

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

**FACULTAD DE INGENIERÍA Y GESTIÓN
ESCUELA PROFESIONAL DE
INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**



**“PROPUESTA DE DISEÑO DE UN DISPOSITIVO DEDIL INTELIGENTE,
PARA MEJORAR LA COMUNICACION EN
PACIENTES TRAQUEOSTOMIZADOS POST-UCI EN LA IPRESS
HOSPITAL MARIA AUXILIADORA EN EL DISTRITO DE SAN JUAN DE
MIRAFLORES”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

PRESENTADO POR EL BACHILLER

QUITO HUAYCAÑE, VICTOR ALEXANDER

Villa El Salvador

2020

DEDICATORIA

El presente proyecto lo dedico a mi padre Julio Victor Quito Calvo, quien fue la inspiración y el motivo principal por el que se desarrolló este proyecto. Que Dios lo tenga siempre en su gloria.

AGRADECIMIENTO

Agradezco principalmente a mis Padres, Hermanos y Profesores quienes fueron los pilares de mi formación académica y sembraron en mis los valores que todo profesional necesita para ser un hombre exitoso.

INDICE

DEDICATORIA.....	ii
AGRADECIMIENTO	iii
LISTADO DE FIGURAS.....	vii
LISTADO DE TABLAS	ix
RESUMEN.....	x
INTRODUCCIÓN	1
OBJETIVOS.....	2
a. Objetivo General.....	2
b. Objetivos Específicos	2
CAPÍTULO I: MARCO TEÓRICO.....	3
1.1 Antecedentes.....	3
1.1.1 Antecedentes Internacionales	3
1.1.2 Antecedentes Nacionales	4
1.2 Bases Teóricas	6
1.2.1 Hardware de referencia.....	6
1.2.1.1 Modulo ESP-WROOM-32	6
1.2.1.2 Sensor MPU6050 Acelerómetro y Giroscopio	8
1.2.1.3 Tecnología Smart Bluetooth Low Energy	10
1.2.2 Software de Referencia para Machine Learning	14
1.2.2.1 Lenguaje de programación Python.....	14
1.2.2.2 Herramienta de Data Science Scikit-Learn	17
1.2.2.3 Librerías de Machine Learning	18
1.2.2.4 Aprendizaje en Machine Learning	20
1.2.3 Software de Diseño Electrónico	28
1.2.3.1 Diseño y Esquemas Electrónicos - EAGLE	28
1.2.3.2 Diseño y Simulación de Circuitos - PROTEUS.....	29
1.3 Definición de Términos Básicos	30

CAPÍTULO II: METODOLOGIA DE DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL.....	32
2.1 Delimitación Temporal y Espacial del Proyecto	32
2.1.1 Teórica	32
2.1.2 Temporal.....	32
2.1.3 Espacial.....	32
2.2 Determinación y Análisis del problema	33
2.2.1 Descripción de la Realidad Problemática	33
2.2.2 Justificación del Problema	34
2.2.3 Formulación del Problema	35
a. Problema General.....	35
b. Problemas específicos	35
2.3 Modelo de Solución Propuesto.....	36
2.3.1 Diseño del Sistema	36
2.3.1.2 Requerimiento de Diseño.....	36
2.3.1.2 Requerimiento de Diseño de Sistema Dedil Inteligente Portátil de Uso Medico.....	37
2.3.1.3 Diagrama de Bloques del Sistema del Dispositivo Propuesto	43
2.3.2 Desarrollo del Prototipo del Hardware	43
2.3.2.1 Sistema de Adquisición de Datos	43
2.3.2.2 Diseño del PCB del Dispositivo Dedil.....	50
2.3.2.3 Estructura de Soporte del Dispositivo Dedil.....	51
2.3.3 Desarrollo del Software del Prototipo.....	52
2.3.4 Entrenamiento del Clasificador y Análisis de la Matriz de Confusión	57
2.3.5 Implementación en el Hospital María Auxiliadora	70
2.4 Resultados.....	71
2.4.1 Resultado 1.....	71
2.4.2 Resultado 2.....	73
2.4.3 Resultado 3.....	74
2.4.4 Resultado 4.....	76
CONCLUSIONES.....	78
RECOMENDACIONES.....	79

BIBLIOGRAFÍA.....	80
ANEXOS.....	83

LISTADO DE FIGURAS

Figura 1. Distribución de los pines del ESP32.....	7
Figura 2. Imagen real Módulo ESP-WROOM 32.....	8
Figura 3. Proceso de funcionamiento del MPU 6050.....	9
Figura 4. Modulo MPU 6050.....	10
Figura 5. Protocolo de capas utilizados en el BLE.....	12
Figura 6. Capa Física del BLE.....	13
Figura 7. Evolución del Lenguaje de Programación Python en el tiempo.....	15
Figura 8. Principales Librerías Usadas en Scikit-Learn.....	18
Figura 9. Hiperplano de la ecuación $1+2x_1 +3x_2 =0$	22
Figura 10. Visualización de un Máximo Margen de Hiperplano Teórico.....	23
Figura 11. Imagen del clasificador de un SVM.....	24
Figura 12. Muestra Grafica de los distintos tipos de Kernel en SVM.....	27
Figura 13. Diagrama Esquemático de Microcontrolador y SW en EAGLE.....	28
Figura 14. Proteus trabajando en PIC y periféricos.....	29
Figura 15. Habitación Médica Del Hospital María Auxiliadora.....	36
Figura 16. Diagrama de Bloques de un Dispositivo Medico Portátil.....	38
Figura 17. Peso de Bateria 18650.....	39
Figura 18. Pruebas de medida aproximada del Dispositivo Dedil.....	40
Figura 19. Circuito del Hardware de Adquisición de Datos.....	48
Figura 20. Diagrama de Flujo del Software de Adquisición de Datos.....	49
Figura 21. Diseño del PBC con medidas finales.....	50
Figura 22. Diseño del Case en 3D en software Ultimaker.....	51
Figura 23. Estructura de Software.....	52
Figura 24. Forma de la trama de datos para el entrenamiento del "1".....	54
Figura 25. Proceso básico de predicción de caracteres.....	57
Figura 26. Valores posibles de los parámetros del Clasificador SVM.....	58
Figura 27. Valores de entrenamiento con 240 muestras.....	60
Figura 28. Matriz de confusión con 240 muestras.....	61
Figura 29. Valores de entrenamiento con 480 muestras.....	62
Figura 30. Matriz de confusión con 480 muestras.....	63
Figura 31. Valores de entrenamiento con 960 muestras.....	64

Figura 32. Matriz de confusión para 960 muestras	65
Figura 33. Predicción con una tasa del 96%.....	66
Figura 34. Fórmulas para el análisis de la Matriz de Confusión	67
Figura 35. Dispositivo que se adapta en el dedo del paciente	72
Figura 36. Prueba de entrenamiento con Dispositivo Dedil	73
Figura 37. Prueba de entrenamiento del Dispositivo.....	76

LISTADO DE TABLAS

Tabla 1. Comparativa entre el Bluetooth Clásico y el BLE	11
Tabla 2. Comparativa entre Python 2.7 y Python 3.6	17
Tabla 3. Resultados del Entrenamiento del SVM en base a los parámetros del Kernel C.....	58
Tabla 4. Resultados de valor de Precisión.....	68
Tabla 5. Resultados del cálculo del Recall.....	68
Tabla 6. Resultados del cálculo del F1-Score.....	69
Tabla 7. Resultados del cálculo del Accuracy.....	69
Tabla 8. Resultados de los entrenamientos de voluntarios	70
Tabla 9. Costos del Proyecto.	71
Tabla 10. Consumo Energético de los componentes	74
Tabla 11. Rango de Voltajes de Operación	75

RESUMEN

La Medicina junto a otras ciencias trabajan y convergen en distintas investigaciones para poder dar tratamiento a diversas enfermedades. Por ejemplo, en la actualidad el Virus SARS-CoV-2, responsable de la pandemia en la que vivimos, ataca principalmente al sistema respiratorio, cuando esto sucede el virus se traslada hacia los pulmones, provocando una tos seca e inflamación de los alvéolos, por consiguiente, dificultad para respirar. La traqueostomía en pacientes con Síndrome de Distrés respiratorio agudo (SDRA), es considerado como alternativa de tratamiento cuando se estima que el paciente estará en un largo periodo de recuperación o la desentubación (destete) puede contraer muchos riesgos. Según las recomendaciones médicas, el proceso de traqueostomía se considera necesario cuando un paciente lleva en entubación entre 9 a 10 días, especialmente en pacientes con debilidad muscular, reserva respiratoria limitada o con abundantes secreciones espesas. (Martínez-Téllez et al., 2020, pp. 1–4)

Actualmente, para mejorar la comunicación en pacientes traqueostomizados existen pocas técnicas y dispositivos, entre ellas se clasifican como *Low-Tech* y *High-Tech*, de las primeras tenemos la escritura a papel o pizarra, vocalización silenciosa, deletreo con tableros de alfabeto, escaneos o barrido asistido visual o pictogramas, mientras que en las High-Tech tenemos dispositivos generadores de voz simples y complejos, paneles touch entre otros. Lo ideal sería tener una combinación de ambas. (Perelló-Campaner, 2015)

Por tal motivo en el presente trabajo de suficiencia propone el diseño de un dispositivo dedil de bajo costo, como medio de comunicación paciente-enfermera-familiares, teniendo en cuenta el limitado movimiento de los músculos que se tiene en la mayoría de pacientes Post-UCI, el dispositivo está dirigido a su uso en algún dedo de la mano, previamente el paciente deberá entender la matriz que se desarrolló en base al diagrama Pictorrino, el cual tuvo un previo estudio para desarrollarse en base a las

necesidades de pacientes laringectomizados, el mismo que fue aceptado para su uso masivo en hospitales de España. (Orduña Beuzón, Vicente Bardón, Martínez Gago, & Plaza Mayor, 2009)

El dispositivo será colocado en algún dedo del paciente para obtener los datasets, se realizó un experimento y se determinó que por cada carácter de la matriz se debe tener como mínimo 80 muestras, estas muestras son obtenidas por el procesamiento de los datos que se hace del MPU6050 mediante la placa de desarrollo ESP32 usando el *Bluetooth Low Energy* (BLE), posteriormente se realizará el entrenamiento supervisado de Machine Learning usando Python con la técnica de Soporte de Máquina Vectorial (SVM), se hizo uso de la librería *Pygarl* basado en *Scikit-Learn*, el cual es recomendado para las actividades señaladas, como lo demuestran otros artículos donde se ha empleado dicha herramienta, en donde además se concluye la importancia de esta para abordar problemas neurocientíficos y de otras categorías (Abraham et al., 2014, pp. 92–30). En los resultados obtenidos, se hizo un estudio de los 3 voluntarios y de los análisis de la matriz de confusión de cada uno de ellos, teniéndose valores aceptables que rondan entre el 75 % al 99% según se pueden apreciar en las tablas 4,5,6 y 7 respectivamente. Sobre el modelo se tiene una tasa de predicción del 96% (Figura 31), porcentaje aceptable para cumplir con los objetivos propuestos. Se determino igualmente que el consumo del dispositivo es en total de 42.7 mA (Tabla 9), y que con unas baterías de 10000mAh, el dispositivo podría funcionar continuamente durante 10 días aproximadamente.

En tal sentido, por lo mencionado anteriormente, se concluye que la propuesta del dispositivo dedil inteligente, ayuda significativamente a mejorar la calidad de vida y comunicación en pacientes traqueostomizados, hasta que estos recuperen su fonación, y puedan comunicarse mediante el habla o algún otro método que involucre el uso de sus propias capacidades.

INTRODUCCIÓN

Actualmente en el Perú, se cuenta con más de 800 mil casos Covid-19 confirmados, de los cuales hay más de 30 mil fallecidos y se encuentran en el área de cuidados intensivos 983 personas y se tiene una disponibilidad de 480 camas con ventiladores mecánicos (“Covid 19 En El Perú - Ministerio Del Salud,” n.d.). En la mayoría de pacientes UCI por Covid 19, se utiliza la traqueostomización para facilitar la ventilación mecánica invasiva a largo plazo y el destete con menor riesgo. (Rappoport W et al., 2020).

Estos pacientes en su proceso de recuperación, presentan diferentes tipos de complicaciones postoperatorias, como neumotórax, neumomediastino, infecciones entre otras. Viendo de esta forma afectada su calidad de vida, en cuanto a sus limitantes físicas, psicológicos y de dimensión social. Muchas de ellas asociadas al hecho de no poder comunicarse. (Pinazo Mendoza & Rodriguez Yauri, 2012).

Por otro lado, la tecnología también ha brindado mejoras en cuanto a herramientas para el tratamiento de diversos tipos de enfermedades, estas pueden mejorar la calidad de vida de los afectados. Actualmente se está utilizando el *Machine Learning* para el diagnóstico, tratamiento y recuperación de pacientes con enfermedades de todo tipo. En el proyecto propuesto, se utilizaron las técnicas de Machine Learning de Soporte Vectorial, para resolver las necesidades de comunicación de los pacientes traqueostomizados, se propuso el desarrollo de un dispositivo dedil que pueda interpretar los movimientos del dedo, para poder generar alertas y estas sean atendidas en el menor tiempo posible para salvaguardas y proteger las necesidades de los pacientes con habla afectada.

OBJETIVOS

a. Objetivo General

- Diseñar un dispositivo dedil inteligente que mejore la comunicación en pacientes traqueostomizados Post-Uci en la IPRESS Hospital María Auxiliadora.

b. Objetivos Específicos

- Diseñar el dispositivo dedil inteligente, para que pueda adaptarse fácilmente y obtener datos correctamente en los pacientes traqueostomizados Post-Uci.
- Realizar una propuesta de implementación y capacitación sobre el uso del dispositivo dedil inteligente tanto para médicos como para los pacientes.
- Diseñar un sistema de emergencia en el dispositivo dedil inteligente en caso esté presente fallas.
- Calcular el consumo energético del dispositivo dedil inteligente, de tal modo que se asegure la continuidad del funcionamiento del mismo.

CAPÍTULO I: MARCO TEÓRICO

1.1 Antecedentes

1.1.1 Antecedentes Internacionales

(Saravanan, Govindrajan, Bansal, & Singh ,2018) En su Paper “*Character Recognition through Gestures Using SVM Algorithm*” presentado en la India, nos muestra el trabajo de reconocimiento de caracteres a través de gestos mediante el algoritmo de Support Vector Machine. El Paper se basa en desarrollar un sistema de Machine Learning que será entrenado por diversos gestos, y estos al ser reconocidos puedan interpretar diferentes funciones. Se basa en un principio en la recolección de datos que son ingresados en la lectura del Acelerómetro, utilizando un algoritmo de Python que posteriormente es convertido en un script. Posteriormente la data recolectada y guardada es interpretada, estos *dataset* tienen tablas de 40 datos. Posteriormente los mismos datos son analizados e interpretados por el algoritmo de SVM, optimizado por una búsqueda de cuadrícula, utilizando la función *GridSearchCV*. Finalmente nos muestra como el dispositivo entrenado predice el gesto, utilizando el *script* clasificador creado en Python.

(Alfaro, 2010) En su tesis “*Maquinas de soporte vectorial con algoritmos basados en poblaciones para el pronóstico del precio de acciones Lan Chile*”, nos muestra en su estudio un aporte al estado del arte de los algoritmos de máquinas de soporte vectorial aplicadas tanto para regresión como para clasificación, además de mostrarnos los procesos de implementación de análisis de máquinas de soporte vectorial con la técnicas de mínimos cuadrados (LV-SVM), Algoritmos genéticos (AG) y optimización por enjambre de partículas (PSO). Posteriormente nos muestra el proceso de entrenamiento de los algoritmos, y el testing realizado con los mismos. Finalmente, en las conclusiones, destaca la importancia de las pruebas resultantes en los distintos modelos

propuestos, además de la importancia del uso de las SVM para la predicción del precio de las acciones en Lan Chile con una confiabilidad del 94% para el modelo LS-SVM PSO.

(Sideridis, Zacharakis, Tzagkarakis, & Papadopouli, 2019) En su Paper titulado *“GestureKeeper: Gesture Recognition for Controlling Devices in IoT Environments”*, nos muestra un sistema de reconocimiento de gestos de manos basados en una unidad de medida inercial portátil (IMU), este será el encargado de captar los gestos, una vez captado el gesto en una ventana de tiempo predeterminada, se emplea una máquina de vectores de apoyo para reconocer el gesto específico. El método propuesto puede identificar correctamente el inicio de un gesto con 87% de probabilidad, y clasificar correctamente el gesto realizado por la mano con una media superior al 96%, destacando que el proyecto está basando únicamente en las señales recibidas por parte de un acelerómetro. También se aprecia la comparativa que se realiza entre los métodos de análisis de clasificadores, como son el SVM y el de bosques aleatorios, para poder llegar a la sensibilidad deseada. El trabajo concluye que el método escogido, está dentro del rango de aceptabilidad, además de que el proyecto es modular, pudiendo ampliarse los 12 gestos entrenados en un principio.

1.1.2 Antecedentes Nacionales

(Diaz, 2014) En la tesis de Maestría *“Diseño de un sistema inalámbrico de detección de caídas aplicado a personas de la tercera edad basado en acelerómetro y teléfono móvil”*, el autor nos define los problemas que tienen las personas mayores de edad y los riesgos a los que están propensos, según cuadros estadísticos del INEI. Posteriormente al desarrollar su propuesta, nos muestra su dispositivo de control de posturas que evalúa diferentes variables del entorno como medición de fuerzas, rangos de movimientos, alineación biomecánica entre otros. En el mismo también se hace referencia a los requerimientos del diseño a nivel de portabilidad y de fácil uso médico, así como un análisis de la

adquisición de datos de un sensor acelerómetro ADXL345, la calibración del mismo y la adaptación al entorno del microcontrolador para que sea fácilmente manejable por un software. Se detalla también el diseño electrónico y los parámetros del mismo para ajustar la tasa de error a interferencias externas. Concluyendo que el diseño propuesto, que contiene las etapas de adquisición de datos y detección de caídas, tiene una precisión del 100% para detectar caídas y posteriormente enviar la alerta utilizando mensajes de texto mediante la red celular, por tal motivo el diseño propuesto tiene un alto grado de confiabilidad, sensibilidad y especificidad.

(Tincopa Flores, 2019) *“Diseño y evaluación de un sistema de ayuda al diagnóstico para neuropatía diabética basado en la lectura de puntos de presión plantar y machine learning”*. El trabajo mencionado, realiza un diseño de un sistema basado en la recolección de datos de sensores de presión plantar, como manipulan estos datos en grupos de datasets, que son clasificados de acuerdo a un algoritmo de *Machine Learning* para un mejor diagnóstico de neuropatía diabética. Por otro lado, nos muestra cómo realizar un uso adecuado de la cantidad de muestras, y como concluye que lo ideal, para su proyecto, es utilizar 65% de los datos para el entrenamiento y 25% para la evaluación. Adicionalmente, nos detalla cómo trabaja el algoritmo para la predicción de casos en cuadros de positivos y negativos, utilizando arboles de decisiones y las Maquinas de Soporte Vectorial para la clasificación y predicción. Concluyendo, luego de un método de validación cruzada, que el algoritmo de árboles de decisiones tiene un mejor desempeño. En el desarrollo de su tesis, también nos muestra un exhaustivo análisis de los datos a nivel de modelo predictivo, a nivel de Interfaz de Usuario y usabilidad.

1.2 Bases Teóricas

1.2.1 Hardware de referencia

1.2.1.1 Modulo ESP-WROOM-32

El ESP-WROOM-32 es un módulo de desarrollo que integra tecnología Wifi y Bluetooth, de amplio uso en aplicaciones del Internet de las Cosas. Con las tecnologías que tiene integrada, por ejemplo, con el módulo WIFI nos permite conectarnos a cualquier tipo de red, y poder transmitir y recibir señales desde cualquier parte del mundo, por otro lado con el Bluetooth Low Energy, nos permite poder conectarnos a otros dispositivos o receptores con bajo consumo de recursos y corriente.

El ESP32 fue elegido para este proyecto debido a que su chip central tiene un consumo en reposo inferior a 5 uA, motivo por el cual es adecuado para poder usarlo en dispositivos portátiles. Otra característica importante, es que el ESP32 trabaja con dos núcleos, pudiéndose configurar para uso general el principal y el secundario para monitorizar distintos tipos de periféricos. El ESP32 es adaptable y escalable, al igual que funciona correctamente con distintos tipos de sensores y periféricos que existen en el mercado.

Para el presente trabajo el procesamiento de datos será realizado en este módulo ESP 32, utilizando las librerías de Arduino, ya que este módulo es programable mediante diferentes lenguajes de programación, entre ellas Python. El fabricante también proporciona su propio entorno de desarrollo, que aún está en desarrollo, Las especificaciones técnicas las podemos ver en el Anexo B.

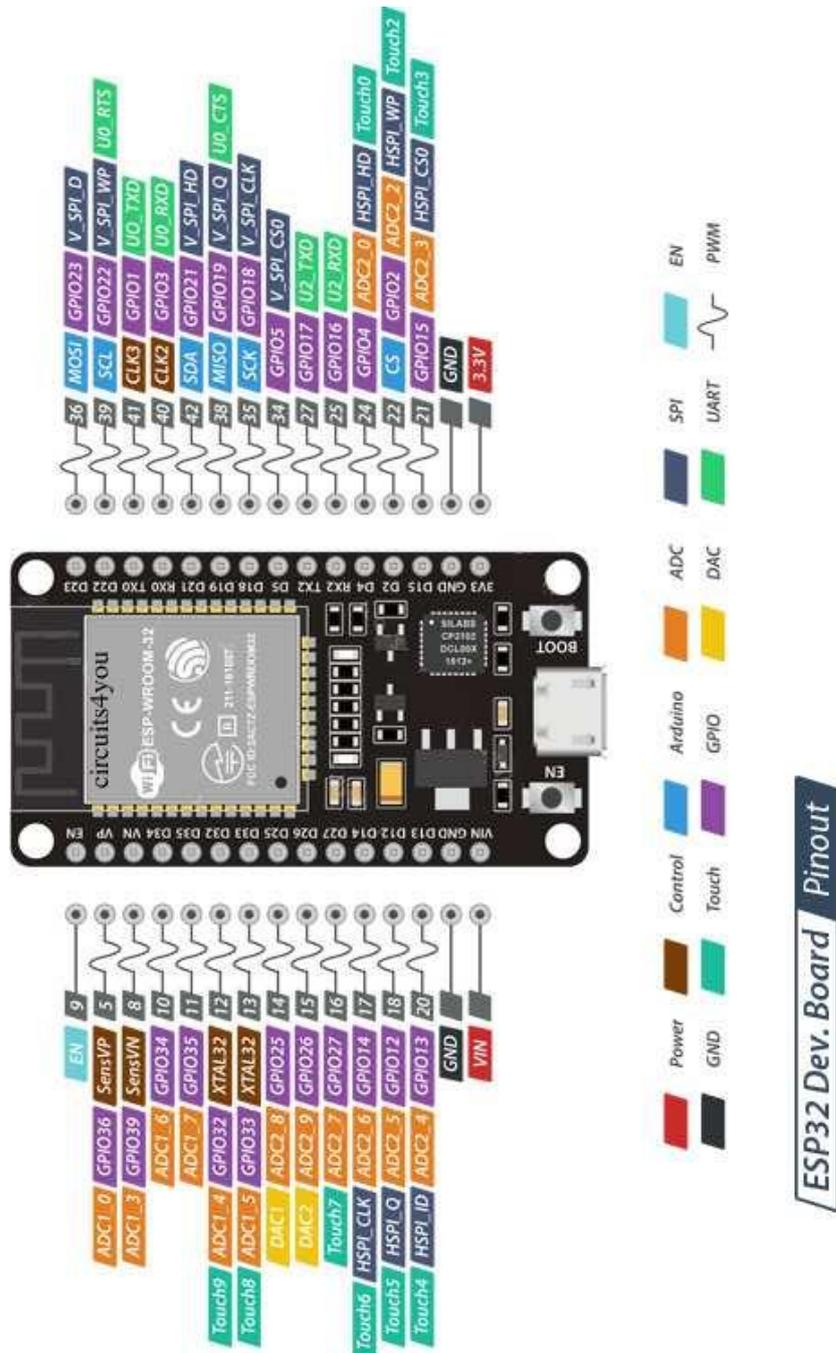


Figura 1. Distribución de los pines del ESP32.

