

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Facultad De Ciencias e Ingeniería



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**GUANTE INTELIGENTE QUE RECONOCE GESTOS DE
MANOS PARA REPRODUCCIÓN DE AUDIO**

**Tesis para optar por el Título de Ingeniera Mecatrónica, que
presenta la bachillera:**

Johana Indira Dominguez Medrano

ASESOR: Elizabeth Roxana Villota Cerna, PhD

Lima, noviembre del 2019

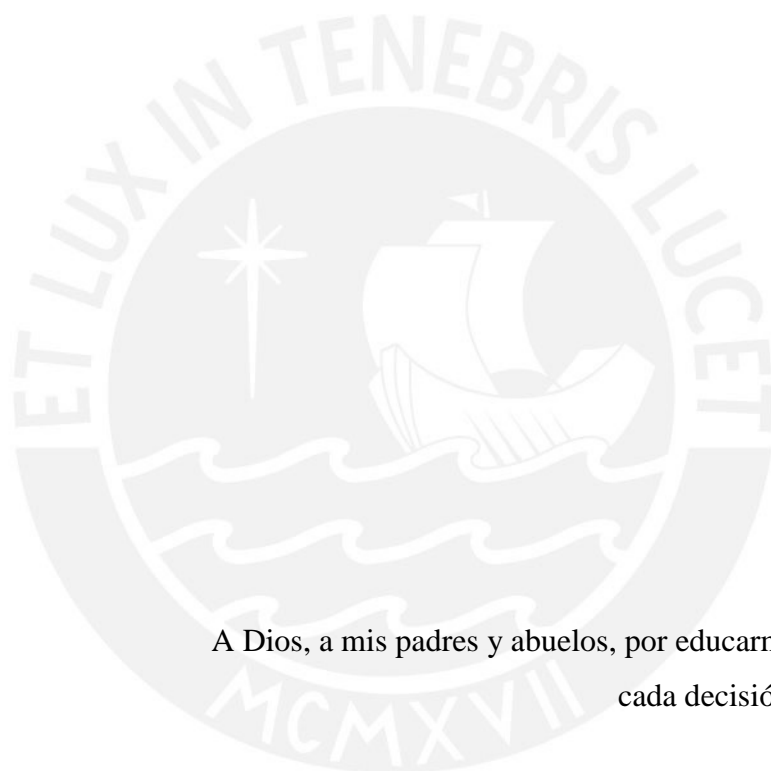
RESUMEN

Debido al cambio continuo en la industria de la música, específicamente en la industria de la música electrónica, lo que siempre se ha buscado es innovar; esto ha sido posible gracias al crecimiento de la tecnología que ha brindado nuevas posibilidades de componer música y hacer performances en vivo. Sin embargo, lo que el *performer* también busca son nuevas maneras de interactuar con su audiencia en una performance en vivo, ya que no siempre es evidente para la audiencia que es lo que sucede cuando el *performer* está detrás de su laptop, loops u otros dispositivos.

Por ello, para que los *performers* puedan interactuar con su música y audiencia de una nueva manera se propone el diseño de un guante, que reconozca gestos que el *performer* realice y un software, mediante el cual se permita reproducir audios que estén vinculados a un gesto; y así, dé la impresión al espectador de cómo “crea música” en frente de él.

El guante está conformado por sensores de flexión, los cuales miden los ángulos de las articulaciones metacarpofalángicas de los dedos de la mano, y un sensor IMU, que mide los movimientos articulares de la muñeca. Los datos que se adquieren de los sensores son enviados a una PC para que sean ingresados al modelo de reconocimiento de gestos construido y consiguientemente se reproduzca el archivo de audio que esté vinculado al gesto entrenado. El entrenamiento del modelo de reconocimiento de gestos se realizó con el algoritmo de clasificación *Random Forest*. La ventaja de este sistema es que se proponen dos modos de uso: modo de usuarios múltiples y modo de usuario específico, dándose en el último caso la oportunidad al usuario de obtener resultados más precisos a cambio de que se tomen el tiempo de tomar los datos de sus gestos para el entrenamiento del modelo. Los resultados de precisión que se obtuvieron para los modelos fueron de 92.80% para el modelo de usuarios múltiples y 92,52% para el modelo de usuario específico.

Por último, se obtuvo un costo del sistema electrónico y mecánico de S/. 1438.40 y un costo de diseño de S/. 9000 para el proyecto, con lo cual se concluye que el costo de diseño se podría repartir y obtener un costo menor por guante en el caso que sea fabricado en volúmenes de producción alto.



A Dios, a mis padres y abuelos, por educarme y apoyarme en cada decisión que he tomado.

A mi asesora Dra. Elizabeth Roxana Villota Cerna, por sus recomendaciones cada vez que tuve alguna pregunta y guiarme en la dirección correcta a lo largo del proceso de la elaboración del trabajo de tesis

A mis hermanas, por enseñarme algo nuevo cada día.

A mis amigos, por apoyarme en todo momento y siempre lograr convertir un mal día en uno bueno.

INDICE DE CONTENIDO

RESUMEN	i
INDICE DE TABLAS	vi
LISTA DE SIMBOLOS	x
INTRODUCCIÓN	1
1. ANTECEDENTES	5
1.1. Tecnológico	5
1.2. Técnico	9
1.2.1. Patentes	9
1.2.2. Artículos técnicos y tesis	11
2. DISEÑO CONCEPTUAL	15
2.1. Requerimientos del sistema guante inteligente	15
2.1.1. Requerimiento general	15
2.1.2. Requerimientos mecánicos	16
2.1.3. Requerimientos electrónicos y de control	19
2.1.4. Requerimientos adicionales	20
2.2. Estructura de funciones	20
2.3. Matriz morfológica	24
2.4. Conceptos de solución	29
2.4.1. Concepto de solución 1	29
2.4.2. Concepto de solución 2	30
2.4.3. Concepto de solución 3	31
2.5. Evaluación técnico-económica	33
2.5.1. Evaluación técnica	33
2.5.2. Evaluación económica	34
3. DISEÑO DE GUANTE INTELIGENTE	36
3.1. Integración de guante inteligente que reconoce gestos de manos para reproducir música	36
3.1.1. Funcionamiento	38
3.2. Diseño electrónico	41
3.2.1. Diagrama de bloques	41

3.2.2.	Componentes electrónicos	42
3.2.3.	Diagramas esquemáticos y PCB	53
3.3.	Diseño mecánico	55
3.3.1.	Ensamble general	56
3.3.2.	Planos de sub-ensamble	56
3.3.3.	Moldes de guante	65
3.4.	Diseño del software	68
3.4.1.	Diagramas de flujo	68
4.	RECONOCIMIENTO DE GESTOS DE MANOS	74
4.1.	Fundamento teórico para análisis de base de datos	75
4.2.	Obtención de base de datos para entrenamiento y validación	77
4.3.	Construcción del modelo de reconocimiento de gestos	79
4.3.1.	Características para reconocimiento de gestos	79
4.3.2.	Métodos de selección de características y de clasificación	80
4.3.3.	Comparación de diferentes modelos	83
4.3.3.1.	Modelo genérico para usuarios múltiples	83
4.3.3.2.	Modelo para usuario específico	88
5.	COSTOS	94
5.1.	Costos de componentes para diseño electrónico	94
5.2.	Costos componentes y materiales para diseño mecánico	95
5.3.	Costo del sistema	96
6.	CONCLUSIONES Y RECOMENDACIONES	98
	REFERENCIAS	100

ANEXOS

ANEXO A: LISTA DE REQUERIMIENTOS

ANEXO B: ESTRUCTURA DE FUNCIONES

ANEXO C: TEORÍA PARA DISEÑO DE *SNAP-FITS* EN *ENCLOSURES* Y
DISEÑO DE *SNAP-FITS* DE *ENCLOSURE* PARTE ELECTRÓNICA

ANEXO D: TEORÍA SELECCIÓN DE CARACTERÍSTICAS

ANEXO E: PRUEBAS PARA MODELO DE RECONOCIMIENTO DE GESTO
TOMANDO DIFERENTES MUESTRAS

ANEXO F: CÓDIGO

ANEXO G: PLANOS Y MOLDES

ANEXO H: COTIZACIÓN

ANEXO I: DATASHEETS Y TABLA DE PROTECCIÓN IP



INDICE DE TABLAS

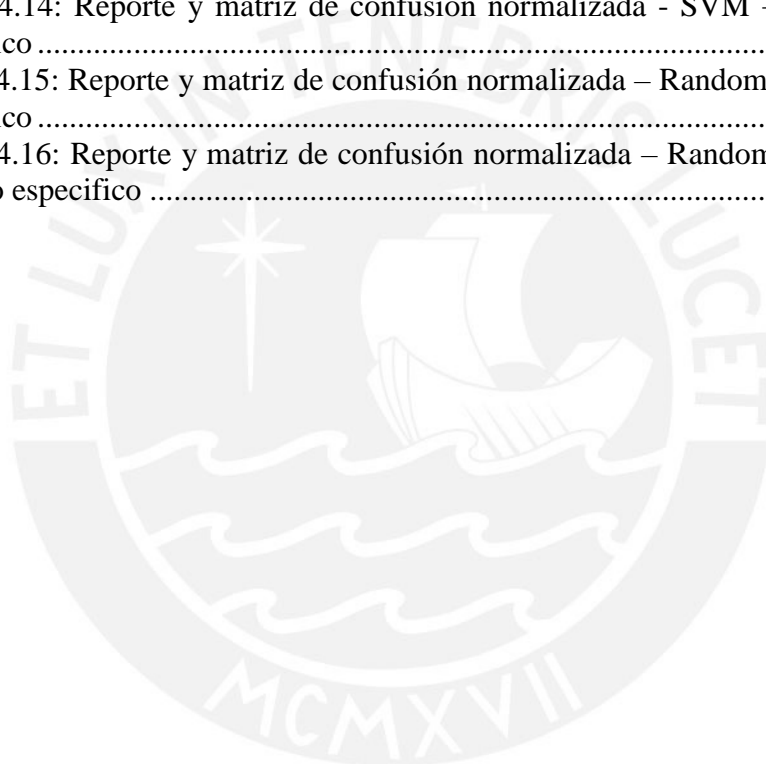
Tabla 1.1: Comparación de modelos para identificación de movimientos de las manos	9
Tabla 2.1: Leyenda – Soluciones para matriz morfológica	24
Tabla 2.2: Matriz morfológica – Energía	25
Tabla 2.3: Matriz morfológica – Sensores	25
Tabla 2.4: Matriz morfológica – Control	26
Tabla 2.5: Matriz morfológica – Comunicación	27
Tabla 2.6: Matriz morfológica – Mecánica	27
Tabla 2.7: Matriz morfológica – Mecánica continuación	28
Tabla 2.8: Evaluación técnica	33
Tabla 2.9: Evaluación económica	34
Tabla 3.1: Parámetros para selección de sensor IMU	43
Tabla 3.2: Alternativas para sensor IMU	44
Tabla 3.3: Parámetros módulo MPU 920	44
Tabla 3.4: Parámetros para selección de sensor de flexión	45
Tabla 3.5: Alternativas para selección de sensor de flexión	46
Tabla 3.6: Alternativas para selección de módulo Wi-Fi	47
Tabla 3.7: Parámetros para selección de microcontrolador	48
Tabla 3.8: Alternativas para selección de microcontrolador	48
Tabla 3.9: Corriente de alimentación	49
Tabla 3.10: Parámetros de batería LiPo seleccionada	50
Tabla 3.11: Parámetros para regulador <i>step-up</i>	50
Tabla 3.12: Alternativas para regulador <i>step-up</i>	50
Tabla 3.13: Parámetros para regulador <i>step-down</i>	52
Tabla 3.14: Alternativas para regulador <i>step-down</i>	52
Tabla 4.1: Codificación para gestos	78
Tabla 4.2: Características para reconocimiento de gestos	80
Tabla 4.3: Resultados para clasificador SVM (lineal) sin PCA	84
Tabla 4.4: Comparación de resultados – Usuarios múltiples	88
Tabla 4.5: Resultados para clasificador LDA sin PCA	89
Tabla 4.6: Comparación de resultados – Usuario específico	92
Tabla 5.1: Costo de diseño eléctrico	95
Tabla 5.2: Costo de fabricación enclosures	95
Tabla 5.3: Costo de materiales para el diseño del guante – tela algodón	95
Tabla 5.4: Costo de materiales para el diseño del guante – tela Scuba	96
Tabla 5.5: Costo total del sistema	96

INDICE DE FIGURAS

Figura 1.1: Leon Theremin tocando su instrumento: El theremin.....	6
Figura 1.2: Theremin actual.....	6
Figura 1.3: Hasbro’s “GrooveIt” diseñado por KID.....	6
Figura 1.4: Sayre Glove.....	7
Figura 1.5: Laetitia Sonami usando el Lady’s Glove.....	7
Figura 1.6: Mi.mu Gloves.....	8
Figura 1.7: 5DT Data Gloves.....	9
Figura 1.8: Diseño de digital entry data gloves.....	10
Figura 1.9: Diseño de Virtual reality glove system with fabric conductors.....	11
Figura 1.10: Esquemático del cableado de los guantes.....	11
Figura 1.11: Sistema integrado para Smart Glove.....	12
Figura 1.12: Esquemático del dispositivo basado en sensores MEMS.....	13
Figura 1.13: Sensores MEMs asegurados en mano de sujeto.....	13
Figura 1.14: Sección transversal del antebrazo en los extremos distales del radio y cúbito.....	14
Figura 1.15: Prototipo usado alrededor de la muñeca.....	14
Figura 2.1: Articulaciones interfalángicas.....	17
Figura 2.2: Movimientos articulares de la muñeca.....	17
Figura 2.3: Black Box.....	20
Figura 2.4: Bloque de Energía.....	21
Figura 2.5: Bloque de Comunicación.....	21
Figura 2.6: Bloque de Interfaz.....	22
Figura 2.7: Bloque de Sensores.....	23
Figura 2.8: Bloque de Software (Control).....	23
Figura 2.9: Bloque de Mecánica.....	24
Figura 2.10: Bosquejo primer concepto solución.....	30
Figura 2.11: Bosquejo segundo concepto solución.....	31
Figura 2.12: Bosquejo tercer concepto solución.....	32
Figura 2.13: Análisis técnico-económico.....	35
Figura 3.1: Integración del Sistema.....	37
Figura 3.2: Integración parte electrónica.....	38
Figura 3.3: Integración alimentación.....	38
Figura 3.4: Integración sensores en guante.....	38
Figura 3.5: Diseño interfaz de usuario – Pantalla Inicio.....	39
Figura 3.6: Diseño interfaz de usuario - Entrenamiento.....	39
Figura 3.7: Diseño interfaz de usuario - Interacción.....	40
Figura 3.8: Diagrama de operaciones del sistema para usuarios específicos.....	41
Figura 3.9: Diagrama de bloques.....	42
Figura 3.10: posición, velocidad y aceleración de movimientos promedios de mano de un sujeto de modo lento, natural y rápido.....	43
Figura 3.11: Estimación y medición de velocidad angular de movimiento flexión - extensión de la muñeca de un sujeto promedio.....	43
Figura 3.12: Sensor IMU – MPU 9250.....	44
Figura 3.13: Sensor IMU – MPU 9250 en enclosure.....	44
Figura 3.14: Ángulos de articulación de dedos y muñeca en diferentes sujetos.....	45
Figura 3.15: Sensor de flexión SEN-08606.....	46
Figura 3.16: Sensores de flexión SEN-08606 en guante.....	46
Figura 3.17: Módulo Wi-Fi ESP-01.....	47

Figura 3.18: Módulo Wi-Fi ESP-01 en enclosure	47
Figura 3.19: Arduino Nano.....	48
Figura 3.20: Arduino Nano en enclosure.....	48
Figura 3.21: Batería LiPo	49
Figura 3.22: Batería LiPo en enclosure	49
Figura 3.23: Módulo step-up U3V12F5	51
Figura 3.24: Módulo step-up U3V12F5 en enclosure	51
Figura 3.25: Corriente máxima de salida vs Voltaje de entrada.....	51
Figura 3.26: Módulo step-down D24V5F3	53
Figura 3.28: Diagrama de conexiones para sensor de flexión.....	53
Figura 3.29: Rango de voltaje vs Resistencia R1	54
Figura 3.30: Diagrama esquemático general del sistema	54
Figura 3.31: PCB – Vista superior e inferior.....	55
Figura 3.32: Vista isométrica del dispositivo	56
Figura 3.33: Vista isométrica enclosure - batería	57
Figura 3.34: Vista principal enclosure - batería.....	57
Figura 3.35: Detalle B.....	57
Figura 3.36: Detalle C.....	57
Figura 3.37: Corte A-A enclosure - batería	58
Figura 3.38: Diseño Snap-fit.....	59
Figura 3.39: Factor de magnificación de deflexión Q vs Ratio L/t	59
Figura 3.40: Vista lateral enclosure - batería.....	61
Figura 3.41: Vista isométrica enclosure - electrónica	61
Figura 3.42: Partes de enclosure - electrónica.....	62
Figura 3.43: Vista principal enclosure - electrónica.....	62
Figura 3.44: Detalle B.....	62
Figura 3.45: Detalle C.....	62
Figura 3.46: Corte A-A enclosure – electrónica y vista posterior	63
Figura 3.47: Vista lateral enclosure - electrónica	64
Figura 3.48: Detalle D	65
Figura 3.49: Detalle G	65
Figura 3.50: Detalle E.....	65
Figura 3.51: Detalle F.....	65
Figura 3.52: Diseño de Molde guante.....	66
Figura 3.53: Diseño de Molde - pockets y adiciones.....	67
Figura 3.54: Diseño de molde - cobertor	67
Figura 3.55: Diagrama de flujo general de dispositivo (guante) - recolección y envío de data.....	69
Figura 3.56: Diagrama de flujo Encapsular_data	70
Figura 3.57: Diagrama de flujo de reconocimiento de gestos y reproducción de audio	71
Figura 3.58: Diagrama de flujo para obtención del modelo de reconocimiento de gestos	72
Figura 3.59: Diagrama de flujo de vinculación de gestos y audios.....	73
Figura 4.1: Código sensor de flexión (Arduino).....	75
Figura 4.2: Gestos correspondientes a base de datos.....	77
Figura 4.3: Posición de sensores EMG.....	78
Figura 4.4: Código PCA (Python)	81
Figura 4.5: Código LDA (Python).....	82

Figura 4.6: Código modo usuarios múltiples - separación y estandarización de datos (Python)	84
Figura 4.7: Análisis de 2 componentes principales – Usuarios múltiples	85
Figura 4.8: Reporte y matriz de confusión normalizada - SVM - Usuarios múltiples	86
Figura 4.9: Reporte y matriz de confusión normalizada - SVM + PCA - Usuarios múltiples	86
Figura 4.10: Reporte y matriz de confusión normalizada – Random Forest + PCA - Usuarios múltiples	87
Figura 4.11: Reporte y matriz de confusión normalizada – Random Forest - Usuarios múltiples	87
Figura 4.12: Análisis de 2 componentes principales – Usuario específico	89
Figura 4.13: Reporte y matriz de confusión normalizada - SVM - Usuario específico	90
Figura 4.14: Reporte y matriz de confusión normalizada - SVM + PCA – Usuario específico	91
Figura 4.15: Reporte y matriz de confusión normalizada – Random Forest - Usuario específico	91
Figura 4.16: Reporte y matriz de confusión normalizada – Random Forest + PCA - Usuario específico	92



LISTA DE SIMBOLOS

$\alpha_{diseño}$: ángulo de diseño <i>snap-fit</i> [°]
$\alpha_{efectivo}$: ángulo efectivo <i>snap-fit</i> [°]
b	: ancho de <i>snap-fit</i> [mm]
ϵ_o	: strain [%]
E	: módulo de Young [MPa]
$F_{montaje-max}$: Fuerza de montaje máxima [N]
h	: espesor de <i>snap-fit</i> [mm]
l	: longitud <i>snap-fit</i> [mm]
P	: Fuerza de deformación [N]
Q	: factor deflexión
μ	: coeficiente de fricción
y	: deformación admisible [mm]



INTRODUCCIÓN

El origen de la música es casi imposible de determinar, por lo que muchas fuentes la consideran una adaptación evolucionaria. Con el pasar del tiempo, la tecnología también ha ejercido una influencia en la música, no sólo transformando la manera en la que esta es producida, sino también en cómo es compuesta. Con la invención de los primeros instrumentos musicales electrónicos, el telarmonio en 1897 y el theremin en 1920, se da paso a lo que hoy llamamos música electrónica. Estos instrumentos no eran vendidos, sino que se hacía uso de ellos en performances públicas, en las cuales se demostraba como es que funcionaban.

Hasta la fecha, en la industria del entretenimiento, han ocurrido abundantes modificaciones, y la industria de música electrónica no ha sido la excepción. Actualmente, los servicios de música por suscripción han cambiado la manera en cómo el consumidor adquiere la música que desea escuchar. Sin embargo, contra lo que se podría esperar, el número de conciertos y performances no ha dejado de crecer en los últimos años, esto se puede atribuir a que ahora los consumidores tienen más fácil acceso a escuchar a sus artistas favoritos y en lugar de comprar álbumes físicos o digitales prefieren comprar tickets de conciertos y performances.

De acuerdo con el *International Music Summit Business Report* del 2017, el valor global anual de la industria de música electrónica fue de 7.4 billones de dólares americanos en el periodo 2016/2017, creciendo un 3% en comparación al periodo 2015/2016 [1]. Sin embargo, en el mismo reporte para el presente año, se declara que el valor global anual disminuyó a 7.3 millones de dólares americanos, decayendo en 2%. *IMS* reporta que la caída en el valor de la industria se debe a el *crossover* que sucede actualmente entre los diferentes géneros musicales, por lo que muchas pistas de música electrónica están siendo clasificadas como pop o R&B [2].

El *crossover* de géneros también indica que está ocurriendo una influencia y transformación en los géneros ya existentes y estables debido al género de música electrónica. No obstante, el género de música electrónica, aunque en innovación

continua, no está progresando en su totalidad. Además, lo que una vez hizo que la música electrónica sea única era la posibilidad de ver en vivo como esta era creada, pero con el pasar del tiempo se ha creado una barrera que no permite que el *performer* se conecte con la audiencia y no hay nada innovador que esta espere de él.

En los últimos 30 años, el crecimiento de la tecnología ha transformado radicalmente las expectativas y posibilidades de composición y performances de música en vivo. Hoy en día, la idea de una performance de música electrónica en vivo ya no es extraña, sin embargo, el *performer* es de una nueva generación, y aunque se puede influenciar por artistas que admira, debe buscar nuevas maneras de innovar e interactuar con su audiencia. De acuerdo a una entrevista de *CDM* con Imogen Heap, artista inglesa quien es conocida por su participación en la industria de música electrónica y pionera en la fusión de tecnología wearable y música, explica que su razonamiento detrás del desarrollo de *wearables* para música es que no quería “sujetar” nada ni romper el flujo de una performance, pero quería control realmente minucioso sin tener que caminar hacia otro lugar e interactuar con una máquina [3]. Chagall, artista y colaboradora de Imogen Heap, hablando de Imogen Heap, agregó que otra de las razones para el desarrollo de *wearables* era que sentía que no siempre era claro para la audiencia que es lo que estaba sucediendo en la performance cuando el artista debía ir a otro lado del escenario moviendo y presionando sliders y botones [4].

Por lo expuesto previamente, se puede afirmar que para un músico que se desarrolla en la industria de música electrónica, lo más difícil es tomar decisiones sobre cómo presentar su trabajo, ya que uno de los mayores problemas es la interacción del *performer* con su audiencia, esta puede verlo crear algo, pero no cómo es que lo está creando. Al estar detrás de mezcladoras de audio, pantallas o laptops el *performer* está interactuando con las pistas que usa para crear una nueva pero no interactúa con la audiencia de una forma directa, por lo que muchos *performers*, como por ejemplo los DJs se apoyan en algo visual, lo cual no es posible para todos los *performers* de música electrónica en vivo.

La presente tesis busca proponer una solución innovadora no solo para que los *performers* puedan interactuar directamente con su audiencia, sino también para que

encuentren una nueva manera de crear e interactuar con la música. Se plantea como solución así el diseño de un sistema con el objetivo de permitir al *performer* innovar en sus presentaciones mediante el uso de un *wearable* que reconozca sus gestos y permita reproducir archivos de audio previamente vinculados mediante un software, además, será de bajo costo y permitirá el fácil movimiento de las articulaciones de la mano. Asimismo, para llevar a cabo la investigación se realizará la revisión del estado de arte, definiendo los requerimientos de diseño. Luego, se desarrollará el concepto de solución óptimo para el diseño del proyecto basándose en los requerimientos, limitaciones u otros asociados, cumpliendo con normas y prevenciones de seguridad electrónica y de alergias dado que el guante estará en contacto con el usuario. Con respecto al diseño del guante se puede resaltar que el diseño electrónico se realizará de forma que sea compacto y ergonómico para el usuario, eligiendo los sensores u otros componentes que se adecuen mejor a los requerimientos, por otro lado, el diseño de los *enclosures* será de fácil ensamble para mantener el guante lo más compacto posible. Finalmente, se construirá un modelo de reconocimiento de gestos y se diseñará una interfaz donde el usuario pueda visualizar los datos de los sensores y realizar la vinculación de gestos con audios. En lo relativo a los costos, estos se basarán en precios del mercado local para fácil adquisición de repuestos y considerando costos de materiales y de mano de obra.

En cuanto a la limitación y alcances de la tesis, esta no incluye la fabricación de un prototipo del guante, ni la implementación de los softwares descritos. Se limitará la implementación del sistema de interpretación de gestos a reconocer gestos de una base de datos obtenida de un repositorio. Solo se implementarán ciertos gestos correspondientes a la base de datos usada. De este modo, el modelo de reconocimiento de gestos para una base de datos proveniente de guante y la vinculación de los gestos que el modelo reconoce con archivos de audio serán trabajados en una investigación posterior. Cabe indicar también que todos los sistemas a diseñar y plantear cumplirán con la norma IEC 61508 “Seguridad funcional de sistemas electrónicos eléctricos / electrónicos / programables relacionados con la seguridad”.

La tesis está organizada en seis capítulos, comenzando por el capítulo antecedentes, donde se presentará inventos, productos, patentes, artículos y trabajos de tesis relacionados al reconocimiento de gestos. En el segundo capítulo se elaborará el

diseño conceptual, para lo cual se realizará la lista de requerimientos y la estructura de funciones, en las cuales se basarán y desarrollarán tres conceptos de solución, luego se realizará una evaluación técnico-económica para elegir el diseño conceptual óptimo. En el tercer capítulo se realizará la integración del sistema y se desarrollará el diseño electrónico, mecánico y del software basándose en el concepto óptimo elegido previamente. Aquí se presentarán diagramas esquemáticos, PCB, planos, moldes y diagramas de flujo. En el cuarto capítulo, llamado reconocimiento de gestos de manos, se realizará la construcción del modelo de reconocimiento de gestos. Precediendo al capítulo final se calcularán los costos del sistema. Por último, se especificarán las conclusiones y recomendaciones.



CAPÍTULO 1

ANTECEDENTES

Este capítulo comprende la investigación de antecedentes, la cual se divide en antecedentes tecnológicos, los cuales abarcan productos y proyectos; y antecedentes técnicos, que abarcan patentes, artículos y trabajos de tesis; todos relacionados al reconocimiento de gestos.

1.1. Antecedentes tecnológicos

En esta sección se presentarán productos y proyectos relacionados a identificar movimientos de las manos.

a. Theremin [5]

Inventado por Leon Theremin en 1920 y patentado en 1928. Es uno de los primeros instrumentos musicales electrónicos que se toca por medio de gestos sin necesidad de contacto físico con el mismo. La sección de control del instrumento usualmente consta de 2 antenas metálicas que detectan la posición relativa de las manos y osciladores para controlar la frecuencia con una mano y el volumen con la otra. Las señales emitidas por el instrumento se amplifican y se derivan a un altavoz. Actualmente, el precio del instrumento varía entre \$100 y \$600.

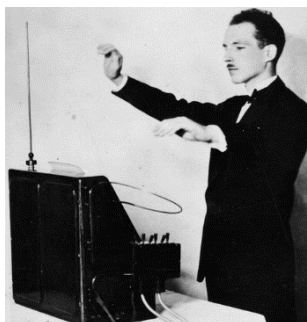


Figura 1.1: Leon Theremin tocando su instrumento: El theremin
Fuente:[6]



Figura 1.2: Theremin actual
Fuente:[7]

b. Hasbro's "GrooveIt" [8]

Producto diseñado por Hasbro que rastrea los movimientos de las manos de los jugadores cuando estos pasan por sensores infrarrojos en una serie de patrones llamados "Groove Moves". Los jugadores aprenden una variedad de movimientos y no alcanzan el estado de experto hasta que puedan seguir al menos 250 movimientos consecutivos. El precio de este producto es de \$44.



Figura 1.3: Hasbro's "GrooveIt" diseñado por KID
Fuente: [8]

c. The Sayre Glove [9]

En 1977, Daniel J. Sandin and Thomas Defanti crearon el *Sayre Glove* en el *Electronic Visualization Laboratory* en la Universidad de Illinois en Chicago. Fue el primer *data glove*. El guante estaba basado en la idea de Richard Sayre: un guante económico y ligero que monitoree los movimientos de la mano, suministrando un método efectivo para control multidimensional, como por ejemplo imitar un set de sliders. El guante usaba sensores a base de luz con tubos flexibles, una fuente de luz en un extremo y

una fotocélula en el otro. Cuando los dedos eran flexionados, la cantidad de luz que llegaba a las fotocélulas variaba, proporcionando de ese modo una medida de flexión de los dedos.



Figura 1.4: Sayre Glove
Fuente: [10]

d. *Lady's Glove* [11]

Desarrollado en STEIM, centro holandés para la investigación del rendimiento electrónico en Amsterdam, y creado por Laetitia Sonami, artista de sonido, *performer* e investigadora. Es uno de los modelos más avanzados para controlar gestos. Es un sistema que cuenta con sensores de flexión para medir la inclinación de ambas articulaciones para los 3 dedos medios de la mano. Además, cuenta con microswitches al final de los dedos para control táctil, sensores de efecto hall para medir la distancia entre los dedos y un imán en el pulgar. Con los guantes, Sonami puede crear performances en las cuales sus movimientos pueden “moldear” la música y en ocasiones entornos visuales.



Figura 1.5: Laetitia Sonami usando el Lady's Glove
Fuente: [11]

e. *Mi.mu Gloves* [12]

Desarrollado por el equipo Mi.mu, encabezado por Imogen Heap, cantautora y productora musical, quien describe a los *Mi.mu gloves* como una fusión de textiles y sensores electrónicos, combinado con detección de gestos que ofrecen una nueva propuesta para el control de la música con movimientos intuitivos. La última versión de estos guantes cuenta con 8 sensores de flexión, un motor de vibración y un led RGB que le da *feedback* háptico al usuario y que puede usar como metrónomo. Cuentan con un IMU para detección de movimiento y tarjeta x-OSC, ambos desarrollados por x-io Technologies, compañía la cual es dirigida por Sebastian Madgwick quien también es parte del equipo mi.mu. El precio por un guante es de £2500.



Figura 1.6: *Mi.mu Gloves*
Fuente: [12]

f. *5DT Data Glove* [13]

El *5DT Data Glove Ultra* está diseñado para satisfacer los estrictos requisitos de los modernos profesionales de captura de movimiento y animación. La alta calidad de captación de datos, la baja correlación cruzada y la alta velocidad de captación de datos lo hacen ideal para la animación realista en tiempo real. El sistema se conecta con la computadora a través de un cable USB. Cuenta con una resolución de flexión de 10 bits, comodidad extrema, baja deriva y una arquitectura abierta. El kit *5DT Data Glove Ultra Wireless* se conecta con la computadora a través de la tecnología bluetooth (hasta 20 m de distancia) para una conectividad de alta velocidad de hasta 8 horas con una sola batería. El precio por un solo guante varía entre \$995 y \$5495.



Figura 1.7: 5DT Data Gloves
Fuente: [13]

En la tabla 1.1 se puede observar la comparación de las principales características de los modelos revisados previamente. Se detalla la facilidad de portabilidad, el funcionamiento y costo de cada modelo, dependiendo si es un producto o proyecto.

