario considerar los componentes electrónicos disponibles y el aporte que pudieran dar si son implementados en la mano a construir. Básicamente los componentes principales que conformarían la mano robótica se pueden dividir en dos grupos: los actuadores, y los sensores.

Los actuadores corresponden a todo componente que permita realizar un movimiento luego de que la señal EMG ha sido procesada, y los sensores permiten tener una retroalimentación de la mano con los microcontroladores con el fin de tomar decisiones correctas ya sea con respecto a su ubicación, movimiento y presión empleada en los dedos. A continuación se describirá las características más importantes de los componentes utilizados.

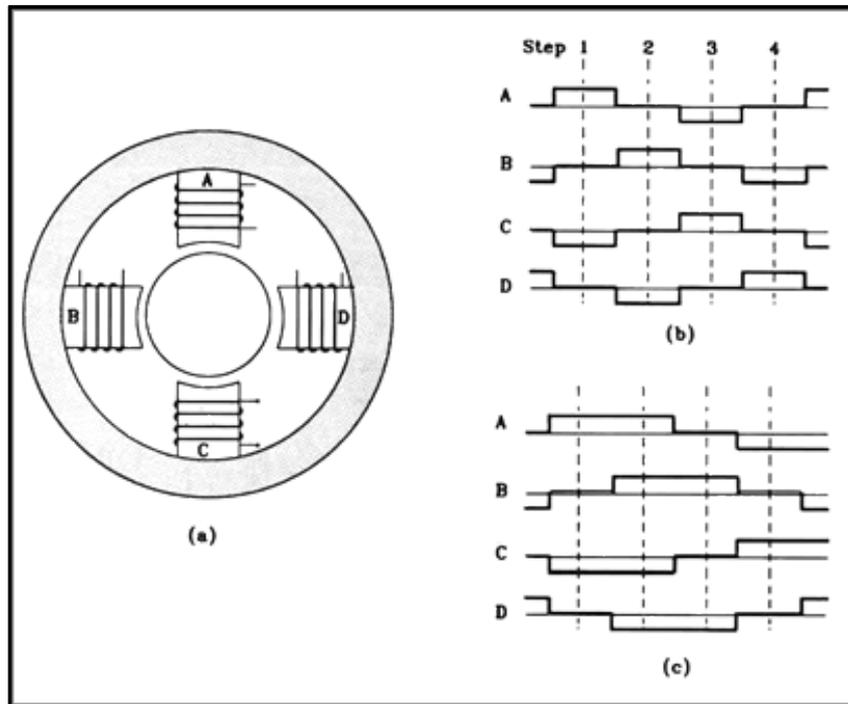
4.2.1 Actuadores eléctricos

Para el movimiento de la mano robótica, se optó por el uso de micro motores de DC con caja reductora y encoder integrado para los dedos y un motor de pasos para el control de la rotación de la muñeca, en los siguientes párrafos se describe las características principales de los componentes propuestos.

4.2.1.1 Motor de pasos

Cuando se trata de aplicaciones que requieren movimientos de alta precisión se recurre al uso de motores de pasos, esto es, debido a las características propias que los diferencian de cualquier otro tipo de motores. Como su nombre lo indica, este tipo de motor puede ser movido por pulsos o “pasos”, es decir, que dependiendo del tipo de motor, este puede girar un ángulo o paso desde 1.8° a 90° cada vez que se le dé un pulso en la bobina adecuada, esto implica que si se tiene un motor de 1.8° por paso se necesitan 200 pasos para dar una vuelta completa.

Figura 4. 5 Esquema interno típico de un motor de pasos a) secuencia para pasos de 90 grados b) pasos de 45 grados.



Fuente: <http://zone.ni.com>

Adicionalmente a la capacidad de controlar el giro del motor mediante secuencias de pasos, este tipo de actuadores también pueden quedar enclavados en una posición si se energiza a una o más bobinas, esto permite, tener un control total sobre el giro y freno del motor, facilitando el control del mismo.

Figura 4. 6 Secuencia de giro de un motor de pasos.

| PASOS | SECUENCIAS | | | | | | | | | | | |
|-------|------------|---|---|---|------------|---|---|---|-----------|---|---|---|
| | NORMAL | | | | WAVE DRIVE | | | | HALF STEP | | | |
| | A | B | C | D | A | B | C | D | A | B | C | D |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 5 | | | | | | | | | 0 | 0 | 1 | 0 |
| 6 | | | | | | | | | 0 | 0 | 1 | 1 |
| 7 | | | | | | | | | 0 | 0 | 0 | 1 |
| 8 | | | | | | | | | 1 | 0 | 0 | 1 |

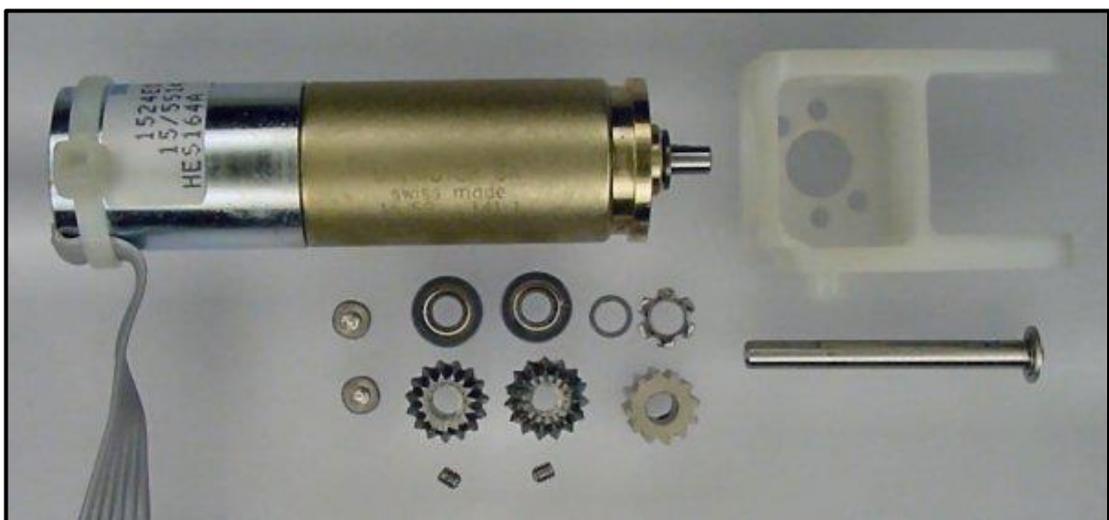
En el caso específico del presente tema de investigación se optó por el motor SY35ST26-0284A que será el encargado de realizar el movimiento de la muñeca de la mano robótica.

4.2.1.2 Motor DC Faulhaber 1524E006S123

Si bien es cierto que, como se pudo apreciar en el apartado anterior, los motores de pasos facilitan el control en aplicaciones donde se requieren movimientos de precisión, en el desarrollo del presente trabajo de investigación se optó por el uso de motores de precisión de corriente directa, esto se debe a que los motores de pasos son bastante costosos en comparación con los de corriente directa de similares dimensiones.

Con el fin de obtener un buen desempeño para en el control de los dedos de la mano robótica, se optó por el uso de motores de corriente directa de marca Faulhaber, que como característica adicional posee una caja reductora y un sistema de encoder incorporado de dos canales, de esta forma, se tiene una retroalimentación de la posición a la cual se encuentra un dedo de la mano robótica.

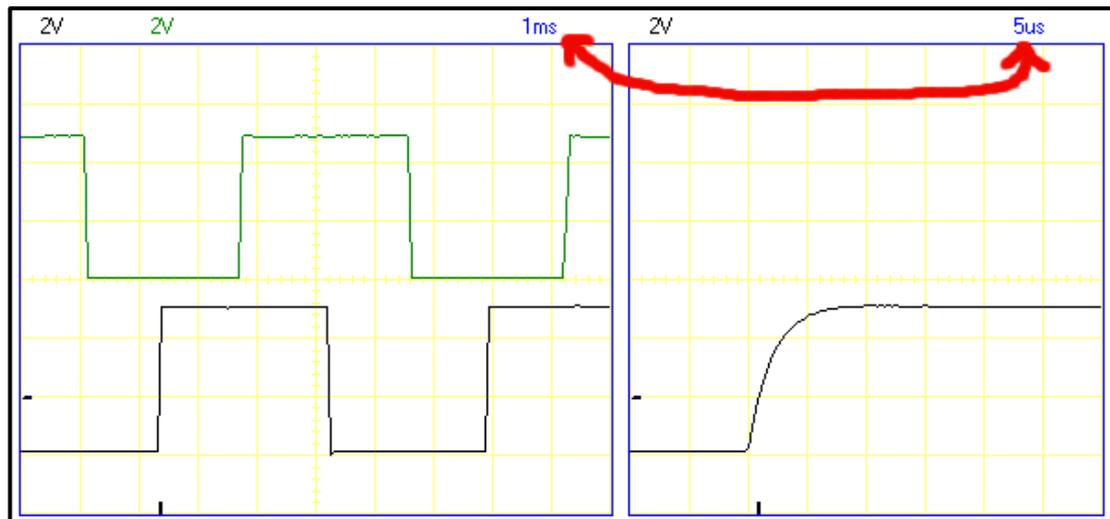
Figura 4. 7 Motor Faulhaber 1524E006S123.



Fuente: <http://www.robotroom.com>

La caja reductora posee una relación de 144 a 1, este dato hay que tenerlo presente al momento de realizar el firmware de control, pues el encoder incorporado produce un pulso alto por cada vuelta de eje, sin embargo, gracias a la caja reductora, hay que contar 144 pulsos por cada revolución que realiza la cabeza de engrane de la caja reductora.

Figura 4. 8 Pulsos de salida del encoder del motor Faulhaber.



Fuente: <http://www.robotroom.com>.

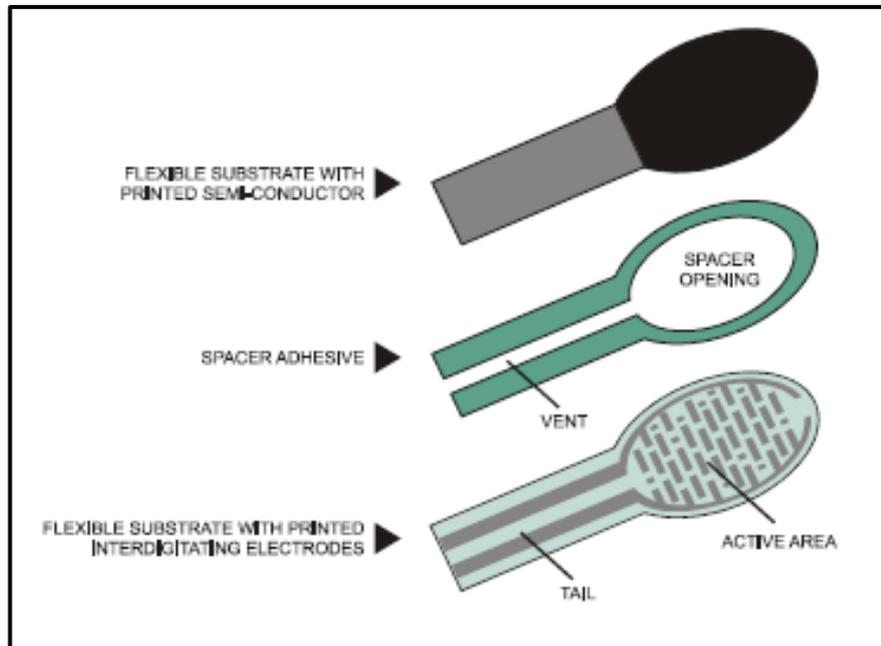
4.2.2. Sensores de presión

Una característica imprescindible que debe tener una mano robótica moderna, es sin lugar a dudas el poder controlar la presión ejercida sobre un objeto determinado, de esta forma se puede manipular objetos frágiles con relativa seguridad, sin embargo, la meta de alcanzar los rangos de presión que logra desempeñar una mano humana está todavía distante (incluso en prótesis electrónicas de alta gama) y abarca una área del conocimiento cuyo enfoque se extiende mucho más del presente tema de investigación.

Con el fin de obtener una retroalimentación de la presión ejercida, se recurre al uso de resistencias de censado de fuerza (FSR por sus siglas en inglés), están compuestos

de un polímero que presenta una disminución de su resistencia eléctrica cuando se produce un incremento en la fuerza aplicada sobre éste (Electronics, Interlink, [s.a]).

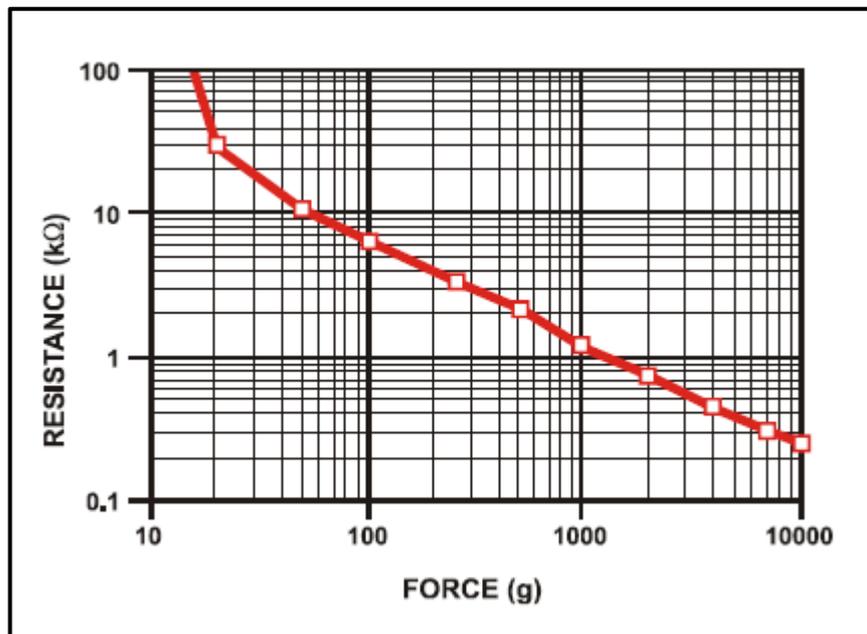
Figura 4. 9 Composición de un sensor FSR.



Fuente: Interlink, FSR Force Sensing Resistor, Integration Guide and Evaluation Parts Catalog.

En el gráfico inferior, se muestra el comportamiento característico de una resistencia FSR con respecto a la fuerza aplicada, cabe recordar que estos sensores no tienen una variación lineal con respecto a la fuerza aplicada.

Figura 4. 10 Curva característica de un sensor FSR.



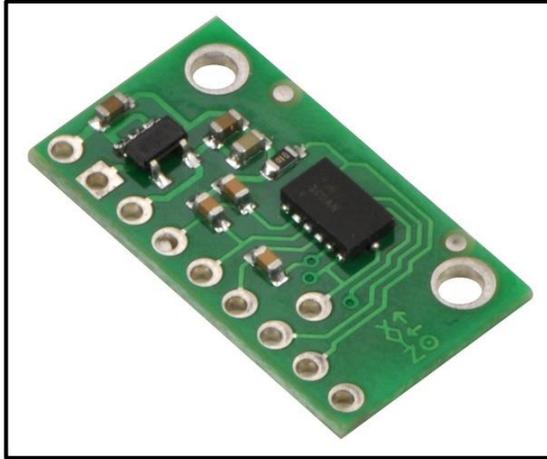
Fuente: Interlink, FSR Force Sensing Resistor, Integration Guide and Evaluation Parts Catalog.

4.2.3 Acelerómetro

En varios artículos de investigación donde se propone el uso de señales EMG para el control de prótesis inteligentes, se recurre a sistemas complejos de procesamiento multicanal de señales EMG para discriminar el tipo de movimiento que se desea realizar, sin embargo, esto implica el uso de algoritmos y sistemas de procesamiento complejos que hacen que el control de una prótesis se vuelva bastante engorroso, y por lo tanto, se tenga que recurrir a procesadores mucho más potentes, o sistemas que resultarían casi imposibles de hacerlos embebidos.

Por tal razón y con el fin de reducir en gran parte el proceso de discriminar cuando una señal EMG es empleada, ya sea, para producir un agarre o para rotar la muñeca, se recurre al uso de un acelerómetro, que al ser colocado a manera de reloj, permite conocer su ubicación sobre el eje z.

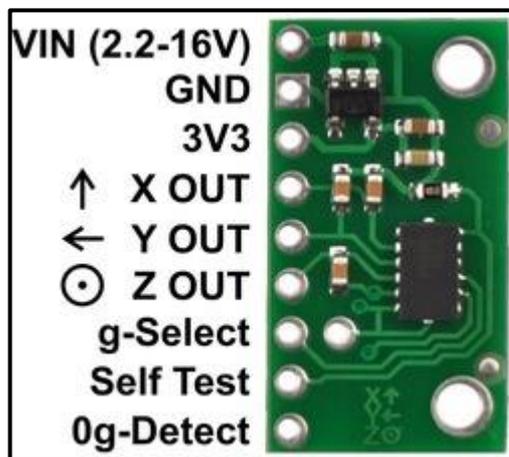
Figura 4. 11 Acelerómetro MMA7341L.



Fuente: <http://www.pololu.com>

El acelerómetro MMA7341L proporcionado por el proveedor de internet Pololu viene montado en un PCB con una regulador de voltaje que permite un rango de alimentación de 2.2V a 16V, posee tres salidas analógicas de voltaje y además su sensibilidad puede ser regulada (± 3 g or ± 11 g).

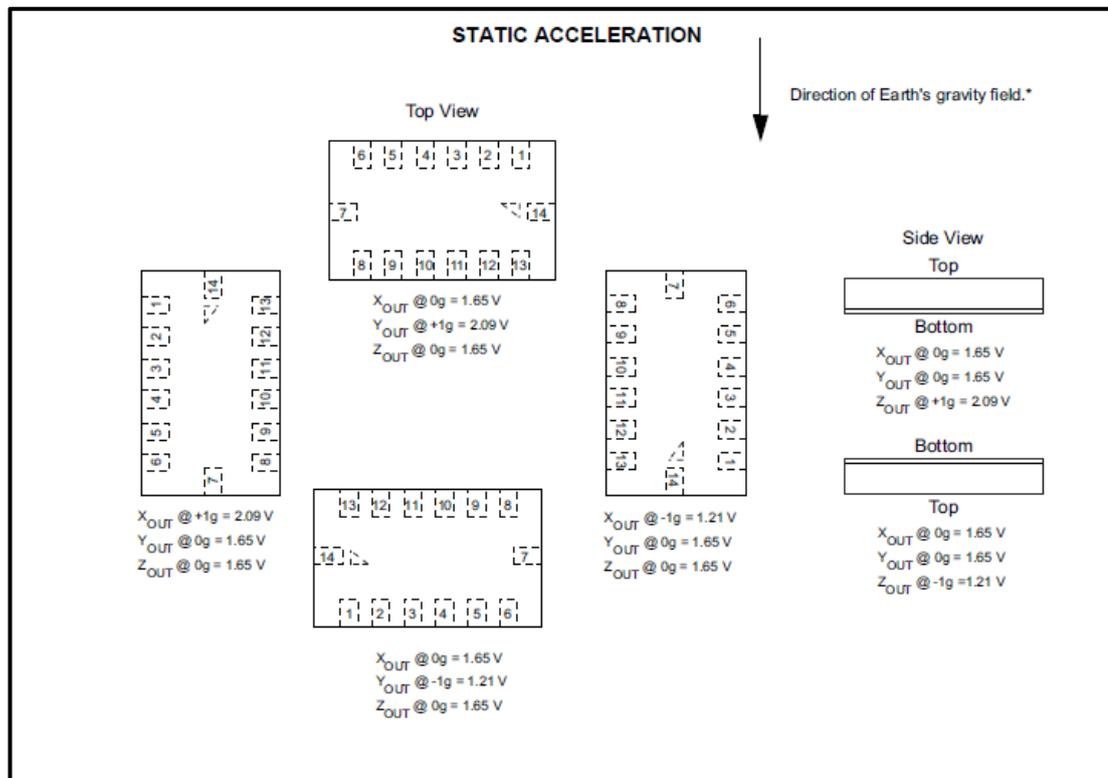
Figura 4. 12 Salidas existentes en la tarjeta del acelerómetro MMA7341L.



Fuente: <http://www.pololu.com>

En la figura 4.13 se puede observar los valores característicos de voltaje presentes en las salidas analógicas del MMA7341L cuando se producen movimientos por los diferentes ejes.

Figura 4. 13 Valores característicos de voltaje de MMA7341L.



Fuente: Freescale MMA7341L ±3g, ±11g Three Axis Low-g Datasheet

4.3 Diseño y construcción

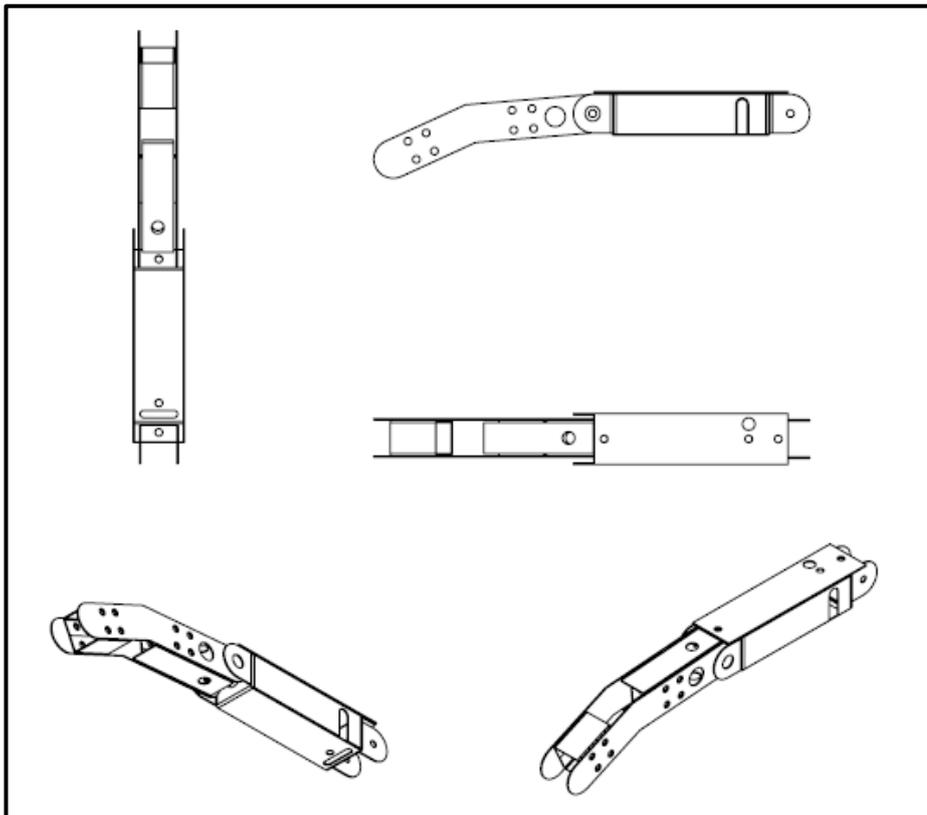
Teniendo en cuenta, tanto las proporciones anatómicas de una mano, y las limitaciones de tamaño que se tiene con los componentes seleccionados, se procede a diseñar una mano robótica con una relación de 1.5 a 1 con respecto a una mano normal, esto se debe principalmente al tamaño de los motores faulhaber que hacen imposible replicar una mano a escala real. Como material de construcción se opta por el empleo de aluminio pues además de ser bastante maleable, es bastante ligero.

La mano robótica constará de 4 dedos (índice, medio, anular y pulgar), cada uno de ellos pueden ser flexionados tanto a nivel del metacarpo (nudillos) como también al nivel de las falanges proximales (mitad del dedo), adicionalmente se propone la construcción del pulgar con un soporte a manera de pivote para que el pulgar pueda cumplir con al menos dos modelos prensiles de Schlesinger (Figura 4.4, cilíndrica palmar y subtermino lateral). Por último se colocara un soporte para que actúe a manera de muñeca para que el motor de pasos pueda girar la mano.

4.3.1 Diseño de los dedos

Considerando que, lo que se plantea construir es una estructura autoportable para los dedos, es decir, que el motor sea el que mantiene a la estructura junta, es sumamente importante tener en cuenta las dimensiones del motor Faulhaber para de esta forma, y gracias a las proporciones obtenidas anteriormente en el apartado 4.1, tener como punto de partida para obtener las dimensiones faltantes del resto del dedo a construir utilizando las relaciones de la figura 4.2.

Figura 4. 14 Estructura base para la construcción de los dedos.



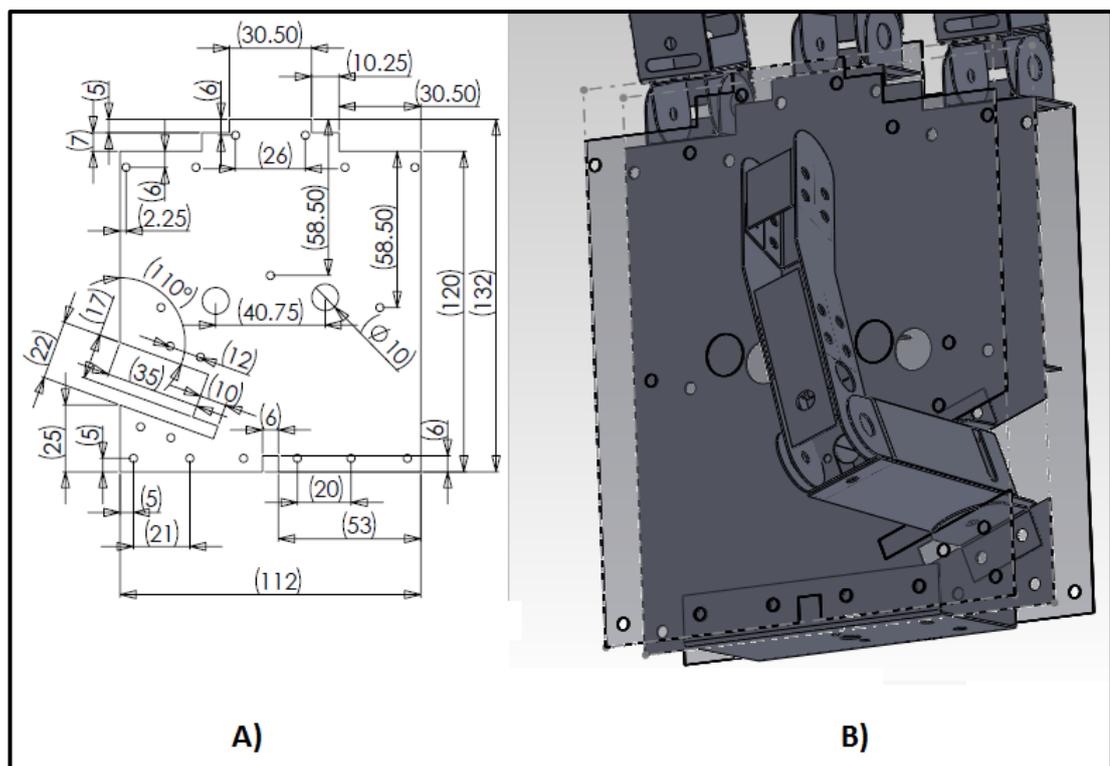
Para el caso de las puntas de los dedos se optó por el diseño semi flexionado de las mismas, de esta forma se obtiene un mejor agarre y una apariencia más natural de la mano, al igual que la estructura base del dedo, estos ángulos son proporcionados por las relaciones obtenidas en el apartado 4.1.

4.3.1 Diseño de la palma

Partiendo de la relación de proporciones palma vs dedo que posee una mano, se procedió a diseñar la palma de la mano robótica, en ella irán alojados los motores que permiten el movimiento de los dedos a nivel del metacarpo y las placas PCB que permiten el control de los motores y los hubs de sensores.

La única característica especial, físicamente hablando, consiste en el agujero a un ángulo de 110 grados, dejado para la ubicación del pulgar, nuevamente este ángulo parte de la observación de una mano humana.

Figura 4. 17 a) Diseño de la palma de la mano b) Mano con pulgar en posición de agarre cilíndrica palmar.



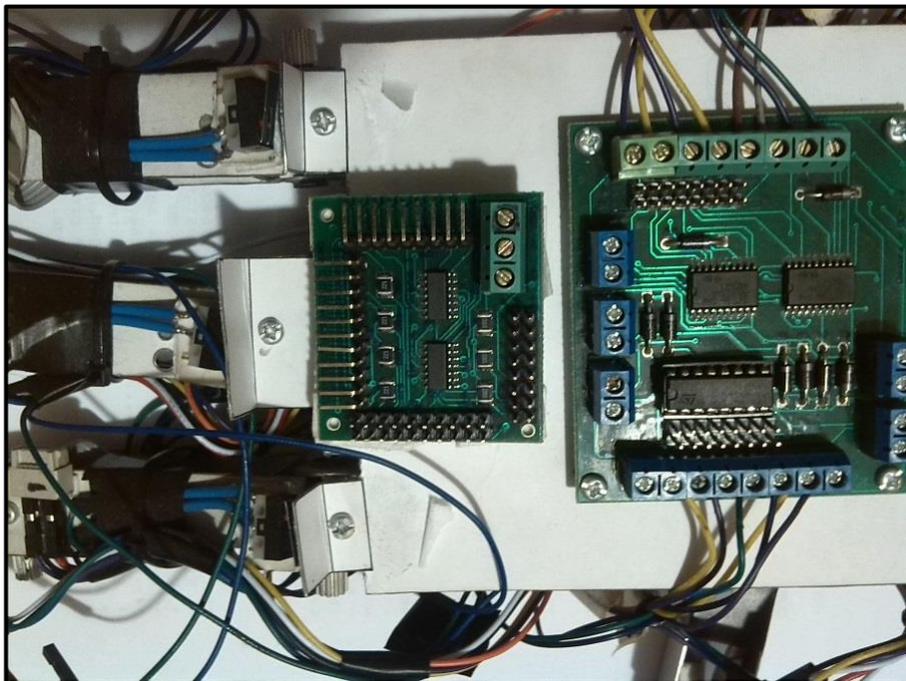
Los diseños completos y el montaje final se los puede apreciar en los anexos 3 y 4.

4.4 Montaje Final

Con las piezas que componen la mano robótica ya construidas, se procede al montaje final, en donde básicamente lo que se busca es ubicar de mejor forma tanto las tarjetas electrónicas para el control, los sensores, los actuadores, el cableado necesario para la comunicación de datos, etc.

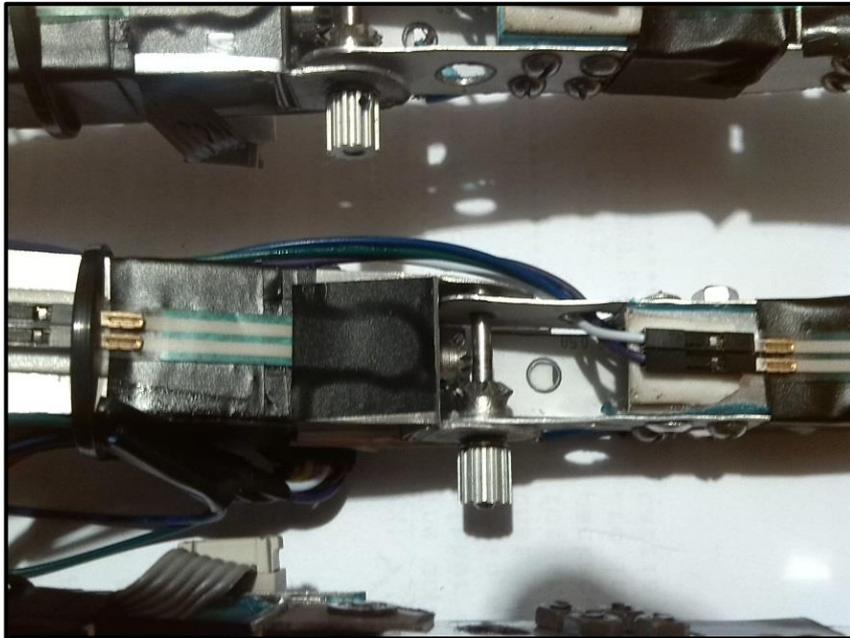
Para poder realizar las pruebas correspondientes y con el fin de tener un acceso rápido a la circuitería de la mano robótica, se optó por colocar las tarjetas MDrive y HSens en el dorso de la mano.

Figura 4. 18 Ubicación de la circuitería en el dorso de la mano.



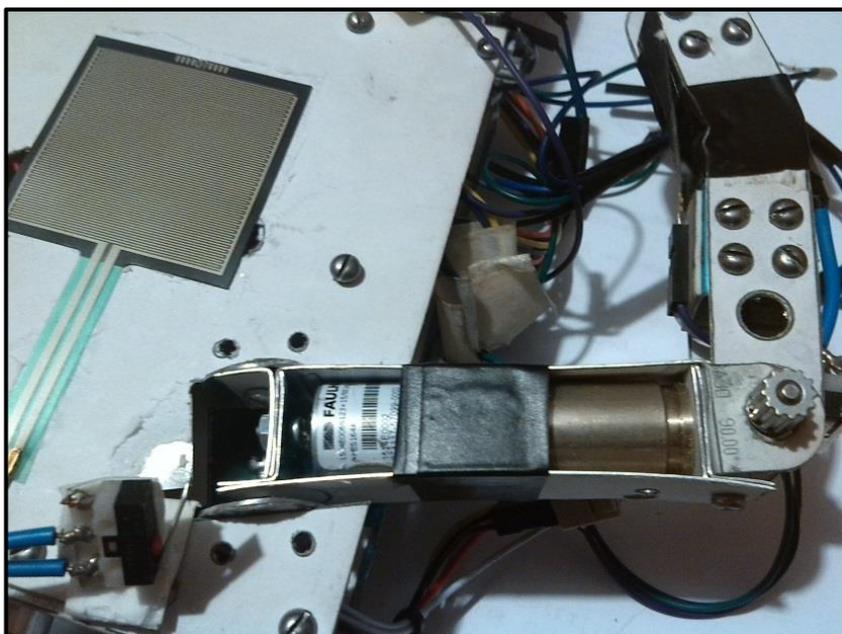
Cada dedo (a excepción del pulgar) requiere dos motores para su funcionamiento, uno a nivel de los nudillos ubicado dentro de la estructura de la palma, y otro a nivel del pliegue de flexión interfalángica media (para fines prácticos, la mitad del dedo), interconectados por un sistema de engranes ubicados en la estructura de los componentes de los dedos.

Figura 4. 19 Disposición de los dedos en la mano robótica.