Fuente: Bruno, 2019



Figura 2. Imagen real Módulo ESP-WROOM 32

Fuente: Elaboración Propia

1.2.1.2 Sensor MPU6050 Acelerómetro y Giroscopio

Es un dispositivo que cuenta con un chip integrado que es reconocido a nivel mundial para diversas aplicaciones que requieren precisión y cálculos en diferentes ejes. El que usaremos es fabricado por la empresa INVENSENSE, este cuenta en su interior con un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Se utiliza y configura utilizando el protocolo I2C, por eso la importancia de contar con los pines adecuados en el microcontrolador que se usa. Entre sus principales características,

tenemos, que cuenta con un muestreo de 8Khz, y los 6 grados de libertad cuentan con una resolución de 16 bits. (Bernal Iñiguez, 2014, p. 23).

Al comenzar a utilizar el sensor se debe inicializarse y configurarse para asegurar su correcto funcionamiento. Durante este proceso se debe indicar al acelerómetro el valor de la escala a utilizar en el trabajo, para lo mismo se dispone de las siguientes instrucciones y secuencia, que podemos apreciar en el Anexo C. De igual manera el PINOUT se puede apreciar en la Figura 3.

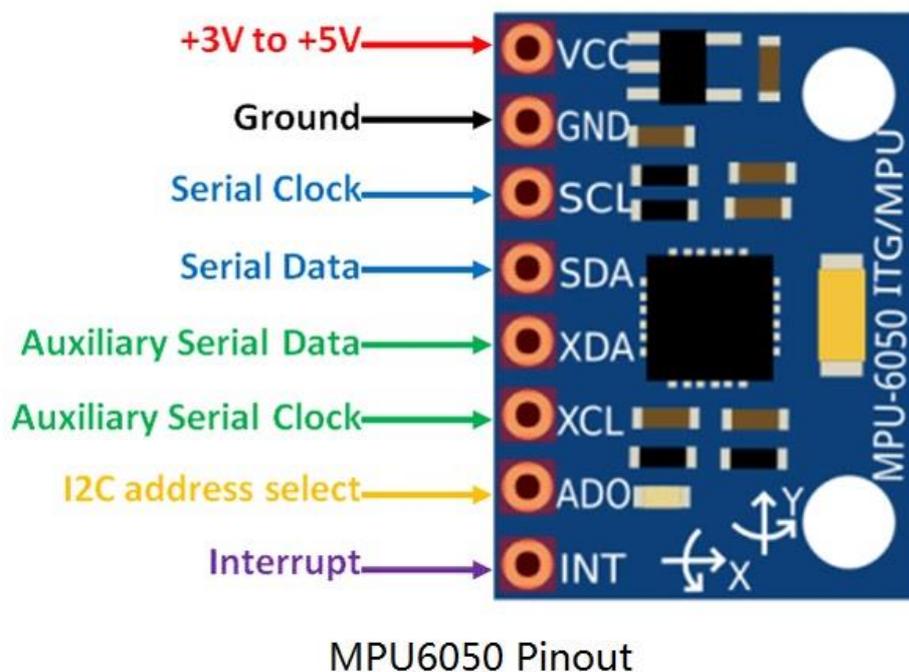


Figura 3. Proceso de funcionamiento del MPU 6050

Fuente: Comisión Nacional de Actividades Espaciales Argentina

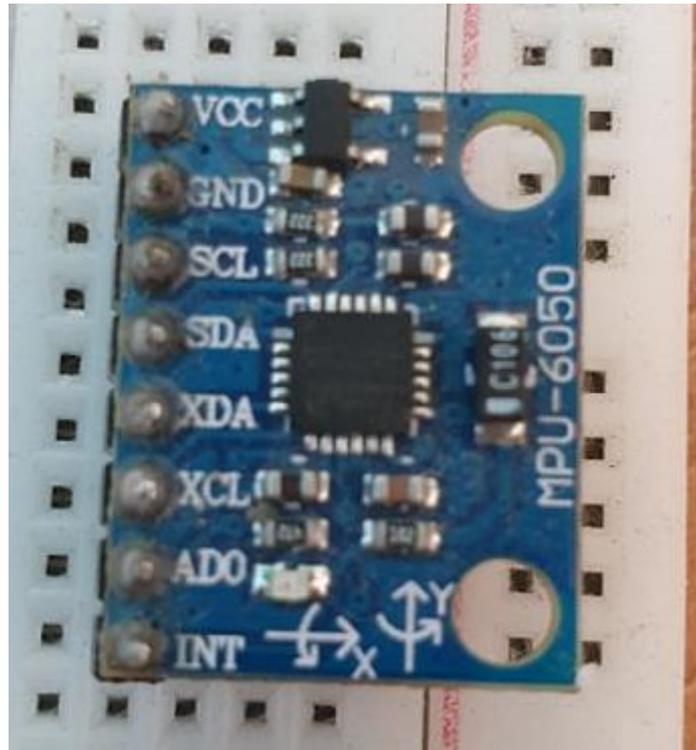


Figura 4. Modulo MPU 6050

Fuente: Elaboración Propia

1.2.1.3 Tecnología Smart Bluetooth Low Energy

Bluetooth Low Energy, BLE para abreviar, es una variante de Bluetooth que ahorra energía. La aplicación principal de BLE es la transmisión a corta distancia de pequeñas cantidades de datos (ancho de banda bajo). A diferencia de Bluetooth que está siempre encendido, BLE permanece en modo de suspensión constantemente, excepto cuando se inicia una conexión.

Esto hace que consuma muy poca energía. BLE consume aproximadamente 100 veces menos energía que Bluetooth (según el caso de uso). A continuación, se presenta una tabla de comparativa entre el BLE y el Bluetooth Clásico.

	BLUETOOTH LOW ENERGY	BLUETOOTH CLASIC
Optimizado	Transmisiones cortas	Transmisiones Continuas
Banda de Frecuencia	2.4 Ghz ISM Band (2.402 -2.480 Ghz utilizados)	2.4 Ghz ISM Band (2.402 - 2.480 Ghz utilizados)
Canales	40 canales	79 canales
Canales Usados	FHSS	FHSS
Modulación	GFSK	GFSK, DQPSK, DPSK
Poder de Consumo	~0.01tx a 0.5x de referencia	1 valores de referencia
Velocidad de Datos	L2M PHY : 2 Mb/s / LE 1m PHY: 1 Mb/s / LE Coded PHY (S=2) = 500 Kb/s / LE Codes PHY (S=8): 156 Kb/s	EDR PHY (8DPSK) : 3 Mb/s / EDR PHY (DPSK) : 2 Mb/s /BR PHY (8DPSK) : 1 Mb/s
Máxima Transferencia	Class 1: 100 mW (+20 dBm) / Class 1.5 : 10 mW (+10 dbm) /Class 2: 2.5 mW(+4 dbm) / Class 3: 1mW (0dBm)	Class 1: 100 mW (+20 dBm) / Class 2: 2.5 mW(+4 dbm) / Class 3: 1mW (0dBm)
Topología de Red	Point-to-Point / Broadcast / Mesh	Point to Point

Tabla 1. Comparativa entre el Bluetooth Clásico y el BLE

**Fuente: DataSheet del Fabricante Espressif Systems Version 2.4
2020**

Las características del BLE son las propiedades de un servicio autónomo y es donde los datos actuales están contenidos en jerarquía (valor). El BLE siempre tiene dos atributos: declaración de características (que proporciona metadatos sobre los datos) y el valor de dichas características.

Los protocolos utilizados en el BLE, podemos observarlos en la figura 5.

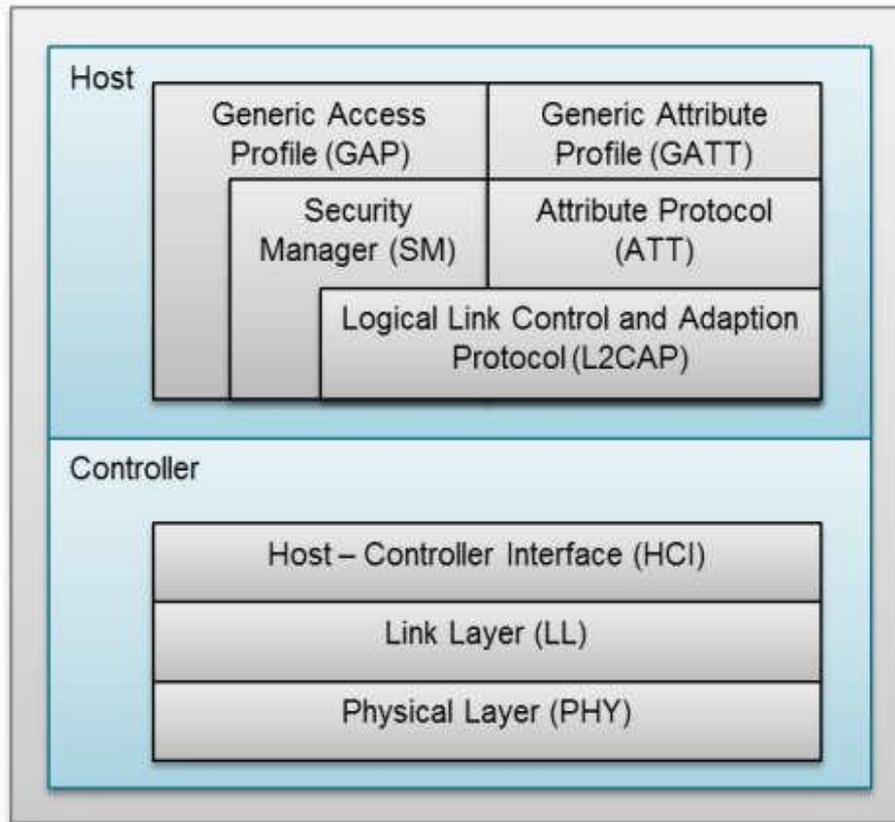


Figura 5. Protocolo de capas utilizados en el BLE

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

La capa física es la que contiene toda la parte de integrados y circuitería, capaz de realizar los procesos de modulación y demodulación, y de transformación de señales analógicas en digitales. La tecnología del BLE es capaz de utilizar hasta 40 canales de 2Mhz.

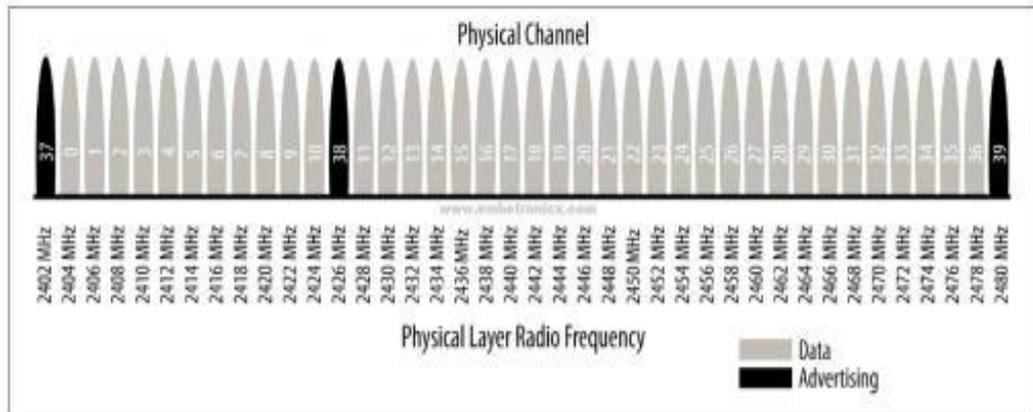


Figura 6. Capa Física del BLE

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

La capa de enlace, es la encargada de gestionar las características y revisión de los mensajes, tanto para el envío como para la recepción. Además de que nos permite definir los roles del BLE (*Advertiser, Scanner, Master o Slave*).

El HCI, es un protocolo que permite la comunicación entre un host y un controlador que utilice el puerto serie de los microcontroladores. Un ejemplo claro de esto, es cuando utilizas tu celular y conectas a la PC, y te permite seleccionar el tipo de conexión UART o USB.

La capa L2CAP (*Logic Link Control and Adaptation Protocol*), es la encargada de los procesos de multiplexación, es decir el traspaso de información entre las diferentes capas. Además de encargarse de la fragmentación y recombinación de los distintos paquetes de información.

Para BLE, la capa L2CAP es fundamental porque además de lo señalado es la encargada de dar acceso y soporte a los dos protocolos fundamentales. Por un lado, ATT (*Attribute Protocol*), un protocolo con arquitectura cliente-servidor, que permite el intercambio de información. Por otro lado, SMP (*Security Manager Protocol*), protocolo que proporciona un *framework* para generar y distribuir claves de seguridad entre dos dispositivos.

En el nivel más alto de la capa de protocolos, encontraremos de forma paralela las capas GAP y GATT. La capa GAP (*Generic Acces Profile*), es la que permite poder generar visibilidad para que pueda emparejarse el BLE con otro dispositivo. Además de establecer los procesos y formas de interacción de las capas bajas, como son:

- Roles de interacción
- Modos de operación y transición entre ellos
- Procedimientos para establecimiento de comunicación
- Modos de seguridad y procedimientos

El protocolo GATT (*Generic Attribute Profile*), es la encargada de definir como dos dispositivos o dos BLE van a intercambiar información. Este proceso es realizado, cuando han entablado comunicación ambos dispositivos emparejamiento dado en la capa GAP.

1.2.2 Software de Referencia para Machine Learning

1.2.2.1 Lenguaje de programación Python

Python es un lenguaje de programación, muy versátil, sencillo, de código abierto y con muchas aplicaciones. Actualmente, está dirigido fundamentalmente a la programación orientada a objetos, sin embargo, su evolución le ha permitido poder desarrollarse en distintos entornos para diferentes tipos de aplicaciones. Una de las ventajas principales de Python, es que permite la creación de código con rutinas mas cortas frente a otros lenguajes, y a la vez cuenta con una gran cantidad de librerías que pueden ser adaptadas fácilmente a distintas necesidades.

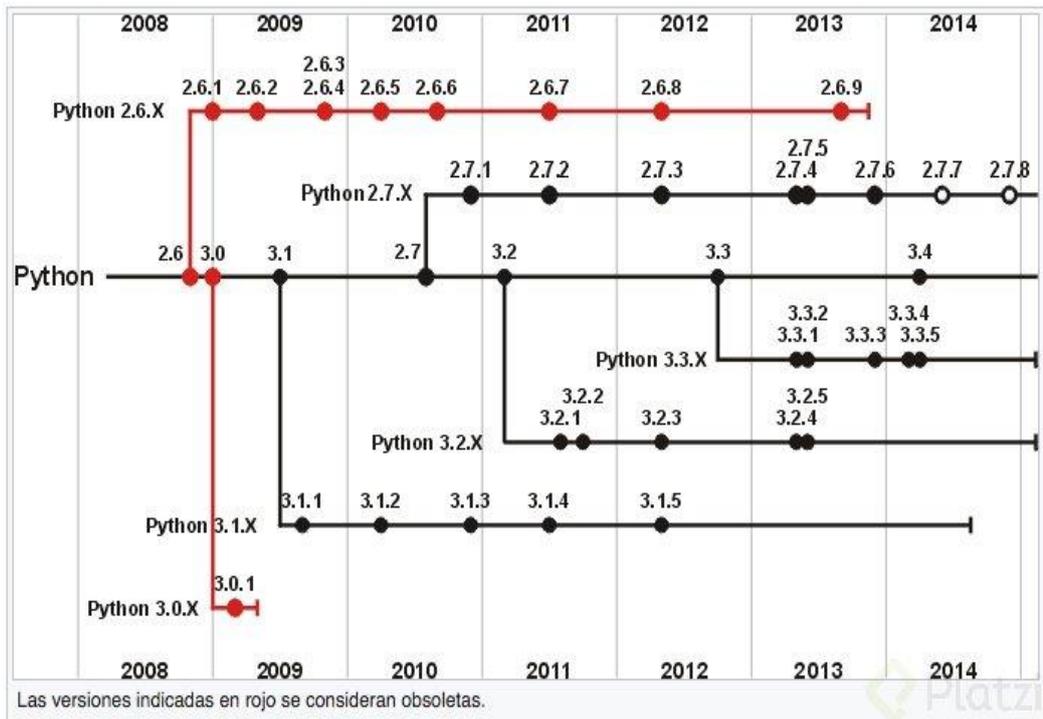


Figura 7. Evolución del Lenguaje de Programación Python en el tiempo

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

Python en Big Data

El uso para este tipo de procesos es usado para la extracción, análisis y conclusiones con grandes cantidades de datos. Para este tipo de procesos, podemos encontrar librerías, muy usadas, como *Pydoop*, ya que permite escribir código con MapReduce y procesar datos tipo *clusters*. Entre otras también encontramos las librerías *Dask* y *Psypark*, que realizan procedimientos similares, en tiempo real.

Python en Data Science

Para este apartado contamos con gran cantidad de librerías para el análisis de datos, usados en diferentes campos de la ciencia, y que actualmente sirve como un apoyo en los trabajos de investigación. Entre sus librerías más conocidas tenemos a Pandas, para el tratamiento de datasets, tenemos a *Numpy*, para trabajar con matrices y listas numéricas, tenemos a *Matplotlib*, para la generación de graficas de acuerdo a los arreglos numéricos, entre otras.

Python en Desarrollo Web

El uso de Python en el desarrollo web, evoluciono constantemente, y hoy permite trabajos más ágiles, con mayor eficiencia y adaptable a todo tipo de entornos. Permite crear páginas más complejas, con menos líneas de código. Entre sus *Framework* más conocidos tenemos a Django, que proporciona la ventaja de crear aplicaciones dinámicas y de gran fluidez. Otro conocido es *Scraping*, que permite obtener información de páginas web, de manera muy sencilla. Es tan popular que es usada en webs conocidas como Instagram, Pinterest, entre otras.

Lo mencionado anteriormente son 3 campos tecnológicos, en donde se puede observar la importancia de Python en el desarrollo de aplicaciones y análisis de datos para un entorno tan cambiante como el que vivimos actualmente.

Para desarrollar el presente trabajo de suficiencia profesional, desarrollamos los distintos códigos mostrados en **Python 2.7**, el cual sigue hoy en día en uso debido a la versatilidad que presente a diferencia de Python 3. Python 2 sigue teniendo aun fuerza en los distintos proyectos, ya que fue en esta versión que se crea la *Python Software Foundation* que actualmente es dueña de la documentación y especificaciones del lenguaje. La fundación baso su modelo en la *Apache Software Foundation*.

A continuación, se presenta una tabla con las principales diferencias entre Python 2 y Python 3.

PYTHON 2.7	PYTHON 3.6
Tiene dos tipos de string: Unicode y No Unicode	Todos los String son Unicode por defecto
La función Xrange () es usada para crear objetos	La función Xrange es reemplazada por la función range()
Separación de variables Int y Long for números no flotantes	Solo hay una variable tipo entero, llamada int, que se comporta como Long.
El tipo de retorno de la división de enteros es int, lo que significa que 5/4 devolverá 1 en lugar de 1,25	El tipo de retorno de la división de enteros es flotante, lo que significa que 5/4 devuelve 1.25
Imprimir es una declaración especial en lugar de una función	Print () es una función incorporada en python 3.6

Tabla 2. Comparativa entre Python 2.7 y Python 3.6

Fuente: <http://www.differencebetween.net/>

1.2.2.2 Herramienta de Data Science Scikit-Learn

Scikit-Learn es una herramienta importante para la ciencia de datos, y que usamos como base para el desarrollo para el presente proyecto. Es un conjunto de librerías, que está desarrollada en Python, y que cuenta con algoritmos para la clasificación, regresión, reducción y un sinfín de posibilidades para en análisis de datos.

La gran variedad de algoritmos que podemos encontrar en *Scikit-learn* la convierten en una herramienta imprescindible para los que comienzan en el mundo del análisis de datos, ya que contiene una interfaz muy sencilla para aprender a programar y analizar nuestros modelos matemáticos, entre ellas sus más destacables como Pandas o *Pybrain* (“Scikit-Learn, Herramienta Básica Para El Data Science En Python,” n.d.)

1.2.2.3 Librerías de Machine Learning

```
# =====  
import numpy as np  
import pandas as pd  
from tabulate import tabulate  
  
# Gráficos  
# =====  
import matplotlib.pyplot as plt  
from matplotlib import style  
import matplotlib.ticker as ticker  
import seaborn as sns  
  
# Preprocesado y modelado  
# =====  
from sklearn.model_selection import train_test_split  
from sklearn.model_selection import GridSearchCV  
from sklearn.model_selection import RepeatedKFold  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.datasets import make_blobs  
from sklearn.metrics import euclidean_distances  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.linear_model import Ridge  
  
from skopt.space import Real, Integer  
from skopt.utils import use_named_args  
from skopt import gp_minimize  
from skopt.plots import plot_convergence  
  
# Varios  
# =====  
import multiprocessing  
from itertools import product  
import random
```

Figura 8. Principales Librerías Usadas en Scikit-Learn

Fuente: https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html

Pandas, Manejo de Datos

Pandas es una de las principales herramientas usadas para el análisis de datos, pues nos permite manejar con facilidad grandes cantidad de datos. Con esta librería podemos leer archivos, de diferentes extensiones, así como también agrupar, dividir, seleccionar, ordenar, priorizar entre otras los distintos tipos de datos. (“10 Librerías Python Para Data Science y Machine Learning - SolidQ Blogs,” n.d.)

Numpy, Operar Matrices

Numpy, es otra librería muy usada en análisis de datos también desarrollada en Python, su función principal es la de crear estructuras numéricas y poder realizar todo tipo de operaciones matemáticas con las mismas. Es usada adicionalmente como base, para la creación de otras librerías, puesto que su motor de procesamiento consume pocos recursos (“10 Librerías Python Para Data Science y Machine Learning - SolidQ Blogs,” n.d.)

Plotly, Imagen y Clic

Plotly, es una librería que nos permite poder realizar gráficas y visualización de datos, de gran ayuda porque de esta manera se puede tener un mejor análisis de los datos procesados. A diferencia de *Matplotlib*, este permite interactuar en tiempo real con la gráfica. Otra aplicación importante para el desarrollo de gráficas, es *Seaborn*, a diferencia de *Plotly* esta permite realizar mayor cantidad de ajustes. (“10 Librerías Python Para Data Science y Machine Learning - SolidQ Blogs,” n.d.)