Tabla 1.1: Comparación de modelos para identificación de movimientos de las manos
Fuente: Propia

Modelo	Facilidad de portabilidad	Funcionamiento	Producto/ Proyecto	Costo (\$)
Theremin	Baja	Cambio de frecuencias	Producto	100 – 600
Hasbro's "GrooveIt"	Alta	Sensores infrarrojos	Producto	44
The Sayre Glove	Alta	Por guante (sensores)	Proyecto	-
Lady's Glove	Alta	Por guante (sensores)	Proyecto	-
Mi.mu Gloves	Alta	Por guante (sensores)	Producto	3300
5DT Data Glove	Alta	Por guante (sensores)	Producto	995 -5495

1.2. Antecedentes técnicos

Esta sección se divide en patentes y artículos técnicos relacionados a identificación de gestos.

1.2.1. Patentes

a. *Digital data entry glove interface device* [14]

Inventor: Gary J. Grimes

Patente: US4414537A

Fecha: 08/11/1983

Una interfaz máquina - hombre capaz de interpretar posiciones básicas de las manos en señales eléctricas que representan caracteres alfanuméricos. La interfaz consiste de un guante que contiene sensores para detectar la flexión de las articulaciones de los dedos y sensores para detectar el contacto entre varias partes de la mano. Sensores adicionales detectan el movimiento de la mano con respecto a un vector gravitacional y a un plano horizontal de referencia. Aún más sensores se encuentran instalados para detectar el giro y flexionamiento de la muñeca.

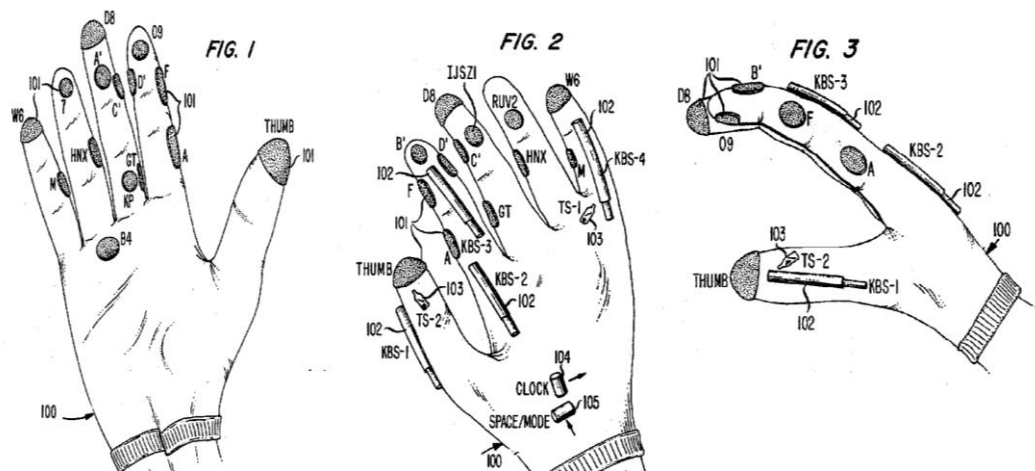


Figura 1.8: Diseño de digital entry data gloves
Fuente: [14]

b. *Virtual reality glove system with fabric conductors* [15]

Inventor: Ian E. McDowall, Mark T. Bolas, Russell C. Mead, Jr., Christian Greuel

Patente: US6128004A

Fecha: 03/10/2000

Un guante de entrada de datos diseñado para ser usado con un sistema de computadora o un sistema de realidad virtual, con numerosos electrodos colocados en el guante e interconectados con controles electrónicos que permiten la detección de contacto entre cualquier combinación de electrodos, los electrodos están hechos de tela flexible y pintura eléctricamente conductora.

FIG. 1

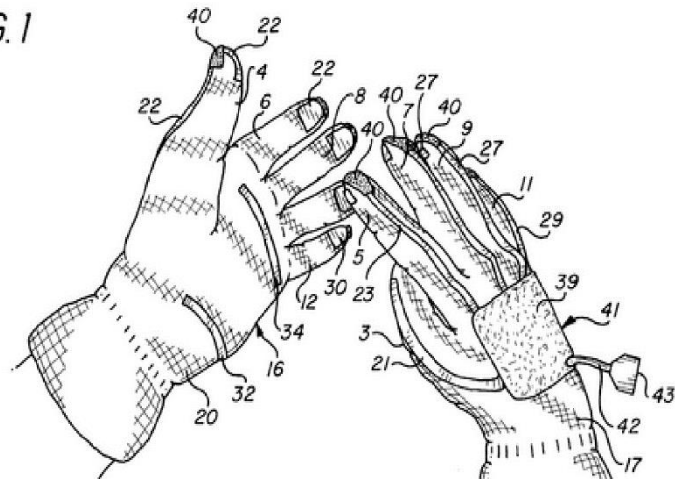


Figura 1.9: Diseño de Virtual reality glove system with fabric conductors
Fuente: [15]

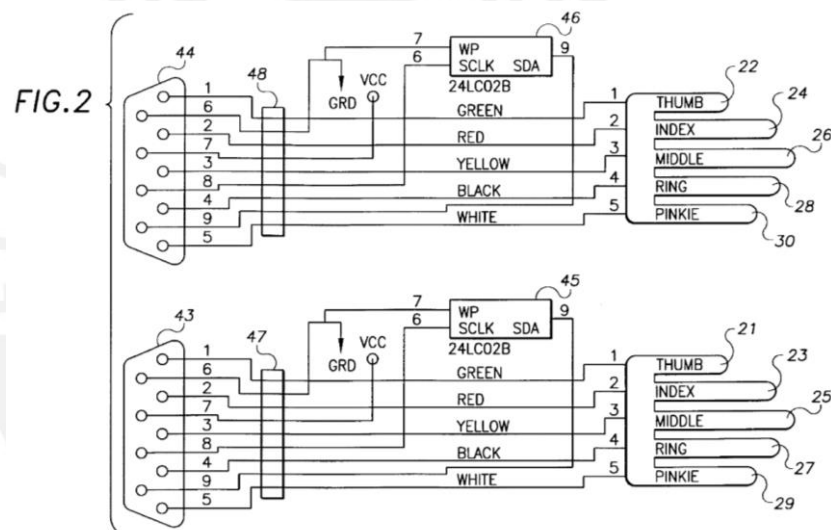


Figura 1.10: Esquemático del cableado de los guantes
Fuente: [15]

1.2.2. Artículos técnicos y tesis

a. *Gesture Controlled Musical Instrument* [16]

Artículo realizado por Sree Ganeshji, Mohammed Irshad, Harshitha K.L, Tunga R, Ranjan S Rao en el año 2015. Este artículo explora la factibilidad de un instrumento musical incorporando procesamiento de señales digitales combinado con SONAR (acrónimo de *Sound Navigation And Ranging*) para producir notas musicales, emulando instrumentos convencionales con uso de hardware de bajo costo.

b. *Smart Glove: An Assistive Device to Enhance Recovery of Hand Function During Motor Rehabilitation* [17]

Tesis realizada por Smrithi Sasidharan en el año 2015. Guante que ayuda a recuperar la movilidad de la mano en pacientes que requieran rehabilitación. El modo en el que se capturan datos es mediante sensores de flexión que se insertan en los bolsillos cosidos en el guante. Hay 2 tipos de sensores implementados en el diseño, uno para detectar la flexión de los dedos, el otro para detectar la flexión de la muñeca.

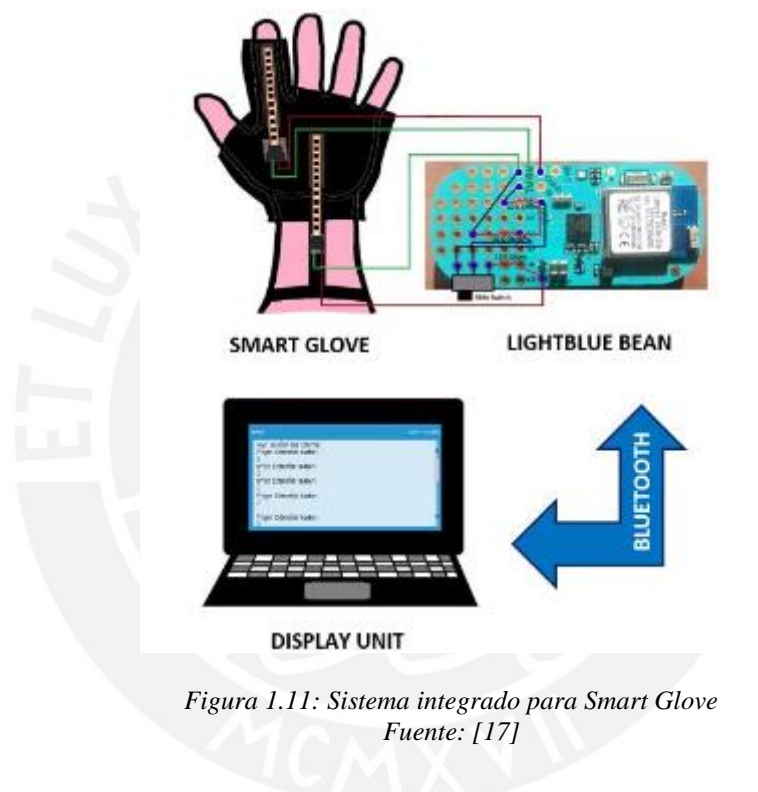


Figura 1.11: Sistema integrado para Smart Glove
Fuente: [17]

c. *Finger motion sensors for fMRI motor studies* [18]

Artículo publicado por NeuroImage en el año 2006, que propone el uso de sensores MEMS para estudios de motores de imágenes de resonancia magnética funcional (fMRI). Se demuestra la utilidad de un dispositivo basado en sensores MEMS a través de experimentos que examinan la relación entre la cinemática del movimiento de los dedos y la activación de imágenes de resonancia magnética funcional en cerebros sanos y heridos. En la figura 1.12 se puede observar el esquemático del dispositivo con sus partes componentes. Se utilizaron giroscopios de un solo chip de dimensiones 7mm x 7mm x 3mm y de peso menor a 0.5 g. En la figura 1.13 se muestran los sensores

posicionados en la mano de un sujeto. Cada sensor fue montado en una plumilla de guitarra remodelada, de modo que cuando se sujeta a la parte dorsal del dedo el eje z de detección de velocidad sea perpendicular al eje que pasa por el largo del dedo. Esta posición permite la detección de la velocidad angular durante la flexión y extensión de la articulación metacarpofalángica del dedo.

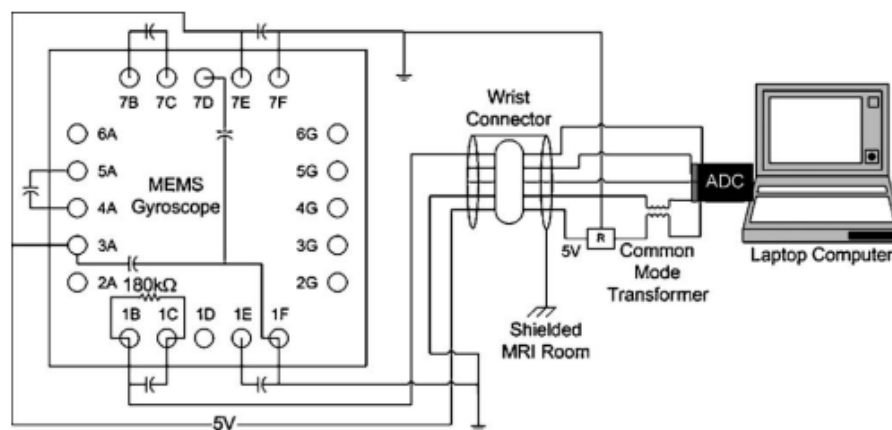


Figura 1.12: Esquemático del dispositivo basado en sensores MEMS
Fuente: [18]



Figura 1.13: Sensores MEMs asegurados en mano de sujeto
Fuente: [18]

d. *EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing* [19]

Artículo respaldado por *EPSRC Doctoral Training*, publicado en el año 2016, que propone el uso de sensores EMG y sensores de fuerza para clasificar gestos de la mano. Este enfoque detecta ambas articulaciones de los dedos y rotación de la muñeca, cubriendo un gran rango de gestos, con una precisión de clasificación de validación cruzada de 10 iteraciones del 96%. Se demuestra que los sensores EMG se adaptan especialmente a detectar los movimientos de los dedos, mientras que los sensores de

presión de fuerza se adaptan más a detectar las rotaciones de la muñeca y antebrazo, y su combinación es significativamente más precisa que cualquiera de los dos métodos por si solos. En la figura 1.14 se puede ver la sección transversal del antebrazo y la posición de los sensores EMG y de fuerza situados en la muñeca. Además, en la figura 1.15 se muestra un prototipo de la propuesta colocado en un sujeto. Por otro lado, se propone usar el método *SVM (Support vector machine)* para el reconocimiento de gestos.

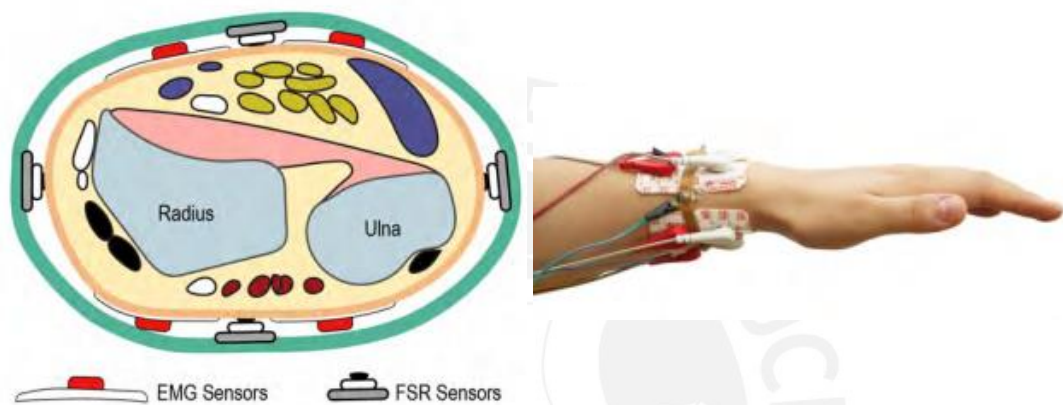


Figura 1.14: Sección transversal del antebrazo en los extremos distales del radio y cúbito.
Fuente: [19]

Figura 1.15: Prototipo usado alrededor de la muñeca
Fuente: [19]

- e. *A Segmented-Mean Feature Extraction Method for Glove-based System to Enhance Physiotherapy for Accurate and Speedy Recuperation of Limbs* [20]

Artículo publicado en 2016, presentado en la Conferencia Internacional en Avances en Informática, Comunicaciones e Informática (*ICACCI*). Se propone un algoritmo de construcción de características con enfoque en el método *Segmented-Mean* para señales adquiridas por un guante *wearable* que consiste en sensores de flexión. Con el propósito de ayudar a la construcción de características de manera más fácil que otros métodos convencionales, reduciendo la complejidad computacional y por ende el tiempo.

Al concluir este capítulo, se alcanza un entendimiento de los antecedentes tecnológicos y técnicos de la presente tesis. En el siguiente capítulo, se desarrollará el diseño conceptual del proyecto.

CAPÍTULO 2

DISEÑO CONCEPTUAL

En el presente capítulo se describe los requerimientos, estructuras de funciones, matriz morfológica y conceptos de solución.

2.1. Requerimientos del sistema guante inteligente

En el presente subcapítulo se describe los requerimientos del sistema, para los cuales se toma como base los antecedentes y las necesidades del usuario expuestos en el capítulo I. La lista de requerimientos se divide en general, donde se definen los requerimientos principales para cumplir con las necesidades específicas del usuario; mecánico, donde se detalla la geometría, cinemática y uso del sistema; electrónico y de control, donde se detalla los requerimientos de los sensores a emplear en el sistema, así como algoritmos que se emplearan en el sistema. La lista con todos los requerimientos detallados se encuentra en el Anexo A, donde se indica con una D o E si el requerimiento en un deseo o exigencia, al igual que en esta sección.

2.1.1. Requerimiento general

El requerimiento funcional del sistema es reproducir música a través de gestos de la

mano, interactuando con un software de música, y de este modo crear nuevas pistas de música.

2.1.2. Requerimientos mecánicos

Los requerimientos mecánicos se basan en requerimientos de geometría, cinemática, uso, seguridad, ergonomía, materiales, fabricación, montaje y mantenimiento.

- Para el óptimo desempeño de los guantes, estos deben ser diseñados para cada usuario, sin embargo, se diseñará guantes con dimensiones para un usuario promedio. Así, se toma en cuenta un tamaño de palma promedio de 80 mm y un tamaño de mano promedio de 170 mm. (E)
- Para que el sistema sea compacto, se diseñarán *enclosures* con las siguientes medidas máximas: ancho no mayor a 45 mm, altura no mayor a 60 mm y profundidad no mayor a 20 mm. (E)
- Los movimientos articulares de los dedos de una mano se clasifican en: abducción/aducción y flexión/extensión, de las articulaciones interfalángicas relacionadas a estos movimientos se enfocará principalmente en las articulaciones metacarpofalángicas, dejando de lado las articulaciones interfalángica distal e interfalángica proximal, como se explica en la figura 2.1. Además, no se tomarán en cuenta los movimientos de abducción/ aducción para la articulación mencionada. Por otro lado, los movimientos articulares de la muñeca (figura 2.2) se clasifican en: abducción/aducción, extensión/flexión y supinación/pronación, correspondientes a las rotaciones *yaw*, *pitch* y *roll*, respectivamente; todos los movimientos de la muñeca mencionados se tomarán en cuenta. El sistema contará con 1 GDL por cada dedo y 3 GDL en la muñeca. (E)



Figura 2.1: Articulaciones interfalángicas
Fuente: Propia

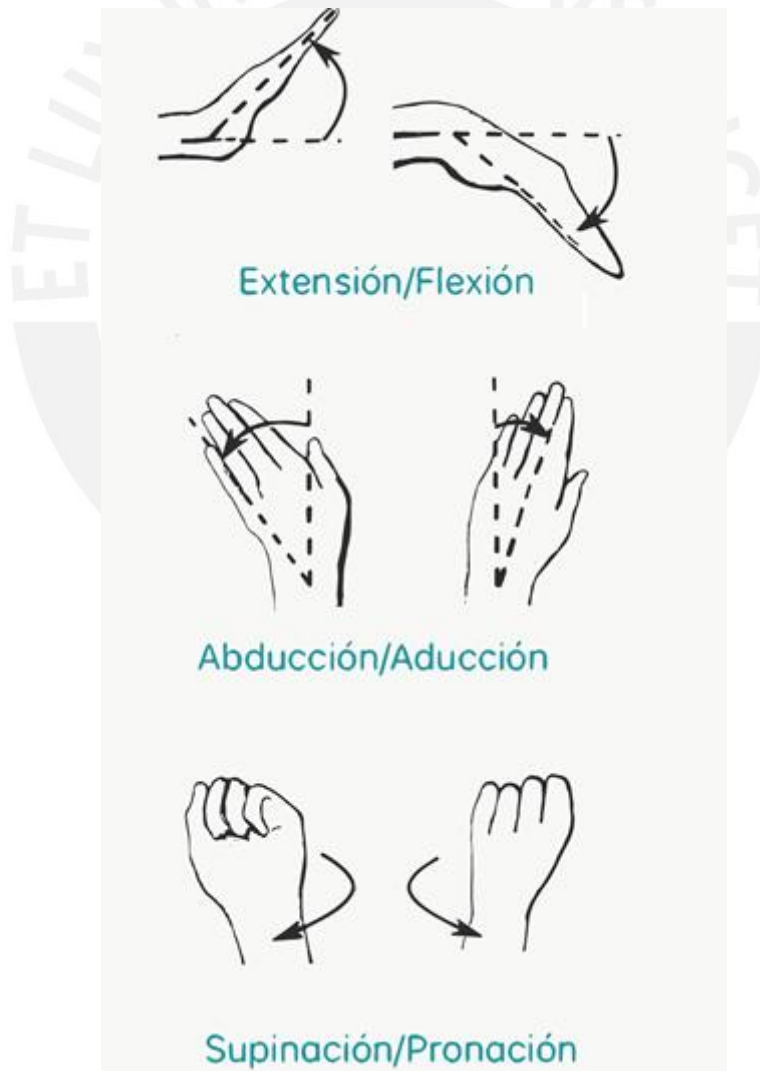


Figura 2.2: Movimientos articulares de la muñeca
Fuente: Propia

- El sistema se podrá usar entre 10°C y 35°C, ya que este es el rango promedio de temperatura en Perú. Además, se deberá mantener alejado del agua, por lo que se podrá usar solo en espacios abiertos mientras no llueva. (E)
- El sistema deberá ser de fácil desmontaje y montaje de partes electrónicas para que el guante pueda ser lavado cuando el usuario lo desee y luego pueda volver a ensamblar las partes por su cuenta. (E)
- El sistema se deberá mantener alejado del agua por lo que se deberá aislar y proteger la parte eléctrica del dispositivo en consideración a la norma IEC 61508 “Seguridad funcional de sistemas electrónicos eléctricos / electrónicos / programables relacionados con la seguridad”. (E)
- El dispositivo permitirá el fácil movimiento de las articulaciones de la mano del usuario sin restringirlo o hacer que esté se sienta incómodo. (E)
- Las piezas electrónicas del sistema deberán ser adquiridas en el mercado local para así tener fácil acceso a repuestos. (D)
- Se deberá diseñar un *enclosure*/case óptimo, de modo que el sistema electrónico sea compacto. Dicho *enclosure* será impreso en 3D para optimizar costos. El material que se use para la impresión de los *enclosures* deberá ser resistente a altas temperaturas, ya que las partes electrónicas tienden a calentarse. (E)
- Se utilizarán materiales que puedan estar en contacto con la piel por un largo tiempo sin que el usuario se sienta incómodo. Dichos materiales deberán ser no alérgicos, no haber sido procesados con agentes tóxicos y permitirán la dispersión natural de calor, flujo de aire y la liberación de calor para apropiada ventilación, cumpliendo con las normas ISO 9237 “Textiles - Determinación de la permeabilidad de las telas al aire” e ISO 11092 “Textiles - Efectos fisiológicos - Medición de la resistencia térmica y al vapor de agua en condiciones de estado estacionario”. (D)
- La limpieza de los materiales no eléctricos del dispositivo deberá ser realizada periódicamente debido a que están en contacto continuo y directo con la piel. (E)
- El sistema contará con dos tipos de sensores (capturar movimiento de articulación de dedos y capturar orientación de la muñeca), los cuales podrán ser reemplazados por repuestos con facilidad en caso de ser necesario. (E)

2.1.3. Requerimientos electrónicos y de control

Los requerimientos electrónicos y de control se basan en requerimientos de señales, control, energía, interfaz, electrónica y comunicación.

- El circuito debe ser alimentado por baterías recargables que cumplan con las condiciones para que el sistema sea compacto. El voltaje de alimentación para los microcontroladores más comerciales está entre 3.3 y 5 V, por lo cual en el caso que la batería elegida no llegue a ese voltaje se deberá usar un regulador *step-up*. (E)
- El sistema deberá contar con sensores para detectar movimiento de las articulaciones de los dedos, un sensor para detectar el movimiento y orientación de la muñeca y mano y un módulo *wireless* para transmitir datos entre dispositivo y software. (E)
- La comunicación entre el dispositivo y la PC se realizará por un módulo Wi-Fi o bluetooth, ya que se busca que el usuario no tenga la necesidad de conectar el dispositivo (guante) a una PC mientras está haciendo uso de él. (E)
- El sistema tendrá como señales de entrada los gestos que el usuario realice y como señales de salida el audio vinculado a dicho gesto. Por otro lado, tendrá como señales intermedias los ángulos de movimiento de las articulaciones metacarpofalángicas (dedos) y los movimientos articulares de la muñeca (rotaciones *yaw*, *pitch* y *roll*). Además, también tendrá como entrada los archivos de audio que el usuario guarde en el sistema. (E)
- El sistema deberá tener un algoritmo de procesamiento de datos obtenidos de sensores y algoritmo de reconocimiento de gestos. (E)
- El sistema contará con un algoritmo de control de parámetros de sonido (intensidad, altura, timbre) para procesar audio, y este será adaptable según el tipo de gesto. (D)
- El sistema tendrá una interfaz donde el usuario podrá ver los datos provenientes de los sensores del sistema y donde podrá realizar la vinculación de tipo de gestos con parámetro de procesamiento de audio. (D)

2.1.4. Requerimientos adicionales

- El costo máximo del sistema deberá ser \$950, basándose en los costos de dispositivos similares que existen en el mercado. (D)

2.2. Estructura de funciones

De acuerdo a lo establecido en la lista de requerimientos, se determina la estructura de funciones del sistema. Dicha estructura se divide en: mecánica, interfaz, sensores, software, comunicación y energía. Seguidamente, se detalla cada uno de los bloques de la estructura de funciones. El diagrama de estructura de funciones completo puede ser revisado en el Anexo B.

En la figura 2.3 se muestra la caja negra (*black box*) del sistema, cuyas entradas son base de datos de gestos, entradas de audio, energía y mano (gestos). Por otro lado, las salidas son los datos de sensores que se mostraran en la interfaz, el audio seleccionado que se obtiene cuando se realiza un gesto que sea correspondiente a este. Por último, la mano con captura de movimiento también se considera como una salida y USUARIO como una entrada.

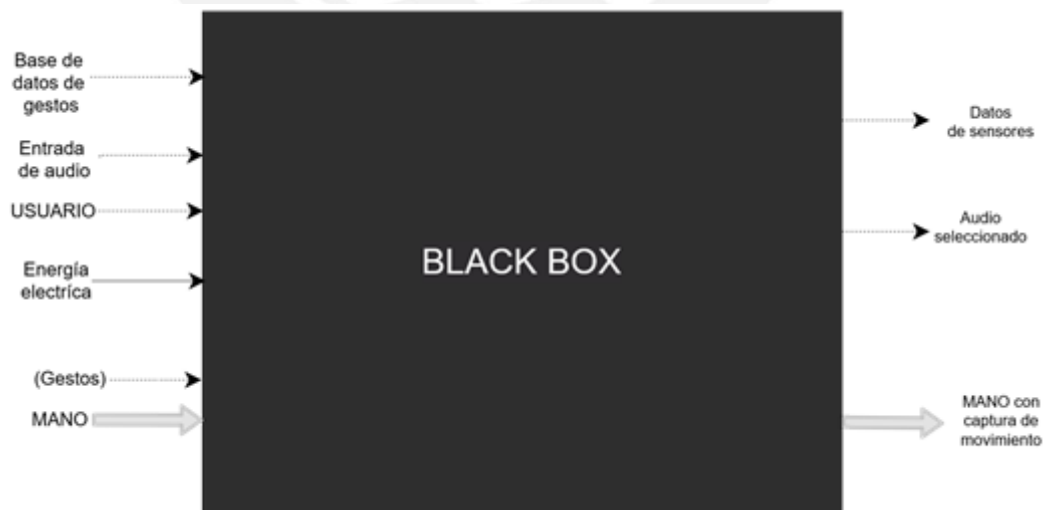


Figura 2.3: Black Box
Fuente: Propia

Energía

En este bloque (figura 2.4) se detalla como los sensores, microcontroladores y módulos de comunicación serán energizados y se indica la regulación de voltaje respectiva para cada componente. La energía del sistema provendrá de baterías, ya que este debe ser portable. Luego de esto, se dirige a los bloques control, sensores y comunicación.

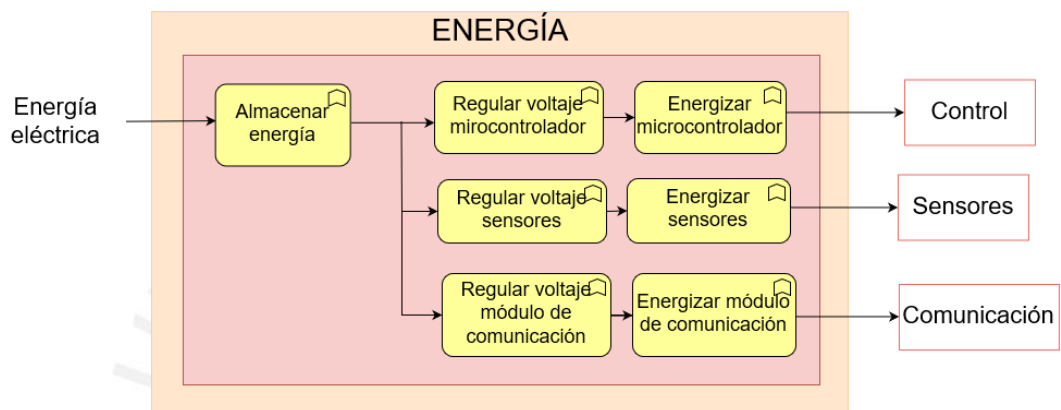


Figura 2.4: Bloque de Energía
Fuente: Propia

Comunicación

En el bloque de comunicación (figura 2.5) se transmiten los datos pre-procesados (encapsulados), provenientes del microcontrolador, a la PC mediante un módulo *wireless*.

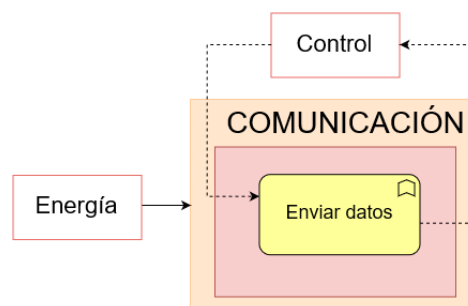


Figura 2.5: Bloque de Comunicación
Fuente: Propia

Interfaz

En el bloque interfaz (figura 2.6) se registran los datos procesados provenientes del microcontrolador (datos de sensores y estado de batería del sistema), los datos provenientes de los algoritmos de reconocimiento de gestos (datos de gestos) y los archivos de audio guardados, para luego ser mostrados en la interfaz.

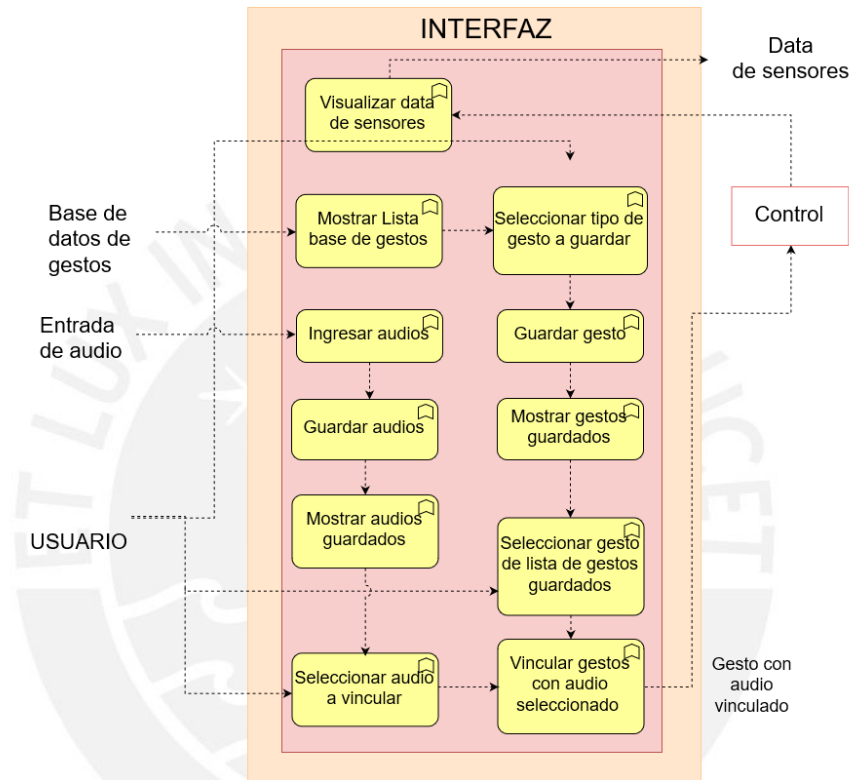


Figura 2.6: Bloque de Interfaz
Fuente: Propia

Sensores

En este bloque (figura 2.7) se detallan las funciones de los sensores que son requeridos para el funcionamiento del sistema. Estos se encargan de medir los ángulos de las articulaciones de los dedos y medir orientación de la muñeca. Los datos mencionados anteriormente dependen del gesto que realice el usuario. Por último, los datos obtenidos deberán ser pre-procesados (encapsulamiento de datos), motivo por el cual se dirigen al bloque de control.