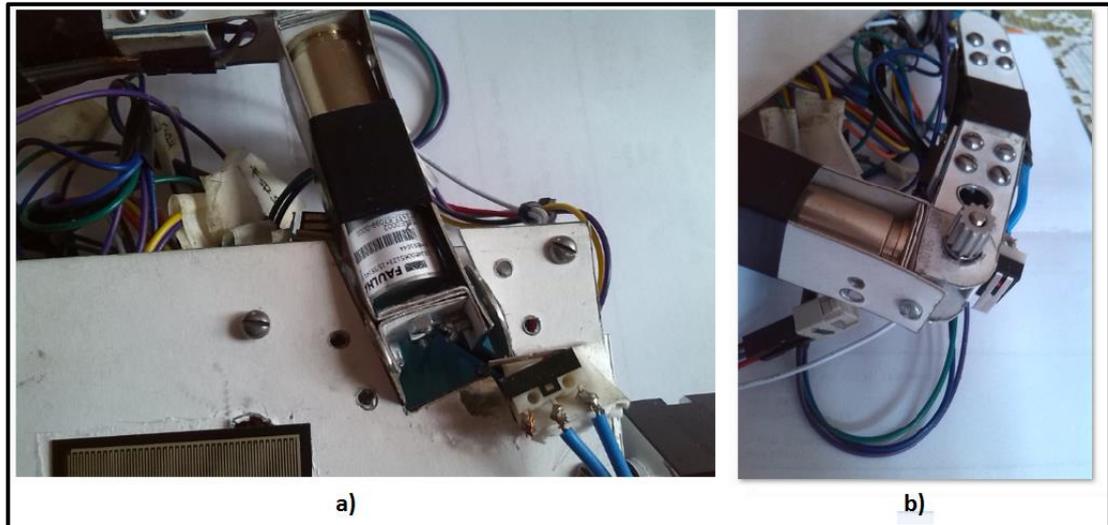
Con respecto al pulgar, este solo requiere un motor ubicado al nivel de la flexión interfalángica media para su funcionamiento, sin embargo además posee una base que funciona como pivote para poder seleccionar los diferentes modos de agarre.

Figura 4. 20 Disposición de la estructura del pulgar.



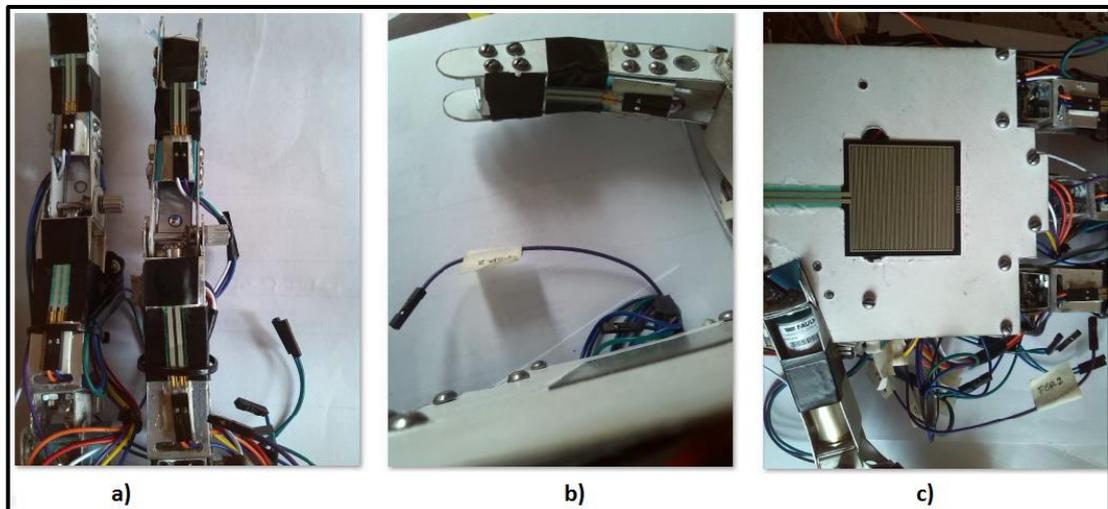
Adicionalmente, la mano robótica, posee varios pulsantes que son activados cuando el dedo recorre hacia una posición final; en el caso del pulgar, además de poseer un pulsante que le indica su posición final, posee dos pulsantes adicionales que le permiten accionar los diferentes tipos de agarre propuestos.

Figura 4. 21 Ubicación de los pulsantes en el pulgar a) pulsante de posición final b) pulsantes de tipo de agarre.



Como se mencionó en el apartado 4.2.2, esta mano posee sensores de presión que se encuentran ubicados a la altura de la primera y tercera falange (en el caso de los dedos anular, índice y medio) y en la segunda falange en el caso del pulgar. Adicionalmente, se tiene un sensor de presión en la palma de la mano también.

Figura 4. 22 Ubicación de los sensores de presión en a) Dedos índice, medio y anular b) Dedo pulgar y c) Palma de la mano.



Finalmente para permitir que el motor de pasos se conecte con el resto de la mano, se recurre a un Hub proporcionado por la empresa Pololu (Pololu, [s.a]) para el motor de pasos seleccionado anteriormente.

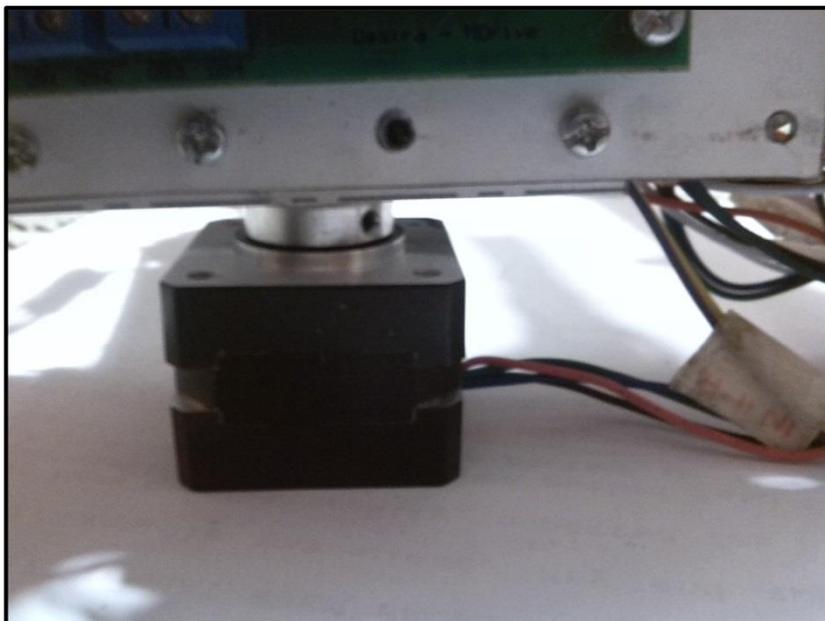
Figura 4. 23 Hub de montaje para el motor de pasos.



Fuente: www.pololu.com

Este hub irá unido mediante tornillos a la base de la mano robótica, tal y como se puede observar en la figura 4.24.

Figura 4. 24 Montaje del Hub del motor de pasos en la mano robótica.



CAPITULO 5:

SOFTWARE PARA EL CONTROL Y PRUEBA DE LA MANO ROBOTICA

Para el desarrollo del sistema de control de actuadores eléctricos mediante señales EMG, es necesario conocer los valores específicos que se generan cuando una persona produce una contracción muscular. Debido a las características propias de las señales EMG, que varían de acuerdo al individuo y a la ubicación donde se posicionan los electrodos, es indispensable tener un sistema que permita obtener estos valores para de esta forma calibrar al sistema electrónico.

Tomando en cuenta lo expuesto anteriormente, se decide utilizar el software para la creación de instrumentación virtual LabView en conjunto con la tarjeta de adquisición NI DAQ 6008, para visualizar y probar los diferentes circuitos que involucran el desarrollo del presente tema investigativo.

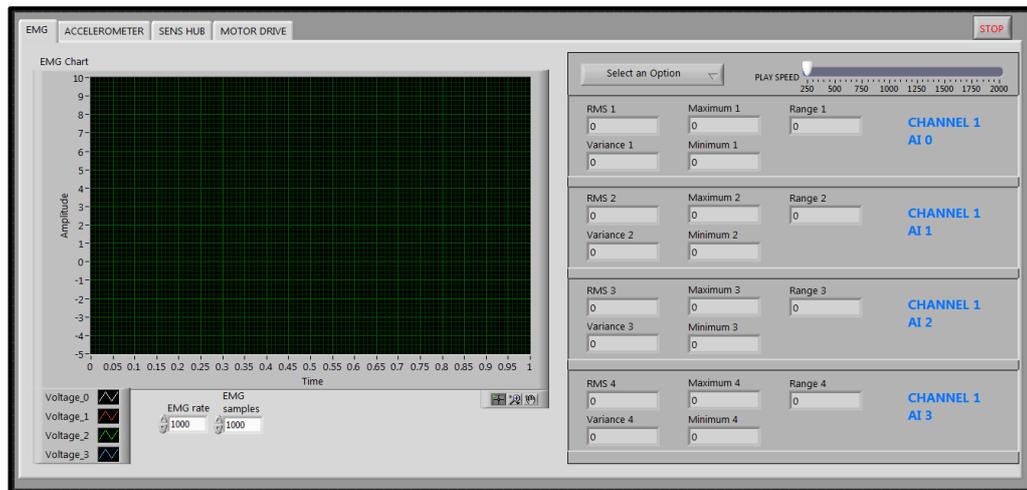
5.1 Diagramas de flujo, bloques y descripción de funcionamiento.

Considerando que, la principal función del software desarrollado será el medir y cuantificar los diferentes parámetros que existen en los circuitos PCB implementados para control de la mano robótica, se opta por dividir en cuatro pestañas (EMG, ACCELEROMETER, SENS HUB, MOTOR DRIVE), las mediciones más importantes según el circuito electrónico a probar. A continuación se explicarán los diagramas de flujo (en el caso de ser necesario) o los diagramas de bloque que componen cada una de las pestañas mencionadas.

5.1.1 Pestaña EMG

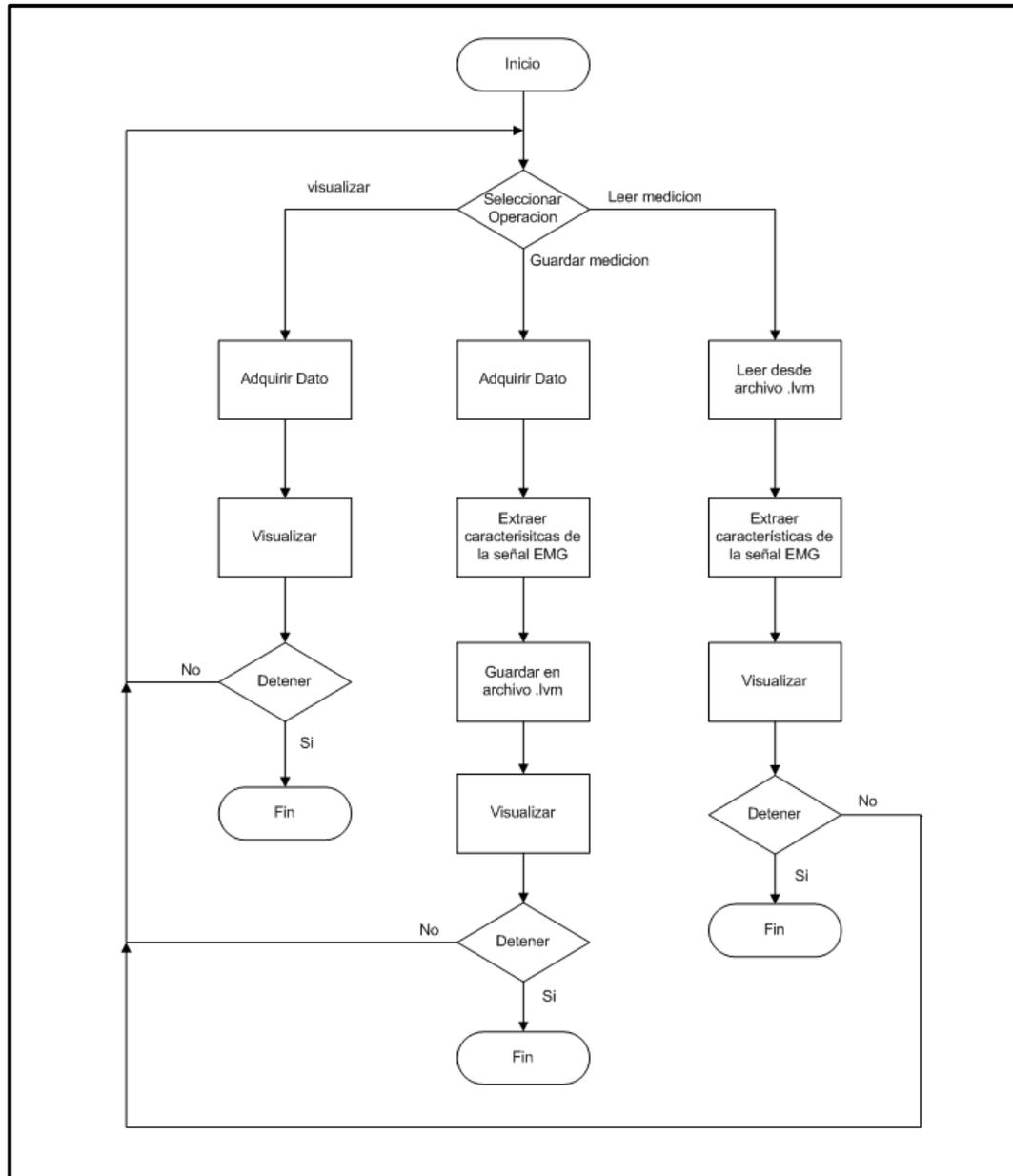
Es sin lugar a dudas el corazón del software de control y pruebas, permite visualizar, grabar y leer las señales EMG que son adquiridas por el circuito PCB EMG Acq4, además como una característica adicional, extrae valores relevantes para caracterizar las señales EMG, los cuales pueden ser empleados en otras aplicaciones (estudio de fatiga muscular, análisis de fuerza ejercida, etc) (Beng Guey, 2009).

Figura 5. 1 Panel Frontal pestaña EMG.



Al observar en el diagrama de flujo de la figura inferior, esta pestaña presenta tres modalidades de funcionamiento y en todos los casos se extraen las características más relevantes de la señal EMG.

Figura 5. 2 Diagrama de flujo de la pestaña EMG.



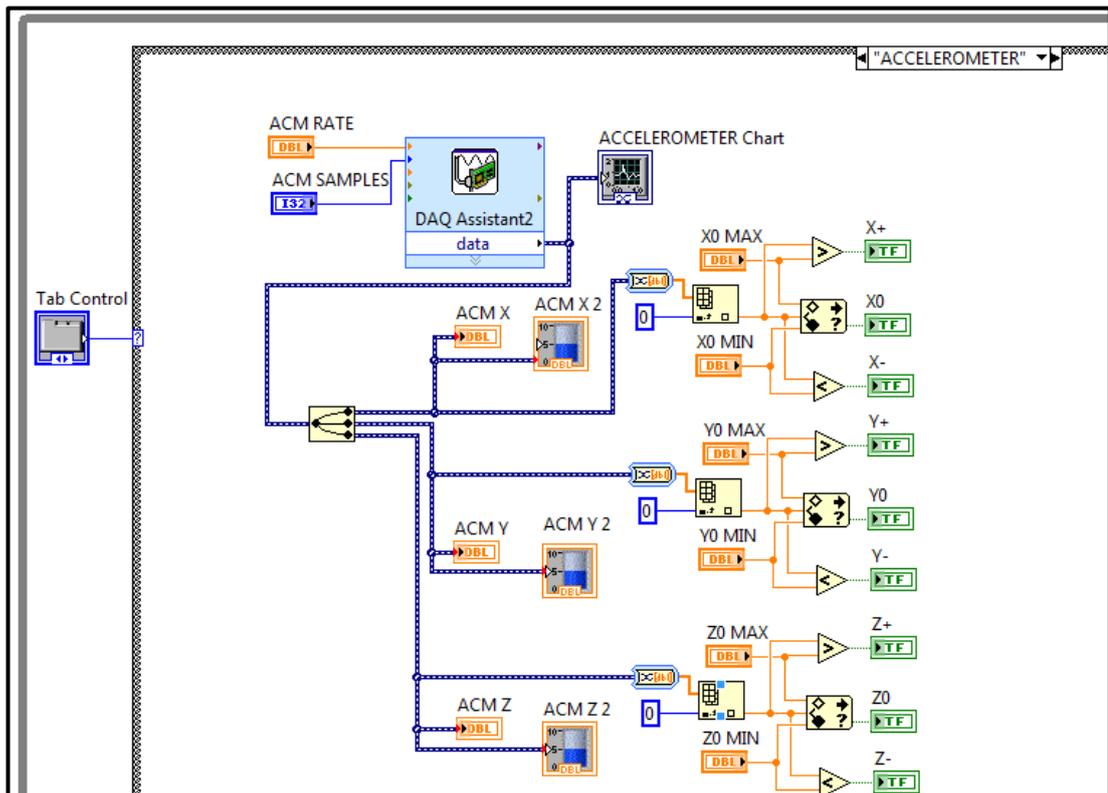
Cuando se trata de la visualización, únicamente se compone de un bucle que repite el muestreo hasta que sea detenida por el usuario (botón de stop en el panel frontal), en el caso de que se desee grabar la señal ingresada, el software utiliza una función express que permite grabar a un archivo de medición de extensión “.lvm” (extensión propietaria de LabView) para registrar tanto las mediciones como los valores a los que se encontraban calibrados los instrumentos al momento de realizar el muestreo. Por último, cuando se desea reproducir un archivo, al igual que en el caso anterior, se

recurre al uso de una función que permite a LabView leer un archivo de medición y reproducirlo, además se tiene la posibilidad variar el tiempo de reproducción mediante el uso de un temporizador.

5.1.2 Pestaña ACCELEROMETER

Esta pestaña básicamente adquiere las señales que se generan en la tarjeta del acelerómetro MMA7341L (Freescale Semiconductor, 2012), de esta forma es posible conocer dentro de que rangos de voltaje la muñeca permanece inmóvil y cuando esta produce un giro ya sea con la palma hacia arriba o hacia abajo.

Figura 5. 3 Diagrama de bloques del panel ACCELEROMETER.

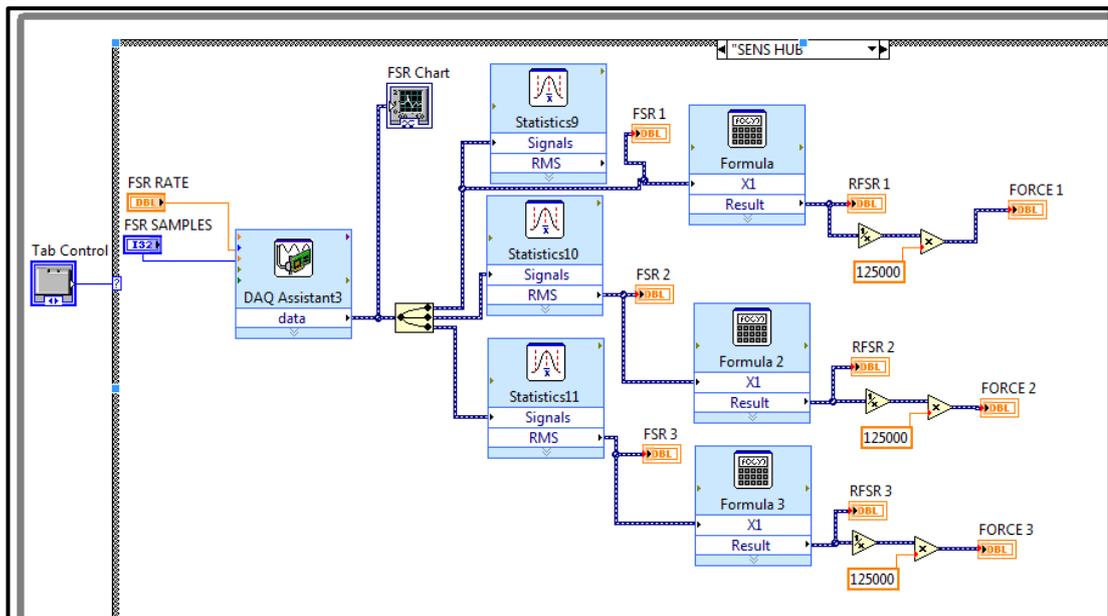


Al observar el diagrama de bloques de la figura 5.3, se puede constatar que, al igual que en la pestaña EMG existe un bucle que permite adquirir los datos generados por el MMA7341L y con la herramienta “In Range and Coerce” comparar si el acelerómetro funciona correctamente con los valores proporcionados en su hoja de datos (Freescale Semiconductor, 2008) o con cualquier valor que se desee contrastar.

5.1.3 Pestaña SENS HUB

Es la encargada de probar el funcionamiento de los sensores de presión en conjunto con la placa PCB H-Sens. En la figura 5.4, se observa que luego de adquirir la señal proveniente de los sensores, ésta es sometida a analizada mediante las formulas proporcionadas por el fabricante para determinar la resistencia del FSR y la fuerza que es ejercida sobre el sensor (Electronics, Interlink, [s.a]).

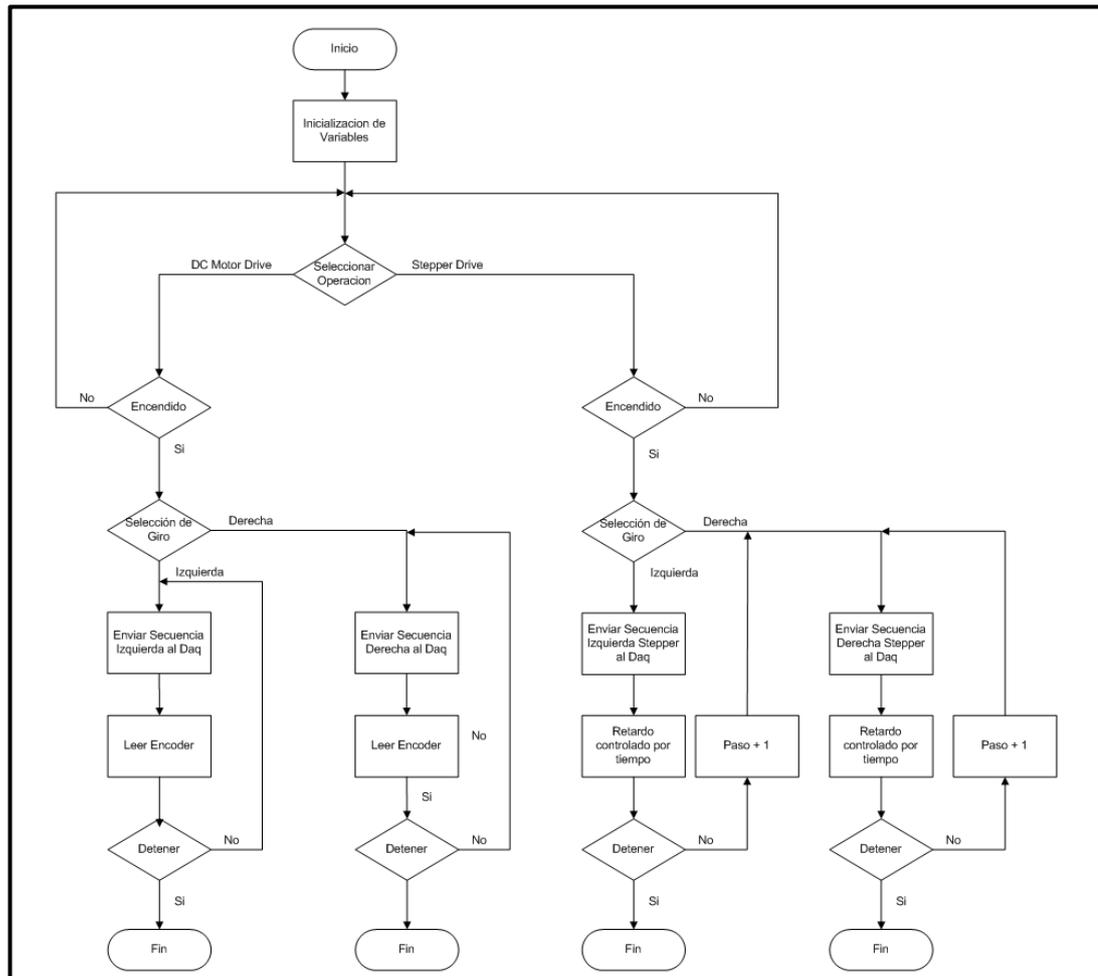
Figura 5. 4 Diagrama de bloques del panel SENS HUB.



5.1.4 Pestaña MOTOR DRIVE

Este módulo, permite hacer pruebas en la tarjeta MDrive que es la que controla tanto a los motores de corriente continua como a los motores de pasos, por tal razón, en esta pestaña consta de dos bancos de pruebas ya sea para motores DC como para motores de pasos.

Figura 5. 5 Diagrama de flujo del panel MOTOR DRIVE.



En la figura 5.5 se aprecia dos ramas en las que se divide el diagrama de flujo, la primera que es la encargada de las pruebas y control de motores DC y la segunda se encarga de los motores de pasos.

En el primer caso, se propone el control de un motor de corriente continua con su correspondiente cambio de giro, además como se mencionó en el apartado 4.2.1.2, el motor Faulhaber posee un encoder interno, razón por la cual además de controlar el motor también se propone adquirir los pulsos generados por su encoder. En el caso del motor de pasos, además del cambio de giro, se propone controlar la velocidad de giro del motor mediante demoras en el tiempo de ejecución de cada secuencia de paso; al igual que en el caso anterior, es necesario conocer la ubicación relativa del motor de pasos, es por esta razón que se propone un contador de pasos. El panel frontal y el diagrama de bloques completo de todas las pestañas detalladas en este capítulo se las puede apreciar en el anexo 5.

CONCLUSIONES

A lo largo del presente tema de investigación se planteó el uso de las señales electromiográficas superficiales SEMG para el control de actuadores eléctricos, si bien es cierto que el tema propuesto se logró realizar a cabalidad, también es necesario recalcar que el presente tema investigativo corresponde una pequeña parte dentro de un marco mucho más grande que constituye el estudio de señales EMG para su uso en prótesis robóticas u otras aplicaciones.

Sin embargo, el principal obstáculo de cualquier nuevo desarrollo de este tipo, es precisamente el alcanzar el conocimiento necesario para poder dar los primeros pasos, pues como es conocido, muy pocas veces existe una retroalimentación entre los estudios médicos y la ingeniería electrónica, por tal razón, es muy importante la síntesis teórica planteada en los capítulos 1 y 2 como punto de arranque para cualquier emprendimiento de este tipo. Cuando ya se posee el “Know How” (término acuñado por los países anglosajones para referirse al saber hacer algo), es relativamente sencillo transformar ese conocimiento en algo físico, en el caso específico del presente tema de investigación, en un circuito de adquisición de señales EMG.

El adquirir señales EMG y transformarlas en valores que puedan ser medibles y cuantificables, es la base para cualquier emprendimiento electrónico que desee usar señales SEMG, pues al tener valores con los cuales se pueden trabajar, ya la aplicación que se desee dar corresponde a conceptos con los cuales cualquier ingeniero o estudiante de electrónica se siente familiarizado.

Es por tal razón que el enfoque modular con el que se planteó el diseño de las tarjetas PCB constituye una gran ventaja al momento de realizar cualquier mejora o cambio a futuro, pues al ser modular es fácil reemplazar cualquier placa según las

necesidades que se presenten en determinado desarrollo investigativo, ya sea el procesador utilizado (por ejemplo usar un PIC 32) cambiar el tipo de sensores o simplemente usar la tarjeta de adquisición para una aplicación totalmente diferente al tema propuesto.

Y es precisamente la facilidad de uso de la tarjeta de adquisición en aplicaciones diferentes al planteado en el presente trabajo investigativo, por la cual se propone usar el software de control y pruebas para obtener mediciones de señales EMG y grabarlas para que luego otros investigadores puedan utilizarlas de acuerdo a sus necesidades (por ejemplo el estudio de fatiga muscular de un paciente).

Si bien es cierto que todavía queda mucho camino por adelante para lograr la meta final que sería el replicar con todas sus funcionalidades una mano humana; el desarrollo de un sistema embebido para el control de actuadores mediante señales EMG corresponde un avance importante en el desarrollo de aplicaciones bioelectrónicas pues constituye el punto de partida para aplicaciones mucho más complejas.

RECOMENDACIONES

Aunque se cumplió a cabalidad con el tema investigativo planteado, durante su desarrollo se pudo observar varias características que pueden ser mejoradas en futuros proyectos, pues como se explicó anteriormente, este trabajo constituye un paso inicial en el desarrollo de este tipo de aplicaciones bioelectrónicas y como todo emprendimiento nuevo siempre es susceptible a ser mejorado.

Tomando como meta la implementación de todo lo desarrollado a lo largo del presente trabajo investigativo en una prótesis robótica lo más similar a una mano humana, es necesario considerar tres factores importantes: tamaño, rendimiento energético y diseño mecánico.

Con respecto al tamaño, es necesario reducir las dimensiones de los circuitos PCB empleados, pues todavía son relativamente grandes para ser ubicados dentro de una mano de tamaño real; esto se podría lograr, reduciendo el número de canales de adquisición de la tarjeta EMG-ACQ4, usando de componentes de montaje superficial en todas las tarjetas y haciéndolas multilayer. También se podría unificar la tarjeta de control motores MDrive con el circuito HSens de los sensores y de igual manera, la tarjeta de adquisición EMG-ACQ4 con la tarjeta de los microprocesadores MCUPro.

Además sería bueno implementar una comunicación vía bluetooth entre las tarjetas de control y las de adquisición, pues con esto se evita cableado innecesario entre la mano y la muñeca, y además que esto permitiría desarrollar aplicaciones móviles donde se pueda calibrar la mano mediante un teléfono inteligente sin necesidad de recurrir a una computadora.

En cuanto al requerimiento energético, en el proyecto realizado, lamentablemente en el proyecto planteado no se pudo disponer de motores que presten las mismas

características de torque y fuerza con menor consumo energético, sin embargo en futuras aplicaciones se debe tener en cuenta este factor para poder utilizar baterías comunes y que el tiempo entre cargas permita por lo menos una autonomía de 6 horas.

Por último, en cuanto al diseño mecánico, como se explicó desde el inicio, el desarrollo de una prótesis robótica implica un estudio multidisciplinario, entre los cuales está también incluido la mecánica o mecatrónica, pues un correcto diseño mecánico permite reducir tanto el tamaño, el peso y la cantidad de motores a emplear, pues con un arreglo eficiente de engranes se podría controlar completamente un dedo con un solo motor y eso reduce tanto el tamaño, consumo energético y costo de la fabricación de una prótesis inteligente.

Todas estas recomendaciones son hechas en base a la apreciación particular del autor, mas no implica que sea el único camino a seguir, pues como se explicó anteriormente, este campo de estudio abarca muchas disciplinas las cuales pueden dar su enfoque particular en base a sus necesidades específicas.

BIBLIOGRAFÍA

- **Anatolandia. [s.a].** Planos Anatómicos. [Online] [s.a]. [Cited: 12 10, 2011.] <http://anatolandia.blogspot.com/2013/10/planos-anatomicos.html>.
- **Artrología. [s.a].** Clasificación de los movimientos articulares. *Artrología*. [Online] [s.a]. [Cited: 11 03, 2011.] <http://anatomica10.galeon.com/productos1842366.html>.
- **Beng Guey, Lau. 2009.** *An Intelligent Prosthetic Hand using Hybrid*. School of Mechanical Engineering, The University of Leeds. 2009. Tesis doctoral.
- **Burr Brown. 2005.** Precision, Low Power INA 128-129. *texas instruments*. [Online] 2005. [Cited: 10 04, 2012.] <http://www.ti.com/lit/ds/symlink/ina128.pdf>.
- **Coombs, Clyde F. 2001.** *Printed Circuits Handbook*. s.l. : McGraw-Hill, 2001.
- **De Luca, Carlo. 2006.** Electromyography. *Encyclopedia of Medical Devices and Instrumentation*. Boston : Boston University, John Wiley Publisher, 2006, pp. 98-109.
- **De Luca, Carlos J. 2008.** A Practicum on the Use of Surface EMG Signals in Movement Sciences. *Delsys*. [Online] 2008. [Cited: 11 1, 2011.] <http://www.delsys.com/KnowledgeCenter/Practicum.html>.
- **De Luca, Carlos J. 2002.** Surface Electromyography: Detection and Recording. *Delsys*. [Online] 2002. [Cited: 11 1, 2011.] http://www.delsys.com/KnowledgeCenter/Tutorials_Technical%20Notes.html.
- **De Luca, Carlos J. 1997.** THE Use of Surface Electromyography in Biomechanics. *Delsys*. [Online] 1997. [Cited: 11 10, 2011.] http://www.delsys.com/KnowledgeCenter/Tutorials_Technical%20Notes.html.
- **Delsys. 2001.** Fundamental Concepts in EMG Signal Acquisition. *Delsys*. [Online] 2001. [Cited: 11 10, 2011.] http://www.delsys.com/KnowledgeCenter/Tutorials_Technical%20Notes.html.
- **Delsys. [s.a].** Surface EMG Sensors. *Delsys*. [Online] [s.a]. [Cited: 11 10, 2011.] <http://www.delsys.com/Products/EMGSensors.html>.

- **Electronics, Interlink. [s.a].** FSR Force Sensing Resistor, Integration Guide and Evaluation Parts Catalog. *interlikelectronics*. [Online] [s.a]. [Cited: 02 20, 2012.] www.interlikelectronics.com.
- **Ferato. [s.a].** Electromiografía. *Ferato*. [Online] [s.a]. [Cited: 11 05, 2011.] <http://www.ferato.com/wiki/index.php/Electromiograf%C3%ADa>.
- **Freescale Semiconductor. 2008.** MMA7341L $\pm 3g$, $\pm 11g$ Three Axis Low-g. *pololu*. [Online] 2008. <http://www.pololu.com/file/0J378/MMA7361L.pdf>.
- **Freescale Semiconductor. 2012.** MMA7341L 3-Axis Accelerometer $\pm 3/11g$ with Voltage Regulator. *pololu*. [Online] 2012. [Cited: 01 02, 2012.] <http://www.pololu.com/catalog/product/1252>.
- **infovisual. [s.a].** Hueso de la mano. *infovisual*. [Online] [s.a]. [Cited: 11 02, 2011.] http://www.infovisual.info/03/027_es.html.
- **Jones, David L. 2004.** alternazone. [Online] 07 29, 2004. [Cited: 02 14, 2012.] <http://www.alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>.
- **Khandpur, R. S. 2006.** *Printed Circuit Boards, Design Fabrication and Assembly*. New Delhi : McGraw-Hill, 2006.
- **Microchip Technology Incorporated. 2006.** PIC18F2455/2550/4455/4550. [Online] 2006. [Cited: 04 10, 2012.] <http://ww1.microchip.com/downloads/en/DeviceDoc/39632c.pdf>.
- **Microchip Technology Incorporated. 2007.** PIC18F2458/2553/4458/4553. [Online] 2007. [Cited: 04 10, 2012.] <http://ww1.microchip.com/downloads/en/DeviceDoc/39887b.pdf>.
- **National Instruments. [s.a].** Types of Stepper Motors (detailed). *ni*. [Online] [s.a]. [Cited: 12 05, 2012.] <http://zone.ni.com/devzone/cda/ph/p/id/287>.
- **Oscar F. Aviles, Pedro L. Simanca, Paola A. Niño. [s.a].** Diseño y construcción de un dedo para grippers roboticos. s.l. : Universidad Estatal de Campinas, [s.a].
- **Pololu. [s.a].** Pololu Universal Aluminum Mounting Hub for 5mm Shaft, #4-40 Holes (2-Pack). *pololu*. [Online] [s.a]. [Cited: 12 01, 2012.] <http://www.pololu.com/catalog/product/1203>.
- **revista la reunion. 2008.** historia de las protesis de mano. [Online] 2008. [Cited: 04 03, 2012.] <http://revistalareunion.blogspot.com/2008/09/historia-de-las-prtesis-de-mano.html>.
- **Robot Room. [s.a].** Faulhaber Gear Motor and Encoder. *Robot Room*. [Online] [s.a]. <http://www.robotroom.com/FaulhaberGearmotor.html>.