Estas 3 librerías constituyen las principales, entre otras, usadas en el presente proyecto, ya que la librería base que estamos usando que es *Pygarl*, utiliza estas, para la recolección, procesamiento, análisis y testeo de los datos.

1.2.2.4 Aprendizaje en Machine Learning

En esta sección hablaremos del tipo de aprendizaje que hemos elegido en *Machine Learning*, sobre todo dándole un énfasis especial al de las Maquinas de Soporte Vectorial, o mas conocidas como SVM (Support Vector Machine), en el cual se basa nuestro proyecto para la predicción correcta de los gestos.

Los casos de estudio de Machine Learning, podemos tener fundamentalmente dos métodos el de clasificación y el de regresión. De manera sencilla podemos indicar que los casos de regresión tienen como objetivo predecir valores continuos, mientras que los de clasificación, tienen como objetivo poder asignar un conjunto de datos a una clase.

Las Maquinas de Soporte Vectorial (SVM), se fundamentan en los conceptos de máximo margen de clasificación, que está a la vez se basa en los conceptos de hiperplano.

Hiperplano y Clasificador de Rango Máximo

Es un espacio p -dimensional, donde un hiperplano se define como un subespacio plano y afín de dimensiones $p-1$. Cuando mencionamos afín, indicamos que el subespacio no tiene porque pasar por el origen. En un espacio de dos dimensiones, tenemos que el hiperplano es un subespacio de 1 dimensión, por lo tanto, podemos indicar que se trata de una Recta. En un espacio tridimensional, un hiperplano es un subespacio de 2 dimensiones, es decir se trata de un plano convencional. Cuando tenemos dimensiones con subespacios > 3 es complejo de visualizar el hiperplano, pero el concepto inicial de $p-1$ dimensiones se cumple.

La definición matemática de un hiperplano, en el caso de dos dimensiones, se escribe de acuerdo a la ecuación de una recta.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

Dados los parámetros $\beta_0 + \beta_1$ y β_2 todos los pares de valores $x = (x_1, x_2)$ para los que se cumple la igualdad son puntos del hiperplano. Si generalizamos la ecuación podemos tener para p-dimensiones, lo siguiente:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0$$

Y de igual manera, para todos los puntos definidos por el vector $(x = x_1, x_2, \dots, x_p)$ que cumplen con la ecuación perecerán al hiperplano.

Un caso especial es cuando X no satisface la ecuación:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p < 0$$

o

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p > 0$$

Podemos observar que el punto x, cae a un lado o al otro del hiperplano. Así pues, se comprueba que el hiperplano divide un espacio p-dimensional en dos mitades. Para saber en que lado se encuentra un determinado punto x, solo hay que calcular el signo de la ecuación.

Un ejemplo, para entender lo antes mencionado, es lo que podemos observar en la Figura 9, en donde tenemos un espacio bidimensional. La ecuación que describe el hiperplano, es una recta, es $1+2x_1 +3x_2 =0$. Si analizamos a fondo, tenemos que la región de azul representa el espacio en el que se encuentran todos los puntos para los que $1+2x_1 +3x_2 > 0$ y la región roja para los puntos en los cuales $1+2x_1 +3x_2 =0 < 0$.

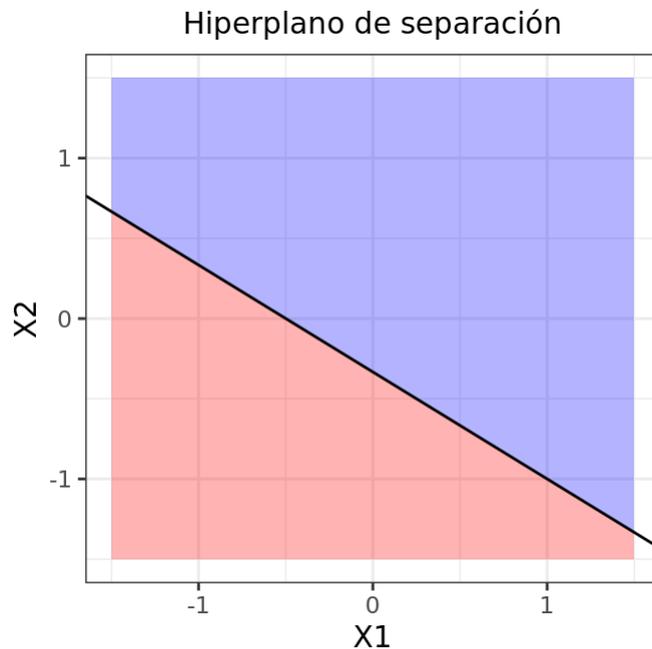


Figura 9. Hiperplano de la ecuación $1+2x_1 +3x_2 =0$

Fuente: (Amar Rodrigo, 2020)

Support Vector Classifier o Clasificador de Vectores de Soporte (SVC).

Para entender mejor este concepto, debemos entender que un *Maximal Margin Hiperplane* (Margen máximo de hiperplano) está definido como un hiperplano que consigue un mayor margen, es decir, que la distancia mínima entre el hiperplano y la observación es lo más grande posible.

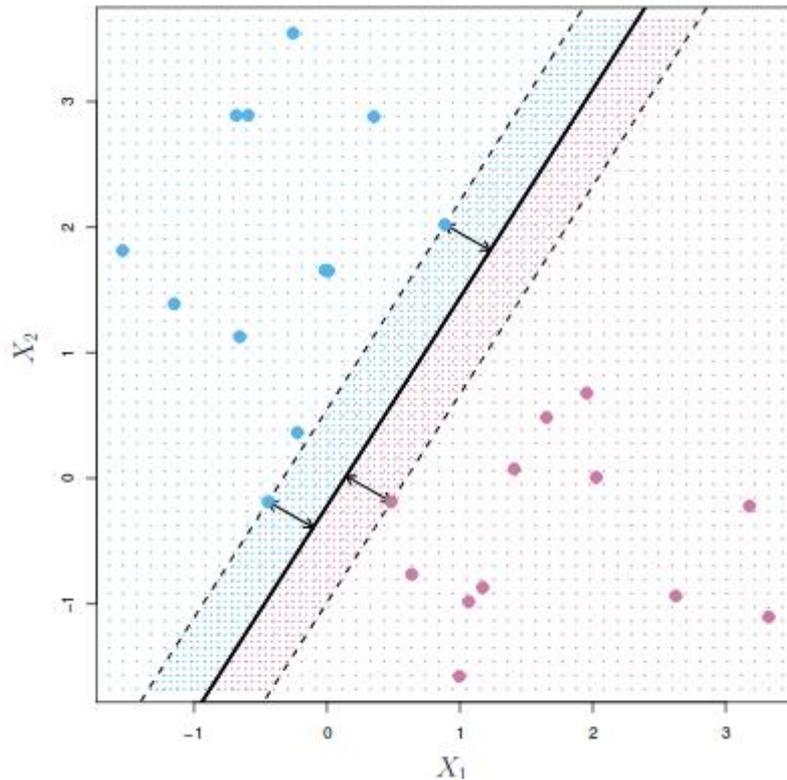


Figura 10. Visualización de un Máximo Margen de Hiperplano Teórico

Fuente: (Amar Rodrigo, 2020)

En la imagen podemos observar el margen máximo del hiperplano para un entrenamiento de 5 variables. Los tres puntos que se separan de la línea recta, que a la vez son equidistantes a la recta, y se encuentran sobre las líneas punteadas son las que definen la anchura del margen. A estos vectores que salen desde el Margen Máximo del Hiperplano, se le conoce como vectores de soporte, estos son vectores en un espacio p -dimensional y soportan al Margen Máximo de Hiperplano. Cualquier modificación de estos puntos (vectores de soporte) conlleva en la alteración del Margen Máximo del Hiperplano.

Los clasificadores de vectores de soporte (SVM), nacen de la necesidad de poder separar o clasificar lo más perfectamente posible (alta probabilidad) las muestras en diferentes clases. Los SVM son modelos robustos y con una alta capacidad predictiva, y son ideales para muestras que están en constante evolución

Los SVM para poder lograr un mejor modelo predictivo, en vez de buscar el rango máximo del hiperplano que consigue que las muestras estén de lado correcto de las clasificaciones, se permite que las muestras estén en el lado incorrecto o inclusive en el hiperplano.

Para entender lo mencionado en el apartado anterior, analizaremos la figura 11. En el cual se tiene un modelo que desea clasificar las variables en números de 1 dígito y de 2 dígitos. Tenemos que la línea continua representa el hiperplano, y las líneas discontinuas representan los márgenes máximos a cada lado. Los puntos 2, 3, 4, 5, 6 y 7 son muestras que están en el lado correcto del margen, por lo cual podemos concluir que están bien clasificadas. Las muestras 1 y 8, a pesar que se encuentran dentro del margen, están en el lado correcto del hiperplano por lo cual podemos concluir que también están bien clasificadas. Las muestras 11 y 12, se encuentran en el lado erróneo del hiperplano por lo cual podemos concluir que están mal clasificadas. Lo ideal hubiera sido que el 11 y el 12 se encontraran alrededor del 10.

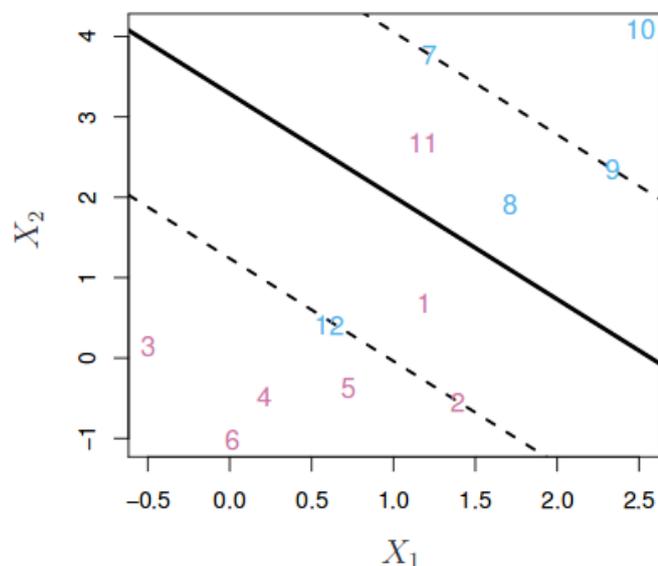


Figura 11. Imagen del clasificador de un SVM

Fuente: (Amar Rodrigo, 2020)

La identificación de un hiperplano de un clasificador de vectores de soporte, que clasifique la mayoría de las muestras de manera correcta, es un problema de optimización convexa. Este proceso de optimización es llamado el hiperparámetro de Tuning C, este C controla el número y la severidad de las violaciones del margen que se toleraran en el proceso de ajuste. Por ejemplo, si C fuera infinito, no se permite ninguna violación del margen y, por lo tanto, el resultado es equivalente al máximo margen de clasificación perfectamente separables. Cuando más se aproxima C a cero, menos errores y muestras estarán en el lado incorrecto del plano y la clasificación será mejor. El parámetro C a fin de cuentas es el hiperparámetro encargado de controlar el balance entre bias y la varianza del modelo. En la práctica, su valor optimo se identifica mediante una validación cruzada, de posibles valores del mismo.

El proceso de optimización, tiene una característica en la cual las muestras que se encuentra justo en el margen o escapan de ellas influyen en el hiperplano. A estas muestras, representadas como observaciones en el hiperplano, se le conoce como vectores de soporte y son las que definen el clasificador obtenido. Por tal motivo, el parámetro C es el que controla el balance entre bias y varianza. Cuando C es pequeño, el margen es más ancho, y más observaciones violan el margen, convirtiéndose en puntos de vectores de soporte. Cuando mayor es C, menor es el margen, menos observaciones serán vectores de soporte y el clasificador resultante tendrá menor bias, pero mayor varianza. En resumen el SVC, es ideal cuando el límite de separación es aproximadamente lineal (Amar Rodrigo, 2020).

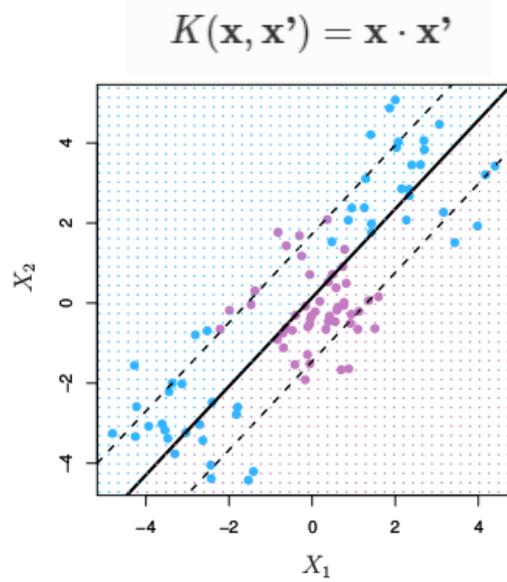
Máquinas de Vectores de Soporte (SVM)

Existen muestras en las cuales los grupos no podrán ser separados linealmente, lo cual no significa que no lo sean en espacios de mayores dimensiones, en este caso hablaríamos de tres dimensiones a más.

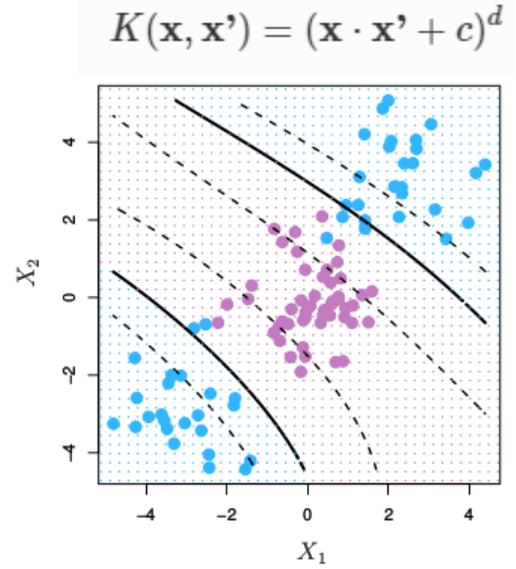
El SVM, se puede considerar como una extensión del SVC obtenida al aumentar las dimensiones de los datos y por lo tanto sus hiperplanos que en muchas ocasiones se entrecruzan.

La dimensión es un conjunto de datos que puede transformarse combinando o modificando cualquier de sus valores, para poder hacer ello hablamos de *Kernels*. Un **kernel**, es una función que devuelve el resultado del producto escalar de dos vectores realizando en un nuevo espacio dimensional distinto al original en el que se encuentran los vectores originarios. Si sustituimos este producto escalar por un kernel, se obtienen directamente los vectores de soporte en la dimensión correspondiente al kernel. Entre los tipos de Kernel, tenemos los 3 principales (Figura 12).

- a) **Kernel Lineal.** Si se emplea este tipo de kernel, el resultado será un SVC.
- b) **Kernel Polinómico.** Cuando se emplea $d=1$ y $c=0$, el resultado es el mismo el del kernel lineal. Si $d>1$, se generan límites de decisión no lineales, aumentando la no linealidad a medida que aumenta d . No se recomienda emplear valores mayores a 5 por problemas de sobreajuste.
- c) **Kernel Gaussiano (RFB).** El valor de γ controla el comportamiento del kernel, cuando es muy pequeño, el modelo es similar al lineal, a medida que aumenta su valor, lo hace su flexibilidad. Este kernel, tiene una ventaja en el cual solo tiene dos parámetros a optimizar, tanto γ como C .

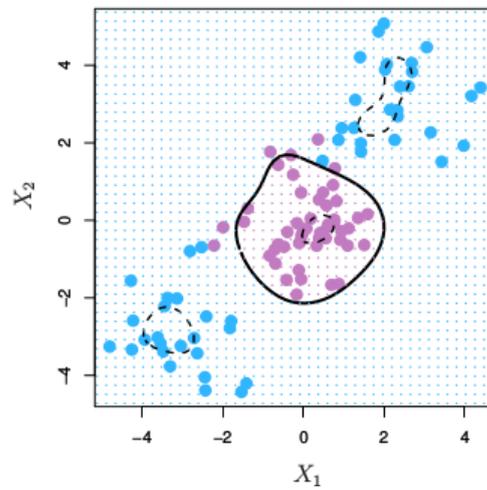


KERNEL LINEAL



KERNEL POLINOMICO

$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$



KERNEL GAUSSIANO

Figura 12. Muestra Grafica de los distintos tipos de Kernel en SVM

Fuente: (Amar Rodrigo, 2020)

1.2.3 Software de Diseño Electrónico

1.2.3.1 Diseño y Esquemas Electrónicos - EAGLE

EAGLE es un software de automatización de diseño electrónico (EDA) que permite que los desarrolladores puedan tener un entorno como de trabajo para realizar diseños de circuitos impresos (PCB), a la vez que permite también conectar componentes, enrutarlos entre otras funcionalidades. Actualmente Eagle forma parte de la familia de Autodesk, con lo cual se ve repotenciado porque permite utilizar comandos y funcionalidades propias de software de diseño, como el conocido AutoCad.

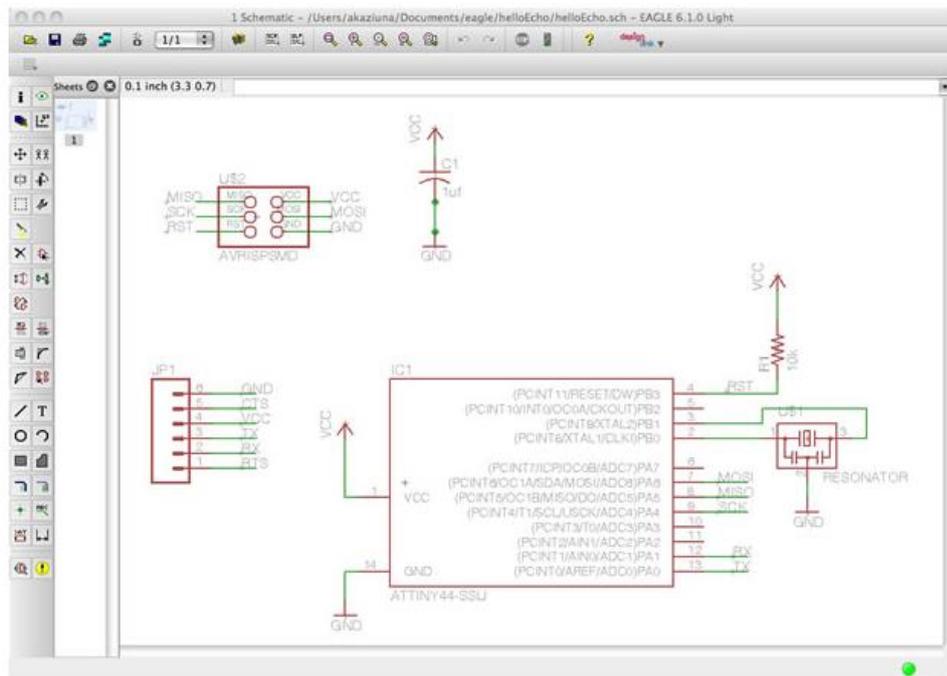


Figura 13. Diagrama Esquemático de Microcontrolador y SW en EAGLE

Fuente: http://archive.fabacademy.org/archives/2016/doc/electronics_design_eagle_spanish.html

1.3 Definición de Términos Básicos

- **Acelerómetro.** Un acelerómetro es un sensor que sirve para medir la fuerza de aceleración, ya sea estática o dinámica, estos sensores son útiles para medir vibraciones y movimientos en un sistema.
- **Microcontrolador.** (abreviado μ C, UC o MCU) Es un circuito integrado que es programable, capaz de ejecutar diferentes códigos guardados en su memoria. Consta de diferentes bloques funcionales para cada categoría de trabajo.
- **Bluetooth.** Es una especificación industrial para redes inalámbricas de área personal (WPAN) creado por Bluetooth *Special Interest Group, Inc.* que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz.
- **Pulsadores e Interruptores.** Un pulsador es un operador eléctrico que, cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, lo interrumpe. - Un interruptor es un operador eléctrico, cuya función es abrir o cerrar un circuito de forma permanente.
- **Traqueostomía.** Una traqueostomía es un procedimiento quirúrgico realizado con objeto de crear una abertura dentro de la tráquea, a través de una incisión ejecutada en el cuello, y la inserción de un tubo o cánula para facilitar el paso del aire a los pulmones.
- **UCI.** El área de cuidados intensivos - UCI - es el lugar en donde se brindan cuidados de alta complejidad para los pacientes graves de otros servicios del hospital.
- **Ventilación Mecánica.** También conocida como ventilación asistida/controlada por volumen y regulada por presión: en el ventilador se programan el volumen corriente, la frecuencia respiratoria y el tiempo inspiratorio.

- **Machine Learning.** Es una disciplina científica del ámbito de la Inteligencia Artificial, que entre sus funciones está la de desarrollar modelos que puedan analizar y ayuden a interpretar diferentes cantidades de datos.
- **Punto a Punto.** Las comunicaciones punto-a-punto, en general, se refieren a una conexión limitada a dos extremos o nodos, por ejemplo: computadoras o dispositivos móviles.
- **Soporte de Maquina Vectorial.** Es un Modelo Supervisado de aprendizaje con algoritmos asociados que analizan los datos y reconocen patrones, que se utiliza para la clasificación y el análisis de regresión en la Inteligencia de Negocios.
- **Bias.** También conocido como Sesgo, pertenecen al concepto de exactitud. Es la diferencia entre la predicción esperada de nuestro modelo y los valores verdaderos.
- **Varianza.** Hace referencia a la cantidad que la estimación de la función objetivo cambiara si se utiliza diferentes datos de entrenamiento
- **Placa o Circuito Impreso.** Son superficies construidas con buses, en los cuales los materiales conductores permiten la interacción entre distintos componentes electrónicos.
- **Dispositivos Dedil.** Son Dispositivos creados para uso en los dedos de la mano, por ejemplo, son oxímetros, o férulas.
- **Baterías 18650.** Las baterías 18650 son pilas de litio recargables muy parecidas a las AA, pero con un tamaño más grande. Son baterías ligeras, pero con una elevada capacidad energética y resistencia a las descargas.

CAPÍTULO II: METODOLOGIA DE DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL

2.1 Delimitación Temporal y Espacial del Proyecto

2.1.1 Teórica

Para realizar la propuesta del diseño del dispositivo dedil inteligente, recurrimos a manuales y conocimientos técnicos de alta fiabilidad por expertos en temas de inteligencia Artificial. De tal modo que el presente proyecto está basado en conocimiento de microcontroladores, lenguaje de programación en Python, algoritmos de Machine Learning y de redes basadas en IoT.

2.1.2 Temporal

El diseño propuesto en el proyecto se estará realizando entre los meses de agosto a diciembre del año 2020. Durante este tiempo se realizarán las pruebas de Hardware y Software, inicialmente se realizarán las pruebas simulando las condiciones en el que se encuentran estos pacientes Post-UCI. Las pruebas podrían aplazarse debido al estado de emergencia, los permisos necesarios por parte del hospital y a la delicada salud de los pacientes antes mencionados.