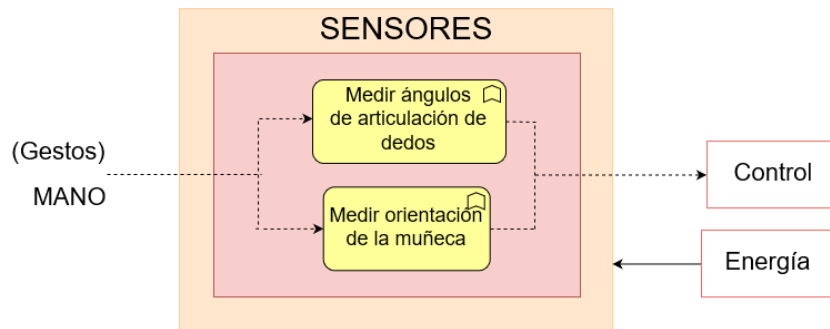


Figura 2.7: Bloque de Sensores
Fuente: Propia

Software

En el bloque de software (figura 2.8) se realiza el encapsulamiento de datos, el reconocimiento de gestos y la selección de audio. El encapsulamiento de datos se hace en el microcontrolador donde entran los datos provenientes de los sensores y se obtienen los datos encapsulados, los cuales entran al bloque de comunicación, para luego ser recibidos en PC, como se indica en la sección de reconocimiento de gestos. Seguidamente, se dirigen al bloque de extracción de características, donde se obtiene la matriz de características para luego clasificar el gesto. Una vez que se obtiene el gesto reconocido, este se dirige al bloque de selección de audio, donde el software selecciona el audio vinculado a dicho gesto.

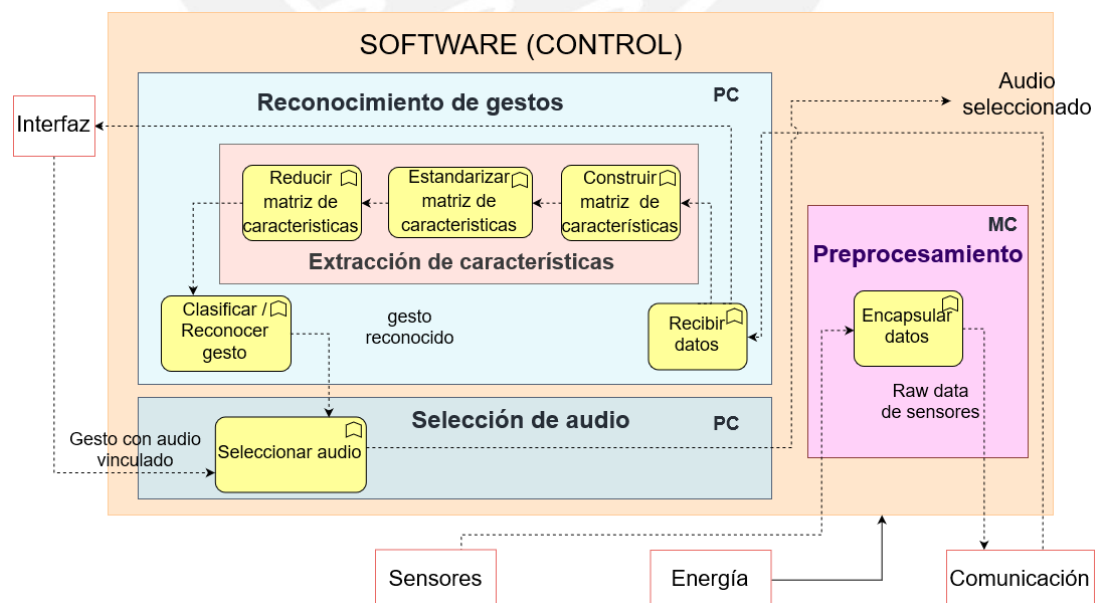


Figura 2.8: Bloque de Software (Control)
Fuente: Propia

Mecánica

En el bloque mecánica (figura 2.9) se describe las funciones de albergar, soportar y proteger los sensores y electrónica del sistema (*enclosure*). La entrada al bloque es mano (gesto), esta se dirige a la función de capturar gesto para obtener mano con captura de movimiento.

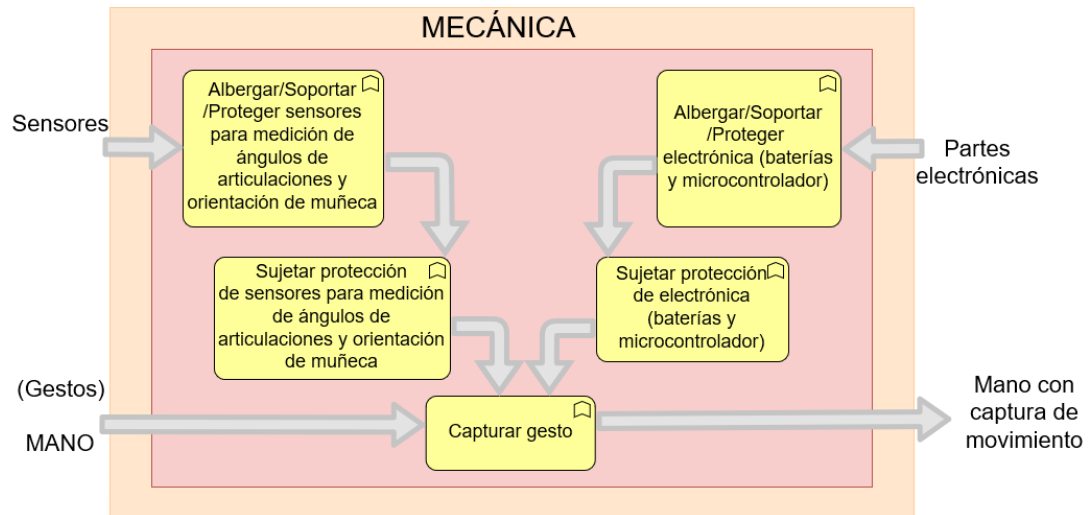


Figura 2.9: Bloque de Mecánica
Fuente: Propia

2.3. Matriz morfológica

Realizada la estructura de funciones, se procede a elaborar la matriz morfológica analizando qué componentes son los más adecuados para efectuar las funciones de los dominios de energía, sensores, control, comunicación y mecánica. Seguidamente, en base a la matriz morfológica se determina 3 alternativas de concepto de solución para el sistema, escogiendo una opción para cada función mencionada en la matriz. Los conceptos de solución se encuentran detallados e ilustrados posteriormente.

Tabla 2.1: Leyenda – Soluciones para matriz morfológica
Fuente: Propia

Concepto de solución 1	— — — — — →
Concepto de solución 2	- - - - - →
Concepto de solución 3 →

Tabla 2.2: Matriz morfológica – Energía
Fuente: Propia


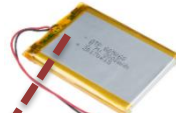
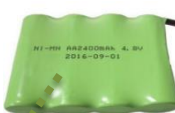



Tipo	Función	Alternativa 1	Alternativa 2	Alternativa 3
Energía	Almacenar energía	2S Baterías de ion-litio (7.4V) 	Batería LiPo (3.7V) 	4 Baterías Ni-MH (1,2V) 
	Regular voltaje (microcontrolador/sensores/módulo de comunicación)	Boost Converter 	MOSFET 7805 	
	Energizar componentes (microcontrolador/sensores/módulo de comunicación)	Cables y conectores de baterías 		

Tabla 2.3: Matriz morfológica – Sensores
Fuente: Propia







Tipo	Función	Alternativa 1	Alternativa 2	Alternativa 3
Sensores	Medir ángulos de articulación de dedos	Electrodos EMG 	Sensores de flexión 	MEMS giroscopio 
	Medir orientación de la mano	Sensores de fuerza 	Giroscopio y Acelerómetro 	IMU 

Tabla 2.4: Matriz morfológica – Control
Fuente: Propia



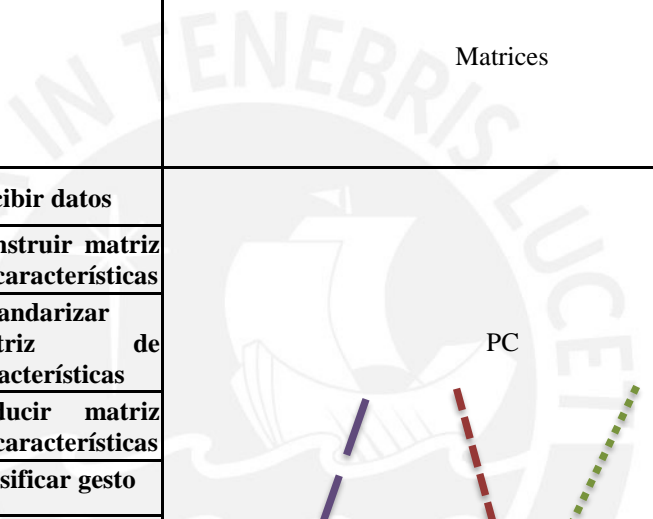
Tipo		Función	Alternativa 1	Alternativa 2	Alternativa 3	
Control	Pre procesamiento	Hardware	Encapsular datos Arduino Pro micro (5V) 	Encapsular datos Arduino Nano (5V) 		
		Software	Encapsular datos			
	Reconocimiento de gestos y Selección de audio	Hardware	Recibir datos	PC 		
			Construir matriz de características			
			Estandarizar matriz de características			
			Reducir matriz de características			
			Clasificar gesto			
			Seleccionar audio			
		Software	Recibir datos	Bluetooth	Wi-Fi	
		Construir matriz de características	IEMG, VAR, Kutorsis, etc	Segmented – mean, mean, VAR, etc	Scale Invariant Feature Transform (SIFT)	
		Estandarizar matriz de características	Método de estandarización $x' = \frac{x - \bar{x}}{\sigma}$			
		Reducir matriz de características	Principal Component Analysis (PCA), métodos de correlación			
Clasificar gesto	Linear and Discriminant Analysis (LDA)	Support Vector Machine (SVM)	Random Forest			
Seleccionar audio	Algoritmo de seleccion por vinculacion					

Tabla 2.5: Matriz morfológica – Comunicación
Fuente: Propia



Tipo	Función	Alternativa 1	Alternativa 2	Alternativa 3
	Enviar datos	Módulo bluetooth 	Módulo Wi-Fi 	

Tabla 2.6: Matriz morfológica – Mecánica
Fuente: Propia

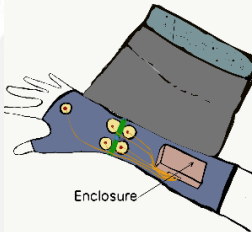
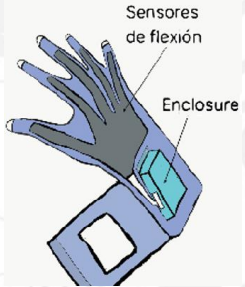
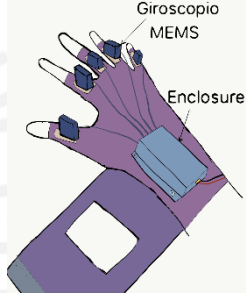
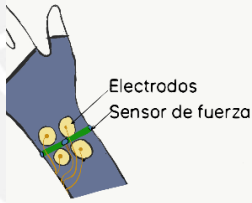
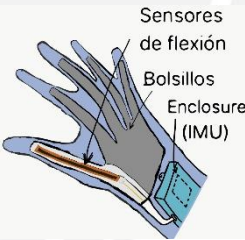
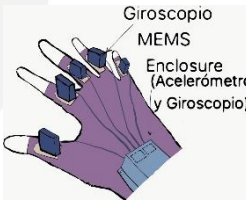
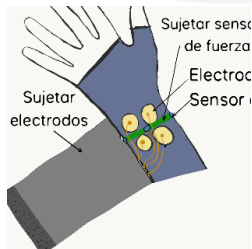
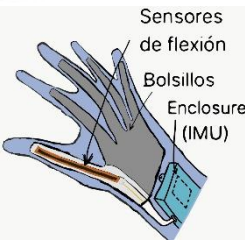
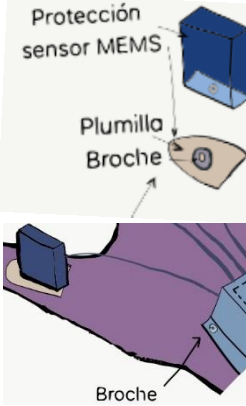
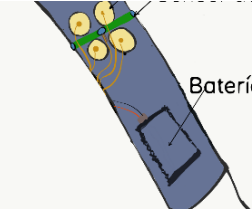
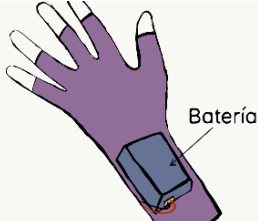
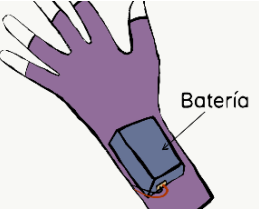
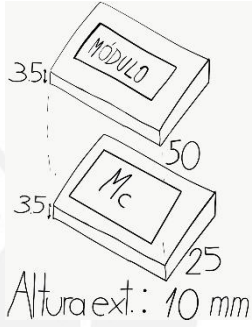
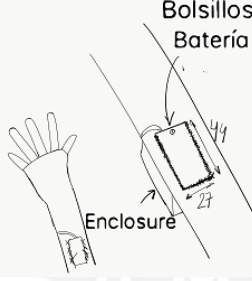
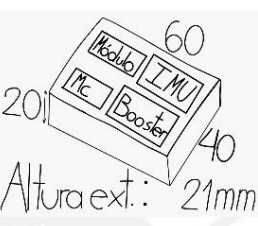
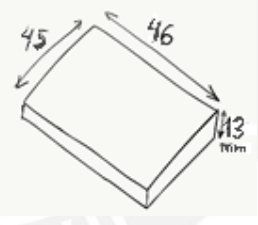
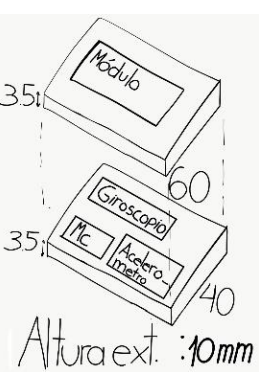
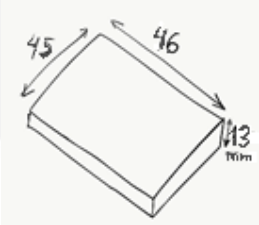
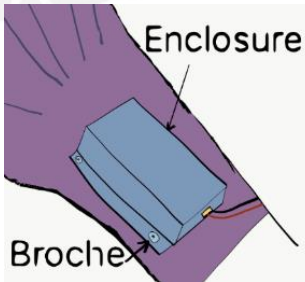
Tipo	Función	Alternativa 1	Alternativa 2	Alternativa 3
	Capturar gesto		Sensores de flexión Enclosure 	Giroscopio MEMS Enclosure 
	Albergar sensores para medición de ángulos y orientación de muñeca	Electrodos Sensor de fuerza 	Sensores de flexión Bolsillos Enclosure (IMU) 	Giroscopio MEMS Enclosure (Acelerómetro y Giroscopio) 
	Sujetar sensores para medición de ángulos y orientación de muñeca	Sujetar sensores de fuerza Electrodo Sensor Sujetar electrodos 	Sensores de flexión Bolsillos Enclosure (IMU) 	Protección sensor MEMS Plumilla Broche 

Tabla 2.7: Matriz morfológica – Mecánica continuación
Fuente: Propia

Tipo	Función	Alternativa 1	Alternativa 2	Alternativa 3
Mecánica	Albergar electrónica			
		<p data-bbox="643 622 754 651">Enclosure</p>  <p data-bbox="659 1059 738 1088">Batería</p> 	<p data-bbox="914 622 1026 651">Enclosure</p>  <p data-bbox="930 1070 1010 1099">Batería</p> 	<p data-bbox="1185 622 1297 651">Enclosure</p>  <p data-bbox="1201 1081 1281 1111">Batería</p> 
	Sujetar electrónica			

2.4. Conceptos de solución

Los conceptos de solución presentados a continuación están desarrollados en base a las alternativas propuestas en la matriz morfológica y los requerimientos del sistema. Posteriormente, se realizará una evaluación de cada uno de los conceptos para obtener la solución óptima.

2.4.1. Concepto de solución 1

En este concepto se propone el uso de electrodos y sensores de fuerza para detectar el movimiento de las articulaciones de los dedos y de la muñeca, los cuales irán colocados como se muestra en la figura correspondiente a la función albergar sensores de la matriz morfológica (Tabla 2.6). Se hará uso de una manga con orificios circulares, en los cuales se indique que electrodo va en cada orificio, para que así el usuario siempre sepa dónde colocarlos y cada canal siempre obtenga lecturas de la misma zona de la mano. Además, cabe resaltar que se hará uso de electrodos secos en lugar de electrodos flotantes para evitar que el usuario aplique pasta electrolítica cada vez que haga uso del dispositivo. Para evitar que estos se muevan se cubrirá la manga como se muestra en la figura 2.10, sosteniendo así todos los sensores en su lugar. Se propone que el *enclosure*, el cual almacena toda la electrónica, esté colocado al final del brazo para no interferir con las posiciones de los electrodos y sensores de fuerza. Dentro del *enclosure* estarán el microcontrolador y el módulo bluetooth, tendrá dos aberturas, una para conectar la batería al microcontrolador y alimentar el sistema, y otra para conectar los cables de los electrodos y sensores de fuerza y poder obtener los datos para el reconocimiento de gestos. Para alimentar el sistema se usarán baterías de ion litio 2s, las cuales no irán colocadas en el *enclosure*, si no en el reverso del brazo debido a su tamaño; la posición de la batería se muestra en la figura correspondiente a la función albergar electrónica de la matriz morfológica (Tabla 2.7). Por otro lado, a través del módulo bluetooth, los datos tomados de los sensores se envían a la PC, donde se construye la matriz de características y se clasifica el gesto mediante el método LDA, si el gesto realizado está registrado en el sistema, este reproducirá el archivo de audio correspondiente.

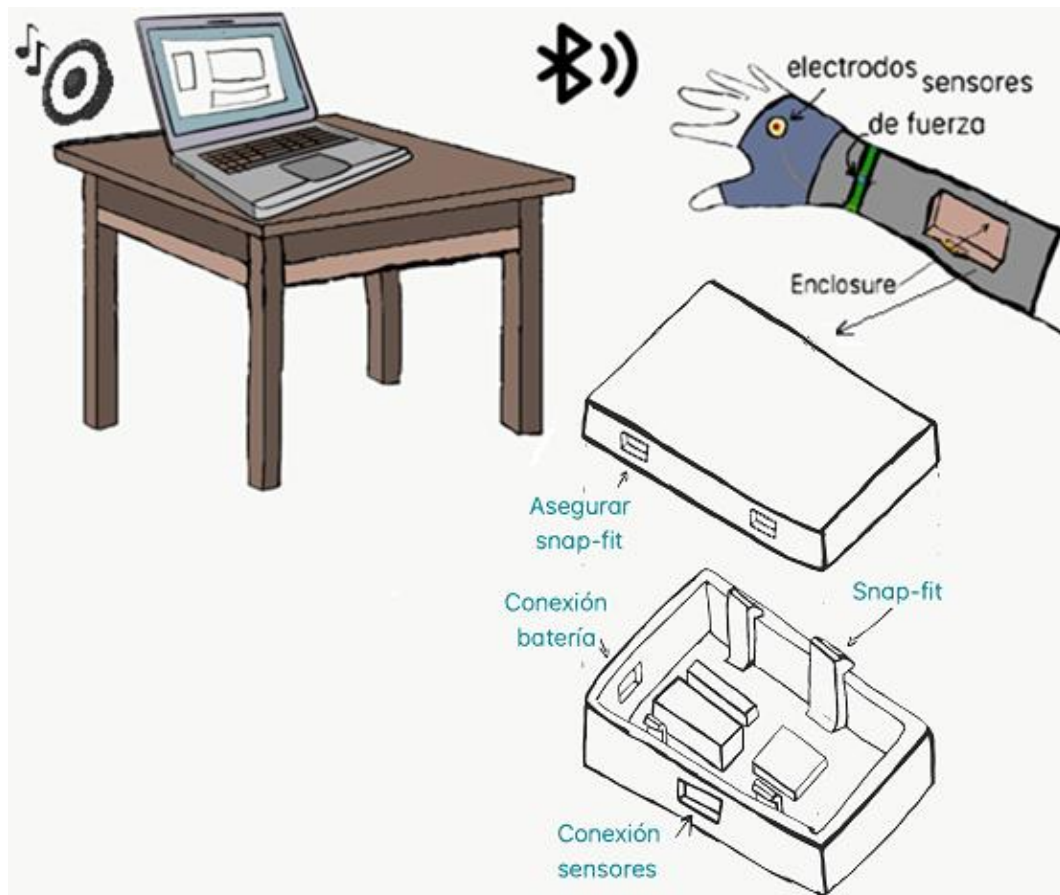


Figura 2.10: Bosquejo primer concepto solución
Fuente: Propia

2.4.2. Concepto de solución 2

En este concepto se propone usar sensores de flexión y una IMU para detectar el movimiento de las articulaciones de los dedos y de la muñeca. Los sensores de flexión irán colocados en unos *pockets* que estarán cocidos sobre el guante, posicionando los sensores por encima de las articulaciones de los dedos, como se muestra en la figura correspondiente a la función albergar sensores de la matriz morfológica (Tabla 2.6). Se diseñará dos *enclosures*, uno en el cual se encontrará el IMU, el microcontrolador, el módulo Wi-Fi y los reguladores de voltaje necesarios, y otro donde se colocará la batería LiPo. El *enclosure* que almacena la electrónica ira posicionado en la parte superior de la muñeca, mientras que el otro *enclosure* se colocara en el reverso de la muñeca, ambos tendrán prolongaciones, donde se colocaran broches para posicionarlos en el guante, como se puede ver en la figura 2.11. Además, las prolongaciones serán cubiertas por un sujetador alrededor de la muñeca para mayor

seguridad. El reconocimiento de gestos sigue un proceso similar al del concepto de solución 1, sin embargo, los datos tomados de los sensores se envían a la PC mediante un módulo Wi-Fi y los gestos se clasifican mediante el método SVM.

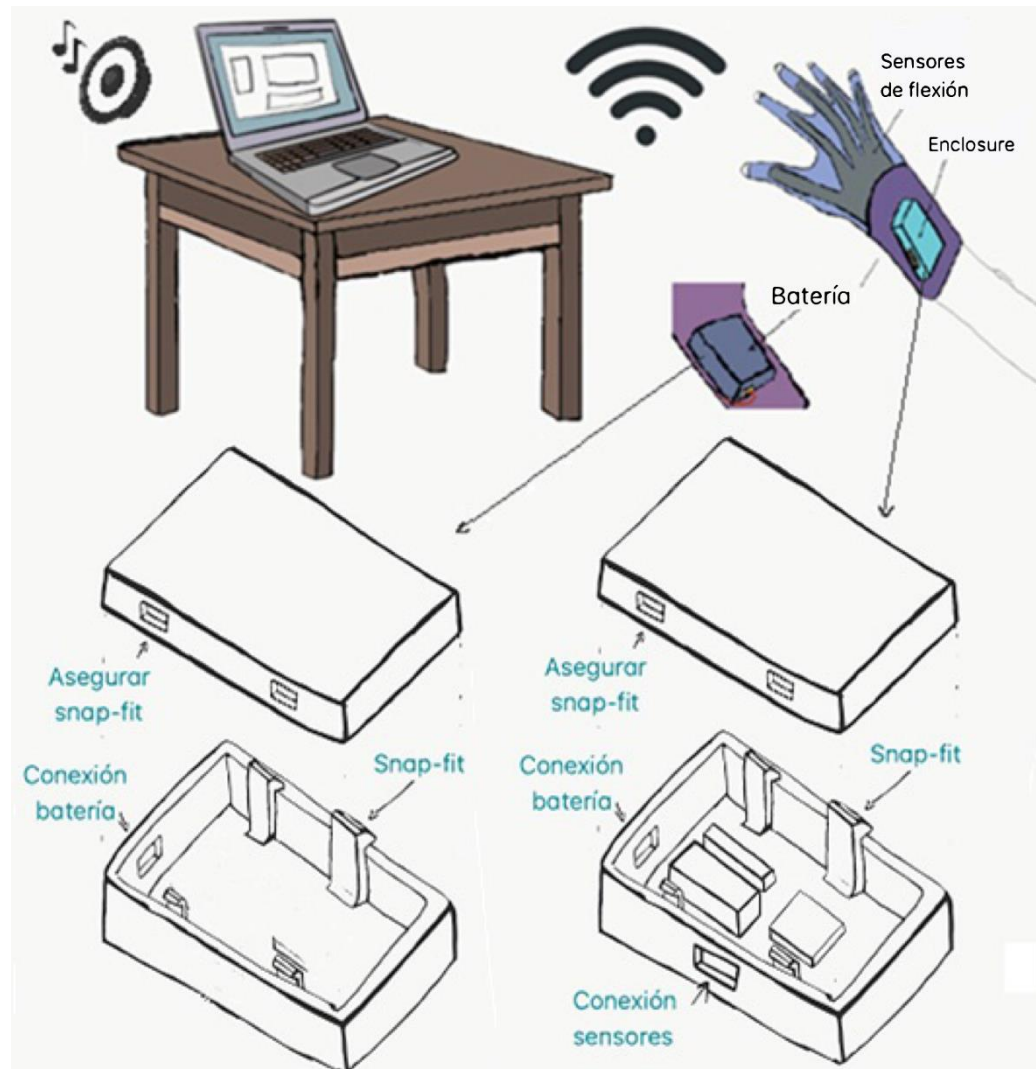


Figura 2.11: Bosquejo segundo concepto solución

Fuente: Propia

2.4.3. Concepto de solución 3

En este concepto se propone usar giroscopios MEMS para detectar el movimiento de las articulaciones de los dedos, cada giroscopio MEMS ira colocado como se muestra en la figura 2.12. Además, se usará un giroscopio y acelerómetro para detectar la orientación de la muñeca. El *enclosure* irá colocado en la parte superior de la muñeca donde se albergará el microcontrolador, el giroscopio, acelerómetro y el módulo Wi-

Fi. Para alimentar el sistema, se usarán baterías Ni-mh, las cuales, debido a su gran tamaño, irán colocadas en otro *enclosure*, posicionado en el reverso de la muñeca como se muestra en la figura correspondiente a la función albergar electrónica de la matriz morfológica (Tabla 2.7). El reconocimiento de gestos se realizará de manera similar a los conceptos de solución 1 y 2, con la diferencia que en este concepto los datos tomados de los sensores se envían a la PC mediante un módulo Wi-Fi y los gestos se clasifican mediante el método *Random Forest*.

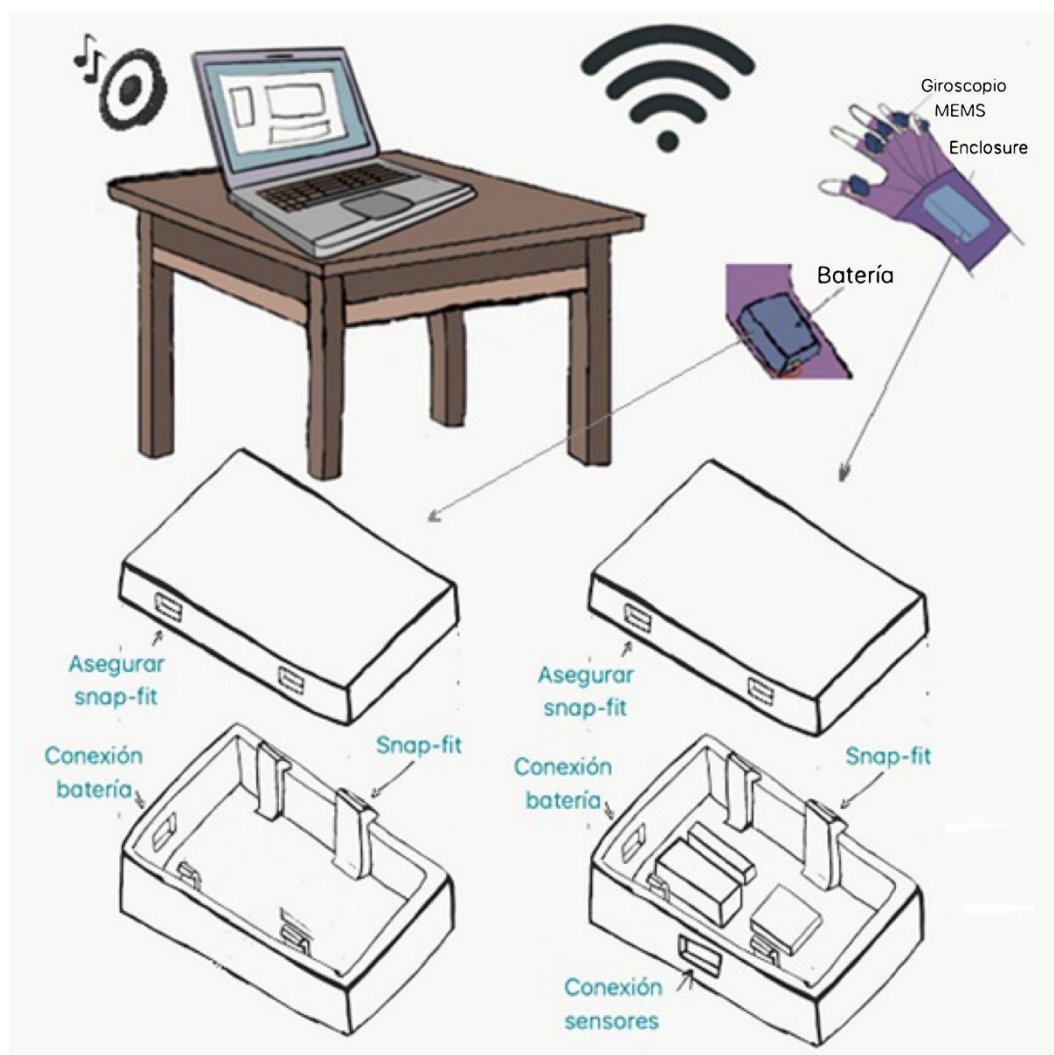


Figura 2.12: Bosquejo tercer concepto solución
Fuente: Propia

2.5. Evaluación técnico-económica

Para cada concepto de solución, se realiza la evaluación técnico-económica en base a la recomendación VDI 2225. Las tablas con la información detallada se encuentran posteriormente. En la tabla 2.8 se ha realizado la evaluación técnica de los conceptos de solución mostrados anteriormente. Por otro lado, la tabla 2.9 muestra la evaluación económica. Los criterios de evaluación poseen un respectivo peso basándose en el nivel de importancia del criterio, así mismo, a cada criterio se le asigna un puntaje. Luego el puntaje de cada solución es comparado con el puntaje ideal para obtener la solución óptima.

2.5.1. Evaluación técnica

En la tabla 2.8 se puede observar que el valor X_i que se acerca más a la solución ideal es el del concepto solución 2, esto se debe principalmente a que es la solución más compacta y ergonómica de las 3 alternativas.

Tabla 2.8: Evaluación técnica
Fuente: Propia

Diseño mecatrónico – Evaluación técnica										
Valor Técnico(X_i)										
Proyecto: GUANTES INTELIGENTES QUE DETECTAN MOVIMIENTOS DE LAS MANOS PARA PROCESAMIENTO DE AUDIO										
Criterios de evaluación para diseños en fase de conceptos										
p: puntaje de 0 a 4 (Escala de valores según VDI 2225)										
0 = No satisface, 1 = Apenas satisface, 2 = Suficiente, 3 = Bien, 4 = Excelente (Ideal)										
g: peso ponderado en función de la importancia de los criterios de evaluación										
4 = muy importante, 3= importante, 2=poco importante										
Conceptos de solución			CS1		CS2		CS3		CS Ideal	
N°	Criterios de evaluación	g	p	pg	p	pg	p	pg	p	pg
1	Función principal	4	3	12	3	12	3	12	4	16
2	Forma (compacto)	4	3	12	4	16	3	12	4	16
3	Seguridad	3	3	9	3	9	3	9	4	12
4	Ergonomía	4	2	8	4	16	3	12	4	16
5	Método de control de algoritmos	4	3	12	3	12	3	12	4	16
6	Captura de datos	4	3	12	3	12	4	16	4	16
7	Electrónica	3	3	9	3	9	3	9	4	12
8	Fabricación	2	3	6	4	8	3	6	4	8
9	Energía	3	3	9	3	9	3	9	4	12
10	Mantenimiento / repuestos	2	3	6	4	8	2	4	4	8
Puntaje máximo $\sum g / \sum pg$		33		95		111		101		132
Valor Económico Y_i				0.72		0.84		0.77		1
Orden				3		1		2		

2.5.2. Evaluación económica

En la tabla 2.9 se puede observar que el valor Y_i que se acerca más a la solución ideal es el del concepto solución 2, esto se debe principalmente a que es la solución más económica y cuyos componentes son más sencillos de conseguir repuestos en caso sea necesario.