- **Romo, Harold A, Realpe, Judy C and Jojoa, Pablo E. 2007.** Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano. Medellín : s.n., 2007, Vol. 4, pp. 127-136.
- **saberespractico. [s.a].** Huesos de la mano. *saberespractico*. [Online] [s.a]. [Cited: 12 14, 2012.] <http://www.saberespractico.com/curiosidades/cuantos-huesos-hay-en-la-mano-y-cuales-son-con-imagen/>.
- **Salud y medio ambiente. [s.a].** Planos anatómicos. *Salud y medio ambiente*. [Online] [s.a]. [Cited: 11 02, 2011.] <http://benitosyma.blogspot.com/2008/10/planos-anatmicos.html>.
- **Sandoval Gonzalez, Oscar Osvaldo. [s.a].** EMG Signals Real-Time Processing & Pattern Classification. *Skills*. [Online] [s.a]. [Cited: 11 08, 2011.] <http://www.skills-ip.eu/events/rss08/pdf/2.Skills%20Capturing.pdf>.
- **STMicroelectronics. [s.a].** Push-pull four channel driver L293B-L293E. *st*. [Online] [s.a]. [Cited: 04 10, 2012.] http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000059.pdf.
- **STMicroelectronics. 2003.** Push-pull four channel driver with diodes L293D-L293DD. *st*. [Online] 2003. [Cited: 10 04, 2012.] http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000059.pdf.
- **Weir, Richard F. and Sensinger, Jonathon W. 2009.** *Biomedical Engineering and Design Handbook*. [ed.] Myer Kutz. Second Edition. s.l. : McGraw-Hill, 2009. Vol. 2. ISBN: 978-0-07-170474-8.
- **Wikipedia. [s.a].** Amplificador no inversor. *Wikipedia*. [Online] [s.a]. [Cited: 04 12, 2012.] <http://es.wikipedia.org/wiki/Archivo:Opamponinverting.png>.
- **Wikipedia. [s.a]** Amplificador operacional. *Wikipedia*. [Online] [Cited: 04 12, 2012.] http://es.wikipedia.org/wiki/Amplificador_operacional.
- **Wikipedia. [s.a].** Lógica difusa. *Wikipedia*. [Online] [s.a]. [Cited: 11 11, 2011.] http://es.wikipedia.org/wiki/Logica_difusa.
- **Wikipedia. [s.a].** Neurona. *Wikipedia*. [Online] [s.a]. [Cited: 11 05, 2011.] <http://es.wikipedia.org/wiki/Neurona>.
- **Wikipedia. [s.a].** Plano anatómico. *Wikipedia*. [Online] [s.a]. [Cited: 11 02, 2011.] http://es.wikipedia.org/wiki/Plano_anat%C3%B3mico.
- **Wikipedia. [s.a].** Sallen–Key topology. *Wikipedia*. [Online] [s.a]. [Cited: 02 15, 2012.] http://en.wikipedia.org/wiki/Sallen%E2%80%93Key_topology.
- **Wikipedia. [s.a].** Terminología anatómica de localización. *Wikipedia*. [Online] [s.a]. [Cited: 11 02, 2011.]

http://es.wikipedia.org/wiki/Terminolog%C3%ADa_anat%C3%B3mica_de_localizaci%C3%B3n.

- **Wilson, Peter. 2012.** *The circuit designers companion*. s.l. : Newnes, 2012.