2.1.3 Espacial

El presente trabajo está orientado hacia cualquier paciente que presente Traqueostomía o problemas de comunicación sonora, a lo largo de nuestro territorio nacional, no obstante, debido al estado de emergencia que atravesamos por la pandemia del Covid-19, las pruebas serán realizadas en pacientes del hospital María Auxiliadora de San Juan de Miraflores.

2.2 Determinación y Análisis del problema

2.2.1 Descripción de la Realidad Problemática

En el distrito de San Juan de Miraflores, existe una IPRESS Hospital Maria Auxiliadora que, en la época actual, se encuentra atendiendo a pacientes con Covid-19. Estos pacientes muchas veces son ingresados e internados por falta de oxígeno, saturación baja, siendo los casos más graves ingresados al área de Cuidados Intensivos para que posteriormente estén por un tiempo prolongado en Ventilación Mecánica hasta que sus pulmones puedan recuperarse de la Neumonía Covid. Actualmente los datos estadísticos de estos casos no son de libre acceso en el Perú, sin embargo, podemos darnos una idea de la cantidad de pacientes traqueostomizados, en el año 2018 en el departamento de Cirugía y Cuello del Hospital Maria Auxiliadora se realizaron 330 intervenciones quirúrgicas, de las cuales 89 fueron Traqueostomizados, representando un total de 26.9% de casos que fueron sometidos a estos procedimientos, teniendo mayor incidencias en adultos mayores en el rango de 30 a 59 años. (Datos, 2018, p. 154)

Un caso interesante de estudio internacional, se dio en España en el año 2008, en el cual se observó que, de un grupo de 521 pacientes en ventilación mecánica, el 9.8% estaba con traqueostomía. Además, se observó que la mortalidad en UCI y en el hospital fue menor en pacientes traqueostomizados siendo del 13.7% frente a los 26.4% de los pacientes sin traqueostomía. Por tal motivo, se concluye que la traqueostomía es un proceso habitual que puede salvar la vida de muchos pacientes en UCI. (Salcedo & Frutos-Vivar, 2008, pp. 91–93).

Según Ángel Cañete, Roberto Pawlowicz, Haydee Zotarez, Mario Gerónimo & Alberto (2012, pp. 12–13) en la *Revista de Medica del Sanatorio Sagrado Corazón de Argentina*, hacen un recopilatorio de distintas recomendaciones para tratar a los pacientes traqueostomizados, por ejemplo; es de vital importancia tratar de buscar alternativas de

comunicación para el paciente, se recomienda facilitar las expresiones del paciente por medio de gestos y escrituras en cuanto sea posible, el paciente puede experimentar frustración con cualquier método si no es efectivo, permitir la autonomía e involucrar al paciente en la toma de decisiones sobre su cuidado. Por lo antes expuesto, se concluye la importancia del uso de herramientas tecnológicas que permitan una comunicación efectiva en este tipo de pacientes.

2.2.2 Justificación del Problema

El presente proyecto se justifica en que el diseño propuesto mejorara el proceso de recuperación de los pacientes traqueostomizados, luego de pasar por el área de cuidados intensivos. Mejorando sus capacidades de comunicación muchas veces disminuidas por el estado prolongado en el ventilador mecánico.

Además, se busca que los pacientes puedan desarrollar su autonomía e involucrarse en la toma de decisiones con respecto a sus cuidados. Nuestro dispositivo dedil, ayudará en mejorar el proceso de comunicación y transmisión de ideas mientras el paciente aprende nuevamente a exhalar aire y expulsarlo por el esófago mediante la vibración de cuerdas vocales como si fuese un eructo, se espera que con el tiempo adquiera la calidad de voz y entonación que se tuvo regularmente. Por otro lado, también se espera, que mientras va recuperando las fuerzas musculares, pueda usar otros medios convencionales de comunicación.

Nuestra propuesta permite, además, que sea accesible y de fácil uso para todo los pacientes y personal médico, además que los pacientes en estas condiciones, mantengan el mínimo contacto posible frente a enfermeras o parientes al cuidado del mismo, pero siempre en constante comunicación, evitando de esta manera posibles recaídas por contaminación de agentes externos.

2.2.3 Formulación del Problema

a. Problema General

- ¿De qué manera la propuesta de diseño de un dispositivo dedil inteligente mejorará la calidad de vida en la recuperación de los pacientes Post-UCI Traqueostomizados en la IPRESS Hospital María Auxiliadora?

b. Problemas específicos

- ¿Cuáles serían las medidas y características específicas del dispositivo dedil inteligente para que pueda adaptarse mejor e interpretar correctamente los datos obtenidos en pacientes traqueostomizados?
- ¿Cómo se llevaría a cabo la implementación y capacitación al personal médico y pacientes sobre el uso del dispositivo dedil inteligente?
- ¿Cuál sería el sistema de emergencia en caso el dispositivo presentara fallas de funcionamiento o falsos positivos?
- ¿Cuál es el consumo energético y como se asegura la continua funcionalidad del dispositivo dedil inteligente?

2.3 Modelo de Solución Propuesto

2.3.1 Diseño del Sistema

2.3.1.2 Requerimiento de Diseño

El sistema propuesto debe, básicamente, detectar los gestos del dedo del paciente y poder predecir cuál es su necesidad en base a la matriz pictórica que se muestra en el Anexo A. El diseño es bajo un escenario de una habitación del Hospital María Auxiliadora. Eventualmente dirigido a pacientes traqueostomizados, luego de pasar largas estancias en UCI, se asume las funciones motoras y/o mentales están limitadas, el cual les impide una correcta comunicación de sus necesidades y manipulación de sistemas mecánicos o electrónicos complejos.



Figura 15. Habitación Médica Del Hospital María Auxiliadora

Fuente: <https://www.facebook.com/HospitalMariaAuxiliadoraOficial>

2.3.1.2 Requerimiento de Diseño de Sistema Dedil Inteligente Portátil de Uso Medico

En este apartado se menciona los requerimientos generales del diseño del prototipo propuesto, considerando su uso de tipo portátil y de uso médico. La fabricación de este tipo de dispositivos, actualmente está normado por el ISO, entre ellos el ISO 13485, que es el sistema de gestión de la calidad del diseño específico para fabricante de dispositivos médicos, este amplifica y perfecciona el ISO 9001. Existen fabricantes en china que ayudan a la fabricación masiva de dispositivos cumpliendo estos estándares. (NQA, n.d.)

Se espera una vez aprobado el prototipo y con la correcta financiación poder llegar a cumplir los estándares de ISO. Lo presentado a continuación, forma parte del análisis realizado por el Magister Edgar Oporto en su Tesis de Maestría. (Diaz, 2014, pp. 92–94)

a) Sistema Electrónico Autónomo

Un sistema de uso portátil debe ser autónomo; por ello, debe disponer de todos los recursos necesarios para trabajar por si solo. En la siguiente figura se muestra un diagrama de bloque básico para un dispositivo médico portátil. Teniendo en cuenta que el diagrama

incorpora varias piezas clave de tecnología y cada una de ellas debe evaluarse antes de poder seleccionar una plataforma de uso de hardware y software para su aplicación. (Chris Edwards, n.d.)

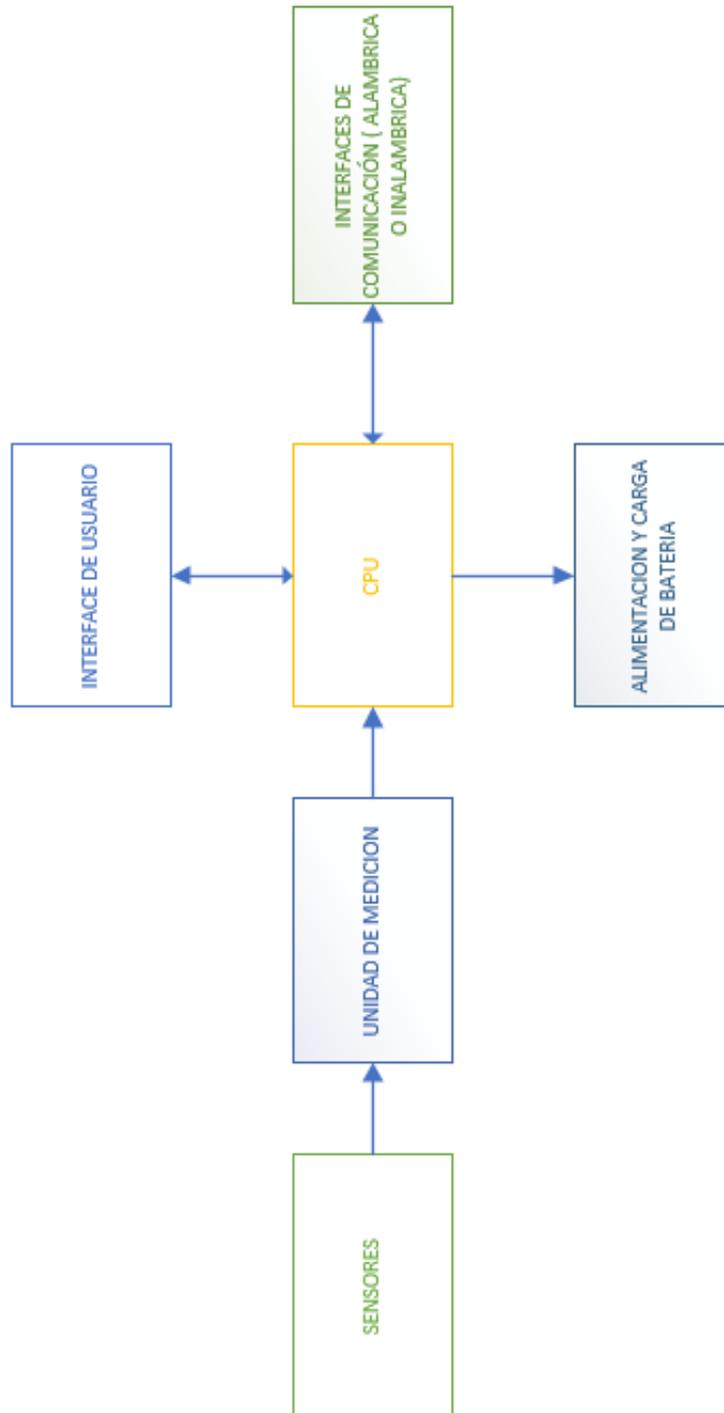


Figura 16. Diagrama de Bloques de un Dispositivo Medico Portátil

Fuente: Modificado de <https://cutt.ly/kgllCV3>

b) De muy bajo peso

El peso es un factor importante en el diseño, ya que este debe ser el adecuado para evitar inconvenientes al paciente, teniendo en cuenta la poca fuerza muscular que posee. El peso total dependerá de los pesos parciales de los componentes, siendo el más importante el de la batería de litio 18650 que tiene un peso de 29.8 gramos, otro componente es el peso de la tarjeta electrónica y carcasa (MPU6050 tiene 2.1 gramos, ESP32 tiene 6.8 gramos, cables SW entre otros 29g) teniendo 37.9 gramos. Teniendo en total para el dispositivo propuesto un peso 67.7 gramos.



Figura 17. Peso de Batería 18650

Fuente: <https://www.bateriasdelitio.net/>

c) De reducido Volumen

El volumen propuesto para el dispositivo dedil debe ser lo más reducido posible, de tal modo que pueda acoplarse sin problemas a los dedos y manos del paciente. Esto indicaría que sea lo más plano y sin protuberancias posibles. De tal modo que el tamaño aproximado del prototipo es de 100 mm de largo, la parte del sensor colocado en el dedo es de 25mm longitud x 18 mm ancho x 4 mm de altura, mientras que la parte colocada en la muñeca de la mano es de 30mm largo x 40mm ancho x 10 mm altura.

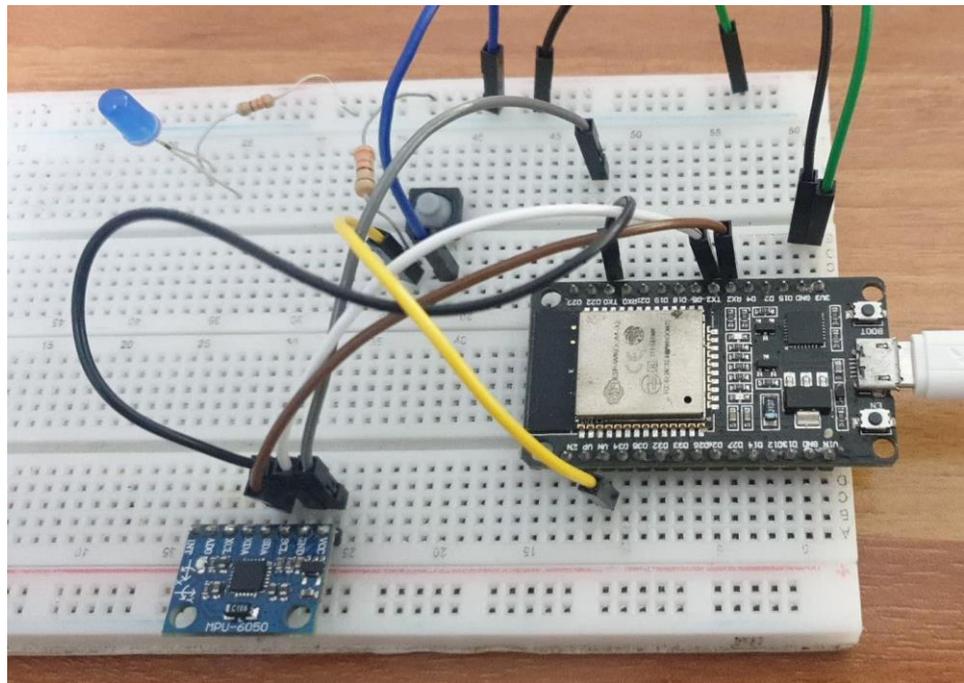


Figura 18. Pruebas de medida aproximada del Dispositivo Dedil

Fuente: Elaboración Propia.

d) Fácilmente adosable al cuerpo

Nuestro dispositivo dedil, debe ser fácilmente adosable a los dedos de la mano del paciente, de tal manera que todo el sistema quede fijo, y el mismo no cuelgue o entre en contacto con otras superficies, para evitar falsos datos, deterioro o mal funcionamiento.

e) Funcionamiento a baterías

El dispositivo propuesto estará en funcionamiento las 24 horas del día, ya que ayudará a la recuperación del paciente, por tal motivo se descarta el uso de pilas o baterías convencionales. Para nuestro prototipo se usaron baterías de litio.

f) Robusto

El dispositivo en su construcción debe tener materiales que lo hagan resistente a los movimientos bruscos, agitaciones, ajeteo y al uso permanente del mismo. Por eso los materiales usados en el case, las soldaduras, y cableado deben estar sujetos firmemente para evitar fallas por uso prolongado o malas maniobras.

g) Residente a la humedad y a los fluidos del cuerpo

El cuerpo humano genera distintos tipos de fluidos, entre ellos el sudor que debemos evitar entre en contacto con el dispositivo, al igual que algún otro fluido propio del ambiente o causado por manipulación de líquidos como medicinas o comida. Por lo antes expuesto el dispositivo debe ser impermeable.

h) Facilidad de Uso

El dispositivo debe ser fácilmente programado por el personal médico, y una vez esté en la mano del paciente, deberá funcionar sin ninguna intervención o manipulación del paciente.

i) Alta confiabilidad

El dispositivo esta creado para fines médicos por tal motivo su función principal es la proporcionar una alternativa de comunicación al paciente, que podría salvaguardar la vida del mismo. Por tal motivo el dispositivo debe ser completamente funcional y confiable en su funcionamiento.

j) No emplear cables

Este requerimiento es uno de los más importantes, ya que el tipo de paciente al que va dirigido el dispositivo, se encuentra en un área donde sus funciones vitales son monitoreadas constantemente por tal motivo es altamente recomendable que el dispositivo sea de uso inalámbrico, para evitar interferencias con otros dispositivos. El dispositivo se debe comunicar en forma inalámbrica con los elementos que solo sean necesarios, utilizando los protocolos de seguridad electrónica más actualizados.

2.3.1.3 Diagrama de Bloques del Sistema del Dispositivo Propuesto

El sistema propuesto para desarrollar el Dispositivo Dedil Inteligente, se detalla en el Anexo D. En los siguientes apartados de este capítulo se detallará cada uno de los elementos del diagrama, para que pueda cumplirse con los requerimientos de diseño impuesto.

2.3.2 Desarrollo del Prototipo del Hardware

2.3.2.1 Sistema de Adquisición de Datos

El sistema de adquisición de datos consta de los siguientes elementos:

- Selección del sensor.
- Selección del Microcontrolador
- Hardware de Adquisición de datos
- Código de Adquisición de datos

a) Selección del Sensor

Dentro de la gama de fabricantes de este tipo de sensor acelerómetro con giroscopio disponibles comercialmente tenemos a los siguientes fabricantes.

- Analog Devices
- Freescale Semiconductor
- STMicroelectronics
- Invensense

Todos ellos fabrican sensores de aceleración, tanto de salida analógica como digital. Se decide por los de tipo digital pues permiten una conexión directa con el procesador sin necesidad de circuitos previos. Esta facilidad evita la conexión de circuitos de filtrado de ruido, circuitos de amplificación o acondicionamiento de señal. (Díaz, 2014, p. 98). Por lo anterior expuesto es que se eligió el MPU6050, ya que cuenta en su arquitectura con Acelerómetro y giroscopio, a diferencia de otros módulos que vienen ambas características separadas. En el apartado 1.2.1.2 se señalan mayores características de este módulo.

b) Selección del Microcontrolador y Placa de Desarrollo

En la tesis del Magister Edgar Oporto (Díaz, 2014, pp. 101–102) se hace un análisis de las principales características que debería tener un microcontrolador para este tipo de aplicaciones propuestas, para la propuesta de nuestro Dispositivo Dedil Inteligente se está usando el ESP32 WROOM .

- Soporte de lenguaje de alto nivel.

El microcontrolador debe estar optimizado nativamente para su programación en distintos lenguajes de programación. Actualmente la tendencia es programar en C y Python. Teniendo en cuenta las ventajas de esos lenguajes en cuanto al menor tiempo de desarrollo y elaboración de códigos más complejos.

- Velocidad de Proceso.

Característica no crítica, puesto que el sensor MPU6050 incorpora lógica que facilita la detección de los 6 ejes que posee en complementación con los procesos del microcontrolador. En general, la aplicación no requiere de altas tasas de velocidad se trabaja en el prototipo a velocidades de 38400 baudios.

- **Herramientas de Desarrollo.**

La disponibilidad de entornos de desarrollo IDE, depuradores y simuladores, equipo de programación en sistema, emuladores en circuito o emuladores en sistema, empleo de monitorización para reducir tiempos de desarrollo. Para nuestra propuesta se usa el IDE del Arduino, porque nos permite usar las distintas librerías que ya existen para el sensor MPU6050. Sin embargo, el fabricante del ESP32, Espressif Systems también cuenta con su propio entorno de desarrollo.

- **Interfaces de comunicación**

El microcontrolador debe contar con buses de comunicación como RS-232 (usados para el Bluetooth común) e I²C (usados para distintos tipos de sensores). Para nuestro caso nuestro modulo ESP32 cuenta con lo propuesto.

- **Facilidad de Programación**

Deben existir diferentes y abundantes librerías que sean fácilmente adaptables o de uso con el entorno de desarrollo del microcontrolador.

- **Consumo de potencia.**

Característica importante puesto que el microcontrolador o la placa de desarrollo debe permitir el uso con baterías, y a la vez debe contar con reguladores de voltajes para poder usar distintos tipos de sensores. Nuestro modulo ESP32 consume una corriente de operación de 80mA.

- **Voltaje de Operación.**

El empleo de bajo voltaje reduce apreciablemente el consumo de potencia de un circuito (Potencia = voltaje x corriente). Existen dispositivos que trabajan desde 1.8 V, sin embargo, deben tomarse en cuenta las tensiones de los demás dispositivos a emplear.

- **Tamaño de memorias RAM y ROM.**

Este apartado no es crítico para nuestro proyecto porque se está tomando en cuenta que el microcontrolador ejecutara una aplicación en específico. Para nuestro proyecto no se consume más del 4% de la RAM del Módulo ESP32.

- **Interfaces de E/S**

Disponibilidad de entradas y salidas, analógicas y digitales para el uso de temporizadores, interrupciones, puertos paralelos entre otros. Nuestro modulo usa E/S Tipo GPIO.

- **Costos**

Actualmente no presenta problema, ya que la mayoría de dispositivos electrónicos para nuestro prototipo son de bajo costo y accesibles.

- **Dimensiones y Encapsulado**

Debe ser de tamaño reducido, en el mejor de los casos en versiones SMD el microcontrolador o los distintos sensores. Para nuestro caso si existe una versión SMD, sin embargo, para nuestro prototipo utilizaremos el módulo con las dimensiones propuestas la tabla 1.

c) Hardware de Adquisición de Datos

El circuito de adquisición de datos consta del sensor de acelerómetro y giroscopio MPU 6050 de *Invensense*. Las conexiones al procesador se realizan mediante la interface de comunicación serial I²C a través del cual el procesador accede a recoger los datos del sensor MPU6050. Como estamos usando el módulo ESP32, encontraremos la línea de comunicación SDA (Serial Data) en el pin GPIO21 y la línea de comunicación SCL (Serial Clock) en el GPIO22. Los datos son leídos una vez que el SW sea presionado. La información del PinOut del ESP32 se muestra en la figura 1.

Para el presente proyecto existen dos etapas, primero la etapa de entrenamiento del modelo para la detección de gestos; este hardware funciona básicamente con el de adquisición de datos, en el cual en el microcontrolador está grabado un código que es capaz de leer los datos del MPU6050, posteriormente estos datos también son leídos por el script creado en Python para que pueda generarse un DataSet y posteriormente pasar al modelo de entrenamiento.

En la segunda etapa, con el modelo ya entrenado, se trabaja con el hardware final del prototipo que se basa en el de la primera etapa, pero que incluye el soporte el caso de emergencia (El Laser y Buzzer).