Tabla 2.9: Evaluación económica
Fuente: Propia

Diseño mecatrónico – Evaluación económica										
Valor Económico(Y_i)										
Proyecto: GUANTES INTELIGENTES QUE DETECTAN MOVIMIENTOS DE LAS MANOS PARA PROCESAMIENTO DE AUDIO										
Criterios de evaluación para diseños en fase de conceptos										
p: puntaje de 0 a 4 (Escala de valores según VDI 2225) 0 = No satisface, 1 = Apenas satisface, 2 = Suficiente, 3 = Bien, 4 = Excelente (Ideal)										
g: peso ponderado en función de la importancia de los criterios de evaluación 4 = muy importante, 3= importante, 2=poco importante										
Conceptos de solución			CS1		CS2		CS3		CS Ideal	
N°	Criterios de evaluación	g	p	pg	p	pg	p	pg	p	pg
1	Costo de componentes	4	3	12	4	16	4	16	4	16
2	Repuesto de piezas	4	2	8	3	12	2	8	4	16
3	Costo de mantenimiento	2	2	4	3	6	3	6	4	8
4	Costo de operaciones	2	3	6	3	6	3	6	4	8
Puntaje máximo $\sum g / \sum pg$		12		30		40		36		48
Valor Económico Y_i			0.63		0.83		0.75		1	
Orden			3		1		2			

Con los resultados de las Tablas 2.8 y 2.9, se realizó el análisis técnico-económico para determinar el concepto de solución óptimo mostrado en la Figura 2.13. El concepto óptimo es aquel que se encuentra lo más cerca de la recta ideal. Como se puede ver, el concepto de solución 2 y 3 se encuentran muy cerca de la solución ideal, sin embargo, el concepto de solución 2 obtuvo mayor puntaje, por lo cual es el concepto de solución óptimo.

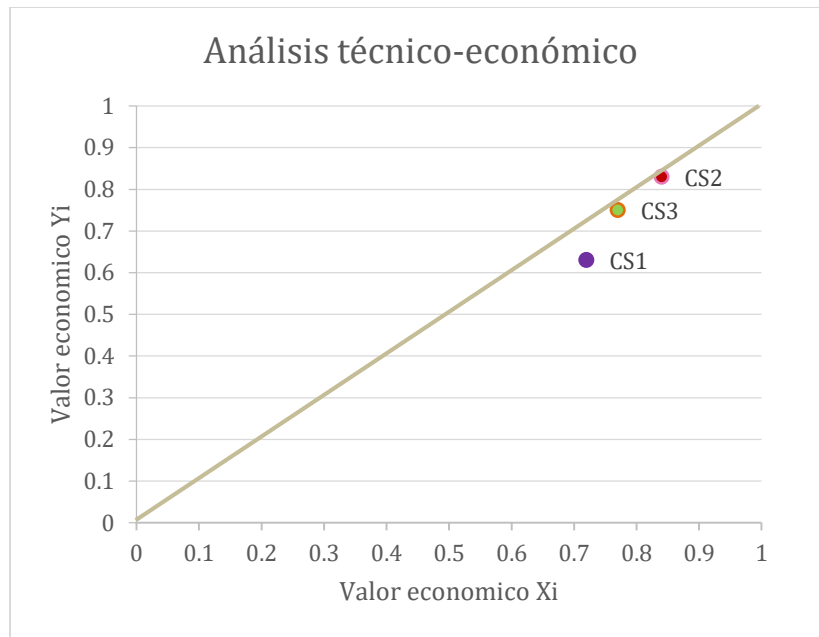


Figura 2.13: Análisis técnico-económico
Fuente: Propia

Al finalizar este capítulo, se concluye el diseño conceptual para la problemática planteada en el capítulo anterior. El diseño conceptual incluye los requerimientos del sistema, estructuras de funciones, matriz morfológica y conceptos de solución, escogiendo el concepto 2 como el más indicado, se llegó a este resultado luego de realizar las evaluaciones técnica y económica. En el siguiente capítulo, se desarrollará el diseño del guante inteligente.

CAPÍTULO 3

DISEÑO DE GUANTE INTELIGENTE

En este capítulo se presenta el desarrollo del concepto de solución óptimo, basándose en los capítulos anteriores. En primer, lugar se muestra el sistema integrado. Luego, se describe la parte electrónica del sistema, donde se detalla la elección de componentes, el diagrama de bloques y el diagrama esquemático. Seguidamente, se describe la parte mecánica del sistema, donde se muestran los planos, selección de tornillos y moldes del sistema. Y, por último, se desarrolla la construcción de un modelo de reconocimiento de gestos.

3.1. Integración de guante inteligente que reconoce gestos de manos para reproducir música

El sistema está compuesto por un dispositivo tipo *wearable* (guante) y un software con el cual el usuario podrá vincular gestos con archivos de audio para que sean reproducidos cuando el gesto correspondiente sea realizado. En la figura 3.1 se puede observar el sistema integrado, en la parte superior del guante se puede ver el *enclosure* que contiene la parte electrónica (figura 3.2), el cual contiene el microcontrolador, un sensor IMU y un módulo Wi-Fi para enviar los datos del guante al software de la PC; y en la parte inferior, el *enclosure* que contiene la batería LiPo (figura 3.3) que alimenta el dispositivo, ambos *enclosures* se sujetan al guante por medio de broches

de presión de acero inoxidable tamaño T3 (10 mm). Para asegurar que no se muevan y que los cables de conexión no se enreden al momento de hacer movimientos, se utiliza una pieza de tela Scuba para asegurar los *enclosures* y para mantener los cables en su lugar. Por otro lado, los sensores de flexión se colocan en *pockets* que están cosidos en la parte superior de los dedos del guante (figura 3.4), además para que estos no se muevan de su posición se sujetan con una pieza de velcro. En la figura 3.1, también, se muestra como el archivo de audio (notas) cambia cuando se realiza otro gesto.



Figura 3.1: Integración del Sistema
Fuente: Propia

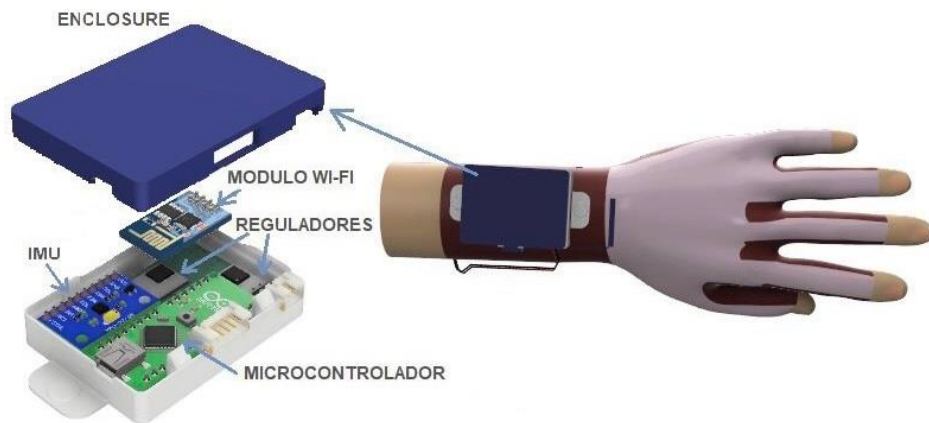


Figura 3.2: Integración parte electrónica
Fuente: Propia

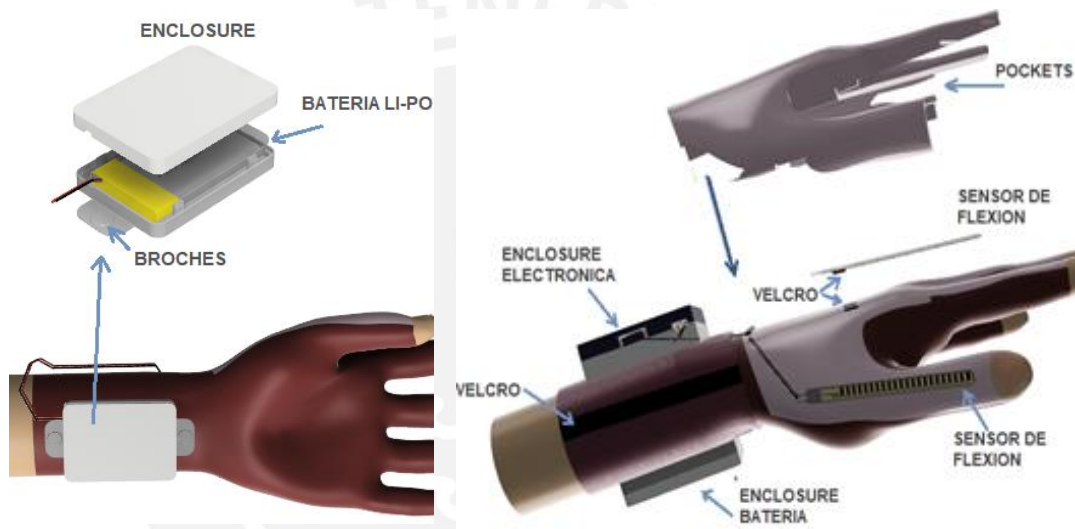


Figura 3.3: Integración alimentación
Fuente: Propia

Figura 3.4: Integración sensores en guante
Fuente: Propia

3.1.1. Funcionamiento

El funcionamiento del sistema se separa en el guante y el software. El microcontrolador colocado en el guante se encarga de almacenar los datos recolectados de los sensores de flexión e IMU, para que luego los datos encapsulados sean enviados por el módulo Wi-Fi a la PC. Al iniciar la interfaz, el usuario deberá elegir qué modo de reconocimiento de gestos quiere utilizar en la pantalla de Inicio (figura 3.5), donde se muestran las opciones “usuarios múltiples” y “usuarios específicos”.



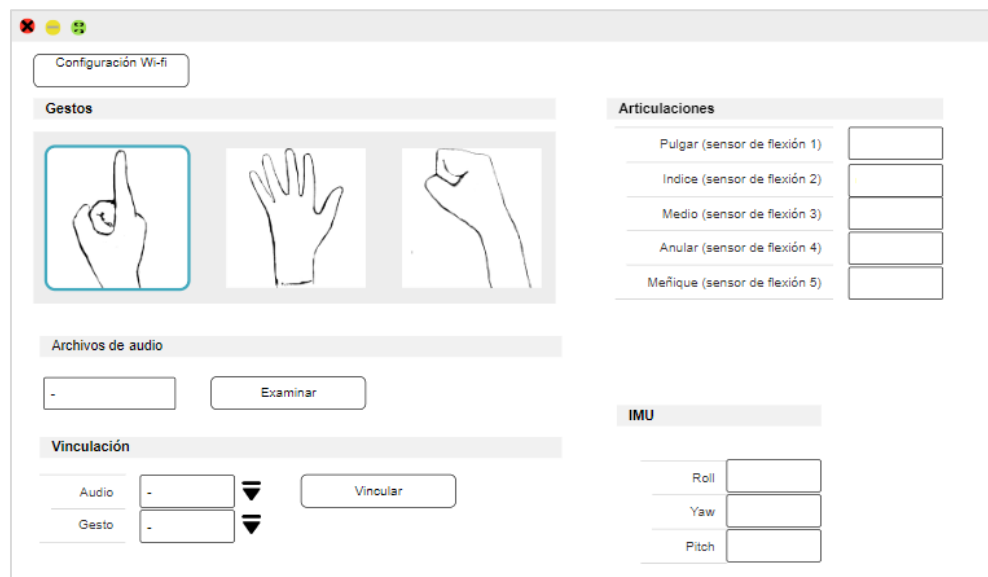
Figura 3.5: Diseño interfaz de usuario – Pantalla Inicio
Fuente: Propia

Al hacer clic en “usuarios específicos” el usuario se dirigirá a la interfaz de entrenamiento, la cual se muestra en la figura 3.6, donde deberá grabar los gestos que sean necesarios, haciendo clic en el gesto a grabar y realizando ese gesto repetidamente por 60 segundos, luego de terminar este procedimiento hará clic en “Done” y se dirigirá a la interfaz de “Interacción” (figura 3.7). Al utilizar este modo, se tiene la ventaja de precisión, ya que el modelo de clasificación se crea con los datos actuales del usuario.



Figura 3.6: Diseño interfaz de usuario - Entrenamiento
Fuente: Propia

Por otro lado, regresando a la pantalla inicio (figura 3.5), si el usuario hace clic en “usuarios múltiples” estaría eligiendo usar el modelo del mismo nombre, el cual tiene la desventaja de ser menos preciso que el modelo para usuarios específicos, sin embargo, no perdería tiempo en grabar sus gestos y se dirigiría a la interfaz de “Interacción”, la cual se muestra en la figura 3.7.



*Figura 3.7: Diseño interfaz de usuario - Interacción
Fuente: Propia*

Cuando el usuario se encuentre en la interfaz de “Interacción”, podrá observar los datos de orientación de los sensores y podrá agregar audios con los que quiera trabajar en la sección “Archivos de audio” (figura 3.7), estos archivos serán almacenados para que luego en la sección “Vinculación” (figura 3.7) puedan ser vinculados con alguno de los gestos guardados. En la sección “Gestos” (figura 3.7), el usuario hará clic en el gesto que desee actualizar, en caso que el reconocimiento de gestos no esté funcionando de forma óptima, de este modo puede entrenar de mejor manera el algoritmo y obtener mejores resultados. Luego que el usuario haya vinculado al menos un gesto con un archivo de audio, al realizar dicho gesto se reproducirá el archivo de audio.

En la figura 3.8 se muestra el diagrama de operaciones del sistema para usuarios específicos, en el cual el usuario elige grabar sus propios gestos.

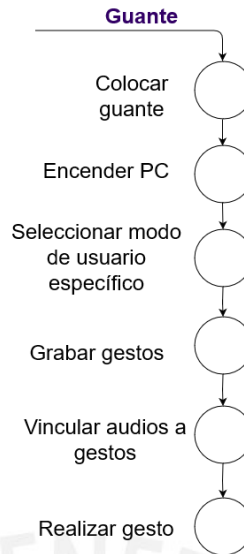


Figura 3.8: Diagrama de operaciones del sistema para usuarios específicos
Fuente: Propia

3.2. Diseño electrónico

En esta sección se desarrolla el diseño electrónico del guante. Primero se presenta el diagrama de bloques, cuya elección de componentes está basada en la estructura de funciones detallada en el capítulo anterior. Para la selección se evaluaron los parámetros que se requerían para el sistema, luego se realizó un estudio de las especificaciones de las alternativas que podrían usarse y finalmente se seleccionó el módulo que mejor se acomodó a los parámetros. Por último, se muestran el diagrama esquemático del sistema, especificando los códigos de los componentes, y el PCB diseñado para la fácil conexión de los componentes.

3.2.1. Diagrama de bloques

En la figura 3.9 se muestra el diagrama de bloques con las diferentes conexiones del sistema. El encapsulamiento de datos para envío a la PC se realiza por medio de una placa Arduino Nano, la cual se encarga de la recepción de datos de los sensores de flexión e IMU. La placa se encuentra conectada a un módulo Wi-Fi, el cual se encarga de enviar los datos de los sensores a la PC, donde se realizará el procesamiento de estos datos para el reconocimiento de gestos. El sistema se alimenta por una batería

LiPo de 3.7 V, y esta se conecta a un regulador *step-up* con voltaje de salida fijo de 5 V, el cual se usa para alimentar el Arduino Nano y un regulador *step-down*, del cual se obtiene un voltaje de salida fijo de 3.3 V para alimentar los módulos Wi-Fi e IMU.

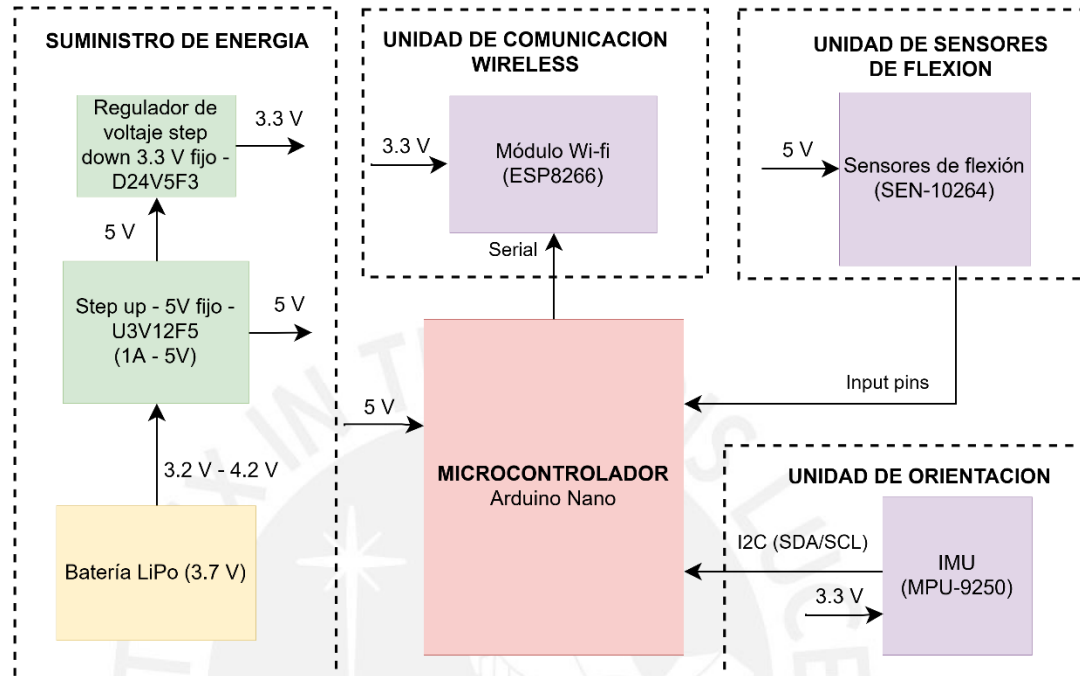


Figura 3.9: Diagrama de bloques
Fuente: Propia

3.2.2. Componentes electrónicos

En esta sección se muestran los componentes electrónicos elegidos para el dispositivo, así como también se muestran las alternativas que los pudieron haber reemplazado.

Sensores

IMU

Un sensor IMU se encarga de medir la orientación, la velocidad y las fuerzas gravitacionales de un objeto al que este haga referencia, en el caso de este sistema, se mide la orientación de la muñeca. Un IMU está conformado por un acelerómetro, un giroscopio y un magnetómetro, en caso de ser de 9DOF y en otros casos también cuenta con un compás haciéndolo un IMU de 10 DOF. En la tabla 3.1 se detallan los parámetros para la selección del sensor.

Tabla 3.1: Parámetros para selección de sensor IMU
Fuente: Propia

Parámetros	
Rango acelerómetro	+/- 3g
Rango giroscopio	+/- 500
DOF	9

Por otro lado, en la figura 3.10 se puede ver que la velocidad para un perfil de movimiento rápido de mano típico es de 2.5 m/s, y aceleración de +/- 20 m/s² por lo cual se debe tener en cuenta para escoger el sensor IMU. En la figura 3.11 se puede ver que la velocidad angular para la flexión y extensión de una muñeca promedio varía entre +/- 500 grados/s.

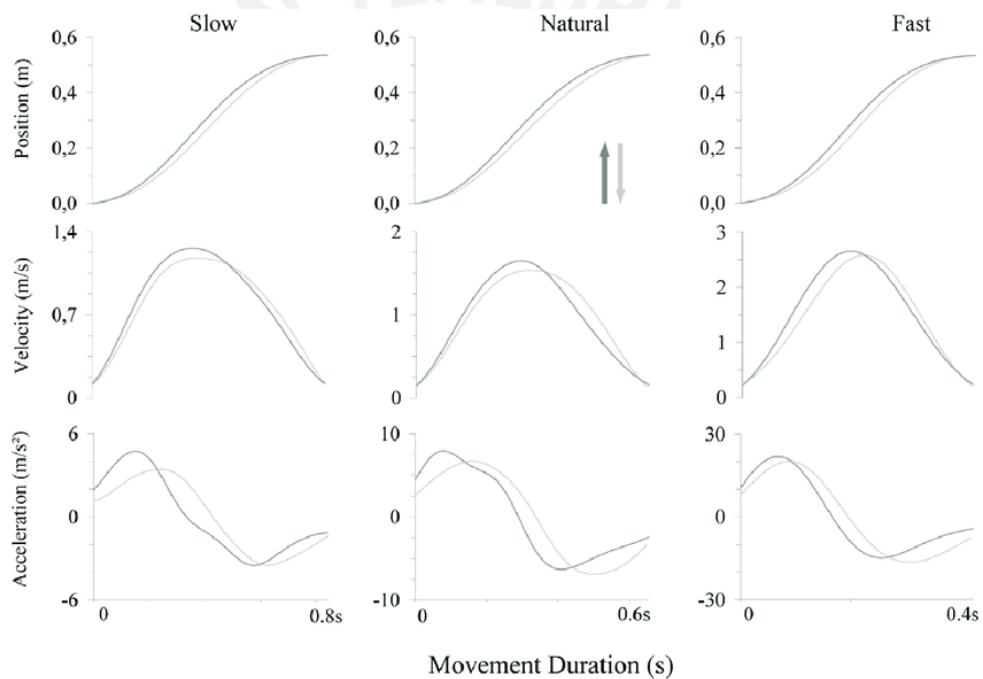


Figura 3.10: posición, velocidad y aceleración de movimientos promedios de mano de un sujeto de modo lento, natural y rápido
Fuente: [21]

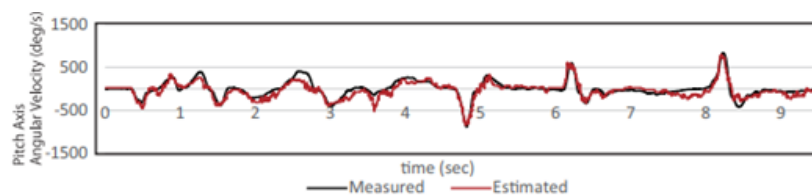


Figura 3.11: Estimación y medición de velocidad angular de movimiento flexión - extensión de la muñeca de un sujeto promedio
Fuente: [22]

En la tabla 3.2 se muestran las alternativas para elegir el sensor IMU a usar en el dispositivo. Se detallan las especificaciones por DOF, rango del acelerómetro y rango del giroscopio.

Tabla 3.2: Alternativas para sensor IMU
Fuente: Propia

Alternativas para el sensor IMU				
Modelo	DOF	Rango acelerómetro	Rango giroscopio	Costo (\$)
Adafruit IMU Breakout (L3GD20 + LSM303)	9	$\pm 2g, \pm 4g, \pm 8g, \pm 16$	$\pm 250, \pm 500, \pm 2000$	19.95
Invensense MPU-9250	9	$\pm 2g, \pm 4g, \pm 8g, \pm 16$	$\pm 250, \pm 500, \pm 1000, \pm 2000$	14.95
Adafruit IMU Breakout (L3GD20 + LSM303 + BMP180)	10	$\pm 2g, \pm 4g, \pm 8g, \pm 16$	$\pm 250, \pm 500, \pm 2000$	29.95

Todos los sensores descritos anteriormente cumplen con los parámetros establecidos. Sin embargo, para este dispositivo no es necesario el uso de un IMU de 10 DOF, por lo cual se descarta el uso de este sensor. Por otro lado, de las otras 2 opciones restantes se escoge el modelo MPU 9250 de Invensense, ver figura 3.12, debido a que su costo es más accesible. En la figura 3.13 se puede ver la posición del sensor dentro del *enclosure*. En la tabla 3.3 se pueden ver los parámetros de alimentación para el módulo escogido.

Tabla 3.3: Parámetros módulo MPU 920
Fuente: Propia

	Voltaje de alimentación de chip	Voltaje de alimentación de módulo
Invensense MPU-9250 - MPU 92/65	2.4 V - 3.6 V	3.3 V - 5 V



Figura 3.12: Sensor IMU – MPU 9250

Fuente: <https://es.gearbest.com>



Figura 3.13: Sensor IMU – MPU 9250 en enclosure

Fuente: Propia

Sensores de flexión

Un sensor de flexión es aquel que mide el ángulo de flexión del objeto al que esté adherido, esto se logra convirtiendo el voltaje que se mide entre el sensor y tierra. Este sensor funciona como un resistor y la resistencia varía mientras esté sea flexionado. En la figura 3.14 se muestran datos del estudio *Flexion and Extension Angles of Resting Fingers and Wrist*, realizado por Kyung-Sun Lee y Myung-Chul Jung [23]. De la tabla que se muestra en la figura se puede resaltar que el rango en el cual los sujetos llegan a doblar las articulaciones MCP (articulación en la cual se enfoca el dispositivo) van desde 45 hasta 90 grados, por lo tanto, se usara este rango para la selección de sensor.

Study	Finger Joint			Wrist
	MCP	PIP	DIP	
Daniels, Zook & Lynch [12]	45°–70°	10°	10°	n.d.
Taylor, Hanna & Belcher [13]	70°–90°	0°	0°	–40°––30°
Bach, Draslov & Jørgensen [14]	60°	0°	0°	–20°–0°
Bednar [15]	60°–70°	0°–10°	n.d.	–15°––10°
Clark [16]	60°	10	5°	–25
Taams, Ash & Johannes [17]	70°–90°	0°–20°	n.d.	–45°––30°
Tan, Mathis & El-Gamal [18]	80°	n.d.	n.d.	–30°

Notes. MCP = metacarpophalangeal joint, PIP = proximal interphalangeal joint, DIP = distal interphalangeal joint, n.d. = not discussed.

Figura 3.14: Ángulos de articulación de dedos y muñeca en diferentes sujetos
Fuente: [23]

En la tabla 3.4 se detallan los parámetros a tomar en consideración para la correcta selección del sensor de flexión.

Tabla 3.4: Parámetros para selección de sensor de flexión
Fuente: Propia

Parámetros	
Gama de flexión	45 - 90 grados
Longitud	2"

En la tabla 3.5 se detallan las alternativas para la elección de los sensores de flexión.

Tabla 3.5: Alternativas para selección de sensor de flexión
Fuente: Propia

Modelo	Longitud	Rango resistencia	Rango de flexión	Tecnología	Costo (\$)
Sensor de flexión SEN-10264	2.2"	30 k Ω - 70 k Ω	0 - 90 grados	Basado en tinta conductora	7.95
Sensor de flexión SEN-08606	4.5"	30 k Ω - 70 k Ω	0 - 90 grados	Basado en tinta conductora	12.95
Sensor de fibra óptica	varia	-	0 - 135 grados	Tecnología fibra óptica	Según longitud

Todos los sensores descritos anteriormente cumplen con los requisitos para el dispositivo, sin embargo, los sensores de fibra óptica tienen un costo muy elevado en comparación a los sensores de flexión basados en tinta conductora. Por otro lado, entre los otros dos sensores descritos se escoge el sensor de longitud 2.2" SEN-08606 (figura 3.15) ya que es suficiente para cubrir la articulación MCP de los dedos. En la figura 3.16 se puede ver la posición de los sensores en el guante.

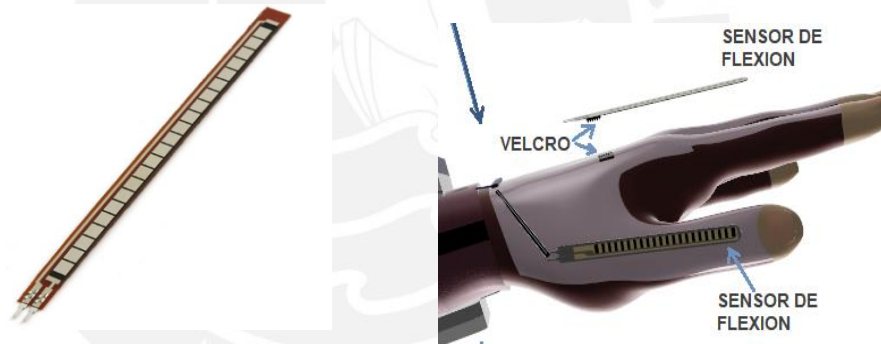


Figura 3.15: Sensor de flexión SEN-08606

Fuente: <https://www.sparkfun.com>

Figura 3.16: Sensores de flexión SEN-08606 en guante

Fuente: Propia

Comunicación

Módulo de comunicación *wireless*

Según el capítulo de estructura de funciones, la función de enviar datos a la PC se realizará por un módulo Wi-Fi. Esto se debe a que de este modo el usuario será más libre de realizar los movimientos que él desee. El rango de temperatura en la que se desea usar el dispositivo, como se encuentra indicado en el Anexo A, es de 10 °C a 35°C, y el rango de frecuencia debe ser de 2.4 GHz. En la tabla 3.6 se puede observar las alternativas de los módulos Wi-Fi que podrían ser usados en el dispositivo.

Tabla 3.6: Alternativas para selección de módulo Wi-Fi
Fuente: Propia

Alternativas para módulo Wi-Fi						
Modelo	Interfaz	Corriente promedio	Voltaje operativo	Temperatura	Rango de frecuencia	Costo (\$)
Módulo Wi-Fi ESP-01 - ESP8266	I2C/GPIO	80 mA	2.5 V - 3.6 V (3.3V)	-40 °C - 120°C	2.4G-2.5GHz	6.95
SPWF01SA.11	UART	105 mA	3.1 V - 3.6 V (3.3V)	-40 °C - 85°C	2.4GHz	19
Wi-Fi Module (802.11) EMW3166	ADC, GPIO, I2C, SPI	124 mA	3 V - 3.6 V (3.3V)	-30 °C - 85°C	2.4GHz	9.95

El módulo Wi-Fi que será parte del dispositivo es el ESP-01 (figura 3.17), se escogió este módulo debido a que, aunque todos los módulos cumplen con los requisitos para el sistema, esta es la opción económica más efectiva. En la figura 3.18 se puede ver el módulo posicionado en el *enclosure* del guante.

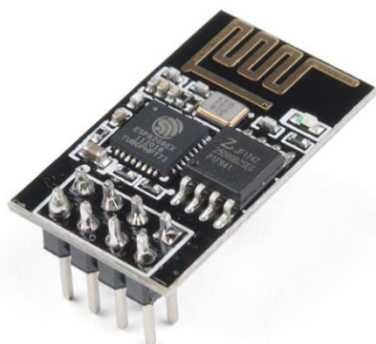


Figura 3.17: Módulo Wi-Fi ESP-01
Fuente: <https://www.sparkfun.com>

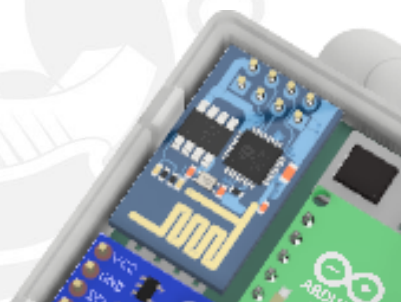


Figura 3.18: Módulo Wi-Fi ESP-01 en *enclosure*
Fuente: Propia

Dispositivo de control

Microcontrolador

El microcontrolador se encarga de recolectar los datos, encapsularlos y enviarlos a la PC, para esto deberá tener al menos 7 entradas analógicas para poder recolectar los datos de los sensores de flexión e IMU y ser capaz de comunicarse con el módulo Wi-Fi e IMU. En la tabla 3.7 se detallan los parámetros para la selección del microcontrolador.

Tabla 3.7: Parámetros para selección de microcontrolador
Fuente: Propia

Parámetros	
Numero de pines analógicos	7
Numero de pines digitales	2
Comunicación I2C y serial	si

En la tabla 3.8 se puede observar las alternativas de los microcontroladores que se podrían utilizar.