ANEXOS

ANEXO 1: ESQUEMAS Y DISEÑOS DE PCB

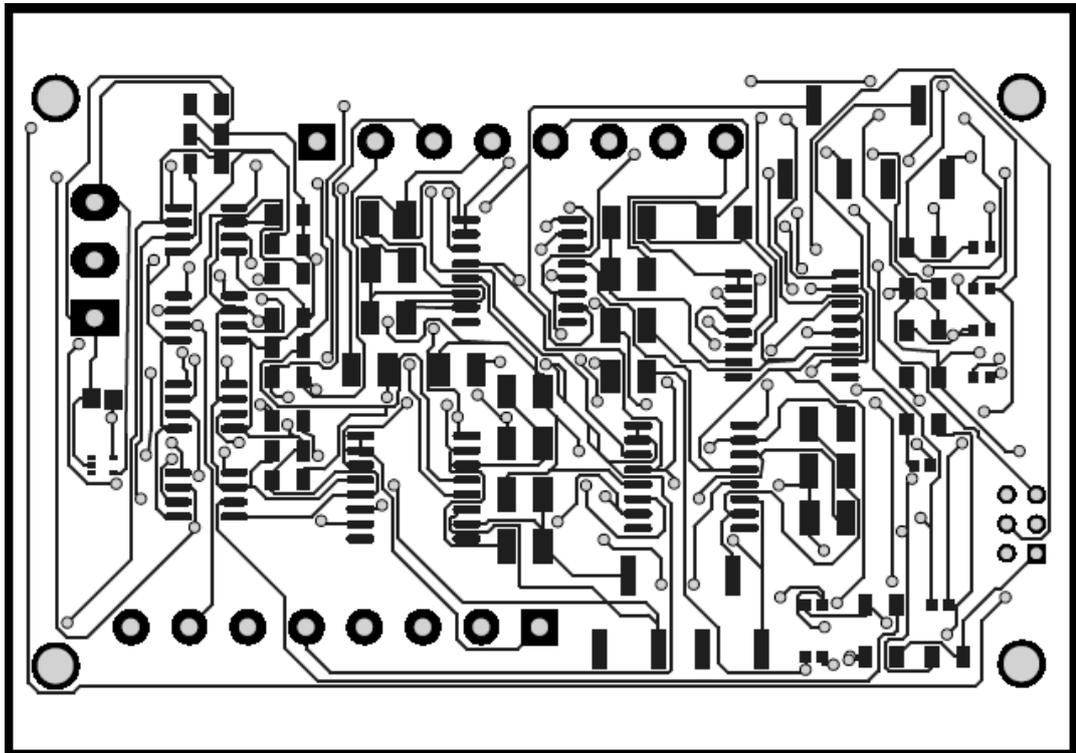


Figura A1. 1 Cara superior del PCB Emg-Acq 4

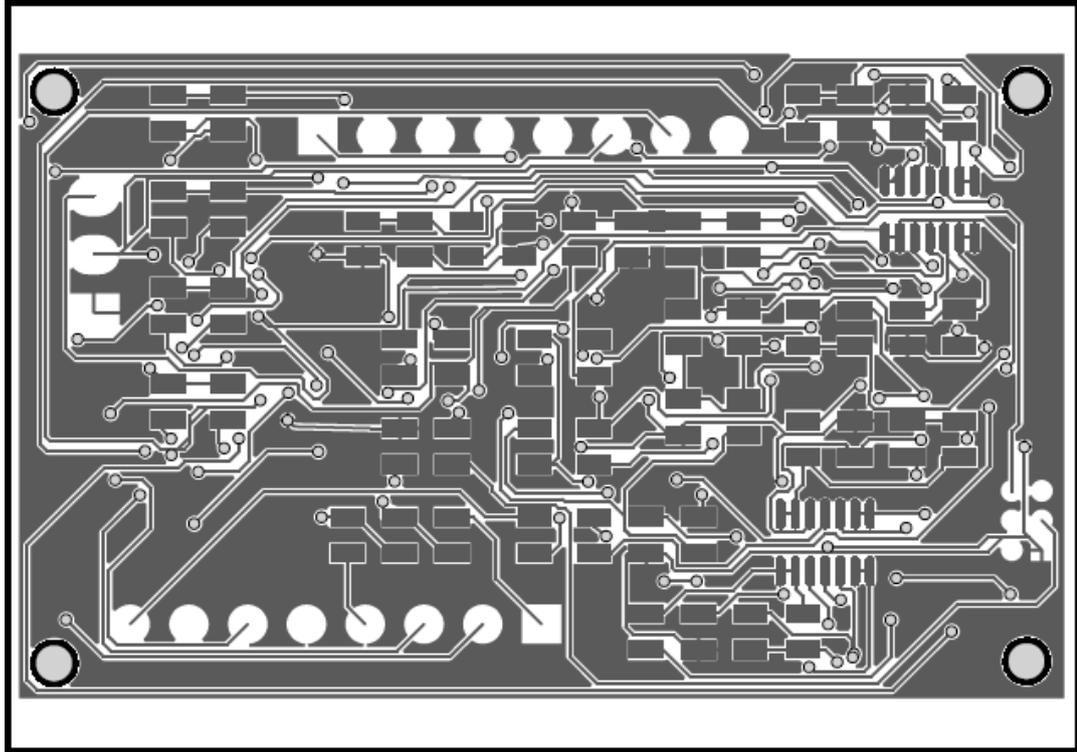


Figura A1. 2 Cara inferior del PCB Emg-Acq 4

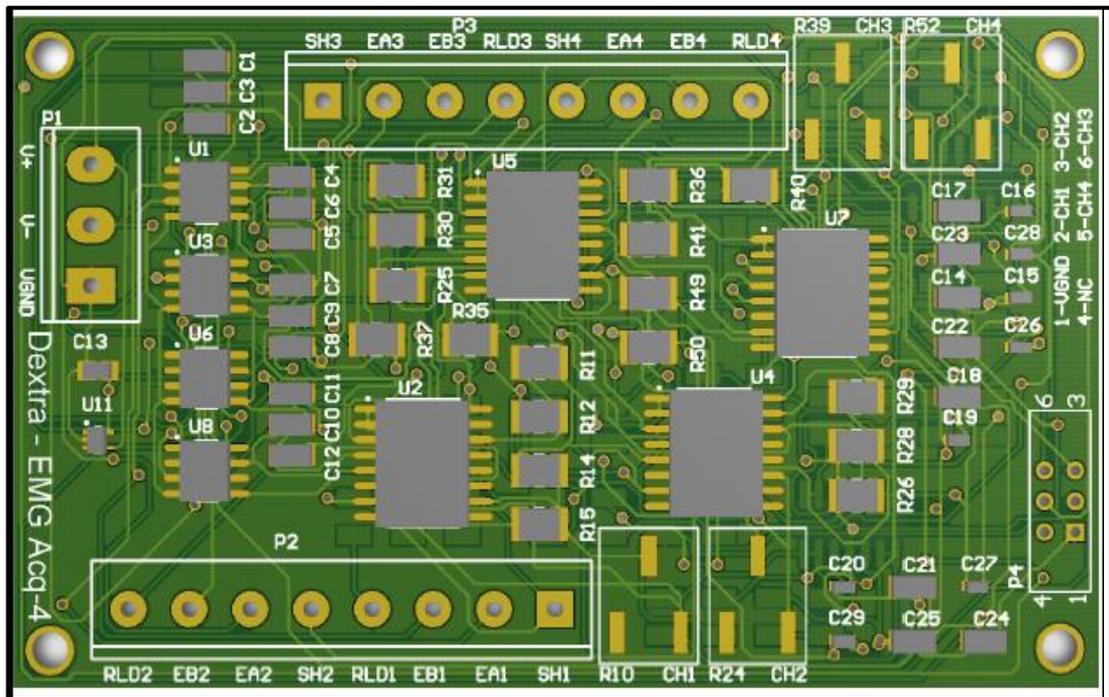


Figura A1. 3 Vista 3D cara superior EMG – Acq 4

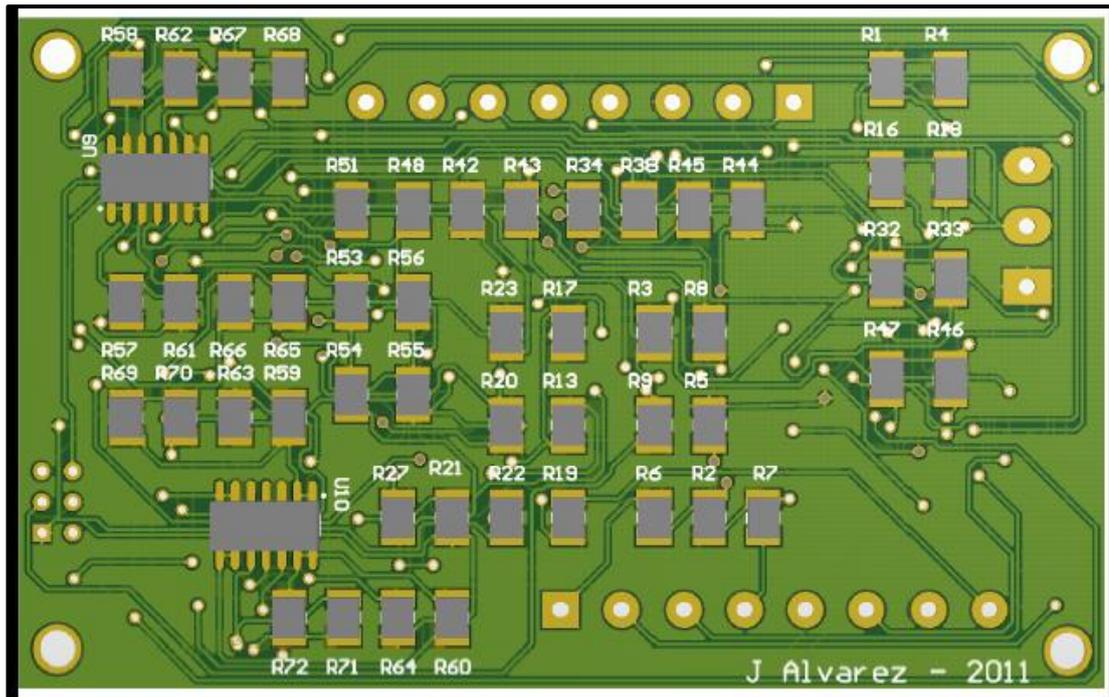


Figura A1. 4 Vista 3D cara inferior EMG – Acq 4

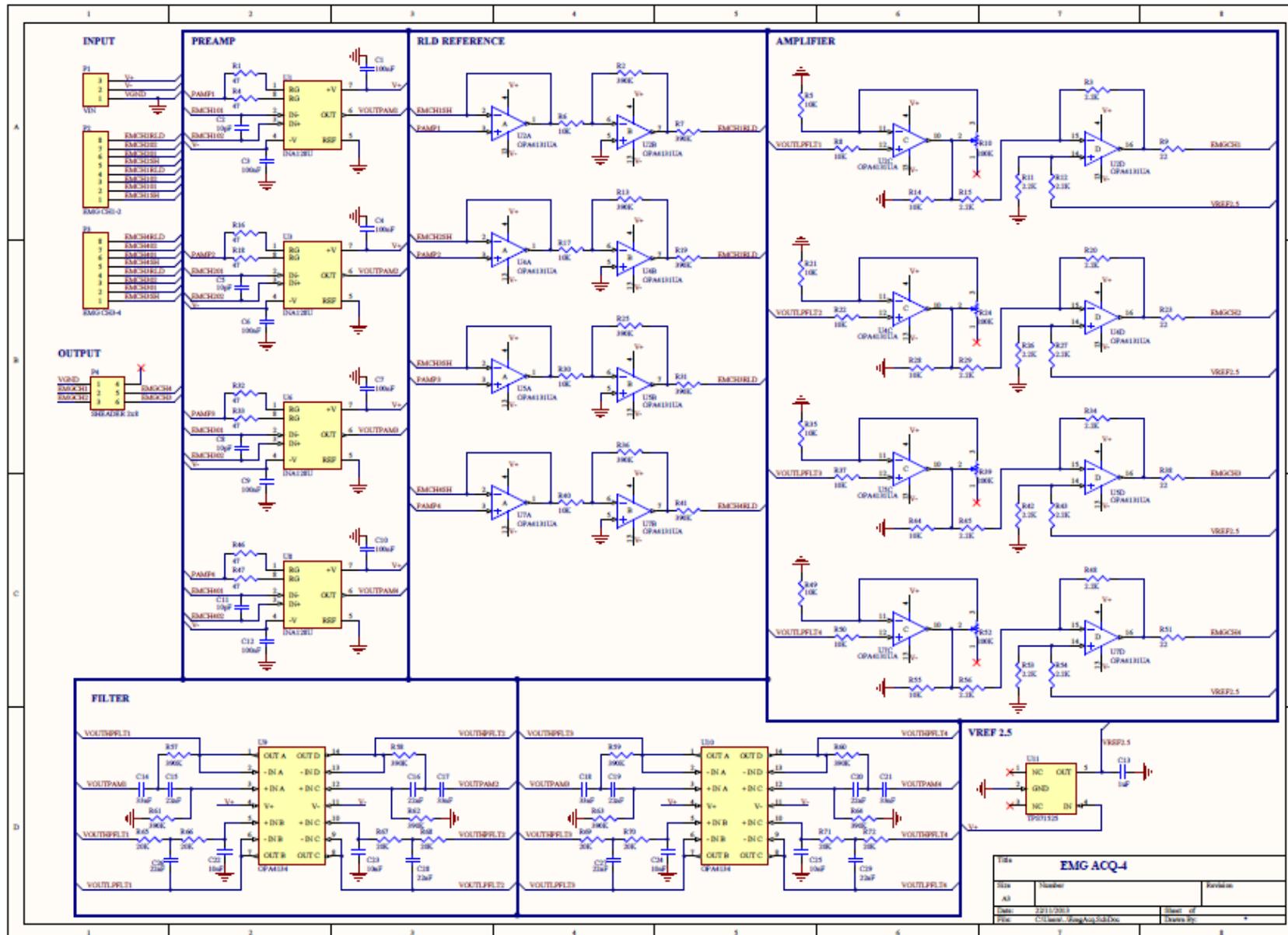


Figura A1. 5 Esquemático del circuito EMG – Acq 4

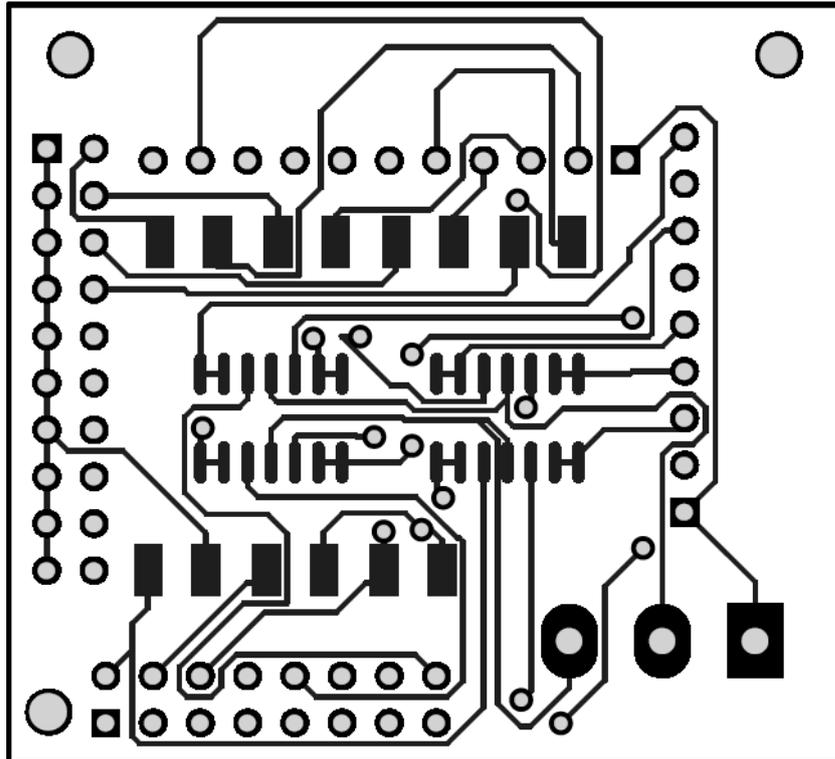


Figura A1. 6 Cara superior del PCB H-Sens

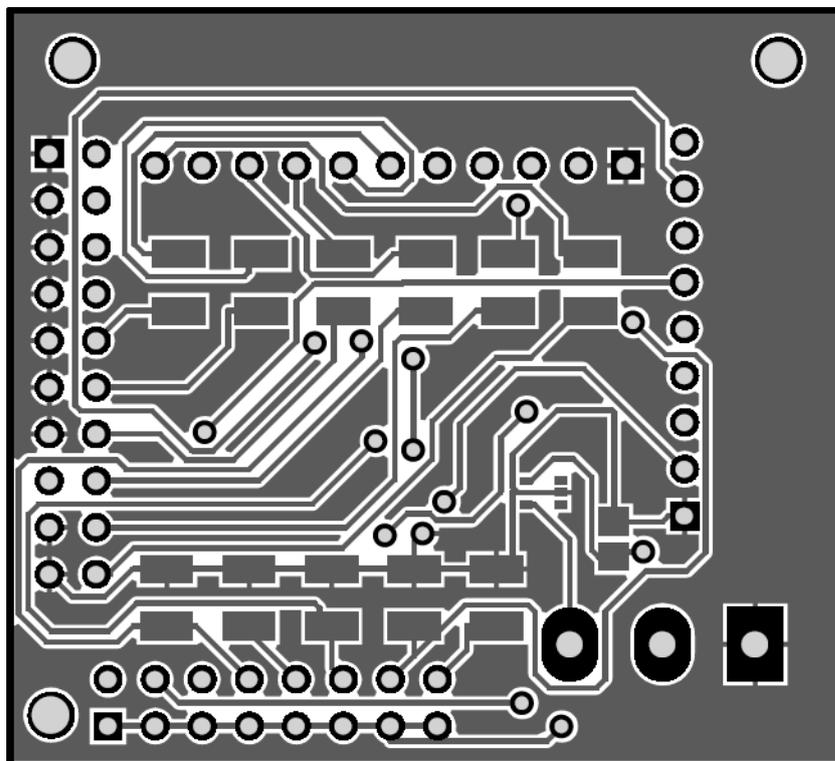


Figura A1. 7 Cara inferior del PCB H-Sens

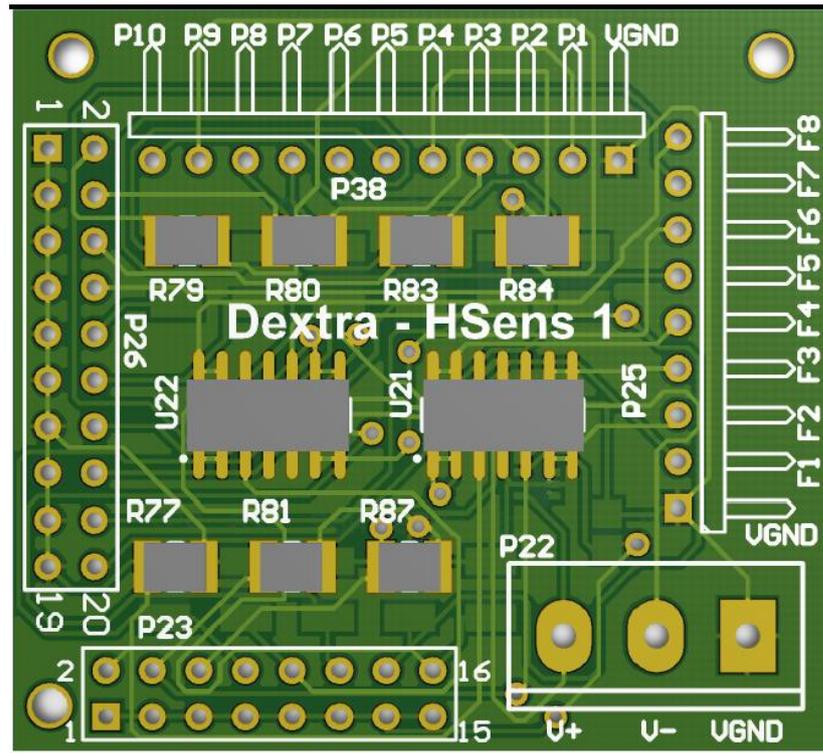


Figura A1. 8 Vista 3D cara superior H-Sens

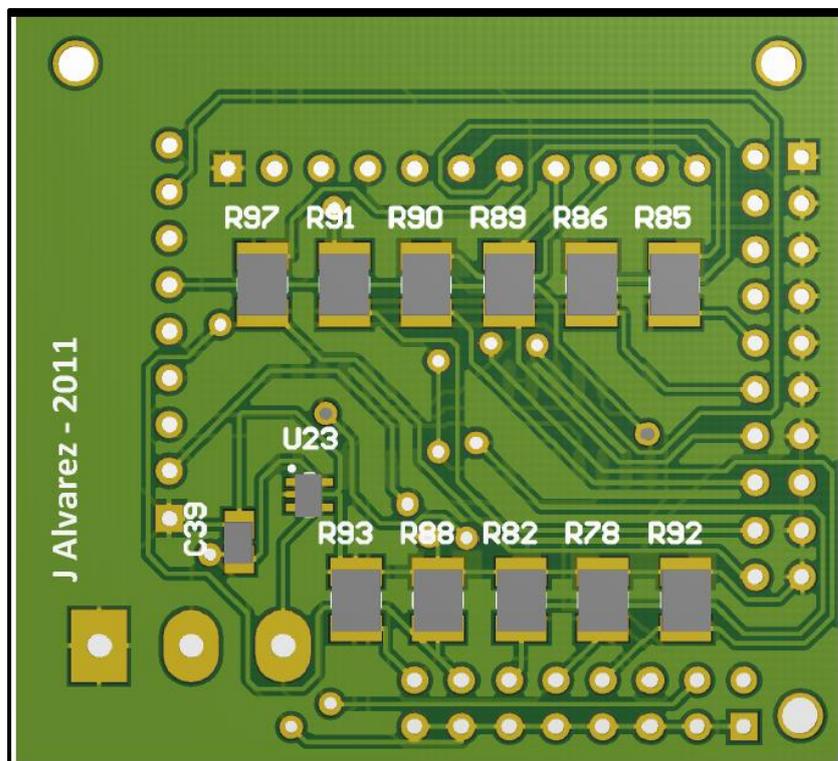


Figura A1. 9 Vista 3D cara inferior H-Sens

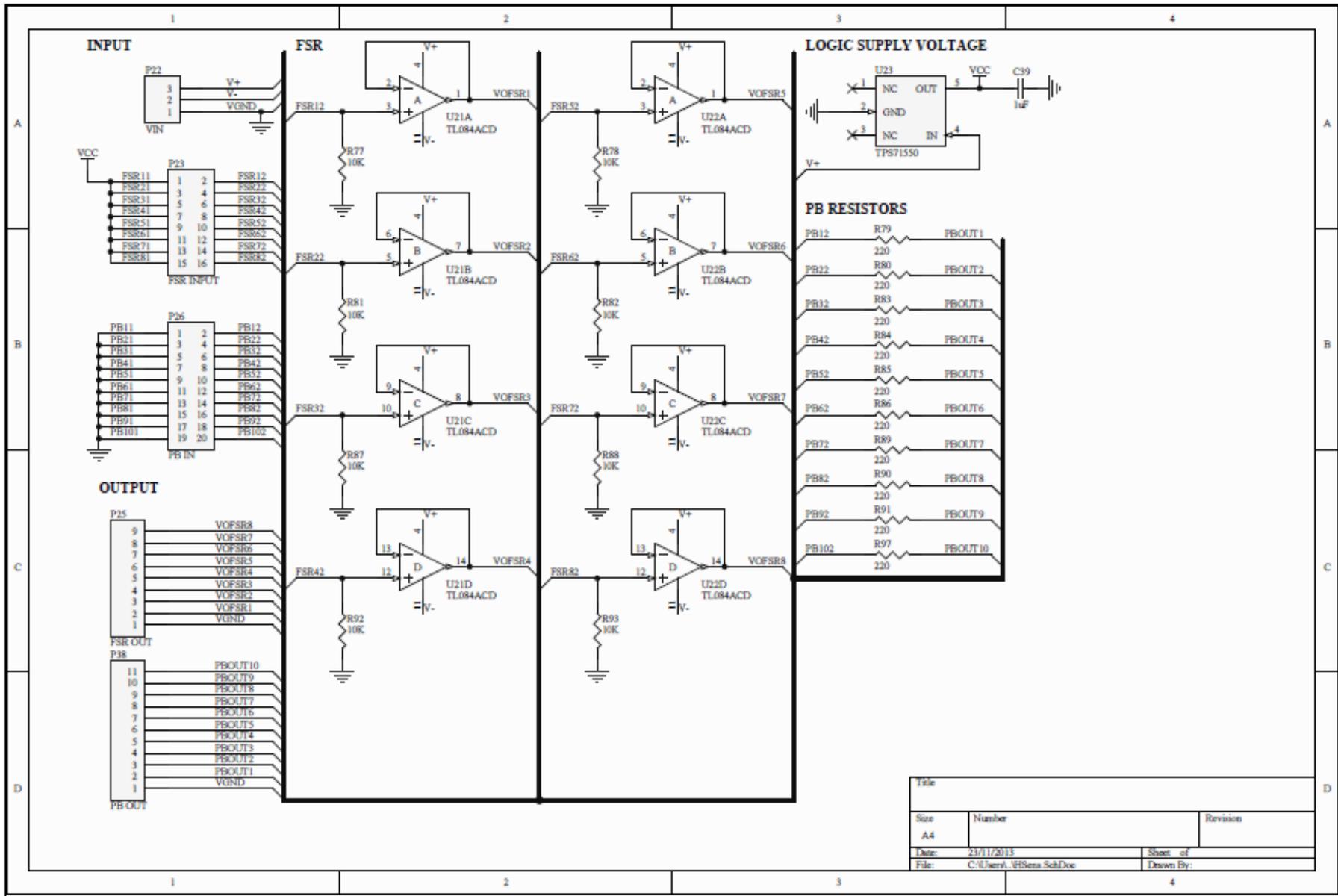


Figura A1. 10 Esquemático del circuito H-Sens

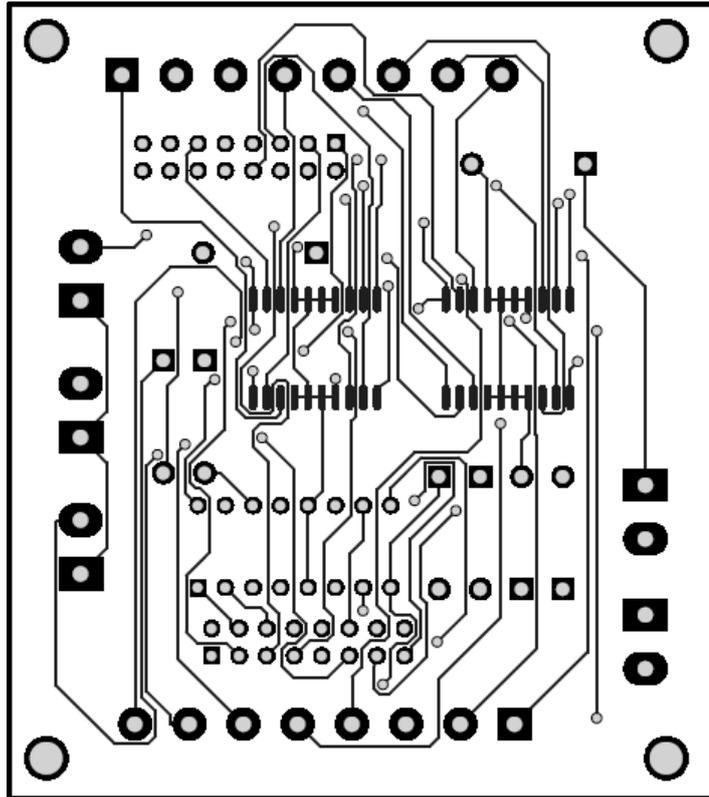


Figura A1. 11 Cara superior del PCB MDrive

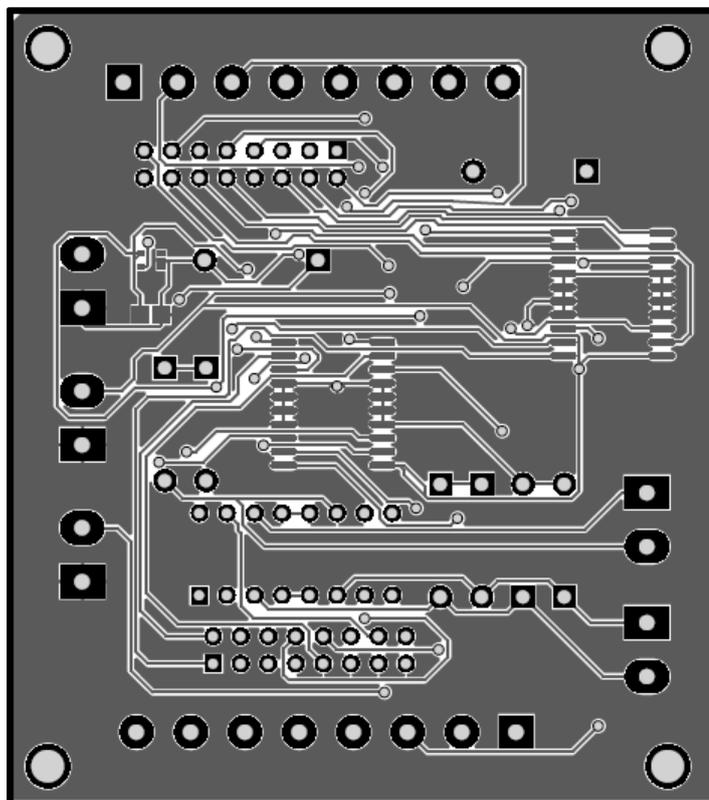


Figura A1. 12 Cara inferior del PCB MDrive

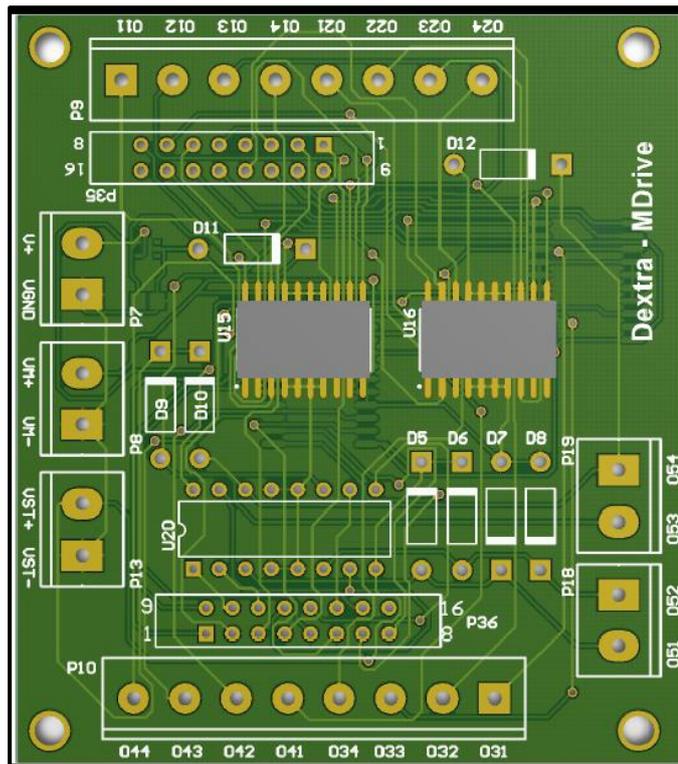


Figura A1. 13 Vista 3D cara superior MDrive

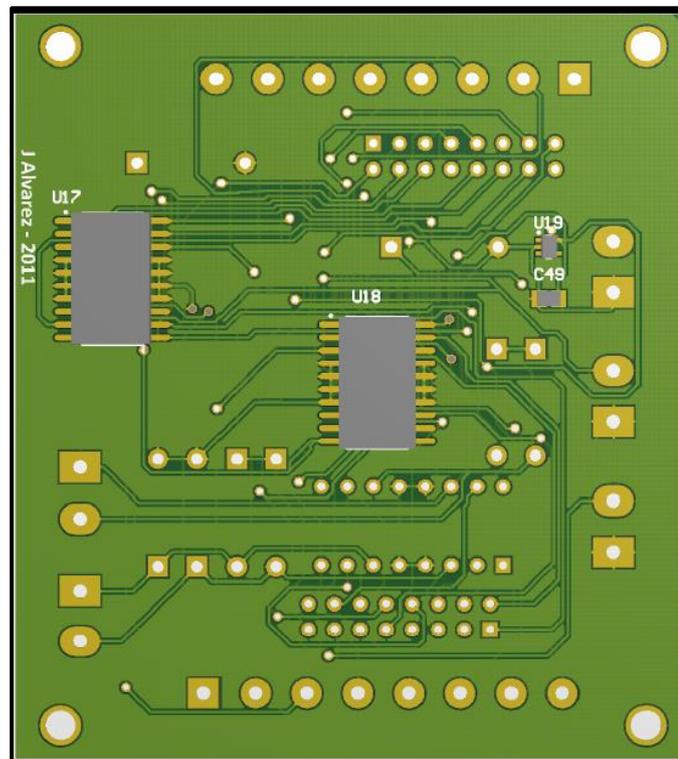


Figura A1. 14 Vista 3D cara inferior MDrive

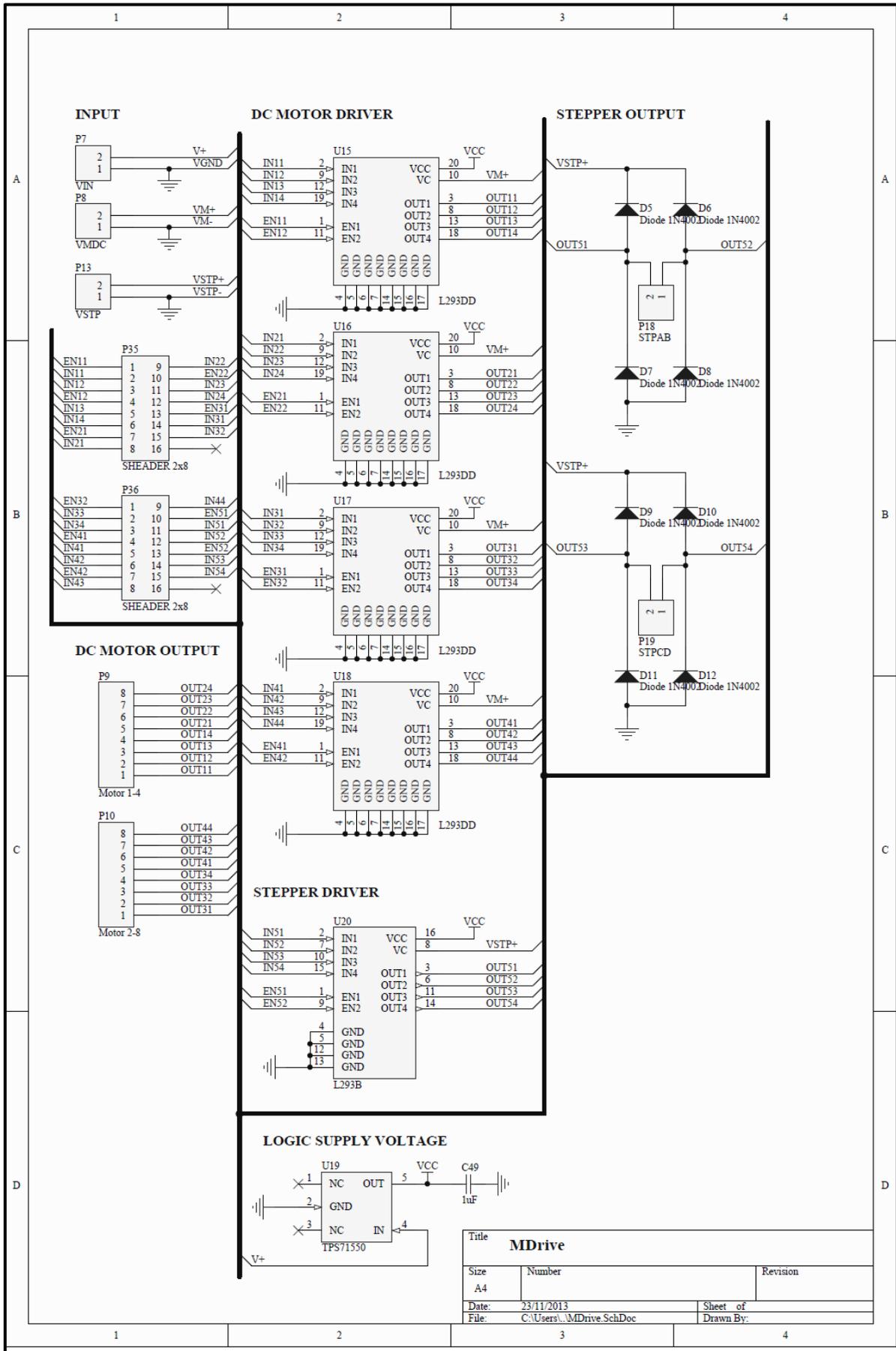


Figura A1. 15 Esquemático del circuito MDrive

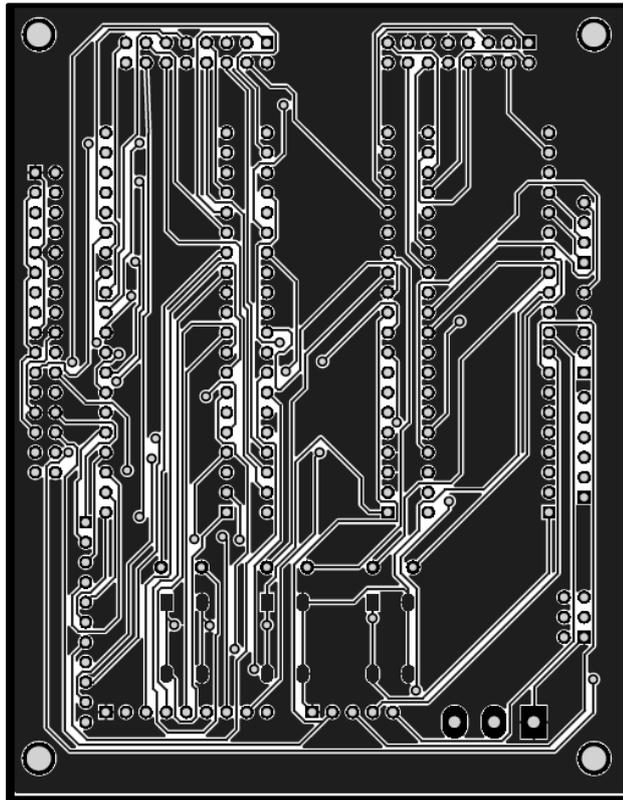


Figura A1. 16 Cara superior del PCB MCUPro

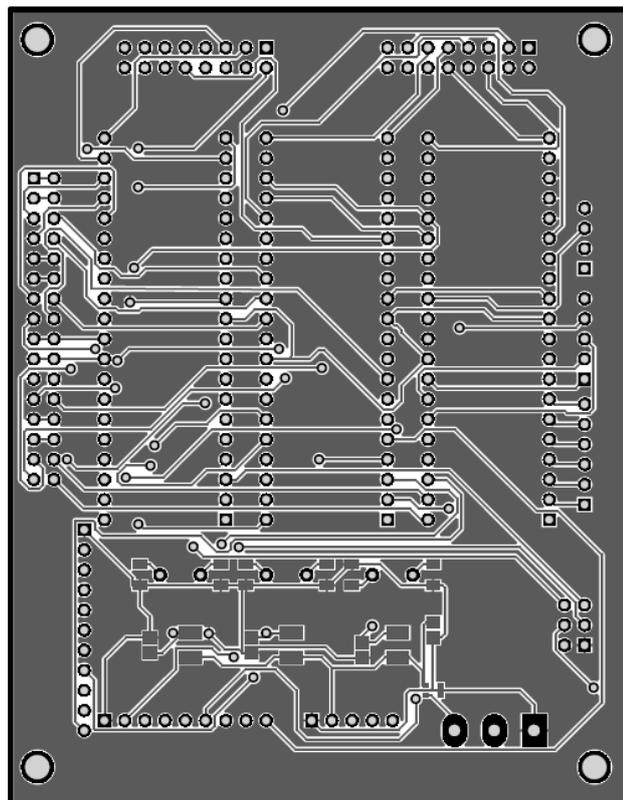


Figura A1. 17 Cara inferior del PCB MCUPro

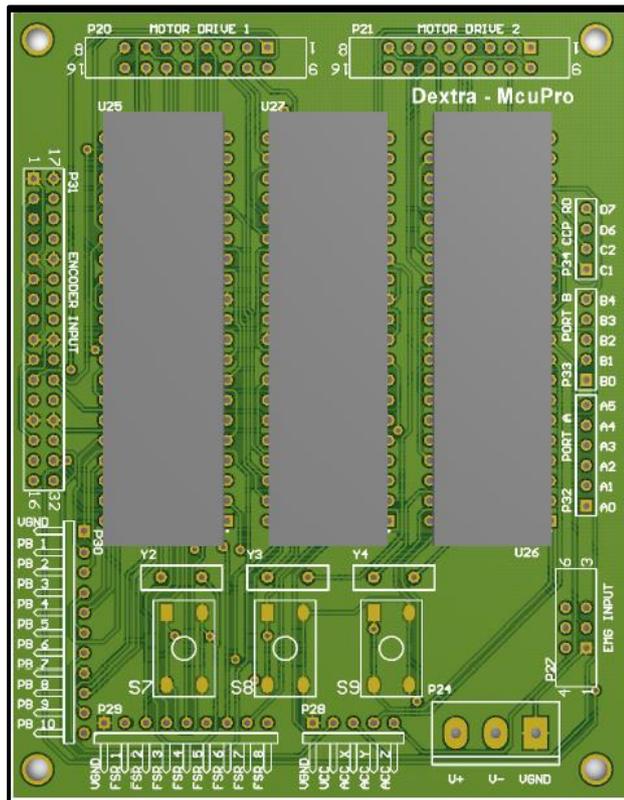


Figura A1. 18 Vista 3D cara superior MCUPro

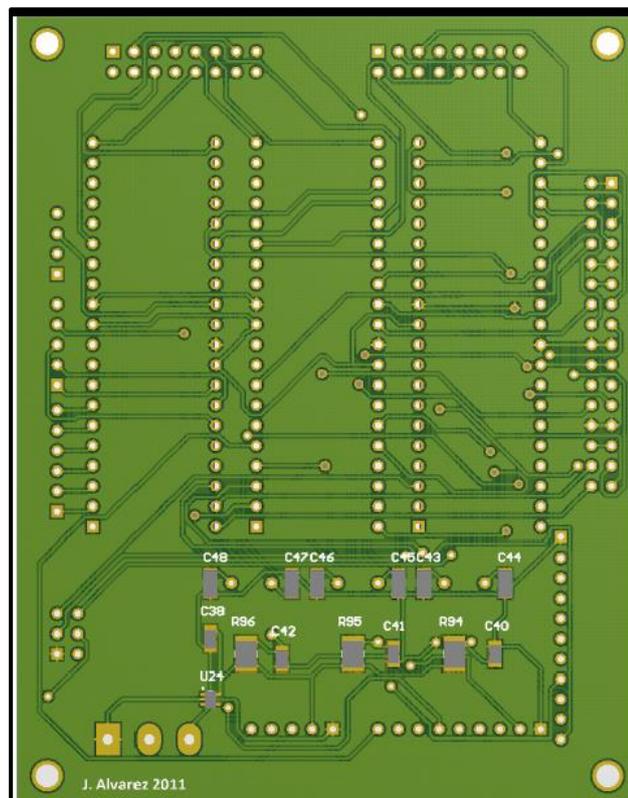


Figura A1. 19 Vista 3D cara inferior MCUPro

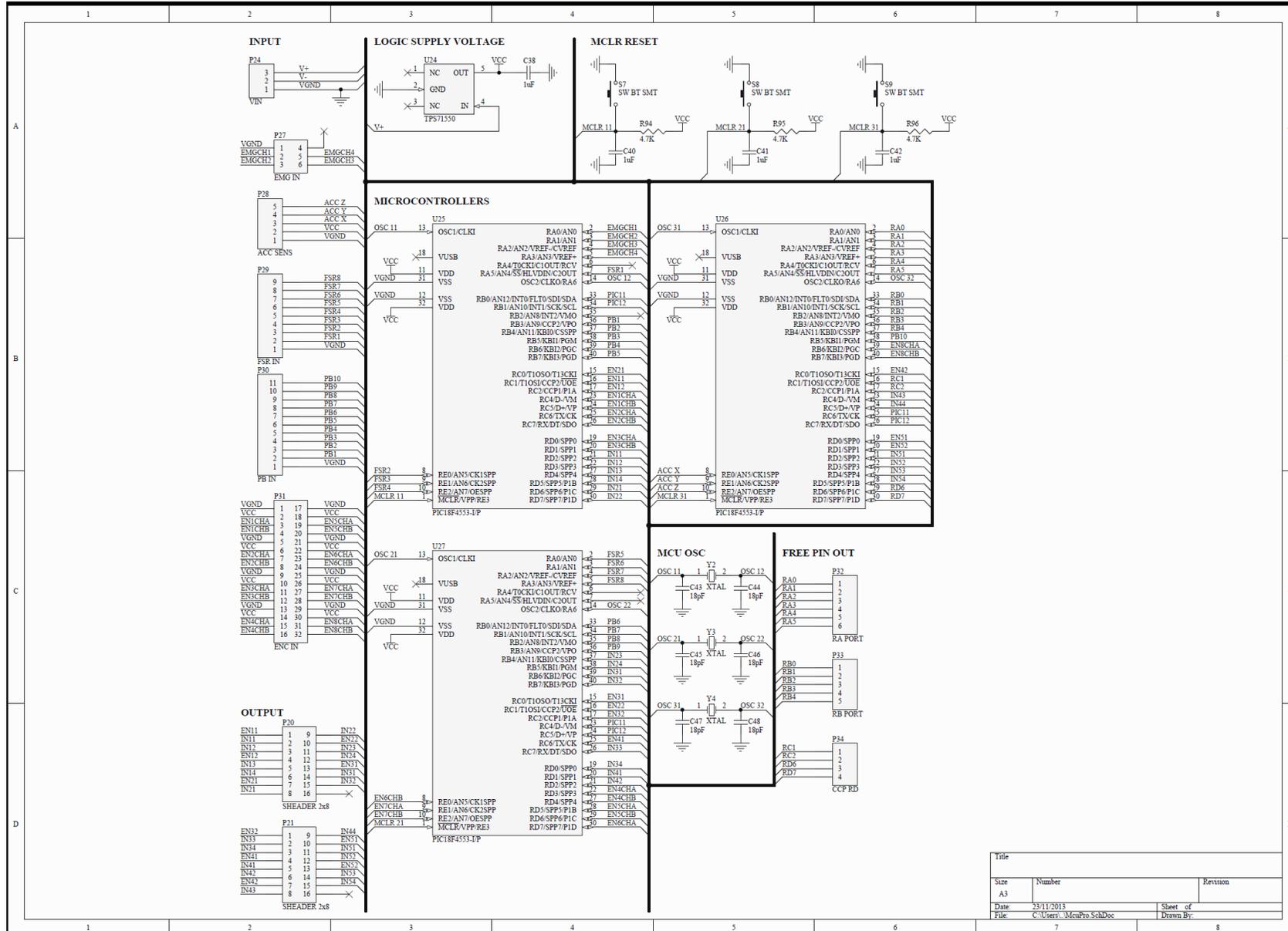


Figura A1. 20 Figura A1.20 Esquemático del circuito MCUPro

| Title | | |
|-------|------------------------|-----------|
| Size | Number | Revision |
| A3 | | |
| Date: | 23/11/2013 | Sheet of |
| File: | C:\Users\ManPro\SchDoc | Drawn By: |

ANEXO 2: FIRMWARE

Anexo 2.1: Dextra PIC1 (Adquisición señal EMG, procesamiento y control de dedos pulgar e índice) Firmware

```

/*
 * File: DextraPIC1_V0.c
 * Author: Julio Alvarez
 *
 * Program description: Programa para la adquisicion Y procesamiento de la señal EMG,
 * ademas del control de los dedos pulgar e indice.
 * Created on February 5, 2012, 4:26 AM
 */

#include <p18f4550.h> // PIC empleado
#include <stdio.h>
#include <stdlib.h>
#include <delays.h> // Libreria demoras
#include <adc.h> // Libreria conversor adc
#include <timers.h> // Libreria temporizadores

/*****
INITIAL CONFIGURATION
*****/

// PLL Prescaler Selection = 5, Divide by 5 (20 MHz oscillator input)
// System Clock Postscaler Selection = CPUDIV = OSC1_PLL2, Primary Oscillator Src: /1][96 MHz PLL Src: /2]
// USB Clock Selection bit = 1, USB clock source comes directly from the primary oscillator
// OSC = HSPLL_HS, 20 MHz (48MHz Oscilador interno)*/
#pragma config PLLDIV = 5, CPUDIV = OSC1_PLL2, USBDIV = 1, FOSC = HSPLL_HS

// Fail-Safe Clock Monitor = OFF
// Internal/External Oscillator Switchover = OFF
// Power-up Timer = ON */
#pragma config FCMEN = OFF, IESO = OFF, PWRT = ON

// Brown-out Reset = OFF
// Brown-out Reset Voltage bits = 0
// USB Voltage Regulator Enable = OFF

```

```

// Watchdog Timer Enable = OFF
// Watchdog Timer Postscale Select = 1, 1:1*/
#pragma config BOR = OFF, BORV = 0, VREGEN = OFF, WDT = OFF
// CCP2 MUX bit = OFF, CCP2 Multiplexed OFF
// PORTB A/D Enable = OFF, Pines PORTB 0 al 4 Configurados como Entradas Analogicas */
#pragma config CCP2MX = ON, PBADEN = OFF
// Low-Power Timer 1 Oscillator Enable = OFF
// MCLR Pin Enable = ON
// Stack Full/Underflow Reset Enable = ON
// Single-Supply ICSP Enable = OFF
// Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable = OFF */
#pragma config LPT1OSC = OFF, MCLRE = ON, STVREN = ON, LVP = OFF, ICPR1 = OFF

// Extended Instruction Set Enable = OFF
// Background Debugger Enable = ON */
#pragma config XINST = OFF, DEBUG = OFF

// Code Protection bit = OFF, CP0
// Code Protection bit = OFF, CP1
// Code Protection bit = OFF, CP2
// Code Protection bit = OFF, CP3
// Boot Block Code Protection bit = OFF */
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF, CPB = OFF

// Data EEPROM Code Protection = OFF
// Write Protection bit = OFF, BLOCK 0
// Write Protection bit = OFF, BLOCK 1
// Write Protection bit = OFF, BLOCK 2
// Write Protection bit = OFF, BLOCK 3 */
#pragma config CPD = OFF, WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF

// Configuration Register Write Protection bit = ON
// Boot Block Write Protection bit = OFF
// Data EEPROM Write Protection bit = OFF */
#pragma config WRTC = ON, WRTB = OFF, WRTD = OFF

// Table Read Protection = OFF, BLOCK 0
// Table Read Protection = OFF, BLOCK 1
// Table Read Protection = OFF, BLOCK 2
// Table Read Protection = OFF, BLOCK 3

```

```

// Boot Block Table Read Protection = OFF */
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF, EBTRB = OFF

/*****
PROGRAM
*****/

//BLOQUE DE DEFINICIONES

#define PB1 PORTBbits.RB3           //PB1 Y PB2 PULSANTES PARA MODO AGARRE
#define PB2 PORTBbits.RB4
#define PB3 PORTBbits.RB5           //PB3 PULSANTE TOPE PULGAR
#define PB4 PORTBbits.RB6           //PB4 PULSANTE TOPE MITAD INDICE
#define PB5 PORTBbits.RB7           //PB5 PULSANTE TOPE TOPE INDICE
#define EN11 LATCbits.LATC1         //EN11 ENABLE MOTOR PULGAR
#define IN11 LATDbits.LATD2         //IN11 SALIDA AL DRIVER PULGAR
#define IN12 LATDbits.LATD3         //IN12 SALIDA AL DRIVER PULGAR
#define EN12 LATCbits.LATC2         //EN12 ENABLE MOTOR INDICE MEDIO
#define IN13 LATDbits.LATD4         //IN13 SALIDA AL DRIVER INDICE MEDIO
#define IN14 LATDbits.LATD5         //IN14 SALIDA AL DRIVER INDICE MEDIO
#define EN21 LATCbits.LATC0         //EN21 ENABLE MOTOR INDICE ALTO
#define IN21 LATDbits.LATD6         //IN21 SALIDA AL DRIVER INDICE ALTO
#define IN22 LATDbits.LATD7         //IN22 SALIDA AL DRIVER INDICE ALTO
#define ENCAPUL PORTCbits.RC4       //ENCODER CANAL A PULGAR
#define ENCBPUL PORTCbits.RC5       //ENCODER CANAL B PULGAR
#define ENCAINDM PORTCbits.RC6      //ENCODER CANAL A INDICE MEDIO
#define ENCBINDM PORTCbits.RC7      //ENCODER CANAL B INDICE MEDIO
#define ENCAINDA PORTDbits.RD0      //ENCODER CANAL A INDICE ALTO
#define ENCBINDA PORTDbits.RD1      //ENCODER CANAL B INDICE ALTO

//PROTOTIPOS DE FUNCIONES
void PulgarAbrir (void);
void IndiceAbrir (void);
void CerrarPul (void);
void CerrarInda (void);
void CerrarIndm (void);

//VARIABLES
unsigned int emg1 = 0, dato1 = 0, emg2 = 0, dato2 = 0, emg3 = 0, dato3 = 0, emg4 = 0, dato4 = 0;
unsigned int fsr1 = 0, fsr2 = 0, fsr3 = 0, fsr4 = 0, fs1 = 0, fs2 = 0, fs3 = 0, fs4 = 0;
unsigned int encoderpul = 0, encoderinda = 0, encoderindm = 0, AnteriorPulA = 0, AnteriorIndaA = 0, AnteriorIndmA
= 0;

```

```

unsigned int flag = 0, flagswitch = 0, flaggrip = 0;
//valores emg donde no existe actividad
unsigned int emgmin1 = 5117; //2.53v
unsigned int emgmin2 = 5117;
unsigned int emgmin3 = 5117;
unsigned int emgmin4 = 5117;
//VALORES MAXIMOS FSR
unsigned int FsrMax1 = 3000; //1.7v
unsigned int FsrMax2 = 3000;
unsigned int FsrMax3 = 3000;
unsigned int FsrMax4 = 3000;

//VALORES MAXIMOS DE ENCODERS
unsigned int EncaPulMax = 60; //35 pulsos
unsigned int EncbPulMax = 60;
unsigned int EncaIndmMax = 60;
unsigned int EncbIndmMax = 60;
unsigned int EncaIndaMax = 50;
unsigned int EncbIndaMax = 60;

void main(void) {
    //CONFIGURACIONES
    //MODULO ADC
    ADCON1 = 0b00000000; //VREF 0-5V
    ADCON1bits.PCFG = 0b0111; //AN0:AN7 HABILITADOS COMO ENTRADAS ANALOGICAS
    ADCON2bits.ADFM = 1; //DATO JUSTIFICADO A LA DERECHA 10 bits
    ADCON2bits.ACQT = 0b111; //Tad = 2
    ADCON2bits.ADCS = 0b010; //Fosc = 32
    ADCON0bits.ADON = 1; // MODULO AD HABILITADO

    UCONbits.USBEN = 0;
    UCFGbits.UTRDIS = 1;

    INTCON2bits.RBPU = 0; // PULL UPS PUERTO B HABILITADOS
    //PUERTOS
    TRISA = 0b11111111; // RA0:RA6 CONFIGURADAS COMO ENTRADAS
    TRISB = 0b11111100; // RB2:RB7 CONFIGURADAS COMO ENTRADAS, RB0:RB1 SALIDAS

    TRISC = 0b11110000; // RC0:RC2 SALIDAS RC4:RC7 ENTRADAS RC3 NO IMPLEMENTADO

```

```

TRISD = 0b00000011;    // RD0:RD1 ENTRADAS RD2:RD7 SALIDAS
TRISE = 0b00000111;    // RE0:RE2 ENTRADAS

//CONFIGURACION TIMER
OpenTimer0(TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_2);
while (1){
    //ADC SEÑALES EMG
    emg1 = 0;
    flag = 0;
    encoderinda = 0;
    encoderindm = 0;
    encoderpul = 0;
    AnteriorIndaA = 0;
    AnteriorIndmA = 0;
    AnteriorPulA = 0;
    SetChanADC(ADC_CH0);                // Selecciono AN0 para convertir
    Delay10TCYx(5);
        ConvertADC();                    // Comienza conversión
        while(BusyADC()==1){ }          // Hasta que se finalice conversión
        dato1 = ReadADC();              // Realizo lectura
    emg1 = dato1 * 10;                  // Multiplica por 10 para precisión de un decimal
    SetChanADC(ADC_CH1);                // Selecciono AN1 para convertir
    Delay10TCYx(5);
        ConvertADC();                    // Comienza conversión
        while(BusyADC()==1){ }          // Hasta que se finalice conversión
        dato2 = ReadADC();              // Realizo lectura
    emg2 = dato2 * 10;                  // Multiplica por 10 para precisión de un decimal
    SetChanADC(ADC_CH2);                // Selecciono AN1 para convertir
    Delay10TCYx(5);
        ConvertADC();                    // Comienza conversión
        while(BusyADC()==1){ }          // Hasta que se finalice conversión
        dato3 = ReadADC();              // Realizo lectura
    emg3 = dato3 * 10;                  // Multiplica por 10 para precisión de un decimal
    SetChanADC(ADC_CH3);                // Selecciono AN1 para convertir
    Delay10TCYx(5);
        ConvertADC();                    // Comienza conversión
        while(BusyADC()==1){ }          // Hasta que se finalice conversión
        dato4 = ReadADC();              // Realizo lectura
    emg4 = dato4 * 10;                  // Multiplica por 10 para precisión de un decimal
//ADC FSRS

```

```

SetChanADC(ADC_CH4); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){ } // Hasta que se finalice conversión
    fs1 = ReadADC(); // Realizo lectura
fsr1 = fs1 * 10; // Multiplica por 10 para precisión de un decimal

SetChanADC(ADC_CH5); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){ } // Hasta que se finalice conversión
    fs2 = ReadADC(); // Realizo lectura
fsr2 = fs2 * 10; // Multiplica por 10 para precisión de un decimal
SetChanADC(ADC_CH6); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){ } // Hasta que se finalice conversión
    fs3 = ReadADC(); // Realizo lectura
fsr3 = fs3 * 10; // Multiplica por 10 para presicion de un decimal
SetChanADC(ADC_CH7); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){ } // Hasta que se finalice conversión
    fs4 = ReadADC(); // Realizo lectura
fsr4 = fs4 * 10; // Multiplica por 10 para precisión de un decimal

//DETERMINAR EL TIPO DE AGARRE DE LA MANO
if (PB1 == 0) {
    flagswitch = 1; //PULGAR POSICION GRIP
}
if (PB2 == 0) {
    flagswitch = 2; //PULGAR POSICION MANO ABIERTA
}
//SEÑAL EMG DENTRO DE RANGO?
if (emg1 < 4068 ){
    if(flagswitch == 1 && flag == 0){ //ESTA EN MODO GRIP?
        if (flaggrip == 0) { //ESTA CERRADA LA MANO?

            while (PB4 == 1) { // ABRIR INDICE HASTA QUE PULSANTE SE ACTIVE

```

```

    EN12 = 1;
    EN21 = 1;
    IndiceAbrir();
}
while (PB3 == 1) { // ABRIR PULGAR HASTA QUE PULSANTE SE ACTIVE
    EN11 = 1;
    PulgarAbrir();
}
LATBbits.LATB0 = 0; //SEÑALES PARA EL CONTROL DE OTROS
DEDOS PIC 2
LATBbits.LATB1 = 1;
EN11 = 0;
IN11 = 0;
IN12 = 0;
EN12 = 1; //MATENER ACTIVO PARA QUE NO CAIGA POR
EL PESO
IN13 = 0;
IN14 = 1;
while (flaggrip == 0){

    //DETERMINAR EL TIPO DE AGARRE DE LA MANO
    if (PB2 == 0) {
        flagswitch = 2; //PULGAR POSICION MANO ABIERTA
    }
    SetChanADC(ADC_CH0); // Selecciono AN0 para convertir
    Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    dato1 = ReadADC(); // Realizo lectura
    emg1 = dato1 * 10; // Multiplica por 10 para precisión de un decimal
    if (emg1 < 4068){
        flaggrip = 1;
    }
}
// VUELVO A CERO EL VALOR PARA OTRA LECTURA
}
if (flaggrip == 1){
    AnteriorPulA = ENCAPUL;
    AnteriorIndaA = ENCAINDA;
}

```

```

AnteriorIndmA = ENCAINDM;
LATBbits.LATB0 = 0; //SEÑALES PARA EL CONTROL DE
OTROS DEDOS PIC2
LATBbits.LATB1 = 0;
while (encoderindm < EncaIndmMax && fsr3 < FsrMax3){ //MOVIMIENTO 3
    EN12 = 1;
    CerrarIndm();
}
EN12 = 0;
IN13 = 0;
IN14 = 0;
while (encoderinda < EncaIndaMax && fsr4 < FsrMax4){ //CERRAR DEDOS
    EN21 = 1;
    CerrarInda();
}
EN21 = 0;
IN21 = 0;
IN22 = 0;
while (encoderpul < EncaPulMax && fsr2 < FsrMax2){ //CERRAR DEDOS
    EN11 = 1;
    CerrarPul();
}
EN11 = 0;
IN11 = 0;
IN12 = 0;
while (flaggrip == 1){
    SetChanADC(ADC_CH0); // Selecciono AN0 para convertir
    Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    dato1 = ReadADC(); // Realizo lectura
    emg1 = dato1 * 10; // Multiplica por 10 para presicion de
un decimal
    if (emg1 < 4068){
        flaggrip = 0;
        flag = 1; //SALE
    }
}

```

```

    }
}
Delay10KTCYx(50000);
Delay10KTCYx(50000);
Delay10KTCYx(50000);
Delay10KTCYx(50000);
if(flagswitch == 2 && flag == 0){
    if (flagrip == 0) {
        LATBbits.LATB0 = 1;
        LATBbits.LATB1 = 1;

        while (PB3 == 1) {
            EN11 = 1;
            PulgarAbrir();
        }
        while (PB4 == 1) {
            EN12 = 1;
            EN21 = 1;
            IndiceAbrir();
        }

        EN11 = 0;
        IN11 = 0;
        IN12 = 0;
        EN12 = 1;

        IN13 = 0;
        IN14 = 1;
        while (flagrip == 0){
            SetChanADC(ADC_CH0);
            Delay10TCYx(5);
            ConvertADC();
            while(BusyADC()==1){}
            dato1 = ReadADC();
            emg1 = dato1 * 10;
        }
    }
}

```

//ESTA EN MODO ABIERTO?

//ESTA CERRADA LA MANO?