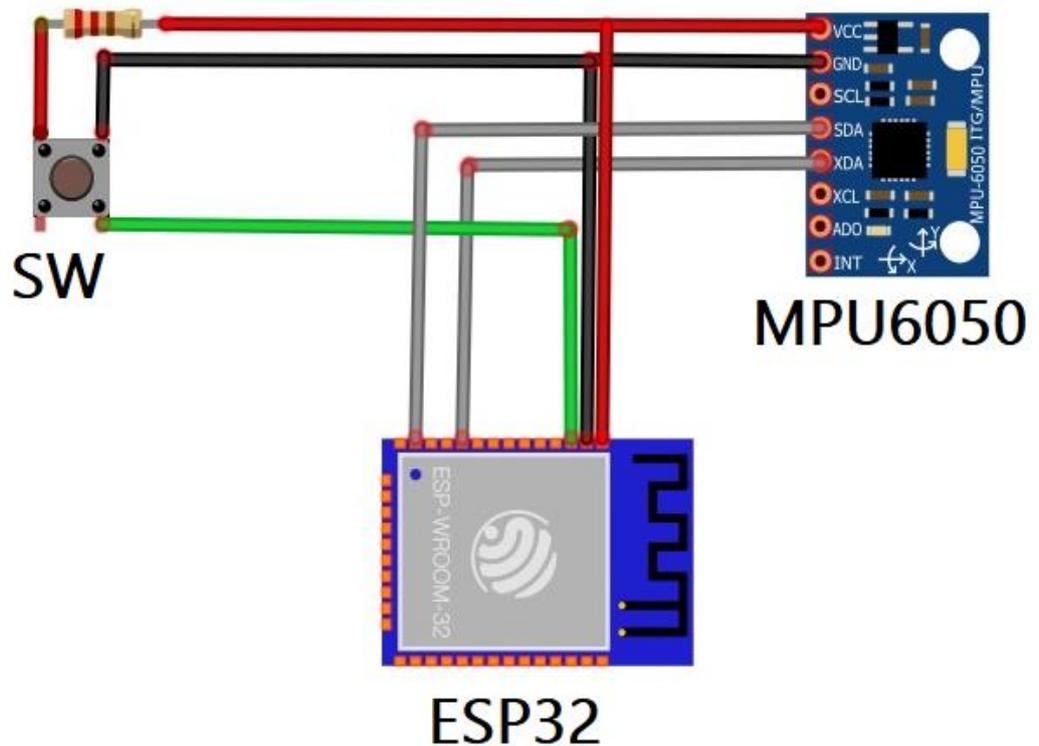


Figura 19. Circuito del Hardware de Adquisición de Datos.

Fuente: Elaboración propia

d) Código de Adquisición de Datos

El siguiente código presentado, forma parte de un boceto creado inicialmente por el User JohnChi (JohnChi, n.d.) en la web oficial de Arduino, nos sirvió como base para trabajar el código del proyecto luego se modificó parámetros para poder trabajar con el ESP32 y los Switches. El código presentado en este apartado, lee los valores brutos del MPU6050. El código completo puede ser observado en el Anexo C.

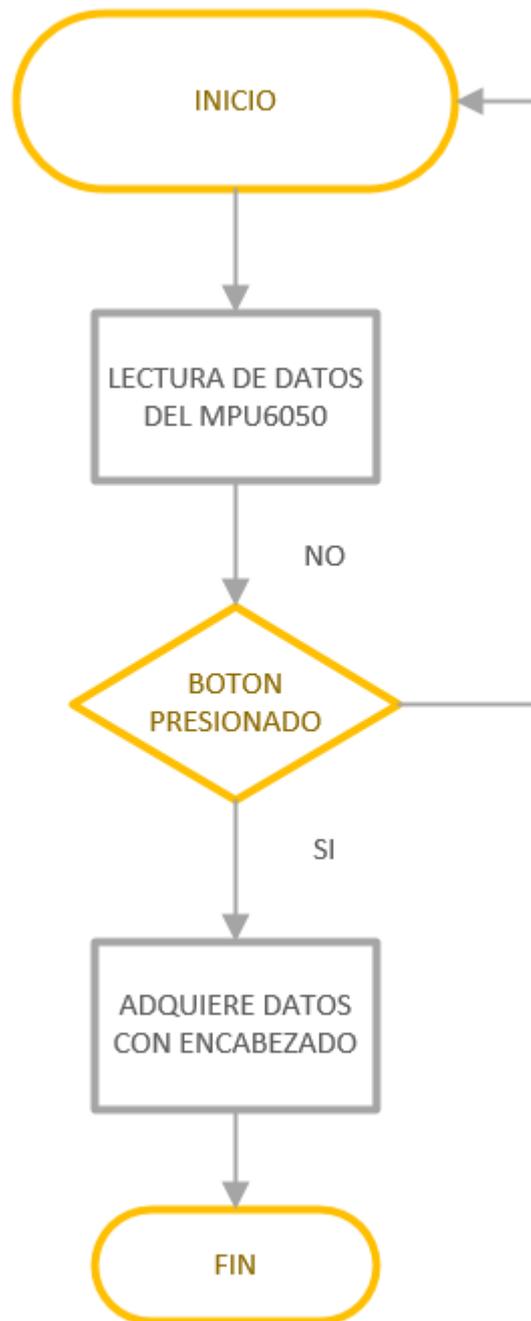


Figura 20. Diagrama de Flujo del Software de Adquisición de Datos

Fuente: Elaboración propia

2.3.2.2 Diseño del PCB del Dispositivo Dedil

Para un mejor desempeño del dispositivo propuesto, se diseñó una placa en circuito impreso (PCB), para el diseño de usaron los componentes electrónicos; Microcontrolador ESP32, Switches, MPU6050, Leds, Borneras y Resistencias. Todos los componentes usados son de tecnología tipo *through-hole*. El diseño de la placa se realizó en fibra de vidrio de 1 sola capa.

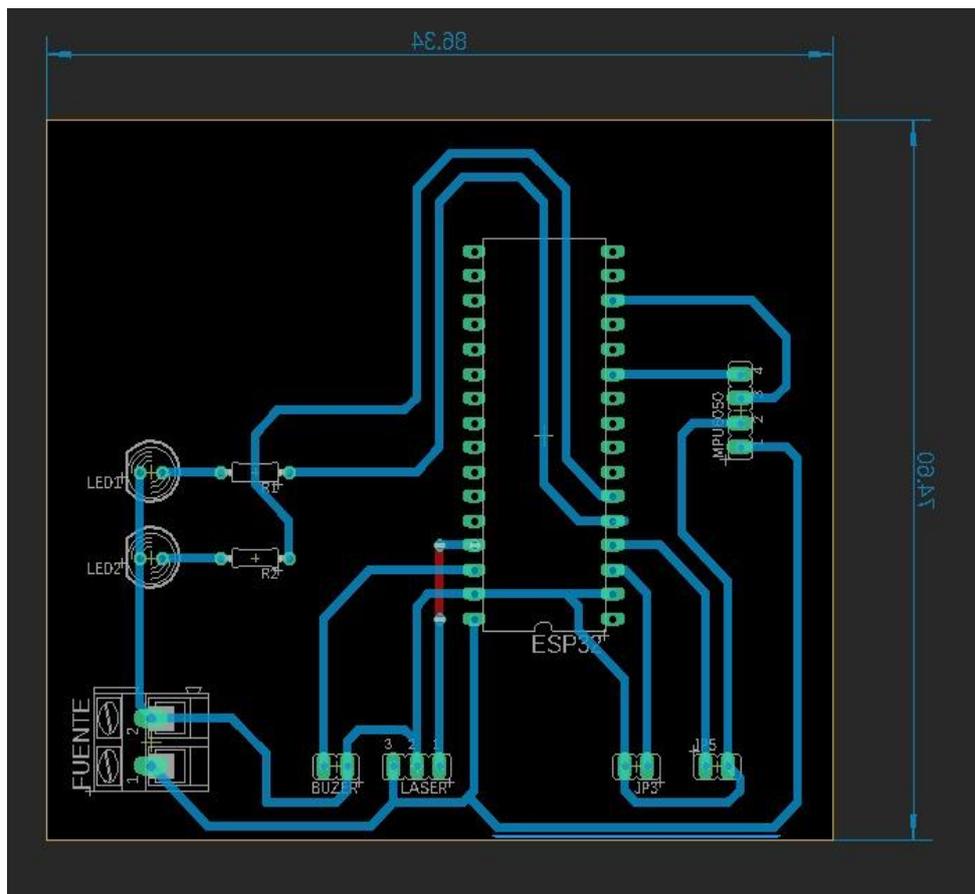


Figura 21. Diseño del PBC con medidas finales

Fuente: Elaboración Propia.

2.3.2.3 Estructura de Soporte del Dispositivo Dedil

Dado que el dispositivo estará cerca al paciente en todo momento, debemos asegurar que los componentes y la PBC no sufran ningún tipo de daño. Por tal motivo, fue necesario diseñar e imprimir en 3d un case que pueda mantener las condiciones adecuadas para el correcto funcionamiento de nuestro dispositivo. Este diseño está construido para poder albergar al PCB y al case de las baterías tipo Li-ion. Esta estructura fue impresa usando filamento del tipo ABS.

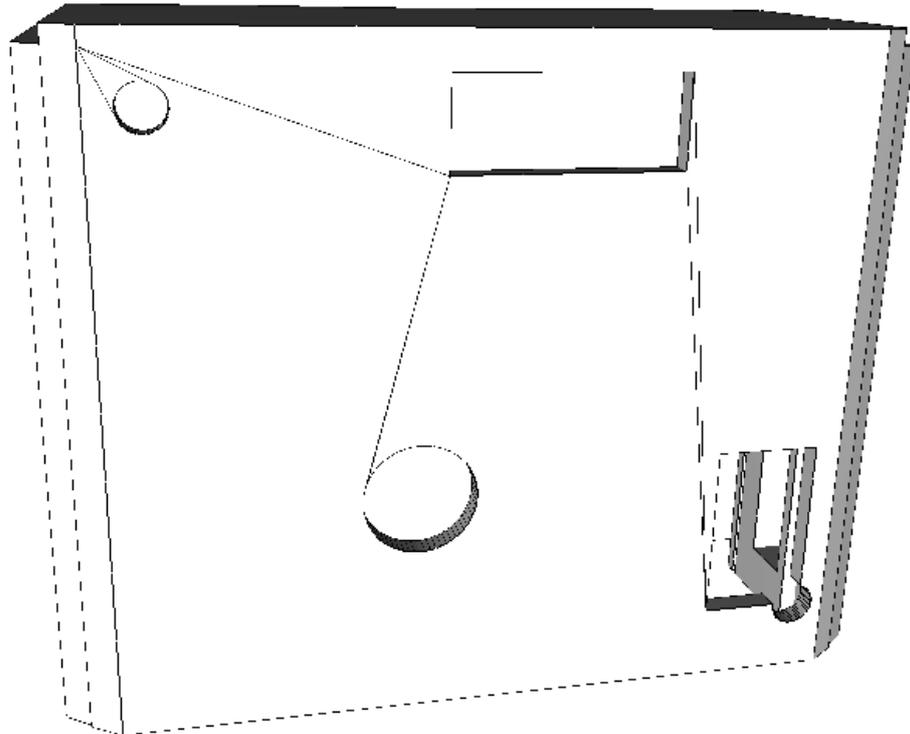


Figura 22. Diseño del Case en 3D en software Ultimaker

Fuente: Elaboración propia

2.3.3 Desarrollo del Software del Prototipo

La Pipeline del procesamiento de datos, se realizó basándonos en la librería *Pygarl* (Terzi Federico, n.d.) el código puede ser observado en el Anexo E, desarrollado en Python con módulos de *Scikit-Learn*, este es utilizado especialmente para el reconocimiento de gestos. La estructura del código usado esta esquematizado en la Figura 19, fragmentada en una serie de componentes que se ocupan de procesar progresivamente los datos recibidos. Cada módulo implementa el patrón *Observer*, con el fin de propagar de manera eficiente el evento a todas las capas posteriores. Las razones y objetivos de cada nivel se explicarán en los siguientes subpárrafos.

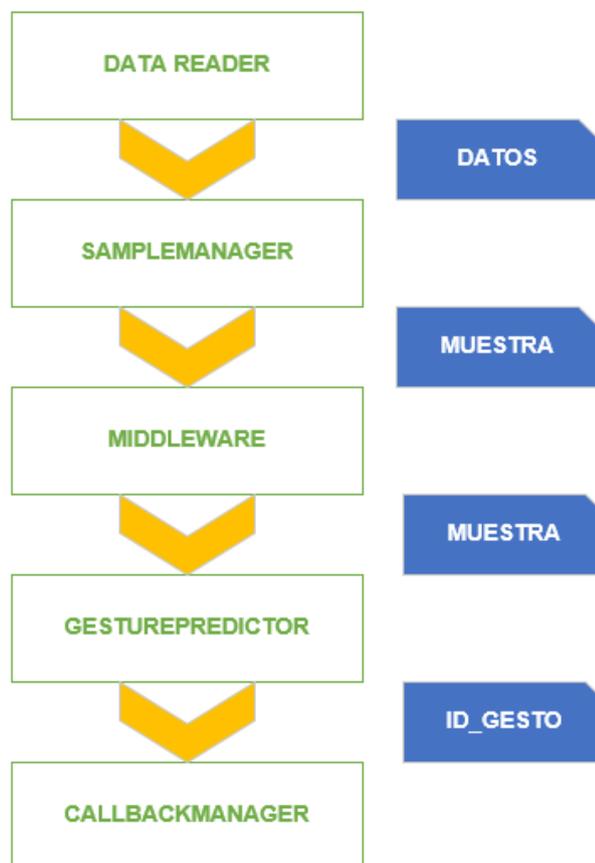


Figura 23. Estructura de Software

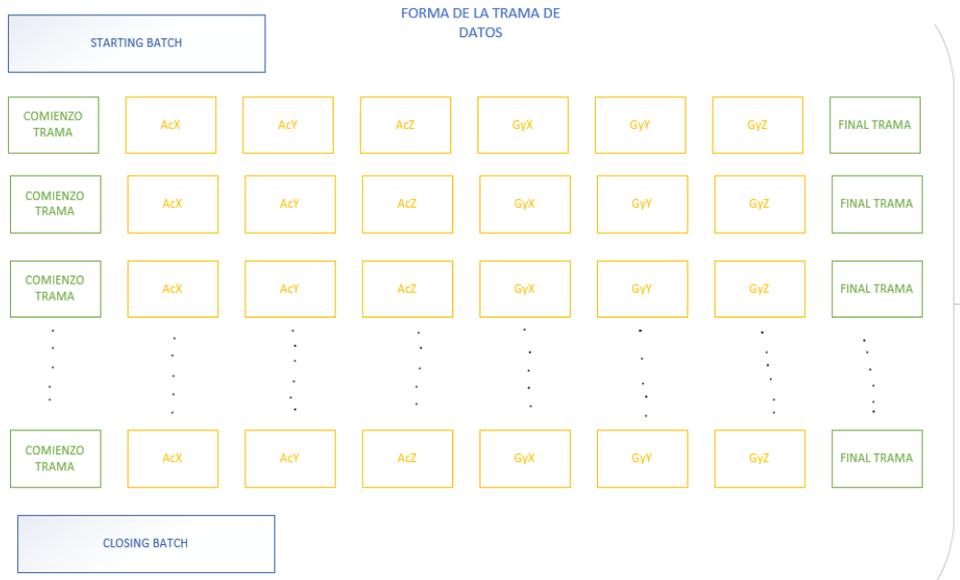
Fuente: Modificación de estructura de Librería PyGarl

a) Software para la Recolección de Datos – Data Reader

Para recolectar y almacenar los datos obtenidos por los sensores se desarrolló un Script en Python 2, basándonos en código de Frederic Terzi (Terzi, n.d.). Este Script trabaja principalmente leyendo los datos del puerto COM, es por este mismo puerto que el IDE de Arduino recepción los datos que envía el MPU6050.

Para que el Script, empiece a recolectar datos es necesario que este presionado el Switch del dispositivo, e inmediatamente el paciente realice el gesto que desea grabar, posteriormente se debe dejar de presionar el Switch para que se haya concluido con la generación de la 1 trama de datos.

Estos datos son guardados en un archivo de texto, que posteriormente será utilizado en la etapa del entrenamiento. Los datos que son obtenidos al presionar el Switch, y que forman tramas, se diferencian de otros datos o falsos positivos, porque tienen un indicador la palabra STARTING BATCH al comienzo y CLOSING BATCH al final. Entre estas palabras claves, se podrá notar una serie de 6 grupos de números, estos corresponden a los valores del MPU6050 en el siguiente orden; AcX, AcY, AcX (valores del acelerómetro) y GyX, GyY, GyX (valores del giroscopio). En el archivo de texto donde son guardados estos valores, pueden contener diferentes cantidades de tramas, puesto que no todos los gestos se realizan en los mismos milisegundos.



```

STARTING BATCH
START 5308.0, 336.0, 16688.0, 416.0, 146.0, -136.0 END
START 5324.0, 280.0, 16712.0, 392.0, 160.0, -145.0 END
START 5232.0, 320.0, 16748.0, 403.0, 139.0, -145.0 END
START 5328.0, 272.0, 16640.0, 377.0, 151.0, -138.0 END
START 5328.0, 236.0, 16628.0, 383.0, 135.0, -144.0 END
START 5328.0, 272.0, 16640.0, 377.0, 151.0, -138.0 END
CLOSING BATCH

```

Figura 24. Forma de la trama de datos para el entrenamiento del “1”

Fuente: Elaboración Propia

b) Sample Manager

Es el módulo que, habiendo recibido las tramas que contienen datos y señales de control del DataReader, se encarga de generar Muestras. La Muestra es un conjunto de marcos que identifican un movimiento, pero que aún no están asociados a ningún gesto. Una vez empaquetada, la muestra se envía a la siguiente capa, que realizará más manipulaciones en los datos recibidos.

c) MiddleWare

Es un componente que se ocupa del procesamiento y manipulación de las muestras. En el proceso recoge y devuelve las muestras, aplicando una serie de transformaciones. Esta sección forma parte del SVM Classifier, puesto que es la encargada que las muestras cumplan con su estructura mínima como se observa en la Figura 24. En caso no cumplan con la misma esta las descarta.

d) Gesture Predictor

Es el componente que se ocupa de asociar un gesto con la Muestra recibida. Es importante señalar que la lógica utilizada para estimar el gesto depende de la implementación específica. Una vez que obtenga la muestra de la capa anterior, envíe una cadena a la siguiente, llamada ID DE GESTO, que identifica el gesto.

e) *CallBack Manager*

Se encarga de llamar al controlador de eventos, si se ha recibido el gesto asociado a él. En la fase inicial, se configuro el ID de Gesto con las funciones CallBack, esta misma es la encargada de llamar al evento, cuando se tiene configurada la opción de predicción para que pueda identificar y asociar el ID de Gesto.

f) *Clasificador SVM*

Es un clasificador que usa el algoritmo Máquinas de vectores de soporte, o SVM, para determinar el gesto asociado con una muestra en particular. Esto se debe a la posibilidad de poder cambiar la función de decisión, es decir, el mecanismo subyacente a la clasificación, para adaptarse de manera óptima a una gran variedad de casos.

Uno de los factores más importantes a tener en cuenta a la hora de utilizar estos algoritmos es la elección de los parámetros, cuya correcta calibración es fundamental para obtener buenos resultados. Para nuestro clasificador se utiliza una técnica combinatoria, se crea una cuadrícula que contiene los valores posibles para cada parámetro y luego se prueba cada combinación para evaluar cuál tiene el mejor resultado. También se deduce que el número de valores debe limitarse al mínimo necesario para evitar el aumento exponencial del tiempo computacional.

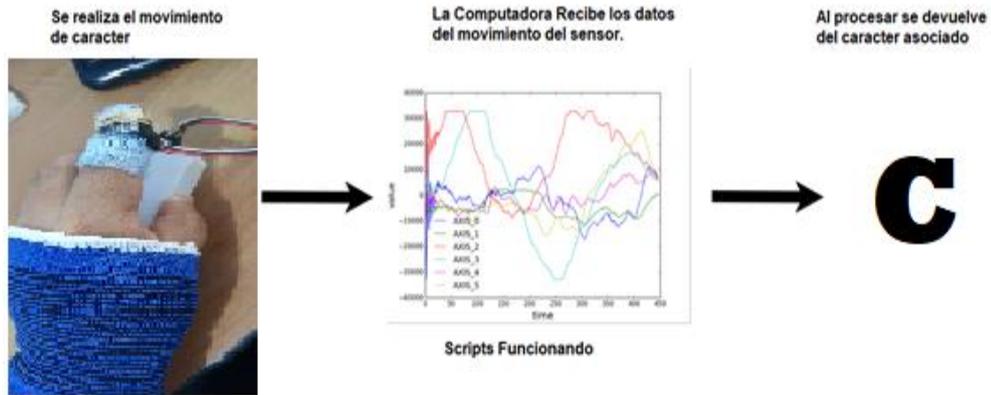


Figura 25. Proceso básico de predicción de caracteres

Fuente: Elaboración Propia

2.3.4 Entrenamiento del Clasificador y Análisis de la Matriz de Confusión

Una vez creado el conjunto de datos, el siguiente paso fue crear un modelo capaz de determinar el gesto asociado con una nueva muestra (Anexo G). Por lo tanto, fue necesario crear un clasificador adecuado para el propósito y, para encontrar el clasificador óptimo para el proyecto nosotros elegimos el clasificador SVM, basándonos en otros trabajos realizados como el de Chandra (Chandra, Rajkumar, & Kumar, 2019), donde los parámetros obtenidos para procesamiento de señas dio un mejor resultado frente a otros clasificadores.

Como se explicó en el apartado anterior, utilizamos una técnica combinatoria. Se elaboró una cuadrícula de posibles valores y, posteriormente, se probaron todas las combinaciones para encontrar la mejor.

Cuadrícula de posibles valores:

Los parámetros considerados fueron la función de decisión, o núcleo, y el parámetro de penalización por error. Los posibles valores se muestran en la Tabla 4.

Funcion de Decision (Kernel)			
Lineal	Polinomial	Sigmoide	RBF
Parametro de Penalizacion por error (C)			
10^0	10^{-1}	10^{-2}	10^{-3}

Figura 26. Valores posibles de los parámetros del Clasificador SVM

Fuente: Código Fuente de Pygarl

Estos valores llevaron a la creación de 16 combinaciones posibles, todas probadas individualmente para evaluar la mejor. Los resultados del entrenamiento del clasificador se pueden observar en la Tabla 4.

Kernel	C	resultado
Lineal	0.001	0.994
RBF	0.001	0.078
Polinomial	0.001	0.987
Sigmoide	0.001	0.078
Linear	0.01	0.993
RBF	0.01	0.078
Polinomial	0.01	0.987
Sigmoide	0.01	0.078
Linear	0.01	0.993
RBF	0.01	0.077
Polinomial	0.01	0.992
Sigmoide	0.01	0.124
Linear	1	0.993
RBF	1	0.078
Polinomial	1	0.981
Sigmoide	1	0.141

Tabla 3. Resultados del Entrenamiento del SVM en base a los parámetros del Kernel C

Fuente: Elaboración Propia

La combinación de parámetros que condujo al mejor resultado se resalta en verde, es decir, aquella cuya función de decisión es lineal y tiene el parámetro de penalización por error igual a 0.001.

Habiendo obtenido este valor, y teniendo el DATASET, en el cual del total se usa 75% para entrenamiento y 25 % para testear el mismo, procedemos con el entrenamiento. Para ello fue necesario testear para poder obtener la mínima cantidad de data necesaria para que los pacientes puedan dar a conocer sus necesidades mediante nuestro dispositivo. Para el presente caso se obtuvieron datos de 3, simulando el entorno de poder mover mínimamente el brazo y usen únicamente el dispositivo dedil.

Persona 1: Elsa Crespo Santillan – Edad 37 años – DNI 43724233

Con el primer voluntario al ser un total de 12 caracteres, se decidió probar con 20 muestras por carácter, teniendo en total 220 muestras. El tiempo empleado para la obtención de las muestras fue de 18 minutos 40 segundos. A continuación, se muestran los resultados que ejecuta el SckitLearn para los valores del clasificador SVM y la Matriz de Confusión.