Tabla 3.8: Alternativas para selección de microcontrolador
Fuente: Propia

Alternativas para microcontroladores					
Modelo	Microcontrolador	Pines Analógicos / Digitales	I2C/Serial	Dimensiones	Costo (\$)
Arduino Nano	ATmega328	8 / 22	Si	18 x 45 mm	22
Adafruit Trinket M0	ATSAMD21E18	12 / 5	Si	15.3 x 27 mm	8.95

Basándose en la tabla 3.7 y 3.8 se escoge usar la placa Arduino Nano (figura 3.19) la cual está basada en el microcontrolador ATmega328 y cumple con los parámetros necesitados para el dispositivo. En la figura 3.20 se puede ver la placa posicionada en el *enclosure* del guante.



Figura 3.19: Arduino Nano
Fuente: <https://store.arduino.cc/usa/>

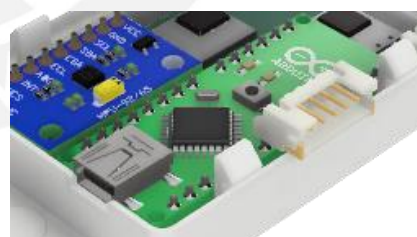


Figura 3.20: Arduino Nano en enclosure
Fuente: Propia

Energía

Batería

Uno de los requisitos del sistema es que el dispositivo debe ser compacto, por lo cual se utilizará una batería LiPo de 1 celda con voltaje nominal de 3.7 V (figura 3.21), se utilizara solo una batería, ya que el mayor voltaje de alimentación del sistema es de 5

V, para lo cual se usará un regulador *step-up*. En la figura 3.22 se puede ver la posición del *enclosure* donde se almacena la batería. Estas baterías tienen un rango de seguridad de voltaje de descarga y carga de 3.2 V a 4.2 V, sin embargo, los módulos que se deben alimentar requieren de 3.3 V y 5 V, por lo cual se requiere reguladores *step-up* y *step-down* para obtener dichos voltajes.

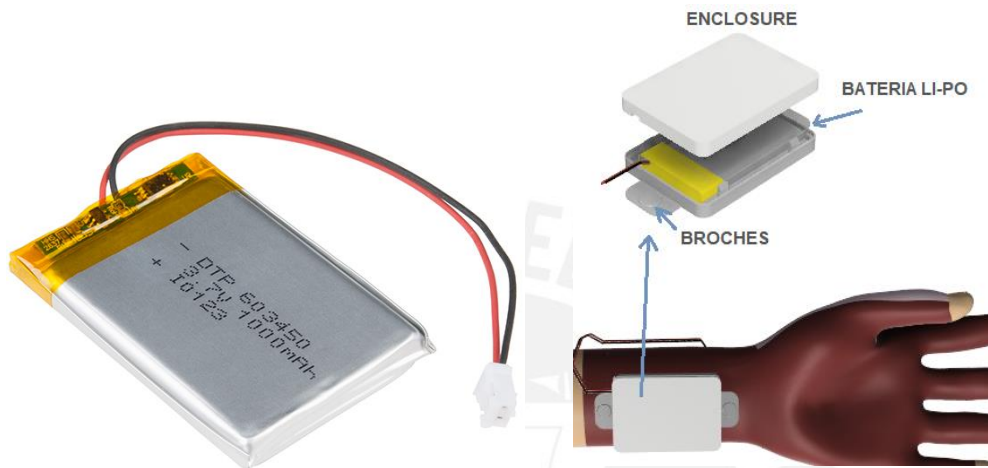


Figura 3.21: Batería LiPo

Fuente:

<https://www.sparkfun.com>

Figura 3.22: Batería LiPo en enclosure

Fuente: Propia

En la tabla 3.9 se puede observar los valores de corriente que se necesita para alimentar cada módulo.

Tabla 3.9: Corriente de alimentación

Fuente: Propia

Corriente necesaria para módulos		
Módulo Wi-Fi ESP-01 - ESP8266		
Corriente de operación promedio	Corriente para pin rx	Corriente por pin tx
80 mA	56 mA	170 mA
Arduino Nano		
Número de pines utilizados	Corriente máxima por pin i/o	
10 (9 para datos – 1 de alimentación)	40mA	
MPU 9250		
Corriente de operación cuando todos los sensores y DPM activado	Corriente de operación cuando todos los sensores y DPM desactivado	
3.5 mA	3.7 mA	

DPM: *Digital Motion Processing*

Debido a que se utilizaran 9 pines en el Arduino Nano, se estima que para alimentar la placa se necesita un estimado de 360 mA, es decir, 40 mA por pin. Por lo tanto, el regulador deberá poder dar esta cantidad de amperaje. Con respecto al módulo Wi-Fi

e IMU, se estima un amperaje de 170 mA y 3.7 mA, respectivamente. Por lo tanto, el regulador de entrega de 3.3 V deberá poder entregar la cantidad mencionada.

En la tabla 3.10 se detalla las características de la batería LiPo escogida para alimentar a los reguladores, esta se eligió basándose en lo requerimientos para alimentar el sistema detallados en la tabla 3.9.

Tabla 3.10: Parámetros de batería LiPo seleccionada
Fuente: Propia

Parámetros batería LiPo				
Corriente	Dimensiones	Celdas	Voltaje nominal	Costo (\$)
1 A	50.8 x 33.5 x 5.9 mm	1	3.7 V	5.95

Regulador *step-up*

Debido a que la alimentación del sistema se dará por una batería LiPo de una celda, para poder alimentar al microcontrolador 5V se da como solución el uso de un regulador *step-up*. En la tabla 3.11 se muestran los parámetros a tomar en cuenta para la selección del regulador.

Tabla 3.11: Parámetros para regulador step-up
Fuente: Propia

Parámetros	
Voltaje entrada	3.2 V a 4.2 V
Voltaje salida	5 V
Corriente de salida	400 mA

En la tabla 3.12 se muestran alternativas de reguladores *step-up*.

Tabla 3.12: Alternativas para regulador step-up
Fuente: Propia

Alternativas para LiPo Booster					
Modelo	Voltaje entrada	Voltaje salida	Alimentación	Dimensiones	Costo (\$)
<i>Adafruit PowerBoost 1000</i>	1.8 V - 5.5 V	1.8 V - 5.5 V	Conector LiPo JST	23 x 45 x 10 mm	19.95
<i>SparkFun LiPo Charger/Booster</i>	0.3 V - 5.5 V	3.3 V ó 5 V	Conector LiPo JST o input de alimentación de 2 pines	22 x 32 x 3 mm	14.95
<i>AK-MP3209 – Boost Converter Breakout</i>	2.6 V - 5 V	5 V ó 12 V	input de alimentación de 2 pines	29 x 14 x 1 mm	9.10
<i>Pololu step-up Voltage Regulator U3V12F5</i>	2.5 V - 5.5 V	5 V	input de alimentación de 2 pines	8.1 × 13.1 × 2.5 mm	3.95

De las alternativas anteriormente detalladas, se puede observar que todas cumplen con los parámetros necesarios para el dispositivo. Sin embargo, se escoge usar el regulador U3V12F5 debido a que es el que tiene menor dimensión y es la opción más económica.

El módulo U3V12F5 (figura 3.23) utiliza el IC TPS61202; en la figura 3.24, se puede ver el módulo posicionado en el *enclosure*. Por otro lado, en la figura 3.25 se puede observar que en el rango de 3,2 V a 4.2 V se entrega corriente en el rango de 0.75 A a 1 A, lo cual es suficiente para alimentar el microcontrolador y el regulador *step-down*. Además, la corriente de entrada puede llegar hasta 1.4 A, por lo cual una batería LiPo de una celda es la que alimentara este regulador.



Figura 3.23: Módulo step-up U3V12F5

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



Figura 3.24: Módulo step-up U3V12F5 en enclosure

Fuente: Propia

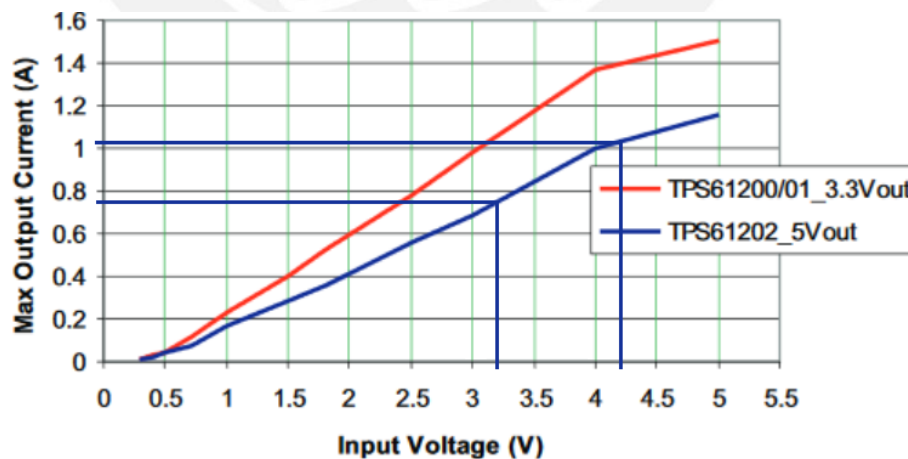


Figura 3.25: Corriente máxima de salida vs Voltaje de entrada

Fuente: [24]

Regulador *step-down*

Debido a que la alimentación del sistema se dará por una batería LiPo de una celda, para poder alimentar los módulos IMU y Wi-Fi se regulará el voltaje que se obtenga de la salida fija de 5 V del regulador *step-up*. En la tabla 3.13 se muestran los parámetros a tomar en cuenta para la selección del regulador.

Tabla 3.13: Parámetros para regulador *step-down*
Fuente: Propia

Parámetros	
Voltaje entrada	5 V
Voltaje salida	3.3 V
Corriente requerida	180 mA

En la tabla 3.14 se muestran alternativas de reguladores *step-down* para obtener un valor de voltaje fijo de 3.3 V.

Tabla 3.14: Alternativas para regulador *step-down*
Fuente: Propia

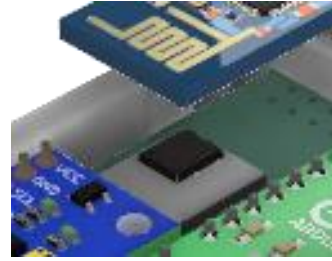
Alternativas para regulador <i>step-down</i>					
Modelo	Voltaje entrada	Voltaje salida	Corriente de salida máxima	Dimensiones	Costo (\$)
<i>Pololu step-up/step-down Voltage Regulator S7V8F3</i>	2.7 V - 11.8 V	3.3 V	1 A	11.43mm x 16.51mm x 2.54mm	5.95
<i>LM3940 1-A</i>	5 V	3.3 V	1 A	10mm x 4.7mm x 25.8mm	1.45
<i>Pololu step-down Voltage Regulator D24V5F3</i>	3.4 V - 36 V	3.3 V	500 mA	10.16 mm x 12.7 mm x 2.54 mm	4.95

De las alternativas anteriormente detalladas, se puede observar que todas cumplen con los parámetros necesarios para el dispositivo. Sin embargo, se escoge usar el regulador *step-down* D24V5F3 (figura 3.26) debido a que, aunque no es el que tiene menor dimensión ni es la opción más económica, es el que entrega la suficiente cantidad de corriente para alimentar a los módulos. En la figura 3.27 se puede ver el módulo posicionado en el *enclosure*, este se encuentra posicionado debajo del módulo Wi-Fi para que no hay necesidad de usar más espacio y así el *enclosure* sea de las dimensiones más óptimas posibles.



Figura 3.26: Módulo step-down D24V5F3

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



Fuente: Propia

3.2.3. Diagramas esquemáticos y PCB

Calculo para resistencia de circuito divisor de tensión para sensor de flexión

En la figura 3.28 se muestra el diagrama de conexiones para un sensor de flexión.

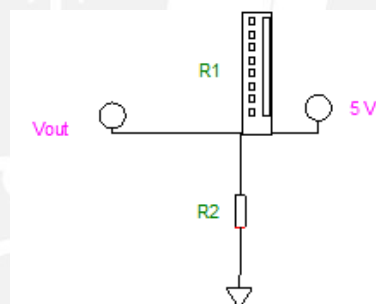


Figura 3.27: Diagrama de conexiones para sensor de flexión

Fuente: Propia

A continuación, en (1) se muestra la fórmula para obtener el voltaje de salida (V_{out}).

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} . \quad (1)$$

Se reemplaza V_{in} en (1):

$$V_{in} = 5 V .$$

Luego, ya que el rango de resistencia para los sensores de flexión (R_1) elegidos es de 30 k Ω a 70 k Ω , se debe escoger un valor R_2 que optimice el rango de voltaje que se pueda obtener para el sensor. Se realizaron los cálculos con valores de 10 k Ω a 75 k Ω en Excel, obteniéndose la gráfica en la figura 3.29.

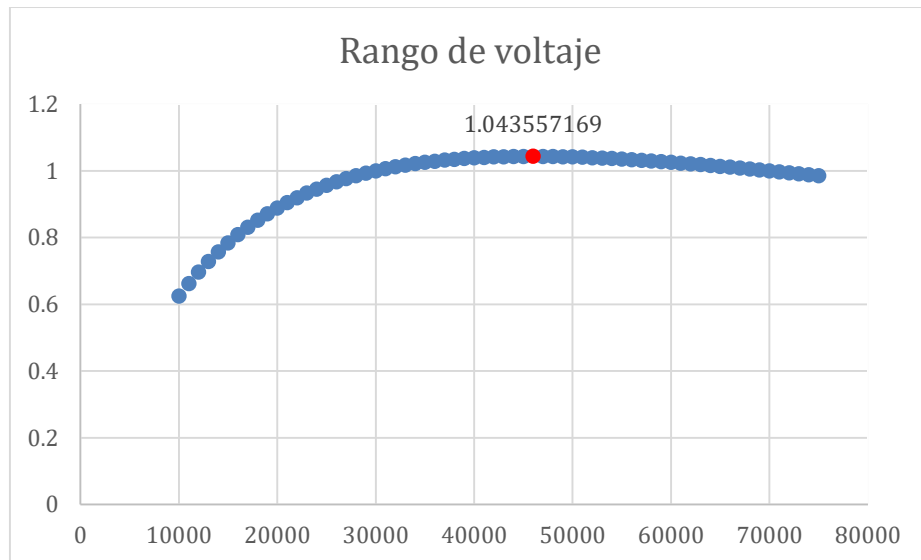


Figura 3.28: Rango de voltaje vs Resistencia R1
Fuente: Propia

El valor de R_2 para obtener el mayor rango de voltaje es de 46 kΩ.

Diagrama esquemático general

En la figura 3.30 se muestra el diagrama esquemático general del sistema. La placa Arduino Nano es alimentada por el regulador *step-up*, el cual también alimenta el regulador *step-down*, cuyo voltaje de salida alimenta a los módulos IMU y Wi-Fi.

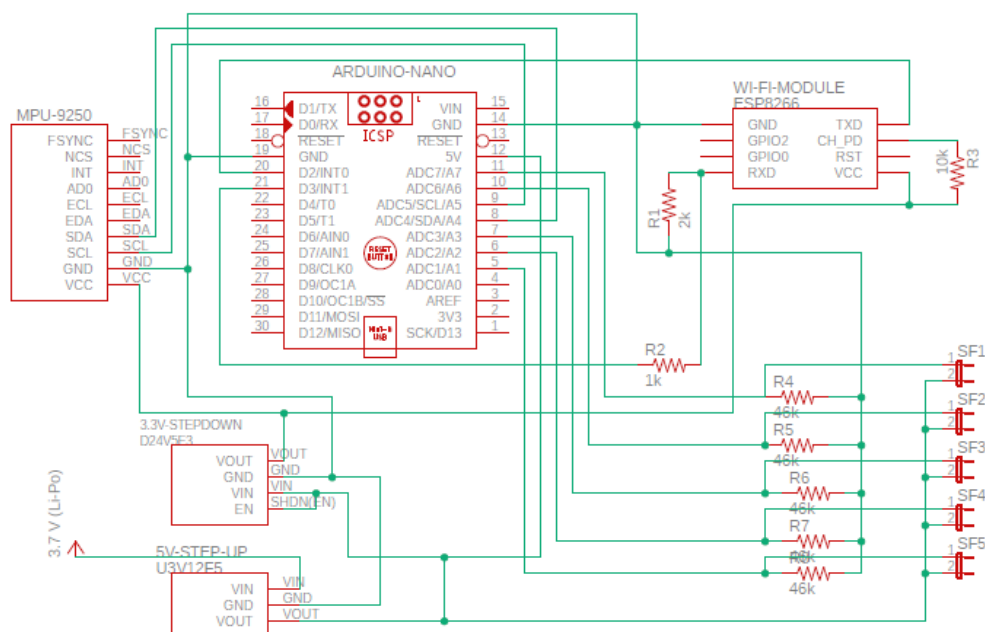


Figura 3.29: Diagrama esquemático general del sistema
Fuente: Propia

Diseño de PCB

Se diseñó una PCB (placa de circuito impreso) de 2 capas mediante el software Eagle para facilitar el ensamble de las partes electrónicas. En la figura 3.31 se puede ver la vista superior e inferior de la PCB, así como también las dimensiones en mm.

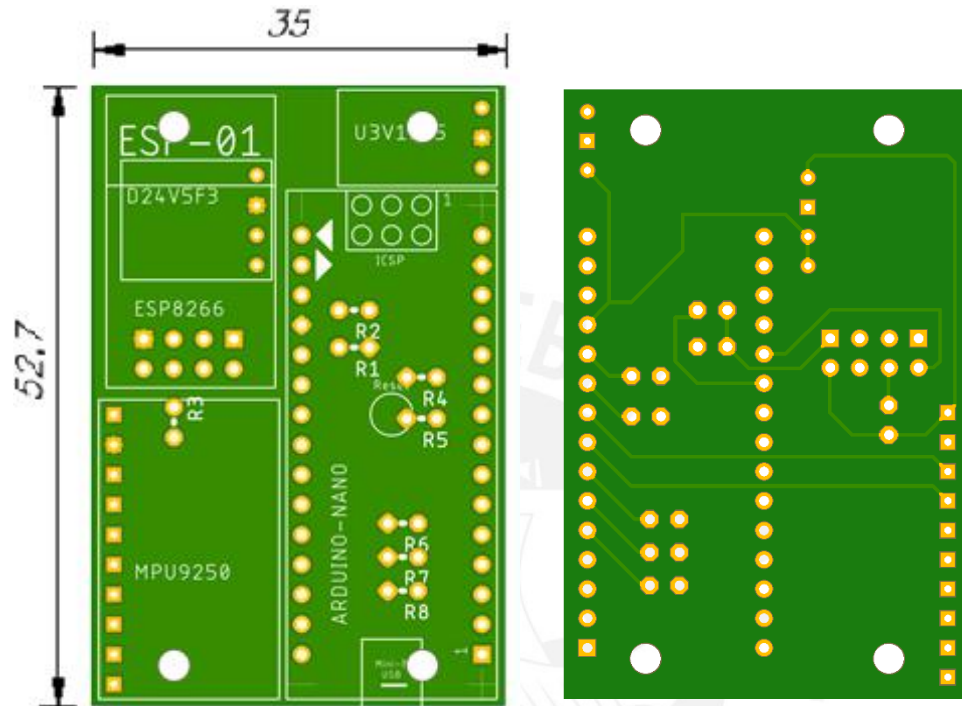


Figura 3.30: PCB – Vista superior e inferior
Fuente: Propia

En el Anexo G se puede observar el diagrama esquemático y el diseño del PCB en escala 2:1.

3.3. Diseño mecánico

En esta sección se presentan los cálculos de los *snap-fit* para los *enclosures* de la batería y la parte electrónica del dispositivo; por último, se muestran los planos de dichos *enclosures*, así como también el molde del guante indicando donde se colocan los *enclosures*. La teoría para diseño de *snap-fits* en *enclosures* se encuentra detallada en el Anexo C; por otro lado, los planos completos y moldes se encuentran en el Anexo G.

3.3.1. Ensamble general

En esta sección se presenta el ensamble general del dispositivo como se muestra en la figura 3.32. Los componentes 1 y 2 hacen referencia al guante y los *pockets*, respectivamente, donde se almacenarán los sensores de flexión. Los moldes para estos componentes de muestran posteriormente. Los componentes 3 y 4 hacen referencia al *enclosure* de electrónica y al *enclosure* de la batería, respectivamente. Por último, el componente 5 hace referencia al cobertor que se usara para asegurar que los *enclosures* no se desprendan del guante.

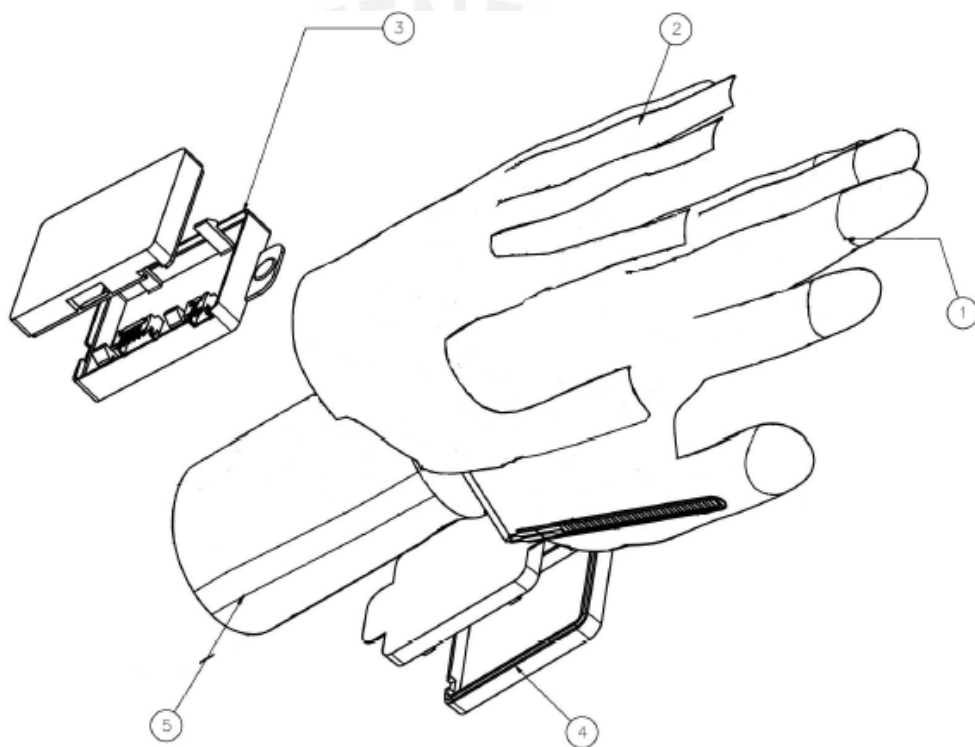


Figura 3.31: Vista isométrica del dispositivo
Fuente: Propia

3.3.2. Planos de sub-ensamble

En esta sección se muestran los planos de los *enclosures* diseñados para albergar la parte electrónica del dispositivo y la batería. Debido a que estos son diseños con ensamblaje por *snap-fits*, serán fabricados en ABS por impresión en 3D.

Plano sub-ensamble *enclosure* batería

En la figura 3.33 se muestra la vista isométrica del *enclosure*. Las dimensiones generales del *enclosure* son 41 mm de ancho, 82 mm de largo y 11 mm de altura, estas medidas son adecuadas para que la batería LiPo pueda ser albergada.

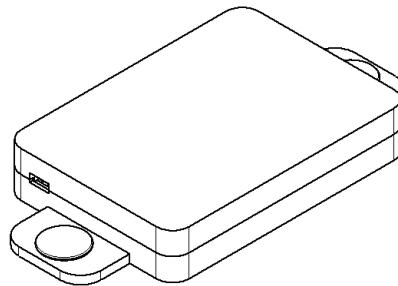


Figura 3.32: Vista isométrica *enclosure* - batería
Fuente: Propia

En la figura 3.34 se puede ver la vista principal del *enclosure*, donde se muestran el detalle B, en el cual se presenta la abertura por donde sale el cable de la batería LiPo (figura 3.35) y el detalle C, donde se muestra el diseño del *snap-fit* (figura 3.36).

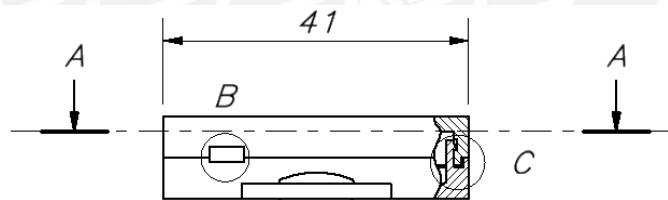


Figura 3.33: Vista principal *enclosure* - batería
Fuente: Propia

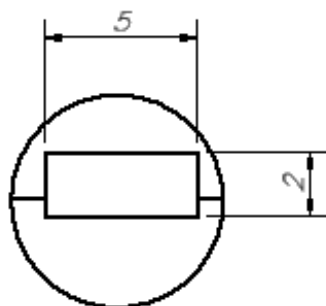


Figura 3.34: Detalle B
Fuente: Propia

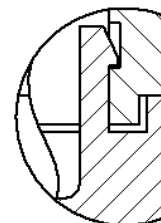


Figura 3.35: Detalle C
Fuente: Propia

Como se puede observar en la figura 3.35, debido a las dimensiones de la apertura para este *enclosure* se puede decir que está protegido contra cuerpos sólidos de

diámetro mayor a 5 mm. Por lo tanto, el *enclosure* cuenta con protección IP20 contra cuerpos de diámetro mayor a 12 mm, se llegó a esta conclusión basándose en la tabla de características de protección IP, la cual se encuentra en el Anexo I.

En la vista principal se realizó un corte A-A, cuya vista se muestra en la figura 3.37, aquí se puede apreciar la batería LiPo, la cual está señalada como elemento 4 del ensamblaje de este *enclosure*. También, se puede observar las extensiones a cada extremo del *enclosure*, en las cuales se colocará broches a presión. En cada extensión hay un agujero donde se posicionará el broche para sujetar el *enclosure* en el guante, luego de esto se asegura con el cobertor, cubriendo las extensiones.

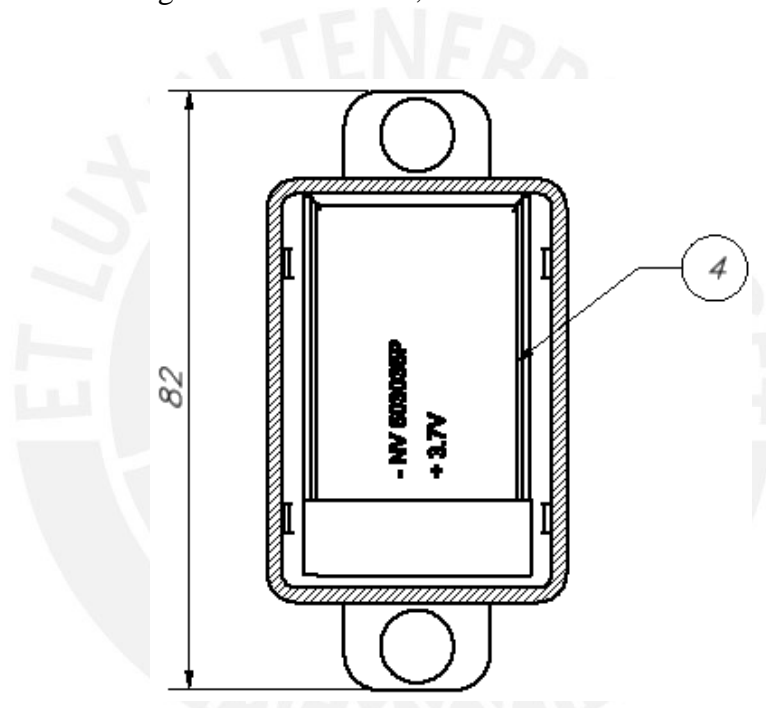


Figura 3.36: Corte A-A enclosure - batería
Fuente: Propia

El *snap-fit* en el detalle C (figura 3.36) fue diseñado según el libro *First Snap-fit Handbook* edición 3 y el Manual de Diseño de *Snap-fits* de BASF [27]. El *enclosure* tiene 4 *snap-fits* ubicados simétricamente, cuyos cálculos se muestran a continuación.

Deformación admisible

Usando la ecuación en (2)

$$y = \frac{0.67 * \epsilon_o * l^2 * Q}{h} \quad , \quad (2)$$

y deformación unitaria (strain) admisible para ABS: $\epsilon_o(ABS) = 2.7\%$.

El *enclosure* para la batería LiPo deberá ser separado frecuentemente, por lo que se usa solo el 60% del valor de la deformación unitaria permisible para los cálculos.

$$\epsilon_o = 2.7\% * 0.6 = 1.62\%$$

Los valores propuestos para l , h y b se detallan a continuación:

$l = 4.5 \text{ mm}$; $h = 1 \text{ mm}$; $b = 4 \text{ mm}$; $h' = 0.8 * 1 = 0.8 \text{ mm}$. Las variables mencionadas se muestran en la figura 3.38.

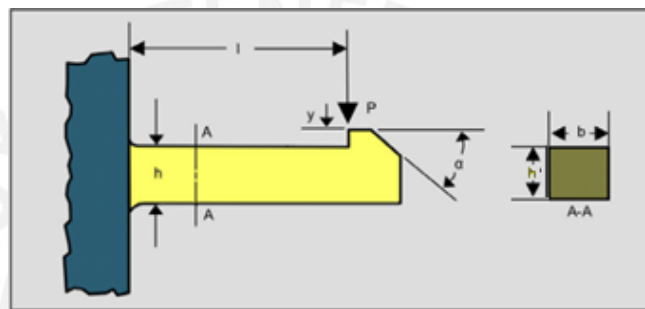


Figura 3.37: Diseño Snap-fit
Fuente: [26]

Se halla el ratio $\frac{l}{h}$ para poder hallar el factor Q :

$$\frac{l}{h} = 4.5.$$

Ya que es un *snap-fit* de tipo 2, por el gráfico en la figura 3.39 se obtiene: $Q = 1.25$.

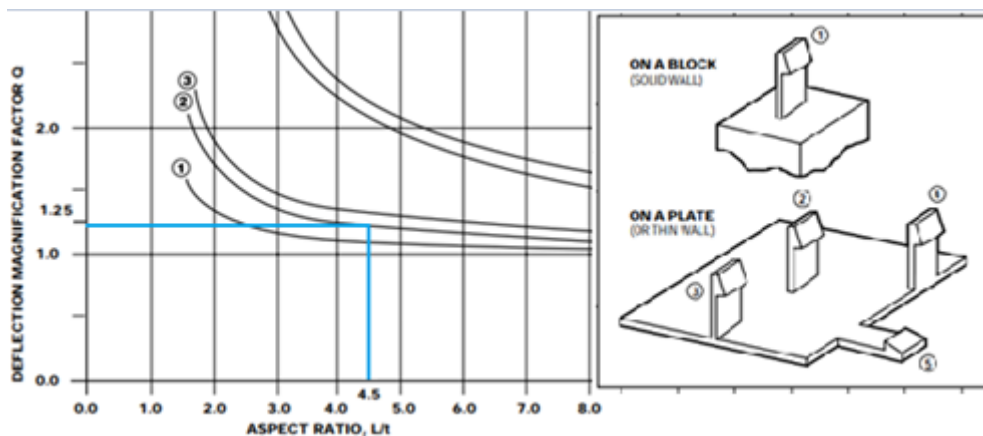


Figura 3.38: Factor de magnificación de deflexión Q vs Ratio L/t
Fuente: [27]

Finalmente, se reemplazan los valores en la ecuación de deformación admisible y se obtiene: $y_{max} = 0.27 \text{ mm}$. Según el libro *First Snap-fit Handbook* el ángulo de inserción $\alpha_{diseño}$ debe estar entre 25° y 35° , por lo cual se escoge $\alpha_{diseño} = 25^\circ$.

Fuerza de deformación

Usando la ecuación en (3)

$$P = \frac{b * h^2}{6} * \frac{E * \epsilon_o}{l} \quad . \quad (3)$$

Y el valor de módulo de Young: $E_{(ABS)} = 2020 \text{ MPa}$. Se obtiene: $P = 4.85 \text{ N}$.