//SEÑALES PARA EL CONTROL DE

OTROS DEDOS PIC 2

// ABRIR PULGAR HASTA QUE PULSANTE SE

ACTIVE

// ABRIR INDICE HASTA QUE PULSANTE SE

ACTIVE

//MATENER ACTIVO PARA QUE NO

CAIGA POR EL PESO

// Selecciono AN0 para convertir

// Comienza conversión

// Hasta que se finalice conversión

// Realizo lectura

// Multiplica por 10 para presicion de un

decimal

```

    if (emg1 < 4068){
        flaggrip = 1;
    }
}

// VUELVO A CERO EL VALOR PARA OTRA LECTURA
}

if (flaggrip == 1){
    AnteriorPulA = ENCAPUL;
    AnteriorIndaA = ENCAINDA;
    AnteriorIndmA = ENCAINDM;
    LATBbits.LATB0 = 1;                                     //SEÑALES PARA EL CONTROL DE
OTROS DEDOS PIC 2
    LATBbits.LATB1 = 0;
    while (encoderindm < EncaIndmMax && fsr3 < FsrMax3){    //MOVIMIENTO 3
        EN12 = 1;
        CerrarIndm();
    }

    EN12 = 0;
    IN13 = 0;
    IN14 = 0;

    while (encoderinda < EncaIndaMax && fsr4 < FsrMax4){    //CERRAR DEDOS
        EN21 = 1;
        CerrarInda();
    }
    EN21 = 0;
    IN21 = 0;
    IN22 = 0;

    while (encoderpul < EncaPulMax && fsr2 < FsrMax2){    //CERRAR DEDOS
        EN11 = 1;
        CerrarPul();
    }
    EN11 = 0;
    IN11 = 0;
    IN12 = 0;
    while (flaggrip == 1){
        SetChanADC(ADC_CH0);                               // Selecciono AN0 para convertir
        Delay10TCYx(5);
    }
}

```

```

ConvertADC(); // Comienza conversión
while(BusyADC()==1){ // Hasta que se finalice conversión
    dato1 = ReadADC(); // Realizo lectura
    emg1 = dato1 * 10; // Multiplica por 10 para presicion de
un decimal
    if (emg1 < 4068){
        flagrip = 0;
        flag = 1; //SALE
    }
}

}
Delay10KTCYx(50000);
Delay10KTCYx(50000);
Delay10KTCYx(50000);
Delay10KTCYx(50000);
}
}

while (!INTCONbits.TMR0IF);
INTCONbits.TMR0IF = 0;
}
return;
}

void PulgarAbrir (void){
    LATDbits.LATD2 = 0;
    Delay10KTCYx(50000);
    LATDbits.LATD3 = 1;
    Delay10KTCYx(50000);
    LATDbits.LATD2 = 0;
    Delay10KTCYx(50000);
    LATDbits.LATD3 = 0;
    Delay10KTCYx(50000);
}

void IndiceAbrir (void){
    while (PB5 == 1){
        LATDbits.LATD6 = 0;
        //Delay10KTCYx(50);

```

```

LATDbits.LATD7 = 1;
//Delay10KTCYx(50);
}
IN21 = 0;
IN22 = 0;
LATDbits.LATD4 = 0;
Delay10KTCYx(50);
LATDbits.LATD5 = 1;
Delay10KTCYx(50);
}
void CerrarPul(void){
    IN11 = 1;
    IN12 = 0;
    if (ENCAPUL == AnteriorPulA ){
        AnteriorPulA = ENCAPUL;

    }else{
        encoderpul = encoderpul + 1;
        AnteriorPulA = ENCAPUL;
    }
    SetChanADC(ADC_CH5);      // Selecciono AN1 para convertir
    Delay10TCYx(5);
    ConvertADC();              // Comienza conversión
    while(BusyADC()==1){}     // Hasta que se finalice conversión
    fs2 = ReadADC();           // Realizo lectura
    fsr2 = fs2 * 10;          // Multiplica por 10 para presicion de un decimal
}
void CerrarInda (void){
    IN21 = 1;
    IN22 = 0;
    if (ENCAINDA == AnteriorIndaA ){
        AnteriorIndaA = ENCAINDA;

    }else{
        encoderinda = encoderinda + 1;
        AnteriorIndaA = ENCAINDA;
    }
    SetChanADC(ADC_CH7);      // Selecciono AN1 para convertir
    Delay10TCYx(5);
    ConvertADC();              // Comienza conversión

```

```

while(BusyADC()==1){}           // Hasta que se finalice conversión
fs4 = ReadADC();                // Realizo lectura
fsr4 = fs4 * 10;                // Multiplica por 10 para presicion de un decimal
}
void CerrarIndm (void){
  IN13 = 1;
  IN14 = 0;
  if (ENCAINDM == AnteriorIndmA){
    AnteriorIndmA = ENCAINDM;
  }else {
    encoderindm = encoderindm + 1;
    AnteriorIndmA = ENCAINDM;
  }
  SetChanADC(ADC_CH6);         // Selecciono AN1 para convertir
  Delay10TCYx(5);
  ConvertADC();                // Comienza conversión
  while(BusyADC()==1){}       // Hasta que se finalice conversión
  fs3 = ReadADC();             // Realizo lectura
  fsr3 = fs3 * 10;            // Multiplica por 10 para presicion de un decimal
}

```

Anexo 2.2: Dextra PIC2 (control dedos medio y anular) Firmware

```

/*
 * File: DextraPIC1_V0.c
 * Author: Julio Alvarez
 *
 * Program description: programa para el control de los dedos anular y medio en base a la
 * entrada digital que proporcion el PIC1 en los puertos RC4 Y RC5
 *
 * Created on February 5, 2012, 4:26 AM
 */

#include <p18f4550.h> // PIC empleado
#include <stdio.h>
#include <stdlib.h>
#include <delays.h> // Libreria demoras
#include <adc.h> // Libreria conversor adc
#include <timers.h> // Libreria temporizadores

```

```
#include <portb.h>
```

```
/******
```

```
INITIAL CONFIGURATION
```

```
*****
```

```
// PLL Prescaler Selection = 5, Divide by 5 (20 MHz oscillator input)
```

```
// System Clock Postscaler Selection = CPUDIV = OSC1_PLL2, Primary Oscillator Src: /1][96 MHz PLL Src: /2]
```

```
// USB Clock Selection bit = 1, USB clock source comes directly from the primary oscillator
```

```
// OSC = HSPLL_HS, 20 MHz (48MHz Oscilador interno)*/
```

```
#pragma config PLLDIV = 5, CPUDIV = OSC1_PLL2, USBDIV = 1, FOSC = HSPLL_HS
```

```
// Fail-Safe Clock Monitor = OFF
```

```
// Internal/External Oscillator Switchover = OFF
```

```
// Power-up Timer = ON */
```

```
#pragma config FCMEN = OFF, IESO = OFF, PWRT = OFF
```

```
// Brown-out Reset = OFF
```

```
// Brown-out Reset Voltage bits = 0
```

```
// USB Voltage Regulator Enable = OFF
```

```
// Watchdog Timer Enable = OFF
```

```
// Watchdog Timer Postscale Select = 1, 1:1*/
```

```
#pragma config BOR = OFF, BORV = 0, VREGEN = OFF, WDT = OFF
```

```
// CCP2 MUX bit = OFF, CCP2 Multiplexed OFF
```

```
// PORTB A/D Enable = OFF, Pines PORTB 0 al 4 Configurados como Entradas Analogicas */
```

```
#pragma config CCP2MX = OFF, PBADEN = OFF
```

```
// Low-Power Timer 1 Oscillator Enable = OFF
```

```
// MCLR Pin Enable = ON
```

```
// Stack Full/Underflow Reset Enable = ON
```

```
// Single-Supply ICSP Enable = OFF
```

```
// Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable = OFF */
```

```
#pragma config LPT1OSC = OFF, MCLRE = ON, STVREN = ON, LVP = OFF, ICPRT = OFF
```

```
// Extended Instruction Set Enable = OFF
```

```
// Background Debugger Enable = ON */
```

```
#pragma config XINST = OFF, DEBUG = OFF
```

```
// Code Protection bit = OFF, CP0
```

```
// Code Protection bit = OFF, CP1
```

```

// Code Protection bit = OFF, CP2
// Code Protection bit = OFF, CP3
// Boot Block Code Protection bit = OFF */
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF, CPB = OFF

// Data EEPROM Code Protection = OFF
// Write Protection bit = OFF, BLOCK 0
// Write Protection bit = OFF, BLOCK 1
// Write Protection bit = OFF, BLOCK 2
// Write Protection bit = OFF, BLOCK 3 */
#pragma config CPD = OFF, WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF

// Configuration Register Write Protection bit = ON
// Boot Block Write Protection bit = OFF
// Data EEPROM Write Protection bit = OFF */
#pragma config WRTC = OFF, WRTB = OFF, WRTD = OFF

// Table Read Protection = OFF, BLOCK 0
// Table Read Protection = OFF, BLOCK 1
// Table Read Protection = OFF, BLOCK 2
// Table Read Protection = OFF, BLOCK 3
// Boot Block Table Read Protection = OFF */
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF, EBTRB = OFF

/*****
PROGRAM
*****/

//BLOQUE DE DEFINICIONES
#define PB6 PORTBbits.RB0 //PB1 Y PB2 PULSANTES PARA MODO AGARRE
#define PB7 PORTBbits.RB1
#define PB8 PORTBbits.RB2 //PB3 PULSANTE TOPE PULGAR
#define PB9 PORTBbits.RB3 //PB4 PULSANTE TOPE MITAD INDICE

#define PIC11 PORTCbits.RC4 //BIT DE CONTROL ON OFF DEDO MEDIO
#define PIC12 PORTCbits.RC5 //BIT DE CONTROL ON OFF DEDO ANULAR

#define IN23 LATBbits.LATB4 //IN23 SALIDA DRIVER MEDIO
#define IN24 LATBbits.LATB5 //IN24 SALIDA DRIVER MEDIO
#define IN31 LATBbits.LATB6 //IN31 SALIDA DRIVER MEDIO
#define IN32 LATBbits.LATB7 //IN32 SALIDA DRIVER MEDIO

```

```

#define EN31 LATCbits.LATC0 //EN31 ENABLE MOTOR MEDIO
#define EN22 LATCbits.LATC1 //EN22 ENABLE MOTOR MEDIO
#define EN32 LATCbits.LATC2 //EN32 ENABLE MOTOR ANULAR
#define EN41 LATCbits.LATC6 //EN41 ENABLE MOTOR ANULAR
#define IN33 LATCbits.LATC7 //IN33 SALIDA AL DRIVER MOTOR ANULAR

#define IN34 LATDbits.LATD0 //IN34 SALIDA AL DRIVER MOTOR ANULAR
#define IN41 LATDbits.LATD1 //IN41 SALIDA AL DRIVER MOTOR ANULAR
#define IN42 LATDbits.LATD2 //IN42 SALIDA AL DRIVER MOTOR ANULAR

#define ENCBMEDM PORTDbits.RD3 //ENCODER CANAL A MEDIO MEDIO
#define ENCBMEDM PORTDbits.RD4 //ENCODER CANAL B MEDIO MEDIO
#define ENCBMEDA PORTDbits.RD5 //ENCODER CANAL A MEDIO ALTO
#define ENCBMEDA PORTDbits.RD6 //ENCODER CANAL B MEDIO ALTO
#define ENCAANUM PORTDbits.RD7 //ENCODER CANAL A ANULAR MEDIO

#define ENCBANUM PORTEbits.RE0 //ENCODER CANAL B ANULAR MEDIO
#define ENCBANUM PORTEbits.RE1 //ENCODER CANAL B ANULAR ALTO
#define ENCBANUM PORTEbits.RE2 //ENCODER CANAL B ANULAR ALTO

//PROTOTIPOS DE FUNCIONES
void MedioAbrir (void);
void AnularAbrir (void);
void CerrarMedm (void);
void CerrarMeda (void);
void CerrarAnum (void);
void CerrarAnua (void);

//VARIABLES
unsigned int fsr5 = 0, fsr6 = 0, fsr7 = 0, fsr8 = 0, fs5 = 0, fs6 = 0, fs7 = 0, fs8 = 0;
unsigned int encodermeda = 0, encodermedm = 0, encoderanum = 0, encoderanua = 0, AnteriorMedA = 0,
AnteriorMedM = 0, AnteriorAnuA = 0, AnteriorAnuM = 0;

//Valores maximos FSR
unsigned int FsrMax5 = 3000; //1.7v
unsigned int FsrMax6 = 3000;
unsigned int FsrMax7 = 3000;
unsigned int FsrMax8 = 3000;
//Valores maximos de encoders

```

```

unsigned int EncaMedaMax = 60; //35 pulsos
unsigned int EncbMedaMax = 60;
unsigned int EncaMedmMax = 60;
unsigned int EncbMedmMax = 60;
unsigned int EncaAnumMax = 60;
unsigned int EncbAnumMax = 60;
unsigned int EncaAnuaMax = 60;
unsigned int EncbAnuaMax = 60;

void main(void) {

    //CONFIGURACIONES
    //MODULO ADC
    //ADCON1 = 0b00000000; //VREF 0-5V
    ADCON1bits.PCFG = 0b1011; //AN0:AN3 HABILITADOS COMO ENTRADAS ANALOGICAS
    ADCON2bits.ADFM = 1; //DATO JUSTIFICADO A LA DERECHA 10 bits
    ADCON2bits.ACQT = 0b010; //Tad = 2
    ADCON2bits.ADCS = 0b010; //Fosc = 32
    ADCON0bits.ADON = 1; // MODULO AD HABILITADO
    ADCON1bits.VCFG0 = 0;
    ADCON1bits.VCFG1 = 0;

    SSPCON1bits.SSPEN = 0;
    UCONbits.USBEN = 0;
    UCFGbits.UTRDIS = 1;

    INTCON2bits.RBPU = 0; // PULL UPS PUERTO B HABILITADOS
    //PUERTOS
    TRISA = 0b11111111; // RA0:RA7 CONFIGURADAS COMO ENTRADAS
    TRISB = 0b00001111; // RB0:RB3 CONFIGURADAS COMO ENTRADAS, RB4:RB7 SALIDAS

    TRISC = 0b00110000; // RC0:RC2 SALIDAS, RC4:RC5 ENTRADAS, RC6:RC7 SALIDAS, RC3 NO
IMPLEMENTADO
    TRISD = 0b11111000; // RD0:RD2 SALIDAS, RD3:RD7 ENTRADAS
    TRISE = 0b00000111; // RE0:RE2 ENTRADAS

    //CONFIGURACION TIMER
    OpenTimer0(TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_2);

    while (1){

```

```

//ADC FSRS
SetChanADC(ADC_CH0); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    fs5 = ReadADC(); // Realizo lectura
fsr5 = fs5 * 10; // Multiplica por 10 para presicion de un decimal
SetChanADC(ADC_CH1); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    fs6 = ReadADC(); // Realizo lectura
fsr6 = fs6 * 10; // Multiplica por 10 para presicion de un decimal
SetChanADC(ADC_CH2); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    fs7 = ReadADC(); // Realizo lectura
fsr7 = fs7 * 10; // Multiplica por 10 para presicion de un decimal
SetChanADC(ADC_CH3); // Selecciono AN1 para convertir
Delay10TCYx(5);
    ConvertADC(); // Comienza conversión
    while(BusyADC()==1){} // Hasta que se finalice conversión
    fs8 = ReadADC(); // Realizo lectura
fsr8 = fs8 * 10; // Multiplica por 10 para presicion de un decimal

if (PIC11==0 && PIC12==1 ){
    //ABRIR DEDO MEDIO Y ANULAR
    while (PB6 == 1) {
        EN22 = 1;
        EN31 = 1;
        MedioAbrir();
    }
    IN23 = 1;
    IN24 = 0;
    IN31 = 0;
    IN32 = 0;
    while (PB8 == 1) {
        EN32 = 1;

```

```
    EN41 = 1;
    AnularAbrir();
}
IN33 = 0;
IN34 = 0;
IN41 = 0;
IN42 = 0;
}
if (PIC11==1 && PIC12==0 ){

    while (PB8 == 1) {
        EN32 = 1;
        EN41 = 1;
        AnularAbrir();
    }
    IN33 = 0;
    IN34 = 0;
    IN41 = 0;
    IN42 = 0;
}
if (PIC11==0 && PIC12==0 ){
    //CERRAR MEDIO
    EN22 = 1;
    EN31 = 1;
    CerrarMedm();
    CerrarMedm();
    //CERRAR ANULAR
    EN32 = 1;
    EN41 = 1;
    CerrarAnum();
    CerrarAnua();

}
if (PIC11==1 && PIC12==1 ){
    //CERRAR MEDIO
    EN22 = 1;
    EN31 = 1;
    CerrarMedm();
    CerrarMedm();
```

```

    }

}

return;
}

void MedioAbrir (void){
    while (PB7 == 1){
        LATBbits.LATB6 = 0;
        //Delay10KTCYx(50);
        LATBbits.LATB7 = 1;
        //Delay10KTCYx(100);
    }
    IN31 = 0;
    IN32 = 0;
    LATBbits.LATB4 = 0;
    Delay10KTCYx(50);
    LATBbits.LATB5 = 1;
    Delay10KTCYx(50);
}

void AnularAbrir (void){
    while (PB9 == 1){
        LATDbits.LATD1 = 0;
        //Delay10KTCYx(50);
        LATDbits.LATD2 = 1;
        //Delay10KTCYx(100);
    }
    IN41 = 0;
    IN42 = 0;
    LATCbits.LATC7 = 0;
    Delay10KTCYx(50);
    LATDbits.LATD0 = 1;
    Delay10KTCYx(50);
}

void CerrarMeda (void){
    IN31 = 1;
    IN32 = 0;
    if (ENCAMEDA == AnteriorMedA ){
        AnteriorMedA = ENCAMEDA;

    }else{

```

```

    encodermeda = encodermeda + 1;
    AnteriorMedA = ENCAMEDA;
}
SetChanADC(ADC_CH1);    // Selecciono AN1 para convertir
Delay10TCYx(5);
ConvertADC();           // Comienza conversión
while(BusyADC()==1){}  // Hasta que se finalice conversión
fs6 = ReadADC();       // Realizo lectura
fsr6 = fs6 * 10;       // Multiplica por 10 para presicion de un decimal
}
void CerrarMedm (void){
    IN23 = 1;
    IN24 = 0;
    if (ENCAMEDM == AnteriorMedM ){
        AnteriorMedM = ENCAMEDM;

    }else{
        encodermedm = encodermedm + 1;
        AnteriorMedM = ENCAMEDM;
    }
    SetChanADC(ADC_CH1);    // Selecciono AN1 para convertir
    Delay10TCYx(5);
    ConvertADC();           // Comienza conversión
    while(BusyADC()==1){}  // Hasta que se finalice conversión
    fs5 = ReadADC();       // Realizo lectura
    fsr5 = fs5 * 10;       // Multiplica por 10 para presicion de un decimal
}
void CerrarAnua (void){
    IN41 = 1;
    IN42 = 0;
    if (ENCAANUA == AnteriorAnuA ){
        AnteriorAnuA = ENCAANUA;

    }else{
        encoderanua = encoderanua + 1;
        AnteriorAnuA = ENCAANUA;
    }
    SetChanADC(ADC_CH3);    // Selecciono AN1 para convertir
    Delay10TCYx(5);
    ConvertADC();           // Comienza conversión

```

```

while(BusyADC()==1){          // Hasta que se finalice conversión
fs8 = ReadADC();             // Realizo lectura
fsr8 = fs8 * 10;            // Multiplica por 10 para presicion de un decimal
}
void CerrarAnum (void){
    IN33 = 1;
    IN34 = 0;
    if (ENCAANUM == AnteriorAnuM ){
        AnteriorAnuM = ENCAANUM;

    }else{
        encoderanum = encoderanum + 1;
        AnteriorAnuM = ENCAANUM;
    }
    SetChanADC(ADC_CH2);      // Selecciono AN1 para convertir
    Delay10TCYx(5);
    ConvertADC();              // Comienza conversión
    while(BusyADC()==1){      // Hasta que se finalice conversión
    fs7 = ReadADC();          // Realizo lectura
    fsr7 = fs7 * 10;         // Multiplica por 10 para presicion de un decimal
    }
}

```

Anexo 2.3: Dextra PIC3 (control de muñeca) Firmware

```

/*
* File: DextraPIC3_V0.c
* Author: Julio Alvarez D.
*
* Program description: Lee entradas analogicas de los puertos RE0 al RE2 y en base
* a la entrada recibida del acelerometro, genera una salida digital para el control del motor de
* pasos en los puertos RBA4 al RB7, RC0 al RC7, RD0 al RD2, el motor detiene dependiendo de los
* valores generado por el acelerometro
*
* Created on February 5, 2012, 4:26 AM
*/
#include <p18f4550.h> // PIC empleado
#include <stdio.h>
#include <stdlib.h>
#include <delays.h> // Libreria demoras

```

```

#include <adc.h> // Libreria conversor adc
#include <timers.h> // Libreria temporizadores
/*****
INITIAL CONFIGURATION
*****/
// PLL Prescaler Selection = 5, Divide by 5 (20 MHz oscillator input)
// System Clock Postscaler Selection = CPUDIV = OSC1_PLL2, Primary Oscillator Src: /1][96 MHz PLL Src: /2]
// USB Clock Selection bit = 1, USB clock source comes directly from the primary oscillator
// OSC = HSPLL_HS, 20 MHz */
#pragma config PLLDIV = 5, CPUDIV = OSC1_PLL2, USBDIV = 1, FOSC = HSPLL_HS

// Fail-Safe Clock Monitor = OFF
// Internal/External Oscillator Switchover = OFF
// Power-up Timer = ON */
#pragma config FCMEN = OFF, IESO = OFF, PWRT = ON

// Brown-out Reset = OFF
// Brown-out Reset Voltage bits = 0
// USB Voltage Regulator Enable = OFF
// Watchdog Timer Enable = OFF
// Watchdog Timer Postscale Select = 1, 1:1*/

#pragma config BOR = OFF, BORV = 0, VREGEN = OFF, WDT = OFF

// CCP2 MUX bit = OFF, CCP2 Multiplexed OFF
// PORTB A/D Enable = OFF, Pines PORTB 4 al 0 Configurados como Digital I/O */
#pragma config CCP2MX = OFF, PBADEN = OFF

// Low-Power Timer 1 Oscillator Enable = OFF
// MCLR Pin Enable = ON
// Stack Full/Underflow Reset Enable = ON
// Single-Supply ICSP Enable = OFF
// Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable = OFF */
#pragma config LPT1OSC = OFF, MCLRE = ON, STVREN = ON, LVP = OFF, ICPR1 = OFF

// Extended Instruction Set Enable = OFF
// Background Debugger Enable = ON */
#pragma config XINST = OFF, DEBUG = OFF

// Code Protection bit = OFF, CP0

```

```

// Code Protection bit = OFF, CP1
// Code Protection bit = OFF, CP2
// Code Protection bit = OFF, CP3
// Boot Block Code Protection bit = OFF */
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF, CPB = OFF

// Data EEPROM Code Protection = OFF
// Write Protection bit = OFF, BLOCK 0
// Write Protection bit = OFF, BLOCK 1
// Write Protection bit = OFF, BLOCK 2
// Write Protection bit = OFF, BLOCK 3 */
#pragma config CPD = OFF, WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF

// Configuration Register Write Protection bit = ON
// Boot Block Write Protection bit = OFF
// Data EEPROM Write Protection bit = OFF */
#pragma config WRTC = OFF, WRTB = OFF, WRTD = OFF

// Table Read Protection = OFF, BLOCK 0
// Table Read Protection = OFF, BLOCK 1
// Table Read Protection = OFF, BLOCK 2
// Table Read Protection = OFF, BLOCK 3
// Boot Block Table Read Protection = OFF */
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF, EBTRB = OFF

/*****
PROGRAM
*****/
//BLOQUE DE DEFINICIONES

#define STPA LATDbits.LATD2 // BOBINA A DEL STEEPER
#define STPB LATDbits.LATD3 // BOBINA B DEL STEEPER
#define STPC LATDbits.LATD4 // BOBINA C DEL STEEPER
#define STPD LATDbits.LATD5 // BOBINA D DEL STEEPER

//PROTOTIPOS DE FUNCIONES
void palmaup (void);
void palmadown (void);
//VARIABLES
unsigned int AcZ = 0, dato1 = 0; // Variables para adquirir dato acelerometro

```

```

unsigned char delaystp = 1;          // Variable para el tiempo entre pasos max 1255
int step,step1;                     // Variable para el conteo de pasos
int maxstep = 24;                   // Variable de cantidad maxima de pasos
int palmflag = 0;                   // Variable indicadora de posicion palma

void main(void) {
//CONFIGURACION ADC
  ADCON1 = 0b00000000;              //VREF 0-5V
  ADCON1bits.PCFG = 0b0111;         //AN0:AN7 HABILITADOS COMO ENTRADAS ANALOGICAS
  ADCON2bits.ADFM = 1;               //DATO JUSTIFICADO A LA DERECHA
  ADCON2bits.ACQT = 0b010;          //Tad = 2
  ADCON2bits.ADCS = 0b010;          //Fosc = 32
  ADCON0bits.ADON = 1;              // MODULO AD HABILITADO

//CONFIGURAR PUERTOS DE ENTRADA Y SALIDA
  INTCON2bits.RBPU = 0;              // PULL UPS PUERTO B HABILITADOS
  TRISB = 0b11111111;               // PUERTO B TODAS ENTRADAS
  TRISC = 0b11000000;               // RC0:RC5 SALIDAS Y RC6:RD7 ENTRADAS
  TRISD = 0b11000000;               //RD0:RD5 SALIDAS Y RD6:RD7 ENTRADAS

//VALORES INICIALES DE SALIDAS
  LATD = 0b00000000;                //PUERTO D EN CERO

//INICIALIZACION DE VARIABLES
  step = 0;
  step1 = 0;

//CONFIGURACION TIMER
  OpenTimer0(TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_32);

  while (1){
    SetChanADC(ADC_CH7);             // Selecciono AN7 para convertir PIN RE2
    Delay10TCYx(5);
    ConvertADC();                     // Comienza conversi3n
    while(BusyADC()==1){}            // Hasta que se finalice conversi3n
    dato1 = ReadADC();                // Realizo lectura
    AcZ = dato1 * 10;                 // Multiplica por 10 para presicion de un decimal

    if (AcZ < 2864){ //MENOR QUE 1.40 V?
      //PALMA HACIA ABAJO

```

```

LATDbits.LATD0 = 1;    //ENABLES DEL STEPPER EN 1
LATDbits.LATD1 = 1;
palmadown();
}
if (AcZ > 3703){ //ES MAYOR QUE 1.81 V??
    //PALMA HACIA ARRIBA
    LATDbits.LATD0 = 1;    //ENABLES DEL STEPPER EN 1
    LATDbits.LATD1 = 1;
    palmaup();
}
if (AcZ >=2864 && AcZ <=3703){ //ESTA ENTRE 1.40 Y 1.81 V?
    //DETENER LA MARCHA
    LATDbits.LATD0 = 0;    //ENABLES DEL STEPPER EN 0
    LATDbits.LATD1 = 0;
    STPA = 0;
    STPB = 0;
    STPC = 0;
    STPD = 0;
}

while (!INTCONbits.TMR0IF);
INTCONbits.TMR0IF = 0;

}
return;
}

void palmaup (void){
    STPA = 0;
    STPB = 1;
    Delay10KTCYx(delaystp);
    STPC = 0;
    STPD = 1;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
    STPA = 1;
    STPB = 0;
    Delay10KTCYx(delaystp);
    STPC = 0;
    STPD = 1;
    Delay10KTCYx(delaystp);
}

```

```
    step1 = step1 + 1;
    STPA = 1;
    STPB = 0;
    Delay10KTCYx(delaystp);
    STPC = 1;
    STPD = 0;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
    STPA = 0;
    STPB = 1;
    Delay10KTCYx(delaystp);
    STPC = 1;
    STPD = 0;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
}
void palmadown (void){
    STPA = 1;
    STPB = 0;
    Delay10KTCYx(delaystp);
    STPC = 1;
    STPD = 0;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
    STPA = 1;
    STPB = 0;
    Delay10KTCYx(delaystp);
    STPC = 0;
    STPD = 1;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
    STPA = 0;
    STPB = 1;
    Delay10KTCYx(delaystp);
    STPC = 0;
    STPD = 1;
    Delay10KTCYx(delaystp);
    step1 = step1 + 1;
    STPA = 0;
    STPB = 1;
```

```
Delay10KTCYx(delaystp);  
STPC = 1;  
STPD = 0;  
Delay10KTCYx(delaystp);  
step1 = step1 + 1;  
}
```