Podemos observar en la figura 27, se obtiene el kernel Lineal como el mejor resultado. Al igual que el entrenamiento tiene 78% de score.

Adicionalmente Observamos en la figura 28, la matriz de confusión, da resultados aun con dificultades para poder reconocer los caracteres, por ejemplo, el carácter “1” se confunde fácilmente con el carácter “7” y la “b” al igual que otros casos. Descartando de esta manera que con 240 muestras no es la cantidad de muestras ideales para el entrenamiento.

```

9.
"avoid this warning.", FutureWarning)
[CV] ..... kernel=rbf, C=0.001, score=0.115384615385, total= 0.0s
[Parallel(n_jobs=1)]: Done 4 out of 4 | elapsed: 0.0s remaining: 0.0s
[CV] kernel=rbf, C=0.001 .....
[CV] ..... kernel=rbf, C=0.001, score=0.1, total= 0.0s
[Parallel(n_jobs=1)]: Done 5 out of 5 | elapsed: 0.0s remaining: 0.0s
[CV] kernel=rbf, C=0.001 .....
[CV] ..... kernel=rbf, C=0.001, score=0.116279069767, total= 0.0s
[Parallel(n_jobs=1)]: Done 6 out of 6 | elapsed: 0.1s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.807692307692, total= 0.0s
[Parallel(n_jobs=1)]: Done 7 out of 7 | elapsed: 0.1s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.78, total= 0.0s
[Parallel(n_jobs=1)]: Done 8 out of 8 | elapsed: 0.1s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.883720930233, total= 0.0s
[Parallel(n_jobs=1)]: Done 9 out of 9 | elapsed: 0.2s remaining: 0.0s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.115384615385, total= 0.0s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.1, total= 0.0s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.116279069767, total= 0.0s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.807692307692, total= 0.0s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.78, total= 0.0s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.883720930233, total= 0.0s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.115384615385, total= 0.0s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.1, total= 0.0s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.116279069767, total= 0.0s
[Parallel(n_jobs=1)]: Done 18 out of 18 | elapsed: 0.4s finished
C:\Users\Victo\anaconda3\envs\py2\lib\site-packages\sklearn\model_selection\_search.py:842:
DeprecationWarning: The default of the 'iid' parameter will change from True to False in v
ersion 0.22 and will be removed in 0.24. This will change numeric results when test-set siz
es are unequal.
DeprecationWarning)
SVC(C=0.001, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
kernel='linear', max_iter=-1, probability=True, random_state=None,
shrinking=True, tol=0.001, verbose=False)
Confusion Matrix:
[[7 0 0 0 0 1 0 0 0 0 0 1]
 [0 3 1 1 0 0 1 0 0 0 0 0]
 [0 0 4 1 0 0 0 0 0 0 0 0]
 [0 1 1 6 0 0 0 0 0 0 0 0]
 [0 0 0 0 4 0 0 0 0 0 0 0]
 [0 0 0 0 2 5 0 0 0 0 0 1]
 [1 0 0 0 0 0 6 0 0 0 0 0]
 [0 0 1 0 0 1 0 5 0 0 0 0]
 [0 0 0 0 0 0 0 0 7 0 1 0]
 [0 0 1 0 0 0 0 0 0 4 0 0]
 [1 0 0 0 0 0 0 0 0 0 5 0]
 [0 0 0 0 0 0 0 0 0 0 0 6]]
FINAL SCORE: 0.7848101265822784
Saving the model to the output file: model.svm
DONE

```

Figura 27. Valores de entrenamiento con 240 muestras.

Elaboración: Fuente Propia

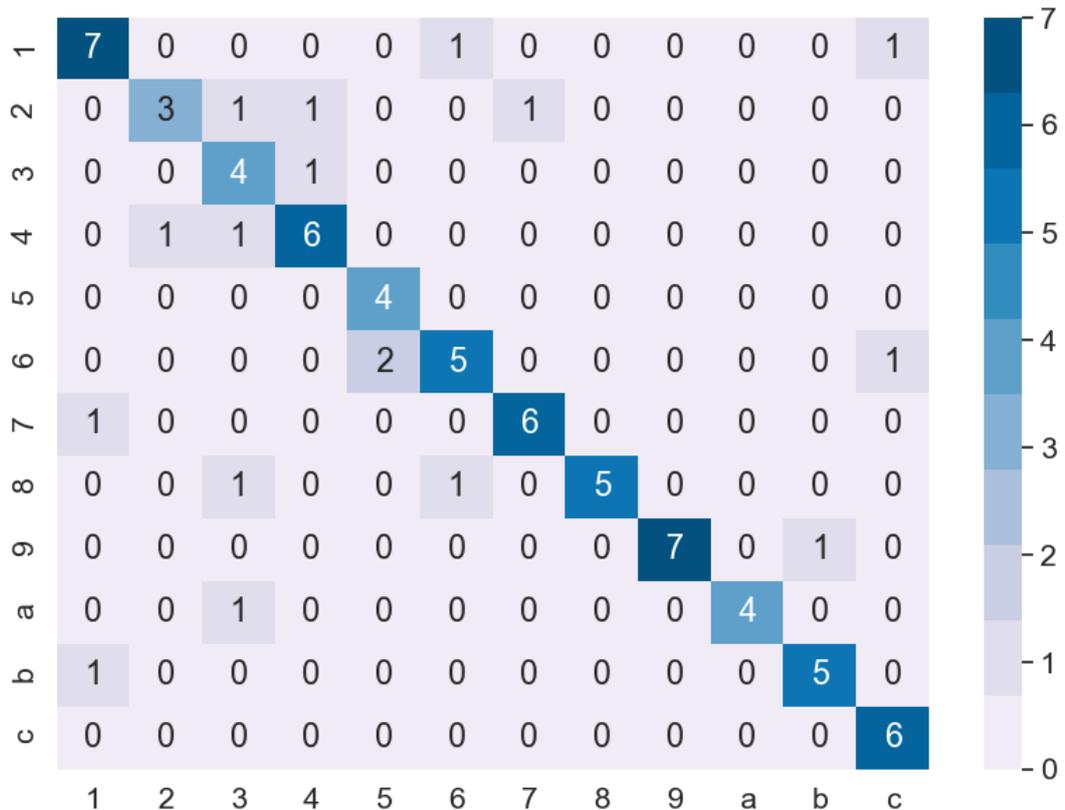


Figura 28. Matriz de confusión con 240 muestras

Fuente: Elaboración Propia.

Persona 2: Carlos Bullon Delgado – Edad 32 años – DNI 45281164

Con el segundo voluntario al ser un total de 12 caracteres, se decidió probar con 40 muestras por carácter, teniendo en total 480 muestras. El tiempo empleado para la obtención de las muestras fue de 33 minutos 10 segundos. A continuación, se muestran los resultados que ejecuta el SckitLearn para los valores del clasificador SVM y la Matriz de Confusión.

Podemos observar en la figura 29, se obtiene el kernel Lineal como el mejor resultado. Al igual que el entrenamiento tiene 88% de score.

Adicionalmente Observamos en la figura 30, la matriz de confusión, da resultados aun con dificultades para poder reconocer los caracteres, por ejemplo, el carácter “5” se confunde fácilmente con el carácter “6” al igual

que otros casos. Descartando de esta manera que con 480 muestras no es la cantidad de muestras ideales para el entrenamiento.

```

[Parallel(n_jobs=1)]: Done 5 out of 5 | elapsed: 0.3s remaining: 0.0s
[CV] kernel=rbf, C=0.001 ..... kernel=rbf, C=0.001, score=0.1, total= 0.0s
[Parallel(n_jobs=1)]: Done 6 out of 6 | elapsed: 0.5s remaining: 0.0s
[CV] kernel=linear, C=0.01 ..... kernel=linear, C=0.01, score=0.881818181818, total= 0.0s
[Parallel(n_jobs=1)]: Done 7 out of 7 | elapsed: 0.5s remaining: 0.0s
[CV] kernel=linear, C=0.01 ..... kernel=linear, C=0.01, score=0.922330097087, total= 0.0s
[Parallel(n_jobs=1)]: Done 8 out of 8 | elapsed: 0.6s remaining: 0.0s
[CV] kernel=linear, C=0.01 ..... kernel=linear, C=0.01, score=0.9, total= 0.0s
[Parallel(n_jobs=1)]: Done 9 out of 9 | elapsed: 0.7s remaining: 0.0s
[CV] kernel=rbf, C=0.01 ..... kernel=rbf, C=0.01, score=0.1, total= 0.0s
[CV] kernel=rbf, C=0.01 ..... kernel=rbf, C=0.01, score=0.0970873786408, total= 0.0s
[CV] kernel=rbf, C=0.01 ..... kernel=rbf, C=0.01, score=0.1, total= 0.0s
[CV] kernel=linear, C=0.1 ..... kernel=linear, C=0.1, score=0.881818181818, total= 0.0s
[CV] kernel=linear, C=0.1 ..... kernel=linear, C=0.1, score=0.922330097087, total= 0.0s
[CV] kernel=linear, C=0.1 ..... kernel=linear, C=0.1, score=0.9, total= 0.0s
[CV] kernel=rbf, C=0.1 ..... kernel=rbf, C=0.1, score=0.1, total= 0.0s
[CV] kernel=rbf, C=0.1 ..... kernel=rbf, C=0.1, score=0.0970873786408, total= 0.0s
[CV] kernel=rbf, C=0.1 ..... kernel=rbf, C=0.1, score=0.1, total= 0.0s
[Parallel(n_jobs=1)]: Done 18 out of 18 | elapsed: 1.7s finished
C:\Users\Victo\anaconda3\envs\py2\lib\site-packages\sklearn\model_selection\_search.py:842: DeprecationWarning: The default of the 'iid' parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results when test-set sizes are unequal.
  DeprecationWarning)
SVC(C=0.001, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=True, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
Confusion Matrix:
[[12  0  0  1  0  0  1  0  0  1  0  0]
 [ 0 17  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  9  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 12  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  8  0  0  0  0  4  0  0]
 [ 0  0  0  0  2 12  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 12  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 15  0  1  1  0]
 [ 0  0  0  1  1  0  0  0 13  0  0  0]
 [ 2  1  0  0  0  0  0  0  0  8  0  0]
 [ 0  0  0  1  0  0  0  0  0  0 13  1]
 [ 0  0  0  0  0  0  0  0  0  0  0 18]]
FINAL SCORE: 0.8816568047337278
Saving the model to the output file: model1.svm
DONE

```

Figura 29. Valores de entrenamiento con 480 muestras

Fuente: Elaboración Propia

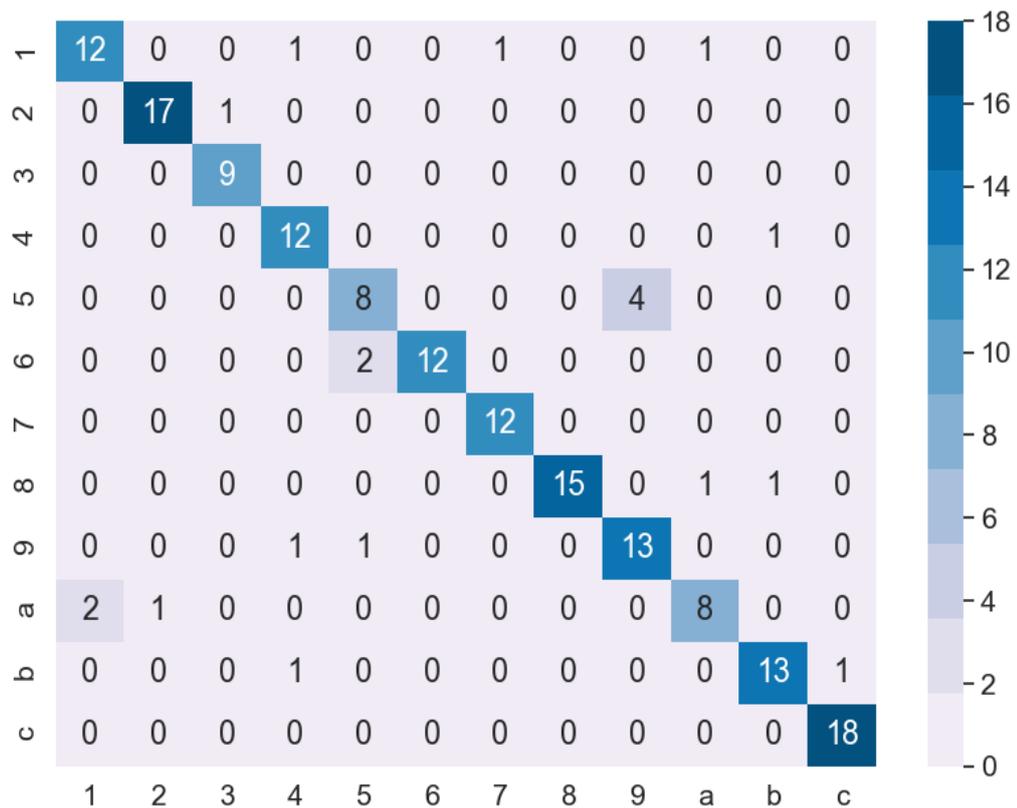


Figura 30. Matriz de confusión con 480 muestras

Fuente: Elaboración Propia.

Persona 3: Jeferson Castro Quito – Edad 26 años – DNI 48341278

Con el tercer voluntario al ser un total de 12 caracteres, se decidió probar con 80 muestras por carácter, teniendo en total 960 muestras. El tiempo empleado para la obtención de las muestras fue de 46 minutos 20 segundos. A continuación, se muestran los resultados que ejecuta el SckitLearn para los valores del clasificador SVM y la Matriz de Confusión.

Podemos observar en la figura 31, se obtiene el kernel Lineal como el mejor resultado. Al igual que el entrenamiento tiene 96% de score.

Adicionalmente Observamos en la figura 32, la matriz de confusión, da mejores resultados frente a los otros casos.

```

[CV] kernel=rbf, C=0.001 .....
C:\Users\Victo\anaconda3\envs\py2\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
"avoid this warning." FutureWarning)
[CV] ..... kernel=rbf, C=0.001, score=0.107296137339, total= 0.5s
[Parallel(n_jobs=1)]: Done 4 out of 4 | elapsed: 0.9s remaining: 0.0s
[CV] kernel=rbf, C=0.001 .....
[CV] ..... kernel=rbf, C=0.001, score=0.109170305677, total= 0.5s
[Parallel(n_jobs=1)]: Done 5 out of 5 | elapsed: 1.5s remaining: 0.0s
[CV] kernel=rbf, C=0.001 .....
[CV] ..... kernel=rbf, C=0.001, score=0.11013215859, total= 0.5s
[Parallel(n_jobs=1)]: Done 6 out of 6 | elapsed: 2.2s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.961373390558, total= 0.0s
[Parallel(n_jobs=1)]: Done 7 out of 7 | elapsed: 2.3s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.938864628821, total= 0.0s
[Parallel(n_jobs=1)]: Done 8 out of 8 | elapsed: 2.4s remaining: 0.0s
[CV] kernel=linear, C=0.01 .....
[CV] ..... kernel=linear, C=0.01, score=0.95154185022, total= 0.0s
[Parallel(n_jobs=1)]: Done 9 out of 9 | elapsed: 2.6s remaining: 0.0s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.107296137339, total= 0.4s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.109170305677, total= 0.4s
[CV] kernel=rbf, C=0.01 .....
[CV] ..... kernel=rbf, C=0.01, score=0.11013215859, total= 0.4s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.961373390558, total= 0.0s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.938864628821, total= 0.0s
[CV] kernel=linear, C=0.1 .....
[CV] ..... kernel=linear, C=0.1, score=0.95154185022, total= 0.0s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.107296137339, total= 0.4s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.109170305677, total= 0.5s
[CV] kernel=rbf, C=0.1 .....
[CV] ..... kernel=rbf, C=0.1, score=0.11013215859, total= 0.5s
[Parallel(n_jobs=1)]: Done 18 out of 18 | elapsed: 6.6s finished
SVC(C=0.001, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
kernel='linear', max_iter=-1, probability=True, random_state=None,
shrinking=True, tol=0.001, verbose=False)
confusion Matrix:
[[34  2  0  0  0  0  0  0  0  0  0  0]
 [ 0 28  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 26  0  0  0  0  0  0  0  1  0]
 [ 0  0  0 28  0  0  0  0  0  0  0  0]
 [ 3  0  0  0 34  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 37  0  0  0  1  0  1]
 [ 0  0  0  0  0  0 20  0  0  0  0  0]
 [ 0  1  1  0  0  0  0 40  0  0  0  0]
 [ 0  0  1  0  0  0  0  0 24  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 32  0  0]
 [ 0  0  0  0  1  0  0  0  0  0 18  0]
 [ 0  0  0  0  0  0  0  1  0  0  0 36]]
FINAL SCORE: 0.9596774193548387
saving the model to the output file: model2.svm
DONE

```

Figura 31. Valores de entrenamiento con 960 muestras

Fuente: Elaboración Propia


```

VALIDACION3.py
from __future__ import print_function
from pygar1.base import CallbackManager
from pygar1.classifiers import SVMClassifier
from pygar1.mocks import VerboseMiddleware
from pygar1.data_readers import SerialDataReader
from pygar1.predictors import ClassifierPredictor
from pygar1.sample_managers import DiscreteSampleManager

MODEL_PATH="C:\\Users\\Victo\\model1.svm"
PORT="COM10"

def receive_character(number):
    print(number)

# Crea el SerialDataReader
sdr = SerialDataReader(AUTONORMALIZE=False,
                      AUTOSCALE_SIZE=50,
                      GESTURES=[u'1', u'2', u'3', u'4', u'5', u'6', u'7', u'8', u'9', u'a', u'b', u'c'],
                      MIDDLEWARES=[])

# Crea el SampleManager
manager = DiscreteSampleManager()

# Administra los datos al sdr
sdr.attach_manager(manager)

# Crea el clasificador
classifier = SVMClassifier()

# carga el modelo propuesto
classifier.load()

# imprime la informacion
classifier.print_info()

# Crea el ClassifierPredictor
predictor = ClassifierPredictor()

# administra la clasificacion
manager.attach_receiver(predictor)

# Crea el CallbackManager
callback_mgr = CallbackManager(verbose=True)

# administra la callback manager
predictor.attach_callback_manager(callback_mgr)

for c in ["1", "2", "3", "4", "5", "6", "7", "8", "9", "a", "b", "c"]:
    callback_mgr.attach_callback(c, receive_character)

# abre la conexion serial
sdr.open()
print("Opened!")

# Inicia el bucle principal
sdr.mainloop()

```

Figura 33. Predicción con una tasa del 96%

Fuente: Elaboración Propia

Donde tenemos:

Verdadero Positivo: Son las que se encuentran en la diagonal, las predicciones correctas.

Verdaderos Negativos: Son los que no pertenecen a la clase diagonal.

Falsos Positivos: Son aquellas evaluaciones que se asignaron a una clase, pero no pertenecen a dicha clase. Valores verticales al Verdadero Positivo.

Falsos Negativo: Es similar al Falso Positivo, pero son los valores de manera horizontal.

Precisión: Nos ayuda a medir la calidad del modelo de Machine Learning en la tarea de clasificación.

Recall: también conocida como sensibilidad, es la proporción de casos positivos que fueron correctamente identificadas por nuestro modelo de algoritmo.

F1-Score: Se utiliza para combinar las medidas de precisión y *recall* en un solo valor. Esto nos ayuda a poder comparar el rendimiento y *recall* entre varias soluciones.

Accuracy: Es la exactitud se refiere a lo cerca que esta el resultado de una medición del valor verdadero.

Para el presente caso analizaremos nuestra matriz de confusión de la Figura 32, donde calcularemos para cada carácter los valores de *precisión*, *recall*, *f1 score* y *accuracy*

PRECISION =	VERDADERO POSITIVO					
	VERDADERO POSITIVO + FALSO POSITIVO					
RECALL =	VERDADERO POSITIVO					
	VERDADERO POSITIVO + FALSO NEGATIVO					
F1- SCORE =	(PRECISION * RECALL)	* 2				
	(PRECISION + RECALL)					
ACCURACY =	VERDADERO POSITIVO + VERDADERO NEGATIVO					
	VERDADERO POSITIVO + FALSO POSITIVO + TRUE NEGATIVO + FALSO NEGATIVO					

Figura 34. Fórmulas para el análisis de la Matriz de Confusión

Fuente: Elaboración Propia

CALCULO DE PRECISION				
CARACTERES	V. POSITIVO	V. POSITIVO	F.POSITIVO	RESULTADO
1	12	12	4	0.75
2	17	17	3	0.85
3	9	9	2	0.818181818
4	12	12	0	1
5	8	8	1	0.888888889
6	12	12	0	1
7	12	12	0	1
8	15	15	2	0.882352941
9	13	13	0	1
a	8	8	2	0.8
b	13	13	1	0.928571429
c	18	18	1	0.947368421

Tabla 4. Resultados de valor de Precisión

Fuente: Elaboración Propia

CALCULO RECALL				
CARACTERES	V. POSITIVO	V. POSITIVO	F.NEGATIVO	RESULTADO
1	12	12	2	0.857142857
2	17	17	0	1
3	9	9	1	0.9
4	12	12	0	1
5	8	8	3	0.727272727
6	12	12	2	0.857142857
7	12	12	0	1
8	15	15	2	0.882352941
9	13	13	2	0.866666667
a	8	8	0	1
b	13	13	2	0.866666667
c	18	18	1	0.947368421

Tabla 5. Resultados del cálculo del Recall

Fuente: Elaboración Propia

CALCULO F1 SCORE			
CARÁCTER	PRECISION	RECALL	RESULTADO
1	0.75	0.857142857	0.8
2	0.85	1	0.918918919
3	0.818181818	0.9	0.857142857
4	1	1	1
5	0.888888889	0.727272727	0.8
6	1	0.857142857	0.923076923
7	1	1	1
8	0.882352941	0.882352941	0.882352941
9	1	0.866666667	0.928571429
a	0.8	1	0.888888889
b	0.928571429	0.866666667	0.896551725
c	0.947368421	0.947368421	0.947368421

Tabla 6. Resultados del cálculo del F1-Score

Fuente: Elaboración Propia

CALCULO DEL ACCURACY					
CARÁCTER	V. POSITIVO	V.NEGATIVO	F. POSITIVO	F. NEGATIVO	RESULTADO
1	12	323	4	2	0.982404692
2	17	329	3	0	0.991404011
3	9	331	2	1	0.991253644
4	12	329	0	0	1
5	8	323	1	3	0.988059701
6	12	320	0	2	0.994011976
7	12	337	0	0	1
8	15	317	2	2	0.988095238
9	13	333	0	2	0.994252874
a	8	325	2	0	0.994029851
b	13	339	1	2	0.991549296
c	18	321	1	1	0.994134897

Tabla 7. Resultados del cálculo del Accuracy

Fuente: Elaboración Propia

A continuación, se muestra un resumen de los resultados del entrenamiento.