Fuerza de montaje máxima

Usando la ecuación en (4)

$$\Delta\alpha = \tan^{-1} \frac{\delta}{L_b} \quad . \quad (4)$$

Y usando los valores: $\delta = y = 0.27 \text{ mm}$, $L_b = l = 4.5 \text{ mm}$. Se obtiene:

$$\Delta\alpha = 3.43^\circ .$$

Luego en (5), reemplazado los valores obtenidos, se obtiene:

$$\alpha_{efectivo} = \alpha_{diseño} + \Delta\alpha = 28.43^\circ \quad . \quad (5)$$

Usando la ecuación en (6):

$$F_{montaje-max} = P * \frac{\mu + \tan \alpha_{efectivo}}{1 - \mu * \tan \alpha_{efectivo}} \quad , \quad (6)$$

y usando el valor de coeficiente de fricción: $\mu_{(ABS)} = 0.5$. Se obtiene:

$$F_{montaje-max} = P * 1.43 = 6.94 \text{ N} \quad .$$

El *enclosure* tendrá 4 *snap-fits*, por lo tanto, la fuerza de montaje total es:

$$F_{montaje-max-TOTAL} = 4 * 6.94 \text{ N} = 27.76 \text{ N} .$$

El valor de fuerza aceptable para ensamblaje, el cual se puede ver en la figura C.5 en el Anexo C, con referencia a una posición de ensamblaje con el antebrazo a 30° y el brazo a 45° para un objeto de tamaño medio es de 9.69 kgf (95 N); por lo que se puede concluir que el diseño es viable.

En la figura 3.40 se muestra la vista lateral del *enclosure*, en el cual se realiza una mordedura para poder observar el corte interior en el cuerpo superior (2) del *enclosure* donde debe ensamblarse el *snap-fit*, el cual se encuentra en la parte inferior (1) del *enclosure*. Como se mencionó anteriormente, el *enclosure* cuenta con cuatro *snap-fits*, por lo cual también cuenta con cuatro de estos cortes interiores para poder ensamblarlo, el ensamble para este *enclosure* es temporal, ya que se requerirá retirar la batería LiPo cuando el dispositivo no esté siendo usado. También se realizó otra mordedura, en la cual se observa el ensamblaje de un broche a presión en una de las extensiones del *enclosure*.

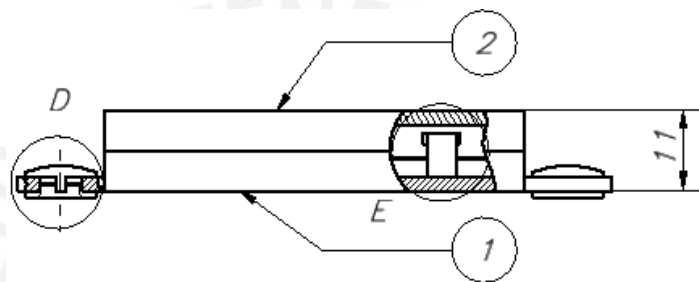


Figura 3.39: Vista lateral enclosure - batería
Fuente: Propia

Plano sub-ensamble *enclosure* parte electrónica

En la figura 3.41 se muestra la vista isométrica del *enclosure* que alberga la parte electrónica. Las dimensiones generales del *enclosure* son 43 mm de ancho, 59 mm de largo y 17 mm de altura, con espesor de 2.5 mm. Este cuenta con tres aberturas, una para el conector USB en caso sea necesario actualizar el código, una abertura para conectar los sensores de flexión y la última, para conectar la batería LiPo.

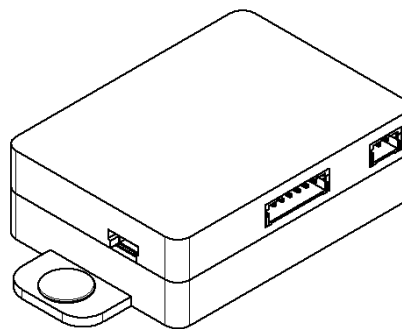


Figura 3.40: Vista isométrica enclosure - electrónica
Fuente: Propia

A continuación, en la figura 3.42 se muestra la lista de las partes que componen el ensamblaje del *enclosure* de la parte electrónica. En las siguientes figuras, se presentarán los números correspondientes a las partes descritas en esta lista.

12	1	MÓDULO Wi-Fi ESP-01
11	4	SEPARADOR DE PCB PST-1-01
10	1	ARDUINO NANO
9	1	IMU MPU-9250
8	1	REGULADOR STEP-DOWN D24V5F3
7	1	REGULADOR STEP-UP U3V12F5
6	1	DISEÑO PCB
5	1	CONECTOR JST PH 6 PINES
4	1	CONECTOR JST PH 2 PINES
3	2	BROCHES DE PRESION
2	1	ENCLOSURE SUPERIOR ELECTRÓNICA
1	1	ENCLOSURE INFERIOR ELECTRÓNICA
POS.	CANT.	DESCRIPCIÓN

Figura 3.41: Partes de *enclosure* - electrónica
Fuente: Propia

En la figura 3.43 se puede ver la vista principal del *enclosure*, en la cual se realizó dos detalles para mostrar el *snap-fit* en el detalle B (figura 3.44) y la abertura para la conexión de la placa Arduino Nano en el detalle C (figura 3.45).

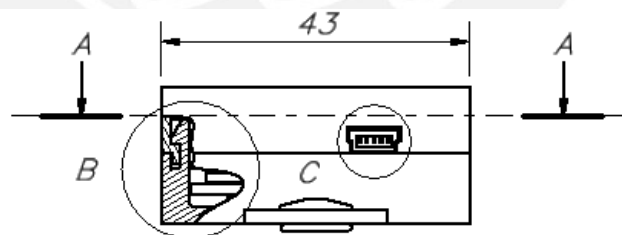


Figura 3.42: Vista principal *enclosure* - electrónica
Fuente: Propia

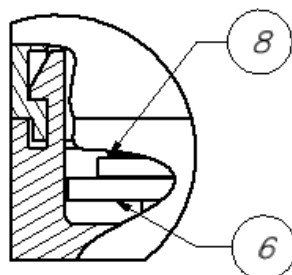


Figura 3.43: Detalle B
Fuente: Propia

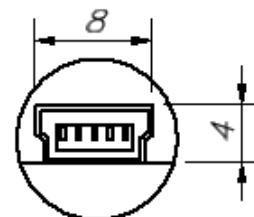


Figura 3.44: Detalle C
Fuente: Propia

En la vista principal se realizó un corte A-A, cuya vista se muestra en la parte derecha de la figura 3.46, aquí se puede observar la posición de algunos de los sensores y partes que conforman el *enclosure* de la parte electrónica. Al igual que en el *enclosure* para la batería LiPo, este cuenta con extensiones en los extremos, donde se colocarán broches a presión para poder asegurar el *enclosure* al guante y que no se desprenda mientras que el usuario este usando el dispositivo. En la parte izquierda de la figura descrita, se muestra la posición de otras partes del ensamble.

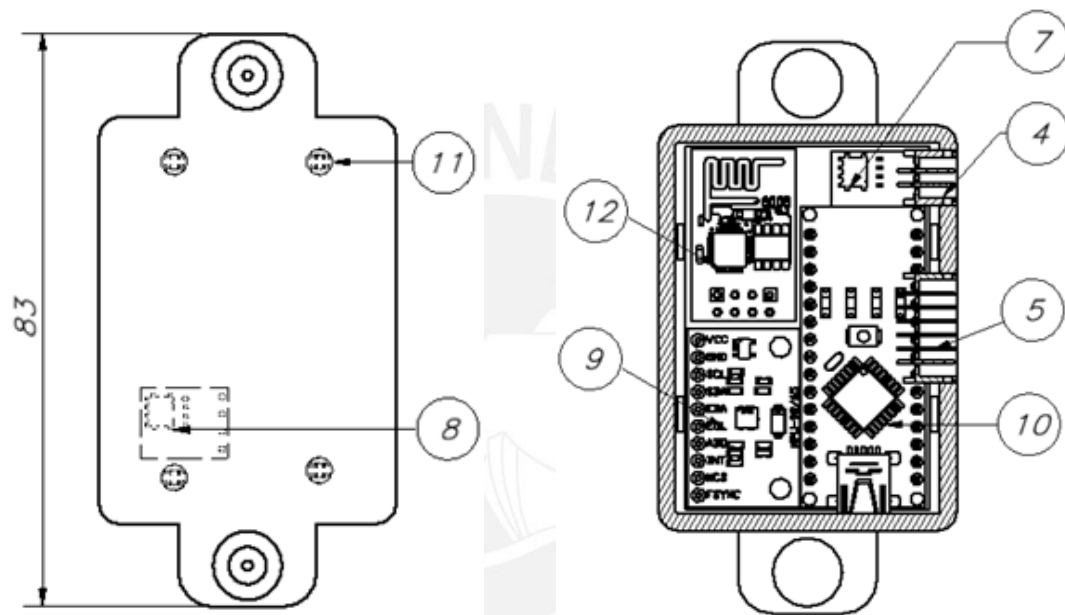


Figura 3.45: Corte A-A *enclosure* – electrónica y vista posterior
Fuente: Propia

Al igual que el *enclosure* diseñado previamente, este también cuenta con 4 *snap-fits* ubicados simétricamente y los cálculos se realizaron siguiendo los mismos pasos, con la única diferencia de que este *enclosure* deberá ser ensamblado solo una vez, por lo que se usa el valor total de la deformación unitaria permisible para los cálculos. El detalle de los cálculos se puede ver en el Anexo C. A continuación, se muestran los resultados.

Deformación admisible

Los valores propuestos para l , h y b se detallan a continuación:

$$l = 9 \text{ mm}; h = 1.5 \text{ mm}; b = 5 \text{ mm}; h' = 0.8 * 1.5 = 1.2 \text{ mm}.$$

Reemplazando los valores en la ecuación de deformación admisible se obtiene

$$y_{max} = 1.17 \text{ mm} ,$$

y se escoge $y = 1 \text{ mm}$. Para el ángulo de inserción $\alpha_{diseño}$ se escoge $\alpha_{diseño} = 30^\circ$.

Fuerza de deformación

Reemplazando los valores en la ecuación de fuerza de deformación: $P = 11.36 \text{ N}$.

Fuerza de montaje máxima

Reemplazando los valores para hallar el ángulo efectivo se obtiene $\Delta\alpha = 6.34$,

$$\alpha_{efectivo} = \alpha_{diseño} + \Delta\alpha = 36.34^\circ .$$

Luego, se calcula la fuerza de montaje máxima: $F_{montaje-max} = 22.15 \text{ N}$.

Finalmente, se obtiene la fuerza de montaje total $F_{montaje-max-TOTAL} = 88.60 \text{ N}$. Ya que la fuerza aceptable para ensamblaje de piezas pequeñas es 95 N, se puede concluir que el diseño es factible.

En la figura 3.47 se muestra la vista lateral del *enclosure*, en el cual se realizan dos mordeduras, una para poder observar uno de los cuatro separadores que sujetan la PCB al *enclosure*, estos separadores tienen una base adhesiva para facilitar su instalación.

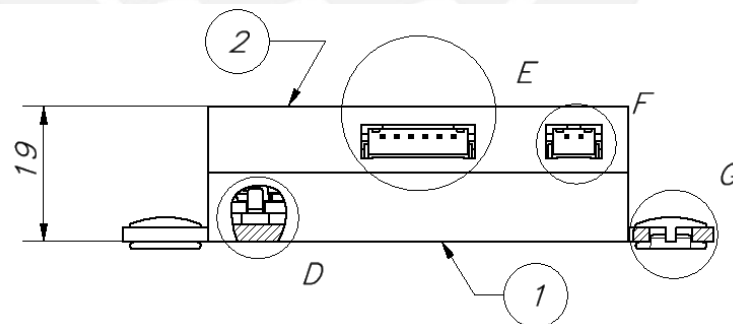


Figura 3.46: Vista lateral enclosure - electrónica
Fuente: Propia

La mordedura descrita se muestra en el detalle D (figura 3.48). La otra mordedura se muestra en el detalle G (figura 3.49) donde se observa el ensamble de uno de los broches a presión. Por otro lado, en esta vista se hacen dos detalles más para mostrar las aberturas, una para los conectores provenientes de los sensores de flexión, la cual se muestra en el detalle E (figura 3.50) y otra, la cual se muestra en el detalle F (figura

3.51), para el conector de la batería LiPo que se encuentra almacenada en el otro *enclosure* descrito previamente.

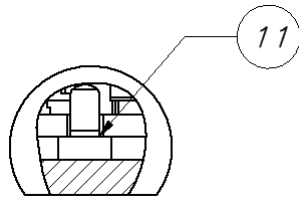


Figura 3.47: Detalle D
Fuente: Propia

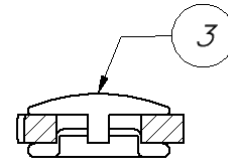


Figura 3.48: Detalle G
Fuente: Propia

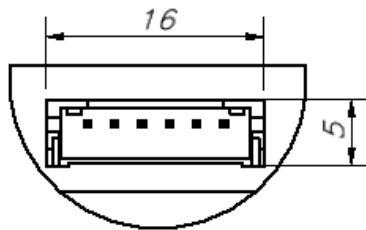


Figura 3.49: Detalle E
Fuente: Propia

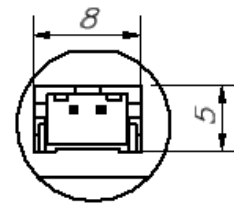


Figura 3.50: Detalle F
Fuente: Propia

A diferencia del *enclosure* para la batería LiPo, este posee tres aberturas, las cuales se muestran en las figuras 3.45, 3.50 y 3.51, con sus medidas correspondientes. En la figura 3.50, se puede apreciar la abertura para el conector jst de 6 pines, la cual tiene la mayor medida de las aberturas. Por lo tanto, este *enclosure* estará protegido contra cuerpos solidos de diámetro mayor a 16 mm; de modo que, según la tabla de características de protección IP (Anexo I), cuenta con protección IP 10 contra cuerpos de diámetro mayor a 50 mm.

3.3.3. Moldes de guante

En la figura 3.52 se puede ver el molde del guante, en el cual se indica donde se debe adjuntar el velcro para la sujeción de los sensores de flexión y donde se deben adjuntar los broches para la sujeción de los *enclosures*. Además, se indica donde se debe coser los *pockets* donde se colocan los sensores de flexión, cuyo molde y demás adiciones del guante se encuentran en la figura 3.53. El tipo de hilo y tela que se usarán serán de algodón Pima, ya que sus fibras son más largas que las del algodón convencional, lo cual ocasiona que sea más suave y menos irritable para personas con piel sensible, lo que es fundamental, puesto que estará en contacto contante con la piel del usuario. Los mencionados materiales deberán contar con certificado GOTS (*Global Organic*

Textile Standard), el cual garantiza que no se ha utilizado sustancias perjudiciales en su proceso de fabricación.

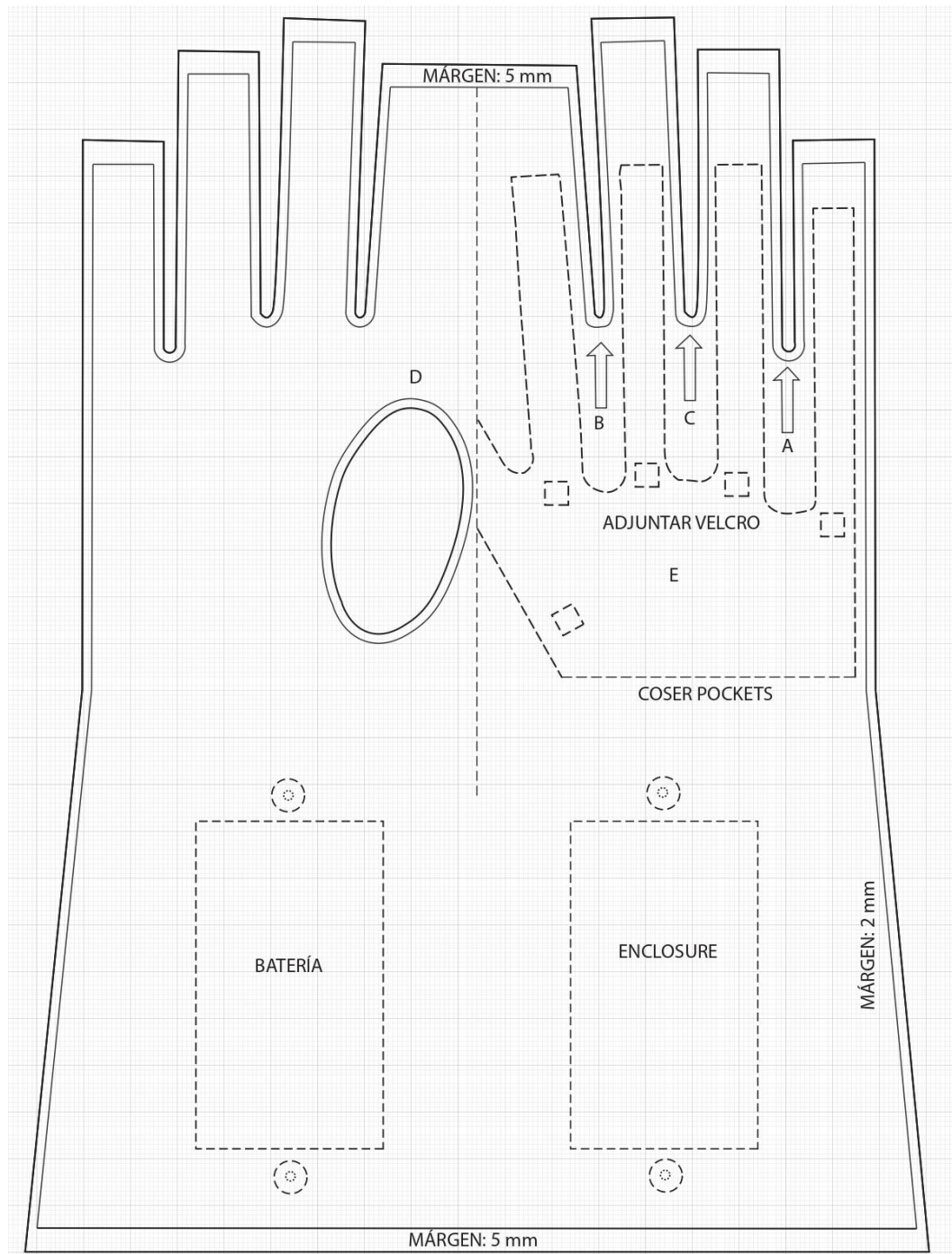


Figura 3.51: Diseño de Molde guante
Fuente: Propia

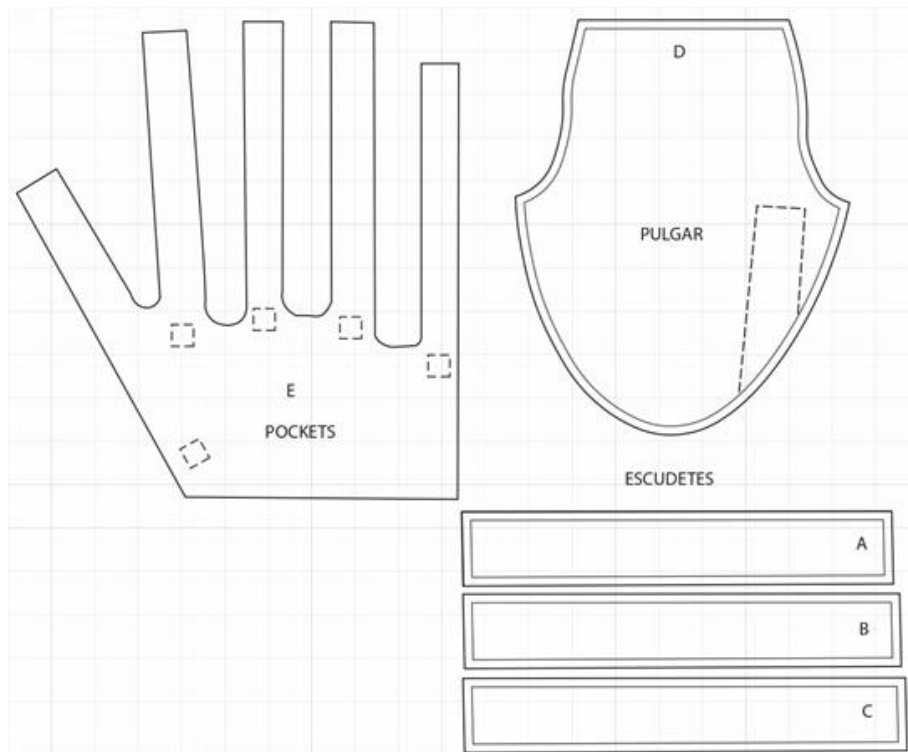


Figura 3.52: Diseño de Molde - pockets y adiciones
Fuente: Propia

En la figura 3.54 se muestra el molde para el cobertor que será de material Scuba. En este se indican las líneas donde se debe hacer los cortes para el encaje con los *enclosures*. Un extremo se cosera en la parte lateral del guante en la posición de la muñeca y en el otro extremo se colocará una pieza de velcro para poder asegurar los *enclosures*.

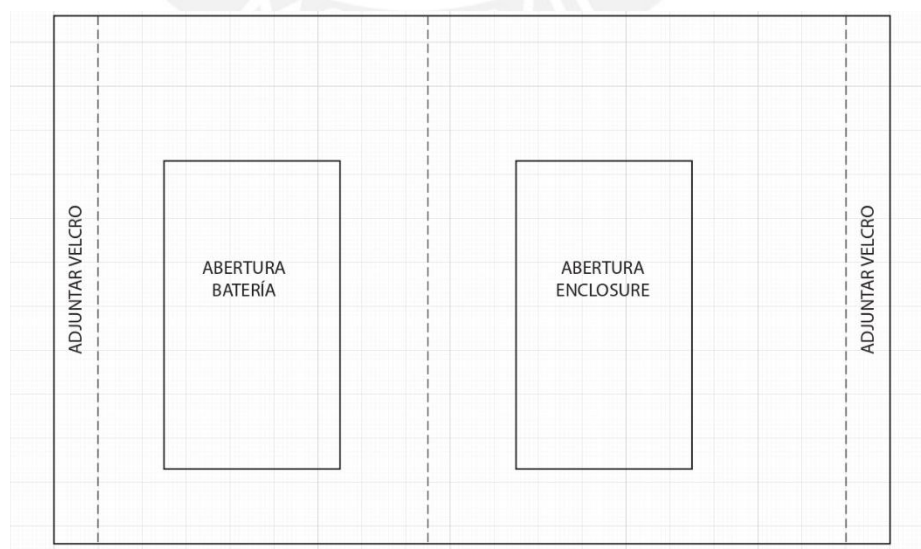


Figura 3.53: Diseño de molde - cobertor
Fuente: Propio

3.4. Diseño del software

En esta sección se desarrolla el software del sistema, es decir el software de comunicación para la encapsulación y envío de datos del guante a la PC y el de reconocimiento de gestos y vinculación con archivos de audio que el usuario usara en una PC. La base de datos para el reconocimiento de gestos estará basada en los datos que se recolectaran de los sensores de flexión y el sensor IMU, capturándose así los movimientos de los dedos y la orientación y movimiento de la mano. Para el reconocimiento de gestos, se elegirán las características (*features*), las cuales han sido previamente calculadas con los datos (series temporales). Para elegir el método de elección de características, se realizó un análisis con los métodos PCA y correlación de Pearson, así como también, se analizó el caso de construir el modelo con todas las características extraídas. Del mismo modo, para elegir el método de clasificación d gestos, se realizó un análisis con los métodos LDA, SVM y *Random Forest*. Por último, se vincularán gestos a archivos de audio para cumplir con el propósito del proyecto.

3.4.1. Diagramas de flujo

En esta sección se muestran los diagramas de flujo para la recolección de datos usando el guante, el reconocimiento de gestos, y la vinculación de audios y gestos en la PC.

Diagrama de flujo general de dispositivo (guante)

General

En la figura 3.55 se presenta el diagrama de flujo del dispositivo. En este se describen las funciones que implican inicializar el sistema, es decir configurar los pines e inicializar las variables, luego de esto se leen los datos de los sensores y se encapsulan para ser enviados a la PC a través de un módulo Wi-Fi para ser procesados posteriormente. Este es el único diagrama de flujo que describe el proceso que realiza el microcontrolador, el cual se encuentra instalado en el *enclosure* superior del guante. Para los demás procedimientos se tomó la decisión de realizarlos en la PC, ya que la placa Arduino Nano no tiene suficiente poder computacional, y en el caso de

realizarlos en la placa se corre el riesgo de que el dispositivo presente fallas y que los datos no sean precisos. Para ambos tipos de sensores, los valores se toman con una tasa de muestreo de 200 Hz (cada 5ms), esto se debe a que los algoritmos de procesamiento de movimiento deben ejecutarse a alta velocidad y este es el valor que recomienda el *datasheet* del sensor IMU MPU-9250. Luego, los valores se almacenan en un vector, el cual se obtiene de Encapsular_datos(). Dicho vector se actualiza cada 2ms y los datos se envían continuamente a la PC para que sigan siendo procesados y pueda reconocer un gesto.

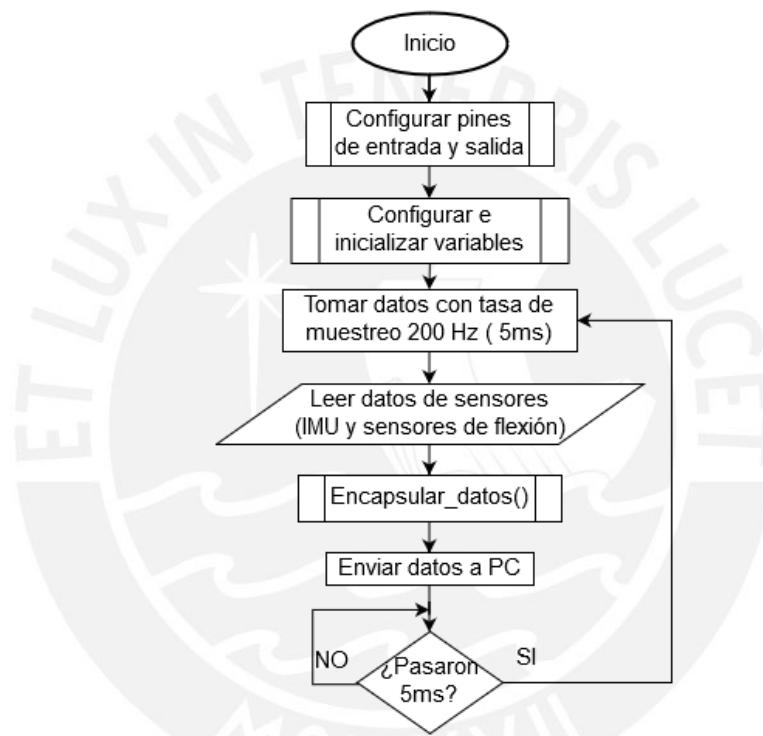


Figura 3.54: Diagrama de flujo general de dispositivo (guante) - recolección y envío de data
Fuente: Propio

Encapsular datos

En la figura 3.56 se muestra el diagrama de flujo correspondiente al bloque Encapsular_datos() en el diagrama de flujo del dispositivo (figura 3.55). El diagrama de flujo presentado se encarga de conseguir los valores de los 5 sensores de flexión, los cuales se obtienen con la función get_sf_Values(), y los valores para acelerómetro, magnetómetro y giroscopio en los ejes x, y, z, los cuales se obtienen con la función get_imu_Values() que utiliza el algoritmo de Sebastian Madgwick [32], estos valores se hallan a partir de los datos crudos obtenidos del sensor IMU. Finalmente, todos los datos obtenidos se almacenan en un vector (V_s).

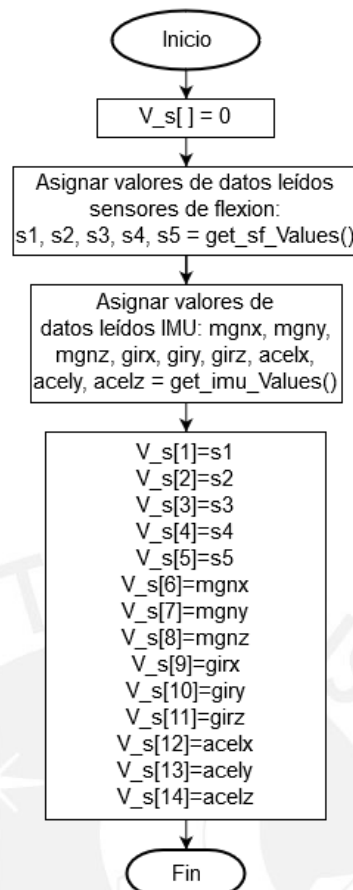


Figura 3.55: Diagrama de flujo Encapsular_data
Fuente: Propio

Diagramas de flujo de software (PC)

General

En la figura 3.57 se muestra el diagrama de flujo de reconocimiento de gesto y reproducción de audio. En este diagrama se configuran e inicializan las variables, luego se reciben los datos del guante por Wi-Fi, estos datos son capturados continuamente y se guardan en la variable `data_gestos []`, posteriormente se extraen las características de modo similar como se extraen de la base de datos para la generación del modelo de reconocimiento de gestos (figura 3.58). Subsiguientemente, se obtiene el gesto por medio del modelo generado previamente (figura 3.58). Este gesto se guarda en la variable `gesto` y se evalúa en la función `Obtener_audio()`, en la cual se compran con el vector de `gestos_vinculados` descrito en el diagrama de flujo de vinculación de gestos y audios (figura 3.59) para así obtener el audio al que este gesto está vinculado y almacenarlo en `audio_vinculado`. Finalmente, se reproduce el audio en un software externo.

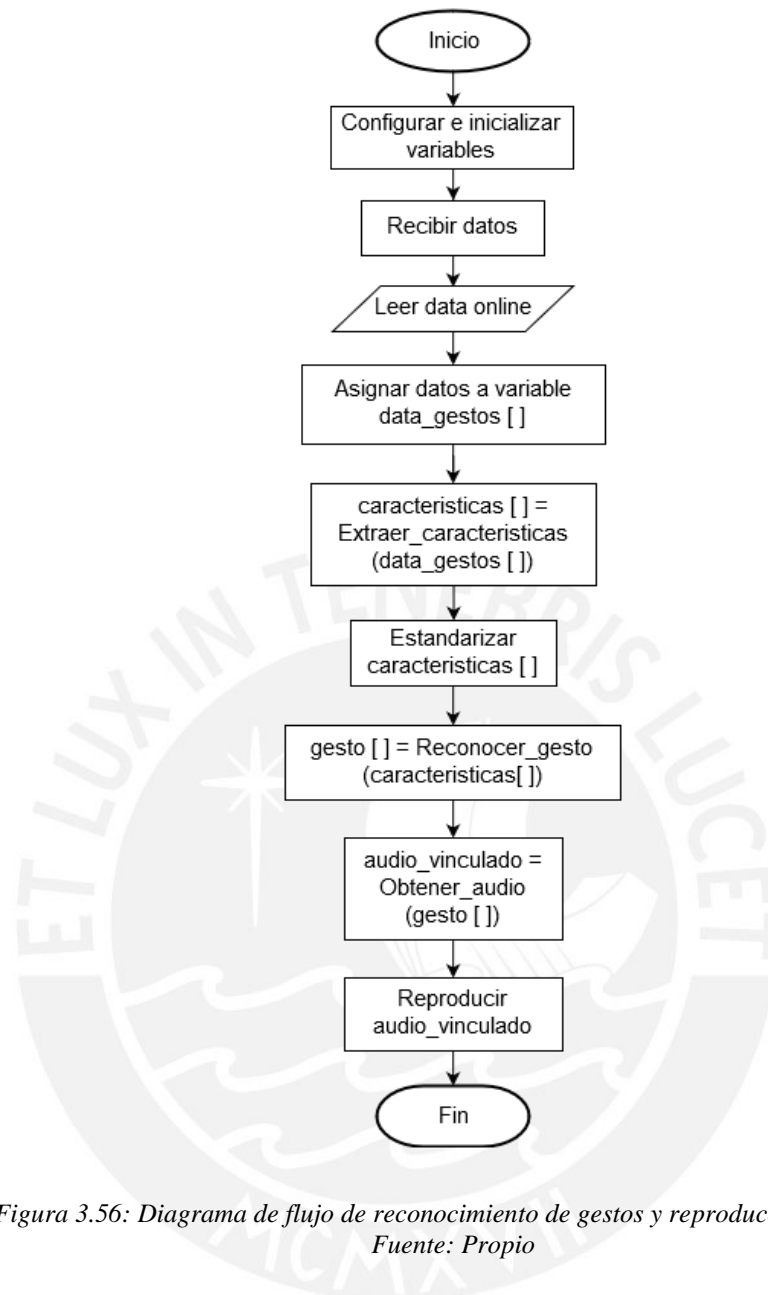


Figura 3.56: Diagrama de flujo de reconocimiento de gestos y reproducción de audio
Fuente: Propio

Entrenamiento y validación

En la figura 3.58 se muestra el diagrama de flujo para el entrenamiento y validación de reconocimiento de gestos. En este se trabaja con una base de datos de series de tiempo que contendrá los gestos y por cada lectura de estos se tendrá cinco valores pertenecientes a los sensores de flexión y nueve valores pertenecientes al sensor IMU. En primer lugar, se extraen las características, las cuales se detallarán en el siguiente capítulo. Luego, se obtendrán los gestos que correspondan a las características con la función `Obtener_gesto()`. La matriz de características seleccionadas y el vector de gestos se usarán para generar el modelo de reconocimiento de gestos, se tomará un porcentaje de datos para validar el modelo y otro para entrenarlo. Seguidamente, se

estandarizarán las características de entrenamiento y validación. Los datos de entrenamiento se usarán para generar el modelo y luego se usarán los datos de validación para verificar que el modelo reconozca el gesto y con qué porcentaje lo hace.