ANEXO 3: PLANOS DE LA MANO ROBÓTICA

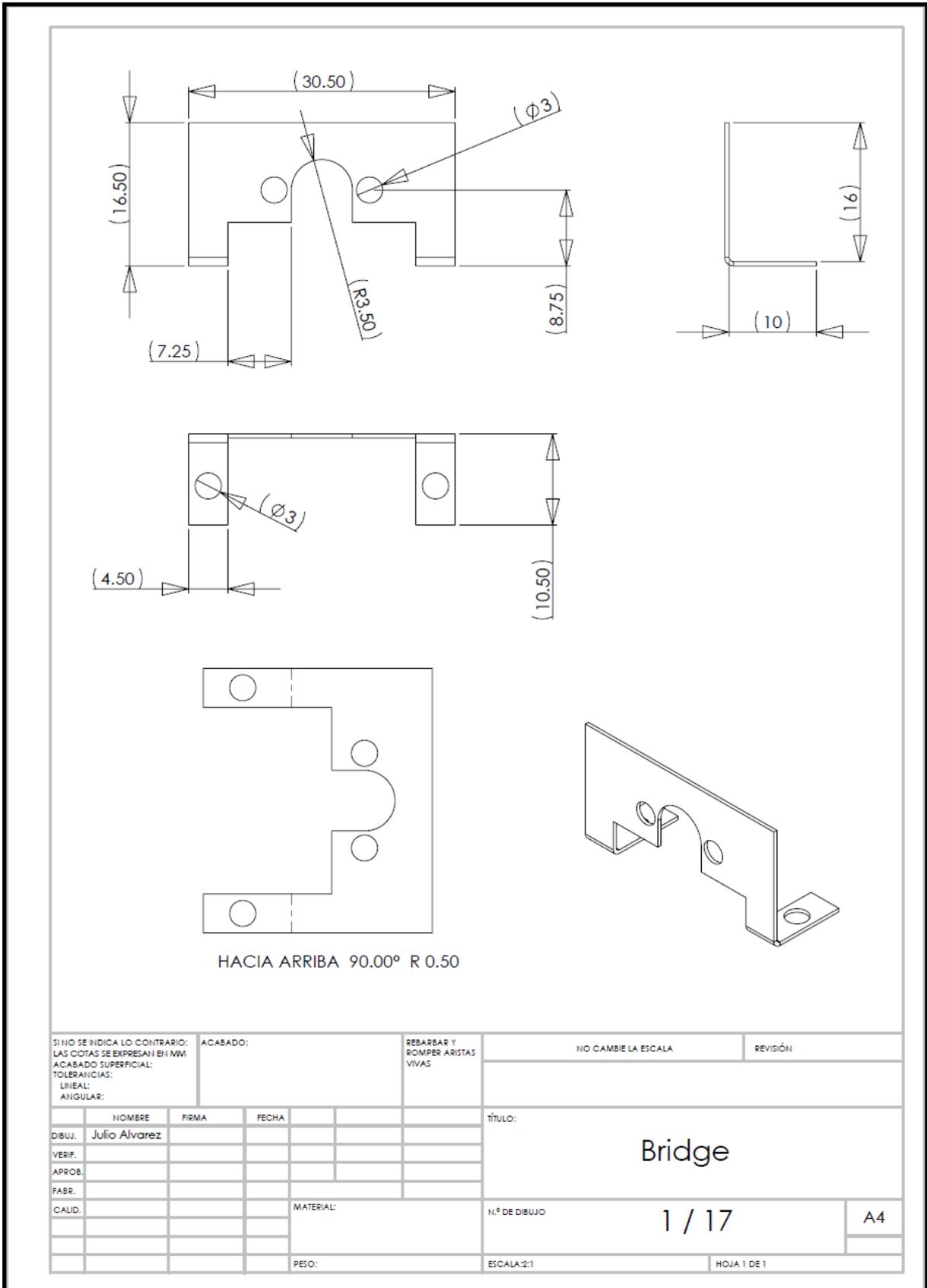


Figura A3. 1 Plano pieza Bridge

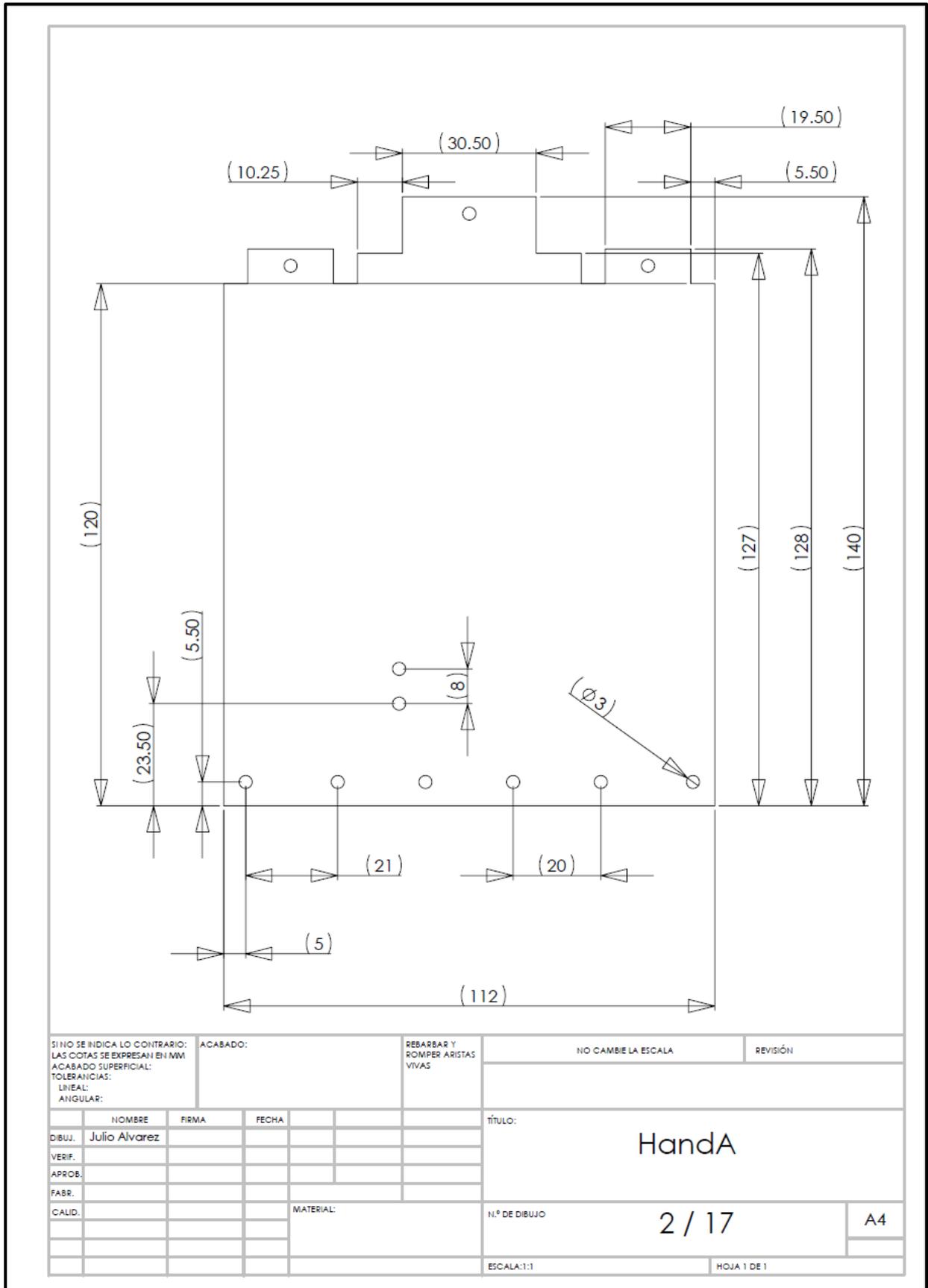


Figura A3. 2 Plano pieza Hand A

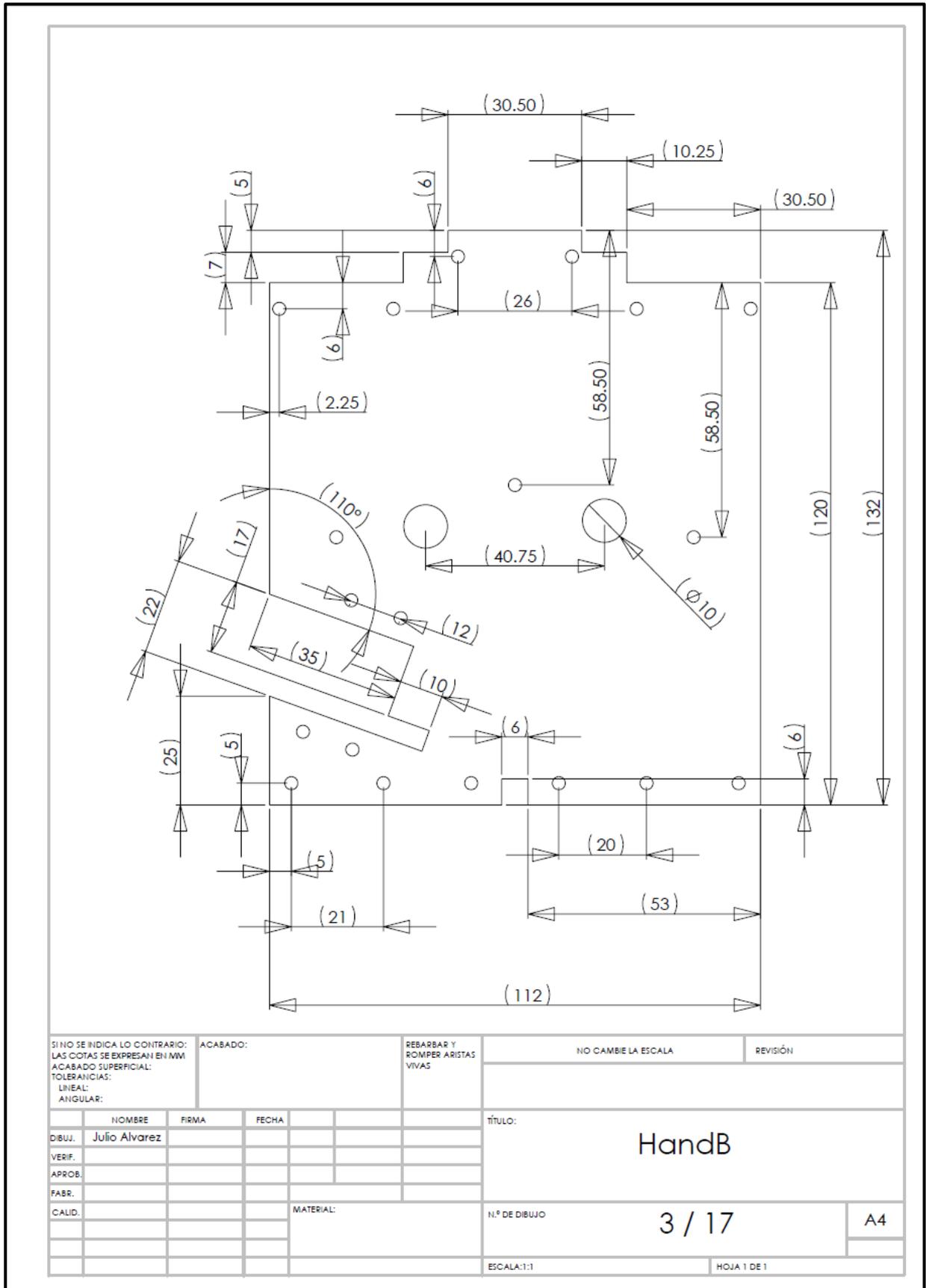


Figura A3. 3 Plano pieza Hand B

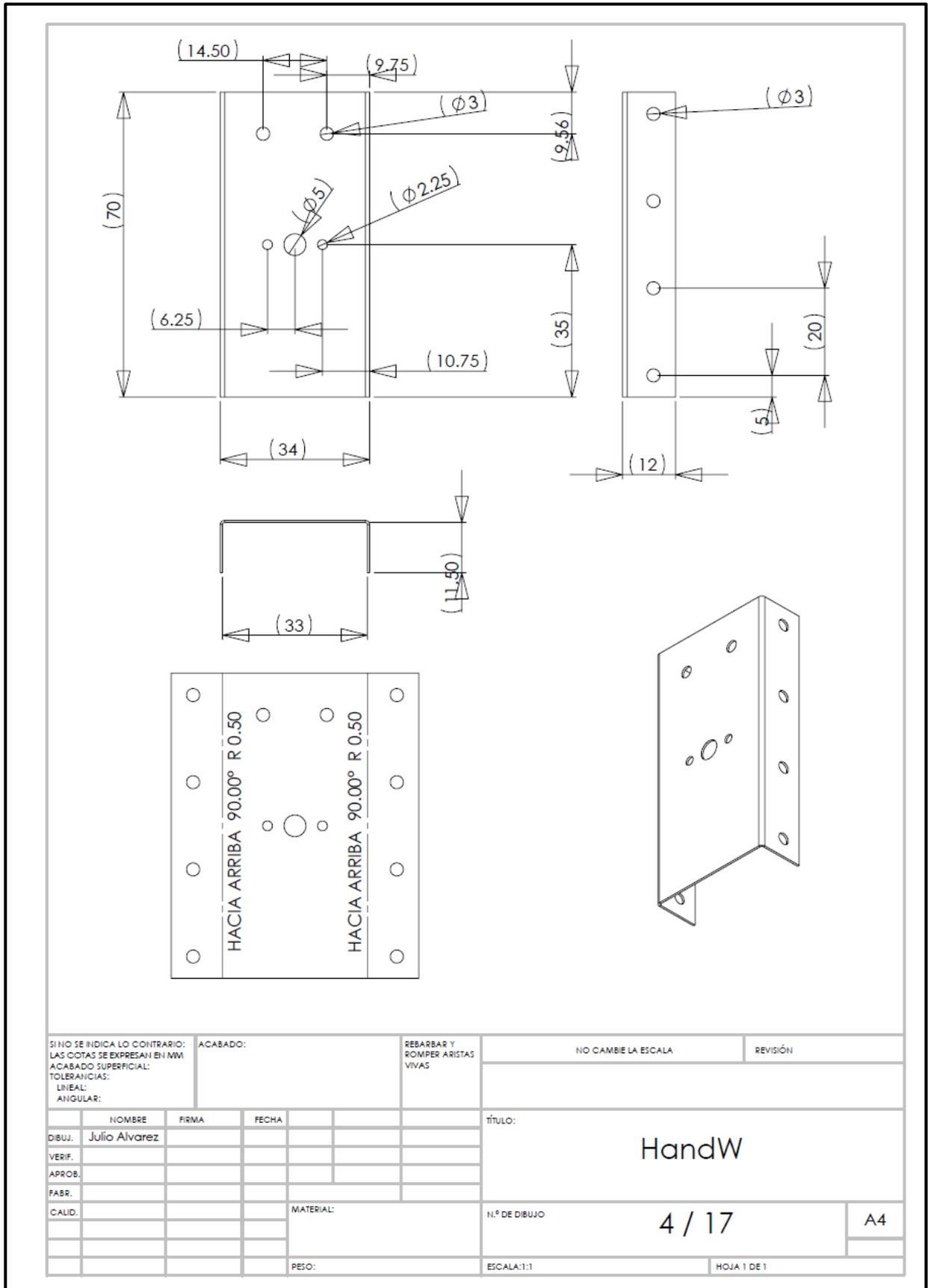


Figura A3. 4 Plano pieza Hand W

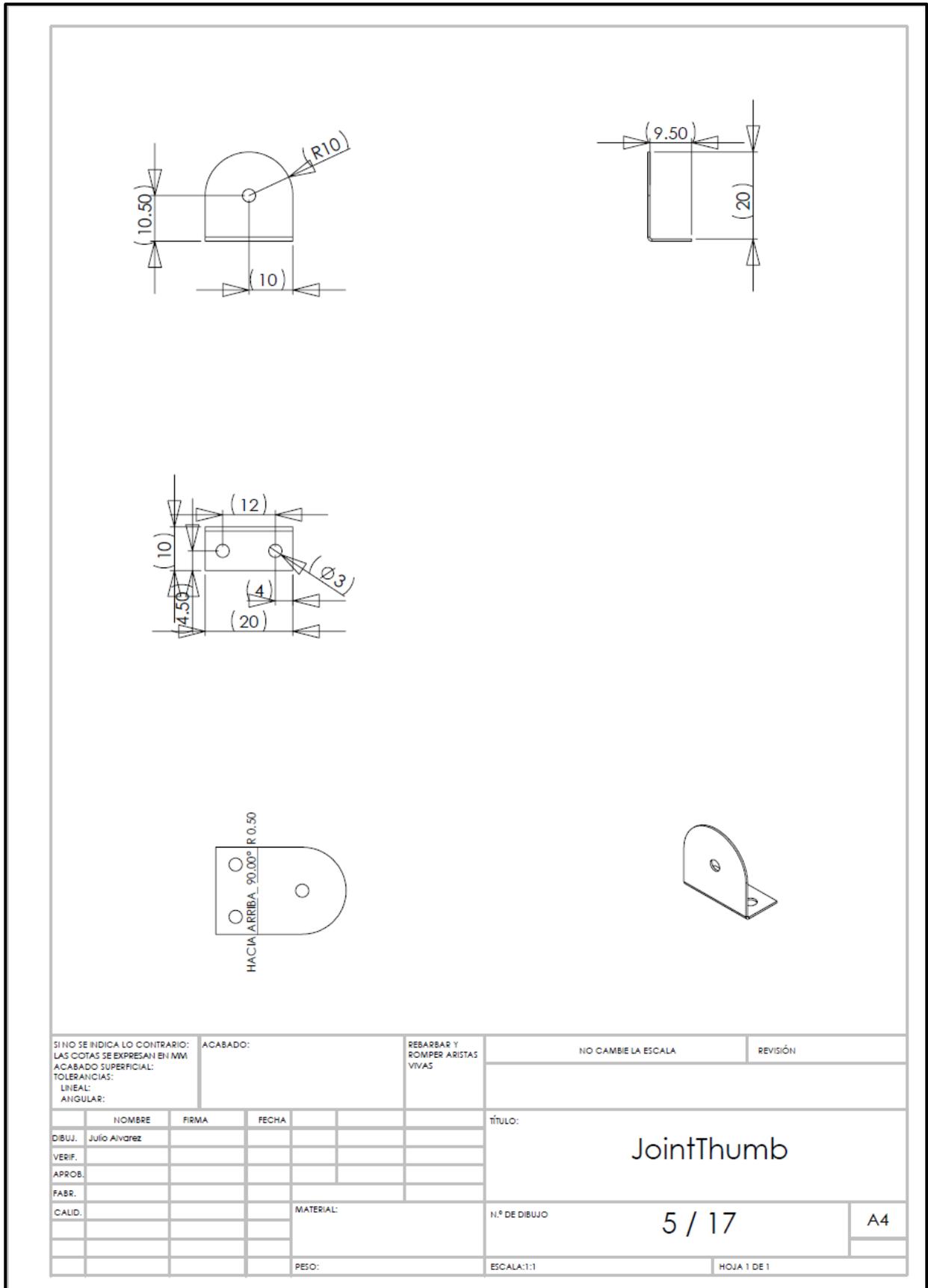


Figura A3. 5 Plano pieza JointThumb

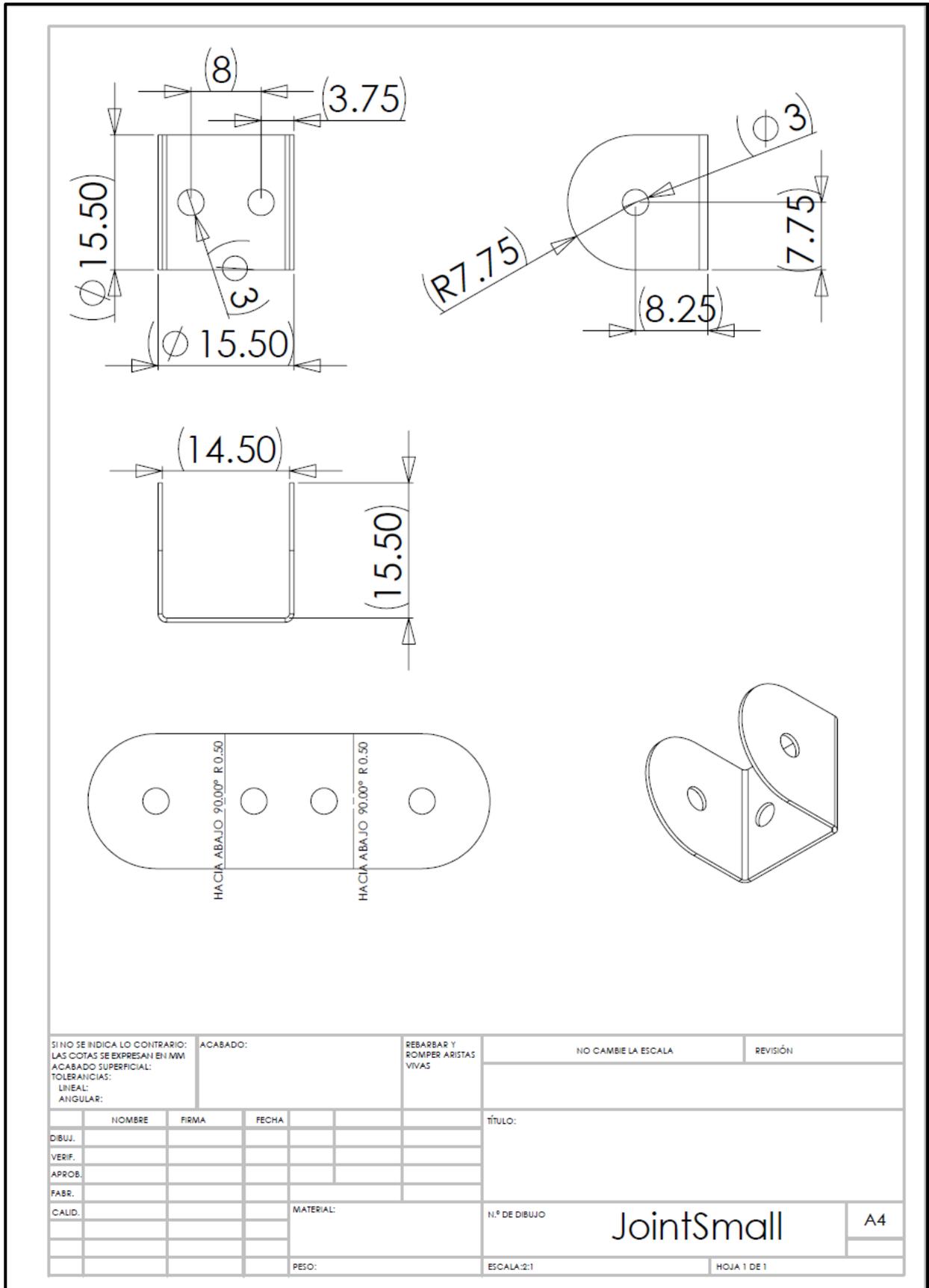


Figura A3. 6 Plano pieza JointSmall

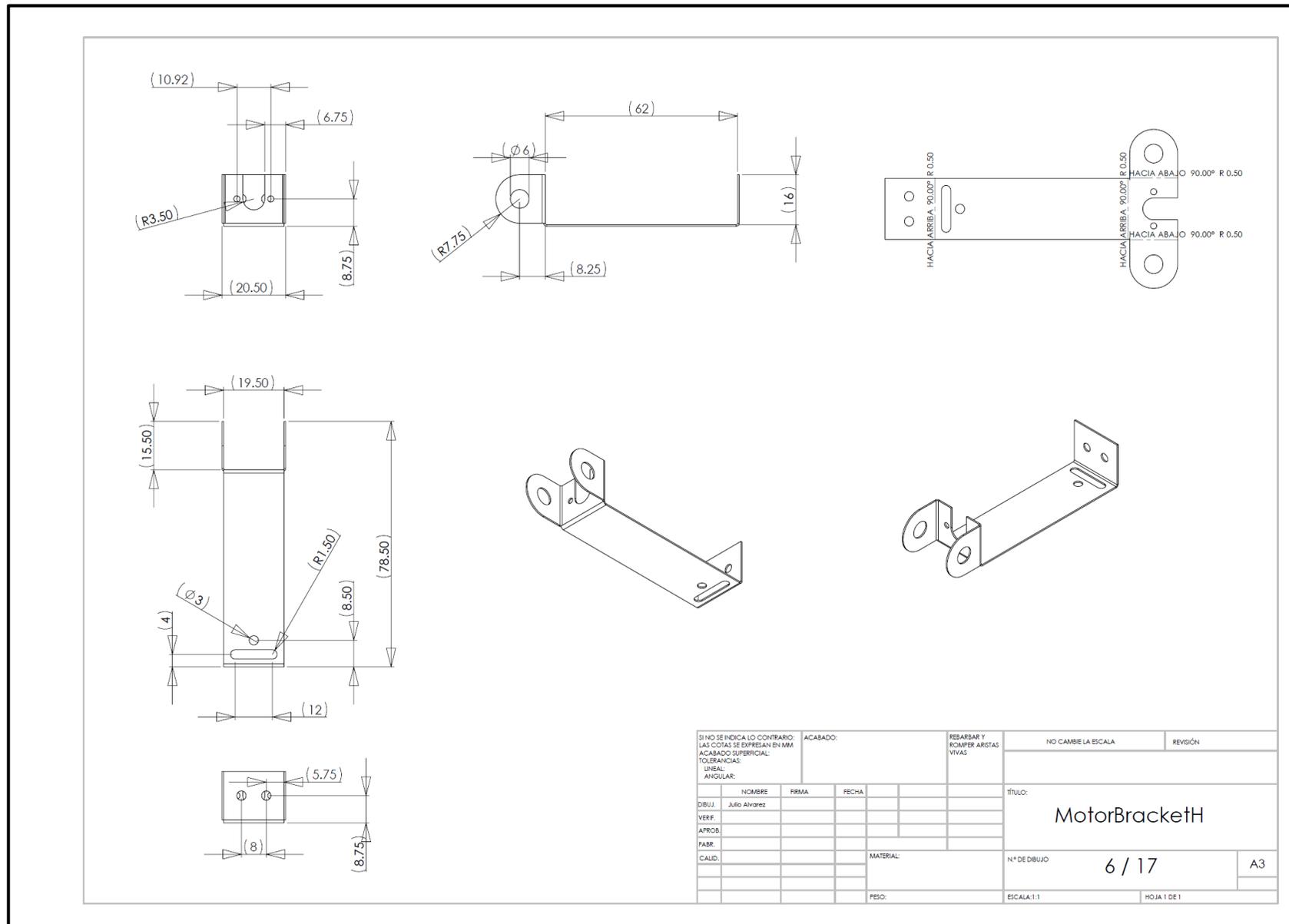


Figura A3. 7 Plano pieza MotorBracketH

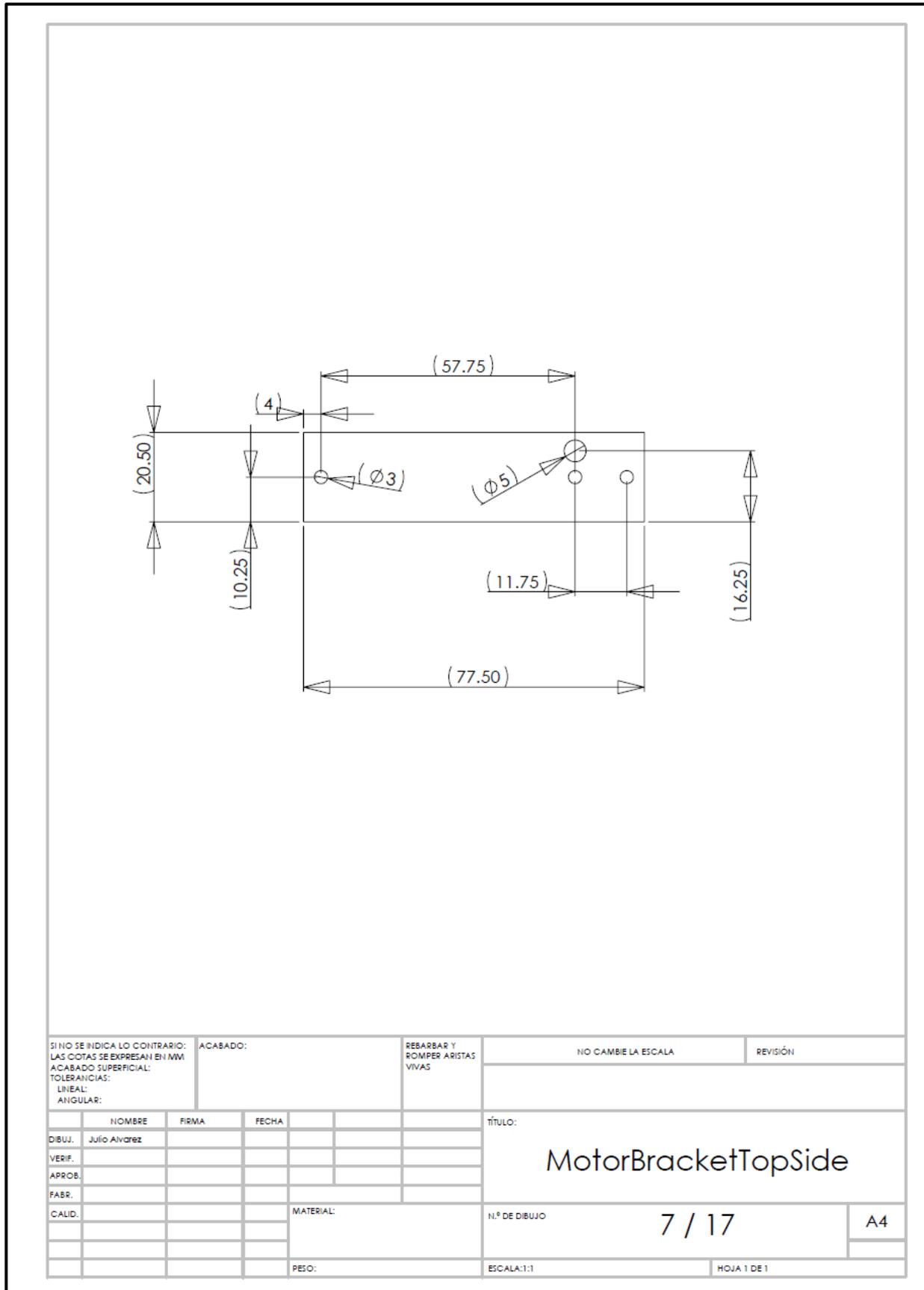


Figura A3. 8 Plano pieza Motor Bracket Top Side

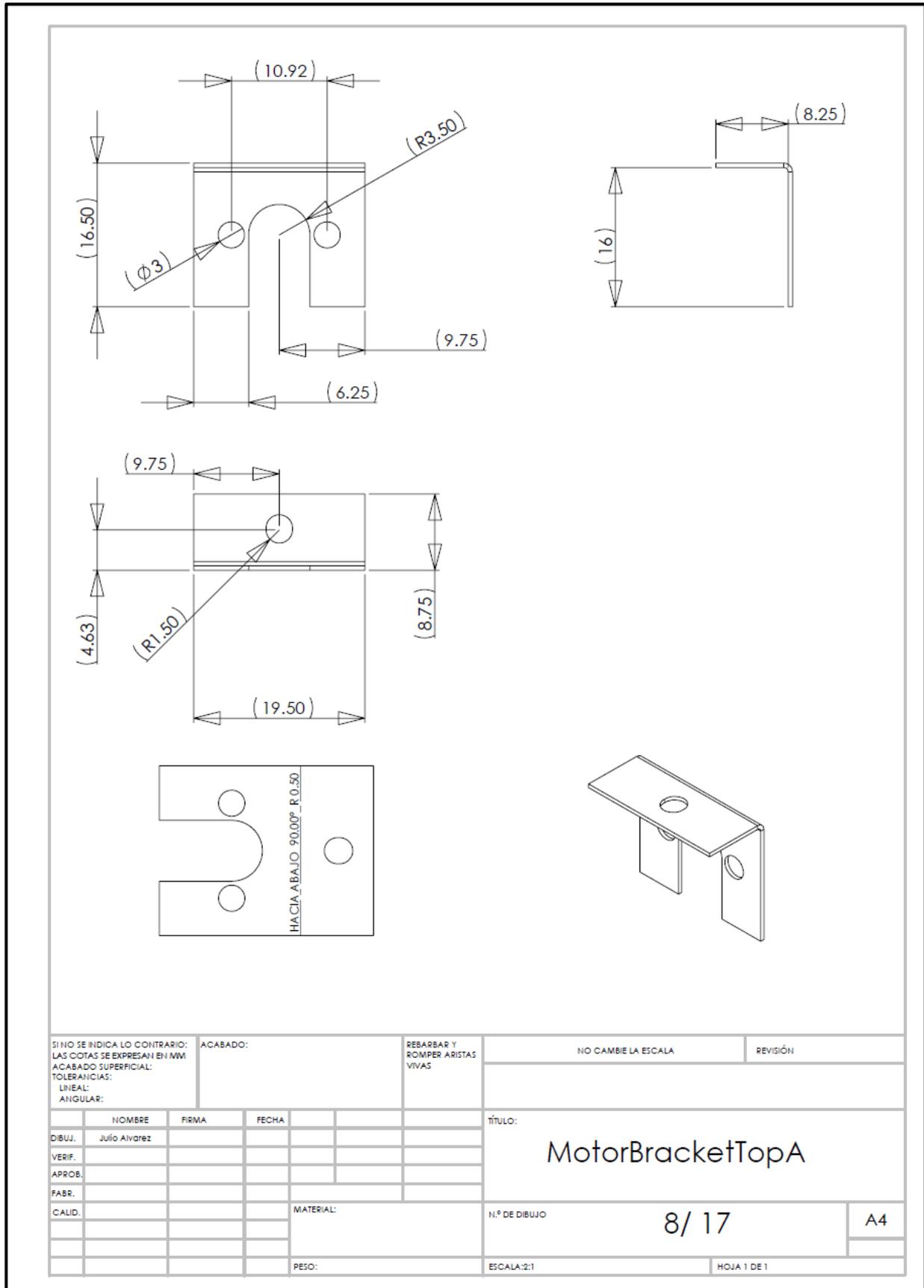


Figura A3. 9 Plano pieza Motor Bracket Top A

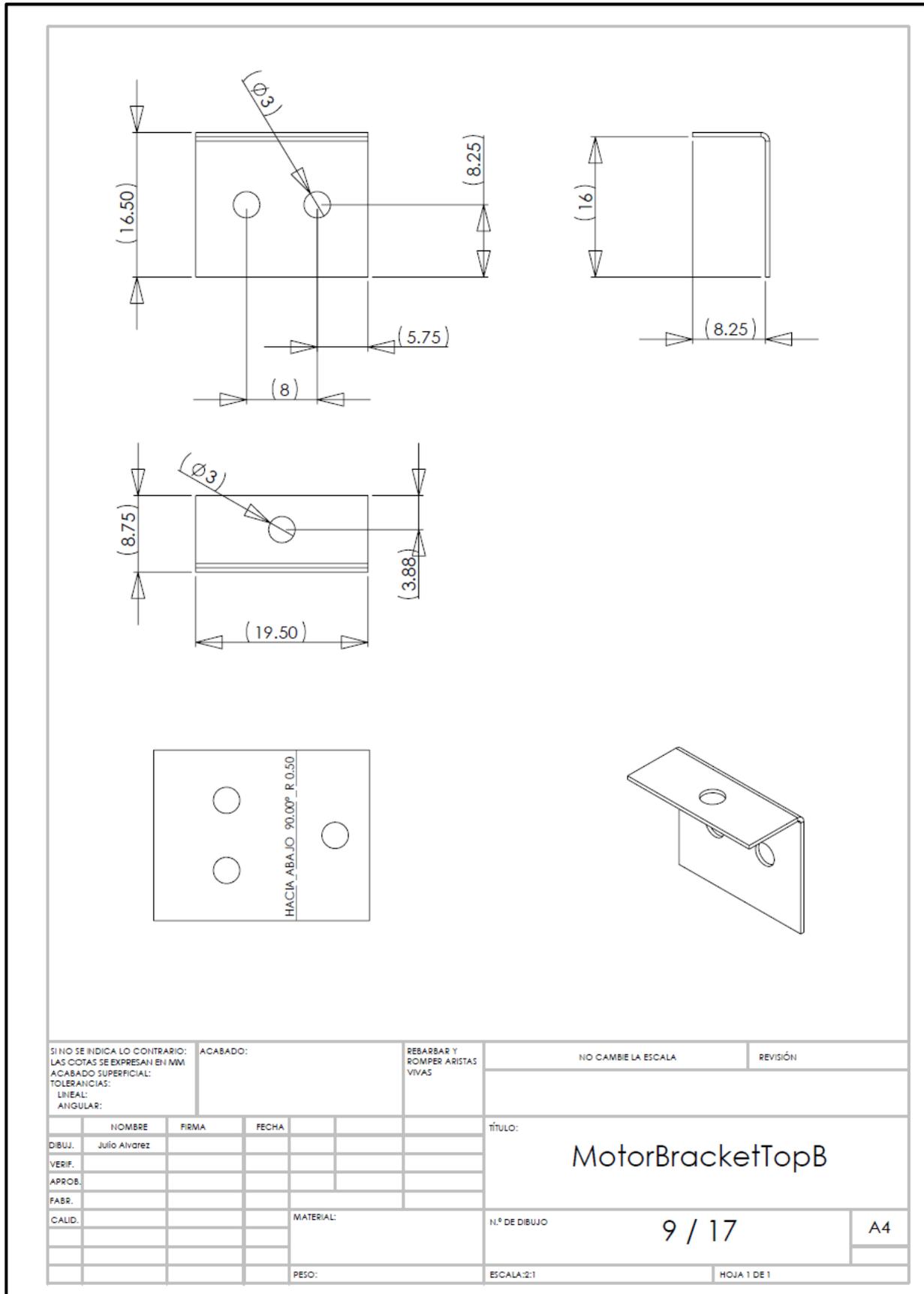


Figura A3. 10 Plano pieza Motor Bracket Top B

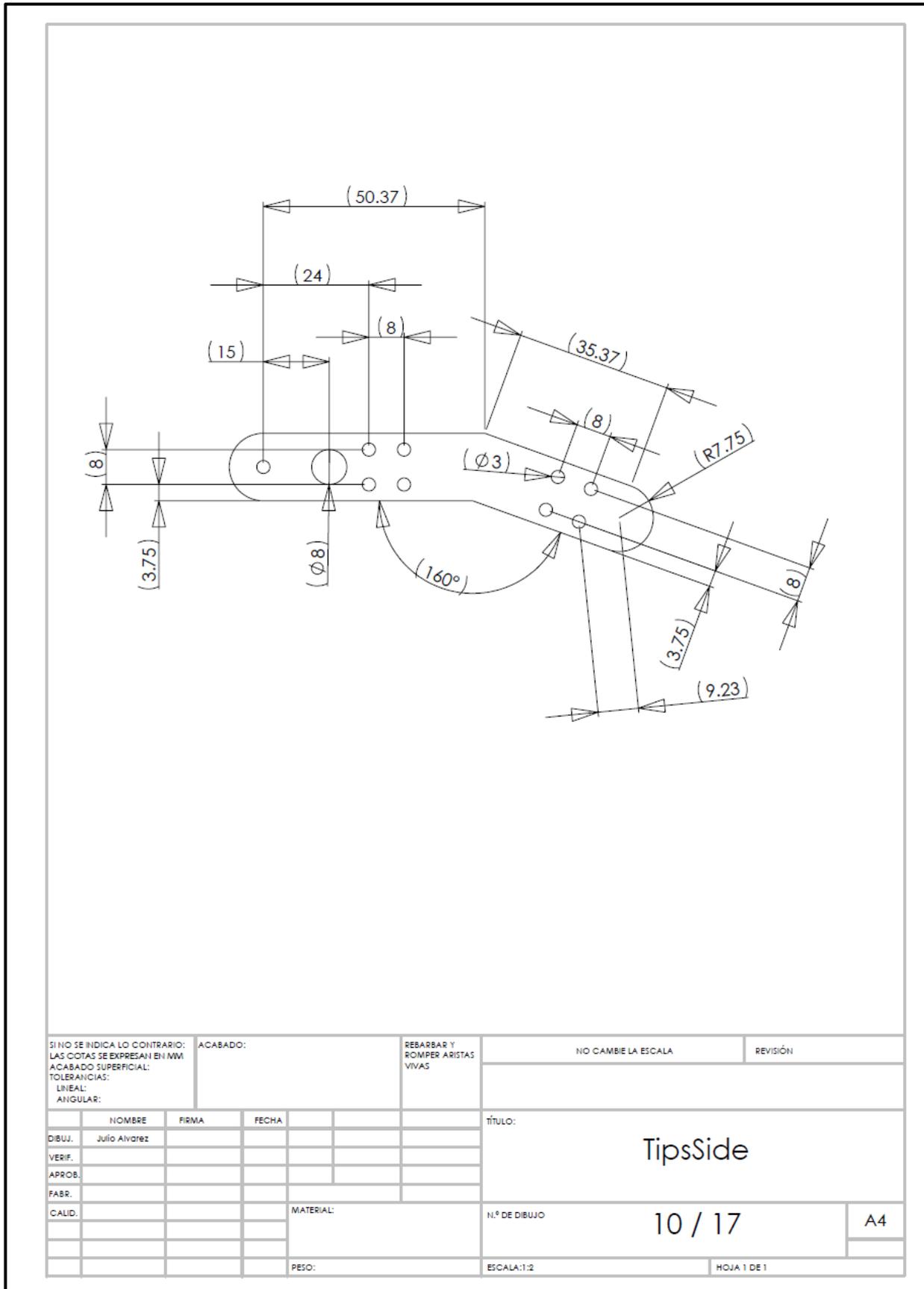


Figura A3. 11 Plano pieza Tips Side