RESULTADOS DE ENTRENAMIENTO Y PREDICCIÓN			
	<i>C</i>	<i>DEGREE</i>	<i>FINAL SCORE</i>
VOLUNTARIO 1	0.001	3	0.7848
VOLUNTARIO 2	0.001	3	0.8816
VOLUNTARIO 3	0.001	3	0.9596

Tabla 8. Resultados de los entrenamientos de voluntarios

Fuente: Elaboración Propia

2.3.5 Implementación en el Hospital María Auxiliadora

Dada la coyuntura actual, se está tramitando los permisos poder realizar las pruebas en pacientes traqueostomizados en el Hospital María Auxiliadora, Véase Anexo J. Se han cruzado documentos al respecto y se está a la espera de una respuesta. Para efectos del presente proyecto, se simularon las condiciones de inmovilidad de los brazos, para que el entrenamiento sea solo realizado con un dedo. El trabajo de implementación, pasa por las siguientes etapas:

- a) Implementación en el ambiente del paciente.
- b) Designación del personal a cargo.
- c) Capacitación en el uso de tecnologías.

Los procesos explicados detalladamente a cada nivel son observados en el Diagrama de Gantt adjunto en el Anexo F.

2.4 Resultados

2.4.1 Resultado 1

Se desarrollo el proyecto seleccionando los componentes, bajo el enfoque de “inteligente” basándonos en la filosofía del internet de las cosas (bajo consumo de energía y económico). Los costos a nivel hardware que se pueden observar en la tabla 9, nos dan cuenta de lo económico del dispositivo.

Por otro lado, las pruebas no se realizaron directamente en un paciente traqueostomizados, debido a las condiciones actuales en el País. Sin embargo, se probaron en los 3 voluntarios inmovilizando totalmente el antebrazo y muñeca, colocando el dispositivo en el dedo como se observa en la Figura 36.

TABLA DE COSTO			
DESCRIPCION	Cantidad	P. UNITARIO	P. TOTAL
ESP 32	1	s/ 40.00	s/40.00
MPU6050	1	s/ 15.00	s/15.00
Cables MM 10 cm x 20 tiras	1	s/ 4.50	s/4.50
Cables HM 10 cm x 20 tiras	1	s/ 4.50	s/4.50
cables MH 10 cm x 20 tiras	1	s/ 4.50	s/4.50
Cable Flex 20 cm	2	s/3.50	s/7.00
Pulsadores	2	s/0.25	s/0.50
Leds	2	s/0.10	s/0.20
Baterias 16850	2	s/20.00	s/40.00
Placa Fibra de vidrio	1	s/25.00	s/25.00
Case Impresión 3d	1	s/40.00	s/40.00
Cable mini usb a Usb	1	s/8.00	s/8.00
Laser	1	s/5.00	s/5.00
Buzzer	1	s/1.00	s/1.00
Espadines Macho	2	s/0.50	s/1.00
Espadines Hembra	2	s/0.50	s/1.00
Borneras de 2 orificios	2	s/0.50	s/1.00
		TOTAL	s/198.20

Tabla 9. Costos del Proyecto.

Fuente: Elaboración Propia

Adicionalmente se utilizaron cables de tipo flexible, y se realizó el prototipo en placa perforada para obtener la menor tasa de error a nivel hardware. A nivel de Software, como se observa en la tabla 8, el entrenamiento arroja resultados con una precisión del 96%. En el cuadro presentado, no se especifica el guante que utilizaría el paciente, puesto que estos deben ser aprobados y recomendados por el personal médico de acuerdo al estado del paciente, en el cual se haría la adaptación del dispositivo.



Figura 35. Dispositivo que se adapta en el dedo del paciente

Fuente: Basado en el dispositivo de Federic Terzi



Figura 36. Prueba de entrenamiento con Dispositivo Dedil

Fuente: Elaboración propia

2.4.2 Resultado 2

Se realizó el plan de la implementación en el Hospital María Auxiliadora, según el Diagrama de Gantt que se puede visualizar en el Anexo F, en un periodo de 30 días, en el que incluye etapas de selección del personal para entrenamiento, etapa de implementación y las etapas de testeo con el paciente. Además de dar especial énfasis en la interpretación de la matriz de confusión, para que puedan obtener los valores más óptimos y se pueda ir calibrando el dispositivo de acuerdo a las necesidades de los pacientes.

Adicionalmente, según hemos analizado se calculó que la muestra ideal para que haya una correcta predicción es de 80 muestras por carácter, como se observa en la Figura 31. Lo cual en las pruebas tomó un tiempo de 46 minutos aproximadamente, con intervalos de trabajo de 3 minutos y de descanso de 5 minutos. Esto debido a que, al tener el antebrazo inmovilizado para simular debilidad muscular, los músculos se resentían al repetir constantemente los caracteres con el dedo. Por tal motivo, se recomienda que al momento de la adquisición de datos se trabaje por 3 minutos y el descanso sea de 5 minutos como mínimo.

2.4.3 Resultado 3

Para poder asegurar la calidad en el rendimiento del dispositivo, y las baterías de este sea cargado por lo menos 1 sola vez a la semana debemos partir de la determinación de la capacidad de la batería. Después del entrenamiento realizado, podemos tener valores reales para estos cálculos. En la Tabla 9, se especifica el consumo de los dispositivos empleados en proyecto. Se incluyen en la tabla al microcontrolador, el acelerómetro y otros dispositivos.

La corriente máxima exigida al sistema de alimentación es de 42.7 mA. Es importante tomar en cuenta que esta demanda de corriente no es permanente y es esporádica cuando el paciente presente una necesidad. El BLE, que está incluido en el ESP32, permanecerá en estado de reposo hasta sea activado con un botón y comience a enviar las tramas de datos.

DISPOSITIVO	I maxima
ESP 32	2.4 mA
MPU 6050	0.05 mA
LEDS	0.25 mA
OTROS	40 mA
TOTAL	42.7 mA

Tabla 10. Consumo Energético de los componentes

Fuente: Elaboración propia - Datasheet de Componentes

Para calcular la capacidad de las baterías o pilas que vamos a usar , estas deben estar en función al consumo de corriente de los dispositivos, y al tiempo de autonomía que se requiere, en este caso las 24 horas del día. La fórmula que relaciona estas variables es la expresan de la siguiente manera.

Donde:

C = Capacidad de la batería, dada por el fabricante

I = Corriente consumida por el circuito

T = Tiempo de autonomía

Reemplazando los datos:

$T = 24$ horas

$I = 42.7$ mA

DISPOSITIVO	VOLTAJE DE OPERACIÓN	VOLTAJE SELECCIONADO
ESP32	3.1 - 3.3 V	3.3 V
MPU6050	3.3 - 5.0 V	3.3 V
LEDS	1.5 A 3.3 V	1.5 V

Tabla 11. Rango de Voltajes de Operación

Fuente: Elaboración Propia – Datasheet de Componentes

Se obtiene:

$C = 42.7$ mA x 24 horas

$C = 1024.8$ mAh

Por tanto, la capacidad mínima de la batería debe ser de unos 1000 mAh. Dentro de las baterías disponibles en el mercado existen modelos de distintas marcas con esa capacidad y con voltaje de salida de 3.3 V.

Con la batería de 1000 mAh seleccionada, el tiempo de autonomía, para un consumo de corriente de 42.82 mA es de:

$T = C / I = 1000$ mAh / 42.7 mA

$T = 23.4$ horas \approx 1 día

Para el presente proyecto se está usando un cargador portátil de 10000mAh, por lo cual podemos asegurar en promedio 10 días de uso continuo en teoría.

En pruebas reales como se observa en la Figura 37, se utilizó un cronometro y un multímetro para medir el consumo en el tiempo que se estuvo usando el dispositivo. En total en los 3 voluntarios, el tiempo total de entrenamiento fue de 98 minutos aproximadamente y se consumió un total de 2.8 mA x h. De tal forma que aplicando una regla de tres simple tendríamos que en un día consumiría alrededor de 41.14 mAh.

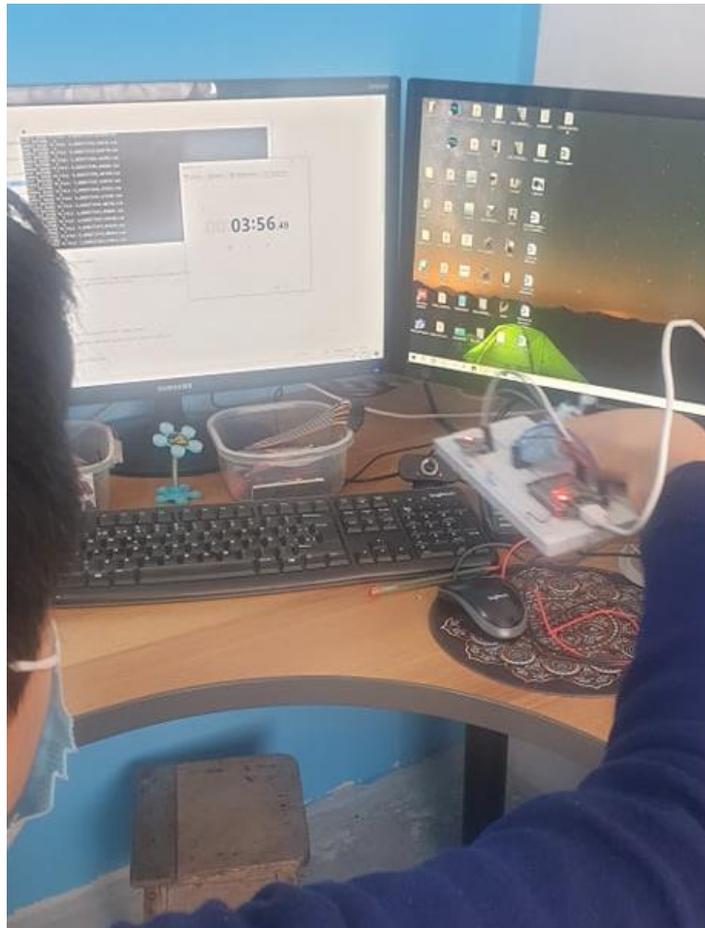


Figura 37. Prueba de entrenamiento del Dispositivo.

Fuente: Elaboración Propia

2.4.4 Resultado 4

En las pruebas realizadas, como se explicó anteriormente, se obtuvo el valor de precisión del 96% en el conjunto de la Matriz de Confusión de 80 muestras por caracteres, en comparación con las otras muestras tomadas

como se puede apreciar en el cuadro de resumen Tabla 8. En el análisis de la Matriz de Confusión para esta predicción del 96%, podemos observar los cálculos de precisión para cada carácter (Tabla 4). Obteniendo una baja precisión para el carácter "1", debido a que este carácter está causando falsos positivos con el carácter "5" y precisiones altas para los caracteres "4", "6", "7", "9".

Por otro lado, en los cálculos del *Recall* (Tabla 5) son aceptables para cada uno de los caracteres, tienen alto grado de relación de casos correctamente clasificados. En los cálculos del F1-Score (Tabla 6), que nos muestra en una sola métrica los resultados de precisión y *Recall*, nos muestran los altos grados de asertividad para cada carácter empleado, siendo la mayoría de ellos superior al 88%. La exactitud, que nos hace referencia a lo cerca que está el resultado de una medición real, para cada uno de los caracteres son de precisión muy alta, lo cual concluye que el modelo realizado es correcto y funcional.

Para evitar falsos positivos, y colocar al dispositivo en estado de emergencia, al igual de que sirva al paciente en casos de necesitar auxilio inmediato, se decidió colocar el *Láser* y el *Buzzer*, para que puedan ser activados mediante un botón de pánico. Esto de tal modo que se, utilizara el hardware para los estados de emergencia y el software netamente para las predicciones. La decisión de usar los dispositivos de emergencia colocados directamente en el dispositivo y activados mecánicamente, obedece a las recomendaciones médicas dadas por personal del Hospital María Auxiliadora, en el cual indican que todo dispositivo medico debe tener un botón de parada de emergencia o debe existir el mismo, para que el paciente pueda alertar de algún problema de manera inmediata. Véase Anexo K

CONCLUSIONES

El dispositivo creado, en combinación con los componentes de hardware y software que se muestran en este proyecto, demostró ser capaz de gestionar un conjunto de gestos y, en consecuencia, reaccionar con alta precisión a los gestos de los pacientes (voluntarios). Tabla 8.

Por otro lado, las acciones que se pueden realizar en reacción a los gestos son personalizables de acuerdo con los requerimientos exigidos por el personal a cargo o del paciente. De este modo, no solamente será aplicable para pacientes traqueostomizados sino también para cualquiera que presente problemas para la fonación del habla. Como se puede observar en el código de la Figura 33.

Se observó que aún se presenta confusión, en un bajo nivel, en la predicción de algunos gestos, y la solución no es alimentar de más datos al algoritmo sino por el contrario de lo ya estimado como mínimo de 80 muestras por carácter, diferenciar bien los caracteres que tienen similitud en el entendimiento de escritura del paciente en el aire. Por ejemplo, algunos escriben el "1" como una línea vertical y otros con una línea jalada, lo cual podría ocasionar confusión con otros caracteres. Tabla 4-5-6-7.

La fabricación, implementación y de puesta en marcha del proyecto, es viable de acuerdo a nuestra tabla de costos y a nuestro diagrama de Gantt (Anexo F). En la medida del proyecto, que surgió como una necesidad ante la falta de dispositivos para pacientes traqueostomizados, se cumple con la idea primordial que es mejorar la calidad de vida y la comunicación en los pacientes.

RECOMENDACIONES

Para poder hacer más entendible los procesos de entrenamiento para el personal médico, se recomienda desarrollar una interfaz gráfica, que mediante comandos sencillos pueda obtener los datos necesarios de los pacientes. A la vez que el personal de monitoreo, pueda tener una reacción más eficaz ante el llamado de los pacientes.

Para el presente proyecto, solo se realizó el modelo propuesto en el dedo de los pacientes, sin embargo, sería de utilidad poder realizar los entrenamientos en otras partes del cuerpo, ya que muchos pacientes al salir de UCI, presentan debilidad muscular en los brazos, pero no en las piernas.

En el proyecto se utilizaron las máquinas de soporte vectorial, ya que en las investigaciones se concluyó que es el método más utilizado, sin embargo, existen otras técnicas que combinan distintos tipos de entrenamiento y podrían generar mejores predicciones, por ello es recomendable utilizar una validación cruzada con otros métodos.

Para poder masificar el proyecto, es recomendable realizar las placas del dispositivo en doble capa, la batería sea de tipo LiPo y con componentes tipo SMD lo cual hará que el dispositivo sea de menor tamaño, de menor peso y más cómodo para los pacientes.

BIBLIOGRAFÍA

- 10 Librerías Python para Data Science y Machine Learning - SolidQ Blogs. (n.d.). Retrieved October 23, 2020, from <https://blogs.solidq.com/es/poder-del-dato/10-librerias-python-para-data-science-y-machine-learning/>
- Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., ... Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 92–93. <https://doi.org/10.3389/fninf.2014.00014>
- Alfaro, E. A. C. (2010). *Máquinas de soporte vectorial con algoritmos basados en poblaciones para el pronóstico del precio de acciones LAN Chile*. PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO. Retrieved from http://opac.pucv.cl/pucv_txt/txt-4500/UCG4828_01.pdf
- Amar Rodrigo, J. (2020, December 9). Máquinas de Vector Soporte (Support Vector Machines, SVMs). Retrieved December 9, 2020, from https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines
- Ángel Cañete, Roberto Pawlowicz, Haydee Zotarez, Mario Gerónimo, L., & Alberto. (2012). *Revista médica. Sagrado Corazon Sanatorio de Alta Complejidad*. 1(1), 12–13. Retrieved from http://intranet.sagrado-corazon.com.ar:1102/intranet/revistas/rm-sc_editorial_001.pdf
- Bernal Iñiguez, J. D. (2014). Diseño, construcción e implementación de un sistema de captura de movimiento para análisis ergonómico de riesgo laboral de extremidades superiores. *Universidad Politécnica Salesiana*, 1–99. Retrieved from <https://dspace.ups.edu.ec/bitstream/123456789/7508/1/UPS-CT004422.pdf>
- Bruno, A. (2019). *Esp32 node mcu*. 7.
- Chandra, M. M., Rajkumar, S., & Kumar, L. S. (2019). Sign Languages to Speech Conversion Prototype using the SVM Classifier. *IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2019-October*, 1803–1807. <https://doi.org/10.1109/TENCON.2019.8929356>
- Chris Edwards. (n.d.). Addressing portable medical device needs - Tech Design Forum Techniques. Retrieved October 31, 2020, from

<https://www.techdesignforums.com/practice/technique/addressing-portable-medical-device-needs/>

Covid 19 en el Perú - Ministerio del Salud. (n.d.). Retrieved November 30, 2020, from https://covid19.minsa.gob.pe/sala_situacional.asp

Datos, U. de P. de. (2018). Compendio Estadístico Hospital María Auxiliadora 2018. *Unidad de Procesamientos de Datos -Hosp.María Auxiliadora*, 154. Retrieved from <http://www.hma.gob.pe/pdf/publicaciones/29.pdf>

Díaz, E. O. (2014). *Diseño de un sistema inalámbrico de detección de caídas aplicado a personas de la tercera edad basado en acelerómetro y teléfono móvil* (PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU ESCUELA DE POSGRADO). PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU ESCUELA DE POSGRADO. Retrieved from <http://hdl.handle.net/20.500.12404/5914>

JohnChi, A. U. (n.d.). Arduino Playground - MPU-6050. Retrieved November 1, 2020, from <https://playground.arduino.cc/Main/MPU-6050/>

Martínez-Téllez, E., Orús Dotú, C., Trujillo-Reyes, J. C., Guarino, M., Cladellas Gutiérrez, E., Planas Cánovas, G., ... Belda-Sanchis, J. (2020). Traqueotomía en pacientes COVID-19: un procedimiento necesario de alto riesgo. Experiencia de dos centros. *Archivos de Bronconeumología*, 1–4. <https://doi.org/10.1016/j.arbres.2020.05.018>

NQA. (n.d.). What Standards Apply to Medical Devices Manufacturing? | NQA. Retrieved October 31, 2020, from <https://www.nqa.com/en-us/certification/sectors/medical-devices-services>

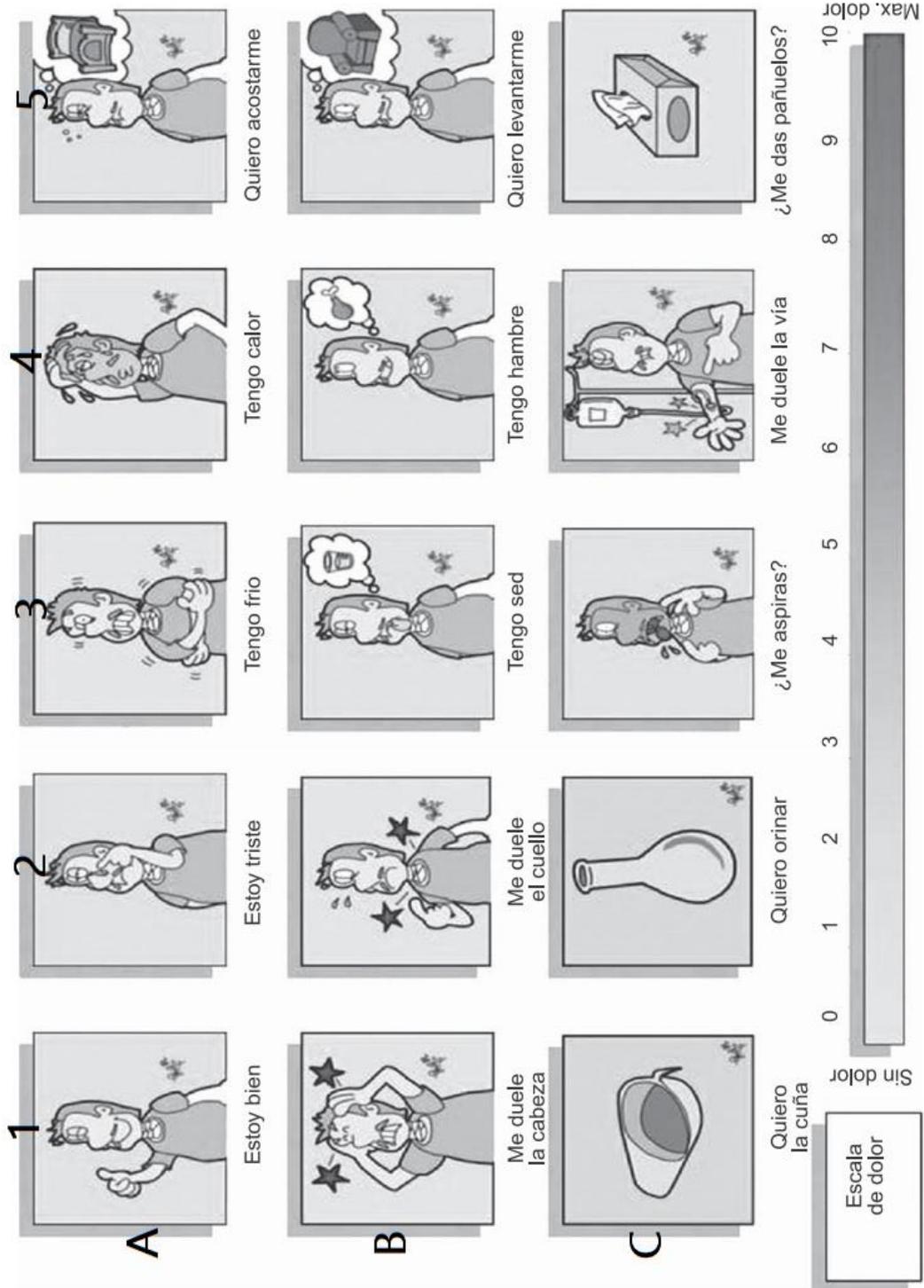
Orduña Beuzón, A. J., Vicente Bardón, J. M., Martínez Gago, A. L., & Plaza Mayor, G. (2009). Communication of needs in laryngectomized patients: Pictorrino©. *Acta Otorrinolaringologica (English Edition)*. [https://doi.org/10.1016/s2173-5735\(09\)70156-5](https://doi.org/10.1016/s2173-5735(09)70156-5)

Perelló-Campaner, C. (2015). Objetivos Paciente crítico intubado. *Universitat de Les Illes Balears*, 1–22. Retrieved from https://riuma.uma.es/xmlui/bitstream/handle/10630/9677/comunicacion_pacientes_intubados.pdf?sequence=1&isAllowed=y