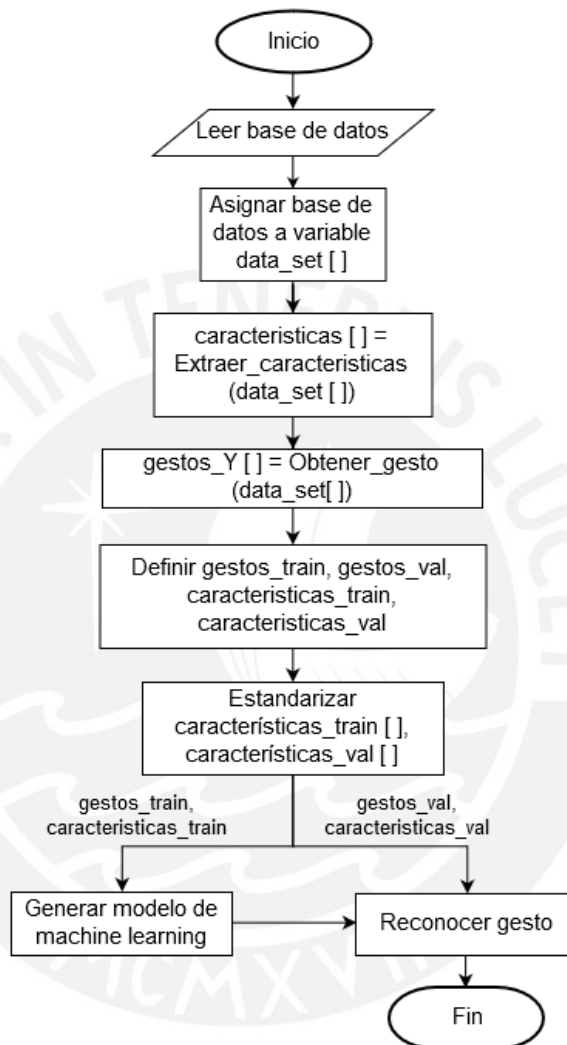


Figura 3.57: Diagrama de flujo para obtención del modelo de reconocimiento de gestos
Fuente: Propio

Vinculación de gestos con audio

En la figura 3.59 se muestra el diagrama de flujo de la vinculación de gestos con audio. En este se leen los vectores de gestos_guardados [] y de audio_guardado [], estos valores se podrán vincular entre ellos para luego ser guardados en el vector gestos_vinculados [].



Figura 3.58: Diagrama de flujo de vinculación de gestos y audios
Fuente: Propio

Al finalizar este capítulo, se concluye el diseño del guante inteligente, habiéndose desarrollado la integración de guante y software. Así como también, explicando el funcionamiento del guante y desarrollando el diseño electrónico, mecánico y de software. En el siguiente capítulo, se desarrollará el tema de reconocimiento de gestos.

CAPÍTULO 4

RECONOCIMIENTO DE GESTOS DE MANOS

El reconocimiento de gestos de manos es un campo emergente que tiene una amplia variedad de aplicaciones en los campos de automatización, animaciones 3D, reconocimiento de lenguaje de señas e interacción entre computadora y usuario. La aplicación específica de esta tesis es reconocimiento de gestos, para que cuando el modelo construido reconozca el gesto realizado un archivo de audio sea reproducido. Los tres métodos populares para el reconocimiento de gestos de manos se basan en visión, EMG (Electromiografía), sensores de flexión y sensores IMU. En el capítulo 2 se escogió como solución usar sensores de flexión en conjunto con un sensor IMU. El diseño para la solución escogida se detalla en el capítulo 3, así como también los componentes que lo conforman.

Este capítulo está dividido en el fundamento teórico para análisis de base de datos, la obtención de base de datos para entrenamiento y validación del modelo, y construcción del modelo de reconocimiento de gestos. Cabe resaltar que en la sección de fundamento teórico se describe el proceso para obtener una base de datos para la solución óptima, mas no se obtiene la base de datos, ya que para este proyecto no se realizó un prototipo para poder obtener una base de datos propia. Por consiguiente, en la sección de obtención de base de datos para entrenamiento y validación del modelo se describe las bases de datos que se usaran para la construcción de modelo de reconocimiento de gestos, las cuales fueron obtenidas del repositorio de Machine

Learning de la Universidad de California en Irvine [29]. Además, ya que no se pudo encontrar una base de datos con señales de sensores de flexión e IMU para reconocimiento de gestos, cabe resaltar que los datos de la base de datos utilizada fueron recolectados de señales obtenidas de sensores EMG.

4.1. Fundamento teórico para análisis de base de datos

Obtención de base de datos para concepto solución

En esta sección se detalla cómo se realizará la obtención de datos de los sensores pertenecientes a la solución escogida y descrita en el capítulo anterior.

Sensores de flexión

Ya que se utilizará un Arduino Nano, la obtención de los datos se realizará en Arduino IDE. En la figura 4.1 se presenta un código para obtener datos de un sensor de flexión.

```
int flexSensorPin = A5; //analog pin 5
void setup(){
  Serial.begin(9600);
}
void loop(){
  int flexSensorReading = analogRead(flexSensorPin);
  Serial.println(flexSensorReading);
  delay(250);
}
```

Figura 4.1: Código sensor de flexión (Arduino)

Fuente: Propia

Como se puede observar en el código, se utiliza el comando `map` para normalizar los valores entre 0 y 90, esto se realiza para obtener el ángulo de flexión del dedo.

IMU

En el caso del sensor IMU, existen muchos métodos para convertir los datos crudos a valores con los que se puede trabajar, sin embargo, para el trabajo posterior a este proyecto se propone utilizar el algoritmo de Sebastian Madgwick [32]. El algoritmo utiliza representaciones en cuaterniones, permitiendo que los datos del magnetómetro y del acelerómetro sean usados en un algoritmo de descenso de gradiente optimizado

para calcular la dirección el error de la medida el giroscopio como una derivativa de cuaterniones.

Luego de obtener lecturas de todos los sensores, a una cierta frecuencia (200 Hz), se da paso a la creación de la base de datos, en la cual los datos se tomarán de diferentes sujetos para diferentes gestos por un periodo pequeño (5 segundos), repitiéndolos por una cierta cantidad de veces y cada lectura de datos se almacenará en una matriz correspondiente a cada sensor. Luego de esto se seguirá con el proceso que se detalla en la siguiente sección para la construcción del modelo de reconocimiento de gestos.

Conceptos a evaluar para calificación de modelos

Ya que el modelo de machine learning será implementado en Python, se hará uso de las herramientas que proporciona para evaluar el rendimiento de un modelo. A continuación, se presentan 3 conceptos a evaluar.

Precisión (*Precision*)

La precisión es la capacidad de un clasificador para no etiquetar una instancia como positiva que en realidad es negativa. Para cada clase se define como la proporción de verdaderos positivos con respecto a la suma de verdaderos y falsos positivos.

Exhaustividad (*Recall*)

Exhaustividad es la capacidad de un clasificador para encontrar todas las instancias positivas. Para cada clase se define como la relación de verdaderos positivos a la suma de verdaderos positivos y falsos negativos.

Valor-F (*F1-score*)

El valor-F es una media armónica ponderada de precisión y exhaustividad, de modo que el mejor puntaje es 1.0 y el peor es 0.0. En términos generales, los valores-F son más bajos que las medidas de precisión, ya que incorporan valores de precisión y exhaustividad en sus cálculos. Como regla general, el promedio ponderado de valor-F debe usarse para comparar modelos de clasificador y no valores de exactitud global.

4.2. Obtención de base de datos para entrenamiento y validación

Para la construcción de los modelos se usaron 2 bases de datos obtenidas del repositorio de Machine Learning de la Universidad de California en Irvine [30] [31].

Para ambas bases de datos se tomaron en cuenta 6 gestos, los cuales se muestran en la figura 4.2. Además, la toma de datos fue colectada usando una frecuencia de muestreo de 500 Hz. Las señales se filtraron con un filtro pasa banda Butterworth con un valor de frecuencia de corte inferior y superior de 15 Hz y 500 Hz respectivamente y un filtro de rechazo de banda de 50 Hz para eliminar artefactos electrocardiográficos, como interferencia de la fuente de alimentación. Las señales fueron tomadas de dos sensores EMG y transmitidas a un sistema EMG de 2 canales. El experimento para la obtención de datos consistió de agarrar repetidamente diferentes elementos, con los cuales se realizarán los gestos de la mano a tomarse en cuenta. La velocidad y la fuerza se dejaron a voluntad del sujeto [31].

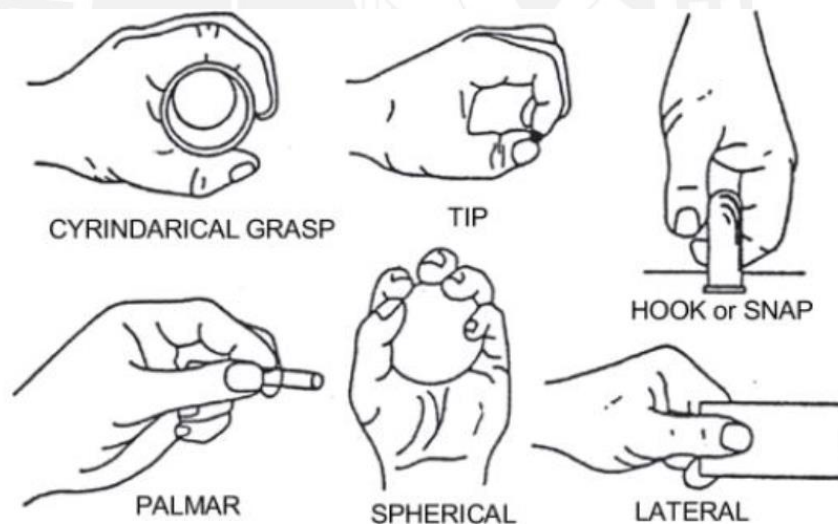


Figura 4.2: Gestos correspondientes a base de datos
Fuente: [31]

Los sensores EMG se colocaron en los músculos del antebrazo: Extensor Carpi Radialis (azul) y Flexor Carpi Ulnaris (rojo) como se puede ver en la figura 4.3.

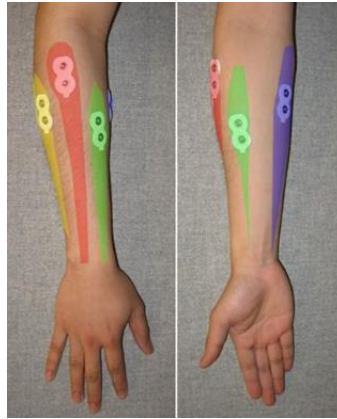


Figura 4.3: Posición de sensores EMG
Fuente: [31]

Para las pruebas realizadas se asignó un número del 0 a 5 a cada gesto, de modo que se pudo realizar el análisis los datos (ver tabla 4.1)

Tabla 4.1: Codificación para gestos
Fuente: Propia

Gesto	Numero asignado
<i>Cylindrical Grasp</i>	0
<i>Hook</i>	1
<i>Lateral</i>	2
<i>Palmar</i>	3
<i>Spherical</i>	4
<i>Tip</i>	5

Descripción de bases de datos

Para la base de datos 1, se tomaron datos de 5 sujetos, 3 mujeres y 2 hombres, con edades en el rango de 20 a 22 años. Se pidió a los sujetos que realicen los 6 gestos, 30 veces cada uno. Los datos se tomaron por 2 canales con tiempo de grabación de 6 segundos por gesto y frecuencia de muestreo de 500 Hz. Los datos se almacenaron en un archivo mat por sujeto con 12 matrices de tamaño 30 x 3000 por cada canal y gesto. Esta base de datos será usada para el modelo genérico de usuarios múltiples.

Para la base de datos 2, se tomaron datos de un mismo sujeto de 22 años por 3 días consecutivos, en este caso, el sujeto realizo cada uno de los 6 gestos 100 veces por día. Los datos se tomaron por 2 canales con tiempo de grabación de 5 segundos por gesto y frecuencia de muestreo de 500 Hz. Los datos se almacenaron en un archivo

mat por día con 12 matrices de tamaño 100 x 2500 por cada canal y gesto. Ya que esta base de datos cuenta con varias grabaciones de un solo sujeto, se usará para el modelo de usuario específico

4.3. Construcción del modelo de reconocimiento de gestos

La construcción de un modelo de reconocimiento de gestos consiste básicamente en alimentar datos a un algoritmo de *machine learning*, el cual se entrena con dichos datos y aprende a asociar un gesto de mano con una clase o etiqueta. Para realizar esto, se utilizan datos de entrenamiento y datos de validación, y en caso que no se tenga una base de datos específica para entrenamiento y validación, se deberá separar la base de datos, utilizando un porcentaje para datos de validación y otro para datos de entrenamiento. Los datos de entrenamiento se utilizan para entrenar el algoritmo. Luego de esto, se utilizan los datos de validación para evaluarlo y se comparan los resultados obtenidos (gestos) con los valores reales (gestos) pertenecientes a los datos de validación.

En esta sección se hace una comparación del puntaje de exactitud que se obtiene al usar los diferentes métodos de selección de características y de clasificación en un modelo genérico para usuarios múltiples y un modelo para usuario específico. Para la evaluación de los modelos, se tomó en cuenta 20% de los datos para validación y el resto para entrenamiento.

4.3.1. Características para reconocimiento de gestos

Ya que el problema de reconocimiento de gesto es uno con series temporales, se usarán 10 características en el dominio del tiempo, las cuales son descritas en la tabla 4.2. Como se mencionó anteriormente la base de datos que se utilizara para construir el modelo pertenece a señales EMG, por lo cual también se incluirán características que sean exclusivas para sensores EMG, las cuales fueron escogidos de acuerdo al artículo *EMG based classification of basic hand movements based on time-frequency features* [30]. El código para conseguir las características se encuentra en el Anexo F.

Tabla 4.2: Características para reconocimiento de gestos
Fuente: [30]

Característica	Descripción
<i>Zero Crossing (ZC)</i>	ZC cuenta las veces que la señal cambia de signo. Dadas dos muestras de amplitud de EMG contiguas, ZC se calcula como $\sum f(x)$. $f(x) = \begin{cases} 1, & (x_k > 0 \text{ y } x_{k+1} < 0) \text{ o } (x_k < 0 \text{ y } x_{k+1} > 0) \\ 0, & \text{otros} \end{cases}$
<i>Waveform Length (WL)</i>	WL es una variación acumulativa de una señal EMG que puede indicar el grado de variación sobre la señal. Y está dada por $WL = \sum x_{k+1} - x_k $
<i>Willison Amplitude (WAMP)</i>	WAMP es el número de conteos para cada cambio de la amplitud de la señal EMG entre dos muestras adyacentes que exceden un umbral definido. Está dada por $f(x) = \begin{cases} \sum f(x_{k+1} - x_k), & x > \text{umbral} \\ 0, & \text{otros} \end{cases}$
<i>Slope Sign Changes (SSC)</i>	SSC cuenta las veces que la pendiente de la señal cambia de signo. Dadas tres muestras de amplitud de EMG contiguas, el número de SSC se calcula como $\sum f(x)$. $f(x) = \begin{cases} 1, & (x_k < x_{k+1} \text{ y } x_k < x_{k-1}) \text{ o } (x_k > x_{k+1} \text{ y } x_k > x_{k-1}) \\ 0, & \text{otros} \end{cases}$
<i>Integrated Electromyogram (IEMG)</i>	IEMG es el valor promedio de los valores absolutos de EMG. $IEMG = \frac{1}{N} \sum_{k=1}^N x_k $
Varianza (VAR)	VAR es una medida de la densidad de potencia de la señal EMG que se calcula con la siguiente ecuación. $VAR = \frac{1}{N-1} \sum_{k=1}^N x_k^2$
Mediana	Si el número de datos en una secuencia (vector) es impar, la mediana es el número medio en dicha secuencia, pero en el caso que el número de datos sea par, la mediana es el promedio de los dos números medios.
Desviación Estándar (STD)	La desviación estándar es la raíz cuadrada de la varianza (VAR).
Oblicuidad (<i>Skewness</i>)	La oblicuidad es una distribución dada por $S = \frac{E(x - \mu)^3}{\sigma^3}$
Kurtosis	La kurtosis es una distribución dada por $k = \frac{E(x - \mu)^4}{\sigma^4}$

4.3.2. Métodos de selección de características y de clasificación

A continuación, se describen el método de selección de características que se utilizaran para este estudio, en el Anexo D se describe otro método que también se consideró usar en un primer instante. La selección de características se utiliza para simplificar un modelo y reducir el tiempo de entrenamiento, y así este sea más fácil

de interpretar. Sin embargo, no siempre se obtienen los mejores resultados cuando se utiliza una de estas técnicas, ya que algunas veces no se cuenta con demasiadas características y es mejor utilizar todas para la construcción del modelo. En cualquiera de los casos, se deberán estandarizar las características de entrenamiento y validación, previamente separadas. El modelo se construirá en Python, por lo que para estandarizar las características se utilizará el comando `StandardScaler`, el cual convierte las características a valores que como conjunto tienen media = 0 y varianza = 1, lo cual es un requerimiento para el rendimiento óptimo de muchos algoritmos de machine learning. El comando `StandardScaler` equivale al método de estandarización; así:

$$z = \frac{x - \mu}{\sigma},$$

con $\mu = \frac{1}{N} \sum_{i=1}^N x_i$ y $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$.

Principal Component Analysis (PCA)

PCA se ha utilizado ampliamente para la generación de características y la reducción de dimensionalidad en el reconocimiento de patrones, en este caso gestos. PCA transforma linealmente el espacio original proyectando los datos de dimensión d sobre los vectores propios l ($l \leq d$) de su matriz de covarianza correspondiente a los valores propios mayores. Incluso si se conserva todo el conjunto de vectores propios, lo que resulta en una falta de reducción de la dimensionalidad, esto puede llevar a una mejora del rendimiento de clasificación debido a que el nuevo conjunto de características es no correlacionado. A continuación, se muestra cómo implementar PCA en Python (figura 4.4). En el comando PCA se coloca 0.99, esto hace que se escoja el mínimo número de componentes principales para conservar el 99% de la varianza. Luego, se entrena el algoritmo con los datos de entrenamiento y; por último, se aplica el método PCA a los datos de entrenamiento y a los datos de validación.

```
from sklearn.decomposition import PCA
pca = PCA(0.99)
pca.fit(X_train)
X_train_t = pca.transform(X_train)
X_val_t = pca.transform(X_val)
```

Figura 4.4: Código PCA (Python)
Fuente: Propia

A continuación, se describen los métodos de clasificación que se utilizarán para construir un modelo para este estudio. Para construir un modelo se utilizan los datos de entrenamiento, consiguientemente se evalúa este modelo con las características de los datos de validación y se compara en que porcentaje se obtuvo respuestas acertadas para el objetivo, en este caso gestos.

Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) es un método usado en machine learning y otras áreas para encontrar una combinación lineal de características que diferencie dos o más clases de objetos o eventos. La combinación resultante puede ser usada como un clasificador lineal. LDA está relacionado con el análisis de varianza (ANOVA) y análisis de la regresión, el cual también tiene como objetivo expresar una variable dependiente como una combinación lineal de otras características. Sin embargo, ANOVA utiliza variables independientes categóricas y una variable dependiente continua, mientras que LDA tiene variables independientes continuas y una variable dependiente categórica. A continuación, se muestra cómo implementar la construcción de un modelo usando LDA en Python (figura 4.5). Se puede observar que el modelo se construye en base a los datos de entrenamiento y luego este modelo se usa para predecir los resultados utilizando los datos de validación.

```
lda=LinearDiscriminantAnalysis(solver="svd",store_covariance=True)
.fit(X_train_t, y_train)
y_pred = lda.predict(X_val_t)
```

Figura 4.5: Código LDA (Python)

Fuente: Propia

Support Vector Machine (SVM)

SVMs (*Support Vector Machines*) son técnicas estadísticas diseñadas para separar o clasificar datos usando límites lineales o no lineales entre diferentes clases de puntos. Un modelo eficiente se puede obtener a través de funciones kernel sin perder propiedades de unicidad de convergencia. SVMs son adecuados para problemas de reconocimiento o de gestos gracias a los típicos kernels (lineal, función de base radial

(rbf), gaussiano, polinomial, etc.). En la siguiente sección se comprobará que kernel entrega mejores resultados y como es que sus valores de parámetros influyen en estos.

Random Forest

Al combinar los predictores de árboles de modo que cada árbol dependa de los valores de un vector aleatorio muestreado de manera independiente y con la misma distribución para todos los árboles en el bosque, se obtiene la técnica *de Random Forest*, la cual ha sido usada en reconocimiento de gestos en tiempo real en numerosas ocasiones.

4.3.3. Comparación de diferentes modelos

En esta sección, se desarrolla y describe los 2 modelos a considerarse en el proyecto: modelo para usuarios múltiples y modelo para usuario específico. Además, se compara diferentes clasificadores para elegir el mejor para cada modo de usuario.

4.3.3.1. Modelo genérico para usuarios múltiples

En este modelo se utilizaron los datos de los 5 sujetos de la base de datos 1, la cual consiste de 5 archivos mat. Las primeras pruebas realizadas se hicieron tomando diferentes números de muestra de datos para la generación de archivos de las características. Cada una de las 12 matrices perteneciente en un archivo mat es de tamaño 30 x 3000, por lo cual se utilizó 24, 12, 6, 3 y 1 muestra por la duración de cada grabación, es decir, para 24 muestras, el tamaño de cada muestra era de 125, y así sucesivamente.

Como se mencionó previamente, se tomaron los datos por 2 sensores EMG, consiguientemente, por cada sujeto y tamaño de muestra se generan 2 archivos txt con las 10 características descritas anteriormente, para luego ser unidas en una sola matriz que contenga todas las características de los dos canales y de todos los 5 sujetos. Además, se generó un vector que contenía las etiquetas de los gestos utilizando los

números del 0 al 5 para representarlos. Antes de realizar cada una de las pruebas, se separaron los datos de la matriz de características en datos de entrenamiento y validación, tomando el 20% para validación y el 80% restante para entrenamiento. Luego de esto, se estandarizan ambas matrices de características como se muestra en la figura 4.6.

```
X_train, X_val, y_train, y_val = train_test_split(features_allCH,
gestos, test_size=0.20, shuffle=True, random_state=25)
scaler = StandardScaler()
scaler.fit(X_train)

X_train = pd.DataFrame(scaler.transform(X_train))
X_train.columns=features
X_val = pd.DataFrame(scaler.transform(X_val))
X_val.columns=features
```

Figura 4.6: Código modo usuarios múltiples - separación y estandarización de datos (Python)
Fuente: Propia

Las primeras pruebas consistieron en el uso o no de del método PCA para selección de características y los métodos LDA y SVM con kernel lineal para modelos de clasificación. En estas pruebas se llegó a la conclusión que en cualquiera de los casos siempre se obtuvo mejores resultados cuando se tomó 1 sola muestra de tamaño 3000, es decir cuando se evalúan los 6 segundos de toma de datos como conjunto. Con este tamaño de muestra el tamaño de la matriz de características, concatenando las dos matrices de cada canal, fue de 900 x 20 y el tamaño del vector que contiene las etiquetas de los gestos fue de 900 x 1. Eligiendo que el tamaño de muestra sea 1, quedaron 4 opciones de las cuales se eligió como más apropiada cuando no se usa PCA y cuando se usa SVM con kernel lineal como clasificador. Los resultados de la opción más apropiada se muestran en la tabla 4.3 y los resultados de todas las opciones se pueden consultar en el Anexo E, los resultados mencionados fueron obtenidos usando los datos de validación.

Tabla 4.3: Resultados para clasificador SVM (lineal) sin PCA
Fuente: Propia

	Precision	Recall	F1-score
Resultados	0.7411	0.7333	0.7333

Antes de realizar las segundas pruebas, se aplicó el método PCA para 2 componentes principales para analizar en qué gestos podría haber dificultad en la clasificación. En la figura 4.7 se muestra el resumen de los datos, antes de aplicar el método las

características fueron estandarizadas. Se puede observar que el gesto 2 (verde) pertenece a alguna región de cualquier otro gesto, por lo que puede que sea difícil clasificarlo.

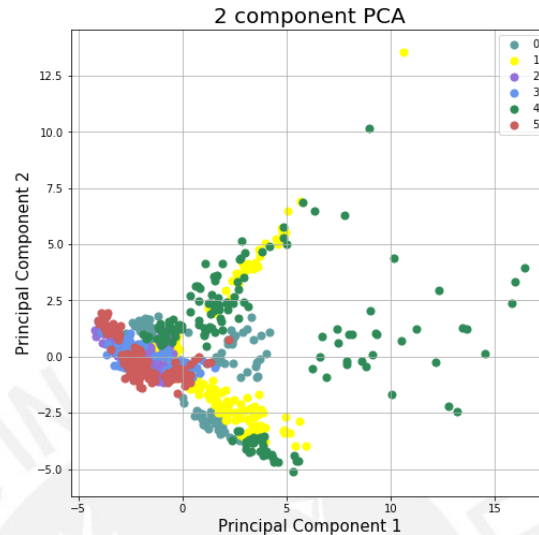


Figura 4.7: Análisis de 2 componentes principales – Usuarios múltiples
Fuente: Propia

Debido a los resultados de las primeras pruebas, las segundas pruebas se realizaron tomando 1 sola muestra de tamaño 3000, es decir que el tamaño de las matrices de características fue de 900 x 20. Se siguió el mismo procedimiento que en las primeras pruebas con respecto a la separación de datos de entrenamiento (80%) y validación (20%), y la estandarización de los mismos. Estas pruebas consisten en el análisis de los casos cuando para la selección de características se usó PCA o se usó todas las características y para la clasificación de modelos se usó el clasificador SVM con función de base radial o el clasificador *Random Forest*. Cuando se usa el clasificador SVM con función base radial se debe tomar en cuenta el parámetro C, el cual compensa la correcta clasificación de ejemplos de entrenamiento con la maximización del margen de la función de decisión, y el parámetro gamma, el cual define la influencia de un solo ejemplo de entrenamiento, influyendo más cuando el valor es mayor. En los casos donde se utilizó PCA, se decidió conservar el 99% de la varianza de las características y el procedimiento para transformar ambos sets de datos tomo aproximadamente un total de 3987 ms. A continuación, se muestran los resultados obtenidos para cada caso usando los datos de validación.

Para el clasificador SVM con función de base radial cuando no se utiliza PCA se escoge $C=3.1$ y $\gamma=0.4$ para obtener los resultados más óptimos. En la figura 4.8 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 1122 ms.

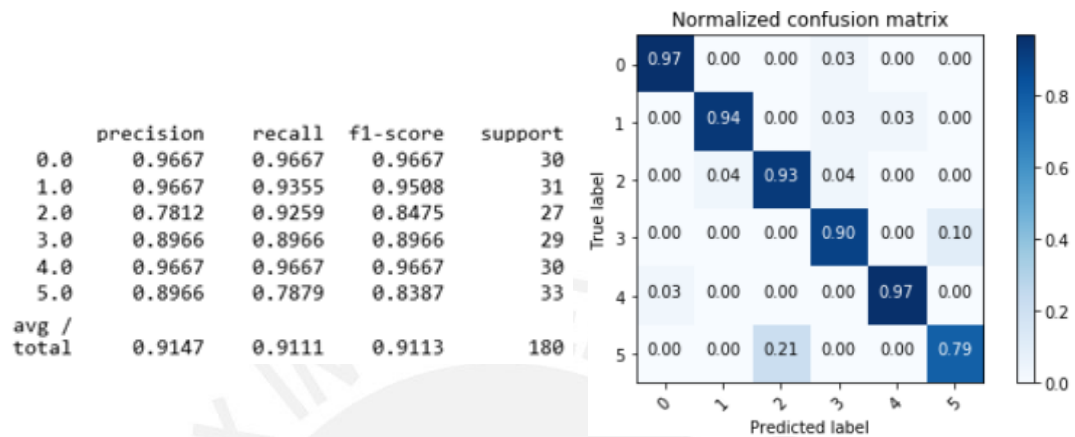


Figura 4.8: Reporte y matriz de confusión normalizada - SVM - Usuarios múltiples
Fuente: Propia

Para el clasificador SVM con función de base radial cuando se utiliza PCA se escoge $C=2.6$ y $\gamma=0.5$ para obtener los resultados más óptimos. En la figura 4.9 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 1033 ms, sin incluir el tiempo que tomó aplicar PCA.

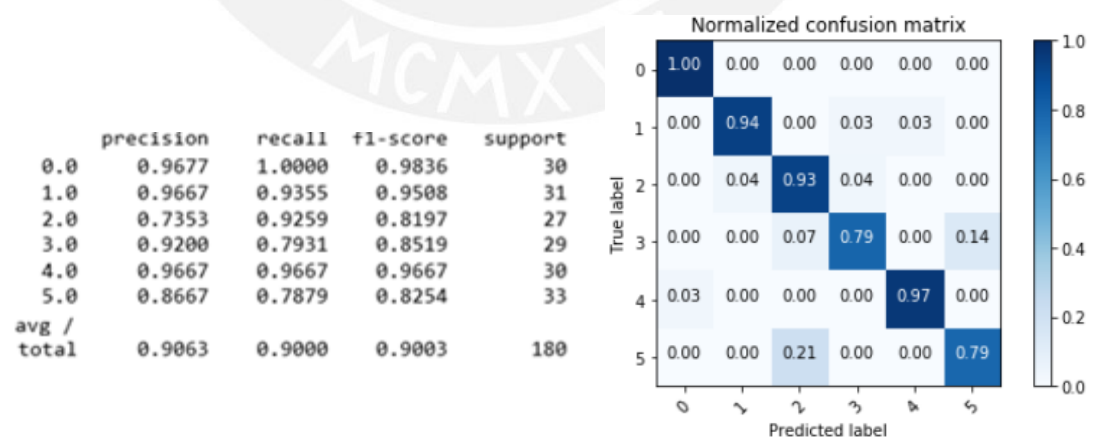


Figura 4.9: Reporte y matriz de confusión normalizada - SVM + PCA - Usuarios múltiples
Fuente: Propia

Para el clasificador *Random Forest* cuando se utiliza PCA se escoge $n_estimators=400$, $criterion='entropy'$. En la figura 4.10 se muestran el reporte

obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 2171 ms, sin incluir el tiempo que tomó aplicar PCA.

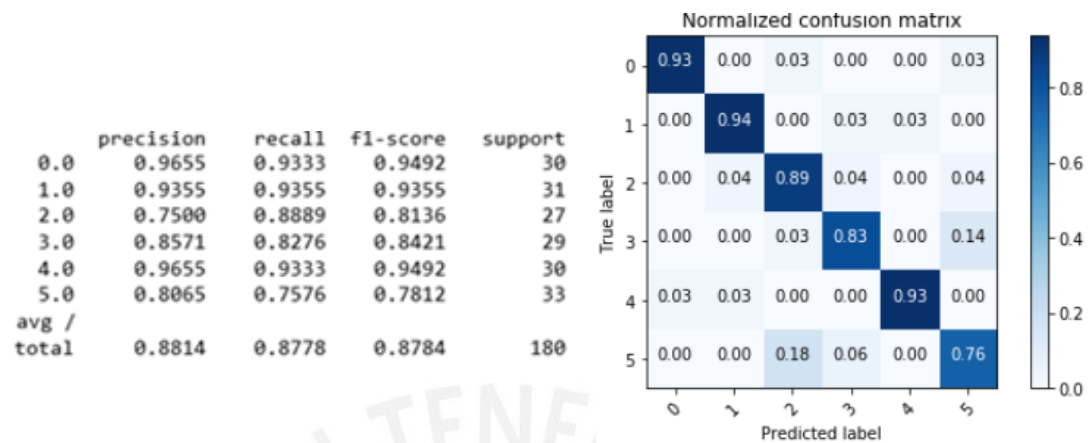


Figura 4.10: Reporte y matriz de confusión normalizada – Random Forest + PCA - Usuarios múltiples
Fuente: Propia

Para el clasificador *Random Forest* cuando no se utiliza PCA se escoge `n_estimators=300, criterion='entropy'`. En la figura 4.11 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 1706 ms.