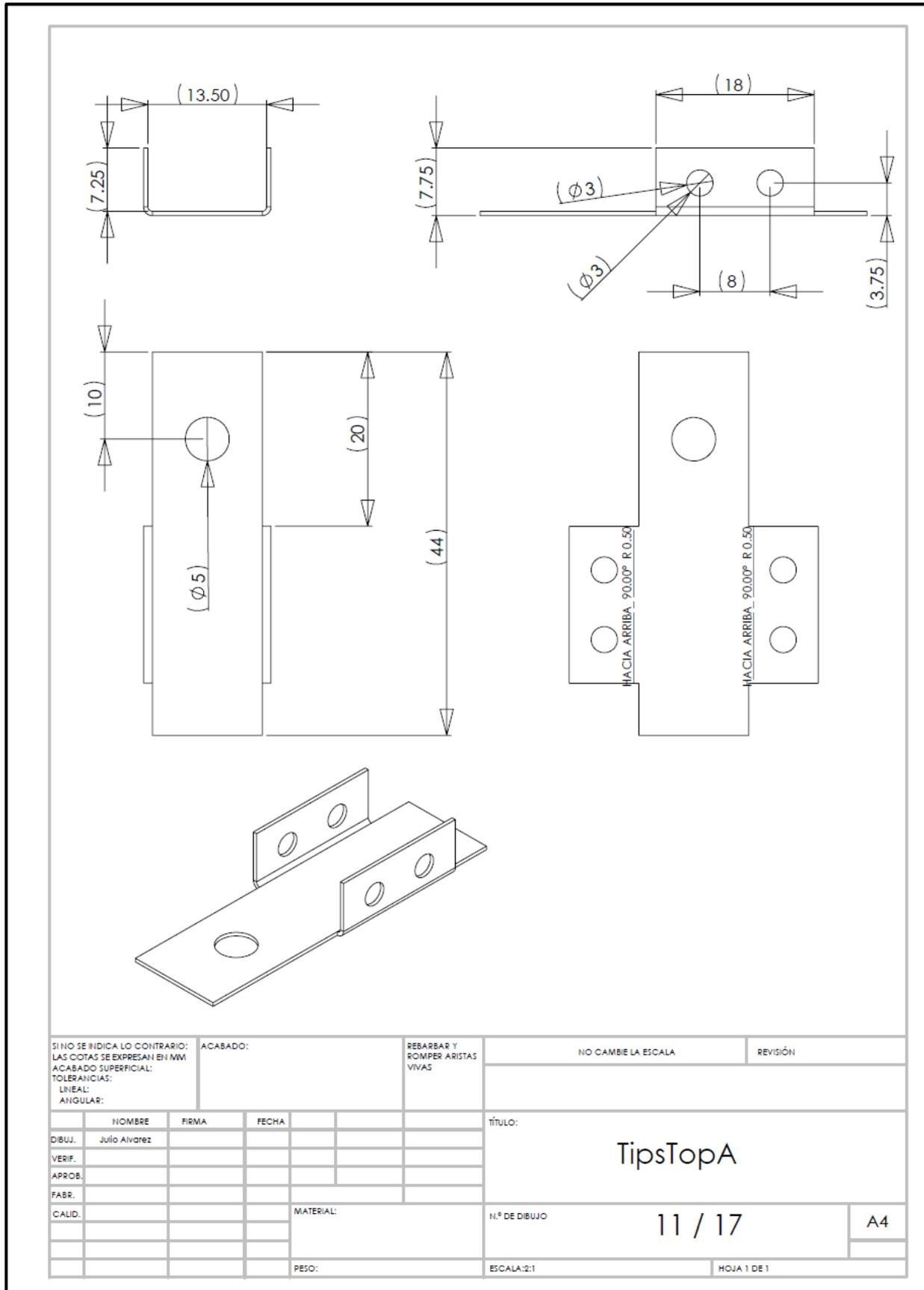


Figura A3. 12 Plano pieza TipsTop A

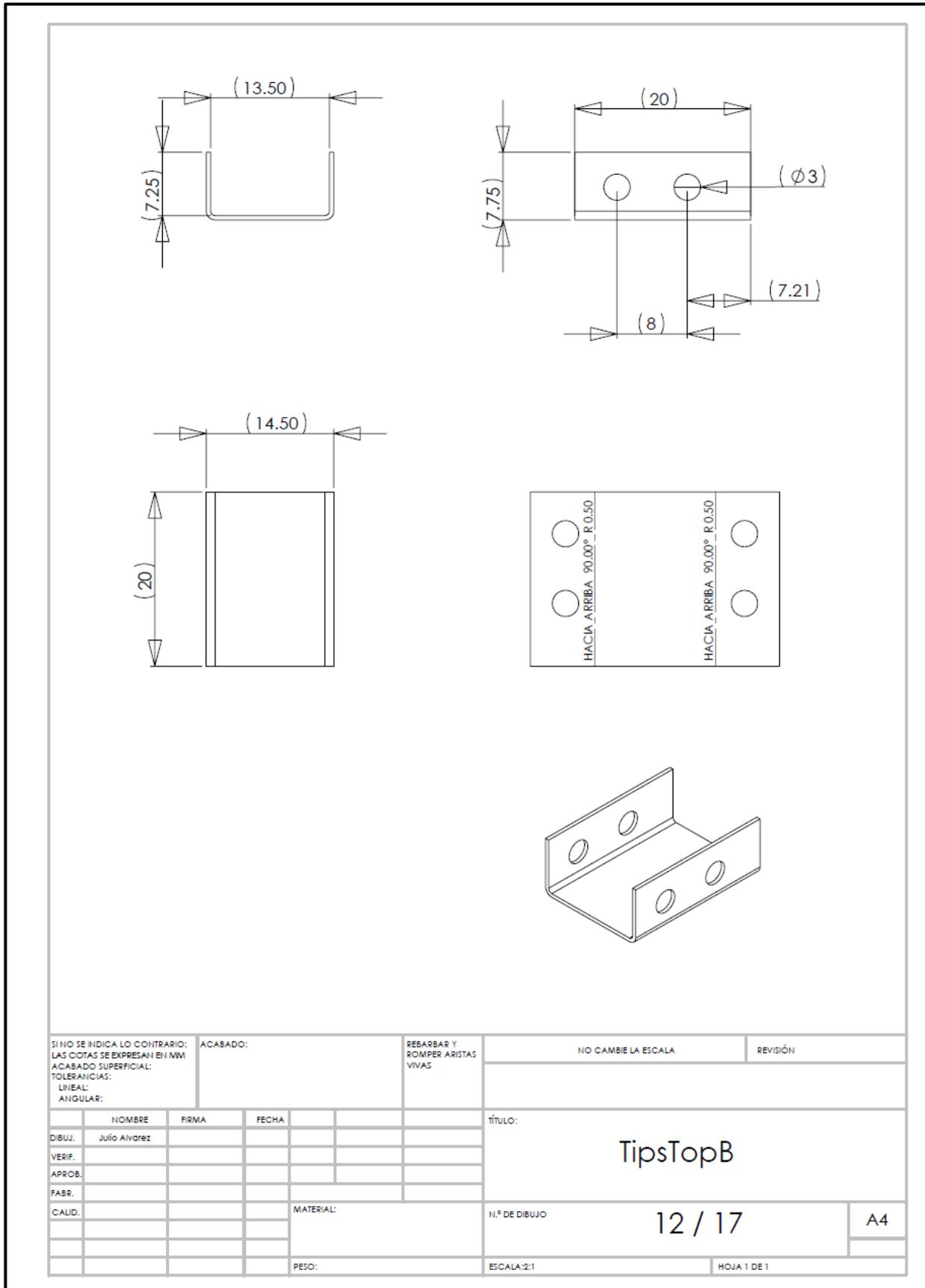


Figura A3. 13 Plano pieza TipsTops B

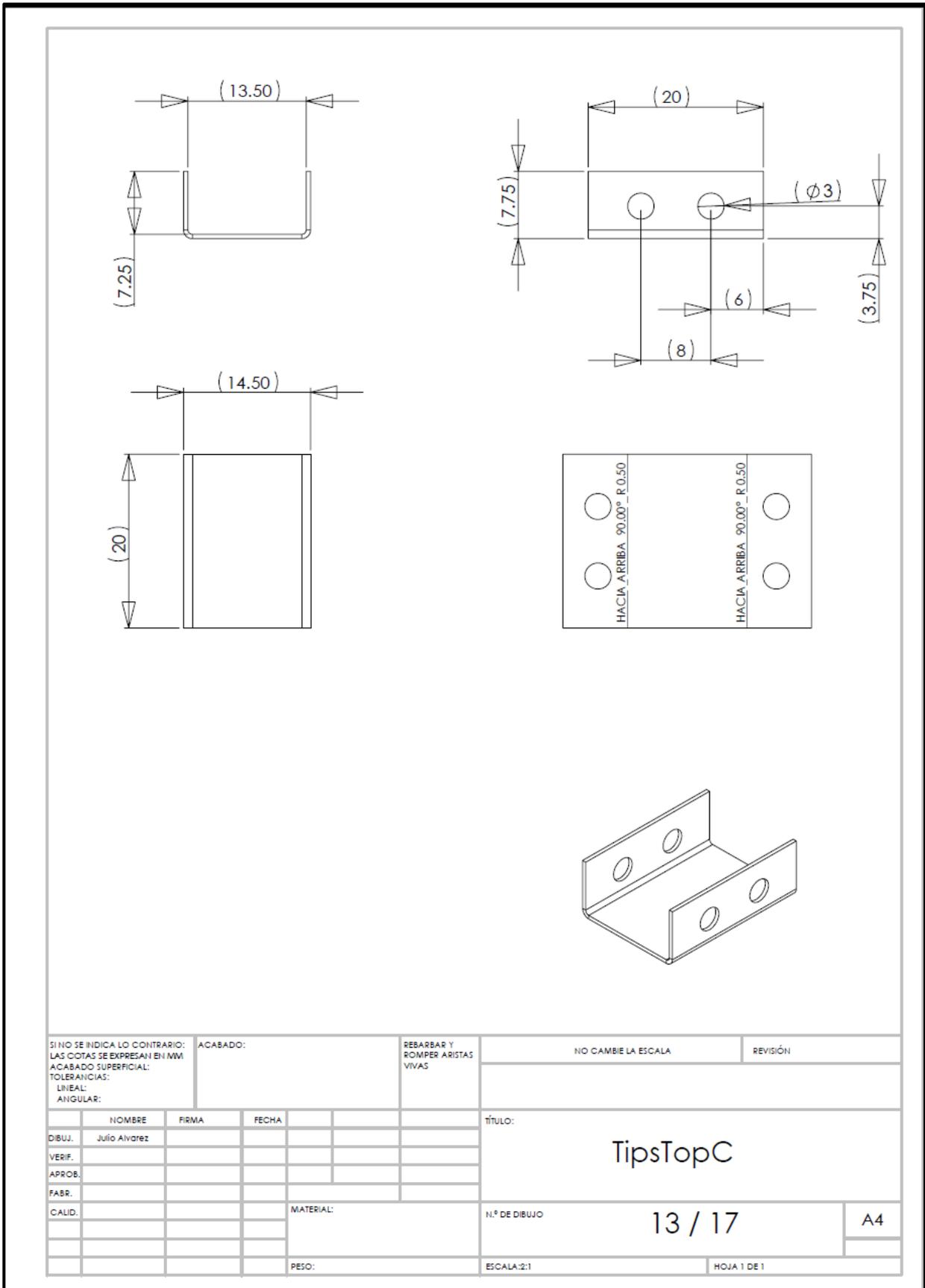


Figura A3. 14 Plano pieza TipsTop C

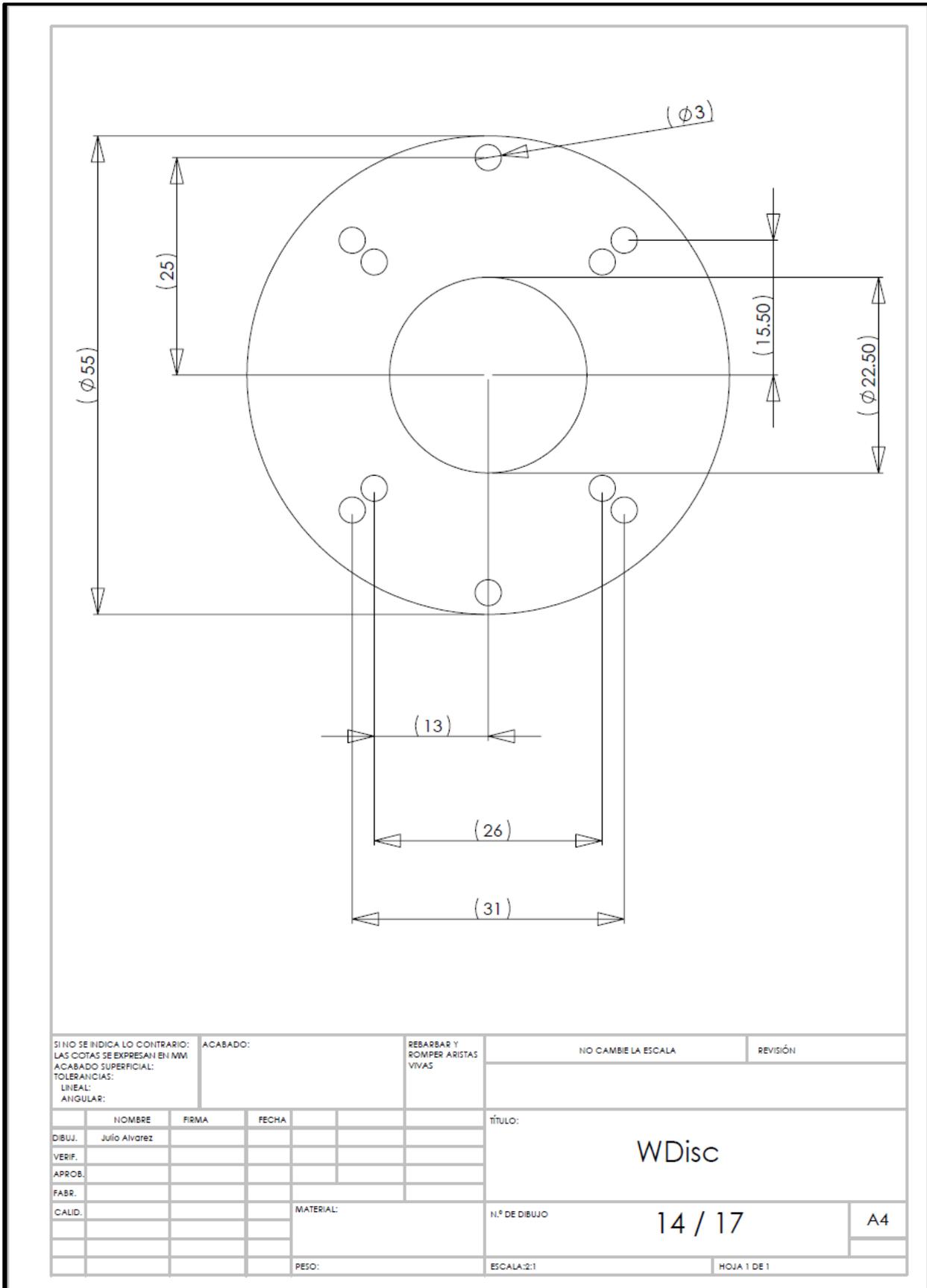


Figura A3. 15 Plano pieza WDisc

ANEXO 4: MONTAJE FINAL

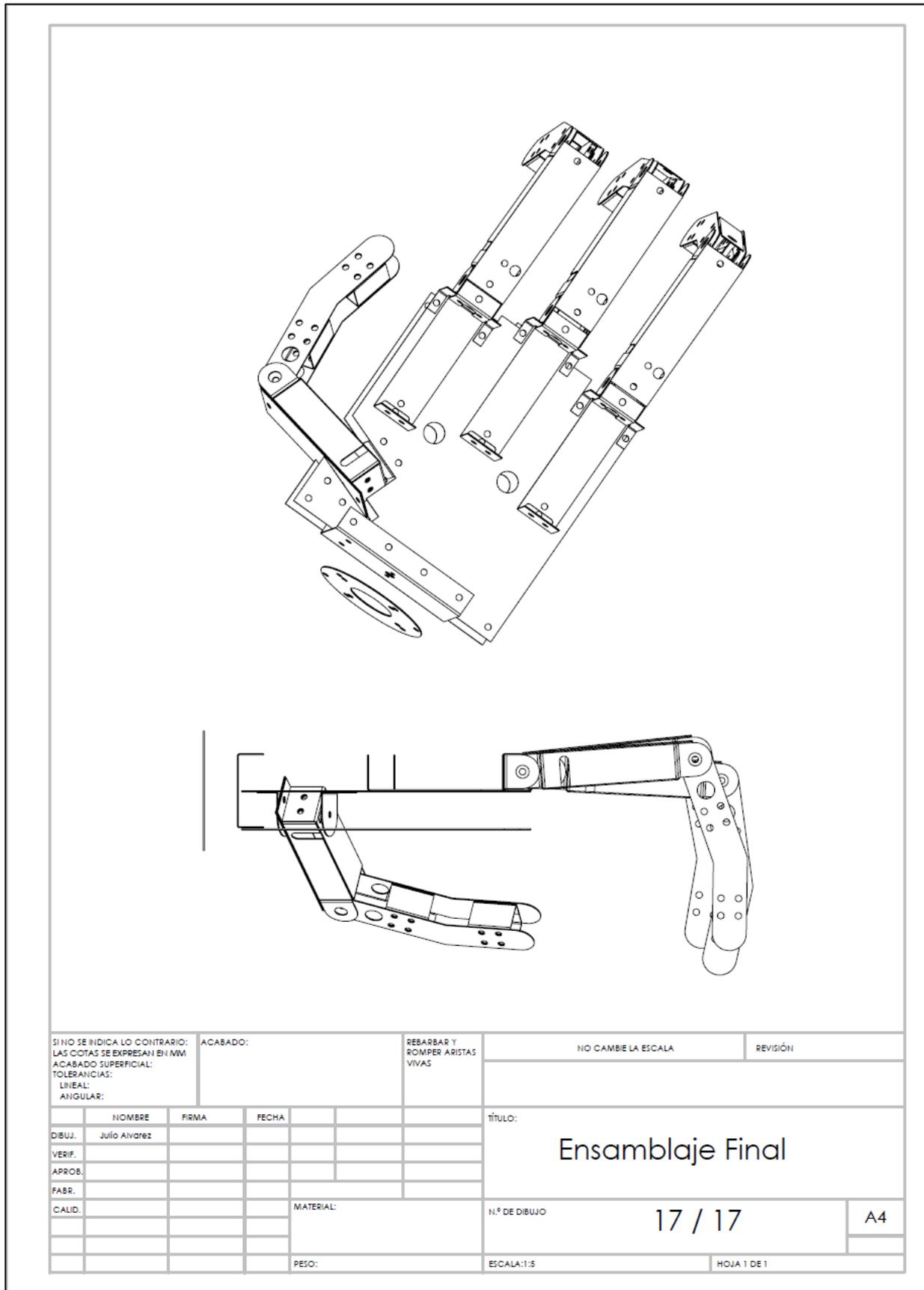


Figura A4. 1 Ensamblaje final mano robótica

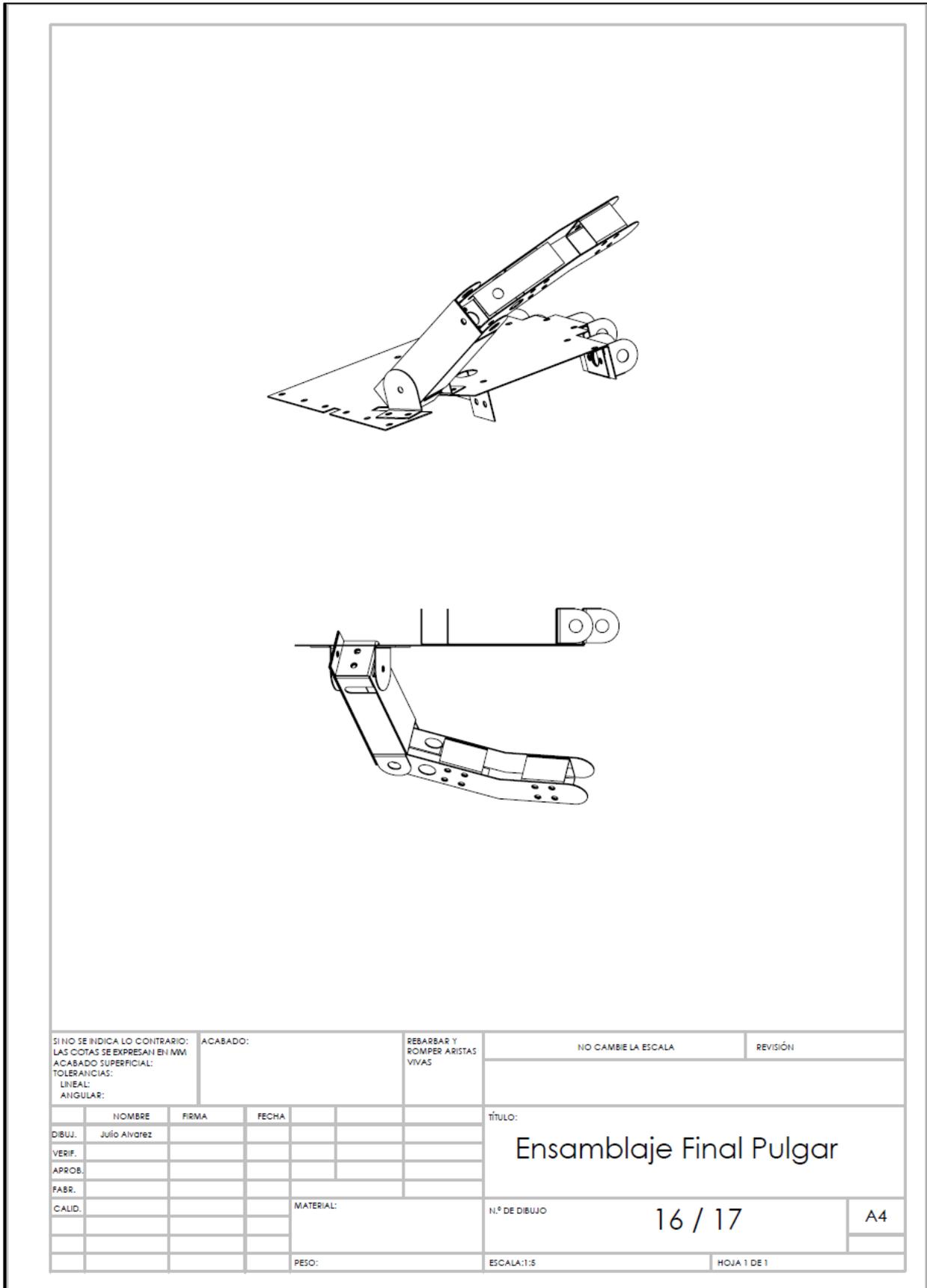


Figura A4. 2 Ensamblaje final pulgar

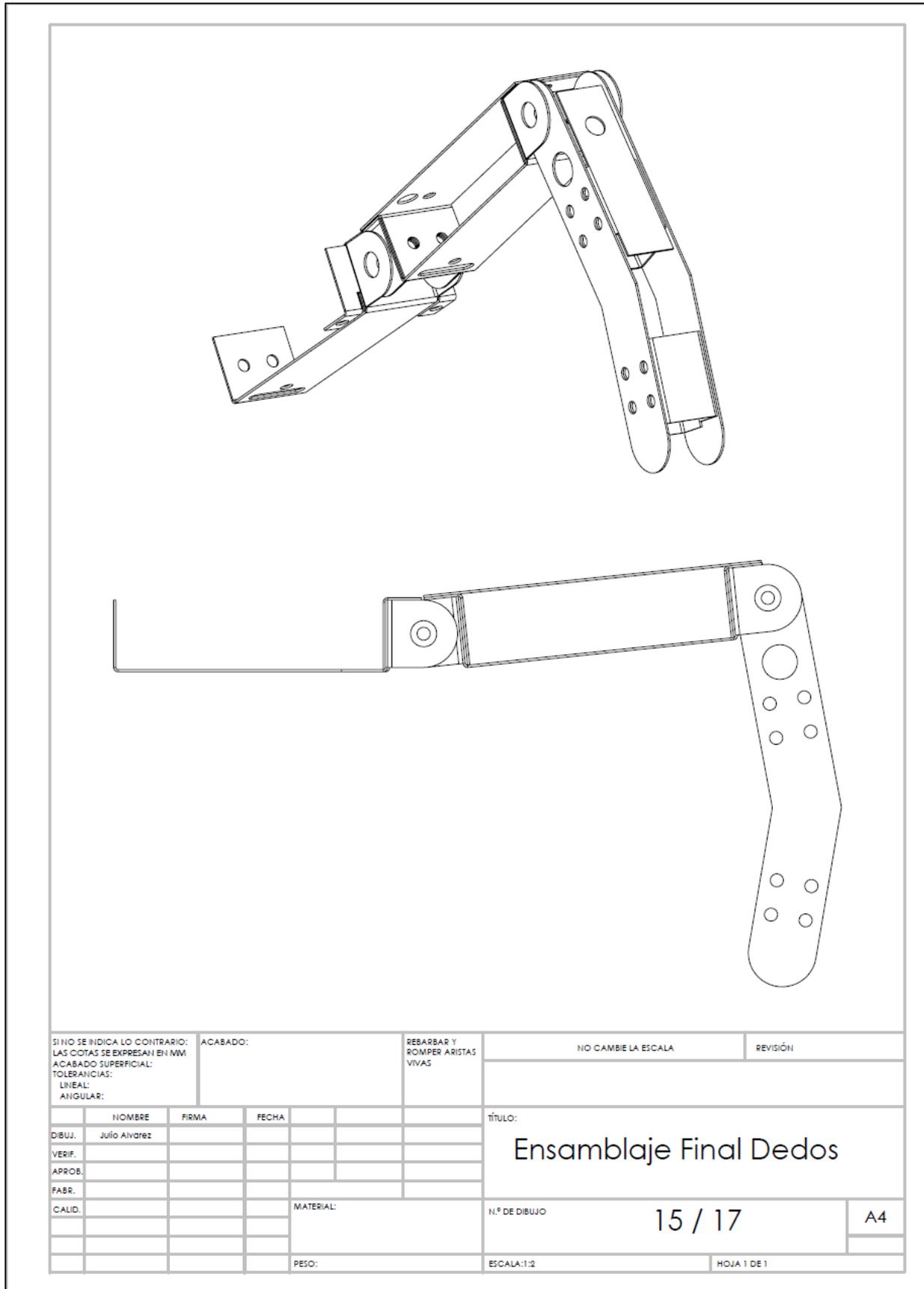


Figura A4. 3 Ensamblaje final mano robótica dedos

ANEXO 5: SOFTWARE DE MEDICIÓN

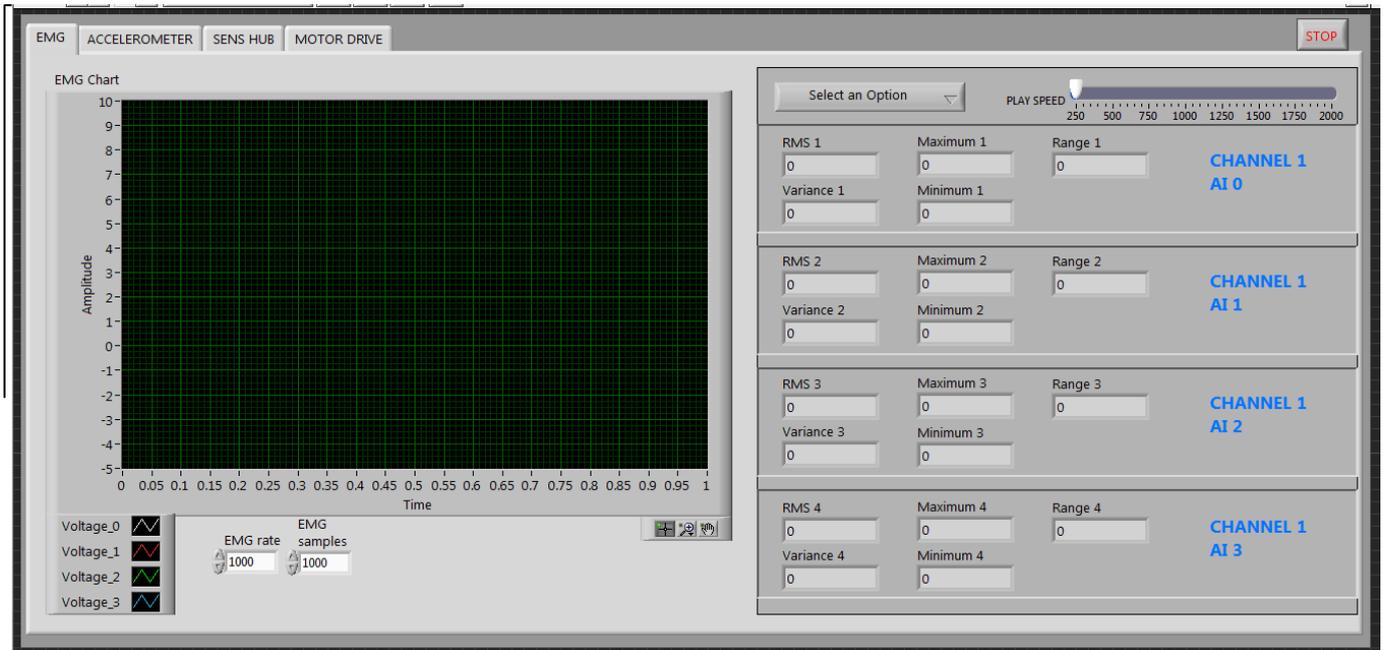


Figura A5. 1 Panel Frontal pestaña EMG

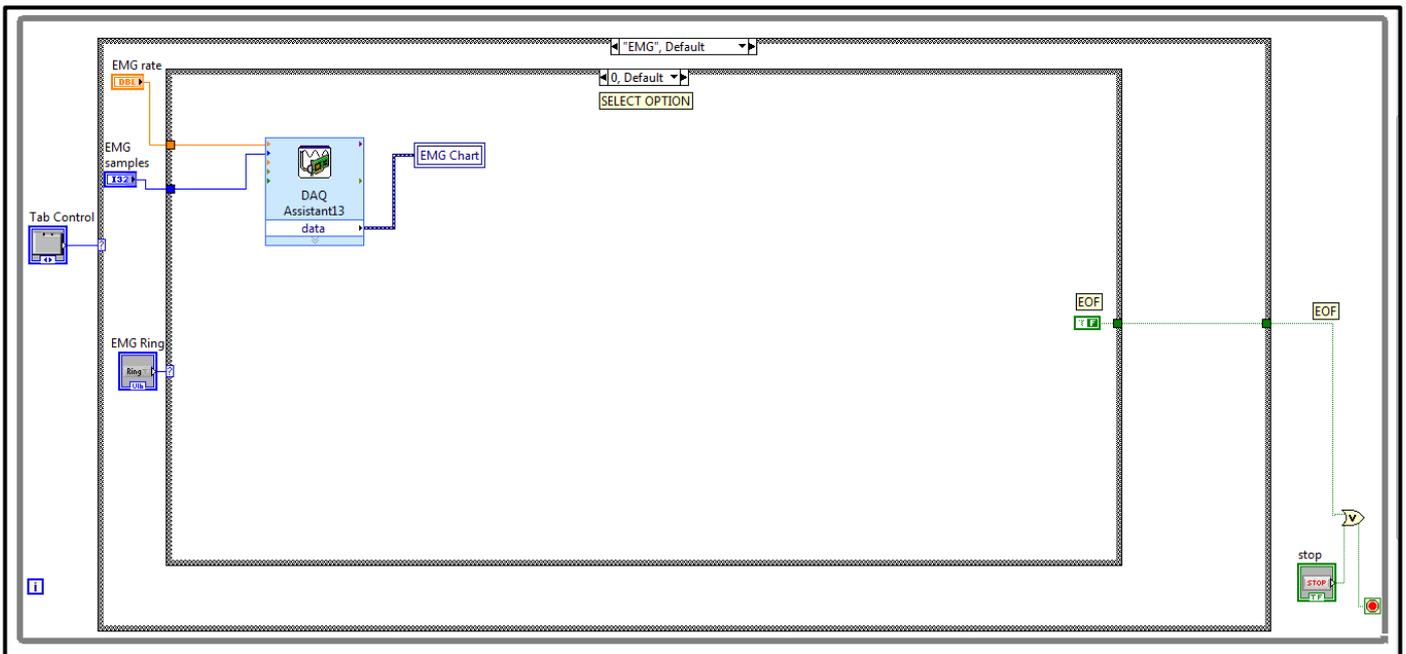


Figura A5. 2 Diagrama de Bloques pestaña EMG (SELECT OPTION)

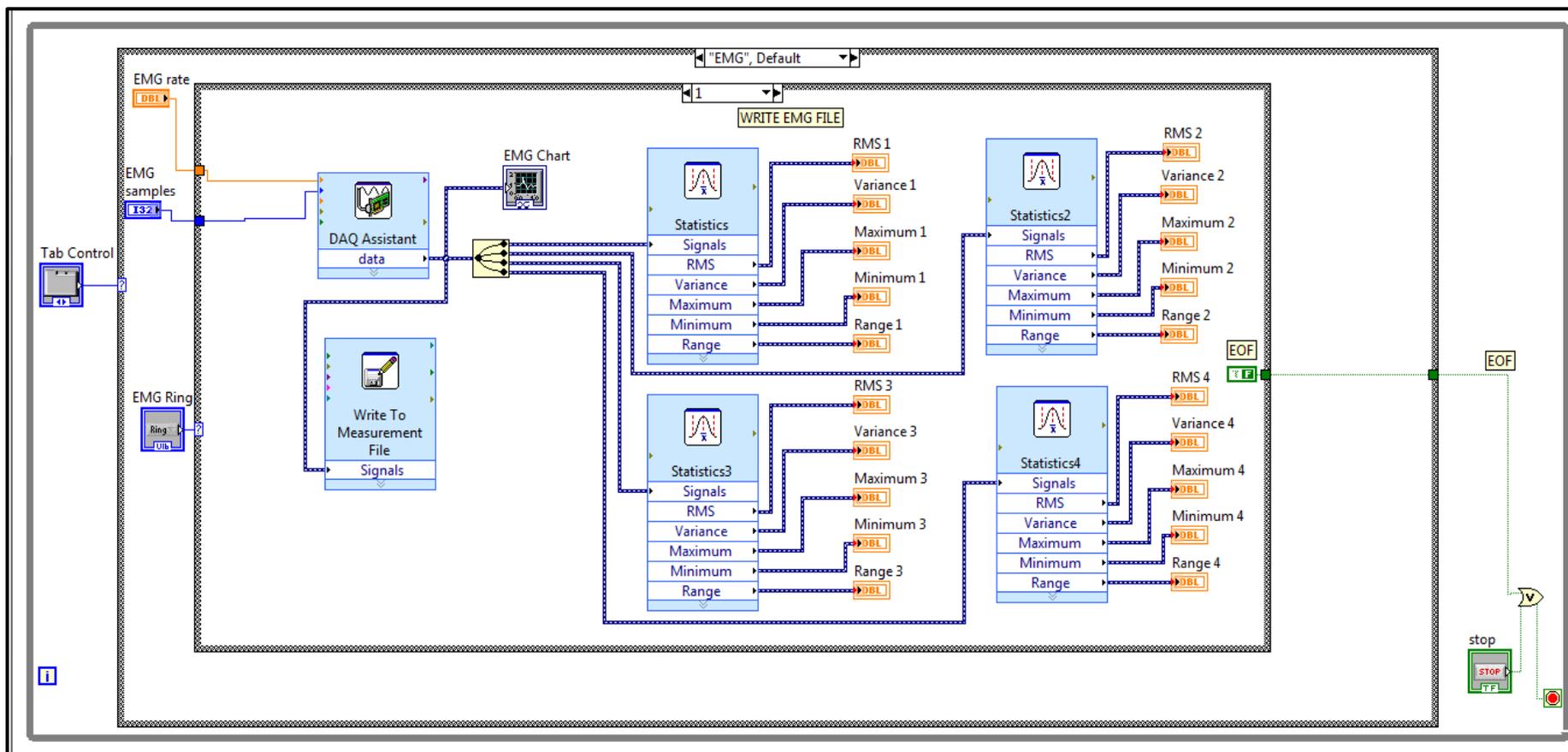


Figura A5. 3 Diagrama de Bloques pestaña EMG (WRITE FILE)