- Pinazo Mendoza, G., & Rodriguez Yauri, M. (2012). *Complicaciones y Calidad de vida de los pacientes traqueostomizados en los servicios de Hospitalización y Emergencia del H . B . C . A . S . E . , Arequipa – Perú*. Univesidad Catolica Santa Maria.
- Rappoport W, D., Gonzalez A., T., Capdeville F, F., Valdés G, F., Rojas P, H., Zanolli De S, L., & Madrid M, A. (2020). Traqueostomía en pacientes con COVID-19: Recomendaciones actuales. *Revista de Cirugía*, 72(5), 449–454. <https://doi.org/10.35687/s2452-45492020005870>
- Salcedo, O., & Frutos-Vivar, F. (2008). Tracheostomy in ventilated patients. What do we do it for? | Traqueostomía en pacientes ventilados: ¿para qué la hacemos? *Medicina Intensiva*. [https://doi.org/10.1016/S0210-5691\(08\)70913-6](https://doi.org/10.1016/S0210-5691(08)70913-6)
- Saravanan, S., Govindrajan, S., Bansal, H., & Singh, A. (2018). *Character Recognition through Gestures Using SVM Algorithm*. 13(9), 7233–7238.
- Scikit-Learn, herramienta básica para el Data Science en Python. (n.d.). Retrieved October 23, 2020, from <https://www.master-data-scientist.com/scikit-learn-data-science/>
- Sideridis, V., Zacharakis, A., Tzagkarakis, G., & Papadopouli, M. (2019). Gesturekeeper: Gesture recognition for controlling devices in IoT environments. *European Signal Processing Conference, 2019-Septe*, 6. <https://doi.org/10.23919/EUSIPCO.2019.8903044>
- Terzi, F. (n.d.). GitHub - federico-terzi/gesture-keyboard: A library to convert accelerometer data into characters using machine learning. Retrieved November 18, 2020, from <https://github.com/federico-terzi/gesture-keyboard>
- Terzi Federico. (n.d.). pygarl · PyPI. Retrieved November 21, 2020, from <https://pypi.org/project/pygarl/>
- Tincopa Flores, J. P. (2019). *Design and evaluation of a diagnostic support system for diabetic neuropathy based on the reading of plantar pressure points and machine learning* (Universidad Peruana Cayetano Heredia). Universidad Peruana Cayetano Heredia. Retrieved from <http://190.116.48.43/handle/upch/7239>

ANEXOS

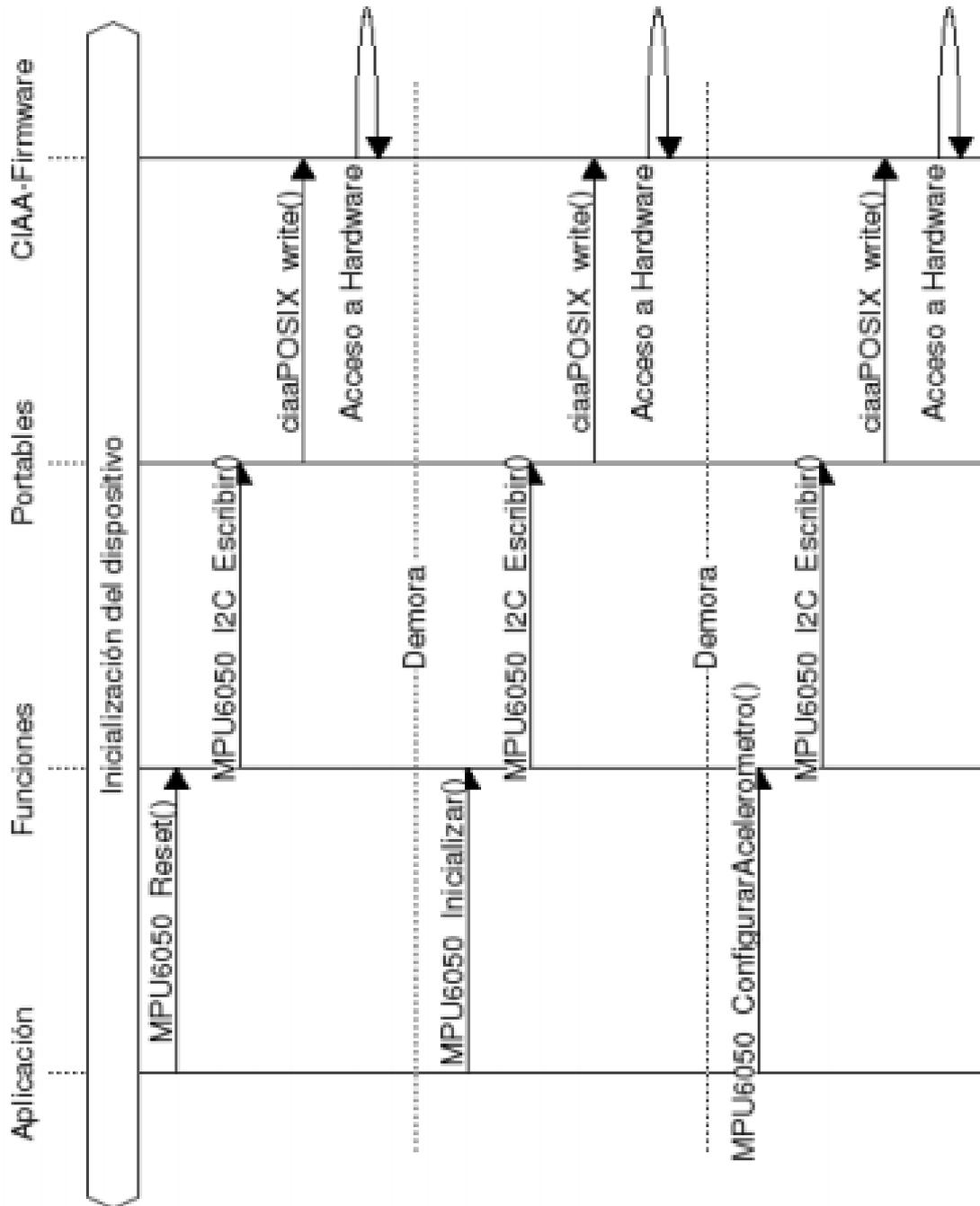
ANEXO A. MATRIZ DE GESTOS ELABORADO EN BASE AL PICTORRINO



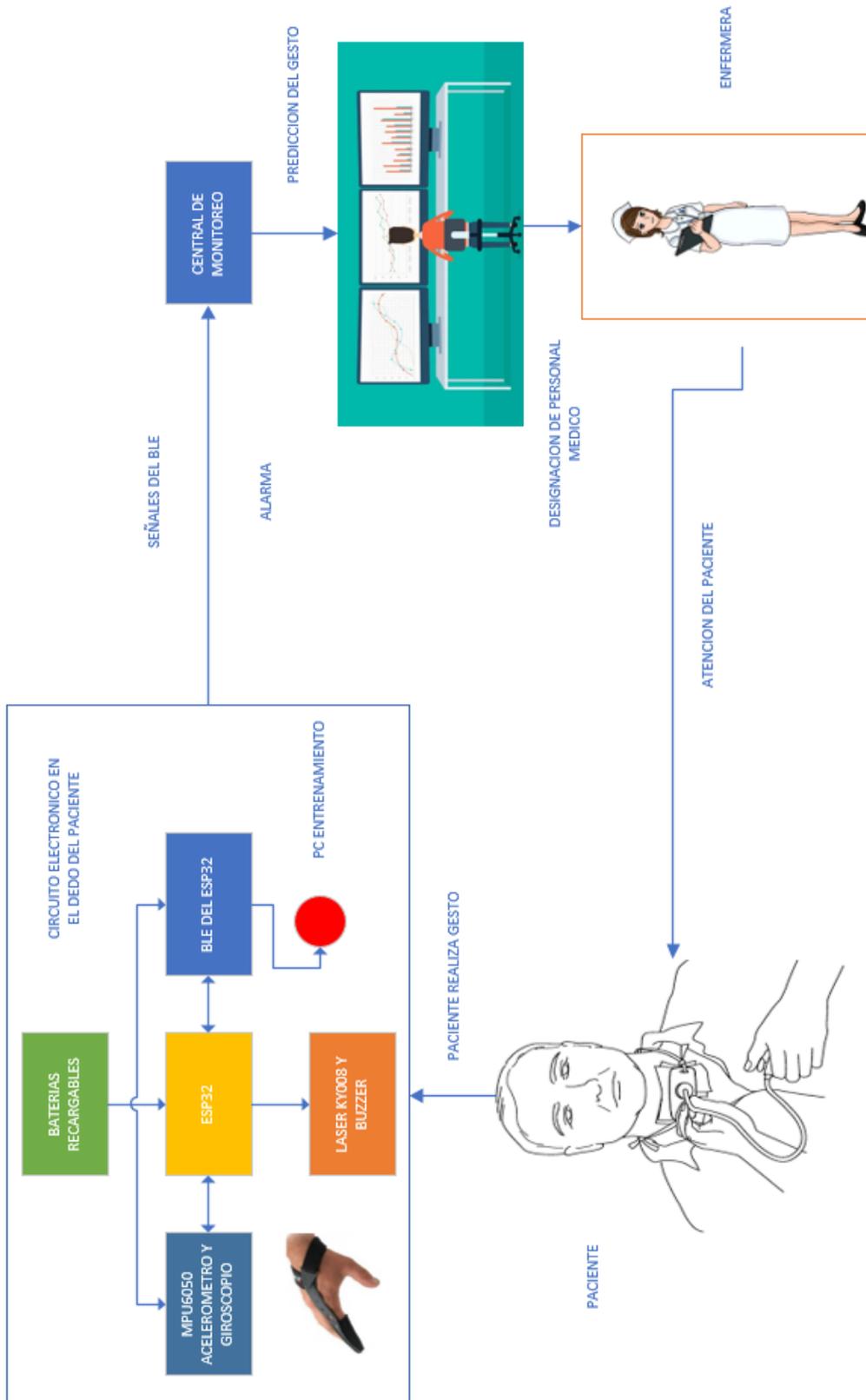
ANEXO B. CARACTERISTICAS TECNICAS DEL ESP 32 WROOM

ESPECIFICACIONES TECNICAS
Voltaje de Alimentación: 3.3V DC (2.7~ 3.6V)
Voltaje lógico entradas/salidas (GPIO): 3.3V
Corriente de Operación: ~80mA (fuente superior a 500mA)
SoM: ESP-WROOM-32
SoC: ESP32 (ESP32-D0WDQ6)
CPU: Dual core Tensilica Xtensa LX6 (32 bit)
Frecuencia de Reloj: 240MHz
SRAM: 520KB
Memoria Flash Externa: 4MB
Pines Digitales GPIO: 34 (incluyendo todos los periféricos)
UART: 2
SPI: 3
I2C: 2
Capacitive touch sensors: 10
Interfaz SD
Timers: 3 (16-bit)
PWM Led: 16 canales independientes (16-bits)
ADC: 2 (12-bit)
DAC: 2 (8-bit)
Wi-Fi, Protocolo 802.11 b/g/n/e/i (802.11n up to 150 Mbps)
Wi-Fi, certificación RF: FCC/CE/IC/TELEC/KCC/SRRC/NCC
Wi-Fi, rango de Frecuencia: 2.4 ~ 2.5 GHz
Wi-Fi mode Station/SoftAP/SoftAP+Station/P2P
Wi-Fi Security WPA/WPA2/WPA2-Enterprise/WPS
Network protocols IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
Bluetooth, Protocolos: V4.2 BR/EDR and BLE specification
Bluetooth, Radios: NZIF receiver with -97 dBm sensitivity, Class-1, class-2 and class-3 transmitter, AFH
Bluetooth, Audio: CVSD and SBC
Stack de Protocolo TCP/IP integrado
Dimensiones: 18*25.5*3.1 mm
Peso: 3 gramos

ANEXO C. SECUENCIAS Y FUNCIONAMIENTO DEL SENSOR MPU6050



ANEXO D. DIAGRAMA GENERAL DEL FUNCIONAMIENTO DEL DISPOSITIVO DEDIL



ANEXO E. CODIGO DEL HARDWARE DE ADQUISICION DE DATOS

```
#include <Wire.h>

// Pins GPIO
int btnPin1 = 15;

// I2C MPU-6050
const int MPU_addr=0x68; // cambiar x68 ante cualquier problema
// variables del sensor
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

// estado del boton
int precBtn1 = HIGH;

void setup(){
  // boton en pullup
  pinMode(btnPin1, INPUT);

  // Comienza la comunicacion con el sensor
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // registro a 0 (enciende the MPU-6050)
  Wire.endTransmission(true);

  // comienza la comunicacion serial
  Serial.begin(38400);
}

void loop(){
  // lee el estado del boton
  int resultBtn1 = digitalRead(btnPin1);

  // cuando el boton 1 esta presionado
  if (precBtn1 == HIGH && resultBtn1 == LOW)
  {
    startBatch();
  }

  // si el boton 1 esta presionado, lee la data y la envia
  if (resultBtn1==LOW)
  {
    // comienza la transmision del mpu6050
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // comienza con el resgistro 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true); //
```

```

Wire.write(0x3B); // comienza con el resgistro 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,14,true); //

// lee la data del sensor
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

Serial.print("START");
Serial.print(" "); Serial.print(AcX);
Serial.print(" "); Serial.print(AcY);
Serial.print(" "); Serial.print(AcZ);
Serial.print(" "); Serial.print(GyX);
Serial.print(" "); Serial.print(GyY);
Serial.print(" "); Serial.print(GyZ);
Serial.println(" END");
}

// cierra el ciclo cuando el boton deja de estar presionado
if (precBtn1 == LOW && resultBtn1 == HIGH)
{
  closeBatch();
}

// guarda el estado del boton
precBtn1 = resultBtn1;
}

// envia la data cuando esta presionado el boton
void startBatch()
{
  Serial.println("STARTING BATCH");
}

// cierra la data cuando el boton deja de estar presionado
void closeBatch()
{
  Serial.println("CLOSING BATCH");
}

```

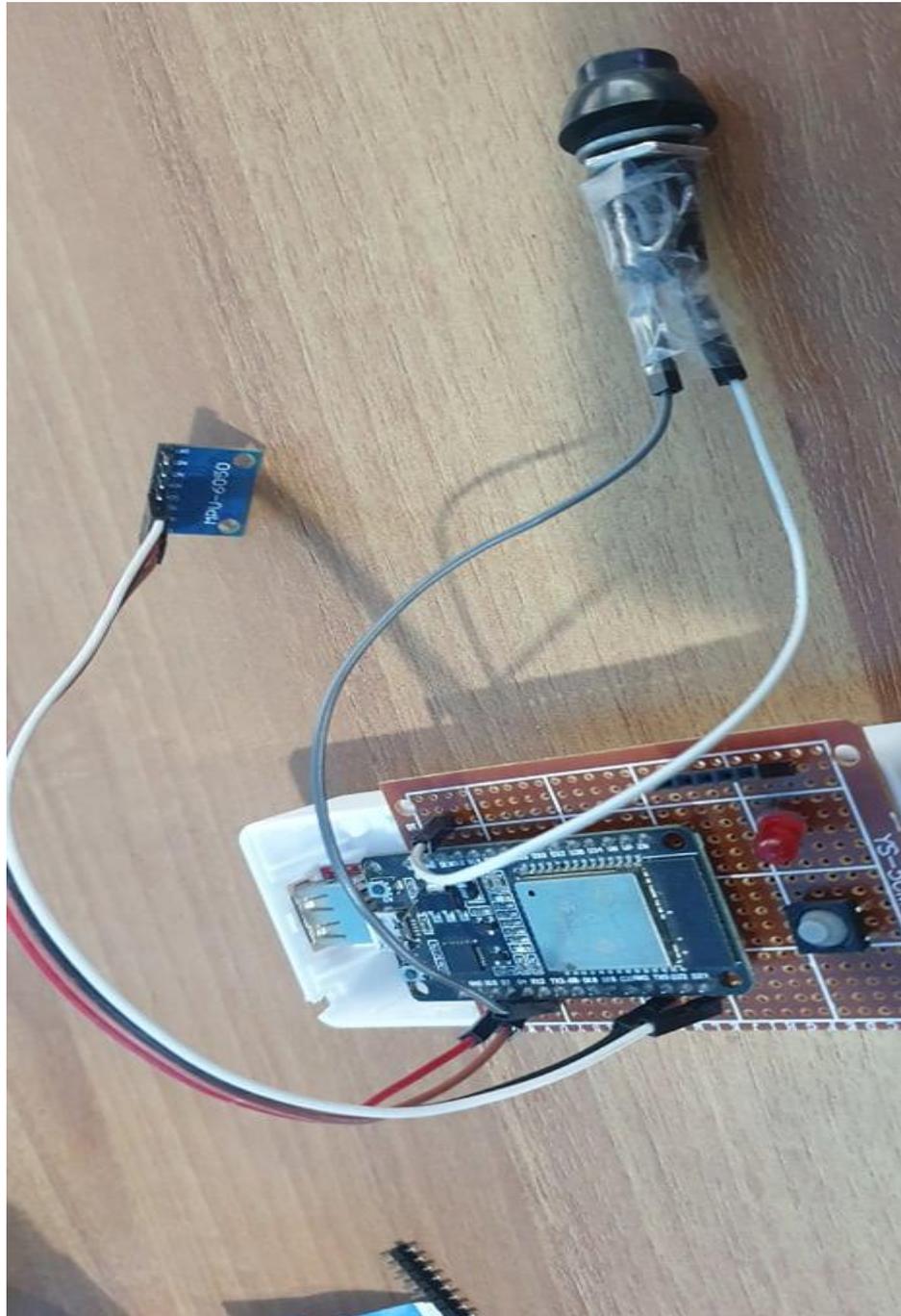

ANEXO G. CODIGO MODIFICADO DE LA LIBRERÍA PYGARL PARA NUESTRO DISPOSITIVO DEDIL

```
codigomodificado_tesis.py  ardu.cpp  predictors.py  data_readers.py
1  from __future__ import print_function
2  from pygarl.base import CallbackManager
3  from pygarl.classifiers import SVMClassifier
4  from pygarl.mocks import VerboseMiddleware
5  from pygarl.data_readers import SerialDataReader
6  from pygarl.predictors import ClassifierPredictor
7  from pygarl.sample_managers import DiscreteSampleManager
8
9  MODEL_PATH="C:\\Users\\Victo\\model.svm"
10 PORT="COM8"
11
12 def receive_character(number):
13     print(number)
14
15
16 # Crea el SerialDataReader
17 sdr = SerialDataReader(PORT, expected_axis=6, verbose=False)
18
19 # Crea el SampleManager
20 manager = DiscreteSampleManager()
21
22 # Administra los datos al manager
23 sdr.attach_manager(manager)
24
25 # Crea el clasificador
26 classifier = SVMClassifier(model_path=MODEL_PATH)
27
28 # carga el modelo propuesto
29 classifier.load()
30
31 # imprime la informacion del clasificador
32 classifier.print_info()
33
34 # Crea el ClassifierPredictor
35 predictor = ClassifierPredictor(classifier)
36
37 # administra la classifier predictor
38 manager.attach_receiver(predictor)
39
40 # Crea el CallbackManager
41 callback_mgr = CallbackManager(verbose=True)
42
43 # administra la callback manager
44 predictor.attach_callback_manager(callback_mgr)
45
46
47 for c in ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "e",
48 "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]:
49     callback_mgr.attach_callback(c, receive_character)
50
51 # abre la conexion serial
52 sdr.open()
53 print("Opened!")
54
55 # Inicia el bucle principal
56 sdr.mainloop()
```

ANEXO H. PRUEBAS INICIALES DEL DISPOSITIVO DEDIL EN DEDO DEL VOLUNTARIO



ANEXO I. PROTOTIPO DEL DISPOSITIVO DEDIL INTELIGENTE



ANEXO J. SOLICITUD ENVIADA AL HOSPITAL MARIA AUXILIADORA PARA TESTEO DEL PROYECTO



Solicito: Permiso y facilidades técnicas para testear dispositivo dedil inteligente para pacientes traqueostomizados.

Dra. Susana J. Oshiro Kanashiro de Otta
Directora de la IPRESS Hospital María Auxiliadora

Presente:

El que suscribe, Victor Alexander Quito Huaycañe con DNI N° 45906731, con código universitario 2011100065, en mi condición de bachiller de la Universidad Nacional Tecnológica de Lima Sur, y como parte del desarrollo de mis habilidades profesionales, he desarrollado el Trabajo de Suficiencia Profesional que lleva por título "PROPUESTA DE DISEÑO DE UN DISPOSITIVO DEDIL INTELIGENTE, PARA MEJORAR LA COMUNICACION EN PACIENTES TRAQUEOSTOMIZADOS POST-UCI EN LA IPRESS HOSPITAL MARIA AUXILIADORA EN EL DISTRITO DE SAN JUAN DE MIRAFLORES", para optar el Título Profesional de Ingeniero Electrónico y de Telecomunicaciones, por tal motivo solicito ante su distinguida persona lo siguiente:

1.- Las facilidades para poder capacitar al personal médico, enfermeras y personal de TI, que ustedes seleccionen, en el uso del dispositivo dedil inteligente, según el cronograma adjunto en el Diagrama de Gantt, las cuales pueden ser adaptadas a las disponibilidades dadas. Las capacitaciones al personal no tendrán una duración mayor a 2 horas.

2.- Las facilidades para poder testear la propuesta de Dispositivo Dedil Inteligente en algún paciente traqueostomizados o con dificultades del habla. Se necesitará un tiempo no mayor de 1 hora con el paciente para obtener los datos del sensor colocado en el dedo, estos datos pueden ser recolectados por el personal capacitado previamente. Posteriormente para las etapas de entrenamiento y testeo, se estará coordinando con el personal médico a cargo.

3.- Las facilidades para la implementación de un espacio, según el diagrama de planificación adjunto en el Trabajo de Suficiencia Profesional Anexo D. Donde se requiere un ambiente con cama para el paciente, una toma de corriente eléctrica, un espacio en la pared para el diagrama pictórico. Posteriormente en otro ambiente, un espacio para una PC que este en constante funcionamiento durante las 24 horas del día.

Agradezco se me notifique digitalmente la respuesta ante esta solicitud al correo victorquitoh@outlook.com o al celular 922324118.

Me despido, agradeciendo su apoyo incondicional en la etapa de tratamiento de mi padre y su trabajo extraordinario en aras de la salud de la población de Lima Sur.

Villa el Salvador, 27 de noviembre del 2020

Atentamente.

Victor Alexander Quito Huaycañe

DNI 45906731



ANEXO K. CARTA DE RECOMENDACIÓN DE MEDICO CON RESPECTO AL SISTEMA DE EMERGENCIA DEL DISPOSITIVO DEDIL

CARTA DE RECOMENDACIONES MEDICAS

Lima 04 de Diciembre 2020

Dr. Raul Edgar Montes Quispe
Medico Cardiólogo Hospital María Auxiliadora
CMP 037905 RNE 030598

El que suscribe Medico Cardiólogo con más de 10 años de trayectoria, me siento en la necesidad de realizar unas recomendaciones al proyecto titulado "PROPUESTA DE DISEÑO DE UN DISPOSITIVO DEDIL INTELIGENTE, PARA MEJORAR LA COMUNICACION EN PACIENTES TRAQUEOSTOMIZADOS POST-UCI EN LA IPRESS HOSPITAL MARIA AUXILIADORA EN EL DISTRITO DE SAN JUAN DE MIRAFLORES", por consiguiente:

1. El Dispositivo debe estar contenido dentro de una caja que pueda ser esterilizada constantemente, sin dañar los circuitos del mismo.
2. El peso del Dispositivo debe ser modificados, ya que para los fines de debilidad muscular se recomienda pesos menores a 200gr. En el mejor de los casos separar el sensor dedil, de la caja de circuitos.
3. Integrar al dispositivo alguna pantalla, donde se muestre el porcentaje de batería y los caracteres que el paciente está invocando.
4. Integrar el botón de parada de emergencia o auxilio, separado de todo el sistema de software, y que sea activado mecánicamente.

Me despido ante usted, esperando tenga en cuenta las recomendaciones dadas.

Atte.



Dr. Montes Quispe Raul Edgar
CMP: 037905 RNE: 030598
CARDIOLOGIA