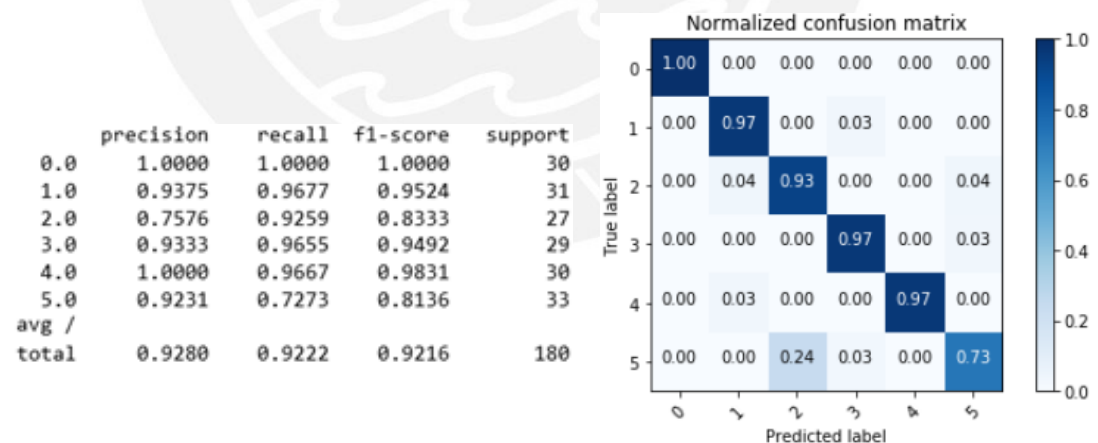


Figura 4.11: Reporte y matriz de confusión normalizada – Random Forest - Usuarios múltiples
Fuente: Propia

En la tabla 4.4 se muestra la comparación de los resultados de los 4 casos, se resalta los resultados más óptimos para cada resultado.

Tabla 4.4: Comparación de resultados – Usuarios múltiples
Fuente: Propia

Caso	Precision	Recall	F1-score	Tiempo (ms)
SVM	0.9147	0.9111	0.9113	1122
SVM + PCA	0.9063	0.9000	0.9003	5020
RF	0.9280	0.9222	0.9216	1706
RF + PCA	0.8814	0.8778	0.8784	6158

Se puede concluir que cuando se usa PCA el procedimiento toma más tiempo y, además, no se obtienen los mejores resultados, esto se debe probablemente a que la matriz de características no es de una dimensión grande, por lo cual no influye positivamente hacerla aún más pequeña. Además, como se previó en el análisis de PCA para 2 componentes principales, en los resultados de precisión del gesto 2 se obtuvo un porcentaje bajo con respecto a los demás gestos en todos los casos, por lo que se concluye que existen dificultades para clasificar el gesto 2 con esta base de datos. Con respecto a los clasificadores, SVM se ejecuta mucho más rápido que *Random Forest*, pero ya que este modelo será para usuarios múltiples se desea escoger el mejor resultado el cual sería el modelo construido con el clasificador *Random Forest*.

4.3.3.2. Modelo para usuario específico

En este modelo se utilizaron los datos del sujeto de la base de datos 2, la cual consiste de 3 archivos mat, uno por cada día que se hizo la toma de datos, cada archivo contiene 12 matrices de tamaño 100 x 2500. De manera similar que, en el modelo para usuarios múltiples, las primeras pruebas se hicieron tomando diferentes números de muestra de datos para la generación de archivos de las características, en este caso para 20, 10, 5, 2 y 1 muestra. Así, en el caso de tomar 20 muestras, el tamaño de cada muestra sería de 125. Por cada día y tamaño de muestra se generaron 2 archivos txt usando 10 características, para luego unir ambos archivos en una sola matriz que contenga todas las características de los dos canales y los 3 días. Luego, se separaron los datos de la matriz de características en datos de entrenamiento y validación, tomando el 20% para validación y el 80% restante para entrenamiento y se estandarizan ambas matrices de características. Se generó un vector conteniendo las etiquetas de los gestos utilizando los números del 0 al 5 para representarlos.

Las pruebas consistieron en la comparación del uso o no del método PCA para selección de características y el uso del método LDA para la clasificación. Al igual que en el modelo para usuarios múltiples, se llegó a la conclusión que siempre se obtenían mejores resultados cuando se toman solo una muestra de tamaño 2500, es decir cuando se evalúan los 5 segundos de toma de datos. Con este tamaño de muestra la matriz de características tiene un tamaño de 1800 x 20 y el vector que contiene las etiquetas de los gestos, 1800 x 1. La opción con la que se obtuvo mejores resultados fue cuando se construyó el modelo con LDA y sin aplicar el método PCA, los resultados se muestran en la tabla 4.5 y los resultados de las demás opciones se encuentran en el Anexo E, los resultados mencionados fueron obtenidos usando los datos de validación.

Tabla 4.5: Resultados para clasificador LDA sin PCA
Fuente: Propia

	Precision	Recall	F1-score
Resultados	0.8189	0.8083	0.8112

En la figura 4.12 se muestra el resumen de los datos al aplicar el método PCA para 2 componentes principales. Se puede observar que el gesto 2 (púrpura), al igual que en el análisis del modelo anterior, pertenece a alguna región de cualquier otro gesto, por lo que puede que sea difícil clasificarlo.

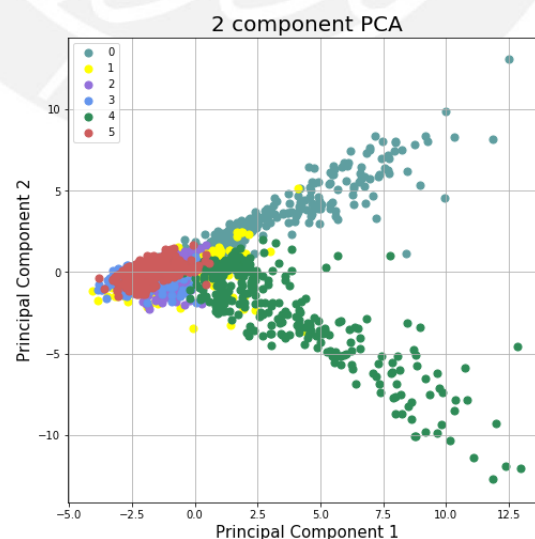


Figura 4.12: Análisis de 2 componentes principales – Usuario específico
Fuente: Propia

Debido a los resultados de las pruebas anteriores, las segundas pruebas se realizaron tomando 1 sola muestra de tamaño 2500, es decir que el tamaño de la matriz de características fue de 1800 x 20. Se siguió el mismo procedimiento con respecto a la separación de datos de entrenamiento (80%) y validación (20%), y la posterior estandarización de los mismos. Se analizó los casos cuando para la selección de características se usó PCA o se usó todas las características y para la clasificación de modelos se usó el clasificador SVM con función de base radial o el clasificador *Random Forest*. En los casos donde se utilizó PCA, se conservó el 99% de la varianza de las características y el procedimiento para transformar ambos sets de datos tomó aproximadamente un total de 4793 ms. A continuación, se muestran los resultados obtenidos para cada caso usando los datos de validación.

Para el clasificador SVM con función de base radial cuando no se utiliza PCA se escoge $C=3$ y $\gamma=0.3$ para obtener los resultados más óptimos. En la figura 4.13 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 2741 ms.

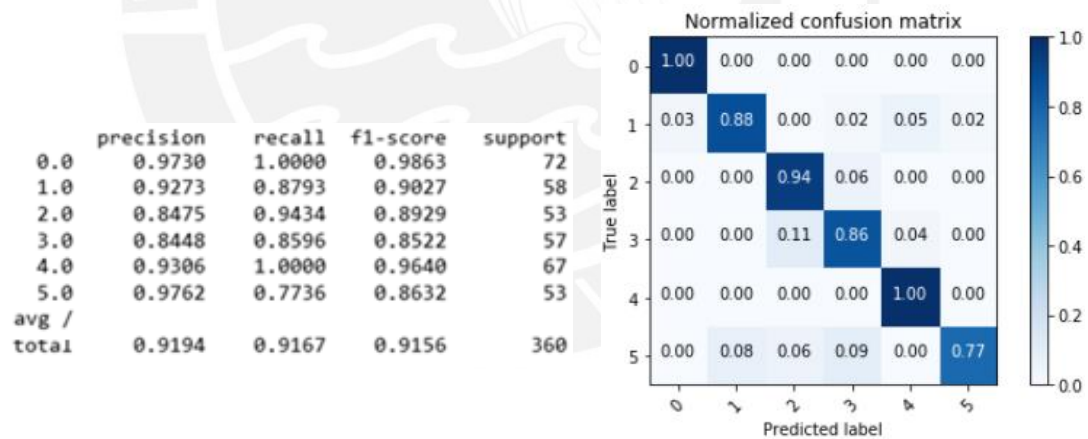


Figura 4.13: Reporte y matriz de confusión normalizada - SVM - Usuario específico
Fuente: Propia

Para el clasificador SVM con función de base radial cuando se utiliza PCA se escoge $C=2.0$ y $\gamma=0.5$ para obtener los resultados más óptimos. En la figura 4.14 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 1714 ms, sin incluir el tiempo que tomó aplicar PCA.

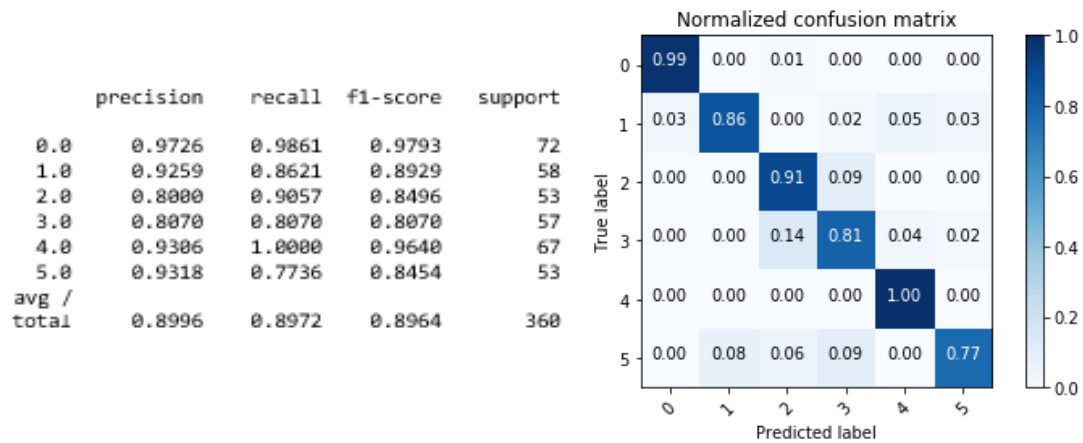


Figura 4.14: Reporte y matriz de confusión normalizada - SVM + PCA – Usuario específico
Fuente: Propia

Para el clasificador *Random Forest* cuando no se utiliza PCA se escoge `n_estimators=200, criterion='entropy'`. En la figura 4.15 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 2182 ms.

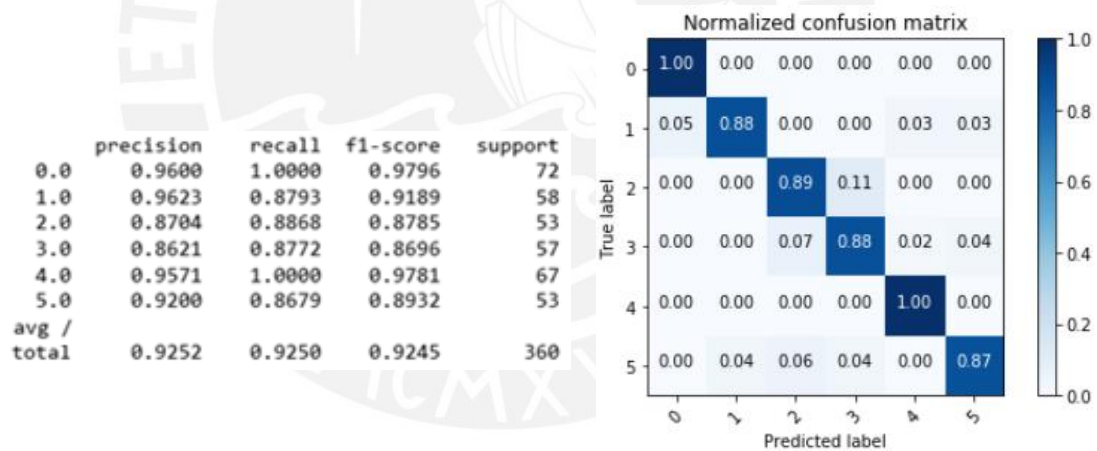


Figura 4.15: Reporte y matriz de confusión normalizada – Random Forest - Usuario específico
Fuente: Propia

Para el clasificador *Random Forest* cuando se utiliza PCA se escoge `n_estimators=300, criterion='entropy'`. En la figura 4.16 se muestran el reporte obtenido y la matriz de confusión normalizada, respectivamente. El procedimiento tomó aproximadamente 3166 ms, sin incluir el tiempo que tomó aplicar PCA.

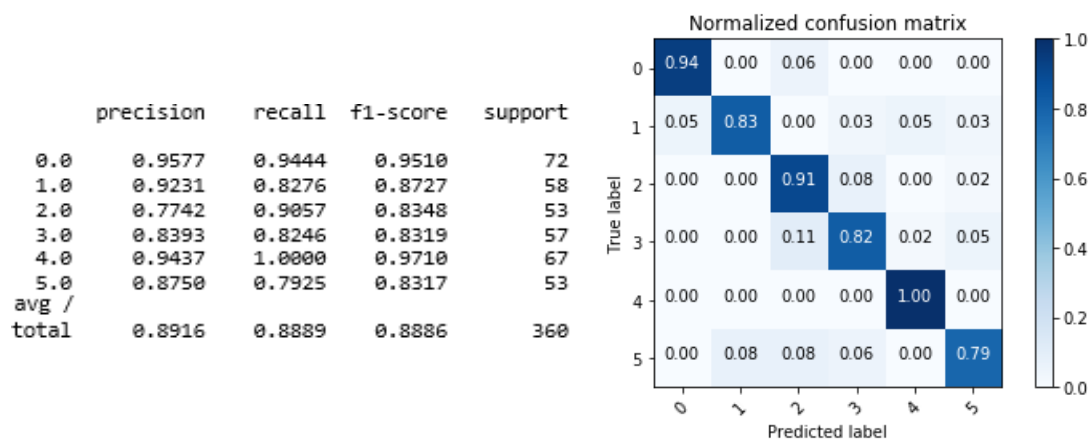


Figura 4.16: Reporte y matriz de confusión normalizada – Random Forest + PCA - Usuario específico
Fuente: Propia

En la tabla 4.6 se muestra la comparación de los resultados de los 4 casos, se resalta los resultados más óptimos para cada resultado.

Tabla 4.6: Comparación de resultados – Usuario específico
Fuente: Propia

Caso	Precision	Recall	F1-score	Tiempo (ms)
SVM	0.9194	0.9167	0.9156	2741
SVM + PCA	0.8996	0.8972	0.8964	6507
RF	0.9252	0.9250	0.9245	2182
RF + PCA	0.8916	0.8889	0.8886	7959

Se puede concluir que al igual que en las pruebas para usuarios múltiples, cuando se usa PCA el procedimiento toma más tiempo y en este caso no se obtiene mejores resultados, ya que la de características no es de una dimensión grande. Además, nuevamente como se previó en el análisis de PCA para 2 componentes principales, el porcentaje de precisión del gesto 2 ese bajo con respecto a los demás gestos en todos los casos, por lo que se recomendaría no usar este gesto al mismo tiempo que los otros 5 gestos, ya que es similar a algunos de los otros gestos de esta base de datos y existe dificultad para clasificarlo. Con respecto a los clasificadores, en este caso, el clasificador *Random Forest* se ejecutó más rápido que el clasificador SVM, debido a que se decidió disminuir la variable `n_estimators` (número de árboles en los cuales se divide). Por lo descrito, se escoge el modelo construido con el clasificador *Random Forest*.

Los resultados de precisión para el modelo de usuarios múltiples y el modelo de usuario específico fueron de 92.80% y 92.52%, respectivamente. Se esperaría que el resultado para el modelo de usuarios múltiples sea menor, sin embargo, es ligeramente mayor; esto se debe a que los datos de validación con los que se compara para obtener el resultado en ambos modelos pertenece a la misma base de datos y en el caso de la base de datos 1, la cual se usó para el modelo de usuarios múltiples, la toma de datos se llevó a cabo en el mismo día y, por el contrario, aunque la toma de datos para la base de datos 2 pertenezca a un solo sujeto, esta se llevó a cabo en 3 días y los artefactos electrocardiográficos pueden haber variado y afectado la toma de datos, lo cual en realidad da un resultado más preciso, ya que se acerca más al uso del dispositivo en la vida real.



CAPÍTULO 5

COSTOS

En este capítulo se presentan los costos totales del sistema. Los costos se distribuyen en costo del diseño electrónico, en el cual se incluye el costo de los componentes electrónicos, costo del diseño mecánico, en el cual se incluye el costo de fabricación de los dos *enclosures* y del guante. Para los componentes electrónicos que sean importados se considerara como costo de envío el 40 % del subtotal, esto también engloba los gastos en aduanas y otros como los costos de resistencias, cables y conectores. Las cotizaciones se pueden consultar en el Anexo H.

5.1. Costos de componentes para diseño electrónico

En la tabla 5.1, se muestran los costos de los componentes electrónicos que se usaran en el diseño electrónico del dispositivo. El costo de los componentes se presenta en dólares considerando el valor del dólar como S/. 3.40.

Tabla 5.1: Costo de diseño eléctrico
Fuente: Propia

Componente	Proveedor	Precio unitario (\$)	Cantidad	Precio total (\$)
Sensor IMU – MPU 9250	GearBest	7.34	1	7.34
Sensor de flexión SEN-08606	SparkFun	7.95	5	39.75
Módulo Wi-Fi ESP-01	SparkFun	6.95	1	6.95
Arduino Nano	Arduino Store	22	1	22
Batería LiPo 1S	SparkFun	9.95	1	9.95
Módulo <i>step-up</i> U3V12F5	Pololu	3.95	1	3.95
Módulo <i>step-down</i> D24V5F3	Pololu	4.95	1	4.95
Impresión circuito PCB	ADE Services	7.35	1	7.35
Subtotal				102.24
Envío + otros				40.90
Total (\$)				143.14
Total (S/.)				486.68

El costo de integración del diseño electrónico se dará en un tiempo de 5 horas, siendo el precio de trabajo por hora S/.40, el costo de integración será de S/. 200. Por lo tanto, el costo total para el diseño electrónico es de S/. 686.68.

5.2. Costos componentes y materiales para diseño mecánico

En esta sección se detallan los costos de fabricación de los *enclosures* y los materiales para el diseño del guante. En la tabla 5.2 se muestran los costos de fabricación de los dos *enclosures* diseñados. La cotización pertenece a la Sala VEO PUCP.

Tabla 5.2: Costo de fabricación *enclosures*
Fuente: Propia

Componente	Cantidad	Precio total (S/.)
<i>Enclosure</i> – Batería - ABS	1	44
<i>Enclosure</i> – Electrónica - ABS	1	44
Total		88

En la tabla 5.3 y 5.4 se detallan los costos de los materiales para el diseño del guante, se considera el valor del dólar como S/. 3.40.

Tabla 5.3: Costo de materiales para el diseño del guante – tela algodón
Fuente: Propia

Materiales	Proveedor	Cantidad (kg)	Precio / kg (S/.)	Precio total (S/.)
Tela 100% algodón Pima corte 1kg – 3 m	BERGMAN RIVERA SAC	1	40	40

Tabla 5.4: Costo de materiales para el diseño del guante – tela Scuba
Fuente: Propia

Materiales	Proveedor	Cantidad (yarda)	Precio total (\$)
Tela Scuba Knit Solid Champagne	Fabric.com	1	4.98
		Envío	1.99
		Total (\$)	6.97
		Total (S/.)	23.70

El costo total para la fabricación de *enclosures* y materiales para diseño mecánico es S/. 151.70.

El costo de integración del diseño mecánico, fabricación del guante e incluyendo pruebas de diseño, se dará en un tiempo de 15 horas, siendo el precio de trabajo por hora S/.40, el costo de integración será de S/. 600. Por lo tanto, el costo total para el diseño electrónico es de S/. 751.70.

5.3. Costo del sistema

En esta sección se muestra el costo total del sistema, el cual consiste en la suma de los costos de los componentes y materiales para los sistemas mecánico y electrónico. Además, también se detalla el costo de diseño para el sistema, para el cual se considera un precio de S/. 40 por hora, un tiempo invertido de 15 horas por semana y 15 semanas para la duración del proyecto. De este modo, se obtiene un costo de diseño de S/. 9000, en este costo también está incluido los costos de las pruebas de diseño de modelo de reconocimiento de gestos. En la tabla 5.5 se muestra el costo total del sistema. Se debe resaltar que el costo puede variar ya que la vigencia de las cotizaciones es de 30 días.

Tabla 5.5: Costo total del sistema
Fuente: Propia

Concepto	Costo total (S/.)
Costos sistema electrónico: componentes y fabricación	686.68
Costos sistema mecánico: componentes y fabricación	751.70
Costo sistema electrónico y mecánico	1438.38
Costo de diseño	9000
Costo total (S/.)	10438.38

Se obtiene un costo total de S/. 10, 438.38, equivalente a aproximadamente \$ 3070 si se toma en cuenta S/. 3.40 como tasa de cambio. En la lista de requerimientos se indicó, que se deseaba que el costo sea máximo \$950, sin embargo, este costo hace referencia al costo de un guante cuando este sea fabricado en volúmenes de producción alto, ya que los costos de diseño se repartirían entre cada guante. No obstante, cabe resaltar que el costo para el diseño mecánico y electrónico fue de S/.1438.38, es decir \$423 aproximadamente, y ya que es mucho menor al precio límite requerido, se puede concluir que con suficientes guantes el costo de diseño se podría repartir y obtener un costo menor de \$950 por guante.



CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

En este capítulo se plantean las conclusiones a las que se llegó al finalizar el trabajo de tesis. Finalmente, se presentan las recomendaciones para trabajos futuros. A continuación, se detallan las ideas que concluyen la tesis.

- Se logró diseñar el dispositivo para toma de datos de reconocimiento de gestos de la mano de la forma más económica posible, aplicando conocimientos de electrónica, mecánica, control y salud y cumpliendo con los requerimientos planteados.
- Se logró construir un modelo de reconocimiento de gestos usando el clasificador *Random Forest*, obteniendo un porcentaje de precisión para datos de validación de 92.80% para un modelo de usuarios múltiples y 92,52% para un modelo de un usuario.
- Aunque los componentes electrónicos son del extranjero, muchos de estos también se distribuyen en el país por lo que se cuenta con fácil acceso a repuestos.
- La vinculación de los gestos que el modelo reconoce con archivos de audio y la implementación del prototipo serán trabajados en una investigación posterior a esta tesis, tal como se indicó en los objetivos.
- El costo de la parte física del dispositivo será aproximadamente S/. 1440, por lo que en caso de producir guantes en grandes cantidades el costo de estos cumpliría

con ser menor al costo de los guantes de toma de datos para reconocimiento de gestos ya existentes en el mercado.

Finalmente, se presentan las siguientes recomendaciones:

- Se logró el diseño de un dispositivo relativamente compacto, sin embargo, la parte electrónica podría reducirse mucho más si se diseñara una tarjeta propia con el microcontrolador perteneciente a la placa Arduino Nano, usando solo las partes necesarias pertenecientes a la placa. Cabe resaltar que esto aumentaría los costos de diseño.
- Se recomienda realizar un modelo de reconocimiento de gestos con una nueva base de datos, proveniente de un prototipo del guante diseñado, ya que el realizado en esta tesis trabaja con una base de datos de sensores EMG.



REFERENCIAS

- [1] K. Watson, *IMS Business Report 2017*. Ibiza: *International Music Summit*, 2017, p. 34 [Online]. Available: <https://www.internationalmusicsummit.com/wp-content/uploads/2017/09/IMS-Business-Report-2017-vFinal3.pdf>. [Accessed: 01- Sep- 2018]
- [2] R. Smirke, "IMS Biz Report: Global EDM Market Falls 2 Percent to \$7.3 Billion", *Billboard*, 2018. [Online]. Available: <https://www.billboard.com/articles/business/8457536/ims-biz-report-2018-global-edm-market-electronic-dance>. [Accessed: 11- Sep- 2018]
- [3] "How Gloves and Wearable Tech Could Change Music Performance: In Depth with Imogen Heap and Team - CDM Create Digital Music", *CDM Create Digital Music*, 2018. [Online]. Available: <http://cdm.link/2014/04/making-musical-expression-wearable-depth-chat-imogen-heap-gloves-project/>. [Accessed: 04- Sep- 2018]<http://cdm.link/2014/04/making-musical-expression-wearable-depth-chat-imogen-heap-gloves-project/> [Accessed 4 Sep. 2018].
- [4] T. Ferguson, "Interview with Chagall on Her Music, Mi.mu Gloves and Gestural Performance | Keith McMillen Instruments", *Keith McMillen Instruments*, 2018. [Online]. Available: <https://www.keithmcmillen.com/blog/interview-chagall-music-mi-mu-gloves-gestural-performance/>. [Accessed: 04- Sep- 2018]
- [5] A. Glinsky, *Theremin: Ether Music and Espionage*. Urbana, Ill: Univ. of Illinois Press, 2000, pp. 23-42.
- [6] Topical Press Agency, *Professor Leon Theremin demonstrating his theremin.1927* [Online]. Available: <https://www.gettyimages.com/detail/news-photo/professor-leon-theremin-demonstrating-his-theremin-the-news-photo/3311852#/12th-december-1927->

professor-leon-theremin-demonstrating-his-theremin-picture-id3311852.

[Accessed: 01- Sep- 2018]

- [7] Moog Music Inc, *Moog Etherwave Theremin Plus*. 2009 [Online]. Available: https://www.thomann.de/es/moog_etherwave_theremin_plus.htm. [Accessed: 01- Sep- 2018]
- [8] T. Blaine, "*The Convergence of Alternate Controllers and Musical Interfaces in Interactive Entertainment*", *Hands-on-tech.com*, 2005. [Online]. Available: <http://www.hands-on-tech.com/articles/nime05.pdf>. [Accessed: 01- Sep- 2018]
- [9] D. Sturman and D. Zeltzer, "*A survey of glove-based input - IEEE Journals & Magazine*", *Ieeexplore.ieee.org*, 1994. [Online]. Available: <https://ieeexplore.ieee.org/document/250916/>. [Accessed: 01- Sep- 2018]
- [10] Viper, *Sayre Glove*. 2016 [Online]. Available: <http://www.viper-computing.be/2016/11/16/historique/>. [Accessed: 01- Sep- 2018]
- [11] "*Instruments – Lady's Glove – LAETITIA SONAMI*", *Sonami.net*. [Online]. Available: <http://sonami.net/ladys-glove/>. [Accessed: 01- Sep- 2018]
- [12] "Tech", *mi.mu*, 2014. [Online]. Available: <https://mimugloves.com/tech/>. [Accessed: 01- Sep- 2018]
- [13] "*Data Gloves | 5DT*", *5dt.com*, 2010. [Online]. Available: <http://www.5dt.com/data-gloves/>. [Accessed: 01- Sep- 2018]
- [14] G. Grimes, "*Digital data entry glove interface device*", 1983, US4414537A1981.
- [15] McDowall, M. Bolas, R. Mead, Jr. and C. Greuel, "*Virtual reality glove system with fabric conductors*", 2000, US6128004A2018.
- [16] S. Ganeshji, M. Irshad, H. K.L, T. R and R. S Rao, "*Gesture Controlled Musical Instrument*", Bachelor Thesis, BNM Institute of Technology, 2015.

- [17] S. Sasidharan, "*Smart Glove: An Assistive Device to Enhance Recovery of Hand Function During Motor Rehabilitation*", Master Thesis, ARIZONA STATE UNIVERSITY, 2015.
- [18] J. D. Schaechter, C. Stokes, B. Connell, K. Perdue and G. Bonmassar, "*Finger motion sensors for fMRI motor studies*", Ph.D, NeuroImage, 2006.
- [19] J. McIntosh, C. McNeill, M. Fraser, F. Kerber, M. Lochtefeld and A. Kruger, "*EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing*", Ph.D, University of Bristol & German Research Center for Artificial Intelligence (DFKI) and Saarland University Saarbrücken, 2016.
- [20] A. Samraj, K. Rajendran and R. Palaniappan, "*A Segmented-Mean Feature Extraction Method for Glove-based System to Enhance Physiotherapy for Accurate and Speedy Recuperation of Limbs*", *Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.
- [21] J. Gaveau and C. Papaxanthis, *The Temporal Structure of Vertical Arm Movements*. 2011. [Online]. Available: https://www.researchgate.net/publication/51499121_The_Temporal_Structure_of_Vertical_Arm_Movements. [Accessed: 01- Nov- 2018]
- [22] K. Horie, A. Suemitsu, T. Tanno and M. Morita, "*Direct Estimation of Wrist Joint Angular Velocities from Surface EMGs by Using an SDNN Function Approximator*", University of Tsukuba and Sapporo University of Health Sciences, 2015. [Online]. Available: <https://pdfs.semanticscholar.org/143e/8ed95c69b6110d42a50668ab25d53a952bcc.pdf>. [Accessed: 01- Nov- 2018]
- [23] K. Lee and M. Jung, "*Flexion and Extension Angles of Resting Fingers and Wrist*", Ajou University, 2014. [Online]. Available: <http://archiwum.ciop.pl/64247>. [Accessed: 01- Nov- 2018]
- [24] Texas Instruments, TPS6120xEVM-179 *datasheet*, 2007. [Accessed: 01- Nov- 2018]

- [25] MatWeb, *Overview of materials for Acrylonitrile Butadiene Styrene (ABS)*, Sheet. Available: <http://www.matweb.com/search/DataSheet.aspx?MatGUID=c8bc69525dd04bd9bca54c475f6b38c3>. [Accessed: 20- Nov- 2018]
- [26] P. Bonenberger, *First Snap-Fit Handbook*, 3rd ed. 2016.
- [27] BASF, *Snap-Fit Design Manual*. 2007. [Online]. Available: <http://www8.basf.us//PLASTICSWEB/displayanyfile?id=0901a5e1801499d5>. [Accessed: 01- Nov- 2018]
- [28] K. Suk Lee and S. Hwan Cha, “*Acceptable force and maximum force in inserting small objects using finger gripping*”, Hongik University, 2007. [Online]. Available: https://www.jstage.jst.go.jp/article/jje/49/Supplement/49_S492/_pdf/-char/en. [Accessed: 20- Nov- 2018]
- [29] C. Sapsanis, “*UCI Machine Learning Repository: sEMG for Basic Hand movements Data Set*”, 2013. Available at: <https://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+movements> [Accessed 28-Nov-2018].
- [30] C. Sapsanis, G. Georgoulas and A. Tzes. “*EMG based classification of basic hand movements based on time-frequency features*” University of Patras, Greece, 2013. Available: https://www.researchgate.net/profile/Christos_Sapsanis/publication/261313999_EMG_based_classification_of_basic_hand_movements_based_on_time-frequency_features/links/56dfb78c08ae9b93f79a9e88/EMG-based-classification-of-basic-hand-movements-based-on-time-frequency-features.pdf. [Accessed: 25-Nov-2018]
- [31] C. Sapsanis. “*Recognition of basic hand movements using electromyography*”, 2013. Available: <https://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+movements>. [Accessed: 25-Nov-2018]
- [32] S. Madgwick. “*Open source IMU and AHRS algorithms*”, 2012. Available: <http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>. [Accessed: 28-Nov-2018]