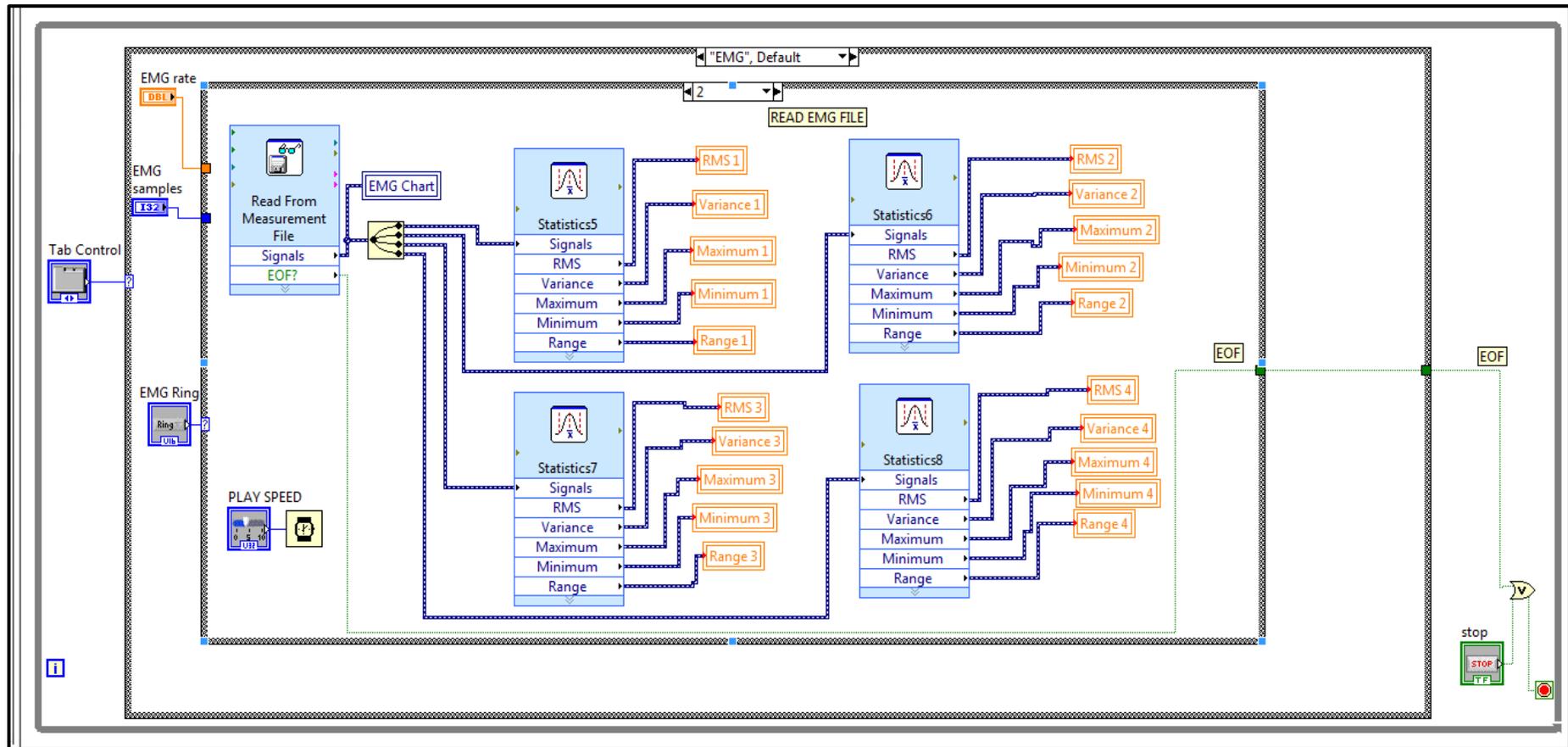


Figura A5. 4 Figura E1.4 Diagrama de Bloques pestaña EMG (READ FILE)

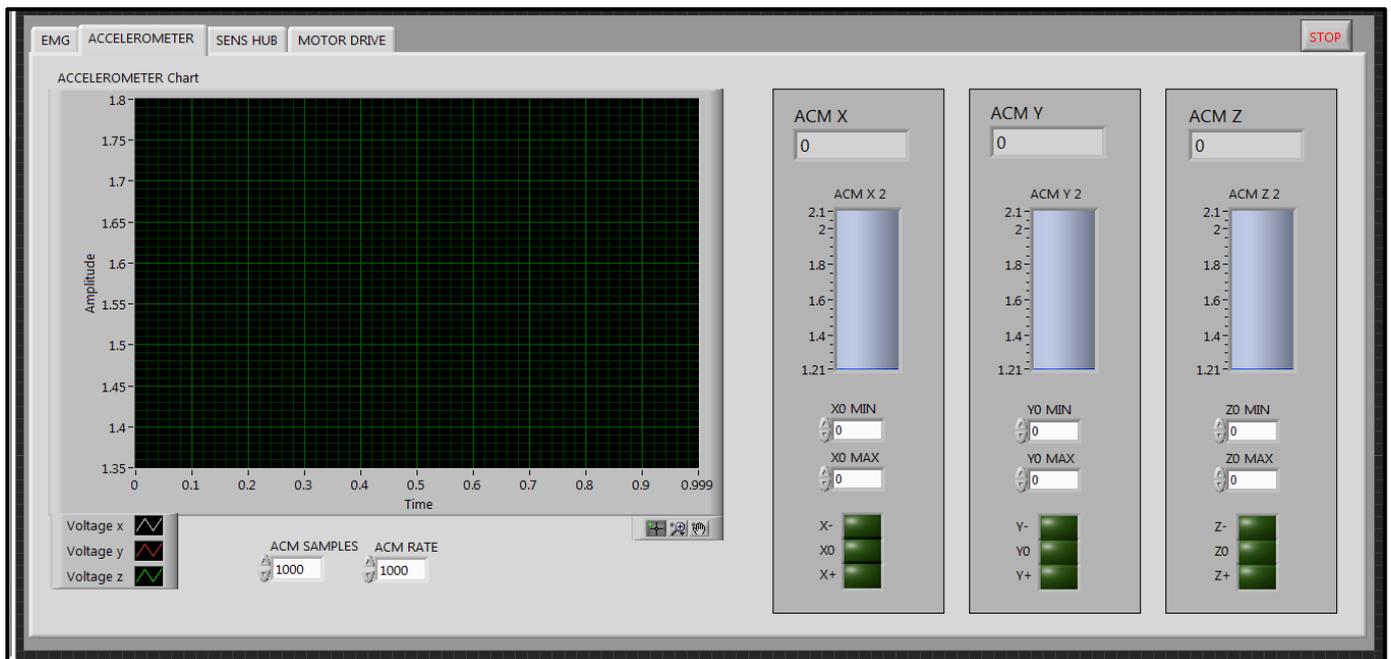


Figura A5. 5 Panel Frontal pestaña ACCELEROMETER

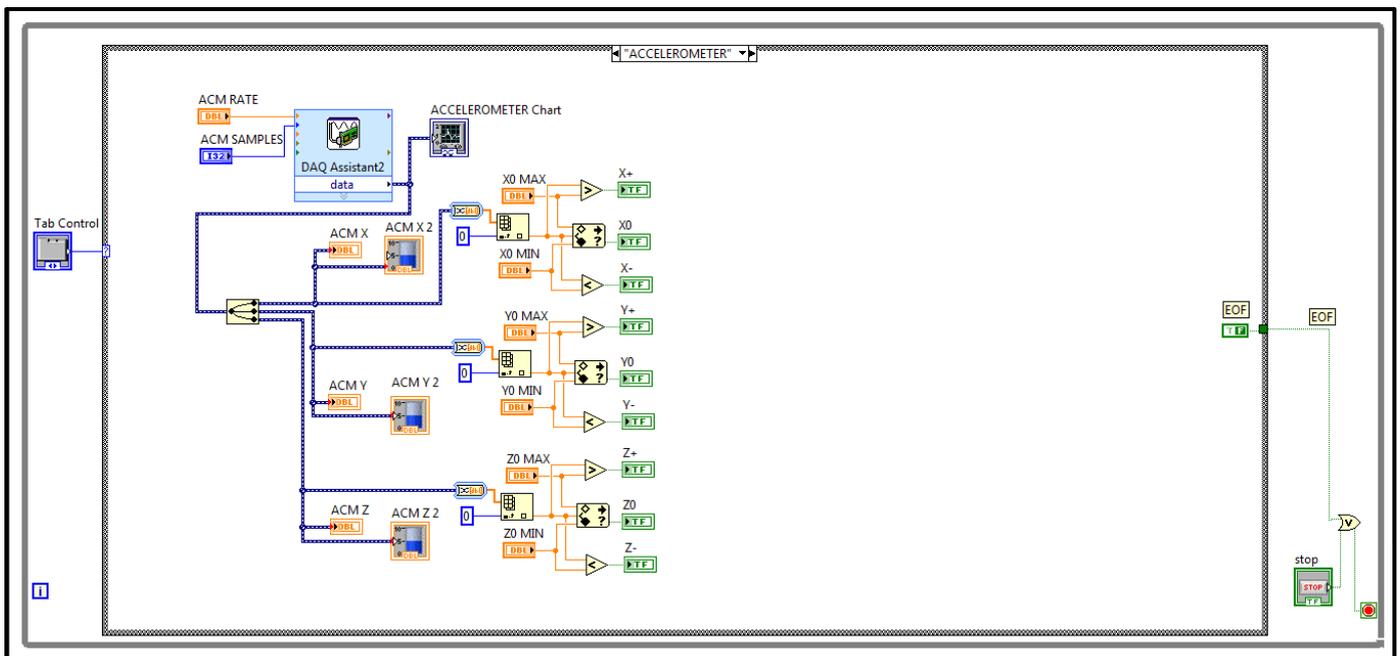


Figura A5. 6 Diagrama de Bloques pestaña ACCELEROMETER

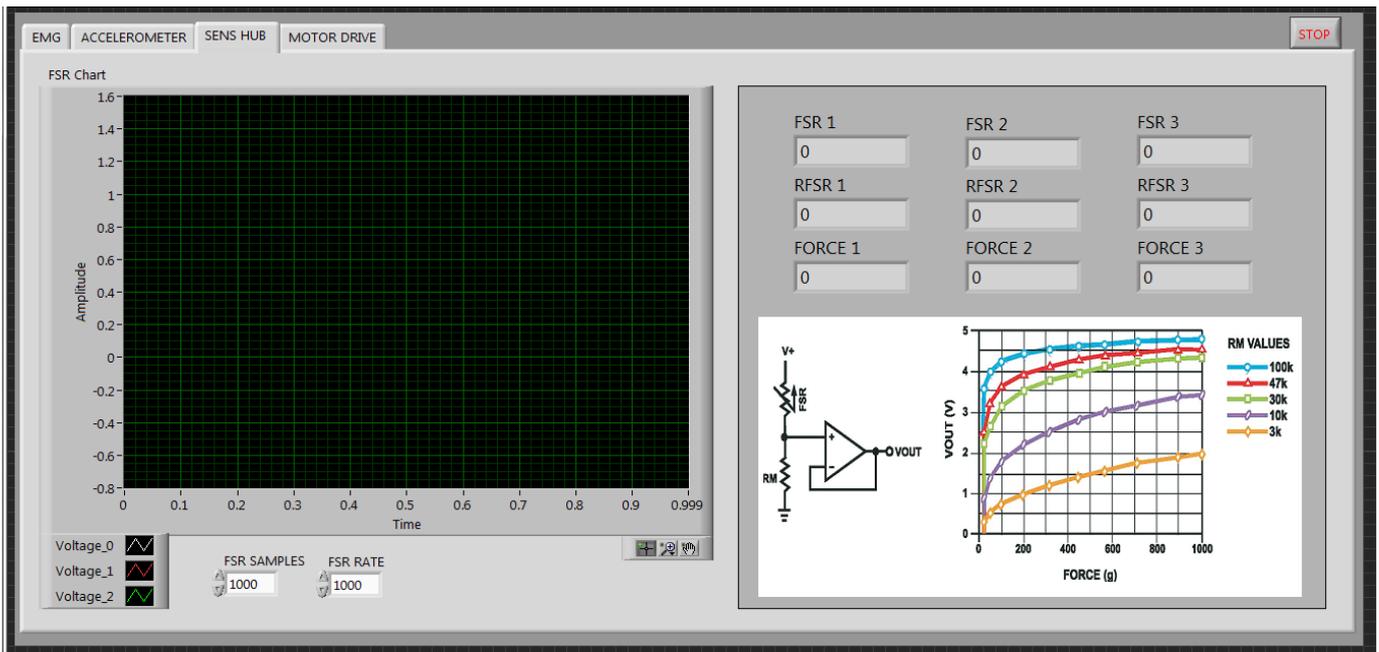


Figura A5. 7 Panel Frontal pestaña SENS HUB

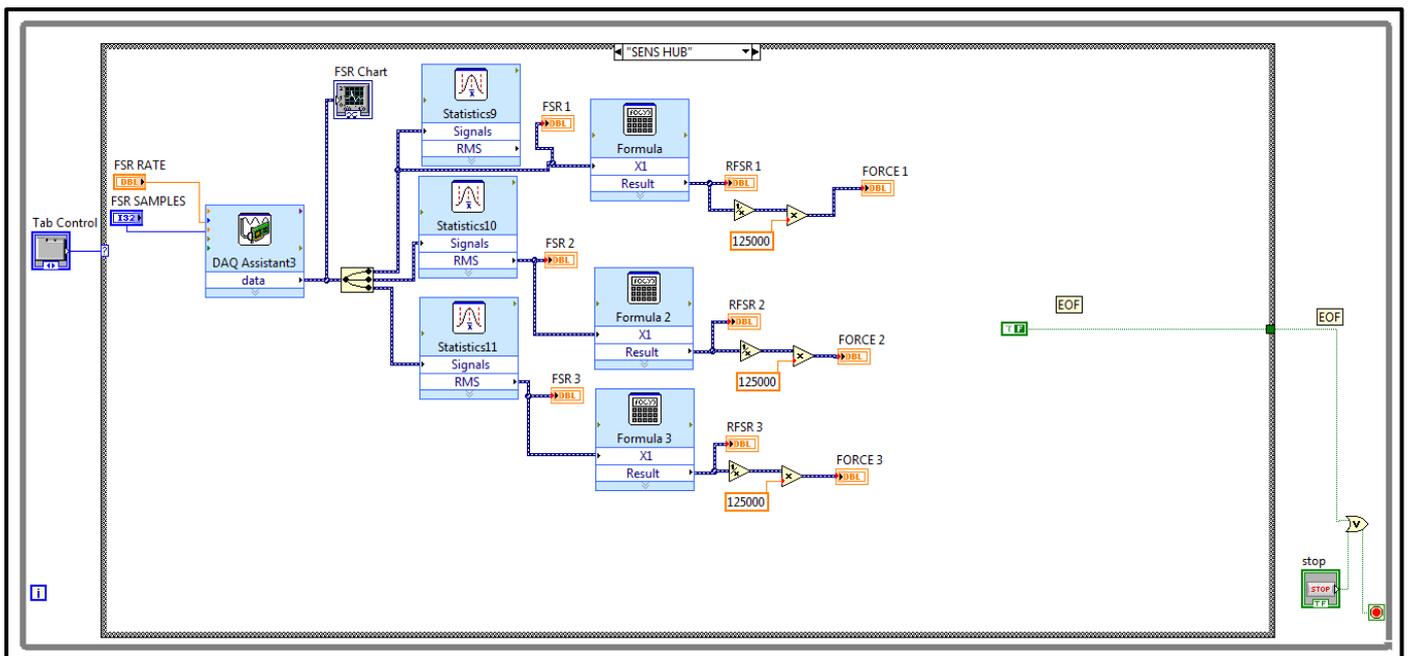


Figura A5. 8 Diagrama de Bloques pestaña SENS HUB

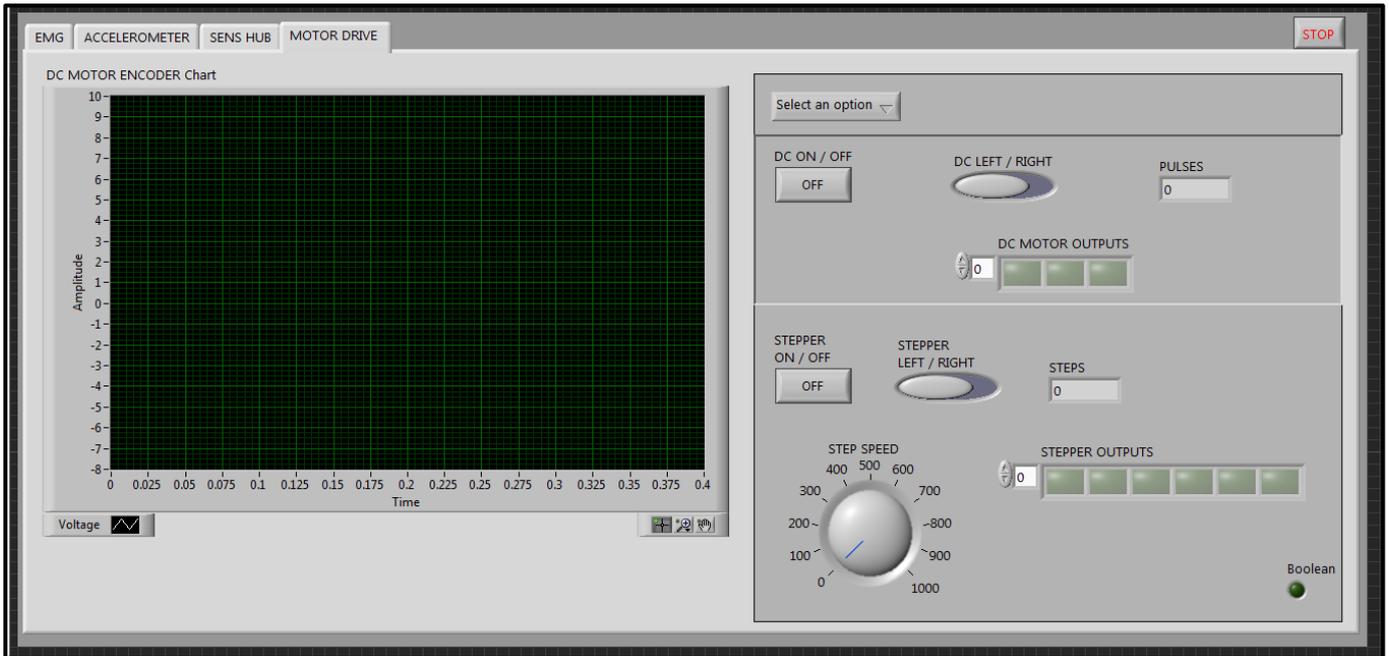


Figura A5. 9 Panel Frontal pestaña MOTOR DRIVE

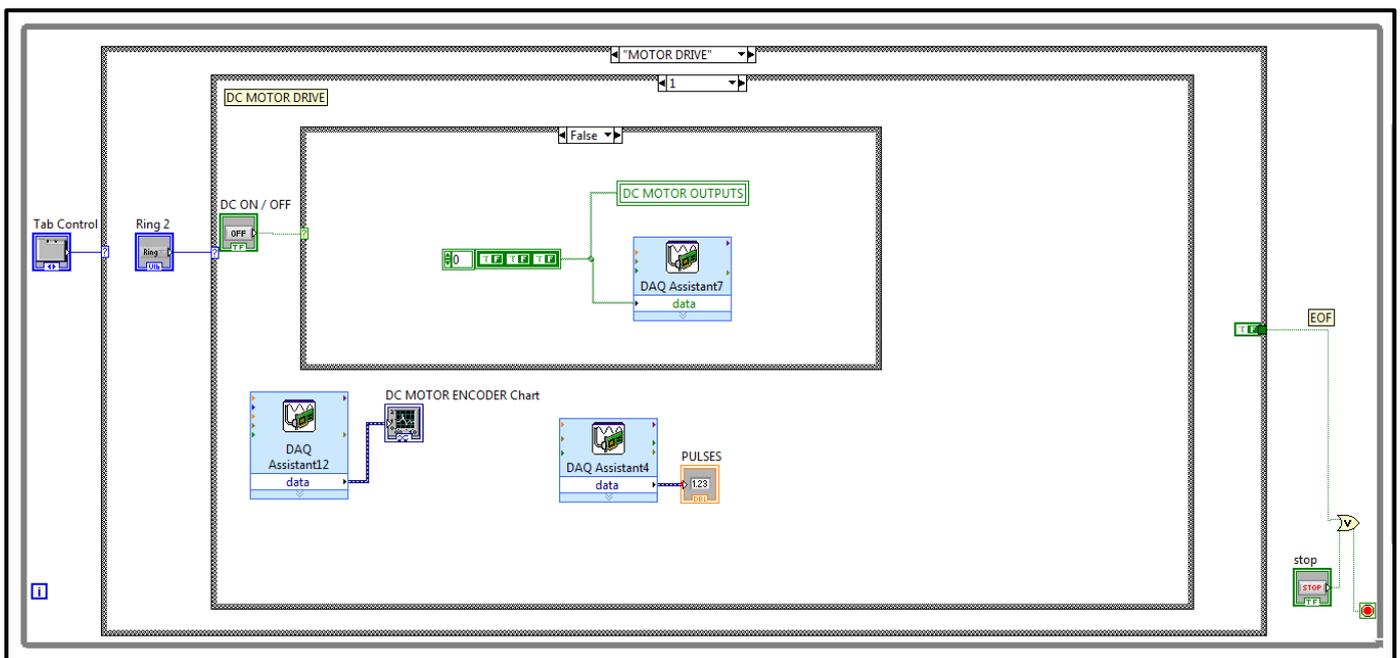


Figura A5. 10 Diagrama de Bloque pestaña MOTOR DRIVE (DC MOTOR OFF)

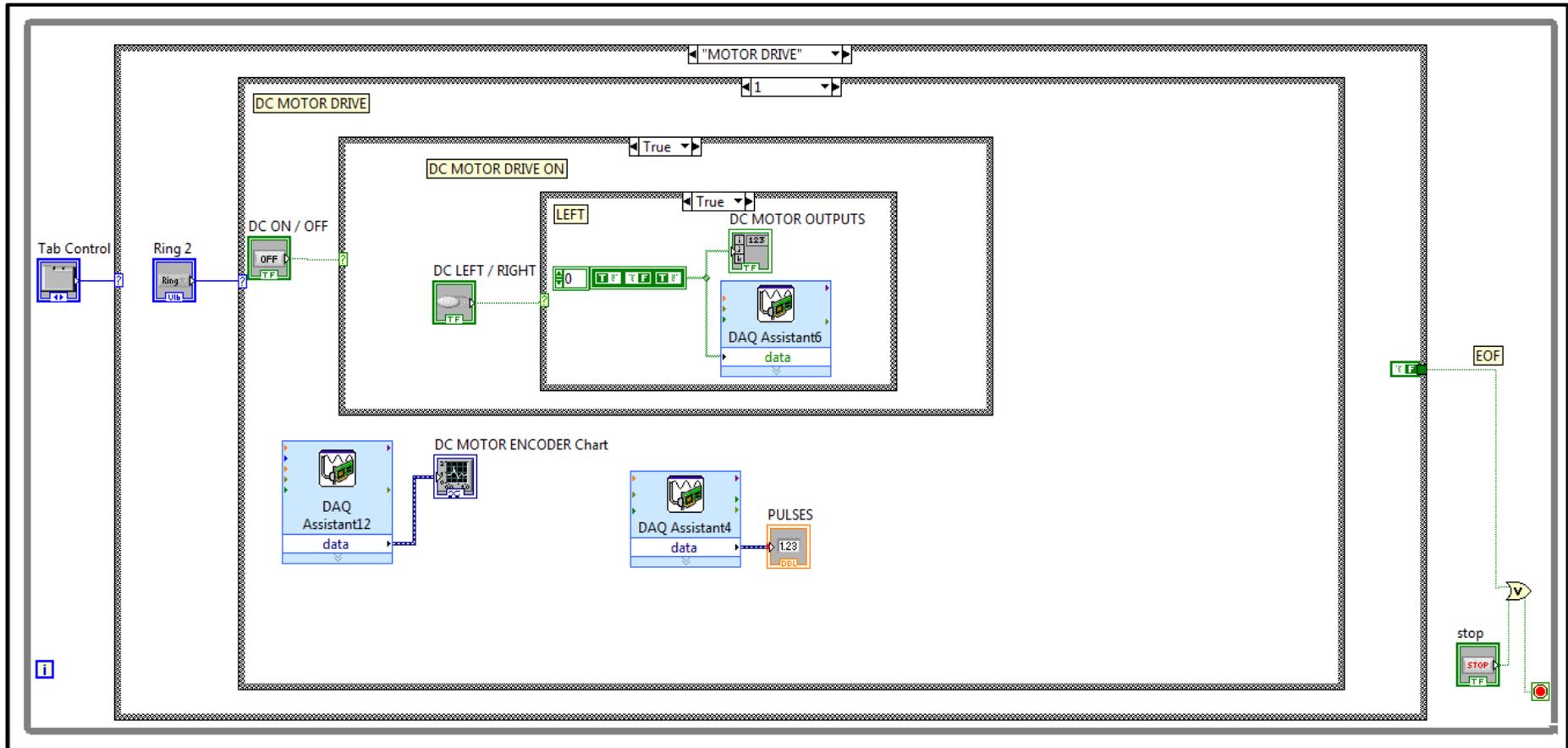


Figura A5. 11 Diagrama de Bloque pestaña MOTOR DRIVE (DC MOTOR ON AND LEFT)

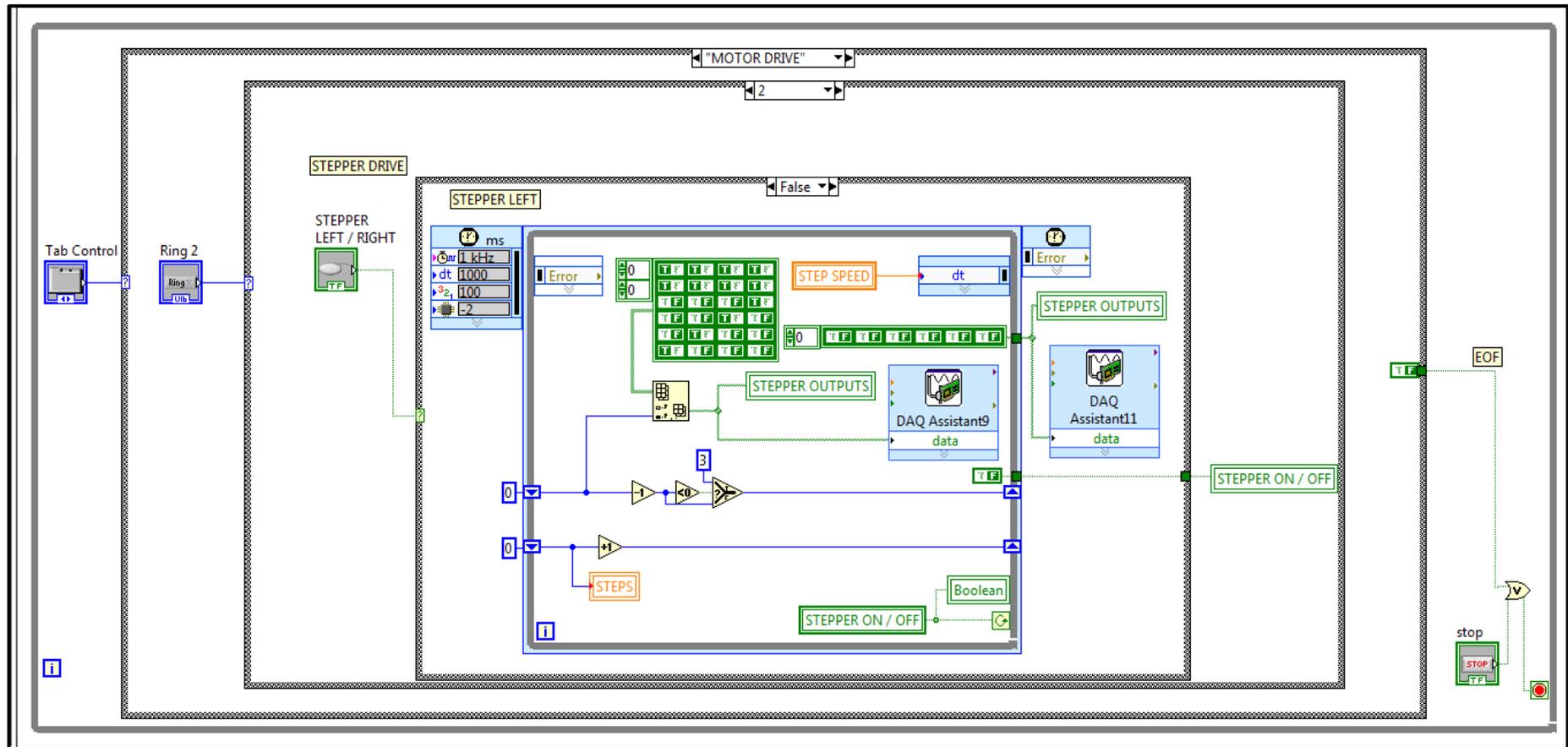


Figura A5. 12 Diagrama de Bloque pestaña MOTOR DRIVE (STEPPER ON AND LEFT)

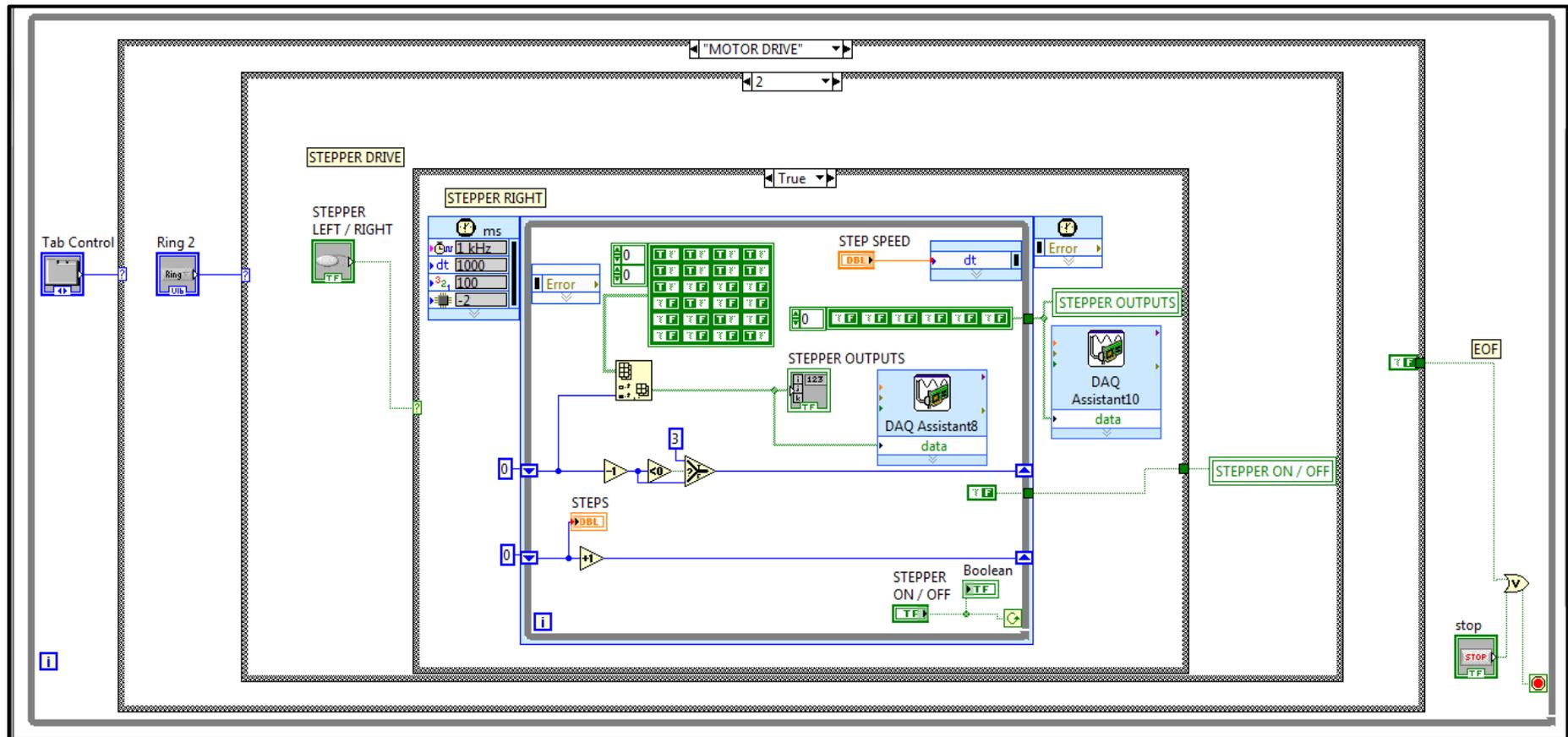


Figura A5. 13 Diagrama de Bloque pestaña MOTOR DRIVE (STEPPER ON AND RIGHT)

